

Universidad de las Ciencias Informáticas
Facultad 2



Módulo para la gestión de expedientes de proyecto de la Red de Centros de la UCI en el Gestor de Documentos Administrativos Xabal-eXcriba 3.1.

Trabajo de Diploma para Optar por el Título de Ingeniero en Ciencias.

Autor: Andrés Maikel Guerra Manrique
Tutor: Damian Troya Martinez
Co-Tutor: Wisbel Sánchez de la Noval

La Habana, 20 de junio de 2018

Declaración de autoría

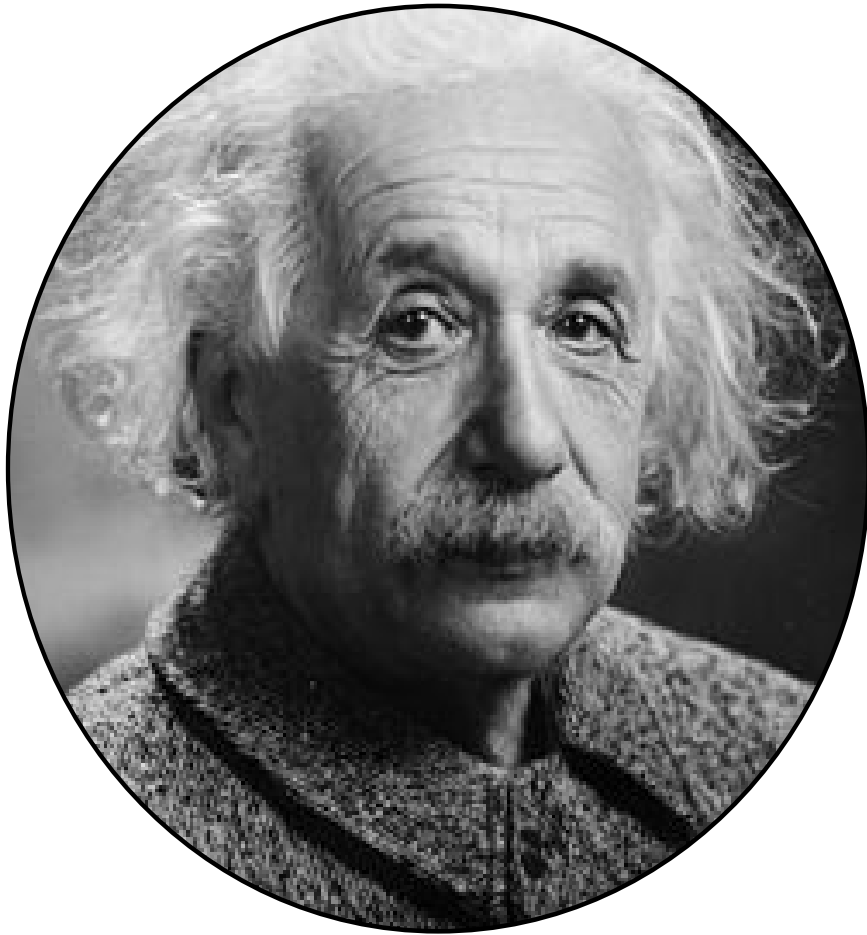
Se declara que Andrés Maikel Guerra Manrique es el único autor del presente Trabajo de Diploma que tiene por título “Módulo para la gestión de expedientes de proyecto de la Red de Centros de la UCI en el Gestor de Documentos Administrativos Xabal-eXcriba 3.1”, se concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales del mismo con carácter exclusivo.

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

Andrés Maikel Guerra Manrique

Ingeniero
Damian Troya Martinez

Ingeniero
Wisbel Sánchez de la Noval



"Hay una fuerza motriz más poderosa que el vapor, la electricidad, y la energía atómica, la voluntad."

Albert Einstein (1879-1955)

Agradecimientos

Una tesis es un trabajo que no sólo es fruto del esfuerzo personal del tesista, sino que necesita de la ayuda de muchas personas, tanto en lo profesional como en lo personal. Con estas líneas quisiera mostrar mi agradecimiento a todas ellas.

A mis tutores, a Nadiela por la guía que representaron durante la elaboración el trabajo. A los profesores que de alguna manera fueron la fuente del apoyó durante los 5 años, a los profesores Yaniselis y Leonil, de los que aprendí no sólo lo que es ser un INGENIERO con mayúsculas. A Yordi por la ayuda incondicional que me brindo en este proceso.

A todos mis compañeros de curso, muchos de ellos amigos, que consiguieron darme el afecto necesario para disfrutar del día a día. Al grupo del técnico, los viejos amigos, Ernesto, Yiyi, Cynthia, Jorge y Bayron, por estar presente desde el inicio. A mis amigos, especialmente a David, Suri, Mayde, Osmani y Wilmer, por escucharme, aguantarme y animarme a seguir adelante. Gracias por estar presentes.

A mis padres, por su apoyo incondicional durante todos los años. Gracias, con vuestro cariño todo ha sido mucho más fácil. A mi abuela mimame que siempre esta pendiente de mi. A esa persona especial que me ha dado fuerzas para seguir adelante, Yon, en las buenas y en las malas estas presentes para brindar tu apoyo.

Dedicatoria

A mi mamá y mi familia.

Resumen

Actualmente la información es la unidad más importante con que cuenta una empresa, por lo que el crecimiento de la misma en cada una de las entidades ha conllevado a la creación de sistemas de gestión documental, capaces de salvaguardar la información y mantenerla disponible, evitando la duplicidad y dispersión de la misma.

Este trabajo tuvo como objetivo el desarrollo de un módulo para la gestión de expedientes de proyecto de la Red de Centros de la Universidad de Ciencias Informáticas en el Gestor de Documentos Administrativos Xabal-eXcriba 3.1. Como medio para alcanzar el objetivo propuesto se utilizaron las herramientas y tecnologías: UML 2.0 como lenguaje de modelado, Visual Paradigm 8.0 como herramienta CASE, JavaScript 1.6 como lenguaje de programación, Visual Studio Code v1.23.1 como editor de código, AUP-UCI en su escenario 2 como metodología de desarrollo, además de PostgreSQL v9.4 y PgAdmin III como gestor y administrador de base de datos respectivamente. Se implementaron las funcionalidades propuestas y se validaron mediante pruebas unitarias, funcionales y de aceptación. Las pruebas arrojaron resultados satisfactorios, lo cual demuestra la calidad de la solución propuesta.

Una vez concluido el proceso de desarrollo se obtuvo como resultado un módulo que permite gestionar los expedientes de proyecto mediante la asignación de lista de datos de miembros y de artefactos, garantizando la asignación de permisos sobre los documentos del expediente, además permitió contribuir en el proceso de despliegue de los expedientes de proyecto en la red de Centros.

Palabras claves: módulo, gestión documental, gestión de expedientes.

Abstract

Currently the information is the most important unit of a company, so the growth in each of the entities has led to the creation of document management systems, able to safeguard the information and keep it available, avoiding duplication and dispersal of the same.

This work aimed at the development of a module for the management of the network of centres of the University of Informatics Sciences project manager administrative documents Xabal-eXcriba 3.1. As means to achieve the proposed objective be used tools and technologies: UML 2.0 modeling, Visual Paradigm 8.0 as a tool language CASE, JavaScript 1.6 as a programming language, Visual Studio Code v1.23.1 as a code editor, AUP-UCI in your scenario 2 as a methodology of development, as well as PostgreSQL v9.4 and PgAdmin III as Manager and database administrator respectively. The proposed features were implemented and were validated through testing unit, functional and acceptance. Tests showed satisfactory results, which demonstrates the quality of the proposed solution.

Upon completion of the development process resulted in a module allowing to manage records of project by assigning data members and artifacts list, ensuring the assignment of permissions on documents of the record, also allowed to contribute in the process of deployment of records project in the network of centres.

Key words: module, document management, records management.

Índice General

Índice de figuras	IX
Índice de tablas	X
Introducción	1
1. Gestión de expedientes de proyecto en el GDA Xabal eXcriba: Fundamentación teórica.	6
1.1. Conceptos asociados al tema.	6
1.2. Centros productivos en la UCI.	11
1.3. Gestión de expedientes de proyecto en el GDA Xabal eXcriba 3.0	13
1.4. GDA Xabal eXcriba 3.1	14
1.5. Metodologías de desarrollo de software.	16
1.5.1. Metodología AUP-UCI.	16
1.6. Lenguajes de programación.	16
1.7. Herramientas informáticas.	18
1.8. Tecnologías informáticas.	20
1.9. Conclusiones del capítulo.	24
2. Módulo para la gestión de expedientes de proyecto en el GDA Xabal eXcriba 3.1: Propuesta de solución.	25
2.1. Descripción de la propuesta de solución.	25
2.2. Modelo conceptual.	26
2.3. Diccionario de datos.	27
2.4. Especificación de los requisitos de <i>software</i>	28
2.4.1. Técnicas para la captura de los requisitos.	28
2.4.2. Requisitos funcionales.	29

2.4.3. Requisitos no funcionales.	33
2.5. Definición de los casos de uso del sistema.	34
2.5.1. Definición de los actores del sistema.	34
2.5.2. Diagrama de caso de uso del sistema.	35
2.5.3. Matriz de trazabilidad.	36
2.5.4. Descripción del caso de uso 4: Gestionar elemento de la lista de datos de miembros.	37
2.6. Arquitectura del sistema.	40
2.7. Patrones de diseño.	42
2.8. Conclusiones del capítulo	43
3. Módulo para la gestión de expedientes de proyecto en el GDA Xabal eXcriba 3.1: Implementación y validación.	44
3.1. Implementación.	44
3.1.1. Estándares de codificación.	44
3.2. Modelo de despliegue.	45
3.2.1. Diagrama de despliegue.	46
3.3. Pruebas del módulo	47
3.3.1. Método de caja negra.	47
3.3.2. Método de caja blanca	51
3.4. Conclusiones del capítulo	54
Conclusiones Generales	55
Recomendaciones	56
Glosario de términos	57
Referencias bibliográficas	58
Bibliografía	61
A. Anexos	63
A.1. Diccionario de datos de la entidad expediente de proyecto.	63
A.2. Diccionario de datos de la entidad lista de datos de miembros.	64
A.3. Diccionario de datos de la entidad lista de datos de artefactos.	65

A.4. Diccionario de datos de la entidad administrador.	66
A.5. Diccionario de datos de la entidad especialista de calidad.	66
A.6. Diccionario de datos de la entidad asesor de planificación.	67
A.7. Modelo conceptual a requisito.	68
A.8. Requisito a requisito.	69
A.9. Caso de uso a caso de prueba.	70
A.10.Gestionar área.	71
A.11.Gestionar documento a partir de plantilla.	73
A.12.Gestionar expediente de proyecto.	76

Índice de figuras

1.1. Proceso de gestión de un nuevo expediente de proyecto.	13
2.1. Propuesta de solución.	26
2.2. Modelo conceptual.	26
2.3. Diagrama de casos de uso del sistema.	36
2.4. Matriz de caso de uso a requisito.	37
2.5. Estilo arquitectónico del sistema	41
2.6. Ejemplo de patrones GRASP: Experto.	42
2.7. Ejemplo de patrones GRASP: Bajo acoplamiento.	43
2.8. Ejemplo de patrones GRASP: Alta cohesión.	43
3.1. Estándares de codificación indentación y líneas	45
3.2. Estándares de codificación variables	45
3.3. Estándares de codificación funciones	45
3.4. Diagrama de despliegue.	46
3.5. Resultado de las pruebas.	51
3.6. Técnica de ruta básica en el método <code>getInformationFromPMDDataList(datalist)</code>	52
3.7. Representación del grafo de flujo de camino básico	52
A.1. Matriz de modelo conceptual a requisito.	68
A.2. Matriz de requisito a requisito.	69
A.3. Matriz de caso de uso a caso de prueba.	70

Índice de tablas

2.1.	Diccionario de datos de la entidad Área	28
2.2.	Definición de los actores del sistema.	35
2.4.	Descripción del CU_4: Gestionar elemento de la lista de datos de miembros.	37
3.1.	Variables para el caso de prueba: Gestionar elemento de la lista de datos de miembros. . . .	49
3.2.	Caso de prueba: CU Gestionar elemento de la lista de datos de miembros.	49
3.3.	Caso de prueba: ruta independiente # 4	54
A.1.	Diccionario de datos de la entidad expediente de proyecto.	63
A.2.	Diccionario de datos de la entidad lista de datos de miembros.	64
A.3.	Diccionario de datos de la entidad lista de datos de artefactos.	65
A.4.	Diccionario de datos de la entidad administrador.	66
A.5.	Diccionario de datos de la entidad especialista de calidad.	66
A.6.	Diccionario de datos de la entidad asesor de planificación.	67
A.7.	Descripción del CU_2: Gestionar documento a partir de plantilla.	73
A.8.	Descripción del CU_6: Gestionar expediente de proyecto.	76

Introducción.

La gestión de documentos es tan antigua como la escritura misma y se concibió para suplir la necesidad del hombre de fundamentar actos administrativos, transacciones legales y comerciales. En sus inicios, la gestión de documentos utilizó tablillas de arcilla, hojas de papel, pergaminos, entre otros medios; pero como los fondos documentales fueron creciendo a medida que los negocios se tornaron más complejos, el hombre necesitó de otros medios para llevar a cabo la administración de sus documentos. Todo el trabajo relacionado con la gestión de documentos fue llevado durante siglos, por administradores, archiveros y bibliotecarios cuyas herramientas manuales eran los libros de registro y carpetas. Con el transcurso del tiempo fue necesario el uso de las tecnologías de la informática y las comunicaciones (TIC) y surgieron programas que se encargaban de esta función.

Como resultado del desarrollo tecnológico surgieron los Sistema de Gestión Documental (SGD). Un SGD es un sistema informático utilizado para rastrear y archivar documentos electrónicos y/o imágenes de documentos de papel escaneados. Actualmente en el mercado existen muchas soluciones disponibles para poder implantar los SGD en la empresa, tal es el caso de SharePoint de Microsoft, Documentum de Dell EMC o Interwoven de Autonomy, todas ellas soluciones privativas. Otros sistemas están disponibles en la modalidad de software libre (*Open Source*) y en algunos casos con una reducción de costes importante. Dentro de las soluciones en el mercado de software libre las más conocidas en el ámbito internacional son Alfresco, Nuxeo y OpenKM.

Por otra parte, Alfresco Community Edition es una plataforma de código abierto para la gestión de contenido o ECM¹. La solución de gestión de documentos de Alfresco permite controlar el contenido de la empresa. Los documentos importantes, como los contratos legales, activos de *marketing* y documentos de ingeniería, pueden encontrarse, compartirse y protegerse fácilmente. Alfresco maximiza el valor del contenido integrándolo en los procesos empresariales centrales. La información llega a las personas adecuadas en el momento justo, en las aplicaciones y dispositivos que usan para hacer sus trabajos. [[Community, 2015](#)]

La Universidad de Ciencias Informáticas (UCI) tiene como objetivos fundamentales la formación de personal capacitado en las Ciencias Informáticas, así como la producción de software a través de sus Centros Productivos. Cada uno de estos centros cuenta con varios productos de gran impacto en la informatización

¹Sistema de Gestión de Contenidos, en inglés: Content Management System.

de la sociedad cubana, en áreas como la seguridad ciudadana, administración pública, industria, educación, gestión empresarial, salud y biotecnología, entre otros sectores. Además, se cuenta con la experiencia de numerosos clientes nacionales y extranjeros. Entre los centros productivos de la UCI esta el Centro de Informatización de la Gestión Documental (CIGED), donde se desarrolló un software para la gestión documental denominado Gestor de Documentos Administrativos Xabal eXcriba, cuyo objetivo principal es automatizar los flujos documentales que se ejecutan dentro de cualquier entidad, desde la elaboración de un documento en su fase de inicio, hasta su conservación o eliminación. Incluye acciones sobre los documentos, tales como: creación, clasificación, el control de versiones, la gestión de flujos de trabajo, almacenar documentos en diferentes formatos electrónicos y gestionar los trámites de los documentos que se generan o reciben. Este sistema informático tiene como núcleo el sistema Alfresco en su versión Community Edition 4.2f. Actualmente el GDA Xabal eXcriba se encuentra disponible en su versión 3.1.

La gestión de expedientes de proyecto en la Red de Centros de la UCI utiliza el GDA Xabal eXcriba en su versión 3.0, presentando las siguientes limitantes en el proceso de gestión:

1. En el proceso de implantación de un nuevo tipo de expediente se especializa el expediente y los documentos del mismo. Dicho proceso de especialización se realiza de forma manual, por el administrador del sistema.
2. El módulo no permite al administrador del área, asignar una lista de miembros nueva al expediente de proyecto.
3. No existe una lista de artefactos única por cada tipo de expediente, como consecuencia la lista esta replicada en cada una de las áreas.
4. No se puede realizar una actualización automática de los permisos en el expediente de proyecto, cuando se modifica la lista de miembros del expediente.
5. El sistema no cuenta con la posibilidad de editar un expediente después de creado.
6. La creación de documentos a partir de plantillas según el tipo de expediente de proyecto no está implementada, actualmente en la funcionalidad solo se muestran las plantillas del último expediente de proyecto definido.
7. La Dirección de Calidad, no forma parte de las áreas del sistema, como consecuencia para cualquier actualización y revisión en los expedientes de proyecto, se depende del administrador del sistema y del equipo de desarrollo.

