

Universidad de las Ciencias Informáticas

Facultad 3



Módulo de Gestión de Expedientes genéricos del XABAL eXcriba 4.0

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

Autor: Annet Herrero González

Tutor: Msc. Aurelio Antelo Collado.

Ing. Liomar Rodríguez Guerra.

La Habana, 2020



*“La medida de lo que somos es lo que hacemos con lo que tenemos”*

*Vince Lombardi.*

## Declaración de autoría

---

Declaro ser la autora de la presente tesis que tiene por título: “Modulo de Gestión de Expedientes genéricos para del XABAL eXcriba 4.0.” y se reconoce a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

---

Annet Herrero González

Autora

---

MsC. Aurelio Antelo Collado

Tutor

---

Lic. Liomar Rodríguez Guerra

Tutor

## Agradecimientos

---

A mis padres por siempre estar ahí y apoyarme en todo momento, por nunca dejar que me rindiera incluso en los momentos más difíciles, por cada consejo y cada palabra de aliento, por creer en mí incluso cuando yo misma no lo hacía. Gracias a uds soy la mujer de hoy y mis palabras y mis acciones jamás serán suficientes para demostrarles lo mucho que los quiero y el orgullo enorme que siento de que sean mis padres.

A mis tutores y mi tribunal por toda la ayuda y el apoyo brindado durante este último año de mi carrera.

A todos mis profesores en el transcurso de estos 5 años por aportar cada uno su granito de arena en mi formación como ingeniera.

A mis compañeros de universidad a todos los que estuvieron y a todos los que están, sobre todo a esos que se mantuvieron a mi lado durante estos 5 años y logramos crecer y conseguir nuestro sueño juntos.

A mis amigos fuera de la uci y mis familiares, tíos, primos y mi hermana que siempre estuvieron pendientes de mí y me dieron ánimos.

Y a mi universidad, mi uci, por darme los 5 mejores años de mi vida y ayudarme a desarrollarme tanto personal como profesionalmente.

## Dedicatoria

---

Por encima de todo a mis padres por ser mi ejemplo a seguir y lo más importante de mi vida.

Y a mí porque a pesar que existieron muchos momentos malos nunca desistí del sueño de convertirme en ingeniera.

## Resumen

---

En los últimos años, el uso de las llamadas Tecnologías de la Información y las Comunicaciones se ha incrementado, brindando grandes beneficios y adelantos en todas las esferas. Actualmente la información es la unidad más importante con que cuenta una empresa, por lo que el crecimiento de la misma ha generado la creación de sistemas capaces de salvaguardar la información en formato electrónico.

El presente trabajo evidencia el proceso de implementación del módulo de gestión de expedientes genéricos en el producto XABAL eXcriba 4.0 que permita a las empresas e instituciones mantener un control y seguimiento de sus expedientes electrónicos.

Este contribuye a la seguridad, consistencia y disponibilidad de la información de forma segura y ordenada en formato electrónico. La solución favorece a la automatización del proceso de estudio de la actividad científica en el contexto universitario.

El desarrollo de la solución estuvo guiado por la metodología de desarrollo Proceso Unificado Ágil en la versión de la Universidad de Ciencias Informáticas, se utiliza el lenguaje de programación Java Script 1.6 y HTML5 y como repositorio de información el gestor de contenido empresarial Alfresco 5.2, fue utilizada la herramienta Visual Paradigm v8.0 para el modelado y diseño del sistema, y la validación mediante diferentes pruebas de software.

**Palabras clave:** gestión de expedientes, gestión documental.

## **Abstract**

In recent years, the use of so-called Information and Communication Technologies has increased, providing great benefits and advances in all spheres. Currently, information is the most important unit that a company has, so its growth has generated the creation of systems capable of safeguarding information in electronic format.

This work shows the implementation process of the generic file management module for the GDA eXcriba, it is developed with the aim of facilitating the monitoring and control of the files. This contributes to the security, consistency and availability of information in a safe and orderly way in electronic format. The solution favors the automation of the study process of scientific activity in the university context.

The development of the solution was guided by the Agile Unified Process development methodology in the version of the University of Informatics Sciences, the Java Script 1.6 and HTML5 programming language is used and the business content manager Alfresco 5.2 was used as a repository of information. used the Visual Paradigm v8.0 tool for modeling and design of the system, and validation through different software tests.

**Keywords:** file management, document management.

# Índice de contenidos

---

<b>Declaración de autoría .....</b>	<b>2</b>
<b>Agradecimientos.....</b>	<b>3</b>
<b>Dedicatoria .....</b>	<b>4</b>
<b>Resumen.....</b>	<b>5</b>
<b>Índice de contenidos .....</b>	<b>7</b>
<b>Introducción .....</b>	<b>9</b>
<b>Capítulo 1: Fundamentación teórica .....</b>	<b>14</b>
1.1    Conceptos fundamentales .....	14
1.1.1    Gestión documental (GD): .....	14
1.1.2    Sistemas de Gestión Documental (SGD): .....	15
1.1.3    eXcriba.....	15
1.1.4    Expediente electrónico: .....	16
1.1.5    Sistema de gestión de expedientes electrónicos. ....	17
1.2    Análisis de sistemas homólogos. ....	18
1.2.1    abox ecm .....	18
1.2.2    Archive.....	19
1.2.3    Gedson. ....	20
1.2.4    Q-expeditive.....	21
1.2.5    GETET:.....	22
1.3    Valoración del estudio realizado.....	23
1.4    Metodología de Desarrollo.....	23
1.5    Lenguajes de programación, herramientas y tecnologías.....	24
1.5.1    Lenguajes de programación. ....	25
1.5.2    Herramientas .....	26
1.5.3    Tecnologías informáticas.....	28
1.6    Conclusiones del capítulo.....	29
<b>Capítulo 2: Análisis, diseño e implementación de la propuesta de solución .....</b>	<b>30</b>
2.1    Propuesta de solución. ....	30
2.2    Modelo conceptual.....	31
2.2.1    Diagrama del modelo del dominio .....	31
2.3    Requisitos de software .....	32
2.3.1    Técnicas utilizadas para la obtención de requisitos .....	32
2.3.2    Requisitos funcionales.....	32
2.3.3    Requisitos no funcionales.....	34
2.4    Casos de uso del sistema.....	35
2.4.1    Definición de los actores.....	35
2.4.2    Definición de casos de uso del sistema .....	35
2.4.3    Descripción de casos de uso del sistema .....	36
2.5    Arquitectura del sistema .....	37
2.5.1    Arquitectura en capas.....	37
2.6    Patrones de diseño de software .....	39



2.6.1	Patrones GRASP .....	39
2.6.2	Patrones GoF.....	41
2.7	Validación de la propuesta de solución a través de métricas del diseño.....	42
2.7.1	Tamaño operacional de clase (TOC) .....	43
2.7.2	Relaciones entre clases (RC).....	43
2.8	Resultado de las métricas.....	44
2.9	Conclusiones del capítulo.....	46
<b>Capítulo 3:</b>	<b>Implementación y validación de la solución.....</b>	<b>47</b>
3.1	Modelo de implementación.....	47
3.2	Diagrama de componentes.....	47
3.3	Modelo de Despliegue.....	48
3.3.1	Métodos para el despliegue de extensiones de Alfresco.....	49
3.3.2	Diagrama de despliegue.....	51
3.4	Estándares de codificación.....	52
3.4.1	Nomenclatura.....	52
3.4.2	Indentación y líneas.....	52
3.4.3	Comentarios.....	53
3.4.4	Buenas prácticas .....	53
3.5	Pruebas de software.....	53
3.5.2	Pruebas unitarias.....	54
3.5.3	Pruebas funcionales.....	57
3.5.4	Pruebas de rendimiento.....	60
3.6	Conclusiones de capítulo.....	61
<b>Conclusiones generales</b>	.....	<b>63</b>
<b>Recomendaciones</b>	.....	<b>64</b>
<b>Referencias bibliográficas</b>	.....	<b>65</b>
<b>Anexos</b>	.....	<b>67</b>
Anexo 1	Descripción del CUS "Gestionar tipo de expediente" .....	67
Anexo 2:	Matriz de trazabilidad RF-RF.....	71
Anexo 3:	Prototipo de interfaz (Panel Principal del módulo).....	72
Anexo 4:	Prototipo de interfaz (Vista eliminar expediente del módulo).....	73
Anexo 5:	Prototipo de interfaz (Panel editar expediente del módulo).....	74
Anexo 6:	Atributos de calidad para las pruebas TOC .....	75
Anexo 7:	Umbrales de prueba RC.....	76
Anexo 8:	Caso de prueba para el escenario "Modificar expediente". .....	77

## Introducción

---

La gestión documental es una actividad casi tan antigua como la escritura, nació debido a la necesidad de documentar actos administrativos y transacciones legales y comerciales por escrito para dar fe de los hechos.

Durante siglos la gestión documental en las organizaciones fue de dominio exclusivo de administradores, archiveros y bibliotecarios, cuyas herramientas manuales básicas eran los libros de registro, las carpetas, archivadores, cajas y estanterías en las que se guardan los documentos de papel y más tarde los audiovisuales y los documentos en soportes magnéticos u ópticos, y una larga lista de técnicas de recuperación de información mediante sistemas de codificación y clasificación.

Recientemente se han ido sumando a ellos los informáticos, que son cada vez más necesarios debido al nivel de complejidad y modernización que van alcanzando los sistemas computacionales de apoyo de la actividad administrativa[1].

La gestión de documentos siempre ha representado un problema para las empresas debido al gran cúmulo de información, los gastos de materiales, el amplio espacio ocupado, el engorroso trabajo en la búsqueda de información y en la conservación de los documentos.

Aunque hoy en día las empresas utilizan cada vez menos documentos en soporte físico de papel, aún existen muchos documentos dentro de la misma que deben existir en este soporte, por ejemplo, contratos firmados, escrituras, pedidos y facturas que los proveedores o clientes envían en papel, por lo que el volumen de documentos en papel sigue siendo importante y complicado de gestionar.

Esta información almacenada en papel no continuó demasiado tiempo, cuando el uso de las tecnologías de la información y las comunicaciones (TIC) se hizo común en la administración pública y privada, surgió la necesidad de capturar y conservar los documentos en formato electrónico. Conseguir esto representó un nuevo salto en la complejidad y exigencias a los sistemas informatizados y significó un gran cambio en lo que a gestión documental se refiere. Con este desarrollo tecnológico se crearon los sistemas electrónicos de gestión documental, con el objetivo de resolver las grandes acumulaciones de información que existían en papel hasta el momento.

Un Sistema Electrónico de Gestión Documental es una aplicación informática utilizada para archivar documentos electrónicos y/o imágenes de documentos de papel escaneados. Actualmente en el mercado existen muchas soluciones disponibles para implantar los Sistemas Electrónicos de Gestión Documental en las empresas, incluso algunas de ellas disponible en la modalidad de software libre, con

las ventajas de reducción de costes y la facilidad de implementación y adaptación que ello implica. Entre ellas, las más conocidas son: Alfresco, Archivista, OpenKM, Knowledge Tree entre otras [2].

Estos sistemas resuelven en gran medida problemas de la sociedad actual, dando acceso rápido a la información necesaria, reducen el tiempo a la hora de consultar y localizar grandes volúmenes de documentos y liberan espacio en las empresas.

Alfresco es una de las aplicaciones de gestión documental creadas con el fin de guardar, clasificar, catalogar, describir y almacenar documentos digitales. Es uno de los gestores de contenido más conocidos a nivel mundial, el cual ofrece muchas herramientas y funciones, entre ellas se puede mencionar la gestión documental, gestión de contenidos web, administración de contenido colaborativo (wikis, redes Sociales) y la gestión de información [3].

La cantidad de expedientes que actualmente se acumulan en las instituciones a nivel mundial, deben administrarse mediante nuevas herramientas electrónicas capaces de actuar ágilmente para lograr la solución y ejecución de los trámites requeridos por las empresas, teniendo en cuenta las necesidades requeridas en la explotación de expedientes. En la gestión de los documentos los expedientes se almacenan y se organizan de acuerdo a una estructura lógica o cuadro de clasificación. Los expedientes están conformados por volúmenes que no son más que divisiones de un expediente que facilita la gestión de los documentos contenidos en el mismo [4].

En la Universidad de las Ciencias Informáticas en el centro de informatización de la gestión documental (CIGED) de la facultad 2, debido a su desarrollo de sistemas y servicios informáticos vinculados a la gestión documental surgió la necesidad de utilizar Alfresco para manejar el almacenamiento de los documentos electrónicos y así facilitar la gestión de los mismos controlando sus contenidos y flujos; pero con el objetivo de darle al usuario final una interfaz más usable y un área de trabajo más fácil de interactuar por usuarios no especializados que el panel de administración del Alfresco, se crea el Gestor de Documentos Administrativos (GDA) eXcriba, el cual permite automatizar los procesos de la institución.

Las instituciones y empresas de nuestro país realizan la gestión documental utilizando normas descritas por el Archivo Nacional de Cuba (ANC) referentes a la gestión de documentos físicos, en la actualidad el auge de las TIC ha aumentado la cantidad de documentación electrónica y por consiguiente se necesita una herramienta que permita gestionar el ciclo de vida de un documento (capturar, guardar, preservar, compartir, versionar, aprobar, publicar y archivar) para ello el ANC decidió que uno de los sistemas a utilizar es el producto XABAL-eXcriba. Este sistema a pesar de que presenta una solución para la gestión de los expedientes de los documentos electrónicos de los proyectos del proceso productivo de la UCI, el mismo presenta las siguientes limitaciones:

- La estructura del expediente y los documentos electrónicos que lo conforman son introducidos al sistema primeramente creando la estructura de carpetas y luego subiendo los documentos de cada carpeta, porque el sistema no permite subir una carpeta.
- Los permisos a los documentos se otorgan por un script y la persona que realice esa tarea en una empresa o institución debe tener conocimientos de programación para implementar ese script.
- No presenta un mecanismo de seguimiento y control de los expedientes y de los documentos electrónicos que componen al expediente.
- Permite crear búsquedas de los expedientes y documentos, pero no permite aplicarles filtros a los expedientes creados.

A partir del contexto anterior, se identifica el siguiente **problema científico**:

¿Cómo lograr un mayor control y seguimiento de los expedientes de documentos electrónicos de las empresas e instituciones con el producto XABAL-eXcriba 4.0?

Considerándose como **objeto de estudio**:

Gestión de expedientes.

Para darle solución a la problemática existente se establece como **objetivo general**:

Desarrollar el módulo de gestión de expedientes genéricos en el producto XABAL eXcriba 4.0 que permita a las empresas e instituciones mantener un control y seguimiento de sus expedientes electrónicos.

**Campo de acción.**

Automatización de la gestión de expedientes electrónicos.

A partir del objetivo general se derivan las siguientes preguntas de investigación:

1. ¿Cuáles son los referentes teóricos a tener en cuenta para abordar la solución del problema planteado relacionado con la automatización de la gestión de expedientes electrónicos en la tecnología de Alfresco 5.2?
2. ¿Qué propuesta de solución se define para implantar en el módulo desarrollado la automatización de la gestión de expedientes electrónicos en la tecnología de Alfresco 5.2?
3. ¿Cómo desarrollar un módulo de gestión de expedientes electrónicos en la tecnología de Alfresco 5.2, que permita el seguimiento y control de los expedientes electrónicos generados en las diferentes empresas y organizaciones?

4. ¿Cómo se valida el correcto funcionamiento del módulo de gestión de expedientes electrónicos en la tecnología de Alfresco 5.2?

#### **Tareas de investigación.**

- Análisis de los principales conceptos asociados a la gestión de expedientes electrónicos en la tecnología de Alfresco 5.2, para obtener los fundamentos teóricos necesarios para el desarrollo de la solución.
- Análisis de los sistemas homólogos para identificar tendencias actuales relacionadas con las funcionalidades y mecanismos utilizados.
- Diseño del mecanismo de control y seguimiento a utilizar en el módulo de los expedientes electrónicos.
- Implementación del módulo de gestión de expedientes electrónicos basada en el mecanismo de control y seguimiento definido.
- Validación del módulo de gestión de expedientes electrónicos en la tecnología de Alfresco 5.2 para comprobar su correcto funcionamiento

#### **Aportes prácticos.**

- Un conjunto de expedientes electrónicos automatizados en el producto XABAL-eXcriba 4.0.
- Un mecanismo de control y seguimiento de los expedientes electrónicos generados en una empresa o institución.
- Un módulo de gestión de expedientes electrónicos en el producto XABAL-eXcriba 4.0 que le permitirá a las empresas o instituciones llevar el control y seguimiento de los mismos.

Para desempeñar la investigación se utilizaron los siguientes métodos teóricos y empíricos:

- **Teóricos**

1. Analítico-Sintético: se utilizó en el análisis bibliográfico del tema y para realizar una síntesis de los aspectos esenciales relacionados con el desarrollo de la gestión documental. Recopilando la información necesaria mediante la revisión de documentos, artículos científicos y sitios webs especializados, además se analizó las diferentes herramientas, metodologías y tecnologías de desarrollo de software a utilizar en la solución.
2. Histórico-Lógico: previamente se hizo un estudio del desarrollo de la gestión documental en el Centro de Informatización de la Gestión Documental, permitiendo realizar así un estudio y análisis del módulo para la gestión de expedientes en el eXcriba 4.0, para lograr una mejor comprensión del negocio y de esta manera tener una visión tanto de los principales problemas como de los avances alcanzados en el módulo.

