

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS  
Facultad 3



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

Análisis y Diseño del Sistema para la Gestión de Documentos Digitales  
del grupo de Arquitectura y Estándares de Información de la Dirección  
Técnica de la Infraestructura Productiva.

**Autores:** Ivonne Leyva García  
Idaliana Legrá Puig

**Tutor:** Ing. Yanicet Avelaira Rodríguez  
**Co-Tutor:** Lic. Keyttia Pintón Almenares

Ciudad de la Habana  
Enero de 2009



*“La inteligencia consiste no sólo en el conocimiento, sino también en la destreza de aplicar los conocimientos en la práctica.”*  
*Aristóteles*

## *Declaración de Autoría*

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Ivonne Leyva García

---

Autor

Idaliana Legrá Puig

---

Autor

Ing. Yanicet Avelaira Rodríguez

---

Tutor

## *Datos de Contacto*

**Tutor:**

Yanicet Aveleira Rodríguez

Ing. en Ciencias Informáticas, graduada de la UCI, curso 2007/2008.

[yaveleira@uci.cu](mailto:yaveleira@uci.cu)

**Co-Tutor:**

Keyttia Pintón Almenares

Lic. en Bibliotecología y Ciencias de la Información de la Universidad de La Habana.

[keyttia@uci.cu](mailto:keyttia@uci.cu)

## *Dedicatoria*

### *De Ivón*

*A mami, papi y mis abuelos Cuca y Rodrigo, por darme la vida, por ser los mejores progenitores del mundo, por ser mi TODO. Los quiero mucho.*

*A mis hermanitas Ivecita y Lili, por ser lo que más quiero en este mundo.*

### *De Idaliana*

*A mis padres que son la razón por la que vivo, por apoyarme en cada momento de mi vida por darme las fuerzas para seguir adelante, por amarme tanto y darme lo mejor de ellos, gracias por todo, gracias por no exigir, por no presionar, por saber guiarme y sobre todo gracias por confiar. Los quiero mucho.*

*A mi abuelita Yoya, eres la mejor abuela que pueda tener cualquier persona, gracias por estar siempre preocupada por mí.*

*A la memoria de mi abuelita Rafaela que aunque no pueda disfrutar de este momento, sé que se sentiría muy orgullosa de su nieta, siempre te llevaré en mi corazón, que Dios te tenga en la gloria bien en alto donde tú te mereces estar*

# *Agradecimientos*

Quiero agradecer a:

*De Ivón*

*Mis padres, mis abuelos (que son mis segundos padres), porque nunca será suficiente lo que haga por ellos y para ellos. Por ser mi guía, mi apoyo, por todo su amor, comprensión, por darme las fuerzas para seguir adelante, por siempre confiar en mí. Los quiero mucho.*

*A mis hermanitas por todo su amor y cariño sinceros. Las quiero mucho mucho.*

*A Ana Ivis, mi Mejor Amiga, mi Hermana, por su apoyo incondicional, no bastarán mil vidas para agradecer por tu amistad. Te quiero mucho.*

*A Ida, mi compañera de tesis, mi amiga y también hermanita durante todo este tiempo en la universidad. Gracias por estar siempre conmigo, por no dejarme cejar en el empeño. Por a veces pelearnos, pero enseñarnos mutuamente a crecer y ser mejores mujeres, mejores profesionales. Sabes cuánto te quiero. Muchas gracias.*

*A mis amigas Yami, Tamy, Lisandra, Ariana, Sahily, Malenilla y Yadira, y a mis amigos Mario, Marcel, José Armando por convertirse en mi familia durante todo este tiempo.*

*A Liubert, un amigo que ha sido mi sombra en toda la investigación, que constantemente estaba alentándome.*

*A mis tíos y mi familia en general por brindarme tanto cariño y apoyo siempre.*

*A mis amigos de la vocacional Danibys, Dariena, Dayi, Neydita, Ililiana, Leonelito, Eduard por mantenernos unidos aunque estemos lejos.*

*A todos mis compañeros de grupo, por haber compartido conmigo estos años inolvidables de la universidad.*

# *Agradecimientos*

Quiero agradecer a:

*De Idaliana*

*A mis padres porque todo lo que soy se lo debo a ellos, atribuyo todos mis éxitos en esta vida a la enseñanza moral, intelectual y física que recibí de ellos y sobre todas las cosas porque siento de corazón que tenerlos es poseer el mayor tesoro del mundo.*

*A mi novio (mi ojote) por su comprensión y paciencia sin límites; por enseñarme que sólo porque alguien no te ama de la forma que quieres, no significa que no te ame con todo lo que puede, porque hay personas que nos aman, pero que no saben cómo demostrarlo.*

*Se dice que los amigos son los hermanos que Dios olvidó darnos por eso quiero agradecerle a Ivón por acompañarme en mis logros, fracasos, por haber compartido conmigo en estos 5 años sonrisas, lágrimas y dolor y no juzgarme por mis errores, en fin, por comportarse como una amiga y más que eso como una hermana. Gracias y mil gracias!!!*

*A Yami por el cariño y apoyo moral que siempre he recibido de ella, por sus consejos los cuales me han enseñado mucho en la vida. Gracias.*

*A mis compañeros de grupo, en especial a mis compañeras de cuarto las cuales han sido de una forma u otra mi gran familia aquí en la universidad, en la que hemos compartido momentos de tristezas, estrés, alegrías, fiestas, borracheras y griterías por causa de las ranas y bichos, a todas ustedes muchísimas gracias por haberme aguantado durante estos años: Ivón, Ariana, Lisandra, Malena y Sahily.*

*Al abuelito de mi novio y a mi suegra por su preocupación y por los consejos que me han dado los cuales han sido de gran ayuda para continuar el camino que me queda por recorrer.*

*A toda mi familia por educarme y apoyarme siempre.*

## *Agradecimientos Generales*

*Queremos agradecerles:*

*A nuestras tutoras la Lic. Keyttia Pintón Almenares y la Ing. Yanicet Aveleira, de quienes nació la idea original de este tema de tesis, por permitirnos con su apoyo constante la materialización de esa idea en este trabajo de diploma.*

*A Enrique J. González y a Fidel Jiménez, por las horas de sueño que les robamos para la realización de este trabajo.*

*A todos los amigos y profes que construyeron nuestro día a día en la escuela y contribuyeron a nuestra formación como persona y como profesional.*

*A las especialistas del grupo de A&EI por brindarnos su ayuda.*

*A todos los que de una forma u otra contribuyeron a la realización de este trabajo.*

*A Fidel y a la Revolución Cubana por darnos la oportunidad de convertirnos en profesionales.*

*A todos: Muchas Gracias*



## *Resumen*

En el presente trabajo se realiza un estudio de la situación actual de la gestión de la documentación digital que se generan por las funciones que realiza el Grupo de Arquitectura y Estándares de la Información (grupo de A&EI) de la Dirección Técnica (DT) de la Infraestructura Productiva (IP) de la Universidad de Ciencias Informáticas (UCI).

Para el desarrollo del trabajo se realiza un estudio del arte de las principales metodologías y herramientas a utilizar en el proceso de desarrollo de software, consultando diferentes bibliografías que fundamenten el por qué de su selección.

Este documento tiene como principal objetivo servir de guía a los desarrolladores de la propuesta de sistema web para la gestión de los documentos digitales del grupo de A&EI, ya que se describe detalladamente el análisis y diseño de un sistema informático que dará solución a los problemas que presenta actualmente el grupo y que influyen en la calidad de su funcionamiento.

### **Palabras claves:**

Sistemas de Gestión de Documentos Digitales, Documentos Digitales, Sistema de gestión, Sistema de gestión documental, Gestión documental, Análisis de software, Diseño de software, Ingeniería de Software.

## Índice

Dedicatoria .....	IV
Agradecimientos.....	V
Resumen .....	VIII
Introducción .....	1
Capítulo I Fundamentación Teórica.....	1
1.1 Introducción.....	1
1.2 Fundamentación del tema.....	1
1.2.1 Definiciones importantes en el proceso de Gestión documental .....	1
1.2.2 Características de los Sistemas de Gestión Documental .....	2
1.3 Tendencias y tecnologías actuales para el proceso de desarrollo de software.....	3
1.3.1 Metodologías de desarrollo de software.....	3
1.3.2 Lenguaje de Modelado .....	10
1.3.3 Herramientas CASE.....	12
1.4 Tendencias y tecnologías actuales para los Sistemas de Gestión Documental. ....	13
1.4.1 Herramientas de Gestión Documental .....	13
1.4.2 ¿Por qué Alfresco?.....	16
1.5 Lenguajes de Programación y Tecnologías del lado del Cliente. ....	18
1.6 Lenguajes de Programación y Tecnologías del lado del Servidor. ....	20
1.6.1 ¿Por qué PHP?.....	21
1.7 Sistemas Gestores de Base de Datos .....	22
1.7.1 ¿Por qué PostgreSQL? .....	24

1.8 Servidores Web.....	24
1.9 Framework.....	26
1.9.1 ¿Por qué CodeIgniter?.....	28
1.10 Patrones.....	28
1.10.1 Patrones de diseño.....	29
1.10.2 Patrones de casos de uso.....	34
1.11 Requerimientos.....	35
1.12 Flujo de Análisis y Diseño en RUP.....	38
1.13 Conclusiones.....	39
Capítulo II Características del Sistema.....	40
2.1 Introducción.....	40
2.2 Flujo actual de los procesos.....	40
2.3 Objeto de automatización.....	40
2.4 Modelado del Sistema de Gestión Documental.....	41
2.4.1 Descripción de los procesos de negocio.....	42
2.4.2. Tratamiento archivístico: Descripción de los documentos.....	43
2.4.3 Cuadro de Clasificación.....	43
2.5 Modelado del Negocio.....	47
2.5.1 Justificación de Actores y Trabajadores del Negocio.....	48
2.5.2 Diagrama de Casos de Uso del Negocio.....	49
2.5.3 Descripción de los Casos de Uso del Negocio.....	50
2.5.4 Diagrama de Actividades.....	56
2.5.5 Modelo de Objeto.....	57

2.5.6 Reglas del Negocio. ....	57
2.5.7 Glosario de Términos.....	58
2.5.8 Propuesta de Sistema. ....	58
2.5.9 Especificación de Requerimientos de Software .....	58
2.6 Actores del Sistema.....	63
2.6.1 Definición de los Casos de Uso del Sistema. ....	64
2.6.2 Descripción Textual de los Casos de Uso. ....	64
Capítulo III Análisis & Diseño del Sistema .....	83
3.1 Introducción.....	83
3.2 Modelo de Análisis.....	83
3.3 Diagramas de interacción .....	84
3.3.1. Diagramas de Secuencia. ....	84
3.4 Modelo de Diseño .....	85
3.4.1 Diagrama de Clases del Diseño .....	85
3.4.2 Extensiones UML para Web utilizadas en el sistema. ....	85
3.5 Patrones aplicados en el Sistema de Gestión de Documentos Digitales.....	86
3.6 Descripción de la arquitectura propuesta en el sistema .....	88
3.7 Métricas.....	89
3.7.1 Introducción .....	89
3.7.2 Métricas de Casos de uso .....	90
3.7.3 Métricas de la calidad de la especificación para determinar la ambigüedad.....	92
3.7.4 Métricas para el Modelo de Diseño .....	94
3.8 Conclusiones.....	95

---

Conclusiones Generales .....	96
Recomendaciones.....	97
Bibliografía Referenciada .....	98
Bibliografía Consultada .....	100
Anexos.....	102
Anexo 1 Herramientas de Gestión Documental .....	102
Anexo 2 Diagrama de Actividades .....	106
Anexo 3 Glosario de Términos.....	110
Anexo 4 Prototipos de Interfaz de Usuarios.....	112
Anexo 5 Diagrama de Clases del Análisis .....	117
Anexo 6 Diagramas de Clases del Diseño.....	121
Anexo 7 Diagramas de secuencia .....	132

## INTRODUCCIÓN

Hoy constituye la información el recurso clave de la economía, de las organizaciones, del mundo cultural y de la política, es por eso que se trata de organizar de manera que los usuarios puedan encontrar las respuestas correctas a sus necesidades informativas. Los grandes beneficios de la informática en el campo de los sistemas de información han hecho que su uso se masifique por todas las latitudes y que sea, cada vez mayor, el número de usuarios que prefieren utilizar las computadoras y las tecnologías para automatizar procesos que involucren documentos electrónicos en su trabajo diario.

Actualmente, en las empresas se manejan un volumen considerable de documentación (procedimientos, notas, normativas, actas de reuniones, informes de trabajo, entre otros), por lo que se hace necesario la implantación de sistemas que permitan controlar la gestión de documentos que se genere en la entidad, lo cual posibilita establecer la organización y tratamientos adecuados a la documentación y por ende una eficiente y rápida recuperación de la información.

A esto se añade la necesidad, en algunos entornos, de controlar las distintas versiones de los documentos, así como su trayectoria, desde la creación (nace) del documento hasta su fase de expurgo (eliminación) o conservación permanente; por tanto, el hecho de poseer un sistema que gestione el ciclo de vida de los documentos, se convierte en una necesidad y en una gran ventaja para las instituciones modernas de hoy día.

La creciente necesidad de gestionar toda la generación de información en una entidad, y el reconocimiento de la informática como herramienta y no como base de la gestión de la información, adquiere gran importancia estratégica. En los diferentes ámbitos informáticos, desde el más simple hasta el más complejo el desarrollo está dado, en gran medida, por la precisión y efectividad del sistema de información que soporte su funcionamiento. Un sistema con información veraz, confiable, precisa y bien estructurada garantizará el éxito de las decisiones de sus usuarios (trabajadores de la entidad), y apoya al mismo tiempo la toma de decisiones de la institución.

En el grupo de Arquitectura y Estándares de Información de la Dirección Técnica (DT) de la Infraestructura Productiva (IP) de la Universidad de Ciencias Informáticas (UCI), se genera gran cantidad de información y documentación como: planes de trabajo, autoevaluaciones, actas de reuniones, planes de capacitación,

informes de trabajo con los proyectos, normativas de trabajo, presentaciones, artículos, entre otros, y no se tiene un control eficiente de los mismos. La documentación muchas veces está repetida, no está clasificada y es de difícil acceso por lo que se ven afectados los procesos internos lo cual incide directamente en el resultado de trabajo del grupo hacia el entorno y viceversa.

Por consiguiente, existe la necesidad de establecer un sistema que soporte la gestión documental del grupo de A&EI y contribuya a un desarrollo organizado, rápido y con calidad de estos procesos.

Partiendo de esta situación, se identifica el siguiente **problema a resolver**:

La inexistencia de un sistema informático para la gestión de documentos digitales en el grupo de A&EI provoca que haya ineficiencia en el control del flujo de documentos que se genera en cada uno de los procesos internos del grupo.

De esta forma, se define como **objetivo general** del presente trabajo:

Realizar el Análisis y Diseño de un sistema que permita una adecuada y rápida gestión de los documentos digitales del grupo de A&EI de la DT en la IP.

De este objetivo general se derivan los siguientes **objetivos específicos**:

- 1) Efectuar un levantamiento de información de sistemas de gestión de documentos digitales y metodologías para el proceso de desarrollo de software y herramientas existentes para la gestión documental.
- 2) Identificar las deficiencias existentes en cuanto a la gestión de la documentación en el grupo de Arquitectura y Estándares de Información de la DT en la IP.
- 3) Proponer las herramientas para la gestión de documentos digitales del grupo de A&EI.
- 4) Obtener los artefactos correspondientes al flujo de trabajo de análisis y diseño para el sistema de gestión de documentos digitales.
- 5) Validar los resultados obtenidos en el análisis y diseño del sistema de gestión de documentos digitales.

Se define como **objeto de estudio**: proceso de gestión de documentos digitales y como **campo de acción**: análisis y diseño dentro del proceso de desarrollo de software de un sistema para la gestión de documentos digitales del grupo de Arquitectura y Estándares de Información de la DT en la IP.

La **hipótesis** es la siguiente: Si se desarrolla un adecuado análisis y diseño del sistema para la gestión de documentos digitales en el grupo de A&Ei entonces los programadores podrán implementar el sistema que erradicaría las deficiencias en el proceso de gestión de documentos digitales, agilizaría las consultas a las fuentes documentales existentes y todo esto posibilitará un adecuado seguimiento del flujo de la información y control de cambios.

Para dar cumplimiento a los objetivos planteados se definieron las siguientes **tareas de investigación**:

- Consulta de material bibliográfico acerca de los diferentes sistemas para la gestión de documentos digitales y las metodologías para el desarrollo de software, así como de las herramientas para la gestión de documentos digitales.
- Descripción de la propuesta de herramienta para la gestión de documentos digitales.
- Descripción de la propuesta de la metodología a utilizar para el proceso de desarrollo.
- Captura de requerimientos que respondan a las necesidades del sistema.
- Descripción de los casos de uso del negocio y sistema.
- Realización del diagrama de casos de usos del negocio y del sistema.
- Realización de una propuesta de diseño acorde con las condiciones planteadas en el análisis.
- Aplicación de un método que permita teóricamente evaluar los resultados de la investigación.

Los **resultados esperados del trabajo** son los siguientes:

La continuidad del estudio de este trabajo en la implementación de un sistema para la gestión de documentos digitales conlleva a un aumento del nivel tecnológico de informatización en el grupo de Arquitectura y Estándares de Información de la DT en la IP para favorecer la gestión y control de la información generada por los procesos que se realizan.

Para el desarrollo de las tareas de investigación se emplearon los siguientes **métodos científicos**:

**Métodos Teóricos:**

- **Analítico- Sintético:** Para entender fenómenos relacionados con el proceso de gestión documental, así como las tendencias actuales del desarrollo de software, a partir de la documentación analizada, lo que permitió la extracción de elementos importantes relacionados con el objeto de estudio.



- **Histórico- Lógico:** Permite constatar teóricamente la evolución del proceso de gestión documental así como las herramientas actuales que permiten la materialización de este proceso.

### **Métodos Empíricos**

- **Entrevista:** Para obtener información acerca de las principales características de los procesos de negocio al cual responde el sistema.
- **Observación:** Posibilita percibir el desarrollo de la investigación en distintos momentos de la misma, conociendo el problema y el objeto de investigación. (Hernández León & Coello González, 2002)

El trabajo está estructurado en 3 capítulos de los que se realiza una breve descripción a continuación:

El **Capítulo I** contiene la **Fundamentación Teórica** del trabajo, abordando todos los elementos teóricos que sustentan el problema científico y los objetivos del trabajo. Se describen aspectos importantes sobre el proceso de gestión de documentos digitales y se realiza un análisis de las tendencias de las tecnologías y metodologías actuales en el desarrollo de software. También se fundamenta la selección de las tecnologías y herramientas con las cuales se debe desarrollar el sistema.

En el **Capítulo II** se describen las **Características del sistema**, realizando el diseño documental del sistema y una propuesta de la investigación y los procesos que están vinculados al campo de acción. Se plantean los requisitos funcionales y no funcionales que debe tener el sistema para lograr óptimos resultados, además de hacer referencia a las personas vinculadas al mismo.

En el **Capítulo III** se desarrolla el **Análisis y Diseño** para el sistema, exponiéndose los aspectos relacionados con el análisis del sistema basándose en una detallada descripción de las diferentes funcionalidades para una mejor comprensión de las mismas facilitando el trabajo de los desarrolladores y mediante el diseño del sistema se ofrece una visión de cómo será el proyecto que se propone. Se aplican métricas de calidad para validar los resultados obtenidos en el análisis y diseño del sistema.

## Capítulo I Fundamentación Teórica

### **1.1 Introducción**

En el presente capítulo se tratan los principales conceptos relacionados con el tema de los sistemas de gestión documental y se explican las características de las principales tecnologías y herramientas que son utilizadas para el desarrollo de estas aplicaciones. También se aborda acerca de las diferentes metodologías para el proceso de desarrollo de software y se realiza una valoración de las ventajas y desventajas de cada elemento tratado, llegando finalmente a la conclusión de qué se debe utilizar en la solución que se propone.

La información obtenida en este análisis ha sido consultada y recopilada a través de revisiones bibliográficas, entrevistas a expertos y especialistas de la Dirección Técnica, así como otras fuentes de información.

### **1.2 Fundamentación del tema**

#### **1.2.1 Definiciones importantes en el proceso de Gestión documental**

*Documento electrónico:* Toda representación de un hecho, imagen o idea que sea creada, enviada, comunicada o recibida (por una persona o institución) por medios electrónicos y almacenada de un modo idóneo para permitir su uso posterior.

*Gestión documental:* Conjunto de normas, técnicas y prácticas usadas para administrar el flujo de documentos de todo tipo en una organización, permitir la recuperación de información desde ellos, determinar el tiempo que los documentos deben guardarse, eliminar los que ya no son útiles y asegurar la conservación indefinida de los documentos más valiosos, aplicando principios de racionalización y economía. (Codina, 2003)

Otra definición para Gestión documental.

*Gestión documental:* Conjunto de procedimientos y operaciones que pretenden el establecimiento de normas y directrices sobre la creación y tratamiento de documentos electrónicos y su conservación.

*Sistemas de gestión de documentos electrónicos:* Sistema computarizado, conjunto de programas utilizado para rastrear y almacenar documentos electrónicos. El término puede ser relacionado con conceptos como sistemas de administración de contenido (CMS) y es comúnmente visto como un

sistema de administración de contenido corporativo y relacionado con el término Digital Asset Management.

### **1.2.2 Características de los Sistemas de Gestión Documental**

Según Calzado en su artículo Suplemento ComputerWorld #876 de Gestión documental del 2003, los sistemas de gestión documental reúnen las siguientes características:

- **Rápido Almacenamiento:** Disponen la totalidad de información que se desee exponer a través de la aplicación en una única base de datos, de manera que la información no se encuentra diseminada en la organización en diferentes espacios físicos, sino que a partir de esa Base de Datos, se dispone la información en un único espacio virtual.
- **Creación de documentos de manera estructurada.** Definen plantillas de manera tal que los documentos que se crean van a estar estructurados de acuerdo al tipo de documento que se desee.
- **Control de documento en su ciclo de vida.** Estos sistemas controlan el ciclo de vida del documento, desde su creación y tratamiento hasta su fase final de expurgo o conservación permanente.
- **Fácil Recuperación:** La búsqueda en SGD es muy fácil, se puede realizar búsquedas de información de documentos específicos dentro del repositorio o dentro del propio documento. Para ello se utilizan los metadatos.
- **Óptima Clasificación:** Los documentos están clasificados respetando el principio de procedencia y organizado en árbol de navegación dinámico. Toda esta información y documentación referida, estará estructurada y organizada bajo un árbol de navegación cuyo uso intuitivo a modo de explorador, permita una sencilla actualización por parte de la totalidad de usuarios, que a través de directorios (al estilo de las carpetas del explorador del ordenador) se ubican en espacios o carpetas por series, subseries documentales y unidades documentales.
- **Autenticación de los usuarios:** Cada persona involucrada en el SGD tiene un rol asociado, con los permisos correspondientes, de manera que el acceso sea diferenciado.
- **Workflow:** Conjunto de reglas establecidas para el tratamiento de los documentos a través de su ciclo de vida.

- **Control de Versiones:** Los documentos que se van a almacenar en la aplicación son los actualizados, de manera que se evita que ocurran problemas por la existencia todavía en la organización de informaciones antiguas, descatalogadas que puedan provocar errores.
- **Dato único:** No se genera la misma información en diferentes lugares, sino que a partir de un único original, se llega a la totalidad de las personas que necesiten la información vía web.

### **1.3 Tendencias y tecnologías actuales para el proceso de desarrollo de software.**

#### **1.3.1 Metodologías de desarrollo de software.**

##### **Introducción**

En los principios de la historia del software, no existía una guía, procedimiento, ni metodología para el desarrollo del mismo, los clientes no tenían muchas exigencias y el software no jugaba un papel importante en la sociedad. Como el desarrollo de la informática está en constante evolución y la demanda de los clientes es cada vez mayor se han creado estándares y metodologías para organizar el trabajo, basándose en métodos comunes para el desarrollo. Estas metodologías fueron diseñadas bajo un conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para lograr la calidad, que es el principal objetivo estratégico en las organizaciones.

En un proyecto de desarrollo de software la metodología define *Quién* debe hacer *Qué*, *Cuándo* y *Cómo* debe hacerlo, de ahí la importancia que tiene hacer una correcta elección de la metodología a emplear. Hoy día no existe una metodología que sea enteramente universal por eso para su selección hay que tener en cuenta las características de cada proyecto (equipo de desarrollo, recursos, etc.). Existen metodologías tradicionales y metodologías ágiles. Entre las tradicionales se hace un estudio y análisis de la metodología orientada a objeto: RUP y MSF y dentro de las ágiles: SCRUM, XP, para poder escoger la más factible de acuerdo a las características del proyecto.

##### **1.3.1.1 Metodologías tradicionales o convencionales.**

Las metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el objetivo de conseguir un software más eficiente y predecible. Para ello, se hace un especial hincapié en la planificación total de todo el trabajo a realizar y una vez que está todo detallado, comienza el ciclo de desarrollo del producto del software. Este planteamiento está basado

en el resto de disciplinas de ingeniería, a pesar de que el software no pueda considerarse como la construcción de una obra clásica de ingeniería. (Cáceres & Marcos, 2000).

### **RUP. Rational Unified Process.**

La metodología RUP, proceso unificado de desarrollo de software, es una metodología pesada, está basada en una notación gráfica, la cual permite especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos. (García Rubio).

Sus principales características son:

- **Guiado por casos de uso:** Los casos de uso son el instrumento para describir el comportamiento del software y extraer los casos de prueba con los que se valida el sistema.
- **Centrado en la arquitectura:** Los modelos son proyecciones del análisis y el diseño, describe la arquitectura del producto a desarrollar.
- **Iterativo e incremental:** Durante todo el proceso de desarrollo se producen versiones superiores.

RUP utiliza como lenguaje de modelado el Unified Modelling Language (UML) para describir todo el proceso. Se divide en ciclos de trabajo, teniendo un producto superior como resultado de cada ciclo. Éstos se componen en su interior por varias fases, en la cuales se llevan a cabo un conjunto de flujos para el desarrollo de todo el proyecto.

Esta metodología divide en 4 fases el desarrollo de software:

- **Inicio:** El Objetivo en esta etapa es determinar la visión del proyecto.
- **Elaboración:** En esta etapa el objetivo es determinar la arquitectura óptima.
- **Construcción:** En esta etapa el objetivo es llegar a obtener la capacidad operacional inicial.
- **Transición:** El objetivo es llegar a obtener el release del proyecto.

En RUP las actividades son agrupadas en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo.

### Flujos de trabajo:

- **Modelamiento del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
  - **Prueba (Testeo):** Busca los defectos a lo largo del ciclo de vida.
  - **Instalación:** Produce release del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
  - **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
  - **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
  - **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

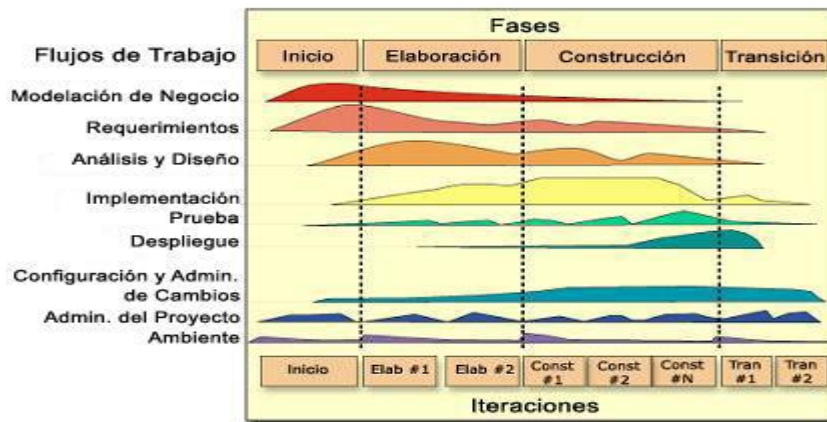


Figura 1.1 Rational Unified Process (RUP).

### **MSF. Microsoft Solutions Framework**

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. (Zabala, 2008)

### **MSF tiene las siguientes características:**

- **Adaptable:** es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.
- **Escalable:** puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas a más.
- **Flexible:** es utilizada en el ambiente de desarrollo de cualquier cliente.
- **Tecnología Agnóstica:** porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

### **MSF consta de 5 fases fundamentales:**

- 1- Visión y Alcances.
- 2- Planificación.
- 3- Desarrollo.
- 4- Estabilización.
- 5- Implantación



Figura 1.2 Microsoft Solutions Framework (MSF)

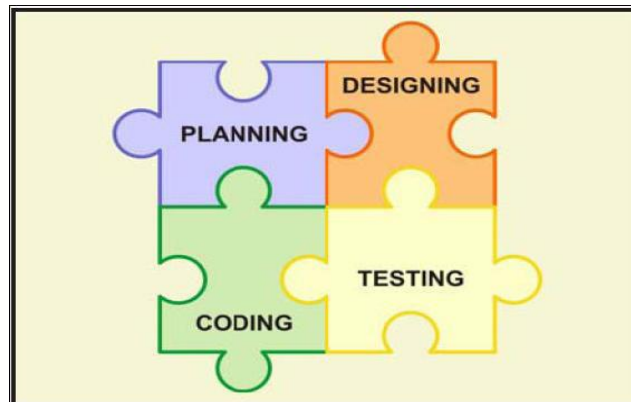
### 1.3.1.2 Metodologías ágiles o ligeras

Las Metodologías Ágiles o “ligeras” constituyen un nuevo enfoque en el desarrollo de software, mejor aceptado por los desarrolladores de e-projects (proyectos) que las metodologías convencionales (ISO-9000, CMM, etc.) debido a la simplicidad de sus reglas y prácticas, su orientación a equipos de desarrollo de pequeño tamaño, su flexibilidad ante los cambios y su ideología de colaboración. (Canós, Letelier, & Penadés)

#### ***XP (Extreme Programming – Kent Beck) (Programación Extrema)***

XP es una de las metodologías ágiles que más terreno ha ganado en los últimos tiempos, la misma fue diseñada para su uso en proyectos de corto plazo que por lo general necesitan desarrollar software de manera rápida en un ambiente en el cual los requerimientos suelen ser rápidamente cambiantes. Es orientada al cliente, y de iteraciones cortas. La base para el desarrollo del software que usa esta metodología son las llamadas User Stories, historias escritas por el cliente en las que describen escenarios sobre el funcionamiento del sistema y que no sólo están limitados los formularios de usuario, sino que también pueden describir modelos.





**Figura 1.3 Metodología Programación Extrema.**

Estas User Stories junto a la arquitectura que se persigue, sirven de base para crear un plan de “entregas de software” entre el equipo de desarrollo y el cliente. Propone que en el equipo de desarrollo se necesita un representante constante del cliente que conozca al dedillo el negocio y que esté a disposición para cualquier duda o necesidades que los desarrolladores necesiten. El cliente se mantiene todo el tiempo informado paso por paso de las cosas que se están desarrollando, y a medida que se va liberando de cualquier entregable se discutirá con el representante, y se repite la nueva iteración del software.

La programación del software siempre se define en pareja con el objetivo principal de lograr mayores resultados y los menores errores posibles, mientras uno codifica haciendo hincapié en la calidad de la función o método que está implementando, el otro analiza si ese método o función es adecuado y está bien diseñado. Es necesario que los desarrolladores se reúnan diariamente y expongan sus problemas, soluciones e ideas de forma conjunta. (Thomson, 2003)

***SCRUM (Jeff Sutherland, Ken Schwaber)***

**SCRUM** define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. (Canós, Letelier, & Penadés)

## Características

- Equipos auto dirigidos
- Utiliza reglas para crear un entorno ágil de administración de proyectos
- No prescribe prácticas específicas de ingeniería
- Los requerimientos se capturan como ítems de la lista reserva del producto
- El producto se construye en una serie de Sprints de un mes de duración.

## Proceso. Scrum consta de tres fases:

- 1- Preamble (antes del juego).
- 2- Development.(Desarrollo)
- 3- Postgame.(después del juego)

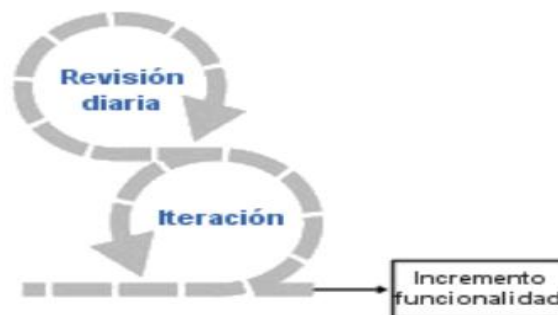


Figura 1.4 SCRUM

### 1.3.1.3 ¿Por qué RUP?

Luego de analizar las metodologías antes mencionadas se puede apreciar que la metodología RUP es más adaptable a las deficiencias encontradas en el grupo de Arquitectura y Estándares de Información(A&EI) por ser una metodología orientada a objetos, además la metodología XP no cuenta con la documentación necesaria, solamente se limita a documentar la que le corresponde al código fuente, por ejemplo la documentación de los diagramas UML es prácticamente nula, esto representa una desventaja ya que no presenta un documento que registre la experiencia ganada a

través del trabajo, y también entorpece la reutilización de las próximas versiones del software. En cambio RUP se basa en la documentación de todo el trabajo desarrollado, ejemplo de esto es que desarrolla todos los diagramas UML, logrando así que las próximas versiones del software puedan reutilizar, sin problemas mayores, aspectos de la primera versión a entregar. Un aspecto fundamental es que en la metodología XP no existe un análisis de los requerimientos profundo, lo que trae como consecuencia que durante el desarrollo de la aplicación se tengan que realizar cambios que se podían identificar anteriormente, en cambio RUP si da una mayor importancia a la fase en que se realiza el levantamiento de requisitos, teniendo definidos roles y artefactos que describen el trabajo realizado durante la ingeniería de requisitos.

### **1.3.2 Lenguaje de Modelado**

#### ***Unified Modeling Language (UML)***

UML es un lenguaje para especificar, construir, visualizar y documentar los artefactos (información que se utiliza o produce mediante un proceso de software). Este lenguaje de modelado no es una guía para realizar el análisis y diseño orientado a objeto, es decir, no es un proceso, es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos.

El modelado visual permite manejar la complejidad de los sistemas a analizar o diseñar. Un modelo es expresado en un lenguaje de modelado compuesto por vistas, elementos de modelos y un conjunto de mecanismos generales o reglas que indican como utilizar los elementos. Las vistas muestran diferentes aspectos del sistema modelado. Una vista es una abstracción que se puede definir como una fotografía completa del sistema. Los diagramas son la representación gráfica de un conjunto de elementos describiendo el contenido de una vista y visualizan un sistema desde diferentes perspectivas. (Anónimo, 2007-2008)

UML intenta solucionar el problema de propiedad de código que ocurre con muchos desarrolladores. Al implementar un lenguaje de modelado común para todos los desarrollos se crea una documentación común de igual manera, que cualquier desarrollador con conocimientos de UML será capaz de entender, independientemente del lenguaje utilizado para este.

Su utilización es independiente del lenguaje de programación y de las características de los proyectos, ya que UML ha sido diseñado para modelar cualquier tipo de proyectos, tanto informáticos como de arquitectura, o de cualquier otra rama.

### ***Business Process Management Language (BPML)***

- Es utilizado para el modelado de procesos de negocio.
- Define un modelo abstracto y la gramática para expresar procesos genéricos, lo que le permite ser usado para una variedad de propósitos entre los que se incluyen: definición de procesos empresariales, definición de complejos servicios Web y definición de colaboraciones multiparte.
- Unifica la semántica de definición de proceso, permitiendo desarrollar un solo procesador que pueda ejecutar cualquier proceso de negocio.
- Sobre este lenguaje se puede decir que, aunque presente características bastante significativas, no es considerado como el más adecuado para ser utilizado en el desarrollo del futuro sistema, ya que se enmarca sólo en los procesos del negocio y precisamente lo que se busca es una notación que abarque todo el ciclo de vida del desarrollo del software. (García, 2005)

### ***1.3.2.1 ¿Por qué UML?***

De forma general las principales características y ventajas de UML que contribuyeron a su elección son:

- Lenguaje unificado para la modelación de sistemas.
- Tecnología orientada a objetos.
- El cliente participa en todas las etapas del proyecto.
- Corrección de errores viables en todas las etapas.
- Aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y cliente/servidor.

Ventajas de UML

- Mayor rigor en la especificación.

- Permite realizar una verificación y validación del modelo realizado.
- Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y a la inversa (a partir del código fuente generar los modelos). Esto permite que el modelo y el código estén actualizados, con lo que siempre se puede mantener la visión en el diseño, de más alto nivel, de la estructura de un proyecto.

### **1.3.3 Herramientas CASE**

Dentro de las herramientas más utilizadas para el desarrollo de software hoy día están las herramientas CASE (Computer Aided Software Engineering) o Ingeniería de Software Asistida por Computadora, éstas representan un conjunto de aplicaciones informáticas que tienen como objetivo automatizar los aspectos claves de todo el proceso de desarrollo de un sistema, compensando de esta manera su coste inicial en forma de ahorro de tiempo y recursos para el proyecto en cuestión.

#### ***Visual Paradigm***

Visual Paradigm (VP) utiliza UML como Lenguaje de Modelado, es una Herramienta CASE poderosa y fácil de usar. Permite representar todo tipo de diagramas UML para las distintas fases como la captura de requisitos, análisis, diseño e implementación, permite además generar códigos, aplicar ingeniería inversa en los lenguajes Java, C++, CORBA IDL, PHP, XML, Ada y Python. Además la generación de código apoya C#, Visual Basic .NET, Object Definition Language (ODL), Flash, ActionScript, Delphi, Perl, Objective-C, y Ruby. Mediante esta herramienta se puede desarrollar un producto de calidad teniendo el diseño centrado en casos de usos y enfocado al negocio. Facilita la comunicación ya que utiliza un lenguaje estándar común a todo el equipo de desarrollo. También posee disponibilidad de múltiples versiones, en múltiples plataformas y es fácil de instalar y actualizar. Presenta la posibilidad de la interoperabilidad con otras aplicaciones como es el Rational Rose. Tiene disponible distintas versiones: Enterprise, Professional, Standard, Modeler, Personal y Community (que es gratuita). También facilita licencias especiales para fines académicos.

#### ***Rational Rose.***

Rational Rose Enterprise Edition es una de las herramientas más poderosas para el modelamiento visual. Se utiliza para modelar un sistema antes de desarrollarse. Permite crear y refinar, logrando así un modelo completo que representa el dominio del problema y el sistema de software. Permite especificar, analizar y diseñar el sistema antes de codificarlo.

Con dicha herramienta se puede generar código en distintos lenguajes de programación a partir de un diseño en UML y proporciona mecanismos para realizar la denominada ingeniería inversa, es decir, a partir del código de un programa podemos obtener información sobre su diseño. Una de las deficiencias que posee este tipo de herramienta es que no es soportada en el Sistema Operativo Linux, es decir no es multiplataforma, otra deficiencia es que es propietaria ya que se tiene que pagar por su licencia para poder trabajar. (Ramos Rodríguez, 2008)

### ***1.3.3.1 ¿Por qué Visual Paradigm?***

Después del análisis de estas dos herramientas CASE: Visual Paradigm y Rational Rose, se seleccionó Visual Paradigm ya que es una herramienta multiplataforma, algunas de sus versiones son gratuitas en cambio Rational Rose es una herramienta muy usada pero es software propietario y no es multiplataforma. Además: es una herramienta que permite el modelado orientado a objetos y está orientada al Lenguaje de Modelado seleccionado anteriormente, UML.

## ***1.4 Tendencias y tecnologías actuales para los Sistemas de Gestión Documental.***

### ***1.4.1 Herramientas de Gestión Documental***

Un Sistema de Gestión de Documentos Digitales permite la creación y tratamiento de documentos principalmente en páginas Web.

### **ECM (Manejo de Contenidos y documentos Empresariales)**

Existen en la actualidad muchas empresas en las que se generan gran cantidad de documentos, esto provoca que no esté organizada y se pierda información, la cual puede ser útil e importante, para evitar que esto ocurra se desarrollan aplicaciones (ECM), las cuales se encargan del manejo de contenidos y documentos. Aplicaciones como: Magnolia, Nuxeo, DocManager, Alfresco, entre otras.

**(Ver Anexo 1)**



**Figura 1.7 Logo de la herramienta Magnolia**

### ***Magnolia***

La herramienta Magnolia es utilizada en muchas empresas con el objetivo de administrar contenidos y funciona de manera similar a los CMS, por lo que funciona desde un navegador y se instala y/o hospeda en un servidor o hosting. Esta ha sido reconocida por su eficiencia y resultados óptimos en el manejo de documentos y contenidos.

Magnolia ofrece grandes ventajas y provechos por ejemplo: está elaborada totalmente en Código Libre por lo que está disponible para cualquier usuario que desee implementarla para que lo adapte a sus necesidades, su distribución e instalación es igualmente gratuita, es de fácil uso, robusta, extensible y adaptable a cualquier organización en la que se implemente su sistema.

El sistema cumple con grandes perspectivas de trabajo, es rápido y permite actualizar los documentos de forma sencilla y constante, además de evaluar periódicamente la calidad, estado y cantidad del contenido administrado y el uso de los contenidos almacenados, en caso de que se encuentren algunos que ya no deseen ser conservados.

Incorpora en su sistema gran variedad de herramientas entre las que más se destacan se encuentran las herramientas de backup, DMS o documentos, registro de eventos, filtro de imágenes, Foros, entre otros, también integra módulos de trabajo los cuales permiten que se le agreguen nuevas herramientas y/o capacidades adicionales de trabajo para la aplicación ejemplo de estos módulos se encuentran: Módulo o herramienta de backup, Módulo DMS o administrador de documentos, Módulos de Foros, entre otros. (Helade, 2008)

### ***Alfresco***



**Figura 1.8 Logo herramienta Alfresco**

Alfresco es un sistema de administración de contenidos empresariales de código libre/abierto, que permite gestionar documentos, expedientes, contenido web e imágenes, además permite la

colaboración entre los distintos usuarios del sistema mediante flujos de trabajo para los documentos, está basado en estándares abiertos y de escala empresarial para Windows y sistemas operativos similares a Unix.

Gestiona el ciclo de vida de los documentos: crear, compartir, versionar, aprobar, publicar, retirar, archivar, facilita el trabajo colaborativo mediante foros, notificaciones, RSS, Blogs, Wiki.

Debido a que Alfresco está diseñado para ser un sistema ECM (Enterprise Content Management) puede utilizarse de distintas formas, algunas de las cuales son: repositorio empresarial de documentos, intranet, portal de gestión de conocimiento, repositorio escalable de contenidos, publicación bajo demanda (como si se tratase de un medio de comunicación), portales colaborativos (para compartir información y aplicaciones), manejo de formularios y procesos de aprobación.

Alfresco constituye la alternativa líder en el mercado de gestión de contenidos empresariales al combinar la innovación del software Open Source, la experiencia en el campo de gestión de documentos por parte de sus creadores y la facilidad de uso de una plataforma de nivel empresarial.

Alfresco suministra características importantes en cuanto a robustez seguridad y escalabilidad, para llevar contenidos confiables y relevantes a clientes, proveedores y empleados de una organización.

**Características** que se destacan de Alfresco y que lo diferencia de otras soluciones para sistemas de gestión de documentos a nivel empresarial se pueden mencionar: (Fuente, 2008)

- **Integral:** Con Alfresco es posible integrar el manejo de documentos de toda una organización mediante el control automatizado del flujos de documentos, además de asignar roles y permisos para los usuarios que intervienen en el ciclo de vida de los documentos.
- **Seguro:** Alfresco provee funcionalidades de seguridad sobre el acceso a la información y la privacidad de datos. Se pueden asignar permisos a espacios (carpetas) o elementos individuales dentro de un espacio. La autenticación de usuarios se consigue a través de su propia base de datos o identificación a través de otros mecanismos, como LDAP, NTLM, Kerberos y Active Directory.
- **Accesible vía Web:** En lugar de depender únicamente de programas específicos para cada sistema operativo, los usuarios pueden acceder a todas las funcionalidades que ofrece Alfresco desde cualquier ubicación, vía Web. El cliente está basado en tecnologías que proveen a los usuarios de una interfaz amigable y que proporcionan todas las funcionalidades



ofrecidas por Alfresco de manera remota.

- **Compatibilidad:** Alfresco soporta múltiples esquemas para compartir documentos en contextos empresariales (como CIFS / Samba, FTP y WebDAV), facilitando la adaptación de los usuarios. La versión desplegable en servidor puede ejecutarse en sistemas Windows, Linux y Mac OS X, asimismo se pueden usar distintos manejadores de bases de datos (MySQL, Oracle, MS SQL Server), servidores de aplicaciones web (JBoss Application Server, Apache Tomcat, entre otros) y portales de uso opcional (JBoss Portal, Liferay).
- **Escalable:** En cualquier empresa, la cantidad de contenidos administrables crece con el paso del tiempo, por lo cual la escalabilidad es una característica clave de un ECM. Debido a su arquitectura ligera y modular, Alfresco provee escalabilidad tanto a nivel de capas de despliegue como particionamiento y balanceo de carga en múltiples servidores. De manera similar, Alfresco puede ser escalable respecto a cantidad de información, mediante el uso de búsquedas complejas, estructura y clasificación de información. Todo ello se consigue sin incurrir en costos adicionales por volumen de datos.
- **Open Source:** Alfresco es una plataforma de gestión y generación colaborativa de documentos basada en tecnologías de código, formatos y protocolos abiertos, muchos de los cuales son de uso extendido entre los desarrolladores de aplicaciones web. Con ello, Alfresco, consigue hacer uso de lo mejor de las contribuciones de la comunidad Open Source para obtener un software de calidad que mejora rápidamente.

