



**Universidad de las Ciencias Informáticas**

**Facultad 10**

**Título: Implementación de la Interfaz Web para el Gestor  
de Contenido Empresarial Alfresco.**

*excríba*

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS  
INFORMÁTICAS**

**AUTORES:**

Osvanys González Ríos.

Jorge Ariel Salomón Rodríguez.

**TUTOR:**

Ing. Misael Fonseca Mata.

**Ciudad de la Habana, junio de 2009.**

*"Si salgo llego, si llego entro, si entro triunfo"*

*Fidel Castro.*

## DECLARACIÓN DE AUTORÍA.

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

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año 2009.

### AUTORES

---

Osvanys González Rios.

---

Jorge Ariel Salomón Rodríguez

### TUTOR

---

Ing. Misael Fonseca Mata

## ***Agradecimientos:***

*A la Revolución por hacer realidad el sueño de graduarme.*

*A la Universidad de las Ciencias Informáticas por forjarme como profesional.*

*A toda mi familia, por siempre estar ahí para mí. A mis amigos de la universidad que siempre han sido y serán un gran apoyo. Le agradezco a mi grupo: el 10x01, por mantenernos 5 años inolvidables.*

*A mi tutor Misael por estar pendiente siempre de nuestro trabajo y ayudarnos en todo lo que nos hizo falta. Al grupo de proyecto Gestión Documental y Archivística que todos fueron de mucha ayuda.*

*Gracias a todas las personas que de una forma u otra, contribuyeron en mi formación profesional así como al desarrollo de este trabajo.*

*Osvany.*

*A la Universidad de las Ciencias Informáticas por permitirme aprender algo nuevo cada día y por formarme como ingeniero en Ciencias Informáticas.*

*A mi familia, a mi madre que siempre estuvo junto a mí, a mi hermana. A todos mis compañeros del aula, los amigos del apartamento, a Jorge Luis Rodríguez Monteagudo (el tigre), Yosbel Santana Castillo (el curro) y Ailet Ávila Pérez que han sido los mejores amigos durante estos 5 años.*

*A mi tutor Misael Fonseca Mata, que siempre estuvo ayudándonos a salir adelante con cualquier problema que se presentara.*

*A todos los que de una forma u otra ayudaron a la realización de este trabajo, Michel David Suarez, Rafael Espinoza Añel, y otros, le agradezco toda su ayuda.*

*Jorge Ariel.*

## ***Dedicatoria:***

*A mis padres por toda su dedicación y amor, por la educación que me brindaron y el apoyo para llegar hasta aquí. A mi hermano, mi ejemplo a seguir en mi vida estudiantil. A mi tía Nilda por luchar tanto con nosotros, a toda mi familia y amigos que siempre han estado presentes.*

*Osvany.*

*A mi madre, y a mi hermana que son las personas que más quiero.*

*Jorge Ariel.*

## RESUMEN

La información es un recurso fundamental para la realización de las actividades de una organización y para la toma de decisiones por parte de una administración. Adicionalmente, tenemos que aproximadamente un 80 % de la información de una institución reside en los documentos y que los avances tecnológicos han provocado la aparición de nuevos productos y herramientas que se han hecho más requeridos y utilizados incluso en nuestras vidas.

El fácil entorno de trabajo de un usuario con un servicio WEB ha sido siempre la tarea principal para lograr un mayor entendimiento del cliente. Con el objetivo de implementar un servicio WEB con una interfaz amigable, donde debe permitir llevar el control de todo el flujo de documentos de cualquier institución para agilizar los procesos relacionados con el control del contenido entrantes y/o salientes, este trabajo de diploma se enmarca en la necesidad de la interoperabilidad entre una capa de presentación y el ECM Alfresco para la Gestión Documental y Archivística.

En esta ponencia se pretende presentar la experiencia desarrollada a través del análisis de los elementos que intervinieron en su puesta en marcha y su integración con los procesos habituales, como reflejo de una adecuada Gestión Documental y Archivística.

Tabla de contenido:	
Agradecimientos:.....	III
Dedicatoria:.....	IV
RESUMEN.....	V
INTRODUCCIÓN:.....	1
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....</b>	<b>5</b>
1.1 Definición y objetivos de Gestión Documental. ....	5
1.1.2 Gestión Documental Eficiente.....	6
1.2 Concepto de sitio web.....	6
1.3 Clasificación de sitios web. ....	6
1.3.1 Clasificados por la audiencia.....	6
1.3.2 Clasificado por su dinamismo. ....	7
1.3.3 Clasificado por su apertura.....	7
1.3.4 Clasificado por su profundidad. ....	7
1.3.5 Clasificado por sus objetivos.....	7
1.4 Páginas Dinámicas y Estáticas. ....	8
1.4.1 Páginas Estáticas.....	8
1.4.2 Páginas Dinámicas ....	8
1.4.2.1 Paginas Dinámicas del lado del cliente.....	9
1.4.2.2 Paginas Dinámicas del lado del servidor .....	9
1.5 Lenguajes de programación para la web.....	10
1.5.1 Lenguaje HTML. ....	10
1.5.2 Lenguaje Java script.....	11
1.5.3 Lenguaje PHP.....	12
1.5.4 Lenguaje Python. ....	13
1.5.5 Lenguaje Java. ....	13
1.6 Sistemas Gestores de Bases de Datos (SGBD). ....	14
1.6.1 MySQL. ....	14
1.6.2 PostgreSQL. ....	14
1.6.3 Oracle (Relational Data Base Management System). ....	14
1.7 Metodologías de desarrollo de software.....	15

1.7.1 Metodología RUP. ....	15
1.7.2 XP (eXtreme Programming).....	17
1.7.3 FDD (Feature Driven Development). ....	17
1.7.4 Metodología Métrica .....	18
<b>1.8 Gestores de Contenido Empresariales (ECM Enterprise Content Management).....</b>	<b>19</b>
1.8.1.1 Gestor de Contenido Empresarial Alfresco. ....	20
1.8.1.1.1 Servicios de red de Alfresco para el cliente.....	21
1.8.1.1.2 Características. ....	22
1.8.2 Framework Code Igniter. ....	23
1.8.2.1 Razones por la cual elegir Code Igniter.....	23
1.8.3 ¿Qué es SDK? .....	24
1.8.3.1 ¿Qué es una API? .....	25
1.8.3.2 API de PHP .....	25
1.8.3.2.1 Inconvenientes de la API PHP.....	26
<b>1.9 Estado del Arte de los Sistemas Implementados.....</b>	<b>26</b>
1.9.1 Descripción de los Sistemas.....	26
1.9.1.1 Módulo gmv .....	26
1.9.1.2 DoCASU .....	27
1.9.1.3 Módulo Joomla.....	29
1.9.1.4 CMS Opencms.....	30
<b>1. 10 Necesidad de un Sistema Gestor de Contenido.....</b>	<b>30</b>
<b>1.11 ¿Qué ventajas posibilitaría una capa de presentación configurable para un Sistema de Gestión Empresarial? .....</b>	<b>31</b>
<b>1.12 Propuesta de solución al problema de la investigación.....</b>	<b>31</b>
<b>CAPÍTULO 2: PROPUESTA DEL SISTEMA.....</b>	<b>33</b>
2.1 Problema. ....	33
2.2 Información que se maneja.....	33
2.3 Propuesta del Sistema. ....	34
2.4 Modelo de Dominio. ....	34
2.4.1 Entidades y Conceptos.....	34
2.4.2 Representación del diagrama de clases Modelo de Dominio. ....	35
2.5 Levantamiento de Requerimiento. ....	36



2.5.1 Requisitos Funcionales.....	36
2.5.2 Requisitos no Funcionales.....	37
2.6 Definición de los Casos de Usos y Actores. ....	38
2.6.1 Definición de los Actores del Sistema.....	39
2.6.2 Casos de Usos del Sistema.....	39
2.6.3 Diagrama de casos de Usos del Sistema. ....	40
2.6.4 Descripción de los Casos de Usos del Sistema. ....	41
<b>CAPÍTULO 3: DISEÑO DEL SISTEMA. ....</b>	<b>48</b>
3.1 Diseño. ....	48
3.2 Representación UML del diagrama de clases del Diseño.....	49
3.3 Representación UML de los diagramas de Estereotipos Web del Diseño. ....	52
3.4 Representación UML de los diagramas de interacción. ....	57
3.4.1 Diagrama de secuencia CU Autenticar Usuario.....	57
3.5.1 Diagrama de secuencia por escenario. ....	58
3.5.1.1 Caso de Uso Gestionar Espacio. ....	58
3.5.1.2 Caso de Uso Gestionar Contenido ....	61
3.5.1.3 Caso de Uso Gestionar Metadatos. ....	64
3.5.1.4 Caso de Uso Gestionar Reglas. ....	65
3.6 Descripción de las Clases del Diseño.....	66
<b>CAPÍTULO 4: IMPLEMENTACIÓN. ....</b>	<b>73</b>
4.2 Diagrama de Despliegue. ....	73
4.3 Estructuración del Modelo de Implementación.....	73
4.4 Prueba de caja Negra ....	77
4.4.1 Casos de Prueba de Integración.....	77
<b>CONCLUSIONES:.....</b>	<b>86</b>
<b>RECOMENDACIONES:.....</b>	<b>87</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>87</b>
<b>BIBLIOGRAFÍA. ....</b>	<b>88</b>
<b>GLOSARIO DE TÉRMINOS:.....</b>	<b>90</b>

### **INTRODUCCIÓN:**

El ser humano siempre ha tenido la necesidad inherente a su existencia de dejar plasmado extensiones de su propia memoria, ya sea para registrar información relativa a los diferentes roles que asume en su devenir o simplemente materializar su intrínseco afán por trascender en el mundo de las ideas, para ello se ha valido de los soportes que su entorno inmediato le ha posibilitado, recurriendo como lo muestran los registros históricos de materiales que han permitido registrar las tecnologías de lenguaje escrito que en diferentes regiones del mundo se han desarrollado, recurriendo a muros en cavernas, tablillas de barro, papiros, cueros, celulosa, etc.

Se debe señalar que hablar de gestión no es otra cosa que el proceso mediante el cual se obtiene, despliega o utiliza una variedad de recursos básicos para apoyar y lograr los objetivos de una institución. En este caso el recurso es la información contenida en los documentos. La gestión de documentos es definida como una parte del sistema de información de la empresa desarrollado con el propósito de almacenar y recuperar documentos, que debe estar diseñado para coordinar y controlar todas aquellas funciones y actividades específicas que afectan la creación, recepción, almacenamiento, acceso y preservación de los documentos, salvaguardando sus características estructurales y contextuales, y garantizando su autenticidad y veracidad.

Con el desarrollo del diseño de una interfaz web amigable para el Sistema de Control de Documentos (SCD) se pretende modelar una Aplicación Web de entorno libre que permita el control eficiente de la correspondencia entrante y saliente del personal que interactúa con la aplicación, así como otros documentos generados en el mismo.

Es necesario implementar una aplicación web capaz de gestionar, controlar y dar seguimiento a todo tipo de información de forma dinámica. Para lograrlo se ha hecho un estudio de los Gestores de Contenidos Empresariales (ECM) y herramientas o aplicaciones para la realización de una interfaz amigable al usuario final, los cuales son sistemas muy utilizados en el mundo entero en cuanto a la elaboración de servicios web, debido a que contienen funcionalidades necesarias para el manejo de contenido dinámico y el diseño de un servicio web.

La gestión de contenido empresarial o Enterprise Content Management (ECM) es una de las estrategias y tecnología empleadas en la industria de la tecnología de la información para manejar la captura, almacenamiento, seguridad, control de versiones, recuperación, distribución, conservación y destrucción de documentos y contenidos. Evidentemente se debe entender que los documentos también son contenidos y deben ser gestionados como tales.

Los Gestores de Contenidos Empresariales son la tecnología empleada para capturar, gestionar, almacenar, conservar y distribuir documentos y contenidos relacionados con los procesos de la organización, ofrecer un repositorio centralizado, junto con una serie de servicios que facilite su creación, organización y recuperación. Permiten la gestión de la información no estructurada de una institución.

Alfresco es un sistema de administración de contenidos de código libre / abierto, basado en estándares abiertos y de escala empresarial. Está diseñado para usuarios que requieren un alto grado de modularidad y rendimiento escalable. Este incluye un repositorio de contenidos, un framework de portal web para administrar y usar contenido estándar en portales, una interfaz CIFS (Sistema de Archivo Común de Internet) que provee compatibilidad de sistemas de archivos en Windows y sistemas operativos similares a Unix, un sistema de administración de contenido web, capacidad de visualizar aplicaciones web y sitios estáticos. Está desarrollado en Java.

Es una alternativa muy importante de código abierto (open source) para la Gestión de Documentos. El modelo de código abierto permite utilizar todo tipo de tecnologías y contribuciones que brinda la comunidad de código abierto de software libre, lo que permite un desarrollo más rápido y menos costoso.

Ofrece el principal sistema de gestión de documentos de código abierto que brinda búsquedas y colaboración de documentos con servicios completos de biblioteca y gestión de ciclo de vida.

Alfresco cuenta con disímiles características para una adecuada Gestión Documental y Archivística, es versionado a nivel de repositorio (similar a Subversion), Superposición transparente, cuenta con Gestión de registros y Gestión de imágenes, posee una publicación integrada, servicios descentralizados, soportes en varios idiomas, empaquetamiento de aplicación portable, soporte multiplataforma, etc.

En la actualidad han surgido diversos sistemas que permiten la gestión del contenido empresarial, y que por ende vienen a facilitar la integración de todos los intentos de homogeneizar y gestionar la información de una institución. Esto sugiere cambios en las empresas desde la perspectiva de adoptar las facilidades que brindan estos sistemas de integración y con ello adaptarlos a las especificidades y necesidades de cada una, entendiendo así cuestiones como la forma de visualizarse los contenidos, las funcionalidades, etc. Sin embargo, los sistemas gestores del contenido empresarial, y en particular Alfresco no cuenta con una capa de presentación amigable, por tal razón surge como **problema** de investigación la imposibilidad del ECM Alfresco de brindar una interfaz amigable y adaptable a las

necesidades del usuario, para realizar una adecuada Gestión Documental y Archivística. Es por ello que la presente investigación centra **su objeto de estudio** en los Gestores de Contenidos Empresarial.

El **campo de acción** es la interoperabilidad del Gestor de Contenidos Empresarial Alfresco.

Se propone como **objetivo general** implementar una interfaz amigable al usuario, para una adecuada Gestión Documental y Archivística.

Para cumplimentar este objetivo general se han definido algunas **tareas específicas**:

1. Estudiar las características fundamentales de un Sistema de Gestión Documental y Archivística.
2. Analizar el SDK (Software Development Kit) para el lenguaje de programación PHP (Hypertext Pre-processor) que tiene integrado ECM Alfresco para lograr la interoperabilidad.
3. Analizar las características y funcionalidades que brinda los Framework para la implementación de la interfaz.
4. Implementar la interfaz web al usuario, a partir de la capa de servicios web que brinda el ECM Alfresco.
5. Entrevistar especialistas en Gestión Documental y Archivística en la universidad para conocer los procesos elementales de sistemas de este tipo.

Los métodos teóricos que son utilizados para darle cumplimiento a estas tareas son el Analítico-Sintético, al identificar los conceptos y definiciones más importantes relacionados con la descripción de documentos que permita generar luego, una propuesta adecuada a la situación planteada y la tecnología estudiada; y el Histórico-Lógico que permite llevar a cabo un estudio de las diferentes herramientas de autor existentes, sus características, ventajas y desventajas. También utilizamos métodos empíricos como la entrevista para obtener toda la información necesaria para diseñar e implementar el sistema, en lo referente a tipos de entradas, tipos de salidas, descripción de flujos de trabajos, definición de responsabilidades de los trabajadores del negocio.

### **Resultados Esperados:**

Un sistema con una interfaz web amigable para la Gestión Documental y Archivística que sea adaptable a las necesidades de cualquier institución.

El presente trabajo consta de una introducción, cuatro capítulos, conclusiones generales, recomendaciones, referencias bibliográficas utilizadas durante el desarrollo del trabajo y por último, los anexos que complementan el cuerpo del trabajo.

En el Capítulo 1 "Fundamentación teórica" se aborda de forma general los principales conceptos que fundamentan la investigación, entre los que están CMS, EMC, entre otros. Además se realiza un estudio sobre las principales Herramientas en la actualidad, sus ventajas, funcionalidades y estándares.

En el Capítulo 2 "Características del Sistema", se describe el flujo actual de los procesos involucrados en el campo de acción, estudiando cómo se deben ejecutar actualmente dichos procesos; también se describe el objeto de automatización, la propuesta del sistema, se determinan los requerimientos funcionales y no funcionales y se describen los casos de uso y los actores del sistema.

En el Capítulo 3 "Diseño del Sistema", se detallan, mediante un grupo de artefactos, la solución del sistema hasta el diseño. Para su modelado se utilizan los diagramas interacción, de clases del diseño.

En el Capítulo 4 "Implementación" se abordará todo lo relacionado con el flujo de trabajo de implementación realizando el diagrama de despliegue el cual describe cómo y dónde el sistema será puesto en funcionamiento, y el diagrama de implementación para dar una visión de cómo las clases, artefactos y otros elementos de bajo nivel, se unen para formar componentes de alto nivel así como las conexiones entre ellos.

El Glosario de términos se localizará al final del documento con los términos o palabras de difícil comprensión.

## **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.**

En el presente capítulo se define que es Gestión Documental y Archivística, se tratan varios temas relacionados con las aplicaciones Web, que abarca desde el concepto de sitios Web hasta la situación internacional de los mismos.

Se nombran algunos de los sitios Web existentes, centrando la atención en los enfocados en el contenido que ofrece a los usuarios. Se realiza una comparación entre los sitios estáticos y dinámicos. Se relacionan algunos de los lenguajes de programación destinados a la Web, con una breve explicación de cada uno de ellos. Se especifican aspectos relacionados con el uso de las tecnologías en la informática, se describen brevemente algunos gestores de bases de datos, caracterizándolos de forma sencilla y clara, además se trabaja el tema de las metodologías de desarrollo de software más conocidas, un tema muy común en el mundo de la informática. Se presenta el concepto de ECM, se caracteriza el ECM Alfresco. También se presenta el estado del arte en el mundo, en nuestro país y en nuestra universidad.

### **1.1 Definición y objetivos de Gestión Documental.**

Se entiende por gestión documental el conjunto de normas, técnicas y prácticas usadas para administrar el flujo de documentos de todo tipo en una organización, permitir la recuperación de información desde ellos, determinar el tiempo que los documentos deben guardarse, eliminar los que ya no sirven y asegurar la conservación indefinida de los documentos más valiosos, aplicando principios de racionalización y economía.

Es una actividad casi tan antigua como la escritura, que nació debido a la necesidad de "documentar" o fijar actos administrativos y transacciones legales y comerciales por escrito para dar fe de los hechos.

Este tipo de documentos se plasmaron sucesivamente en tablillas de arcilla, hojas de papiro, pergaminos y papel, cuya gestión se fue haciendo cada vez más compleja a medida que crecía el tamaño de los fondos documentales. Una adecuada Gestión Documental y Archivística se traza como objetivo principal conseguir menos documentos pero mejores, asegurando una documentación adecuada, evitando lo no esencial, simplificando los sistemas de creación y el uso de papeleo y mejorando las formas de cómo se organiza y recuperan los documentos.(1)

### **1.1.2 Gestión Documental Eficiente.**

La gestión documental eficiente implica suministrar el documento adecuado a la persona apropiada, en el momento apropiado con el menor costo posible. La correcta Gestión Documental permite a la institución realizar operaciones administrativas más eficientes, tomar decisiones; y crear, mantener, usar, guardar y hacer una evaluación de los documentos de manera eficaz y eficiente.(2)

### **1.2 Concepto de sitio web.**

En inglés website o web site, un sitio web es un sitio (localización) en la World Wide Web que contiene documentos (**páginas web**) organizados jerárquicamente. Cada documento (página web) contiene texto y o gráficos que aparecen como información digital en la pantalla de una computadora. Un sitio puede contener una combinación de gráficos, texto, audio, vídeo, y otros materiales dinámicos o estáticos.

Los mismos contienen información de un tema en específico, que se encuentra accesible para la totalidad del planeta mediante la red de redes.

La Web no sólo se limita a presentar textos y enlaces, sino que también puede ofrecer imágenes, videos y sonido, llegando a ser el servicio más rico en medios que tiene Internet. Por esta razón, para referirse al sistema que implementa la Web (hipertexto), se ha acuñado un nuevo término que es hipermedia, haciendo referencia a que la Web permite contenidos multimedia. (3)

### **1.3 Clasificación de sitios web.**

Los sitios se pueden clasificar de muchas maneras. Cada tipo de sitio tiene unas características y limitaciones propias. Una buena organización es vital para conseguir los objetivos del Sitio.(4)

#### **1.3.1 Clasificados por la audiencia**

**Públicos:** Es un Website normal, una página dirigida al público general, sin restricciones de acceso en principio.

**Extranet:** Son Sitios limitados por el tipo de usuarios que pueden acceder, por ejemplo los proveedores de una empresa determinada, o los clientes.

**Intranet:** Son sitios cuyo acceso está restringido a una empresa u organización, normalmente funcionan dentro de redes privadas, aunque no siempre es así.

### **1.3.2 Clasificado por su dinamismo.**

**Sitios Interactivos:** El usuario puede influir sobre el contenido del sitio que variará en función de cada usuario y de los objetivos de éste. Normalmente, las páginas se generan cuando el usuario las solicita, personalizando la información que se le ofrece.

**Sitios estáticos:** Los usuarios no pueden modificar o añadir nada al sitio, de cuyos contenidos se encargan exclusivamente sus diseñadores.

### **1.3.3 Clasificado por su apertura.**

**Estructura abierta:** Todos los documentos disponen de su dirección y los usuarios pueden acceder a cualquier punto del Website.

**Estructura cerrada:** Limita el acceso a unos pocos puntos de entrada (incluso a uno sólo). Un ejemplo sería un sitio que requiere un registro previo para entrar, el usuario siempre tendría que pasar primero por el registro antes de poder acceder al resto de la página.

**Estructura semicerrada:** A medio camino entre ambas, obliga a los usuarios a acceder por unos puntos específicos, como por ejemplo sólo la página principal y las páginas de entrada a las secciones más importantes.

### **1.3.4 Clasificado por su profundidad.**

Basada en el número de enlaces que hay que pulsar para llegar al contenido. En general los usuarios prefieren sitios poco profundos. Una buena regla a seguir es que el usuario no tenga que pulsar más de 3 enlaces para encontrar lo que busca.

### **1.3.5 Clasificado por sus objetivos.**

**Comerciales:** Están creados para promocionar los negocios de una empresa. Su finalidad es económica. Su audiencia puede estar formada por clientes (actuales y potenciales), inversores



(actuales y potenciales), empleados (actuales y potenciales) e incluso la competencia y los medios de comunicación. Podemos a su vez dividirlos en Corporativas (Informan sobre la empresa) y Promocionales (promocionan productos).

**Informativos:** Su finalidad principal es distribuir información. La audiencia de este tipo de sitios depende del tipo de información que distribuyen.

**Navegación:** Su finalidad es ayudar al usuario a encontrar lo que busca en Internet. Dentro de este grupo se sitúan los llamados portales, que intentan abarcar prácticamente todo dentro del propio sitio.

**Personales:** Al igual que los anteriores, son un medio de expresión de su creador o creadores. Sus objetivos y su audiencia pueden ser de lo más variopinto. Dentro de este grupo puede haber de todo desde colecciones de fotos de la familia hasta tratados científicos de primer orden.

#### **1.4 Páginas Dinámicas y Estáticas.**

Todos los sitios existentes en la web son derivados de dos tipos fundamentales de páginas: las estáticas que son las que no presentan ningún tipo de movimiento ni funcionalidades y las dinámicas que son aquellas con la que el usuario puede interactuar.(5)

##### **1.4.1 Páginas Estáticas.**

Un sitio Web estático consta de un conjunto de páginas y de archivos HTML relacionados alojados en un equipo que ejecuta un servidor Web.

Los sitios **Web estáticos** son aquellos sitios enfocados principalmente a mostrar una información permanente, donde el navegante se limita a obtener dicha información, sin que pueda interactuar con la página Web visitada, las Web estáticas están construidas principalmente con hipervínculos o enlaces (links) entre las páginas Web que conforman el sitio, este tipo de Web son incapaces de soportar aplicaciones Web como gestores de bases de datos, foros, consultas online, e-mails inteligentes.

##### **1.4.2 Páginas Dinámicas**

Las páginas dinámicas son aquellas con las que el usuario puede interactuar, es por esto que sólo con el lenguaje HTML, no se pueden construir, es necesario utilizar otros lenguajes de programación, por ejemplo PHP.

