



**Universidad de las Ciencias Informáticas**  
**Facultad 10**

**Título: Análisis y Diseño de una herramienta de autor Web interoperable.**

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

**Autores:** Judith Carrodegua Argote  
Henry Ricardo Labañino

**Tutores:** MSc. David Leyva Leyva  
MSc. Daymy Tamayo Avila  
Ing. Dunia María Colomé Cedeño

Ciudad de La Habana  
Junio de 2008

*Profundizar en el conocimiento científico es una de las mejores vías para lograr plenitud y libertad.*

*- Pilar Álvarez Pellicero*

## Declaración de autoría

Por este medio declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los \_\_ días del mes de \_\_ del 2008.

### Autores

---

Judith Carrodegua Argote

---

Henry Ricardo Labañino

### Tutores

---

MSc. David Leyva Leyva

---

MSc. Daymy Tamayo Avila

---

Ing. Dunia María Colomé Cedeño

## *Agradecimientos*

*De Judith:*

A mis padres por todo su apoyo, por sus consejos y preocupaciones.

A mi abuelita y mi tío Tito por preocuparse siempre por mí y apoyarme en todo lo que me hizo falta.

A mi hermana Juliette que aunque lejos supo darme fuerzas y consejos para que todo saliera adelante.

A Guille por estar a mi lado y brindarme su amor.

A mis compañeros de proyecto Rafael, Anny, Rolando y Ernesto que estuvieron en todo momento dispuestos a ayudarme.

A mis amigas Sandra y Patricia por compartir estos años de estudio y por todos los momentos felices y tristes que pasamos juntas.

A Yunieski, Sissi y Dunel por ayudarme en todo lo que me hizo falta para la realización de este trabajo.

A mis tutores por servirme de guía.

A mi tutora Dunia en especial porque siempre estuvo presente para brindarme toda su ayuda.

A mi compañero de tesis Henry.

A todas las personas que a lo largo de mi vida y carrera me han apoyado, principalmente a todos los que han aportado un granito de arena en la realización de este trabajo.

## *Agradecimientos*

*De Henry*

A mi mamá por ser mi principal motivación para hacer todo esto.

Al resto de mi familia, especialmente a mis tías Arelis, Estela y Odisa, (por orden alfabético).

A mi amiga Yohandra Antón Medina.

A mi amigo Vladimir Guerra Miller, por ser mi hermano en las buenas y en las malas.

A mi amigo Raudel Cuza Sánchez.

Al resto de mis amigos, en especial a Lester, Heiner, Alioscha, Benjamín, César y Yolanda, conocerlos fue lo mejor de estar en la universidad.

Al profesor de Matemática del Instituto Pre Vocacional de Ciencias Exactas José Martí Pérez de Holguín, César Leyva, por ser el mejor profesor de toda mi carrera de estudiante, un educador y un hombre de verdad.

A mi compañera de tesis Judith, por ser tan especial.

*Dedicatoria*

*Judith*

*A mi querida y preciosa abuelita Cuca por brindarme confianza y darme todo  
su apoyo.*

*A mi mamita por estar al tanto de que en mis estudios todo marchara bien.*

*Henry*

*A mi hermano Erick, por ser la persona que más quiero en el mundo.*

## Resumen

En la actualidad el proceso de enseñanza-aprendizaje a distancia exige sistemas que sean rápidos, flexibles e intuitivos; para lo cual se utilizan las ventajas de las Tecnologías de la Informática y las Comunicaciones (TICs). La unión de ambos conceptos da surgimiento al e-learning, contexto en el cual los contenidos educativos tienen un papel esencial, siendo los Objetos de Aprendizaje la tendencia actual para el diseño de los mismos.

Para la creación y edición de los mismos, es necesario la utilización de Herramientas de Autor (HA). El objetivo de este trabajo es realizar el análisis y diseño de una HA capaz de crear OA, que contemplen las especificaciones de los estándares Information Modular Systems Content Packaging (*IMS CP*) y Sharable Content Object Reference Model (*SCORM*), además de poseer características que le permitan interactuar con otras aplicaciones que conforman un sistema e-learning (como pueden ser repositorios y plataformas LMS), utilizando mecanismos de interoperabilidad.

Para el diseño de la Herramienta de Autor Web se propone utilizar la metodología de desarrollo RUP, el lenguaje de modelado UML y la herramienta CASE Visual Paradigm. Para su desarrollo se define herramientas libres, como son PostgreSQL como gestor de base de datos relacional; Apache como servidor Web; PHP, HTML, JavaScript y CSS como lenguajes de programación.

Palabras Clave:

Interoperabilidad; Objetos de Aprendizaje, SCORM, IMS CP, e-learning, Herramienta de Autor.

