



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Diseño e implementación de una plataforma web para gestión documental (WebDoc)

Proyecto Final de Carrera

Ingeniería Técnica en Informática de Sistemas

(Especialidad en Ingeniería del Software)

Autor: Alberto Rodríguez Baixauli

Director: Juan Vicente Capella Hernández

Julio de 2013

Agradecimientos

Querría expresar mi agradecimiento a toda mi familia, en especial a mi mujer, a mi hijo, a mis padres y hermana, ya que, sin ellos, no podría haber llegado hasta aquí.

También querría agradecer a mi director de proyecto sus indicaciones y paciencia, y el haberme hecho comprender lo que parecía incomprensible.

“El que habla por su cuenta, busca su propia gloria; pero el que busca la gloria del que le ha enviado, ese es veraz; y no hay impostura en él.”

Jn 7:18

Resumen

En esta memoria se expondrá el desarrollo de una plataforma web (**WebDoc**), diseñada y desarrollada bajo la premisa de automatizar la gestión documental de una organización lo máximo posible. Para ello se ha diseñado un método semi-automático de clasificación de diferentes tipos de archivos, así como un sistema de monitorización sobre la creación y borrado de archivos, un etiquetado unívoco de los mismos (accesible desde un ordenador o desde un dispositivo móvil) así como la inclusión de un sistema ligero de búsqueda y un inventariado basado en el tipo de archivo.

Para la realización de dicha plataforma se ha utilizado de base el gestor de contenidos **Joomla!**, el cual se ha modificado radicalmente en la parte de usuario, incluyendo elementos en **HTML5**, **CSS3** y **JavaScript**, así como código **PHP** incrustado en componentes **Joomla!** y distribuido por diversas partes de todo el sistema.

Palabras clave: gestión documental, web, php, html5, css3, Joomla.



Tabla de contenidos

- 1. Introducción.
 - 1.1. *Motivación y tecnologías utilizadas*.....11
 - 1.2. *Objetivos*.....11
 - 1.3. *Estructura*.....12
- 2. Tecnologías, entornos de ejecución e implementación.
 - 2.1. *Tecnologías utilizadas*.....13
 - 2.2. *Entorno de Ejecución*.....14
 - 2.3. *Herramientas de Desarrollo*.....14
- 3. Ingeniería Web.
 - 3.1. *Introducción*.....15
 - 3.2. *Especificación de Requisitos Software*.....15
 - 3.2.1. *Especificación de Requisitos Software*.....15

ESPECIFICACIÓN DE REQUISITOS SOFTWARE	1. Introducción.	
	1.1. <i>Propósito</i>	17
	1.2. <i>Ámbito</i>	17
	1.3. <i>Definiciones, acrónimos y abreviaturas</i>	18
	1.4. <i>Referencias</i>	19
	1.5. <i>Visión global</i>	19
	2. Descripción general.	
	2.1. <i>Perspectiva del producto</i>	20
	2.2. <i>Funciones del producto</i>	20
	2.3. <i>Características del usuario</i>	21
	2.4. <i>Restricciones generales</i>	21
	2.5. <i>Supuestos y dependencias</i>	22
	3. Requisitos Específicos.	
	3.1. <i>Requisitos de interfaces externos.</i>	
	3.1.1. <i>Interfaces de usuario</i>	23
	3.1.2. <i>Interfaces hardware</i>	23
	3.1.3. <i>Interfaces software</i>	23
	3.1.4. <i>Interfaces de comunicaciones</i>	23
	3.2. <i>Requisitos funcionales.</i>	
	3.2.1. <i>Modo Creación De Documentos</i>	24
3.2.2. <i>Modo Visualización De Documentos</i>	25	
3.2.3. <i>Modo Administración</i>	27	
3.3. <i>Requisitos de rendimiento</i>	28	
3.4. <i>Restricciones de diseño.</i>		
3.4.1. <i>Estándares cumplidos</i>	28	
3.4.2. <i>Limitaciones hardware</i>	28	
3.5. <i>Atributos.</i>		
3.5.1. <i>Seguridad</i>	28	
3.5.2. <i>Mantenimiento</i>	29	
3.6. <i>Otros requisitos.</i>		
3.6.1. <i>Bases de datos</i>	29	



3.2.2. Diagramas de requerimientos.	
3.2.2.1. Introducción.....	31
3.2.2.2. Modo Creación De Documentos.....	31
3.2.2.3. Modo Visualización De Documentos.....	32
3.2.2.4. Modo Administración.....	32
3.2.3. Diagramas de casos de uso.	
3.2.3.1. Introducción.....	33
3.2.3.2. Modo Creación De Documentos.....	33
3.2.3.3. Modo Visualización De Documentos.....	34
3.2.3.4. Modo Administración.....	34
3.3. Análisis.	
3.3.1. Introducción.....	35
3.3.2. Modo Creación De Documentos.	
3.3.2.1. Análisis del caso de uso <code>CREAR_CARPETA</code>	35
3.3.2.2. Análisis del caso de uso <code>CREAR_DOCUMENTO</code>	36
3.3.2.3. Análisis del caso de uso <code>ELIMINAR_CARPETA</code>	36
3.3.2.4. Análisis del caso de uso <code>ELIMINAR_DOCUMENTO</code>	37
3.3.3. Modo Visualización De Documentos.	
3.3.3.1. Análisis del caso de uso <code>LISTAR_TODOS_LOS_DOCUMENTOS</code>	37
3.3.3.2. Análisis del caso de uso <code>LISTAR_TODOS_LOS_DOCUMENTOS_DE_UN_TIPO</code>	37
3.3.3.3. Análisis del caso de uso <code>LISTAR_UNA_CATEGORIA_DADO_UN_TIPO_DE_DOCUMENTO</code> ..	37
3.3.3.4. Análisis del caso de uso <code>BUSCAR_DOCUMENTO</code>	38
3.3.3.5. Análisis del caso de uso <code>CREAR_PERFIL_USUARIO</code>	38
3.3.3.6. Análisis del caso de uso <code>ACTUALIZAR_PERFIL_USUARIO</code>	39
3.3.4. Modo Administración.	
3.3.4.1. Análisis del caso de uso <code>LISTAR_REGISTROS_DOCUMENTOS_BORRADOS</code>	39
3.3.4.2. Análisis del caso de uso <code>ELIMINAR_REGISTRO_DOCUMENTO_BORRADO</code>	40
3.3.4.3. Análisis del caso de uso <code>CREAR_PERFIL_USUARIO</code>	40
3.3.4.4. Análisis del caso de uso <code>ACTUALIZAR_PERFIL_USUARIO</code>	41
3.3.4.5. Análisis del caso de uso <code>AUTORIZAR_PERFIL_USUARIO</code>	41
3.3.4.6. Análisis del caso de uso <code>DESAUTORIZAR_PERFIL_USUARIO</code>	41
3.3.4.7. Análisis del caso de uso <code>BORRAR_PERFIL_USUARIO</code>	42
3.4. Diseño.	
3.4.1. Introducción.....	43
3.4.2. Modo Creación De Documentos.	
3.4.2.1. Diseño del caso de uso <code>CREAR_CARPETA</code>	43
3.4.2.2. Diseño del caso de uso <code>CREAR_DOCUMENTO</code>	44
3.4.2.3. Diseño del caso de uso <code>ELIMINAR_CARPETA</code>	44
3.4.2.4. Diseño del caso de uso <code>ELIMINAR_DOCUMENTO</code>	45
3.4.3. Modo Visualización De Documentos.	
3.4.3.1. Diseño del caso de uso <code>LISTAR_TODOS_LOS_DOCUMENTOS</code>	45
3.4.3.2. Diseño del caso de uso <code>LISTAR_TODOS_LOS_DOCUMENTOS_DE_UN_TIPO</code>	45
3.4.3.3. Diseño del caso de uso <code>LISTAR_UNA_CATEGORIA_DADO_UN_TIPO_DE_DOCUMENTO</code> ..	45
3.4.3.4. Diseño del caso de uso <code>BUSCAR_DOCUMENTO</code>	46

3.4.3.5. Diseño del caso de uso <i>CREAR_PERFIL_USUARIO</i>	46
3.4.3.6. Diseño del caso de uso <i>ACTUALIZAR_PERFIL_USUARIO</i>	47
3.4.4. Modo Administración.	
3.4.4.1. Introducción.....	47
3.4.4.2. Diseño del caso de uso <i>LISTAR_REGISTROS_DOCUMENTOS_BORRADOS</i>	48
3.4.4.3. Diseño del caso de uso <i>ELIMINAR_REGISTRO_DOCUMENTO_BORRADO</i>	48
3.4.4.4. Diseño del caso de uso <i>CREAR_PERFIL_USUARIO</i>	49
3.4.4.5. Diseño del caso de uso <i>ACTUALIZAR_PERFIL_USUARIO</i>	49
3.4.4.6. Diseño del caso de uso <i>AUTORIZAR_PERFIL_USUARIO</i>	50
3.4.4.7. Diseño del caso de uso <i>DESAUTORIZAR_PERFIL_USUARIO</i>	50
3.4.4.8. Diseño del caso de uso <i>BORRAR_PERFIL_USUARIO</i>	50
3.5. Implementación.	
3.5.1. Introducción.....	51
3.5.2. Implementación en Joomla!	51
3.5.3. Modo Creación De Documentos.	
3.5.3.1. Implementación del caso de uso <i>CREAR_CARPETA</i>	53
3.5.3.2. Implementación del caso de uso <i>CREAR_DOCUMENTO</i>	53
3.5.3.3. Implementación del caso de uso <i>ELIMINAR_CARPETA</i>	59
3.5.3.4. Implementación del caso de uso <i>ELIMINAR_DOCUMENTO</i>	59
3.5.4. Modo Visualización De Documentos.	
3.5.4.1. Implementación del caso de uso <i>LISTAR_TODOS_LOS_DOCUMENTOS</i>	64
3.5.4.2. Implementación del caso de uso <i>LISTAR_TODOS_LOS_DOCUMENTOS_DE_UN_TIPO</i>	64
3.5.4.3. Implem. del caso de uso <i>LISTAR_UNA_CATEGORIA_DADO_UN_TIPO_DE_DOCUMENTO</i>	64
3.5.4.4. Implementación del caso de uso <i>BUSCAR_DOCUMENTO</i>	64
3.5.4.5. Implementación del caso de uso <i>CREAR_PERFIL_USUARIO</i>	64
3.5.4.6. Implementación del caso de uso <i>ACTUALIZAR_PERFIL_USUARIO</i>	64
3.5.5. Modo Administración.	
3.5.5.1. Implem. del caso de uso <i>LISTAR_REGISTROS_DOCUMENTOS_BORRADOS</i>	64
3.5.5.2. Implem. del caso de uso <i>ELIMINAR_REGISTRO_DOCUMENTO_BORRADO</i>	64
3.5.5.3. Implementación del caso de uso <i>CREAR_PERFIL_USUARIO</i>	64
3.5.5.4. Implementación del caso de uso <i>ACTUALIZAR_PERFIL_USUARIO</i>	64
3.5.5.5. Implementación del caso de uso <i>AUTORIZAR_PERFIL_USUARIO</i>	65
3.5.5.6. Implementación del caso de uso <i>DESAUTORIZAR_PERFIL_USUARIO</i>	65
3.5.5.7. Implementación del caso de uso <i>BORRAR_PERFIL_USUARIO</i>	65
3.6. Pruebas.	
3.6.1. Introducción.....	66
3.6.2. Desarrollo de las Pruebas.	
3.6.2.1. Introducción.....	66
3.6.2.2. Caso de prueba de integración para el caso de uso <i>CREAR_CARPETA</i>	67
3.6.2.3. Caso de prueba de integración para el caso de uso <i>CREAR_DOCUMENTO</i>	68
3.6.2.4. Caso de prueba de integración para el caso de uso <i>ELIMINAR_CARPETA</i>	69
3.6.2.5. Caso de prueba de integración para el caso de uso <i>ELIMINAR_DOCUMENTO</i>	70
3.7. WebDoc.	
3.7.1. Acceso (login).....	71
3.7.2. Desktop.....	72
3.7.3. Documents Manager.....	73



3.7.4. Archivos tipo PDF.....	75
3.7.5. Categoría 1. Archivos tipo PDF.....	75
3.7.6. Administración.	
3.7.6.1. Acceso (Login).....	76
3.7.6.2. Listado Registros Documentos Borrados.....	76
3.7.6.3. Crear Perfiles de Usuario.....	77
3.7.6.4. Autorizar un Perfil de Usuario.....	77
4. Conclusiones finales.	
4.1. Consecución de Objetivos.....	79
4.2. Trabajo futuro.....	80
4.3. Conclusiones.....	82
5. Bibliografía.....	83

1. Introducción

1.1. Motivación y tecnologías utilizadas.

La gestión documental en una organización es una tarea ardua, incluso hoy en día, con todos los medios informáticos al alcance de la mayoría de la población.

Ello es debido a que no existen herramientas que utilicen tecnologías estandarizadas para la gestión de documentos y que estén disponibles para el amplio conjunto de las organizaciones.

En este proyecto se pretende elaborar una plataforma web para la gestión documental que, mediante tecnologías web, unifique la gestión documental en un único lugar, donde, desde cualquier organización, se pueda acceder en todo momento al conocimiento documental de la empresa. Las tecnologías web utilizadas, y detalladas en el punto 2, son, básicamente, **PHP** [1], **HTML5** [2], **CSS3** (todo lo publicado después del estándar **CSS 2.1**) [3] y **Javascript** [4]. Todo ello desarrollado sobre la plataforma de gestión de contenidos (CMS) **Joomla!** [5].

1.2. Objetivos.

Los objetivos a conseguir por el presente proyecto son los siguientes:

- (1) Implementar un sistema de gestión de usuarios, con altas automáticas y acceso restringido por defecto.
- (2) Elaborar un sistema de clasificación de archivos, basándose en el tipo de archivo y procesando la información que dicho archivo contiene.
- (3) Diseñar una clasificación, basada en categorías, para separación de los archivos por conceptos.
- (4) Implementar un módulo web que se encargue de la creación, modificación y borrado de archivos dentro de la plataforma web, basándose en los puntos 2 y 3.
- (5) Diseñar e implementar un sistema de control de la creación y el borrado de archivos dentro de la plataforma, indicando el usuario, la fecha y la hora.
- (6) Desarrollar un sistema de avisos al/los administrador/es de la aplicación mediante email, parametrizándolo de forma que quede condicionado a la opción de avisar/no avisar a cada administrador/es.

1.2. Estructura.

Esta memoria está organizada en 4 grandes secciones:

- *Sección 1: Introducción.*

En esta sección se hace una breve introducción de toda la memoria, indicando la motivación de la misma y nombrando las principales tecnologías utilizadas.

Asimismo se hace un listado de objetivos que se quieren alcanzar en la consecución de esta memoria y se expone su estructura.

- *Sección 2: Tecnologías, entornos de ejecución e implementación.*

En la segunda sección, se profundiza en la descripción de las tecnologías utilizadas, haciendo una breve introducción de sus orígenes, por establecer un orden cronológico de las mismas.

También se describe el entorno de ejecución de la plataforma **WebDoc** que es el mismo donde se ha desarrollado y donde se ha comprobado que funciona convenientemente.

Por último, se describen las herramientas informáticas que se han utilizado para elaborar la plataforma y desarrollar esta memoria.

- *Sección 3: Ingeniería Web.*

En la tercera sección, amén de ser la más larga de la memoria, es donde se desarrolla todo el proceso de ingeniería que se ha llevado a cabo para desarrollar la plataforma **WebDoc**.

Se empieza con una breve descripción de lo que se entiende por “Ingeniería Web”, donde se describe, de manera sucinta, todos los fundamentos sobre los que se desarrolla la plataforma **WebDoc**.

A continuación, se introduce un documento de Especificación de Requisitos, el cual es un documento independiente integrado en esta memoria, seguido de los Diagramas de Requerimientos [19] y los Diagramas de Casos de Uso [20].

Seguidamente, se describe el análisis que se ha llevado a cabo sobre la Especificación de Requisitos y que se plasma mediante los Diagramas de Clase del Análisis.

Posteriormente se describe el diseño, realizado sobre el análisis mediante los Diagramas de Clase del Diseño.

Finalmente, se muestra parte del código en la Implementación y se definen los Diagramas de Secuencia realizados para efectuar las pruebas sobre la plataforma **WebDoc**.

En último lugar se presentan diversas capturas de pantalla para presentar la implementación de la plataforma **WebDoc**.

- *Sección 4: Conclusiones finales.*

En esta sección y última, se habla de los objetivos cumplidos, del trabajo futuro a continuar sobre el desarrollo y las conclusiones a las que se ha llegado.

2. Tecnologías, entornos de ejecución e implementación.

2.1. Tecnologías utilizadas.

- **HTML:** Lenguaje de marcado diseñado e implementado por Sir Tim Berners-Lee, a finales de 1989, y presentado a la comunidad en 1990 sobre un ordenador NeXT. Basado en el SGML e HyperText, creando un tipo de HyperText global, utilizando la nomenclatura DNS y con su propio protocolo para la recuperación de archivos mediante enlaces de HyperText (**HTTP**) [6].

- **CSS:** Hojas de estilo en cascada. Mecanismo para añadir apariencia gráfica a los documentos Web (**HTML**). Aunque ya Sir Tim Berners-Lee en 1990 separó la estructura del documento de su representación gráfica (mediante una hoja de estilos muy simple), no fue presentado a la comunidad hasta Abril de 1995 en la conferencia WWW por Håkon Wium Lie y Bert Bos [7].

- **PHP:** Lenguaje de scripting ejecutado en el servidor web, es el sucesor de un producto llamado PHP/FI. Creado en 1994 por Rasmus Lerdorf, la primera versión fue un conjunto simple de archivos binarios CGI escritos en C. Con el tiempo llamó al conjunto de scripts “Personal Home Page Tools”, más conocido como “PHP Tools”. En Julio de 2004 fue lanzado PHP 5, con un núcleo Zend Engine 2.0 que contiene un nuevo modelo de objetos y docenas de nuevas opciones [8].

- **SQL:** En la década de los setenta, después de que E.F. Codd presentara su propuesta sobre el Modelo Relacional, aparecieron varios productos comerciales que seguían esta propuesta [...]. En la década de los 80 el organismo ANSI publicó la primera versión estándar del SQL (Structured Query Language). A partir de ese momento, este organismo publicó varias versiones que mejoraban las deficiencias encontradas en las versiones previas e introducían las mejoras que las investigaciones en el campo de las bases de datos iban descubriendo [9].

- **JavaScript:** Lenguaje de programación interpretado, del que surgió el estándar ECMAScript, se utiliza, usualmente, embebido en documentos HTML o referenciado como archivos .js dentro del código HTML y ejecutado, en un alto porcentaje de los casos, en el navegador del usuario, no en el servidor. Sirve para añadir interacción con el usuario a los documentos estáticos HTML. Fue creado en 10 días en Mayo de 1995 por Brendan Eich, trabajador de Netscape, actualmente en Mozilla. Entre 1996 y 1997 fue llevado ante la European Computer Manufacturers Association (ECMA) (Ecma International) del que surgió el estándar ECMAScript, que, actualmente va por la edición 5.1 [10].



- **JQuery:** Es una biblioteca que añade funcionalidad a JavaScript, creada por John Resig, permite interactuar con los documentos HTML de una manera sencilla: manipular el árbol DOM del HTML, manejar eventos y agregar interacción AJAX [11]. Fue presentada como idea el 22 de Agosto de 2005 [12] y liberada como primera versión estable en Enero de 2006 [13].

- **Joomla!: Joomla!** es un destacado gestor de contenido (o CMS, Content Management System) que permite construir sitios Web y potentes aplicaciones online. Diversos aspectos, incluidos su facilidad de uso y su extensibilidad, hacen a **Joomla!** el software más popular para construir sitios Web [14].

2.2. Entorno de Ejecución.

- **Sistema Operativo:** Linux version 2.6.28.8-20120713a-iscsi-ntacker-3ware-sec8-fmon-igb-dif2-grsec (root@buildd-i386) (gcc version 4.4.5 (Debian 4.4.5-8)) #1 SMP Fri Jul 13 13:37:07 CEST 2012

- **Sistema Gestor de Base de Datos:** MySQL version 5.1.67-log

- **Lenguaje de Desarrollo en Servidor:** PHP version 5.4.16

- **Servidor Web:** Apache

- **Joomla! Version:** Joomla! 3.1.1 Stable [Ember] 26-April-2013 14:00 GMT

- **Joomla! Platform Version:** Joomla Platform 12.2.0 Stable [Neil Armstrong] 21-September-2012 00:00 GMT

2.3. Herramientas de Desarrollo.

- **Sistema Operativo:** Mac OS X Mountain Lion

- **Integrated Development Environment:** NetBeans 7.3

- **Herramienta FTP:** Yummy FTP 1.9.1

- **Herramienta de composición de documentos:** Apple Pages '09

- **Herramienta de gestión de diagramas UML:** Visual Paradigm for UML (Modeler Edition) Version 10.2

- **Herramienta de diseño gráfico y retoque:** Pixelmator 2.2

3. Ingeniería Web

3.1. Introducción.

¿Qué es? Los sistemas y aplicaciones basados en Web (WebApps) ofrecen un conglomerado de contenido y funcionalidad a una gran cantidad de usuarios finales. La ingeniería Web (IWeb) es un proceso por el que se producen WebApps de alta calidad. No es un calco exacto de la Ingeniería del Software pero toma prestadas muchos conceptos y principios fundamentales.

¿Quién lo hace? Ingenieros Web ayudados por desarrolladores de contenido.

¿Por qué es importante? Las WebApps han adquirido gran importancia en los últimos años, ya que se han integrado en los negocios de grandes y pequeñas empresas (por ejemplo, en el comercio electrónico, en la gestión del conocimiento, etc.), por lo que aumenta la importancia de construir sistemas fiables y robustos bajo una disciplina de desarrollo basada en los principios de la ingeniería del software.

¿Cuáles son los pasos? Al igual que cualquier disciplina de ingeniería, la IWeb comienza con una búsqueda de los requisitos por parte de los ingenieros, continuando con un detallado análisis y diseño, pasando a una implementación con tecnologías y herramientas específicas para la Web. Finalmente se revisa el resultado con los clientes y se refina, estableciendo un control de versiones y configuraciones que asegure una calidad de la aplicación y un posible soporte a largo plazo que fije su evolución.

¿Cómo se puede asegurar que el resultado es correcto? Es algo complicado hasta que los usuarios no prueban la aplicación. Esto es debido a que las aplicaciones Web están sometidas a un estrés mayor que una aplicación convencional, debido a que las cargas del sistema son muy variables, yendo desde unas pocas decenas de personas hasta posibles miles, ejecutando todas ellas el sistema concurrentemente. Por ello, el período de prueba de una aplicación web debe ser mucho más amplio que el de una aplicación convencional. Debido a esto último, es muy difícil el establecimiento de versiones sobre una aplicación Web, siendo adoptado, en casi un 100% de los casos, un sistema de iteración continua, donde la última versión de la aplicación es la que está funcionando en ese momento [15].

3.2. Especificación de Requisitos Software.

3.2.1. Especificación de Requisitos Software.

NOTA: Debido a la naturaleza de documento independiente (IEEE Std. 830-1998), este punto tendrá su propia numeración, recuperándose la numeración correlativa de esta memoria en el punto 3.2.2.



Especificación de Requisitos Software

(IEE Std. 830-1998)

(Revisión del IEEE Std. 830-1993)

1.Introducción.

1.1. Propósito.

El propósito de esta Especificación de Requisitos Software (**ERS**) es definir cuáles son los requerimientos que debe tener una plataforma Web que se encargue de la gestión documental en una organización.

Se toma, como modelo, una posible gestión documental para el Departamento de Informática de Sistemas y Computadores (DISCA) de la Universidad Politécnica de Valencia.

1.2. Ámbito.