3. Modelado: se utiliza en la modelación de los diagramas correspondientes a la etapa de análisis, diseño e implementación del módulo a desarrollar en la presente investigación.
  4. Sistémico-estructural-funcional: se utilizó para sistematizar la jerarquía del sistema, la estructura y las relaciones funcionales de la esencia del proceso y sobre esta base proponer una metodología de la enseñanza de la comprensión de lectura.
- **Empíricos**
    5. Entrevista: utilizada en el levantamiento de información a partir de un grupo de entrevistas realizadas al personal del centro, para obtener información rápida y precisa de sus principales problemas, necesidades y posibles resultados esperados.
    6. Observación: para comprender la forma en que se realiza el proceso de la Gestión de los expedientes.

La investigación se estructura en introducción, tres capítulos, bibliografía y anexos donde se abarca todo lo relacionado con la investigación realizada.

**Capítulo 1 Fundamentación teórica:** En este capítulo se hace un estudio de los procesos de gestión documental, se definen los conceptos más importantes para la investigación, se abordan los diferentes elementos que brindan la base teórico-conceptual para el desarrollo de la solución propuesta, así como la realización de un estudio del módulo para la gestión de expedientes en eXcriba 4.0. Además, se realiza un análisis sobre la metodología utilizada, las normas por las que se rige la gestión documental y las herramientas necesarias para dar cumplimiento al objetivo de la presente investigación.

**Capítulo 2 Análisis, diseño e implementación de la propuesta de solución:** En este capítulo se describe la solución que se propone, dando a conocer primeramente el modelo conceptual. Se definen los requisitos funcionales y no funcionales que ofrecen una solución al problema de investigación. Además, se generan los artefactos según la metodología seleccionada AUP versión UCI en su escenario número dos: casos de uso del sistema. Además de la arquitectura propuesta para el componente a desarrollar, generando todos los artefactos necesarios.

**Capítulo 3 Implementación y validación de la solución:** En este capítulo se exponen el modelo de implementación mediante los diagramas de componentes y de despliegue. Se describen los resultados de la implementación del módulo para la gestión de expedientes en eXcriba 4.0. Finalmente se realiza la validación de la investigación y se realizan pruebas comprobando el cumplimiento del objetivo general.

# Capítulo 1: Fundamentación teórica

---

En el presente capítulo serán analizados los fundamentos de la gestión de expedientes, así como algunas soluciones de sistemas de gestión de expedientes existentes en el mundo y en la Universidad de las Ciencias Informáticas. Se muestran además las diferentes herramientas, tecnologías y metodología utilizadas para la implementación del sistema.

## 1.1 Conceptos fundamentales

A continuación, se evidencian un conjunto de conceptos asociados a la investigación que permitirán una mayor comprensión de la misma.

### 1.1.1 Gestión documental (GD):

La gestión documental es el conjunto de normas que se aplican para gestionar los documentos de todo tipo que se crean y reciben en una organización. Esta gestión de documentos debe facilitar su recuperación, permitir la extracción de información, la conservación de los que son importantes durante el tiempo que sean útiles, así como la destrucción cuando ya no se necesitan, utilizando para todo esto métodos eficaces y eficientes[5].

La Gestión Documental posibilita la optimización de archivos desde su origen hasta la disposición final garantiza su correcta conservación, por lo que la aplicación de estrategias de gestión ayuda a llevar a cabo la custodia de archivos, la eliminación técnica y el almacenamiento y recuperación de los mismos[6].

En sí, la gestión documental permite:

- Revisión de información.
- Ahorro y eficiencia.
- Categorización de archivos.
- Eliminación de archivos que han perdido vigencia.
- Almacenamiento ordenado.
- Custodia de documentos.

De este modo, permite a la empresa administrar de manera eficaz todo lo relacionado a su archivo, brindando ayuda de manera considerable a la ejecución y revisión de todo lo que es calificado como importante para la empresa[6].

### **1.1.2 Sistemas de Gestión Documental (SGD):**

Un sistema de gestión documental está diseñado para almacenar, administrar y controlar el flujo de documentos dentro de una organización. Se trata de una forma de organizar los documentos e imágenes digitales en una localización centralizada a la que los empleados puedan acceder de forma fácil y sencilla [7].

Los sistemas de gestión documental se están convirtiendo en una pieza cada vez más importante para las empresas. La cantidad de información generada por las compañías actuales crece de forma exponencial y los antiguos sistemas de almacenamiento presentan demasiadas carencias. Un sistema de gestión documental asegura una información organizada y eficiente, que favorezca la productividad empresarial [7].

Con un sistema de gestión documental, se evita la pérdida de documentos, ya que están todos seguros en un servidor documental central. También se evita la pérdida de tiempo debido a que varias personas tengan que recuperar los documentos desde diferentes lugares y departamentos.

Con un sistema de gestión documental, los documentos son inmediatamente recuperados por la persona que está buscando la información. Esto cambia el enfoque del problema de la recuperación de documentos, para que las personas solo se preocupen de realizar trabajo productivo para su empresa[8].

### **1.1.3 eXcriba**

Sistema de gestión documental orientado a la automatización de los procesos documentales de las organizaciones. Contiene todos los instrumentos para cubrir desde la generación de documentos, su revisión, administración, distribución, custodia y disposición, cumpliendo con las normas internacionales para la gestión de documentos, manteniendo la seguridad de los documentos en formato electrónico. Entre sus principales funcionalidades se destacan el control de versiones, accesos y permisos, automatización de los flujos documentales, gestión de documentos y carpetas, gestión de notas sobre los documentos y listas de datos. Contiene módulos para la gestión de acuerdos, gestión de reportes, notificaciones, gestión de registros de entrada y salida de documentos y gestión del cuadro de clasificación de la entidad [9].



#### **1.1.4 Expediente electrónico:**

Se entiende por expediente electrónico al conjunto de documentos electrónicos que forman parte de un procedimiento administrativo. Para completar la definición se necesita definir qué es un documento electrónico. Se llama documento electrónico a cualquier información en formato electrónico que se archiva en soporte electrónico con un formato predeterminado que permite identificación y tratamiento diferencial. Por ende un expediente electrónico está formado por todos los documentos electrónicos que son necesarios para la gestión, que una persona pueda solicitar [10].

La estructura de un expediente electrónico la establece la norma técnica de interoperabilidad de expediente electrónico la cual indica que el expediente electrónico contiene:

- Documentos electrónicos.
- Índice electrónico.
- Firma electrónica.
- Un mínimo de metadatos obligatorios.
- Especificaciones para servicios de remisión y puesta a disposición.

#### **Ventajas del expediente electrónico[10]:**

- Descentralización. El expediente electrónico permite el inicio de la tramitación desde distintas ubicaciones sin limitación de restricciones horarias. Se puede interactuar sin ser necesario la presencia física de los usuarios. De este modo, los trámites administrativos se acercan al usuario.
- Validez legal. El expediente electrónico está regulado por un marco legal que lo legitima y legaliza. Toda actuación en un expediente electrónico tiene el mismo valor legal que las realizadas con expedientes en papel.
- Transparencia. Un expediente electrónico puede ser consultado en cualquier momento por el interesado, conocer su estado o la etapa en la que se encuentra, y cuáles van a ser los siguientes pasos. Todo esto se actualiza cada vez que el expediente electrónico se mueve, y está a disposición del usuario las 24 horas del día mediante Internet.
- Interoperabilidad. Un expediente electrónico puede ser interoperado entre diversos organismos sin que se necesite mover papeles y sin que se produzcan demoras. Se evita la posibilidad de deterioro de expediente o del extravío de documentos.

- Rápida localización. Un expediente electrónico está almacenado en un sistema de gestión documental con bases de datos indexadas. Esto permite que pueda ser almacenado sin necesidad de utilizar enormes espacios físicos como los que se necesitan con los expedientes en papel. La tecnología nos permite localizar documentos que pueden ser accedidos por cualquier usuario autorizado en cualquier momento y a mucha velocidad.
- Heterogeneidad documental. En un expediente electrónico se pueden incorporar diferentes tipos de documentos, algunos de los cuales tienen nacimiento directamente en digital, cómo puede ser una hoja de cálculo, un correo electrónico, un documento PDF, etc. El expediente electrónico funciona a modo de contenedor en el que cualquier elemento relevante puede ser incluido.
- Copia fiel. Un expediente electrónico y sus documentos pueden ser reproducidos tantas veces como se necesite, siendo cada una de estas veces una copia fiel del original.
- Seguridad. El expediente electrónico es más seguro que el de papel. La posibilidad de extravío de páginas de un documento desaparece. No se necesita imprimir en papel por lo que el documento no puede ser interceptado por nadie que no tenga acceso electrónico autorizado. Los documentos de un expediente electrónico no se deterioran.

### **1.1.5 Sistema de gestión de expedientes electrónicos.**

Un sistema de gestión de expedientes electrónicos podría definirse como la solución que permite a las instituciones o empresas la tramitación y seguimiento integral de cualquier expediente y documento. El sistema de gestión de expedientes electrónico cubre todo el proceso administrativo, simplificando y optimizando además permitiendo el diseño, implantación y desarrollo de las prácticas de gestión de documentos electrónicos establecidas en forma de programa de actuación, dotándose de los recursos oportunos para su funcionamiento[11].

El sistema de gestión de expedientes se articula sobre las fases del ciclo de vida de los documentos y se nutre de los documentos incorporados a través del proceso de captura, con independencia de que los documentos hayan sido creados dentro o fuera de la propia organización [11].

## **1.2 Análisis de sistemas homólogos.**

Teniendo en cuenta el estudio del problema científico y enfocando la investigación en el campo de acción: “Automatización de la gestión de expedientes electrónicos” se realiza un análisis de sistemas homólogos con el objetivo de estudiar las funcionalidades propias del sistema a desarrollar y adquirir nuevos conocimientos sobre soluciones ya existentes, para luego llegar a una conclusión en base a diferentes criterios.

### **1.2.1 abox ecm**

La mayor parte de las aplicaciones de gestión documental agrupan documentos por carpetas, abox ecm se diferencia de otras plataformas por su potente gestor de expedientes, que permite ejercer un control exhaustivo de cualquier trámite agrupando documentos y tratando el conjunto como unidad documental [12].

Un expediente electrónico debe almacenar todos los contenidos de un trámite y controlar la manera y el orden en que son cargados a la herramienta.

abox ECM permite realizar una gestión sobre los expedientes, como unidad documental, adicional a la gestión sobre los documentos que en ellos se contienen.

A continuación, se muestran algunas prestaciones fundamentales: control de tipos documentales, tratamiento del expediente como unidad documental, gestor de expedientes off-line, visor on-line de contenidos, cierre certificado, workflow<sup>1</sup>, organización jerárquica, opciones de carga y descarga masivas, etc.[12]

Control de tipos documentales:

- Tipos documentales esperados.
- Requisitos previos por tipo documental.
- Tipos documentales libres.
- Control automático de progreso del expediente.
- Control de tipos documentales existentes o pendientes.

Tratamiento del expediente como unidad documental:

- Ficha completa del expediente.
- Metadatos exclusivos por serie documental.
- Relaciones entre expedientes.

---

<sup>1</sup> Flujo de trabajo: estudio de los aspectos operacionales de una actividad de trabajo.

- Comentarios y redes sociales.
- Workflow del expediente.
- Auditoría por expediente.

Visor online de expedientes:

- Formatos admitidos: (PDF y PDF/A – TIFF – Office – PNG, JPEG, GIF – EML, MSG (Outlook). y MHT – Multimedia (WAV, MP3, MP4, AVI, ...) – ZIPs y otros formatos de archivo.

Cierre certificado y archivado:

- Los expedientes pueden cerrarse o archivarse, según se defina en su flujo.
- Archivado manual o por workflow (el expediente se oculta a los usuarios sin privilegios).

Servidor de aplicación:

- Plataforma Microsoft .NET
- Internet Information Server.
- Escalamiento por balanceo de carga S.O. Windows Server.

Servidor de base de datos (Funciona sobre 3 motores de B.D en Windows):

- Sql Server 2008 o superior.
- Oracle 10g o superior.
- PosgreSql 8.4 o superior.

Servidor de ficheros:

- Permite el almacenamiento distribuido de los ficheros de documentos.
- Repositorios escalables.
- Funciona sobre Windows.

### **1.2.2 Archive.**

Archive es una aplicación web de archivo definitivo de expedientes y documentos electrónicos. Proporciona las herramientas necesarias para la creación por parte de un super-administrador de un sistema de administración y gestión de Centros de Archivo multidepartamental, así como la integración en Archive de las correspondientes aplicaciones consumidoras y la gestión de los documentos y expedientes electrónicos remitidos por las mismas[13].

Los expedientes y documentos serán importados y clasificados dentro de Archivos y series documentales, y pasarán por diversos estados (tanto de forma automática, como manual), en función

de la política de conservación asignada. Archive proporciona también, mecanismos de notificación de los cambios de estados acaecidos en los expedientes, así como distintos módulos de gestión (usuarios y roles, archivos documentales, clasificación) [13].

Por otro lado, dispone de una serie de operaciones accesibles a través de servicios web, tales como enviar SIP<sup>2</sup>, buscar expedientes u obtener actas de ingreso. Archive cubre todo el ciclo de vida de los expedientes y sus sucesivas fases de archivo a largo plazo.

Permite generar, para cada archivo y serie documental, los metadatos del esquema de metadatos de la Norma Técnica de Interoperabilidad de Política de Gestión de Documentos Electrónicos<sup>3</sup>, necesarios para la gestión de Expedientes[13].

Entre las funcionalidades y características que posee se contemplan:

- Destinatarios: Cualquier Administración Pública.
- Modo de Uso:
  - Producto instalable.
  - Servicio en red de infraestructuras.
  - Servicio en red para usuario final.
- Tipo de Solución: Web y Aplicación.
- Licencia: EUPL (European Public License).
- Nivel Interoperabilidad: Semántico, Técnico.
- Lenguaje de Programación: J2EE, Servicios Web.

### 1.2.3 Gedson.

Gedson facilita la creación y seguimiento de los expedientes jurídicos, facilitando el acceso de forma rápida a toda la información contenida en el mismo, actuaciones, documentos, provisiones y facturación. Posee generación automática de documentos, puede generar, en distintos formatos y de forma rápida aquellos documentos de uso común. Estos documentos se formalizan a partir de las plantillas indicadas y se adapta a tu imagen corporativa. De esta manera se asegura de cubrir toda la información de los expedientes. Gedson brinda el seguimiento y gestión de tus expedientes, y puedes acceder desde cualquier lugar, sin tener que instalarlo en el PC de tu despacho gracias a su plataforma web[14].

---

<sup>2</sup> Sesión Initiation Protocol (Protocolo de inicio de sesión)

<sup>3</sup> Documento que establece conceptos de las políticas de gestión de documentos electrónicos.

Gedson permite reducir el tiempo en la elaboración y seguimiento de los expedientes y verificar en todo momento y de una forma rápida el estado de los expedientes y la actividad e información generada en cada una de las actuaciones[14].

Características:

- Consulta de forma sencilla todos los expedientes.
- Asigna a los expedientes actuaciones, avisos/eventos, documentos y provisiones.
- Gestión de los documentos, pudiendo asignar permisos de acceso y bloqueando la modificación del mismo a un usuario.
- Módulo de plantillas y documentos donde puedes incluir en el sistema aquellos documentos necesarios para la cumplimentación del expediente o actuaciones, teniéndolo disponible en todo momento para facilitar su acceso y uso.
- Permisos de acceso, permite decidir si los expedientes son públicos o privados, pudiendo indicar qué usuarios pueden acceder a su información y el modo en que lo hacen, consulta o edición.
- Proporciona un conjunto de informes que facilitan el control de los expedientes, clientes, contrarios, procuradores, facturación, etc.
- Seguimiento de la actividad del expediente.
- Generación automática y descarga de documentos en los formatos más habituales (Word, Excel, PDF, OpenOffice)
- Lenguaje de Programación: JAVA, Servicios Web.
- Sistema Operativo: Linux, Windows.
- Posee licencia de pago para la utilización de todas sus funcionalidades.

#### **1.2.4 Q-expeditive.**

Es una solución integral de expediente electrónico que permite definir los trámites en base a su ruta, formularios y documentación anexa. Q-expeditive ofrece una solución bajo licencia de pago que integra las modernas tecnologías de digitalización de documentos, firma digital, Internet, correo electrónico, workflow y bases de datos. Permite definir gráficamente los trámites que una organización lleva a cabo, y ejecutar automáticamente estos. Su arquitectura orientada a servicios le confiere una gran capacidad de integración con otros sistemas, lo cual hace de Q-expeditive un producto especialmente apropiado para grandes organizaciones que cuentan con sistemas heterogéneos, por ejemplo, entidades gubernamentales y grandes empresas. Q-expeditive está construido sobre la sólida base de Q-flow, producto con el que se integra naturalmente. Q-flow es una completa herramienta de BPM (Business

Process Management) y le provee a Q-expeditive avanzadas herramientas de modelado de procesos, manejo de seguridad y organigrama, integración con sistemas externos y mucho más [15].

Beneficios de usar Q-expeditive [15].

- Mayor eficiencia operativa.
- Registro automático de información para auditoría.
- Prevención de atrasos.
- Defensa contra alteraciones malintencionadas de documentos.
- Disminución de requerimientos de espacio resultantes de la necesidad de almacenar grandes cantidades de papel.
- Disponibilidad de información del estado de un trámite para consulta del público.

### 1.2.5 GETET.

Herramienta para la gestión de expedientes técnicos de las estaciones de trabajo desarrollada por el centro CEIGE.