### 1.4.2 ¿Por qué Alfresco?

Se seleccionó la herramienta Alfresco en su versión 2.1.0 bajo una licencia GPL por las ventajas que brinda por ejemplo:

- Recuperación completa desde un repositorio en formato y estructura Alfresco-ECM a un sistema de ficheros y directorios.
- Los documentos son renombrados al nombre original con el que se integraron en Alfresco-ECM y la estructura lógica de espacios a su equivalente en directorios dentro del sistema de ficheros.
- Incorpora la gestión documental que permite tener organizados por carpetas y por categorías grandes volúmenes de documentos por ejemplo: documentos ofimáticos, presentaciones, imágenes, xml, multimedia, etc, Este tipo de clasificación es muy flexible y permite que

documentos que lógicamente deben estar en diferentes carpetas, compartan la misma categoría facilitando una búsqueda más rápida de la información.

- Es seguro en el acceso y conservación de los documentos.
- Reutilización del conocimiento. Como Alfresco es un gestor documental garantiza que toda la documentación de la empresa, grupo o institución se mantengan conservada y al alcance del que la necesite por ejemplo: si se publica en un espacio un material, todo el que tenga acceso a ese espacio podrá hacer uso de ese material, pues se comporta como un archivo en una biblioteca.
- Al estar orientado al entorno web, es muy fácil acceder a Alfresco desde cualquier sitio, incluso sin estar físicamente en la propia red en la que esté alojado Alfresco.
- Su programa de pruebas hospedadas en sus servidores permite evaluar si Alfresco cubre las necesidades que tengamos sin tener que invertir dinero y tiempo en una máquina o en la instalación de Alfresco en un servidor propio.
- Alfresco está desarrollado con tecnología J2EE y la forma de acceso básica es a través de su cliente web, lo que convierte Alfresco en un software multiplataforma casi sin trabajo de desarrollo adicional. (Fuente, 2008)

Otras de las ventajas que ofrece Alfresco es que permite:

- **Gestión documental:** Alfresco proporciona una solución intuitiva que simplifica la captura, búsqueda y colaboración de documentos con una completa colección de servicios y administración de ciclo de vida en un repositorio unificado y robusto.
- Se incluye soporte para Virtual File System (en lugar de discos compartidos, pero con la misma facilidad para acceder a los documentos desde el Explorador de Windows), y una interfaz web que proporciona búsqueda de archivos de manera similar a la ofrecida por Google y navegación de archivos similar a Yahoo.
- **Gestor de Imágenes:** Las imágenes pueden ser capturadas a través de distintos mecanismos (Common Internet File System: CIFS, Kofax Ascent Capture; y en el caso de la interfaz web es posible realizar la captura mediante servicios Web, HTTP y HTTPS), que luego pueden ser incorporadas al repositorio y clasificadas.

- Respecto a la administración de las imágenes es posible transformarlas entre varios formatos (JPEG, TIFF, PNG, MS Office, PDF y Flash), editar sus propiedades (metadatos) y reutilizarlas en conjunto con las reglas de negocio definidas.
- **Gestor de Expedientes:** Se dispone de una solución para administrar expedientes durante el curso normal del negocio que involucra tareas de control, almacenamiento y uso.
- Contempla captura de expedientes a través de una interfaz estándar del Explorador de Windows, integración con auditorías y flujos de trabajo desde el panel de tareas, exportación simple para guardar copias de expedientes, y conversión a formatos PDF y ODF; además de soportar las especificaciones del DOD 5015.2 (certificación del Departamento de Defensa estadounidense para el manejo de registros digitales).
- **Gestor de Contenido Web:** Ofrece un repositorio único para todo un equipo dedicado a la elaboración de contenidos para Internet.
- Este repositorio es una moderna plataforma para la gestión del contenido al estilo Web 2.0, incluyendo alta disponibilidad, tolerancia a fallos, vista preliminar de actualizaciones en contexto, publicación de contenidos en múltiples sitios. Además de lo anterior, esta funcionalidad contempla soporte para proyectos, servidores de prueba, instalación de cambios, capas y visualización del sitio en fechas anteriores.

### ***1.5 Lenguajes de Programación y Tecnologías del lado del Cliente.***

Los lenguajes de programación del lado del cliente son aquellos que solamente pueden ser interpretados por una aplicación cliente como el navegador Web entre estos lenguajes están DHTML, Javascript, etc.

El HTML Dinámico o DHTML (del inglés Dynamic HTML) es el arte de construir sitios Web dinámicos combinando HTML estático con un lenguaje interpretado con el lado del cliente (Ej. Javascript), las CSS (Cascading Style Sheets) y la jerarquía de objetos de un DOM (Document Object Model). Otra manera de entender qué es DHTML es mediante lo afirmado por la W3C (World Wide Web Consortium) “El HTML dinámico es un término usado por algunos vendedores para describir la combinación del HTML, de las hojas del estilo y las scripts que permiten que las documentos Web sean animados” (Introduction to DHTML). Con el DHTML se pueden desarrollar entre otras cosas menús desplegables, imágenes que cambian al pasar el cursor sobre ellas, objetos en movimiento, botones que permiten desplazar el texto que se está mostrando, textos explicativos que aparecen al

situar el cursor sobre ciertas palabras clave, cronómetros, etc. DHTML ofrece a diferencia de otras herramientas la ventaja de que no requiere ningún tipo de plug-in para poder utilizarlo.

### **Javascript**

Básicamente Javascript 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. De Javascript se puede decir que es un lenguaje orientado a objetos, ya que dispone del mecanismo de herencia, si bien esta se realiza siguiendo el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad. Javascript también utilizado para crear pequeños algoritmos, funciones, etc., encargados de realizar acciones dentro del ámbito de una página Web. En la arena internacional Javascript fue utilizado de forma masiva por la mayoría de sitios de Internet hasta la aparición de Flash ya que este permitía realizar algunas acciones imposibles de llevar a cabo mediante el uso de Javascript. a su vez sin embargo, la aparición de las aplicaciones AJAX programadas con JavaScript le ha devuelto una popularidad sin igual dentro de los lenguajes de programación Web.

### **CSS**

Las Hojas de Estilo en Cascada (Cascading Style Sheets) o CSS, es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos. En la Web las CSS son utilizadas para dar estilo a documentos y XML, separando el contenido de la presentación. Los Estilos definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento. (Guía Breve de CSS). Las CSS sin duda constituyen una de las tecnologías que han permitido revolucionar el concepto de Webs personalizadas.

### **AJAX**

AJAX o (Asynchronous Javascript And XML) no es más que una técnica de desarrollo Web, mediante la cual se puede crear aplicaciones Web más rápidas y cómodas para el usuario. Por medio de esta técnica el cliente puede interactuar con el servidor de manera asincrónica, actualizando las páginas, sin necesidad de volver a cargarlas. Esto se traduce en un aumento significativo de la interactividad, velocidad y usabilidad en las mismas. De AJAX se ha dicho: “Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes.” (Garrett, 2005)

Entre las tecnologías que conforman AJAX están:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

### **1.6 Lenguajes de Programación y Tecnologías del lado del Servidor.**

Cuando se habla de lenguajes del lado servidor se refiere a aquellos lenguajes que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Entre los lenguajes del lado del servidor más utilizados a escala internacional para el desarrollo de páginas Web están PHP, JAVA, ASP, PERL.

### **Java**

Java es un lenguaje de programación orientado a objetos de alto nivel, gran rendimiento, sencillo, gran nivel de seguridad, multiplataforma y contiene las herramientas necesarias para desarrollar cualquier tipo de aplicación. En la actualidad se utiliza en los principales sectores de la industria de todo el mundo y está presente en un gran número de dispositivos, ordenadores y redes de cualquier tecnología de programación, por su versatilidad y eficiencia, la portabilidad de su plataforma y la seguridad que aporta, la han convertido en la tecnología ideal para su aplicación a redes, de manera que hoy en día, más de 2.500 millones de dispositivos utilizan dicha tecnología como son; ordenadores, teléfonos móviles, tarjetas inteligentes, impresoras, Web cams, juegos, sistemas de

navegación para automóviles, terminales de lotería, dispositivos médicos, cajeros de pago en aparcamientos, etc. (ZUKOWSKI, 2003)

### **PHP**

Es el acrónimo de Hipertext Preprocesor, lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, de código abierto, ejecutado al lado del servidor y soportado por la mayoría de los servidores Web de hoy en día, por ejemplo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape e iPlanet, Oreilly Website Pro Server, Caudium, Xitami, OmniHTTPd entre otros, con gran volumen de documentación y con una gran librería de funciones con un espíritu generoso ya que estas funciones son progresivamente construidas por colaboradores desinteresados en nuevas versiones del lenguaje. PHP nos permite embeber sus pequeños fragmentos de código dentro de la página HTML y realizar determinadas acciones de una forma fácil y eficaz sin tener que generar programas programados íntegramente en un lenguaje distinto al HTML. Este lenguaje de programación está preparado para realizar muchos tipos de aplicaciones Web con páginas dinámicas gracias a la extensa librería de funciones con la que está dotado.

#### **1.6.1 ¿Por qué PHP?**

- Específicamente para el desarrollo del sistema que se propone se escogió PHP debido a una serie de características que lo hacen el ideal en lo que a la aplicación se refiere.
- Un potente soporte para gran cantidad de bases de datos entre las que se pueden mencionar InterBase, SQL, MySQL, Oracle, Informix, PostgreSQL, entre otras.
- Integración con varias bibliotecas externas, que dan al desarrollador la posibilidad de realizar cualquier tarea, desde generar documentos en pdf hasta analizar código XML.
- Es software libre, lo que implica menos costes y servidores más baratos que otras alternativas.
- Muy rápido y su integración con la base de datos MySQL y el servidor Apache, le permite constituirse como una de las alternativas más atractivas del mercado.
- Multiplataforma, funciona tanto para Unix (con Apache) como para Windows (con Microsoft Internet Information Server) de forma que el código que se haya creado para una de ellas no tiene porqué modificarse al pasar a la otra.
- Sintaxis inspirada en C, ligeramente modificada para adaptarlo al entorno en el que trabaja.

- Tiene una de las comunidades más grandes en Internet, con lo que no es complicado encontrar ayuda, documentación, artículos, noticias, y más recursos.

### **1.7 Sistemas Gestores de Base de Datos**

Los Sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.

Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

De manera general se pueden definir los objetivos más importantes de los SGBD como los siguientes:

- Evitar la redundancia de los datos, eliminando así la inconsistencia de los mismos.
- Mejorar los mecanismos de seguridad de los datos y la privacidad. Podemos distinguir cuatro tipos de contextos para usar mecanismos de seguridad: seguridad contra accesos indebidos a los datos, seguridad contra accesos no autorizados a la base de datos, seguridad contra destrucción causada por el entorno (fuego, inundación, robo, etc.), seguridad contra fallos del propio sistema (fallos del hardware, del software, etc.).
  - Asegurar la independencia de los programas y los datos, es decir, la posibilidad de modificar la estructura de la base de datos (esquema) sin necesidad de modificar los programas de las aplicaciones que manejan esos datos.
  - Mantener la integridad de los datos realizando las validaciones necesarias cuando se realicen modificaciones en la base de datos.
  - Mejorar la eficacia de acceso a los datos, en especial en el caso de consultas imprevistas.

Entre los sistemas Gestores de Bases de Datos más usados en la actualidad se encuentran MySQL, Oracle, SQL Server, PostgreSQL, etc.

A continuación se hace un análisis de tres de los SGBD más utilizados tanto en el mundo como en nuestro país y universidad:

- **MySQL:** se puede decir que es un sistema de gestión de base de datos relacional, multihilo y multiusuario. Fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca. Gracias a la colaboración de muchos usuarios, la base de datos se ha ido mejorando optimizándose en velocidad razón por la cual una de

las bases de datos más usadas en la Internet. Por su sencillez y sus características es usado por muchas personas ya que consume muy pocos recursos, es usado tanto en aplicaciones sencillas como complejas. Es utilizado también en aplicaciones Web como Drupal, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla, además sus conexiones generalmente son muy seguras.

- **Oracle (Relational Database Management System):** básicamente una herramienta cliente/servidor para la gestión de base de datos, es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hace que solo se vea en empresas muy grandes y multinacionales, por norma general. El mismo es considerado uno de los sistemas de bases de datos más completos destacándose por su:

- Soporte de transacciones.
- Estabilidad.
- Escalabilidad.
- Es multiplataforma.

Entre los aspectos que le han sido fuertemente criticados a Oracle por los especialistas están la seguridad de la plataforma y las políticas de suministro de parches de seguridad, modificadas a comienzos de 2005 y que incrementan el nivel de exposición de los usuarios. En los parches de actualización provistos durante el primer semestre de 2005 fueron corregidas 22 vulnerabilidades públicamente conocidas, algunas de ellas con una antigüedad de más de 2 años.

- **PostgreSQL:** sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) libre y bajo la licencia BSD. PostgreSQL en la actualidad está ampliamente considerado como el sistema de bases de datos de código abierto más avanzado del mundo, a esta realidad contribuye mucho el hecho de que no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales la cual trabaja en su desarrollo y perfeccionamiento, esta comunidad es denominada (PostgreSQL Global Development Group). PostgreSQL también sirve de soporte a muchos de los lenguajes más utilizados hoy día tal es el caso de PHP, C, C++, Java, Python, Ruby, etc, constituyendo una alternativa seria a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle y SQLServer.



### **1.7.1 ¿Por qué PostgreSQL?**

Para el desarrollo del sistema de gestión documental se escoge usar el SGBD, PostgreSQL debido a que gracias al estudio realizado del tema así como la recomendación realizada por otros profesionales y de personal de experiencia de la UCI se pudo constatar que la herramienta constituye la principal alternativa dentro del software libre lo cual es una de las principales premisas que se trata de seguir en la universidad, en adición a esto podríamos sumar entre otra bondades que esta base de datos no necesita ningún tipo de configuración, ni mantenimiento para los desarrolladores lo cual ahorraría significativamente tiempo de trabajo a los desarrolladores. También, y muy importante, se integra con el gestor documental propuesto, Alfresco.

Entre las innumerables ventajas que ofrece PostgreSQL que contribuyeron a su selección están las siguientes:

- Posee una gran escalabilidad. Es capaz de ajustarse al número de procesadores y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta (en algunos benchmarks se dice que ha llegado a soportar el triple de carga de lo que soporta MySQL).
- Implementa el uso de rollbacks, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz.
- Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos.
- Es Multiusuario, multiprogramado, con arquitectura cliente-servidor y control de privilegios de acceso.

### **1.8 Servidores Web.**

Un servidor Web no es más que un programa que se ejecuta continuamente en una computadora (También llamada por lo general servidor) que interpreta las peticiones HTTP (hypertext transfer protocol) que recibe por parte de un cliente (un navegador de Internet) y las satisface. Dependiendo del tipo de la petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor, devolviendo algún tipo de resultado HTML al cliente o navegador que realizó la petición.

En nuestros días, a pesar de que existen cierto número de servidores Web, el mercado de estos softwares esta prácticamente dominado por Apache. Se estima según estudios realizados en la red

mundial que alcanzó su máxima cota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios Web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años. ([www.netcraft.com](http://www.netcraft.com))

Entre las características más destacables de Apache están las siguientes:

- Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Apache es una tecnología gratuita de código fuente abierto. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto. Esto le da una transparencia a este software de manera que si queremos ver que es lo que estamos instalando como servidor, puede ser estudiado, sin ningún secreto o puerta trasera nos sorprenda.
- Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Actualmente existen muchos módulos para Apache que son adaptables a este, y están ahí para que los instalemos cuando los necesitemos. Otra cosa importante es que cualquiera que posea cierta experiencia en la programación de C o Perl puede escribir un modulo para realizar una función determinada.
- Apache trabaja con gran cantidad de Perl, PHP y otros lenguajes de script. Perl destaca en el mundo del script y Apache utiliza su parte del pastel de Perl tanto con soporte CGI como con soporte mod perl. También trabaja con Java y páginas jsp. Teniendo todo el soporte que se necesita para tener páginas dinámicas.
- Apache te permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.
- Tiene una alta configurabilidad en la creación y gestión de logs. Apache permite la creación de ficheros de log a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en tu servidor. (Ricardo J. Vargas Del Valle, Programación en Capas.)

Debido a las características anteriormente mencionadas y a que Apache es un servidor Web potente y flexible que puede funcionar una amplia variedad de plataformas y entornos fue que determinamos

que este es el servidor Web que se debe utilizar para el desarrollo del sistema. En esta decisión también influyó el amplio dominio que presentan los estudiantes de la universidad en la herramienta.

### **1.9 Framework**

Un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Son diseñados con el intento de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel de proveer un sistema funcional.

### ***Symfony***

Es un completo framework desarrollado completamente con PHP5 diseñado para optimizar el desarrollo de las aplicaciones Web, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación Web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación Web compleja, automatiza las tareas más comunes permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. (POTENCIER, 2008)

### **Características principales:**

- Fácil de instalar y configurar en la mayoría de las plataformas tanto \*nix (Unix, Linux, etc.) como en plataformas Windows.
- Es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Basado en la premisa de “convenir en vez de configurar”, en la que el desarrollador solo debe configurar aquello que no es convencional.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.

- Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa.
- Código fácil de leer que incluye comentarios y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.
- Se trata del framework PHP mejor documentado: miles de páginas en el wiki oficial, tutoriales de hasta 250 páginas y un libro gratuito de casi 500 páginas. Además, el libro está completamente traducido al español.
- Symfony sigue una política de tipo LTS (long term support). Las versiones estables se mantienen durante 3 años sin cambios pero con una continua corrección de los errores conocidos.

### ***CodeIgniter***

CodeIgniter es un poderoso framework para PHP Entorno de Desarrollo de una Aplicación - una herramienta de sistema - para gente que construye sitios web usando PHP. El objetivo es habilitar el desarrollo de proyectos mucho más rápido de lo que podría si escribiese código desde cero, a través de proveer un rico conjunto de librerías para tareas comúnmente necesarias, tanto como una simple interface y estructura lógica para acceder a estas librerías. CodeIgniter le permite concentrarse creativamente en su proyecto minimizando el volumen de código necesario para una tarea determinada. (Martínez, Ruiz Díaz, & Waisbrot, 2008)

### ***Características principales***

- Sistema Basado en Modelo-Vista-Controlador
- Compatible tanto PHP 4 como para PHP5
- Extremadamente Liviano
- Clases de base de datos llenas de características con soporte para varias plataformas.
- Formulario y Validación de Datos
- Manejo de Sesión
- Clase de Envío de Email. Soporta Archivos Adjuntos, email de texto/HTML, múltiples protocolos (sendmail, SMTP, and Mail) y más.
- Paginación
- Encriptación de Datos

- Historial de Errores
- Clase de Agente del Usuario
- URLs amigables a motores de búsqueda
- Magnífica documentación y enorme comunidad de desarrolladores
- Es relativamente sencillo de aprender y utilizar, sobre todo porque cuenta con un manual de usuario muy completo y comprensible aún para programadores no muy experimentados en PHP

### 1.9.1 ¿Por qué CodeIgniter?

Después de realizarse un estudio de algunos de los frameworks para PHP existentes, reconociendo las potencialidades de Symfony pero teniendo en cuenta las características propias del sistema tales como: que es un proyecto simple, pequeño y con pocos casos de uso, que se necesita un desempeño excepcional, un entorno de trabajo que requiere casi nula implementación del código, que evite entornos de trabajo que requieran usar línea de comando, que el paquete de librerías no sean complejas como el caso Symfony con PEAR, se decide utilizar como framework de desarrollo el CodeIgniter por las principales características del framework anteriormente mencionadas, que responden a las necesidades del sistema. Otro motivo del por qué de su selección es que ya se conoce de la experiencia en la facultad 10 de la UCI, de la implementación con resultados positivos de sistemas de gestión documental haciendo uso de este framework. La principal desventaja que posee es que hay mucha documentación en inglés, tema que se convierte en un reto para los informáticos.

### 1.10 Patrones

Un patrón es una solución a un problema en un contexto, codifica conocimiento específico acumulado por la experiencia en un dominio. Un sistema bien estructurado está lleno de patrones.

Los patrones surgen de la experiencia de seres humanos de tratar de lograr ciertos objetivos. Capturan la experiencia existente y probada para promover buenas prácticas.

De manera más simple, un *patrón es un par problema/solución* con nombre que se puede aplicar en nuevos contextos, con consejos acerca de cómo aplicarlo en nuevas situaciones y discusiones sobre sus compromisos.

### **Categorías de patrones**

Según la escala o nivel de abstracción:

- **Patrones de arquitectura:** Aquellos que expresan un esquema organizativo estructural fundamental para sistemas software.
- **Patrones de diseño:** Aquellos que expresan esquemas para definir estructuras de diseño (o sus relaciones) con las que construir sistemas software.
- **Idiomas:** Patrones de bajo nivel específicos para un lenguaje de programación o entorno concreto.

### **1.10.1 Patrones de diseño**

#### **1.10.1.1 Patrones Arquitectónicos**

##### **Modelo-Vista-Controlador (MVC)**

El MVC (Model View Controller) o Modelo Vista Controlador constituye uno de los patrones más usados para aplicaciones Web.

“La lógica de una interfaz de usuario cambia con más frecuencia que el almacenado de datos y la lógica de negocio. Si realizamos un diseño ofuscado, es decir, un pastiche que mezcle los componentes de interfaz y de negocio, entonces la consecuencia será que, cuando necesitemos cambiar el interfaz, tendremos que modificar trabajosamente los componentes de negocio. Mayor trabajo y más riesgo de error”. (Lago, Patrones de diseño software, 2007)

Se trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos. (Lago, Patrones de diseño software, 2007).

Los elementos principales del patrón son:

- **Modelo:** datos y reglas de negocio
- **Vista:** muestra la información del modelo al usuario
- **Controlador:** gestiona las entradas del usuario

### ***El modelo es el responsable de:***

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define las reglas de negocio (la funcionalidad del sistema).
- Lleva un registro de las vistas y controladores del sistema.

### ***El controlador es responsable de:***

- Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método "Actualizar ()". Una petición al modelo puede ser "Obtener\_tiempo\_de\_entrega (nueva\_orden\_de\_venta)".

### ***Las vistas son responsables de:***

- Recibir datos del modelo y mostrarlos al usuario.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- Pueden dar el servicio de "Actualización ()", para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes). (Lago, Patrones de diseño software, 2007).

### ***Patrón 3 capas***

El **patrón tres capas** se utiliza en la arquitectura dividida en capas ya que ofrece entre sus principales ventajas las que se listan a continuación:

- Aísla la lógica de la aplicación y la convierte en una capa intermedia bien definida y lógica del software.
- En la capa de presentación se realiza relativamente poco procesamiento de la aplicación.
- Simplifica la comprensión y organización del desarrollo del sistema, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle de las superiores.
- Facilita la reutilización. (Larman, 2003).

### 1.10.1.2 Patrones GRASP

Los patrones GRASP describen los principios fundamentales de diseño de objetos para la asignación de responsabilidades y de cierto modo ayudan a determinar las clases que estarán en el diseño.

A continuación se ofrece una relación de los principales patrones GRASP y el objetivo específico que cumple cada uno de ellos:

- **Experto:** Cada objeto es responsable por mantener su propia información (principio de encapsulamiento) o sea que este conoce y puede informar el valor de sus atributos así como modificarlos. Si tiene relación de agregación (fuerte) con otros objetos (sus partes), también será responsable de conocer la información de ellos, de crearlos (patrón creador) y de delegarles las operaciones.
- **Creador:** El objeto B tiene la responsabilidad de crear objetos de la clase A si:
  - B agrega objetos A.
  - B contiene objetos A.
  - B registra objetos A.
  - B usa (exhaustivamente) objetos A.
  - B posee la información necesaria para inicializar A.
- **Bajo Acoplamiento:** El acoplamiento es la medida de cuánto una clase esta conectada (tiene conocimiento) de otras clases por lo que éste patrón es evaluativo. Un bajo acoplamiento permite que el diseño de clases sea más independiente reduciendo el impacto de los cambios y aumentando la reutilización.
- **Alta Cohesión:** La cohesión funcional dentro de una clase es una medida que indica cuán relacionadas están las responsabilidades de una clase por lo que también éste es un patrón evaluativo. Entre más alta cohesión resulta más fácil de entender, cambiar y reutilizar.
- **Controlador:** Consiste en Asignar la responsabilidad de manejar los mensajes eventos del sistema a una clase facilitando centralizar las actividades (tales como validación, seguridad entre otras).



### 1. 10.1.3 Patrones GoF.

#### Patrones de creación:

- **Abstract Factory:** proporciona una interfaz para crear familias de objetos o que dependen entre sí, sin especificar sus clases concretas.
- **Builder:** separa la construcción de un objeto complejo de su representación, de forma que el mismo proceso de construcción pueda crear diferentes representaciones.
- **Factory Method:** define una interfaz para crear un objeto, pero deja que sean las subclases quienes decidan qué clase instanciar. Permite que una clase delegue en sus subclases la creación de objetos.
- **Prototype:** especifica los tipos de objetos a crear por medio de una instancia prototípica, y crear nuevos objetos copiando este prototipo.
- **Singleton:** garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella.

#### Patrones estructurales:

- **Adapter:** convierte la interfaz de una clase en otra distinta que es la que esperan los clientes. Permiten que cooperen clases que de otra manera no podrían por tener interfaces incompatibles.
- **Bridge:** desvincula una abstracción de su implementación, de manera que ambas puedan variar de forma independiente.
- **Composite:** combina objetos en estructuras de árbol para representar jerarquías de parte-todo. Permite que los clientes traten de manera uniforme a los objetos individuales y a los compuestos.
- **Decorator:** añade dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia para extender la funcionalidad.
- **Facade:** proporciona una interfaz unificada para un conjunto de interfaces de un subsistema. Define una interfaz de alto nivel que hace que el subsistema sea más fácil de usar.
- **Flyweight:** usa el compartimiento para permitir un gran número de objetos de grano fino de forma eficiente.
- **Proxy:** proporciona un sustituto o representante de otro objeto para controlar el acceso a éste.

### Patrones de comportamiento

- **Chain of Responsibility:** evita acoplar el emisor de una petición a su receptor, al dar a más de un objeto la posibilidad de responder a la petición. Crea una cadena con los objetos receptores y pasa la petición a través de la cadena hasta que esta sea tratada por algún objeto.
- **Command:** encapsula una petición en un objeto, permitiendo así parametrizar a los clientes con distintas peticiones, encolar o llevar un registro de las peticiones y poder deshacer la operaciones.
- **Interpreter:** dado un lenguaje, define una representación de su gramática junto con un intérprete que usa dicha representación para interpretar las sentencias del lenguaje.
- **Iterator:** proporciona un modo de acceder secuencialmente a los elementos de un objeto agregado sin exponer su representación interna.
- **Mediator:** define un objeto que encapsula cómo interactúan un conjunto de objetos. Promueve un bajo acoplamiento al evitar que los objetos se refieran unos a otros explícitamente, y permite variar la interacción entre ellos de forma independiente.
- **Memento:** representa y externaliza el estado interno de un objeto sin violar la encapsulación, de forma que éste puede volver a dicho estado más tarde.
- **Observer:** define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambia de estado se notifica y actualizan automáticamente todos los objetos.
- **State:** permite que un objeto modifique su comportamiento cada vez que cambia su estado interno. Parecerá que cambia la clase del objeto.
- **Strategy:** define una familia de algoritmos, encapsula uno de ellos y los hace intercambiables. Permite que un algoritmo varíe independientemente de los clientes que lo usan.
- **Template Method:** define en una operación el esqueleto de un algoritmo, delegando en las subclases algunos de sus pasos. Permite que las subclases redefinan ciertos pasos del algoritmo sin cambiar su estructura.
- **Visitor:** representa una operación sobre los elementos de una estructura de objetos. Permite definir una nueva operación sin cambiar las clases de los elementos sobre los que opera. (Gracia, 2005).

## 1.10.2 Patrones de casos de uso.

- **Patrón CRUD** (Creating, Reading, Updating and Deleting): Este patrón se basa en la fusión de casos de uso simples para formar una unidad conceptual.
  - *Completo*: Este patrón consta de un caso de uso, llamado Información CRUD o Gestionar información, modela todas las operaciones que pueden ser realizadas sobre una parte de la información de un tipo específico, tales como creación, lectura, actualización y eliminación. Suele ser utilizado cuando todos los flujos contribuyen al mismo valor del negocio, y estos a su vez son cortos y simples.
  - *Parcial*: Este patrón alternativo modela una de las vías de los casos de uso como un caso de uso separado. Es preferiblemente utilizado cuando una de las alternativas de los casos de uso es más significativa, larga o más compleja que las otras.
- **Actores Múltiples**:
  - *Rol Común (Multiple Actors: Common Role)*: dos actores juegan el mismo papel hacia el caso de uso. Este rol es representado por otro actor, heredado por los actores que comparten este rol. Este patrón es aplicable cuando, desde del punto de vista de un caso de uso hay solo una entidad externa interactuando con cada instancia del caso de uso.
  - *Roles diferentes (Multiple Actors: Different Role)*: Captura la concordancia entre actores manteniendo roles separados. Consiste de un caso de uso y por lo menos dos actores. Es utilizado cuando dos actores juegan diferentes roles en un caso de uso, o sea, interactúan de forma diferente con el caso de uso.
- **Concrete Inclusión or Extensión**: el objetivo principal de este patrón es modelar flujos como parte de un caso de uso base y otra parte en otro caso de uso que complete al caso de uso base, es un patrón estructural.
  1. *Inclusión*: en este patrón hay una relación de inclusión del caso de uso base al caso de uso incluido, el cual puede ser instanciado por si mismo, este patrón puede ser usado cuando un flujo puede ser incluido en flujo de otro caso de uso.
  2. *Exclusión*: consiste en la extensión de un caso de uso a otro caso de uso base. El caso de uso extendido es concreto, o sea, puede estar instanciado por si mismo tanto como el

caso de uso base. Este patrón puede ser usado cuando un flujo puede extender el flujo de otro caso de uso.

- **Concordancia** (*Commonality*): Extrae una subsecuencia de acciones que aparecen en diferentes lugares del flujo de casos de uso y es expresado por separado.
  - *Reusabilidad*: Consta de 3 casos de uso. El primero llamado subsecuencia común, modela una secuencia de acciones que aparecerán en múltiples casos de uso en el modelo. Los otros casos de uso modelan el uso del sistema que comparte la subsecuencia común de acciones. De manera que deben existir al menos dos de ellos.
  - *Adición*: En el caso de este patrón alternativo, la subsecuencia común de casos de uso, extiende los casos de uso compartiendo la subsecuencia de acciones. Los otros casos de uso modelan el flujo que será expandido con la subsecuencia. Este patrón es preferible usarlo cuando otros casos de uso se encuentran propiamente completos, o sea, que no requieren de una subsecuencia común de acciones para modelar los usos completos del sistema.
  - *Especialización*: Otro patrón de concordancia que contiene casos de uso del mismo tipo. En este caso, estos son modelados como una especialización de casos de uso de tipo de uso común. Todas las acciones en estos casos de uso son heredadas por los casos de uso hijos, donde otras acciones serán adicionadas o acciones heredadas que serán especializadas. Este patrón es aplicable cuando la utilización de los casos de uso que han sido modelados son del mismo tipo, y este tipo debe hacerse visible en el modelo.

### 1.11 Requerimientos

Los requerimientos son uno de los aspectos más importantes en el proceso de desarrollo de software, debido a que reflejan las necesidades que debe cumplir el sistema.

La IEEE Standard Glossary of Software Engineering Terminology define un requerimiento como:

1. Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.

2. Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.
3. Una representación documentada de una condición o capacidad como en 1 o 2.

Los requisitos se pueden clasificar en: **funcionales** (Son capacidades o condiciones que el sistema debe cumplir) y **no funcionales** (Son propiedades o cualidades que el producto debe tener). (Anónimo, 2007-2008)

### ***Técnicas empleadas en el levantamiento de requisitos.***

La captura de requisitos es la actividad a través de la cual el equipo de desarrollo de un sistema de software obtiene, de cualquier fuente de información disponible, las necesidades que debe cumplir dicho sistema. Por la complejidad que todo este proceso puede implicar, la ingeniería de requisitos define técnicas que permiten hacer este proceso de una forma más eficiente y precisa. Estas técnicas son:

- Entrevistas
- JAD
- Tormenta de ideas
- Casos de Uso
- Cuestionarios
- Comparación de Terminología
- Estudio de otros sistemas

### **Entrevistas**

Resulta una técnica muy aceptada y de un uso ampliamente extendido dentro de la ingeniería de requisitos. Las entrevistas le permiten al analista obtener conocimiento del problema y comprender los objetivos de la solución esperada. A través de esta técnica el equipo de trabajo se acerca al problema de una forma natural. Existen muchos tipos de entrevistas. Básicamente, la estructura de la entrevista abarca tres pasos: identificación de los entrevistados, preparación de la entrevista, realización de la entrevista y documentación de los resultados.

La entrevista no es una técnica sencilla de aplicar, requiere que el entrevistador sea experimentado y

tenga capacidad para elegir bien a los entrevistados y obtener de ellos toda la información posible en un período de tiempo siempre limitado. Es por ello que la preparación de la entrevista juega un papel fundamental. (José Escalona & Koch, 2002).

### **JAD**

Esta técnica se realiza como una alternativa a las entrevistas. Es una práctica de grupo que se desarrolla durante varios días y en la que participan analistas, usuarios, administradores del sistema y clientes. Está basada en cuatro principios fundamentales: dinámica de grupo, el uso de ayudas visuales para mejorar la comunicación, mantener un proceso organizado y racional y una filosofía de documentación WYSIWYG (What You See Is What You Get, lo que ve es lo que obtiene), es decir, durante la aplicación de la técnica se trabajará sobre lo que se generará. En cada una de las sesiones que se realizan se establecen los requisitos de alto nivel a trabajar, el ámbito del problema y la documentación, llegándose a una serie de conclusiones que se documentan. En cada sesión se van concretando más las necesidades del sistema.

A la hora de ver las ventajas que presenta esta técnica frente a las entrevistas que es la que por tradición más se utiliza, vemos que se ahorra tiempo al evitar que a los clientes se les tenga que contrastar por separados, sin embargo se requiere de un grupo de participantes bien integrados y organizados.

### **Tormenta de ideas**

Es también una técnica de reuniones en grupo, cuyo objetivo es que los participantes muestren sus ideas de forma libre. Esto no es más que la acumulación de ideas y/o información sin evaluar las mismas. Como técnica de captura de requisitos es sencilla de usar y de aplicar, contrariamente al JAD, puesto que no requiere tanto trabajo en grupo como este. Tiene como desventaja que no nos sirve para obtener de ellos detalles concretos ya que estos se suelen usar solo en los primeros encuentros, a pesar de ofrecer una visión general de las necesidades del sistema.

### **Casos de uso**

Esta técnica aunque inicialmente se desarrollaron como técnica para la definición de requisitos algunos autores proponen casos de uso como técnica para la captura de requisitos. Los casos de uso permiten mostrar el contorno (actores) y el alcance (requisitos funcionales expresados como casos de uso) de un sistema. Un caso de uso describe la secuencia de interacciones que se producen entre el sistema y los actores del mismo para realizar una determinada función.

La ventaja esencial de los casos de uso es que resultan muy fáciles de entender para el usuario o cliente, sin embargo, carecen de la precisión necesaria si no se acompañan con una información textual o detallada con otra técnica, como pueden ser los diagramas de actividades. (José Escalona & Koch, 2002).

### **Cuestionarios**

Esta técnica requiere que el analista conozca el ámbito del problema en el que está trabajando. Consiste en redactar un documento con preguntas cuyas respuestas sean cortas y concretas, o incluso cerradas por unas cuantas opciones en el propio cuestionario (Checklist). Este cuestionario será cumplimentado por el grupo de personas entrevistadas o simplemente para recoger información en forma independiente de una entrevista. (José Escalona & Koch, 2002).

### **Comparación de Terminología**

Uno de los problemas que surge durante la elicitación de requisitos es que usuarios y expertos no llegan a entenderse debido a problemas de terminología. Esta técnica es utilizada en forma complementaria a otras técnicas para obtener un consenso respecto de la terminología a ser usada en el proyecto de desarrollo. Para ello es necesario identificar el uso de términos diferentes para los mismos conceptos (correspondencia), misma terminología para diferentes conceptos (conflictos) o cuando no haya concordancia exacta ni en el vocabulario ni en los conceptos (contraste). (José Escalona, y otros, 2002).

Las técnicas descritas anteriormente representan las más utilizadas, pero existen otras como el análisis de otros sistemas y el estudio de la documentación que también pueden ser aplicadas para el desarrollo del presente trabajo de diploma.

### **1.12 Flujo de Análisis y Diseño en RUP**

El Análisis y Diseño es uno de los 9 flujos de trabajo de la metodología RUP y su objetivo fundamental es traducir los requisitos a una especificación que describe cómo implementar el sistema. El análisis consiste en obtener una visión del sistema que se preocupa de ver QUÉ hace, de modo que sólo se interesa por los requisitos funcionales. Por otro lado, el diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva CÓMO cumple el sistema sus objetivos. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades. De hecho, cuando la precisión del diseño es muy grande, la

implementación puede ser hecha por un generador automático de código.

Para este flujo se establecen como **roles**: Diseñador, Analista, Analista de sistemas y dentro de los **artefactos** están: Documento de arquitectura de software, Modelo de despliegue, Modelo de análisis, Modelo de diseño, Prototipo de interfaz de usuario, Modelo de datos y Modelo de análisis.

De todos estos artefactos serán desarrollados como parte de la propuesta de solución el modelo de análisis y el modelo de diseño.

### **Modelo de Análisis**

El modelo de análisis es utilizado fundamentalmente por los desarrolladores para comprender cómo debería darse forma al sistema, es decir, como debería ser diseñado e implementado. Este modelo sirve como una primera aproximación al diseño. (Rumbaugh, y otros, 2004).

### **Modelo de Diseño**

El Modelo de Diseño es un modelo de objetos que describe la realización física de los casos de uso, sirve como una abstracción del modelo de implementación y su código fuente. El modelo de diseño es utilizado como una entrada esencial a las actividades de implementación y pruebas. (Rational Software Corporation, 2003).

### **1.13 Conclusiones**

Con el desarrollo de este capítulo, donde se realizó un estudio del arte de los principales conceptos relacionados con el tema de gestión documental, así como de las tendencias y tecnologías actuales en el proceso de desarrollo de software y de gestión de documentos digitales. Se seleccionaron las herramientas y tecnologías a utilizar en el proceso de desarrollo de software y en el diseño del sistema de gestión de documentos digitales, haciendo una fundamentación de por qué su elección.



### Capítulo II Características del Sistema

#### **2.1 Introducción**

En este capítulo se describe la situación actual del grupo de Arquitectura y Estándares de Información. Se modela el diseño del sistema de gestión documental y el modelo de negocio, incluyendo actores, trabajadores y la descripción de los casos de uso del negocio que corresponden a las principales funciones que se realizan en el grupo. Además, se hace una propuesta del sistema a desarrollar, se describen sus principales características y se especifican los requisitos funcionales y no funcionales, así como los casos de uso del sistema y su descripción.

#### **2.2 Flujo actual de los procesos.**

El Grupo de A&EI realiza diferentes funciones como: Servicios de consultoría y asesoría a los proyectos de Investigación y Desarrollo (I+D), Formación del rol Arquitecto de Información, Diseño de procedimientos para la aplicación de la AI en el proceso de desarrollo de software, entre otras, en las que intervienen tanto los Especialistas del grupo, el Jefe de Grupo y el Director de la Dirección Técnica. Actualmente existen problemas con el control y organización de la documentación que estas funciones generan, muchas veces la información está repetida, no clasificada o no disponible por lo que se hace un poco engorroso tener acceso a ella, saber con qué versión del documento se está trabajando y esto trae consigo ineficiencia en los resultados del grupo.

#### **2.3 Objeto de automatización**

Con el propósito de solucionar los problemas que se están presentando en el Grupo de A&EI se propone realizar una aplicación web que realice la función de un sistema que permita la gestión de los documentos digitales que se generan como resultado de las funciones del grupo.

Durante el ciclo de desarrollo de esta actividad existen varios procesos que deben ser automatizados, puesto que su ejecución actualmente resulta tediosa y propensa a errores, además de consumir una valiosa porción de tiempo de los especialistas. Muchos de estos procesos son de vital importancia para lograr un nivel de calidad óptimo en los productos finales.

Se automatiza el proceso de ciclo de vida del documento teniendo en cuenta:

- Creación de plantillas definidas según tipos documentales.
- Creación de documentos según la plantilla definida para este tipo de documento.
- Registro de las versiones de cada documento que se genera.
- Registro / control de los usuarios que inciden en el ciclo de vida de cada documento.
- La organización y almacenamiento de los documentos una vez creados y aprobados.

### **2.4 Modelado del Sistema de Gestión Documental**

La necesidad de controlar y organizar la documentación que se genera en el grupo de A&EI es la razón por la que se hace imprescindible diseñar un sistema de gestión de documentos digitales que facilite la organización de los documentos que generan las funciones que se llevan a cabo en el grupo.

Para hacer un diseño lo más aproximado posible al SGDD que daría solución a las necesidades del grupo de A&EI se deben tener en cuenta algunos aspectos "... para la apreciación de un documento cualquiera es esencial saber exactamente quién lo produjo, en qué circunstancias, dentro de qué marco de procedimiento, con qué fin, con destinación a quién, cuándo y cómo fue recibido por su destinatario, y porqué vías llegó hasta nosotros. Bien: tal conocimiento no es posible sino en la medida en que el conjunto de documentos que lo acompañan se haya conservado exacto..." (Duchein, 1985). Por eso el principio de respeto a la procedencia de los documentos y orden natural es un aspecto básico a tener en cuenta para lograr una óptima clasificación de los documentos y por tanto una óptima organización.

Dentro de las agrupaciones documentales reconocidas por la archivística española que conforman una estructura jerárquica se encuentran: las unidades documentales (simples y complejas), las series documentales, las secciones y el fondo documental.

El **fondo documental**: es la colección de documentos, organizados en una categoría que especifica el orden en el que provienen.

Las **secciones**: se crean teniendo en cuenta la organización, grupo, institución que realiza las funciones que generan los documentos.

La **serie documental**: se conoce como el conjunto de documentos que responden a un mismo tipo documental producido por un mismo órgano en el ejercicio de una función determinada dentro de la institución. (Duplá del Moral, 1997).

Cada serie entonces está formada a su vez por una unidad documental (o documento) que puede ser simple (formado por un solo documento, Ej: carta) o compleja (conjunto de documentos, Ej: expediente).



Figura 2.1: Agrupaciones documentales.

**2.4.1 Descripción de los procesos de negocio.**

Para describir los procesos del negocio que se llevan a cabo en el Grupo de A&EI, se desarrolla una tabla que relaciona el origen funcional de de la documentación generada en el grupo.

Funciones	Norma que regula la función	Reflejo documental de las funciones (Serie documental)
Controlar y organizar las actividades que desempeñan los especialistas del grupo.	Resolución rectoral que respalda el nombramiento de directivos en la universidad.	Planes de trabajo. Autoevaluación Evaluaciones de desempeño Actas de reuniones
Asesorar y dar consultoría a los proyectos de I+D relacionadas con AI.	Resolución rectoral de respalda la toma de decisiones de los directivos en la UCI.	Informes técnicos de revisión de los proyectos Normativas de trabajo.
Formar al Arquitecto de Información.	Procedimiento para el proceso de Arquitectura de Información.	Estrategia de formación y capacitación Planes de formación y capacitación,

	Programa analítico para diseño de cursos (P1).	
Gestionar los expertos para impartir cursos de capacitación.	Estrategia para la gestión de expertos de la Dirección Técnica	Solicitud de expertos (Planilla de solicitud, informe curricular y carta de solicitud).
Elaborar los cursos que se van a impartir.	Programa analítico para diseño de cursos (P1).	P1, Programa analítico de la asignatura.
Definir normativas de AI para los diferentes proyectos productivos.	IPP1000_2008: Elaboración y aprobación de procedimientos y lineamientos.	Procedimientos Lineamientos

### **2.4.2. Tratamiento archivístico: Descripción de los documentos.**

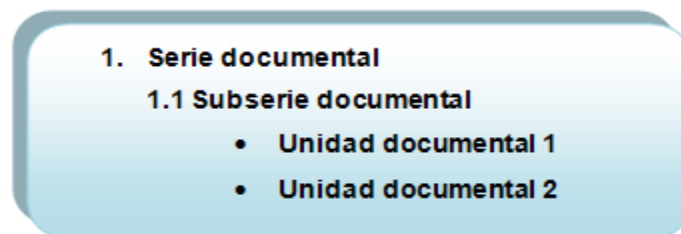
El sistema que se propone tiene como alcance la documentación que se genera en el grupo de A&EI, por lo que se considera una sección dentro del diseño del sistema de gestión documental de la Universidad de Ciencias Informáticas. Por ello, fue necesario consultar el diseño documental de dicho sistema, para respetar las normas de tratamiento archivístico que se utilizan en el mismo, con una visión futura de poder integrar el sistema del grupo al sistema de la universidad. Dichas normas para la descripción archivística son: ISAD(G), ISAAR(CPF) y la EAD por estar orientada a formatos electrónicos. Para la organización funcional en la clasificación de la documentación se utilizaron las técnicas de Roberch, y como metodologías para el diseño e implementación de sistemas la ISO 15489 y la DIRKS del 2001. Para la transferencia de documentos se deberá regir el sistema por resoluciones rectorales que respalden dicha actividad.