La plataforma que vamos a desarrollar se identifica con el nombre **WebDoc**. La misma se encargará de unificar en un único lugar, clasificar (por tipo de documento y por categoría, hasta un número de 10 por tipo de documento) y presentar los documentos, que los usuarios de la misma decidan almacenar. Las 10 categorías por tipo de archivo, aunque fijas en número, podrán cambiar su nombre, desde la parte de administración de la plataforma, para facilitar su identificación por los usuarios.

Mediante una nomenclatura para el nombre del archivo, definida a continuación, la plataforma hará inventario de cada archivo, colocándolo en su categoría correspondiente. Si el archivo no cumple con esta nomenclatura, la plataforma se encargará de clasificar el archivo en una categoría superior genérica.

NOMENCLATURA DEL ARCHIVO
<p style="text-align: center;">[Nombre del archivo] _ [Nº].[extensión]</p> <p>Donde:</p> <p>[Nombre del archivo] := Nombre del archivo que queramos ponerle, sin espacios en blanco y sin incluir el guión bajo [_]</p> <p>[Nº] := [1..10] Que corresponde, de 1 a 10, con el número de categoría, de entre las 10 posibles que existen por tipo de archivo.</p> <p>[extensión] := Extensión del archivo.</p>

En ambos casos (cumpliendo la nomenclatura y no cumpliendo) se hará una clasificación previa de los archivos, basada en el tipo de archivo que, a su vez, vendrá determinada por su extensión.

Los tipos de archivo que se contemplan para la plataforma son:

- Portable Document Format (PDF)
- Microsoft® Word
- Microsoft® Excel®
- Microsoft® PowerPoint®
- Open Document Text
- Open Document Presentation
- Open Document Spreadsheet
- Other (*Otro tipo de documentos que no se encuentran reflejados en el resto de categorías*)

La plataforma informará, mediante correo electrónico a todos los administradores que tienen activado el envío de mensajes del sistema, de todas las operaciones que ejecuten los usuarios autorizados.

1.3. Definiciones, acrónimos y abreviaturas.

ERS: Especificación de Requisitos Software.

Archivo: Equivalente a un archivo en un sistema informático. Archivo con un nombre y extensión, y que se encuentra en una URL de la plataforma, o está en proceso de integrarse en la misma.

Documento: Artículo en **Joomla!** que lleva asociado un archivo de forma unívoca. El artículo en **Joomla!** se crea automáticamente cuando se envía un archivo a la plataforma.

UUID: Identificador Universalmente Único (Universally Unique Identifier, aunque de forma precisa es un pseudo-UUID) creado para la plataforma y que está basado en la generación de una cadena de bytes pseudo-aleatoria.

Crear un documento: Proceso por el cual, se selecciona un archivo del ordenador del usuario, se envía, mediante formulario integrado, a la plataforma **WebDoc**, y la plataforma, crea un artículo en **Joomla!** automáticamente, que lleva asociado dicho archivo de manera unívoca, al tiempo que clasifica dicho archivo, basándose en la nomenclatura del mismo y en su extensión. Se pueden crear documentos con el mismo nombre y la misma categoría, incluso por el mismo usuario, si dichos documentos tienen archivos asociados en diferentes sitios del sistema de archivos del servidor. Se manda un correo de aviso a todos los administradores que tienen activado el envío de mensajes del sistema.

Borrar un documento: Proceso por el cual se elimina un archivo del espacio físico del servidor de la plataforma **WebDoc**, se marca el artículo de **Joomla!** con una marca de tiempo unívoca y se marca el artículo como “trasladado a la papelera”. Dicha marca unívoca de tiempo servirá a los administradores para llevar un control de las bajas de los archivos en la plataforma. Se manda un correo de aviso a todos los administradores que tienen activado el envío de mensajes del sistema.

Actualizar un documento: Proceso por el cual, un archivo del espacio físico del servidor se actualiza, sin crear un nuevo documento en el sistema. Esto ocurre cuando ya hay creado un documento en el sistema con ese nombre, en esa categoría y en el mismo espacio físico del sistema de archivos del servidor. Se manda un correo de aviso a todos los administradores que tienen activado el envío de mensajes del sistema.

Documents Manager: Módulo integrado en **Joomla!**, que contiene toda la lógica de la plataforma para crear, actualizar, borrar documentos y enviar correos a los administradores, que tienen activado el envío de mensajes del sistema, con la actividad de la plataforma. Es un módulo sensible al contexto, y muestra a un usuario sólo su directorio donde puede subir sus archivos. Si el usuario es un usuario privilegiado (Super User) entonces le muestra el directorio raíz (UpLoad) de donde cuelgan todos los directorios, de todos los usuarios que han hecho uso del módulo subiendo algún archivo. Cada usuario tiene su propio directorio (colgando del raíz) cuyo nombre es igual al nombre del usuario.

1.4. Referencias.

[1] ANSI/IEEE Std. 830-1998 Recommended Practice for Software Requeriments Specifications

1.5. Visión global.

En el resto de la especificación de requisitos software vamos a realizar una descripción general del producto que vamos a desarrollar: funciones, características del usuario, restricciones y describir requisitos específicos.



2. Descripción general.

2.1. *Perspectiva del producto.*

La aplicación **WebDoc** debe aunar, en un único lugar, la gestión documental de una organización, utilizando tecnologías web, para asegurar su independencia de la plataforma del cliente y poder llevar un control de accesos al sistema por los usuarios, así como una gestión de la unicidad de documentos a nivel de URL.

2.2. *Funciones del producto.*

1. Menú Documentos

1. Gestor de Documentos

2. Menú Usuario

1. Mi perfil

2. Salir del sistema

3. Menú Tipos de Archivo

1. Portable Document Format (PDF)

- "Categoría 1"
- "Categoría 2"
- "Categoría 3"
- "Categoría 4"
- "Categoría 5"
- "Categoría 6"
- "Categoría 7"
- "Categoría 8"
- "Categoría 9"
- "Categoría 10"

2. Microsoft® Word

- "Categoría 1"
- [...]
- "Categoría 10"

3. Microsoft® Excel®

- "Categoría 1"
- [...]
- "Categoría 10"

4. Microsoft® PowerPoint®

- "Categoría 1"
- [...]
- "Categoría 10"

5. Open Document Text

- ☑ “Categoría 1”
- ☑ [...]
- ☑ “Categoría 10”

6. Open Document Presentation

- ☑ “Categoría 1”
- ☑ [...]
- ☑ “Categoría 10”

7. Open Document Spreadsheet

- ☑ “Categoría 1”
- ☑ [...]
- ☑ “Categoría 10”

8. Other

- ☑ “Categoría 1”
- ☑ [...]
- ☑ “Categoría 10”

4. Menú Búsqueda

2.3. Características del usuario.

Esta plataforma la utilizará todo el personal, de una organización, que necesite llevar una gestión documental. Además deberá haber un usuario Administrador (Super User) que se encargará de autorizar usuarios y llevar un control de las altas y bajas de documentos. Aunque la herramienta es casi auto-gestionada (con la excepción de los permisos de usuario) debe haber un usuario Administrador que entre de forma esporádica en el panel de control de la herramienta (backend).

Aunque la plataforma está diseñada sobre tecnologías web, un usuario, para una utilización habitual y elemental de la herramienta, no es necesario que sepa de código sobre tecnologías web. El usuario Administrador (Super User), sin embargo, sería muy conveniente que supiera algo de desarrollo sobre **HTML** [2] y **PHP** [1], debido a que los módulos internos de la herramienta tienen código de este tipo (que, aunque no es necesario que se modifiquen, sí puede haber errores al intentar personalizar alguno de ellos mediante el cambio de su código).

2.4. Restricciones generales.

Debido a la orientación a URL de la plataforma y al método semi-automático diseñado para crear documentos en la misma, se han de tener en cuenta las siguientes restricciones:

Archivos: Los archivos se convertirán en documentos en la plataforma **WebDoc** y, dichos documentos tienen una estructura de URL. Por lo tanto, la nomenclatura de los archivos deberá contemplar las siguientes restricciones:



- El nombre del archivo no puede contener espacios en blanco.
- El nombre del archivo no puede contener el símbolo “guión bajo” [_], excepto para indicar la categoría a la que pertenecerá el futuro documento, tal y como se indica en **1.2 - NOMENCLATURA DEL ARCHIVO**.

Carpetas: Las carpetas contendrán archivos que se convertirán en documentos en la plataforma **WebDoc** y, dichos documentos tienen una estructura de URL. Por lo tanto, la nomenclatura de las carpetas deberá contemplar las siguientes restricciones:

- El nombre de la carpeta no podrá contener espacios en blanco.

2.5. Supuestos y dependencias.

La plataforma **WebDoc** se supone que tiene asignada una URL completa y, por lo tanto, tiene asociado un registro en un servidor de DNS, como por ejemplo <http://webdoc.upv.es/>, ya que es necesaria para realizar copias de seguridad y migraciones de la plataforma.

3. Requisitos Específicos.

3.1. Requisitos de interfaces externos.

3.1.1. Interfaces de usuario.

Aplicación Web donde los usuarios podrán seleccionar todos los menús descritos en **2.2 Funciones del producto** que conforman la plataforma **WebDoc**.

3.1.2. Interfaces hardware.

Requisitos	Procesador	Memoria	Conexión Local
Mínimo	[Intel® Xeon® CPU X3323 @ 2.50GHz] x3	12 GB	1 Gb

3.1.3. Interfaces software.

S.O.	Linux version 2.6.28.8-20120713a-iscsi-ntacker-3ware-sec8-fmon-igb-dif2-grsec (root@buildd-i386) (gcc version 4.4.5 (Debian 4.4.5-8)) #1 SMP Fri Jul 13 13:37:07 CEST 2012
SGBD	MySQL version 5.1.67-log
Lenguaje Servidor	PHP version 5.4.16
Servidor Web	Apache
Joomla! Version	Joomla! 3.1.1 Stable [Ember] 26-April-2013 14:00 GMT
Joomla! Platform Version	Joomla! Platform 12.2.0 Stable [Neil Armstrong] 21-September-2012 00:00 GMT

3.1.4. Interfaces de comunicaciones.

Es una plataforma web, usada como intranet, y la interfaz de comunicación está sustentada por el protocolo TCP/IP.

3.2. Requisitos funcionales.

3.2.1. Modo Creación De Documentos.

CREAR_CARPETA	
ENTRADAS	Nombre de la carpeta.
PROCESO	Crea una nueva carpeta [sin espacios en blanco] en la carpeta personal del usuario. Si el usuario es un usuario Administrador (Super User) la carpeta se creará en el punto donde se encuentre el directorio de trabajo actual.
SALIDAS	Confirmación de que la operación se ha llevado con éxito. < O > Mensaje de error.

CREAR_DOCUMENTO	
ENTRADAS	Archivo [cuyo nombre sigue la nomenclatura de 1.2 - NOMENCLATURA DEL ARCHIVO]. Departamento (opcional). Descripción (opcional).
PROCESO	<p>Se crea un documento en WebDoc, clasificando el mismo de la siguiente forma:</p> <ul style="list-style-type: none"> 1.- Se extrae, de su extensión, información acerca del tipo de archivo que se está manejando. <ul style="list-style-type: none"> 1.1.- Si la extensión no está soportada por la plataforma, el documento se clasificará como "Other" (Otros). 2.- Se extrae, de su nombre, a qué categoría (de las 10 posibles para cada tipo de archivo) debe pertenecer el documento asociado a dicho archivo. <ul style="list-style-type: none"> 2.1.- Si no es posible extraer a qué categoría pertenece, por error u omisión, se clasifica el documento como perteneciente a la supercategoría de cada tipo de archivo. <p>Si las entradas Departamento y Descripción están disponibles, se adjuntan a la información en el documento.</p> <p>Se crea un código BIDI único, con la URL del archivo (que no del documento) para permitir una descarga directa del archivo a un dispositivo móvil.</p> <p>Se crea un UUID para el archivo y se adjunta a la información en el documento y se crea un enlace gráfico para poder descargar el archivo desde cualquier dispositivo.</p> <p>Finalmente, se sube el archivo al sistema de archivos del servidor mediante el uso de un formulario HTML, y se asocia un icono representativo del tipo de archivo que se está manejando.</p>
SALIDAS	<p>Confirmación de que la operación ha sido realizada con éxito. < O > Mensaje de error.</p> <p>Se manda un correo de aviso a todos los administradores que tienen activado el envío de mensajes del sistema. < O > Mensaje de error de que ha sido imposible el envío del correo.</p>

ELIMINAR_CARPETA	
ENTRADAS	Nombre de la carpeta (mediante selección gráfica)
PROCESO	Se elimina una carpeta (vacía) de la carpeta personal del usuario (o del directorio de trabajo actual del sistema, si el usuario es un usuario Administrador [Super User]).
SALIDAS	<p><SI> [la carpeta está llena] Aviso de que la carpeta está llena</p> <p><SI NO> Confirmación de que la operación se ha llevado con éxito < O > Mensaje de error</p>

ELIMINAR_DOCUMENTO	
ENTRADAS	Nombre del archivo (mediante selección gráfica).
PROCESO	Se elimina el archivo del sistema de archivos del servidor. Se mueve el documento de WebDoc a la papelera, y se le añade una marca de fecha y tiempo (medido en milisegundos) para que los usuarios Administradores (Super Users) puedan llevar un registro de documentos eliminados.
SALIDAS	<p>Confirmación de que la operación ha sido realizada con éxito. < O > Mensaje de error.</p> <p>Se manda un correo de aviso a todos los administradores que tienen activado el envío de mensajes del sistema. < O > Mensaje de error de que ha sido imposible el envío del correo.</p>

3.2.2. Modo Visualización De Documentos.

LISTAR_TODOS_LOS_DOCUMENTOS	
ENTRADAS	Proceso de Autenticación. <O> Selección gráfica del icono WebDoc .
PROCESO	Después de que un usuario introduce su usuario y contraseña, la plataforma WebDoc muestra un listado de todos los documentos que hay disponibles para su visualización. Si no existe ningún documento en la plataforma, WebDoc mostrará un resultado en blanco.
SALIDAS	Listado gráfico.

LISTAR_TODOS_LOS_DOCUMENTOS_DE_UN_TIPO	
ENTRADAS	Seleccionar el tipo de documento (mediante selección gráfica).
PROCESO	Se muestran todos los documentos que corresponden con el tipo de documento seleccionado en la entrada. Si no existe ningún documento en la plataforma, para el tipo de documento seleccionado, WebDoc mostrará un resultado en blanco.
SALIDAS	Listado gráfico.

LISTAR_UNA_CATEGORIA_DADO_UN_TIPO_DE_DOCUMENTO	
ENTRADAS	Seleccionar una categoría dentro de un tipo de documento (mediante selección gráfica).
PROCESO	Se muestran todos los documentos que pertenecen a la categoría seleccionada en la entrada y cuyo tipo de documento coincide con el tipo de documento al que pertenece la categoría. Si no existe ningún documento en la plataforma, para el tipo de documento seleccionado, WebDoc mostrará un resultado en blanco.
SALIDAS	Listado gráfico.

BUSCAR_DOCUMENTO	
ENTRADAS	Palabras clave a buscar.
PROCESO	Se muestran todos los documentos que coinciden con las palabras clave de la entrada. Cada documento tiene una serie de campos, por cuya información puede buscarse en la plataforma. Dichos campos son: <ul style="list-style-type: none"> - Nombre (nombre del archivo). - ID (UUID). - Usuario (su correo electrónico). - Departamento. - Descripción.
SALIDAS	Listado.

CREAR_PERFIL_USUARIO	
ENTRADAS	Nombre, nombre de usuario, contraseña, email, tipo de editor html, franja horaria, idioma de la plataforma.
PROCESO	Cuando un usuario se da de alta (él mismo) se registra en la plataforma perteneciendo, por defecto, al grupo "Registered". Básicamente, este grupo, dentro de la plataforma WebDoc no puede hacer nada, excepto ver la información de su perfil y actualizarla.
SALIDAS	Se muestra el perfil.

ACTUALIZAR_PERFIL_USUARIO	
ENTRADAS	Nombre, nombre de usuario, contraseña, email, tipo de editor html, franja horaria, idioma de la plataforma.
PROCESO	Se actualiza la información del perfil del usuario en WebDoc según lo que haya puesto el usuario en el formulario que concuerda con las entradas.
SALIDAS	Se muestra el perfil.

3.2.3. Modo Administración.

LISTAR_REGISTROS_DOCUMENTOS_BORRADOS	
ENTRADAS	Seleccionar: Articles > Filter: Trashed (mediante selección gráfica)
PROCESO	La plataforma filtra todos los documentos que están en estado “trashed”.
SALIDAS	Listado de todos los documentos que han sido borrados en el sistema, con su autor, la fecha de creación y la fecha y hora de borrado.

ELIMINAR_REGISTRO_DOCUMENTO_BORRADO	
ENTRADAS	Seleccionar: Articles > Filter: Trashed (mediante selección gráfica) Seleccionar uno o más documentos
PROCESO	Elimina el registro de documento borrado.
SALIDAS	(Ninguna)

CREAR_PERFIL_USUARIO	
ENTRADAS	Nombre, nombre de usuario, contraseña, email, tipo de editor html, franja horaria, idioma de la plataforma.
PROCESO	Se crea el perfil de un nuevo usuario, asignándolo al grupo “Publisher”, si así se necesitara, para darle acceso inmediato a la plataforma WebDoc .
SALIDAS	Se envía un correo a la dirección indicada en la creación, de que se ha creado un usuario con dicha dirección de correo.

ACTUALIZAR_PERFIL_USUARIO	
ENTRADAS	Nombre, nombre de usuario, contraseña, email, tipo de editor html, franja horaria, idioma de la plataforma.
PROCESO	Se actualiza la información del perfil del usuario en WebDoc según lo que haya puesto el administrador en el formulario que concuerda con las entradas.
SALIDAS	Se muestra el perfil.

AUTORIZAR_PERFIL_USUARIO	
ENTRADAS	Seleccionar un usuario (mediante selección gráfica)
PROCESO	Cambiar a un usuario de grupo “Registered” a “Publisher”, autorizando, con dicho cambio, a ese usuario, a usar la plataforma WebDoc .
SALIDAS	Listado de todos los grupos a los que pertenece el usuario.

DESAUTORIZAR_PERFIL_USUARIO	
ENTRADAS	Seleccionar un usuario (mediante selección gráfica)

DESAUTORIZAR_PERFIL_USUARIO	
PROCESO	Cambiar a un usuario de grupo “Publisher” a “Registered”, desautorizando, con dicho cambio, a ese usuario, a usar la plataforma WebDoc .
SALIDAS	Listado de todos los grupos a los que pertenece el usuario.
BORRAR_PERFIL_USUARIO	
ENTRADAS	Seleccionar un usuario (mediante selección gráfica)
PROCESO	Se elimina el perfil de un usuario, eliminándolo de la plataforma WebDoc .
SALIDAS	(Ninguna)

3.3. Requisitos de rendimiento.

No se han determinado Requisitos de eficiencia, aunque es recomendable que se intente optimizar todo lo posible el servidor web **Apache** [16] y la base de datos **MySQL** [17], puesto que **WebDoc** está implementado sobre una plataforma compleja (**Joomla!** [5]), sobre un lenguaje de scripting pesado, como se considera en este documento a **PHP** [1].

3.4. Restricciones de diseño.

3.4.1. Estándares cumplidos.

La plataforma **WebDoc** intenta seguir los estándares HTML5 [2] y el CSS3 [3], ambos no completamente especificados, a fecha de Julio de 2013.

3.4.2. Limitaciones hardware.

Las limitaciones hardware de la plataforma **WebDoc** son las limitaciones de poder instalar las herramientas que la plataforma necesita y que están especificadas en **3.1.2 Interfaces software**.

3.5. Atributos.

3.5.1. Seguridad.

La seguridad en **WebDoc**, viene fijada por las directrices de seguridad definidas en la plataforma **Joomla!** [5].

Por defecto, los usuarios de la plataforma **WebDoc**, se crean (sea auto-gestionado por el propio futuro usuario de la plataforma, sea dado de alta por un usuario Administrador (Super User)) perteneciendo al grupo “Registered”. Posteriormente, un usuario Administrador (Super User) cambiará a cada usuario nuevo del grupo “Registered” al grupo

“Publisher”. De esta forma, los usuarios en el grupo “Publisher” podrán usar la plataforma **WebDoc**.

Todo cambio en un documento en **WebDoc**, creación de uno nuevo y borrado, será comunicado, mediante correo electrónico a todos los usuarios Administradores (Super User) que tengan activado el envío de mensajes del sistema.

3.5.2 Mantenimiento.

El mantenimiento se facilitará en lo posible, incluyendo abundantes comentarios en el código fuente de la aplicación, además de disponer de la presente **ERS**.

Desde el punto de vista de los datos, habría que auditar ocasionalmente el funcionamiento de la plataforma **Joomla!** desde la administración (backend) mediante un usuario Administrador (Super User).

El log de archivos borrados, podría resultar muy largo con el tiempo, por lo que habría que vaciar las entradas más antiguas usando la papelera estándar de **Joomla!** (*Articles > Filter: Trashed*).

3.6. Otros requisitos.

3.6.1. Bases de datos.

Debido a que la plataforma **WebDoc** está desarrollada sobre **Joomla!**, en la base de datos se almacena, tanto la información de los usuarios de la plataforma, como los *metadatos* de los documentos creados en la plataforma, así como, también, gran parte de la propia plataforma **WebDoc**. Es decir, si hubiera un fallo en la base de datos, no sólo se perderían los datos de los usuarios que hacen login y los metadatos de los documentos, sino que se perdería gran parte de la plataforma **WebDoc**, ya que la parte de lógica de la aplicación, está definida en la misma base de datos que está definida **Joomla!**

Por ello se recomienda encarecidamente que se realicen copias de seguridad periódicas, tanto del sistema de archivos del servidor (que es donde se encuentran los archivos convertidos en documentos en **WebDoc**) como de la base de datos (que están todos los datos puntualizados en el párrafo anterior).

Para facilitar dicha tarea, se ha incluido una herramienta de clonado de plataformas **Joomla!** en la plataforma **WebDoc** [18].

[FIN]

Especificación de

Requisitos

Software

3.2.2. Diagramas de requerimientos.

3.2.2.1. Introducción.

Un Diagrama de requerimientos, basado en el estándar **SysML**, muestra visualmente las relaciones construidas entre los requerimientos y los elementos del modelo que cumplan y verifiquen dichos requerimientos.

Los Diagramas de requerimientos pueden usarse para especificar y administrar los requerimientos tanto funcionales como no-funcionales (por ejemplo, el rendimiento).

Nota de Uso: Los Diagramas de requerimientos que especifican requerimientos funcionales pueden solapar semánticamente los Diagramas de Casos de Uso que definan funciones similares [19].

3.2.2.2. Modo Creación De Documentos.

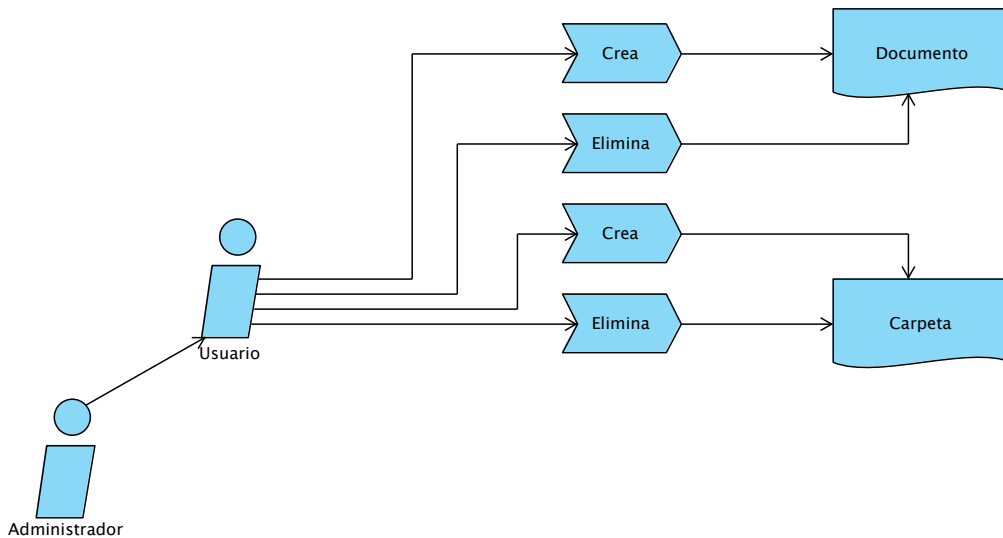


Figura 1. Diagrama de Requerimientos para el modo Creación De Documentos.