#### **1.4.2.1 Páginas Dinámicas del lado del cliente.**

Las páginas dinámicas del lado del cliente son las que se ejecutan en el navegador del usuario, es por esto que todo el procesamiento de los efectos y funcionalidades lo soporta éste.

Para crear los efectos y funcionalidades es necesario incluir código dentro del mismo archivo HTML, este código incluido es conocido como Script. Estos script el navegador se encarga de interpretarlos y ejecutarlos para realizar los efectos y funcionalidades. Los lenguajes más utilizados para escribir las páginas dinámicas son: Java script y Visual Basic Script (VBScript).

Ventajas del lado del cliente.

Las páginas dinámicas descargan al servidor algunos trabajos, brindan respuestas inmediatas a las acciones del usuario, además permiten la utilización de algunos recursos de la máquina local. Desventajas del lado del cliente Las páginas del cliente son muy dependientes del sistema donde se están ejecutando y esa es su principal desventaja, ya que cada navegador tiene sus propias características, incluso cada versión, y lo que puede funcionar en un navegador puede no funcionar en otro.

#### **1.4.2.2 Páginas Dinámicas del lado del servidor**

Las páginas dinámicas del lado del servidor son aquellas que son reconocidas, interpretadas y ejecutadas por el servidor.

Las páginas del lado del servidor tienen muchas aplicaciones, por ejemplo se pueden hacer con ellas agendas a foros, sistemas de documentación, estadísticas, juegos y chat.

Cuando se debe acceder a una información centralizada, situada en una base de datos en el servidor, éstas son imprescindibles. También son muy utilizadas cuando es necesario que algunos cálculos no sean realizados en la computadora del usuario, por problemas de seguridad. El funcionamiento sería el siguiente: cuando un cliente solicita una página de este tipo, el servidor ejecuta los scripts y se genera una página resultado, que solamente contiene código HTML, es decir .el cliente no va a ver nunca el scripts. Este resultado final es el que se envía al cliente y puede ser interpretado sin lugar a errores ni incompatibilidades, puesto que sólo contiene HTML.

El servidor es el que maneja toda la información de las bases de datos y cualquier otro recurso, como imágenes o servidores de correo y luego envía al cliente una página Web con los resultados de todas las operaciones.

Existen varios lenguajes que permiten escribir páginas dinámicas del lado del servidor como Perl, Active Server Pages (ASP), Hipertext Preprocesor (PHP), y Java Server Pages (JSP).

### Ventajas

Estas páginas son independientes del navegador del usuario, pues el código que reciben es HTML que es fácil de interpretar. Los script se ejecutan y transforman antes de enviarlos y esto trae consigo que el cliente no los puede ver.

### Desventajas

Como desventajas se puede señalar que será necesario un servidor más potente y con más capacidades que el necesario para las páginas de cliente. Además, estos servidores podrán soportar menos usuarios concurrentes, porque se requerirá más tiempo de procesamiento para cada uno.

## **1.5 Lenguajes de programación para la web.**

Actualmente existen diferentes lenguajes de programación para desarrollar en la web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas. En el presente artículo pretende mostrar las ventajas y desventajas de los lenguajes más conocidos.

Desde los inicios de Internet, fueron surgiendo diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. A medida que pasó el tiempo, las tecnologías fueron desarrollándose y surgieron nuevos problemas a dar solución. Esto dió lugar a desarrollar lenguajes de programación dinámicos para la web, que permitieran interactuar con los usuarios y utilizaran sistemas de bases de datos. A continuación daremos una introducción a los diferentes lenguajes de programación para la web.

### **1.5.1 Lenguaje HTML.**

Desde el surgimiento de internet se han publicado sitios web gracias al lenguaje HTML. Es un lenguaje estático para el desarrollo de sitios web (acrónimo en inglés de HyperText Markup Language, en

español Lenguaje de Marcas Hipertextuales). Desarrollado por el World Wide Web Consortium (W3C). Los archivos pueden tener las extensiones (htm, html).

Ventajas:

Sencillo de comprender, presenta un texto estructurado de forma agradable, no se necesita grandes conocimientos cuando se cuenta de un editor de páginas web, contiene archivos pequeños, despliegue rápido, es un lenguaje fácil de aprendizaje y lo admiten todos los navegadores.

Desventajas:

Que es un lenguaje estático, la interpretación de cada navegador puede ser diferente, guarda muchas etiquetas que pueden convertirse en basura que dificulta la corrección, el diseño es más lento y las etiquetas son muy limitadas.(6)

### **1.5.2 Lenguaje Java script.**

Este es un lenguaje interpretado, no requiere compilación. Fue creado por Brendan Eich en la empresa Netscape Communications. Utilizado principalmente en páginas web. Es similar a Java, aunque no es un lenguaje orientado a objetos, el mismo no dispone de herencias. La mayoría de los navegadores en sus últimas versiones interpretan código Java script. El código Java script puede ser integrado dentro de nuestras páginas web. Para evitar incompatibilidades el World Wide Web Consortium (W3C) diseño un estándar denominado DOM (en inglés Document Object Model, en su traducción al español Modelo de Objetos del Documento).

Ventajas:

Es un lenguaje de scripting seguro y fiable, los script tienen capacidad limitada por razones de seguridad y el código se ejecuta en el cliente.

Desventaja:

Presenta un código visible para cualquier usuario y el código debe descargarse completamente.(7)

### **1.5.3 Lenguaje PHP.**

Es un lenguaje de programación utilizado para la creación de sitio web. PHP es un acrónimo recursivo que significa “PHP Hypertext Pre-processor”, (inicialmente se llamó Personal Home Page). Surgió en 1995, desarrollado por PHP Group.

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión (php).

Ventajas:

Es muy fácil de aprender, se caracteriza por ser un lenguaje muy rápido, soporta la orientación a objeto, clase y herencia; en un lenguaje multiplataforma, linux, windows; tiene una capacidad de conexión con la mayoría de los manejadores de bases de datos, es libre por lo que se representa como alternativa de fácil acceso para todos.

Desventajas:

Se necesita instalar un servidor web, todos los trabajos los realiza el servidor y no delega al cliente, la legibilidad del código puede verse afectada al mezclar sentencia de HTML. La programación orientada a objetos puede verse aún muy deficientes para grandes aplicaciones y dificulta la organización por capas de la aplicación.

Seguridad:

PHP es un poderoso lenguaje e intérprete, ya sea incluido como parte de un servidor web en forma de módulo o ejecutado como un binario CGI separado, es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor. Estas propiedades hacen que cualquier cosa que sea ejecutada en un servidor web sea insegura por naturaleza. PHP está diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI que Perl o C, y con la selección correcta de opciones de configuración en tiempos de compilación y ejecución, y siguiendo algunas prácticas correctas de programación.(8)

#### **1.5.4 Lenguaje Python.**

Es un lenguaje de programación creado en el año 1990 por Guido van Rossum, es el sucesor del lenguaje de programación ABC. Python es comparado habitualmente con Perl. Los usuarios lo consideran como un lenguaje más limpio para programar. Permite la creación de todo tipo de programas incluyendo los sitios web.

Su código no necesita ser compilado, por lo que se llama que el código es interpretado, lo cual fuerza a que los programadores adopten por un estilo de programación particular:

- Programación orientada a objeto.
- Programación estructurada.
- Programación funcional.
- Programación orientada a aspectos.

Ventajas:

Es libre y tiene fuente abierta, posee gran cantidad de funciones y librerías, sencillo y rápido de programar, es multiplataforma, tiene licencia de código abierto, es orientado a objeto y portable.

Desventajas:

Lentitud por ser un lenguaje interpretado.

#### **1.5.5 Lenguaje Java.**

Este lenguaje de programación fue desarrollado por James Gosling, y su equipo en Sun Microsystems, entre 1990 y 1994, pensado originalmente como un reemplazo de C++, orientado a dispositivos embebidos, y a la televisión interactiva, posteriormente para rescatarlo del fracaso y fue reorientado hacia su aplicación en la Web. El lenguaje Java tiene cinco características principales entre las que se encuentran: orientado a objetos, multiplataforma, soporte integrado para redes de computadoras, diseñado para ejecutar código de fuentes remotas de modo seguro, y la curva de aprendizaje es más lenta.

## **1.6 Sistemas Gestores de Bases de Datos (SGBD).**

Los Sistemas de Gestión de Base de Datos (SGBD) son una herramienta software que proporcionan una interfaz entre los datos almacenados y las aplicaciones que permiten acceder a los mismos.

Además posibilita crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad. Algunos ejemplos de sistemas Gestores de Bases de Datos son MySQL, PostgreSQL, Oracle y SQL Server.

### **1.6.1 MySQL.**

Debido a que está construido para la economía Web, MySQL es la base de datos de fuente abierta más popular del mundo. Ahora, con los Sistemas Sun para MySQL, usted puede escalar su solución de MySQL, obtener hasta 3 veces más desempeño, recortar los costos hasta la mitad y beneficiarse de la experiencia técnica de Sun en sistemas y software. En la actualidad Sun ha sido comprada por Oracle y que el mayor desarrollador de bases de datos libres haya sido comprado por el mayor desarrollador de bases de datos privativas no es buena noticia.

Por su sencillez y sus características es usado por muchas personas, consume muy pocos recursos, se usa tanto en aplicaciones sencillas como complejas. Es utilizado en aplicaciones Web como Drupal, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla, además sus conexiones son muy seguras. Se decidió usar este sistema gestor de bases de datos en la realización del portal.

### **1.6.2 PostgreSQL.**

Es un magnífico gestor de bases de datos, es multiplataforma. Se encuentra bajo la licencia BSD. Permite una fácil gestión de los usuarios y de las bases de datos que contenga el sistema. Sirve de soporte a los lenguajes más populares como PHP, C, C++, Java, Python, Ruby, etc, y al protocolo de comunicación encriptado por SSL. El número de base de datos que puede contener es ilimitado.

### **1.6.3 Oracle (Relational Data Base Management System).**

Es considerado uno de los sistemas de bases de datos más completos, por su estabilidad, soporte de transiciones, escalabilidad y ser multiplataforma, sin embargo el precio de Oracle es sumamente

costoso. Otro aspecto que ha sido criticado por algunos especialistas es la seguridad de la plataforma, y las políticas de suministro de parches de seguridad, modificadas a comienzos de 2005 y que incrementan el nivel de exposición de los usuarios. En los parches de actualización provistos durante el primer semestre de 2005 fueron corregidas 22 vulnerabilidades públicamente conocidas, algunas de ellas con una antigüedad de más de 2 años.

### **1.7 Metodologías de desarrollo de software.**

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software.

En un proyecto de desarrollo de software la metodología define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo.

No existe una metodología de software universal. Las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable.

Dentro de la ingeniería de software podemos encontrar algunas metodologías que se encargan de elaborar estrategias de desarrollo de software que promueven nuevas prácticas para un buen desarrollo del mismo, entre las más conocidas encontramos: Rational Unified Process (RUP), eXtreme Programming (XP), Metodología Métrica y Feature Driven Development (FDD).

#### **1.7.1 Metodología RUP.**

Es uno de los procesos más generales de los existentes actualmente, ya que en realidad está pensado para adaptarse a cualquier proyecto.

El RUP divide el proceso de desarrollo en ciclos, obteniendo una versión del software al final de cada ciclo que se divide en fases que finalizan con un hito donde se debe tomar una decisión importante:

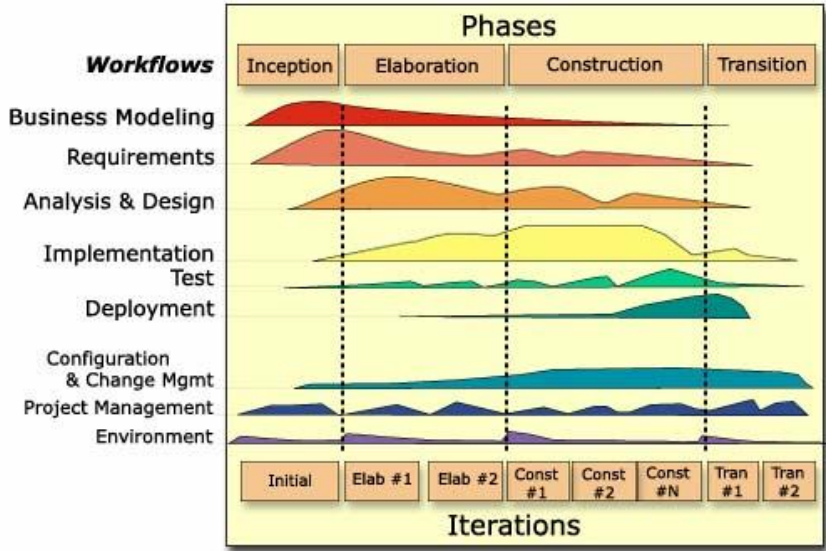
- Fase de inicio: Se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos (puesta en marcha).
- Elaboración: Se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos (definición, análisis, diseño).



- Construcción: Se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario (implementación).
- Transición: Se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados (fin del proyecto y puesta en producción).

Define también 9 actividades a realizar en cada fase del proyecto y en los flujos de trabajo.

- Modelo del Negocio.
- Requerimientos.
- Análisis y Diseño.
- Implementación.
- Prueba.
- Distribución.
- Gestión de configuración y cambio.
- Gestión de proyecto.
- Gestión de entorno.



Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de caso de uso, el código fuente, etc.) y define una serie de roles que se distribuyen entre los miembros del proyecto y las tareas de cada uno a lo largo del proceso.

El ciclo de vida de RUP se basa en casos de uso para describir lo que se espera del software y está muy orientado a la arquitectura del sistema, es iterativo e incremental, basándose en UML (Unified Modeling Language) como herramienta principal. (9)

### **1.7.2 XP (eXtreme Programming).**

La programación extrema es una metodología reciente (alrededor de 5 años) utilizada en el desarrollo de software. La filosofía de XP es satisfacer al completo las necesidades del cliente, por eso, lo integra como una parte más del equipo de desarrollo.

XP está diseñada para el desarrollo de aplicaciones que requieren un grupo de programadores pequeño, dónde la comunicación sea más factible que en grupos de desarrollo grandes. La comunicación es un punto importante y debe realizarse entre los programadores, los jefes de proyecto y los clientes. (10)

- Comunicación.
- Simplicidad.
- Realimentación (Feedback).

### **1.7.3 FDD (Feature Driven Development).**

FDD es un proceso diseñado por Peter Coad, Erich Lefebvre y Jeff De Luca y se podría considerar a medio camino entre RUP y XP.

FDD está pensado para proyectos con tiempo de desarrollo relativamente cortos (menos de un año). Se basa en un proceso iterativo con iteraciones cortas (~2 semanas) que producen un software funcional que el cliente y la dirección de la empresa pueden ver y monitorizar.

Un proyecto que sigue FDD se divide en 5 fases:

- Desarrollo de un modelo general.
- Construcción de la lista de funcionalidades.

- Plan de releases en base a las funcionalidades a implementar.
- Diseñar en base a las funcionalidades.
- Implementar en base a las funcionalidades.

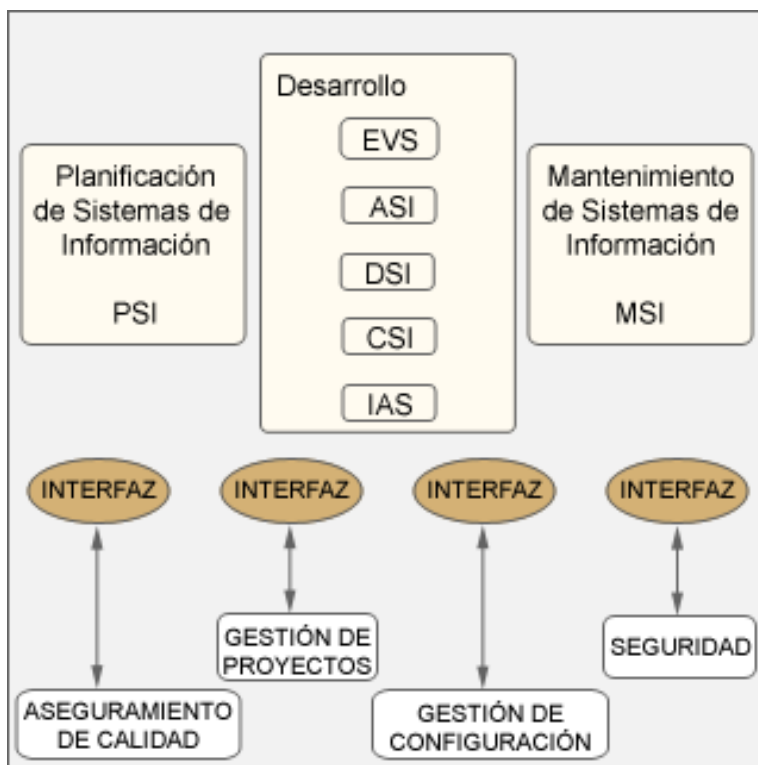
El trabajo (tanto de modelado como de desarrollo) se realiza en grupo, aunque siempre habrá un responsable último (arquitecto jefe o jefe de programadores en función de la fase en que nos encontremos), con mayor experiencia, que tendrá la última palabra en caso de no llegar a un acuerdo. Al hacerlo en grupo se consigue que todos formen parte del proyecto y que los menos inexpertos aprendan de las discusiones de los más experimentados, y al tener un responsable último, se asignan las responsabilidades que todas las empresas exigen. (11)

#### **1.7.4 Metodología Métrica**

Métrica tiene ya varios años de vida y su actual versión, la 3, se crea con la finalidad de incorporar las nuevas técnicas derivadas de la programación y el análisis orientado a objetos, al proceso de desarrollo de software que a través de esta metodología las administraciones públicas pretenden llevar a cabo.

Métrica debería ser obligatoria en cuantos contratos de desarrollo de software realiza la administración pública, sin embargo su utilización aún no está suficientemente universalizada. Todavía es frecuente ver concursos públicos donde no se establece la obligatoriedad para el contratante de seguir las pautas de métrica a la hora de llevar a cabo un proceso de construcción de software.

Todo proyecto Métrica 3 consta de un conjunto de fases que se desglosan en múltiples puntos cuya cronología hay que seguir con claridad para ir avanzando en el desarrollo del proyecto. El siguiente gráfico muestra el esquema general de estas fases:



La metodología MÉTRICA Versión 3 proporciona un conjunto de métodos y técnicas que guía a los distintos profesionales de Sistemas y Tecnologías de la Información y Comunicaciones (STIC) en la obtención de los diversos productos de los procesos del ciclo de vida de un proyecto informático. Con el fin de mejorar la productividad de los distintos participantes y asegurar la calidad de los productos resultantes, la mayoría de las técnicas propuestas están soportadas por herramientas disponibles en el mercado que automatizan en mayor o menor grado su utilización. En cualquier caso, no todos los productos resultantes de cada tarea son susceptibles de obtenerse de forma automatizada. (12)

### **1.8 Gestores de Contenido Empresariales (ECM Enterprise Content Management).**

La automatización de los procesos de negocio y la información asociada permite el trabajo por excepción y facilita la toma de decisiones, reduciendo el tiempo de respuesta e incrementando la productividad.

El volumen de información no estructurada ha crecido exponencialmente en los últimos tiempos y esta diversidad ha hecho necesaria una evolución en los sistemas de Gestión de Contenidos. Su utilización se ha hecho imprescindible para evitar el caos en los sistemas de información.

Un sistema de Gestión de Contenidos totalmente integrado en el funcionamiento de la empresa permite gestionar el ciclo de vida completo de la información, crear y compartir toda esa información es la base imprescindible para la gestión de la empresa en tiempo real.

Todo ello ha servido de impulso para una rápida evolución de toda una gama de aplicaciones especializadas. En este contexto los fabricantes de soluciones de software de Gestión documental (DM), Gestión de contenidos WEB (WCM), Gestión de procesos (BPM) e Integración de aplicaciones (EAI) están haciendo crecer y evolucionar sus productos a un ritmo elevado, a pesar de la difícil situación del mercado.

### **1.8.1 ¿Qué es ECM?**

Un sistema ECM debe permitir la gestión de contenidos y procesos, al tiempo que facilita la conectividad con el resto de sistemas de la empresa.

La gestión de contenidos empresarial es (Enterprise Content Management – ECM) hace que los conocimientos de una organización estén disponibles para todo aquel que tenga derecho a usarlos, cuando y donde los necesite. Esto mejora mucho la eficacia y permite a las empresas cumplir con el creciente número de obligaciones legales y reglamentarias.

Un sistema para organizar y facilitar la creación conjunta de documentos y otros contenidos. Posibilita monitorizar contenidos a través de su ciclo de vida, controlando los cambios del contenido y los usuarios del contenido. En la mayoría de los casos, los sistemas de gestión de contenidos requieren un software de cliente especial para editar y construir artículos.

#### **1.8.1.1 Gestor de Contenido Empresarial Alfresco.**

Con un precio demasiado alto, la gestión de contenido empresarial excesivamente compleja se ha convertido en un objeto de consumo generalizado. Los usuarios quieren la simplicidad y familiaridad de las herramientas que se utilizan día tras día con unidades compartidas.



Gestión documental Alfresco captura, comparte y retiene contenido, permitiendo a los usuarios versionar, buscar y crear de forma sencilla sus propias aplicaciones de contenido, todo ello con las herramientas utilizadas hoy en día.

Alfresco es la alternativa principal del código abierto para la gestión del contenido empresarial. El modelo del código abierto permite a Alfresco utilizar las tecnologías y las contribuciones de la comunidad del código abierto para conseguir un software de alta calidad produciéndolo más rápidamente con un costo mucho más bajo.

Los beneficios de usar Alfresco son disímiles, debido a que es fácil de usar, productividad del desarrollador y del administrador, es la mejor práctica de colaboración, consta con un gestor de búsqueda avanzado y tiene una arquitectura distribuida.

Alfresco se integra con Microsoft Office y Open Office. Por lo tanto, puede seguir trabajando en su entorno de oficina sin necesitar formación adicional. Simplemente obtiene lo que usted y su empresa quieren: la posibilidad de guardar, versionar, compartir, buscar y auditar con una integración sencilla del flujo de trabajo.

Gestión documental Alfresco es una décima parte de un costo de un gestor documental tradicional, es creado sobre estándares de código abierto, está basado en una arquitectura de Java y es el ECM más escalable.(13)

#### **1.8.1.1.1 Servicios de red de Alfresco para el cliente.**

Alfresco ofrece un rango de ayuda completa ofreciendo que pueda ser adaptada a unas necesidades específicas. Alfresco cree que un servicio de apoyo total abarca el ofrecimiento de:

- Certificados estables del producto.

- Pruebas certificadas de escalabilidad.
- Un modelo controlado del lanzamiento.
- Mantenimiento y actualizaciones.
- Resolución de problemas.
- Soporte de actualizaciones.
- Consejo de Configuración.

#### **1.8.1.1.2 Características.**

\* Autenticación y Seguridad:

- JAAS, LDAP, Active Directory, Kerberos.

\* Compatibilidad:

- Linux, MacOS, Unix, Microsoft Windows.
- Base de datos que soporte Hibernate.
- Servidor de Aplicaciones Web sobre jdk 5/6 (Tomcat, Jboss, WebSphere, WebLogic, etc).
- Cliente Web sobre cualquier navegador (Firefox, Internet Explorer).
- Se puede añadir cualquier funcionalidad a través de Spring.

\* Organiza y facilita la gestión de contenidos de todo tipo.

- Documentos ofimáticos, presentaciones, imágenes, XML, multimedia etc.

\* Gestión del ciclo de vida de los contenidos.

- Crear, compartir, versionar, aprobar, publicar, retirar, archivar.

\* Facilita el trabajo colaborativo.

- Foros, Notificaciones, RSS, Blogs, Wiki, 'Social Computing'

\* Provee un repositorio fuente basado en últimas tecnologías y estándares.

- Altamente escalable, disponible y extensible.

Alfresco cuenta con varias aplicaciones frontales tales como DoCASU y Opsoro, que proveen un interfaz bastante amigable, ambas aplicaciones utilizan una librería Java Script para crear las interfaces y gestionar los Web Scripts conocida como ExtJs. Sin embargo estas aplicaciones frontales son muy sencillas de usar gracias a lo simple que son y no cumplen con las exigencias propuestas para nuestro trabajo de diploma, por lo tanto con la ayuda de la API de PHP que trae integrado Alfresco y los Web scripts damos una solución necesaria para la realización de nuestra aplicación.

### 1.8.2 Framework Code Igniter.

Code Igniter es un Framework para el desarrollo de aplicaciones en PHP. Es de código abierto, muy pequeño y posee un acceso a sus librerías bien estructurado.