8. Los documentos de los expedientes no cuenta con descripción, incumpliendo con lo establecido por la Dirección de Calidad de la UCI.
9. Existe dependencia de los especialistas de calidad hacia los administradores y el equipo de desarrollo del módulo para la implantación de un nuevo tipo de expediente, ralentizando el proceso de despliegue y uso del nuevo expediente en los centros productivos.
10. La implantación de un nuevo tipo de expediente no es configurable a nivel de usuario, por tal motivo el equipo de desarrollo tiene que hacer modificaciones en el código del sistema.

Dada la situación problemática anteriormente descrita, se identifica como **problema a resolver**: ¿Cómo contribuir al proceso de gestión de los expedientes de proyecto de la Red de Centros de la Universidad de las Ciencias Informáticas?. Partiendo de este problema se define como **objeto de estudio**: El proceso de gestión de los expedientes de proyecto.

Para resolver el problema identificado se ha propuesto como **objetivo general**: Desarrollar un módulo para la gestión de los expedientes de proyecto de la Red de Centros de la Universidad de las Ciencias Informáticas, en el Gestor de Documentos Administrativos Xabal eXcriba 3.1. Enmarcado en el **campo de acción**: La gestión de los expedientes de proyecto para la Red de Centros de la Universidad de las Ciencias Informáticas.

Para el cumplimiento del objetivo general de la investigación se definen las siguientes **tareas de la investigación**:

1. Elaboración de los fundamentos teóricos y metodológicos de la investigación relacionados con la gestión de los expedientes de proyecto para la Red de Centros de la Universidad de las Ciencias Informáticas.
2. Análisis de las funcionalidades del módulo para la gestión de expedientes de proyecto para la Red de Centros de la UCI en el GDA Xabal eXcriba 3.0.
3. Desarrollo del módulo para la gestión de los expedientes de proyecto de la Red de Centros de la UCI en el GDA Xabal eXcriba 3.1.
4. Validación del módulo desarrollado a partir de los métodos de prueba definidos en la investigación.

En la presente investigación se utilizaron los siguientes métodos teóricos y empíricos:

■ Teóricos:

1. **Analítico-Sintético:** permitió la obtención de los conceptos fundamentales relacionados con la gestión documental. Recopilando la información necesaria mediante la revisión de documentos, artículos científicos y sitios web especializados, además se analizó las diferentes herramientas, metodologías y tecnologías de desarrollo de software a utilizar en la implementación del módulo.

■ Empíricos:

1. **Entrevista:** el tipo de entrevista no estructurada se empleó para obtener la información necesaria para diseñar e implementar el módulo, en lo referente a tipos de entradas, tipos de salidas y para evaluar el comportamiento correcto o no del módulo.

Estructura del Documento:

El presente trabajo se ha estructurado en tres capítulos:

Capítulo 1: Gestión de expedientes de proyecto en el GDA Xabal eXcriba: Fundamentación teórica.

Se abordan los diferentes elementos que brindan la base teórico-conceptual para el desarrollo de la solución propuesta. Se reflejan varios conceptos asociados a la gestión documental y a los expedientes de proyecto; así como la realización de un estudio del módulo para la gestión de expedientes en GDA Xabal eXcriba 3.0, para conocer cómo realiza la gestión de expedientes de proyecto. Además, se plantean las tecnologías, herramientas y metodología de desarrollo de software que se requieren para dar cumplimiento al objetivo de la presente investigación.

Capítulo 2: Módulo para la gestión de expedientes de proyecto en el GDA Xabal eXcriba 3.1: Propuesta de solución. En este capítulo se define la propuesta de solución para el desarrollo del módulo para la gestión de expedientes de proyecto de la Red de Centros de la UCI en el GDA Xabal eXcriba 3.1. Además, se expone a partir de un conjunto de requerimientos funcionales y no funcionales cómo se le dará solución a cada una de las problemáticas identificadas. Además de la arquitectura propuesta para el componente a desarrollar, generando todos los artefactos necesarios.

Capítulo 3: Módulo para la gestión de expedientes de proyecto en el GDA Xabal eXcriba 3.1: Implementación y validación. En este capítulo se explica a partir de los resultados obtenidos en el capítulo anterior, la implementación del módulo. Se definen los estándares de codificación y se propone el modelo de despliegue de la solución. Se define la estrategia a seguir en las pruebas, especificando niveles, métodos y tipos de pruebas. Por último, se muestran los resultados obtenidos a partir del diseño de cada caso de prueba realizado. Además, se abordan los resultados obtenidos en el desarrollo de la solución y las consideraciones

finales mediante la realización de las pruebas pertinentes realizadas al software.

Capítulo 1

Gestión de expedientes de proyecto en el GDA Xabal eXcriba: Fundamentación teórica.

Aunque las técnicas básicas de la gestión documental ya se llevan aplicando desde hace décadas en el uso del control de documentos, y a lo largo de los años ha ido evolucionando. El aumento de documentos en las instituciones ha traído como consecuencia un engorroso trabajo sobre la gestión de documentos, haciendo necesario una mejor gestión documental. En el presente capítulo se abordan los conceptos y aspectos fundamentales relacionados con la gestión documental y la gestión de expedientes de proyecto. Se exponen y explican las herramientas y tecnologías utilizadas en el desarrollo del módulo.

1.1. Conceptos asociados al tema.

A continuación, se relacionan algunos de los conceptos referidos al tema que ocupa la investigación, para permitir una mayor comprensión del funcionamiento de la propuesta de solución apoyando la investigación en curso.

Documento.

La mayoría de los autores nacionales y extranjeros coinciden en conceptualizar al documento como un objeto material producto de la actividad humana destinado a representar por medio de signos externos. Se puede definir como documento al medio en el que se registra o por el que se transmite información en cualquier soporte y que en sentido general contiene la expresión del trabajo de creación humana en formato impreso o no impreso. [Cuba, 2003]. Por parte de la Real Academia Española define por documento: *“escrito en que constan datos fidedignos o susceptibles de ser empleados como tales para probar algo”*. [RAE, 2017]

Por otro lado, el autor Signer, entiende por documento *“toda expresión testimonial generada en cualquier fecha, en cualquier lenguaje, forma o soporte”* [Signer, 2010]. Esta definición que establece Signer completa el concepto de documento agregando que este puede estar expresado en cualquier elemento material que sirva para fines representativos. Se refiere, al soporte del documento, que es el elemento, el sustrato

material sobre el que se asienta la información, como, por ejemplo, papel, fotografías, grabaciones y cintas magnéticas.

Se concluye que un documento es un testimonio material registrado en una unidad de información en cualquier tipo de soporte (papel, dispositivo electrónico, fotografías) en lengua natural o convencional. Es el testimonio de una actividad humana fijada en un soporte.

Tipología documental.

La doctora Vicenta Cortés Alonso definió el tipo documental como "número y disposición de los elementos de la información que corresponden a la actividad que lo ha producido" [Cortés Alonzo, 1989]. Por otra parte la autora Antonia Heredia ha expuesto que la tipología documental es *"la suma de tipología diplomática y tipología jurídico-administrativa. La delimitación de los tipos, su fijación e identificación vendrán determinados por el análisis de los caracteres externos e internos de los documentos y de su mensaje o información"*. [Heredia Herrera, 1991]

Por su parte, dentro de las Ciencias de la Información, se conoce como tipología documental a los distintos formatos y tipos de documentos que una entidad específica elige y desarrolla, para ser aplicados en las tareas de producción o recepción documental, a fin de generar a su vez una dinámica homogénea para todos los departamentos y fuentes de alimentación de archivos.

Existe una estrecha relación entre el tipo documental y la actividad que lo produce, un ejemplo de esto puede ser "informe" que proviene de la actividad de "informar". Dada esta relación se puede afirmar que los tipos documentales pueden ser infinitos, como lo son las actividades humanas y varían en el tiempo y el espacio. Además, están dadas en correspondencia con las actividades que dan lugar a los documentos, sus funciones y su uso.

Importancia de la tipología documental

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

- Aceleración y sistematización para organizar los documentos.
- Eficacia de la recuperación de la información para la toma de decisiones.

- Protección de la información administrativa.
- Estabilidad y continuidad administrativa.
- Optimización de los recursos y racionalización de los espacios.

Por todo lo anteriormente expuesto se concluye que la tipología documental es la encargada de estudiar y profundizar las diferentes clasificaciones que se le dan a los documentos que son originados en una organización. Estos documentos son del mismo tipo y con una misma estructura física. Una de sus características fundamentales es que poseen igual forma de transmitir la información. Otra consideración es, que los tipos de documentos generados por una organización suelen ser amplios, variados y complejos, dependiendo del tiempo de su actividad.

Expediente de proyecto.

El mecanismo tradicional más empleado para ejecutar procedimientos administrativos en la órbita de los organismos públicos y algunos privados, es el expediente.

Según el Archivo General de la Nación de Bogotá, se define como expediente: el conjunto de documentos producidos y recibidos durante el desarrollo de un mismo trámite o procedimiento, acumulados por una persona, dependencia o unidad administrativa, vinculados y relacionados entre sí y que se conservan manteniendo la integridad y orden en que fueron tramitados, desde su inicio hasta su resolución definitiva. [[Archivo General de la Nación, 2015](#)]

El expediente además de ser la esencia de las actuaciones de la administración, reúne de manera orgánica los documentos que se producen o reciben en desarrollo de un mismo trámite o actuación y se acumulan de manera natural reflejando el orden en que dicho trámite es ejecutado, es la base de la organización archivística sobre la cual se establecen las series y subseries documentales que conforman un archivo [[Archivo General de la Nación, 2015](#)].

La autora Aida Navarro precisa como expediente administrativo o expediente electrónico el conjunto de documentos electrónicos sobre un determinado asunto ordenados secuencialmente, cuya gestión desde su creación, tratamiento archivístico comprende la aplicación de los procesos técnicos archivísticos de organización, descripción y valoración, conservación y servicio se ejecuta íntegramente mediante la aplicación de la tecnología informática [[Navarro, 2008](#)].

Un proyecto es una asociación de esfuerzos, limitado en el tiempo, con un objetivo definido, que requiere del acuerdo de un conjunto de especialidades y recursos. También puede definirse como una organización

temporal con el fin de lograr un propósito específico. Cuando los objetivos de un proyecto son alcanzados se entiende que el proyecto está completo [Garzás, 2013].

Por los conceptos anteriormente expuestos, un expediente de proyecto consiste en un conjunto de documentos físicos o electrónicos correspondientes a un procedimiento administrativo, cualquiera que sea el tipo de información que contengan. Garantizan un acceso eficiente y seguro a la información y a todos los documentos desde cualquier sitio, su primordial función es la de conservar la información y transmitirla.

Roles¹ y permisos.

Un punto crítico en los sistemas hoy en día son los permisos y es importante disponer de algún mecanismo que permita establecer que operaciones podrá realizar cada usuario. El objetivo es evitar que, por accidente o de forma intencionada, un usuario pueda realizar operaciones que comprometan la integridad y funcionamiento del sistema o la información.

La norma UNE-ISO15489-1 define el acceso como derechos, modo y medios de localizar, usar o recuperar información [International Organization for Standardization, 2006]. Teniendo en cuenta la definición anterior se puede concebir el control de acceso como el proceso de conceder o denegar permisos a usuarios para acceder a objetos, información o datos. Cuando se accede a un recurso existen tres actividades que están estrechamente relacionadas: la autenticación (quién soy), la autorización (qué puedo hacer), el registro de auditoría (qué he hecho).

Autenticación: Cuando un usuario requiere acceso a un sistema, debe presentar información personal que permita establecer de forma unívoca su identidad dentro del entorno. El sistema procesa estos datos y determina si ese usuario es realmente quien dice ser.

Autorización: Este proceso determina lo que un usuario ya autenticado, tiene derecho o no a hacer.

Auditoría: En muchos sistemas se quiere llevar un control detallado de las operaciones que realizan los usuarios en el mismo, por tal razón existen mecanismos para registrar todas las acciones de los usuarios y posteriormente poder ser usado durante un proceso de auditoría.

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

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

Modelo de autorización

Control de Acceso Basado en Roles (RBAC)

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

Para lograr la confiabilidad en un sistema se debe tener en cuenta los caminos para llegar a ella, los niveles que existen en los usuarios (ejemplo: usuario y súper usuario) y los permisos o privilegios que dichos usuarios puedan llegar a tener. En ocasiones los usuarios de una organización pueden cambiar de un puesto a otro, cambiando también sus permisos y responsabilidades, llegando a ser difícil y de alto coste. La situación anterior puede ser evitada mediante RBAC, ya que son los roles de usuario los que posibilitan el acceso al sistema y no la identificación del usuario. Cada rol realiza una función y puede tener asignados uno o más usuarios, además tienen asociado permisos que determinan los datos y aplicaciones a las que el rol tiene acceso. [[Whitepaper, 2015](#)]

Gestión documental (GD).

La gestión de documentos se ha transformado en un término bastante usual en la sociedad, debido al creciente desarrollo de la información y la generación de contenidos que esto conlleva y su repercusión en las organizaciones que la utilizan cotidianamente. La gestión es un proceso mediante el cual se utiliza una variedad de recursos básicos para apoyar los objetivos de la organización.

La GD puede definirse según Patricia Russo Gallo como: el conjunto de actividades que permiten coordinar y controlar los aspectos relacionados con creación, recepción, organización, almacenamiento, acceso y difusión de documentos [[Gallo, 2011](#)].

La norma UNE-ISO 15489-1 define la gestión documental como el “área de la gestión responsable de un control eficaz y sistemático de la creación, la recepción, el mantenimiento, el uso y la disposición de documentos de archivo, incluidos los procesos para incorporar y mantener en forma de documentos la información y prueba de las actividades y operaciones de la organización” [[International Organization for Standardization, 2006](#)].

Al reflexionar los criterios expuestos por los autores mencionados anteriormente, se puede concluir que la gestión documental permite controlar el ciclo de vida de los documentos, es decir, desde su creación, hasta que son eliminados o archivados; evitando el desorden, duplicidad o pérdida de documentos, lo que permite agilizar los procesos en los que se encuentran involucrados los documentos, ahorrando tiempo y

recursos a las organizaciones.

Sistemas de gestión documental (SGD).

Durante los últimos tiempos, las empresas se han visto en la necesidad de apoyarse en las herramientas de software de gestión documental para evitar el colapso de los sistemas de información. Este se debe en gran medida a la complejidad y diversidad de información que maneja una organización, así como el gran volumen de datos que se genera, sin importar el tamaño del mismo.

Los sistemas de gestión documental o de gestión de documentos *“se definen como el conjunto de elementos relacionados con la gestión documental que interactúan de manera sincronizada y controlada en la creación, conservación, uso y disposición de los documentos de una organización a partir de la aplicación de técnicas administrativas modernas dirigidas a aumentar la eficiencia administrativa, reducir costos y facilitar la gestión administrativa general, a partir de la integración de las diferentes tecnologías de la información y la comunicación, los sistemas de información y la aplicación de los principios básicos de la archivística”*. [Zapata Cardenas, 2005]

La compañía AIIM define a un sistema de gestión documental como las estrategias, métodos y herramientas informáticas utilizadas para crear, capturar, automatizar, entregar, asegurar y analizar contenido y documentos relacionados con procesos organizacionales. [AIIM, 2016]

Teniendo en cuenta los conceptos anteriormente expuestos, se define que un sistema de gestión documental es utilizado para rastrear y almacenar documentos electrónicos y/o imágenes digitales de documentos originalmente en soporte físico, o sea, en papel. Ya que proporcionan un medio para el almacenamiento seguro y centralizado de la información. Son utilizados con el objetivo de administrar el flujo de los documentos, permitiendo controlar los accesos a estos, de forma tal que solo las personas autorizadas tengan permisos sobre ellos.

1.2. Centros productivos en la UCI.

Los centros productivos en la UCI tiene como misión desarrollar productos, servicios y soluciones para distintas ramas de las Ciencias Informáticas, contribuyendo a la formación integral de profesionales y permitiendo un posicionamiento en el mercado nacional e internacional. Dichos centros integran la producción e investigación en el desarrollo de aplicaciones informáticas.

La actividad de desarrollo–producción de la UCI se soporta sobre la Red de Centros, integrada por 14 centros de producción de software. Existen otros 2 entidades que brindan servicios transversales (Direc-

ción de Calidad de y Centro de Soporte). Dichos centros se encuentran vinculados a la Vicerrectoría de producción o a las distintas facultades. A continuación, se hace una relación de ellos.

- Vinculados a la Vicerrectoría de producción.
 1. Centro de Soporte.
 2. Consultoría de Desarrollo de Arquitecturas Empresariales (CDAE).
 3. Dirección de Calidad.
 4. Informática Médica (CESIM).

- Vinculados a las distintas Facultades.
 1. Centro de Desarrollo Geoinformática y Señales Digitales (GEYSED).
 2. Centro de Entorno 3D y Realidad Virtual (VERTEX).
 3. Centro de Gobierno Electrónico (CEGEL).
 4. Centro de Identificación y Seguridad Digital (CISED).
 5. Centro de Informática Industrial (CEDIN).
 6. Centro de Informatización de Entidades (CEIGE).
 7. Centro de Informatización de la Gestión Documental (CIGED).
 8. Centro de Investigación y Desarrollo de Internet (CIDI).
 9. Centro de Software Libre (CESOL).
 10. Centro de Tecnologías de Gestión de Datos (DATEC).
 11. Centro de Tecnologías para la Formación (FORTES).
 12. Centro de Telemática (TLM).