3.2.2.3. Modo Visualización De Documentos.

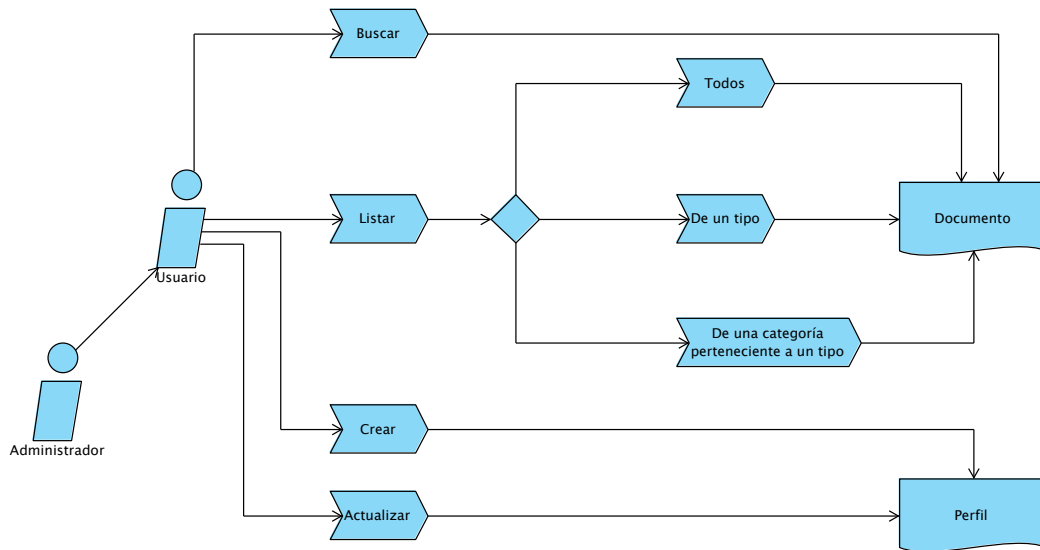


Figura 2. Diagrama de Requerimientos para el modo Visualización De Documentos.

3.2.2.4. Modo Administración.

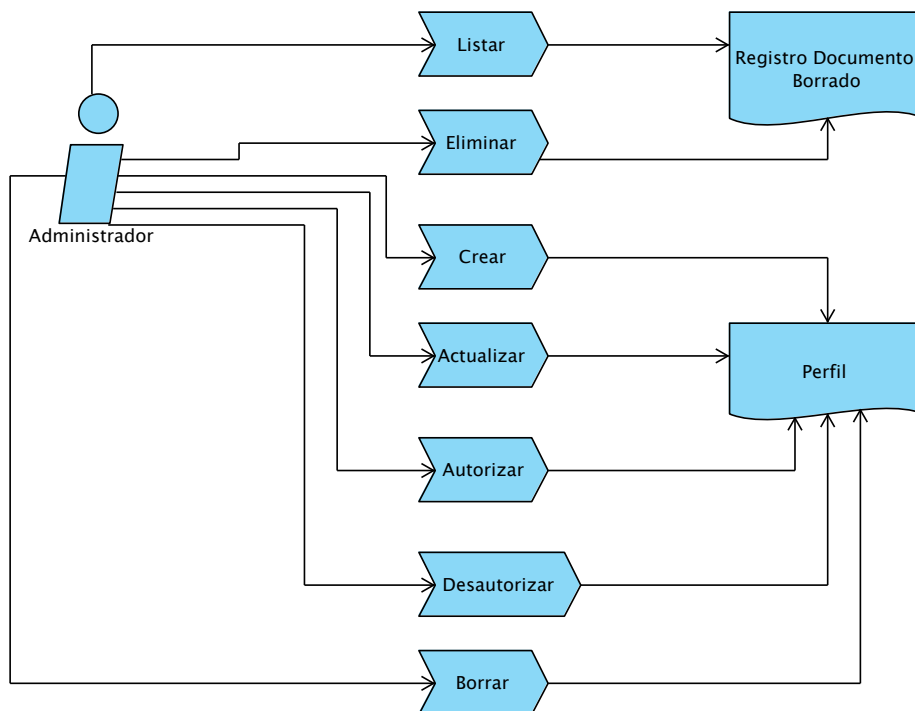


Figura 3. Diagrama de Requerimientos para el modo Administración.

3.2.3. Diagramas de casos de uso.

3.2.3.1. Introducción.

El esfuerzo principal en la fase de requisitos es desarrollar un modelo del sistema que se va a construir, y la utilización de los casos de uso es una forma adecuada de crear ese modelo. Esto es debido a que los requisitos funcionales se estructuran de forma natural mediante casos de uso, y a que la mayoría de los otros requisitos no funcionales son específicos de un solo caso de uso, y pueden tratarse en el contexto de ese caso de uso [...].

Los casos de uso proporcionan un medio intuitivo y sistemático para capturar los requisitos funcionales con un énfasis especial en el valor añadido para cada usuario individual o para cada sistema externo. Mediante la utilización de los casos de uso, los analistas se ven obligados a pensar en términos de quiénes son los usuarios y qué necesidades u objetivos de la empresa pueden cumplir [20].

3.2.3.2. Modo Creación De Documentos.

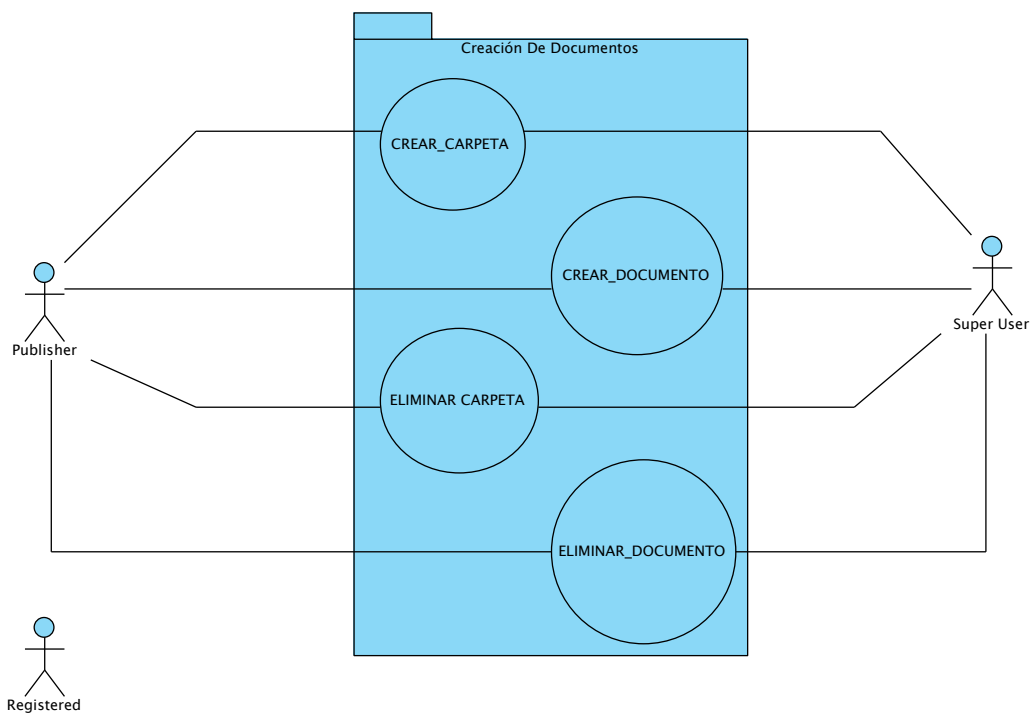


Figura 4. Diagrama de Casos de Uso para el modo Creación De Documentos.

3.2.3.3. Modo Visualización De Documentos.

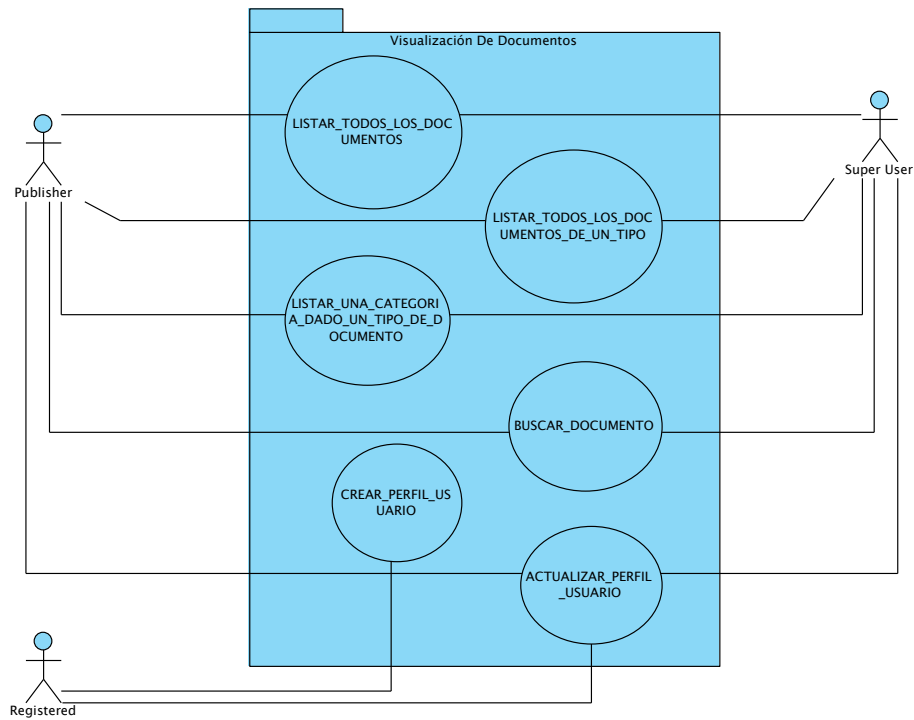


Figura 5. Diagrama de Casos de Uso para el modo Visualización De Documentos.

3.2.3.4. Modo Administración.

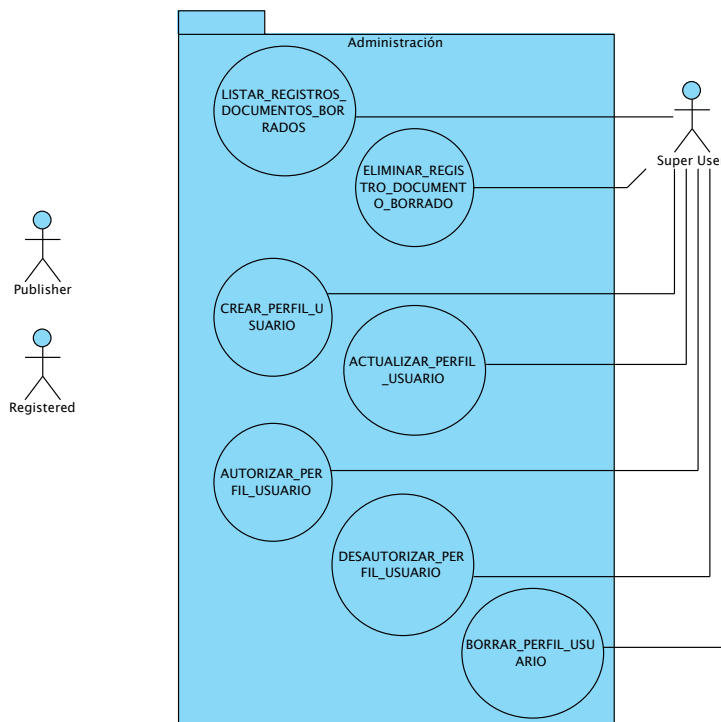


Figura 6. Diagrama de Casos de Uso para el modo Administración.

3.3. Análisis.

3.3.1. Introducción.

Durante el análisis, analizamos los requisitos que se describieron en la captura de requisitos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que nos ayude a estructurar el sistema entero, incluyendo su arquitectura [21].

3.3.2. Modo Creación De Documentos.

3.3.2.1. Análisis del caso de uso

CREAR_CARPETA

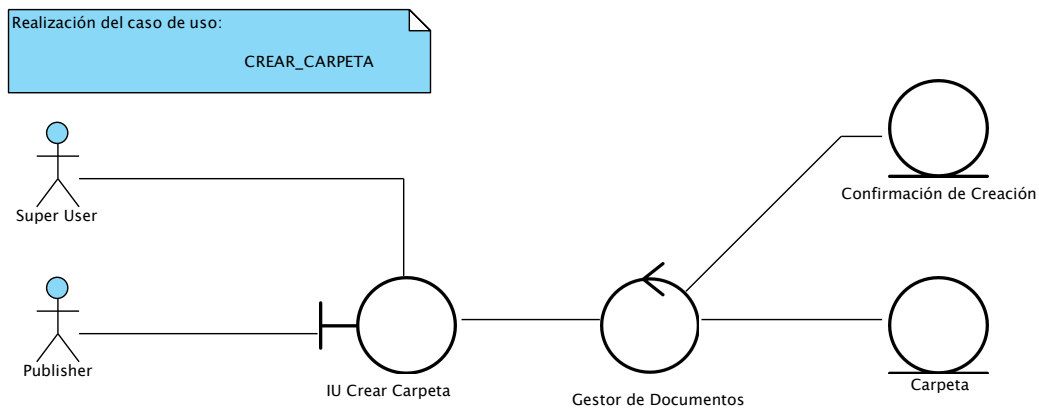


Figura 7. Análisis del Caso de Uso **CREAR_CARPETA**

3.3.2.2. Análisis del caso de uso

CREAR_DOCUMENTO

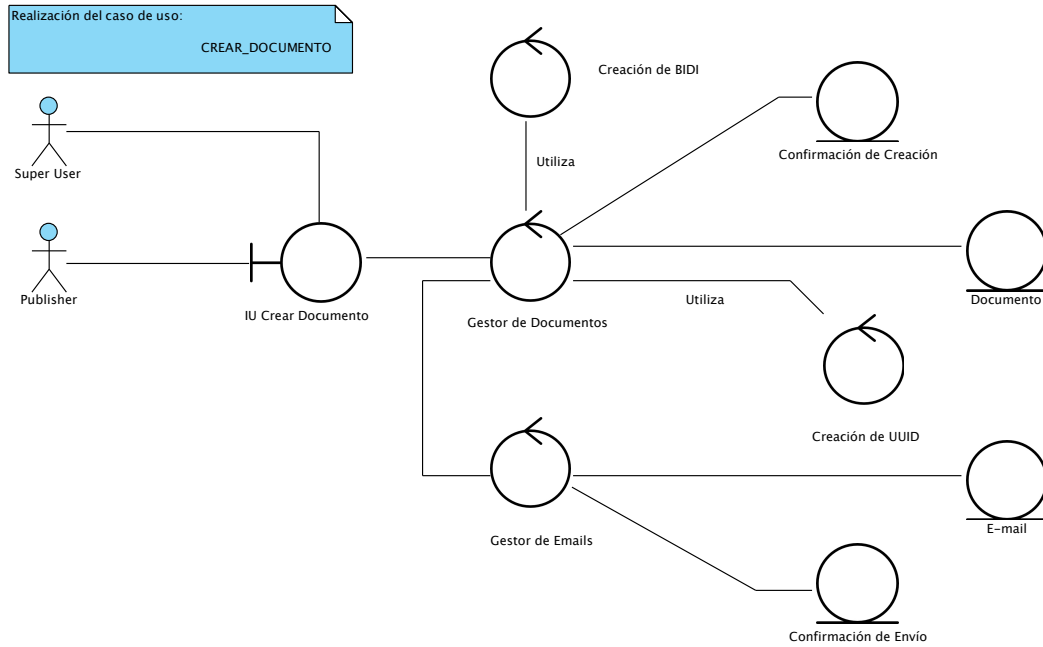


Figura 8. Análisis del Caso de Uso **CREAR_DOCUMENTO**

3.3.2.3. Análisis del caso de uso

ELIMINAR_CARPETA

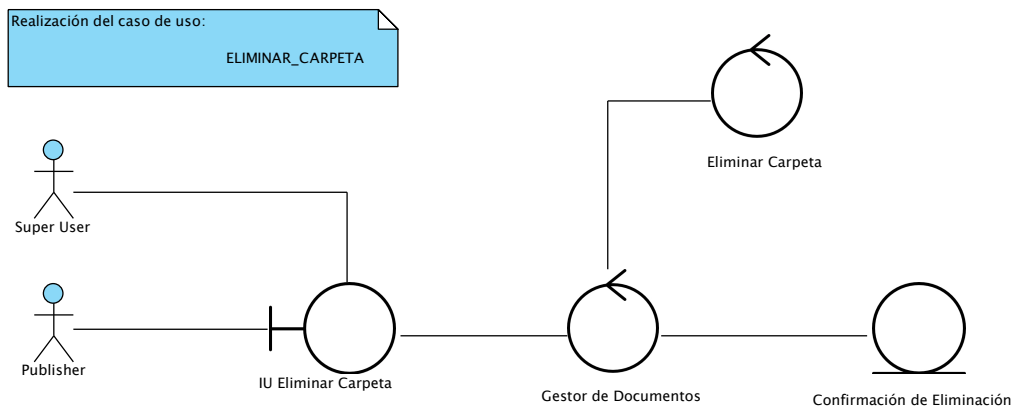


Figura 9. Análisis del Caso de Uso **ELIMINAR_CARPETA**

3.3.2.4. Análisis del caso de uso

ELIMINAR_DOCUMENTO

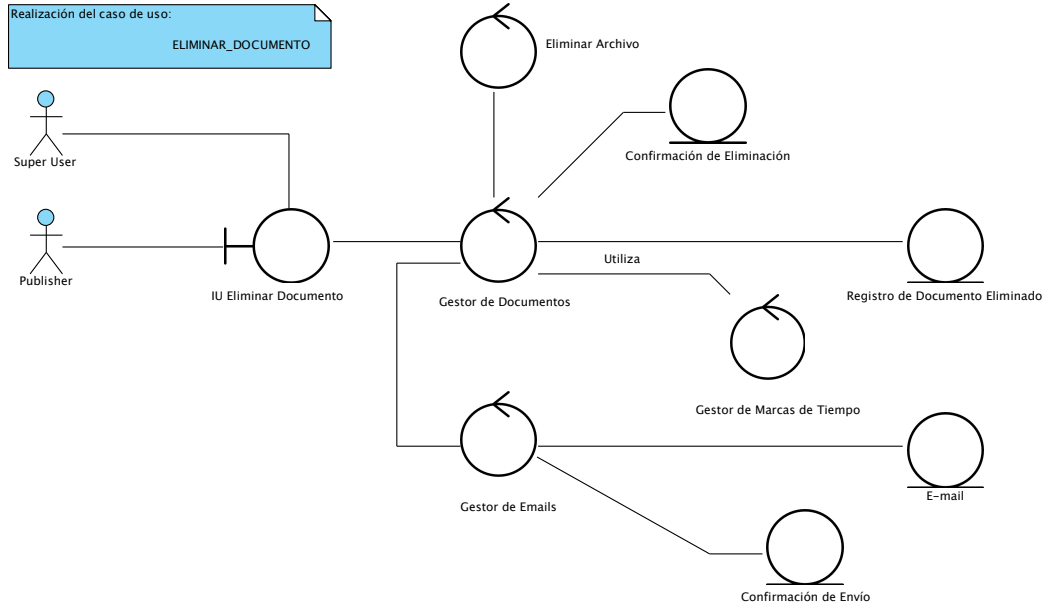


Figura 10. Análisis del Caso de Uso **ELIMINAR_DOCUMENTO**

3.3.3. Modo Visualización De Documentos.

3.3.3.1. - 3.3.3.3. Análisis de los caso de uso

LISTAR_TODOS_LOS_DOCUMENTOS

LISTAR_TODOS_LOS_DOCUMENTOS_DE_UN_TIPO

LISTAR_UNA_CATEGORIA_DADO_UN_TIPO_DE_DOCUMENTO

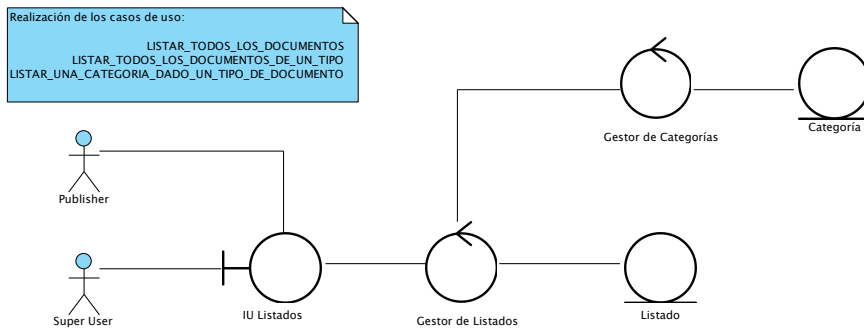


Figura 11. Análisis de los Casos de Uso

LISTAR_TODOS_LOS_DOCUMENTOS

LISTAR_TODOS_LOS_DOCUMENTOS_DE_UN_TIPO

LISTAR_UNA_CATEGORIA_DADO_UN_TIPO_DE_DOCUMENTO



NOTA: Los tres casos de uso presentados en este punto 3.3.3.1 comparten el mismo diagrama de clases de análisis. La razón es que, aunque visualmente haya una gran diferencia, y sean 3 casos de uso diferente, desde el punto de vista de la arquitectura, pueden representarse con las mismas clases de análisis. Esto es debido a que internamente, la plataforma **WebDoc** diferencia los listados que se muestran al usuario únicamente por el número de categoría. Visualmente tiene mucho impacto sobre el usuario, pero internamente, en la plataforma **WebDoc** sólo implica utilizar un módulo u otro.