Esta herramienta permite generar los expedientes y las fichas técnicas de las estaciones de trabajo, garantizando una gestión centralizada de los recursos, aumentando la rapidez y efectividad del proceso. Además, se facilita obtener algunos reportes sobre la información que gestiona el sistema tales como: la relación de las estaciones rotas o con sellos rotos, las estaciones disponibles y otros reportes relacionados las prestaciones de las estaciones de trabajo. La herramienta permite además obtener la información del hardware de las estaciones de una subred seleccionada y comparar esos resultados con los registrados en su base de datos, de esta manera se podrán detectar alteraciones en las características físicas de las computadoras y emitir alertas a usuarios ya predefinidos, informando de cualquier anomalía. GETET es capaz de personalizar la información conforme al área en la que se esté trabajando, resolviendo así el problema relacionado al cúmulo de información que recibían todos los jefes de áreas[16].

Características:

- La aplicación pertenece al grupo de las aplicaciones web por lo que comprende el uso de lenguajes de programación del lado del cliente y del lado del servidor.
  - Programación del lado del cliente Java Script 1.8
  - Programación del lado del servidor PHP 5.5
- RapidSVN 1.6.6 cliente gráfico para Subversion.
- El gestor de base de datos PostgreSQL 9.1
- El servidor webApache 2.2

### 1.3 Valoración del estudio realizado.

Luego de realizado el análisis de los sistemas de gestión de expedientes electrónicos descritos anteriormente y las funcionalidades que estos proveen, se concluye que ninguno cumple del todo con las necesidades de implementación de una solución de gestión de expedientes para el GDA eXcriba, porque no se integran con él mismo, ya que son aplicaciones basadas en plataformas privativas y poseen licencias de uso, no poseen la capacidad de gestionar de manera genérica los expedientes o no poseen plataforma web.

Para confeccionar un gestor de expedientes que cumpla con las necesidades del GDA eXcriba se deberán tener en cuenta algunas de las ventajas que ofrecen varios de los sistemas analizados, tales como la gestión de los expedientes, desde su creación hasta que es cerrado el mismo, admitir la integración y exportación de los documentos en diferentes formatos, como: word, excel, txt, pdf, imágenes, la gestión de los documentos del expediente posibilitando asignación de permisos de acceso y bloqueando la modificación del mismo a usuarios en dependencia de sus roles y permitir un almacenamiento más organizado mediante la clasificación estos. Brindando así una mayor información para la construcción de una solución que resuelva el problema planteado.

*Tabla 1: Características de los sistemas homólogos*

	Expedientes genéricos	Software libre	Web	Seguimiento y control
Abox ecm	si	no	no	si
Archive	si	no	si	si
GEDSON	no	no	si	si
Q-expeditive	si	no	si	si
GETET	no	si	si	si

### 1.4 Metodología de Desarrollo

Las metodologías de desarrollo de software coleccionan un conjunto de pasos y procedimientos que se deben seguir para organizar, controlar y planear el proceso de desarrollo de un software. Surgen ante la necesidad de trabajar mediante el uso de procedimientos, técnicas, herramientas y documentos durante el desarrollo de software[17].

Con vistas a garantizar el correcto funcionamiento del desarrollo productivo, la Universidad de las Ciencias Informática adopta una metodología por la cual se deben regir los proyectos productivos en la



misma, se basa en una variación de la metodología “Proceso Unificado Ágil” (AUP). Esta metodología establece distintas fases y disciplinas por las que se debe transitar durante el desarrollo. Las fases definidas son: Inicio, Ejecución (elaboración, construcción, transición) y Cierre. Las disciplinas son: Modelo (modelado del negocio, requisitos, análisis y diseño), Implementación, Pruebas (internas, liberación y aceptación) y Despliegue. La solución que se desea implementar se enmarca en la fase de ejecución, en la cual se ejecutan las actividades requeridas para desarrollar el software. Durante la ejecución se realizan los procesos de negocio, se refinan los requisitos, se elabora la arquitectura y el diseño, se implementa y se libera el producto. El objetivo de esta fase es obtener un sistema que satisfaga las necesidades de los clientes y usuarios finales[17].

La metodología de desarrollo de software AUP para la UCI constituye una guía para emplear las mejores prácticas en una entidad desarrolladora. Estas prácticas se centran en el desarrollo de productos y servicios de calidad. AUP-UCI propone tres formas de describir los requisitos: casos de uso del sistema (CUS), historias de usuario (HU) y descripción de requisitos por proceso (DRP), agrupados en cuatro escenarios condicionados por el modelado de negocio, los cuales quedan definidos de la siguiente forma[18]:

1. Los proyectos que modelen el negocio con casos de uso del negocio (CUN) solo pueden modelar el sistema con CUS.
2. Los proyectos que modelen el negocio con modelo conceptual (MC) solo pueden modelar el sistema con CUS.
3. Los proyectos que modelen el negocio con descripción de proceso de negocio (DPN) solo pueden modelar el sistema con DRP.
4. Los proyectos que no modelen negocio solo pueden modelar el sistema con HU.

Para la realización de este trabajo se empleó el escenario No. 2 de la metodología AUP-UCI en su fase de ejecución, incluyendo las disciplinas de modelado, implementación y pruebas. Este escenario se aplica a los proyectos que conozcan el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma se modelarían exclusivamente los conceptos fundamentales del negocio. Se recomienda este escenario para proyectos donde el objetivo primario es la gestión y presentación de información.

## **1.5 Lenguajes de programación, herramientas y tecnologías**

A continuación, se describen los principales lenguajes, herramientas y tecnologías a utilizar en el desarrollo de la investigación.

## 1.5.1 Lenguajes de programación.

Para la implementación de las funcionalidades en Alfresco se utiliza JavaScript 1.6 y para la definición de los modelos de contenido y los formularios se utilizó el lenguaje XML en su versión 1.0.

### 1.5.1.1 JavaScript 1.6

El motor de JavaScript integrado en Alfresco utiliza el motor de Mozilla Rhino JavaScript para Java, por lo que permite la utilización de JavaScript en su versión 1.6. JavaScript es un lenguaje script multiplataforma. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. JavaScript es un lenguaje de programación ampliamente utilizado en el mundo del desarrollo web por ser muy versátil y potente, tanto para la realización de pequeñas tareas como para la gestión de complejas aplicaciones[19].

Técnicamente es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios[19].

Se decide utilizar JavaScript v1.6 para la implementación de los servicios, debido a que permite hacer uso de la API<sup>4</sup> de JavaScript que Alfresco provee. Con la utilización de esta API es posible acceder, gestionar y manejar los objetos de la raíz de Alfresco.

### 1.5.1.2 XML 1.0

XML<sup>5</sup> está diseñado especialmente para los documentos de la web, este permite que los diseñadores creen sus propias etiquetas, permitiendo la definición, transmisión, validación e interpretación de datos entre aplicaciones y entre organizaciones. Es un lenguaje de formato universal para documentos y datos estructurados; permite intercambiar información estructurada entre diferentes plataformas de una manera segura, fiable y libre. Entre sus principales ventajas se encuentran[20]:

- Es un estándar internacionalmente conocido.
- No pertenece a ninguna compañía y su utilización es libre.
- Fácilmente procesable tanto por humanos como por software.
- Separa la información o el contenido de su presentación o formato.
- Diseñado para ser utilizado en cualquier lenguaje o alfabeto.
- Posee una estructura jerárquica.

---

<sup>4</sup> Application Programming Interface (Interfaz de programación de aplicaciones)

<sup>5</sup> Extensible Markup Language (Lenguaje de marcado extensible)

- El número de etiquetas es ilimitado.

El uso de este lenguaje en el presente trabajo es debido a que la extensión de los ficheros asociados al modelo de dominio es XML para de esta forma ser interpretados por Alfresco[20].

### **1.5.1.3 HTML 5.0**

HTML<sup>6</sup> es un lenguaje que se utiliza comúnmente para establecer la estructura y contenido de un sitio web, tanto de texto, objetos e imágenes. El lenguaje de HTML funciona por medio de "etiquetas" que describen la apariencia o función del texto enmarcado. Este lenguaje puede llegar a incluir un script o código que tenga incidencia en el comportamiento del navegador web del cliente [25].

### **1.5.1.4 CSS 3.0**

Hojas de estilo en cascada (CSS, del inglés *Cascading Style Sheets*) es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML. El CSS sirve para definir la estética de un sitio web en un documento externo y eso mismo permite que modificando ese documento podamos cambiar la estética entera de un sitio web. Con la utilización de CSS se obtiene un mayor control de la presentación del sitio al poder tener todo el código CSS reunido en uno, lo que facilita su modificación[25].

### **1.5.1.5 Spring 4.3.5 Framework**

Spring, ofrece como elemento clave el soporte de infraestructura a nivel de aplicación, brindando un completo modelo tanto para la configuración como para la programación de aplicaciones empresariales desarrolladas bajo Java, sin discriminación en cuanto al despliegue de la plataforma.

Todo esto trae consigo la ventaja de permitir que los equipos de desarrollo puedan enfocarse directamente en la lógica empresarial que requiere la aplicación, haciendo el proceso más corto, rápido y eficaz, ahorrando líneas de código evitando tareas repetitivas [22].

## **1.5.2 Herramientas**

Las herramientas de la Ingeniería de Software son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software, reduciendo el costo de las mismas en términos de tiempo y dinero. Estas herramientas ayudan en todas las etapas del ciclo de vida de desarrollo del software, en tareas como: el proceso de realizar un diseño del proyecto, cálculo de costos,

---

<sup>6</sup> HiperText Markup Language (Lenguaje de Marcación de Hipertexto)

implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación y detección de errores[17].

A continuación, se hará un breve resumen de las herramientas necesarias para el desarrollo del módulo.

#### **1.5.2.1 Visual Paradigm v8.0**

Es una herramienta, para modelado en UML. Esta herramienta tiene características gráficas muy cómodas, que facilitan la realización de los diagramas de modelado que sigue el estándar de UML; los mismos son: diagramas de clases, casos de uso, comunicación, secuencia, estado, actividad, componentes. Entre sus principales ventajas se encuentran: que presenta entorno de creación de diagramas para UML 2.0; su diseño es centrado en casos de uso y enfocado al negocio, lo cual genera un software de mayor calidad; posee uso de un lenguaje estándar común a todo el equipo de desarrollo, que facilita la comunicación, las capacidades de ingeniería directa e inversa, modelo y código, que permanece sincronizado en todo el ciclo de desarrollo [17].

Las ventajas que proporciona Visual Paradigm para UML son:

- **Dibujo:** Facilita el modelado de UML, ya que proporciona herramientas específicas para ello. Esto también permite la estandarización de la documentación, ya que la misma se ajusta al estándar soportado por la herramienta eXcriba.
- **Coherencia entre diagramas:** Al disponer de un repositorio común, es posible visualizar el mismo elemento en varios diagramas, evitando duplicidades.
- **Generación de código:** Permite generar código de forma automática, reduciendo los tiempos de desarrollo y evitando errores en la codificación del software.
- **Generación de informes:** Permite generar diversos informes a partir de la información introducida en la herramienta.

Para el modelado de los artefactos obtenidos durante el proceso de desarrollo se utilizará Visual Paradigm, ya que está disponible en múltiples plataformas, soporta aplicaciones web, está disponible en varios idiomas, además de disponer de fácil instalación y uso, además de presentar compatibilidad entre versiones.

#### **1.5.2.2 PostgreSQL 9.4**

PostgreSQL es un sistema de gestión de bases de datos relacional orientado a objetos y de código abierto que incluye diversas características tales como la herencia múltiple, funciones, restricciones, etc. El desarrollo de PostgreSQL no es manejado por una empresa o persona, sino que es dirigido por

una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre o apoyados por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group) [23].

### **1.5.2.3 Visual Studio Code v1.42.1**

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y MacOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. Es compatible con varios lenguajes de programación y un conjunto de características que pueden o no estar disponibles para un idioma dado [24].

### **1.5.3 Tecnologías informáticas.**

A continuación, se ejemplifican las tecnologías utilizadas.

#### **1.5.3.1 Alfresco.**

Es una plataforma o manejador de contenido para la Gestión de Contenidos, que ofrece entre sus funcionalidades un Sistema de Gestión Documental que posibilita la búsqueda y obtención de documentos, colaborar en ellos y gestionar su ciclo de vida en un repositorio centralizado[25].

Además, gestiona todos los contenidos en la empresa: documentos, imágenes, fotos, páginas web, registros, documentos XML y cualquier otro fichero. Los servicios que ofrece Alfresco son una de sus grandes ventajas. Permiten gestionar el contenido de los documentos y características, tales como, gestión de metadatos, control de versiones, gestión del ciclo de vida, flujo de trabajo, búsquedas, asociaciones a otros contenidos, etiquetado, comentarios, etc. [25]

#### **ECM<sup>7</sup> Community Alfresco 5.2**

La actual versión de Alfresco presenta las siguientes mejoras:

- Plataforma al aire libre.
- Herramientas de soporte en la consola de administración.
- Eliminación de GhostScript.
- Nuevas API REST.
- Nuevo explorador de API REST.
- Mejoras administrativas.

---

<sup>7</sup> Enterprise Content Management (Gestión de contenido empresarial).

- Facetas de selección múltiple.
- Acciones de archivos a granel en los resultados de búsqueda.

Alfresco incorpora un metamodelo en su estructura, que es la capacidad para declarar nuevos modelos de contenido. Básicamente un modelo contenido es la estructura que define cual información acompañará a los documentos digitales. Su definición permite dotar de semántica a los documentos para que posteriormente se puedan implementar búsquedas y procesos más eficientes e inteligentes [25].

## **1.6 Conclusiones del capítulo.**

En este capítulo se presentó la fundamentación teórica que sustenta el proceso de desarrollo del módulo. Se realizó un estudio detallado sobre los principales conceptos que están asociados al objeto de estudio, que permitió ganar en calidad y brindó un mayor dominio del problema. Se expusieron los aspectos más significativos relacionados con los expedientes electrónicos, siendo el punto de partida de la propuesta de solución. Se desarrolló una amplia investigación sobre los sistemas homólogos que tratan la gestión de expedientes electrónicos para documentar características y funcionalidades que poseen, determinándose que es necesario el desarrollo de un nuevo sistema para la gestión de expedientes que cumpla con los requerimientos establecidos. En este capítulo se define, la metodología de desarrollo, herramientas, así como los lenguajes y tecnologías que van a servir de guía para el proceso de desarrollo del módulo gestión de expedientes genéricos para el GDA eXcriba 4.0 en la red de centros de la UCI, teniendo como resultado:

- Metodología de Desarrollo: AUP UCI.
- Lenguajes de Programación: JavaScript 1.6, XML 1.0
- Lenguaje de Modelado: UML.
- Framework: Spring 4.3.5
- Herramienta CASE: Visual Paradigm 8.0
- Gestor de base de datos: PostgreSQL 9.4
- Tecnología: Alfresco Community 5.2

Quedando así sentadas las bases para el diseño e implementación del módulo Gestión de Expedientes Genéricos propuesto.

## **Capítulo 2: Análisis, diseño e implementación de la propuesta de solución**

---

El presente capítulo se describe el proceso de construcción de la solución y se hace énfasis en las principales funcionalidades que tendrá el sistema siguiendo la metodología de desarrollo AUP-UCI en su escenario número dos. Se realiza la descripción del modelo conceptual y a partir de este, definir los pasos con el fin de comenzar con el diseño e implementación del Módulo de Gestión de Expedientes genéricos del XABAL eXcriba 4.0. Además, se identifican los requisitos funcionales y no funcionales que se deben cumplir, la descripción de casos de uso del sistema, así como el diseño de las clases de la propuesta de solución.

### **2.1 Propuesta de solución.**

Con el objetivo de agilizar y facilitar la gestión de expedientes en el Gestor de Documentos Administrativos eXcriba, se requiere la implementación de un módulo para la gestión de los mismos. El módulo incorpora nuevas funcionalidades, entre las que se encuentran:

- Posibilitará la gestión de todo tipo de expedientes, siendo un módulo de gestión de expedientes genéricos.
- Permitirá el seguimiento y control de los expedientes, así como de los documentos que conforman el mismo, mediante la aplicación de filtros y búsquedas sobre estos.
- Permitirá a cualquier usuario autorizado la asignación de roles de los diferentes miembros de proyecto que lo componen, sin necesidad de contactar con algún miembro del equipo de desarrollo.
- Gestionará los permisos sobre los expedientes de forma automática. Cualquier usuario autorizado puede ser capaz de conceder los permisos a los documentos del proyecto sin tener que crear un script para esto o tener conocimientos de programación.
- Permitirá adicionar carpetas y documentos al sistema para que conformen parte de la estructura del expediente de manera sencilla en el proceso de creación o edición de este.

Con la realización del presente módulo a desarrollar se obtendrá un medio capaz de cumplir con el objetivo perseguido.

## 2.2 Modelo conceptual

Un modelo conceptual, es una representación visual en forma de diagrama de las clases conceptuales u objetos del mundo real que son significativos en un dominio de interés; se trata de un conjunto de diagramas que describen clases u objetos de software con responsabilidades[17].

### 2.2.1 Diagrama del modelo del dominio

La Figura 1 expuesta a continuación muestra el modelo del dominio. El mismo representa la relación entre los conceptos fundamentales del entorno donde se desarrollará la propuesta de solución.