Expedientes de proyecto en la UCI.

La Dirección de Calidad de la UCI es el encargado del aseguramiento de la calidad, medición, análisis, validación de software, y otros aspectos que contribuyan a la mejora de la calidad de procesos y productos. Por tal razón la Dirección de Calidad fue el encargado de definir como están conformados los expedientes de proyecto para la red de centros de la UCI. Para la confección de los artefactos que lo componen se hizo uso en su mayoría de la metodología RUP (*Rational Unified Process*).²

²El Proceso Racional Unificado o RUP (por sus siglas en inglés de Rational Unified Process) es un proceso de desarrollo de software desarrollado por la empresa Rational Software.

El GDA Xabal eXcriba para la Red Centros cuentan con diferentes expedientes de proyecto. Los mismo responden a la necesidad de la gestión de desarrollo de software en los diferentes centros productivos.

Certificaciones.

Actualmente la UCI como centro productivo se encuentra en un proceso de mejoras basados en el modelo CMMI³ con el objetivo de certificar los proceso de desarrollo como organización productora de software. Por tal motivo se realizan modificaciones en los expedientes de proyecto, respondiendo a las prácticas genéricas de CMMI nivel III de madurez. Alcanzar este nivel significa que la forma de desarrollar proyectos está definida, es decir, está establecida y documentada. Mediante estas prácticas genéricas se permite institucionalizar en la organización las prácticas establecidas en cada una de las áreas de procesos, de manera que los procesos puedan evolucionar.

1.3. Gestión de expedientes de proyecto en el GDA Xabal eXcriba 3.0

En la siguiente figura se muestra el proceso de gestión de un nuevo expediente de proyecto en la Red de Centros de la UCI o Áreas.

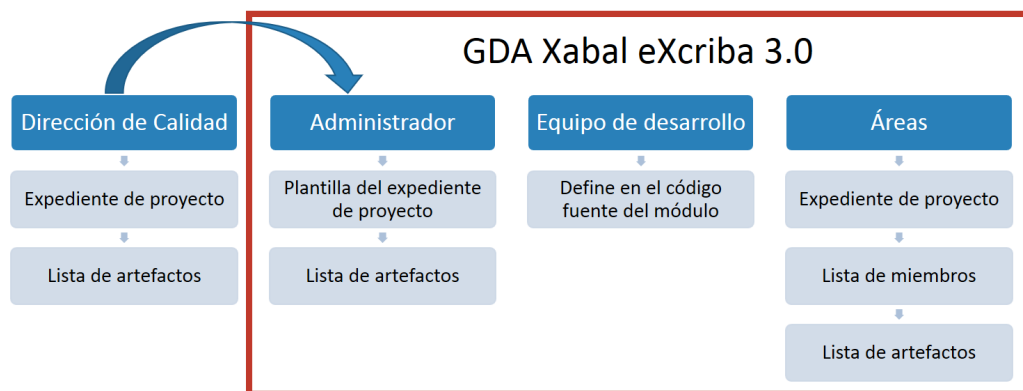


Figura 1.1: Proceso de gestión de un nuevo expediente de proyecto.

El proceso de despliegue de un nuevo expediente de proyecto, inicia cuando la Dirección de Calidad establece un nuevo expediente de proyecto (estructura de carpetas y documentos) en conjunto con un listado donde se establece la relación documento-roles (lista de artefactos). Estos elementos son entregados al administrador del sistema por parte de un especialista de la Dirección de Calidad, que posteriormente los

³Integración de modelos de madurez de capacidades o Capability Maturity Model Integration por sus siglas en ingles.

incorpora al sistema de forma manual, alargando el período de tiempo para su explotación. Posteriormente el administrador replica la lista de artefactos por las 14 áreas que comprenden los centros que producen software, lo que trae como consecuencia la duplicidad de la información en el sistema. A continuación, el equipo de desarrollo tiene que incorporar al código fuente del módulo las nuevas especificaciones del expediente de proyecto y posteriormente efectuar el reinicio del sistema obstaculizando las actividades de las áreas. Para finalmente poder ser utilizados los expedientes por las áreas con sus respectivas listas de miembros y de artefactos.

Seguidamente en las áreas para poder establecer el control de acceso a los elementos del expediente de proyecto, cuenta con la lista de miembros. En la actualidad la gestión de la lista de miembros no cuenta con la actualización automática de los permisos; como consecuencia de ello para poder actualizar los permisos el administrador de cada área tiene que dirigirse al administrador del sistema para realizar el proceso de forma manual. Para poder actualizar la lista de artefactos el administrador del sistema tiene que modificar cada una de las listas existentes en cada centro de forma manual. El módulo no cuenta con la posibilidad de poder actualizar un expediente de proyecto.

Por lo anteriormente planteado se puede afirmar que la gestión de expedientes de proyecto en el GDA Xabal eXcriba 3.0 presenta limitantes y deficiencias, que obstaculizan y retrasan los procesos productivos en la Red de Centros de la UCI así como su evolución, atentando contra el proceso de certificación.

1.4. GDA Xabal eXcriba 3.1

El GDA Xabal eXcriba 3.1 está diseñado para tramitar los documentos administrativos que se generan o reciben dentro de las organizaciones a partir de sus funciones, por lo tanto, involucra todas las áreas de una organización, permitiéndoles gestionar de forma correcta la documentación como prueba, testimonio y evidencia de las actividades organizacionales. La solución de software agiliza el trámite de los documentos, permitiendo controlar el estado de los mismos, evitando la pérdida de información y tributando como herramienta de apoyo para la toma de decisiones.

El sistema, permite trabajar en un entorno de colaboración entre los usuarios de la organización, compartiendo sus documentos con otros usuarios y estableciendo niveles de acceso y permiso. Además, optimiza la organización de los documentos, posibilita la búsqueda de información y gestiona los documentos durante su ciclo de vida [UCI, 2015]. Durante la gestión de los documentos incluye las 3 primeras fases propuestas por la ISO 15489 para el ciclo de vida de un documento: creación, revisión/aprobación y publicación, manteniendo la seguridad de los documentos en formato electrónico. [Lamas, 2007]

El GDA Xabal eXcriba 3.1 propone un conjunto de permisos asociados a los roles que pueden desempeñar los usuarios o grupos de usuarios en el sistema, para garantizar la seguridad de los medios electrónicos en el sistema, a continuación, se describen.

Coordinador: Todos los permisos en el sistema.

Contribuidor: Leer documentos y carpetas, leer propiedades de las carpetas, copiar carpetas y documentos, crear documentos en las carpetas.

Colaborador: Leer documentos y carpetas, leer propiedades de las carpetas, copiar carpetas y documentos, crear documentos en las carpetas, editar contenidos, editar propiedades de carpetas y documentos, realizar una copia de trabajo, actualizar una copia de trabajo.

Editor: Leer documentos y carpetas, leer propiedades de las carpetas, copiar carpetas y documentos, editar propiedades de carpetas y documentos, realizar una copia de trabajo, actualizar una copia de trabajo.

Consumidor: Leer documentos y carpetas, leer propiedades de las carpetas, copiar carpetas y documentos.

El GDA Xabal eXcriba 3.1 incorpora varias funcionalidades dentro de las que se encuentran:

- Gestión de acuerdos.
- Gestión de grupos y usuarios por parte de los administradores de áreas.
- Gestión de notas.
- Gestión de notificaciones.
- Gestión de reportes.

Los beneficios de la actualización del GDA Xabal eXcriba en la red centros a la versión 3.1, se reflejan en la incorporación de todos los módulos desarrollados por el proyecto, que representan un valor agregado al sistema. Contribuyendo al uso de una mayor cantidad de recurso enfocados al proceso de gestión documental.

1.5. Metodologías de desarrollo de software.

Una metodología de desarrollo de software se refiere a un *framework*⁴ que es usado para estructurar, planear y controlar el proceso de desarrollo en sistemas de información. A lo largo del tiempo, una gran cantidad de métodos han sido desarrollados diferenciándose por su fortaleza y debilidad.

1.5.1. Metodología AUP-UCI.

El Proceso Unificado Ágil de Scott Ambler⁵ o Agile Unified Process (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP [Rodríguez, 2014].

La metodología de desarrollo de software AUP para la UCI, es una variedad creada por la universidad a la metodología ágil AUP en unión con el modelo CMMI-DEV v 1.3, la que 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. La metodología AUP-UCI está definida por la casa de altos estudios como el escrito oficial que rige la actividad productora. Para la realización de este trabajo se empleó el escenario No. 2 de la metodología AUP-UCI. El cual se aplica a los proyectos que hayan evaluado 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. [Rodríguez, 2014]

1.6. Lenguajes de programación.

Son herramientas que permiten crear programas. Representan en forma simbólica y en manera de un texto los códigos que podrán ser leídos por una persona. Los lenguajes utilizados en el desarrollo de la presente investigación son:

⁴Un *framework* o entorno de trabajo es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar. [Bhaumik, 2011]

⁵ Scott W. Ambler (nacido en 1966) es un ingeniero de software canadiense, consultor y autor. Es autor de una serie de libros centrados en el marco de decisión del proceso de Entrega Ágil Disciplinado, el proceso Unificado, el desarrollo de software Agile, el Lenguaje de Modelado Unificado y el desarrollo del Modelo de Madurez de Capacidades (CMM).

Lenguaje Unificado de Modelado (UML v2.0).

El UML (Lenguaje Unificado para la Construcción de Modelos) se define como un "lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas de software". Es un sistema notacional (que, entre otras cosas, incluye el significado de sus notaciones) destinado a los sistemas de modelado que utilizan conceptos orientados a objetos [Larman, 2003]. Este lenguaje permite que personas con poco conocimiento de programación puedan participar en el análisis y diseño de un sistema. Para la modelación de los artefactos que son generados durante el proceso de desarrollo del software se hará uso de este lenguaje de modelado.

JavaScript v1.6.

Es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. 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. [Eguiluz, 2015]

Utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos. Con JavaScript se pueden crear diferentes efectos e interactuar con los usuarios. Es un lenguaje seguro y es independiente de la plataforma. [Eguiluz, 2015]

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

XML v1.0.

XML (en inglés Extensible Markup Language) se define como un conjunto de reglas para especificar etiquetas semánticas que organizan un documento en partes. Es un metalenguaje que detalla la sintaxis utilizada para definir otros lenguajes de etiquetas estructuradas. Entre sus principales ventajas se encuentran [Montalvo, 2015]:

- 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.
- El número de etiquetas es ilimitado.

Es un meta-lenguaje que permite diseñar lenguajes de marcado adecuados a usos determinados. Permite definir etiquetas personalizadas para descripción y organización de datos. Permite representar información estructurada en la web (todos documentos), de modo que esta información pueda ser almacenada, transmitida, procesada, visualizada e impresa, por muy diversos tipos de aplicaciones y dispositivos. [EXES, 2016]

En el presente trabajo se hace uso de XML v1.0 para la implementación de los modelos de contenido, porque los ficheros asociados a estos deben tener extensión XML para ser interpretados por Alfresco.

1.7. Herramientas informáticas.

Una herramienta de desarrollo de software es un programa informático que usa un programador para crear, depurar, gestionar o mantener un programa [Mastermagazine, 2010]. A continuación, se hará un breve resumen de las herramientas necesarias para el cumplimiento del objetivo general.

PostgreSQL v9.4.

PostgreSQL es un servidor de base de datos objeto relacional libre, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional, liberado bajo la licencia BSD. Como muchos otros proyectos Open Source, el desarrollo de PostgreSQL no es manejado por una sola compañía, sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo. [Martinez, 2016]

Ventajas:

- Ampliamente popular - Ideal para tecnologías Web.
- Fácil de Administrar.
- Su sintaxis SQL es estándar y fácil de aprender.
- Uso bajo de memoria, bastante poderoso con una configuración adecuada.
- Multiplataforma.

- Capacidades de replicación de datos.
- Soporte empresarial disponible.

Administrador de base de datos

Un administrador de base de datos dirige o lleva a cabo todas las actividades relacionadas con el mantenimiento de un gestor de base de datos. Las responsabilidades incluyen el diseño, implementación y mantenimiento del sistema de base de datos; el establecimiento de políticas y procedimientos relativos a la gestión y la seguridad. [Martinez, 2016]

PgAdmin III es una herramienta de código abierto para la administración de bases de datos. Está diseñada para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL hasta desarrollar bases de datos complejas. La interfaz gráfica soporta todas las características de PostgreSQL y hace simple la administración. Está disponible en más de una docena de lenguajes y para varios sistemas operativos, incluyendo Microsoft Windows y Linux.

Por los aspectos anteriormente expuestos se selecciona como gestor de base de datos PostgreSQL v 9.4 y PgAdmin III como sistema de administración de base de datos para el desarrollo del módulo.

Visual Paradigm v8.0.

Visual Paradigm es una herramienta CASE⁶ concebida para soportar el ciclo de vida completo del proceso de desarrollo de software a través de la representación de todo tipo de diagramas. La misma proporciona ayudas para el desarrollo de programas informáticos, desde la planificación, el análisis y diseño, hasta la generación de código fuente de los programas. Teniendo en cuenta las características citadas anteriormente, Visual Paradigm para UML es la herramienta que se utilizará para la realización del modelado de los diagramas. [Paradigm, 2016]

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.
- Coherencia entre diagramas: Al disponer de un repositorio común, es posible visualizar el mismo elemento en varios diagramas, evitando duplicidades.

⁶Ingeniería de Software Asistida por Computadora. Son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas desde el punto de vista de tiempo y de dinero.

- 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 es una herramienta multiplataforma para modelado UML, facilita la interoperabilidad⁷ con diferentes sistemas, permite la ingeniería inversa, es decir, invertir modelos UML desde el código fuente, genera diferentes tipos de código fuente y captura requisitos mediante el modelado de los casos de uso.

Visual Studio Code v1.23

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. También es personalizable, por lo que se pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto [Microsoft, 2016]. También ofrece la opción IntelliSense que sugiere sentencias según se va escribiendo. Asimismo permite la integración con repositorios como Git. Permite incorporar plugins de tercero, permitiendo que el desarrollo sea más fluido.

Por lo anteriormente expuesto se escoge el Visual Studio Code v1.23 como editor de código por la flexibilidad que brinda para el desarrollo de aplicaciones, además que su mayor potencial se encuentra en las extensiones, y es que desde el mismo editor se puede acceder a una gran variedad de extensiones y complementos que permitirá dotarle de las funciones y características necesarias para el desarrollo del módulo.

1.8. Tecnologías informáticas.

ECM Community Alfresco 4.2.f

Alfresco es un ECM, normalmente utilizado para gestión de documentos digitales en entornos exigentes. Es un proyecto OpenSource creado en 2005 por un equipo procedente de Documentum. Además, gestiona todos los contenidos en la empresa: documentos, imágenes, fotos, páginas web, registros, documentos

⁷Habilidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada.

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. Con la plataforma Alfresco, el contenido y los procesos aúnan fuerzas para optimizar el flujo de la información en toda la organización. Para asegurar que el contenido está en línea con los objetivos del negocio. [Alfresco Software, 2015a]

La actual versión del GDA Xabal eXcriba 3.1, emplea como núcleo del sistema la versión liberada 4.2f de Alfresco, permitiéndole la capacidad de definir modelos de contenidos con metadatos personalizados, flujos de trabajo y para cada flujo espacios inteligentes con características adicionales como búsqueda local, reglas de contenido y notificaciones.

Modelo de Contenido.

Alfresco incorpora un metamodelo en su estructura, es decir, la capacidad para que declarar nuevos modelos de contenido. Básicamente un modelo contenido es la estructura que define qué 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”. [Alfresco Software, 2016b]

Un modelo de contenido presenta la siguiente información clave:

- Los tipos de datos fundamentales y como deben persistirse en base de datos. Sin un modelo de contenido, Alfresco no sabría la diferencia entre un literal y una fecha.
- Los tipos documentales (tipos de datos de alto nivel) como carpetas y contenidos, así como los tipos documentales personalizados (procedimiento, factura, contrato).
- Aspectos como “auditable” o “clasificable” o aspectos personalizados como “Datos de factura” o “Datos de empresa”.
- Propiedades (o metadatos) específicos de cada tipo de contenidos.
- Ligaduras (*constraints*) de ciertas propiedades, por ejemplo, una lista restringida de datos.
- Como indexar el contenido para las búsquedas.
- Las relaciones (asociaciones) entre los tipos documentales.

Reglas y acciones.

Cuando se configura una regla en Alfresco, hay muchas acciones predeterminadas disponibles. Las acciones seleccionadas se realizan en archivos que cumplen los criterios del evento y las condiciones que ha seleccionado. Las acciones no se aplican a los archivos en las subcarpetas, a menos que la regla que se aplica a la opción de subcarpetas se seleccione antes de que se cree una regla. Una acción representa una unidad de trabajo parametrizada que se puede aplicar a un nodo⁸. Usando el servicio de acción, se pueden crear acciones de tipos específicos.

El tipo de acción indica la implementación de la acción que se usará cuando esta sea ejecutada y la información sobre cómo se puede parametrizar la acción. Los valores de los parámetros se pueden establecer en una acción. Algunas acciones pueden tener parámetros obligatorios. Estos deben establecerse antes de la ejecución. Los detalles de los tipos de parámetros, nombres y opcionalidad se pueden encontrar en la definición de acción. Estos están disponibles a través de la API del servicio de acción. El servicio de acción se puede usar para ejecutar una acción contra un nodo, llamado el nodo activado. [Alfresco Software, 2016a]

Las acciones son útiles cuando:

- Desea definir una o más operaciones que pueden ejecutarse repetidamente.
- Desea facilitar que los usuarios finales invoquen operaciones comunes, ya sea haciendo clic en un elemento del menú.
- Configurar una regla en una carpeta que ejecutará las operaciones automáticamente.