### **2.4.3 Cuadro de Clasificación**

Para la implantación del sistema se debe identificar **el cuadro de clasificación**, que representa las funciones que se llevan a cabo en la organización en consecuencia con las unidades documentales que estas generan organizadas jerárquicamente en secciones y series.

Por tanto, se define como sección del fondo documental a Dirección Técnica y como sub-sección a A&EI. Es por ello que el cuadro está estructurado a partir de una Sección y por sub-sección, esta última asociada a las funciones que desarrolla el grupo.

El cuadro de clasificación quedaría entonces estructurado como se representa en la siguiente figura:



**Figura 2.2. Representación del cuadro de clasificación de manera general**

### **Cuadro de Clasificación de la Documentación del grupo de A&EI de la Dirección Técnica de la IP.**

1. Normativas de trabajo
  - 1.1 Procedimientos
    - Normativas
    - Procedimientos
    - Lineamientos
2. Capacitación
  - 2.1 Estrategia de capacitación
    - Estrategia y plan de capacitación
  - 2.2 Cursos de capacitación
    - P1 Programa analítico
3. Gestión de expertos
  - 3.1 Solicitudes
    - Planilla de solicitud
    - Informe curricular del experto
    - Carta de solicitud al comité de compra
4. Reuniones
  - 4.1 Reuniones ordinarias

- Actas de reuniones
5. Evaluación de los especialistas del grupo
    - 5.1 Evaluación del desempeño
      - Acta de Autoevaluación individual
      - Acta de Evaluación del desempeño
  6. Planes de trabajo
    - 6.1 Trimestral
      - Plan de trabajo trimestral
    - 6.2 Semestral
      - Plan de trabajo semestral
    - 6.3 Anual
      - Plan de trabajo anual
      -

### ***Workflow del documento (o flujo de Trabajo del documento)***

Se define como **Workflow del documento** al conjunto de reglas establecidas para el tratamiento de los documentos a través de su ciclo de vida.

El ciclo de vida del documento inicia con su creación. Para facilitar el trabajo de creación de los documentos, se establecieron **contenidos** (plantillas o tipos documentales) que definen un formato único según el tipo documental del que se necesite crear el documento.

Una vez creado el documento, este tiene que ser aprobado y validado por el personal calificado para ello, (en este caso el Jefe de Grupo y el Director Técnico) y se publica finalmente en el sistema en formato .pdf. Finaliza, este ciclo de vida del documento, cuando el sistema permite que el documento pueda ser recuperado, eliminado o que se conserve el tiempo y en el lugar establecidos. El flujo de trabajo anteriormente descrito se refleja en la figura 2.3.



Figura 2.3. Workflow del proceso de gestión del documento en el sistema.

También se debe tener en cuenta en el diseño del workflow del documento que se propone para el SGDD, la asignación de permisos a usuarios según el rol que este desempeña, así se controla qué usuario realiza las acciones que intervienen en el ciclo de vida del documento. La siguiente tabla hace referencia a esto:

Actor	Permisos				
	Leer	Editar Contenido	Crear	Modificar propiedades	Publicar
Jefe de Grupo	x		x	x	
Dirección Técnica	x				x
Especialistas G	x	x	x		
Administrador	x	x	x	x	

### **Metadatos**

Los metadatos son datos altamente estructurados que describen información, describen el contenido, la calidad, la condición y otras características de los datos. Estos juegan un papel importante dentro de la información ya que permiten organizar y mantener el conjunto de datos de una organización, proporcionar información para catálogos de datos y centros de distribución de metadatos y proveer información necesaria para interpretar y procesar datos transferidos por otra organización.

Algunos de los metadatos definidos según el tipo documental (contenido o plantilla) son:

- Metadatos para Evaluación de desempeño:
  - Período de Evaluación
  - Nombre del evaluado
  - Fecha de Evaluación
  - Número del Solapín

- Carnet Identidad
- Metadatos para Autoevaluación
  - Nombre y Apellidos
  - Período de Evaluación
- Metadatos para Currículum
  - Nombre y Apellidos
  - E-mail
- Metadatos para Plan de Trabajo
  - Nombre
  - Fecha
- Metadatos para Programa Analítico
  - Confeccionado por
  - Año de confección
- Metadatos para la Solicitud de cursos en el extranjero
  - Curso
  - Nombre y Apellidos del que lo recibió

### **2.5 Modelado del Negocio.**

Un modelo de negocio (también llamado diseño de negocio) es el mecanismo por el cual un negocio trata de generar ingresos y beneficios. Por tanto, la finalidad del modelado del negocio es describir cada proceso del negocio, especificando sus datos, actividades (o tareas) y roles (o agentes).

- **Modelo del negocio:** Describe los procesos de un negocio y su interacción con elementos internos, es decir, describe las funciones que este realiza. Comprende la descripción de los actores, trabajadores y casos de uso que intervienen.
- **Actor del negocio:** Es el rol que juega algo o alguien que interactúa con el negocio para beneficiarse de sus resultados.
- **Trabajadores del negocio:** Son los que actúan directamente dentro del negocio.
- **Casos de uso del negocio:** Son los procesos que se llevan a cabo en el negocio.
- **Las entidades del negocio:** Representan a los objetos que los trabajadores manipulan o inspeccionan, producen o utilizan durante la realización de los casos de uso del negocio.



- **Diagrama de actividad:** Es un tipo especial de diagrama de estados que muestra el flujo de actividades dentro del negocio, donde se representan los actores, trabajadores y entidades del negocio, así como las actividades realizadas por ellos y sus relaciones.
- **Modelo de objetos del negocio:** Este modelo describe cómo colaboran los trabajadores y las entidades del negocio dentro del flujo de trabajo.
- **Reglas del negocio:** Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio.

### 2.5.1 Justificación de Actores y Trabajadores del Negocio.

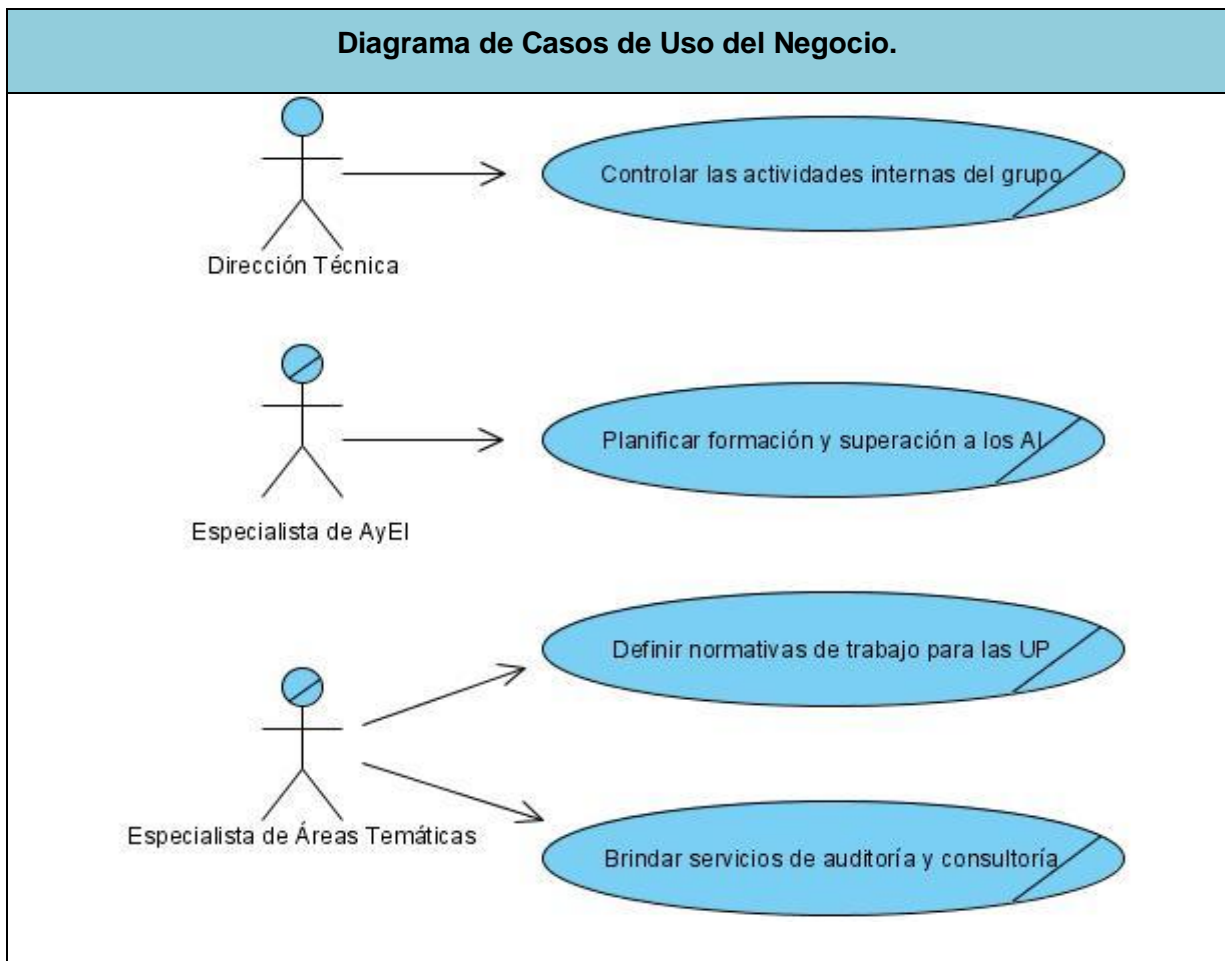
Actores	Justificaciones
<b>Dirección Técnica</b>	Es el responsable de organizar, orientar y controlar las tareas particulares por cada especialista en correspondencia a las funciones del grupo, obteniendo las mejoras que implican una adecuada organización y control del trabajo.
<b>Especialista de áreas temáticas</b>	Persona que logra un mejor control del trabajo ya que tiene que cumplir con las exigencias de las auditorías y consultorías dentro del proyecto.
<b>Especialista de A&amp;EI</b>	Persona que se beneficia con la capacitación en diferentes temas.

En este proceso de desarrollo de software se va a hacer la salvedad de que el mismo actor del negocio puede comportarse como trabajador del negocio; ya que en algunos casos se beneficia de la actividad y en otros casos es quien las realiza.

Trabajadores del Negocio	Justificación
<b>Director Técnico</b>	Persona que controla el funcionamiento del grupo de A&EI.
<b>Jefe Grupo</b>	Es el responsable de organizar, orientar y controlar las tareas particulares por cada especialista en correspondencia a las funciones del grupo.
<b>Especialistas del</b>	Son las personas que ejecutan tareas concretas para el cumplimiento de

<b>grupo</b>	las diferentes funciones y objetivos.
<b>Dirección de Posgrado</b>	Son las personas encargadas de autorizar los cursos de capacitación al personal.

**2.5.2 Diagrama de Casos de Uso del Negocio.**



**Figura 2.4 Diagrama de Casos de Uso del Negocio**

**2.5.3 Descripción de los Casos de Uso del Negocio.**

<b>Nombre del CU</b>	<b>Controlar las actividades internas del grupo</b>
<b>Actor</b>	Dirección Técnica.
<b>Trabajadores</b>	Especialistas del grupo, Jefe de Grupo
<b>Propósito</b>	Organizar y controlar las actividades del grupo para lograr un óptimo funcionamiento.
<p><b>Resumen:</b> El CU inicia cuando de la Dirección Técnica se chequean con el Jefe de Grupo las actividades generales orientadas para el grupo. El Jefe de Grupo solicita el Plan de Trabajo al especialista del grupo, los especialistas diseñan sus planes de trabajo y el Jefe de Grupo le da una evaluación teniendo en cuenta el cumplimiento de las actividades recogidas en el Plan de Trabajo. El CU termina con la entrega que hace el Jefe de Grupo a la Dirección Técnica del informe del cumplimiento de las actividades.</p>	
<b>Precondiciones</b>	Tener previamente definido el Plan de Trabajo.
<b>Flujo normal de los eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del Negocio</b>
1. Chequea cumplimiento de las actividades.	3. El Jefe de Grupo pide Plan de Trabajo individual.
2. Pide informe de valoración de cumplimiento.	
	4. El especialista elabora y entrega Plan de Trabajo
	5. El Jefe de Grupo chequea el cumplimiento de las actividades en el Plan de Trabajo <b>Ver Flujo Alterno</b> (Si no se cumplió con la tarea asignada)

	6. El Jefe de Grupo emite evaluación según los resultados de las tareas ejecutadas.
	7. El Jefe de Grupo registra evaluación en la planilla de evaluación de desempeño  8.El especialista del grupo redacta acta de reunión
	9. Se le entrega a la Dirección Técnica el informe final de evaluación de las tareas.
10. Recibe el informe	
<b>Flujo alterno de eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del Negocio</b>  5.1 El Jefe decide la afectación del incumplimiento del Plan de Trabajo y lo deja explícito en la planilla de evaluación de desempeño.
<b>Poscondiciones:</b>	Se generan documentos como: Actas de reuniones, Plan de Trabajo, Autoevaluación y Evaluaciones de desempeño.
<b>Prioridad:</b>	Representa uno de las principales actividades del grupo. (crítico)

<b>Nombre del CU</b>	<b>Planificar formación y superación para los AI</b>
<b>Actor</b>	Especialista de A&EI.
<b>Trabajadores</b>	Dirección de Posgrado, Jefe de Grupo.
<b>Propósito</b>	Que todo el personal que juegue el rol de Arquitecto de Información tenga al menos la formación básica para ejecutar todas las tareas y

	a su vez reciban una superación constante en todos los temas de la disciplina de AI.
<b>Resumen:</b> El CU inicia cuando los especialistas del grupo piden que se le impartan cursos, el Jefe de Grupo hace la solicitud al encargado de la Dirección de Posgrado, este último pide los datos para la inscripción del curso, los especialistas le envían los datos y decide si se puede impartir el curso o no, terminando así el CU.	
<b>Precondiciones</b>	Que exista el personal calificado para impartir los cursos.
<b>Flujo normal de los eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del Negocio</b>
1 Solicita al Jefe de Grupo que se impartan cursos de interés para la formación y superación.	2. El Jefe de Grupo pide a los especialistas los datos para la inscripción del curso.
3. Se redactan informes como Plan de Capacitación, P1 del curso, Programa analítico, entre otros para enviarlos al Jefe de Grupo.	4. El Jefe de Grupo los revisa y envía a Dirección de Posgrado.  5. Dirección de Posgrado analiza los informes y decide si se puede impartir el curso. (Se crean las condiciones por ejemplo: se buscan a los profesores especializados en el tema, se confecciona el cronograma y todo lo necesario para garantizar el aseguramiento logístico.)  <b>Ver Flujo Alterno</b> (Si no hay condiciones para que se imparta el curso)
6. Reciben aprobación de la solicitud.	
<b>Flujo alterno de eventos</b>	

<b>Acción del actor</b>	<b>Respuesta del Negocio</b>
	5.1 Rechaza la solicitud
<b>Poscondiciones:</b>	Se crean documentos como: Estrategias de formación y capacitación, Planes de formación y capacitación, Solicitud de expertos (planilla de solicitud, informe curricular), elaboración de los cursos (P1, Programa analítico).
<b>Prioridad:</b>	Representa uno de las principales actividades del grupo. (crítico)

<b>Nombre del CU</b>	<b>Definir normativas de trabajo para las UP</b>
<b>Actores</b>	Especialista de áreas temáticas.
<b>Trabajadores</b>	Especialistas del grupo, Jefe de Grupo.
<b>Propósito</b>	Establecer estilos de trabajo que garanticen un cumplimiento efectivo del proceso de AI en el desarrollo de los proyectos.
<b>Resumen:</b> El CU se inicia cuando se realiza una auditoría a una UP y se detectan problemas, haciendo los especialistas un estudio para ver si hay que establecer o no una normativa de trabajo, terminando así el CU.	
<b>Precondiciones</b>	Que exista la necesidad de organizar flujos de trabajo.
<b>Flujo normal de los eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del Negocio</b>
1 Recibe una auditoría.	2 El especialista del grupo estudia situación actual del proyecto.

	<p>3 Identifica problemas.</p> <p>4. Establece normativa de trabajo (procedimientos y lineamientos).  <b>Ver Flujo Alterno1</b> (Si se detectan problemas en menos del 60% de los proyectos auditados)</p> <p>5. El Jefe de Grupo la revisa y aprueba.</p> <p><b>Ver Flujo Alterno 2</b> (Si no se aprueba la normativa por algún error.)</p>
6. Se aplica normativa de AI en el proyecto.	7. Se valida la correcta aplicación de la normativa establecida.
<b>Flujo alternativo de eventos 1</b>	
<b>Acción del actor</b>	<b>Respuesta del Negocio</b>
	4.1. Se capacita al personal en el área temática que presente problemas el proyecto. Se realiza otra auditoría para comprobar si se erradicó el problema identificado en la anterior.
<b>Flujo alternativo de eventos 2</b>	
	5.1 El Jefe de Grupo se la envía al especialista para que la revise. Continúa flujo normal de eventos.
<b>Poscondiciones:</b>	Se crea el documento Normativa de trabajo, Procedimiento, entre otros.
<b>Prioridad:</b>	Representa una de las principales actividades del grupo. (crítico)

<b>Nombre del CU</b>	Brindar servicios de consultoría y asesoría.
<b>Actores</b>	Especialista de áreas temáticas.
<b>Trabajadores</b>	Especialistas del grupo, Jefe de Grupo
<b>Propósito</b>	Aportar al mejoramiento continuo en el desarrollo de los proyectos de software, esto implica que independiente a estar definidas normativas de trabajo, los equipos de proyectos tengan un tratamiento diferenciado en el tema de AI.
<b>Resumen:</b> El CU inicia cuando el especialista de áreas temáticas solicita servicios de consultoría y asesoría al Jefe de Grupo de A&EI. Este asigna un especialista del grupo para que realice la tarea y este la ejecuta. El CU termina cuando el especialista del grupo emite el Informe de Revisión Técnica donde se da una valoración del estado actual del proyecto.	
<b>Precondiciones</b>	Que exista el personal capacitado y que estén definidas normativas de trabajo para AI.
<b>Flujo normal de los eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del Negocio</b>
1. Solicita servicio de consultoría y asesoría.	2. Evalúa solicitud y envía respuesta positiva (aprueba solicitud). <b>Ver Flujo Alterno</b> (Si no se aprueba la solicitud).
3. Recibe respuesta y se prepara para la consultoría y asesoría.	4. Define cronograma y asigna especialistas.
	5. Especialista ejecuta el servicio.
	6. Redacta informe y se lo envía al Jefe de Grupo para que este lo apruebe.
	7. El Jefe de Grupo pide aval del trabajo realizado por el



	especialista del grupo al responsable del proyecto.
8. Avala el trabajo del especialista.	9. El Jefe Grupo registra el aval para la evaluación del desempeño.
<b>Flujo alterno de eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del Negocio</b>
	2.1 Evalúa solicitud y no la aprueba por no tener personal calificado.  2.2 Envía rechazo de solicitud.
<b>Poscondiciones:</b>	Se crean documentos como: Informes técnicos de revisión de los proyectos, Normativas de trabajo, Evaluación del desempeño, entre otros.
<b>Prioridad:</b>	Representa uno de las principales actividades del grupo. (crítico)

#### 2.5.4 Diagrama de Actividades

Un diagrama de actividades es un diagrama diseñado para mostrar una visión simplificada de lo que ocurre durante una operación o un proceso. Es una extensión de un diagrama de estados, pero el diagrama de actividades resalta precisamente a las actividades. A cada actividad se le representa con un rectángulo con las esquinas redondeadas, esta dividido en calles y muestra como se utilizan las entidades que apoyan la realización de los procesos de negocio **(Ver Anexo 2)**.

### 2.5.5 Modelo de Objeto

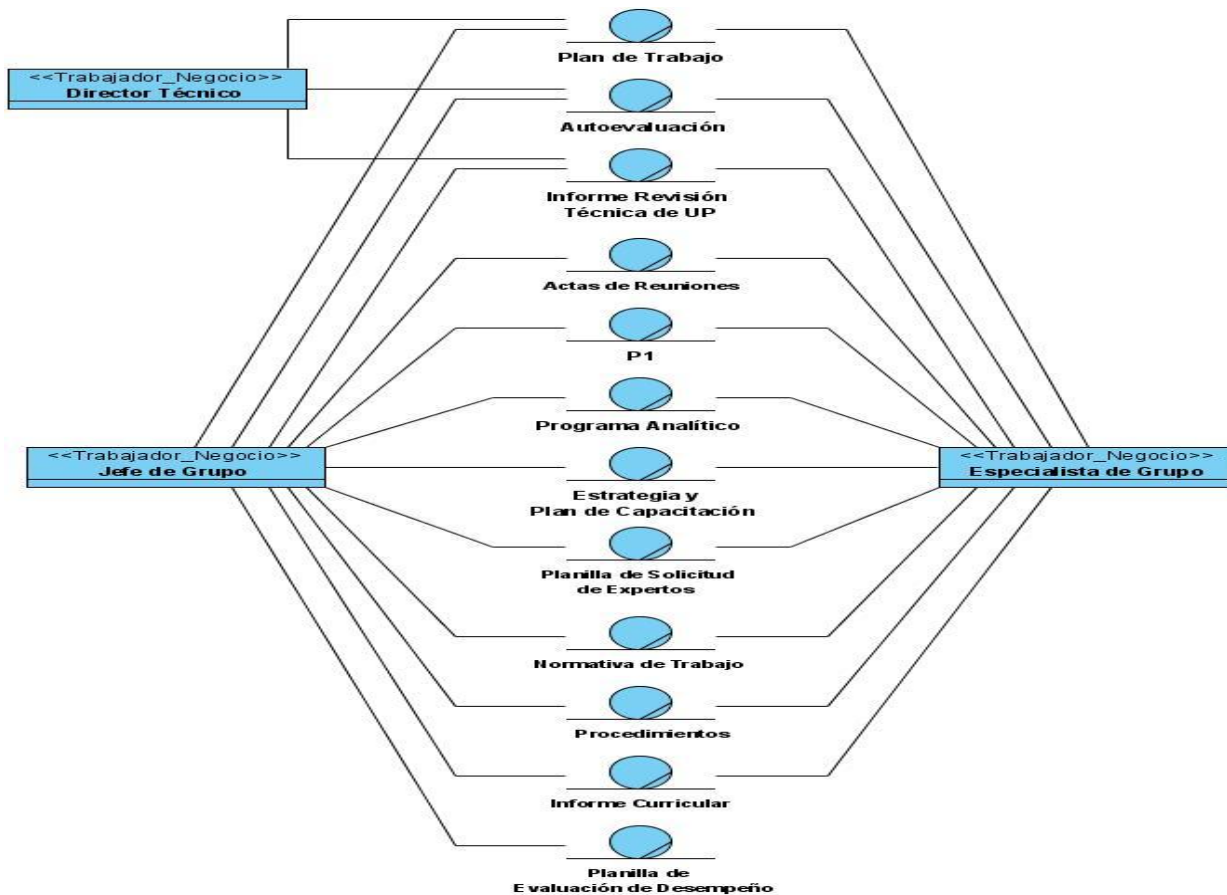


Figura 2.5 Modelo de objetos

### 2.5.6 Reglas del Negocio.

- ❖ **1:** Cuando se identifica un nuevo tipo de documento, a partir de una función del grupo, este debe aprobarse por el Jefe de Grupo y el Director Técnico.
- ❖ **2:** Para crear un documento debe hacerse uso de la plantilla establecida según el tipo documental.
- ❖ **3:** Para archivar un documento debe tenerse en cuenta tipo documental, fecha de creación, quién lo genera y otros aspectos según su clasificación.
- ❖ **4:** Para consultar y recuperar información de un documento deben estar bien definidos los metadatos por cada documento.

### **2.5.7 Glosario de Términos**

EL Glosario de términos es un artefacto muy importante a lo largo del proceso de desarrollo de software, ya que se listan los términos comunes asociados al negocio que emplea el equipo de desarrollo para lograr un vocabulario común con el cliente, evitando confusiones y ambigüedades.

Se define como Glosario de términos: Lista de concepto asociados al negocio que son comúnmente usados y que deben ser del dominio del equipo de desarrollo para poder modelar el negocio y dar una solución a la problemática encontrada. (Anónimo, 2007-2008).

Para consultar el Glosario de Términos (**Ver Anexo 3**)

### **2.5.8 Propuesta de Sistema.**

La UCI representa actualmente la máxima expresión de la informatización de la sociedad cubana como programa priorizado de la Revolución. Como industria del software en Cuba, ha cumplido con la tarea de la creación de importantes aplicaciones informáticas para las diferentes esferas del país, garantizando su mejor funcionamiento.

La solución informática que se da como propuesta de sistema para facilitar la gestión de la información de los procesos llevados a cabo en el grupo de A&EI, eliminando los problemas que presentan hoy, es una aplicación web que va a hacer uso del gestor documental Alfresco, para utilizar los servicios que este brinda en cuanto al tratamiento del documento.

Esta aplicación permitirá crear documentos según plantillas definidas para el tipo documental, registrar los diferentes usuarios que van a tener acceso al sistema, así como los permisos según el rol que desempeñen. Facilitará que la búsqueda de información sea rápida ya que se va a tener control y organización de los documentos registrados en el sistema, sobre qué versión del documento se está trabajando, quién lo escribió, cuándo, quiénes lo leyeron, después de aprobado a quién debe enviarse vía correo electrónico, donde estará almacenado, por cuánto tiempo, etc.

### **2.5.9 Especificación de Requerimientos de Software**

Los requerimientos de software son condiciones o capacidades que debe tener el sistema para satisfacer las necesidades de un cliente, estos deben ser especificados por escrito. A continuación se muestra un listado de los requerimientos tanto funcionales como no funcionales de la aplicación.

### **Requisitos Funcionales**

#### **RF 1: Autenticar usuario**

- Mostrar el formulario de autenticación de usuario.
- Pedir nombre de usuario y contraseña.
- Validar los datos introducidos por el usuario.
  - a. Mostrar un mensaje al usuario si existe alguna dificultad durante la validación.
  - b. Permitir 3 intentos de autenticación en caso de datos incorrectos.

#### **RF 2 Configurar Cuenta de Usuario.**

RF 2.1: Insertar Usuario.

RF 2.3: Eliminar Usuario.

RF 2.4: Buscar Usuario en el Sistema

#### **RF 3 Configurar Permisos**

RF 3.1: Insertar Permisos al usuario.

RF 3.2: Eliminar Permisos al usuario.

RF 3.3: Modificar Permisos al usuario.

#### **RF 4: Configurar Espacios**

RF 4.1: Insertar espacios

RF 4.2: Modificar espacios

RF 4.3: Eliminar espacios

#### **RF 5: Configurar secciones**

RF 5.1: Insertar secciones

RF 5.2: Modificar secciones

RF 5.3: Eliminar secciones

#### **RF 6: Configurar series**

RF 6.1: Insertar series

**RF 6.2:** Modificar series

**RF 6.3:** Eliminar series

### **RF 7: Configurar subseries**

**RF 7.1:** Insertar subseries

**RF 7.2:** Modificar subseries

**RF 7.3:** Eliminar subseries

### **RF 8: Configurar unidades documentales**

**RF 8.1:** Insertar unidades documentales

**RF 8.2:** Modificar unidades documentales

**RF 8.3:** Eliminar unidades documentales

### **RF 9: Configurar contenido.**

**RF 9.1:** Insertar contenido.

**RF 9.2:** Modificar contenido.

**RF 9.3:** Eliminar contenido.

### **RF 10: Configurar metadatos**

**RF 10.1:** Insertar metadatos

**RF 10.2:** Modificar metadatos

**RF 10.3:** Eliminar metadatos

### **RF 11: Gestionar Flujo de Trabajo**

### **RF 12 Consultar y recuperar información**

**RF 12.1** Buscar un documento específico.

**RF 12.2** Buscar información en un documento.

### **RF13: Revisar Documentos**

### **RF14: Publicar Documentos**

#### ***Requisitos No Funcionales***

##### **RNF 1: Apariencia o interfaz externa**

- La interfaz del sistema no contiene numerosas imágenes para evitar demoras en la respuesta a cualquier acción del usuario.
- La misma será sencilla, amigable e intuitiva, de fácil navegación por parte del usuario. Estará diseñada para una óptima visualización siendo adaptable a cualquier resolución.

##### **RNF 2: Usabilidad**

- La aplicación Web deberá facilitar la interacción usuario – sistema con el objetivo de evitar rechazo en el uso de la misma, y guiará mediante mensajes al usuario en las diferentes acciones que realice.
- El usuario deberá poseer conocimientos básicos del manejo de computadoras y estar familiarizado con la gestión y el flujo de la información en el área para la cual se desarrolla el sistema informático.

##### **RNF 3: Soporte**

- La aplicación Web contará con una ayuda donde el usuario podrá suplir las dudas que se le puedan presentar durante la utilización de la misma. El usuario del módulo deberá recibir un adiestramiento previo en la utilización del sistema con el fin de que pueda explotar las prestaciones del sistema sin contratiempos ocasionados por la falta de preparación técnica.

##### **RNF 4: Portabilidad**

- El sistema informático está desarrollado sobre una plataforma Web y podrá ser utilizado desde cualquier sistema operativo, recomendándose para su uso Windows o distribuciones de Linux.

##### **RNF 5: Rendimiento**

- El tiempo de respuesta de una petición al servidor y la velocidad de procesamiento de la información deben ser rápidas para evitar retrasos en la actualización de datos, así como en la toma de decisiones.

### **RNF 6: Software**

#### **RNF 6.1: Software servidor**

- El servidor debe contar con sistema operativo Linux/Debian 4 Etch, Apache.

#### **RNF 6.2: Software Cliente**

- Para utilizar la aplicación Web será necesaria una computadora con el sistema operativo Windows o Linux en cualquiera de sus versiones, recomendándose Windows XP o superior y Ubuntu 7.10 o superior. Además se podrá acceder desde cualquier navegador Web recomendándose Internet Explorer 6 o superior y Firefox 2.0 o superior.

### **RNF 7: Hardware**

- Para garantizar el correcto funcionamiento de la aplicación se necesitan como requerimientos mínimos una computadora con un procesador Pentium II o superior, una memoria RAM de 512 MB o más, un disco duro de 10 GB o más y una tarjeta de red a 128Mbps o más.

### **RNF 8: Restricciones en el diseño y la implementación**

- Se utilizarán los patrones de diseño establecidos.
- La lógica de presentación constituirá una capa independiente de la lógica de negocio, centrando su función en la interfaz de usuario y validaciones simples de la entrada de datos.
- Para el análisis y el diseño del sistema deberá ser utilizada la metodología RUP, usando el lenguaje de modelado UML y como herramienta para llevarlo a cabo el Visual Paradigm.
- Usar el Gestor de Base de Datos PostgreSQL
- Usar el Gestor Documental Alfresco

### **RNF 9: Seguridad**

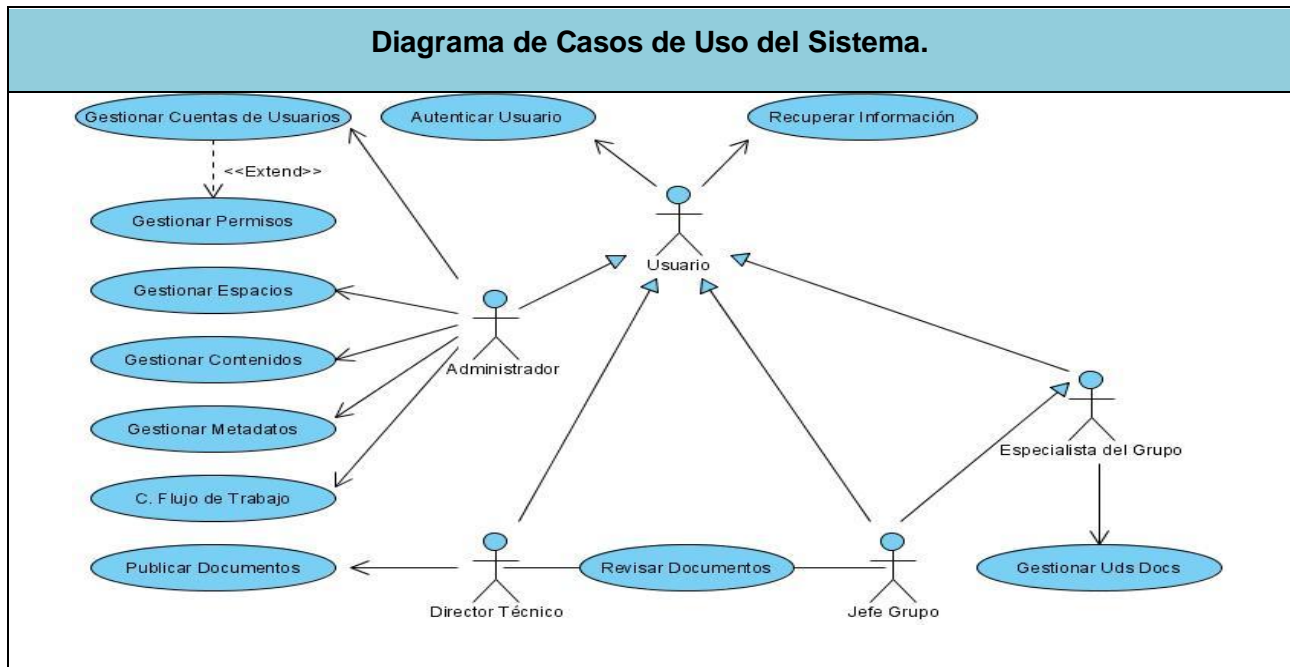
- Identificar al usuario antes de que pueda realizar cualquier acción sobre el sistema.
- Proteger la información manejada por el sistema de accesos no autorizados.
- Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que esté activo.
- Proteger la información manejada por el sistema contra la corrupción de ficheros o estados inconsistentes.

**2.6 Actores del Sistema**

<b>Trabajadores del Negocio</b>	<b>Justificación</b>
<b>Director Técnico</b>	Persona que controla el funcionamiento del grupo de A&Ei.
<b>Jefe Grupo</b>	Es el responsable de organizar, orientar y controlar las tareas particulares por cada especialista en correspondencia a las funciones del grupo.
<b>Especialistas del grupo</b>	Son las personas que ejecutan tareas concretas para el cumplimiento de las diferentes funciones y objetivos.
<b>Usuario</b>	Toda persona que interactúa con el sistema.
<b>Administrador</b>	Es la persona facultada para la gestión del sistema. Es el encargado de administrar las diferentes cuentas de los usuarios autenticados en la aplicación, etc. (los tipos documentales y las acciones sobre ellos así como la organización propia del archivo del sistema).



**2.6.1 Definición de los Casos de Uso del Sistema.**



Para estructurar el diagrama de casos de uso del sistema se aplicaron los siguientes patrones de casos de uso: CRUD completo, CRUD parcial, Múltiples actores por roles diferentes y Múltiples actores por rol común.

**2.6.2 Descripción Textual de los Casos de Uso.**

Caso de Uso:	Autenticar Usuario
<b>Actores:</b>	Usuario
<b>Resumen:</b>	El Caso de Uso (CU) inicia cuando el usuario se autentica para tener acceso al sistema y realizar cualquier acción a la que esté autorizado según los permisos que tenga.
<b>Precondiciones:</b>	Autenticarse antes de realizar cualquier acción sobre el sistema.
<b>Referencias:</b>	RF 1
<b>Prioridad:</b>	Secundario
<b>Flujo Normal de Eventos</b>	

Acción de actor	Respuesta del Sistema
1. El usuario accede al sistema.	2. Muestra formulario para autenticar usuario.
3. El usuario introduce los datos y selecciona la opción "Aceptar".	4. Valida los datos introducidos por el usuario para acceder al sistema y le permite el acceso. <b>Ver Flujo Alternativo</b> (Si el usuario no tiene permiso para acceder al sistema).
5. Realiza la acción que desea.	
Flujo Alternativo de Eventos	
Acción del actor	Acción del actor
	4.1. El sistema muestra un mensaje de error porque el usuario autenticado no es usuario del sistema.
Prototipo de Interfaz de Usuario	Ver Anexo 4 Prototipo de Interfaz de Usuario
Poscondiciones	El sistema debe permitir guardar todos los cambios que el usuario autenticado realice de acuerdo a los permisos que este tenga.

Caso de Uso:	Gestionar Cuenta de Usuario
Actores:	Administrador
Resumen:	El CU inicia cuando el usuario (que en este caso sería el administrador), necesita acceder al sistema para poder realizar la acción a la que tenga permiso por ejemplo: Buscar, Insertar y Eliminar Usuario.
Precondiciones:	Autenticarse antes de realizar cualquier acción sobre el sistema (Ver CU Autenticar usuario).
Referencias:	RF 2
Prioridad:	Crítico
Flujo Normal de Eventos	
Acción de actor	Respuesta del Sistema

1. El administrador accede al sistema.	2. El sistema muestra un formulario con las diferentes opciones a realizar:  <ul style="list-style-type: none"> <li>▪ Si desea Modificar cuenta de usuario, ver sección Modificar Cuenta de Usuario.</li> <li>▪ Si desea Insertar una cuenta de usuario, ver sección Insertar Cuenta de Usuario.</li> <li>▪ Si desea Eliminar cuenta de Usuario, ver sección Eliminar cuenta de Usuario.</li> </ul> Ver CU Gestionar Permisos
3. El administrador selecciona la opción deseada y presiona el botón “Aceptar”.	
<b>Sección “Insertar Cuenta de Usuario”</b>	
<b>Acción de actor</b>	<b>Respuesta del Sistema</b>
	4. El sistema muestra la interfaz para adicionar un usuario.
5. El administrador introduce los datos necesarios para realizar dicha acción y presiona el botón “Aceptar”.	6. El sistema verifica que los datos introducidos sean correctos, los guarda en la BD y muestra un mensaje que se ha realizado dicha operación. <b>Ver Flujo Alterno</b> (Si los datos introducidos no son correctos).
<b>Sección “Modificar Cuenta de Usuario”</b>	
<b>Acción de actor</b>	<b>Respuesta del Sistema</b>
	7. El sistema muestra la interfaz para modificar un usuario.
8. El administrador modifica los datos necesarios para realizar dicha acción y presiona el botón “Aceptar”.	9. El sistema verifica que los datos introducidos sean correctos, los guarda en la BD y muestra un mensaje que se ha realizado dicha operación. <b>Ver Flujo Alterno</b> (Si los datos introducidos no son correctos).
<b>Sección “Eliminar Cuenta de Usuario”</b>	

Acción de actor	Respuesta del Sistema
10. El administrador selecciona la opción “Mostrar Usuarios”	11. Muestra la interfaz donde aparecen los usuarios que están en la BD.
12. El administrador selecciona el usuario al que desea eliminar y oprime el botón “Eliminar”. <b>Ver Flujo Alterno</b> (Si el administrador no seleccionó al usuario que desea eliminar).	13. El sistema elimina el usuario seleccionado, se actualiza la BD y muestra los resultados de la operación.
Flujo Alterno de Eventos	
Acción del actor	Respuesta del Sistema
	<p>6.1. El sistema muestra un mensaje de que no existe ningún usuario que coincida con los datos de búsqueda especificados y el sistema muestra la opción insertar.</p> <p>9.1. El sistema muestra un mensaje de error porque los datos introducidos no son los correctos.</p> <p>12.1. El sistema muestra un mensaje de error porque no se seleccionó el usuario que se quiere eliminar.</p>
<b>Prototipo de Interfaz de Usuario</b>	Ver Anexo 4 Prototipo de Interfaz de Usuario
<b>Poscondiciones</b>	El sistema debe permitir guardar todos los cambios realizados referentes a las cuentas de usuarios.

Caso de Uso:	Gestionar Permisos
<b>Actores:</b>	Administrador
<b>Resumen:</b>	El CU inicia cuando se necesite adicionar, modificar o eliminar permisos de un usuario en específico, el mismo finaliza una vez que se realicen estas operaciones o al menos una de ellas.
<b>Precondiciones:</b>	Antes de iniciarse el CU el administrador tiene que haberse autenticado en

	el sistema. (Ver CU Autenticar usuario)
<b>Referencias:</b>	RF 3
<b>Prioridad:</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción de actor</b>	<b>Respuesta del Sistema</b>
1. El administrador accede al sistema.	<p>2. El sistema muestra las opciones que se puede realizar.</p> <ul style="list-style-type: none"> <li>▪ Si desea Adicionar Permiso a un usuario ver sección Adicionar Permiso.</li> <li>▪ Si desea Modificar Permiso de un usuario ver sección Modificar Permiso.</li> <li>▪ Si desea Eliminar Permiso de un usuario ver sección Eliminar Permiso.</li> </ul>
3. El administrador selecciona la opción deseada y oprime el botón “Aceptar”.	
<b>Sección “Adicionar Permiso”</b>	
	4. El sistema muestra la interfaz con los usuarios y permisos existentes para Adicionar Permiso al usuario. (Ver CU Gestionar Cuentas de Usuarios)
5. El administrador selecciona el usuario al que se le va a adicionar el permiso y adiciona el permiso establecido según el rol que tenga el usuario. Introduce los datos del usuario (usuario y contraseña) y selecciona la opción “Aceptar”. <b>Ver Flujo Alternativo</b> (Si no adiciona el permiso).	6. El sistema guarda en la BD la actualización de los permisos del usuario.
<b>Sección “Modificar Permiso”</b>	
7. El administrador oprime el botón “Mostrar Usuario Permiso”	8. El sistema muestra la interfaz donde aparecen los usuarios con los respectivos permisos.
9. El administrador selecciona del usuario el permiso al que se le realizarán las	10. El sistema guarda los cambios realizados.

modificaciones y oprime el botón “Modificar”, procede a la operación del mismo y una vez culminada la operación selecciona opción “Guardar”. <b>Ver Flujo Alterno</b> (Si no selecciona el permiso al que desea hacerle las modificaciones).	
<b>Sección “Eliminar Permiso”</b>	
11. El administrador oprime el botón “Mostrar Permisos”	12. El sistema muestra la interfaz donde aparecen todos los permisos.
13. El administrador selecciona el permiso que va a eliminar y oprime el botón “Eliminar”. <b>Ver Flujo Alterno</b> (Si no seleccionó el permiso para ser eliminado).	14. El sistema guarda los cambios de la acción realizada, se actualiza la BD y muestra el resultado de la operación.
<b>Flujo Alterno de Eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
	<p>5.1. El sistema muestra un mensaje porque no se adicionó el permiso.</p> <p>9.1. El sistema muestra un mensaje porque no se seleccionó al permiso al que se le iba a realizar las modificaciones.</p> <p>13.1. El sistema muestra un mensaje porque que no seleccionó al permiso que iba ser eliminado.</p>
<b>Prototipo de Interfaz de Usuario</b>	Ver Anexo 4 Prototipo de Interfaz de Usuario
<b>Poscondiciones</b>	El sistema debe permitir guardar todos los cambios realizados referentes a los permisos.

<b>Caso de Uso:</b>	<b>Gestionar Espacios</b>
<b>Actores:</b>	Administrador

<b>Resumen:</b>	El CU inicia cuando se necesite adicionar, modificar o eliminar un espacio, el mismo finaliza una vez que se realicen estas operaciones o al menos una de ellas.	
<b>Precondiciones:</b>	Antes de iniciarse el CU el administrador tiene que haberse autenticado en el sistema. (Ver CU Autenticar usuario)	
<b>Referencias:</b>	RF 4	
<b>Prioridad:</b>	Secundario	
<b>Flujo Normal de Eventos</b>		
<b>Acción de actor</b>	<b>Respuesta del Sistema</b>	
1. El administrador accede al sistema.	2. El sistema muestra las opciones que puede realizar. <ul style="list-style-type: none"> <li>▪ Si desea Adicionar Espacio ver sección Adicionar Espacio.</li> <li>▪ Si desea Modificar Espacio ver sección Modificar Espacio.</li> <li>▪ Si desea Eliminar Espacio ver sección Eliminar Espacio.</li> </ul>	
3. El administrador selecciona la opción que desea realizar y oprime el botón "Aceptar".		
<b>Sección "Adicionar Espacio"</b>		
	4. El sistema muestra la interfaz para adicionar un espacio.	
5. El administrador adiciona el espacio que desea. <b>Ver Flujo Alternativo</b> (Si el administrador no adiciona el espacio).	6. El sistema guarda los datos del espacio creado en la BD.	
<b>Sección "Modificar Espacio"</b>		
7. El administrador presiona el botón "Mostrar Espacios".	8. El sistema muestra la interfaz donde aparecen los espacios creados.	
9. Selecciona el espacio que desea	10. El sistema guarda los cambios realizados del	

<p>modificar y oprime el botón “Modificar”, realiza las modificaciones que necesita y presiona el botón “Guardar”. <b>Ver Flujo Alternativo</b> (Si no selecciona el espacio al que quiere hacerle las modificaciones).</p>	<p>espacio seleccionado.</p>
<b>Sección “Eliminar Espacio”</b>	
<p>11. El administrador presiona el botón “Mostrar Espacios”.</p>	<p>12. El sistema muestra la interfaz donde aparecen los espacios creados.</p>
<p>13. El administrador selecciona el espacio que desea eliminar y oprime el botón “Eliminar”. <b>Ver Flujo Alternativo</b> (Si no se seleccionó el espacio para ser eliminado).</p>	<p>14. El sistema guarda los cambios de la acción realizada, se actualiza la BD y muestra el resultado de la operación.</p>
<b>Flujo Alternativo de Eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
	<p>5.1. El sistema muestra un mensaje porque no se adicionó el espacio requerido por el usuario.</p> <p>9.1. El sistema muestra un mensaje de error porque no se seleccionó el espacio al que se le quiere realizar las modificaciones.</p> <p>13.1. El sistema muestra un mensaje de error porque no se seleccionó el espacio que se quiere eliminar.</p>
<b>Prototipo de Interfaz de Usuario</b>	Ver Anexo 4 Prototipo de Interfaz de Usuario
<b>Poscondiciones</b>	El sistema debe permitir guardar todos los cambios realizados referentes a los espacios.