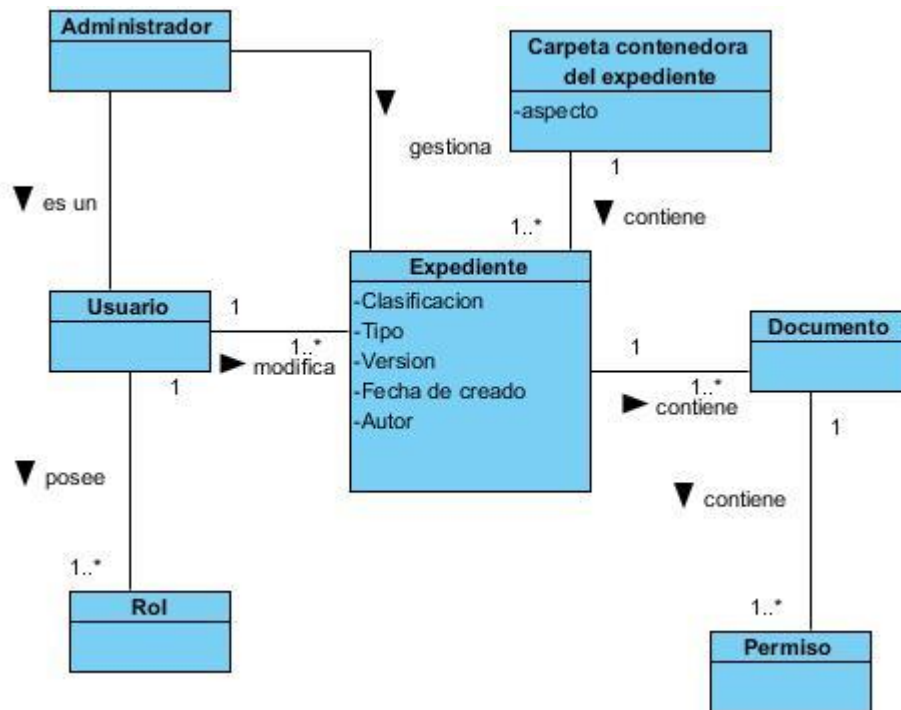


Figura 1: Modelo del dominio

#### 2.2.1.1 Descripción de los conceptos representados en el diagrama

Para lograr una mejor comprensión del dominio del negocio se hace necesaria la descripción de los siguientes conceptos:

- **Usuario:** Es la persona que interactúa con el sistema luego de registrarse en este.
- **Administrador:** Usuario responsable de gestionar y controlar una empresa o entidad.
- **Expediente:** conjunto de los documentos que corresponden a una determinada cuestión.
- **Documento:** archivo electrónico de soporte de información.
- **Rol:** agrupación de responsabilidades que pueden desempeñar los usuarios.



- **Permiso:** reglas otorgadas sobre los roles que definen las acciones que pueden realizar los usuarios sobre el sistema.
- **Carpeta contenedora del expediente:** carpeta donde se encuentra ubicado todo el contenido del expediente.

## 2.3 Requisitos de software

Según el profesor Ian Sommerville: “Un requisito es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de éste. En el otro extremo, es una definición detallada y formal de una función del sistema.”[26]

### 2.3.1 Técnicas utilizadas para la obtención de requisitos

Dentro de las técnicas de obtención de requisitos existentes se utilizaron para la realización de esta investigación las siguientes:

- **Entrevistas:** Se realizaron entrevistas al equipo de desarrollo del GDA eXcriba en las cuales se hicieron una serie de preguntas con el objetivo de obtener toda la información posible sobre las funcionalidades del módulo a desarrollar.
- **Análisis de otros sistemas:** Mediante esta técnica se pudieron estudiar sistemas de gestión de expedientes similares al que se desea desarrollar. El estudio del funcionamiento, características, ventajas y desventajas de estos sistemas aportó en la identificación de los requisitos.

### 2.3.2 Requisitos funcionales.

Es fundamental conocer los Requisitos Funcionales (RF) para poder identificar qué debe hacer el sistema y entender su funcionamiento, por lo que un RF es una capacidad o condición que el sistema debe cumplir [17].

A continuación, se muestra una tabla referente a los requisitos funcionales con la complejidad y prioridad correspondiente a cada uno.

*Tabla 2: Requisitos funcionales.*

Requisitos	Complejidad	Prioridad
1. RF Crear tipo expediente	Alta	Alta
2. RF Modificar tipo expediente	Alta	Alta
3. RF Eliminar tipo expediente	Alta	Alta
4. RF Mostrar tipo expediente	Alta	Alta

5. RF Asignar rol a un miembro del proyecto	Alta	Alta
6. RF Eliminar rol a un miembro del proyecto	Alta	Alta
7. RF Modificar rol a un miembro del proyecto	Alta	Alta
8. RF Mostrar rol de un miembro del proyecto	Alta	Alta
9. RF Asignar permisos a un documento	Alta	Alta
10. RF Eliminar permisos a un documento	Alta	Alta
11. RF Modificar permisos a un documento	Alta	Alta
12. RF Mostrar permisos del documento	Alta	Alta
13. RF Añadir contenido del expediente	Alta	Alta
14. RF Eliminar contenido del expediente	Alta	Alta
15. RF Mostrar contenido del expediente	Alta	Alta
16. RF Modificar contenido del expediente	Alta	Alta
17. RF Mostrar autor del expediente	Alta	Alta
18. RF Mostrar nombre del expediente	Alta	Alta
19. RF Mostrar fecha del expediente	Alta	Alta
20. RF Mostrar estado del expediente	Alta	Alta
21. RF Mostrar clasificación del expediente	Alta	Alta
22. RF Mostrar versión del expediente	Alta	Alta
23. RF Mostrar acciones del expediente	Alta	Alta
24. RF Filtrar por clasificación	Media	Media
25. RF Filtrar por estado	Media	Media
26. RF Filtrar por tipo	Media	Media
27. RF Filtrar por versión	Media	Media
28. RF Filtrar por autor	Media	Media
29. RF Filtrar por fecha	Media	Media
30. RF Filtrar por nombre	Media	Media
31. RF Crear expediente	Alta	Alta
32. RF Modificar expediente	Alta	Alta
33. RF Eliminar expediente	Alta	Alta
34. RF Mostrar expediente	Alta	Alta
35. RF Imprimir expediente	Media	Media
36. RF Exportar expediente	Media	Media

### 2.3.3 Requisitos no funcionales.

Los requisitos no funcionales (RnF), son requisitos que imponen restricciones en el diseño o la implementación como restricciones en el diseño o estándares de calidad. Son propiedades o cualidades que el producto debe tener. Estos están vinculados a los RF, es decir, una vez que se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido debe ser [17].

A continuación, se definen los requisitos no funcionales del sistema.

- **Requisitos de Software:**

**1.RnF** El cliente debe contar con un navegador web (Mozilla Firefox 56.0 o superior, Google Chrome 71.0 o superior)

**2.RnF** El servidor de base de datos debe ser PostgreSQL 9.4 o superior.

**3.RnF** Se puede utilizar el módulo en sistemas operativos Windows y GNU/Linux. Se recomienda utilizar los de GNU/Linux para una mayor rapidez.

- **Requisitos de Hardware:**

**4.RnF** El ordenador del cliente debe poseer un Procesador Intel Core i3 o superior y 4 GB de Memoria RAM, como mínimo.

**5.RnF** La computadora del servidor debe poseer un Procesador Intel Core i5 o superior y una RAM de 8GB o superior.

- **Seguridad y confiabilidad:**

**6.RnF** El módulo debe garantizar la integridad y confidencialidad de la información de los expedientes.

**7.RnF** El módulo debe verificar que el usuario esté autenticado antes de que pueda realizar alguna acción sobre el sistema.

**8.RnF** Se establecerán roles para acceder al sistema, garantizando que la información almacenada solo podrá ser modificada y/o visualizada por los usuarios autorizados.

- **Usabilidad:**

**9.RnF** Idioma: utilizar el idioma español para los mensajes y textos de la interfaz.

**10.RnF** Se elaborará un manual de usuario para el módulo.

- **Disponibilidad:**

**11.RnF** El módulo debe estar disponible las 24 horas del día, excepto cuando se encuentre en mantenimiento.

- **Fiabilidad:**

**12.RnF** La precisión y exactitud de las salidas del sistema se corresponden con la calidad y exactitud de la información contenida en las bases de datos y de la información introducida por los usuarios.

## **2.4 Casos de uso del sistema.**

Un Caso de Uso (CU) especifica una secuencia de acciones que el sistema puede llevar a cabo. El diagrama de CUS se utiliza para ilustrar los requisitos del sistema. Además, es una representación gráfica de los procesos y su interacción con los actores[17].

### **2.4.1 Definición de los actores.**

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados[17]. A continuación, se describen los actores de la aplicación a desarrollar.

- **Usuario:** Individuo que interactúa con el módulo y puede realizar diferentes opciones sobre este dependiendo de los permisos que posea.
- **Administrador:** Usuario con todos los permisos sobre la aplicación. Se encarga de gestionar los usuarios y sus roles, así como modificar toda la información mostrada en el sistema.

### **2.4.2 Definición de casos de uso del sistema**

Los RF definidos anteriormente en el epígrafe 2.3.2 se agrupan en 7 CU, los cuales se muestran a continuación:

- Administrar permisos.
- Administrar roles.
- Gestionar expedientes.
- Gestionar tipos de expedientes.
- Gestionar contenido del expediente.
- Mostrar filtros.
- Mostrar detalles.

En el siguiente diagrama de caso de uso del sistema se puede observar el comportamiento y la comunicación dentro del módulo de gestión de expedientes mediante las diferentes interacciones que realizan los actores, siendo estos los usuarios y administradores.

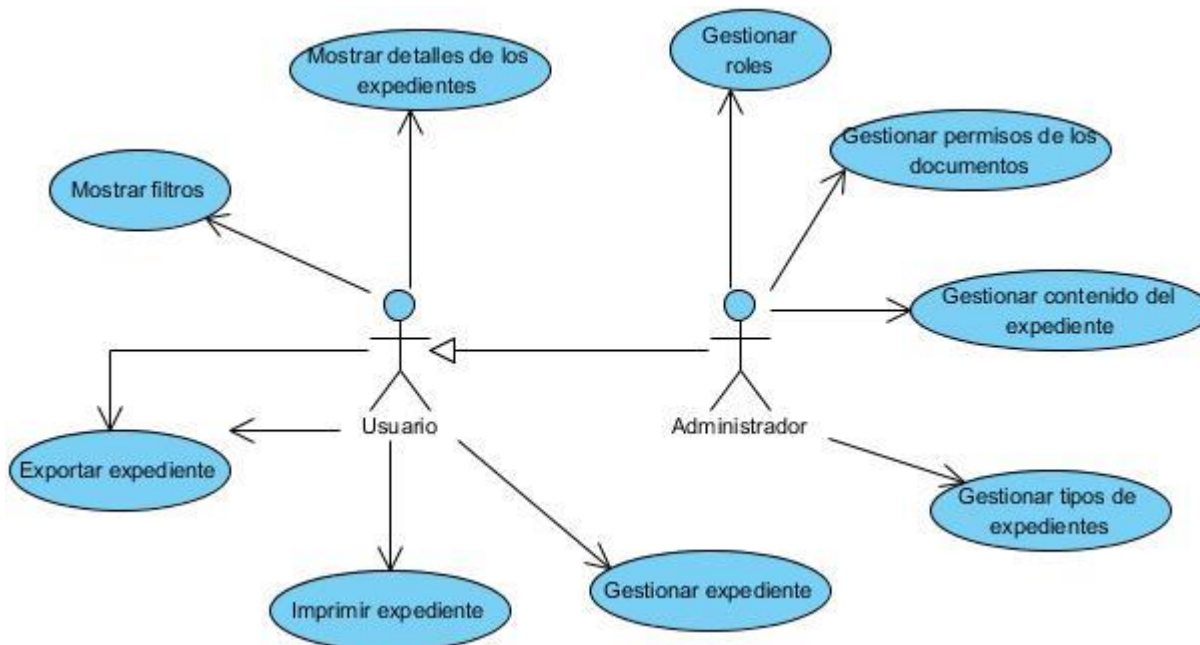


Figura 2: Diagrama de casos de uso del sistema

### 2.4.3 Descripción de casos de uso del sistema

Una vez modelado el diagrama de CUS se describen cada uno de los casos de uso que integran dicho diagrama. El anexo 1 muestra la descripción del CU “Gestionar tipo de expediente”.

A continuación, se muestra en la figura 3 la matriz de trazabilidad requisitos funcionales con su correspondencia a los casos de uso que se asocian a los mismos. Permitiendo así agruparlos para una mayor organización y comprensión y evitando que se queden requisitos sin implementar. En el anexo 2 se puede observar la matriz de trazabilidad de RF-RF.

(9) Use Case	(36) Requirement	Mostrar versión del expediente	Asignar permisos a un documento	Asignar rol a un miembro del proy...	Añadir contenido del expediente	Crear expediente	Crear tipo expediente	Eliminar contenido del expediente	Eliminar expediente	Eliminar permisos a un documento	Eliminar rol a un miembro del proy...	Eliminar tipo expediente	Exportar expediente	Filtrar por autor	Filtrar por clasificación	Filtrar por estado	Filtrar por fecha	Filtrar por nombre	Filtrar por tipo	Filtrar por versión	Imprimir expediente	Modificar contenido del expediente	Modificar expediente	Modificar permisos a un documento	Modificar rol a un miembro del pr...	Modificar tipo expediente	Mostrar acciones del expediente	Mostrar autor del expediente	Mostrar clasificación del expediente	Mostrar contenido del expediente	Mostrar estado del expediente	Mostrar expediente	Mostrar fecha del expediente	Mostrar nombre del expediente	Mostrar permisos del documento	Mostrar rol de un miembro del pr...	Mostrar tipo expediente							
Exportar expediente													✓																															
Gestionar contenido del ex...				✓				✓														✓								✓														
Gestionar expediente					✓				✓														✓										✓											
Gestionar permisos de los ...		✓								✓													✓																					
Gestionar roles			✓								✓													✓																				
Gestionar tipos de expedie...						✓						✓													✓																			
Imprimir expediente																					✓																							
Mostrar detalles de los ex...	✓																										✓	✓	✓		✓		✓	✓	✓									
Mostrar filtros														✓	✓	✓	✓	✓	✓	✓																								

Figura 3: Matriz de trazabilidad requisitos contra casos de uso del sistema

## 2.5 Arquitectura del sistema

Una arquitectura de software define la forma de trabajar en un sistema, implica una solución estructurada que satisfaga todos los requisitos técnicos y operacionales y, a la vez, optimizar los atributos comunes de calidad como rendimiento, seguridad y capacidad de administración. Además, implica una serie de decisiones basadas en una amplia gama de factores, y cada una de esas decisiones puede tener un considerable impacto sobre la calidad, rendimiento, mantenimiento y éxito general de ese software[17].

### 2.5.1 Arquitectura en capas

La arquitectura basada en capas o n-capas proporciona una gran cantidad de beneficios para las empresas que necesitan soluciones flexibles y viables para resolver complejos problemas inmersos en cambios constantes. Esta arquitectura se define como una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior. Desde un punto de vista conceptual es un modelo para construir sistemas de información, que se sustenta en la idea de repartir el tratamiento de la información y los datos por todo

el sistema informático, permitiendo mejorar el rendimiento del sistema global de información[17]. En la figura 4 se aprecia el diagrama de arquitectura del módulo en cuestión.

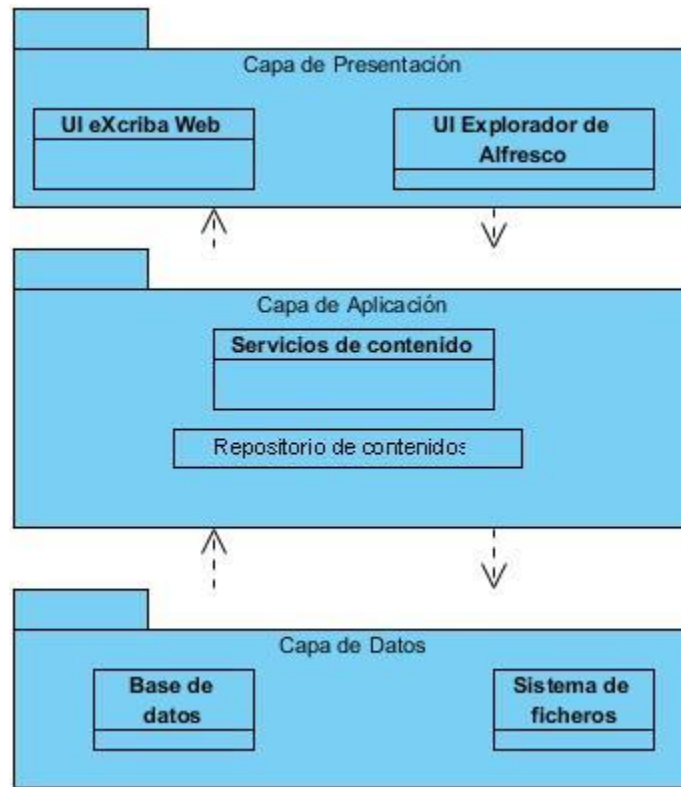


Figura 4: Diagrama de arquitectura

**Capa de Presentación:** En esta capa se brindan dos clientes, el cliente web eXcriba y el explorador de Alfresco. eXcriba es quien interactúa directamente con el usuario, captura la información entrada por este y hace las peticiones a la capa inferior, mostrando al usuario la respuesta proveniente de esta. Únicamente se comunica con la capa de Aplicación mediante el protocolo https. El explorador de Alfresco es incluido en el núcleo como una herramienta de administración de apoyo, ya que expone todas las funcionalidades del ECM utilizado y es requerido para llevar a cabo acciones más avanzadas como el mantenimiento de las reglas de negocio y la gestión de grupos y usuarios[27].