API JavaScript.

Una Interfaz de Programación de Aplicaciones (o API, por sus iniciales en inglés) es un conjunto de funciones previamente implementadas que brindan al programador una interfaz a través de la cual comunicarse con un sistema determinado, añadiéndole nuevas funcionalidades. La API de repositorio de JavaScript permite desarrollar archivos compatibles con JavaScript 1.6 para acceder, modificar y crear objetos de repositorio de Alfresco, como nodos, aspectos y propiedades. Proporciona una API simple, clara y orientada a objetos. [Alfresco Software, 2016c]

La API de JavaScript es capaz de realizar varias funciones esenciales, como, por ejemplo:

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

⁸En Alfresco todos los elementos son tratados como nodos, pero con diferentes propiedades.

- Transformación: se puede transformar el contenido. Por ejemplo, si se desea generar una versión PDF de un documento de MS-Office.
- Etiquetado: la API de etiquetado ayuda a crear etiquetas para los contenidos.
- Clasificación: se puede categorizar o clasificar los contenidos que se utiliza.
- Personas: con el uso de esta API, se puede manejar todas las operaciones relacionadas con el usuario y grupo de usuario, como la creación de un nuevo usuario y cambiar la contraseña de un usuario.
- Búsqueda: permite buscar el contenido haciendo uso de esta API, basada en *Lucene* o *XPath*.
- Operaciones de nodo: se utiliza esta API para realizar varias funciones relacionadas con el nodo como administrar propiedades, gestionar los aspectos, copiar, eliminar y mover.

Webscript.

Un Web script es un servicio mapeado a una URI que responde ante peticiones HTTP de tipo *GET*, *POST*, *PUT* y *DELETE*. Es por eso, que el desarrollo de Web Scripts permite construir fácilmente Servicios Web para gestión de contenidos identificados por una URI y accesibles vía HTTP, es decir, integraciones *REST*. Permite acceder, manejar y enlazar los contenidos a través de una API tipo *REST*.

Los Web Scripts son elementos clave dentro de Alfresco. Permite construir servicios web para la gestión de contenidos accesibles vía HTTP, es decir, convierten el repositorio de Alfresco en un gestor de contenido controlador mediante un servidor HTTP. [[Alfresco Software, 2015b](#)]

Los Web Script se pueden escribir usando simplemente la API de JavaScript de Alfresco y las plantillas FreeMarker. Alfresco tiene incluido el *framework* de Web Script, una API basada en tecnologías RESTful que proporciona una forma fácil, rápida y potente de interactuar con el repositorio de contenido y de integrar Alfresco con otros sistemas.

jQuery Library.

jQuery es una librería de JavaScript OpenSource. Entre sus características destaca la facilidad para manipular el contenido de un documento y la independencia del navegador para gestionar los eventos. Es una biblioteca de funciones JavaScript útiles para todo tipo de proyectos. jQuery brinda una cantidad de utilidades ya implementadas y listas para usar. Uno de sus principales fuertes es su conjunto de facilidades para manejar elementos DOM (Document Object Model). Se trata de una Interfaz de Programación de Aplicaciones

que permite acceder y modificar dinámicamente la estructura de documentos HTML, XHTML y XML, con plena interacción entre lenguajes y diferentes navegadores. [JSON.org, 2014]

Se escoge JQuery como *framework* de JavaScript para el desarrollo de la solución ya que posee una amplia variedad de plugins, widgets (componentes visuales) y utilidades que permiten tener un control total de las interfaces de usuario. Además, permiten el intercambio asíncrono de datos entre cliente y servidor de manera sencilla.

1.9. Conclusiones del capítulo.

Luego de finalizado el capítulo se concluye que se presentó la fundamentación teórica que guiará el proceso de desarrollo del módulo. Se trataron conceptos generales acerca del tema en cuestión. Se expusieron los aspectos más significativos relacionados con los expedientes de proyecto, siendo el punto de partida de la propuesta de solución. El análisis del GDA Xabal eXcriba 3.0 ha sido de gran importancia, porque permite reconocer diferentes funcionalidades del sistema, que son utilizadas, posteriormente como guía para la captura de requerimientos de la propuesta de solución. Finalmente se aprobaron la metodología de desarrollo, herramientas y tecnologías a utilizar, concluyendo que como metodología de desarrollo a utilizar será AUP-UCI en su escenario 2, como lenguajes de programación JavaScript v1.6, como lenguaje de modelado UML 2.0, como gestor de base de datos PostgreSQL en su versión 9.4, como editor de código fuente Visual Studio Code v1.23.1 y como herramienta CASE Visual Paradigm en su versión 8.0. Lo anteriormente expuesto brindo las pautas a seguir para el desarrollo del módulo para la gestión de expedientes de proyecto en el GDA Xabal eXcriba 3.1 para la red de centros de la UCI.

Capítulo 2

Módulo para la gestión de expedientes de proyecto en el GDA Xabal eXcriba 3.1: Propuesta de solución.

En este capítulo se realizará una descripción de la solución propuesta, proporcionando un mejor entendimiento del módulo. Para comprender el entorno en el que trabaja el módulo se incluirá un modelo conceptual con su respectivo diccionario de datos. También se especifican los requisitos funcionales y no funcionales que debe cumplir la solución. Además se presentan la descripción y representación de los casos de uso del sistema. Seguidamente se incluyen el diseño de la arquitectura de la solución propuesta.

2.1. Descripción de la propuesta de solución.

Con el objetivo de agilizar y facilitar la gestión de expedientes de proyecto de la Red de Centros de la UCI en el Gestor de Documentos Administrativos Xabal-eXcriba 3.1, se requiere la implementación de un módulo para la gestión de los mismos.

El módulo incorpora nuevas funcionalidades, capaces de cumplir las necesidades actuales del proceso. En una primera instancia se añade al GDA Xabal-eXcriba 3.1, la Dirección de calidad, como un área del sistema. Manteniendo sus funciones de definir la plantilla de los nuevos expedientes de proyecto y la lista de artefactos.

A nivel de usuario el administrador del sistema sera el único encargado de gestionar las áreas. El módulo pretende satisfacer la necesidad de gestionar los permisos sobre los expedientes de proyecto de forma automática en el momento de actualizar la lista de miembros del área. Además permite a los centros o áreas obtener los datos de la lista de artefactos directamente de la Dirección de calidad, eliminando la duplicidad e inconsistencia de la información.

A continuación, se muestra en la siguiente figura dicha propuesta de solución:

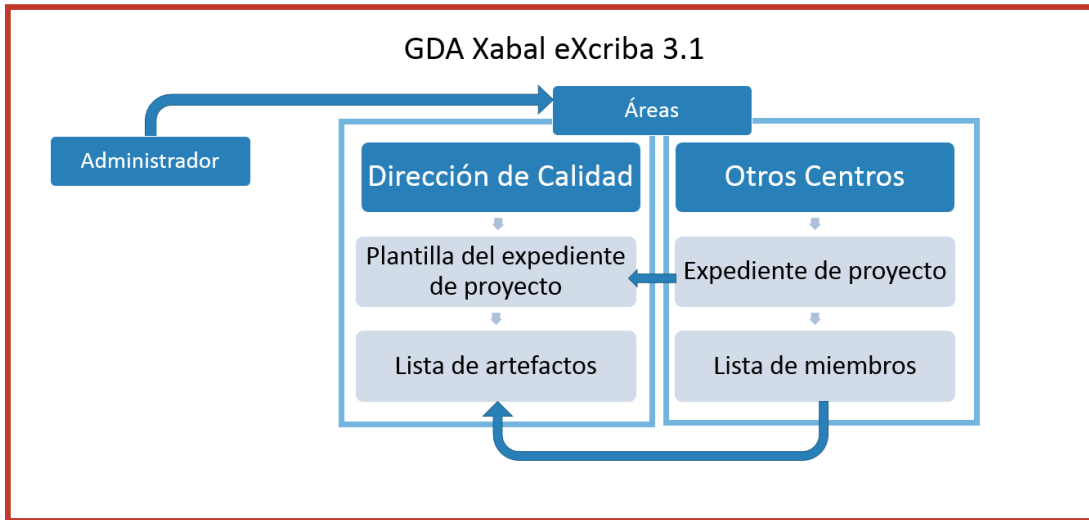


Figura 2.1: Propuesta de solución.

2.2. Modelo conceptual.

El modelo conceptual (también conocido como modelo de dominio) captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan los conceptos que existen o los eventos que suceden en el entorno en el que trabaja el sistema [Quesada, 2014]. A continuación, se muestra el modelo de dominio de la presente investigación:

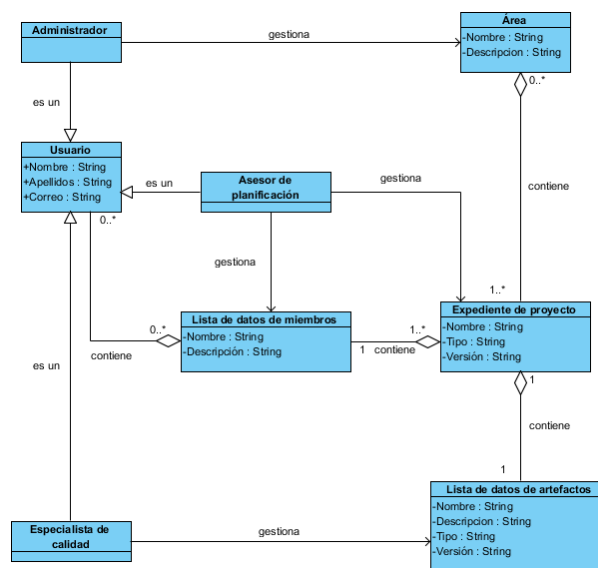


Figura 2.2: Modelo conceptual.

Descripción de las entidades

Usuario: es la persona que interactúa con el negocio.

Área: es el entorno de trabajo asociado a un centro, donde se gestionan los expediente de proyecto por los usuarios.

Expediente de Proyecto: es la estructura lógica de carpetas que contiene documentos referentes a un proyecto de desarrollo de *software*.

Lista de datos de miembros: es el listado donde se establecen la relación usuario-roles asociados a un expediente de proyecto.

Lista de datos de artefactos: es el listado donde se establecen la relación de documentos-roles que tiene un expediente de proyecto.

Administrador: es el administrador de área que posee privilegios administrativos en el sistema.

Asesor de planificación: es un usuario que posee privilegios para gestionar la lista de datos de miembros y el expediente de proyecto.

Especialista de calidad: es un usuario que posee privilegios para gestionar la lista de datos de artefactos del expediente de proyecto.

2.3. Diccionario de datos.

Un diccionario de datos, o repositorio de metadatos, como lo define el IBM Dictionary of Computing, un repositorio centralizado de información sobre datos tales como significado, relación con otros datos, origen, uso y formato. [[McDaniel y International Business Machines Corporation, 2015](#)]

En un diccionario de datos se encuentra la lista de todos los elementos que forman parte del flujo de datos en todo el sistema. Los elementos más importantes son flujos de datos, almacenes de datos y procesos. El diccionario guarda los detalles y descripciones de todos estos elementos. Si se desean conocer cuántos caracteres abarca un determinado dato o qué otros nombres recibe en distintas partes del sistema, o dónde se utiliza, es en el diccionario donde se encuentran reflejados estos datos. A continuación, se muestra el diccionario de datos correspondiente a la entidad Área :

Tabla 2.1: Diccionario de datos de la entidad Área

Descripción		<i>Esta entidad representa a un centro productivo en el sistema.</i>				
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
Nombre	Nombre propio del área en el Gestor de Documentos Administrativos Xabal eXcriba 3.1.	Cadena de caracteres.	No	No	A-Z, a-z, 0-9, algunos caracteres especiales	"" , * , \ , / , < , > , ? , ; , , \$
Descripción	Descripción del área en el Gestor de Documentos Administrativos Xabal eXcriba 3.1.	Cadena de caracteres.	No	No	A-Z, a-z, 0-9, algunos caracteres especiales	"" , * , \ , / , < , > , ? , ; , , \$

Para consultar los diccionario de datos de cada entidad restante, ver Anexo **A.1**.

2.4. Especificación de los requisitos de *software*.

Uno de los primeros procesos que se realizan en un proyecto de construcción de *software* es la especificación de requisitos. Los objetivos de este proceso son identificar, validar y documentar los requisitos de *software*; es decir determinar las características que deberán tener el sistema o las restricciones a cumplir para que sea aceptado por el cliente y los futuros usuarios del sistema.

La ingeniería de requisitos, como todas las demás actividades de la ingeniería del *software*, debe adaptarse a las necesidades del proceso, el proyecto, el producto y las personas que realicen el trabajo. La ingeniería de requisitos proporciona el mecanismo apropiado para entender lo que el cliente quiere, analizar las necesidades, evaluar la factibilidad, validar la especificación y administrar los requisitos. [Sommerville, 2011]

2.4.1. Técnicas para la captura de los requisitos.

La captura de requisitos es la actividad en la que un grupo especializado extrae, de cualquier fuente de información disponible (documentos, aplicaciones existentes y entrevistas), las necesidades de cubrir dicho sistema. El proceso de captura de requisitos puede resultar complejo, debido a esto existen un conjunto de técnicas que permiten hacer este proceso de una forma más eficiente y precisa, obteniéndose necesidades y modelos del sistema. Es por ello que surgen diferentes técnicas que ayudan a comprender el problema,

proponer soluciones, negociar diferentes puntos de vista y finalmente especificar un conjunto básico de requisitos de la solución.

Ejemplo de estas técnicas pueden ser: entrevistas, cuestionarios, tormentas de ideas, análisis de sistemas existentes, arqueología de documentos, prototipos, entre otras. Para la realización del levantamiento de requisitos del módulo se aplicaron las técnicas de Tormentas de ideas y análisis de los sistemas existentes.

Análisis de los sistemas existentes.

Mediante el análisis de sistemas existentes es posible estudiar aplicaciones similares a la que se necesita obtener. Una vez que se tiene la concepción del funcionamiento de un *software* similar en cuanto a funcionalidades y características es más sencillo identificar los requisitos del sistema que se necesita implementar. Durante la investigación se realizó un estudio del GDA Xabal eXcriba 3.0, en las cuales se observaron los diseños de sus interfaces, las funcionalidades que ofrecen, el grado de dificultad a la hora de interactuar con la aplicación, los colores que emplean, entre otros rasgos importantes que contribuyeran a obtener un producto con la mejor calidad posible.

Tormenta de ideas.

Es una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre. Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas. El grupo de personas que participa en estas reuniones no debe ser muy numeroso (máximo 10 personas), una de ellas debe asumir el rol de moderador de la sesión, pero sin carácter de controlador.

Se efectuaron encuentros con el equipo de desarrollo del proyecto Xabal eXcriba y con la Jefa de Departamento de la Dirección de Calidad de *software*, y a partir de un conjunto de ideas propuestas principalmente por la analista del GDA Xabal eXcriba y el administrador del sistema del GDA Xabal eXcriba para la red de centros, se definieron los requisitos con los que debería cumplir el módulo.

2.4.2. Requisitos funcionales.

Los requerimientos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera que este debe reaccionar a entradas en particulares y cómo se debe comportar en situaciones particulares. En algunos casos, pueden declarar lo que el sistema no debe hacer [Somerville, 2011].

Se aplicaron las técnicas antes expuestas para la obtener una visión general de las funcionalidades del módulo.

A continuación, se enumeran los requisitos identificados:

RF_ 1: Crear área

Descripción: el sistema debe permitir al administrador del sistema entrar los datos necesarios para crear un nuevo área.

RF_ 2: Modificar área

Descripción: el sistema debe permitir al administrador del sistema modificar los datos de un área.

RF_ 3: Eliminar área

Descripción: el sistema debe permitir al administrador del sistema eliminar un área.

RF_ 4: Crear lista de datos de artefactos

Descripción: el sistema debe permitir al especialista de calidad, perteneciente al área de la Dirección de Calidad, entrar los datos necesarios para crear una lista de datos de artefactos.

RF_ 5: Modificar lista de datos de artefactos

Descripción: el sistema debe permitir al especialista de calidad, perteneciente al área de la Dirección de Calidad, modificar los atributos de una lista de datos de artefactos.

RF_ 6: Eliminar lista de datos de artefactos

Descripción: el sistema debe permitir al especialista de calidad, perteneciente al área de la Dirección de Calidad, eliminar una lista de datos de artefactos.

RF_ 7: Adicionar elemento en la lista de datos de artefactos

Descripción: el sistema debe permitir al especialista de calidad, perteneciente al área de la Dirección de Calidad, adicionar un nuevo elemento en la lista de datos de miembros.

RF_ 8: Modificar elemento en la lista de datos de artefactos

Descripción: el sistema debe permitir al especialista de calidad, perteneciente al área de la Dirección de Calidad, modificar un elemento en la lista de datos de artefactos.

RF_ 9: Eliminar elemento en la lista de datos de artefactos

Descripción: el sistema debe permitir al especialista de calidad, perteneciente al área de la Dirección de Calidad, eliminar un elemento en la lista de datos de artefactos.