<b>Caso de Uso:</b>	<b>Gestionar Unidades Documentales</b>
<b>Actores:</b>	Jefe de Grupo, Especialista del Grupo
<b>Resumen:</b>	El CU inicia cuando se necesite buscar, adicionar, modificar o eliminar una unidad documental, el mismo finaliza una vez que se realicen estas



	operaciones o al menos una de ellas.
<b>Precondiciones:</b>	<p>Antes de iniciarse el CU el Jefe de Grupo o el Especialista del Grupo tiene que haberse autenticado en el sistema. (Ver CU Autenticar usuario)</p> <p>Solo se creará una unidad documental a partir de un contenido (o plantilla) establecidos.</p>
<b>Referencias:</b>	RF 8
<b>Prioridad:</b>	Crítico
<b>Flujo Normal de Eventos</b>	
<b>Acción de actor</b>	<b>Respuesta del Sistema</b>
1. El Jefe de Grupo o el Especialista del Grupo accede al sistema.	<p>2. El sistema muestra las opciones que puede realizar con una unidad documental en específico.</p> <p>(Ver CU Gestionar Contenido.)</p> <ul style="list-style-type: none"> <li>▪ Si desea Adicionar una Unidad Documental ver sección Adicionar Unidad Documental.</li> <li>▪ Si desea Modificar una Unidad Documental ver sección Modificar Unidad Documental.</li> <li>▪ Si desea Eliminar una Unidad Documental ver sección Eliminar Unidad Documental.</li> </ul>
3. El Jefe de Grupo o el Especialista del Grupo selecciona la opción deseada y presiona el botón “Aceptar”.	
<b>Sección “Adicionar Unidad Documental”</b>	
	4. El sistema muestra la interfaz para adicionar una Unidad Documental.
5. El usuario adiciona la Unidad Documental que desea y selecciona la opción “Aceptar”. <b>Ver Flujo Alterno</b> (Si el administrador no selecciona la opción “Aceptar”).	6. El sistema guarda en la BD la Unidad Documental creada.
<b>Sección “Modificar Unidad Documental”</b>	

7. El usuario oprime el botón “Mostrar Unidades Documentales”.	8. El sistema muestra las unidades documentales existentes.
9. El usuario selecciona la Unidad Documental a la que se le realizarán las modificaciones y oprime el botón “Modificar”, realiza las modificaciones requeridas y presiona el botón “Guardar”. <b>Ver Flujo Alterno</b> (Si no selecciona la Unidad Documental a la que se le harán las modificaciones).	10. El sistema guarda los cambios realizados.
<b>Sección “Eliminar Unidad Documental”</b>	
11. El usuario oprime el botón “Mostrar Unidades Documentales”.	12. El sistema muestra las unidades documentales existentes.
13. El usuario selecciona la Unidad Documental que desea eliminar y oprime el botón “Eliminar”. <b>Ver Flujo Alterno</b> (Si no se seleccionó la Unidad Documental para ser eliminada).	14. El sistema guarda los cambios de la acción realizada, se actualiza la BD y muestra el resultado de la operación.
<b>Flujo Alterno de Eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
	<p>5.1. El sistema muestra un mensaje de error porque no seleccionó la opción “Aceptar”.</p> <p>9.1. El sistema muestra un mensaje de error porque no seleccionó la Unidad Documental que va a ser modificada.</p> <p>13.1 El sistema muestra un mensaje porque no seleccionó la Unidad Documental que será eliminada.</p>
<b>Prototipo de Interfaz de Usuario</b>	Ver Anexo 4 Prototipo de Interfaz de Usuario
<b>Poscondiciones</b>	El sistema debe permitir guardar todos los cambios realizados referentes a las Unidades Documentales.

<b>Caso de Uso:</b>	<b>Gestionar Contenido</b>	
<b>Actores:</b>	Administrador	
<b>Resumen:</b>	El CU inicia cuando se necesite insertar, modificar o eliminar un tipo de contenido en el sistema, el mismo finaliza una vez que se realicen estas operaciones o al menos una de ellas.	
<b>Precondiciones:</b>	Antes de iniciarse el CU el administrador tiene que haberse autenticado en el sistema. (Ver CU Autenticar usuario)	
<b>Referencias:</b>	RF 9	
<b>Prioridad:</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción de actor</b>	<b>Respuesta del Sistema</b>	
1. El administrador accede al sistema.	2. El sistema muestra las opciones que puede realizar en una unidad documental en específico. <ul style="list-style-type: none"> <li>▪ Si desea Insertar Contenido ver sección Adicionar Contenido.</li> <li>▪ Si desea Modificar Contenido ver sección Modificar Contenido.</li> <li>▪ Si desea Eliminar Contenido ver sección Eliminar Contenido.</li> </ul>	
3. El administrador selecciona la opción deseada y presiona el botón “Aceptar”.		
<b>Sección “Insertar Contenido”</b>		
4. El administrador selecciona la opción crear nuevo contenido.	5. El sistema muestra la interfaz en blanco para agregar los campos.	
6. El administrador inserta los campos establecidos según el nuevo tipo de contenido y selecciona la opción “Guardar”. <b>Ver Flujo Alterno</b> (Si el administrador no selecciona la opción	7. El sistema muestra un mensaje de que la operación se realizó correctamente y el contenido adicionado.	

“Guardar”).	
	8. El sistema guarda los cambios realizados.
<b>Sección “Modificar Contenido”</b>	
9. El administrador selecciona la opción “Mostrar Contenidos”.	10. El sistema muestra la interfaz donde aparece los contenidos existentes en el sistema.
11. El administrador selecciona el contenido que será modificado. <b>Ver Flujo Alternativo</b> (Si no se selecciona el contenido que será modificado).	12. El sistema muestra el contenido seleccionado.
13. El administrador realiza las modificaciones en el contenido y presiona el botón “Guardar”. <b>Ver Flujo Alternativo</b> (Si el administrador no selecciona la opción “Guardar”).	14. El sistema guarda los cambios realizados.
<b>Sección “Eliminar Contenido”</b>	
15. El administrador selecciona la opción “Mostrar Contenido”.	16. El sistema muestra la interfaz donde aparece los contenidos existentes en el sistema.
17. El administrador selecciona el contenido que será eliminado. <b>Ver Flujo Alternativo</b> (Si no se seleccionó el contenido que será eliminado).	18. El sistema muestra la interfaz donde aparece el contenido seleccionado.
19. El administrador oprime el botón “Eliminar”. <b>Ver Flujo Alternativo</b> (Si no se oprime el botón “Eliminar”).	20. El sistema guarda los cambios de la acción realizada, se actualiza la BD y muestra el resultado de la operación.
<b>Flujo Alternativo de Eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
	6.1. El sistema muestra un mensaje porque no se seleccionó la opción “Guardar”.
	11.1. El sistema muestra un mensaje porque no se

	<p>seleccionó el contenido que será modificado.</p> <p>13.1. El sistema muestra un mensaje de error porque no seleccionó la opción “Guardar”.</p> <p>17.1 El sistema muestra un mensaje porque no seleccionó el contenido que será eliminado.</p> <p>19.1. El sistema muestra un mensaje de error porque no se oprimió el botón “Eliminar”.</p>
<b>Prototipo de Interfaz de Usuario</b>	Ver Anexo 4 Prototipo de Interfaz de Usuario
<b>Poscondiciones</b>	El sistema debe permitir guardar todos los cambios realizados referentes a los contenidos.

<b>Caso de Uso:</b>	<b>Gestionar Metadatos</b>	
<b>Actores:</b>	Administrador	
<b>Resumen:</b>	El CU inicia cuando se necesite adicionar, modificar o eliminar metadatos para una rápida localización de los documentos digitales, el mismo finaliza una vez que se realicen estas operaciones o al menos una de ellas.	
<b>Precondiciones:</b>	Antes de iniciarse el CU el administrador tiene que haberse autenticado en el sistema.	
<b>Referencias:</b>	RF 10 (Ver CU Autenticar usuario)	
<b>Prioridad:</b>	Crítico	
<b>Flujo Normal de Eventos</b>		
<b>Acción de actor</b>	<b>Respuesta del Sistema</b>	
1. El administrador accede al sistema.	2. El sistema muestra las opciones que puede realizar referente a los metadatos. <ul style="list-style-type: none"> <li>▪ Si desea Adicionar Metadato ver sección Adicionar Metadato.</li> <li>▪ Si desea Modificar Metadato ver sección Modificar</li> </ul>	

	<p>Metadato.</p> <ul style="list-style-type: none"> <li>▪ Si desea Eliminar Metadato ver sección Eliminar Metadato.</li> </ul>
3. El administrador selecciona la opción deseada y presiona el botón “Aceptar”.	
<b>Sección “Adicionar Metadato”</b>	
	4. El sistema muestra la interfaz para seleccionar el documento al que se le va a adicionar metadatos.
5. El administrador selecciona el documento y adiciona los metadatos que desea. <b>Ver Flujo Alterno</b> (Si el administrador no adiciona metadatos).	6. El sistema guarda los metadatos insertados en la BD.
<b>Sección “Modificar Metadato”</b>	
7. El administrador selecciona la opción “Mostrar Metadatos”.	8. El sistema muestra la interfaz con los metadatos existentes por documentos.
9. El administrador selecciona el documento y de este el metadato que será modificado y oprime el botón “Modificar”, realiza las modificaciones que necesita al metadato seleccionado y presiona el botón “Guardar”. <b>Ver Flujo Alterno</b> (Si no selecciona el metadato al que se le quiere realizar las modificaciones).	10. El sistema guarda los cambios realizados del metadato modificado.
<b>Sección “Eliminar Metadato”</b>	
11. El administrador selecciona la opción “Mostrar Metadatos”.	12. El sistema muestra la interfaz con los metadatos existentes por documentos.
13. El administrador selecciona el documento y de este el metadato que se desea eliminar y oprime el botón “Eliminar”. <b>Ver Flujo Alterno</b> (Si no se seleccionó el metadato para ser	14. El sistema guarda los cambios de la acción realizada, se actualiza la BD y muestra el resultado de la operación.

eliminado).	
Flujo Alternativo de Eventos	
Acción del actor	Respuesta del Sistema
	<p>5.1. El sistema muestra un mensaje porque no se adicionó el o los metadato(s) requerido(s).</p> <p>9.1. El sistema muestra un mensaje de error porque no se seleccionó el metadato a la que se le realizarán las modificaciones.</p> <p>13.1. El sistema muestra un mensaje de error porque no se seleccionó el metadato que se quiere eliminar.</p>
Prototipo de Interfaz de Usuario	Ver Anexo 4 Prototipo de Interfaz de Usuario
Poscondiciones	El sistema debe permitir guardar todos los cambios realizados referentes a los metadatos.

Caso de Uso:	Configurar Flujo de Trabajo del Documento
Actores:	Administrador
Resumen:	El CU inicia cuando el administrador configura el proceso de vida del documento y finaliza cuando queda establecido.
Precondiciones:	Autenticarse antes de realizar cualquier acción sobre el sistema. (Ver CU Autenticar usuario)
Referencias:	RF 11
Prioridad:	Crítico
Flujo Normal de Eventos	
Acción de actor	Respuesta del Sistema
1. El administrador accede al sistema.	2. Muestra formulario para configurar flujo de trabajo.

3. El administrador establece pasos del ciclo de vida del documento y presiona el botón "Aceptar".	4. Muestra un mensaje que la operación fue realizada con éxito y guarda los pasos establecidos por el administrador.
<b>Flujo Alternativo de Eventos</b>	
<b>Acción del actor</b>	<b>Acción del actor</b>
	3.1. Muestra un mensaje porque el administrador no seleccionó el botón "Aceptar".
<b>Poscondiciones</b>	El sistema debe permitir guardar todos los cambios realizados referentes al flujo de trabajo.

<b>Caso de Uso:</b>	<b>Recuperar Información</b>
<b>Actores:</b>	Usuario
<b>Resumen:</b>	El CU inicia cuando se necesite recuperar una información, el mismo finaliza una vez que se realice la operación.
<b>Precondiciones:</b>	Antes de iniciarse el CU el usuario tiene que haberse autenticado en el sistema. (Ver CU Autenticar usuario)
<b>Referencias:</b>	RF 12
<b>Prioridad:</b>	Secundario
<b>Flujo Normal de Eventos</b>	
<b>Acción de actor</b>	<b>Respuesta del Sistema</b>
1. El usuario accede al sistema.	2. El sistema muestra la interfaz con las diferentes opciones.
3. El usuario selecciona la opción "Recuperar Información".	4. El sistema muestra una interfaz con las opciones de búsqueda para realizar la recuperación de la información que necesita.
5. El usuario introduce los datos en la opción de búsqueda que brinda la interfaz	6. El sistema muestra la información requerida por el usuario. <b>Ver Flujo Alternativo</b> (Si no se muestra la



y oprime el botón “Buscar”.	información que necesita).
Flujo Alternativo de Eventos	
Acción del actor	Respuesta del Sistema
	6.1. El sistema muestra un mensaje porque no encontró la información requerida por el usuario y brinda la opción de introducir nuevos criterios de búsqueda.
Prototipo de Interfaz de Usuario	Ver Anexo 4 Prototipo de Interfaz de Usuario
Poscondiciones	El sistema debe permitir actualizar, eliminar o salvar la información recuperada.

Caso de Uso:	Revisar Documentos
Actores:	Director Técnico, Jefe de Grupo
Resumen:	El CU inicia cuando se necesita revisar un documento en específico, el mismo finaliza una vez que se realice la operación.
Precondiciones:	Antes de iniciarse el CU el Director Técnico o el Jefe de Grupo tiene que haberse autenticado en el sistema. (Ver CU Autenticar usuario)
Referencias:	RF 13
Prioridad:	Secundario
Flujo Normal de Eventos	
Acción de actor	Respuesta del Sistema
1. El Director Técnico o el Jefe de Grupo accede al sistema.	2. El sistema muestra la interfaz con las diferentes opciones.
3. El usuario autenticado selecciona la opción “Mostrar Documentos”.	4. El sistema muestra la interfaz con los documentos.
5. El usuario selecciona el documento que desea revisar y selecciona el botón	6. El sistema muestra el documento requerido por el

“Aceptar”. <b>Ver Flujo Alterno</b> (Si no selecciona el documento para ser revisado).	usuario.
7. El usuario revisa el documento, en caso de que haya que realizarle cambios se procede a la acción y por último se va a la opción “Aceptar”. <b>Ver Flujo Alterno</b> (Si no selecciona el botón “Aceptar”).	8. El sistema muestra el documento revisado.
<b>Flujo Alterno de Eventos</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
	5.1. El sistema muestra un mensaje de error porque no se seleccionó el documento que será revisado.  7.1. El sistema muestra un mensaje porque no seleccionó la opción “Aceptar”.
<b>Prototipo de Interfaz de Usuario</b>	Ver Figura
<b>Poscondiciones</b>	El sistema debe permitir guardar todos los cambios realizados referentes al documento revisado.

<b>Caso de Uso:</b>	<b>Publicar Documento</b>
<b>Actores:</b>	Director Técnico
<b>Resumen:</b>	El CU inicia cuando se necesita publicar un documento el cual ya está aprobado por las personas autorizadas del grupo, el mismo finaliza una vez que se realice la operación.
<b>Precondiciones:</b>	Antes de iniciarse el CU el Director Técnico tiene que haberse autenticado en el sistema. (Ver CU Autenticar usuario)
<b>Referencias:</b>	RF 14
<b>Prioridad:</b>	Secundario

Flujo Normal de Eventos	
Acción de actor	Respuesta del Sistema
1. El Director Técnico accede al sistema.	2. El sistema muestra la interfaz con las diferentes opciones.
3. El Director Técnico selecciona la opción “Publicar Documento”.	4. El sistema muestra la interfaz para seleccionar el documento que desea publicar.
5. El Director Técnico selecciona el documento que será publicado en la interfaz y selecciona el botón “Publicar”. <b>Ver Flujo Alternativo</b> (Si no se adiciona contenido).	6. El sistema muestra el documento publicado.
Flujo Alternativo de Eventos	
Acción del actor	Respuesta del Sistema
	5.1. El sistema muestra un mensaje de error porque no se adicionó el documento que será publicado.
<b>Prototipo de Interfaz de Usuario</b>	Ver Anexo 4 Prototipo de Interfaz de Usuario
<b>Poscondiciones</b>	El sistema debe permitir que se recuperen y actualicen los documentos publicados.

### 2.7 Conclusiones

En este capítulo, después de realizar un estudio sobre la situación actual en el grupo de A&EI se propone realizar una aplicación informática que mejore el funcionamiento de dicha entidad. Para ello, se obtuvo el diseño documental, donde se detallan los principales conceptos a tener en cuenta. Se realizó además el modelo del negocio, definiéndose actores, trabajadores y procesos descritos por el cliente. Se obtuvo un listado de funcionalidades que debe tener el sistema, expresados en los requerimientos funcionales y no funcionales. Partiendo de este punto, base de todo proceso de desarrollo, se puede comenzar con la construcción de la propuesta, velando por el cumplimiento de todos los requerimientos y funcionalidades consideradas.

## Capítulo III Análisis & Diseño del Sistema

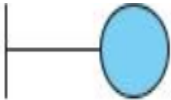
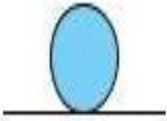
### 3.1 Introducción


En este capítulo se realiza el análisis y diseño de la propuesta de solución del sistema de gestión de documentos digitales, desarrollándose los diagramas de clases del análisis de los casos de usos definidos, así como los diagramas de interacción correspondientes a cada uno de ellos, refinando estos diagramas en el diseño para lograr un mejor entendimiento y una mayor funcionalidad del sistema que se propone. Se aplican métricas para evaluar la calidad de los principales artefactos que se generan.

### 3.2 Modelo de Análisis

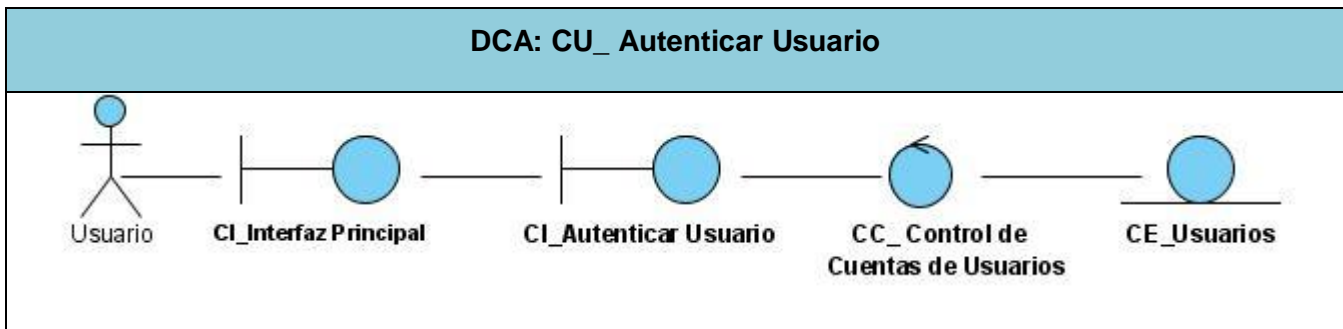
En el flujo de trabajo de análisis se trabaja sobre los aspectos internos del sistema a desarrollar, se pueden estructurar los requisitos de manera que faciliten su comprensión, su preparación, su modificación y en general su mantenimiento.

En el modelo de análisis se obtiene un mayor poder expresivo y una mayor formalización y se puede considerar como una primera aproximación al modelo de diseño. Para la construcción del modelo de análisis se identificaron las clases que describen la realización de los casos de usos, que pueden ser de tres tipos fundamentales:

Clases del análisis	
	<p><b>Clases Interfaz:</b> Se utilizan para modelar la interacción entre el sistema y sus actores, lo que clarifican y reúnen los requisitos en los límites del sistema. Representan abstracciones de ventanas, formularios, paneles, interfaces de comunicaciones, interfaces de impresoras, sensores y terminales.</p>
	<p><b>Clases Entidad:</b> Se utilizan para modelar la información que posee una vida larga y que es a menudo persistente. Modelan la información y el comportamiento asociado de algún fenómeno o concepto y reflejan la información de modo que beneficia a los desarrolladores al diseñar e implementar el sistema, incluyendo su soporte de persistencia.</p>

	<p><b>Clases controladoras:</b> Representan la coordinación, secuencia, transacciones, y control de otros objetos, y se usan con frecuencia para encapsular el control de un caso de uso en concreto. Se utilizan para representar derivaciones y cálculos complejos.</p>
---	---

A continuación se muestra uno de los diagramas de clases del análisis, para obtener más información sobre los diagramas restantes consultar **Anexo 5**.



### 3.3 Diagramas de interacción

Los diagramas de interacción muestran una interacción concreta: un conjunto de objetos y sus relaciones, junto con los mensajes que se envían entre ellos. Modelan el comportamiento dinámico del sistema; el flujo de control en una operación. Describe la interacción entre objetos, los objetos interactúan a través de mensajes para cumplir ciertas tareas. Las interacciones proveen un comportamiento y típicamente implementan un Caso de Uso.

Existen dos tipos de diagramas de interacción en UML:

1. Diagramas de Secuencia (dimensión temporal).
2. Diagrama de Colaboración (dimensión estructural).

#### 3.3.1. Diagramas de Secuencia.

Un diagrama de secuencia muestra las interacciones entre objetos, ordenadas en secuencia temporal durante un escenario concreto. Si los casos de uso tienen varios flujos o subflujos distintos, suele ser útil crear un diagrama de secuencia para cada uno de ellos.

Para consultar los diagramas de secuencia ver **Anexo 7**.

### 3.4 Modelo de Diseño

El diseño tiene como propósito formular los modelos que se centran en los requisitos no funcionales y en el dominio de la solución. Se basa en que el modelo del sistema es una colección de objetos que cooperan entre sí, donde cada objeto es una instancia de una clase en una jerarquía de clases. Además impone una estructura del sistema que se debe conservar lo más exacto posible cuando se construya el mismo.

#### 3.4.1 Diagrama de Clases del Diseño


Un diagrama de clases de diseño es un diagrama que muestra los siguientes elementos: Clases, Interfaces, Colaboraciones y las Relaciones de dependencia, generalización y asociación. Se utilizan para modelar principalmente la vista de diseño estática de un sistema.



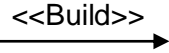
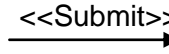
**Clases del Diseño:** Una clase de diseño es una abstracción de una clase o construcción en la implementación del sistema.

Para obtener más información sobre los diagramas de clases del diseño consultar **Anexo 6**.

#### 3.4.2 Extensiones UML para Web utilizadas en el sistema.

UML posee una extensión para el modelado de aplicaciones Web, usada para el diseño de las clases, dicha extensión utiliza diferentes estereotipos que permiten definir un nuevo significado de la semántica para el elemento a modelar, los estereotipos más usados son:

Estereotipos Web según UML para la construcción de Diagramas de Clases del Diseño	
	<p><b>Server Page:</b> Representa la página Web que tiene código que se ejecuta en el servidor. Este código interactúa con recursos en el servidor. Las operaciones representan las funciones del código y los atributos las variables visibles dentro del alcance de la página. Esta clase sólo puede tener relaciones con objetos en el servidor.</p>

	<p><b>Client Page:</b> Una instancia de Página Cliente es una página Web, con formato HTML. Mezcla de datos, presentación y lógica. Son interpretadas por el navegador. Sus atributos son las variables declaradas dentro del script que son accesibles para páginas cualquier función dentro de la página. Cada página cliente es construida por una sola página de servidor.</p>
	<p><b>Form:</b> Colección de elementos de entrada que son parte de una página cliente. Se relaciona directamente con la etiqueta de igual nombre del HTML. Sus atributos son los elementos de entrada del formulario (Text Field, Text Area, Button, Label, Radio Button, Radio Group, Select, Check Box y Hidden Fields).</p>
	<p><b>Build:</b> Representa una asociación especial que relaciona las páginas cliente con las páginas servidor, de forma general se expresa como que las páginas que se encuentran en el servidor construyen las páginas en el cliente. Debe ser una relación direccional, donde una página servidor puede construir una o más páginas cliente.</p>
	<p><b>Submit:</b> Es la relación que se crea siempre entre una página servidor y un formulario, a través de esta relación el formulario manda los valores de sus campos al servidor, para ser procesados por la página servidor.</p>

### 3.5 Patrones aplicados en el Sistema de Gestión de Documentos Digitales.

Dentro de los patrones utilizados tenemos los GRASP (General Responsibility Assignment Software Patterns) permiten asignar correctamente las responsabilidades a cada una de las clases que intervienen en el modelo. En el sistema se utilizaron cuatro de los cinco patrones GRASP fundamentales:

- **Experto:** Este patrón es el principio básico de asignación de responsabilidades. Se asignaron responsabilidades a las clases con la información necesaria para cumplirla, lo cual permitió que se conservara el encapsulamiento en las clases del sistema debido a que los objetos lograron valerse de su propia información para hacer lo se les pedía, esto favoreció el logro del diseño de un sistema más robusto y fácil de mantener..
- **Creador:** Se asignaron responsabilidades a las clases de crear instancias de otras conociendo que las primeras son las que contienen la información para ello lográndose un bajo acoplamiento lo cual se tradujo en menos dependencias respecto al mantenimiento así como en mejores oportunidades de reutilización en el sistema.
- **Alta cohesión:** Se asignaron responsabilidades a las clases de manera que todos sus métodos tuvieran un comportamiento bien definido logrando mantener la complejidad dentro de límites manejables obteniéndose de esta manera clases más fáciles de entender, cambiar y reutilizar.
- **Bajo acoplamiento:** Cada clase está acoplada a las clases estrictamente necesarias por lo que si ocurriera algún cambio en alguna parte del sistema no se afectarían el resto de los componentes, además se pueden entender las clases por separado.

De los patrones GoF se aplicó en el diseño de clases del SGDD:

- **Fachada:** Este patrón se usó para acceder a la Capa Acceso a Datos. La intención del patrón fachada es facilitar una interfaz simple a un subsistema complejo (Gamma, y otros, 1994), permitiendo así estructurar en capas un sistema.

El framework CodeIgniter trabaja haciendo uso del patrón arquitectónico **Modelo-Vista-Controlador**. MVC es una aproximación al software que separa la lógica de la aplicación de la presentación. En la práctica, permite que sus páginas web contengan mínima codificación ya que la presentación es separada del código PHP.

- El **Modelo** representa la estructura de datos. Típicamente sus clases de modelo contendrán funciones que lo ayudarán a recuperar, insertar y actualizar información en su base de datos.
- La **Vista** es la información que es presentada al usuario. La Vista normalmente será una página web, pero en CodeIgniter, una vista también puede ser un fragmento de una página



como un encabezado o un pie de página. También puede ser una página RSS, o cualquier otro tipo de "página".

- El **Controlador** sirve como un intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para procesar la petición HTTP y generar una página web.

CodeIgniter tiene un enfoque bastante flexible del MVC, ya que los Modelos no son requeridos. Si no necesita agregar separación, o descubre que mantener los modelos requiera más complejidad que quería, puede ignorarlos y construir su aplicación mínimamente usando Controladores y Vista. CodeIgniter también le permite incorporar sus códigos existentes, o incluso desarrollar librerías de núcleo para el sistema, habilitándolo a trabajar en una forma que hace que tenga más sentido para usted. (Martínez, Ruiz Díaz, & Waisbrot, 2008)

El uso de estos patrones permitió que el trabajo ganara en organización, se lograra una mayor reutilización de código ahorrando tiempo y lográndose una mayor escalabilidad.

### ***3.6 Descripción de la arquitectura propuesta en el sistema***

Existe en la actualidad una marcada tendencia a que los equipos de desarrollo de software se dividan el trabajo por capas, logrando que al realizar cambios en una de ellas no se vean afectadas las demás, se logra además una óptima integración del sistema, ya que se construyen las piezas por separado y después se integran. Esta arquitectura en capas se ha vuelto muy común a la hora de construir software debido a las facilidades que brinda, pues permite la reutilización y la independencia entre las capas, se pueden realizar cambios en capas sin tener que modificar las otras, facilita la estandarización y la utilización de los recursos.

#### ***Capas utilizadas en el sistema***

**Capa de presentación o de interfaz de usuario:** Está formada por los formularios .html, sus controles o atributos y las client page. Está definida como la capa con la que interactúa el usuario, esta capa se comunica únicamente con la capa de negocio.

**Capa de negocio:** En esta capa se almacenarán todos los componentes que gestionarán el negocio de la aplicación. Está formada por las entidades que se definen en el sistema, que representan objetos que van a ser manejados o consumidos por toda la aplicación y por las clases .php que se encuentran dentro del negocio de la aplicación.

**Capa de acceso a datos:** Contiene clases que interactúan con la base de datos, estas clases altamente especializadas permiten, utilizando los procedimientos almacenados generados, realizar todas las operaciones con la base de datos de forma transparente para la capa de negocio.



Figura 3.2 Organización de las capas en el sistema.

### 3.7 Métricas

La medición es fundamental para cualquier disciplina de ingeniería, y la ingeniería del software no es una excepción. La medición nos permite tener una visión más profunda, proporcionando un mecanismo para la evaluación objetiva. (Pressman, 2005).

El resultado de toda investigación científica tiene que estar validado. Así, lo que está teóricamente descrito, puede ser llevado a la práctica sin mayores problemas. En el presente capítulo, aplicando métricas de medición de la calidad de la especificación de los requisitos, del DCUS y del diseño se va a realizar la validación de los resultados obtenidos en el análisis y diseño.

#### 3.7.1 Introducción

Se definen las métricas de software como un buen medio para entender, monitorizar, controlar, predecir y probar el desarrollo software y los proyectos de mantenimiento (Briand et al., 1996).

En general, la medición persigue tres objetivos fundamentales: ayudarnos a entender qué ocurre durante el desarrollo y el mantenimiento, permitirnos controlar qué es lo que ocurre en nuestros proyectos y poder mejorar nuestros procesos y nuestros productos (Fenton y Pfleeger, 1997).

Una medida proporciona una indicación cuantitativa de la extensión, cantidad, dimensiones, capacidad o tamaño de algunos atributos de un proceso o producto. La medición es el acto de determinar una medida.

### 3.7.2 Métricas de Casos de uso

Principales Métricas de Casos de Uso		
Métricas principales	Rango habitual	Descripción
NOS	[4,9]	Números de Pasos
NOAS/NOS	[30%,60%]	Proporción de pasos de actor
NOSS/NOS	[40%, 80%]	Proporción de pasos de sistema
NOUS/NOS	[0%,35%]	Proporción de pasos de casos de uso
CC	[1,5]	Complejidad (NOCS+NOE+1) <span style="float: right;">Cicломática</span>

#### **Métrica NOAS (Number of Actor Steps- Número de pasos del actor) /NOS (Number of Steps- Número de pasos)**

Esta heurística se basa en la idea de que un caso de uso sirve para expresar una interacción actor-sistema. Por ello, el número de pasos de actor y el de pasos del sistema deben estar en torno al 50%, considerando también la posibilidad de que existan pasos de inclusión o extensión en los que se realice otro caso de uso. Las situaciones que llevan a esta métrica fuera del rango habitual son:

- El hecho de obviar la participación del sistema, por lo que el caso de uso resulta incompleto. Es la situación más habitual.
- El hecho de haber desglosado demasiado las acciones de un actor determinado. En este caso

aparecen varios pasos seguidos, del mismo actor, lo cual podría haberse evitado uniéndolos en uno solo, separando las acciones por comas en el texto del paso.

El rango habitual de esta métrica es [30%,60%]. Un valor alto de la misma puede estar condicionado por el hecho de que el caso de uso no incluye todo lo que debe hacer el sistema para alcanzar el objetivo de este o demasiado desglose de los pasos del actor. Un valor bajo de la misma puede estar condicionado por el hecho de que no incluye todo lo que debe hacer el actor para alcanzar el objetivo del caso de uso, demasiado desglose de los pasos del sistema. (Durand Martínez & Becerra González, Junio, 2008).

<b>Aplicación de la métrica NOAS/NOS al DCUS</b>			
<b>Nombre del CUS</b>	<b>NOAS</b>	<b>NOS</b>	<b>Valor de la métrica</b>
<b>Gestionar Cuentas de Usuario</b>	<b>5</b>	<b>11</b>	<b>45%</b>
<b>Autenticar Usuario</b>	<b>3</b>	<b>6</b>	<b>50%</b>
<b>Gestionar Permisos</b>	<b>7</b>	<b>14</b>	<b>50%</b>
<b>Gestionar Espacios</b>	<b>7</b>	<b>14</b>	<b>50%</b>
<b>Gestionar Uds Docs</b>	<b>7</b>	<b>14</b>	<b>50%</b>
<b>Gestionar Contenidos</b>	<b>10</b>	<b>20</b>	<b>50%</b>
<b>Gestionar Metadatos</b>	<b>7</b>	<b>14</b>	<b>50%</b>
<b>Configurar FT</b>	<b>2</b>	<b>5</b>	<b>41%</b>
<b>Publicar Docs</b>	<b>3</b>	<b>6</b>	<b>50%</b>
<b>Revisar Docs</b>	<b>4</b>	<b>9</b>	<b>44%</b>

<b>Buscar Información</b>	<b>3</b>	<b>6</b>	<b>50%</b>
---------------------------	----------	----------	------------

La métrica NOAS/NOS, para la medición de los CUS fue aplicada a todos y cada uno de los casos de uso del sistema. En la tabla anterior se muestran los valores arrojados después de su aplicación. Los valores obtenidos para cada caso de uso está en el rango habitual, con valores bastante cercanos al 50%, lo que significa que las descripciones están bastante completas, se evidencia que no se obvian participaciones del actor o del sistema, o sea, que se describe cada paso tanto por parte del actor como del sistema cumpliendo el objetivo del caso de uso.

### 3.7.3 Métricas de la calidad de la especificación para determinar la ambigüedad

La calidad del modelo de análisis y la correspondiente especificación de requisitos, según plantean Davis y sus colegas (Pressman, 2005), pueden determinarse empleando una lista de características que proponen para valorar la especificidad (ausencia de ambigüedad), completación, corrección, comprensión, capacidad de verificación, consistencia interna y externa, capacidad de logro, concisión, trazabilidad, capacidad de modificación, exactitud y capacidad de reutilización.

Haciendo uso de la siguiente expresión matemática propuesta por Davis, se puede obtener un valor aproximado de la ambigüedad de la especificación, o lo que es lo mismo, de la **especificidad** (ausencia de ambigüedad) de la interpretación de los revisores para cada requisito.

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

Donde:  $n_{ui}$  es el número de requisitos para los que todos los revisores tuvieron interpretaciones idénticas.

La variable  $n_r$  representa el número de requisitos en una especificación, tal como:

$$n_r = n_f + n_{nf}$$

Donde:  $n_f$  es el número de requisitos funcionales y  $n_{nf}$  es el número de requisitos no funcionales.

Cuando el valor de  $Q$  se encuentra cercano a 1 se dice que es menor la ambigüedad de la especificación.

Para aplicar esta métrica, se realizaron dos revisiones por varios revisores. El objetivo principal de estas revisiones era obtener los requisitos con el mayor grado de claridad posible y sin ambigüedades.

Resultados:

$$n_r = n_f + n_{nf}$$

$$n_r = 14 + 9$$

$$n_r = 23$$

**Primera Revisión**

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

$$Q_i = 19 / 23$$

$$Q_i = 0.8$$

**Segunda Revisión**

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

$$Q_i = 23 / 23$$

$$Q_i = 1$$

### Resultados de aplicar la métrica de especificación de los requisitos

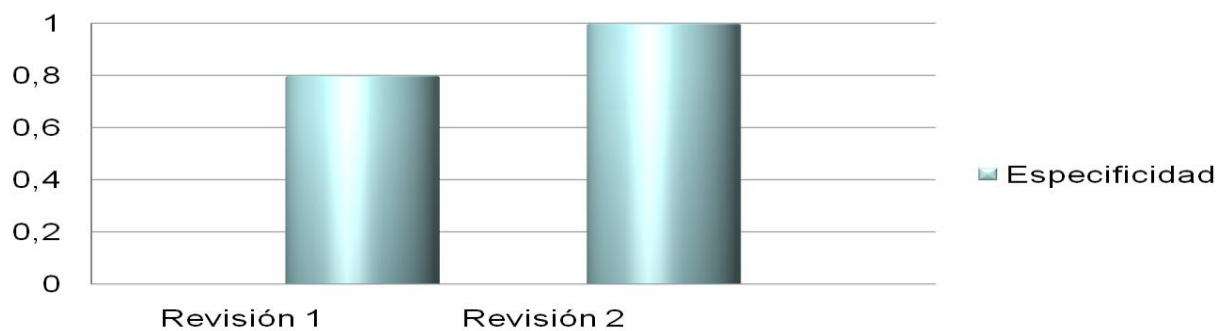


Figura 3.3 Gráfica del resultado de aplicar la métrica de especificidad

La gráfica muestra cómo varía la especificidad de los requisitos entre una revisión y otra. El valor de la primera revisión demuestra, que aunque el valor de la especificidad está próximo a 1, existen ambigüedades en cuanto a la redacción de los requisitos lo que provoca que los revisores no tengan interpretaciones idénticas de estos. Ya para la segunda revisión se rectificaron estos detalles y se logró que aumentara el valor de la especificidad.

### 3.7.4 Métricas para el Modelo de Diseño

#### Métricas Propuestas por Lorenz y Kidd

**Tamaño de Clase (TC).** Lorenz y Kidd plantean que una clase puede medirse por:

- Total de operaciones (heredadas o privadas) encapsuladas en la clase.
- Nro de atributos (heredados o privados) encapsulados por la clase.
- $A > TC$ , menor reutilización, complica la implementación y comprobación.

Un TC grande afecta los indicadores de calidad definidos para esta métrica por los especialistas:

**Reutilización:** reduce la reutilización de la clase.

**Implementación:** complica la implementación.

**Responsabilidad:** la clase debe tener bastante responsabilidad. Las medidas o umbrales para los parámetros de calidad han sido una polémica a nivel mundial en el diseño de sistemas. Algunos especialistas plantean umbrales para estas métricas según se muestra en la siguiente tabla, estos fueron los aplicados en el diseño de este sistema.

No de Operaciones y/o Atributos	
TC	Umbral
Pequeño	$\leq 20$
Medio	$> 20$ y $\leq 30$
Grande	$> 30$

**Resultado:** Esta métrica fue aplicada en la capa de presentación. La capa de presentación tiene un total de 20 clases, todas las clases de acuerdo con los umbrales de la tabla anterior se pueden

considerar pequeñas. Estos valores demuestran que los indicadores de calidad reutilización, implementación y responsabilidad no se ven afectados.

### **3.8 Conclusiones**

En el presente capítulo, concebido como continuación del capítulo II y tomando como base las características que se habían definido en dicho capítulo, se construyeron los artefactos definidos por RUP para el flujo de Análisis y Diseño, los diagramas de clases del análisis y del diseño así como los diagramas de interacción. Se hace uso de patrones de diseño los cuales mejoran considerablemente la calidad del software, y queda establecido el Análisis y Diseño del Sistema de Gestión de Documentos Digitales para que los desarrolladores puedan implementarlo.

Al mismo tiempo que se desarrollaba el trabajo de diploma, con los principales artefactos que fueron definidos, se comenzó parte de la implementación del sistema del Grupo de A&EI y ha quedado demostrado que el diseño ha sido útil para este flujo de trabajo.

Los resultados obtenidos hasta el momento fueron considerados por la Dirección Técnica como punto de partida para ampliar el sistema al alcance de la dirección, lo cual evidencia que el cliente está satisfecho con el trabajo realizado.

Para la validación de los resultados alcanzados en el análisis y diseño se aplicaron métricas que garantizan la efectividad en la medición de la calidad. Se aplicaron métricas para evaluar el DCUS, la especificación de los requisitos, así como el diseño del sistema, obteniendo resultados satisfactorios dentro de los umbrales propuestos por cada métrica.



## Conclusiones Generales

Con la realización de este trabajo:

- ❖ Se demostró la necesidad de diseñar un sistema que permita realizar la gestión de los documentos digitales que se generan en el Grupo de Arquitectura y Estándares de la Información de la Dirección Técnica de la Infraestructura Productiva de la UCI. La propuesta de solución introduce mejoras en los procesos de gestión documental, producidos por un profundo análisis de los procesos en pos de lograr una óptima organización y control de los documentos, facilitar las opciones de búsqueda y recuperación de la documentación y así aumentar la eficiencia y calidad en el cumplimiento de las funciones que realizan como grupo.
- ❖ Una vez realizado el estudio de las tendencias actuales de las metodologías y herramientas más usadas en el desarrollo de sistemas informáticos de este tipo, teniendo en cuenta las características propias del proyecto, los altos niveles de accesibilidad, estabilidad y seguridad que exige el trabajo con la documentación que se genera en el grupo y las ventajas que estas ofrecen, se arribó a la elección de estas metodologías y herramientas: RUP como metodología de desarrollo (por las grandes potencialidades que presenta para el análisis y diseño, así como la documentación que brinda), UML (para el modelado de los artefactos de todas las fases y flujos de trabajo), Visual Paradigm (herramienta de modelado), Alfresco (gestor documental), CodeIgniter (framework de desarrollo para PHP) y PostgreSQL (sistema gestor de la base de datos).
- ❖ La propuesta de solución que se presenta da cumplimiento al objetivo general trazado al principio de la investigación, pues brinda los artefactos correspondientes al flujo que propone RUP de Análisis y Diseño para que se le pueda dar continuidad al trabajo. Fueron aplicadas métricas que evaluaron los resultados como positivos.

## Recomendaciones

Como resultado del proceso de investigación y realización de la aplicación han surgido ideas que serían recomendables tener en cuenta para un futuro perfeccionamiento del sistema, a continuación se listan las mismas:

- ❖ Continuar con la línea de desarrollo del sistema propuesto.
- ❖ Agregar nuevas funcionalidades en la medida del avance del trabajo y de acuerdo con las expectativas de los usuarios finales.
- ❖ Realizar el análisis y diseño para integrar al sistema todos los grupos que conforman a la Dirección Técnica y no solo el grupo de A&EI.

## Bibliografía Referenciada

Hernández León, R. A., & Coello González, S. (2002). *EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTÍFICA*. Ciudad de la Habana.

Canós, J. H., Letelier, P., & Penadés, M. C. (n.d.). *Métodologías Ágiles en el Desarrollo de Software*. Retrieved marzo 1, 2009, from <http://www.willydev.net/descargas/prev/TodoAgil.pdf>

Codina, L. (2003, Mayo). *El Profesional de la Información*. Retrieved marzo 1, 2009, from <http://www.elprofesionaldelainformacion.com/>

Cáceres, P., & Marcos, E. (2000). *Procesos Ágiles para el desarrollo de aplicaciones web*.

Calzado, V. (2003, febrero-marzo 23-1). *Suplemento ComputerWorld #876. Gestión Documental*. Retrieved marzo 2009, from <http://www.idg.es/computerworld>

Larman, C. (2003). *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. Prentice Hall.

García Rubio, F. Ó. *Metodologías de desarrollo de software*. S/A.

Zabala. (31 de marzo de 2008). *Ingeniería de Software*. Recuperado el marzo de 2009, de <http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html#PUyMSF>

Thomson, W. (2003)

Anónimo. (2007-2008). *Asignatura: Ingeniería de Software 1*. Retrieved enero 2009, from Título: Introducción a la Ingeniería de Software.: <http://teleformacion.uci.cu>

García, J. ( 2005, Mayo 7). *IngenieroSoftware*. Retrieved from <http://www.ingenierosoftware.com/analisisydiseño/uml.php>

Ramos Rodríguez, R. S. (2008). *Trabajo de Diploma: Análisis y Diseño del Sistema de Ayuda para la planificación docente en la facultad 3*. Ciudad Habana.

Helade. (2008, octubre 13). *Aplicaciones Empresariales.com*. Retrieved from "Magnolia, Gestión de Contenido Empresarial": <http://www.aplicacionesempresariales.com/magnolia-gestion-de-contenido-empresarial.html>

Fuente, T. (2008, octubre). *Alfresco: gestión documental y gestión de contenidos web*. Retrieved from <http://wiki.alfresco.com>

Duplá del Moral, A. (1997). *Manual de archivos de oficina para gestores: Comunidad de Madrid*. Madrid: Ediciones Jurídicas y Sociales.