**Capa de Aplicación:** está conformada por los paquetes que integran el sistema, los cuales se ajustan a las descripciones de casos de uso y a los requisitos. Se comunica con la capa de Datos y con la de presentación mediante el protocolo https. Los servicios de datos que esta brinda son incluidos dentro del repositorio de Alfresco[27].

**Capa de Datos:** esta capa se implementa encima de la API remota que brinda el repositorio de contenidos de Alfresco, es la encargada de interactuar directamente con el repositorio de contenidos, permitiéndole a la capa de Aplicación abstraerse de la forma en que deben persistir los datos y cómo

deben ser recuperados. En esta capa se guarda toda la información y el contenido que es gestionado por el núcleo cuando realiza sus funciones[27].

## **2.6 Patrones de diseño de software**

Los patrones de diseño son descripciones de clases y objetos relacionados que están particularizados para resolver un problema de diseño general en un determinado contexto. Estos a su vez identifican las clases, instancias, roles, colaboraciones y la distribución de responsabilidades. Entre los patrones de diseño más utilizados se encuentran los Patrones Generales de Software para Asignación de Responsabilidades (GRASP, del inglés: *General responsibility assignment software patterns*) y los patrones Pandilla de los Cuatro (GoF, del inglés: *Gang of four*) [28].

### **2.6.1 Patrones GRASP**

Los patrones de asignación de responsabilidades describen los principios fundamentales del diseño de objetos para la asignación de responsabilidades [28]. Teniendo en cuenta el patrón arquitectónico seleccionado, los patrones de diseño que se aplican durante el desarrollo del módulo de gestión de expedientes son los expuestos a continuación.

#### **2.6.1.1 Bajo acoplamiento**

El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo acoplamiento no depende de muchas otras. El bajo acoplamiento soporta el diseño de clases más independientes, que reducen el impacto de los cambios, que son más reutilizables y más fáciles de entender [29]. En todas las clases del módulo se ve reflejado el bajo acoplamiento ya que estas se comunican con un número pequeño de clases.

#### **2.6.1.2 Alta cohesión**

En la perspectiva del diseño orientado a objetos, la cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas, que no realicen un trabajo enorme y que además colaboran con otras clases para llevar a cabo las tareas. Una clase con baja cohesión hace muchas cosas no afines o un trabajo excesivo. Una clase con mucha cohesión es útil porque es bastante fácil darle mantenimiento, entenderla y reutilizarla, además a menudo genera un bajo acoplamiento [29]. En la Figura 5, se ilustra como la clase "Usuario" delega la responsabilidad de visualizar los miembros de un proyecto al método buscarPersona(), evidenciando el uso de alta cohesión, en cada clase solo se implementaron las funcionalidades correspondientes a esta.



```

function buscarPersona(person){

    if (people.getPerson(person)){
        return true; }
    return false;
}
try {
model.statususer = buscarPersona(json.get('nombreUsuarioTecnico'));

} catch (e) {
    model.error = e.message;
}
}

```

Figura 5: Ejemplo del patrón Alta cohesión.

### 2.6.1.3 Controlador

Este patrón se evidencia en el archivo datalist que está contenido dentro de cada paquete de la aplicación, el cual funciona como un controlador que atiende los distintos eventos del sistema. Este recibe información proveniente del modelo y decide qué datos enviar a la plantilla para su posterior organización e interacción con el usuario.

```

function getInformationFromDataList(dataList)
{
    var node, result = {}, rol;
    for(var i in dataList.children)
    {
        node = dataList.children[i];
        var roles = node.properties['listRoles'];
        if (node.assocs['list:member'] != null)
        {
            var member = node.assocs['xfpm-dl:member'][0];
            for(var r in roles)
            {
                rol = roles[r];
                if(result[rol])
                {
                    result[rol].push(member.nodeRef);
                }
                else
                {
                    result[rol] = [member.nodeRef];
                }
            }
            node.remove();
        }
    }
}

```

Figura 6: Ejemplo del patrón Controlador.

### 2.6.1.4 Experto

Consiste en asignar una responsabilidad al experto en información, es la clase que cuenta con la información necesaria para cumplir la responsabilidad. Si esto se hace de forma adecuada, los sistemas

tienden a ser más fáciles de entender, mantener y ampliar, y se presenta la oportunidad de reutilizar los componentes en futuras aplicaciones [17]. A continuación, se evidencia una de las funciones de la clase expediente donde se ve esto de manifiesto ya que la misma utiliza su propia información para llevar a cabo sus tareas.

```
function getNodo(uuid) {
    var referenceType = "node";
    var reference = ["workspace", "SpacesStore", uuid];
    var node = search.findNode(referenceType, reference);
    return node;
}

try {
    var nombre_exp = json.get('nombre_exp');
    var tipo = json.get('tipo');
    var clasificacion = json.get('clasificacion');
    var autor = json.get('autor');
    var version = json.get('version');
    var uuidExpedienteMod = json.get('uuidExpedienteMod');

    if (tipoAccionExpediente == "modificar") {
        var expediente = getNodo(uuidExpedienteMod);
    } else {
        var expediente = nodoConsultas.createNode(titulo, 'exp:expediente', []);
    }
}
```

Figura 7: Ejemplo del patrón Experto.

## 2.6.2 Patrones GoF

Los patrones GoF son agrupados en tres grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento. Los de creación abstraen el proceso de creación de instancias. Los estructurales se ocupan de cómo clases y objetos son utilizados para componer estructuras de mayor tamaño y los de comportamiento atañen a los algoritmos y a la asignación de responsabilidades entre objetos[28]. En el desarrollo del módulo de gestión de expedientes se hace uso de los patrones siguientes:

### 2.6.2.1 Decorador

Este patrón posibilita añadir responsabilidades adicionales a un objeto de forma dinámica. El sistema cuenta con la clase que contiene todas las funcionalidades adicionales o decoradores que realizarán determinada acción en correspondencia con el objeto ejecutado y sus entradas[26]. Un ejemplo del uso de este patrón en el módulo, lo constituye la clase expediente, que tiene la responsabilidad de listar todos los documentos y cuenta con el decorador que verifica si el usuario tiene permiso para acceder a este.

```

function applyPerms(node, datalist, datalistName){
//Obtener la informacion de la Lista de Miembros y los contenidos del expediente.
var assigns = getInformationFromPMDDataList(datalist),
    contents = search.luceneSearch("(" + PATH:"" + node.qnamePath + "/*" + TYPE:"cm:content");

logger.log("----- ASIGNANDO PERMISOS -----");
var personNode, membersBoss = [],
    siteName = datalist.parent.name,
    siteAbbreviation = datalist.parent.properties["gep:abbreviation"].toUpperCase();
//Obtener los miembros que tienen permisos sobre el expediente
membersBoss = !assigns['Jefe de proyecto'] ? membersBoss : membersBoss.concat(assigns['Jefe de proyecto']);
}

```

Figura 8: Ejemplo del patrón Decorador.

### 2.6.2.2 Observador

Define una dependencia “uno a muchos” entre objetos, para que cuando uno de ellos cambie su estado, todos los que dependan de él sean avisados y puedan actualizarse convenientemente[26]. Este patrón se evidencia en la clase carpeta contenedora del expediente ya que cuando se realiza una modificación sobre el aspecto de la misma se envía un mensaje sobre tipo de documento que puede ser asignado a esta.

```

68 //Verificar aspecto
69 function folderNodeAspect() {
70
71     var props = new Array(1);
72     props["cm:template"] = fileNode.nodeRef;
73     document.addAspect("cm:templatable", props);
74     copy.addAspect("cm:templatable", null);
75     copy.properties["cm:template"] = fileNode.nodeRef;
76     copy.save();
77     if (copy.hasAspect("cm:templatable") && copy.hasPermission("Write"))
78     {
79         copy.name = "templatable " + copy.name;
80         copy.save();
81     }
82 }

```

Figura 9: Ejemplo del patrón Observador.

## 2.7 Validación de la propuesta de solución a través de métricas del diseño.

Una métrica es un instrumento que permite evaluar el software al inicio del proceso, que cuantifica además un criterio y persigue comprender mejor la calidad del producto, estimar la efectividad del proceso y mejorar la calidad del trabajo realizado al nivel de proyecto. La aplicación de métricas al diseño de un producto de software constituye un elemento fundamental a la hora de evaluar la calidad del mismo. Con el fin de desarrollar un diseño robusto y sencillo se

realizó la validación del mismo utilizando las métricas Tamaño operacional de clase (TOC) y Relaciones entre clases (RC)[30].

### 2.7.1 Tamaño operacional de clase (TOC)

Está dado por el número de métodos asignados a una clase y evalúa los siguientes atributos de calidad:

- Responsabilidad: Un aumento del TOC implica un aumento de la responsabilidad asignada a la clase.
- Complejidad de implementación: Un aumento del TOC implica un aumento de la complejidad de implementación de la clase.
- Reutilización: Un aumento del TOC implica una disminución del grado de reutilización de la clase.

### 2.7.2 Relaciones entre clases (RC)

Está dado por el número de relaciones de uso de una clase con otra y evalúa los siguientes atributos de calidad:

- Acoplamiento: Un aumento del RC implica un aumento del Acoplamiento de la clase.
- Complejidad de mantenimiento: Un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.
- Reutilización: Un aumento del RC implica una disminución en el grado de reutilización de la clase.
- Cantidad de pruebas: Un aumento del RC implica un aumento de la cantidad de pruebas de unidad necesarias para probar una clase.

Tabla 3: Clases y relaciones de uso (RC).

Clases	Cantidad de relaciones de uso
Administrador	2
Usuario	3
Rol	1
Expediente	4
Documento	2
Carpeta contenedora del expediente	1
Permiso	1
<b>Total</b>	<b>14</b>

Tabla 4: Clases y cantidad de procedimientos (TOC).

Clases	Cantidad de procedimientos
Administrador	3
Usuario	10
Rol	4
Expediente	8
Documento	3
Carpeta contenedora del expediente	2
Permiso	4
<b>Total</b>	<b>34</b>

Tabla 5: Atributos que posibilita medir las métricas TOC y RC.

Atributos de calidad	Descripción
<b>Acoplamiento</b>	Dependencia o interconexión de una clase o estructura de clase respecto a otra
<b>Complejidad Mantenimiento</b>	Nivel de esfuerzo necesario para sustentar, mejorar o corregir el diseño de software propuesto. Puede influir significativamente en los costes y la planificación del proyecto.
<b>Reutilización</b>	Significa cuán reutilizada es una clase o estructura de clase dentro de un diseño de software
<b>Cantidad de Pruebas</b>	Número o grado de esfuerzo necesario para realizar las pruebas de calidad al producto diseñado.
<b>Responsabilidad</b>	Responsabilidad que posee una clase en un marco conceptual correspondiente al modelado de la solución propuesta
<b>Complejidad Implementación</b>	Grado de dificultad que tiene implementar un diseño de clases determinado

Para la evaluación de las clases fueron utilizados umbrales para el acoplamiento, complejidad de mantenimiento, la reutilización y cantidad de pruebas. (Ver anexos 6 y 7 referentes a los atributos de calidad para las pruebas y los umbrales de prueba utilizados.)

## 2.8 Resultado de las métricas.

Luego de aplicarse las métricas de diseño RC y TOC obtenidos los resultados de la evaluación del instrumento de medición de la métrica, se puede concluir que el diseño propuesto tiene una

calidad aceptable teniendo en cuenta que aproximadamente el 87.05% de las clases empleadas poseen menos de 3 dependencias de otras clases lo que lleva a evaluaciones positivas de los atributos de calidad involucrados (bajo acoplamiento con 36.14%, baja complejidad de mantenimiento con un 58.85%, baja cantidad de pruebas con 56.52%, alta reutilización con un 77.15% y una baja complejidad de implementación con 46,70%). Favoreciendo de esta manera la reutilización de las clases, así como la modificación e implantación del diseño. Lo planteado anteriormente se ve reflejado en las figuras 10 y 11.

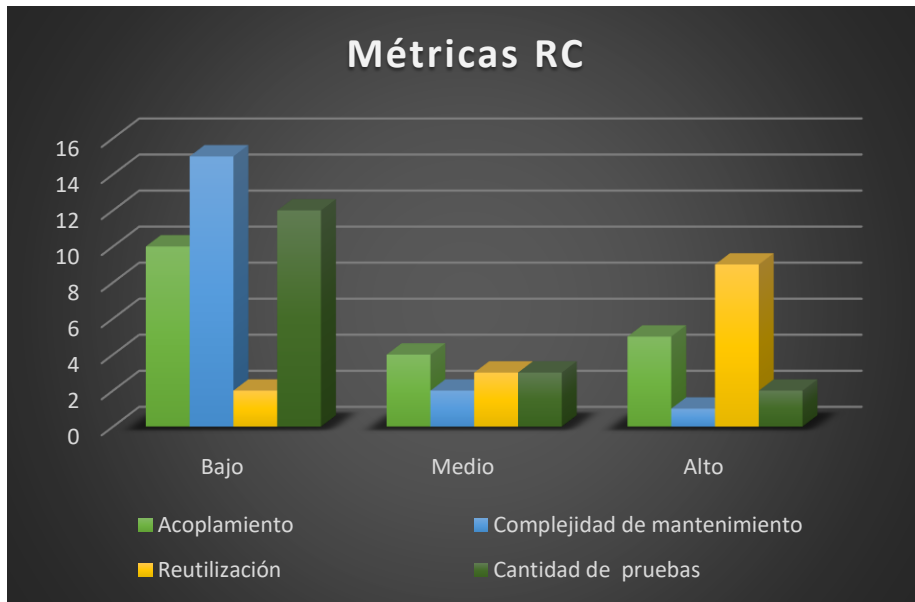


Figura 10: Métricas RC.

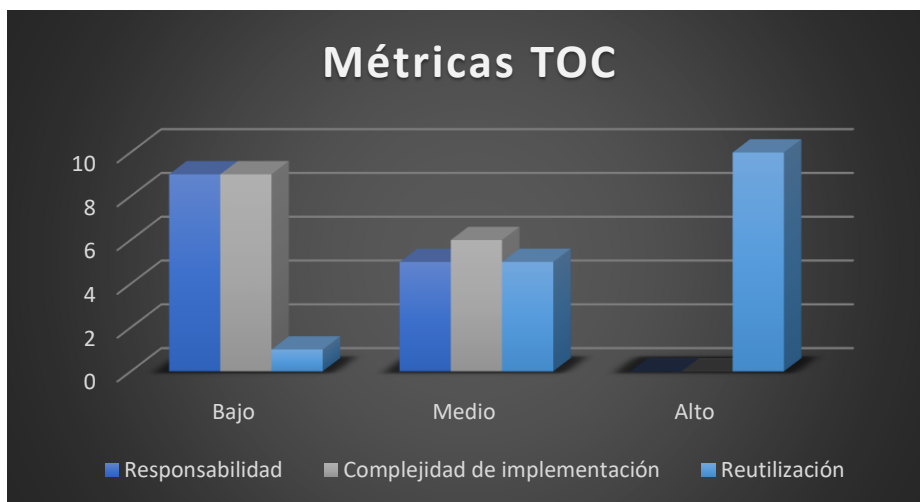


Figura 11: Métricas TOC.

## 2.9 Conclusiones del capítulo

A partir de los resultados presentados en este capítulo del análisis y diseño del módulo, teniendo en cuenta los objetivos específicos de la investigación y haber generado los diferentes artefactos que dispone la metodología AUP-UCI, quedan reflejadas las bases para la posterior implementación del módulo de gestión de expedientes genéricos, resaltando que:

- La especificación de los requisitos funcionales y no funcionales contribuyó a una mejor organización del sistema a partir de las entrevistas y el análisis de los sistemas homólogos, sin descuidar la tecnología previamente seleccionada.
- La descripción de CUS y los diagramas de clases de diseño a partir del lenguaje de modelado UML orientaron efectivamente la implementación del sistema, su documentación y comportamiento acorde a los requisitos.
- La aplicación de patrones de diseño GRASP y GoF, evidencian la utilización de buenas prácticas durante el desarrollo del sistema, lo cual repercute positivamente en la calidad final del mismo.
- La aplicación de las métricas de diseño permitió medir de forma cuantitativa la calidad de los atributos internos del módulo, brindando una referencia de cuál es el por ciento de cumplimiento que tiene el estas características, ayudando así a la toma de decisiones y a mejorar la calidad del desarrollo del proceso de pruebas y del producto.

## **Capítulo 3: Implementación y validación de la solución**

---

En el presente capítulo se muestran los resultados alcanzados en la implementación del sistema, así como los estándares de codificación empleados y algunas interfaces de la aplicación. Se evidencia la organización del sistema mediante el diagrama de componentes el cual representa la vista estática del mismo, y mediante el diagrama de despliegue se representa cómo se distribuirán los servicios que brindará el sistema. Por último, se describirán las pruebas realizadas al sistema y los resultados obtenidos una vez aplicadas las mismas; además de la validación de la investigación.

### **3.1 Modelo de implementación.**

Entre las disciplinas propuestas por la metodología de desarrollo seleccionada, AUP-UCI, se encuentra la de Implementación. En la implementación, a partir de los resultados del Análisis y Diseño se construye el sistema. El objetivo de esta disciplina es lograr la implementación de las clases y subsistemas que fueron encontrados durante el diseño, así como definir la organización del código[31].

El modelo de implementación está conformado por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Fundamentalmente, se describe la relación que existe desde los paquetes y clases del modelo de diseño a subsistemas y componentes físicos[26]. El modelo de implementación muestra, a través del diagrama de despliegue, la estructura del sistema en ejecución y a través del diagrama de componentes, las dependencias entre las partes de código del sistema.