3.3.3.4. Análisis del caso de uso

BUSCAR_DOCUMENTO

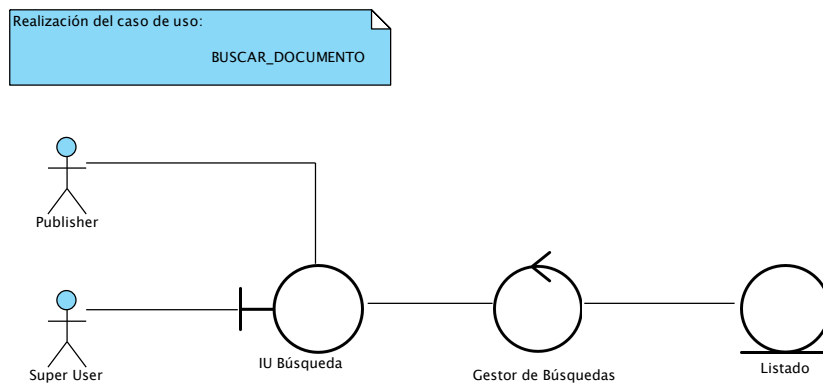


Figura 12. Análisis del Caso de Uso **BUSCAR_DOCUMENTO**

3.3.3.5. Análisis del caso de uso

CREAR_PERFIL_USUARIO

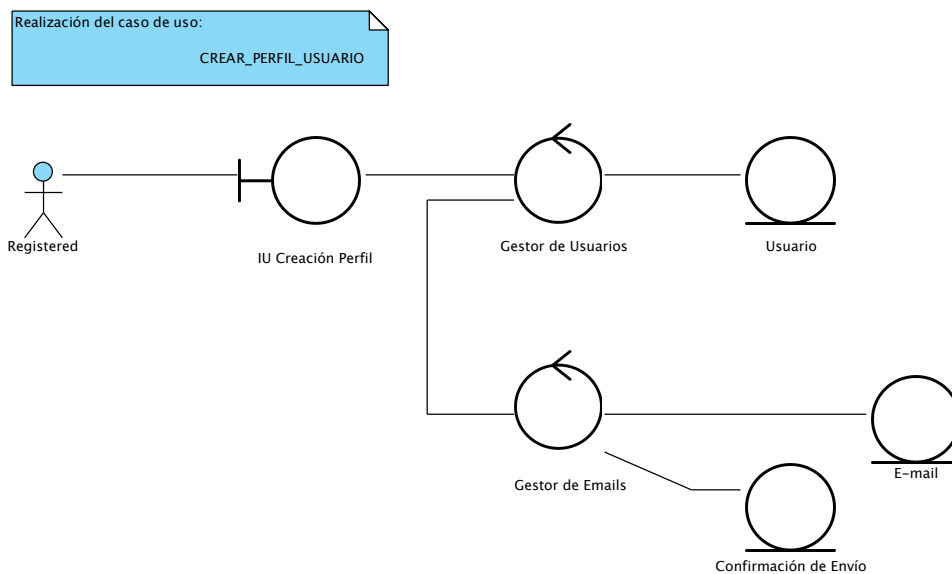


Figura 13. Análisis del Caso de Uso **CREAR_PERFIL_USUARIO**

3.3.3.6. Análisis del caso de uso

ACTUALIZAR_PERFIL_USUARIO

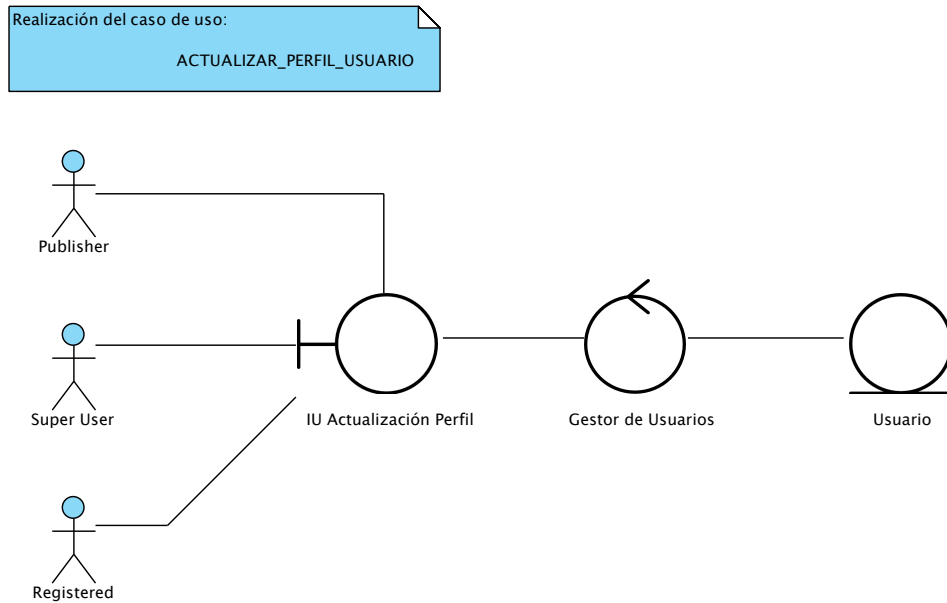


Figura 14. Análisis del Caso de Uso **ACTUALIZAR_PERFIL_USUARIO**

3.3.4. Modo Administración.

3.3.4.1. Análisis del caso de uso

LISTAR_REGISTROS_DOCUMENTOS_BORRADOS

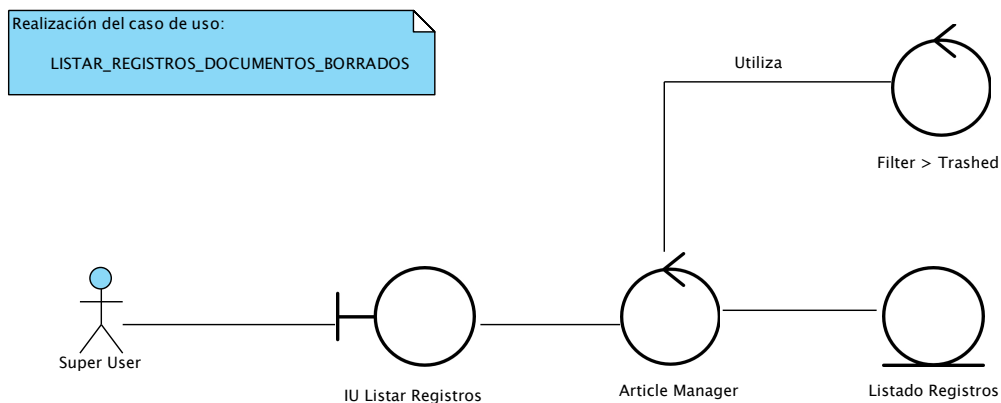


Figura 15. Análisis del Caso de Uso **LISTAR_REGISTROS_DOCUMENTOS_BORRADOS**

3.3.4.2. Análisis del caso de uso

ELIMINAR_REGISTRO_DOCUMENTO_BORRADO

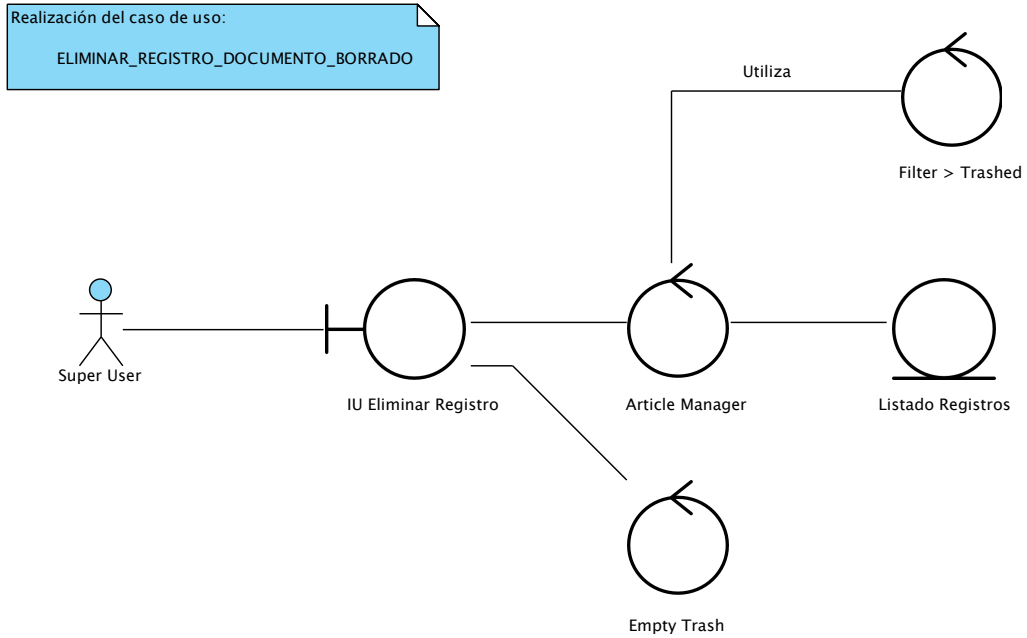


Figura 16. Análisis del Caso de Uso **ELIMINAR_REGISTRO_DOCUMENTO_BORRADO**

3.3.4.3. Análisis del caso de uso

CREAR_PERFIL_USUARIO

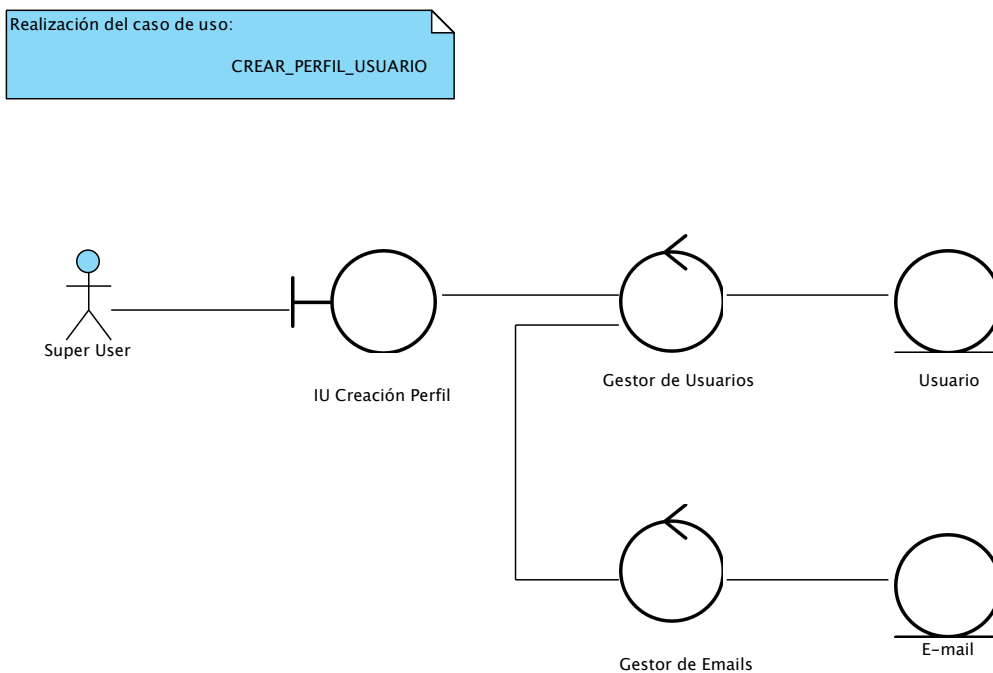


Figura 17. Análisis del Caso de Uso **CREAR_PERFIL_USUARIO**

3.3.4.4. Análisis del caso de uso

ACTUALIZAR_PERFIL_USUARIO

Realización del caso de uso:
ACTUALIZAR_PERFIL_USUARIO

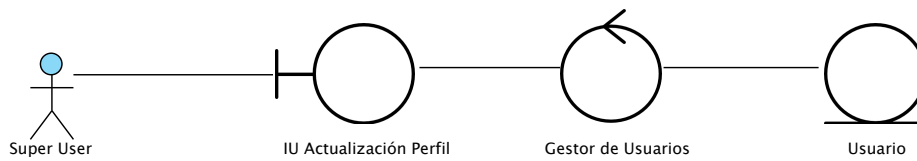


Figura 18. Análisis del Caso de Uso **ACTUALIZAR_PERFIL_USUARIO**

3.3.4.5. - 3.3.4.6. Análisis de los casos de uso

AUTORIZAR_PERFIL_USUARIO

DESAUTORIZAR_PERFIL_USUARIO

Realización de los casos de uso:
AUTORIZAR_PERFIL_USUARIO
DESAUTORIZAR_PERFIL_USUARIO

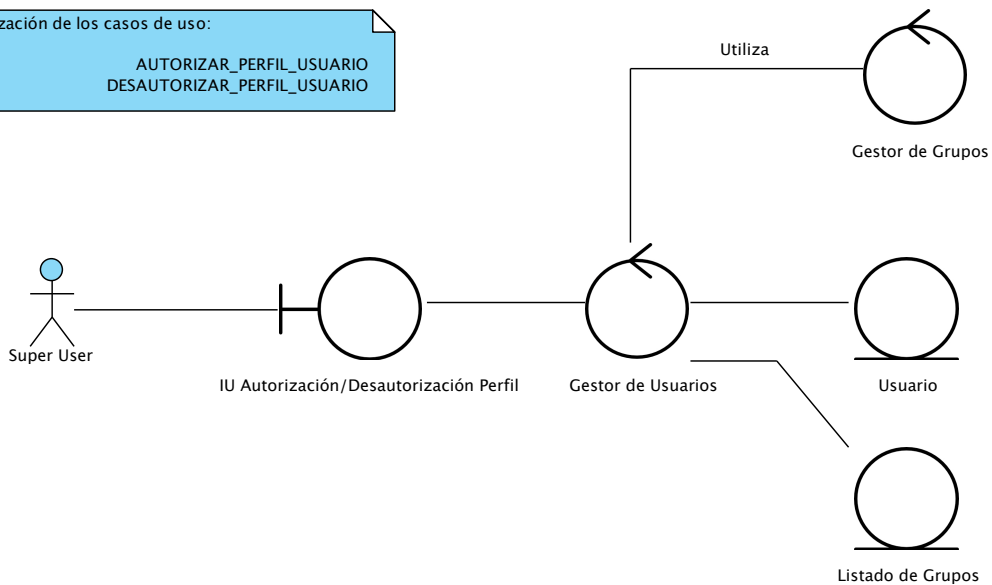


Figura 19. Análisis de los Casos de Uso

AUTORIZAR_PERFIL_USUARIO **DESAUTORIZAR_PERFIL_USUARIO**

NOTA: Los dos casos de uso presentados en este punto 3.3.4.5 comparten el mismo diagrama de clases de análisis. La razón es que, aunque desde el punto de vista de la lógica de la plataforma haya una gran diferencia, y sean 2 casos de uso diferentes, desde el punto de vista de la arquitectura, pueden representarse con las mismas clases de análisis. Esto es debido a que, internamente, la plataforma **WebDoc** autoriza a un perfil de usuario (o no) en base a la pertenencia (o no) al grupo “Publisher”.

3.3.4.7. Análisis del caso de uso

BORRAR_PERFIL_USUARIO

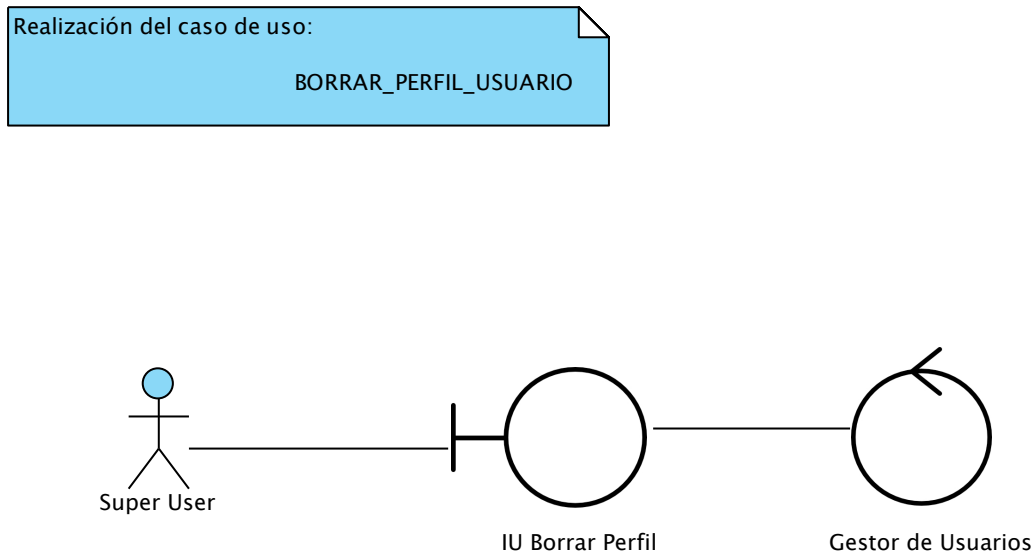


Figura 20. Análisis del Caso de Uso **BORRAR_PERFIL_USUARIO**

3.4. Diseño.

3.4.1. Introducción.

En el diseño modelamos el sistema y encontramos su forma (incluida la arquitectura) para que soporte todos los requisitos (incluyendo los requisitos no funcionales y otras restricciones) que se le suponen.

Una entrada esencial en el diseño es el resultado del análisis, esto es, el modelo de análisis. El modelo de análisis proporciona una comprensión detallada de los requisitos. Y lo que es más importante, impone una estructura del sistema que debemos esforzarnos por conservar lo más fielmente posible cuando demos forma al sistema [22].