ZUKOWSKI, J. (2003). *Java 2*.

Ricardo J. Vargas Del Valle, J. P. [www.di-mare.com](http://www.di-mare.com).: <http://www.di-mare.com/adolfo/cursos/2007-2/pp-3capas.pdf>

Lago, R. (Abril de 2007). [www.proactiva-calidad.com](http://www.proactiva-calidad.com). Recuperado el 8 de 5 de 2008, de [www.proactiva-calidad.com: http://www.proactiva-calidad.com/java/patrones/mvc](http://www.proactiva-calidad.com/java/patrones/mvc)

(n.d.). Retrieved from Multimedia: <http://www.slideshare.net/diegodalceropfw-docmanager>

Ivar Jacobson, Grady Booch, James Rumbaugh. *El Proceso Unificado de desarrollo de Software*. 2000. <http://bibliodoc.uci.cu/pdf/reg00060.pdf>

Duchain, M. (1985). *El respeto de los fondos en archivística: principios teóricos y problemas prácticos*. París.

Durand Martínez, L., & Becerra González, O. (Junio, 2008). *Análisis y Diseño de un Sistema para la Generación de Reportes*.

IEEE. (1993). *IEEE Standars Collection: Software Engineering*.

José Escalona, M., & Koch, N. (2002, 12). *Ingeniería de Requisitos en Aplicaciones para la WEb. Un estudio comparativo*. Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla. Retrieved 2009, from Ingeniería de Requisitos en Aplicaciones para la WEb. Un estudio comparativo. Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla.: <http://www.lsi.us.es/docs/informes/LSI-2002-4.pdf>.

POTENCIER, F. Z. (2008). *Symfony, la guía definitiva*. Obtenido de <http://www.librosweb.es/symfony>

Martínez, P., Ruiz Díaz, P., & Waisbrot, S. (2008). *Manual de CodeIgniter en Español*. Obtenido de <http://www.conocimientovirtual.edu.co>

## Bibliografía Consultada

Hernández León, R. A., & Coello González, S. (2002). *EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTÍFICA*. Ciudad de la Habana.

Canós, J. H., Letelier, P., & Penadés, M. C. (n.d.). *Métodologías Ágiles en el Desarrollo de Software*. Retrieved marzo 1, 2009, from <http://www.willydev.net/descargas/prev/TodoAgil.pdf>

Codina, L. (2003, Mayo). *El Profesional de la Información*. Retrieved marzo 1, 2009, from <http://www.elprofesionaldelainformacion.com/>

García Rubio, F. Ó. *Metodologías de desarrollo de software*. S/A.

Cáceres, P., & Marcos, E. (2000). *Procesos Ágiles para el desarrollo de aplicaciones web*.

Calzado, V. (2003, febrero-marzo 23-1). *Suplemento ComputerWorld #876. Gestión Documental*. Retrieved marzo 2009, from <http://www.idg.es/computerworld>

Ivar Jacobson, Grady Booch, James Rumbaugh. *El Proceso Unificado de desarrollo de Software*. 2000. <http://bibliodoc.uci.cu/pdf/reg00060.pdf>

Duchain, M. (1985). *El respeto de los fondos en archivística: principios teóricos y problemas prácticos*. París.

Durand Martínez, L., & Becerra González, O. (Junio, 2008). *Análisis y Diseño de un Sistema para la Generación de Reportes*.

Anónimo. (2007-2008). *Asignatura: Ingeniería de Software 1*. Retrieved enero 2009, from Título: Introducción a la Ingeniería de Software.: <http://teleformacion.uci.cu>

Comité de Normas de descripción. (2000). *ISAD(G) Norma Internacional General de Descripción Archivística*. Madrid.

Comité de Normas de Descripción Canberra. (2004). *Norma Internacional sobre los Registros de autoridad de archivos relativos a instituciones, personas y familias*. Madrid.

Reyes Serrano, K., & Rodríguez González, Z. (2004). *Estudio de la producción documental de la Asamblea Provincial del Poder Popular de Holguín y la Asamblea Municipal del Poder Popular de Colón*. La Habana.

Mena Mugica, L. M. (2002). *Propuesta de organización del archivo de gestión de la Dirección de Registro Civil y Notarías del Ministerio de Justicia*. Ciudad de La Habana.

Bonal Zazo, D. J. *EL DOCUMENTO ELECTRÓNICO Y EL ARCHIVO*. Universidad de Extremadura: Facultad de Biblioteconomía y Documentación.

Sierra Escobar, L. F. (2004). *Cómo identificar y denominar una serie documental: Propuesta metodológica*. Colombia.

Bustelo Ruesta, C. (15-04-2005). *Estrategias y metodología de diseño e implementación de sistemas de gestión de documentos: ISO 15489 y AS-5090-2003*. Madrid.

Monzón, A. *Calidad de la Especificación: ¿Se pueden medir los Requisitos?*

B. Bernárdez, A., & Durán Toro, M. (2004). UNA PROPUESTA PARA LA VERIFICACIÓN DE REQUISITOS BASADA EN MÉTRICAS. *Revista de Procesos y Métricas de las Tecnologías de la Información (RPM)*, España.

## Anexos

### *Anexo 1 Herramientas de Gestión Documental*

#### *DocManager*



**Figura 1.5 Logo herramienta DocManager**

Portal Framework DocManager (PFW DocManager) es un gestor documental que posibilita el almacenamiento, clasificación, búsqueda, descarga, distribución y actualización de documentos en diferentes formatos (imágenes, textos, vídeos, hojas Excel...) mediante una web visual de fácil usabilidad y máxima accesibilidad manteniendo un exhaustivo control de acceso, mediante roles de usuarios, y actualización de contenidos, gracias a la existencia de un archivo histórico con las versiones anteriores.

Este gestor documental garantiza y conserva la documentación mediante el almacenamiento de un archivo histórico atendiendo a las necesidades del ciclo de vida de cualquier documento. Además de establecer un riguroso control de acceso a la información mediante el establecimiento de permisos para diferentes roles de usuarios finales.

DocManager es un módulo adicional a la plataforma de gestión de contenidos y documental. Portal Framework Enterprise Content Manager se trata de una solución que mejora y proporciona un valor añadido a las aplicaciones ya contenidas en ECM dotando a la gestión documental de un sistema gráfico a través de un acceso web.

Proporciona un repositorio virtual con acceso web, de máxima usabilidad y accesibilidad, que revoluciona el complejo proceso de gestión documental para transformarlo en una cuestión sencilla buscando una finalidad práctica y útil desde la perspectiva del usuario final y el administrador de contenidos. Esto le permitirá mejorar los flujos de información dentro de su organización y disponer siempre de todas las versiones de un documento, evitando problemas de pérdidas de los documentos, fallos del sistema o pérdida de las versiones anteriores.

DocManager, debido a su desarrollo con Portal Framework ECM, posee **características** como:



**Gestión de usuarios y roles:** Permite la creación de áreas privadas a las que únicamente se puede acceder mediante unas claves de usuario y contraseña facilitadas por el administrador.



**Gestión de workflows:** La canalización de procesos de workflows posibilita disponer de un sistema de validación que asegure que los contenidos publicados sean correctos y puedan ser visionados por el resto de los usuarios.



**Gestión de ficheros y contenidos:** Un portal web requiere un gran número de ficheros y contenidos; por este motivo Portal Framework ECM ofrece un sólido sistema de gestión de ficheros y contenidos. Además se incluye un editor de texto enriquecido e imágenes para realizar sencillas modificaciones desde los elementos subidos a la aplicación.



**Buscador:** A consecuencia de la integración de Portal Framework DocFinder, PFW DocManager está dotado de una herramienta de búsqueda rápida y eficaz que permite realizar búsquedas de contenido tanto en el propio portal web como en el back-office del mismo.



**Programación de nuevas funciones:** A pesar de que PFW DocManager se ofrece como una solución estándar, el gestor de contenidos es altamente personalizable y pueden generarse y programarse un gran número de nuevas funcionalidades, donde el cliente es el que pone el límite.

DocManager brinda numerosas **ventajas** entre las que se encuentran:

- Acceso descentralizado gracias a su difusión online permitiendo su uso desde diferentes ubicaciones y soportes.
- Control férreo de acceso a la documentación mediante la existencia de permisos dependiendo de diferentes roles de usuarios.
- Preservación de la información mediante el almacenamiento de un archivo histórico con las



versiones anteriores.

- Flexibilidad de almacenamiento y clasificación ya que permite creación de carpetas y subcarpetas.
- Presentación lógica, coherente y predecible que posibilita un manejo sencillo y accesibilidad máxima.
- Módulo dinámico que permite personalizar el entorno gráfico y configuración en multiidiomas.
- Completo sistema de gestión documental probado en entornos de alta concurrencia y grandes volúmenes de información. (Concatel Concatel)

## **Nuxeo**



**Figura 1.6 Logo herramienta Nuxeo**

Nuxeo es una plataforma que se creó con el objetivo de gestionar contenidos, basado en los últimos estándares y componentes Java bajo una licencia LGPL.

Nuxeo permite una sencilla gestión de versiones y es utilizada para implementar procesos de administración de documentos, gestión de conocimientos, aplicaciones de flujo de trabajo, en fin, Nuxeo es la solución ideal para manejar grandes volúmenes de millones de documentos.

Presenta una estructura orientada a servicios, es extremadamente fácil de configurar, y puede ser integrado y desarrollado sin ningún problema en un sistema de información habitual.

En cuanto a la usabilidad como atributo de calidad que mide la facilidad de usar las interfaces Web, Nuxeo posee una interfaz fácil de usar, proporciona Nuxeo WebEngine como creador de portales web de forma totalmente gratuita además de ser Open Source. Gracias a esta herramienta, se pueden crear portales orientados al usuario sin renunciar a todas las ventajas que ofrece la gestión documental.

Nuxeo disfruta de una aplicación de escritorio para manejar el contenido almacenado de forma rápida, se trata de Nuxeo RCP.

Como funcionalidad específicamente en la utilización del Drag & Drop que no es más que la

capacidad para arrastrar contenidos y para crear contenido, Nuxeo es 100% estable.

Con la herramienta Nuxeo se puede utilizar edición online o edición en vivo permitiendo editar documentos de Microsoft Office, Open Office y otros formatos habituales trabajando directamente con el navegador bajo una gran cantidad de configuraciones: Linux o Windows (XP y Vista) + Firefox o Explorer + Office o Open Office. Además, cada uno de ellos con varias versiones (Office 2003 y 2007, Firefox 2.X y 3.X, Explorer 6 y 7... etc.).

Este Gestor Documental permite al usuario "taggear" (etiquetar) contenido para más adelante poder navegar en la información utilizando estas etiquetas, esta navegación se denomina "virtual" facilitando por ejemplo: navegar por los contenidos de diferentes países y luego cambiar a navegación de carpetas por "temática" por ejemplo imágenes de Salud por un lado, y por otro de Deporte.

Nuxeo basa su búsqueda en herramientas estándares y abiertas como Lucene y Open Search. Estos buscadores indexan la totalidad de los documentos (título y contenido total) en los formatos más habituales, y resuelven las búsquedas en menos de 1 segundo al estilo "Google". (Blog Yerbabuena)

**Anexo 2 Diagrama de Actividades**

**Diagrama de actividades del CUN: Gestionar las actividades del grupo.**

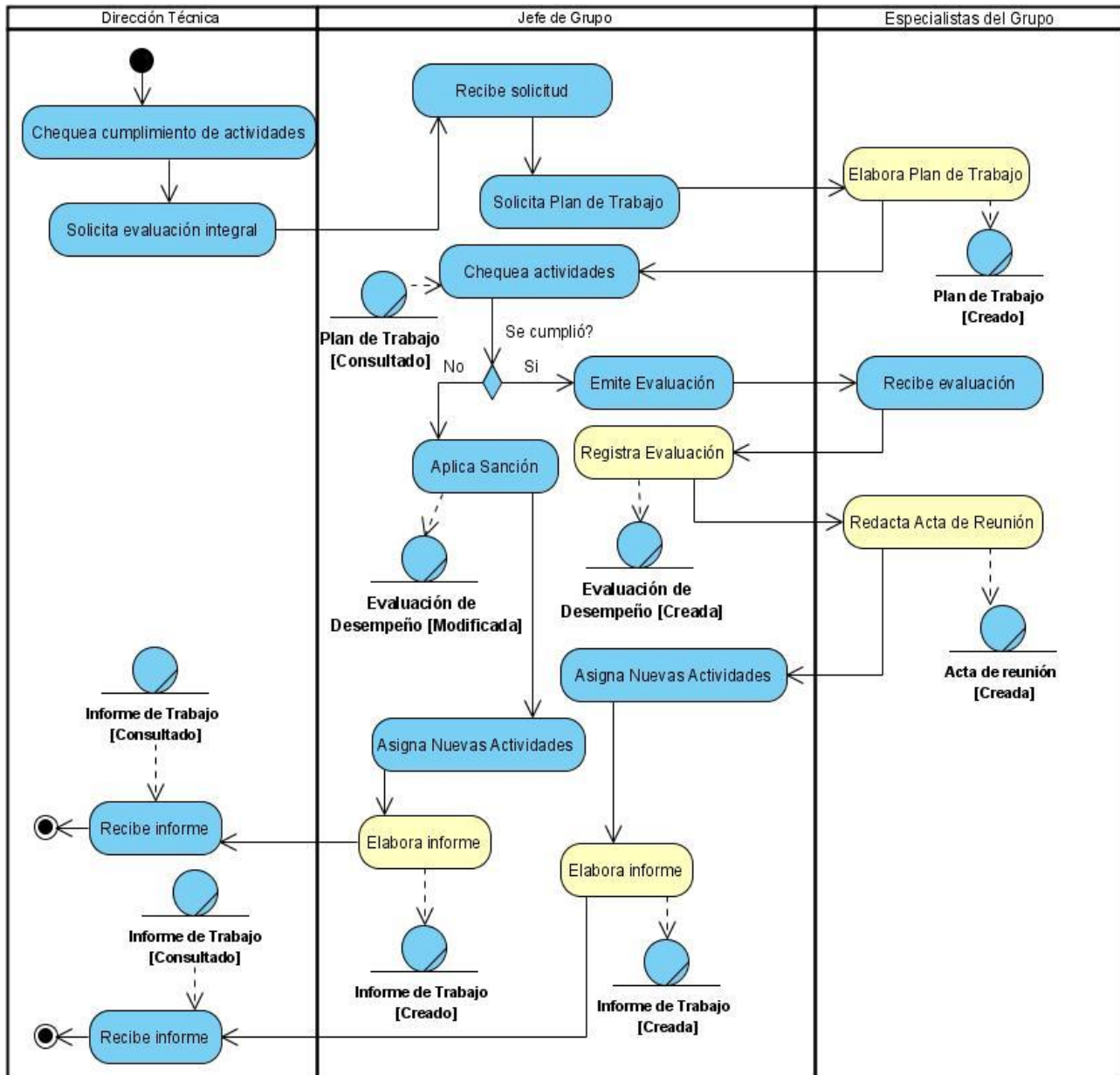


Diagrama de actividades del CUN: Gestionar formación y superación a los AI

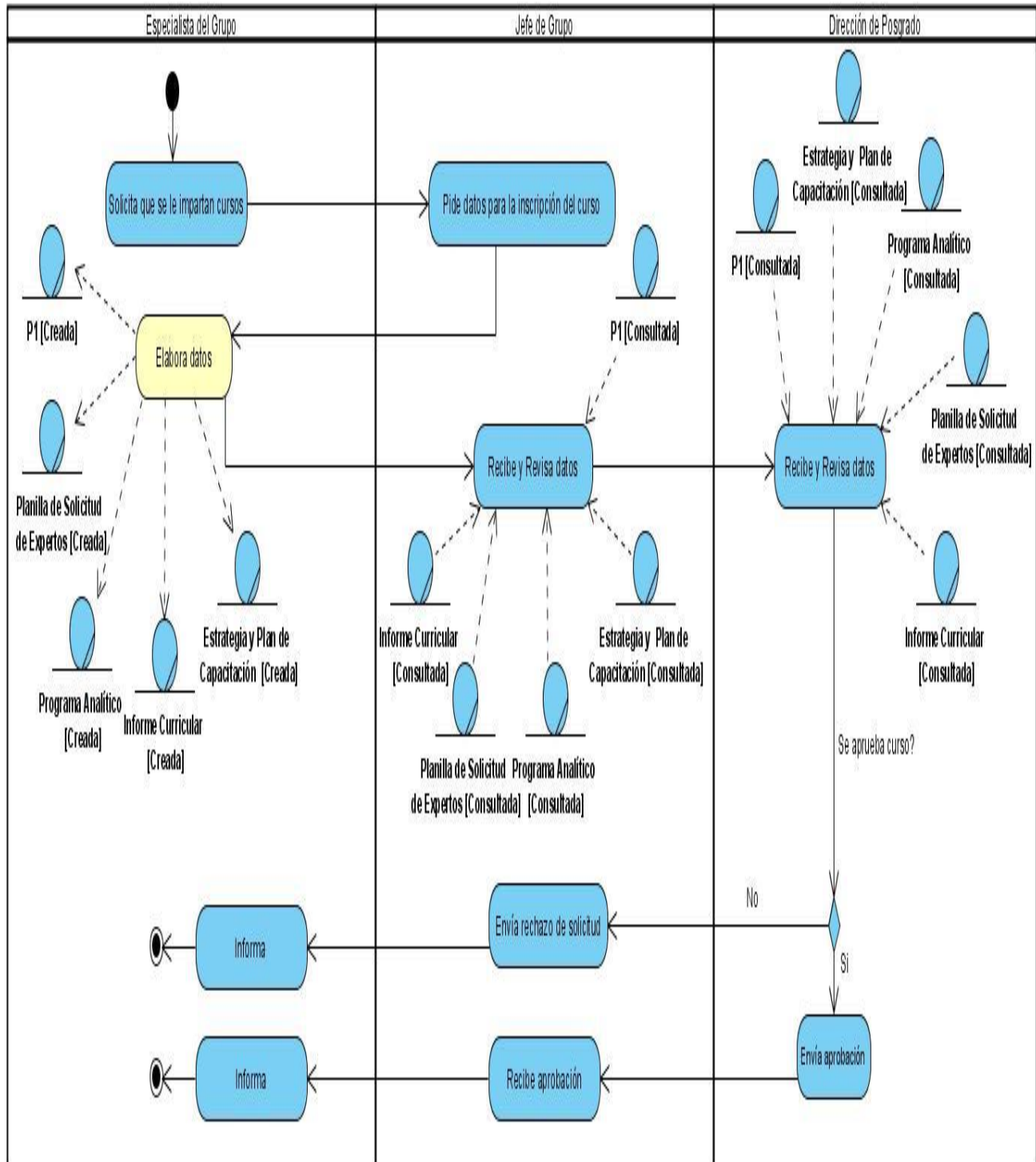


Diagrama de actividades del CUN: Definir normativas de trabajo para las UP.

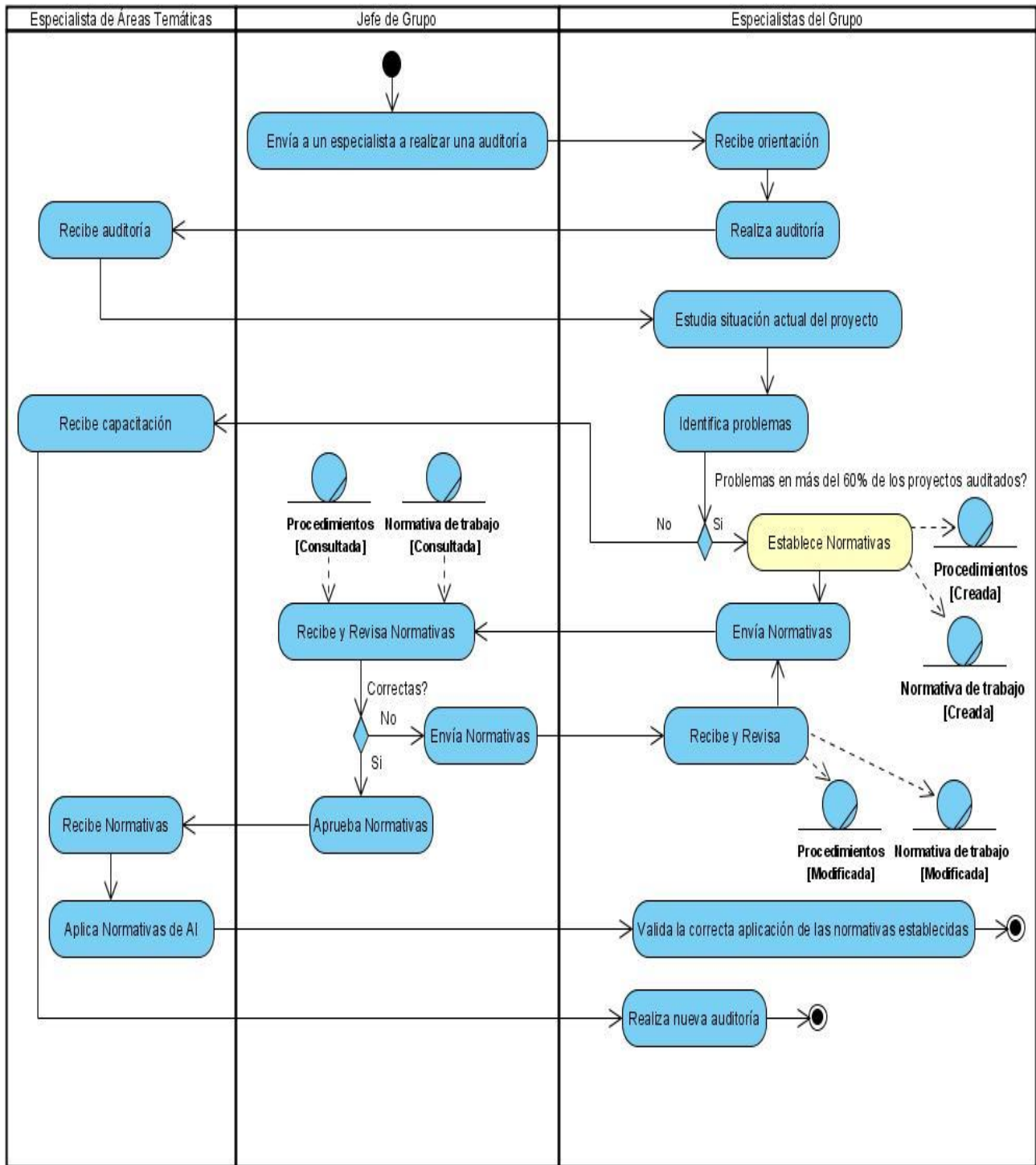
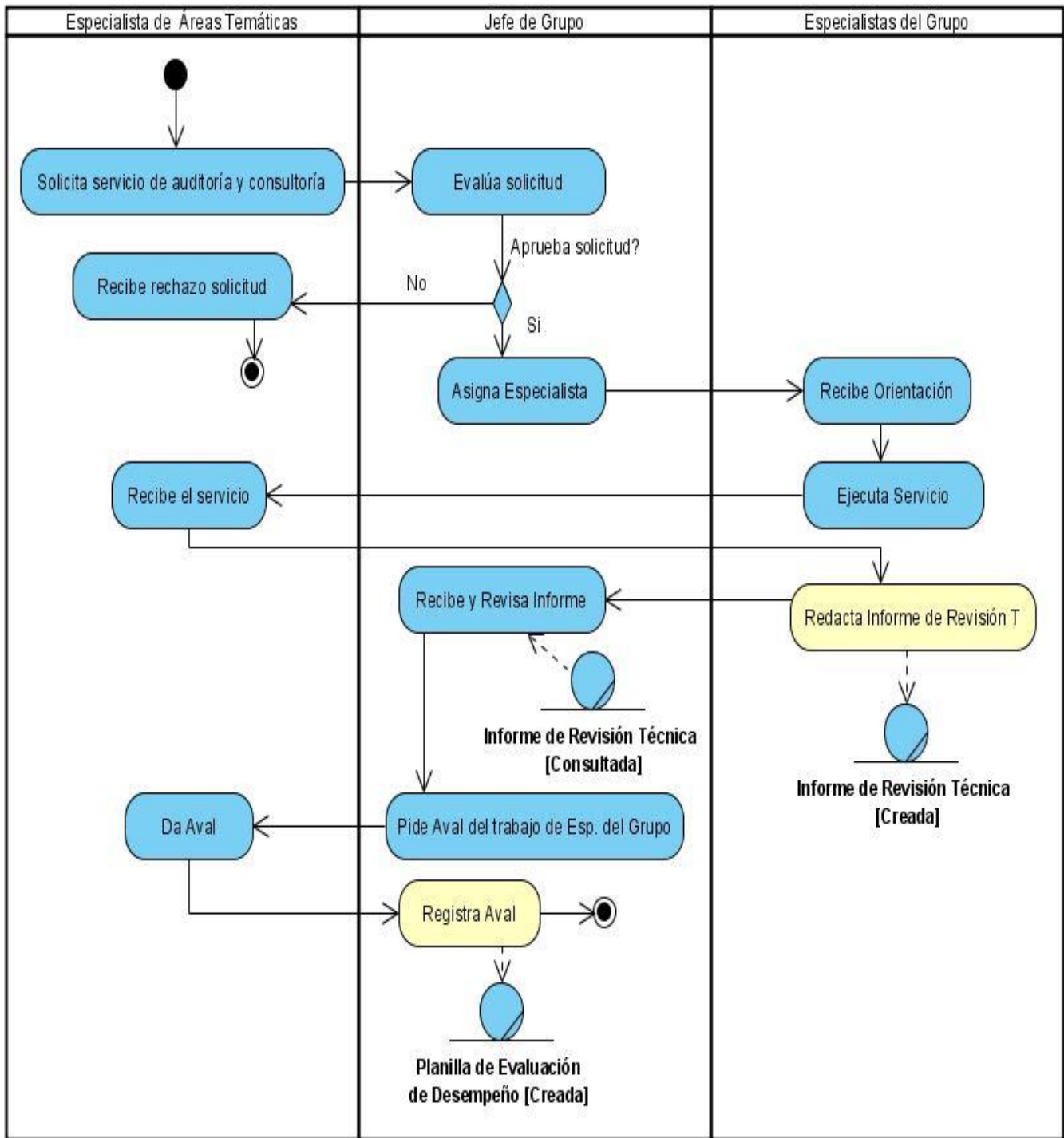


Diagrama de actividades del CUN: Brindar servicios de consultoría y asesoría a UP.



### **Anexo 3 Glosario de Términos**

#### **A**

**Artefacto de software:** Cualquier cosa que resulte del proceso de desarrollo de software; por ejemplo: especificaciones, diseños, software, etc.

**AI:** Arquitectura de Información

Arquitecto Información: Especialista en la disciplina de Arquitectura de Información.

**A&EI:** Grupo de Arquitectura y Estándares de Información.

#### **C**

**CU:** Caso de Uso

**CUN:** Caso de Uso del Negocio

**CUS:** Caso de Uso del Sistema

#### **D**

**DCA:** Diagrama de Clases del Análisis

**DCD:** Diagrama de Clases de Diseño.

**DCS:** Diagrama de Casos de Uso del Sistema.

**DT:** Dirección Técnica

#### **E**

**ECM:** Enterprise Content Management.

#### **H**

Herramientas CASE: Grupo de herramientas utilizadas para el desarrollo de de todo tipo de proyectos de ingeniería de software.

#### **I**

**I+D:** Grupo de Proyectos Investigación y Desarrollo

**IP:** Infraestructura Productiva

**R**

**RF:** Requisitos Funcionales

**RNF:** Requisitos No Funcionales

**RUP:** Metodología para el desarrollo de software, que en español sería Proceso Unificado de Desarrollo.

**Roles:** Para habilitar y administrar las distintas funcionalidades que pueden realizar los usuarios surgen los roles, como Lector, Creador, Editor, Revisor.

**S**

**SGD:** Sistemas de Gestión Documental

**SGDD:** Sistemas de Gestión de Documentos Digitales.

**SGBD:** Sistema de gestión de Base de Datos, es básicamente el software que permite la utilización y la actualización de los datos almacenados en una o varias bases de datos por los usuarios.

**Software:** Sistemas o Aplicaciones expresadas en un lenguaje de máquina.

**U**

**UCI:** Universidad de las Ciencias Informáticas.

**UML:** Unified Modeling Language. Lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software.

**UP:** Unidades Productivas.

**X**

**XP:** eXtreme Programming. Metodología de desarrollo de software basada en valores como simplicidad, comunicación, retroalimentación.



## **Anexo 4 Prototipos de Interfaz de Usuarios**

### **CU: Autenticar Usuario**



Autenticar Usuario

Registrarse

Usuario

Contraseña

Aceptar Cancelar

Detailed description: This is a screenshot of a software window titled 'Autenticar Usuario'. The window has a blue title bar with standard Windows window controls (minimize, maximize, close). Below the title bar, the text 'Registrarse' is displayed in blue. There are two input fields: one for 'Usuario' and one for 'Contraseña'. Below the input fields, there are two buttons: 'Aceptar' and 'Cancelar'.

### **CU: Gestionar Cuentas de Usuarios**



Gestionar Cuentas de Usuario

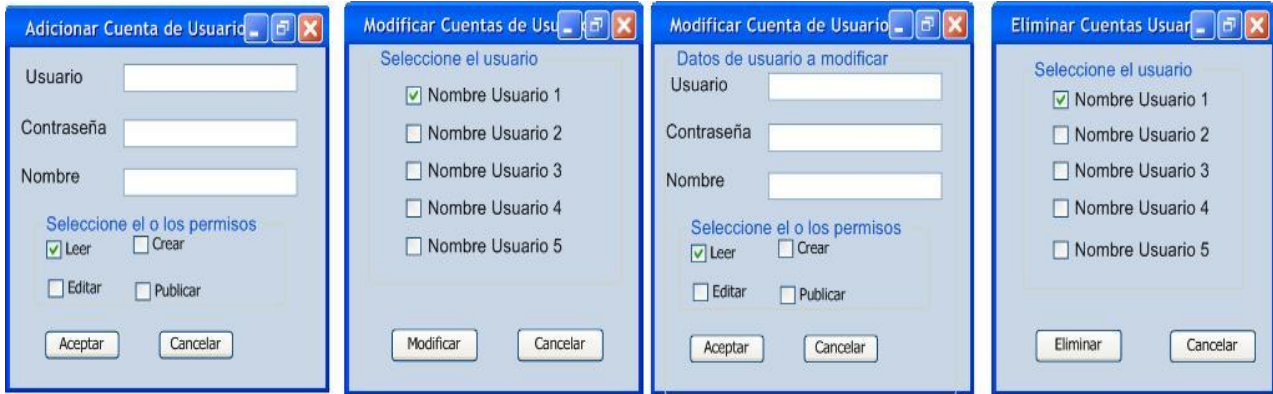
Seleccione la acción a realizar

Adicionar Cuentas de Usuarios

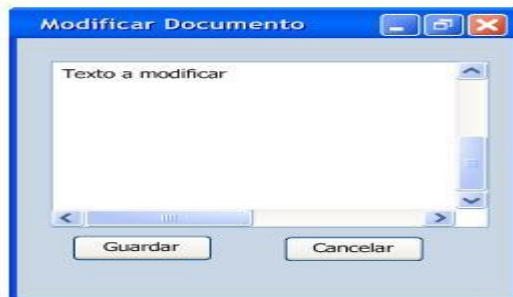
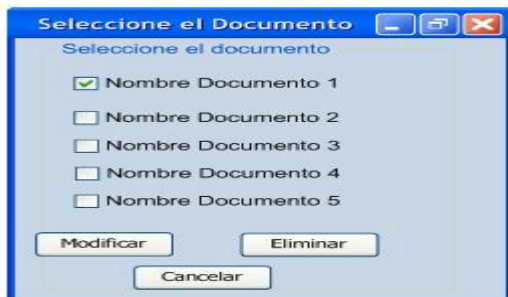
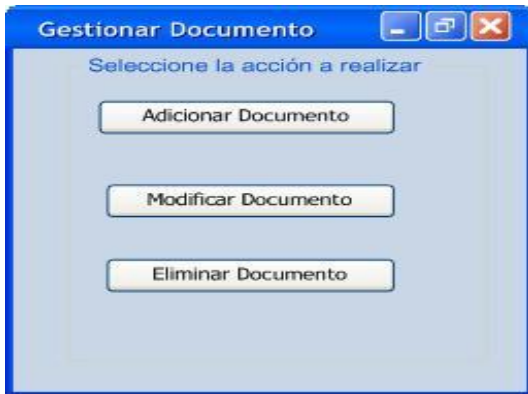
Eliminar Cuentas de Usuarios

Modificar Cuentas de Usuarios

Detailed description: This is a screenshot of a software window titled 'Gestionar Cuentas de Usuario'. The window has a blue title bar with standard Windows window controls. Below the title bar, the text 'Seleccione la acción a realizar' is displayed. There are three buttons stacked vertically: 'Adicionar Cuentas de Usuarios', 'Eliminar Cuentas de Usuarios', and 'Modificar Cuentas de Usuarios'.



**CU: Gestionar Documentos**



**CU: Gestionar Metadatos**

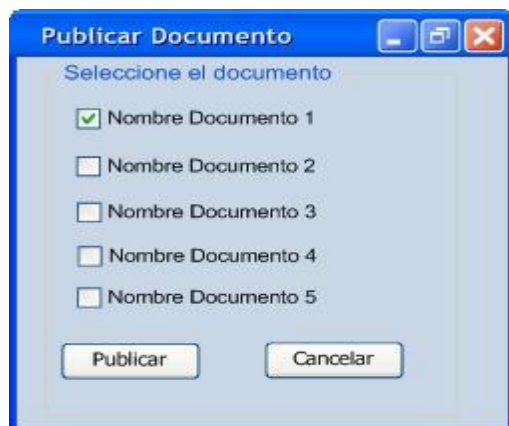
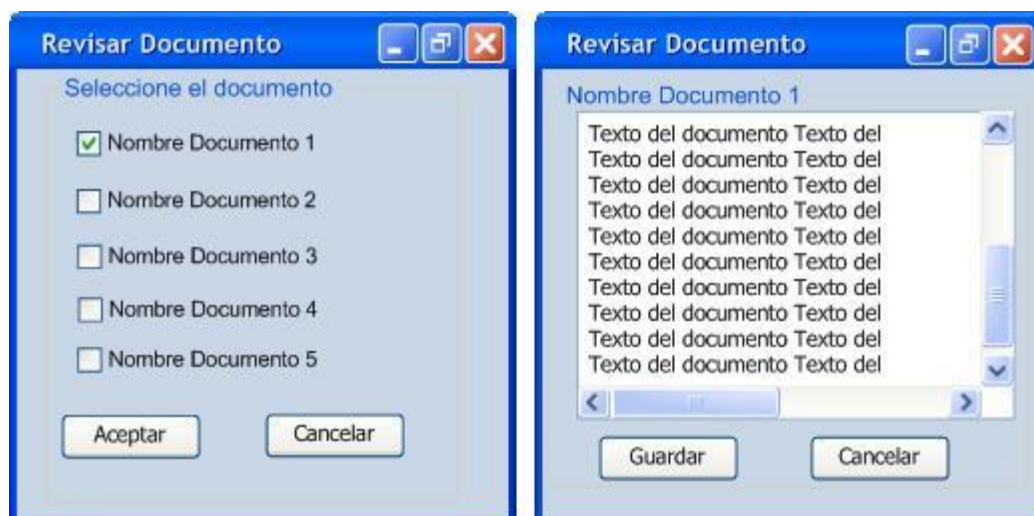


**CU: Gestionar Permisos**



**CU: Gestionar Contenido (o Plantilla)**



**CU: Gestionar Espacios****CU: Publicar Documento****CU: Revisar Documento**

**CU: Buscar Información**

Buscar Información

Criterios de búsqueda por:

Palabra

Formato

Fecha

Autor

Buscar Cancelar

**CU: Configurar Flujo de Trabajo**

Configurar F Trabajo

Seleccione el usuario

Nombre Usuario 1

Nombre Usuario 2

Nombre Usuario 3

Nombre Usuario 4

Nombre Usuario 5

Aceptar Cancelar

Configurar F Trabajo

Nombre usuario 1

Permisos

Crear  Leer

Editar  Publicar

Aceptar Cancelar

Configurar Flujo de Trabajo

Nombre usuario 1

Permisos

Seleccione el documento

Nombre documento 1

Nombre documento 2

Nombre documento 3

Nombre documento 4

Aceptar Cancelar

Configurar Flujo de Trabajo

Usuario 1

Permisos

Documento 1

Seleccione el espacio

Nombre espacio 1

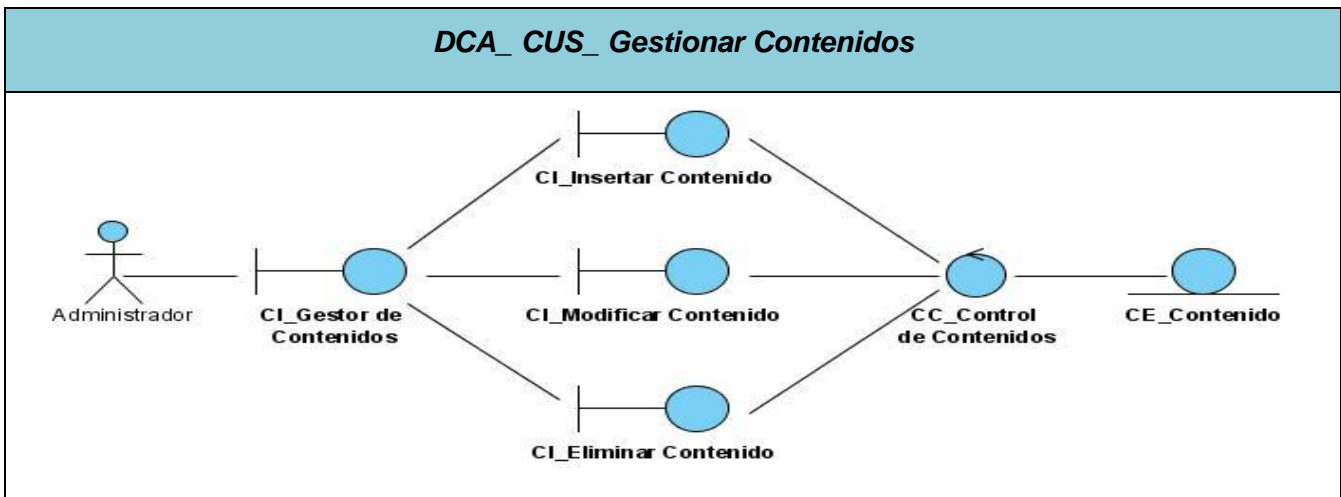
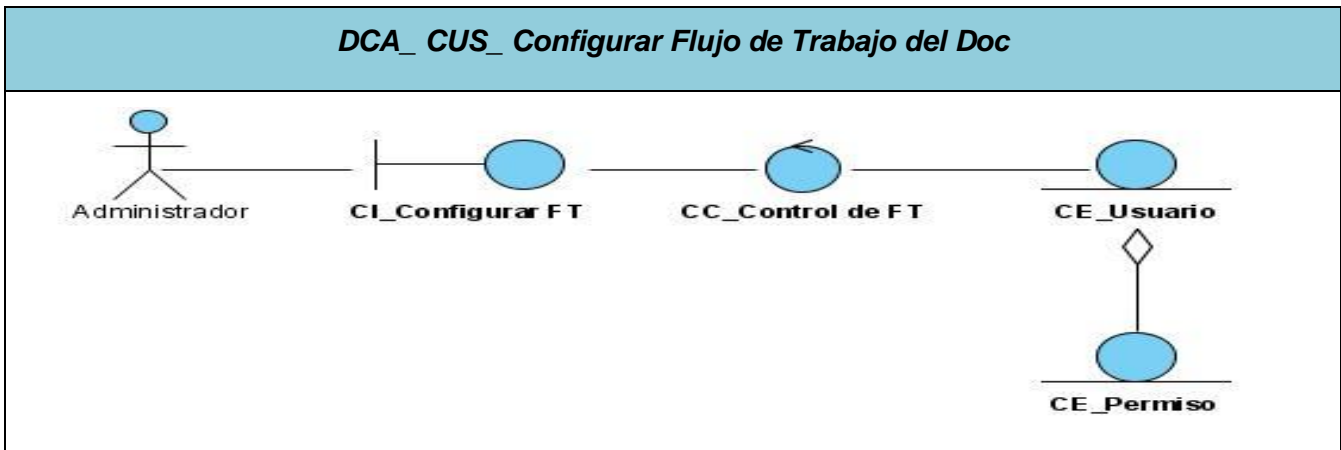
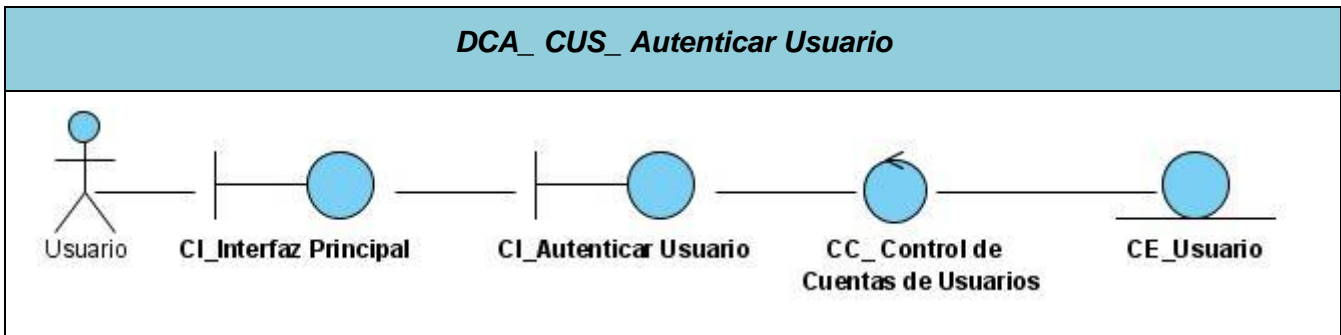
Nombre espacio 2

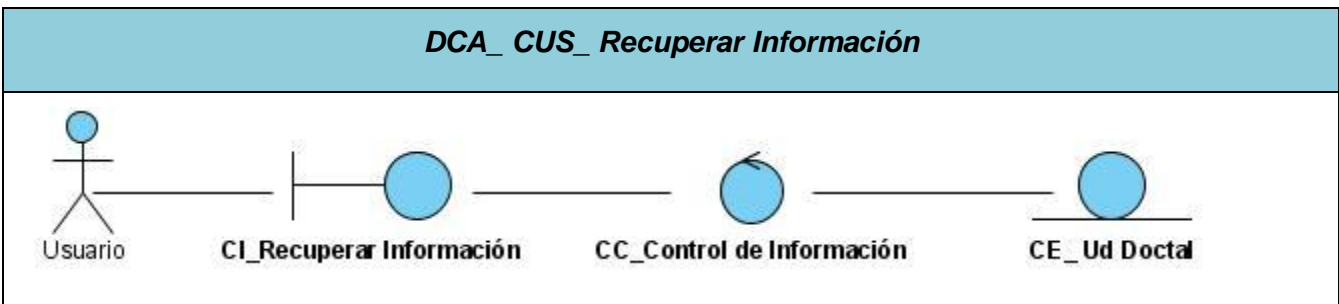
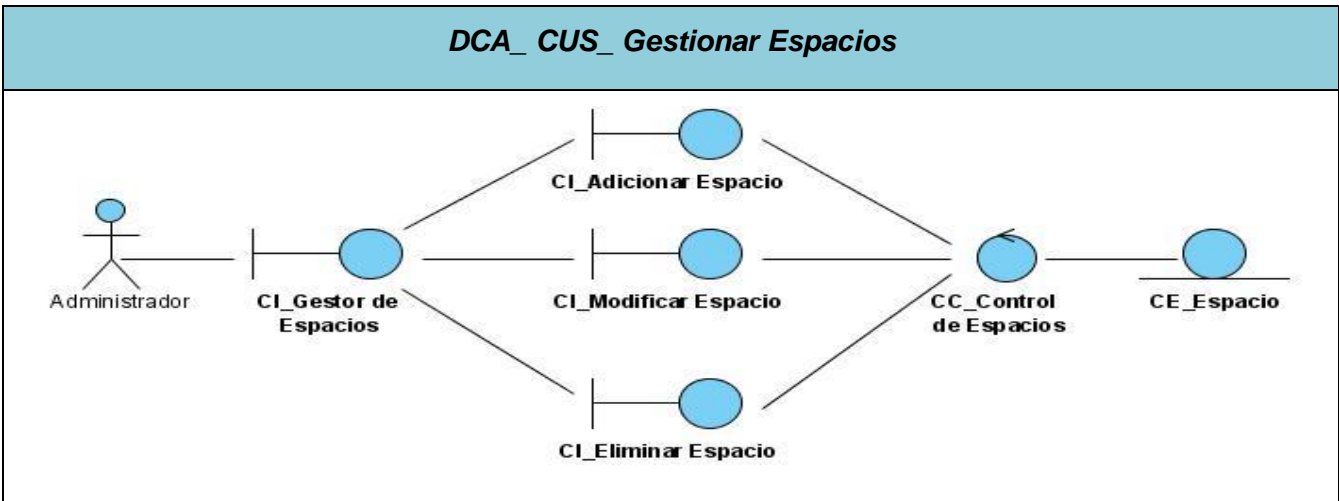
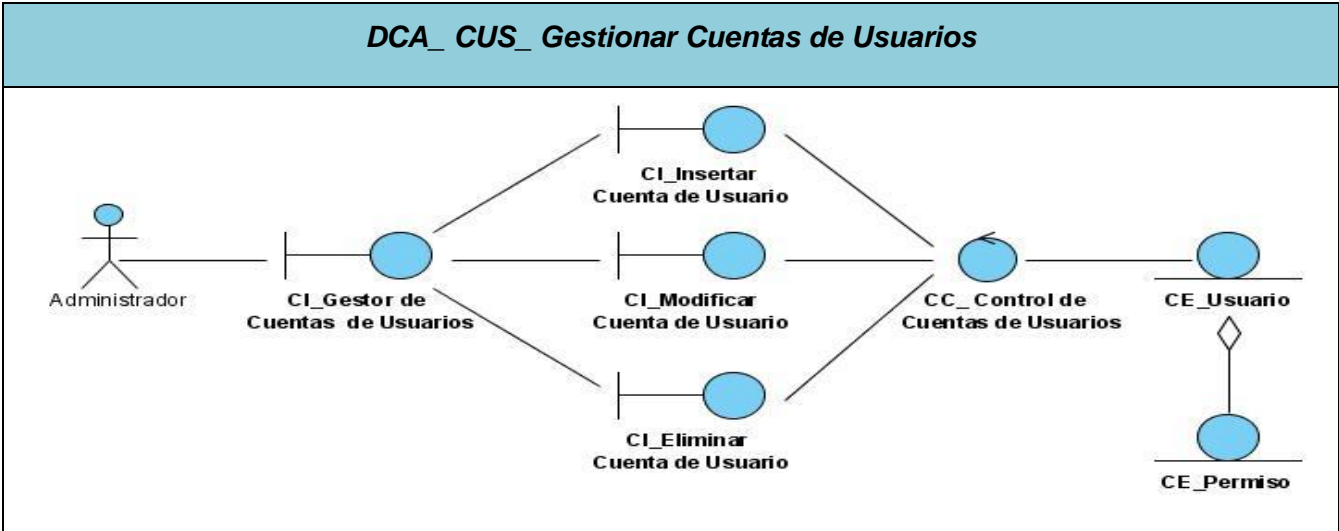
Nombre espacio 3

Dirección : \  Browse...