Code Igniter usa el patrón de diseño Modelo-Vista-Controlador como paradigma de arquitectura de desarrollo, la cual separa en 3 capas distintas: la representación de datos, el interfaz de usuario y el controlador de eventos respectivamente.(14)

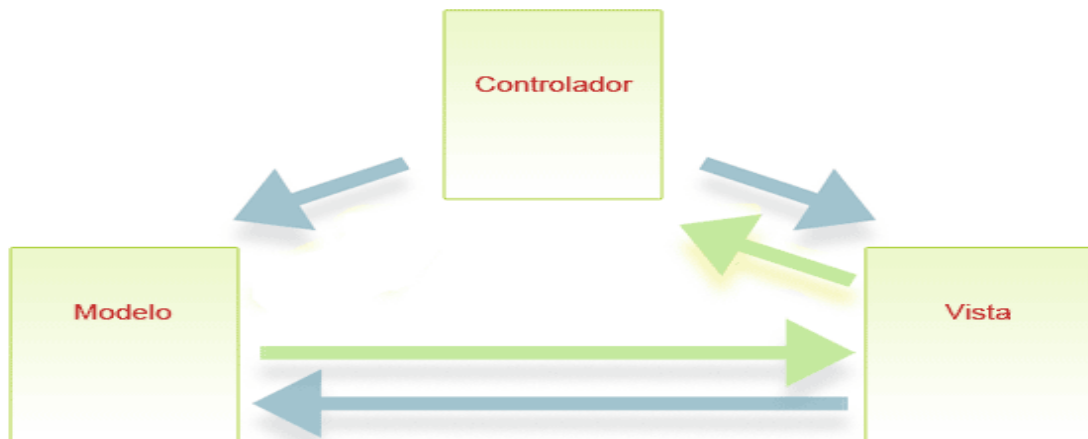


#### 1.8.2.1 Razones por la cual elegir Code Igniter.

- **Performance:** Framework Code Igniter tiene muy buena performance en comparación con Zend o CakePHP.
- **Documentación:** Tiene una muy buena documentación en línea, además de una comunidad de usuarios muy activos.
- **MVC:** Code Igniter usa el patrón de diseño Modelo-Vista-Controlador como paradigma de arquitectura de desarrollo, la cual separa en 3 capas distintas: la representación de datos, el interfaz de usuario y el controlador de eventos respectivamente. Hay una clara separación entre los componentes de un programa; lo cual nos permite implementarlos por separado. La



conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación.



- **Facilidad de uso:** La curva de aprendizaje es sumamente corta. Perfectamente se puede comenzar a utilizar las características habiendo leído el tutorial de Code Igniter o las guías online.
- **Active Record:** El manejo y abstracción de base de datos es realmente bueno, e implementa el patrón Active Record.
- **Estructura organizada:** La estructura de archivos es muy clara y ofrece gran flexibilidad en caso que necesite cambiarse por algún motivo. La división en models, views, controllers, libraries, helpers, config, etc. es muy práctica y comprensible.
- **Ruteo:** Hoy en día la optimización para buscadores es fundamental en algunos casos. Los mecanismos de ruteo de Code Igniter favorecen a éste y otras aplicaciones.

### 1.8.3 ¿Qué es SDK?

SDK, kit de desarrollo de software, es generalmente un conjunto de herramientas de desarrollo que le permite a un programador crear aplicaciones para un sistema concreto.

Es algo tan sencillo como una interfaz de programación de aplicaciones o API (del inglés application programming interface) creada para permitir el uso de cierto lenguaje de programación, o puede, también, incluir hardware sofisticado para comunicarse con un determinado sistema embebido. Las

herramientas más comunes incluyen soporte para la detección de errores de programación como un y otras utilidades. Los SDKs frecuentemente incluyen, también, códigos de ejemplo y notas técnicas de soporte u otra documentación de soporte para ayudar a clarificar ciertos puntos del material de referencia primario.

PHP SDK proporciona las siguientes capacidades:

- Una librería PHP que permite el acceso remoto al repositorio de Alfresco, Alfresco PHP a través de la API.
- Integración de populares aplicaciones PHP.
- Plantillas de PHP y capacidades de scripting en el repositorio utilizando la API de Alfresco PHP.
- Soporte para ejecutar aplicaciones PHP en el repositorio de JVM, con acceso a la API de Alfresco PHP.

### **1.8.3.1 ¿Qué es una API?**

API es la abreviatura de **A**plication **P**rogramming **I**nterface. Conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación. Cuando se intenta estandarizar una plataforma, se estipulan unos APIs comunes a los que deben ajustarse todos los desarrolladores de aplicaciones. Un API no es más que una serie de servicios o funciones que el Sistema Operativo ofrece al programador. Visto desde la perspectiva del código máquina, el API aparece como una serie de llamadas (en otros sistemas operativos se hace mediante saltos a supervisor; en OS/2 se implementan como Far Calls), mientras que si lo vemos desde la de un lenguaje de alto nivel, el API aparece como un conjunto de procedimientos y funciones.(15)

### **1.8.3.2 API de PHP**

Para desarrollos basados en la web. Para su implementación se tomo como base la API Java. De esa herencia se implementaron los eventos.

En la API Simple solo se configura el parámetro customized sender -setCustomizedSender ('CS'). Para quitarlo debes de establecer el valor a vacío.

La API Simple no reconoce los comandos DELIVER, INCOMINGMO, CHECKER y ACUSE. Esto significa que si se recibe uno de estos comandos los ignorara y no realiza ninguna acción. Esto es

interesante, porque si enviara un ACK de un INCOMING ese usuario nunca sabría de ese mensaje recibido, lo mismo para los DELIVER, CHEKER y ACUSE. Esto significa que ignora los comandos enviados por el servidor y no los trata. Para la API completa sí que están activas las respuestas para estos comandos. Para configurar que tipo de comandos se debe de tratar la aplicación PHP teniendo la clase Protocol Properties. Por defecto están desactivados, de esta manera se ahorra en espacio de memoria.

Otra gran diferencia es que la API simple no mantiene en memoria los comandos enviados ni recibidos. Mientras que la completa mantiene 4 arreglos en memoria.

#### **1.8.3.2.1 Inconvenientes de la API PHP**

Requiere el puerto 2048 y muchos usuarios tienen sus webs en servidores compartidos que no abren ese puerto. Para estos casos se implemento a modo de ejemplo los ejemplos:

- Sample-php-curl-xml
- Sample-php-xml

Estos ejemplos usan la API HTTP XML.

### **1.9 Estado del Arte de los Sistemas Implementados.**

Analizando e investigando sistemas que cumplan conjuntos de normas, técnicas y prácticas usadas para administrar el flujo de documentos de todo tipo en una organización, que permitan la recuperación de información desde ellos, determinen el tiempo que los documentos deben guardarse, eliminen los que ya no sirven y poder asegurar la conservación indefinida de los documentos más valiosos, aplicando principios de racionalización y economía, hacemos aquí una descripción de los mismos.

#### **1.9.1 Descripción de los Sistemas.**

**En el Mundo:**

##### **1.9.1.1 Módulo gmv**

Existe en la Comunidad de CMS un módulo que permite la integración de Drupal con Alfresco el cual ha sido denominado Gmv autor Sergio Marín. La integración de ambos conlleva grandes posibilidades,

ventajas y ahorro de trabajo. ¿Por qué preocuparse de que Drupal gestione los archivos que deseamos publicar en sitios web si para eso ya están los Gestores Documentales? La integración entre ambas soluciones se realiza usando HTTP-SOAP. Una prueba más de que la orientación a servicios flexibiliza muchísimo la interoperabilidad entre distintas soluciones. Es posible referenciar los documentos desde otro tipo de contenido dado que el módulo principal proporciona un nuevo tipo de contenido perfectamente referenciable.(16)



#### 1.9.1.2 DoCASU

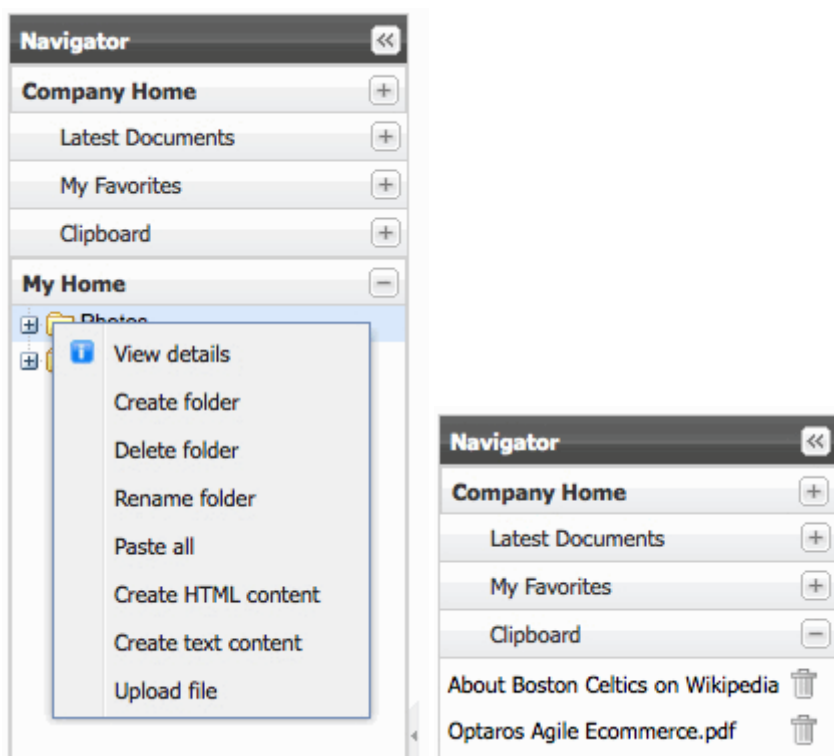
Un nuevo frontal para el gestor documental Alfresco fue diseñado por la empresa Optaros, este nuevo frontal llamado DoCASU, bajo licencia GPL, es muy sencillo de usar, gracias a lo simple que es. Esto una vez más pone de relieve algunas de las grandes ventajas del software de código abierto.

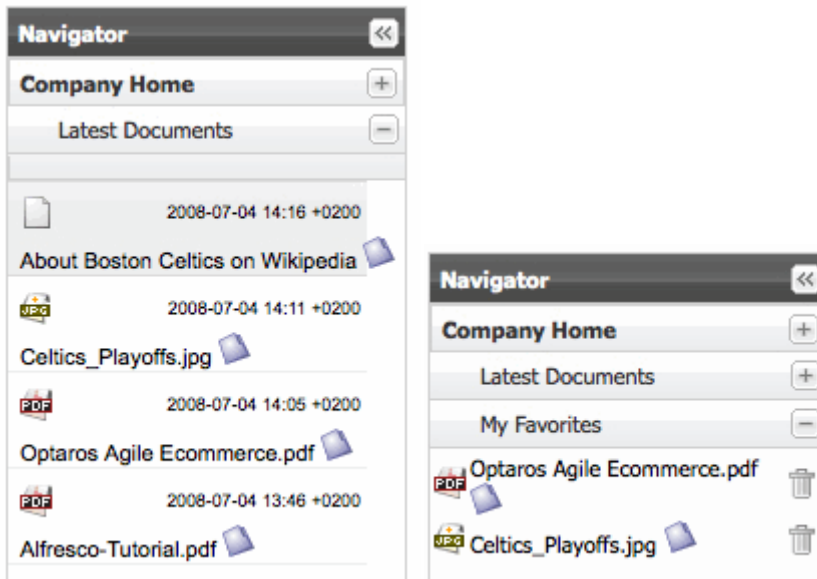
DoCASU es una interfaz de usuario para la plataforma del Gestor de Contenido Empresarial Alfresco. Con DoCASU los usuarios de Alfresco tienen la posibilidad de simplificar y facilitar las soluciones para acceder, agregar, buscar y modificar documentos. DoCASU puede extenderse y adaptarse para encontrar las necesidades específicas de los usuarios proporcionándoles funcionalidades para hacer fácil el trabajo con Alfresco.

Un software de fuente abierta tiene muchas ventajas frente a las alternativas. El hecho de que el código fuente está abierto ni siquiera podría ser el conductor más relevante. A menudo muchos más beneficios de las modernas tecnologías de código abierto es la naturaleza de sus componentes y la capacidad para acceder a las funciones y servicios a través de una API (interfaces de programación de aplicaciones). Esto permite centrar los esfuerzos de desarrollo en lo que es más importante para el uso de la aplicación.

DoCASU es una capa de presentación de la que se puede hacer en la parte superior de la página web de Alfresco con enfoque de comandos de estado de la técnica los resultados de las tecnologías. DoCASU los desarrolladores pueden tomar como punto de partida para una aún más amplia de interfaces de usuario, construido frontends o simplemente como la base para el aprendizaje de cómo construir en la parte superior de la Web Scripts. Optaros ha desplegado DoCASU 1,0 internamente y sigue utilizando en proyectos de cliente.(17)

Mapa de Navegación.





### 1.9.1.3 Módulo Joomla.

Alfresco Software y Joomla!tools han anunciado su primera integración basada en el estándar CMIS (Content Management Interoperability Services). El Módulo Joomla!™ para Alfresco, ha sido diseñado con CMIS REST API para que las organizaciones que utilicen sitios web en Joomla puedan acceder al potente archivo de gestión de contenido de código abierto de Alfresco.

Alfresco es la alternativa líder de código abierto para la gestión de contenido empresarial y aúna la innovación del código abierto con la estabilidad de una verdadera plataforma empresarial con unos costos mucho más económicos. Joomla es un sistema de gestión de contenido web de código abierto que ha sido premiado por su eficacia y permite a las organizaciones construir páginas web y diseñar potentes aplicaciones en línea. Joomla!tools ofrece servicios de asesoramiento y desarrollo de aplicaciones para Joomla.

Esta integración, diseñada con CMIS REST API, permitirá que millones de sitios web creados con Joomla tengan acceso a los servicios del enorme archivo de contenido final de Alfresco, y garantizará seguridad, cumplimiento y adaptabilidad. Los usuarios podrán gestionar, prever y trazar más efectivamente unos volúmenes mayores de contenido y bienes digitales en colaboración con páginas web Joomla utilizando la biblioteca de contenido de Alfresco. Asimismo, los usuarios de Alfresco podrán realizar búsquedas, publicar, compartir, descargar y editar contenido directamente de cualquier sitio Joomla.(18)



#### **1.9.1.4 CMS Opencms.**

Opencms es un Gestor de Contenidos Web (Content Management System) que permite la creación de tipos de contenido XML basados en esquemas (XSD), lo cual facilita la normalización de los procesos de definición de contenidos y su validación. De este modo se podrán definir los tipos de contenido que se publicarán en el Portal (Noticias, Eventos, Documentos, Enlaces, FAQs, etc).

El objetivo del gestor de contenidos consiste en facilitar la tarea de mantenimiento de los contenidos del portal, ocultando la complejidad y particularidades del proceso de publicación, de forma que en la medida de lo posible, los propios contribuidores de información sean quienes mantienen el entorno, sin asistencia técnica.(19)



#### **1. 10 Necesidad de un Sistema Gestor de Contenido.**

Se entiende por gestión documental el conjunto de normas, técnicas y prácticas usadas para administrar el flujo de documentos de todo tipo en una organización, permitir la recuperación de información desde ellos, determinar el tiempo que los documentos deben guardarse, eliminar los que ya no sirven y asegurar la conservación indefinida de los documentos más valiosos, aplicando principios de racionalización y economía.

Durante siglos, la gestión documental en las organizaciones fue el dominio exclusivo de administradores, archiveros y bibliotecarios, cuyas herramientas manuales básicas eran los libros de registro, las carpetas, archivadores, cajas y estanterías en que se guardan los documentos de papel (y más tarde los audiovisuales y los documentos en soportes magnéticos u ópticos), los ficheros que permiten hacer referencias cruzadas y una larga lista de técnicas de recuperación de información mediante sistemas de codificación y clasificación. Más recientemente se fueron sumando a ellos los informáticos, que son cada vez más necesarios debido a la complejidad y nivel de sofisticación que van alcanzando los sistemas computacionales de apoyo de la actividad administrativa.

Sin embargo todavía los gestores de contenido empresariales y en nuestro ejemplo Alfresco no cuenta con una capa de presentación amigable para el usuario, por esto nuestro trabajo de diploma se ha enmarcado la necesidad de implementar una interfaz fácilmente configurable para una adecuada Gestión Documental y Archivística.

### **1.11 ¿Qué ventajas posibilitaría una capa de presentación configurable para un Sistema de Gestión Empresarial?**

La capa de presentación es una parte de vital importancia en una aplicación, ya que, si su arquitectura no es la adecuada, puede incrementar la complejidad de la aplicación, no favorecer su flexibilidad y hacer que la experiencia del usuario sea poco eficaz y frustrante.

La capa de presentación de la mayoría de aplicaciones resulta con mucha frecuencia esencial para su éxito. Después de todo, representa la interfaz entre el usuario y el resto de la aplicación. Es, por así decirlo, el aspecto fundamental. Si el usuario no puede interactuar con la aplicación de forma que le permita realizar su trabajo con eficacia, el éxito global de la misma se verá seriamente perjudicado.

Por lo tanto nuestro trabajo se especializa en que el usuario final pueda interactuar fácilmente con la aplicación, la cual debe ser amigable para una adecuada Gestión Documental y Archivística.

### **1.12 Propuesta de solución al problema de la investigación.**

Al inicio existían dos vías de solución para la realización de la interfaz, que eran desarrollar sobre una aplicación ya existente llamada DOCASU, la cual hace uso de la API Javascript del Alfresco, a través de la librería Javascript Ext, esta aplicación realiza todas las operaciones e intercambio de datos mediante peticiones URL usando los métodos POST, GET y DELETE principalmente, esta vía de



solución gana en accesibilidad y confort al usuario pero no es muy segura, ya que los datos enviados por la URL pueden ser intervenidos por un usuario en la red usando un sniffer o cualquier herramienta para ello. La API de Javascript no es muy amplia y hay funcionalidades que no tiene bien definidas, lo que dificulta el trabajo con el repositorio de Alfresco.

La otra vía de solución y la más factible fue utilizando la SDK que provee Alfresco que está escrita en varios lenguajes, Java, Ruby, PHP entre otros, debido al amplio conocimiento que existe en la Universidad sobre el lenguaje PHP y la facilidad del mismo para su aprendizaje, se acordó el uso de la PHP SDK la cual presenta un amplio desarrollo y permite el control y gestión total sobre el repositorio de Alfresco, entonces se desarrollara la interfaz utilizando esta SDK por las facilidades y ventajas que posee sobre la API de Javascript.

Para la documentación del sistema se utilizó la Metodología RUP debido a que es uno de los procesos más generales de los existentes, se puede adaptar a cualquier proyecto, divide el proceso de desarrollo en ciclos, obteniendo una versión del software al final de cada ciclo que se divide en fases que finalizan con un hito donde se debe tomar una decisión importante

En este capítulo se caracterizaron las tendencias actuales en el mundo de la informática en el tema de los sitios Web y cuáles son las soluciones que se ajustan al objeto de estudio del trabajo. Además se analizó los lenguajes más utilizados para resolver el problema que se plantea, así como los Gestores de Contenidos Empresariales y aplicaciones en el mundo que pueden interactuar con ellos. Se caracterizaron algunos sistemas gestores de base de datos, se hace alusión a las tecnologías del lado del cliente y los procesos de desarrollo de software más conocidos actualmente.

## **CAPÍTULO 2: PROPUESTA DEL SISTEMA.**

En este capítulo entre las actividades fundamentales que se desarrollan se encuentra la descripción detallada del proceso que se encuentra involucrado en el modelamiento del problema, que es necesario para la implementación de una interfaz web. De dicho proceso se realiza una representación mediante una herramienta, así como una descripción dada por pasos lógicos, la cual posibilita comprender y organizar el proceso, permitiendo dar una visión hacia el sistema que se desea implementar.

En este capítulo se realiza una propuesta del sistema que se quiere implementar, luego de haber analizado otros sistemas. Se lleva a cabo la captura de los requisitos no funcionales y funcionales, dando el punto de partida para empezar a desarrollar el sistema con las características y funcionalidades requeridas, por otra parte se realiza la descripción y representación de los casos de uso del negocio y del sistema.

### **2.1 Problema.**

En la actualidad, coexisten en el mundo los más diversos sistemas de gestión documental: desde el simple registro manual de la correspondencia que entra y sale, hasta los más sofisticados sistemas informáticos. Por lo que es necesario que las empresas opten por las facilidades que brindan estos sistemas de integración y con ello adaptarlos a las especificidades y necesidades de cada institución, entendiendo así cuestiones como la forma de visualizarse los contenidos y las funcionalidades.

Sin embargo los Sistemas de Gestión Empresarial y en particular Alfresco no cuenta con una capa de presentación amigable para el usuario, debido a esta problemática el proyecto Gestión Documental y Archivística se ha planteado como objetivo brindar la posibilidad de consumir los servicios que brinda el mismo en una interfaz implementada.

### **2.2 Información que se maneja.**

La información que se maneja en el sistema es cualquier tipo de contenido que la institución necesite gestionar.

### **2.3 Propuesta del Sistema.**

Atendiendo a las necesidades del proyecto Gestión Documental y Archivística se propone desarrollar una interfaz amigable para la interactividad del usuario con la aplicación. Por lo cual se hace necesario el desarrollo de una capa de presentación que consuma todos los servicios que brinda el Gestor de Contenido Empresarial Alfresco. Fue utilizada la SDK que provee Alfresco en el lenguaje PHP. Debido a que PHP SDK presenta un amplio desarrollo y permite el control y gestión total sobre el repositorio de Alfresco.

### **2.4 Modelo de Dominio.**

Después de haber realizado un análisis profundo del problema en cuestión, se determinó que en la presente investigación no se definen concretamente todos los procesos, por lo que se decide dar un nuevo enfoque a todo el proceso. Para ello se debe realizar un modelo de dominio. Este modelo contribuye posteriormente a identificar algunas clases que se utilizarán en el sistema.

Es necesario tener un vasto conocimiento de cómo debe funcionar el proceso en cuestión, para poder capturar correctamente los requisitos y así poder construir un sistema con las características que el cliente desee. Éste modelo va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema.

‘El Modelo de Dominio (o Modelo Conceptual) es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema. Representa conceptos del mundo real, no de los componentes de software.’  
(20)

#### **2.4.1 Entidades y Conceptos.**

**Gestores de Contenidos Empresariales:** es una de las estrategias y tecnología empleadas en la industria de la tecnología de la información para manejar la captura, almacenamiento, seguridad, control de versiones, recuperación, distribución, conservación y destrucción de documentos y contenidos.

**Aplicación:** En informática, una aplicación es un tipo de programa informático diseñado para facilitar al usuario la realización de un determinado tipo de trabajo.

**Contenido:** Información disponible en una página. Se incluyen las imágenes, textos y cualquier otro tipo de información proporcionada independientemente de su formato.

**Carpeta o Espacio:** Directorio: en informática, lugar en el que se agrupa información.

**Reglas:** Se utilizan para validar cada contenido y así gestionarlo correctamente de acuerdo con las características del sistema.

**Metadatos:** Los metadatos son datos asociados a un documento digital que recogen información fundamentalmente descriptiva (autor, título, etc.). También pueden incluir información de administración (creación del recurso, derechos, control de acceso...), y preservación (tipo de formato, etc.).