---

# Índice

Introducción .....	1
Capítulo 1: Fundamentación Teórica .....	6
Introducción .....	6
1.1 El e-learning como variante educativa.....	6
1.1.1 Plataformas para el acceso a e-learning.....	7
1.2 Objetos de aprendizaje .....	10
1.3 Estándares para la creación de objetos de aprendizaje .....	11
1.3.1 Organizaciones que desarrollan estándares.....	13
1.3.2 Estándares para empaquetar OA .....	14
1.3.2.1 Estándar IMS Content Packaging .....	15
1.3.2.2 Estándar SCORM.....	16
1.3.3 Estándares para describir OA.....	17
1.3.3.1 Estándar Learning Object Metadata (LOM) .....	18
1.3.3.2 Estándar Dublin Core .....	21
1.4 Estándares para la interoperabilidad .....	22
1.4.1 OKI (Open Knowledge Initiative) .....	23
1.4.2 IMS-DRI (IMS Digital Repository Interoperability) .....	24
1.4.3 SQI (Simple Query Interface) .....	24
1.5 ¿Qué es la Arquitectura Orientada a Servicios?.....	25
1.6 Servicios Web .....	25
1.6.1 Lenguajes y protocolos asociadas.....	26
1.7 ¿Qué es una herramienta de autor?.....	27
1.8 Lenguajes de Programación para la Web.....	30
1.9 Tecnologías del lado del cliente .....	32
1.10 Servidores Web.....	34
1.11 Gestores de Bases de Datos.....	34
1.12 Metodologías de desarrollo de SW.....	36
1.12.1 Herramientas para la Modelación Visual del Sistema.....	39
1.13 Conclusiones.....	40
Capítulo 2: Características del Sistema .....	41
2.1 Introducción .....	41
2.2 Propuesta actual .....	41
2.3 Modelo de Dominio .....	42
2.3.1 Descripción del Problema de Dominio.....	43
2.3.2 Definición de las entidades y los conceptos principales.....	44
2.3.3 Representación del modelo de dominio.....	45
2.4 Especificación de los requisitos de software. ....	45
2.4.1 Requerimientos Funcionales .....	45
2.4.2 Requerimientos no Funcionales .....	48
2.5 Definición de los casos de uso y actores.....	49
2.5.1 Definición de los actores del sistema.....	49
2.5.2 Casos de uso del sistema .....	49
2.5.3 Diagrama de casos de uso del sistema .....	54
2.5.4 Descripción de los Casos de Uso del Sistema.....	54
2.6 Conclusiones .....	76
Capítulo 3: Análisis y Diseño .....	77
3.1 Introducción .....	77
3.2 Análisis .....	77
3.2.1 Diagramas de clases del análisis .....	77

3.3 Diseño.....	82
3.3.1 Diagrama de Clases del Diseño. ....	84
3.4 Diseño de la Base de Datos .....	84
3.4.1 Modelo de Datos .....	99
3.4.2 Descripción de las tablas de la Base de Datos .....	100
3.5 Patrones de diseño .....	103
3.6 Conclusiones .....	103
Conclusiones .....	104
Recomendaciones .....	105
Bibliografía.....	106
Glosario de Términos:.....	110

---

## Índice de Tablas y Figuras

Figura 1. Diagrama del Modelo de Dominio .....	45
Tabla 1. Definición de los actores del sistema. ....	49
Figura 2. Diagrama de Casos de Uso del sistema. ....	54
Tabla 2. Descripción Textual <CU Autenticar usuario> .....	54
Tabla 3. Descripción Textual <CU Gestionar usuario>.....	55
Tabla 4. Descripción Textual <CU Gestionar rol > .....	58
Tabla 5. Descripción Textual <CU Gestionar perfil >.....	59
Tabla 6. Descripción Textual <CU Mostrar reportes>.....	59
Tabla 7. Descripción Textual <CU Configurar tipo de autenticación>.....	61
Tabla 8. Descripción Textual <CU Configurar HTTPs > .....	62
Tabla 9. Descripción Textual <CU Gestionar mensajes > .....	62
Tabla 10.Descripción Textual <CU Administrar interfaz gráfica >.....	64
Tabla 11.Descripción Textual <CU crear paquete de contenidos >.....	66
Tabla 12.Descripción Textual <CU gestionar paquete > .....	68
Tabla 13.Descripción Textual <CU Gestionar metadatos >.....	69
Tabla 14.Descripción Textual <CU Mostrar vista previa del paquete > .....	70
Tabla 15.Descripción Textual <CU Generar paquete >.....	70
Tabla 16.Descripción Textual <CU Descargar paquete >.....	71
Tabla 17.Descripción Textual <CU Crear contenido > .....	71
Tabla 18.Descripción Textual <CU Gestionar contenido > .....	73
Tabla 19.Descripción Textual <CU Exportar paquete >.....	73
Tabla 20.Descripción Textual <CU Recuperar contenido >.....	74
Tabla 21.Descripción Textual <CU Configurar metadatos >.....	75
Tabla 22.Descripción Textual <CU Parametrizar variables de configuración > .....	76
Figura 3. Modelo de datos (Entidad-Relación) .....	99
Figura 4. Modelo de datos (Entidad-Relación) .....	100