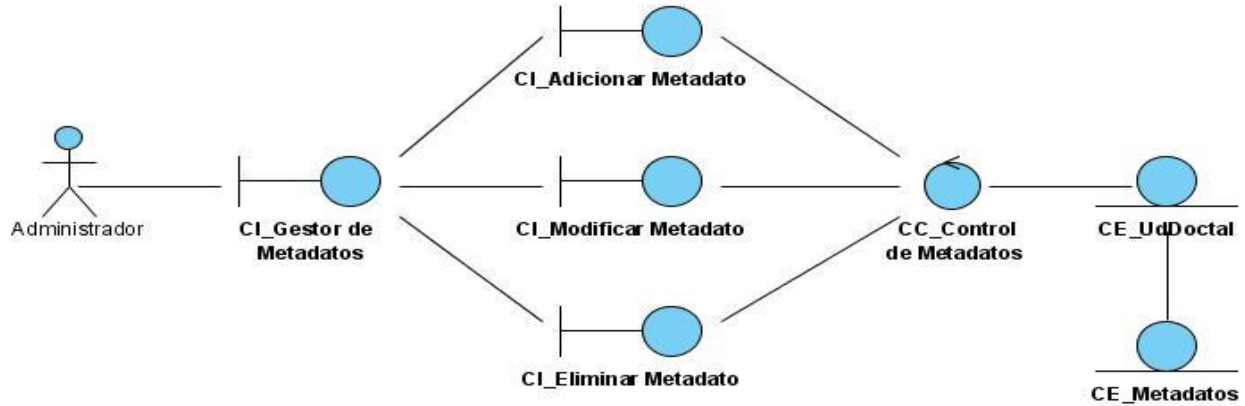
Crear Cancelar

**Anexo 5 Diagrama de Clases del Análisis**

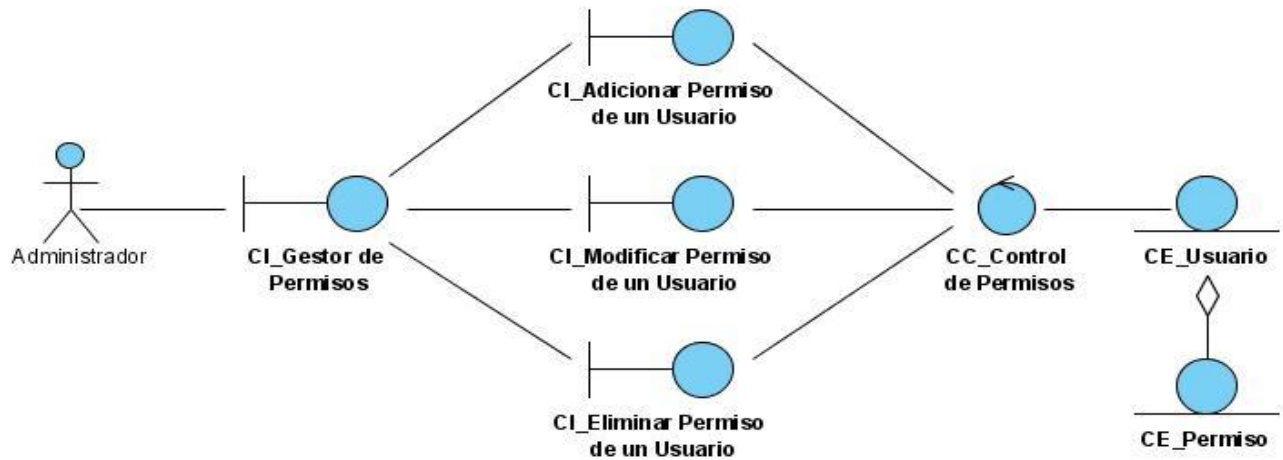




**DCA\_CUS\_Gestionar Metadatos**

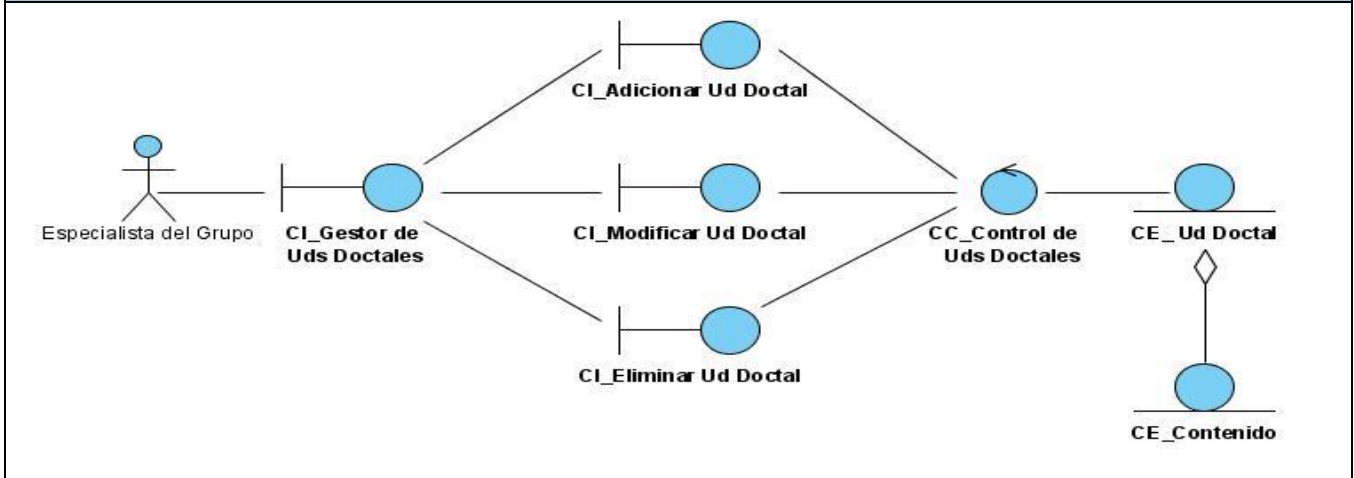


**DCA\_CUS\_Gestionar Permisos\_Usuario**





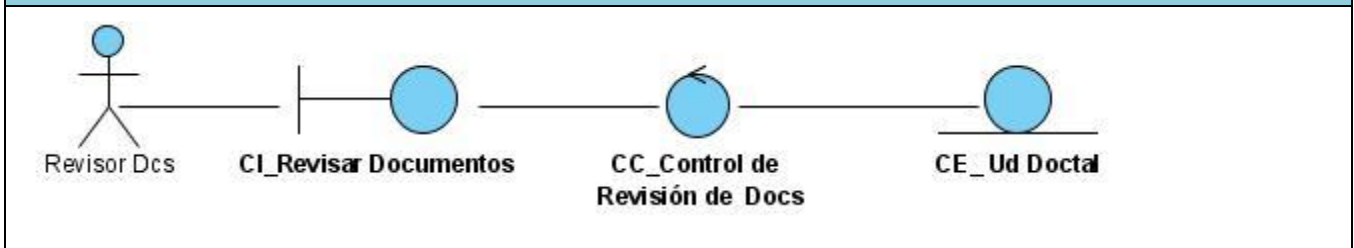
**DCA\_CUS\_Gestionar Uds Doctales**



**DCA\_CUS\_Publicar Documentos**

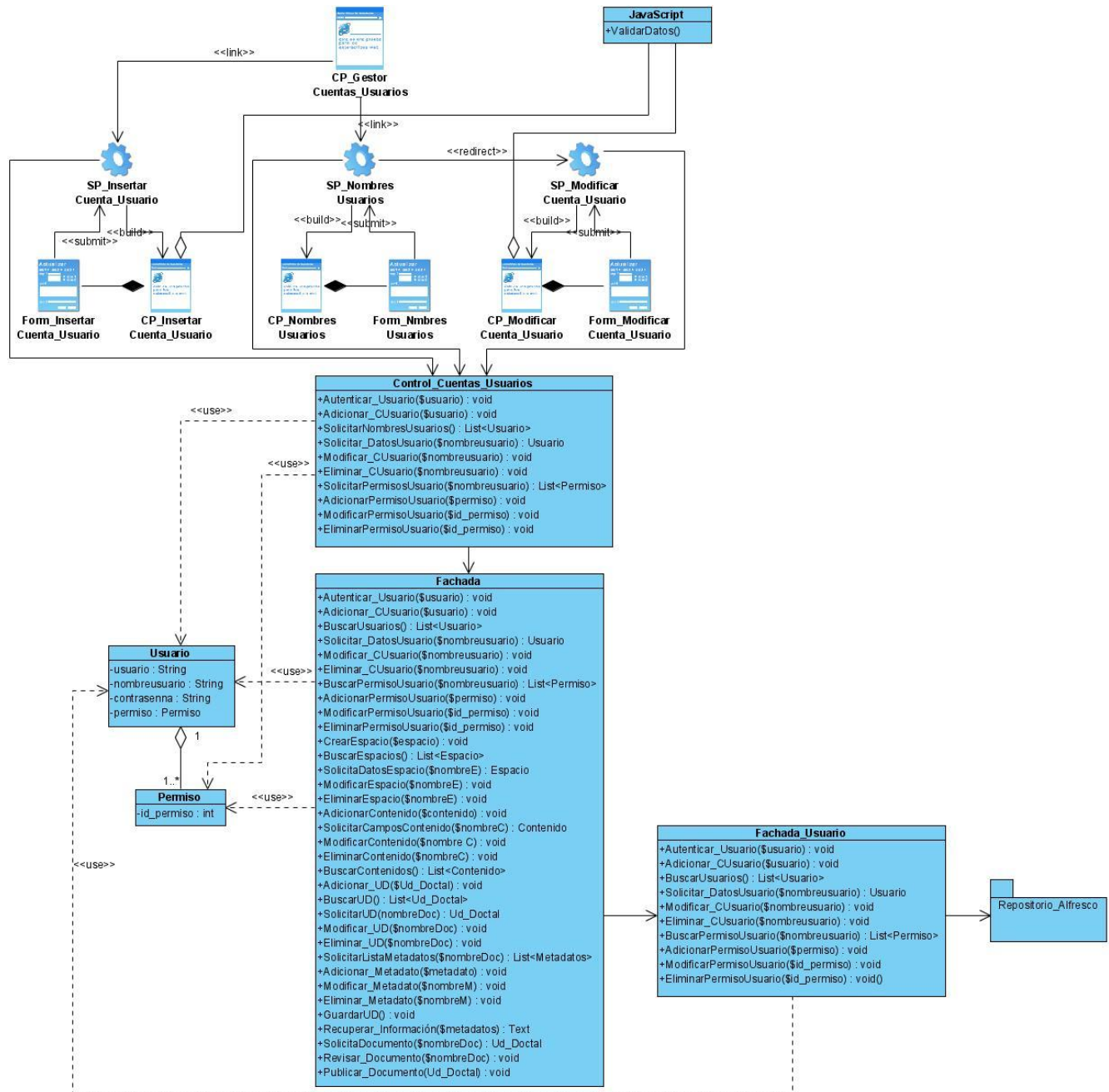


**DCA\_CUS\_Revisar Documentos**

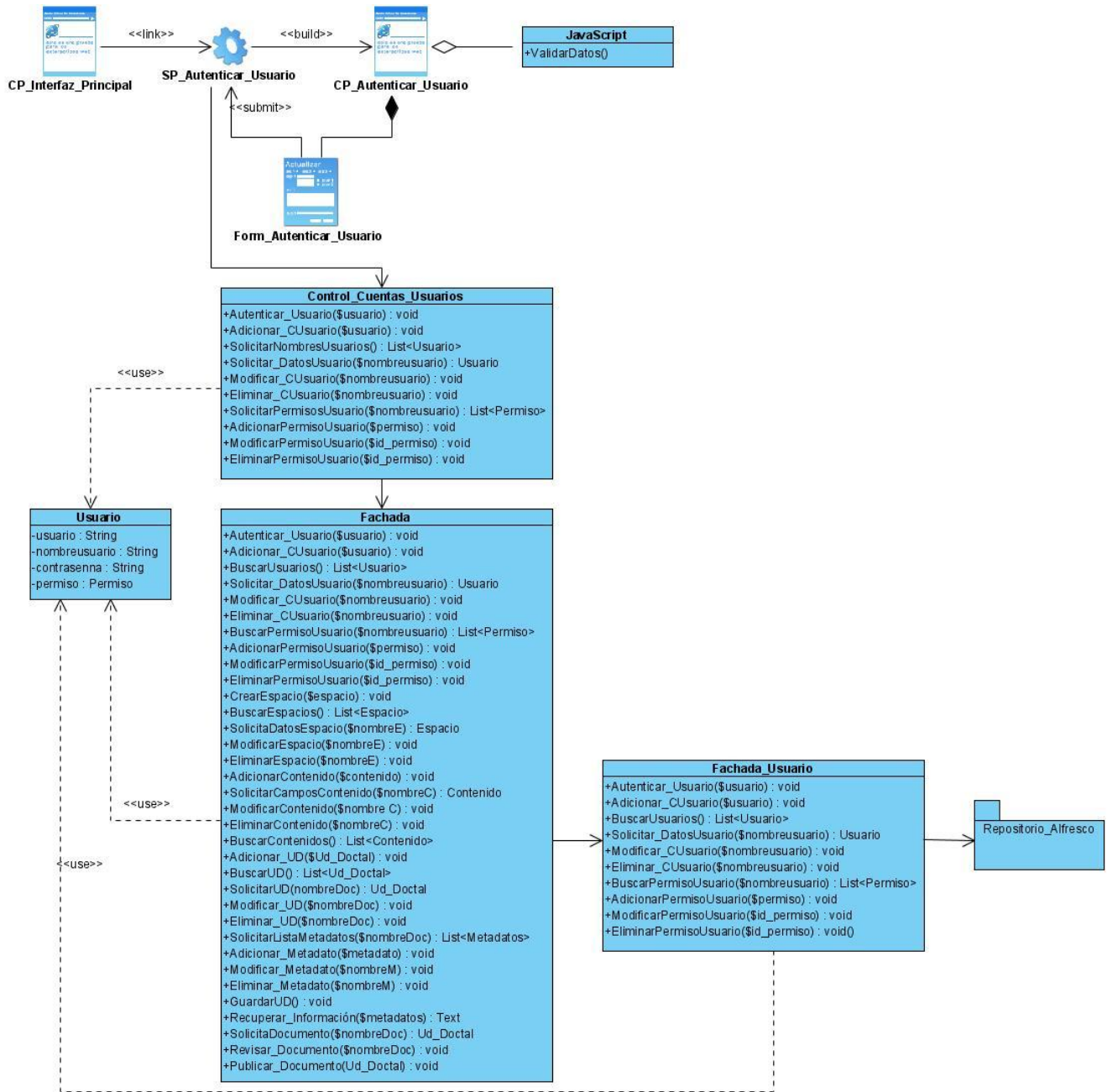


## Anexo 6 Diagramas de Clases del Diseño

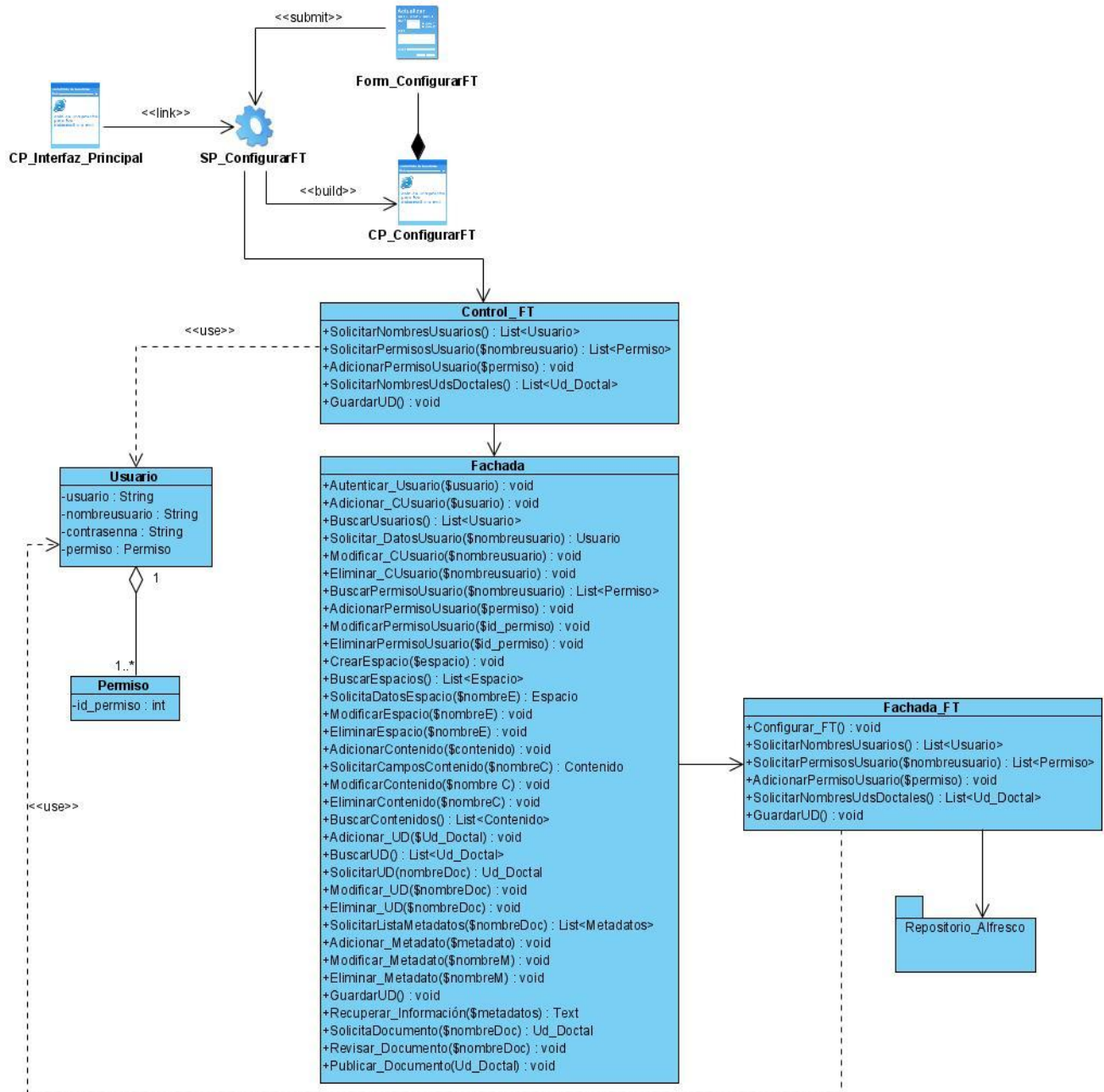
### DCD-EW\_CU\_Gestionar Cuentas de Usuarios



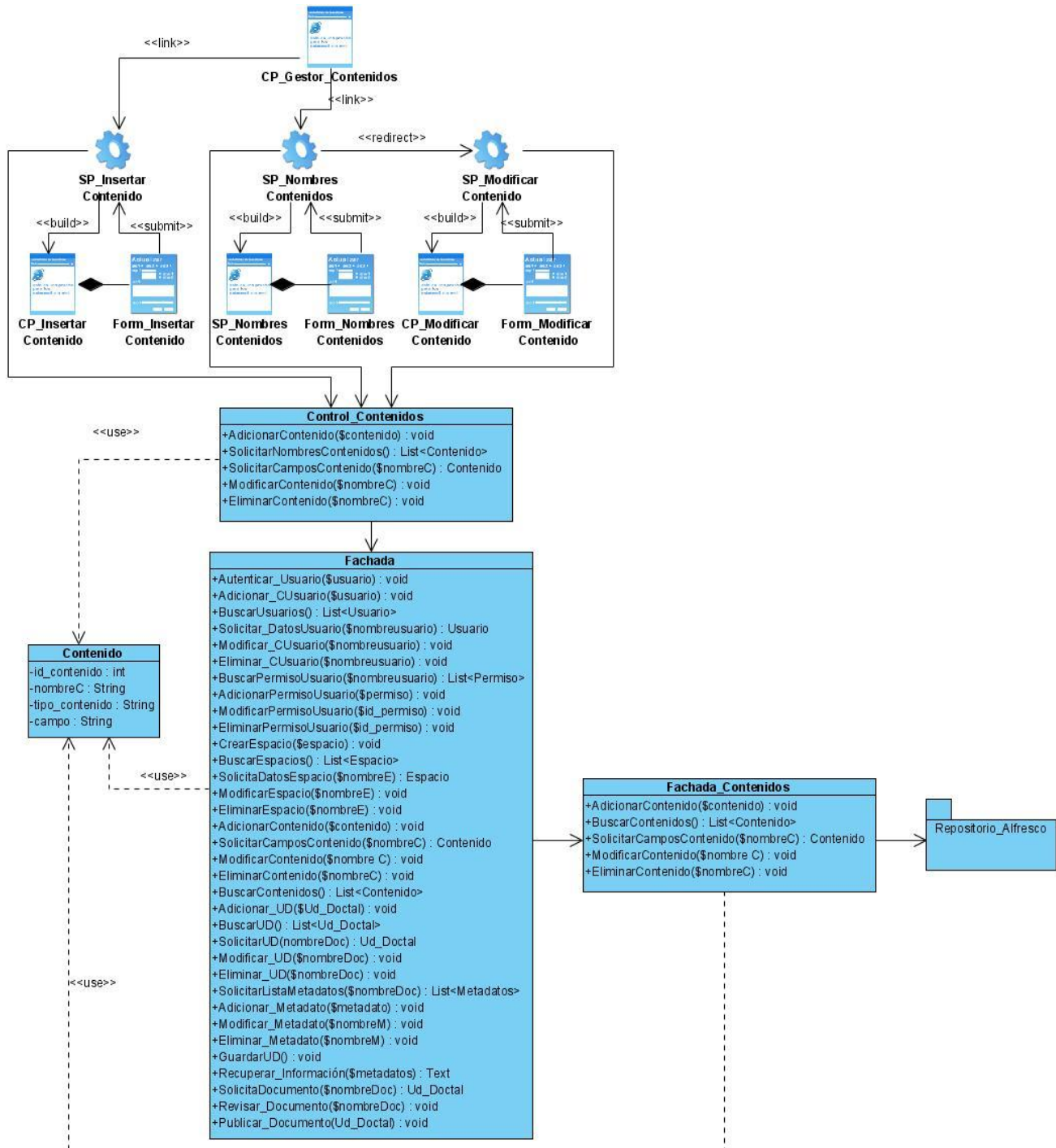
DCD-EW\_CUS\_Autenticar Usuario



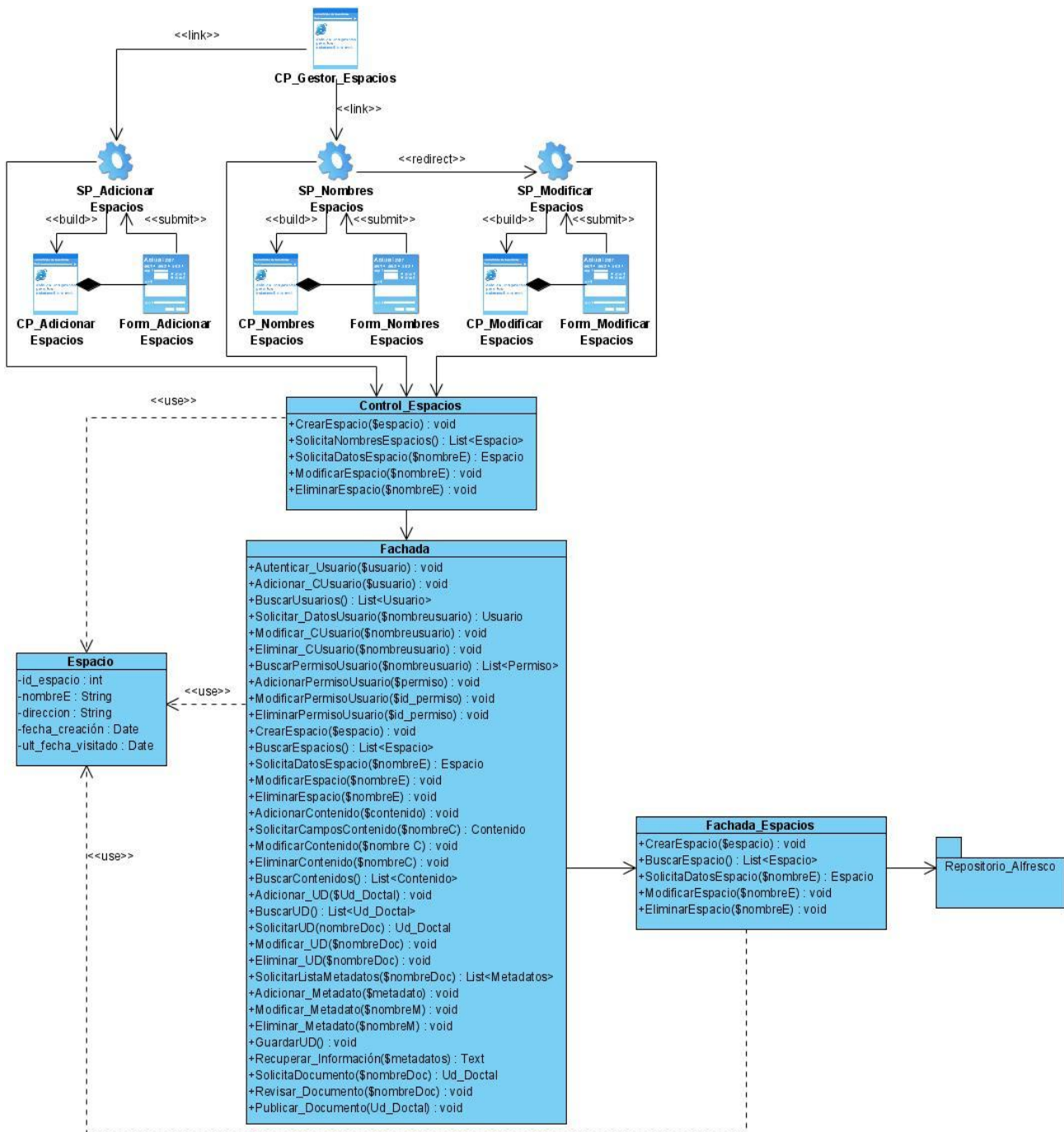
DCD-EW\_CUS\_Configurar Flujo de W del Doc