### 2.4.2 Representación del diagrama de clases Modelo de Dominio.

“Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de contención.” [1]

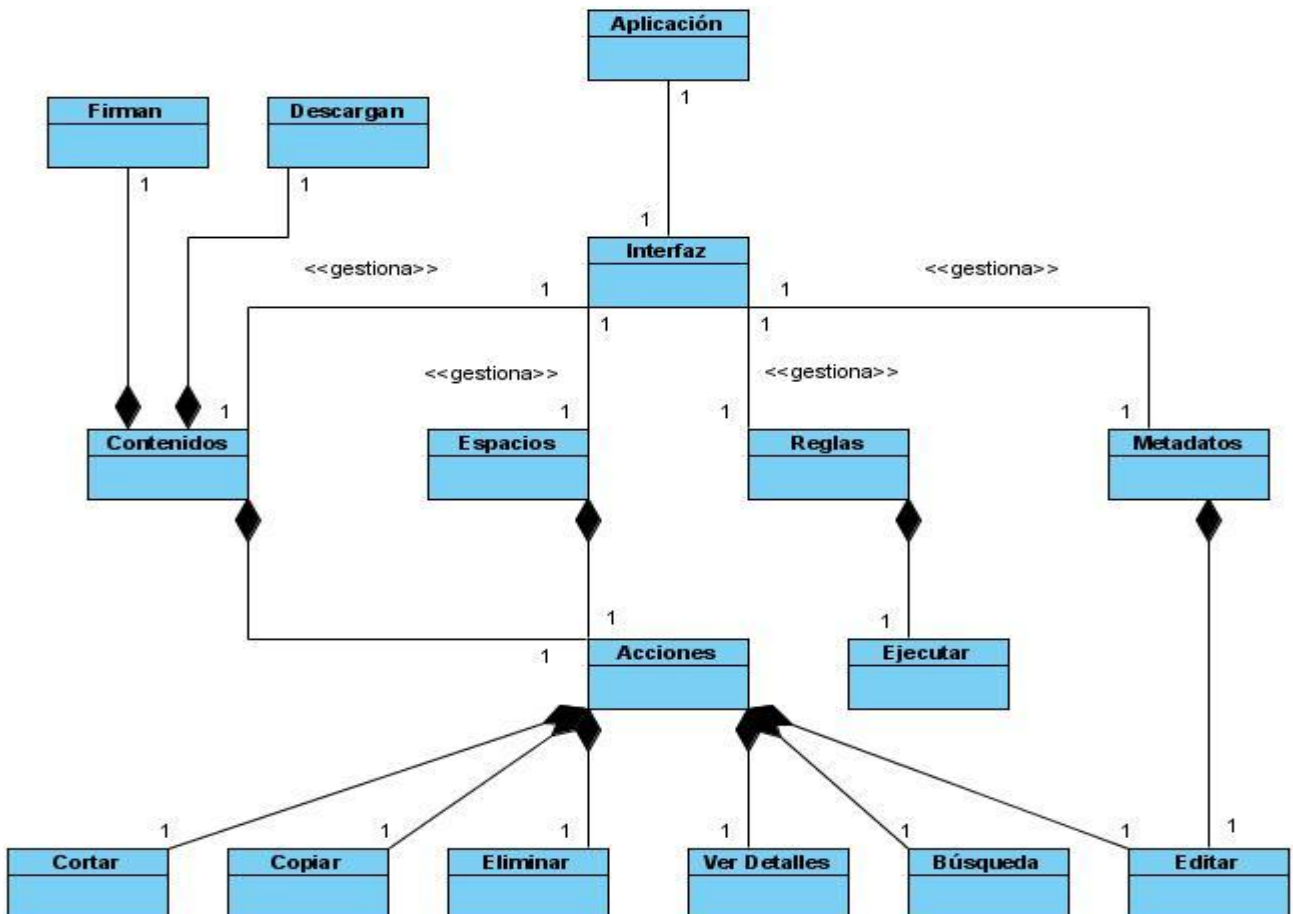


Fig 2.1 Diagrama General de Clases del Modelo del Dominio.

## **2.5 Levantamiento de Requerimiento.**

Es uno de los flujos de trabajo más importantes, porque en él se establece qué es lo que tiene que hacer exactamente el sistema que se construya. En esta línea los requisitos son el contrato que se debe cumplir, de modo que los usuarios finales tienen que comprender y aceptar los requisitos que se especifiquen. Se dividen en dos grupos: los requisitos funcionales y los requisitos no funcionales.

### **2.5.1 Requisitos Funcionales.**

Los requerimientos funcionales definen las funciones que el sistema será capaz de realizar.

RF 1 Autenticar Usuario.

RF 2 Gestionar Espacios.

RF 2.1 Cortar.

RF 2.2 Copiar.

RF 2.3 Ver detalles.

RF 2.4 Editar.

RF 2.5 Eliminar.

RF 2.6 Búsqueda.

RF 3 Gestionar Contenido.

RF 3.1 Cortar.

RF 3.2 Copiar.

RF 3.3 Ver detalles.

RF 3.4 Editar.

RF 3.5 Eliminar.

RF 3.6 Descargar.

RF 3.7 Búsqueda.

RF 4 Gestionar Metadatos.

RF 4.1 Editar.

RF 5 Gestionar Reglas

RF 5.1 Ejecuta

RF 6 Firmar Contenido.

### **2.5.2 Requisitos no Funcionales.**

“Los requerimientos no funcionales especifican propiedades o cualidades que el producto de software debe tener, como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, entre otras.” [2]

Entre los requerimientos no funcionales del sistema propuesto se encuentran:

#### **Usabilidad:**

- ✓ Para utilizar el sistema es necesario poseer conocimientos elementales de computación y sobre el ambiente Web en sentido general.
- ✓ El sistema podrá ser usado de forma fácil por cualquier persona que posea permisos en la institución para gestionar los documentos.

#### **Rendimiento:**

- ✓ Tiempos de respuestas rápidos, aproximadamente de 2 segundos, al igual que la velocidad de procesamiento de la información.

#### **Hardware:**

- ✓ Requiere como mínimo de RAM 512 MB.
- ✓ El disco duro no requiere gran cantidad de espacio.

#### **Software:**

- ✓ Requiere tener instalado Apache y activarle el modulo rewrite.

- ✓ Alfresco 3.0 y PHP 5.0.

**Portabilidad:**

- ✓ Necesidad de que el sistema sea multiplataforma.

**Seguridad:**

- ✓ Garantizar que la información sea editada únicamente por las personas que tienen permisos para realizar esta actividad.
- ✓ Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.

**Legales:**

- ✓ El empleo de este producto no debe violar ninguna ley o licencia por lo que la plataforma escogida para el desarrollo de la aplicación, está basada en la licencia GPLv2.

**Confiabilidad:**

- ✓ La información contenida en el sistema debe ser totalmente confiable.
- ✓ Deben implementarse mecanismos para garantizar la respuesta ante posibles fallos lo más rápido posible.

**Interfaz:**

- ✓ El producto debe ser legible y con colores adecuados, agradables y poco llamativos.
- ✓ Construcción de enlaces rápidos o anclas para los documentos muy largos.

**2.6 Definición de los Casos de Usos y Actores.**

## 2.6.1 Definición de los Actores del Sistema.

Actores del sistema	Justificación
Usuario	Persona con privilegios para autenticarse y formar parte de los usuarios del sistema, mantiene las operaciones básicas de navegación.
Autor	Persona con privilegios para autenticarse y formar parte de los usuarios del sistema, mantiene las operaciones básicas de navegación además de otras como crear espacios y contenidos.
Archivador	Persona encargada de controlar los contenidos.
Administrador	Persona encargada de monitorear el sistema, así como gestionar Documentos, espacios y otras funciones del sistema.

Tabla: Actores del Sistema.

## 2.6.2 Casos de Usos del Sistema.

<b>CU-1</b>	Autenticar Usuario.
<b>Actor</b>	Usuario
<b>Descripción</b>	El usuario se identifica en el sistema para recibir los permisos que se le hayan asignados.
<b>Referencia</b>	RF 1

<b>CU-2</b>	Gestionar Espacio.
<b>Actor</b>	Administrador.
<b>Descripción</b>	El administrador accede al sistema para cortar, copiar, ver detalles, editar, eliminar o buscar un espacio.
<b>Referencia</b>	RF-2

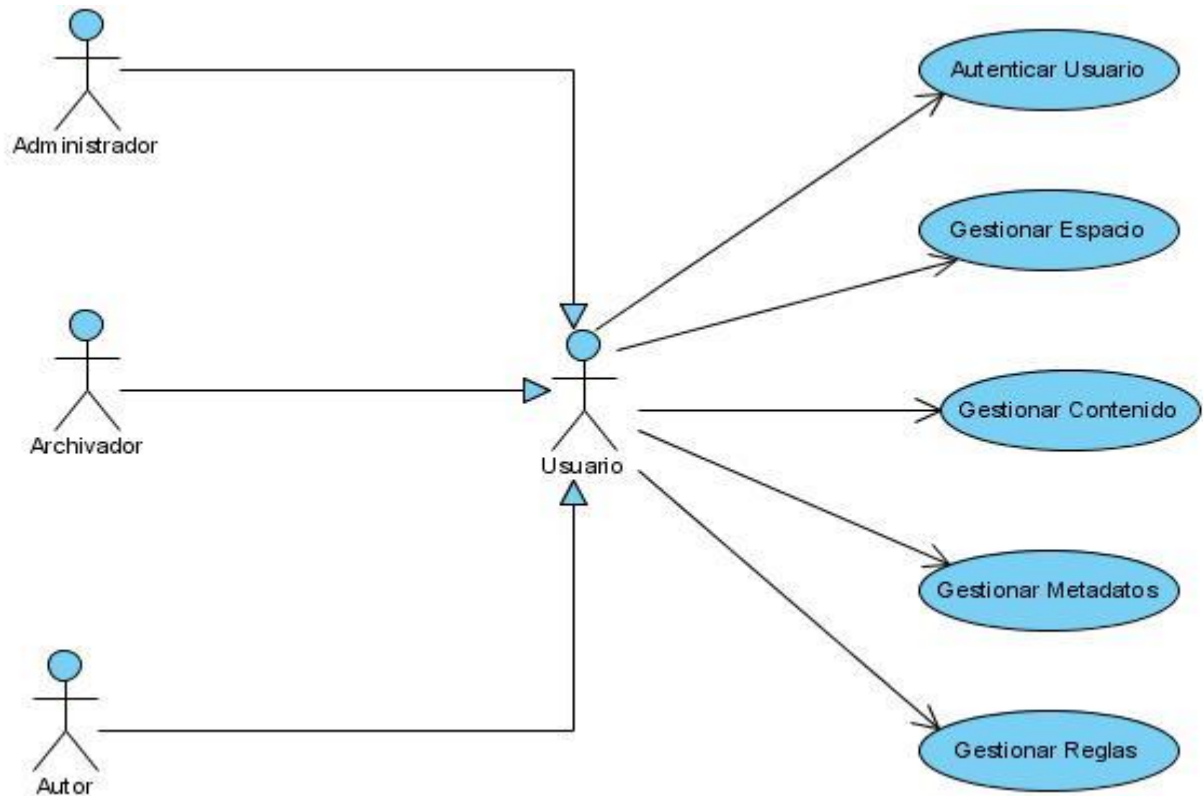


<b>CU-3</b>	Gestionar Contenido.
<b>Actor</b>	Administrador.
<b>Descripción</b>	El administrador accede al sistema para cortar, copiar, ver detalles, descargar, editar, firmar, buscar o eliminar un contenido.
<b>Referencia</b>	RF-3

<b>CU-4</b>	Gestionar Metadatos.
<b>Actor</b>	Usuario.
<b>Descripción</b>	El usuario accede al sistema para editar un metadato.
<b>Referencia</b>	RF-4

<b>CU-5</b>	Gestionar Reglas.
<b>Actor</b>	Usuario.
<b>Descripción</b>	El usuario accede al sistema para ejecutar una regla.
<b>Referencia</b>	RF-5

### 2.6.3 Diagrama de casos de Usos del Sistema.



2.6.4 Descripción de los Casos de Usos del Sistema.

Tabla: descripción Textual <CU Autenticar Usuario.>

<b>CU</b>	Autenticar Usuario	
<b>Propósito</b>	Comprobar las credenciales del usuario y autenticarlo en el sistema.	
<b>Actores</b>	Autor.	
<b>Resumen</b>	El Caso de Uso se inicia cuando el autor desea autenticarse en el sistema y finaliza cuando se autentica.	
<b>Referencia</b>	RF-1	
<b>Flujo Normal</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El autor accede al sistema insertando los datos de usuario y contraseña.	1.1 El sistema chequea al usuario.	
<b>Flujo Alternativo</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
	1.1 El sistema verifica que los datos sean correctos. 1.2 El sistema permite el acceso al usuario	

	asignándole sus privilegios.
--	------------------------------

Tabla: descripción Textual &lt;CU Gestionar Espacio.&gt;

<b>CU</b>	Gestionar espacio	
<b>Propósito</b>	El administrador accede al sistema y puede gestionar determinado espacio.	
<b>Actores</b>	administrador	
<b>Resumen</b>	El Caso de Uso se inicia cuando el administrador necesita cambiar o añadir algún espacio en el sistema.	
<b>Referencia</b>	RF-2	
<b>Flujo Normal</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
1. El administrador selecciona la opción crear carpeta.		1.1 El sistema muestra los datos necesarios que deben ser llenados para la creación del espacio.
<b>Sección-1 Cortar Espacio</b>		
<b>Flujo Normal.</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
1. El administrador selecciona la opción cortar el espacio que desee. 2. El administrador va al portapapeles y escoge el espacio cortado y lo pega en el lugar que desea.		1.1 El sistema corta el espacio seleccionado. 1.2 El espacio cortado se mueve al portapapeles. 2.1 El sistema mueve el nodo al lugar escogido, borrándolo del portapapeles y de su ubicación de origen.
<b>Flujo Alternativo.</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
		1.1 El sistema lanza una excepción en caso de ocurrir error.
<b>Sección-2 Copiar Espacio</b>		
<b>Flujo Normal.</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
1. El administrador selecciona la opción copiar el espacio que desee. 2. El administrador va al portapapeles y escoge el espacio copiado y o pega en el lugar que desee.		1.1 El sistema copia el espacio seleccionado. 1.2 El sistema manda el espacio copiado hacia el portapapeles. 2.1 El sistema copia el nodo del espacio al lugar escogido.
<b>Flujo Alternativo.</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
		1.1 El sistema lanza una excepción en caso de ocurrir

	error.
<b>Sección-3 Ver Detalles</b>	
<b>Flujo Normal.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador selecciona la opción ver detalles del espacio que desee.	1.1 El sistema muestra las propiedades de dicho espacio.
<b>Flujo Alternativo.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>Sección-4 Editar Espacio</b>	
<b>Flujo Normal.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<ol style="list-style-type: none"> <li>1. El administrador selecciona la opción editar el espacio que desee.</li> <li>2. El administrador entra los datos al sistema que deben ser cambiados</li> </ol>	<ol style="list-style-type: none"> <li>1.1 El sistema muestra los datos que deben ser editados.</li> <li>2.1 El sistema cambia la propiedad del nodo cambiándolo por los nuevos datos de entrada.</li> </ol>
<b>Flujo Alternativo.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1.1 El sistema lanza una excepción en caso de que no sean editados los datos correctamente.
<b>Sección-5 Eliminar Espacio</b>	
<b>Flujo Normal.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador selecciona la opción eliminar el espacio que desee.	1.1 El sistema busca el nodo del espacio seleccionado a eliminar y lo elimina completamente del repositorio.
<b>Flujo Alternativo.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>Sección- 6 Búsqueda Espacio</b>	
<b>Flujo Normal.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador selecciona la opción Buscar dando el nombre de lo que desea buscar.	1.1 El sistema muestra los espacios que contengan el nombre entrado por el usuario.
<b>Flujo Alternativo.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1.1 El sistema muestra un mensaje de error si la búsqueda no tuvo éxito.

Tabla: descripción Textual &lt;CU Gestionar Contenido.&gt;

<b>CU</b>	Gestionar Contenido	
<b>Propósito</b>	El administrador accede al sistema y puede gestionar determinado contenido.	
<b>Actores</b>	administrador	
<b>Resumen</b>	El Caso de Uso se inicia cuando el administrador necesita cambiar o añadir algún contenido en el sistema.	
<b>Referencia</b>	RF-3	
<b>Flujo Normal</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El administrador selecciona la opción crear o adicionar contenido.	1.1 El sistema muestra los datos necesarios que deben ser llenados para la crear o adicionar un contenido.	
<b>Sección-1 Cortar Contenido</b>		
<b>Flujo Normal.</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El administrador selecciona la opción cortar el contenido que desee. 2. El Administrador selecciona el contenido del cortapapeles y lo pega en el lugar que desee.	1.1 El sistema corta el contenido seleccionado. 1.2 El sistema manda el contenido cortado para el portapapeles. 2.1 El sistema mueve el nodo del contenido en el lugar seleccionado y lo elimina del portapapeles y de su ubicación de origen.	
<b>Flujo Alternativo.</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
	1.1 El sistema lanza una excepción en caso de ocurrir error.	
<b>Sección-2 Copiar Contenido</b>		
<b>Flujo Normal.</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El administrador selecciona la opción copiar el contenido que desee. 2. El administrador accede al portapapeles y pega el contenido en el lugar que desea.	1.1 El sistema copia el contenido seleccionado. 1.2 El sistema copia el contenido seleccionado en el portapapeles. 2.1 El sistema copia el nodo con la información del contenido	
<b>Flujo Alternativo.</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
	1.1 El sistema lanza una excepción en caso de ocurrir error.	
<b>Sección-3 Ver Detalles</b>		
<b>Flujo Normal.</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El administrador selecciona la opción	1.1 El sistema muestra las propiedades de dicho	

ver detalles del contenido que desee.	contenido.
<b>Flujo Alternativo.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>Sección-4 Editar Contenido</b>	
<b>Flujo Normal.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<ol style="list-style-type: none"> <li>1. El administrador selecciona la opción editar el contenido que desee.</li> <li>2. El administrador entra los datos en el los campos que desea editar.</li> </ol>	<ol style="list-style-type: none"> <li>1.1 El sistema muestra los datos que deben ser editados.</li> <li>2.1 El sistema cambia la propiedad del nodo al contenido que se está editando.</li> </ol>
<b>Flujo Alternativo.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1.1 El sistema lanza una excepción en caso de que no sean editados los datos correctamente.
<b>Sección-5 Eliminar Contenido</b>	
<b>Flujo Normal.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador selecciona la opción eliminar el contenido que desee.	<ol style="list-style-type: none"> <li>1.1 El sistema muestra el contenido a eliminar.</li> <li>1.2 El sistema elimina el contenido del repositorio.</li> </ol>
<b>Flujo Alternativo.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>Sección-6 Descargar Contenido</b>	
<b>Flujo Normal.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador selecciona la opción descargar el contenido que desee.	1.1 El sistema muestra el contenido a descargar abriéndolo.
<b>Flujo Alternativo.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
<b>Sección- 7 Búsqueda Contenido</b>	
<b>Flujo Normal.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El administrador selecciona la opción Buscar el contenido que desee.	1.1 El sistema muestra el contenido solicitado si existe.
<b>Flujo Alternativo.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1.1 El sistema muestra un mensaje de error si la

	búsqueda no tuvo éxito.
--	-------------------------

Tabla: descripción Textual &lt;CU Gestionar Metadatos.&gt;

<b>CU</b>	Gestionar Metadatos	
<b>Propósito</b>	El usuario accede al sistema y puede gestionar determinado metadatos.	
<b>Actores</b>	usuario	
<b>Resumen</b>	El Caso de Uso se inicia cuando el usuario necesita cambiar o añadir algún metadato en el sistema para un contenido.	
<b>Referencia</b>	RF-4	
<b>Flujo Normal</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
1. El usuario selecciona el contenido que quiere gestionarle algún metadato.		1.1 El sistema muestra los datos necesarios que deben ser llenados.
<b>Sección-1 Editar</b>		
<b>Flujo Normal.</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
1. El usuario selecciona la opción editar los metadatos a un tipo de contenido específico. 2. El administrador entra los datos del metadato que deben ser editados.		1.1 El sistema muestra los campos que deben ser llenados. 2.1 El sistema cambia la propiedad del nodo del metadato que se esté editando.
<b>Flujo Alternativo.</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
		1.1 El sistema lanza una excepción en caso de ocurrir error.

Tabla: descripción Textual &lt;CU Gestionar Reglas.&gt;

<b>CU</b>	Gestionar Reglas	
<b>Propósito</b>	El usuario accede al sistema y puede gestionar determinadas reglas.	
<b>Actores</b>	usuario	
<b>Resumen</b>	El Caso de Uso se inicia cuando el usuario necesita cambiar o añadir alguna regla en el sistema.	
<b>Referencia</b>	RF-5	
<b>Flujo Normal</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
1. El usuario selecciona la opción crear regla.		1.1 El sistema muestra los datos necesarios que deben ser llenados para la creación de la regla.
<b>Sección-1 Ejecutar Regla</b>		

<b>Flujo Normal.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El usuario selecciona la opción ejecutar la regla que desee.	1.1 El sistema ejecuta la regla seleccionada.
<b>Flujo Alternativo.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1.1 El sistema lanza una excepción en caso de ocurrir error.

En este capítulo se mostraron las principales clases del dominio para una mayor comprensión del negocio, además se justificó la selección de los actores y trabajadores que intervienen; se analizaron las características y funciones fundamentales del sistema para la creación y empaquetamiento de contenidos, los cuales se representaron mediante un Diagrama de Casos de Uso, llegando así a las descripciones detalladas de cada caso de uso del sistema. Una vez realizado esto es posible comenzar a realizar el diseño de la aplicación teniendo en cuenta los requerimientos especificados en el capítulo.



## **CAPÍTULO 3: DISEÑO DEL SISTEMA.**

En el presente capítulo se expone el diagrama de clases de diseño, el cual se decidió que fuese general, donde agrupara todas las clases que intervienen en la realización de los casos de usos pertenecientes al primer y segundo ciclo de desarrollo de nuestro sistema, con el objetivo de obtener un mejor entendimiento. Además se representan los diagramas de secuencia por escenario de cada caso de usos. Así como el diagrama general de estereotipos web del sistema.

### **3.1 Diseño.**

En la fase de diseño se modela el sistema de manera que soporte todos los requisitos, tanto funcionales como no funcionales, creándose así una entrada apropiada para las actividades de implementación. Lo principal de esta etapa es la elaboración de los diagramas de clases de diseño, donde se muestran las clases participantes en la ejecución de un caso de uso. Para un mayor entendimiento los autores han decidido distribuir las clases por paquetes de acuerdo a sus funcionalidades, y estos a su vez están compuestos por subpaquetes, esto se ha realizado con el objetivo de lograr un mayor entendimiento y organización de los diagramas.

En el **Paquete de Clases Controladoras** contiene las clases controladoras de las entidades relacionadas en el negocio, estas clases permiten diversas funciones como la validación de datos. Contiene las clases de **auth**, **content**, **offline** y **search**.

En el **Paquete de Clases de Librerías** encontramos las clases para hacer posible la persistencia y la búsqueda de objetos. El mismo está dividido en dos subpaquetes: el subpaquete **Librerías** y el **PHP API**.

El subpaquete **Librerías** contiene las clases **CController**, **RepositoryAccess** y **Access**.

### 3.2 Representación UML del diagrama de clases del Diseño.

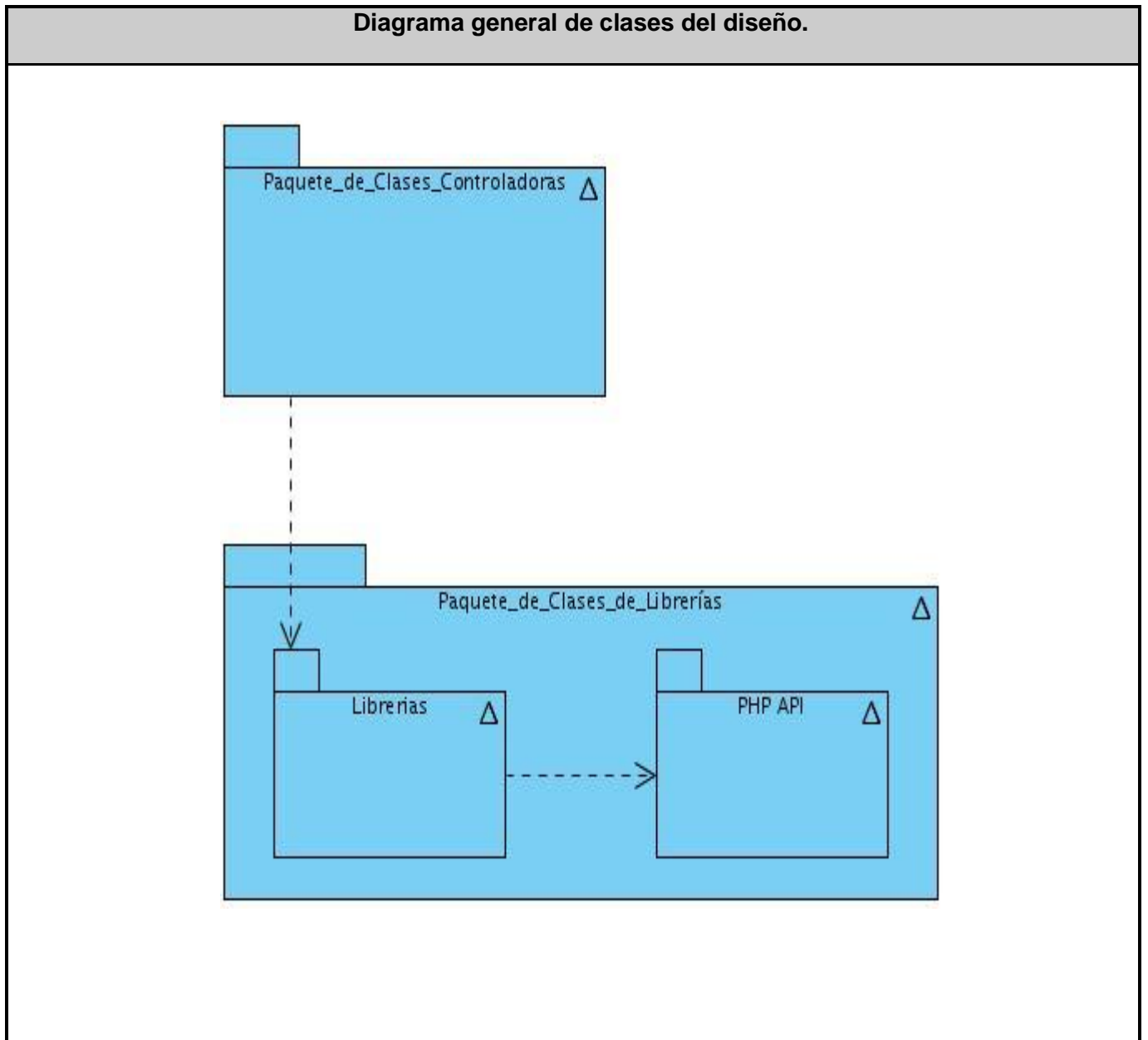


Fig 3.1 Diagrama General de Clases del Diseño.