## Introducción

Actualmente los centros de educación superior de todo el mundo, adoptan la educación virtual como solución a la creciente demanda de conocimientos que el desarrollo provoca en la sociedad. Para la adopción de este tipo de enseñanza se utilizan las TICs en todo el proceso de instrucción y recepción de los conocimientos. Es por eso que es necesario el uso de una estrategia que combine las nuevas tecnologías con las corrientes educativas más eficaces de la actualidad. De esta manera surge el e-learning, que es un *“conjunto de tecnologías, aplicaciones y servicios orientados a facilitar la enseñanza y el aprendizaje a través de Internet/Intranet, que facilitan el acceso a la información y la comunicación con otros participantes”* (Red TTnet, 2005).

Para facilitar la enseñanza y el aprendizaje a distancia es necesario la utilización de un sistema software-hardware encargado del manejo o administración de los contenidos que forman parte de cada programa educativo. Como solución a esta necesidad surgen los Learning Management System (LMS), que son sistemas que posibilitan la gestión de los recursos, el libre acceso a los contenidos y la interacción entre todos los actores vinculados. Las plataformas LMS normalmente controlan los accesos, actividades y permisos de los usuarios (inscripción, control de contenidos visitados, resultados de evaluaciones, creación de informes y estadísticas) y proporcionan herramientas de comunicación tanto síncronas (ejemplo: Chat, videoconferencia, tutorías en tiempo real, etc.) como asíncronas (ejemplo: foros de discusión, tableros de anuncios, etc.).

Entre los LMS de libre distribución más difundidos está la plataforma Moodle<sup>1</sup>, cuyo diseño está basado según las ideas del constructivismo<sup>2</sup>, y no es más que un paquete de software para la creación y gestión de cursos y sitios Web basados en Internet. Una de las desventajas de estas plataformas es que manejan como mínima unidad didáctica el curso. Sin embargo, la tendencia en la actualidad es granular los contenidos en pequeñas unidades llamados Objetos de Aprendizaje (OA), originados bajo el paradigma orientado a objetos, y que tienen la finalidad de maximizar el número de contextos en que puedan ser utilizados (TAMAYO 2007).

En la actualidad existe una gran variedad de conceptos referente a los OA, en este trabajo se define como *“cualquier recurso con una intención formativa, compuesto de uno o varios elementos digitales,*

---

<sup>1</sup> Moodle: Modular Object-Oriented Dynamic Learning Environment (Entorno de Dinámico Aprendizaje Modular Orientado a Objetos)

<sup>2</sup> En pedagogía se denomina **Constructivismo** a una corriente que afirma que el conocimiento de todas las cosas es un proceso mental del individuo, que se desarrolla de manera interna conforme el individuo obtiene información e interactúa con su entorno. O sea, el individuo participa en la creación de su propio conocimiento.

*descrito con metadatos, que pueda ser utilizado y reutilizado dentro de un entorno e-Learning”* (LÓPEZ 2005).

Para su utilización, los OA deben tener características tales como: ser fáciles de migrar a otras plataformas, de localizar, almacenar, reutilizar y acceder, además de ser compatibles con diversos ambientes. Para darle cumplimiento a estos requerimientos, los OA son reunidos y almacenados en repositorios de objetos de aprendizaje (ROA). Con el objetivo de efectuar una futura búsqueda y recuperación de los OA en estos repositorios, se utilizan metadatos, los cuales describen de un modo específico la información contenida en el paquete educativo.

En el proceso de producción de un curso a distancia basado en el uso de las TIC, convergen conocimientos y habilidades que tienen su origen en el campo pedagógico e informático por lo cual resulta bastante complejo la publicación de un curso, exigiendo del profesor nuevos aprendizajes y cambios en su modo de actuación. Esta dificultad puede ser reducida con el empleo de un software que automatiza parte del proceso, llamadas Herramientas de Autor (HA). Las HA son necesarias para la creación de OA. Las mismas permiten conferirle características que posibilitan su reutilización al basarse en estándares.