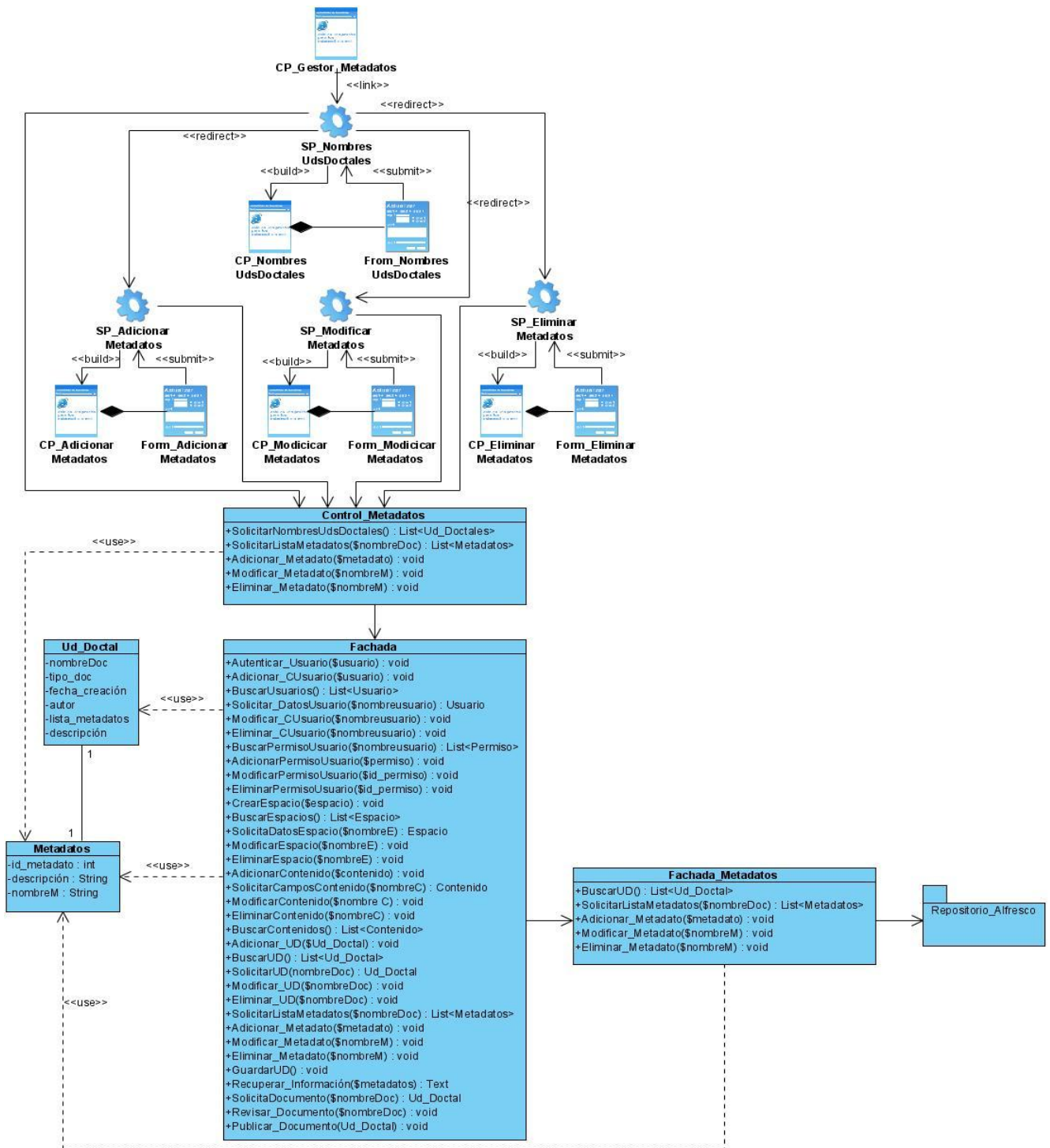
DCD-EW\_CUS\_Gestionar Contenidos



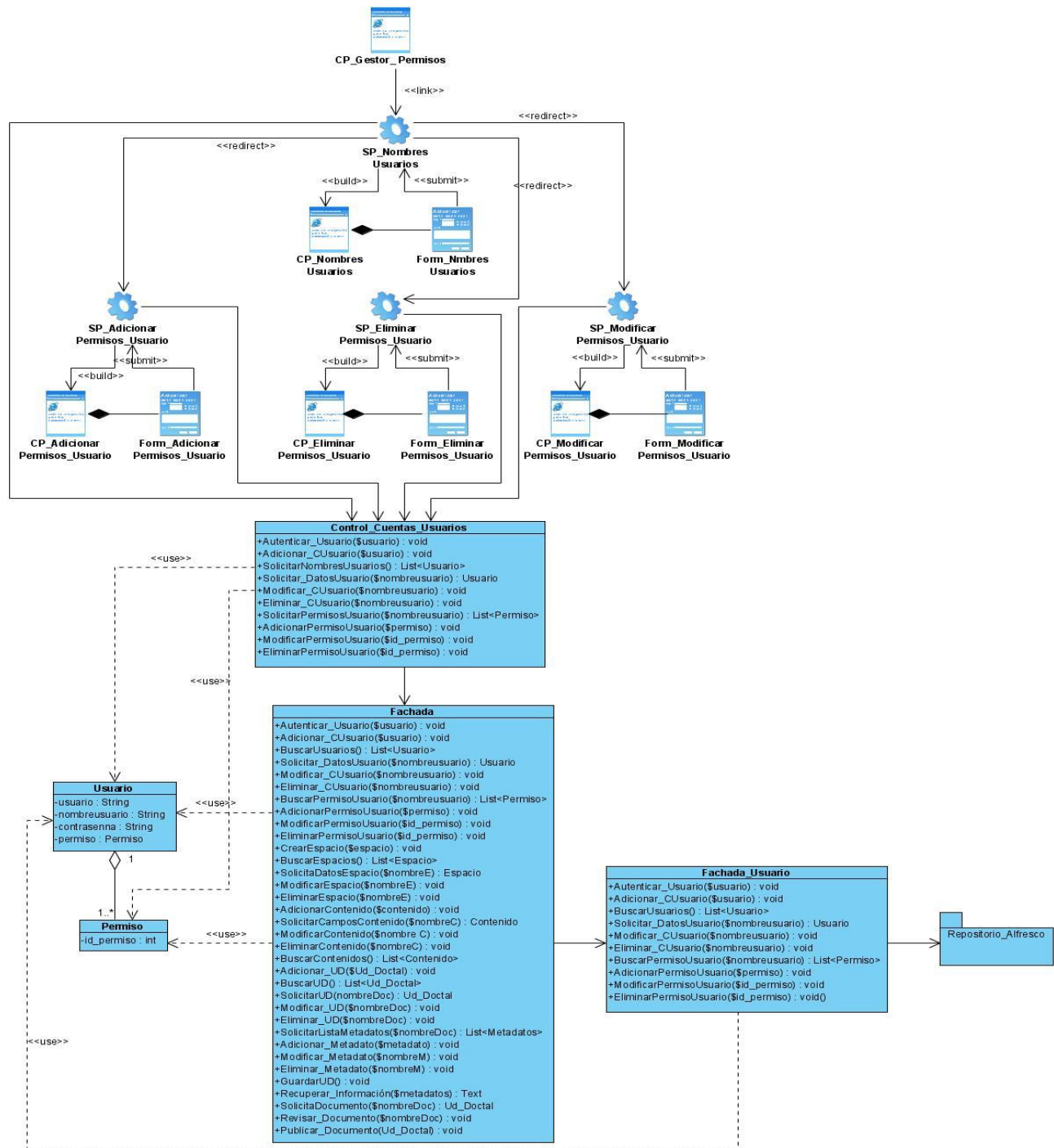
DCD-EW\_CUS\_Gestionar Espacios



DCD\_EW\_CUS\_GestionarMetadatos

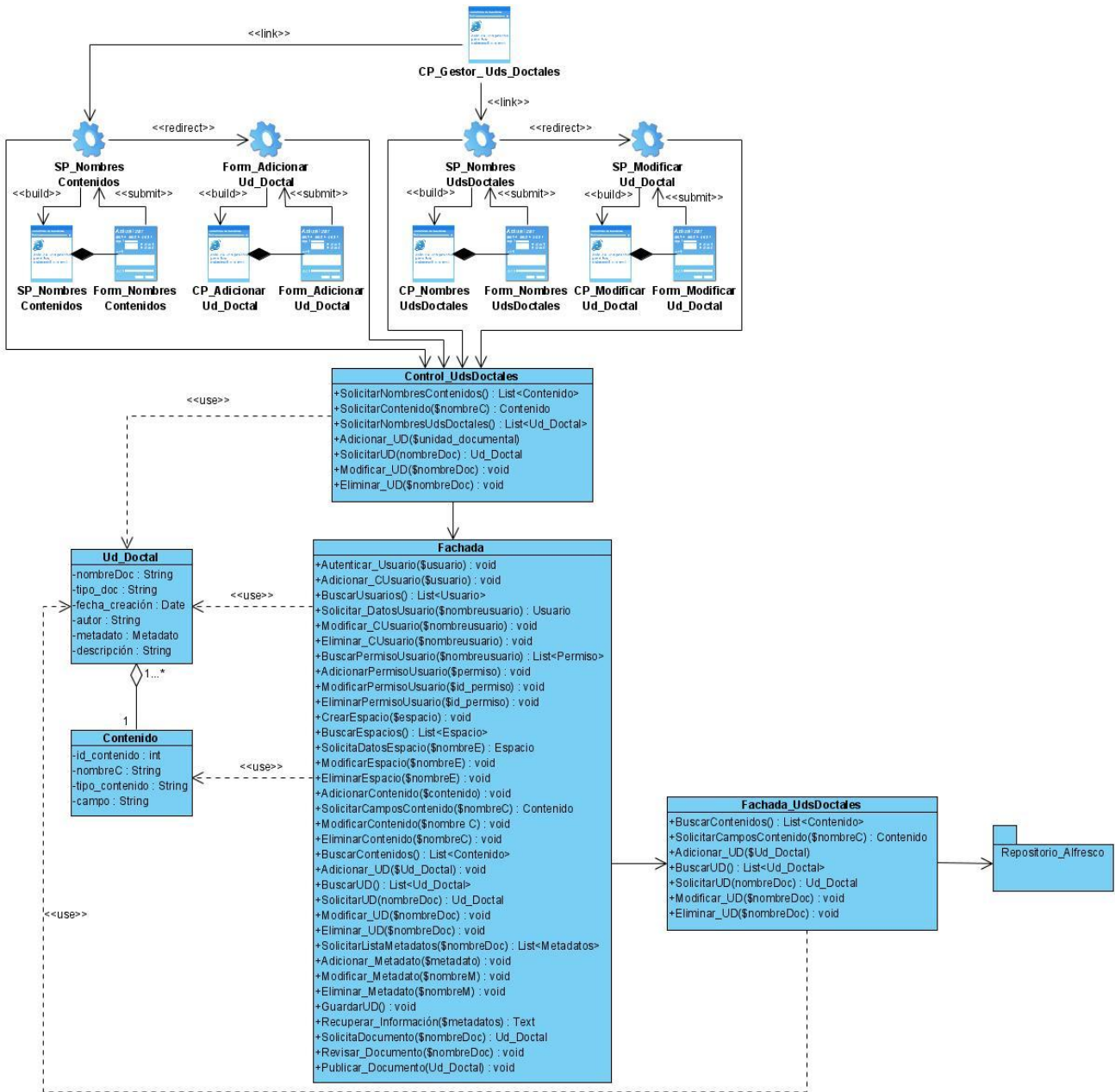


### DCD-EW\_CUS\_Gestionar Permisos\_Usuario

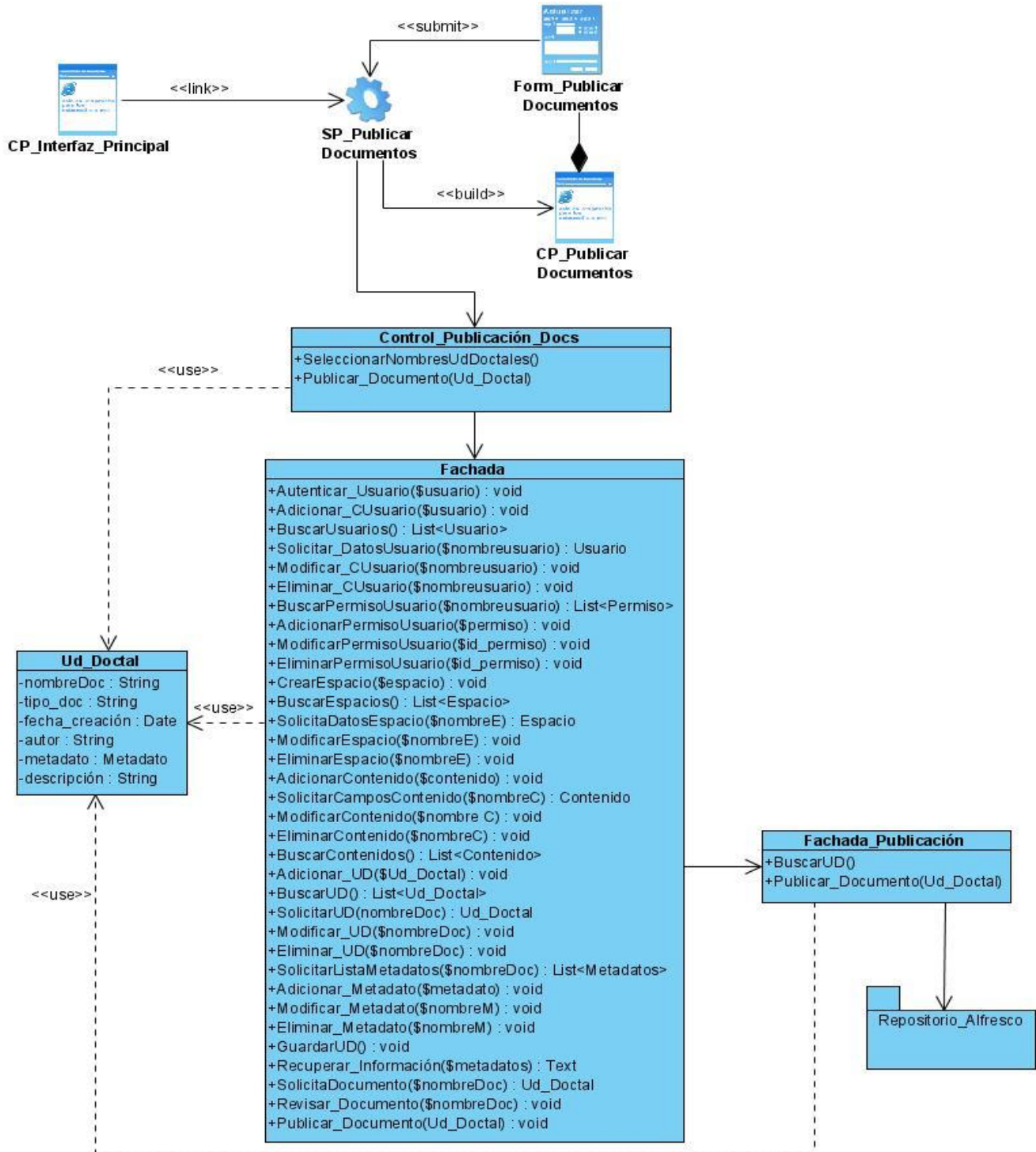




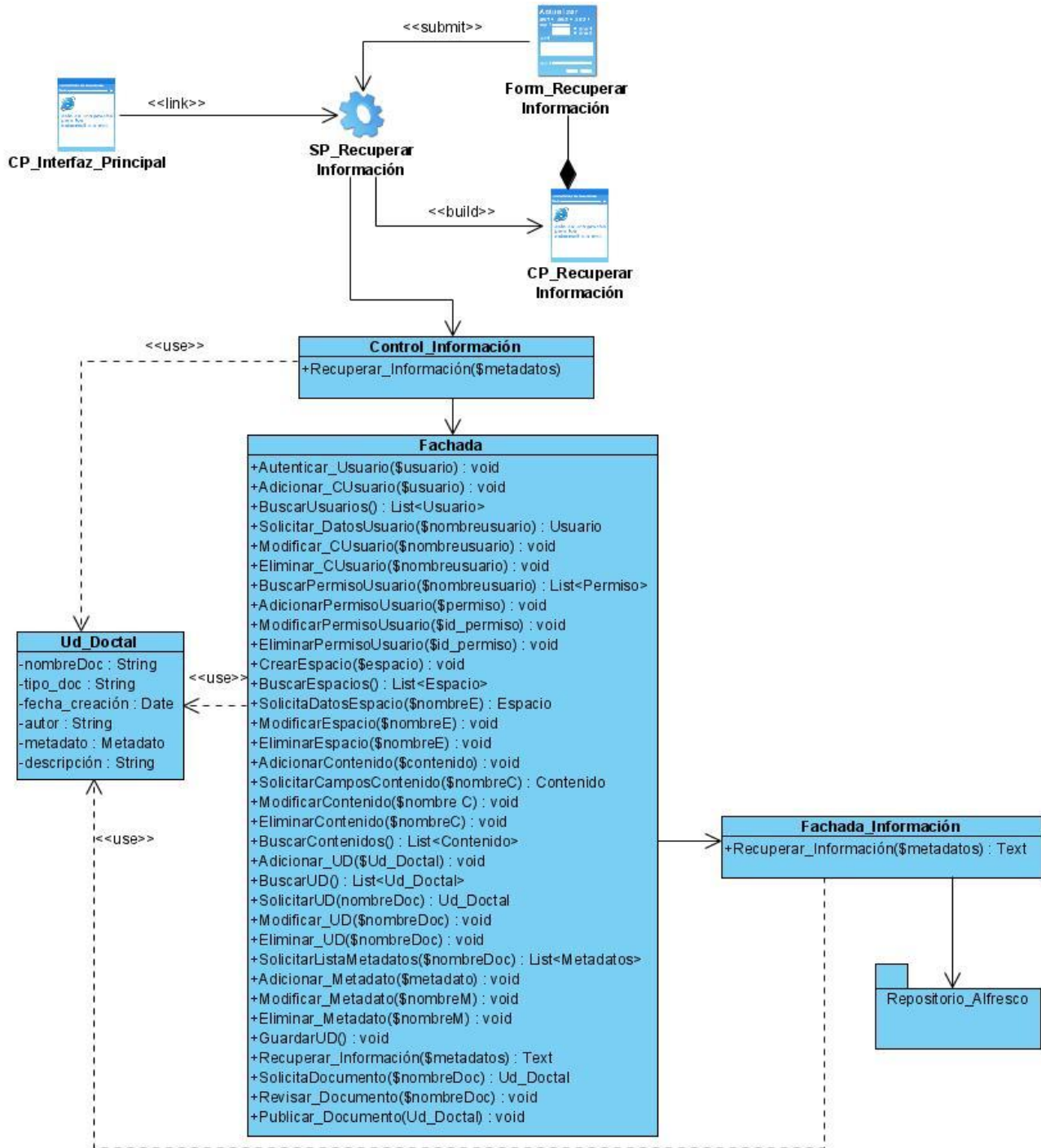
DCD-EW\_CUS\_Gestionar Uds Doctales



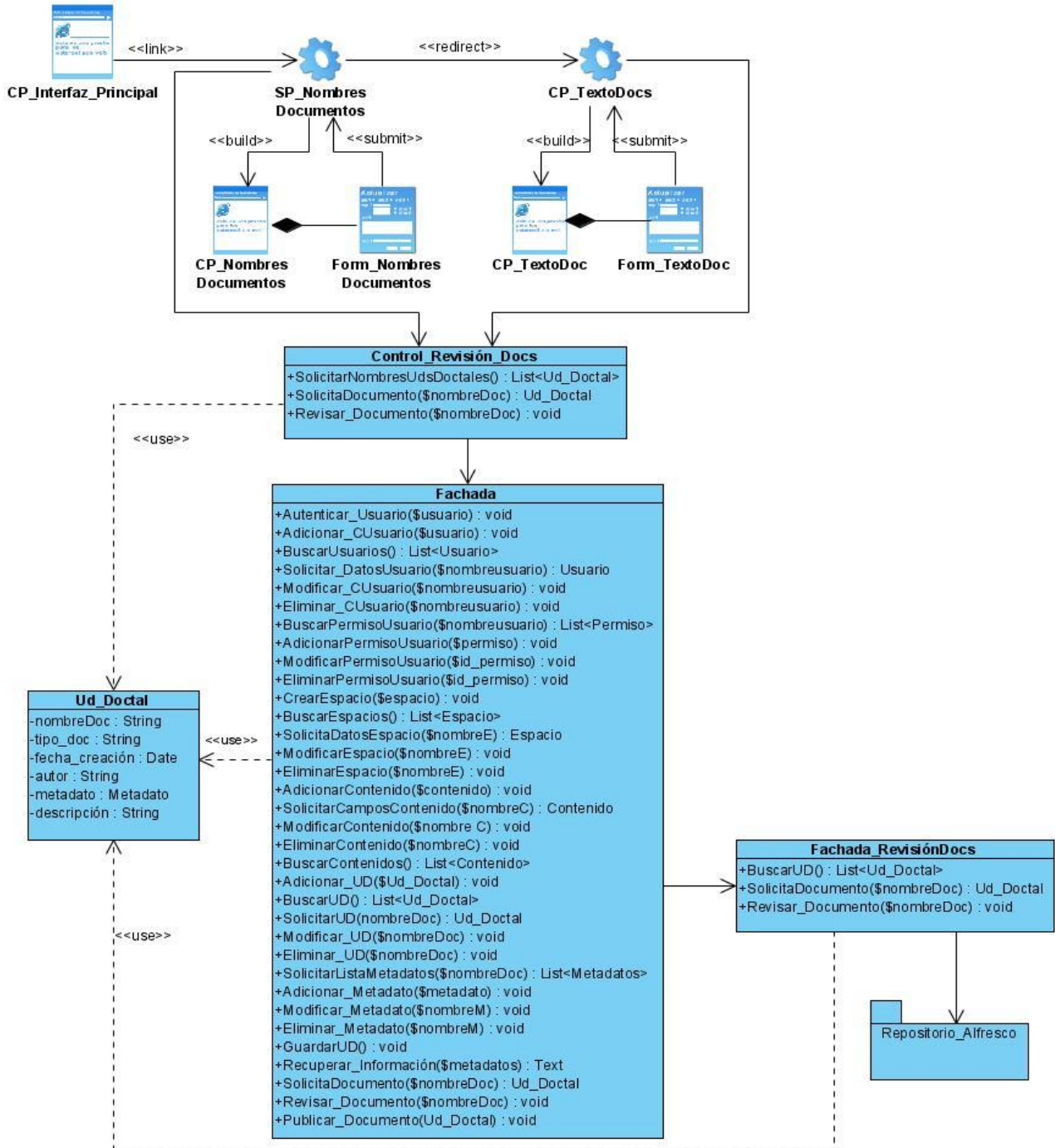
**DCD-EW\_CUS\_Publicar Documentos**



**DCD-EW\_CUS\_Recuperar Información**

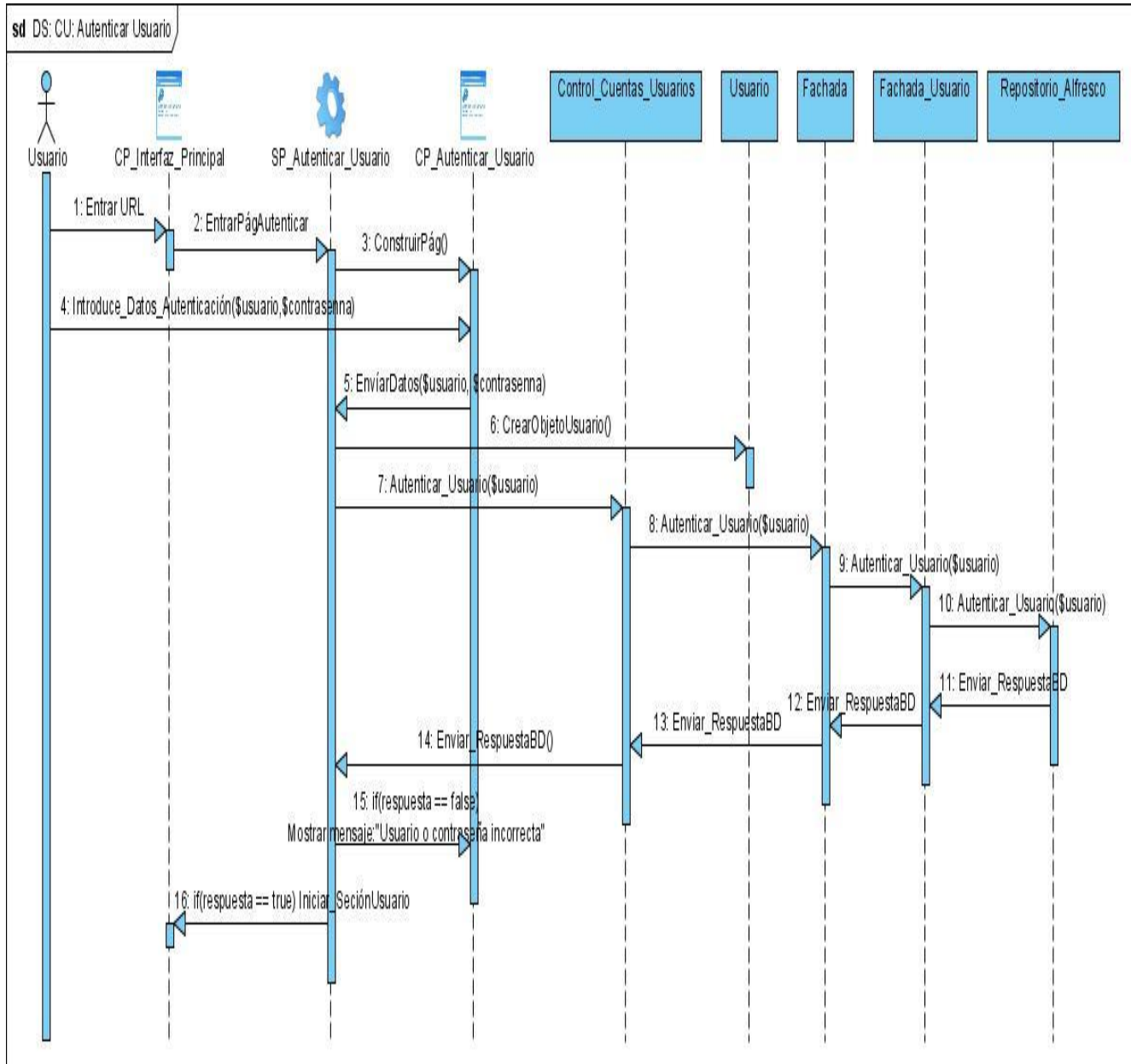


DCD-EW\_CUS\_Revisar Documentos

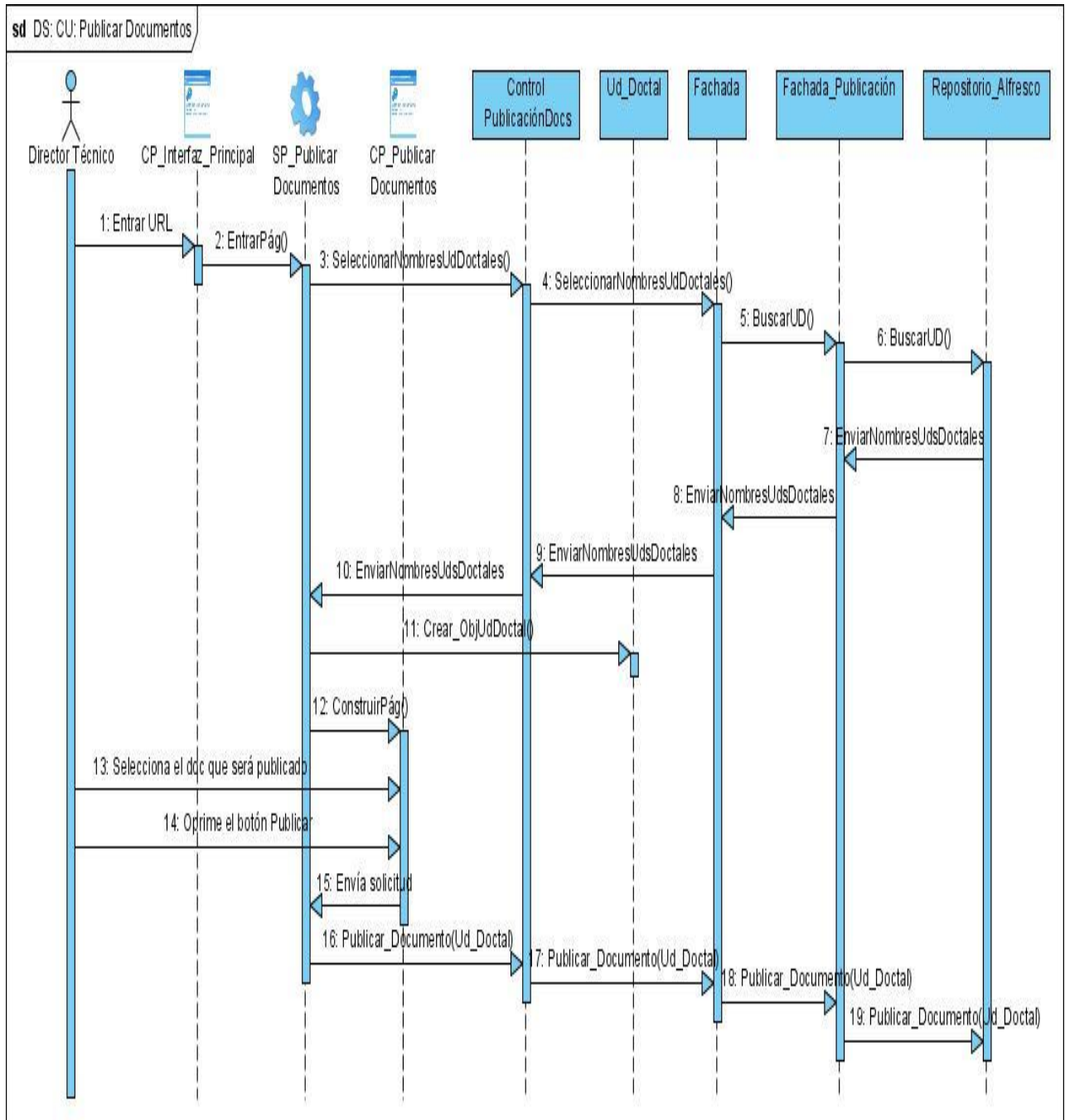


**Anexo 7 Diagramas de secuencia**

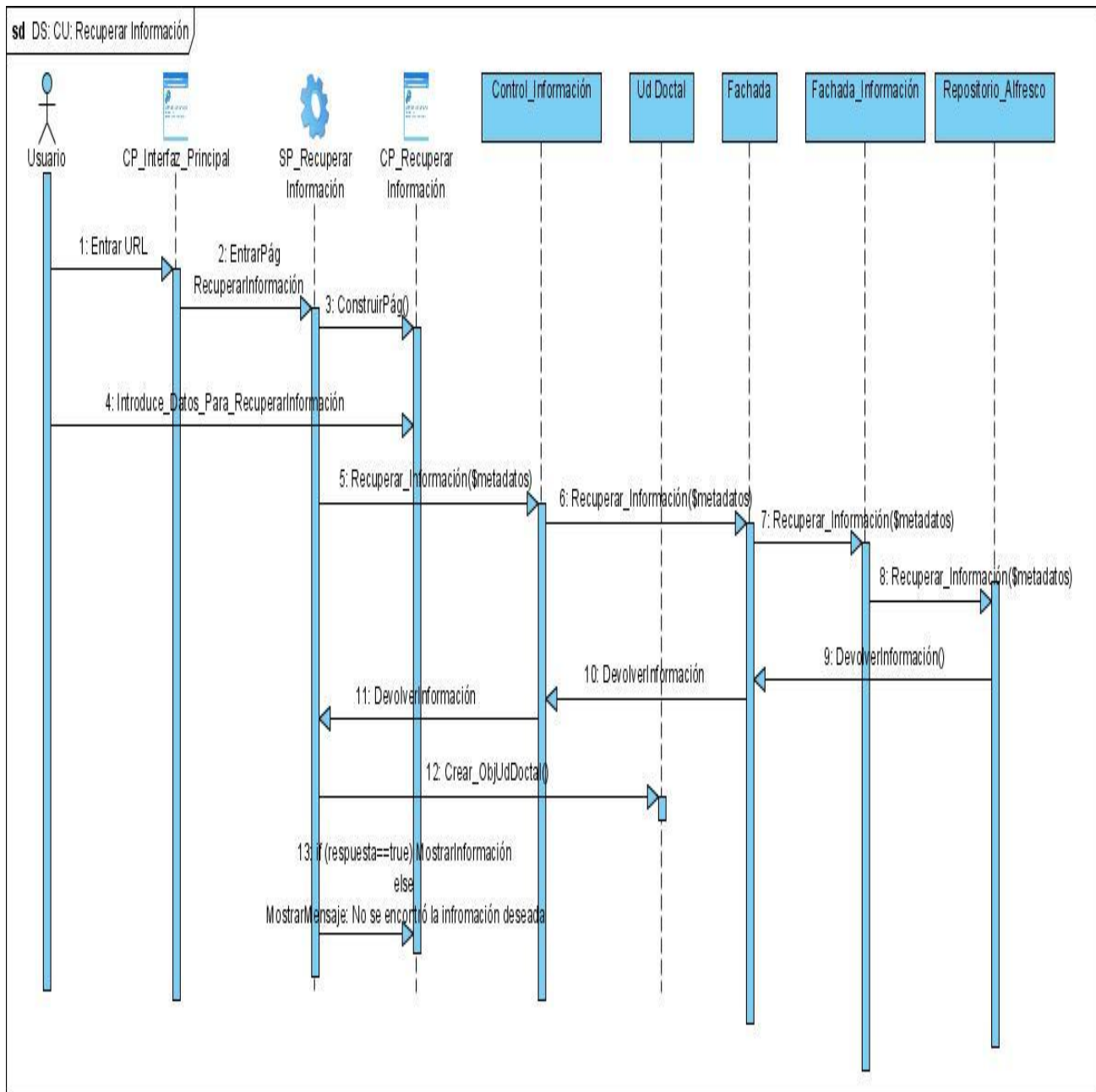
**DS\_CU\_Autenticar Usuario**



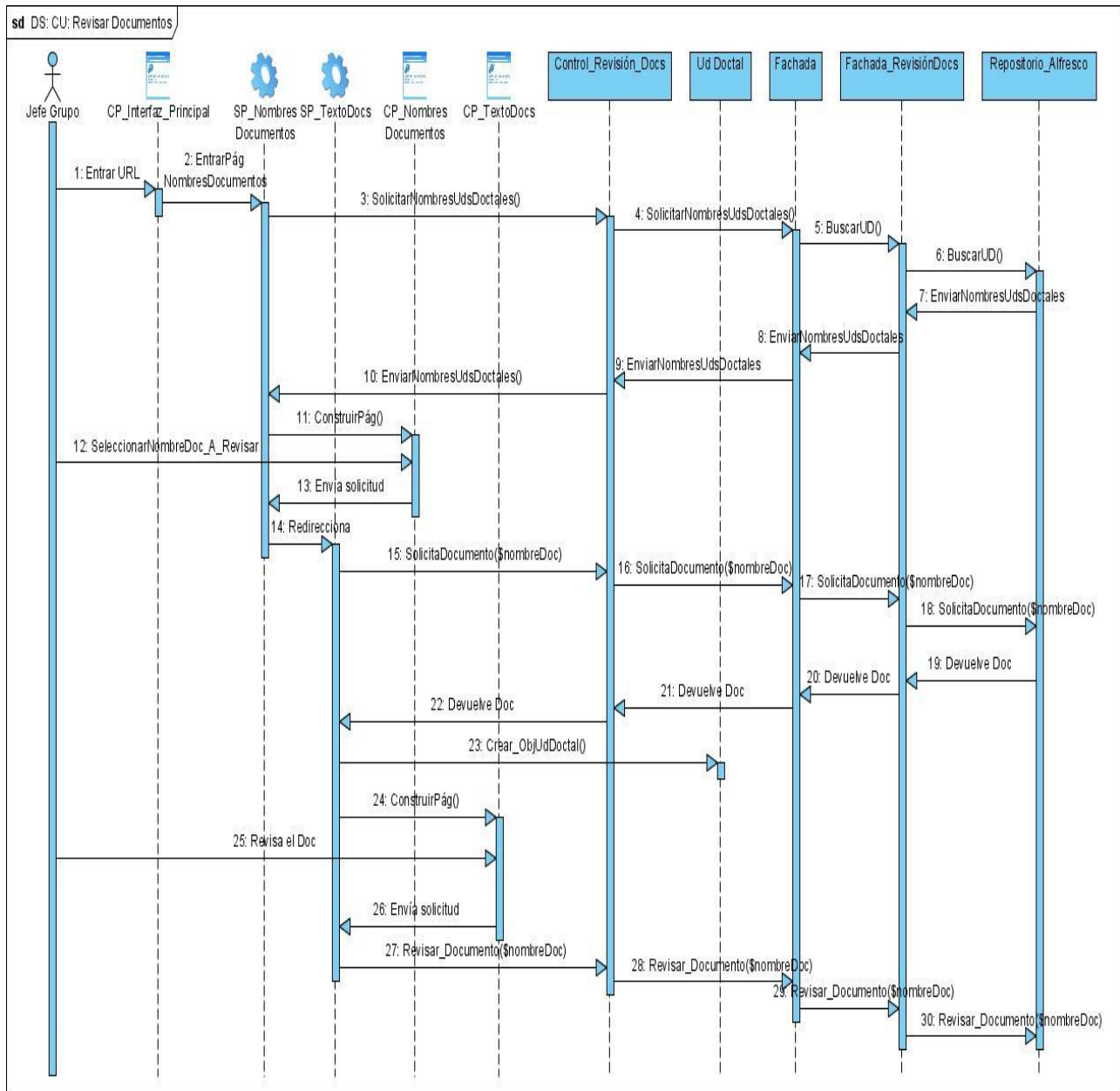
**DS\_CU\_Publicar Documentos**



**DS\_CU\_Recuperar Información**

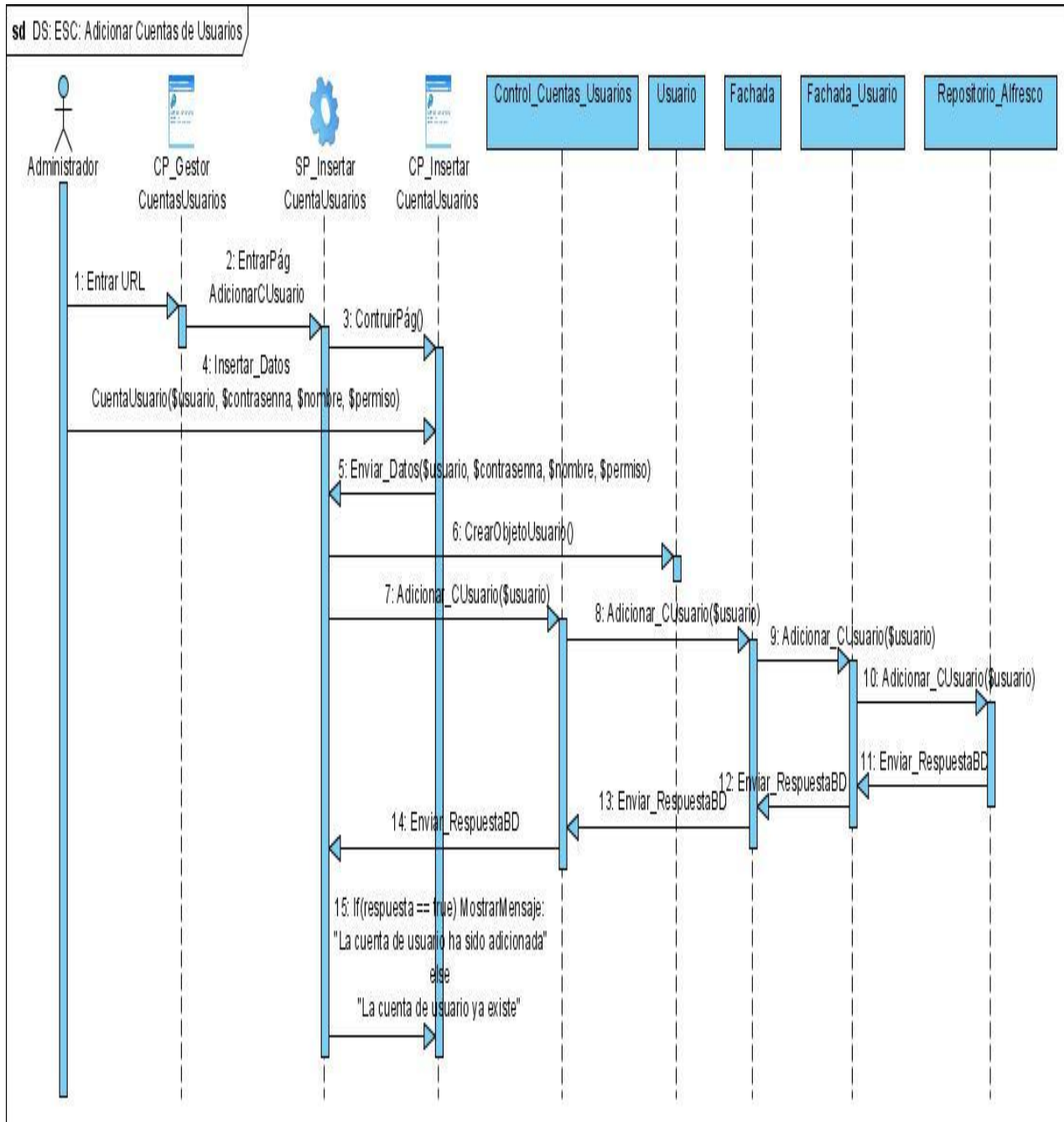


**DS\_CU\_Revisar Documentos**

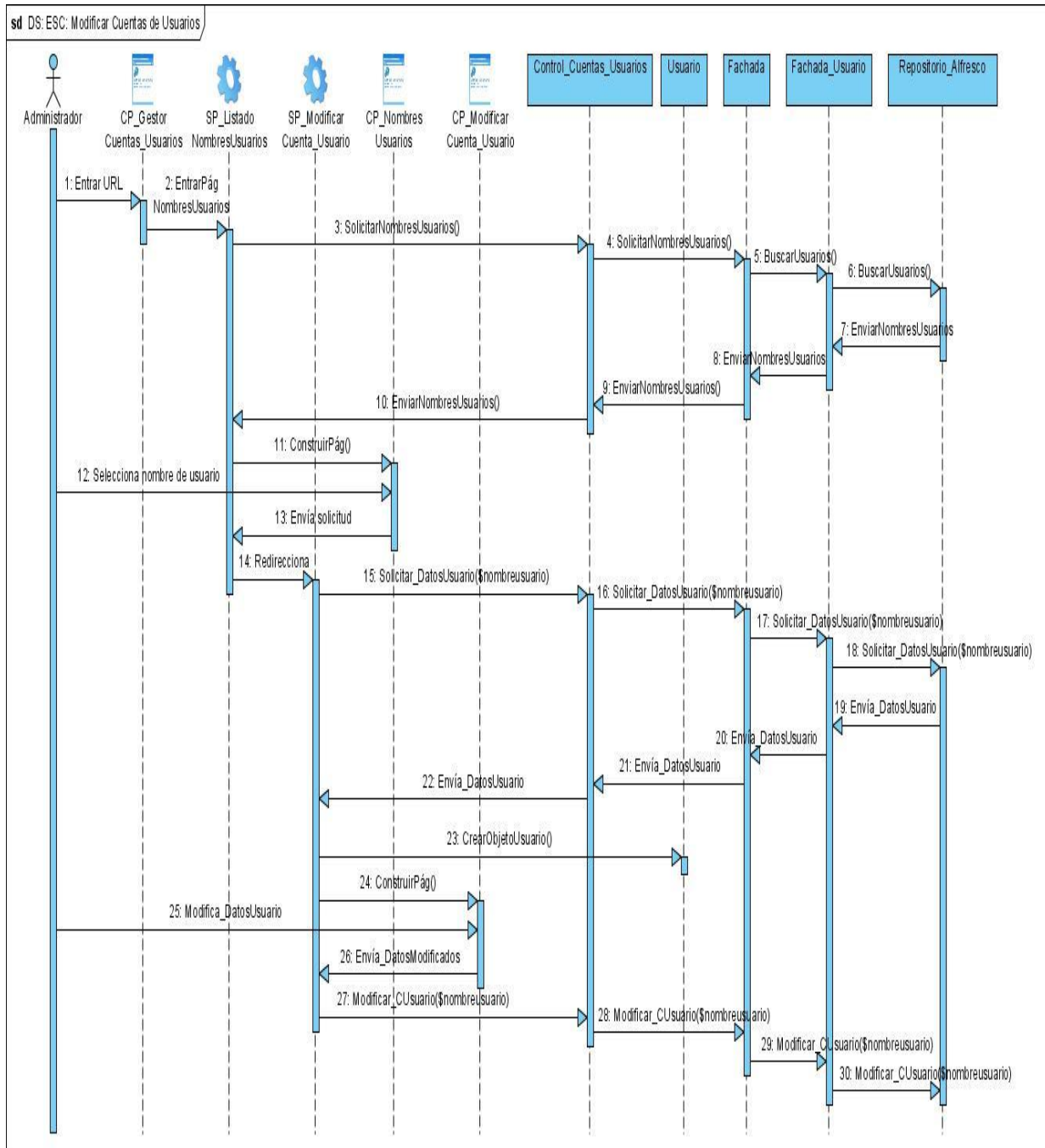




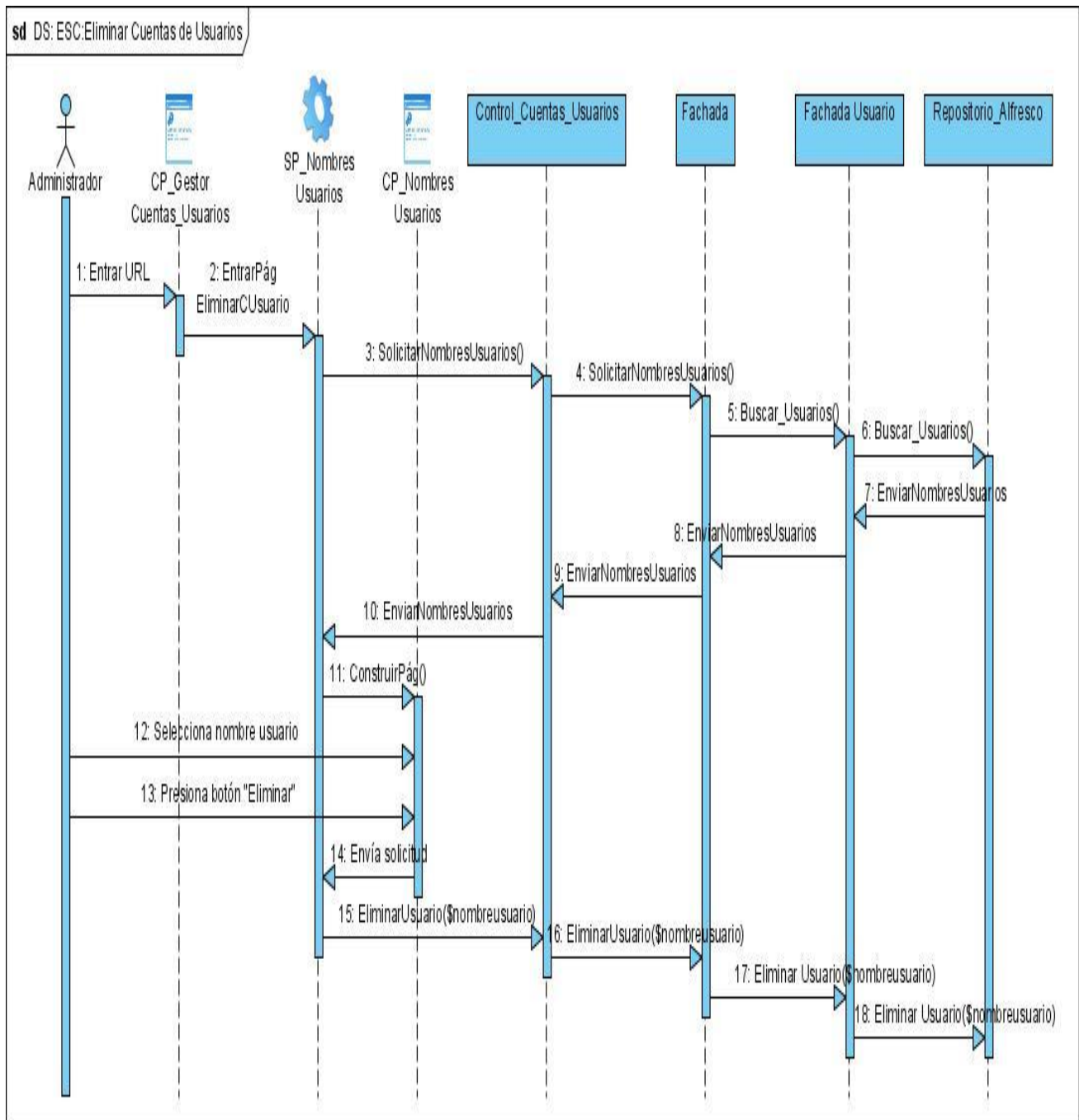
**DS\_CU: Gestionar Cuentas de Usuarios\_ESC: Adicionar**



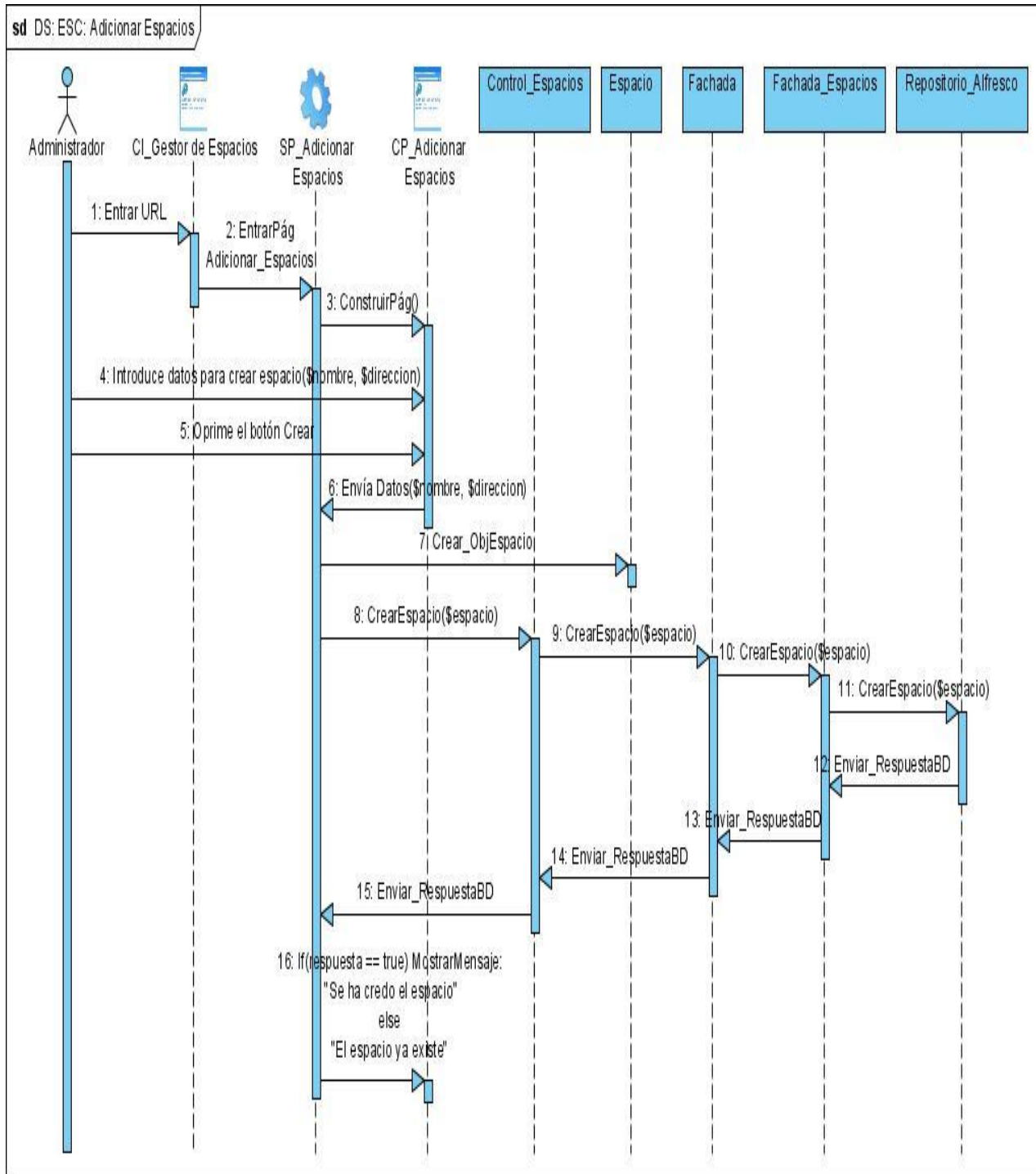
**DS\_CU: Gestionar Cuentas de Usuarios\_ESC: Modificar**



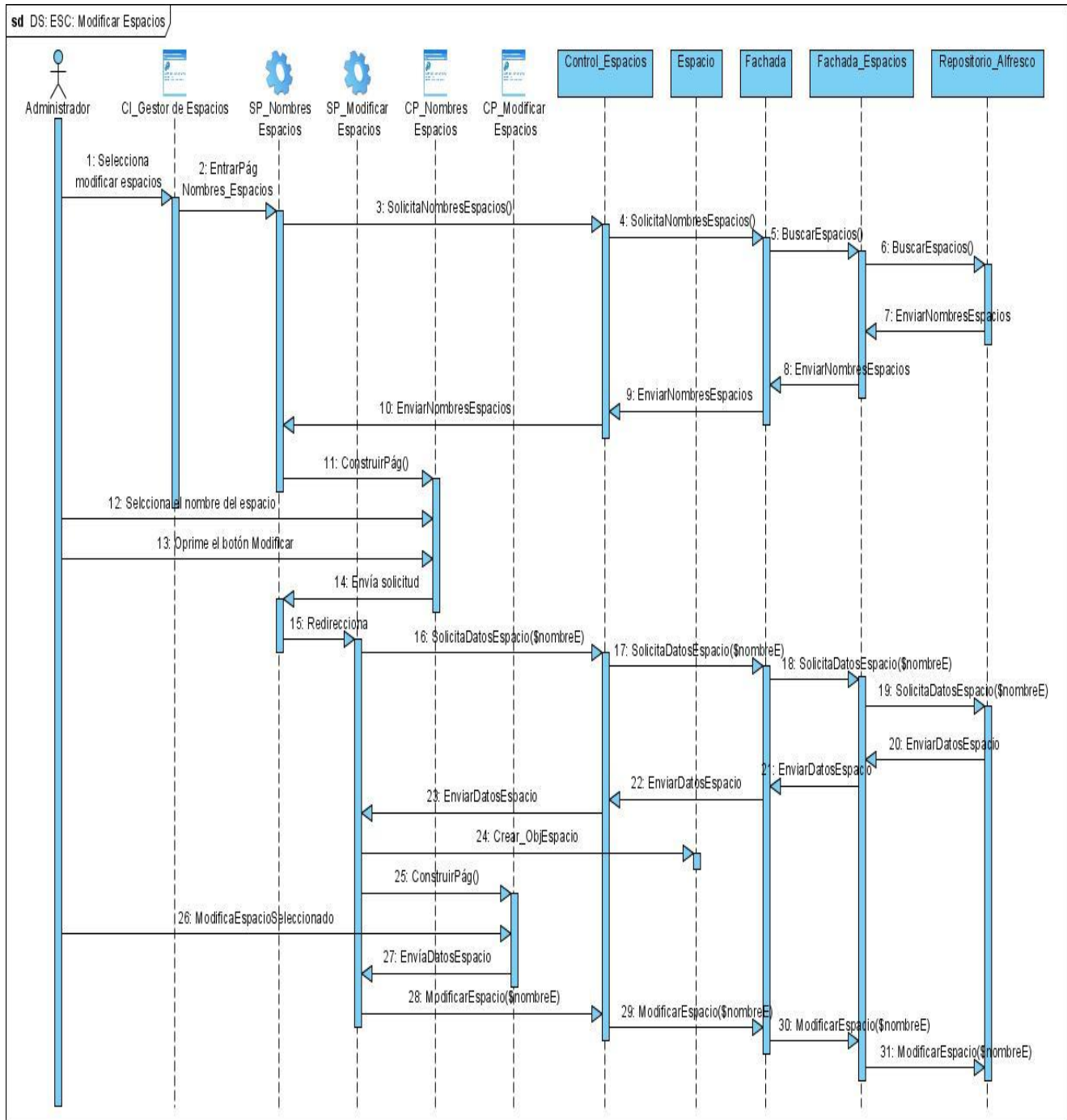
**DS\_CU: Gestionar Cuentas de Usuarios\_ESC: Eliminar**



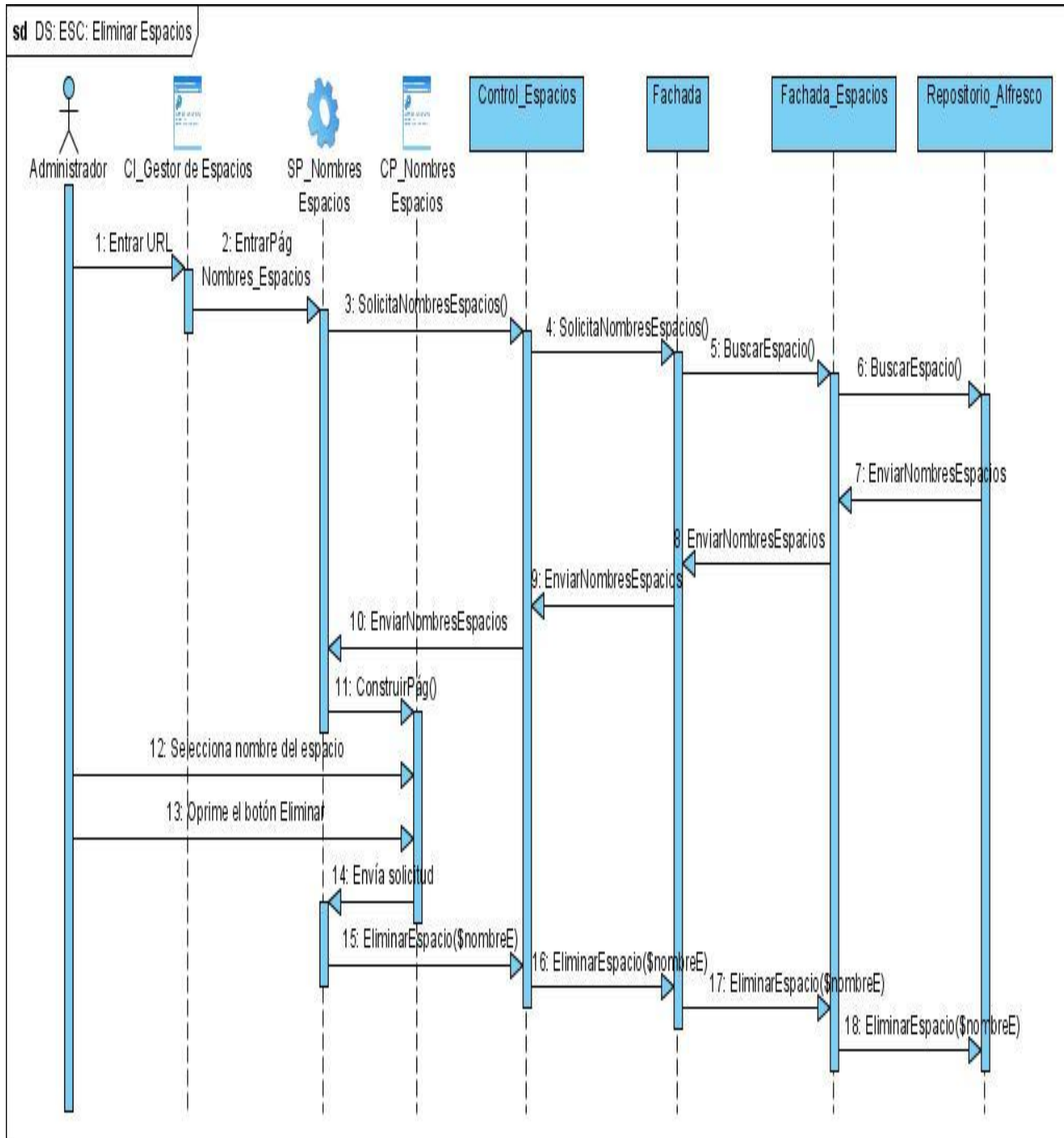
**DS\_CU: Gestionar Espacios\_ESC: Adicionar**