3.4.2. Modo Creación De Documentos.

3.4.2.1. Diseño del caso de uso

CREAR_CARPETA

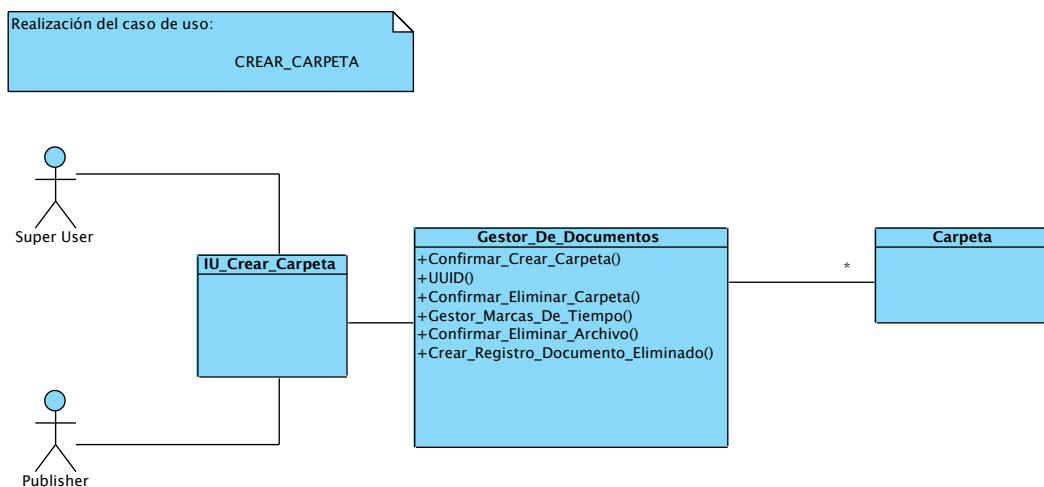


Figura 21. Diseño del Caso de Uso **CREAR_CARPETA**

3.4.2.2. Diseño del caso de uso

CREAR_DOCUMENTO

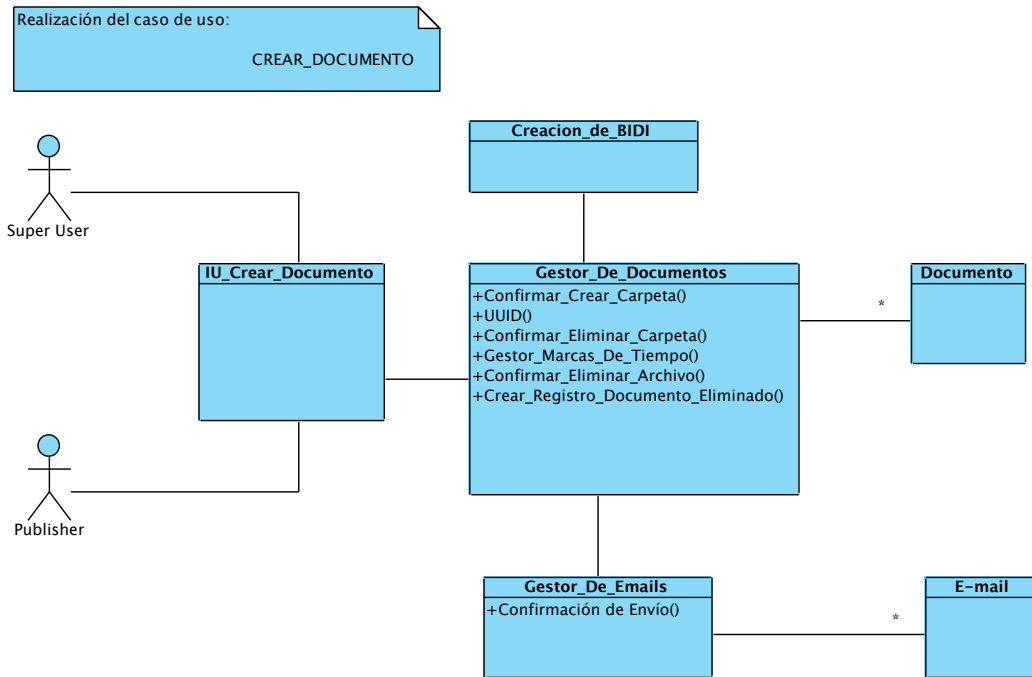


Figura 22. Diseño del Caso de Uso **CREAR_DOCUMENTO**

3.4.2.3. Diseño del caso de uso

ELIMINAR_CARPETA

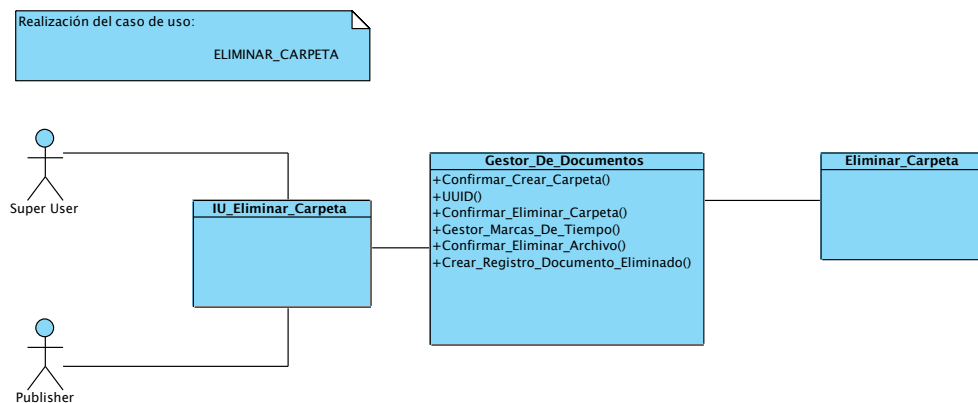


Figura 23. Diseño del Caso de Uso **ELIMINAR_CARPETA**

3.4.2.4. Diseño del caso de uso

ELIMINAR_DOCUMENTO

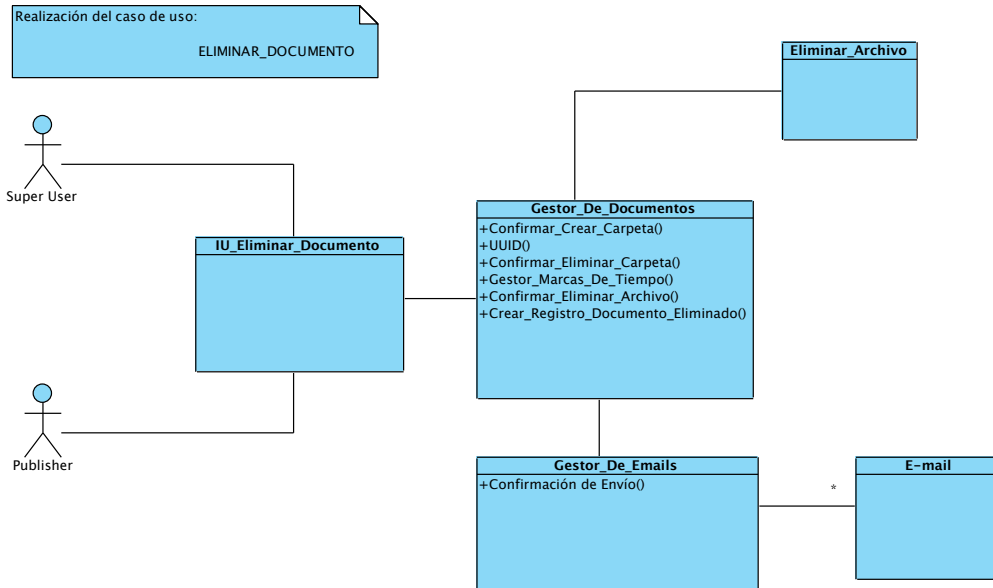


Figura 24. Diseño del Caso de Uso **ELIMINAR_DOCUMENTO**

3.4.3. Modo Visualización De Documentos.

3.4.3.1. - 3.4.3.3. Diseño de los caso de uso

LISTAR_TODOS_LOS_DOCUMENTOS

LISTAR_TODOS_LOS_DOCUMENTOS_DE_UN_TIPO

LISTAR_UNA_CATEGORIA_DADO_UN_TIPO_DE_DOCUMENTO

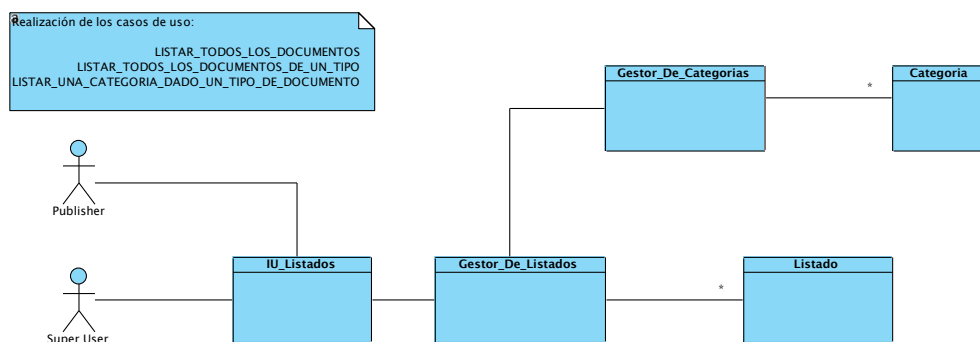


Figura 25. Diseño de los Casos de Uso

LISTAR_TODOS_LOS_DOCUMENTOS

LISTAR_TODOS_LOS_DOCUMENTOS_DE_UN_TIPO

LISTAR_UNA_CATEGORIA_DADO_UN_TIPO_DE_DOCUMENTO



3.4.3.4. Diseño del caso de uso

BUSCAR_DOCUMENTO

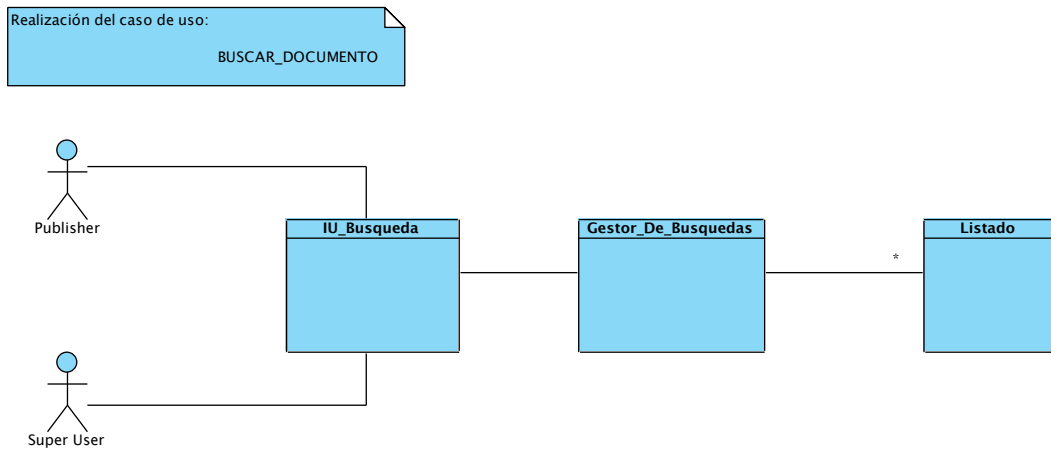


Figura 26. Diseño del Caso de Uso **BUSCAR_DOCUMENTO**

3.4.3.5. Diseño del caso de uso

CREAR_PERFIL_USUARIO

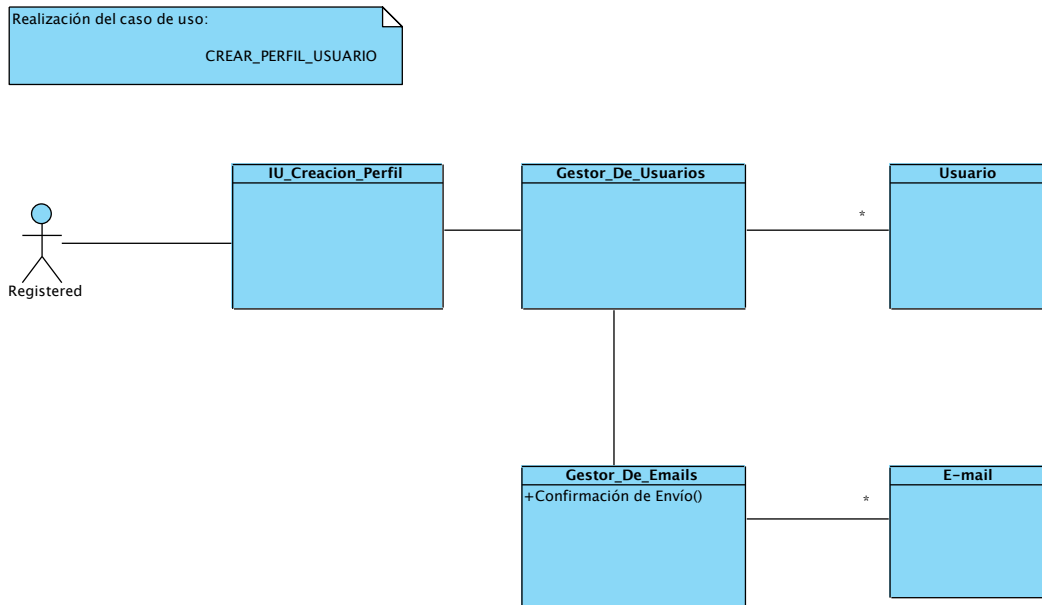


Figura 27. Diseño del Caso de Uso **CREAR_PERFIL_USUARIO**

3.4.3.6. Diseño del caso de uso

ACTUALIZAR_PERFIL_USUARIO

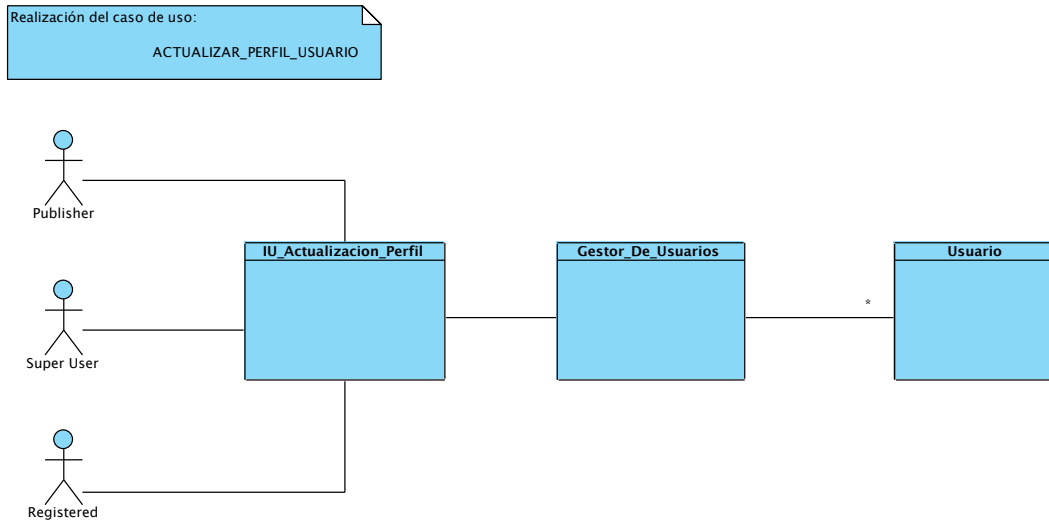


Figura 28. Diseño del Caso de Uso **ACTUALIZAR_PERFIL_USUARIO**

3.4.4. Modo Administración.

3.4.4.1. Introducción

En el modo “Administración” de la plataforma, se va a hacer un uso intensivo de las opciones de administración que nos proporciona la plataforma **Joomla!** Por ello, en los diagramas de clase de los casos de uso, las clases están etiquetadas con la forma **J_XXX** (donde **XXX** es el nombre identificativo de la clase). Ello significa que estas clases forman parte del núcleo de **Joomla!** [5].

3.4.4.2. Diseño del caso de uso

LISTAR_REGISTROS_DOCUMENTOS_BORRADOS

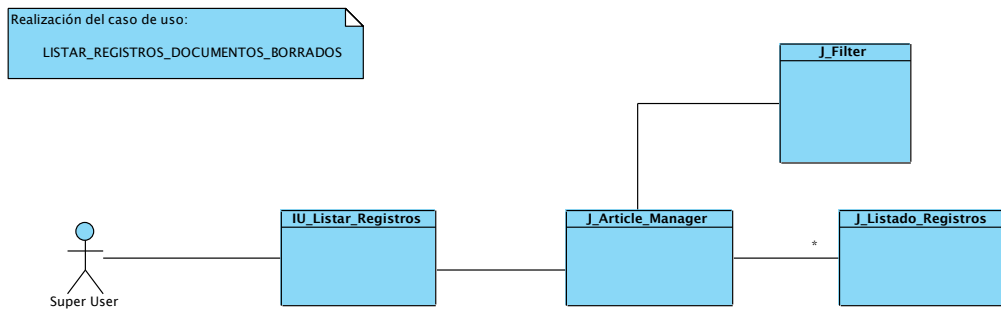


Figura 29. Diseño del Caso de Uso **LISTAR_REGISTROS_DOCUMENTOS_BORRADOS**

3.4.4.3. Diseño del caso de uso

ELIMINAR_REGISTRO_DOCUMENTO_BORRADO

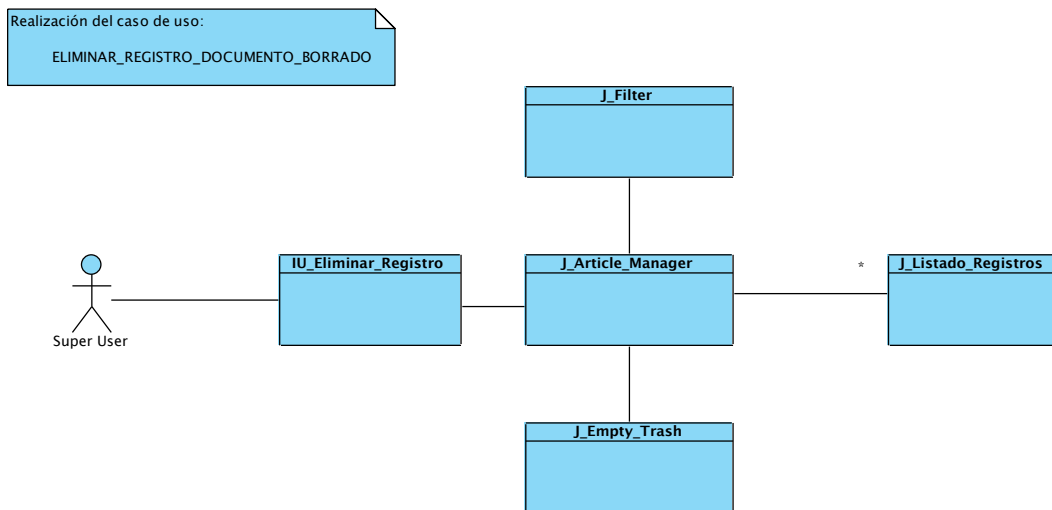


Figura 30. Diseño del Caso de Uso **ELIMINAR_REGISTRO_DOCUMENTO_BORRADO**

3.4.4.4. Diseño del caso de uso

CREAR_PERFIL_USUARIO

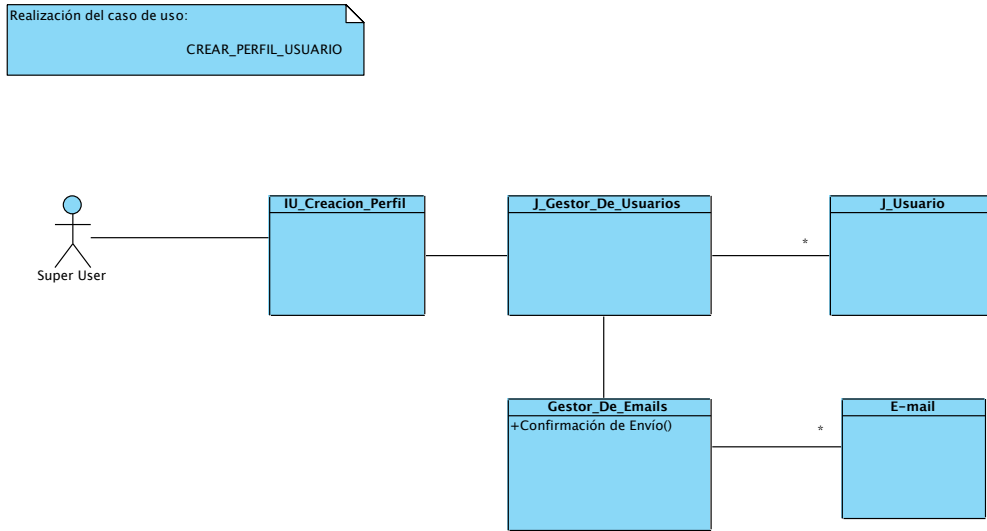


Figura 31. Diseño del Caso de Uso **CREAR_PERFIL_USUARIO**

3.4.4.5. Diseño del caso de uso

ACTUALIZAR_PERFIL_USUARIO

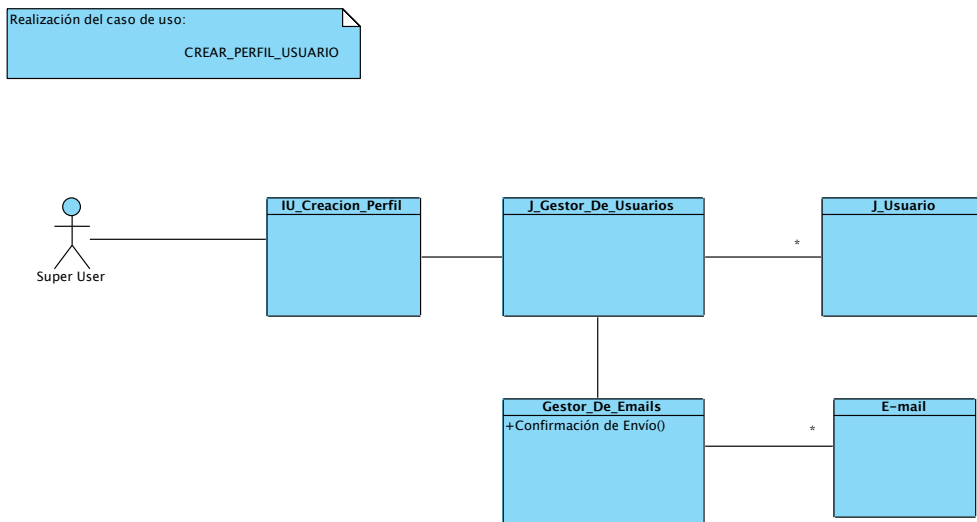


Figura 32. Diseño del Caso de Uso **ACTUALIZAR_PERFIL_USUARIO**

3.4.4.6. - 3.4.4.7. Diseño de los casos de uso

AUTORIZAR_PERFIL_USUARIO

DESAUTORIZAR_PERFIL_USUARIO

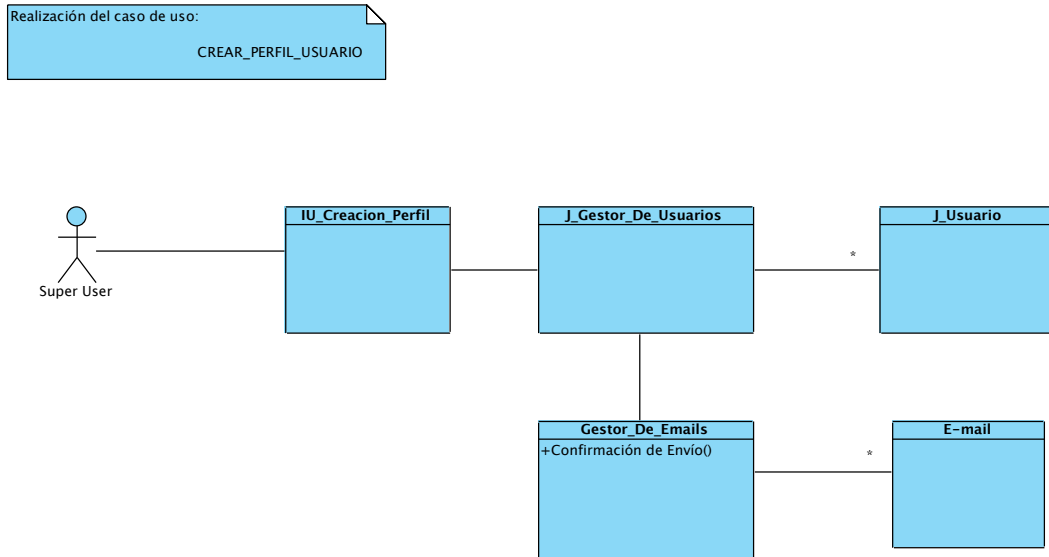


Figura 33. Diseño de los Casos de Uso

AUTORIZAR_PERFIL_USUARIO

DESAUTORIZAR_PERFIL_USUARIO

3.4.4.8. Diseño del caso de uso

BORRAR_PERFIL_USUARIO

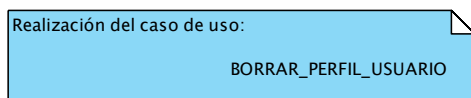


Figura 34. Diseño del Caso de Uso **BORRAR_PERFIL_USUARIO**

3.5. Implementación.

3.5.1. Introducción.

En la implementación empezamos con el resultado del diseño e implementamos el sistema en términos de componentes, es decir, archivos de código fuente, *scripts*, archivos de código binario, ejecutables y similares.

Afortunadamente, la mayor parte de la arquitectura del sistema es capturada durante el diseño, siendo el propósito principal de la implementación el desarrollar la arquitectura y el sistema como un todo [23].

3.5.2. Implementación en **Joomla!**

Para realizar la implementación del sistema **WebDoc** se ha escogido desarrollarlo sobre el gestor de contenido **Joomla!** [14]. Debido a que es una plataforma web ampliamente difundida, con una gestión de usuarios con una granularidad muy fina en cuanto a permisos de acceso y porque existen multitud de componentes ya desarrollados que permiten su integración en nuevos sistemas.

El desarrollo sobre la plataforma **Joomla!** está basado (tal y como se especifica en el punto “3.6.1 Bases de datos” de la **ERS**), a partes iguales, entre el código escrito en los archivos PHP que definen la plataforma (y que se encuentran en el sistema de archivos del servidor) y el código escrito en módulos definidos en la base de datos. Es decir, para hacer funcionar una unidad lógica (sea unidad lógica definida como un módulo, plantilla, componente, etc.) hay que tener 2 partes desarrolladas:

- 1.- El código en el archivo PHP correspondiente.
- 2.- El código que se encuentra en la base de datos.

Por ello, se procede a presentar un mapa mental, que define todos los módulos, categorías, entradas de menú, etc. que se han tenido que crear en **Joomla!** para la implementación de la plataforma **WebDoc**.

Diseño e implementación de una plataforma web para gestión documental (**WebDoc**)

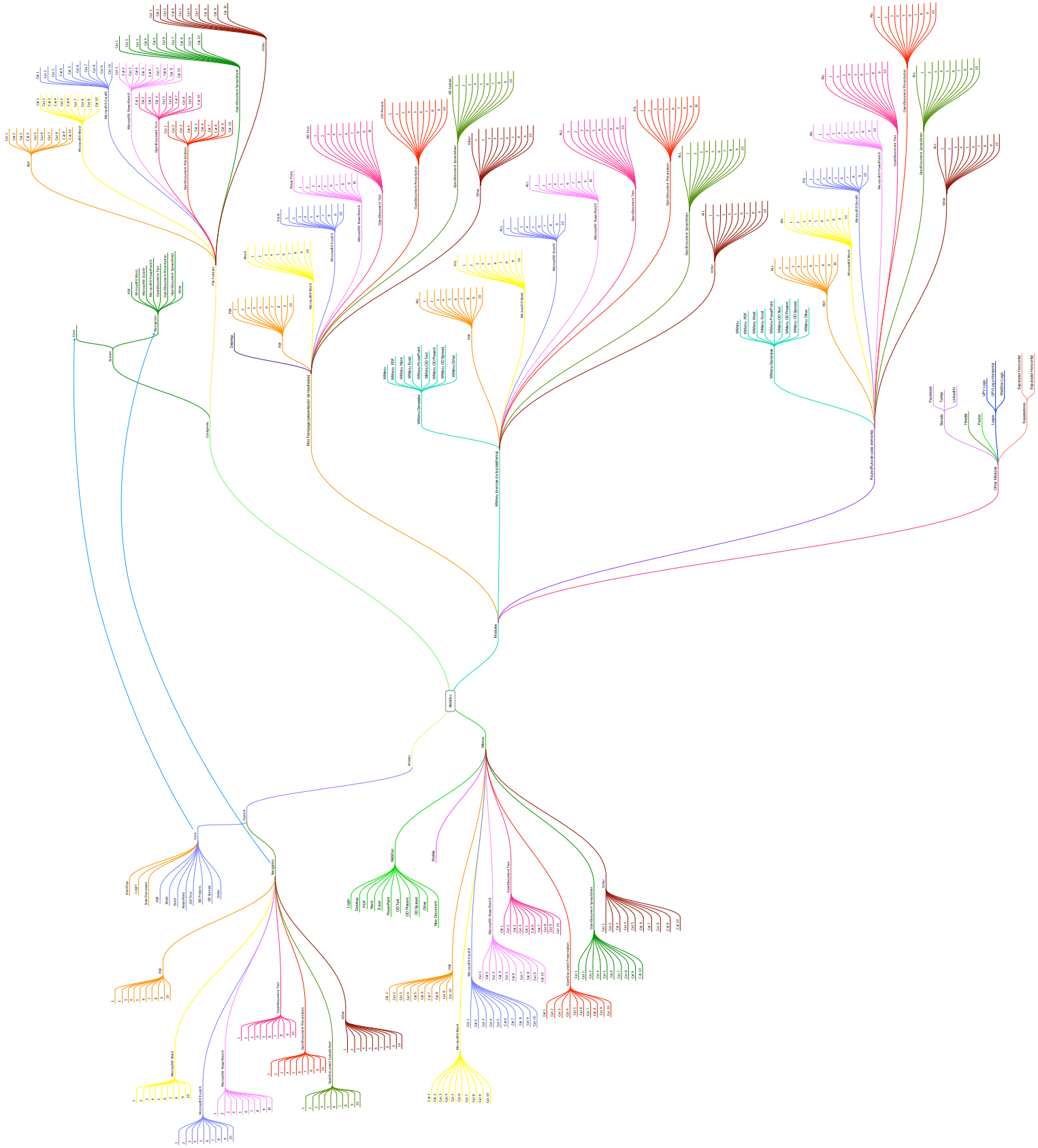


Figura 35. Mapa Mental de la implementación en Joomla! de WebDoc



NOTA: En la implementación sólo se va a poner el código relevante para una mejor comprensión de la plataforma **WebDoc**. El resto del código puede consultarse directamente en los archivos y base de datos de la plataforma. En el caso de uso donde no se incluya código en la memoria, se pondrá **V.E.P.** (Ver En Plataforma).

3.5.3. Modo Creación De Documentos.

3.5.3.1. Implementación del caso de uso

CREAR_CARPETA

(V.E.P.)