RF_ 10: Crear lista de datos de miembros

Descripción: el sistema debe permitir al asesor de planificación del área entrar los datos necesarios para crear una lista de datos de miembros.

RF_ 11: Modificar lista de datos de miembros

Descripción: el sistema debe permitir al asesor de planificación del área modificar una lista de datos de miembros.

RF_ 12: Eliminar lista de datos de miembros

Descripción: el sistema debe permitir al asesor de planificación del área eliminar una lista de datos de miembros.

RF_ 13: Adicionar elemento en la lista de datos de miembros

Descripción: el sistema debe permitir al asesor de planificación del área adicionar un nuevo elemento en la lista de datos de miembros.

RF_ 14: Modificar elemento en la lista de datos de miembros

Descripción: el sistema debe permitir al asesor de planificación del área modificar un elemento en la lista de datos de miembros.

RF_ 15: Eliminar elemento en la lista de datos de miembros

Descripción: el sistema debe permitir al asesor de planificación del área eliminar un elemento en la lista de datos de miembros.

RF_ 16: Crear plantilla del expediente de proyecto

Descripción: el sistema debe permitir al especialista de calidad, perteneciente al área de la Dirección de Calidad, crear una plantilla del expediente de proyecto.

RF_ 17: Eliminar plantilla del expediente de proyecto

Descripción: el sistema debe permitir al especialista de calidad, perteneciente al área de la Dirección de Calidad, eliminar una plantilla del expediente de proyecto.

RF_ 18: Crear el expediente de proyecto

Descripción: el sistema debe permitir al asesor de planificación crear un nuevo expediente de proyecto.

RF_ 19: Modificar el expediente de proyecto

Descripción: el sistema debe permitir al asesor de planificación modificar un expediente de proyecto.

RF_ 20: Establecer los permisos en el expediente de proyecto

Descripción: el sistema debe permitir al asesor de planificación establecer los permisos sobre un expediente de proyecto, cuando actualice la .

RF_ 21: Modificar los permisos en el expediente de proyecto

Descripción: el sistema debe permitir al asesor de planificación modificar los permisos sobre un expediente de proyecto, cuando actualice la .

RF_ 22: Eliminar los permisos en el expediente de proyecto

Descripción: el sistema debe permitir al asesor de planificación eliminar los permisos sobre un expediente de proyecto, cuando actualice la .

RF_ 23: Crear documento a partir de plantilla

Descripción: el sistema debe permitir al usuario crear documentos de los diferentes tipos de expedientes de proyecto a partir de plantillas.

RF_ 24: Modificar documento creado a partir de plantilla

Descripción: el sistema debe permitir al usuario modificar los documentos creados a partir de plantillas.

RF_ 25: Eliminar documento creado a partir de plantilla

Descripción: el sistema debe permitir al usuario eliminar los documentos creados a partir de plantillas.

2.4.3. Requisitos no funcionales.

Los requisitos no funcionales son los requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste, como la fiabilidad y el tiempo de respuesta. [Booch, Jacobson, 2006]

Definen las características o cualidades generales que se esperan de un sistema y establecen restricciones sobre el producto, el proceso de desarrollo de *software* y establecen restricciones externas que el *software* debe lograr.

Usabilidad Estos requerimientos describen los niveles apropiados de usabilidad, dados los usuarios finales del producto.

RnF_ 1: El sistema sólo podrá ser utilizado por los usuarios registrados en él.

RnF_ 2: El sistema debe presentar un acceso fácil y rápido, para facilitar el uso del mismo por usuarios con pocos conocimientos en el campo de la informática.

Seguridad

RnF_ 3: La información estará protegida contra accesos no autorizados utilizando mecanismos de autenticación propios del sistema.

RnF_ 4: La autenticación será la primera acción del usuario en el sistema y consistirá en suministrar un nombre de usuario único y una contraseña que debe ser de conocimiento exclusivo de la persona que se autentica. Si el usuario autenticado no se encuentra registrado se debe reportar un error de acceso.

Hardware para el cliente

RnF_ 5: Se recomienda que tenga al menos 2 GB de memoria RAM, 500 GB de disco duro, 2 núcleos de procesadores como mínimo en el cliente.

Hardware para el servidor

RnF_ 6: Se recomienda que tenga al menos 8GB de memoria RAM y se recomienda 1 TB o más, 4 núcleos de procesadores.

Software para el cliente

RnF_ 7: Se recomienda usar Mozilla Firefox v57 o superior.

2.5. Definición de los casos de uso del sistema.

Un caso de uso especifica una secuencia de acciones, incluyendo variantes, que el sistema puede llevar a cabo, y que producen un resultado observable de valor para un actor concreto [Booch, Jacobson, 2006]. Es decir un caso de uso describe lo que el sistema debe ser capaz de hacer. Muestra la respuesta a una secuencia de acciones realizada por el actor.

A continuación se establecen los casos de uso para el módulo:

CU_ 1: Gestionar área.

CU_ 2: Gestionar plantilla del expediente de proyecto.

CU_ 3: Gestionar lista de datos de artefactos.

CU_ 4: Gestionar elemento de la lista de datos de artefactos.

CU_ 5: Gestionar lista de datos de miembros.

CU_ 6: Gestionar expediente de proyecto.

CU_ 7: Gestionar elemento de la lista de datos miembro .

CU_ 8: Gestionar documento a partir de plantilla.

2.5.1. Definición de los actores del sistema.

Un actor en un conjunto coherente de roles que los usuarios de casos de uso desempeñan cuando interaccionan con estos. [Booch, Jacobson, 2006].

Tabla 2.2: Definición de los actores del sistema.

Actor	Justificación
Usuario	Persona que posee los privilegios necesarios para autenticarse y formar parte de los usuarios del sistema. Mantiene las operaciones de crear o modificar documentos además de crear versiones de los documentos. Todas estas operaciones las puede realizar donde el usuario tenga los permisos apropiados.
Administrador	Es el usuario del sistema con los máximos privilegios en el sistema, es el encargado de gestionar las áreas en el sistema.
Especialista de Calidad	Es el usuario del sistema perteneciente al área <i>Calidad-UCI</i> , con los permisos para gestionar (administrador de área) los elementos de la lista de datos de miembros de un expediente de proyecto y gestionar la plantilla del expediente de proyecto.
Asesor de planificación	Es el usuario del sistema con la responsabilidad de gestionar los expedientes de proyecto del área.

2.5.2. Diagrama de caso de uso del sistema.

Un caso de uso expresa todas las formas de usar un sistema para alcanzar una meta particular para un usuario. En conjunto, los casos de uso le proporcionan todos los caminos útiles de usar el sistema e ilustran el valor que este provee. [Jacobson et al., 2013]

En la figura se muestra el diagrama de casos de uso del sistema, el cual ofrece claramente las acciones que puede realizar el usuario cuando interactúa con el módulo.

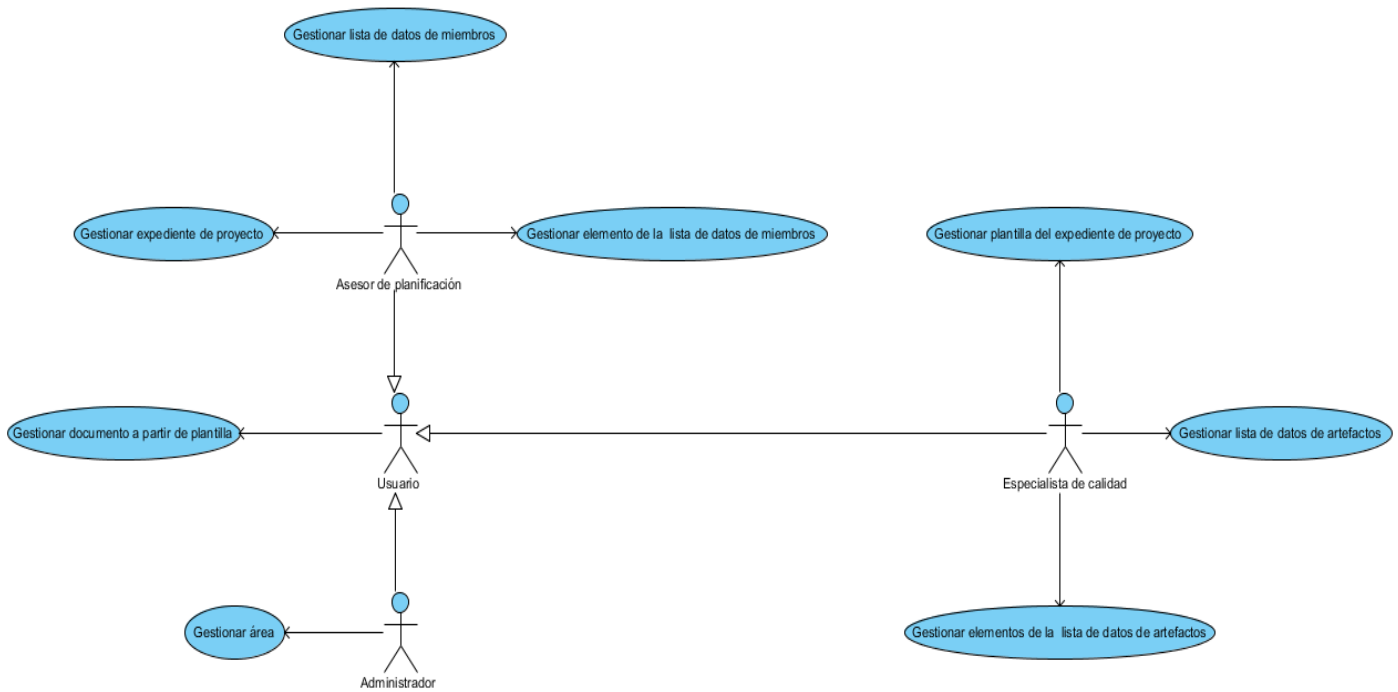


Figura 2.3: Diagrama de casos de uso del sistema.

2.5.3. Matriz de trazabilidad.

En este epígrafe se muestra mediante una matriz de trazabilidad como son recogidos todos los requisitos funcionales en los casos de uso identificados.

La matriz de trazabilidad consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo. Es necesario ir viendo que objetivos cubre cada requisito para detectar inconsistencias u objetivos no cubiertos.

Como se puede observar en la figura anterior cada requisito identificado ha sido cubierto por al menos un caso de uso del sistema, lo que significa que la especificación de caso de uso propuesta satisface todas las necesidades del cliente.

Para consultar las restantes matrices, ver Anexo **A.7**.

(25) Requirement		Crear área.	Modificar área.	Eliminar área.	Crear lista de datos de artefactos.	Modificar lista de datos de artefactos.	Eliminar lista de datos de artefactos.	Adicionar elemento en la lista de datos ...	Modificar elemento en la lista de datos ...	Eliminar elemento en la lista de datos ...	Crear lista de datos de miembros.	Modificar lista de datos de miembros.	Eliminar lista de datos de miembros.	Adicionar elemento en la lista de datos...	Modificar elemento en la lista de datos ...	Eliminar elemento en la lista de datos ...	Crear plantilla del expediente de proye...	Eliminar plantilla del expediente de pro...	Crear el expediente de proyecto.	Modificar el expediente de proyecto.	Establecer los permisos en el expedie...	Modificar los permisos en el expedient...	Eliminar los permisos en el expediente...	Crear documento a partir de plantilla	Modificar documento creado a partir de ...	Eliminar documento creado a partir de ...	
By: Transitor	(8) Use Case	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="radio"/> Gestionar área		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="radio"/> Gestionar lista de datos de artefactos		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="radio"/> Gestionar lista de datos de miembros		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="radio"/> Gestionar plantilla del expediente de proyecto		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="radio"/> Gestionar expediente de proyecto		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="radio"/> Gestionar documento a partir de plantilla		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
<input checked="" type="radio"/> Gestionar elementos de la lista de datos de artefactos		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="radio"/> Gestionar elemento de la lista de datos de miembros		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 2.4: Matriz de caso de uso a requisito.

2.5.4. Descripción del caso de uso 4: Gestionar elemento de la lista de datos de miembros.

Tabla 2.4: Descripción del CU_4: Gestionar elemento de la lista de datos de miembros.

Caso de Uso	Gestionar elemento de la lista de datos de miembros.
Objetivo	Actualizar elemento de la lista de datos de miembros.
Actores	Asesor de planificación: (Inicia) Crea, edita, y elimina los elementos en lista de datos de miembros.
Resumen	El caso de uso inicia cuando el actor selecciona crear, modificar o eliminar un elemento de la lista de datos de miembros, el sistema permite realizar la acción solicitada.
Complejidad	Media
Prioridad	Alta
Precondiciones	<ol style="list-style-type: none"> 1. El actor tiene los permisos necesarios y esta autenticado en el sistema. 2. El actor se encuentra en el entorno Lista de Datos del área. 3. El actor seleccionó la lista de datos de miembros a actualizar.
Post-condiciones	<ol style="list-style-type: none"> 1. Se actualizó un elemento. 2. Se actualizó los permisos en los expedientes de proyecto vinculados a la lista de datos de miembros.

Flujo básico: Gestionar usuario de la lista de datos de miembros.		
	Actor	Sistema
1.	Crea, modifica o elimina un elemento.	
2.		Permite realizar varias acciones con el elemento: <ul style="list-style-type: none"> - Adicionar un elemento. Ver Sección 1: Adicionar elemento. - Modificar un elemento. Ver Sección 2: Modificar elemento. - Eliminar un elemento. Ver Sección 3: Eliminar elemento.
3.		Termina el caso de uso.
Sección 1: Adicionar elemento.		
Flujo básico		
	Actor	Sistema
1.	Selecciona el botón "Nuevo elemento".	
2.		Muestra un formulario con los siguientes datos: <ul style="list-style-type: none"> - Miembro* -Roles*
3.	Insertar valores.	
4.	Presiona el botón "Aceptar".	
5.		Valida la información insertada.
6.		Crea el nuevo usuario con los datos insertados.
7.		Termina el caso de uso.
Flujos alternos		
4.a Presionar el botón "Cancelar".		
	Actor	Sistema
1.	Presiona el botón "Cancelar".	
2.		Cancela la acción y muestra la interfaz anterior.
3.		Termina el caso de uso.
5.a Existen campos obligatorios vacíos.		
	Actor	Sistema

Continúa en la siguiente página.

	Actor	Sistema
1		Muestra encima del campo vacío el mensaje: "El valor no puede estar vacío."
2		Continuar en el paso 3 del flujo básico.
Sección 2: Modificar elemento.		
Flujo básico		
	Actor	Sistema
1.	Presiona el ícono "Editar" del usuario escogido.	
2.		Muestra un formulario con los siguientes datos: - Miembro* -Roles*
3.	Insertar valores.	
4.	Presiona el botón "Aceptar".	
5.		Valida la información insertada.
6.		Edita el usuario con los datos insertados.
7.		Termina el caso de uso.
Flujos alternos		
4.a Presionar el botón "Cancelar".		
	Actor	Sistema
1.	Presiona el botón "Cancelar".	
2.		Cancela la acción y muestra la interfaz anterior.
3.		Termina el caso de uso.
5.a Existen campos obligatorios vacíos.		
	Actor	Sistema
1.		Muestra encima del campo vacío el mensaje: "El valor no puede estar vacío."
2.		Continuar en el paso 3 del flujo básico.

Continúa en la siguiente página.

Sección 4: "Eliminar elemento."		
Flujo básico		
	Actor	Sistema
1.	Presiona el ícono "Eliminar" del elemento escogido.	
2.		El sistema muestra la interfaz de confirmación de eliminación.
3.	Selecciona el botón "Eliminar".	
4.		El sistema elimina el usuario escogido.
5.		Se actualiza la lista.
6.		Termina caso de uso.
Flujos alternos		
3.a Presionar el botón "Cancelar".		
	Actor	Sistema
1	Presiona el botón "Cancelar".	
2		Cerrar la interfaz de confirmación de eliminación.

Para consultar las descripciones más detalladas de los caso de uso, ver Anexo **A.10**.

2.6. Arquitectura del sistema.

Una arquitectura de *software* define la forma de trabajar en un sistema, implica definir 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*. [Microsoft, 2013]

Patrón de arquitectura

Los patrones arquitectónicos son un conjunto de reglas de diseño que identifican las clases de componentes y conectores que se pueden utilizar para componer en sistema o subsistema, junto con las restricciones locales o globales de la forma en que la composición se lleva a cabo. [POSSO, 2010]

Como tecnología, las arquitecturas de n-capas proporcionan una gran cantidad de beneficios para las empresas que necesitan soluciones flexibles y fiables para resolver complejos problemas inmersos en cambios

constantes. 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.