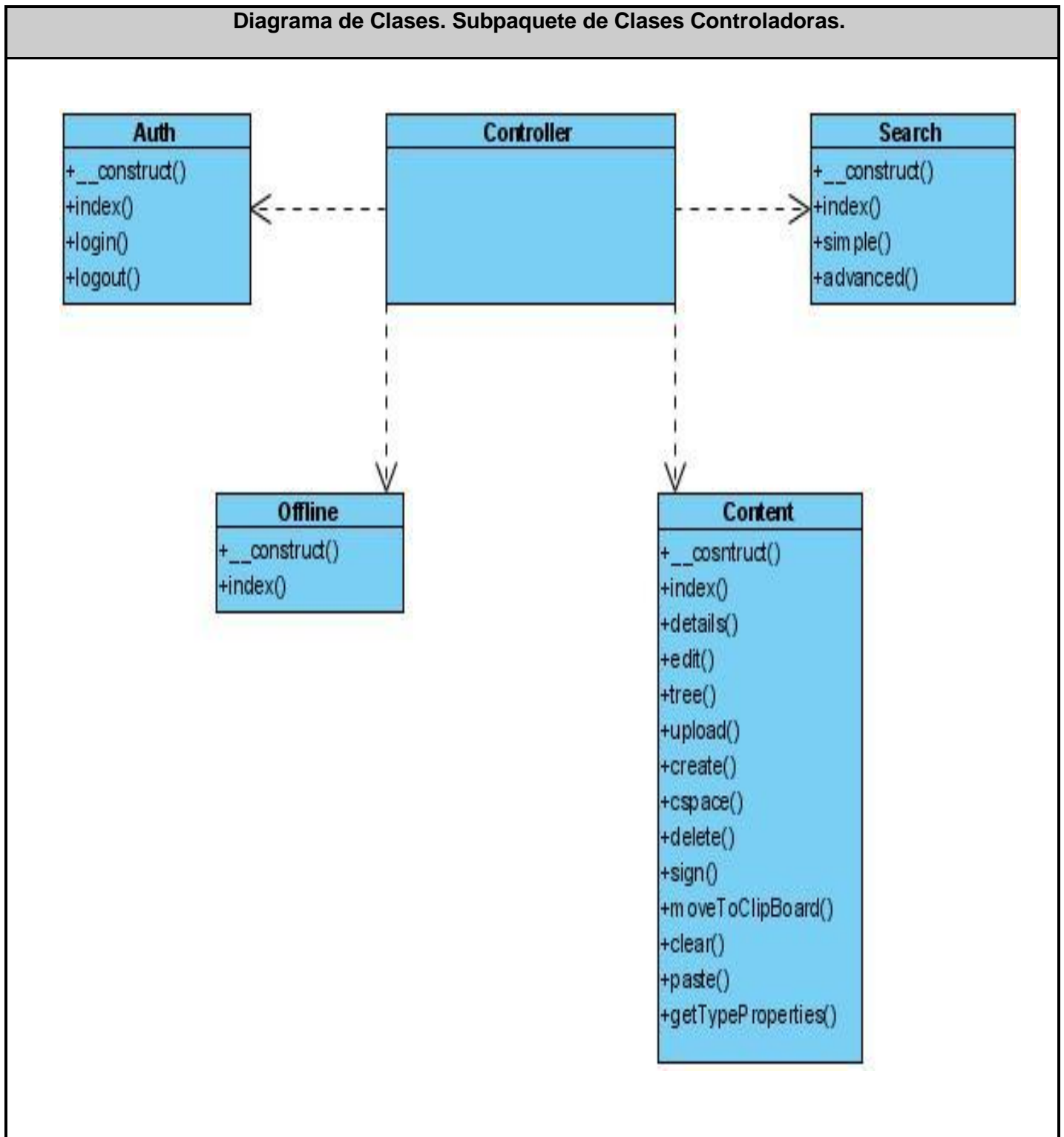


Fig 3.2 Diagrama de clases. Subpaquete de Clases Controladoras.

## Diagrama de Clases. Subpaquete de Clases de Librerías.

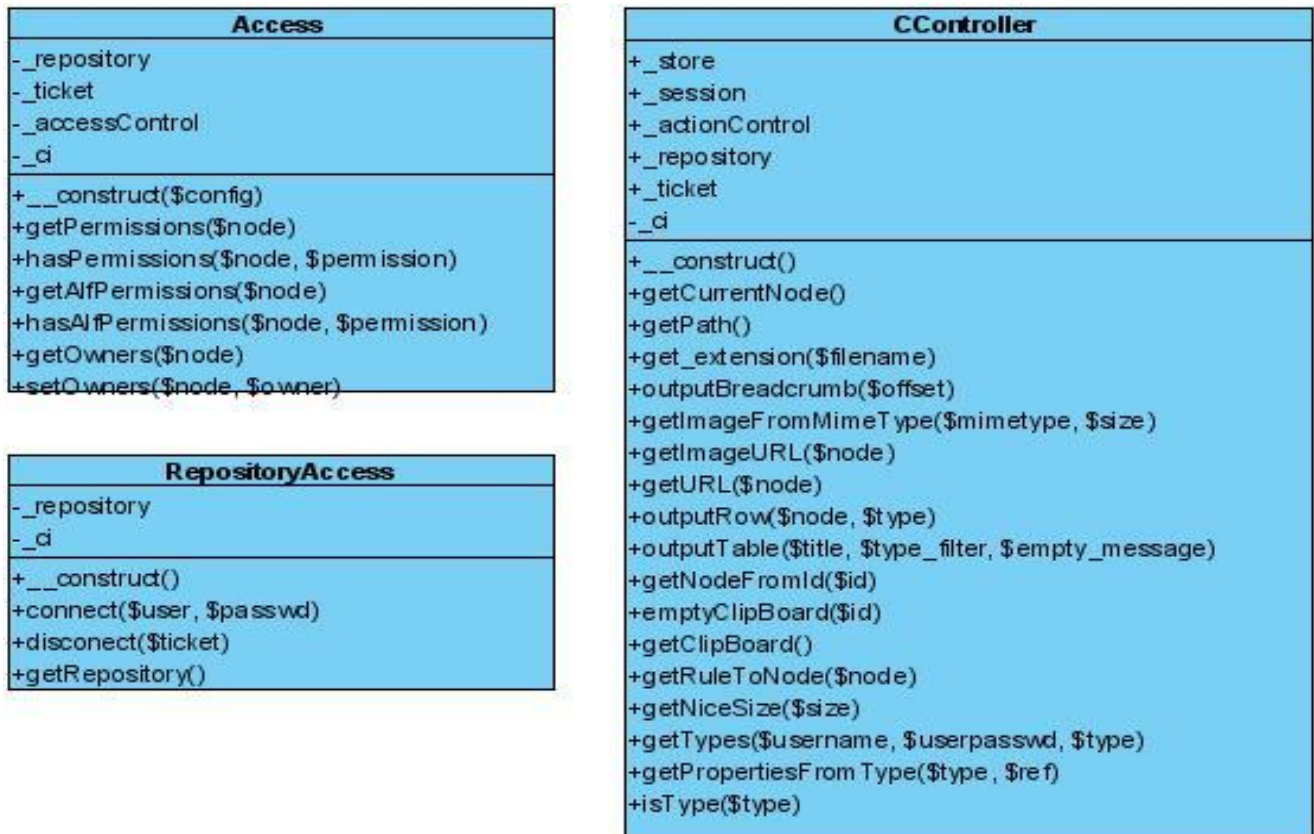


Fig 3.3 Diagrama de Clases. Subpaquete de Clases de Librerías.

Otro objetivo de esta etapa es la elaboración de los diagramas de interacción que muestran gráficamente cómo los objetos se comunican entre ellos con el objetivo de cumplir los requerimientos. Esta interacción se puede expresar en diagramas de colaboración y de secuencia. El diagrama de secuencia es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema. Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase. Este contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementarlo y mensajes pasados entre los objetos.

## 3.3 Representación UML de los diagramas de Estereotipos Web del Diseño.

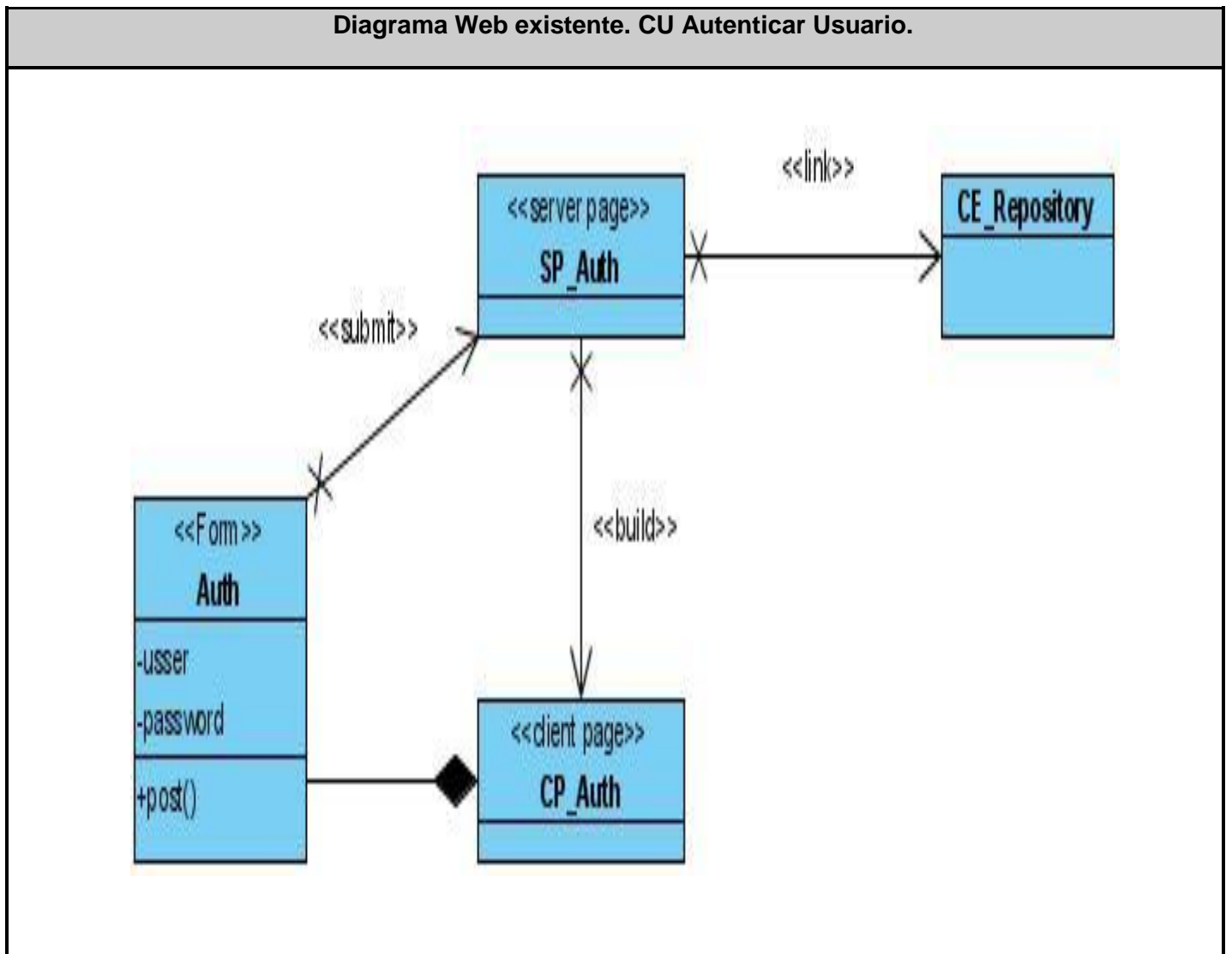


Fig 3.4 Diagrama de Estereotipos Web. CU Autenticar Usuario.

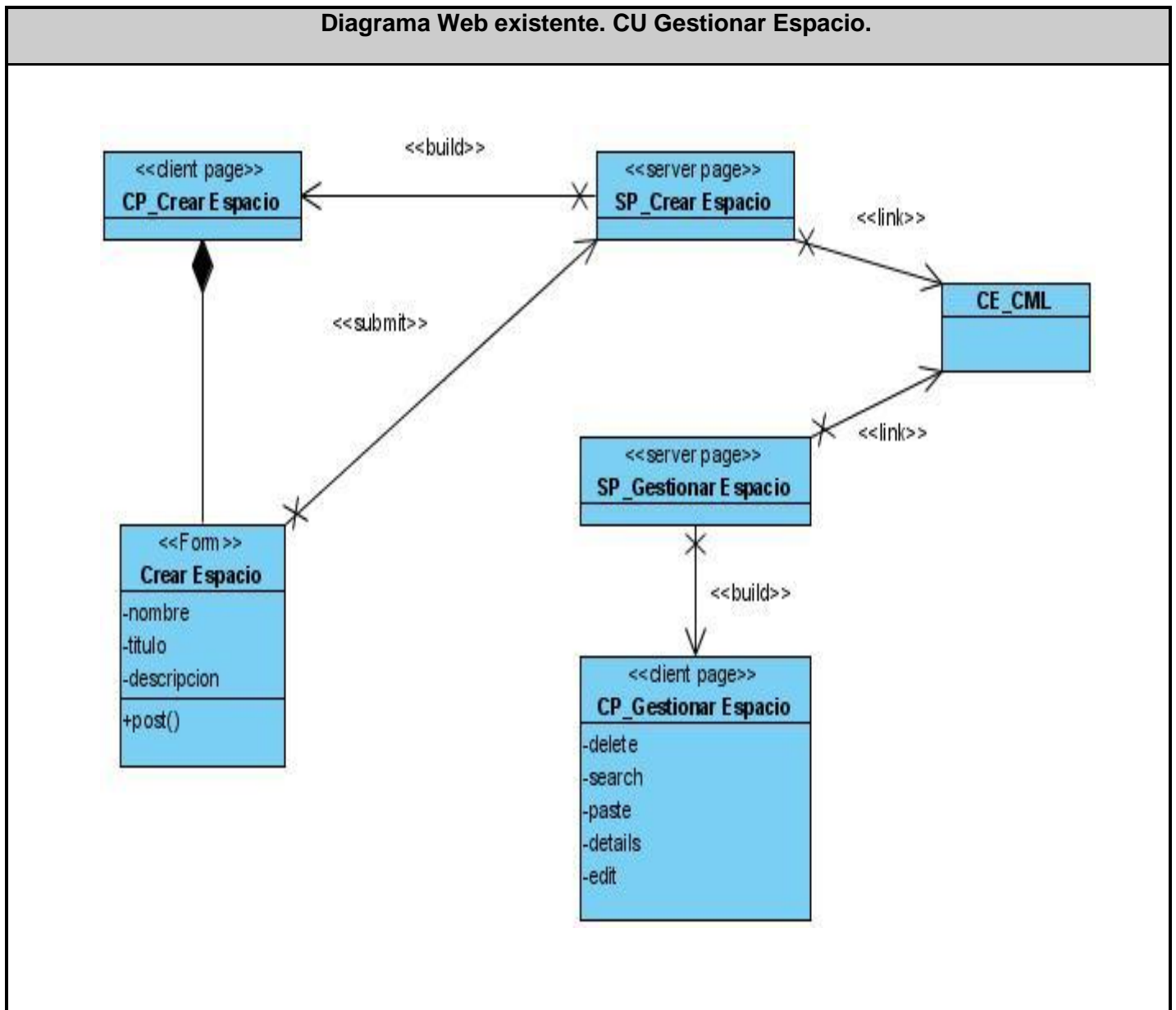


Fig 3.5 Diagrama de Estereotipos Web. CU Gestionar Espacio.

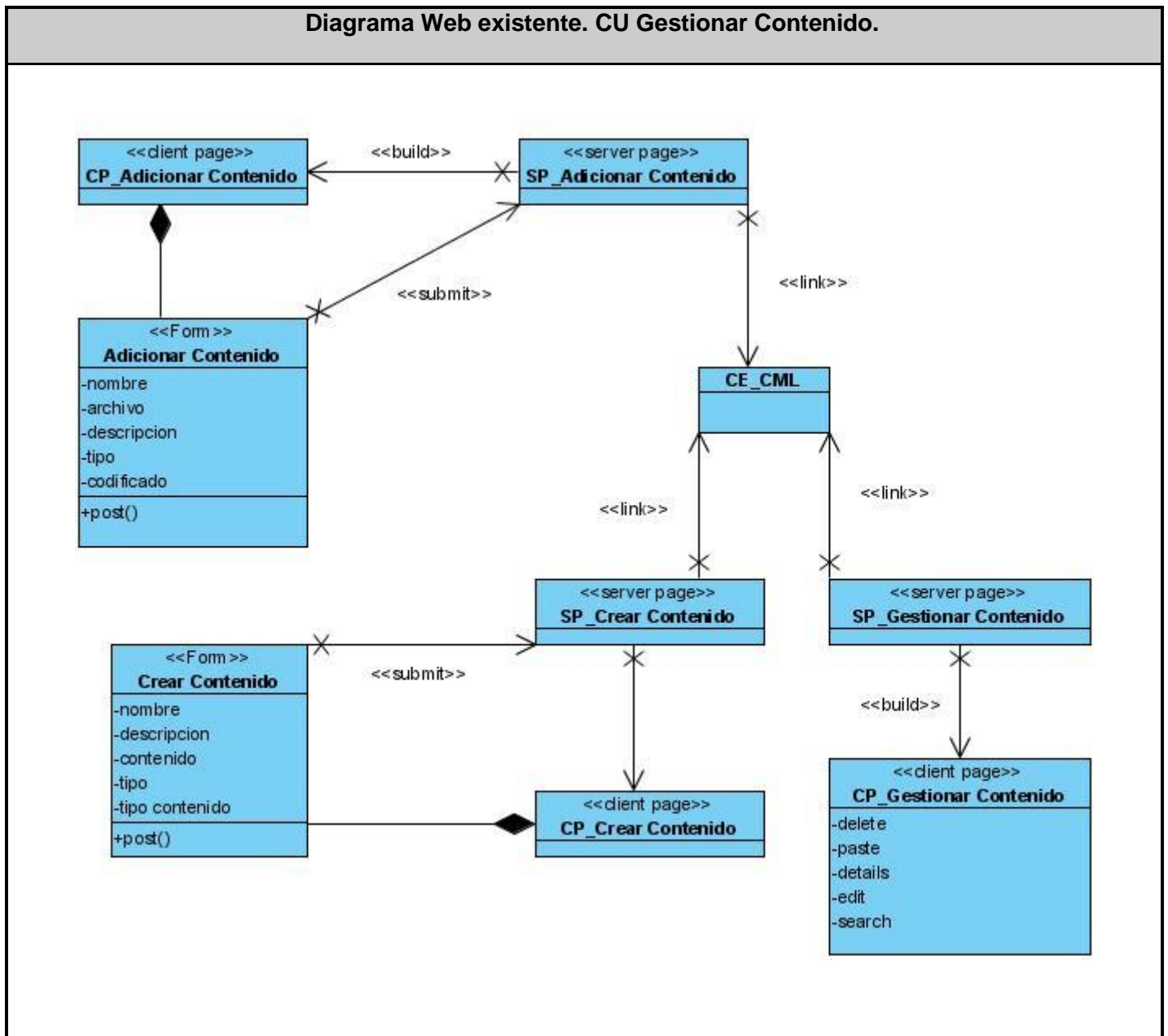


Fig 3.6 Diagrama de Estereotipos Web. CU Gestionar Contenido.

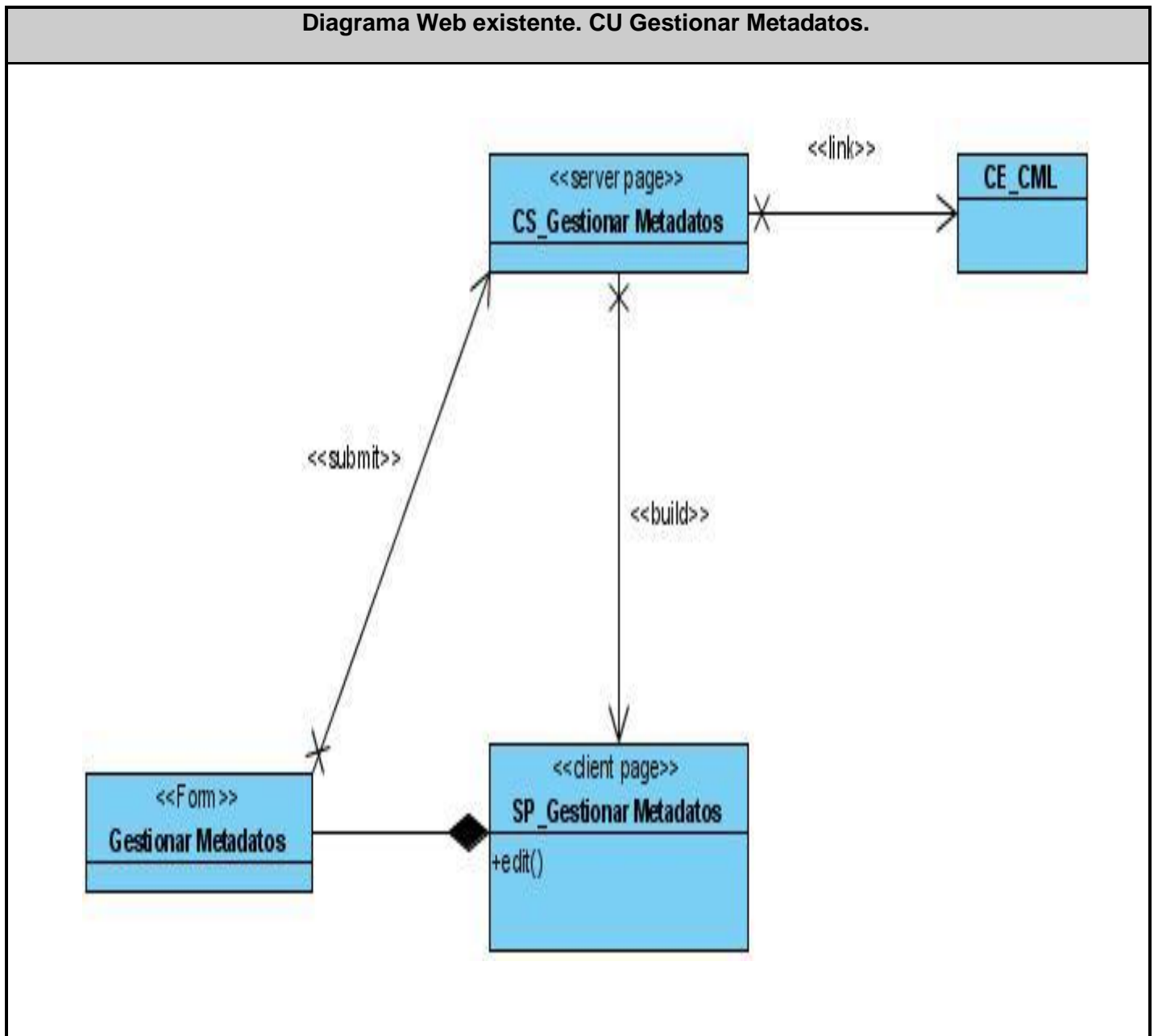


Fig 3.7 Diagrama de Estereotipos Web. CU Gestionar Metadatos.