### **3.2 Diagrama de componentes**

El término de componente se define como una parte modular, desplegable y reemplazable de un sistema que encapsula implementación y expone un conjunto de interfaces. Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Los componentes representan todos los tipos de elementos de software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes y bibliotecas cargadas dinámicamente[17]. La Figura 12 muestra el diagrama de componentes para el caso de uso “Gestionar expedientes”.



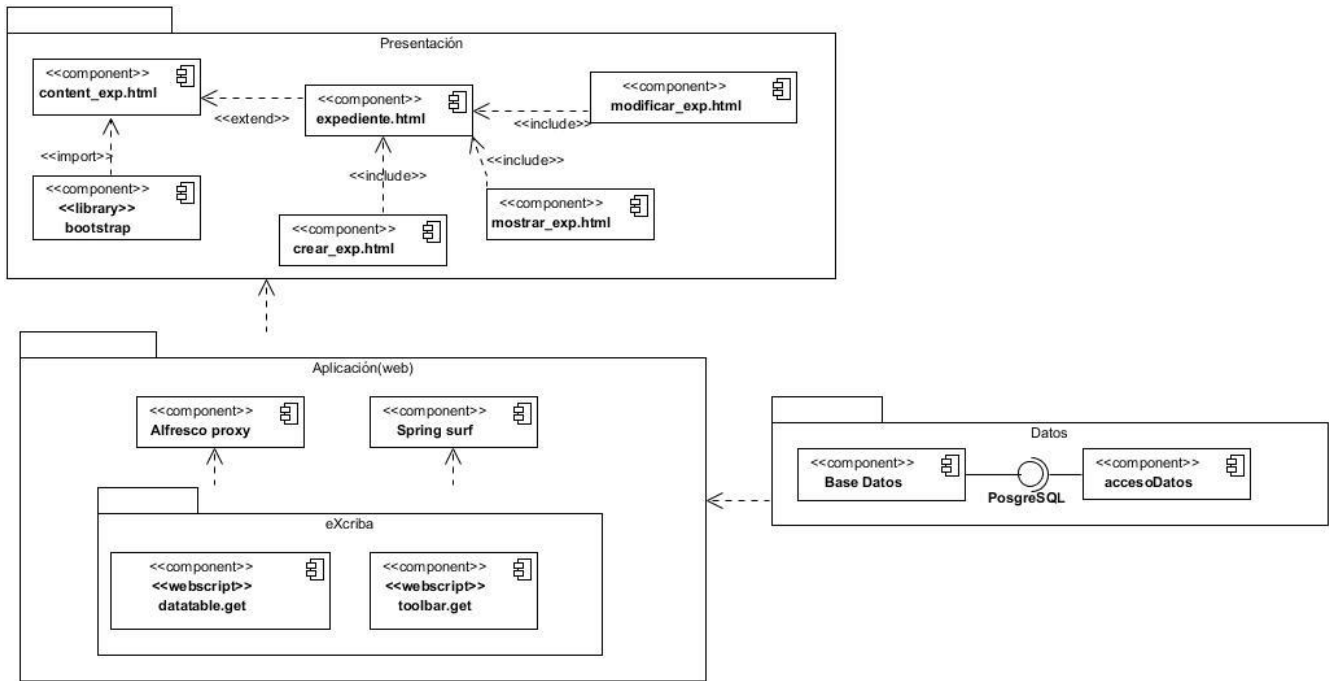


Figura 12: Diagrama de componentes

En el diagrama de componentes mostrado anteriormente se evidencian los componentes `datatable.get` y `toolbar.get` los cuales son parte fundamental del `eXcriba` y son los encargados de controlar las acciones que se realizan sobre los registros de los expedientes y estos acceden a los servicios del `Alfresco` y `Spring surf`. Los componentes anteriores se encuentran conectados a la capa de datos y utilizan el acceso a datos para la gestión de expedientes a nivel de base de datos con el `PosgreSQL`. Además, se encuentra el componente `expediente.html` que es el encargado de construir la clase. Esta se extiende del `content_exp.html` para su completa construcción y visualización por el usuario e incluye a `crear_exp.html`, `modificar_exp.html` y `mostrar_exp.html`. El componente `content_exp.html` contiene lo general de las plantillas, por lo que importa la librería de `bootstrap` utilizada para la visualización del módulo.

### 3.3 Modelo de Despliegue.

El modelo de despliegue consta de uno o más nodos, dispositivos (nodos estereotipados sin capacidad de proceso en el nivel modelado de abstracción) y conectores entre nodos, y entre nodos y dispositivos. El modelo de despliegue también correlaciona procesos con los elementos de proceso, permitiendo la distribución de comportamiento entre los nodos que se deben representar[17].

Un modelo de despliegue puede tener las siguientes propiedades:

- **Introducción:** Una descripción textual que sirve como breve introducción al modelo.
- **Nodos:** Elementos de proceso en el sistema.
- **Dispositivos:** Dispositivos físicos, que no tienen ninguna capacidad de proceso (al nivel modelado de abstracción), que dan soporte a los nodos de procesador. Los dispositivos pueden tener las siguientes propiedades:
- **Conectores:** Conexiones entre nodos, y entre nodos y dispositivos. Los conectores pueden tener información asociada en relación con la capacidad o ancho de banda del conector.

### **3.3.1 Métodos para el despliegue de extensiones de Alfresco.**

Existen varias maneras para realizar un despliegue de una extensión para Alfresco, el método concreto depende de la complejidad de desarrollo y del servidor de aplicaciones que se utilice. Así, por ejemplo, si la personalización consiste solamente en cambios en el fichero de configuración o propiedades bastaría con colocarlos en el classpath del servidor de aplicaciones. A continuación, se plantean algunos de los métodos de despliegue de extensiones de Alfresco [27]:

- Extensión de Alfresco usando librerías JAR.
- Extensiones del Cliente Web.
- Extensiones fuera de librerías JAR.
- Extensión con Módulos AMP.

#### **3.3.1.1 Extensión con Módulos AMP.**

En el desarrollo de módulo de gestión de expedientes se utilizaron extensiones con módulos AMP (Alfresco Module Package) para las nuevas funcionalidades. Los módulos de extensión AMP son ficheros que contienen una colección de código, XML, imágenes, CSS, y otros que extienden la funcionalidad proporcionada por el repositorio de Alfresco. Estos contenidos están distribuidos dentro de una estructura como la siguiente:

```

/
|- /config
|- /lib
|- /licenses
|- /web
|   |- /jsp
|   |- /css
|   |- /images
|   |- /scripts
|- module.properties
|- file-mapping.properties

```

Figura 13: Estructura de extensión con módulos AMP

Esta no es una estructura obligatoria, puede faltar cualquier directorio o estar vacío. Cada uno de ellos será mapeado dentro de un Alfresco WAR usando la Herramienta de Manejo de Módulos (MMT)<sup>8</sup>:

`/config`: El contenido será mapeado dentro del directorio `/WEB-INF/classes` en el fichero WAR. En este directorio se situarán los ficheros que definirán la configuración Spring y la del Interfaz de Usuario. En este directorio debe estar también el fichero `module-context.xml` con la configuración Spring del módulo.

- `/lib`: El contenido será mapeado en el directorio en `/WEB-INF/lib`. Deberá contener los ficheros JAR de nuestro modulo.
- `/licenses`: Si el módulo usa alguna librería que necesite de una licencia, estas deberán estar situadas en este directorio.
- `/web/jsp`: El contenido será mapeado con el directorio `/jsp` del fichero WAR. El directorio debe contener todos los JSP nuevos o modificados del módulo.
- `/web/css`: El contenido será mapeado con el directorio `/css` del fichero WAR. El directorio debe contener las hojas de estilo de nuestro módulo.
- `/web/images`: El contenido será mapeado en el directorio `/images` del WAR. Contendrá las imágenes que use nuestro módulo.
- `/web/scripts`: Los contenidos serán mapeados en el directorio `/scripts` del WAR. Los ficheros con código JavaScript que use el módulo deberán ser puestos aquí.

<sup>8</sup> Module management tool(Herramienta de manejo de módulos)

### 3.3.2 Diagrama de despliegue.

Un diagrama de despliegue permite indicar cómo se ubicarán las funcionalidades y los subsistemas dentro del entorno computacional físico que soportará el software. La vista de despliegue muestra la configuración de los nodos de procesamiento en tiempo de ejecución, los vínculos de comunicación entre ellos y las instancias de los componentes. El propósito del modelo de despliegue es capturar la configuración de los elementos de procesamiento y las conexiones entre estos elementos en el sistema[17]. A continuación, se muestra en la Figura el diagrama de despliegue diseñado para la aplicación, en el cual se evidencia cómo se distribuyen los servicios que brindará el sistema teniendo en cuenta los requisitos no funcionales de software identificados en epígrafes anteriores, destacándose que la comunicación entre los nodos PC-Cliente y el servidor web y el de base de datos con los protocolos de comunicación entre los mismos.

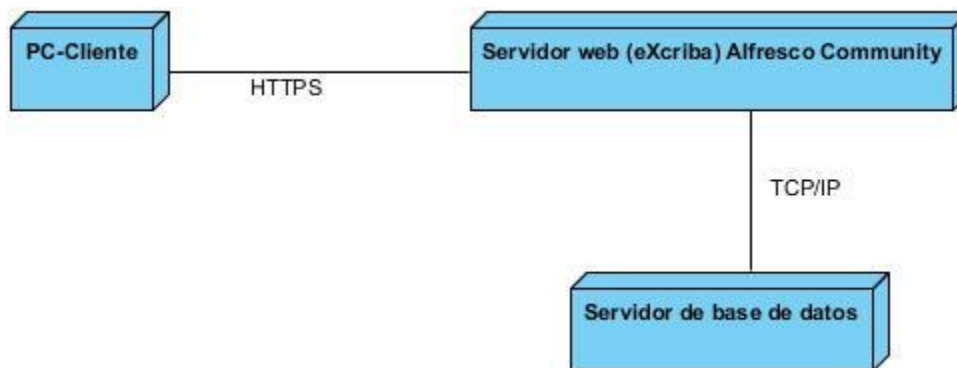


Figura 14: Diagrama de despliegue

**PC Cliente:** computadora en la cual la aplicación se ejecutará y a través de un navegador el usuario accederá a las interfaces del módulo utilizando el protocolo HTTPS.

**Servidor Web:** es donde se encuentran los componentes asociados a la capa presentación, además en este nodo se encuentra alojado el cliente web en un servidor Apache Tomcat 8.5.2, donde se encuentra desplegado el GDA eXcriba 4.0 y el Alfresco Community 5.2.

**Servidor de Base de datos:** servidor de Base de Datos PostgreSQL 9.4, donde se encuentra la base de datos que utiliza el sistema, este puede estar instalado en la misma computadora donde se encuentra el servidor Web.

### 3.4 Estándares de codificación.

Los estándares de codificación son reglas de codificación que permiten tener una programación homogénea, comprendiendo todos los aspectos de la generalización del código, pues la aplicación debe de estar implementada como si un único programador escribiera el código de una sola vez. La usabilidad de estos permite conservar el código fuente entendible y fácil de mantener[26].

#### 3.4.1 Nomenclatura.

El idioma por defecto a la hora de dar sentido funcional al nombre de clases, variables, constantes, etc. será una mezcla entre la nomenclatura tradicional en inglés y la nomenclatura funcional adoptada. Resumiendo, aquella codificación que por estandarización y/o aceptación se pueda escribir en inglés se mantendrá así por convenio, casos como *insert*, *update*, *delete*, *create*, *retrieve*, *list*, *set*, *get*. En el ejemplo de la figura 15 podemos verlo reflejado.

```
function updateNode(node, name, contieneEstructura, contentieneMedio) {  
    node.properties['exp:valorMedio'] = name;  
    node.name = name;  
    node.save();  
    // eliminar asociaciones actuales  
    removeAssocs(node);  
    //agregar asociaciones nuevas  
  
    //agregar las asocciaciones de las estructuras a las que pertenece
```

Figura 15: Ejemplo de estándar de codificación. (Nomenclatura y Comentarios)

#### 3.4.2 Identación y líneas

Usar una Identación<sup>9</sup> sin tabulaciones, con un equivalente a 4 espacios. El uso de las llaves “{}” será en una nueva línea. La longitud de las líneas de código es aproximadamente de 75-80 caracteres. Para mantener la legibilidad del código. Esto se puede evidenciar en la figura 16.

---

<sup>9</sup> Identación: es un anglicismo de la palabra inglesa indentation, de uso común en la informática. En los lenguajes de programación es un tipo de notación secundaria utilizado para mejorar la legibilidad del código fuente por parte de los programadores

```
function getUbicacion(Ubicacion) {  
    //Buscando ubicacion  
    var node = search.findNode(referenceType, reference);  
    return node.name;  
  
}
```

Figura 16: Ejemplo de estándar de codificación. (Indentación)

### 3.4.3 Comentarios.

- Los comentarios serán utilizados para dar información adicional al desarrollador sobre la implementación del diseño de la clase. Por tanto, se tiene que evitar referencias al diseño funcional de la misma.
- El uso abusivo de los comentarios es desaconsejable, principalmente por el trabajo extra necesario para su correcto mantenimiento. Es preferible rediseñar el código para una mejor comprensión del mismo.
- Estándar de Codificación JEE<sup>10</sup>
- Se tienen que evitar el uso de caracteres especiales dentro de los comentarios

### 3.4.4 Buenas prácticas

- Como norma general todas las constantes numéricas no deberían codificarse directamente.
- El acceso/modificación de las propiedades de una clase (no constantes) siempre mediante métodos de acceso get/set.
- La asignación de variables y propiedades no podrá ser consecutiva.
- No utilizar el operador asignación en sitios donde se pueda confundir con el operador igualdad. Ni dentro de expresiones complejas.
- Como norma general no se debe acceder a un método estático desde una instancia de una clase, debemos utilizar la clase en sí misma.

## 3.5 Pruebas de software.

Una vez terminada la implementación del producto que se requiere es necesario realizarle pruebas con el objetivo de detectar errores en la aplicación y la documentación. Las pruebas son un conjunto de actividades que se planean con anticipación y se realizan de manera sistemática, por lo que se debe

---

<sup>10</sup> Java Enterprise Edition: Conjunto de estándares de desarrollo en Java.

definir una estrategia para las pruebas de software que se realizarán, la cual se basa en un conjunto de pasos y procedimientos en que se puedan incluir técnicas y métodos específicos del diseño de casos de prueba[17].

### **3.5.1.1 Estrategia de pruebas.**

Una estrategia de pruebas de software proporciona una guía que describe los pasos que deben realizarse como parte de la prueba, cuándo se planean y se llevan a cabo dichos pasos, y cuánto esfuerzo, tiempo y recursos requieren. Por tanto, cualquier estrategia de prueba debe incorporar la planificación de la prueba, el diseño de casos de prueba, la ejecución de la prueba y la recolección y evaluación de los resultados[17]. Las pruebas que se realizan son:

- Pruebas unitarias (caja blanca).
- Pruebas de funcionales (caja negra).
- Pruebas de rendimiento (carga y estrés).

### **3.5.2 Pruebas unitarias.**

Las pruebas de unidad o unitarias son el proceso de probar componentes del programa tales como métodos o clases de objetos. Las funciones o los métodos individuales son el tipo más simple de componente. Las pruebas deben llamarse para dichas rutinas con diferentes parámetros de entrada[26].

#### **3.5.2.1 Prueba de caja de blanca.**

La prueba de caja blanca también se conoce como prueba de caja transparente o de cristal. Esta prueba consiste específicamente en cómo diseñar los casos de prueba atendiendo al comportamiento interno y la estructura del programa, examinándose la lógica interna sin considerar los aspectos de rendimiento. Dentro de la prueba de caja blanca se incluyen las técnicas de pruebas que serán descritas a continuación[17]:

- Prueba del camino básico: permite obtener una medida de la complejidad lógica de un diseño y usar la misma como guía para la definición de un conjunto de caminos básicos.
- Prueba de condición: ejercita las condiciones lógicas contenidas en el módulo de un programa. Garantiza la ejecución por lo menos una vez de todos los caminos independientes de cada módulo, programa o método.

- Prueba de flujo de datos: Se seleccionan caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa. Garantiza que se ejerciten las estructuras internas de datos para asegurar su validez.
- Prueba de bucles: Se centra exclusivamente en la validez de las construcciones de bucles. Garantiza la ejecución de todos los bucles en sus límites operacionales.

La técnica de caja blanca que se emplea en la solución desarrollada es la Prueba del Camino Básico, la cual permite obtener una medida de la complejidad lógica del diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución, garantizando con estos que durante la prueba se ejecute por lo menos una vez cada sentencia del programa.

Para realizar esta técnica es necesario calcular antes la complejidad ciclomática del algoritmo o fragmento de código a analizar.

```
function removeAllPermission(node)
{
    node.setPermissions(false);
    var permissions = node.getPermissions();
    if (permissions.length > 0)
    {
        for(var i in permissions)
        {
            var temp = permissions[i].split();
            node.removePermission(temp[i]);
        }
    }
    return node;
}
```

Figura 17: código para la prueba de Caja blanca

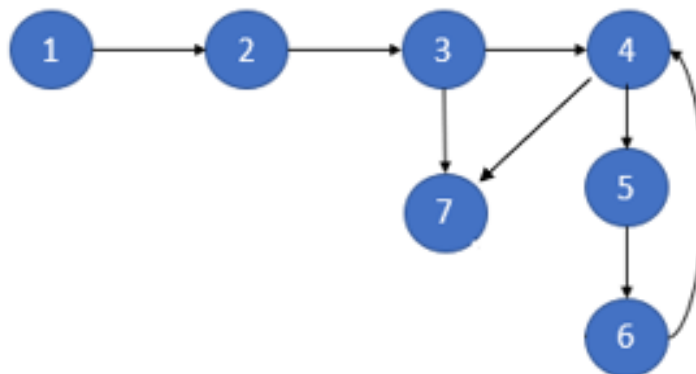


Figura 18: Grafo de la prueba de caja blanca.