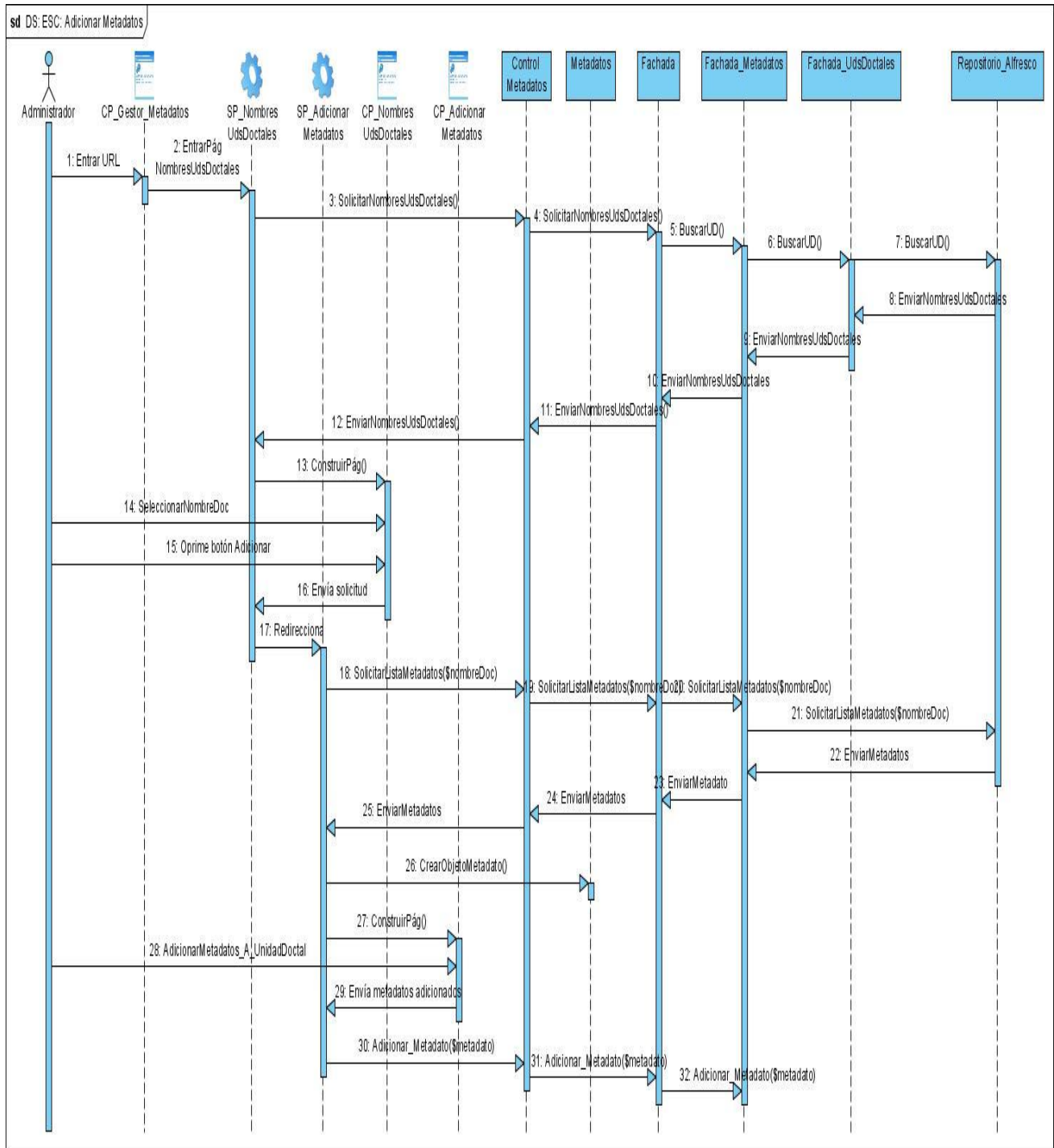
**DS\_CU: Gestionar Espacios\_ESC: Modificar**



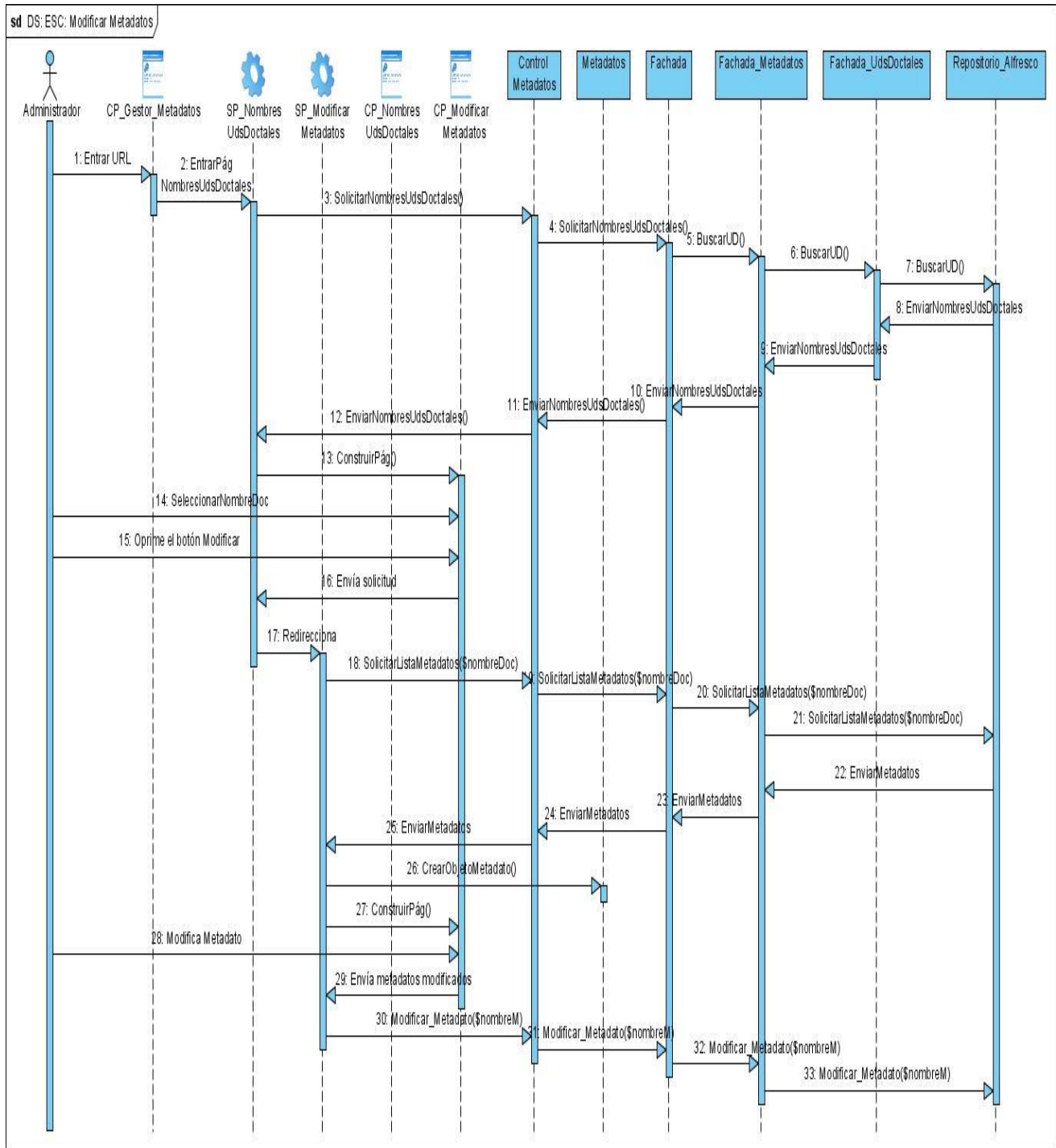
**DS\_CU: Gestionar Espacios\_ESC: Eliminar**



**DS\_CU: Gestionar Metadatos\_ ESC: Adicionar**

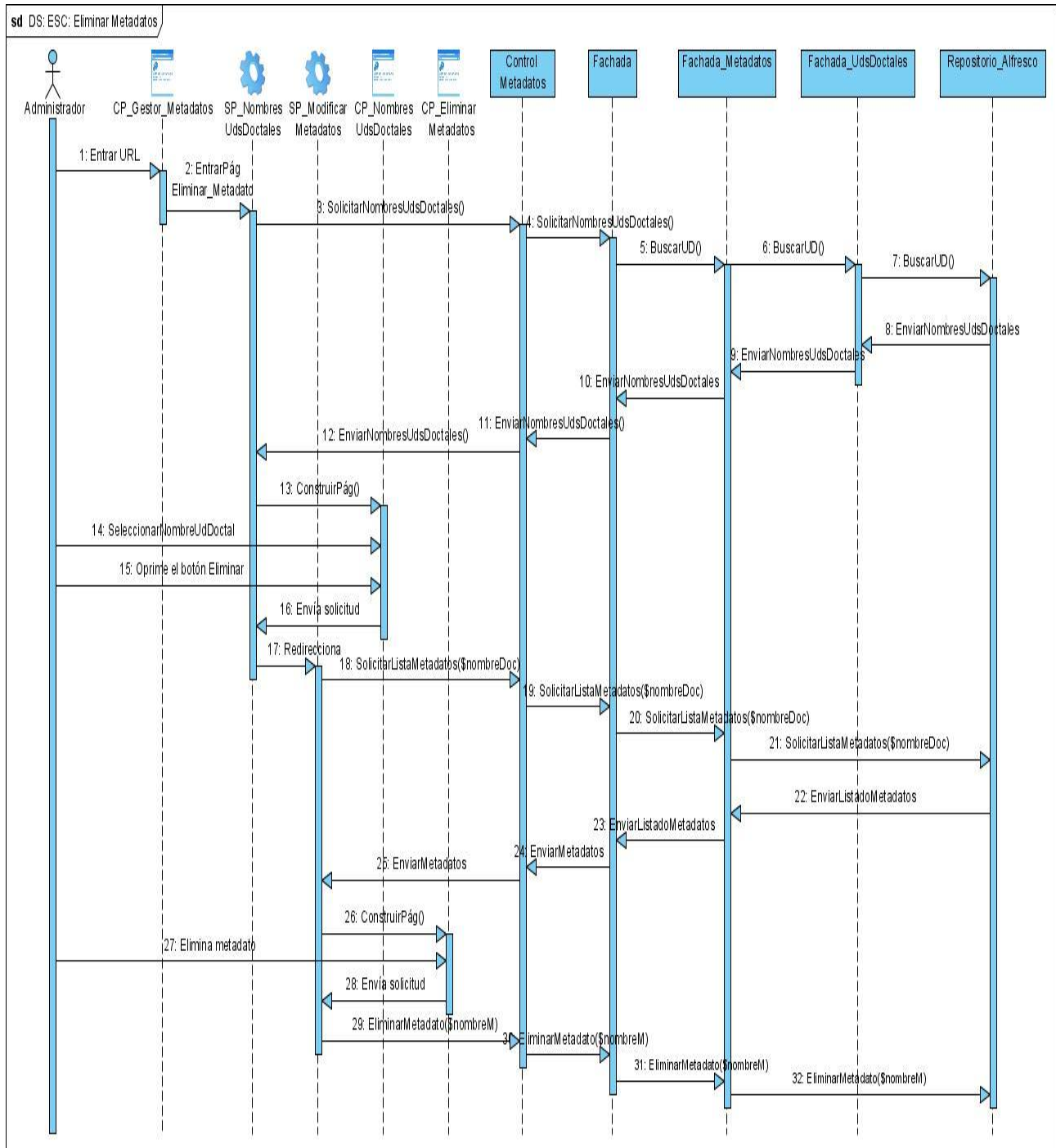


**DS\_CU: Gestionar Metadatos\_ESC: Modificar**

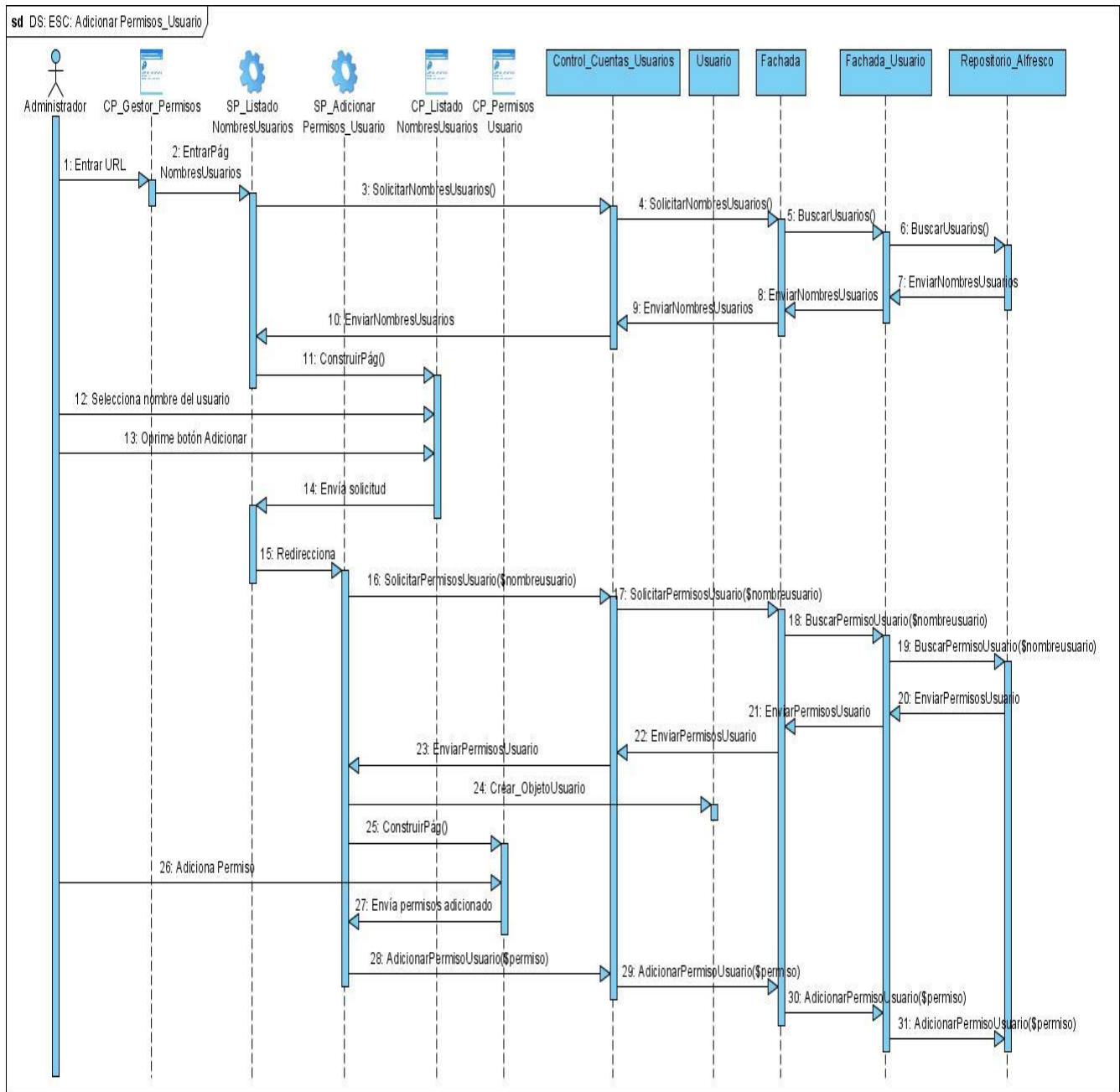




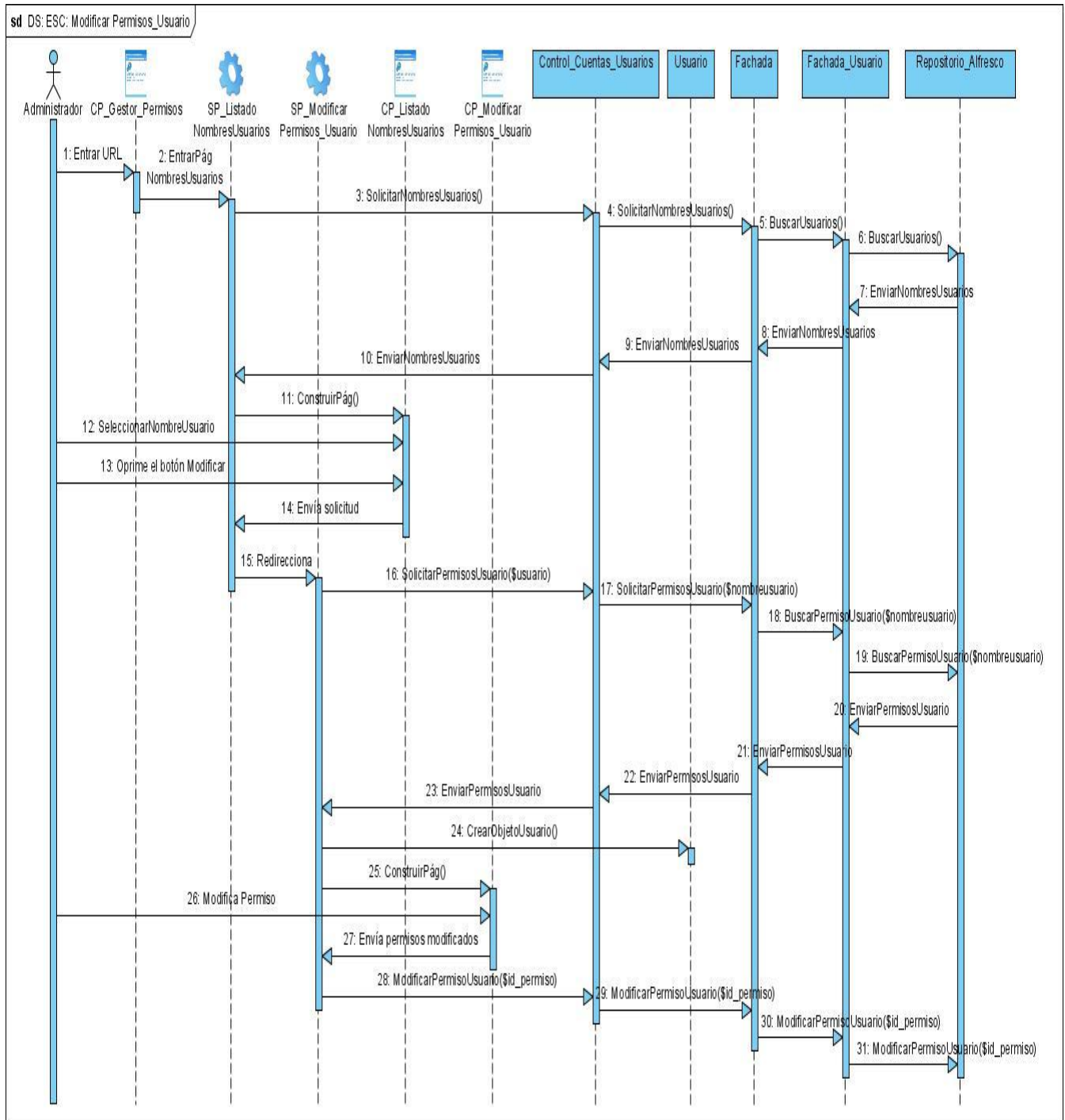
**DS\_CU: Gestionar Metadatos\_ESC: Eliminar**



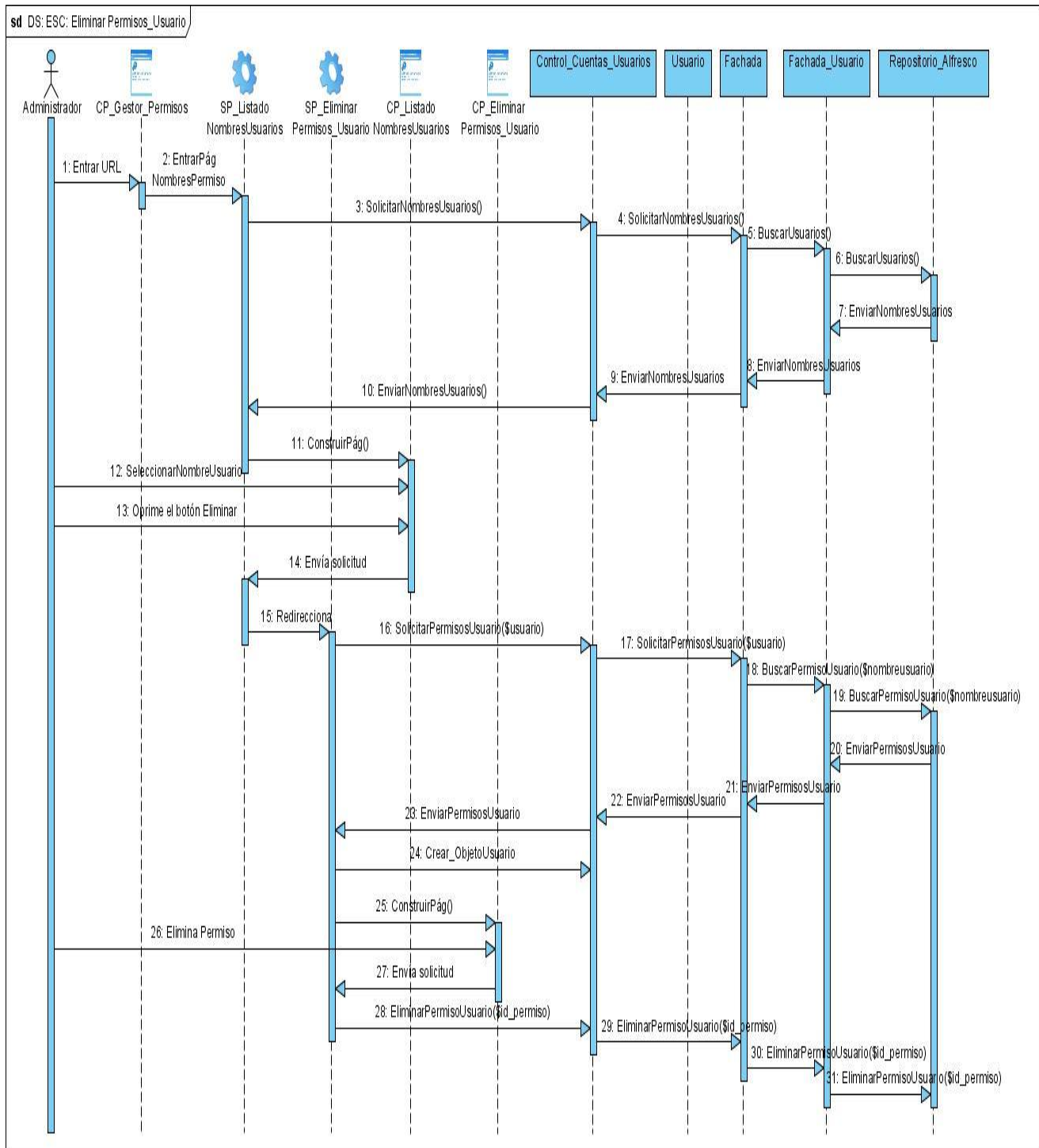
**DS\_CU: Gestionar Permisos Usuario\_ESC: Adicionar**



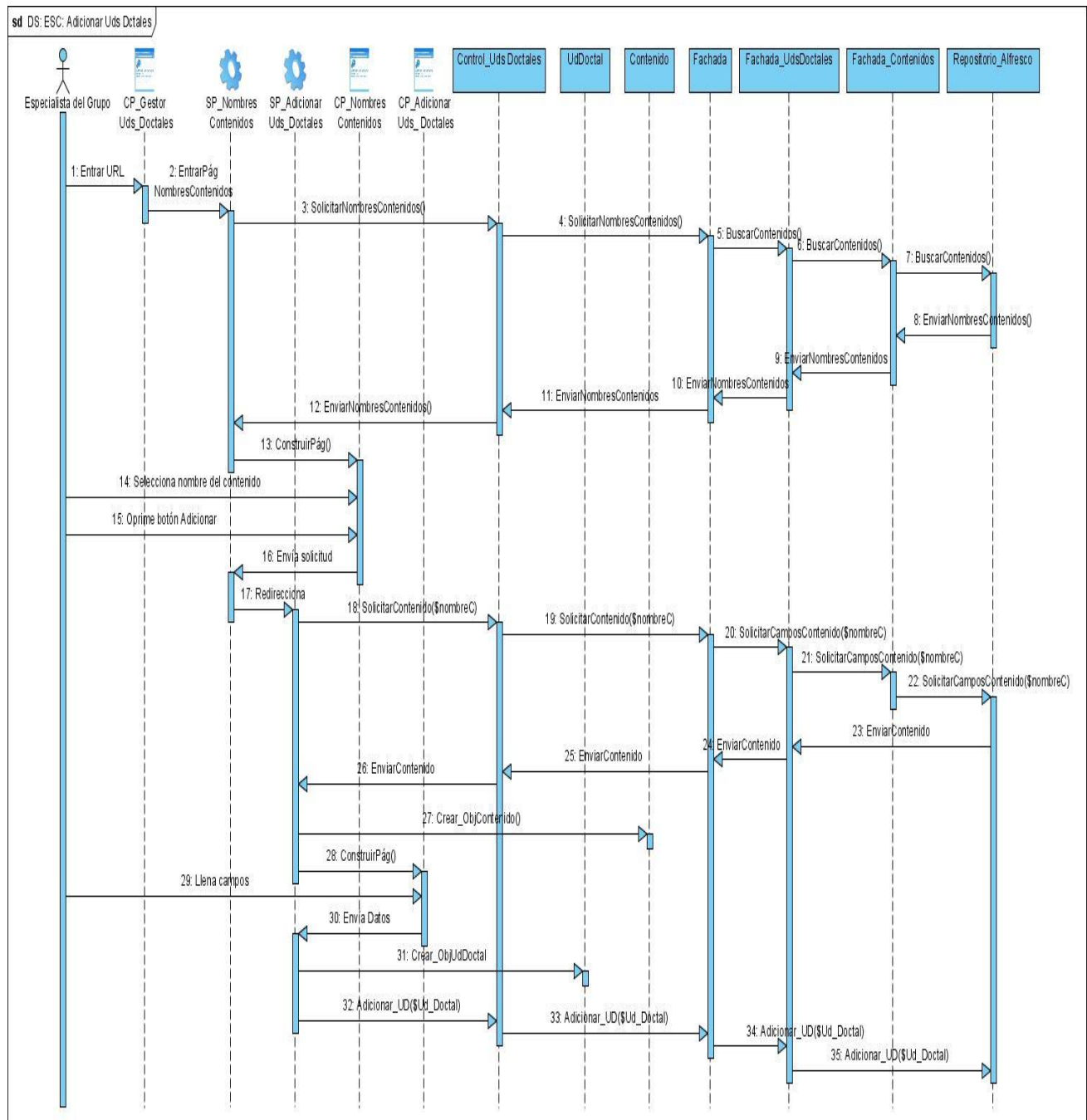
**DS\_CU: Gestionar Permisos Usuario\_ESC: Modificar**



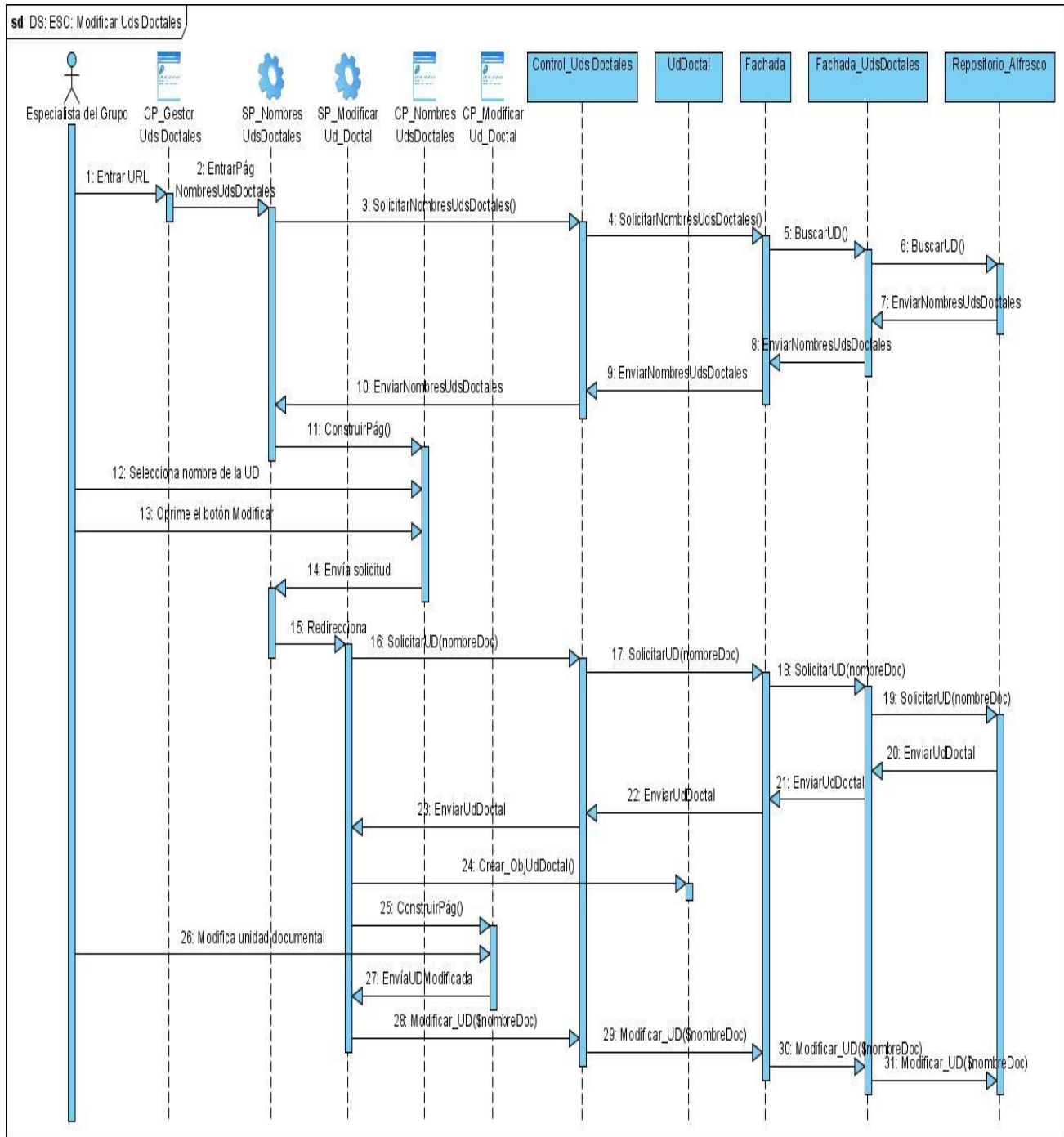
**DS\_CU: Gestionar Permisos Usuario\_ESC: Eliminar**



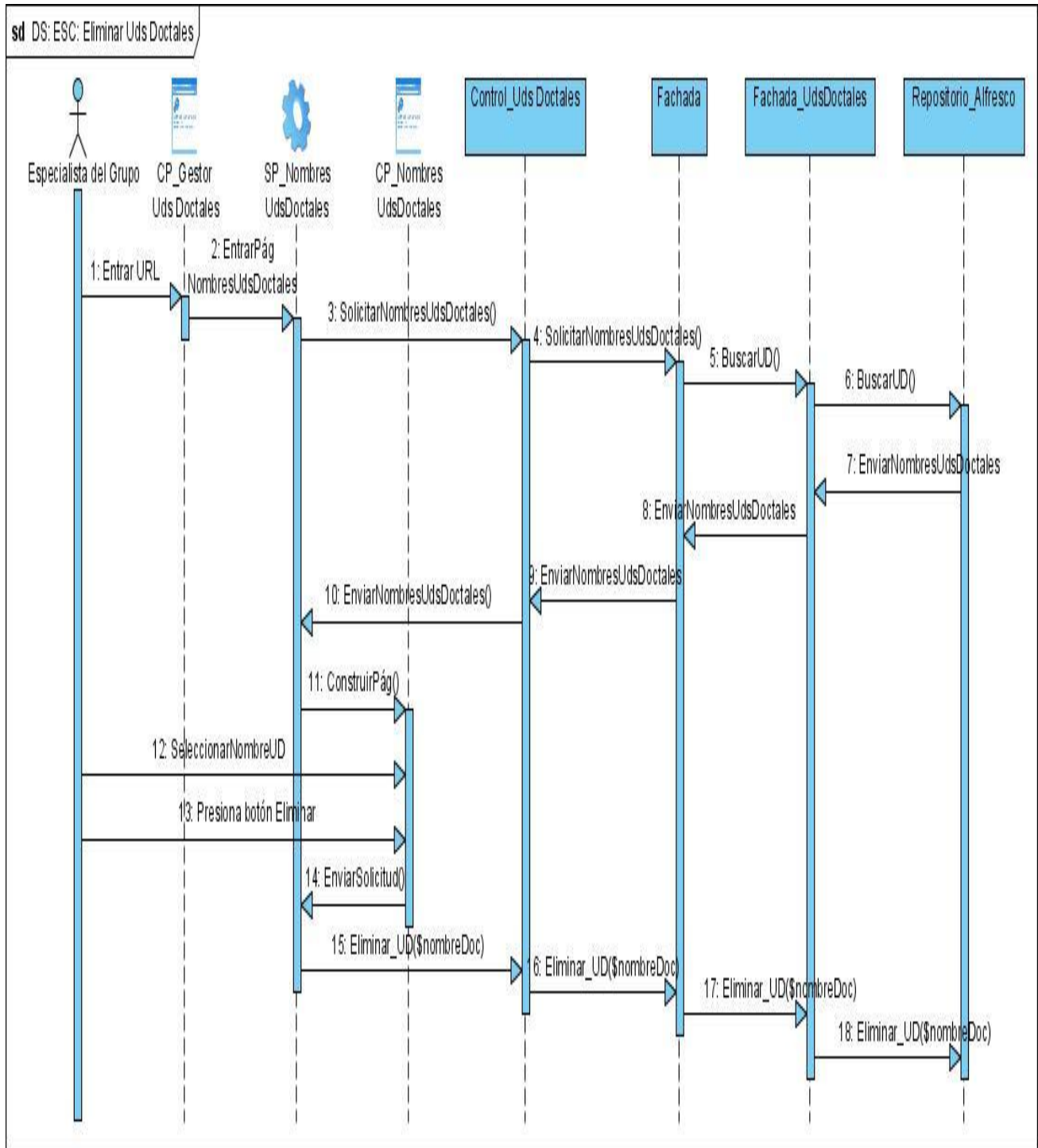
**DS\_CU: Gestionar Uds\_Docs\_ESC: Adicionar**



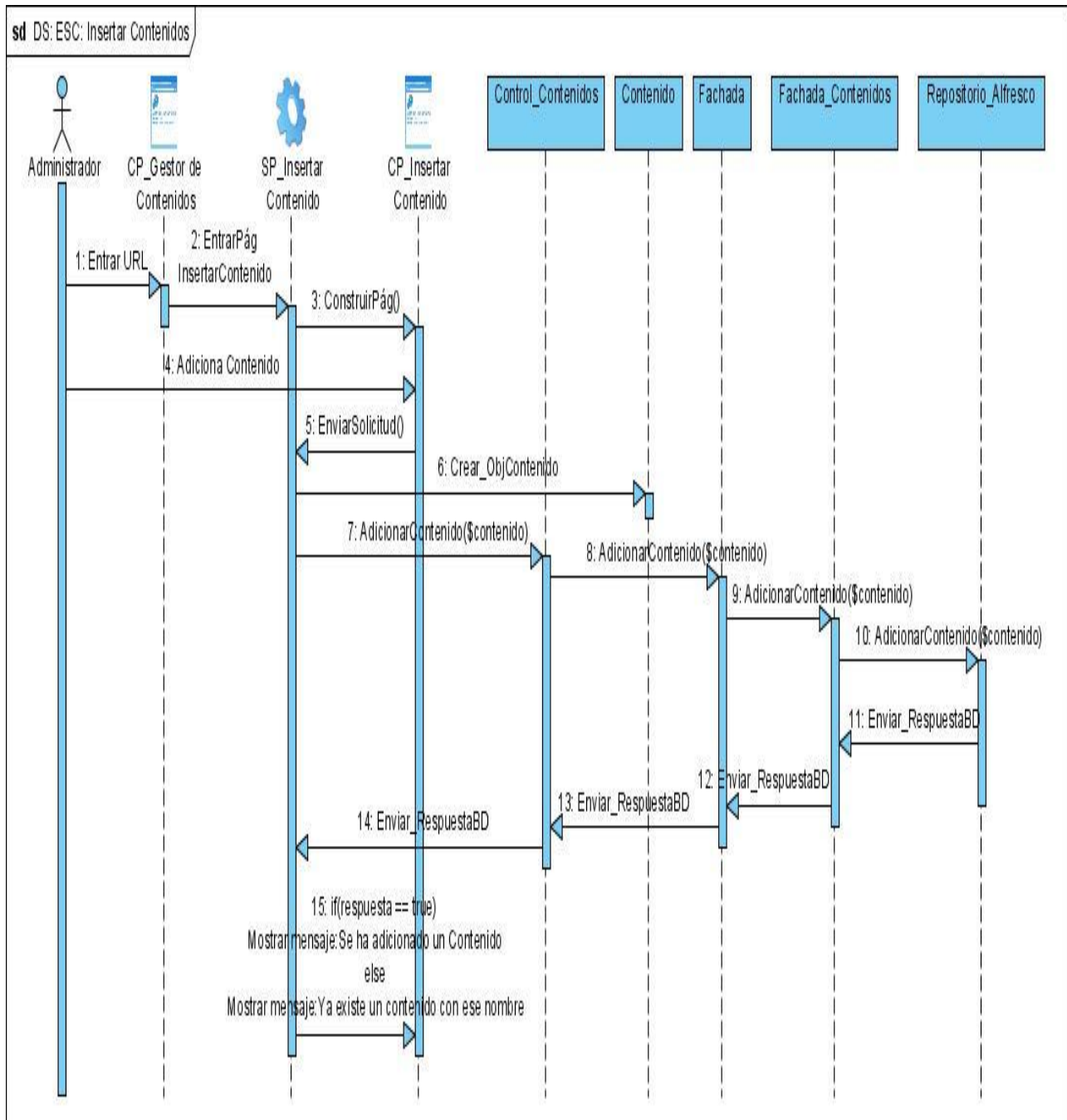
**DS\_CU: Gestionar Uds\_Docs\_ESC: Modificar**



**DS\_CU: Gestionar Uds\_Docs\_ESC: Eliminar**

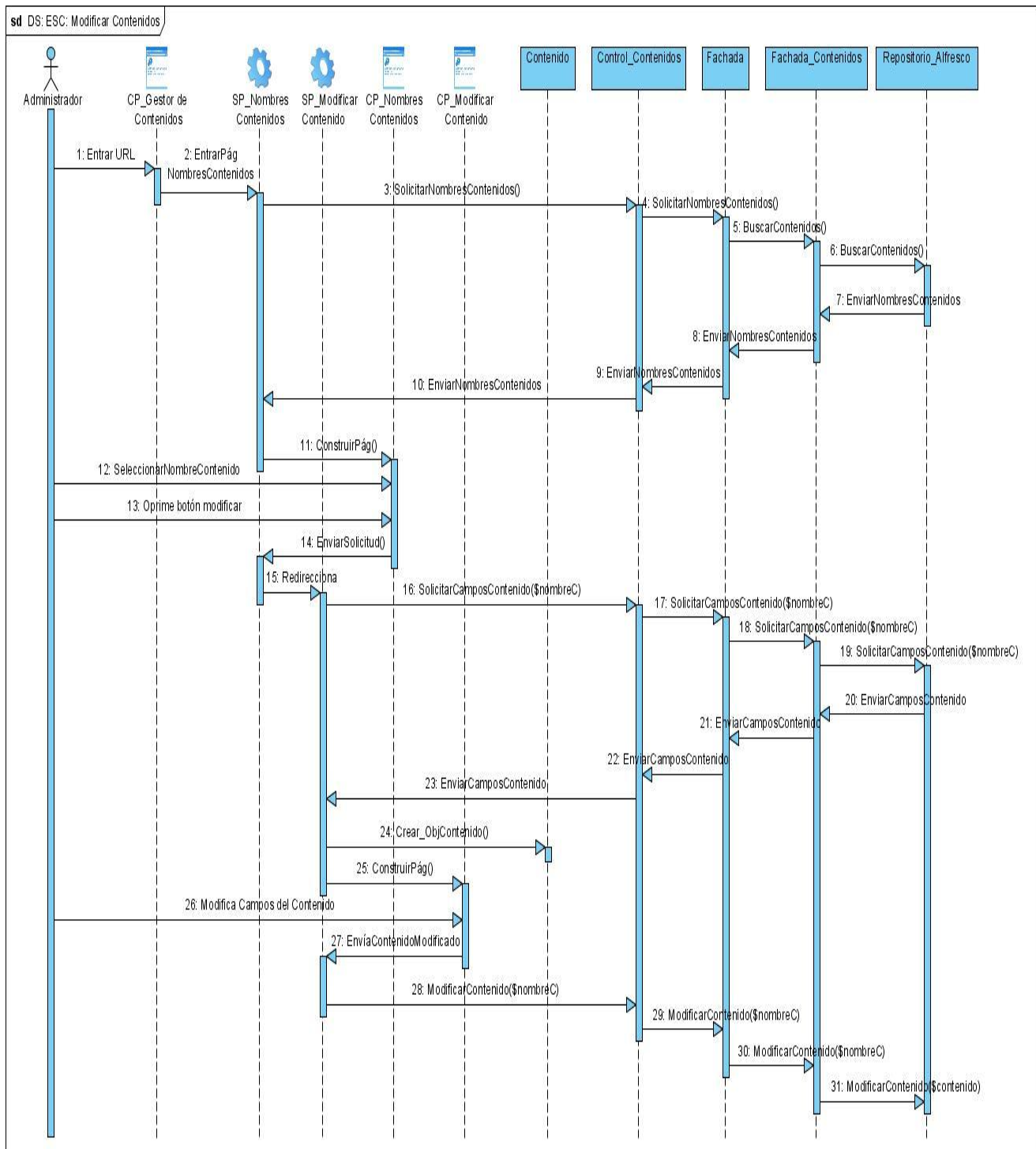


**DS\_CU: Gestionar Contenidos\_Esc: Insertar**

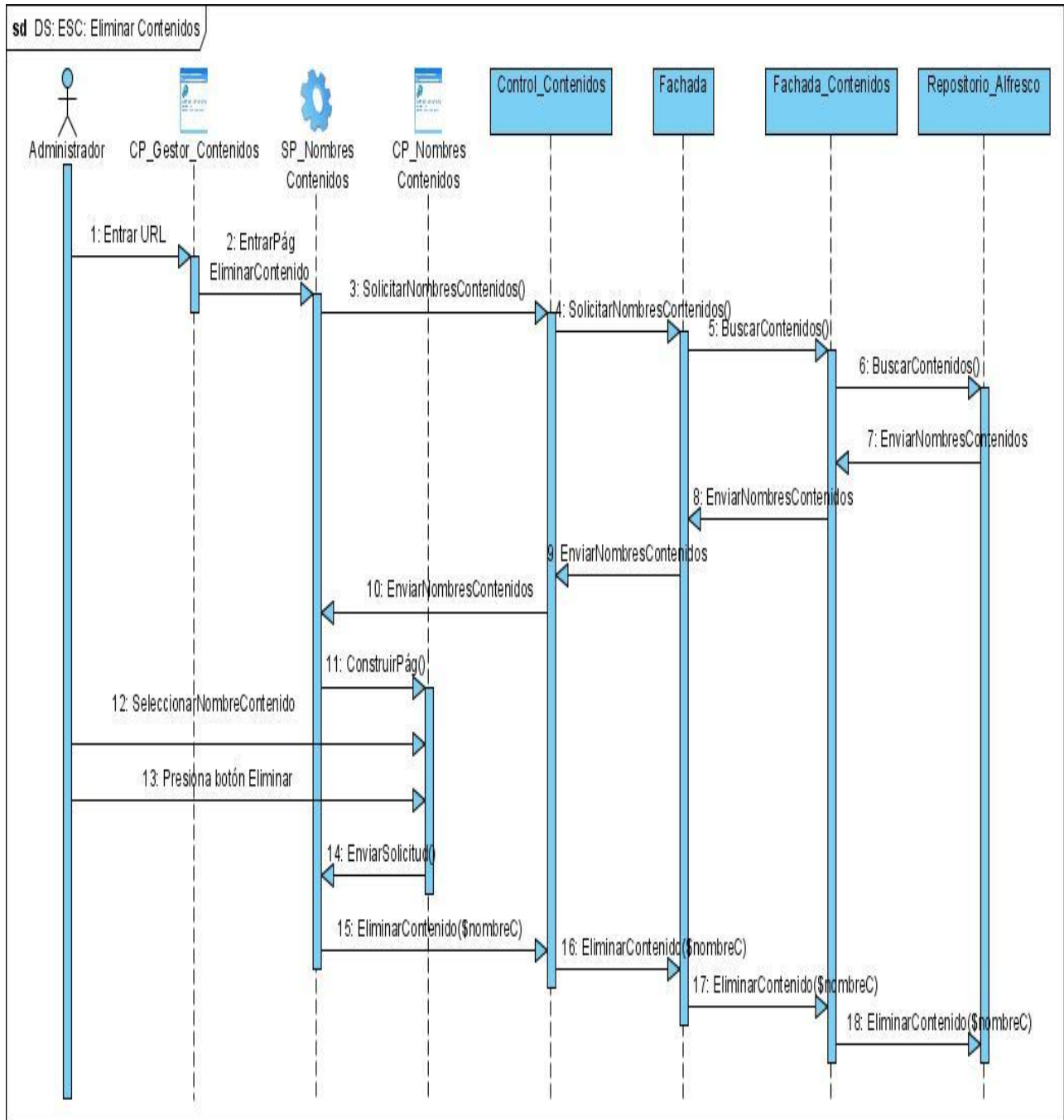




**DS\_CU: Gestionar Contenidos\_Esc: Modificar**



**DS\_CU: Gestionar Contenidos\_Esc: Eliminar**



### DS\_CU: Configurar Flujo de Trabajo