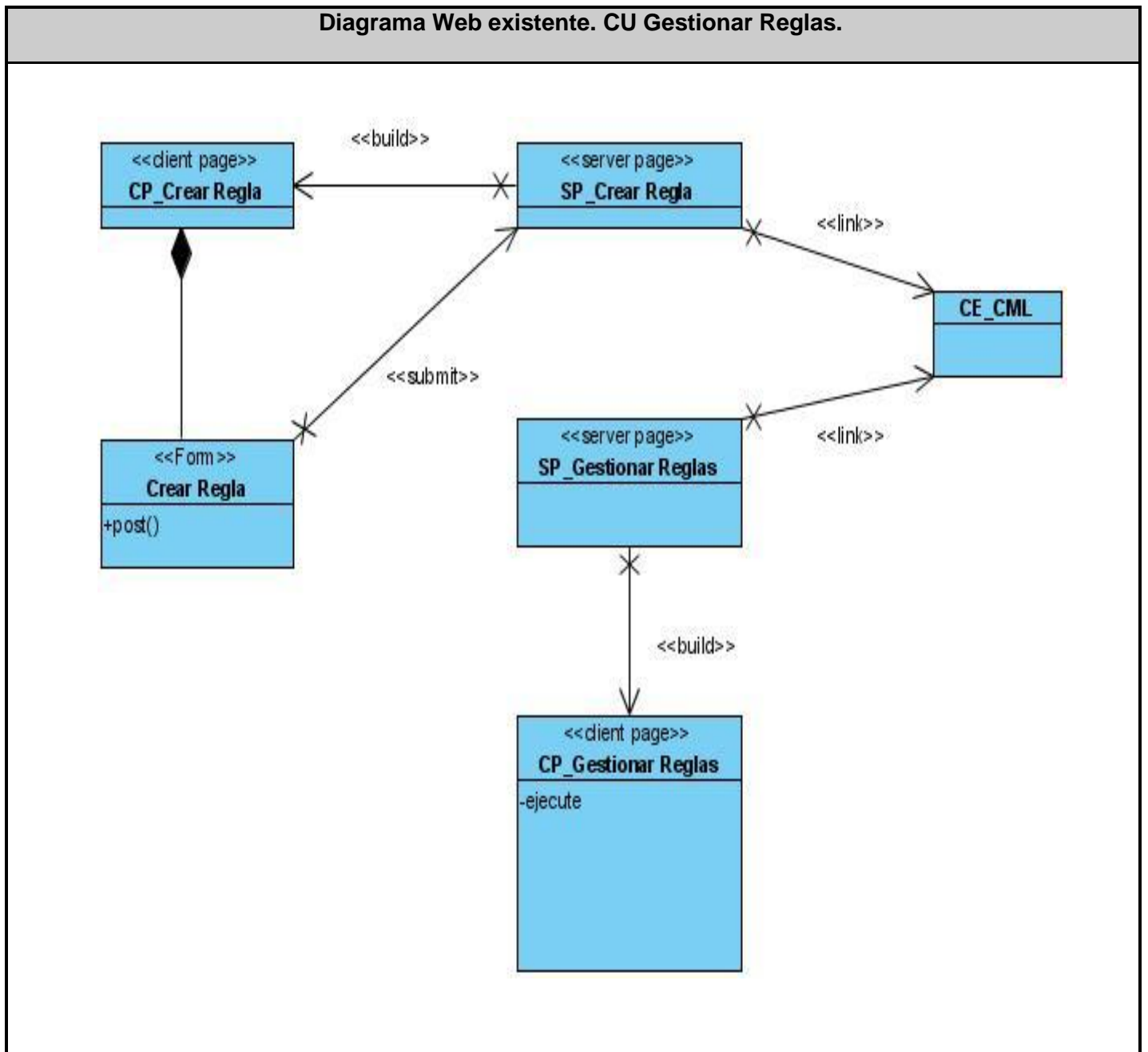
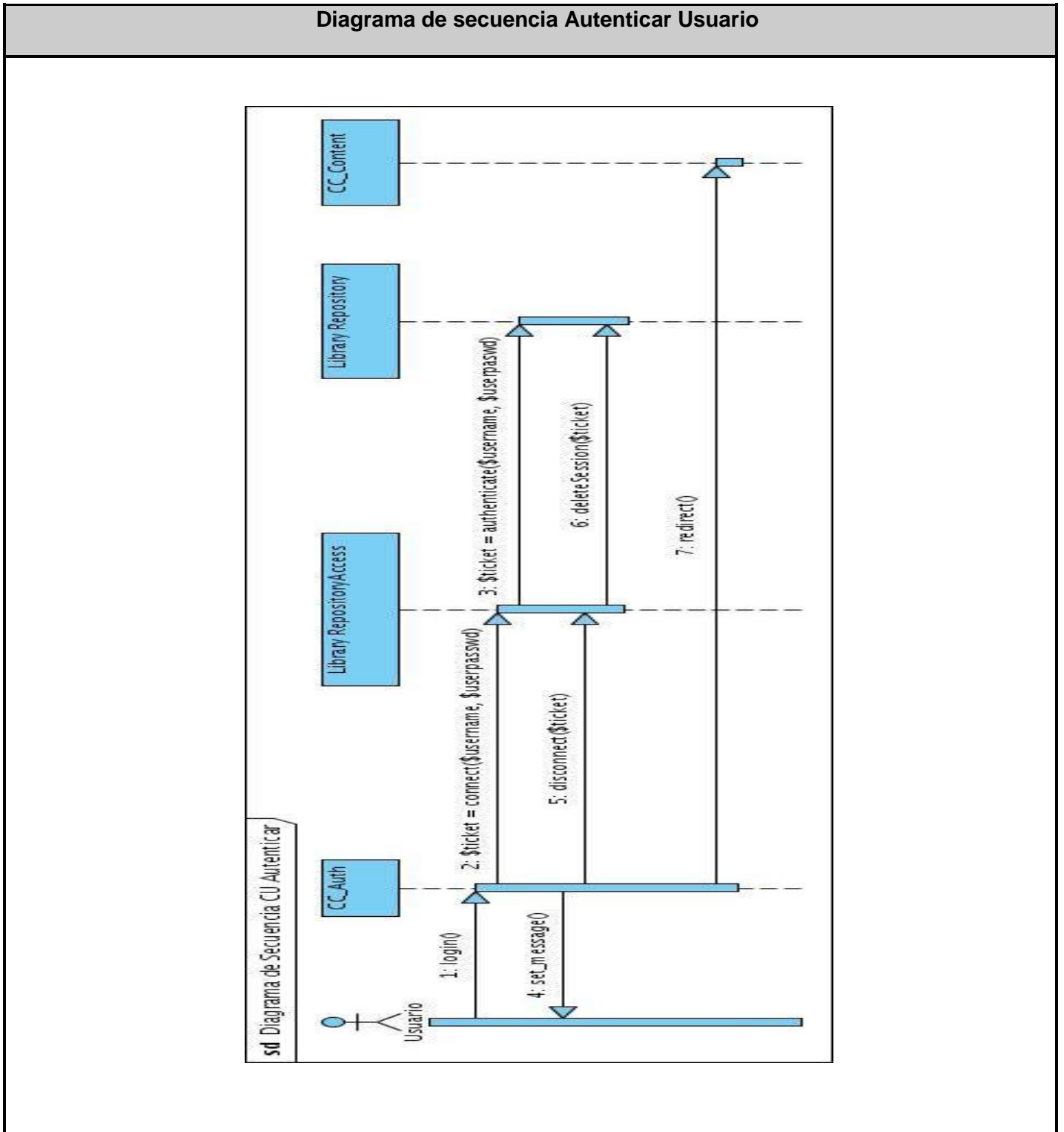


Fig 3.8 Diagrama de Estereotipos Web. CU Gestionar Reglas.

3.4 Representación UML de los diagramas de interacción.

3.4.1 Diagrama de secuencia CU Autenticar Usuario.

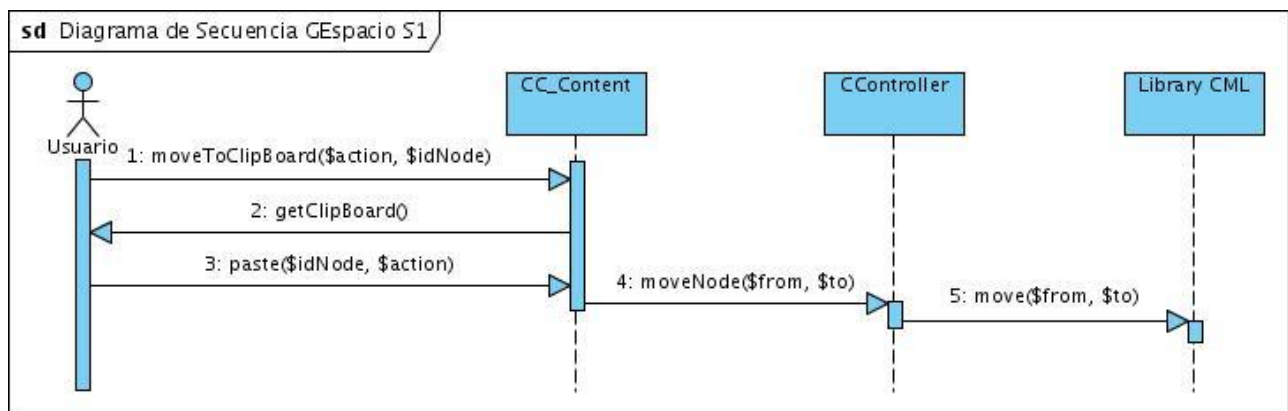


### 3.5.1 Diagrama de secuencia por escenario.

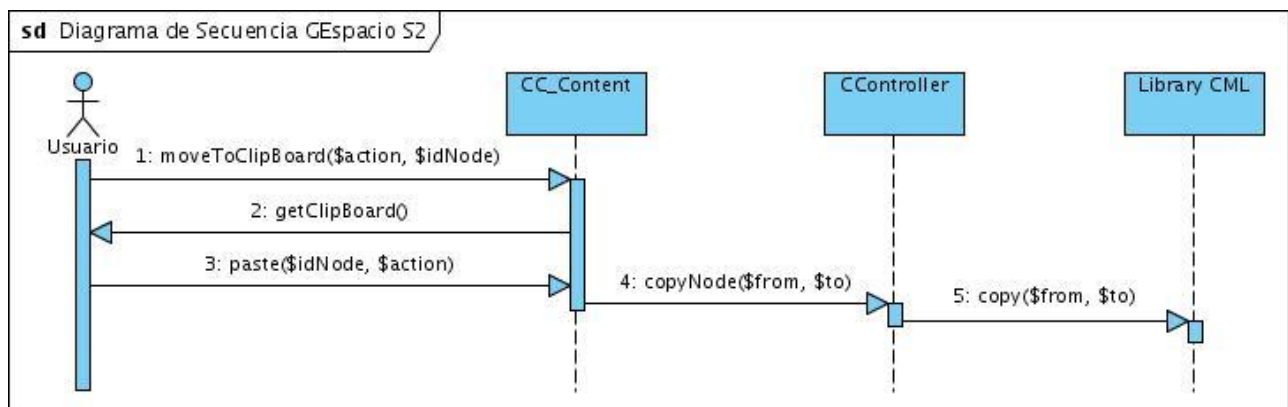
Los casos de usos a continuación se hace necesario dividirlo por sesiones en los cuales están compuestos. Los casos de usos que representamos no tienen un orden consecutivo por lo que el caso de uso se puede iniciar por cualquier sesión.

#### 3.5.1.1 Caso de Uso Gestionar Espacio.

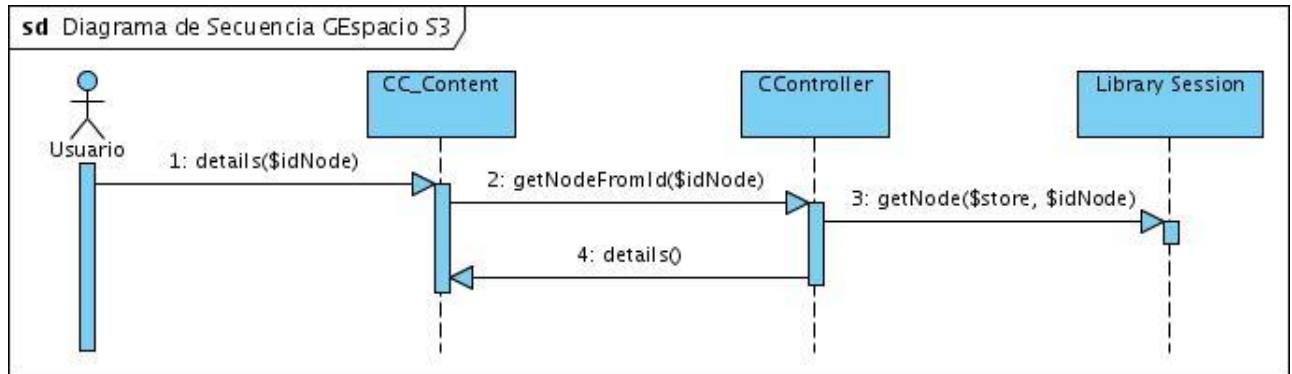
**Diagrama de secuencia por escenario. Gestionar Espacio Sección 1 Cortar**



**Diagrama de secuencia por escenario. Gestionar Espacio Sección Copiar**



**Diagrama de secuencia por escenario. Gestionar Espacio Sección Ver Detalles**



**Diagrama de secuencia por escenario. Gestionar Espacio Sección Editar**

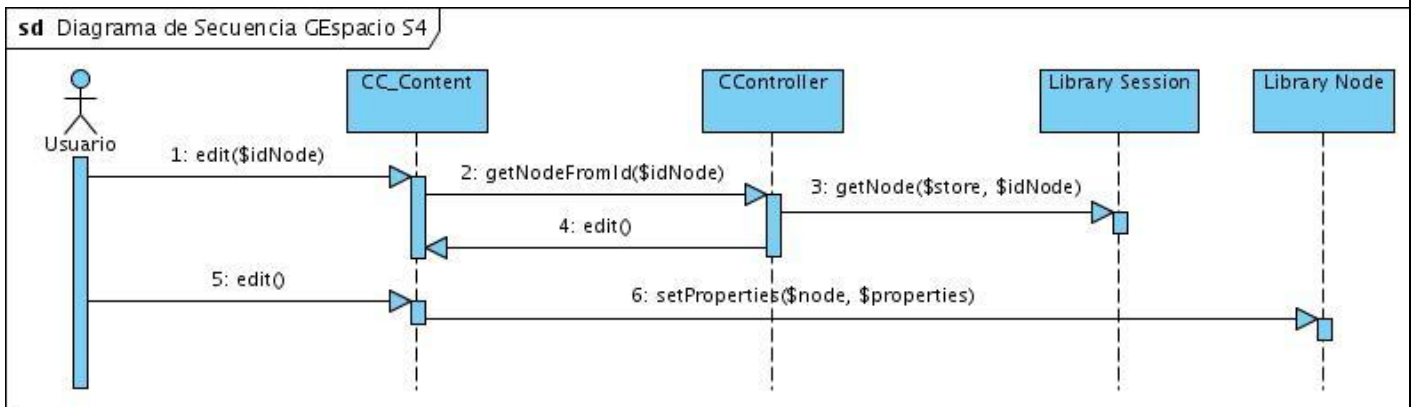


Diagrama de secuencia por escenario. Gestionar Espacio Sección Eliminar

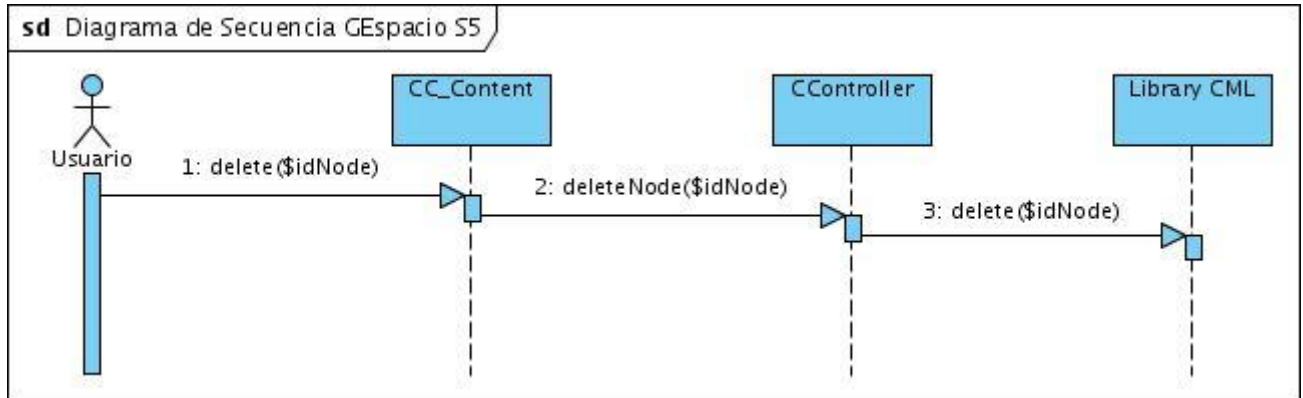
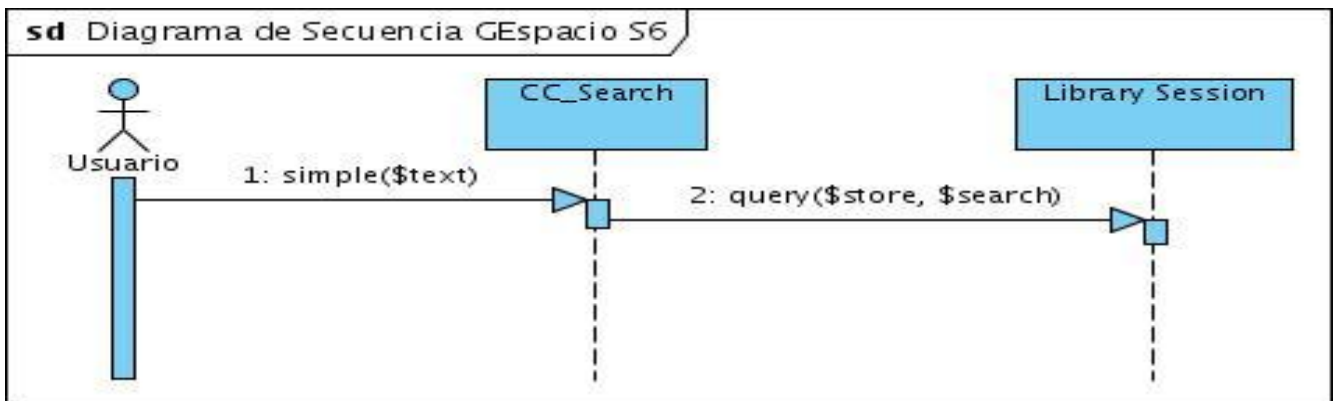


Diagrama de secuencia por escenario. Gestionar Espacio Sección Búsqueda



3.5.1.2 Caso de Uso Gestionar Contenido

Diagrama de secuencia por escenario. Gestionar Contenido Sección Cortar

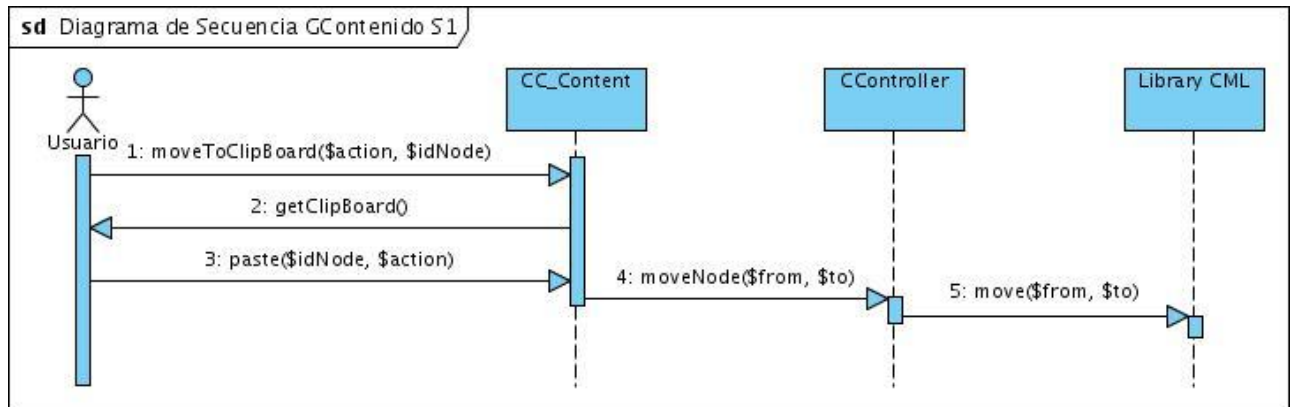
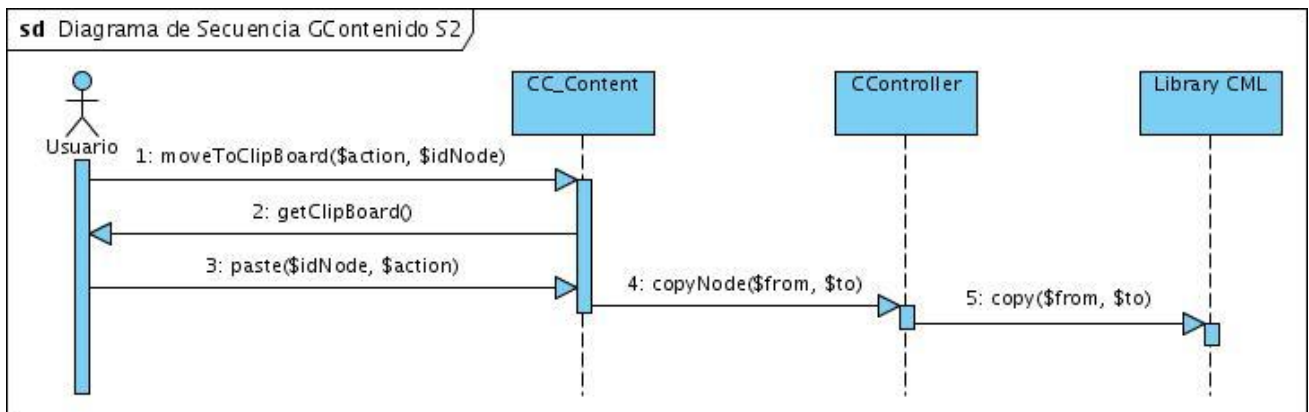
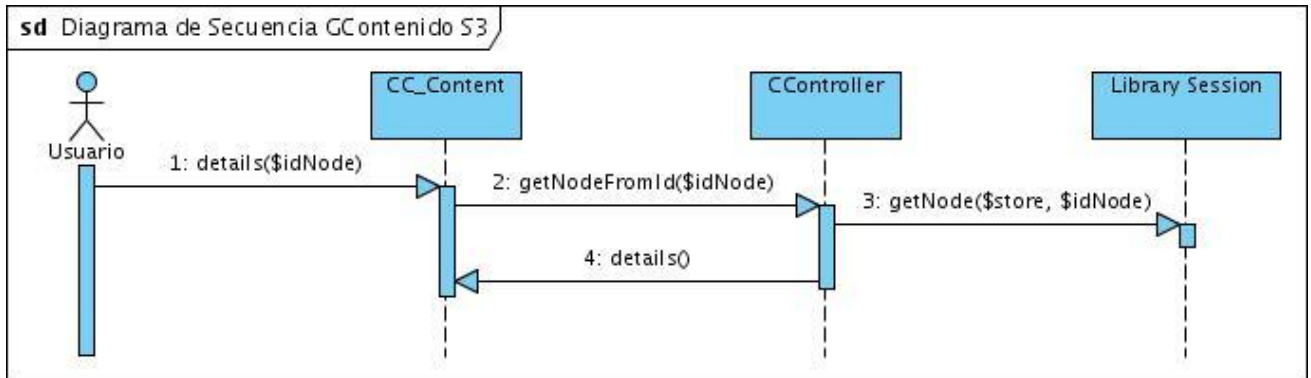