3.5.3.2. Implementación del caso de uso

CREAR_DOCUMENTO



/modules/plugins/content/jsmallfib/jsmallfib.php

```

[ ----- ]
[ -----[ Línea 186 ]----- ]
[ ----- ]

// -----
// Función, muy elegante, generadora de UUID sacada de:
// http://stackoverflow.com/questions/2040240/php-function-to-generate-v4-uuid
// -----

function guidv4()
{
    $data = openssl_random_pseudo_bytes(8);

    return vsprintf('%s-%s-%s', str_split(bin2hex($data), 4));
}

// -----

/**
 * The database object.
 */
var $db;

// -----

[ ----- ]
[ -----[ Línea 2468 ]----- ]
[ ----- ]

// -----
// En este IF se comprueba que el fichero existe, por tanto, habría que poner la variable de comprobación
// en este lugar. Un booleano que nos dijera que el fichero ya existe y que no hace falta crear un nuevo
// artículo en Joomla.
// -----

$ExisteFichero = TRUE;

// -----

[ ----- ]
[ -----[ Línea 2523 ]----- ]
[ ----- ]

// ----- SUBIDA DE FICHEROS AL SERVIDOR ----- //
//
// Las dos líneas que siguen al código de Webtory, se usan para subir ficheros al servidor,
// luego, es aquí donde hay que poner el código para dar de alta un nuevo artículo para WebDoc.
//
// ¡Atención! Hay que tener en cuenta que, si el fichero ya existía no hay que crear un nuevo
// artículo, porque podría haber, con el tiempo, varios artículos refenciando un único fichero.
//
// ----- //

// DEBUG USERS >>>> $user = JFactory::getUser();

if(!$ExisteFichero)
{
    $user=& JFactory::getUser();

    $stable = JTable::getInstance('Content', 'JTable');

    // WEBTORY: El módulo nos da una ruta como ésta:
    // "homepages/33/d27260080/htdocs/webdoc/images/Upload/alrodbai/doc.png"
    // que se usa para las operaciones de subir y mover de carpeta el documento,
    // por lo que nosotros "recortamos" a partir de la palabra "Upload", que es la
    // carpeta donde están todos los documentos, y, el resto, es la dirección del fichero
    // para poder descargarlo posteriormente.
    // Así, la ruta anterior quedaría:
    // $toks[0] = "homepages/33/d27260080/htdocs/webdoc/images/";
    // $toks[1] = "/alrodbai/doc.png";

    $toks = explode("Upload", $upload_file);

    // Volvemos a dividir 2 veces, hasta conseguir un array con el fichero y la extensión.
    $FicheroYrutaPorSeparado = explode("/", $toks[1]);
    $TamanyoFicheroYrutaPorSeparado = sizeof($FicheroYrutaPorSeparado);

    // En esta variable, la posición 0 indica el nombre del fichero, y la 1 indica la extensión.
    $NombreDeFicheroYExtensionPorSeparado = explode(".", $FicheroYrutaPorSeparado[$TamanyoFicheroYrutaPorSeparado-1]);

    // Las extensiones de fichero para Adobe® PDF están sacadas de esta Web:
    // http://en.wikipedia.org/wiki/Portable_Document_Format#Forms_Data_Format_.28PDF.29
    // Las extensiones de fichero para Microsoft® Office® están sacadas de esta Web:
    // http://office.microsoft.com/en-us/help/introduction-to-new-file-name-extensions-HA010006935.aspx
    // Las extensiones de fichero para OpenDocument están sacadas de esta Web:
    // http://es.wikipedia.org/wiki/OpenDocument

    $sicono = "";

    $TipoDeFichero = strtolower($NombreDeFicheroYExtensionPorSeparado[1]);

```

/modules/plugins/content/jsmallfib/jsmallfib.php

```

switch($TipoDeFichero)
{
    case "pdf":
    case "fdf":
    case "xfdf":
        $supercat = 8;
        $cat = [94,95,96,97,98,99,100,101,126,127];
        $icono = "pdf";
        break;

    case "docx":
    case "docm":
    case "dotx":
    case "dotm":
    case "doc":
        $supercat = 9;
        $cat = [102,103,104,105,106,107,108,109,128,129];
        $icono = "doc";
        break;

    case "xlsx":
    case "xslm":
    case "xltx":
    case "xltm":
    case "xlsb":
    case "xlsm":
    case "xls":
        $supercat = 11;
        $cat = [110,111,112,113,114,115,116,117,130,131];
        $icono = "xls";
        break;

    case "pptx":
    case "pptm":
    case "potx":
    case "potm":
    case "ppam":
    case "ppsx":
    case "ppsm":
    case "sldx":
    case "sldm":
    case "thmx":
    case "ppt":
        $supercat = 10;
        $cat = [118,119,120,121,122,123,124,125,132,133];
        $icono = "ppt";
        break;

    case "odt":
        $supercat = 16;
        $cat = [54,55,56,57,58,59,60,61,134,135];
        $icono = "odt";
        break;

    case "odp":
        $supercat = 17;
        $cat = [62,63,64,65,66,67,68,69,136,137];
        $icono = "odp";
        break;

    case "ods":
        $supercat = 18;
        $cat = [70,71,72,73,74,75,76,77,138,139];
        $icono = "ods";
        break;

    default:
        $supercat = 20;
        $cat = [78,79,80,81,82,83,84,85,140,141];
        $icono = "other";
}

// Mediante la convención de el guión bajo: minombre_X
// determinamos a qué subcategoría pertenece el fichero, dentro de
// cada categoría según el tipo de fichero. La X corresponde
// a dicha subcategoría. Si no existe la X o tiene un número mayor de 10,
// se le asigna la supercategoría, que incluye todos los fichero, dentro
// del tipo de fichero que no tienen subcategoría.

// Debug > $subcat = 1;

$NombreSinGuionBajo = explode("_",$NombreDeFicheroYExtensionPorSeparado[0]);

if(sizeof($NombreSinGuionBajo)==2)
switch ($NombreSinGuionBajo[1])
{
    case "1":
        $subcat = 1;
        break;
    case "2":
        $subcat = 2;
}

```



/modules/plugins/content/jsmallfib/jsmallfib.php

```

        break;
        case "3":
            $subcat = 3;
            break;
        case "4":
            $subcat = 4;
            break;
        case "5":
            $subcat = 5;
            break;
        case "6":
            $subcat = 6;
            break;
        case "7":
            $subcat = 7;
            break;
        case "8":
            $subcat = 8;
            break;
        case "9":
            $subcat = 9;
            break;
        case "10":
            $subcat = 10;
            break;
        default:
            $subcat = 0;
            break;
    }
    else
    {
        // Si entramos aquí, es porque no había un subguión bajo en el nombre del fichero,
        // por lo que le asignamos la supercategoría.
        $subcat = 0;
    }
}

// -----

switch ($subcat)
{
    case 1:
        $catid = $cat[0];
        break;
    case 2:
        $catid = $cat[1];
        break;
    case 3:
        $catid = $cat[2];
        break;
    case 4:
        $catid = $cat[3];
        break;
    case 5:
        $catid = $cat[4];
        break;
    case 6:
        $catid = $cat[5];
        break;
    case 7:
        $catid = $cat[6];
        break;
    case 8:
        $catid = $cat[7];
        break;
    case 9:
        $catid = $cat[8];
        break;
    case 10:
        $catid = $cat[9];
        break;
    default:
        $catid = $supercat;
        break;
}

$UUID = $TipoDeFichero.'-'. $this->guidv4();

$fullText = '<div style="text-align: right;">
<table cellpadding="0" cellspacing="0" style="width:748px; border: 0px solid black; color:
black;">
<tr>
<td style="width:25%; padding: 5px; border-left:1px solid black; border-right:1px
solid black; border-top:1px solid black; border-bottom:1px solid black; font-size:12px; background-color:
gainsboro;">Name</td>
<td style="width:75%; text-align: left; padding: 5px; border-right:1px solid
black; border-top:1px solid black; border-bottom:1px solid black; background-color: gainsboro;">'.
$NombreSinGuionBajo[0].'</td>
</tr>
<tr>
<td style="width:25%; padding: 5px; border-left:1px solid black; border-right:1px

```


/modules/plugins/content/jsmallfib/jsmallfib.php

```

solid black; border-bottom:1px solid black; font-size:12px;">ID</td>
<td style="width:75%; text-align: left; padding: 5px; border-right:1px solid
black; border-bottom:1px solid black;">'. $UUID.'</td>
</tr>
<tr>
<td style="width:25%; padding: 5px; border-left:1px solid black; border-right:1px
solid black; border-bottom:1px solid black; font-size:12px; background-color: gainsboro;">User</td>
<td style="width:75%; text-align: left; padding: 5px; border-right: 1px solid
black; border-bottom:1px solid black; background-color: gainsboro;">'. $user->email.'</td>
</tr>
<tr>
<td style="width:25%; padding: 5px; border-left:1px solid black; border-right:1px
solid black; border-bottom:1px solid black; font-size:12px;">Department (Centre)</td>
<td style="width:75%; text-align: left; padding: 5px; border-right:1px solid
black; border-bottom:1px solid black;">'. $_POST['WebDocDepartment'].'</td>
</tr>
<tr>
<td style="width:25%; padding: 5px; border-left:1px solid black; border-right:1px
solid black; border-bottom:1px solid black; font-size:12px;">Description</td>
<td style="width:75%; text-align: left; padding: 5px; border-right:1px solid
black; border-bottom:1px solid black;">'. $_POST['WebDocDescription'].'</td>
</tr>
</table>
</div>';

// -----
// Cálculo de URL única:
// Las URLs únicas, son del tipo >
// http://webdoc.webtory.es/index.php/component/search/?
searchword=ppt_cd06-0270-2c85&ordering=newest&searchphrase=exact&limit=20&areas[0]=content&Itemid=101
// -----

$URLUnique = JURI::base(). 'index.php/component/search/?searchword=' .
$UUID.'&ordering=newest&searchphrase=exact&limit=20&areas[0]=content&Itemid=101';

// -----

$URL_OR = JURI::base(). 'images/Upload'. $stoks[1];

// DEBUG > $error .= $URL_OR;

$data = array(
    'catid' => $catid,
    'title' => $NombreSinGuionBajo[0],
    'alias' => $UUID,
    'introtxt' => '<div style="width:748px; height: 80; vertical-align: middle; text-align: right;"> <a
href="images/Upload'. $stoks[1]."></a></
div>',
    'fulltext' => $fullText,
    'state' => 1,
);

$HayError = FALSE;
$MensajeError = '';

// Bind data
if (!$stable->bind($data))
{
    $MensajeError = 'The data format sent to WebDoc is not correct!<br>';
    $MensajeError .= $stable->getError();
    $HayError = TRUE;
}

// Check the data.
if (!$stable->check())
{
    $MensajeError = 'Check the data sent to WebDoc!<br>';
    $MensajeError .= $stable->getError();
    $HayError = TRUE;
}

// Store the data.
if (!$stable->store())
{
    $MensajeError = $stable->getError();
    $HayError = TRUE;
}

if (!$HayError)
{
    // Reordenamos la categoría de Joomla, que no se hace automáticamente.
    $stable->reorder('catid = '.(int) $stable->catid.' AND state >= 0');

    // -----
    // Correo de confirmación de la creación de un nuevo fichero en WebDoc.
    // -----

    // Fetch the mail object -----

```



/modules/plugins/content/jsmallfib/jsmallfib.php

```

$mail =& JFactory::getMailer();

// Set a sender -----
$config =& JFactory::getConfig();
$sender = array(
    $config->get( 'mailfrom' ),
    $config->get( 'fromname' ) );
$mail->setSender($sender);

// Recipient & Subject -----
$mail->setSubject("WebDoc - UPV - WebDoc New Document");

// Lo dejamos comentado para posibles notificaciones
// futuras al propio usuario que realiza la acción.
// $user =& JFactory::getUser();
// $recipient = $user->email;

// Consulta con la BD para saber los emails de los Super Usuarios,
// teniendo en cuenta que desean recibir emails del sistema, si no
// es así, no se envían correos.

// Get the database object.
$this->db = JFactory::getDBO();

$queryAdminEmail = "SELECT u.email, u.name
FROM #__users u
WHERE
    u.id IN
        (SELECT DISTINCT(j.user_id)
        FROM #__user_usergroup_map j
        WHERE
            j.group_id = 8)
    AND
        u.sendEmail = 1;";

$this->db->setQuery($queryAdminEmail);
$result_queryAdminEmail = $this->db->loadRowList();

// -----
foreach($result_queryAdminEmail as $oneEmailAsArray)
    $mail->addRecipient($oneEmailAsArray[0],$oneEmailAsArray[1]);

// Create the mail -----
$title = '<h2>WebDoc New Document</h2>';
$body = '<div style="text-align:center;">A new document has been created on WebDoc!</div>'
        '<div style="text-align:center;"><br/><a href="', $URLUnique, '"></a></div>';
$mail->isHTML(true);
$mail->Encoding = 'base64';
$mail->setBody($title.$body);

// Create the mail -----
$send =& $mail->Send();
if ( $send != true ) {
    $error .= JText::sprintf('Error sending email. The notification system is disabled for all
users. ');
} else {
    $error .= JText::sprintf($body);
}

// -----

```

3.5.3.3. Implementación del caso de uso

ELIMINAR_CARPETA

(V.E.P.)

3.5.3.4. Implementación del caso de uso

ELIMINAR_DOCUMENTO



/modules/plugins/content/jsmallfib/jsmallfib.php

```
[ ----- ]
[ -----{ Línea 1193 }----- ]
[ ----- ]

// -----
// DELETED: Procedemos a borrar el artículo de Joomla que equivale al fichero que estamos borrando.
// -----
// -----

$deleted_document = explode(".", $_GET["delfile"]);

// DEBUG -----
// $texto_error = '<div><br/>&nbsp;Document to Delete:&nbsp;'.$_GET["delfile"].'</div>';
// $error .= JText::sprintf($texto_error);
// -----

if(sizeof($deleted_document)==2)
{
    $TipoDeFichero = explode(".", $deleted_document[1]);
    $Document_Title = $deleted_document[0];
}
else
{
    $TipoDeFichero = explode(".", $deleted_document[0]);
    $Document_Title = $TipoDeFichero[0];
}

switch($TipoDeFichero[1])
{
    case "pdf":
    case "fdf":
    case "xpdf":
        $supercat = 8;
        $cat = [94,95,96,97,98,99,100,101,126,127];
        break;

    case "docx":
    case "docm":
    case "dotx":
    case "dotm":
    case "doc":
        $supercat = 9;
        $cat = [102,103,104,105,106,107,108,109,128,129];
        break;

    case "xlsx":
    case "xlsm":
    case "xltx":
    case "xltxm":
    case "xlsb":
    case "xlam":
    case "xls":
        $supercat = 11;
        $cat = [110,111,112,113,114,115,116,117,130,131];
        break;

    case "pptx":
    case "pptm":
    case "potx":
    case "potm":
    case "ppam":
    case "ppsx":
    case "ppsm":
    case "sldx":
    case "sldm":
    case "thmx":
    case "ppt":
        $supercat = 10;
        $cat = [118,119,120,121,122,123,124,125,132,133];
        break;

    case "odt":
        $supercat = 16;
        $cat = [54,55,56,57,58,59,60,61,134,135];
        break;

    case "odp":
        $supercat = 17;
        $cat = [62,63,64,65,66,67,68,69,136,137];
        break;

    case "ods":
        $supercat = 18;
        $cat = [70,71,72,73,74,75,76,77,138,139];
        break;

    default:
        $supercat = 20;
        $cat = [78,79,80,81,82,83,84,85,140,141];
}
```



/modules/plugins/content/jsmallfib/jsmallfib.php

```

}

if(sizeof($deleted_document)==2)
switch ($TipoDeFichero[0])
{
    case "1":
        $subcat = 1;
        break;
    case "2":
        $subcat = 2;
        break;
    case "3":
        $subcat = 3;
        break;
    case "4":
        $subcat = 4;
        break;
    case "5":
        $subcat = 5;
        break;
    case "6":
        $subcat = 6;
        break;
    case "7":
        $subcat = 7;
        break;
    case "8":
        $subcat = 8;
        break;
    case "9":
        $subcat = 9;
        break;
    case "10":
        $subcat = 10;
        break;
    default:
        $subcat = 0;
        break;
}
else
{
    // Si entramos aquí, es porque no había un subguión bajo en el nombre del fichero,
    // por lo que le asignamos la supercategoría.
    $subcat = 0;
}

switch ($subcat)
{
    case 1:
        $catid = $cat[0];
        break;
    case 2:
        $catid = $cat[1];
        break;
    case 3:
        $catid = $cat[2];
        break;
    case 4:
        $catid = $cat[3];
        break;
    case 5:
        $catid = $cat[4];
        break;
    case 6:
        $catid = $cat[5];
        break;
    case 7:
        $catid = $cat[6];
        break;
    case 8:
        $catid = $cat[7];
        break;
    case 9:
        $catid = $cat[8];
        break;
    case 10:
        $catid = $cat[9];
        break;
    default:
        $catid = $supercat;
        break;
}

// WEBTORY _____
//
// Procedemos a actualizar la base de datos de Joomla, con la eliminación del artículo.
//
//
// Get the database object.
$this->db = JFactory::getDBO();

```



/modules/plugins/content/jsmallfib/jsmallfib.php

```

$query = $this->db->getQuery(true);
$query->select($this->db->quoteName('id'));
$query->from($this->db->quoteName('#_content'));
$query->where(
    $this->db->quoteName('title') . ' = ' . "'" . $Document_Title . "'"
    " AND "
    $this->db->quoteName('catid') . ' = ' . $catid
);
$this->db->setQuery($query);
$result_query = (String) $this->db->loadResult();

// DEBUG -----
// $texto_error = '<div>-----<br/>Query:&nbsp;&nbsp;&nbsp;'.(String) $query.'</div>'
// $error .= JText::sprintf($texto_error);
// -----

if($result_query != '')
{
    $query2 = $this->db->getQuery(true);
    $query2->update($this->db->quoteName('#_content'));
    $query2->set($this->db->quoteName('state'), ' = -2');
    $query2->where($this->db->quoteName('id') . ' = ' . $result_query);
    $this->db->setQuery($query2);
    $result_query2 = (String) $this->db->loadResult();

    // Actualizamos el nombre del fichero en la base de datos, para que se pueda subir,
    // posteriormente, con el mismo nombre. Esto se hace así, por motivos operativos, ya
    // que el artículo se tiene que borrar desde el administrador de Joomla, pues, al borrar
    // un artículo se actualizan muchas tuplas de muchas relaciones, y no se puede eliminar
    // la tupla simplemente.

    // Obtenemos la fecha con resolución de milisegundos, por evitar duplicidades en accesos
    // concurrentes de usuarios (la probabilidad de que 2 usuarios obtengan la misma fecha
    // en milisegundos es extremadamente baja).

    list($usec, $sec) = explode(" ", microtime());
    list($lambda, $millisecons) = explode(".", $usec);
    $outputMS = date('Y_m_d_H_i_s', $sec) . "." . $millisecons;

    // -----

    $query3 = $this->db->getQuery(true);
    $query3->update($this->db->quoteName('#_content'));
    $query3->set($this->db->quoteName('title'), ' = ' . "'" . $Document_Title . $outputMS . "'" );
    $query3->where($this->db->quoteName('id') . ' = ' . $result_query);
    $this->db->setQuery($query3);
    $result_query3 = (String) $this->db->loadResult();

    // -----
    // Correo de confirmación del borrado de un nuevo fichero en WebDoc.
    // -----

    // Fetch the mail object -----

    $mail =& JFactory::getMailer();
    $user =& JFactory::getUser();

    // Set a sender -----

    $config =& JFactory::getConfig();
    $sender = array(
        $config->get( 'mailfrom' ),
        $config->get( 'fromname' ) );

    $mail->setSender($sender);

    // Recipient & Subject -----

    $mail->setSubject("WebDoc - UPV - WebDoc Deleted Document");

    // Lo dejamos comentado para posibles notificaciones futuras al propio usuario que realiza la
    acción.

    // $user =& JFactory::getUser();
    // $recipient = $user->email;

    // Consulta con la BD para saber los emails de los Super Usuarios,
    // teniendo en cuenta que desean recibir emails del sistema, si no
    // es así, no se envían correos.

    // Get the database object.
    $this->db = JFactory::getDBO();

    $queryAdminEmail = "SELECT u.email, u.name
                        FROM #__users u
                        WHERE
                            u.id IN
                                (SELECT DISTINCT(j.user_id)
                                 FROM #__user_usergroup_map j
                                 WHERE
                                    j.group_id = 8)";

```

/modules/plugins/content/jsmallfib/jsmallfib.php

```

                                AND
                                u.sendEmail = 1;";

    $this->db->setQuery($queryAdminEmail);
    $result_queryAdminEmail = $this->db->loadRowList();

    // -----
    foreach($result_queryAdminEmail as $OneEmailAsArray)
        $mail->addRecipient($OneEmailAsArray[0], $OneEmailAsArray[1]);

    // Create the mail -----
    $title = '<h2>WebDoc Deleted Document</h2>';
    $body = '<div style="text-align:center;">A document has been deleted on WebDoc!</div>'
        // . '<br/>'. $result_query
        // . '<br/>'. $result_query2
        // . '<br/>'. $query . '</div>'
        // . '<br/>'. $query2 . '</div>'
        // . '<br/>'. $query3 . '</div>'
        // . '<br/>'. $query4 . '</div>'
        // . '<br/>'. print_r($result_queryAdminEmail,true) . '</div>'
        . '<div style="text-align:center;"><br/>'. $Document_Title.'</div>'
        . '<div style="text-align:center;"><br/>By '. $user->email.'</div>';

    $mail->isHTML(true);
    $mail->Encoding = 'base64';
    $mail->setBody($title.$body);

    // Create the mail -----
    $send =& $mail->Send();
    if ( $send != true ) {
        $error .= JText::sprintf('Error sending email. The notification system is disabled for all
users. ');
    } else {
        $error .= JText::sprintf('The document has been deleted!');
    }

    // -----
}
else
{
    $texto_error = '<div style="text-align:center;">The document does not exists in database!</div>';
    // . '<div style="text-align:center;"><br/>'.(String) $query.'</div>'; // << DEBUG
    $error .= JText::sprintf($texto_error);
}

// -----
// -----

```



3.5.4. Modo Visualización De Documentos.

3.5.4.1. - 3.5.4.3. Implementación de los caso de uso

LISTAR_TODOS_LOS_DOCUMENTOS
LISTAR_TODOS_LOS_DOCUMENTOS_DE_UN_TIPO
LISTAR_UNA_CATEGORIA_DADO_UN_TIPO_DE_DOCUMENTO

(V.E.P.)

3.5.4.4. Implementación del caso de uso

BUSCAR_DOCUMENTO

(V.E.P.)

3.5.4.5. Implementación del caso de uso

CREAR_PERFIL_USUARIO

(V.E.P.)

3.5.4.6. Implementación del caso de uso

ACTUALIZAR_PERFIL_USUARIO

(V.E.P.)

3.5.5. Modo Administración.

3.5.5.1. Implementación del caso de uso

LISTAR_REGISTROS_DOCUMENTOS_BORRADOS

(V.E.P.)

3.5.5.2. Implementación del caso de uso

ELIMINAR_REGISTRO_DOCUMENTO_BORRADO

(V.E.P.)

3.5.5.3. Implementación del caso de uso

CREAR_PERFIL_USUARIO

(V.E.P.)

3.5.5.4. Implementación del caso de uso

ACTUALIZAR_PERFIL_USUARIO

(V.E.P.)

3.5.5.5 - 3.5.5.6. Implementación de los casos de uso

AUTORIZAR_PERFIL_USUARIO

DESAUTORIZAR_PERFIL_USUARIO

(V.E.P.)

3.5.5.7. Implementación del caso de uso

BORRAR_PERFIL_USUARIO

(V.E.P.)