A continuación se muestra la propuesta para el estilo arquitectónico del sistema:

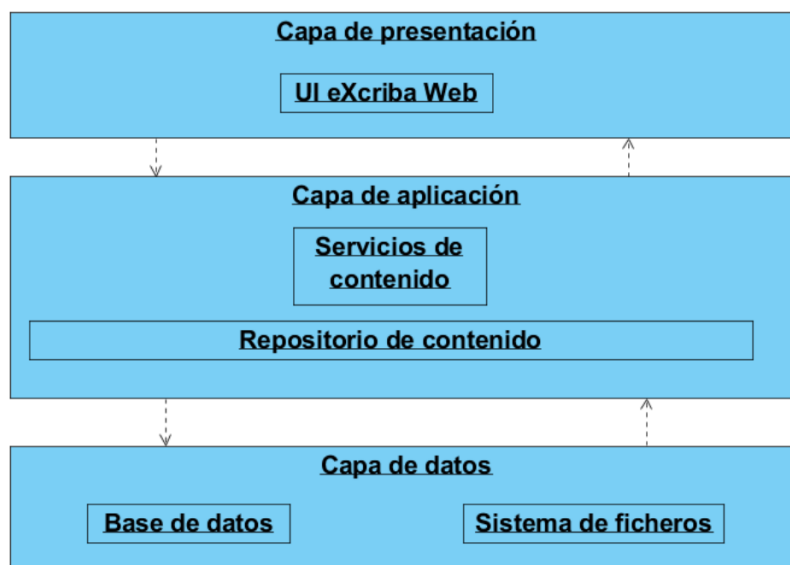


Figura 2.5: Estilo arquitectónico del sistema.

Capa de presentación: es la que 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.

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 acceso a datos y con la de presentación mediante el protocolo TCP/IP.

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 y la base de datos, permitiéndole a la capa de aplicación abstraerse de la forma en que deben persistir los datos y cómo deben ser recuperados.

2.7. Patrones de diseño.

Un patrón de diseño se caracteriza como “una regla de tres partes que expresa una relación entre cierto contexto, un problema y una solución”. Para el diseño de *software*, el contexto permite al lector entender el ambiente en el que reside el problema y qué solución sería apropiada en dicho ambiente. Un conjunto de requerimientos, incluidas limitaciones y restricciones, actúan como sistema de fuerzas que influyen en la manera en la que puede interpretarse el problema en este contexto y en cómo podría aplicarse con eficacia la solución. [Roger S. Pressman, 2010]

Se propone utilizar los Patrones Generales de *software* para Asignación de Responsabilidades (GRASP), debido a que estos describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Constituyen el fundamento de cómo se diseñará el sistema, es decir, una serie de buenas prácticas de aplicación recomendable en el diseño de *software*.

Los principales patrones GRASP son:

Experto: asigna la responsabilidad al experto en información, la clase que tiene información necesaria para cumplir la responsabilidad. Indica, por ejemplo, que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo.

```
/**
 * [Elimina los permisos asignados a un nodo]
 * @node El nodo a eliminarle los permisos
 * Return Void
 */
function removeAllPermission(node) {
    node.setInheritsPermissions(false);
    var permissions = node.getPermissions();
    if (permissions.length > 0) {
        for(var i in permissions) {
            var temp = permissions[i].split(";");
            node.removePermission(temp[2],temp[1]);
        }
    }
}
```

Figura 2.6: Ejemplo de patrones GRASP: Experto.

Bajo acoplamiento: permite tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases, potenciando la reutilización y disminuyendo la dependencia entre las clases.

```
var n = search.findNode(args.n);
model.result = jsonUtils.toJSONString(n.properties);
```

Figura 2.7: Ejemplo de patrones GRASP: Bajo acoplamiento.

Alta cohesión: caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen todo un trabajo enorme, de modo que la información que almacenan debe ser coherente y debe estar relacionada con la clase, permitiendo mejorar la claridad con que se entiende el diseño. [Larman, 2003]

```
function getInformationFromPMDDataList(datalist) {
    var node, result = {}, rol;
    for(var i in datalist.children) {
        node = datalist.children[i];
        var roles = node.properties['xfpm-dl:projectRoles'];
        if (node.assocs['xfpm-dl: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];
                }
            }
        } else {
            node.remove();
        }
    }
    return result;
}
```

Figura 2.8: Ejemplo de patrones GRASP: Alta cohesión.

2.8. Conclusiones del capítulo

Con el desarrollo de este capítulo se obtuvo una descripción de la propuesta realizada, se describieron las características del sistema en términos de requerimientos funcionales y no funcionales. Se logró identificar los actores y los casos de uso del sistema, los cuales fueron modelados gráficamente y se hizo una descripción de los mismos. Además se describió la arquitectura y los patrones de diseño que se utilizaran en el proceso de implementación.

Capítulo 3

Módulo para la gestión de expedientes de proyecto en el GDA Xabal eXcriba 3.1: Implementación y validación.

En el presente capítulo se muestra cómo está implementado el sistema y bajo qué condiciones se hizo, teniendo en cuenta los estándares de codificación. Además, se realizó la selección del tipo de prueba y las técnicas que se utilizaron para comprobar la validez del módulo. Posteriormente se describieron los valores empleados para las pruebas y para concluir una evaluación de su ejecución y los resultados obtenidos.

3.1. Implementación.

El modelo de implementación es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes podemos encontrar datos, archivos, ejecutables, código fuente y los directorios. [[Leovigilda Hernandez, 2013](#)]

3.1.1. Estándares de codificación.

Los estándares de código son parte de las llamadas buenas prácticas o mejores prácticas. Estas son un conjunto no formal de reglas que han ido surgiendo en las distintas comunidades de desarrolladores con el paso del tiempo y las cuales, bien aplicadas pueden incrementar la calidad del código. A continuación, se muestran los estándares de codificación que se utilizaron para la implementación del módulo:

Identación, llaves y líneas:

Identación y líneas: usar una indentación sin tabulaciones, con un equivalente a cuatro espacios. El uso de las llaves es seguido del nombre del método. La longitud de las líneas de código es aproximadamente de 75-80 caracteres para mantener la legibilidad del código.

Ejemplo:


```
function getInformationFromPMDDataList (datalist) {  
    //  
}
```

Figura 3.1: Estándares de codificación indentación y líneas

Variables: deben contener sólo letras minúsculas, en caso de ser nombres compuestos, se rigen por la nomenclatura camelCase. Siempre indicando su propósito y contenido. Las variables muy cortas sólo deben utilizarse como iteradores en bucles for.

Ejemplo:

```
function isUserDuplicate(group, shortName) {  
    var user = group.getChildUsers(),  
        flag = false;  
    for (var i in user) {  
        if (user[i].getShortName() == shortName) {  
            flag = true;  
            break;  
        }  
    }  
    return flag;  
}
```

Figura 3.2: Estándares de codificación variables

Funciones: se rigen por la nomenclatura camelCase. Siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula. Los parámetros son separados por espacio luego de la coma que los separa.

Ejemplo:

```
function isUserDuplicate(group, shortName) {  
    //  
}
```

Figura 3.3: Estándares de codificación funciones

3.2. Modelo de despliegue.

Describe la distribución física del sistema en términos de cómo las funcionalidades se distribuyen entre los nodos de computación sobre los que se va a instalar el sistema. Cada nodo representa un recurso

de computación. Los nodos tienen relaciones entre ellos que representan los medios de comunicación. La funcionalidad de un nodo viene representada por los componentes que se ejecutan en él. [Sparx, 2013].

3.2.1. Diagrama de despliegue.

Los diagramas de despliegue muestran las relaciones físicas entre los distintos nodos que componen un sistema y la distribución de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. [Leovigilda Hernandez, 2013]

A continuación, se muestra el diagrama de despliegue existente para el GDA Xabal eXcriba en la red de centros, el cual demuestra cómo están distribuidos los diferentes nodos. La misma distribución será empleada para el despliegue físico del módulo para la gestión de expedientes de proyecto en la Red de Centros de la UCI en el GDA Xabal eXcriba 3.1:

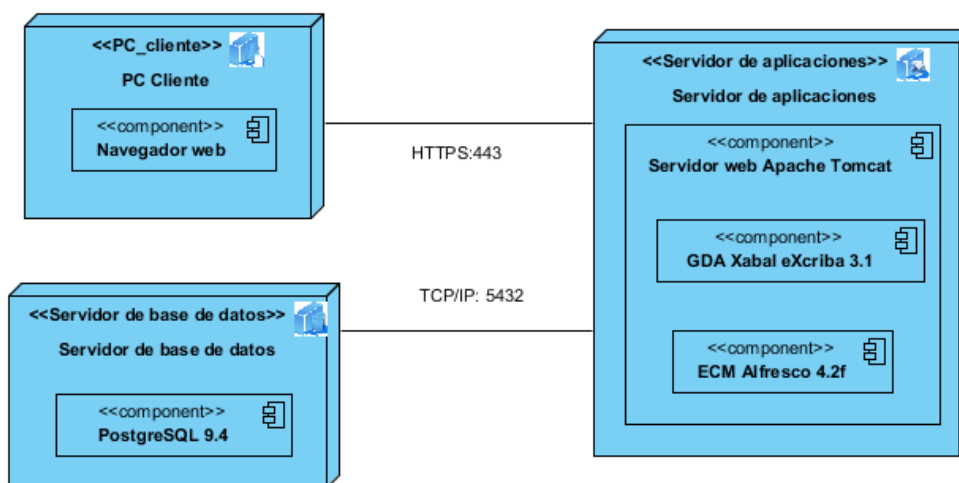


Figura 3.4: Diagrama de despliegue.

A continuación, se detalla una breve descripción de cada uno de los nodos presentes en el diagrama de despliegue.

PC_cliente: en este nodo es donde procesan todas las interfaces de usuario que han sido previamente implementadas. Se encuentra alojado el navegador web mediante el cual el usuario accederá a las interfaces del módulo utilizando el protocolo HTTPS.

Servidor de aplicaciones: 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 7.0.52, donde se encuentra desplegado el GDA Xabal eXcriba 3.1 y el ECM Alfresco 4.2f.

Servidor de base datos: es donde se encuentran los componentes asociados a la base de datos del sistema alojado en el PostgreSQL 9.4.

3.3. Pruebas del módulo

Las pruebas son procesos que se realizan con el fin de emitir una evaluación de un producto desde un punto de vista crítico, sometiéndolo a una serie de acciones que verifiquen si responde con los requerimientos definidos, determinando así la calidad del mismo. Los objetivos de las pruebas son:

- Planificación de pruebas necesarias para comprobar cada versión operacional del sistema (pruebas de integración y de sistema).
- Diseño e implementación de prueba, creando los casos de prueba para especificar qué probar y los procedimientos de prueba, estableciendo componentes de prueba para agilizar las pruebas.
- Realización de las pruebas y análisis de los resultados de las mismas con el propósito de determinar los defectos del sistema y poder corregirlos.

Para determinar si un sistema cumple con las expectativas del usuario final se define una etapa de pruebas, en las que se determinan los errores que presenta el software relacionados con los requisitos funcionales. Partiendo de lo planteado se decide realizar pruebas de caja negra para comprobar que el sistema de recomendación cumple con los requisitos especificados.

Pruebas internas

En esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas [Rodríguez, 2014].

3.3.1. Método de caja negra.

“Las pruebas de caja negra también denominadas pruebas de comportamiento se centran en los requisitos funcionales del software” [Roger S. Pressman, 2010]. Estas se llevan a cabo sobre la interfaz del mismo, obviando el comportamiento interno y la estructura del programa. El objetivo es verificar que el sistema cumpla con los requisitos funcionales establecidos por el cliente. En la misma los casos de prueba pretenden

demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada, que se produce una salida correcta y que la integridad de la información externa se mantiene.

A continuación, se muestra una lista de errores que tratan de encontrar las pruebas de caja negra, los errores están centrados en las siguientes categorías [[Roger S. Pressman, 2010](#)]:

- Errores en la interfaz.
- Funciones que son incorrectas.
- Errores en estructuras de datos a bases de datos externas.
- Errores de comportamiento o desempeño.
- Errores de inicialización.

Técnica de partición de equivalencia

La técnica de partición de equivalencia es considerada una de las técnicas más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software, esta técnica se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar. Una clase de equivalencia representa un conjunto de estados válidos o no válidos para condiciones de entrada. Por lo general, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica [[Roger S. Pressman, 2010](#)]. Por tal razón, se utilizó la técnica de partición de equivalencia para la confección del los caso de prueba, ya que permite examinar los valores válidos e inválidos de las entradas existentes en el software.

Los casos de prueba son un conjunto de entradas de pruebas, condiciones de ejecuciones y resultados esperados desarrollados para cumplir un objetivo en particular o una función esperada. Para la elaboración de los casos de pruebas es necesario un número de datos que ayuden a la ejecución de los mismos y que permitan que el sistema se ejecute en todas sus variantes.

La definición de las variables para los casos de prueba se define en la siguiente tabla:

Tabla 3.1: Variables para el caso de prueba: Gestionar elemento de la lista de datos de miembros.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Miembro	campo de selección	No	Permite seleccionar un usuario de la lista de usuarios del sistema.
2	Roles	lista de selección múltiple	No	Permite seleccionar varios roles, de la lista.

A continuación se muestra el caso de prueba para el CU "Gestionar elemento de la lista de datos de miembros".

Tabla 3.2: Caso de prueba: CU Gestionar elemento de la lista de datos de miembros.

Escenario	Descripción	Variables		Respuesta del sistema	Flujo central
		1	2		
Sección Adicionar.					
EC 1.1. Adicionar elemento a la lista de datos de miembros	Adicionar elemento introduciendo los datos correctos	V amguerra	V Asistente de control	Adiciona el elemento correctamente	1. Seleccionar nuevo elemento. 2. Selecciona los datos 3. Selecciona la opción Aceptar
EC 1.2. Adicionar elemento a la lista de datos de miembros	Adicionar elemento introduciendo los datos vacíos.	V dtroya	V Arquitecto de información	Muestra un mensaje indicando que existen campos vacíos.	1. Seleccionar nuevo elemento. 2. Selecciona los datos 3. Selecciona la opción Aceptar

Sección Modificar.					
EC 2.1. Modificar elemento de la lista de datos de miembros	Modificar elemento introduciendo los datos correctos	V	V	Adiciona el elemento modificado	1. Seleccionar el elemento. 2. Selecciona los datos a modificar 3. Selecciona la opción Aceptar
		wgarcia	Analista		
EC 2.2. Modificar elemento de la lista de datos de miembros	Modificar elemento introduciendo los datos vacíos.	I	V	Muestra un mensaje indicando que existen campo vacíos.	1. Seleccionar el elemento. 2. Selecciona los datos a modificar 3. Selecciona la opción Aceptar
			Arquitecto de información		
		V	I		
		dtroya			
		I	I		
Sección Eliminar					
EC 3.1. Eliminar elemento de la lista de datos de miembros	Eliminar elemento.	V	V	Muestra un mensaje indicando que si desea eliminar el elemento.	1. Seleccionar el elemento. 2. Selecciona eliminar 3. Selecciona la opción Aceptar para confirmar.
		dtroya	Arquitecto de información		

Resultados de las pruebas.

A continuación se muestra un gráfico donde se ven reflejados los resultados de las pruebas al módulo.

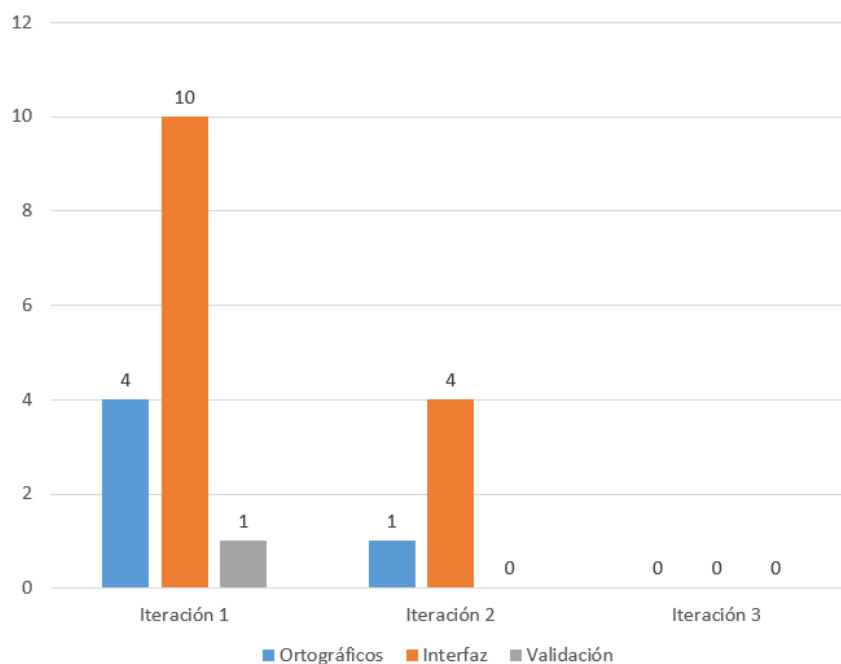


Figura 3.5: Resultado de las pruebas.

Durante la ejecución de estas pruebas se realizaron tres iteraciones, donde los errores detectados fueron solucionados conllevando a que en la 2da iteración no se detectaron errores. Entre los errores detectados se encuentran: errores ortográficos, de validación y de interfaz. Las pruebas se realizaron de forma iterativa e incremental, comprobando en cada iteración que hubiesen sido corregidos los errores detectados en la iteración anterior, lo que contribuyó a mejorar la calidad y funcionalidad del software.

3.3.2. Método de caja blanca

El método de caja blanca del software se basa en el examen cercano de los detalles de procedimiento. Las rutas lógicas a través del software y las colaboraciones entre componentes se ponen a prueba al revisar conjuntos específicos de condiciones y/o bucles. Este método se realiza sobre las funcionalidades internas de un módulo y se encargan de comprobar los caminos lógicos, ciclos (bucles) y condiciones que debe cumplir el programa. Con el empleo de este método es posible desarrollar casos de prueba que garanticen la ejecución, al menos una vez, de los caminos independientes (camino que recorre por lo menos una nueva arista en el grafo de flujo). [Roger S. Pressman, 2010]

Este método se aplicó teniendo en cuenta la técnica de ruta básica. A continuación, se describe los resultados de esta técnica.