Diagrama de secuencia por escenario. Gestionar Contenido Sección Copiar



**Diagrama de secuencia por escenario. Gestionar Contenido Sección Ver Detalles**



**Diagrama de secuencia por escenario. Gestionar Contenido Sección Editar**

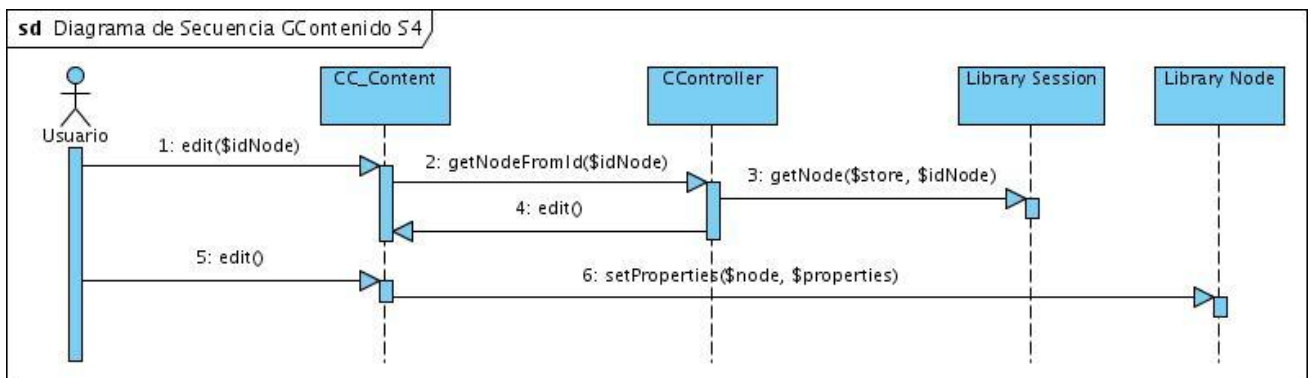


Diagrama de secuencia por escenario. Gestionar Contenido Sección Eliminar

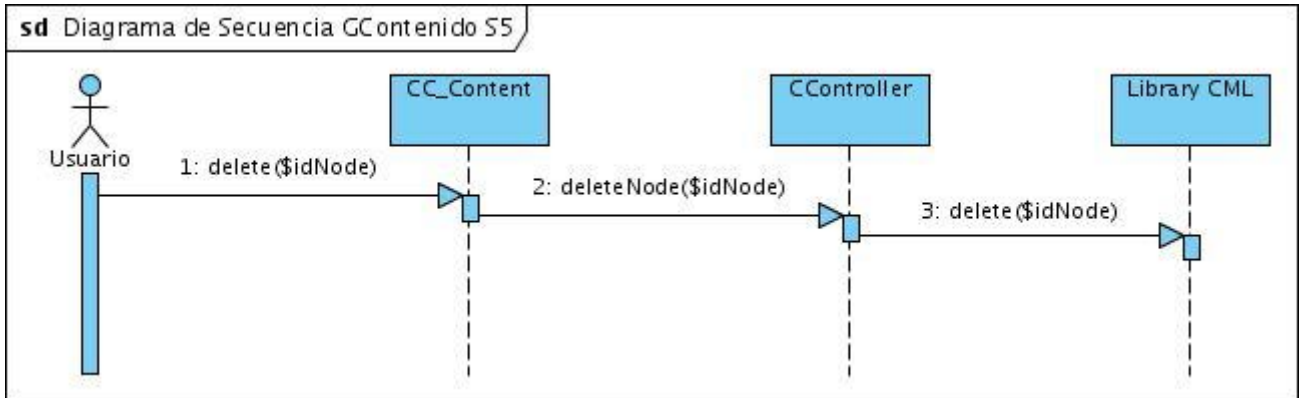


Diagrama de secuencia por escenario. Gestionar Contenido Sección Buscar

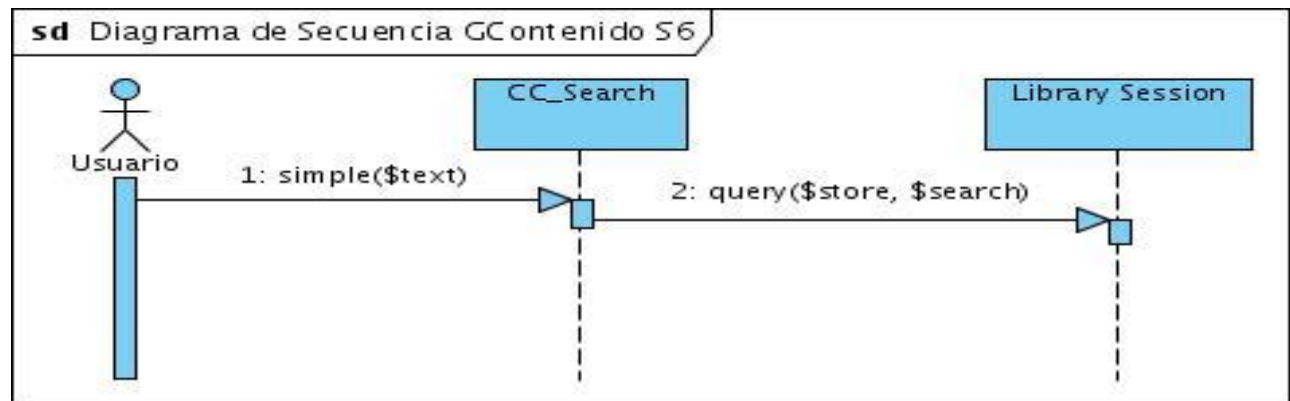
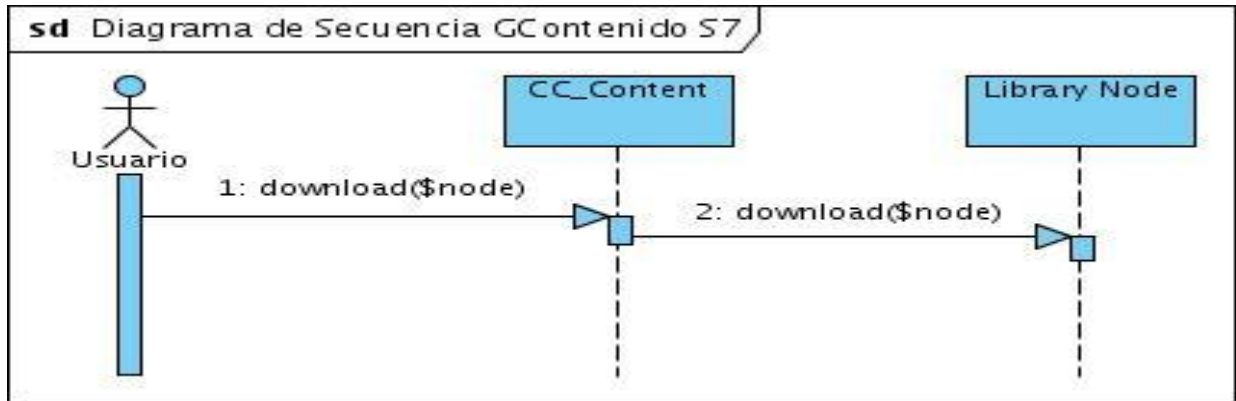


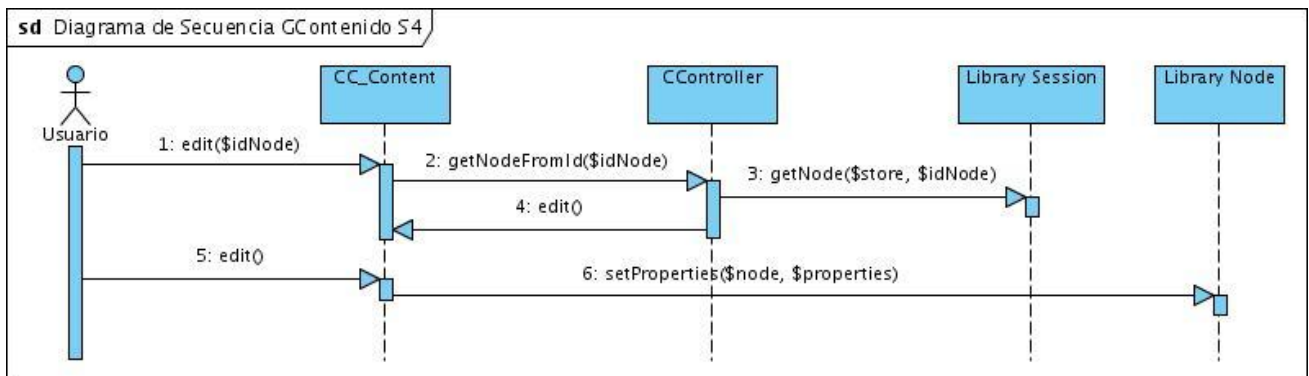


Diagrama de secuencia por escenario. Gestionar Contenido Sección Descargar

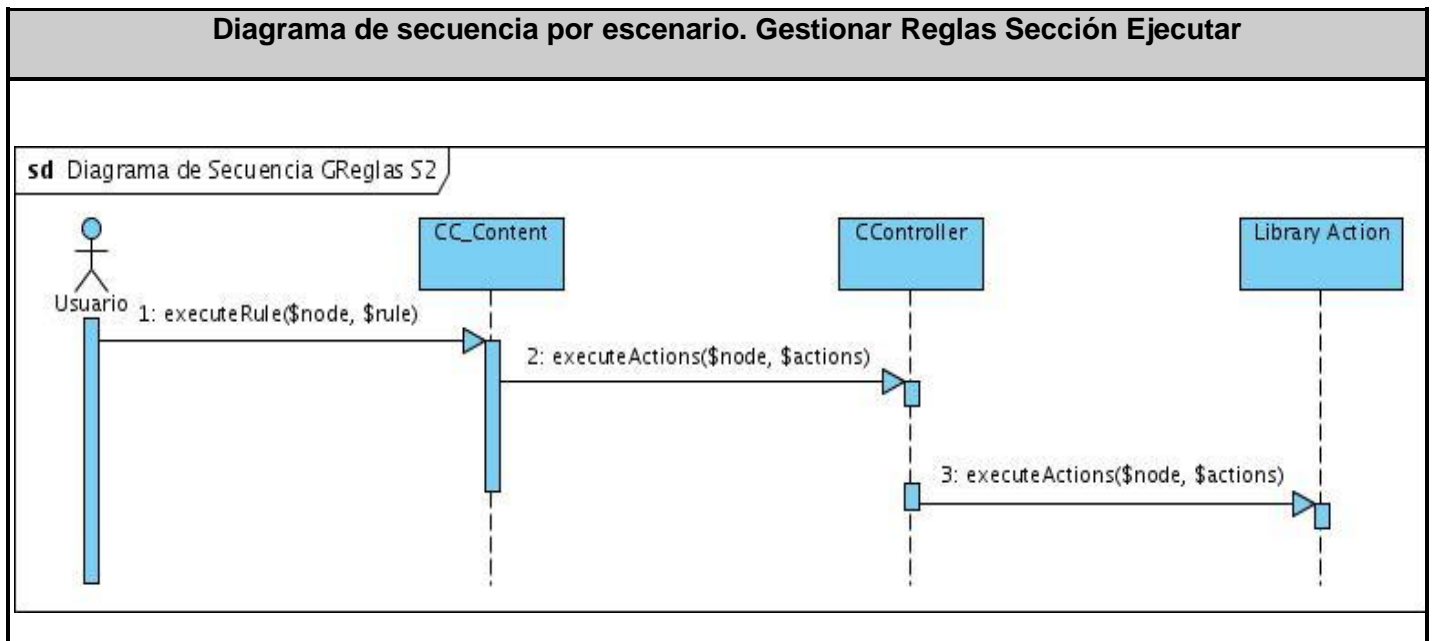


3.5.1.3 Caso de Uso Gestionar Metadatos.

Diagrama de secuencia por escenario. Gestionar Metadatos Sección Editar



## 3.5.1.4 Caso de Uso Gestionar Reglas.



### 3.6 Descripción de las Clases del Diseño.

#### Clase Auth.

<b>Nombre:</b> Auth	
<b>Clase controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Nombre:</b>	index()
<b>Descripción:</b>	Verifica que si el usuario está autenticado, en caso de estarlo lo redirecciona a Content en caso contrario a la vista de autenticación.
<b>Nombre:</b>	login()
<b>Descripción:</b>	Se encarga de autenticar al usuario en el sistema, si las credenciales son correctas lo envía a Content en caso contrario muestra la vista de autenticación.
<b>Nombre:</b>	logout()
<b>Descripción:</b>	Cierra la sesión del usuario en el sistema.

#### Clase Offline:

<b>Nombre:</b> Offline	
<b>Clase controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Nombre:</b>	index()
<b>Descripción:</b>	Muestra una vista con un mensaje que indica que el servidor de Alfresco está fuera de servicio.

**Clase Search.**

<b>Nombre:</b> Search	
<b>Clase controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Nombre:</b>	index()
<b>Descripción:</b>	Muestra la vista con el formulario de búsqueda avanzada.
<b>Nombre:</b>	simple()
<b>Descripción:</b>	Se encarga realizar la búsqueda simple en el repositorio de contenidos Alfresco.
<b>Nombre:</b>	advanced()
<b>Descripción:</b>	Se encarga de realizar la búsqueda avanzada en el repositorio de contenidos Alfresco.

**Clase Content**

<b>Nombre:</b> Content	
<b>Clase controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Nombre:</b>	index()
<b>Descripción:</b>	Se encarga de la navegación dentro del repositorio de contenidos Alfresco.
<b>Nombre:</b>	details()
<b>Descripción:</b>	Muestra los metadatos y propiedades de un contenido o espacio del repositorio.
<b>Nombre:</b>	edit()
<b>Descripción:</b>	Permite editar o modificar las propiedades de un espacio o contenido del repositorio.
<b>Nombre:</b>	tree()

<b>Descripción:</b>	Construye el árbol de navegación de contenidos.
<b>Nombre:</b>	upload()
<b>Descripción:</b>	Sube un archivo al servidor de contenidos.
<b>Nombre:</b>	create()
<b>Descripción:</b>	Crea un contenido en el repositorio de Alfresco.
<b>Nombre:</b>	cespace()
<b>Descripción:</b>	Crea un espacio dentro del repositorio de contenidos de Alfresco.
<b>Nombre:</b>	delete()
<b>Descripción:</b>	Elimina un contenido o espacio del repositorio de contenidos de Alfresco.
<b>Nombre:</b>	sing()
<b>Descripción:</b>	Firma digitalmente un documento o contenido.
<b>Nombre:</b>	moveToClipboard()
<b>Descripción:</b>	Crea una referencia de un contenido o espacio en el portapapeles.
<b>Nombre:</b>	clear()
<b>Descripción:</b>	Elimina todas las referencias creadas en el portapapeles.
<b>Nombre:</b>	paste()
<b>Descripción:</b>	Pega un contenido o espacio en un lugar especificado dentro del repositorio.
<b>Nombre:</b>	getTypeProperties()
<b>Descripción:</b>	Crea un formulario con las propiedades de un tipo de contenido.

**Clase CController.**

<b>Nombre:</b> CController
----------------------------

Clase paquetes de librerías	
<b>Atributos:</b>	<b>Tipo</b>
_store	space store
_session	session
_action control	action
_repository	repository
_ticket	string
_ci	content
_cml	CML
<b>Nombre:</b>	getCurrentNode()
<b>Descripción:</b>	Obtiene el nodo actual donde se está navegando.
<b>Nombre:</b>	getPath()
<b>Descripción:</b>	Obtiene la dirección completa del nodo raíz al nodo actual.
<b>Nombre:</b>	Get_extension(\$filename)
<b>Descripción:</b>	Obtiene la extensión de un archivo.
<b>Nombre:</b>	cutputBreadcrumb(\$offset)
<b>Descripción:</b>	Obtiene el Breadcrumb especificando el tamaño.
<b>Nombre:</b>	getImageFromMim_e_Type(\$minType,\$size)
<b>Descripción:</b>	Obtiene el icono del nodo de acuerdo con el Min e Tipe.
<b>Nombre:</b>	getImageURL(\$node)
<b>Descripción:</b>	Obtiene la dirección de la imagen de un nodo.
<b>Nombre:</b>	getURL(\$node)

<b>Descripción:</b>	Obtiene la dirección física de un nodo.
<b>Nombre:</b>	cutputRow(\$node, \$type)
<b>Descripción:</b>	Crea una fila para mostrar los contenidos.
<b>Nombre:</b>	cutputTable(\$title,\$type_filter,\$empty_messages)
<b>Descripción:</b>	Crea una tabla para mostrar los contenidos y espacios.
<b>Nombre:</b>	getNodefromId(\$id)
<b>Descripción:</b>	Obtiene el nodo completo dando el id del mismo.
<b>Nombre:</b>	emptyClipboard(\$id)
<b>Descripción:</b>	Elimina el nodo completo del id seleccionado del portapapeles.
<b>Nombre:</b>	getClipboard()
<b>Descripción:</b>	Obtiene los contenidos referenciados del portapapeles.
<b>Nombre:</b>	getRuleToNode(\$node)
<b>Descripción:</b>	Obtiene la regla asociado a un nodo.
<b>Nombre:</b>	getNiceSize(\$size)
<b>Descripción:</b>	Obtiene el tamaño de un archivo en formato legible al usuario.
<b>Nombre:</b>	getTypes(\$username, \$userpasswd, \$type)
<b>Descripción:</b>	Obtiene todos los tipos de contenidos creados en el Alfresco.
<b>Nombre:</b>	getPropertiesFromType(\$type, \$ref)
<b>Descripción:</b>	Obtiene las propiedades de un tipo de contenido especificado.
<b>Nombre:</b>	isType(\$type)
<b>Descripción:</b>	Verifica si un tipo de contenido puede ser visible por el usuario

**Clase Access**

<b>Nombre:</b> Access	
<b>Clase paquetes de librerías</b>	
<b>Atributos:</b> _repository _ticket _access_control _ci	<b>Tipo:</b> repository string access content
<b>Nombre:</b>	getPermissions(\$node)
<b>Descripción:</b>	Obtiene todos los permisos de un nodo.
<b>Nombre:</b>	hasPermissions(\$node,\$permissions)
<b>Descripción:</b>	Da un permiso determinado a un nodo.
<b>Nombre:</b>	getAlfPermissions(\$node)
<b>Descripción:</b>	Método que obtiene todos los permisos de un nodo de Alfresco.
<b>Nombre:</b>	hasAlfPermissions(\$node,\$permissions)
<b>Descripción:</b>	Método que da los permisos determinados a un nodo de Alfresco.
<b>Nombre:</b>	getOwners(\$node)
<b>Descripción:</b>	Obtiene las propiedades de un nodo.
<b>Nombre:</b>	setOwners(\$node,\$owners)
<b>Descripción:</b>	Cambia las propiedades de un nodo.



**Clase Repository Access**

<b>Nombre:</b> Repository Access	
<b>Clase paquetes de librerías</b>	
<b>Atributos:</b> _repository _ci	<b>Tipo:</b> repository content
<b>Nombre:</b>	connect(\$usser,\$paswrđ)
<b>Descripción:</b>	Para conectar un usuario al sistema.
<b>Nombre:</b>	Disconnect(\$ticket)
<b>Descripción:</b>	Para desconectar un usuario del sistema.
<b>Nombre:</b>	getRepository()
<b>Descripción:</b>	Obtiene un objeto de tipo Repository.

En este capítulo se ha desarrollado uno de los flujos fundamentales en el ciclo de vida del software: el flujo de trabajo Diseño. A partir de los casos de usos del sistema se modeló el diseño con los artefactos principales que propone, como son los diagramas de clases del diseño, diagramas de interacción y descripción de las clases.

## **CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA.**

Actividades de codificación del software que de acuerdo al diseño, cumplen con los requisitos del sistema. En este capítulo se expone el diagrama de despliegue según nuestra propuesta y el de componentes y luego se detallan cada uno de los paquetes por los que está compuesto, esto se hizo para lograr un buen entendimiento y organización.

### **4.2 Diagrama de Despliegue.**

El modelo de despliegue describe la distribución física del sistema, muestra como están distribuidos los componentes de software entre los distintos nodos de cómputo. Permite comprender la correspondencia entre la arquitectura software y la arquitectura hardware. Las estaciones de trabajo, dispositivos y procesadores son reflejados como nodos.

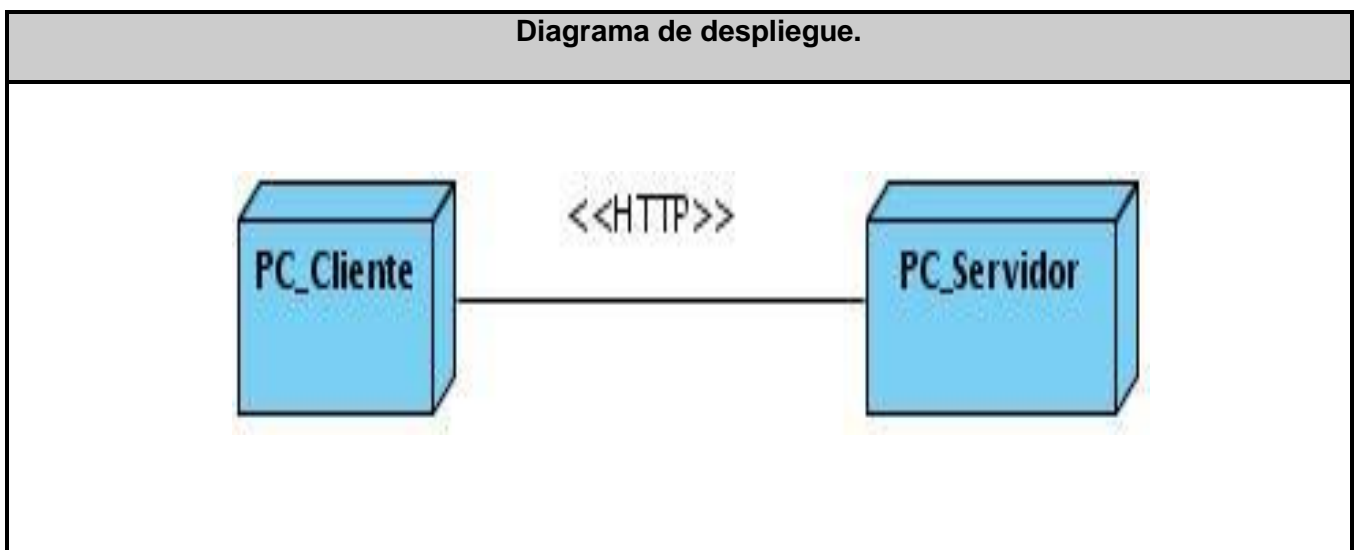


Fig 4.1 Diagrama de Despliegue.

### **4.3 Estructuración del Modelo de Implementación.**

El diagrama de componentes define cómo las clases, artefactos y otros elementos de bajo nivel, se unen para formar componentes de alto nivel y las conexiones entre ellos. Los componentes son artefactos de software compilados que trabajan acoplados para brindar el comportamiento requerido dentro de las restricciones definidas en el proceso de captura de requisitos.

Diagrama de componentes general.

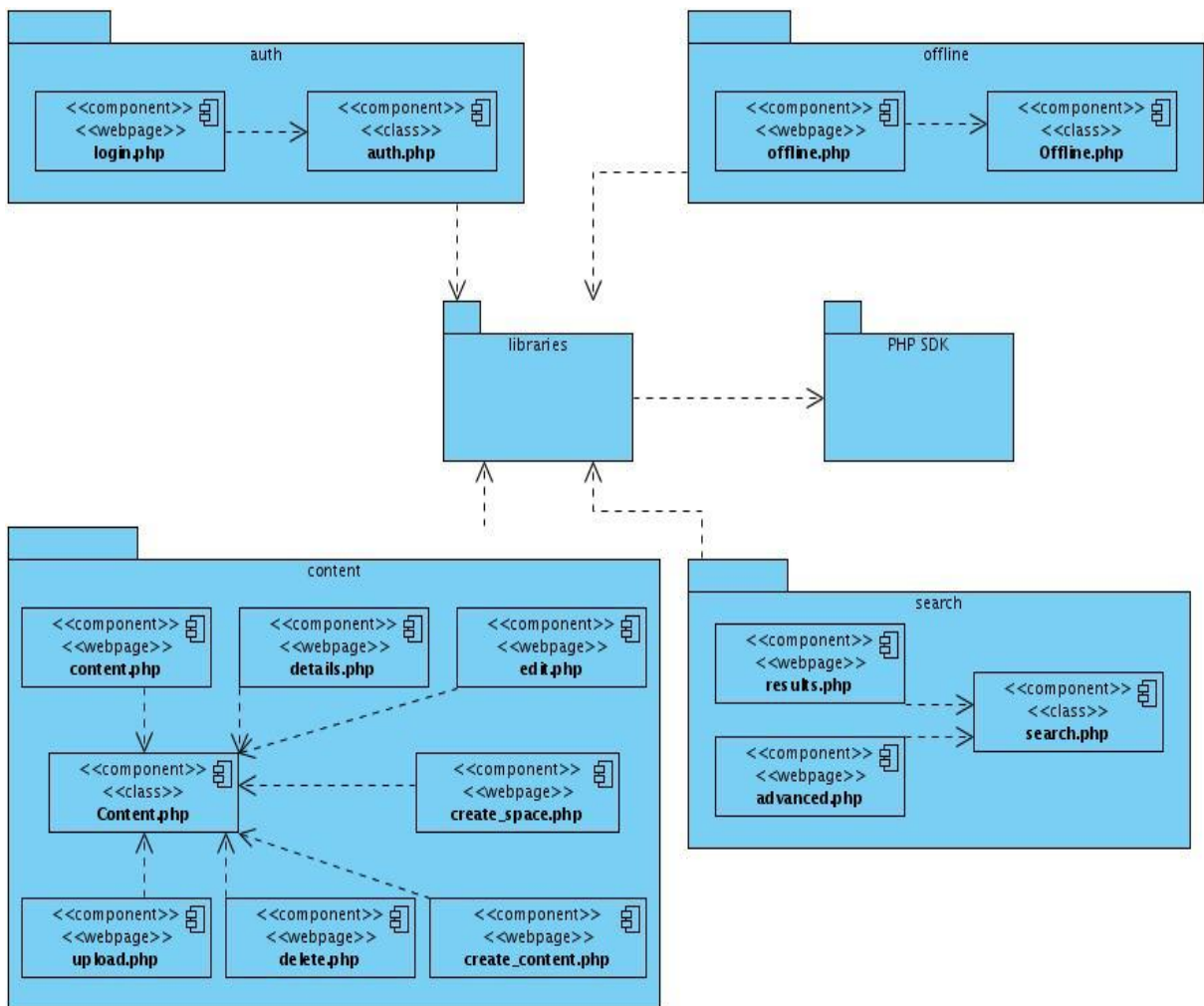


Diagrama de componentes del subpaquete de librerías.