3.6. Pruebas.

3.6.1. Introducción.

Las pruebas son el proceso de encontrar diferencias entre el comportamiento esperado, especificado por los modelos del sistema, y el comportamiento observado del sistema. Las pruebas unitarias encuentran diferencias entre el modelo de diseño de objetos y sus componentes correspondientes. Las pruebas estructurales encuentran diferencias entre el modelo del diseño del sistema y un subconjunto de subsistemas integrados. Las pruebas funcionales encuentran diferencias entre los requerimientos no funcionales y el desempeño real del sistema. Cuando se encuentran diferencias, los desarrolladores identifican el defecto que causa la falla observada y modifican el sistema para corregirlo. En otros casos, se identifica al modelo como causa de la diferencia y se actualiza éste para que refleje el estado del sistema [...].

Las pruebas son el proceso de análisis de un sistema, o componente de un sistema, para detectar las diferencias entre el comportamiento especificado (requerido) y el observado (existente). Por desgracia, es imposible probar por completo un sistema no trivial. En primer lugar, las pruebas no son determinantes. En segundo, es necesario realizar las pruebas bajo restricciones de tiempo y presupuesto. En consecuencia, los sistemas se entregan sin estar probados por completo, lo que conduce a defectos que son descubiertos por los usuarios finales [24].

3.6.2. Desarrollo de las Pruebas.

3.6.2.1. Introducción.

Los casos de prueba de integración se utilizan para verificar que los componentes interaccionan entre sí de la forma apropiada después de haber sido integrados en una construcción. La mayoría de los casos de prueba de integración pueden ser derivados de las realizaciones de casos de uso-diseño, ya que las realizaciones de casos de uso describen cómo interaccionan las clases y los objetos, y, por tanto, cómo interaccionan los componentes [25].

3.6.2.2. Caso de prueba de integración para el caso de uso

CREAR_CARPETA

mediante el diagrama de secuencia

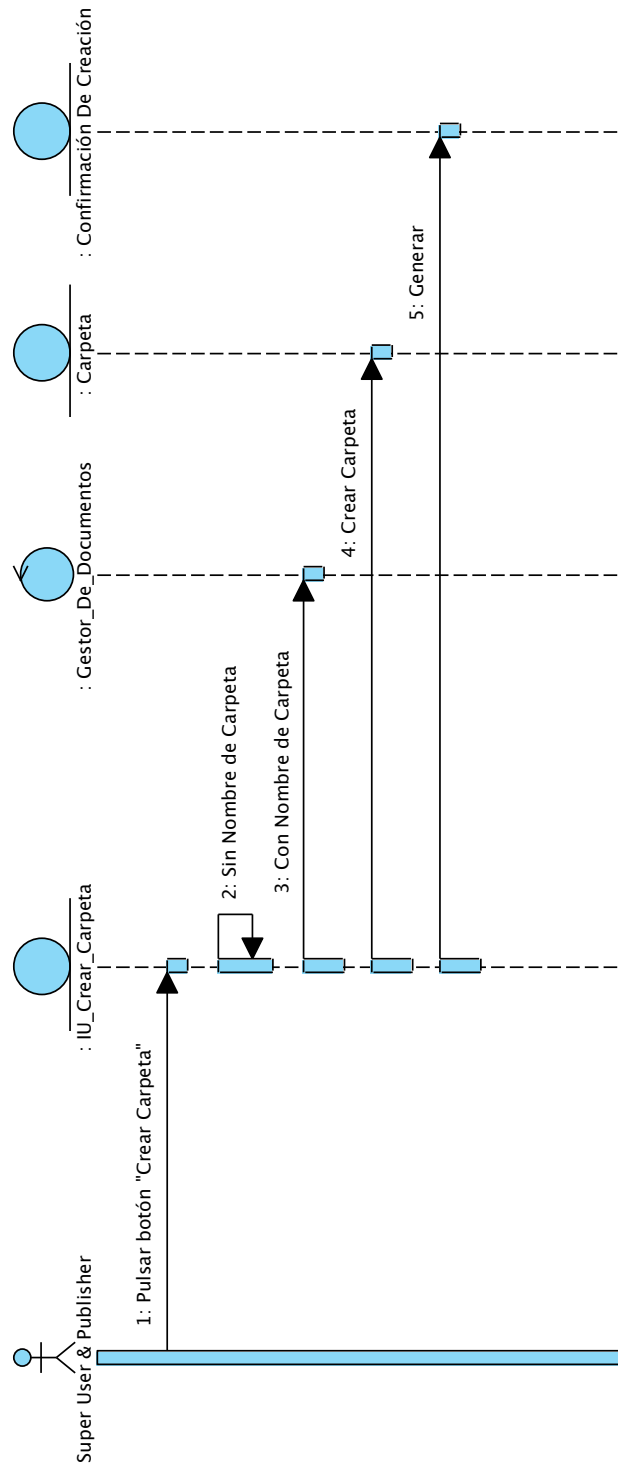


Figura 36. Caso de prueba de integración para el caso de uso **CREAR_CARPETA**

3.6.2.3. Caso de prueba de integración para el caso de uso

CREAR_DOCUMENTO

mediante el diagrama de secuencia

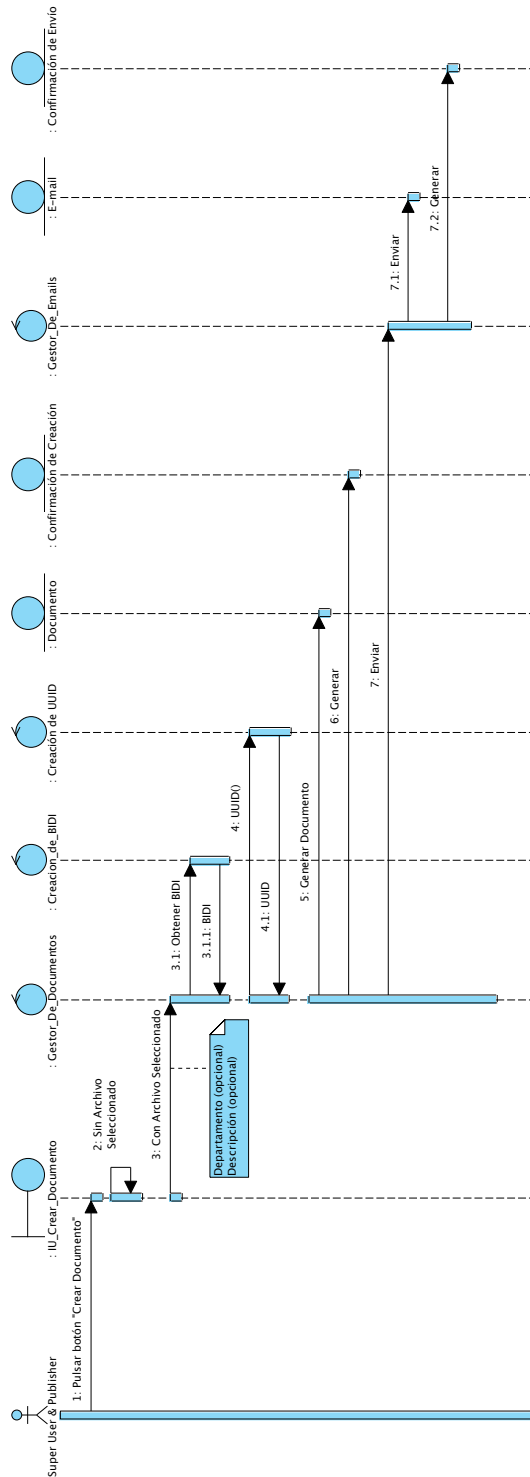


Figura 37. Caso de prueba de integración para el caso de uso **CREAR_DOCUMENTO**

3.6.2.4. Caso de prueba de integración para el caso de uso

ELIMINAR_CARPETA

mediante el diagrama de secuencia

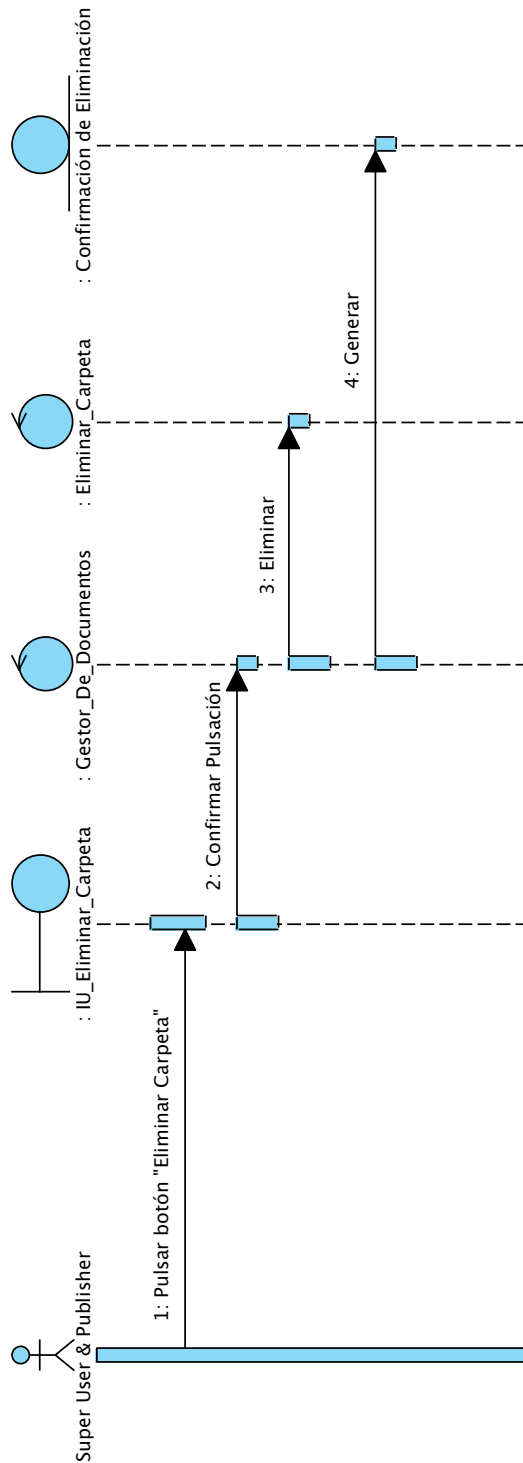


Figura 38. Caso de prueba de integración para el caso de uso **ELIMINAR_CARPETA**



3.6.2.5. Caso de prueba de integración para el caso de uso

ELIMINAR_DOCUMENTO mediante el diagrama de secuencia

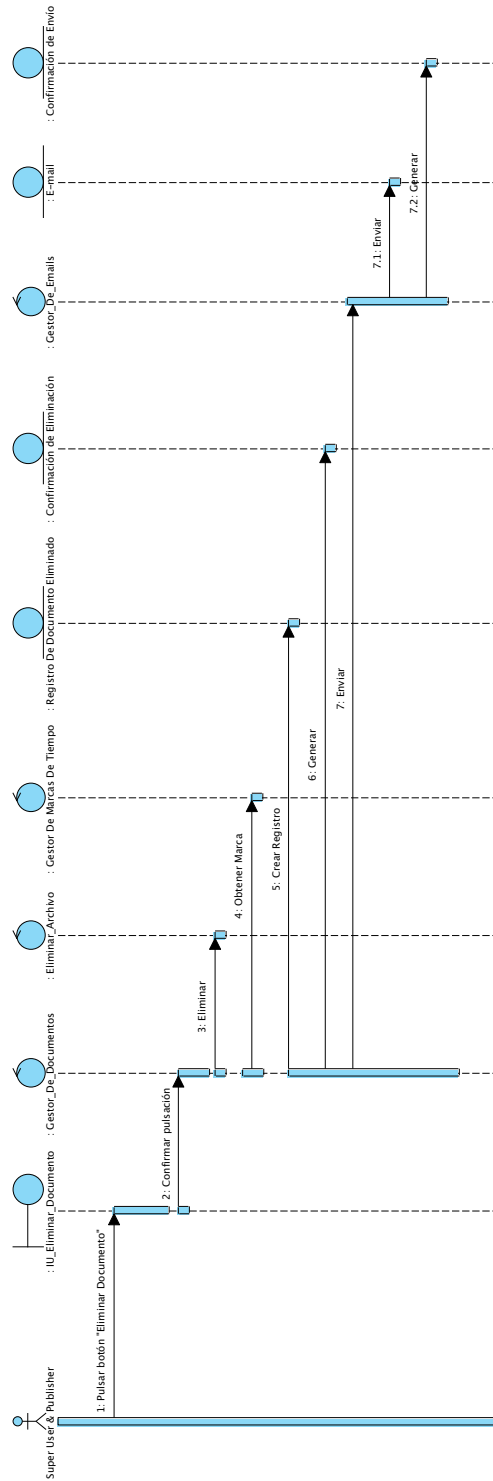


Figura 39. Caso de prueba de integración para el caso de uso **ELIMINAR_DOCUMENTO**

3.7. **WebDoc.**

Se va a presentar la plataforma **WebDoc**, en una instalación de prueba, en la que se ha desarrollado toda la implementación que se presenta en esta memoria.

3.7.1. Acceso (login).



Figura 40. Pantalla de Inicio de la plataforma **WebDoc**.



Figura 41. Pantalla de Acceso de la plataforma **WebDoc**.

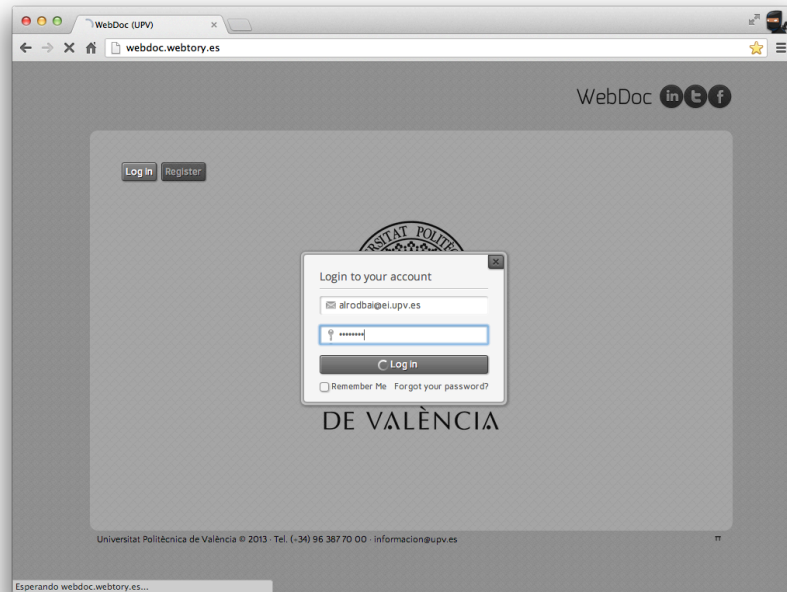


Figura 42. Accediendo a la plataforma **WebDoc**.

3.7.2. Desktop.

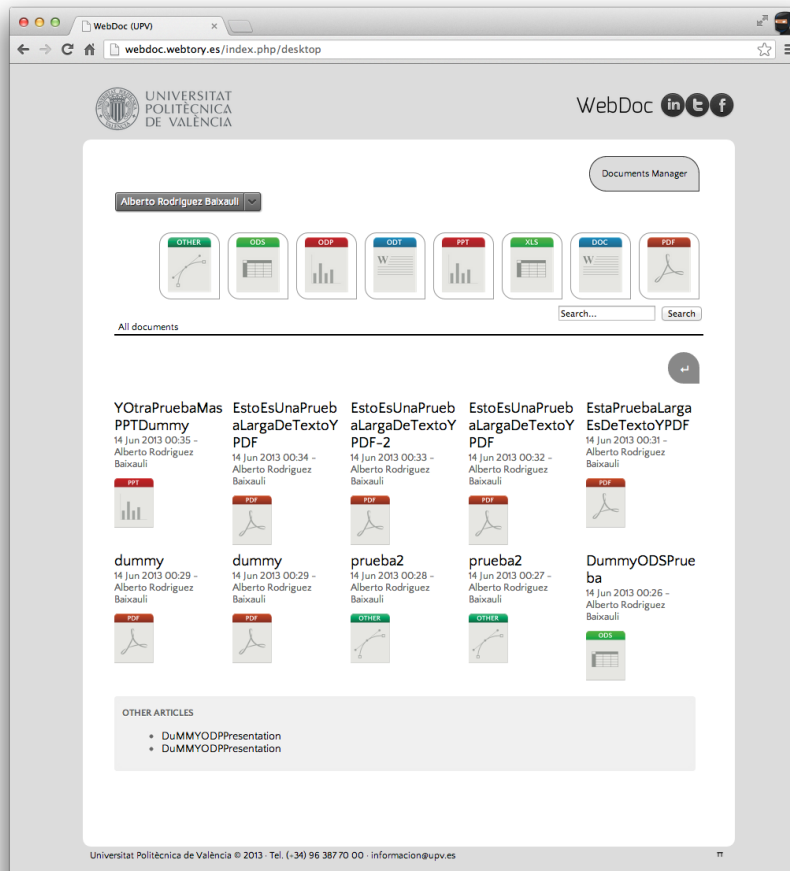


Figura 43. Escritorio (pantalla principal) de la plataforma **WebDoc**.

3.7.3. Documents Manager.

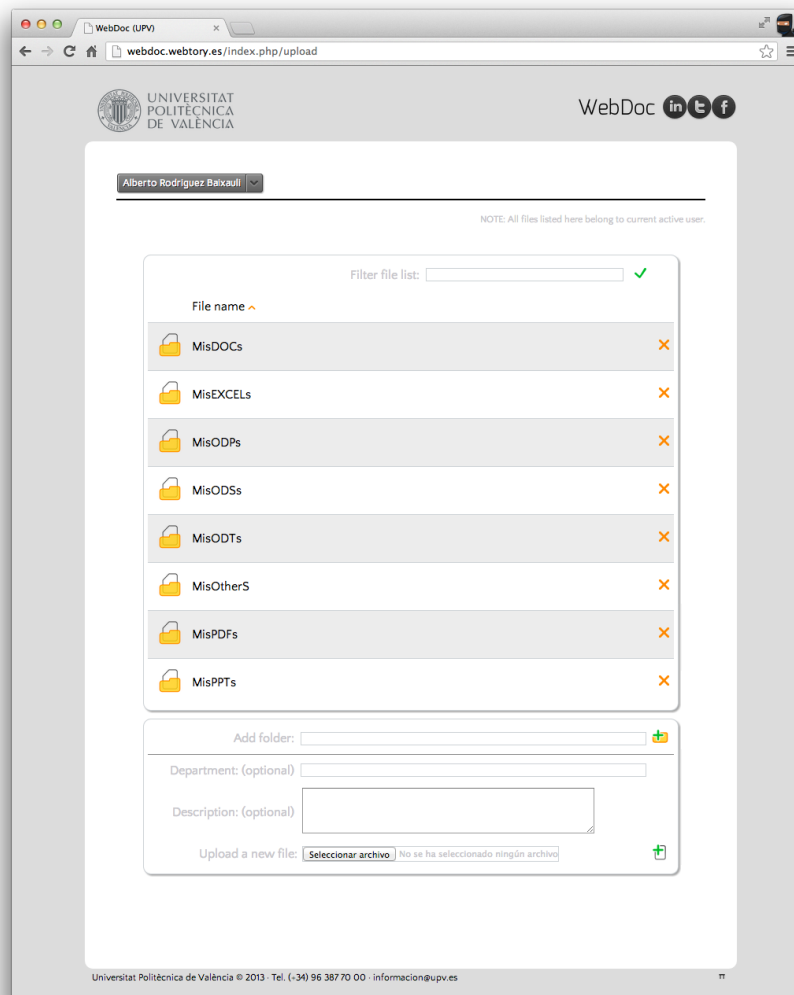


Figura 44. Pantalla principal del Gestor de Documentos de la plataforma **WebDoc**.

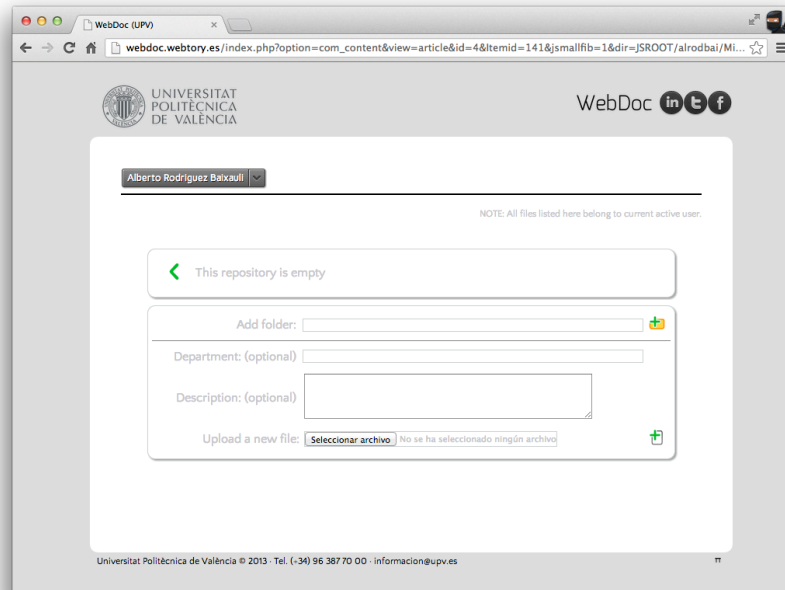


Figura 45. Un directorio vacío en el Gestor de Documentos de la plataforma WebDoc.

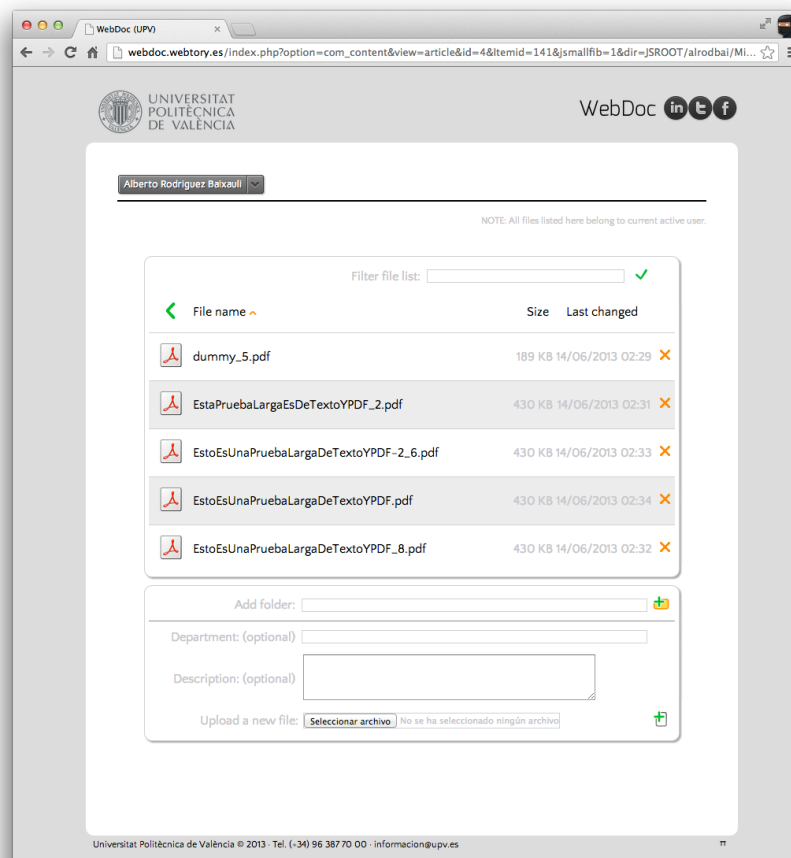


Figura 46. Un directorio con archivos en el Gestor de Documentos de la plataforma WebDoc.

3.7.4. Archivos tipo PDF.



Figura 47. Un directorio con archivos tipo PDF en el Gestor de Documentos de la plataforma **WebDoc**.