Técnica de ruta básica

En la siguiente figura se muestra el método `getInformationFromPMDDataList(datalist)` utilizado en el caso de uso Gestionar elemento de la lista de datos de miembros. El objetivo de este método es obtener la información de los elementos de la lista de datos de miembros. Se selecciona este método teniendo en cuenta la importancia que representa para el resultado de este trabajo.

```
function getInformationFromPMDDataList(datalist) {  
  var node, result = {},           1  
      rol;  
  for (var i in datalist.children) { 2  
    node = datalist.children[i];     3  
    var roles = node.properties['xfpm-dl:projectRoles']; 3  
    if (node.assocs['xfpm-dl:member'] != null) 4  
      var member = node.assocs['xfpm-dl:member'][0]; 5  
    for (var r in roles) {           6  
      rol = roles[r];               7  
      if (result[rol]) {           8  
        result[rol].push(member.nodeRef); 9  
      } else {                     10  
        result[rol] = [member.nodeRef];  
      }  
    }  
  }  
  return result;                   11  
}
```

Figura 3.6: Técnica de ruta básica en el método `getInformationFromPMDDataList(datalist)`.

Posteriormente se procede a la elaboración del grafo de flujo teniendo en cuenta dicha enumeración:

La complejidad ciclomática es la métrica de software con que se define la cantidad de caminos independientes de cada una de las funcionalidades del programa y provee el límite superior para el número de

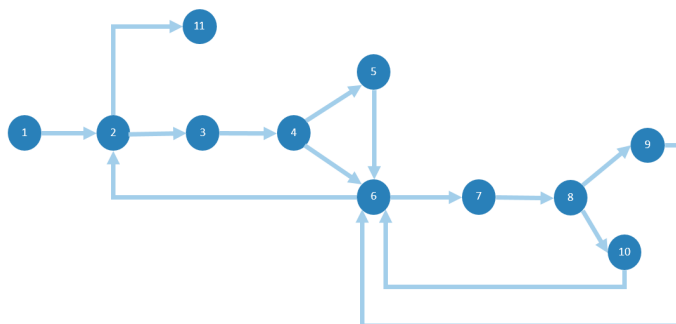


Figura 3.7: Representación del grafo de flujo de camino básico

pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez [Roger S. Pressman, 2010].

La complejidad ciclomática se basa en la teoría gráfica y se calcula de dos maneras distintas, donde, para que el cálculo sea correcto, todas deben arrojar el mismo resultado:

El número de regiones corresponde a la complejidad ciclomática "V (G)".

$V(G) = R$ Donde R es la cantidad total de regiones.

$V(G) = 5$

$V(G) = E - N + 2$ Donde E es el número de aristas y N número de nodos.

$V(G) = 14 - 11 + 2$

$V(G) = 5$

$V(G) = P + 1$ Donde P es la cantidad de nodos predicados

$V(G) = 4 + 1 = 5$

El valor V (G) expresa la cantidad de caminos linealmente independientes de la estructura de control del programa, por lo que se definen los siguientes 5 caminos:

Ruta independiente 1: 1-2-11

Ruta independiente 2: 1-2-3-4-6-2-11

Ruta independiente 3: 1-2-3-4-5-6-2-11

Ruta independiente 4: 1-2-3-4-6-7-8-9-6-2-11

Ruta independiente 5: 1-2-3-4-6-7-8-10-6-2-11

Cada camino independiente es un caso de prueba a realizar, de forma que se garantiza que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa. En el caso anterior se calcularon tres caminos básicos, por tanto, surge la necesidad de hacer igual número de casos de prueba. A continuación, se muestra uno de los casos de pruebas realizados:

Tabla 3.3: Caso de prueba: ruta independiente # 4

Descripción	Este camino verifica la existencia de elementos en la lista pasada por parámetro para luego devolver un arreglo con la información de los elementos contenidos en la lista.
Entrada	Lista de datos de miembros
Resultados Esperados	Devuelve un arreglo con la información de los elementos contenidos en la lista.
Condiciones	Cada elemento de la lista tenga asociado un miembro.

Luego de realizar las pruebas al módulo empleando la técnica de ruta básica al método `getInformationFromPMDDataList(datalist)`, se afirma que se obtuvo el resultado esperado, por lo que esta prueba concluye de manera satisfactoria.

3.4. Conclusiones del capítulo

En este capítulo fue representado el módulo a partir de sus componentes de implementación. Se mostró la distribución física en el que será implantado el mismo para su explotación. La definición de los estándares de codificación permitió mejorar el entendimiento del código perteneciente al módulo y alcanzar una uniformidad en el mismo. Finalmente tras un análisis de los resultados obtenidos en las pruebas, se demostró la solidez de la solución, evidenciado en las 3 iteraciones de pruebas realizadas, las cuales denotaron un número decreciente de no conformidades encontradas.

Conclusiones Generales

Como resultado de la investigación realizada y los resultados obtenidos, se concluye:

- El establecimiento de los fundamentos teóricos y metodológicos relevantes de la gestión de expediente de proyecto, permitió sentar las bases del proceso investigativo y conocer la lógica del negocio.
- El análisis de la gestión de expediente de proyecto para la Red de Centros de la UCI en el GDA Xabal eXcriba 3.0, permitió identificar las limitantes que obstaculizan el proceso de gestión.
- El empleo de la metodología AUP-UCI en su escenario 2, posibilitó obtener la documentación necesaria para llevar a cabo las actividades de implementación.
- El desarrollo del módulo permite la gestión de expedientes de proyecto de la Red de Centros de la UCI en el Gestor de Documentos Administrativos Xabal-eXcriba 3.1, cumpliendo con los requisitos funcionales y no funcionales definidos
- Se realizaron las pruebas y validaciones necesarias para garantizar la calidad del sistema y la satisfacción de los clientes, obteniendo resultados satisfactorios en cada una, ya que todas las no conformidades encontradas fueron corregidas.

Recomendaciones

Se recomienda para el futuro:

- Configurar el buscador del GDA eXcriba para realizar búsquedas sobre expedientes creados.
- Implementar una funcionalidad que permita descargar un volumen o el expediente completo.
- Permitir realizar el trámite de los expedientes el sistema GESPRO .
- Incorporar una funcionalidad que permita gestionar la línea base de los expedientes de proyecto.

Glosario de términos

API Interfaz de Programación de Aplicaciones (Application Programming Interface).

AUP Proceso Agil Unificado(Agile Unified Process).

CASE Ingeniería de Software Asistida por Ordenador (Computer Aided Software Engineering).

CMMI-DEV Integración de modelos de madurez de capacidades-Desarrollo (Capability Maturity Model

ECM Gestión de Contenidos Empresariales (Enterprise Content Management). el flujo de documentos dentro de una organización.

HTTP Es un protocolo de transferencia orientado a transacciones y se le identifica mediante un URL.

REST método para obtener el contenido de información de un sitio web mediante la lectura de una página

REST Transferencia de Estado Representacional (Representational State Transfer).

RUP Rational Unified Process (Proceso Unificado de Desarrollo).

SGD Sistema de Gestión Documental. Sistemas informáticos creados para almacenar, administrar y controlar documentos.

TIC Tecnologías de la información y la comunicación. Son todos aquellos recursos, herramientas y progra-

UML Lenguaje Unificado de Modelado (Unified Modeling Language). Es el lenguaje de modelado de siste-

Web Que contiene designado un XML que describe e incluye el contenido deseado.

XML Lenguaje de Marcado Extensible

Referencias bibliográficas

- AIIM (2016). What is Enterprise Content Management? <https://www.aiim.org/What-is-ECM-Enterprise-Content-Management.aspx#> <http://www.aiim.org/What-is-ECM-Enterprise-Content-Management>.
- Alfresco Software, I. (2015a). Alfresco Community. <http://www.alfresco.com/products/community>.
- Alfresco Software, I. (2015b). Web Scripts. <https://docs.alfresco.com/4.1/concepts/ws-architecture.html>.
- Alfresco Software, I. (2016a). Actions and Responsibility. <https://community.alfresco.com/docs/DOC-4767-actions-and-rules>.
- Alfresco Software, I. (2016b). Content modeling. <https://docs.alfresco.com/4.0/concepts/content-modeling-about.html>.
- Alfresco Software, I. (2016c). JavaScript API Samples.
- Archivo General de la Nación (2015). Archivo General de la Nación. <http://www.archivogeneral.gov.co/>.
- Bhaumik, S. (2011). Alfresco 3 Cookbook. <http://snigbhaumik.com/books/alfresco-3-cookbook/>.
- Booch, Jacobson, R. (2006). El lenguaje Unificado de Modelado. *B. Pearson Educ.*, Segunda Ed(Person Education):1–551.
- Community, A. (2015). Document Management Software Benubird.
- Cortéz Alonzo, V. (1989). *Manual de archivos municipales*.
- Cuba (2003). Gaceta Oficial de la Republica de Cuba. <http://www.gacetaoficial.cu/>.
- Eguiluz, J. (2015). Introducción a JavaScript. http://librosweb.es/libro/javascript/capitulo_1.html.
- EXES (2016). Manual de XML. <http://www.mundolinux.info/index.htm>.
- Gallo, P. R. (2011). Gestión documental en las organizaciones. *Univ. OBERTA CATALUNYA*, page 103.

- Garzás, J. (2013). Gestión ágil de Proyectos de Software. page 226.
- Heredia Herrera, A. (1991). *Archivística General: teoría y práctica*.
- International Organization for Standardization (2006). ISO 15489-1. Información y documentación. Gestión de documentos : Parte 1: Generalidades. *Iso*, 2001:37.
- Jacobson, I., Bittner, K., y Spence, I. (2013). Casos de uso 2.0. La guía para ser exitoso con los casos de uso. https://www.ivarjacobson.com/sites/default/files/field_iji_file/article/use_case_2.0_-_spanish_translation.pdf.
- JSON.org (2014). Introducing JSON. <http://www.json.org/>.
- Lamas, T. A. (2007). Universidad de las Ciencias Informáticas.
- Larman, C. (2003). UML y patrones. pages 1–265.
- Leovigilda Hernandez (2013). MODELO DE IMPLEMENTACIÓN. <http://ithleovi.blogspot.com/2013/06/unidad-5-modelo-deimplementacion-el.html>.
- Martinez, R. (2016). PostgreSQL-ES | Emc2Net. <https://e-mc2.net/es/postgresql-es>.
- Mastermagazine (2010). Definición de herramienta. www.mastermagazine.info/termino/5234.php.
- McDaniel, G. y International Business Machines Corporation (2015). IBM Dictionary of Computing. <https://www-03.ibm.com/ibm/history/documents/pdf/glossary.pdf>.
- Microsoft (2016). Visual Studio Code. <https://code.visualstudio.com/docs#vscode>
<https://code.visualstudio.com/docs> <http://code.visualstudio.com>.
- Microsoft (2013). Microsoft Developer Network. <http://msdn.microsoft.com>.
- Montalvo, M. (2015). XML. El nuevo lenguaje universal 1. —.
- Navarro, A. L. M. (2008). El e-expediente administrativo. http://www.interpares.org/display_file.cfm?doc=ip1-2-3_mexico_dissemination_janr_navarro_rgpd_2_2008.pdf.
- Paradigm, V. (2016). Visual Paradigm - UML, Agile, PMBOK, TOGAF, BPMN and More! <https://www.visual-paradigm.com/features/>.
- POSSO, A. N. R. (2010). Repositorio digital de la Universidad Técnica del Norte. <http://repositorio.utn.edu.ec/>.

- Quesada, J. A. L. (2014). Lenguajes de marcas y sistemas de gestión de información. Universidad de Murcia. http://dis.um.es/~lopezquesada/documentos/IES_1415/LMSGI/curso/material.html#ut5.
- RAE (2017). Diccionario de la lengua española - Edición del Tricentenario. *Real Acad. Española*.
- Rodríguez, T. (2014). Metodología de desarrollo para la Actividad productiva de la UCI. pages 1–16.
- Roger S. Pressman (2010). *Ingeniería de software - Un Enfoque práctico*.
- Signer, B. (2010). What Is Wrong with Digital Documents ? A Conceptual Model for Structural Cross-Media. In *International Conference on Conceptual Modeling*, pages 391–404. Springer.
- Sommerville, I. (2011). *Software Engineering*.
- Sparx (2013). Sparx Systems Pty Ltd. <http://www.sparxsystems.com.au/>.
- UCI (2015). Manual de Usuario de Excriba.
- Whitepaper, N. (2015). Role based access control for world wide web.pdf.
- Zapata Cardenas, C. A. (2005). Directrices Para Estructurar Un Programa De Gestion De Documentos En Las Organizaciones. *Rev. CODICE*, 1:97–111.

Bibliografía

- I. Sommerville, Software Engineering. 2010.
- C. Álvarez de Zayas, A Carlos; SIERRA, V. La investigación científica en la sociedad del conocimiento. Material de apoyo a la docencia. La Habana, 2002.
- C. E. O. Morales, “Análisis, diseño, desarrollo, pruebas y despliegue de software con los estándares de calidad, proceso y tecnologías usadas en Pragma S.A.,” 2012.
- C. Larman, “UML y patrones,” pp. 1–265, 2003.
- D. Crockford, JavaScript: The Good Parts. 2008.
- D. Flanagan, JavaScript: The Definitive Guide 6th Edition. 2011.
- D. Goodman, JavaScript Bible. 2010.
- D. Weisinger, Alfresco 3 Records Management. 2011.
- E. Sierra Valentí, El expediente administrativo. Esbozo de tipología documental, vol. 29/2. 1979.
- G. Documental, “Alfresco 1,” Universidad de Granada, 2010.
- I. M. G. Jorin, “Proceso de pruebas para la liberación de productos software,” 2007.
- I. Sommerville, “Ingeniería del software,” 2005.
- J. Bradenbaugh, Aplicaciones Java Script. 2000.
- J. Pots, “Alfresco Developers Guide,” American journal of epidemiology, 2010.
- J. Rumbaugh, I. Jacobson, and G. Booch, El Lenguaje Unificado de Modelado. 1999.
- L. Guzmán, “Guzmán, de Sistemas Archivo - 1999 - El documento electrónico.pdf.”

-
- M. Bergljung, Alfresco 3 Business Solutions. 2011.
 - M. Haverbeke, Eloquent JavaScript. 2007.
 - M. Haverbeke, Eloquent JavaScript: A Modern Introduction to Programming. 2011.
 - M. Kimberly, G. Deepa, E. Board, E. Principal, I. Primary, F. Systems, E. B. Study, and N. Co-
 - M. Shariff, P. Majmudar, and A. Bhandari, Alfresco 3 Web Content Management. 2010.
 - Pérez, JavaScript. 2009.
 - R. A. Castro Gil, Estructura básica del proceso unificado de desarrollo de software. 2004.
 - R. Alfredo, H. León, and S. C. González, El Paradigma Cuantitativo de la investigación científica. 2002.
 - R. Pressman, Ingeniería Del Software I. 2010.
 - SEI, CMMI R ? para Desarrollo, Versión 1.3. 2010. investigator, Arquitectura Macro de N-Capas. 2014.

Anexo A

Anexos

A.1. Diccionario de datos de la entidad expediente de proyecto.

Descripción		<i>Esta entidad representa un expediente de proyecto en el sistema.</i>				
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
Nombre	Nombre propio del expediente de proyecto en el Gestor de Documentos Administrativos Xabal eXcriba 3.1.	Cadena de caracteres.	No	No	A-Z, a-z, 0-9, algunos caracteres especiales	“”, *, \, /, <, >, ?, :, , \$
Tipo	Determina el tipo de expediente de proyecto.	Lista de selección única.	No	No	*Listado de tipos.	
Versión	Determina la versión del expediente de proyecto.	Lista de selección única.	No	No	3.4 4.0 5.0	

Tabla A.1: Diccionario de datos de la entidad expediente de proyecto.

A.2. Diccionario de datos de la entidad lista de datos de miembros.

Tabla A.2: Diccionario de datos de la entidad lista de datos de miembros.

Descripción		<i>Esta entidad representa la lista de datos de artefactos en el sistema.</i>				
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
Nombre	Nombre propio la lista de datos de artefactos.	Cadena de caracteres.	No	No	A-Z, a-z, 0-9, algunos caracteres especiales	“”, *, \, /, <, >, ?, ;, , \$
Descripción	Descripción de la lista de datos de artefactos.	Cadena de caracteres.	No	No	A-Z, a-z, 0-9, algunos caracteres especiales	

A.3. Diccionario de datos de la entidad lista de datos de artefactos.

Tabla A.3: Diccionario de datos de la entidad lista de datos de artefactos.

Descripción		<i>Esta entidad representa la lista de datos de artefactos en el sistema.</i>				
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
Nombre	Nombre propio de la lista de miembros de artefactos.	Cadena de caracteres.	No	No	A-Z, a-z, 0-9, algunos caracteres especiales	“, *, \, /, <, >, ?, ;, , \$
Descripción	Descripción de la lista de miembros de artefactos.	Cadena de caracteres.	No	No	A-Z, a-z, 0-9, algunos caracteres especiales	
Tipo	Contiene el lista de tipos de expedientes definidos.	Campo de selección única.	No	No	*Listado de tipos.	
Versión	Contiene el listado de versiones de los expedientes de proyecto	Campo de selección única.	No	No	<ul style="list-style-type: none"> • 3.4 • 4.0 • 5.0 	

A.4. Diccionario de datos de la entidad administrador.