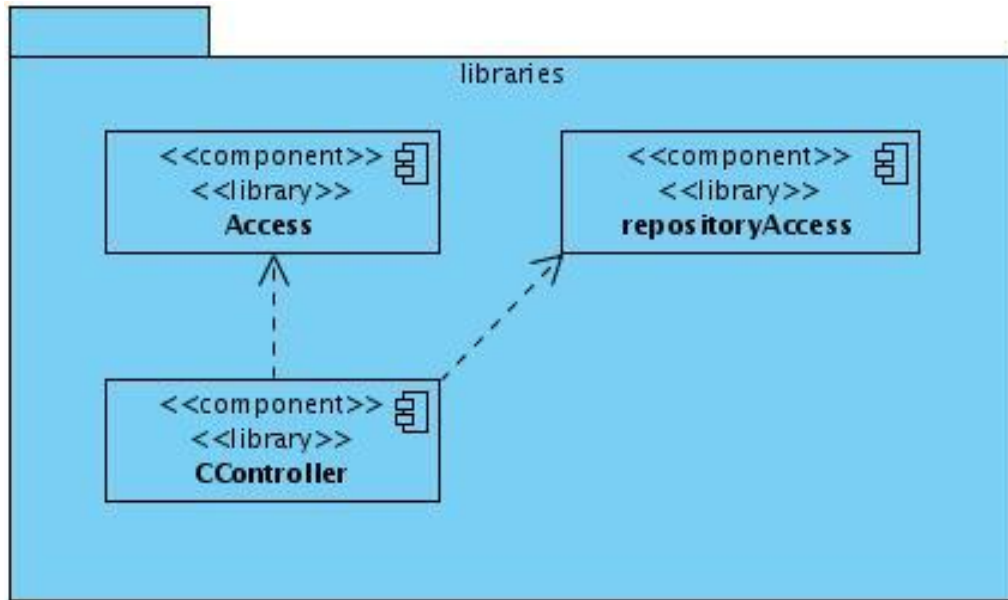
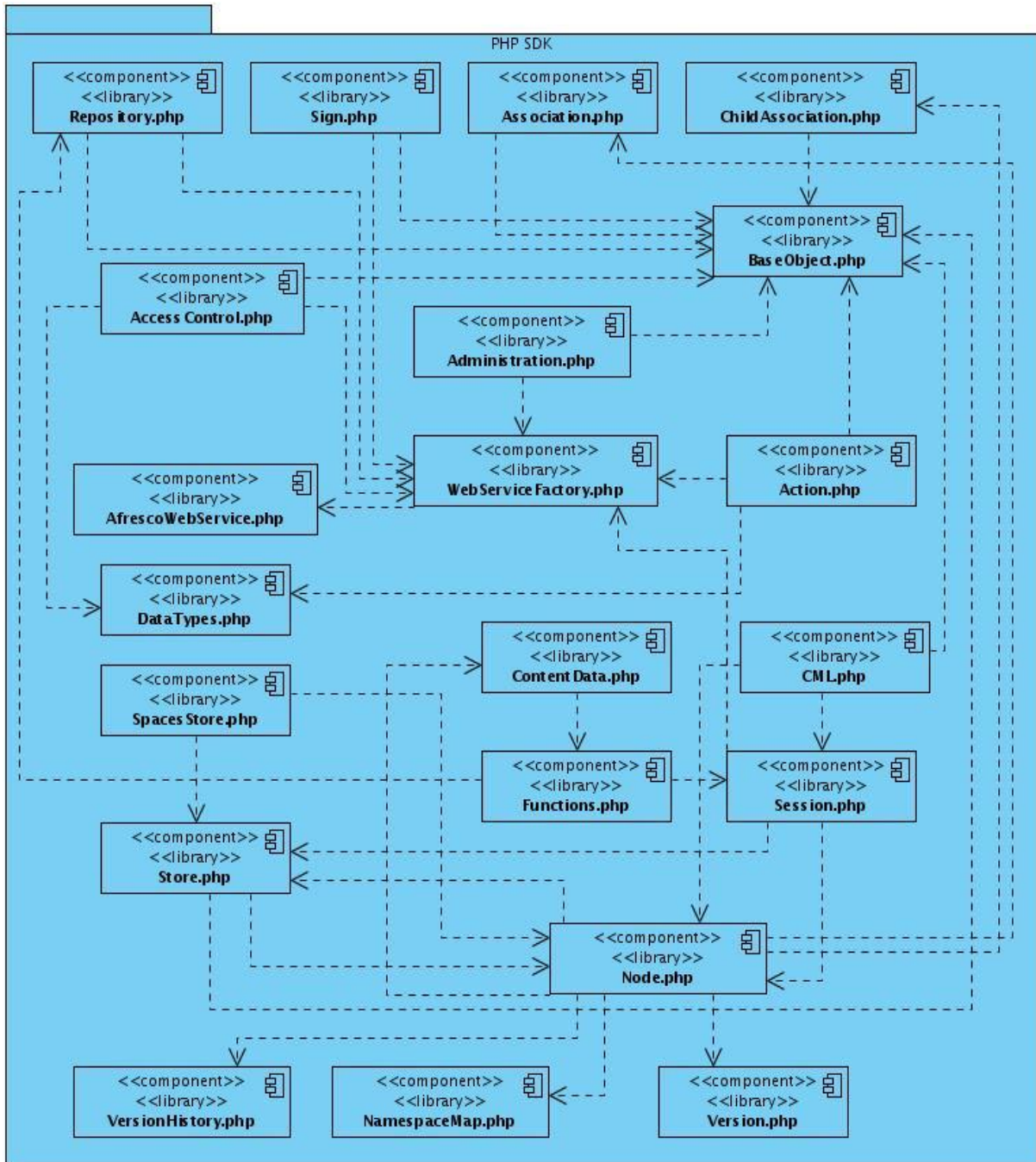


Diagrama de componentes del subpaquete de PHP SDK.



#### 4.4 Prueba de caja Negra

La prueba de Caja Negra se centra principalmente en los requisitos funcionales del *software*. Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, centrándose en los requisitos funcionales del sistema y ejercitándolos.

Para preparar los casos de pruebas hacen falta un número de datos que ayuden a la ejecución de los estos casos y que permitan que el sistema se ejecute en todas sus variantes, pueden ser datos válidos o inválidos para el programa según si lo que se desea es hallar un error o probar una funcionalidad. Los datos se escogen atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el programa corra bien.(21)

##### 4.4.1 Casos de Prueba de Integración.

Los casos de prueba son un conjunto de condiciones mediante las cuales se comprueba si los requisitos funcionan correctamente.

Las descripciones de los casos de prueba de integración realizadas a continuación contienen el nombre del caso de uso, el caso de prueba, se describe el estado de la información almacenada, el resultado que se obtiene y las condiciones que se deben cumplir en el momento que se ejecuta el caso de uso.

Caso de uso: Autenticar Usuario.

<b>Caso de Uso: Autenticar Usuario.</b>
<b>Caso de Prueba:</b> Autenticar usuario existente.
<b>Entrada:</b> El usuario desea entrar al sistema y se autentica poniendo su usuario y contraseña. El usuario está registrado.
<b>Resultado:</b> Se permite la entrada del usuario otorgándosele los permisos correspondientes.

<p><b>Condiciones:</b></p> <p>No se permite que se ejecuten otras instancias del caso de uso para el usuario.</p>
<p><b>Tabla 4.1: Descripción del caso de prueba, Autenticar usuario existente, en el CU, Autenticar usuario</b></p>

<p><b>Caso de Uso: Autenticar Usuario.</b></p>
<p><b>Caso de Prueba:</b> Autenticar usuario no existente.</p>
<p><b>Entrada:</b></p> <p>El usuario desea entrar al sistema y se autentica poniendo su usuario y contraseña. El usuario no está registrado.</p>
<p><b>Resultado:</b></p> <p>No se permite la entrada del usuario y se muestra un mensaje confirmando que el usuario no está registrado.</p>
<p><b>Condiciones:</b></p> <p>No se permite que se ejecuten otras instancias del caso de uso para el usuario.</p>
<p><b>Tabla 4.2: Descripción del caso de prueba, Autenticar usuario no existente, en el CU, Autenticar usuario</b></p>

<p><b>Caso de Uso: Autenticar Usuario.</b></p>
<p><b>Caso de Prueba:</b> Usuario o contraseña incorrecta de usuario registrado.</p>
<p><b>Entrada:</b></p> <p>El usuario desea entrar al sistema y se autentica poniendo su usuario y contraseña. El usuario está registrado.</p>
<p><b>Resultado:</b></p> <p>No se permite la entrada del usuario y se muestra un mensaje confirmando que el usuario o la contraseña no están correctos.</p>

<p><b>Condiciones:</b></p> <p>No se permite que se ejecuten otras instancias del caso de uso para el usuario.</p>
<p><b>Tabla 4.3: Descripción del caso de prueba, Autenticar usuario no existente, en el CU, Autenticar usuario</b></p>

Casos de Uso: Gestionar Espacio.

<p><b>Caso de Uso: Gestionar Espacio.</b></p>
<p><b>Caso de Prueba:</b> Registrar espacio no existente.</p>
<p><b>Entrada:</b></p> <p>El administrador desea registrar un espacio y entra los datos del mismo.</p> <p>Espacio no registrado</p>
<p><b>Resultado:</b></p> <p>Se registra el espacio, se muestra la carpeta en el examinador de espacios.</p>
<p><b>Condiciones:</b></p> <p>No se permite que se ejecuten otras instancias del caso de uso para el administrador.</p>
<p><b>Tabla 4.4: Descripción del caso de prueba, Registrar espacio no existente, en el CU, Gestionar Espacio.</b></p>

<p><b>Caso de Uso: Gestionar Espacio.</b></p>
<p><b>Caso de Prueba:</b> Registrar espacio existente.</p>
<p><b>Entrada:</b></p> <p>El administrador desea registrar un espacio y entra los datos del mismo.</p> <p>Espacio registrado</p>
<p><b>Resultado:</b></p>



No se registra el espacio, se muestra un mensaje confirmando que no se pudo crear la carpeta.
<b>Condiciones:</b> No se permite que se ejecuten otras instancias del caso de uso para el administrador.
<b>Tabla 4.5: Descripción del caso de prueba, registrar espacio existente, en el CU, Gestionar Espacio</b>

<b>Caso de Uso: Gestionar Espacio.</b>
<b>Caso de Prueba:</b> Eliminar espacio existente.
<b>Entrada:</b> El administrador desea eliminar un espacio existente y da en la opción eliminar carpeta
<b>Resultado:</b> El sistema muestra una notificación avisando de que la carpeta va ser borrada del repositorio.
<b>Condiciones:</b> No se permite que se ejecuten otras instancias del caso de uso para el administrador.
<b>Tabla 4.6: Descripción del caso de prueba, eliminar espacio existente, en el CU, Gestionar Espacio</b>

<b>Caso de Uso: Gestionar Espacio.</b>
<b>Caso de Prueba:</b> Editar espacio.
<b>Entrada:</b> El administrador desea editar una carpeta y da en la opción editar.
<b>Resultado:</b> El sistema muestra los campos a llenar o lanza un mensaje de error si los campos son incorrectos o coinciden con otro espacio existente en el repositorio.

<b>Condiciones:</b>
No se permite que se ejecuten otras instancias del caso de uso para el administrador.
<b>Tabla 4.7: Descripción del caso de prueba, editar espacio existente, en el CU, Gestionar Espacio</b>

Casos de Uso: Gestionar Contenido.

<b>Caso de Uso: Gestionar Contenido.</b>
<b>Caso de Prueba:</b> Crear o adicionar un contenido existente.
<b>Entrada:</b>
El administrador desea crear o adicionar un contenido existente y entra los datos. Contenido existente.
<b>Resultado:</b>
No se registra el espacio, se muestra un mensaje confirmando que no se pudo crear el contenido.
<b>Condiciones:</b>
No se permite que se ejecuten otras instancias del caso de uso para el administrador.
<b>Tabla 4.8: Descripción del caso de prueba, crear o adicionar contenido existente, en el CU, Gestionar Contenido</b>

<b>Caso de Uso: Gestionar Contenido.</b>
<b>Caso de Prueba:</b> Crear o adicionar un contenido no existente.
<b>Entrada:</b>
El administrador desea crear o adicionar una carpeta existente y entra los datos. Contenido no existente.
<b>Resultado:</b>

Se registra el contenido. Y se muestra en los elementos de contenido.
<b>Condiciones:</b> No se permite que se ejecuten otras instancias del caso de uso para el administrador.
<b>Tabla 4.9: Descripción del caso de prueba, crear o adicionar contenido no existente, en el CU, Gestionar Contenido</b>

<b>Caso de Uso: Gestionar Contenido.</b>
<b>Caso de Prueba:</b> Editar contenido.
<b>Entrada:</b> El administrador desea editar un contenido y da en la opción editar.
<b>Resultado:</b> El sistema muestra los campos a llenar o lanza un mensaje de error si los campos son incorrectos o coinciden con otro contenido existente en el repositorio.
<b>Condiciones:</b> No se permite que se ejecuten otras instancias del caso de uso para el administrador.
<b>Tabla 4.10: Descripción del caso de prueba, editar contenido existente, en el CU, Gestionar Contenido</b>

Casos de Uso: Gestionar Metadatos.

<b>Caso de Uso: Gestionar Metadatos.</b>
<b>Caso de Prueba:</b> Editar un metadatos.
<b>Entrada:</b> El administrador desea editar un metadatos y accede a la opción editar.
<b>Resultado:</b>

El sistema muestra los campos a llenar o lanza un mensaje de error si los campos son incorrectos.
<b>Condiciones:</b> No se permite que se ejecuten otras instancias del caso de uso para el administrador.
<b>Tabla 4.11: Descripción del caso de prueba, editar metadatos existente, en el CU, Gestionar Metadatos</b>

Casos de Uso: Gestionar Reglas.

<b>Caso de Uso: Gestionar Reglas.</b>
<b>Caso de Prueba:</b> Ejecutar una Regla.
<b>Entrada:</b> El administrador desea ejecutar una Regla existente a un contenido.
<b>Resultado:</b> Se ejecuta la regla, se muestra un mensaje confirmando que al contenido deseado por el usuario se le ejecuto dicha regla.
<b>Condiciones:</b> No se permite que se ejecuten otras instancias del caso de uso para el administrador.
<b>Tabla 4.12: Descripción del caso de prueba, ejecutar regla, en el CU, Gestionar Reglas</b>

En este capítulo se comenzó mostrando el diagrama de despliegue para ver mejor la distribución física del sistema. Luego se explica el diagrama de componentes y por último se realizan los casos de pruebas al sistema. Se aplica las pruebas de caja negra a cada casos de uso del sistema, identificando las principales interacciones entre usuario-sistema.

**CONCLUSIONES:**

En este trabajo se demostró la necesidad de desarrollar e implementar una interfaz web amigable al Gestor de Contenido Empresarial Alfresco, que brinde al usuario final una facilidad a la hora de gestionar todos los recursos de una administración.

Para la creación de esta interfaz web se elaboró la propuesta de utilizar el lenguaje de programación PHP, utilización paquete de librerías API de PHP y el Framework Code Igniter.

Se utilizó la metodología RUP para llevar a cabo el proceso de desarrollo, y lenguaje de modelado UML a través de la herramienta CASE Visual Paradigm.

En la investigación se determinó que no se definían concretamente todos los procesos, para ello se realizó un modelo del dominio.

En el diseño se elaboró el modelado de los diagramas de clases del diseño, se representan los diagramas UML de Estereotipos Web del Diseño, de secuencia y así como la descripción de las clases.

Por otra parte se propuso la realización del diagrama de despliegue en el cual se expone la ubicación física de cada componente de hardware y software, y el diagrama de componente para describir los componentes físicos del sistema y sus relaciones. Se sometió la aplicación a pruebas de caja negra, la cual se centra principalmente en los requisitos funcionales.

Luego de realizado todo este trabajo se puede concluir que la interfaz web implementada brinda la solución al problema de la investigación, la cual posibilitó el origen a esta investigación.

**RECOMENDACIONES:**

Sobre la presente investigación recomendamos:

- Seguir con las actualizaciones en el mundo de las tecnologías implementadas que faciliten la interoperabilidad con ECM Alfresco debido que facilitan la integración de todos los intentos de homogeneizar y gestionar la información de una institución.
- Integrar las funcionalidades de GDOA con el módulo GMV implementado para el CMS Drupal que logra la interoperabilidad con el ECM Alfresco.
- Someter a una prueba de calidad rigurosa al software para su posterior uso.

**REFERENCIAS BIBLIOGRÁFICAS.**

[1]. **Caro, Patricio Salinas y Histchfeld K, Nancy.** Tutorial de UML .  
<http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html>.

[2] **JACOBSON, Ivar.** *Rational Unified Process 2000.* <http://es.wikipedia.org/wiki/RUP>



## BIBLIOGRAFÍA.

1. ARCHIVO. Reglamento General de Archivo. *DOCUARCHIVISTA*, 2009-02-04 2007, vol. 1, nº Disponible en: <http://sturizo.blogspot.es/>.
2. GRANMA, A. H. P. D. Archivo de Gestión Administrativos. *Granma*, 2009-02-17 2006, nº Disponible en: <http://www.archivo.granma.inf.cu/archivo/index.html>.
3. FERRERES, G. Tipologías de Sitios Web. *Mosaic*, 2009-02-25 2006, nº p. 2. Disponible en: <http://mosaic.uoc.edu/articulos/gferrerres1106.html>.
4. MENDOZA, J. A. Definiendo el Género de su sitio web. *desarrollo web*, 05- 02-2009 2001, vol. -, nº -, p. 3. Disponible en: <http://sturizo.blogspot.es/>.
5. ÁLVARES, M. Á. Concepto de páginas dinámicas y estáticas. *desarrollo web*, 2009-02-26 2002, nº p. 2. Disponible en: <http://www.desarrolloweb.com/articulos/712.php>.
6. ---. Conceptos del Lenguaje HTML. *desarrollo web*, 2009-03-02 2002, nº p. 2. Disponible en: <http://www.desarrolloweb.com/articulos/25.php>.
7. ---. Conceptos del lenguaje Script. *desarrollo web*, 2009-03-06 2002, nº p. 2. Disponible en: <http://www.desarrolloweb.com/articulos/25.php>.
8. GALLÚS, C. PHP orientado a objeto. *Web estilo*, 2009-03-10 2002, vol. Publicaciones I, II, III, nº Disponible en: <http://www.webestilo.com/php/articulo.phtml?art=27>.
9. RUP. [Online], [Cited: febrero 20, 2009] 2004, nº Disponible en: <http://www.scribd.com/doc/297224/RUP>.
10. BECK, K. eXtreme Programming. *Wikipedia*, 1999, nº p. 2. Disponible en: [http://es.wikipedia.org/wiki/Programaci%C3%B3n\\_Extrema](http://es.wikipedia.org/wiki/Programaci%C3%B3n_Extrema).
11. COAD, E. L. Y. J. D. Feature Driven Development. 2009-03-07 2000, nº p. 10. Disponible en: <http://www.fing.edu.uy>.
12. ADMINISTRACIÓN., M. Y. O. D. L. Metodología Métrica. 2009-03-09 nº p. 17. Disponible en: <http://www.csi.map.es/csi/metrica3/index.html>.
13. JOHNSON, S. Alfresco. *Sitio de Alfresco*, 2009-03-09 2006, nº Disponible en: <http://www.alfresco.com/es/>.
14. LTD, J. E. Code Igniter. *EllisLab*, 2009-03-12 2009, nº p. 1. Disponible en: [http://translate.google.com/cu/translate?hl=es&sl=en&u=http://codeigniter.com/&ei=wKD\\_SenE\\_FtDHtgfU7KyWBw&sa=X&oi=translate&resnum=1&ct=result&prev=/search%3Fq%3DFramewor+k%2BCode%2BIgniter%26hl%3Des%26client%3Dfirefox-a%26rls%3Dorg.mozilla:en-US:official%26hs%3D89U](http://translate.google.com/cu/translate?hl=es&sl=en&u=http://codeigniter.com/&ei=wKD_SenE_FtDHtgfU7KyWBw&sa=X&oi=translate&resnum=1&ct=result&prev=/search%3Fq%3DFramewor+k%2BCode%2BIgniter%26hl%3Des%26client%3Dfirefox-a%26rls%3Dorg.mozilla:en-US:official%26hs%3D89U).

15. ¿Qué es una API? *Manual de Programación*, 2009-03-16 2008, nº p. 1. Disponible en: <http://www.rastersoft.com/OS2/CURSO/APIEXPL.HTM>.
16. MORILLAS, S. M. Módulo GMV. *Mi Espacio*, 2009-03-17 2008, nº Disponible en: <http://www.manuelrecena.com/blog/archives/565>.
17. ROTZ, B. V. DoCASU. *EOSDirectory*, 2009-03-20 2008, nº Disponible en: <http://www.eosdirectory.com/blogs/2008/07/15/>.
18. JOOMLATOOLS, A. Y. Joomla. *Alfresco*, 2009-03-26 2009, nº Disponible en: <http://www.alfresco.com/es/media/releases/2008/12/joomla/>.
19. KANDZIOR, A. OpenCms. *SlideShare*, 2009-03-27 1999, nº Disponible en: <http://www.slideshare.net/osmarinero/taller-open-cms-instalacion-presentation>.
20. DAPENA, M. D. D. Definición del modelo del negocio y del dominio utilizando Razonamiento Basado en Casos. *Documento*, 2009-03-28 2006, nº p. 11. Disponible en: <http://www.inf.udec.cl/~revista/ediciones/edicion8/Rbc.pdf>.
21. MAÑAS, J. A. Caja Negra. *Pruebas de Programas.*, 2009-04-29 1994 nº p. 1. Disponible en: <http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/testing.htm>.

## GLOSARIO DE TÉRMINOS:

Término	Definición
<b>API</b>	(Del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones) es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta librería para ser utilizado por otro software como una capa de abstracción.
<b>ECM Alfresco</b>	Es la alternativa de Código Abierto <b>para la gestión de contenido empresarial</b> (ECM), proporcionando gestión documental, colaboración, gestión de registros, gestión de información, gestión del contenido web e imágenes.
<b>Interfaz</b>	La capa de presentación del sistema
<b>Interoperabilidad</b>	La condición mediante la cual sistemas heterogéneos pueden intercambiar procesos o datos.
<b>HTML</b>	Lenguaje de Marcación de Hipertexto. Es un lenguaje de marcas diseñado para estructurar textos y representarlos en forma de hipertexto, que es el formato estándar de las páginas web.
<b>PHP</b>	Lenguaje de programación interpretado.
<b>Repositorio</b>	Es un término utilizado en el dominio de las herramientas CASE. El repositorio podría definirse como la base de datos fundamental para el diseño; no sólo guarda datos, sino también algoritmos de diseño y, en general, elementos software necesarios para el trabajo de programación.

<b>RUP</b>	El Proceso Unificado de Racional es un proceso de desarrollo de software.
<b>SDK</b>	<p>Un <b>software Development kit (SDK)</b> o <b>kit de desarrollo de software</b> es generalmente un conjunto de herramientas de desarrollo que le permite a un programador crear aplicaciones para un sistema.</p> <p>Un servicio web (en inglés Web service) es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.</p>
<b>Servicio WEB</b>	Un servicio web (en inglés Web service) es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.
<b>UML</b>	Lenguaje Unificado de Modelado, lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos.
<b>Usuario</b>	Todo el personal con privilegios para acceder al sistema.