### 3.5.2.2 Fórmulas para calcular la complejidad ciclomática

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

Donde "A" es la cantidad de aristas y "N" la cantidad de nodos.

$$V(G) = (8 - 7) + 2$$

$$V(G) = 3$$

- $V(G) = P + 1$

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

$$V(G) = 2 + 1$$

$$V(G) = 3$$

- $V(G) = R$

Donde "R" representa la cantidad de regiones en el grafo.

$$V(G) = 3$$

El cálculo efectuado mediante las fórmulas ha dado el mismo valor, por lo que se puede decir que la complejidad ciclomática del código es de tres, lo que significa que existen 3 posibles caminos por donde el flujo puede circular, este valor representa el límite mínimo del número total de casos de pruebas para el procedimiento tratado.

*Tabla 6 Caminos básicos del grafo de flujo asociado al método removeAllPermisson().*

Número	Caminos básicos
1	1-2-3-4-7
2	1-2-3-7
3	1-2-3-4-5-6-4-7

Posteriormente de haber determinado los caminos básicos se procede a ejecutar los casos de pruebas para cada uno de estos (ver Tablas 7, 8, 9). Para definir los casos de prueba se tiene en cuenta:

- Descripción del proceso: Se describe el caso de prueba y de forma general se tratan los aspectos fundamentales de los datos de entrada.
- Resultados Esperados: Se expone el resultado esperado que debe devolver el procedimiento después de efectuado el caso de prueba.

Tabla 7 Caso de prueba para la trayectoria básica 1 del método de caja blanca

<b>Caso de prueba para la trayectoria básica 1</b>	
<b>Trayectoria</b>	1-2-3-4-7
<b>Descripción del proceso.</b>	Verifica que el arreglo de permisos no se encuentra vacío, entra al for pero el nodo no tiene permisos
<b>Resultado esperado.</b>	Devuelve el nodo en su estado original

Tabla 8 Caso de prueba para la trayectoria básica 2 del método de caja blanca

<b>Caso de prueba para la trayectoria básica 2</b>	
<b>Trayectoria</b>	1-2-3-7
<b>Descripción del proceso.</b>	El arreglo de permisos se encuentra vacío
<b>Resultado esperado.</b>	Devuelve el nodo en su estado original

Tabla 9 Caso de prueba para la trayectoria básica 3 del método de caja blanca

<b>Caso de prueba para la trayectoria básica 3</b>	
<b>Trayectoria</b>	1-2-3-4-5-6-4-7
<b>Descripción del proceso.</b>	Verifica que el arreglo de permisos no se encuentra vacío y elimina los permisos de los nodos.
<b>Resultado esperado.</b>	Devuelve el arreglo de nodos sin permisos

### 3.5.3 Pruebas funcionales.

Las pruebas funcionales están basadas en ejecución, revisión y retroalimentación de las funcionalidades implementadas para el software y las especificaciones definidas por el usuario. Con estas pruebas se busca evaluar el sistema mediante modelos de pruebas. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos. Por ello se denominan pruebas funcionales, y el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro. Las pruebas funcionales que se realizarán a la solución, estarán enfocadas o dirigidas a los CUS para verificar su correcto funcionamiento. En este tipo de pruebas se ejecutan los distintos servicios prestados con datos correctos e incorrectos. En caso de que los datos sean incorrectos se verifica que los mensajes de error sean los deseados y en el caso opuesto que los resultados sean los esperados[17].

#### 3.5.3.1 Prueba de Caja Negra.

Las pruebas de Caja Negra se aplican a la interfaz del software, examinan algún aspecto funcional de un sistema que tiene poca relación con la estructura lógica interna del software. Estas pruebas se concentran en los RF del software, tratando de encontrar los siguientes errores[17]:

- Funciones incorrectas o faltantes.
- Errores de interfaz.

- Errores de estructuras de datos o en acceso a bases de datos externas.
- Errores de comportamiento o desempeño.
- Errores de inicialización y término.

Los casos de pruebas pretenden demostrar que:

- Las funciones del software son operativas.
- La entrada se acepta de forma correcta.
- Se produce una salida correcta.
- La integridad de la información externa se mantiene.

Para el diseño de los casos de pruebas de Caja Negra se utilizó la Técnica de Partición de Equivalencia, la cual divide el dominio de entrada de un programa en clases de datos a partir de las cuales pueden derivarse casos de prueba. La partición equivalente se esfuerza por definir un caso de prueba que descubra ciertas clases de errores, reduciendo así el número total de casos de prueba que deben desarrollarse[17]. A continuación, se presenta el diseño de caso de prueba que se utilizará para comprobar el funcionamiento del sistema.

*Tabla 10 Descripción de variables para el caso de prueba "Gestionar expediente"*

<b>No.</b>	<b>Nombre de campo</b>	<b>Clasificación</b>	<b>Valor Nulo</b>	<b>Descripción</b>
1	Nombre	Campo de texto	No	Contiene el título del expediente (debe comenzar con mayúscula).
2	Clasificación	Lista de selección única	No	En este campo se escogerá entre las opciones de clasificación disponibles.
3	Tipo	Lista de selección única	No	En este campo se escogerá entre los tipos disponibles.
4	Versión	Lista de selección única	No	En este campo se escogerá entre las opciones de versión disponibles.
5	Fecha de creado	Campo de fecha	Sí	En este campo se escoge la fecha en el calendario.
6	Contenidos del expediente	Campo de búsqueda	Sí	Contiene la ruta del directorio donde se cargarán los documentos.
7	Carpeta destino del expediente	Campo de búsqueda	No	Contiene la ruta del directorio donde se guardará el expediente.

### 3.5.3.2 Descripción de Casos de Prueba.

Se evaluó y probó la validez de cada una de las entradas al sistema, utilizando datos válidos (V), inválidos (I), y valores que no son necesarios proporcionar (N/A). En la Figura 19 se muestra el caso de prueba para el caso de uso “Gestionar expediente”, en el escenario “Crear expediente” para tres variantes posibles evidenciando los caminos y respuestas que da el sistema al introducir los datos correctamente, valores inválidos y dejando campos vacíos. En el anexo 8 se puede evidenciar el caso de prueba para el escenario “Modificar expediente”.

Escenario	Descripción	Nombre	Clasificación	Tipo	Versión	Contenido del expediente	Carpeta destino del expediente	Respuesta del sistema	Flujo central
EC1. Adicionar Expediente con los datos correctos.	El usuario introduce los parámetros correctamente y se adicionan los datos al sistema.	Expediente de la secretaria	Laboral	Trabajadores	2.0	N/A	N/A	Adiciona el elemento correctamente.	1. Selecciona la opción crear Expediente 2. Se llenan los datos en la tabla 3. Selecciona la opción aceptar.
EC2. Insertar campos inválidos	El usuario introduce datos inválidos	expediente de usuario	I	I	I	N/A	N/A	Muestra un mensaje de error señalando los campos incorrectos	1. Selecciona la opción crear Expediente 2. No llenan los datos en la tabla 3. Muestra un mensaje de error.
EC3. Dejar campos vacíos	El usuario deja campos vacíos		I	I	I	N/A	N/A	Muestra un mensaje de error diciendo que hay campos obligatorios vacíos.	1. Selecciona la opción crear Expediente 2. No llenan los datos en la tabla 3. Muestra un mensaje de error.

Figura 19: Descripción del caso de prueba “Crear expediente”

### 3.5.3.3 Resultados de las pruebas utilizando el método de caja negra.

Una vez realizadas las pruebas utilizando el método de caja negra mediante la técnica de partición equivalente usando los casos de pruebas asociados para cada CU, se comprobaron las funcionalidades del sistema y la correcta validación de sus campos. Para una primera iteración se detectan un total de doce no conformidades las cuales están distribuidas en cuatro de interfaz, tres de funcionalidad, una ortográfica y dos de validación y diseño. Para una segunda iteración se detectan cinco, siendo estas dos de interfaz y una de funcionalidad, validación y diseño respectivamente, las cuales quedan resueltas para una tercera iteración. Completando la solución de las mismas en un período de 72 horas. La figura a continuación muestra lo planteado anteriormente con el gráfico que representa las iteraciones realizadas y el tipo de no conformidad encontrada.

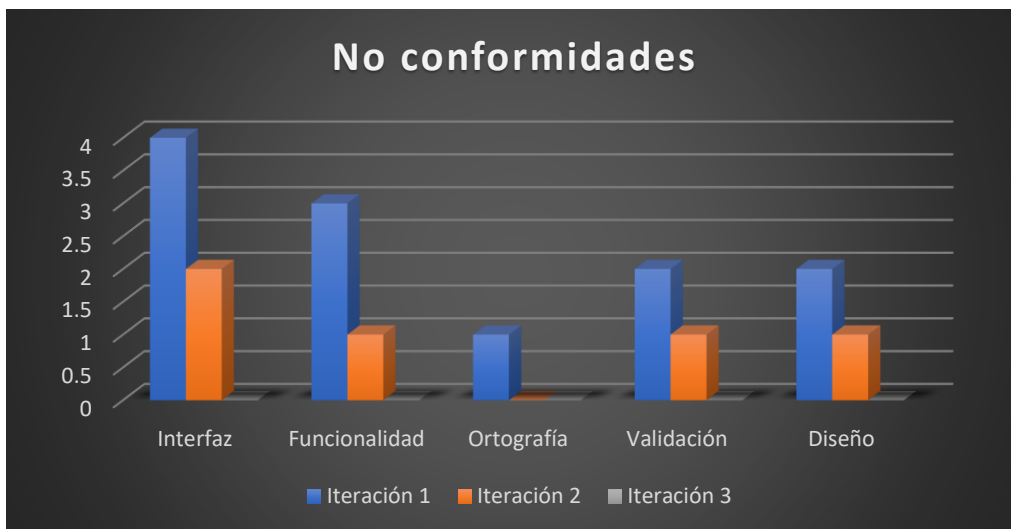


Figura 20: Grafico de las no conformidades.

### 3.5.4 Pruebas de rendimiento.

Las pruebas de rendimiento se realizan para determinar lo rápido que realiza una tarea un sistema en condiciones particulares de trabajo. También puede servir para validar y verificar otros atributos de la calidad del sistema, tales como la escalabilidad, fiabilidad y uso de los recursos. Las pruebas de rendimiento son un subconjunto de la ingeniería de pruebas, una práctica informática que se esfuerza por mejorar el rendimiento, englobándose en el diseño y la arquitectura de un sistema, antes incluso del esfuerzo inicial de la codificación[32]. Para probar el rendimiento del sistema se realizaron pruebas de carga y estrés.

#### 3.5.4.1 Pruebas de carga y estrés

Las pruebas de carga son diseñadas para determinar y validar la respuesta de la aplicación cuando es sometida a una carga de usuarios y/o transacciones que se espera en el ambiente de producción. Entre las dimensiones de calidad de esta prueba se encuentra la de rendimiento. Dichas pruebas se esbozan para asegurar que el sistema pueda procesar una carga esperada. Esto normalmente implica planificar un grupo de pruebas en la que la carga se va incrementado regularmente hasta que el rendimiento del sistema se hace inaceptable. Se ocupan tanto de demostrar que el sistema satisface los requerimientos como de descubrir defectos y problemas en el mismo[26].

Por su parte, las pruebas de estrés son creadas para encontrar el volumen de datos o de tiempo en que la aplicación comienza a fallar o es incapaz de responder a las diferentes peticiones. La fiabilidad, es una de las dimensiones de calidad de este tipo de pruebas[26].

Para la realización de este tipo de pruebas se utiliza la herramienta JMeter haciendo uso de una PC con un procesador Intel Core i5-5500 con una frecuencia de 3.0 GHz y una RAM de 8GB.

A continuación, se muestran las variables analizadas:

- Usuarios: cantidad de usuarios.
- Muestra: cantidad de peticiones realizadas para cada url.
- Media: tiempo promedio en milisegundos en el que se obtienen los resultados.
- Mediana: tiempo en milisegundos en el que se obtuvo el resultado que ocupa la posición central.
- Línea 90 %: máximo tiempo utilizado por el 90 % de la muestra.
- Min: tiempo mínimo que demora un hilo en acceder a una página.
- Max: tiempo máximo que demora un hilo en acceder a una página.
- %Error: porcentaje de error de las páginas que no se llegaron a cargar de manera satisfactoria.
- Rend: rendimiento. Se mide en cantidad de solicitudes por segundo.
- Kb/Seg: medición del rendimiento en cantidad de kilobytes por segundo.

*Tabla 1 Resultados de las pruebas de rendimiento con la herramienta JMeter*

Usuarios	Muestra	Media	Mediana	Línea 90 %	Min	Max	%Error	Rend	Kb/s
100	1900	1130	1320	1276	2	1288	0	83.4	740.7
300	40900	2243	2609	2046	3	423547	0.17	74.2	662.1
500	52200	2950	5125	7946	4	500956	0.32	66.1	548.9

Esta prueba permitió comprobar que los tiempos de respuesta del módulo están dentro del límite permisible para no entorpecer el correcto funcionamiento del GDA eXcriba.

### 3.6 Conclusiones de capítulo

A partir de los resultados obtenidos en el presente capítulo se tiene que:

- Con la elaboración del diagrama de despliegue se logró ilustrar la organización física del módulo.
- La implementación del módulo de gestión de expedientes genéricos del GDA eXcriba cumple con el estándar de codificación descrito lo que permite hacer más fácil el entendimiento del código y el futuro mantenimiento del sistema.
- La ejecución de las pruebas unitarias y el método de caja blanca permitió detectar errores en la implementación del sistema posibilitando la posterior corrección de los mismos; las pruebas funcionales mediante el método de caja negra comprobaron las funcionalidades del sistema y la correcta validación de sus campos. La realización de pruebas de rendimiento

permitió identificar el tiempo de respuesta del sistema cuando es sometido a una carga de usuarios y transacciones esperadas una vez que el mismo esté en explotación.

- La valoración y la comparación realizada teniendo en cuenta un antes y un después del desarrollo el sistema demostró que los usuarios finales están satisfechos con la solución implementada y que esta satisface las necesidades plasmadas.

## Conclusiones generales

---

En el desarrollo del presente trabajo de diploma se obtuvo como resultado la creación del módulo de Gestión de expedientes genéricos del XABAL eXcriba 4.0 dando cumplimiento al objetivo general, por lo que se arriba a las siguientes conclusiones:

- El seguimiento y control de un gestor de expedientes genéricos es un aspecto que no incluía el eXcriba dentro de sus funcionalidades, por lo que constituye una funcionalidad novedosa para el Gestor de Documentos Administrativos Xabal eXcriba 4.0.
- El estudio de los principales conceptos, contribuciones, tecnologías y referentes teóricos relacionados con la gestión documental permitió identificar las características principales del módulo y sentar las bases arquitectónicas e ingenieriles para el desarrollo de esta solución.
- Con la implementación del módulo de gestión de expedientes guiado por la metodología de desarrollo AUP-UCI en correspondencia con los requisitos identificados se obtuvo una solución que favorece el control y seguimiento de los expedientes contribuyendo así a su eficiencia y disponibilidad.
- La validación de la solución propuesta mediante las pruebas de software aplicadas demostró el correcto funcionamiento del módulo, al igual que el cumplimiento de las necesidades plasmadas al comienzo de la investigación con la adecuada calidad y su amplia aptitud para ser usado.



## Recomendaciones

---

Se recomienda la utilización del módulo de gestión de expedientes en el centro CIGED y la Universidad de las Ciencias Informáticas para facilitar una mejor gestión sobre los mismos y de acuerdo a las necesidades que puedan surgir realizar la implementación de nuevas funcionalidades que se puedan acoplar a este.