Tabla A.4: Diccionario de datos de la entidad administrador.

Descripción	<i>Esta entidad representa a un administrador en el sistema.</i>					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
N/A	N/A	N/A	N/A	N/A	N/A	N/A

A.5. Diccionario de datos de la entidad especialista de calidad.

Tabla A.5: Diccionario de datos de la entidad especialista de calidad.

Descripción	<i>Esta entidad representa a un especialista de calidad en el sistema.</i>					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
N/A	N/A	N/A	N/A	N/A	N/A	N/A

A.6. Diccionario de datos de la entidad asesor de planificación.

Tabla A.6: Diccionario de datos de la entidad asesor de planificación.

Descripción	<i>Esta entidad representa a un asesor de planificación en el sistema.</i>					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
N/A	N/A	N/A	N/A	N/A	N/A	N/A

*Listado de tipos:

1. Expediente de desarrollo
2. Expediente de desarrollo-4.0
3. Expediente de desarrollo para portales
4. Expediente de proyecto de servicio
5. Expediente de proyecto de servicio para las personalizaciones
6. Expediente de componente
7. Expediente de migración
8. Expediente de almacenes

A.7. Modelo conceptual a requisito.

(8) Class		Administrador	Área	Asesor de planificación	Especialista de calidad	Expediente de proyecto	Lista de datos de artefactos	Lista de datos de miembros	Usuario
By: <input type="text" value="Transitor"/>	(25) Requirement								
<input type="checkbox"/> Crear área.		✓	✓						
<input type="checkbox"/> Modificar área.		✓	✓						
<input type="checkbox"/> Eliminar área.		✓	✓						
<input type="checkbox"/> Crear lista de datos de artefactos.					✓		✓		
<input type="checkbox"/> Modificar lista de datos de artefactos.					✓		✓		
<input type="checkbox"/> Eliminar lista de datos de artefactos.					✓		✓		
<input type="checkbox"/> Adicionar elemento en la lista de datos de artefactos.					✓		✓		
<input type="checkbox"/> Modificar elemento en la lista de datos de artefactos.					✓		✓		
<input type="checkbox"/> Eliminar elemento en la lista de datos de artefactos.					✓		✓		
<input type="checkbox"/> Crear lista de datos de miembros.				✓				✓	
<input type="checkbox"/> Modificar lista de datos de miembros.				✓				✓	
<input type="checkbox"/> Eliminar lista de datos de miembros.				✓				✓	
<input type="checkbox"/> Adicionar elemento en la lista de datos de miembros.				✓				✓	
<input type="checkbox"/> Modificar elemento en la lista de datos de miembros.				✓				✓	
<input type="checkbox"/> Eliminar elemento en la lista de datos de miembros.				✓				✓	
<input type="checkbox"/> Crear plantilla del expediente de proyecto.					✓				
<input type="checkbox"/> Eliminar plantilla del expediente de proyecto.					✓				
<input type="checkbox"/> Crear el expediente de proyecto.				✓		✓			
<input type="checkbox"/> Modificar el expediente de proyecto.				✓		✓			
<input type="checkbox"/> Establecer los permisos en el expediente de proyecto.				✓					
<input type="checkbox"/> Modificar los permisos en el expediente de proyecto.				✓					
<input type="checkbox"/> Eliminar los permisos en el expediente de proyecto.				✓					
<input type="checkbox"/> Crear documento a partir de plantilla		✓		✓	✓				✓
<input type="checkbox"/> Modificar documento creado a partir de plantilla		✓		✓	✓				✓
<input type="checkbox"/> Eliminar documento creado a partir de plantilla.		✓		✓	✓				✓

Figura A.1: Matriz de modelo conceptual a requisito.

A.8. Requisito a requisito.

By: ▼

(25) Requirement

(25) Requirement	Crear área.	Modificar área.	Eliminar área.	Crear lista de datos de artefactos.	Modificar lista de datos de artefactos.	Eliminar lista de datos de artefactos.	Adicionar elemento en la lista de datos de artefactos.	Modificar elemento en la lista de datos de artefactos.	Eliminar elemento en la lista de datos de artefactos.	Crear lista de datos de miembros.	Modificar lista de datos de miembros.	Eliminar lista de datos de miembros.	Adicionar elemento en la lista de datos de miembros.	Modificar elemento en la lista de datos de miembros.	Eliminar elemento en la lista de datos de miembros.	Crear plantilla del expediente de proyecto.	Eliminar plantilla del expediente de proyecto.	Crear el expediente de proyecto.	Modificar el expediente de proyecto.	Establecer los permisos en el expediente de proyecto.	Modificar los permisos en el expediente de proyecto.	Eliminar los permisos en el expediente de proyecto.	Crear documento a partir de plantilla	Modificar documento creado a partir de plantilla	Eliminar documento creado a partir de plantilla.		
Crear área.	⌕																										
Modificar área.		⌕																									
Eliminar área.			⌕																								
Crear lista de datos de artefactos.				⌕																							
Modificar lista de datos de artefactos.					⌕																						
Eliminar lista de datos de artefactos.						⌕																					
Adicionar elemento en la lista de datos de artefactos.							⌕																				
Modificar elemento en la lista de datos de artefactos.								⌕																			
Eliminar elemento en la lista de datos de artefactos.									⌕																		
Crear lista de datos de miembros.										⌕																	
Modificar lista de datos de miembros.											⌕																
Eliminar lista de datos de miembros.												⌕															
Adicionar elemento en la lista de datos de miembros.													⌕														
Modificar elemento en la lista de datos de miembros.														⌕													
Eliminar elemento en la lista de datos de miembros.															⌕												
Crear plantilla del expediente de proyecto.																⌕											
Eliminar plantilla del expediente de proyecto.																	⌕										
Crear el expediente de proyecto.																		⌕									
Modificar el expediente de proyecto.																			⌕								
Establecer los permisos en el expediente de proyecto.																				⌕							
Modificar los permisos en el expediente de proyecto.																					⌕						
Eliminar los permisos en el expediente de proyecto.																						⌕					
Crear documento a partir de plantilla																								⌕			
Modificar documento creado a partir de plantilla																									⌕		
Eliminar documento creado a partir de plantilla.																										⌕	

Figura A.2: Matriz de requisito a requisito.

A.9. Caso de uso a caso de prueba.

By:

(8) Test Case

(8) Use Case

	CP_Gestionar Lista de Datos de Artefactos	CP_Gestionar documento a partir de plantilla	CP_Gestionar elemento de la lista de datos de artefactos	CP_Gestionar elemento de la lista de datos de miembros	CP_Gestionar expediente de proyecto	CP_Gestionar lista de datos de miembros.	CP_Gestionar plantilla del expediente de proyecto.	CP_Gestionar área
<input checked="" type="radio"/> Gestionar lista de datos de artefactos	✓							
<input checked="" type="radio"/> Gestionar documento a partir de plantilla		✓						
<input checked="" type="radio"/> Gestionar elementos de la lista de datos de artefactos			✓					
<input checked="" type="radio"/> Gestionar elemento de la lista de datos de miembros				✓				
<input checked="" type="radio"/> Gestionar expediente de proyecto					✓			
<input checked="" type="radio"/> Gestionar lista de datos de miembros						✓		
<input checked="" type="radio"/> Gestionar plantilla del expediente de proyecto							✓	
<input checked="" type="radio"/> Gestionar área								✓

Figura A.3: Matriz de caso de uso a caso de prueba.

A.10. Gestionar área.

Caso de Uso	Gestionar área.	
Objetivo	Gestionar las áreas del sistema.	
Actores	Administrador: (Inicia) Crea, edita, y elimina un área del sistema.	
Resumen	El caso de uso inicia cuando el actor selecciona crear, modificar o eliminar un área, el sistema permite realizar la acción solicitada.	
Complejidad	Media	
Prioridad	Media	
Precondiciones	<ol style="list-style-type: none"> 1. El actor tiene los permisos necesarios y esta autenticado en el sistema. 2. El actor se encuentra en el entorno Área. 	
Post-condiciones	<ol style="list-style-type: none"> 1. Se actualizo un área. 	
Flujo de eventos		
Flujo básico: Gestionar área.		
	Actor	Sistema
1.	Crea, modifica o elimina un área.	
2.		Permite realizar varias acciones: <ul style="list-style-type: none"> - Crear un área. Ver Sección 1: Crear un área. - Modificar área. Ver Sección 2: Modificar área. - Eliminar un área. Ver Sección 3: Eliminar área.
3.		Termina el caso de uso.
Sección 1: Crear un área.		
Flujo básico		
	Actor	Sistema
1.	Selecciona el botón "Nuevo área".	
2.		Muestra un formulario con los siguientes datos: <ul style="list-style-type: none"> - Nombre* - Descripción*
3.	Insertar valores.	
4.	Presiona el botón "Aceptar".	
5.		Valida la información insertada.
6.		Crea el nuevo área con los datos insertados.
7.		Termina el caso de uso.
Flujos alternos		
4.a Presionar el botón "Cancelar".		
	Actor	Sistema
1.	Presiona el botón "Cancelar".	

Continúa en la siguiente página.

2.		Cancela la acción y muestra la interfaz anterior.
3.		Termina el caso de uso.
5.a Existen campos obligatorios vacíos.		
	Actor	Sistema
1		Muestra encima del campo vacío el mensaje: "El valor no puede estar vacío."
2		Continuar en el paso 3 del flujo básico.
Sección 2: Modificar área.		
Flujo básico		
	Actor	Sistema
1.	Presiona el ícono "Editar" del área escogida.	
2.		Muestra un formulario con los siguientes datos: - Nombre* -Descripción*
3.	Insertar valores.	
4.	Presiona el botón "Aceptar".	
5.		Valida la información insertada.
6.		Edita el área con los datos insertados.
7.		Termina el caso de uso.
Flujos alternos		
4.a Presionar el botón "Cancelar".		
	Actor	Sistema
1.	Presiona el botón "Cancelar".	
2.		Cancela la acción y muestra la interfaz anterior.
3.		Termina el caso de uso.
5.a Existen campos obligatorios vacíos.		
	Actor	Sistema
1.		Muestra encima del campo vacío el mensaje: "El valor no puede estar vacío."
2.		Continuar en el paso 3 del flujo básico.
Sección 4: "Eliminar área."		
Flujo básico		
	Actor	Sistema
1.	Presiona el ícono "Eliminar" del área escogida.	
2.		El sistema muestra la interfaz de confirmación de eliminación.
3.	Selecciona el botón "Eliminar".	

Continúa en la siguiente página.

4.		El sistema elimina el área escogida.
5.		Se actualiza la lista.
6.		Termina caso de uso.
Flujos alternos		
3.a Presionar el botón "Cancelar".		
	Actor	Sistema
1	Presiona el botón "Cancelar".	
2		Cerrar la interfaz de confirmación de eliminación.

A.11. Gestionar documento a partir de plantilla.

Tabla A.7: Descripción del CU_2: Gestionar documento a partir de plantilla.

Caso de Uso	Gestionar documento a partir de plantilla	
Objetivo	Gestionar documento a partir de plantilla	
Actores	Usuario: (Inicia) Crea, edita, y elimina un documento a partir de plantilla	
Resumen	El caso de uso inicia cuando el actor selecciona crear, modificar o eliminar un documento a partir de plantilla, el sistema permite realizar la acción solicitada.	
Complejidad	Media	
Prioridad	Media	
Precondiciones	<ol style="list-style-type: none"> 1. El actor tiene los permisos necesarios y esta autenticado en el sistema. 2. El actor se encuentra en una carpeta del expediente de proyecto. 	
Post-condiciones	<ol style="list-style-type: none"> 1. Se actualizo un expediente de proyecto. 	
Flujo de eventos	Flujo básico: Gestionar documento a partir de plantilla.	
	Actor	Sistema
1.	Crea, modifica o elimina de un documento a partir de plantilla.	
2.		Permite realizar varias acciones: <ul style="list-style-type: none"> - Crear un documento. Ver Sección 1: Crear un documento. - Modificar un documento. Ver Sección 2: Modificar un documento. - Eliminar un documento. Ver Sección 3: Eliminar un documento.
3.		Termina el caso de uso.
Sección 1: Crear un documento.		
	Actor	Sistema

Continúa en la siguiente página.

1.	Selecciona el botón "Crear".	Muestra una lista desplegable con varias opciones.
2.	Selecciona "Crear documento a partir de plantilla".	Muestra una lista desplegable con los documentos del expediente de proyecto.
3.	Selecciona un documento de la lista.	Muestra un mensaje: Contenido creado a partir de plantilla
4.		Termina el caso de uso.
Flujos alternos		
1,2 .a Presionar el botón "Cancelar".		
	Actor	Sistema
1.	Presiona el botón "Cancelar".	
2.		Cancela la acción y muestra la interfaz anterior.
3.		Termina el caso de uso.
Sección 2: Modificar documento.		
Flujo básico		
	Actor	Sistema
1.	Presiona el ícono "Editar".	
2.		Muestra un formulario con los siguientes datos: - Nombre* -Titulo -Descripción
3.	Insertar valores.	
4.	Presiona el botón "Aceptar".	
5.		Valida la información insertada.
6.		Edita el documento con los datos insertados.
7.		Termina el caso de uso.
Flujos alternos		
4.a Presionar el botón "Cancelar".		
	Actor	Sistema
1.	Presiona el botón "Cancelar".	

Continúa en la siguiente página.

2.		Cancela la acción y muestra la interfaz anterior.
3.		Termina el caso de uso.
5.a Existen campos obligatorios vacíos.		
	Actor	Sistema
1.		Muestra encima del campo vacío el mensaje: "El valor no puede estar vacío."
2.		Continuar en el paso 3 del flujo básico.
Sección 4: "Eliminar documento."		
Flujo básico		
	Actor	Sistema
1.	Presiona el ícono "Eliminar".	
2.		El sistema muestra la interfaz de confirmación de eliminación.
3.	Selecciona el botón "Eliminar".	
4.		El sistema elimina el área escogida.
5.		Se actualiza la lista.
6.		Termina caso de uso.
Flujos alternos		
3.a Presionar el botón "Cancelar".		
	Actor	Sistema
1	Presiona el botón "Cancelar".	
2		Cerrar la interfaz de confirmación de eliminación.

A.12. Gestionar expediente de proyecto.

Tabla A.8: Descripción del CU_6: Gestionar expediente de proyecto.

Caso de Uso	Gestionar expediente de proyecto	
Objetivo	Gestionar expediente de proyecto	
Actores	Asesor de planificación: (Inicia) Crea o modifica un expediente de proyecto.	
Resumen	El caso de uso inicia cuando el actor desea crear o modificar una un expediente de proyecto, el sistema permite realizar la acción solicitada.	
Complejidad	Media	
Prioridad	Alta	
Precondiciones	<ol style="list-style-type: none"> 1. El actor tiene los permisos necesarios y esta autenticado en el sistema. 2. El actor se encuentra en el Área. 3. El actor se encuentra en la carpeta del sistema destinada para los expediente de proyecto. 	
Post-condiciones	<ol style="list-style-type: none"> 1. Se gestionó el expediente de proyecto. 	
Flujo de eventos		
Flujo básico: Gestionar expediente de proyecto.		
	Actor	Sistema
1.	Crea o modifica un expediente de proyecto	
2.		Permite realizar varias acciones: <ul style="list-style-type: none"> - Crear un expediente de proyecto - Ver Sección 1: Crear expediente de proyecto. - Modifica un expediente de proyecto Ver Sección 3: Eliminar expediente de proyecto
3.		Termina el caso de uso.
Sección 1: Crear expediente de proyecto.		
Flujo básico		
	Actor	Sistema
1.	Selecciona el botón "Crear".	
2.		Muestra un formulario con los siguientes datos:

Continúa en la siguiente página.

		- Nombre* -Tipo* -Versión* -Lista de Datos de Miembros*.
3.	Insertar valores.	
4.	Presiona el botón "Aceptar".	
5.		Valida la información insertada.
6.		Crea el expediente de proyecto.
7.		Termina el caso de uso.
Flujos alternos		
1.a Presionar el botón "Cancelar".		
	Actor	Sistema
1.	Presiona el botón "Cancelar".	
2.		Cancela la acción y muestra la interfaz anterior.
3.		Termina el caso de uso.
5.a Existen campos obligatorios vacíos.		
	Actor	Sistema
1.		Muestra encima del campo vacío el mensaje: "El valor no puede estar vacío."
2.		Continuar en el paso 3 del flujo básico.
Sección 2: Modificar expediente de proyecto.		
Flujo básico		
	Actor	Sistema
1.	Presiona el ícono "Editar".	
2.		Muestra un formulario con los siguientes datos: - Nombre* - Lista de Datos de Miembros*.
3.	Insertar valores.	
4.	Presiona el botón "Aceptar".	

Continúa en la siguiente página.

5.		Valida la información insertada.
6.		Edita el expediente de proyecto
7.		Termina el caso de uso.
Flujos alternos		
4.a Presionar el botón "Cancelar".		
	Actor	Sistema
1.	Presiona el botón "Cancelar".	
2.		Cancela la acción y muestra la interfaz anterior.
3.		Termina el caso de uso.
5.a Existen campos obligatorios vacíos.		
	Actor	Sistema
1.		Muestra encima del campo vacío el mensaje: "El valor no puede estar vacío."
2.		Continuar en el paso 3 del flujo básico.