3.7.5. Categoría 1. Archivos tipo PDF.

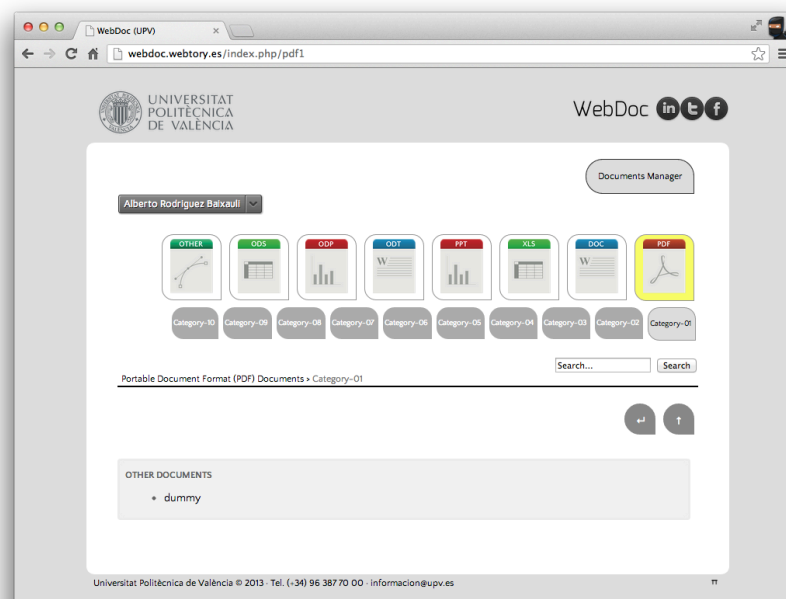


Figura 48. Un directorio con archivos tipo PDF pertenecientes a la categoría 1 en el Gestor de Documentos de la plataforma **WebDoc**.

3.7.6. Administración.

3.7.6.1. Acceso (Login).

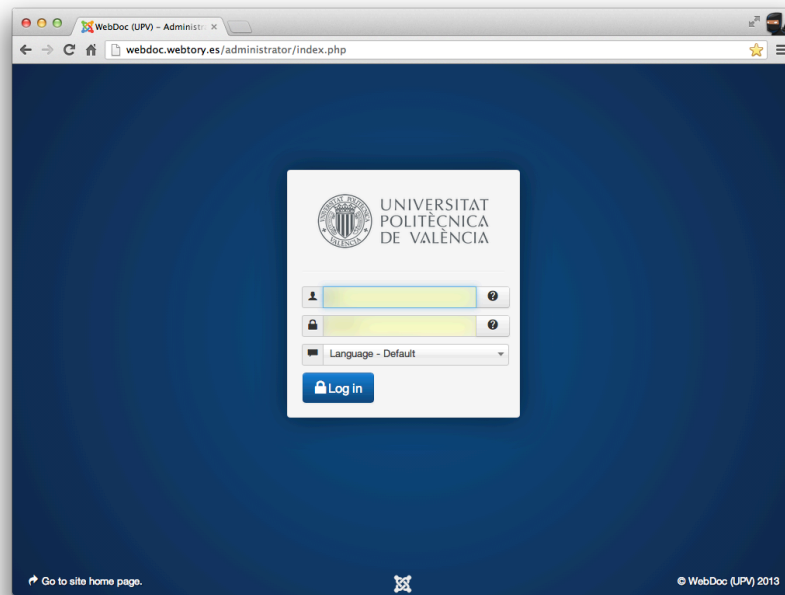


Figura 49. Pantalla de Acceso a la Administración de la plataforma **WebDoc**.

3.7.6.2. Listado Registros Documentos Borrados.

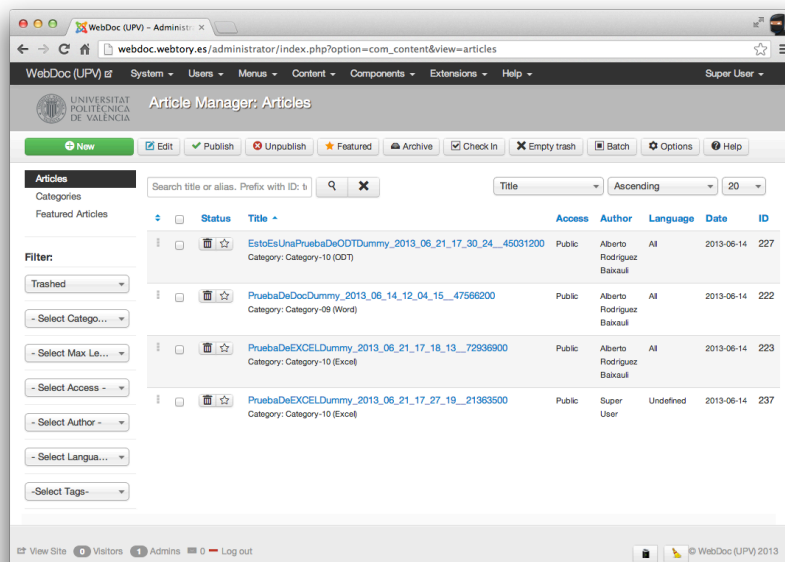


Figura 50. Listado de Registros de Documentos Borrados de la plataforma **WebDoc**.

3.7.6.3. Crear Perfiles de Usuario.

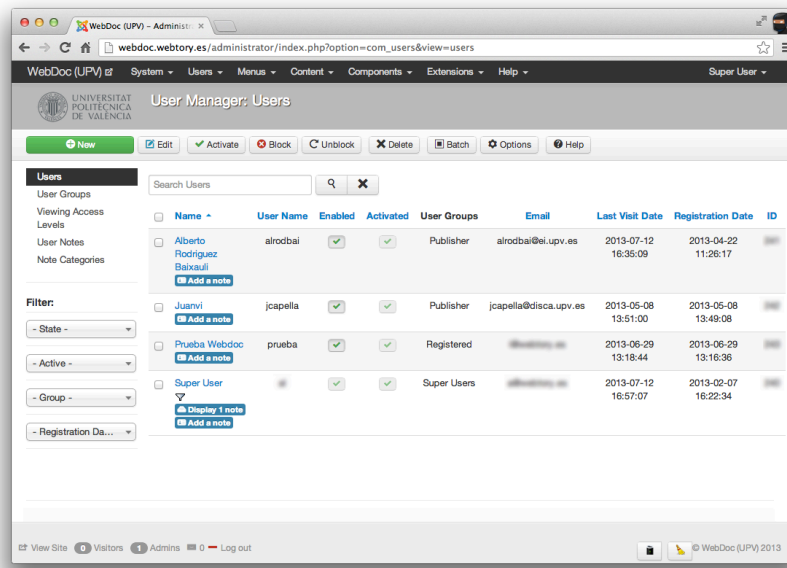


Figura 51. Pantalla de Creación de Perfiles de Usuario de la plataforma **WebDoc**.

3.7.6.4. Autorizar un Perfil de Usuario.

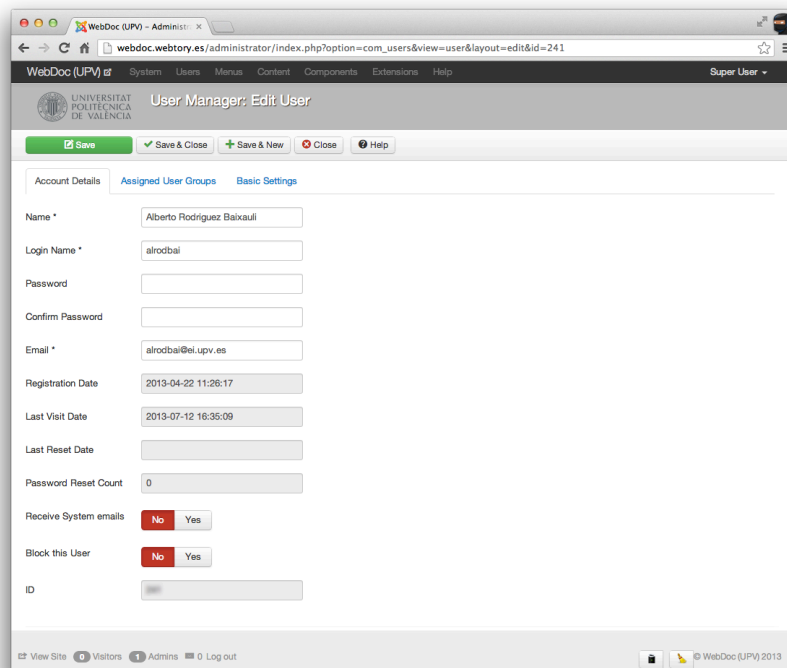


Figura 52. Pantalla de Modificación de Datos de un Perfil de Usuario de la plataforma **WebDoc**.

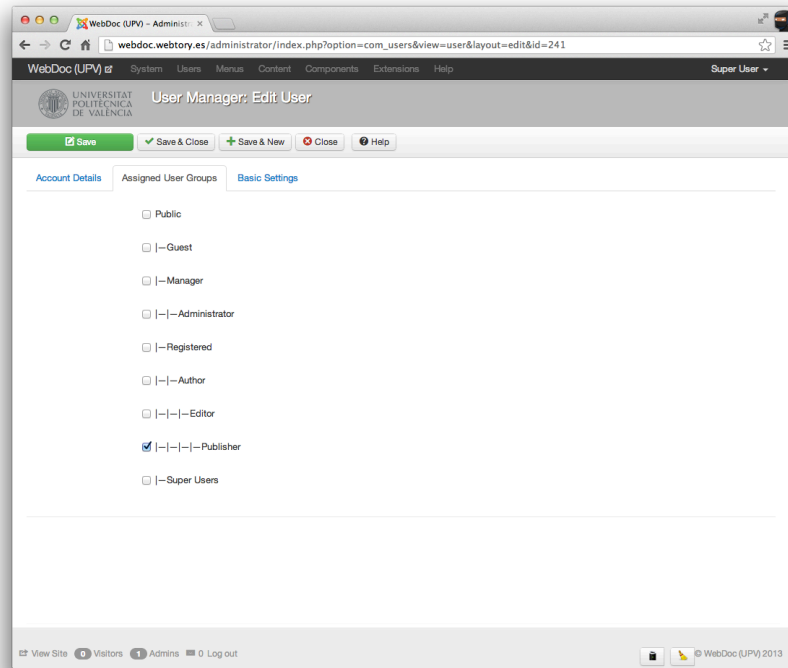


Figura 53. Pantalla de Cambio de Grupo (ACL) de un Perfil de Usuario de la plataforma **WebDoc**.

4. Conclusiones finales

4.1. Consecución de Objetivos.

(1) Se propuso un sistema de gestión de usuarios, con altas automáticas y acceso restringido por defecto. Aprovechando al máximo el sistema de gestión de usuarios de **Joomla!** que se puede describir de la siguiente manera:

***Joomla!** diferencia entre visitantes y usuarios registrados con permisos diferentes [...]. Los usuarios ya registrados saben, en su mayoría, lo que están buscando y llegan a la plataforma con ciertas expectativas.*

*Cuanto más usuarios tiene la plataforma, más compleja se torna la gestión de usuarios y permisos. En las versiones anteriores a **Joomla!** 1.6, había un sistema estático constituido por grupos de usuarios, permisos y niveles de acceso que no se podían cambiar. Con **Joomla!** 2.5 y superiores, este sistema aún se conserva en la configuración predeterminada (por defecto) de la llamada ACL, del inglés ACCESS CONTROL LIST (Lista de Control de Acceso).*

*Cada acceso al sitio será evaluado por un Grupo de Permisos, incluso el de los visitantes. Después de registrarse en la plataforma creada con **Joomla!**, el usuario se convertirá automáticamente en miembro de un Grupo de Permisos. El grupo tiene permisos predefinidos y pertenece a un determinado nivel de acceso. Un nivel de acceso puede tener cualquier número de Grupos de Permisos. Un grupo puede tener cualquier número de usuarios. Los permisos pueden ser heredados y reemplazados en diversos sitios [26].*

Se diseñó la plataforma **WebDoc** para que los usuarios se registraran ellos mismos en la plataforma (aprovechando esta característica de **Joomla!**). Al registrarse, los usuarios pasaban a formar parte del grupo “Registered”. Dicho grupo, dentro de la plataforma **WebDoc**, no puede hacer casi nada: modificar sus propios datos como usuario, hacer “login” y “logout”, pero no pueden acceder a la plataforma.

Esto se consiguió, haciendo que el módulo principal, que lista el contenido de documentos de la plataforma, así como el módulo de creación de documentos, sólo fuera accesibles para la ACL “Publisher”. De esta forma, el acceso de nuevos usuarios a la plataforma, está condicionado a que un Administrador (Super User) cambie a los nuevos usuarios de “Registered” a “Publisher”.

De esta forma se libera a los Administradores (Super Users) de la tarea de introducir la información de cada nuevo usuario (ya que es el propio usuario el que lo hace), pero siguen conservando la potestad de decidir quién puede acceder a la plataforma y quién no.

(2) y (3) En estos dos objetivos, se desglosó la propuesta de clasificación de los archivos en la plataforma **WebDoc**. Basándose en el

tipo de archivo y en una convención en el nombrado del mismo, se diseñó un sistema de clasificación semi-automática de archivos, (transformados en documentos dentro de la plataforma) que requiere de un mínimo esfuerzo por parte del usuario de la plataforma.

Revisando con detenimiento el objetivo (2) no se alcanzó el mismo completamente, debido a la falta de tiempo. La parte que no se desarrolló, fue la referente al proceso de la información contenida en el texto del archivo. Se deja dicha parte para un trabajo futuro, tal y como se expondrá en el punto siguiente de las conclusiones.

(4) y (5) En cuanto a la implementación de un módulo web, que se encargue de la creación, modificación y borrado de documentos dentro de la plataforma, se aprovechó el módulo desarrollado para **Joomla!**, y con licencia GNU, **jsmallfib** [27].

Se amplió dicho módulo, para que, cada vez que se produjera una subida de un archivo al sistema de archivos del servidor, se creara un artículo en la plataforma **Joomla!**; cada vez que se subiera un archivo ya existente en el servidor, sólo se cambiara el archivo y no el artículo en **Joomla!** (ya que para cambiar la meta-información del documento está el editor estándar de artículos de **Joomla!**) y cada vez que se borrara un archivo en el servidor, se borrara el archivo, y el artículo en **Joomla!** pasara a estar en modo “Trashed”, pero con una marca de tiempo, que le indicara al Administrador (Super User) cuándo se borró dicho documento.

(6) Siguiendo este objetivo, se introdujo un código de envío de correos dentro del módulo **jsmallfib**. Dicho código aprovecha toda la infraestructura creada en **Joomla!** para el envío de correos haciendo uso de las clases desarrolladas a tal efecto.

Se tuvo en cuenta el parámetro que hay en la gestión de usuarios de **Joomla!** que indica si un usuario recibirá correos del sistema o no. Es decir, si dicho parámetro está activado, la plataforma **WebDoc** enviará información a dicho usuario por correo, si no, no.

4.2. Trabajo futuro.

Tal y como se ha comentado en el punto anterior, las líneas de trabajo futuro empezarán implementando la segunda parte del objetivo marcado con (2).

Para ello, podría acotarse el desarrollo del procesado de la información contenida en el archivo (por lo menos en primera instancia) a los archivos del tipo *Microsoft® Excel®*.

Una primera aproximación podría procesar los archivos usando una de las bibliotecas PHP que pueden accederse en [28] y [29], contando con que los archivos de Excel contengan un formato similar al siguiente:

FECHA	CONCEPTO	IMPORTE	SALDO

Podría implementarse definiendo ciertas cadenas de caracteres configurables (como "transferencia papelería", "transferencia reprografía", "ingreso efectivo UD", etc) que se incluirían dentro de la columna CONCEPTO.

De esta forma, podría hacerse un desglose sencillo, agrupados por meses, en el que podría sumarse importes de todas las filas que tuvieran una cadena de caracteres común (p.ej. TOTAL en "transferencia papelería" para el mes X, TOTAL en "transferencia reprografía" para el mes Y, etc.).

Las cadenas de caracteres podrían almacenarse como subcategorías, de una nueva categoría llamada "Cadenas configurables" en **Joomla!**

De esta forma, el módulo encargado del proceso de los archivos EXCEL, podría disponer, en un sitio fácilmente accesible de Joomla!, las cadenas configurables. Para ello, habría que incluir un código similar al siguiente:

CÓDIGO PHP
<pre> <?php // ID's de las categorías a buscar su nombre. // Categorías "Cadenas configurables" ----- \$cat = [, , , , , , , ,]; // ----- \$cat_size = sizeof(\$cat); \$cat_name = array(); // Get the database object. \$this->db = JFactory::getDBO(); for(\$i=0;\$i<\$cat_size;\$i++) { \$query = \$this->db->getQuery(true); \$query->select(\$this->db->quoteName('title')); \$query->from(\$this->db->quoteName('#__categories')); \$query->where(\$this->db->quoteName('id') . ' = ' . \$cat[\$i]); \$this->db->setQuery(\$query); \$result_query = (String) \$this->db->loadResult(); array_push(\$cat_name, \$result_query); } ?> </pre>

Donde **\$cat** es un array que contiene los ID's de todas las subcategorías de la categoría "Cadenas configurables".



Se ha sugerido usar categorías, porque es uno de los objetos más sencillos de acceder para obtener una lista de cadenas de caracteres en **Joomla!**

En otro orden de cosas, se plantea, como trabajo futuro, remodelar toda la plataforma **WebDoc**, para que acepte más de 10 categorías por tipo de archivo.

En este momento, la plataforma se ha construido en base a módulos estáticos de **Joomla!** de ahí el número relativamente reducido de categorías por tipo de archivo, ya que el número de módulos estáticos necesarios para cada categoría es ingente, tal y como se puede apreciar en el mapa mental de *3.5.2. Implementación en Joomla!*.

Por ello se propone cambiar el módulo “Mini Frontpage” que lista los documentos de cada categoría, por otro módulo que sea dinámico y que cambie según la cantidad de subcategorías para cada tipo de archivo y que no dependa de un número fijo de las mismas.

4.2. Conclusiones.

En general, puede afirmarse que se han conseguido todos los objetivos propuestos inicialmente.

Se ha diseñado un sistema de clasificación de documentos semi-automático que requiere de una intervención mínima por parte del usuario de **WebDoc**. Se ha definido una interfaz gráfica, se cree que bastante adecuada a la gestión documental, basada en el tipo de archivo. Teniendo en cuenta que el usuario suele saber qué tipo de archivo está buscando para obtener una información concreta.

También se ha diseñado e implementado (en la creación de documentos en la plataforma) un sistema de acceso rápido a plataformas móviles, mediante la inclusión de códigos BIDI a cada documento creado en **WebDoc** para una descarga rápida del documento al dispositivo móvil.

La plataforma desarrollada es un sistema altamente autónomo, que requiere de una intervención mínima de los usuarios Administradores (Super Users), tal y como se espera de una plataforma de gestión documental.

Asimismo la plataforma puede generar una serie de avisos (en forma de correos electrónicos), a los usuarios Administradores (Super Users), para informar de la actividad de la plataforma, sin que los Administradores (Super Users) tenga que entrar en **WebDoc** periódicamente. Avisos que puede activarse o desactivarse desde el panel de control de la plataforma.

Y por último se ha conseguido que la plataforma lleve un registro de control tanto para documentos creados, como un histórico de los documentos que se han eliminado.

5. Bibliografía

- [1] *PHP Definition*. (s.f.) Accedido el 18 de Junio de 2013. <http://php.net/>
- [2] *HTML 5.1 Nightly* (s.f.) Accedido el 18 de Junio de 2013. <http://www.w3.org/html/wg/drafts/html/master/>
- [3] *CSS Specifications & Drafts* (s.f.) Accedido el 18 de Junio de 2013. <http://www.w3.org/Style/CSS/>
- [4] *ECMAScript® Language Specification* (Junio 2011) Accedido el 18 de Junio de 2013. <http://www.ecma-international.org/publications/standards/Ecma-262.htm>
- [5] *Joomla Core Features Overview* (s.f.) Accedido el 18 de Junio de 2013. <http://www.joomla.org/core-features.html>
- [6] *A history of HTML* (1998) Accedido el 19 de junio de 2013. <http://www.w3.org/People/Raggett/book4/cho2.html>
- [7] *The CSS Saga* (s.f.) Accedido el 19 de junio de 2013. <http://www.w3.org/Style/LieBos2e/history/>
- [8] *Historia de PHP* (s.f.) Accedido el 19 de Junio de 2013. <http://www.php.net/manual/es/history.php.php>
- [9] Matilde Celma Giménez, Juan Carlos Casamayor Ródenas, Laura Mota Herranz. “El lenguaje SQL”, *Bases de datos relacionales*, Pearson Prentice-Hall, pp. 107-180, 2003.
- [10] *A short history of javascript* (s.f.) Accedido el 19 de Junio de 2013. http://www.w3.org/community/webed/wiki/A_Short_History_of_JavaScript
- [11] *A JQuery Description* (s.f.) Accedido el 20 de Junio de 2013, <http://es.wikipedia.org/wiki/JQuery>
- [12] *Selectors in JavaScript* (22 de Agosto de 2005) Accedido el 20 de Junio de 2013. <http://ejohn.org/blog/selectors-in-javascript/>
- [13] *Announcing the JQuery Blog* (24 de Enero de 2006) Accedido el 20 de Junio de 2013. <http://blog.jquery.com/2006/01/>
- [14] *What is Joomla?* (s.f.) Accedido el 20 de Junio de 2013. <http://www.joomla.org/about-joomla.html>
- [15] Roger S. Pressman. “Ingeniería Web”, *Ingeniería del Software. Un enfoque práctico* (6ª Edición), Mc Graw Hill Interamericana, pp. 502-503, 2005.
- [16] *The Apache Software Foundation* (s.f.) Accedido el 1 de Julio de 2013. <http://www.apache.org/>
- [17] *MySQL. The world's most popular open source database* (s.f.) Accedido el 1 de Julio de 2013. <http://www.mysql.com/>
- [18] *Akeeba Backup* (s.f.) Accedido el 1 de Julio de 2013. <https://www.akeebabackup.com/>



- [19] *Systems Modeling Language (SysML)* (s.f.) Accedido el 3 de Julio de 2013. <http://www.sysml.org/>
- [20] I. Jacobson, G. Booch, J. Rumbaugh. “Captura de requisitos como casos de uso”, *El proceso unificado de desarrollo de software*, Pearson - Addison Wesley, pp. 125-163, 2000.
- [21] I. Jacobson, G. Booch, J. Rumbaugh. “Análisis”, *El proceso unificado de desarrollo de software*, Pearson - Addison Wesley, pp. 165-204, 2000.
- [22] I. Jacobson, G. Booch, J. Rumbaugh. “Diseño”, *El proceso unificado de desarrollo de software*, Pearson - Addison Wesley, pp. 205-253, 2000.
- [23] I. Jacobson, G. Booch, J. Rumbaugh. “Implementación”, *El proceso unificado de desarrollo de software*, Pearson - Addison Wesley, pp. 255-279, 2000.
- [24] Bernd Bruegge, Allen H. Dutoit. “Pruebas”, *Ingeniería de Software Orientado a Objetos*, Prentice Hall. Pearson Educación de México, pp. 327-369, 2002.
- [25] I. Jacobson, G. Booch, J. Rumbaugh. “Prueba”, *El proceso unificado de desarrollo de software*, Pearson - Addison Wesley, pp. 281-299, 2000.
- [26] *Joomla! 2.5 Guía para Principiantes* (s.f.) Accedido el 16 de Julio de 2013. <http://ayuda.joomlaspanish.org/ayuda-joomla/j25es.pdf>
- [27] *Jsmallfib* (s.f.) Accedido el 16 de Julio de 2013 <http://www.smallerik.com/index.php/joomla-extensions/jsmallfib>
- [28] *PHPExcel* (s.f.) Accedido el 16 de Julio de 2013 <http://phpexcel.codeplex.com/>
- [29] *PHP-ExcelReader* (s.f.) Accedido el 16 de Julio de 2013 <http://sourceforge.net/projects/phpexcelreader/>