## Referencias bibliográficas

---

- [1] «Qué es la Gestión Documental», *Incentro*, 12-feb-2020. [En línea]. Disponible en: <https://www.incentro.com/es-es/blog/stories/que-es-la-gestion-documental/>. [Accedido: 12-feb-2020].
- [2] webmaster, «Pilar gestión documental», *Ats Gestion Documental*. [En línea]. Disponible en: <https://atsgestion.net/gestion-documental/>. [Accedido: 11-feb-2020].
- [3] «Amantium Consulting S.L. - Alfresco, gestor de contenidos de código abierto; Amantium referente para su empresa». [En línea]. Disponible en: <https://www.amantium.net/blogs/post/Alfresco-gestor-de-contenidos-de-c%C3%B3digo-abierto-Amantium-referente-para-su-empresa/>. [Accedido: 12-feb-2020].
- [4] «Gestión Electrónica de Expedientes», *Pixelware*. [En línea]. Disponible en: <https://pixelware.com/gestion-electronica-expedientes/>. [Accedido: 12-feb-2020].
- [5] «La gestión documental. Definición, conceptos clave e importancia en la actualidad | Kyocera». [En línea]. Disponible en: <https://www.kyoceradocumentsolutions.es/es/smarter-workspaces/business-challenges/paperless/la-gestion-documental-definicion-conceptos-clave-e-importancia-en-la-actualidad.html>. [Accedido: 20-feb-2020].
- [6] «Gestión documental - Ats Gestión Documental Colombia». [En línea]. Disponible en: <https://atsgestion.net/gestion-documental/>. [Accedido: 12-feb-2020].
- [7] «¿Qué es un sistema de gestión documental?», *TIC Portal*. [En línea]. Disponible en: <https://www.ticportal.es/temas/sistema-gestion-documental/que-es-sistema-gestion-documental>. [Accedido: 12-feb-2020].
- [8] «Que es la gestion documental. Definicion de gestion documental | TBS-Telecon». [En línea]. Disponible en: <http://www.tbs-telecon.es/que-es-gestion-documental>. [Accedido: 12-feb-2020].
- [9] «EXCRIBA 3.1 | Universidad de las Ciencias Informáticas». [En línea]. Disponible en: <https://www.uci.cu/investigacion-y-desarrollo/productos/xabal/excriba-31>. [Accedido: 27-feb-2020].
- [10] «Expediente electrónico, una realidad inmediata». [En línea]. Disponible en: <https://smarterworkspaces.kyocera.es/blog/expediente-electronico-una-realidad-inmediata/>. [Accedido: 12-feb-2020].
- [11] «¿Qué es un Sistema de Gestión de Documentos Electrónicos?», *Transformación Empresarial*, 03-ago-2017. [En línea]. Disponible en: <https://www.pol.com.co/sistema-de-gestion-de-documentos-electronicos-de-archivo/>. [Accedido: 12-feb-2020].
- [12] «Gestor de expedientes en la Gestión Documental de Abox ECM», *Grupo Adapting*. [En línea]. Disponible en: <https://www.adapting.com/gestor-de-expedientes/>. [Accedido: 12-feb-2020].
- [13] «PAe - CTT - General - ARCHIVE». [En línea]. Disponible en: <https://administracionelectronica.gob.es/ctt/archive>. [Accedido: 13-feb-2020].
- [14] «Funcionalidades». [En línea]. Disponible en: <http://www.gedson.es/descripcion/funcionalidades/index.html>. [Accedido: 13-feb-2020].
- [15] «Q-expediente». [En línea]. Disponible en: <https://www.urudatasoftware.com/qexpeditive/#about>. [Accedido: 13-feb-2020].
- [16] «Revista Cubana de Ciencias Informáticas, vol 12. 2018. Disponible en: <https://www.scielo.sld.cu/articuloderevision/getet/>».
- [17] R. S. Pressman, *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education, 2010.
- [18] «Metodología de Desarrollo de Software Variación de AUP para la UCI - EcuRed». [En línea]. Disponible en: [https://www.ecured.cu/Metodolog%C3%ADa\\_de\\_Desarrollo\\_de\\_Software\\_Variaci%C3%B3n\\_de\\_AUP\\_para\\_la\\_UCI](https://www.ecured.cu/Metodolog%C3%ADa_de_Desarrollo_de_Software_Variaci%C3%B3n_de_AUP_para_la_UCI). [Accedido: 20-feb-2020].
- [19] «Capítulo 1. Introducción (Introducción a JavaScript)». [En línea]. Disponible en: <https://uniwebsidad.com/libros/javascript/capitulo-1>. [Accedido: 27-feb-2020].

- [20] «Extensible Markup Language (XML) 1.0 (Fifth Edition)». [En línea]. Disponible en: <https://www.w3.org/TR/xml/>. [Accedido: 27-feb-2020].
- [21] «Definición de HTML», *Definición ABC*. [En línea]. Disponible en: <https://www.definicionabc.com/tecnologia/html.php>. [Accedido: 29-ene-2020].
- [22] «Conoce qué es Spring Framework y por qué usarlo | OpenWebinars». [En línea]. Disponible en: <https://openwebinars.net/blog/conoce-que-es-spring-framework-y-por-que-usarlo/>. [Accedido: 27-feb-2020].
- [23] «PostgreSQL: The world's most advanced open source database». [En línea]. Disponible en: <https://www.postgresql.org/>. [Accedido: 27-feb-2020].
- [24] «Visual Studio Code - Code Editing. Redefined». [En línea]. Disponible en: <https://code.visualstudio.com/>. [Accedido: 27-feb-2020].
- [25] «Alfresco Content Services 5.2.6 | Alfresco Documentation». [En línea]. Disponible en: <https://docs.alfresco.com/5.2/concepts/welcome.html>. [Accedido: 27-feb-2020].
- [26] «Ingeniería del software - Ian Sommerville - Google Libros». .
- [27] Torres hernandez, «25- Torres Hernández, Oscar Daniel. 2013. Propuesta arquitectónica para el núcleo del Gestor de Documentos Administrativos eXcriba 3.0. La Habana: s.n., 2013».
- [28] Erick Gamma, *Gamma, Erick, Patrones de Diseño: elementos de software orientados a objetos reutilizables, 2015*. .
- [29] Larman, Craig *UML y patrones. Introducción al análisis y diseño orientado a objetos 1999*. .
- [30] Polanco Garay, Leodanny Wuilber; y Moré Soto, Dailien, «Iberoamerican Journal of Project Management (IJoPM). [www.ijopm.org](http://www.ijopm.org). ISSN 2346-9161. Vol.10, No.2, A.T., pp.30-60. 2019. Recepción: 08/10/17. Aceptación: 11/09/19. Publicación: 10/12/19».
- [31] «Sánchez Rodríguez, Tamara. Metodología de desarrollo para la Actividad productiva de la UCI. 2014, Universidad de las Ciencias Informáticas». .
- [32] Molyneaux, Ian *The art of application performance testing: Help for programmers and quality assurance*. « O'Reilly Media, Inc.» 2009. .

## Anexos

### 3.7 Anexo 1 Descripción del CUS "Gestionar tipo de expediente"

<b>Objetivo</b>	Permite crear un nuevo tipo de expediente, mostrar el tipo de expediente, modificarlo, eliminarlo y ver detalles del mismo.
<b>Actores</b>	Usuario (Inicia).
<b>Resumen</b>	El caso de uso inicia cuando el usuario selecciona la opción "Crear nuevo tipo de expediente" del menú. La aplicación muestra una nueva ventana y brinda la opción de crear uno nuevo, de asignar su clasificación, tipo, versión, añadir su contenido y asignar la carpeta donde será creado.
<b>Complejidad</b>	Media
<b>Prioridad</b>	Alta
<b>Precondiciones</b>	El usuario se ha autenticado en el sistema. El usuario posee el rol de 'Administrador' o 'Administrador de área'.
<b>Postcondiciones</b>	Se crea, edita, elimina o muestra un nuevo tipo de expediente.
<b>Referencia</b>	<ol style="list-style-type: none"> <li>1. RF Crear tipo expediente</li> <li>2. RF Modificar tipo expediente</li> <li>3. RF Eliminar tipo expediente</li> <li>4. RF Mostrar tipo expediente</li> </ol>

#### Flujo de eventos

##### Flujo básico Gestionar tipos de expedientes

	Actor	Sistema
1.	Selecciona la opción "Gestionar tipos de expedientes" del menú.	
2.		<p>Muestra el listado de los tipos de expedientes existentes en el sistema</p> <p>Permite realizar varias acciones:</p> <ul style="list-style-type: none"> <li>- Crear tipo de expediente: Ver Sección 1</li> <li>- Modificar tipo de expediente: Ver Sección 2</li> <li>- Eliminar tipo de expediente: Ver Sección 3</li> <li>- Ver detalles del tipo de expediente: Ver Sección 4</li> </ul>
3.		

##### Sección 1: "Crear tipo de expediente"

##### Flujo básico Crear tipo de expediente

	Actor	Sistema
1.	Selecciona la opción "Nuevo tipo de expediente" del menú de la página principal	
2.		<p>Muestra una vista con un formulario con los siguientes campos:</p> <ul style="list-style-type: none"> <li>- Clasificación</li> </ul>

		<ul style="list-style-type: none"> <li>- Tipo</li> <li>- Versión</li> <li>- Fecha de creación</li> <li>- Contenido</li> <li>- Carpeta destino</li> </ul>
3.	Introduce los datos correspondientes y da clic en el botón "Crear".	
4.		Crea un nuevo tipo de expediente, almacena en la base de datos la información del mismo la muestra en el listado de los existentes.
5.		Termina el caso de uso.
<b>Flujos alternos</b>		
<b>3a Deja campos obligatorios vacíos</b>		
	<b>Actor</b>	<b>Sistema</b>
3 <sup>a</sup>		No se crea el expediente y señala los campos que aún estén vacíos, para que el usuario los llene.
3b El usuario selecciona la opción cancelar		
	<b>Actor</b>	<b>Sistema</b>
3 c		Cancela la operación.
<b>Sección 2: "Modificar tipo de expediente"</b>		
<b>Flujo básico Modificar tipo de expediente</b>		
	<b>Actor</b>	<b>Sistema</b>
1	Selecciona la opción "editar" del tipo de expediente que desea modificar, en el menú de acciones del expediente seleccionado.	
2		Muestra una vista con un formulario con los siguientes campos para editar: <ul style="list-style-type: none"> <li>- Clasificación</li> <li>- Tipo</li> <li>- Versión</li> <li>- Fecha de creación</li> <li>- Contenido</li> <li>- Carpeta destino</li> </ul>
3	Modifica los datos que desee y da clic en el botón "Aceptar".	
4		Modifica el expediente y guarda la información modificada en la base de datos.
5		Termina el caso de uso
<b>Flujos alternos</b>		
<b>3a. Deja campos obligatorios vacíos</b>		
	<b>Actor</b>	<b>Sistema</b>
3 <sup>a</sup>		No se inserta el expediente y señala los campos que aún estén vacíos, para que el usuario los llene.
<b>4a. El usuario selecciona la opción "Cancelar"</b>		

Actor		Sistema
4 <sup>a</sup>		Cancela la operación.
<b>Sección 3: "Eliminar tipo expediente"</b>		
<b>Flujo básico: Eliminar tipo expediente</b>		
Actor		Sistema
1	Selecciona la opción "Eliminar" del expediente que desea eliminar, en la columna de acciones del expediente deseado.	
2		Muestra un mensaje para que el usuario confirme la eliminación del expediente seleccionado.
3	Selecciona la opción "Eliminar".	
4		Elimina el expediente seleccionado de la base de datos.
		Termina el caso de uso.
<b>Flujos alternos</b>		
<b>3<sup>a</sup>. El usuario selecciona la opción "Cancelar"</b>		
Actor		Sistema
3 <sup>a</sup>		Cancela la operación.
<b>Sección 4: "Ver detalles del expediente"</b>		
<b>Flujo básico: Ver detalles del expediente</b>		
Actor		Sistema
1	Selecciona la opción "Ver detalles" del expediente que desea visualizar, en la columna de acciones.	
2		Muestra una vista con todos los datos del expediente.
3		Termina el caso de uso.
<b>Relaciones</b>	<b>CU incluidos</b>	No Aplica
<b>Prototipo elemental de interfaz gráfica de usuario</b>		

## Nuevo tipo de Expediente

X

Clasificación:

Proyecto

Fecha:

dd / mm / aaaa

Tipo:

Personalización

Version:

4.0

Cargar contenido....

Carpeta destino....

Documentos	Roles	Acciones
Exped 1		/ ☒
Exped 2		/ ☒

Crear

Crear y Cargar nuevo

Cancelar

	Mostrar versión del exped...	Asignar permisos a un doc...	Asignar rol a un miembro d...	Añadir contenido del expe...	Crear expediente	Crear tipo expediente	Eliminar contenido del exp...	Eliminar expediente	Eliminar permisos a un doc...	Eliminar rol a un miembro d...	Eliminar tipo expediente	Exportar expediente	Filtrar por autor	Filtrar por clasificación	Filtrar por estado	Filtrar por fecha	Filtrar por nombre	Filtrar por tipo	Filtrar por versión	Imprimir expediente	Modificar contenido del ex...	Modificar expediente	Modificar permisos a un do...	Modificar rol a un miembro ...	Modificar tipo expediente	Mostrar acciones del expe...	Mostrar autor del expediente	Mostrar clasificación del ex...	Mostrar contenido del exp...	Mostrar estado del expedi...	Mostrar expediente	Mostrar fecha del expedie...	Mostrar nombre del expedi...	Mostrar permisos del docu...	Mostrar rol de un miembro ...	Mostrar tipo expediente			
Mostrar versión del exped...	✓																																						
Asignar permisos a un doc...		✓																																					
Asignar rol a un miembro d...			✓																																				
Añadir contenido del expe...				✓																	✓								✓										
Crear expediente					✓																	✓									✓								
Crear tipo expediente						✓																		✓														✓	
Eliminar contenido del exp...			✓																		✓								✓										
Eliminar expediente				✓																																			
Eliminar permisos a un doc...		✓																					✓													✓			
Eliminar rol a un miembro d...			✓																					✓													✓		
Eliminar tipo expediente					✓																			✓														✓	
Exportar expediente																										✓													
Filtrar por autor													✓			✓		✓																					







3.8 Anexo 2: Matriz de trazabilidad RF-RF.



- ▼ Herramientas
  - Aplicación
  - Navegador de nodos
- ▼ Expedientes y Roles
  - Expedientes
  - Roles

- Gestionar Expedientes
- Gestionar Permisos

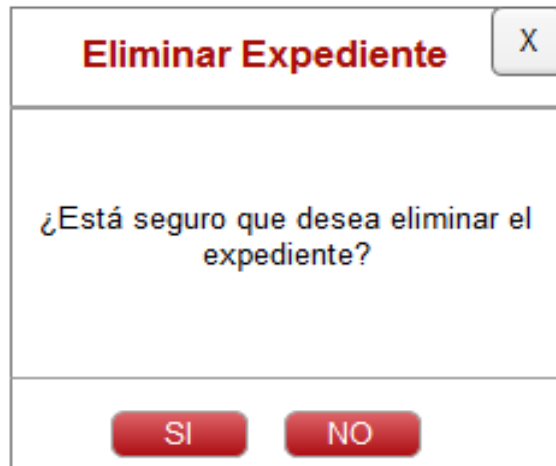
Mostrar  elementos

Clasificación ▾	Tipo ▾	Version ▾	Autor ▾	Fecha de creado ▾	Acciones
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
					  
					  

Mostrando 1 páginas de 1

Anterior  Siguiente

3.9 Anexo 3: Prototipo de interfaz (Panel Principal del módulo)



3.10 Anexo 4: Prototipo de interfaz (Vista eliminar expediente del módulo)

**Editar tipo de Expediente** X

Clasificación:  Fecha:

Tipo:

Version:

---

Contenido:

Documentos	
Exped 1	/
Exped 2	/

**3.11** Anexo 5: Prototipo de interfaz (Panel editar expediente del módulo)

3.12 Anexo 6: Atributos de calidad para las pruebas TOC

	<b>Categoría</b>	<b>Criterio</b>
<b>Responsabilidad</b>	<b>Alta</b>	<b>&gt;2*prom</b>
	<b>Media</b>	<b>Entre prom y 2*prom</b>
	<b>Baja</b>	<b>&lt;=prom</b>

	<b>Categoría</b>	<b>Criterio</b>
<b>Complejidad de implementación</b>	<b>Alta</b>	<b>&gt;2*prom</b>
	<b>Media</b>	<b>Entre prom y 2*prom</b>
	<b>Baja</b>	<b>&lt;=prom</b>

	<b>Categoría</b>	<b>Criterio</b>
<b>Reutilización</b>	<b>Alta</b>	<b>&lt;=prom</b>
	<b>Media</b>	<b>Entre prom y 2*prom</b>
	<b>Baja</b>	<b>&gt;2*prom</b>

3.13 Anexo 7: Umbrales de prueba RC

	Categoría	Criterio
	ninguno	0
<b>Acoplamiento</b>	Baja	1
	Media	2
	Alta	>2

	Categoría	Criterio
<b>Complejidad de mantenimiento</b>	Alta	>2*prom
	Media	Entre prom y 2*prom
	Baja	<=prom

	Categoría	Criterio
<b>Reutilización</b>	Alta	<=prom
	Media	Entre prom y 2*prom
	Baja	>2*prom

	Categoría	Criterio
<b>Cantidad de pruebas</b>	Alta	>2*prom
	Media	Entre prom y 2*prom
	Baja	<=prom

3.14 Anexo 8: Caso de prueba para el escenario "Modificar expediente".

Escenario	Descripción	Nombre	Clasificación	Tipo	Versión	Contenido del expediente	Carpeta destino	Respuesta del sistema	Flujo central
EC1. Modificar expediente con datos correctos	El usuario modifica los parámetros correctamente y se hacen los cambios en el sistema	V Expediente de postgrado	V Laboral	V Centro	V 2.0	N/A	N/A	Modifica el expediente correctamente.	1-Selecciona la opción modificar expediente. 2-Se realizan todas las modificaciones correctamente. 3-Selecciona la opción aceptar.
EC 2. Modificar expediente con campos inválidos.	El usuario modifica los parámetros con campos inválidos, no se hacen los cambios en el sistema y sale un mensaje de error.	I proyecto	N/A	N/A	N/A	N/A	N/A	Muestra un mensaje indicando que existen campos inválidos.	1-Selecciona la opción modificar expediente. 2-No se realizan las modificaciones. 3-Muestra un mensaje de error.
EC 3. Modificar expediente con campos vacíos.	El usuario modifica los parámetros pero deja campos vacíos, no se hacen los cambios en el sistema y sale un mensaje de error.	I	V Trabajo	I	V 4.0	N/A	N/A	Muestra un mensaje indicando que existen campos obligatorios vacíos.	1-Selecciona la opción modificar expediente. 2-No se realizan las modificaciones. 3-Muestra un mensaje de error.