Dentro de los entornos e-learning existen diferentes intereses y objetivos, sistemas informáticos con funciones y tecnologías variadas, donde un reto es la interoperabilidad. Este término define la posibilidad de que las plataformas que componen el sistema trabajen de forma conjunta y exista un flujo de datos transparente para el usuario. Para garantizar dicha interoperabilidad, además de la reusabilidad, durabilidad y accesibilidad de los OA es necesario apegarse a los estándares y a las especificaciones propuestas por importantes grupos de organizaciones internacionales como Aviation Industry Computer-Based Training Comitee (AICC), IMS Global Consortium Inc., Advanced Distributed Learning (ADL), Institute of Electrical and Electronics Engineers/Learning Technology Standard Committee (IEEE/LTSC), World Wide Web Consortium (W3C), entre otros.

Entre los estándares a los que se hace mención, se encuentra SCORM<sup>3</sup> que es un modelo de referencia para el desarrollo e integración de contenidos de formación para ser utilizados en un entorno virtual de aprendizaje, especifica qué estándares y modelos previos se pueden utilizar y cómo hacerlo. SCORM utiliza como modelo de empaquetamiento el Content Packaging Specification de IMS Global Learning Consortium<sup>4</sup> y como modelo de metadatos el Learning object metadata (LOM) de IEEE

---

<sup>3</sup> SCORM: Shareable Content Object Reference Model Initiative

<sup>4</sup> Organización sin ánimo de lucro cuya misión es desarrollar y promover especificaciones abiertas para facilitar las actividades del aprendizaje on-line.

Learning Technology Standards Committee (LTSC)<sup>5</sup>.

Existen otros estándares que tienen como objetivo conferir interoperabilidad entre las aplicaciones que conforman un sistema e-learning. Algunos de los más utilizados son el Z39.50, Open Knowledge Initiative (OKI) Simple Query Interface (SQI), IMS Digital Repositories Interoperability (IMS-DRI), The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). Además del uso de estos estándares, como mejora a la interoperabilidad surgen los servicios Web y las Arquitecturas Orientadas a Servicios (SOA), que plantean nuevos mecanismos de comunicación posibilitando que puedan interactuar sistemas y aplicaciones desarrollados con diferentes lenguajes de programación y en diversas plataformas.

Actualmente, existen varias herramientas de autor a disposición de profesores y educadores para la creación de paquetes de OA en los entornos de enseñanza-aprendizaje a través de Internet. Frente a la proliferación de estas herramientas, uno de los problemas es determinar cuáles herramientas se adaptan mejor a la solución de los objetivos educativos específicos de cada entorno en que se apliquen. Entre ellas se pueden mencionar el eXeLearning, ReLOAD y Lompad que son de libre distribución, basadas en licencias de código abierto, pero en su mayoría estas herramientas son aplicaciones Desktop, obligando al usuario a su previa instalación. Además el diseño de los entornos de trabajo son poco usables, ocasionando dificultades a usuarios inexpertos (COSS 2006).

Existen también herramientas de autor Web, una de ellas es Autore, que tiene como limitación que los objetos de aprendizaje tienen una estructura rígida, no son recursos flexibles. Otra herramienta es el proyecto alemán Weload que imita las funcionalidades de ReLOAD (subir paquetes SCORM, dibujar el árbol de contenidos y lanzar los contenidos). Presenta como desventajas la utilización de un servidor Web apache portátil, que impide el acceso por la red a los recursos contenidos en esta herramienta; además de incorporar muy pocos metadatos. En la Universidad de las Ciencias Informáticas (UCI) se ha implementado una herramienta denominada ROXS, que no describe todos los metadatos de LOM, solamente incorpora SCORM y no implementa estándares para lograr la interoperabilidad a nivel de sistema con otras aplicaciones.

A partir de lo expuesto anteriormente se plantea el **problema científico** ¿Cómo garantizar la creación de objetos de aprendizaje en un ambiente interoperable de e-learning?

---

<sup>5</sup> Organismo creado con el fin de preparar normas técnicas, prácticas y guías recomendadas para el uso informático de componentes y sistemas de educación

Este problema se enmarca en el **objeto de estudio** creación de objetos de aprendizaje y el **campo de acción** las tecnologías que posibilitan la interoperabilidad de los OA y de los sistemas que los manejan.

De acuerdo al problema planteado anteriormente se propone como **objetivo** modelar una herramienta de autor Web basada en código abierto que permita la creación de OA de forma estandarizada, garantizando mecanismos de interoperabilidad para su posterior utilización en otros sistemas.

**Idea a defender:** La creación de OA reutilizables puede lograrse mediante la utilización de una herramienta de autor Web interoperable.

Para dar cumplimiento al objetivo se han planteado las siguientes **tareas**:

- Analizar los principales proyectos relacionados con la creación y estandarización de OA a nivel nacional e internacional.
- Realizar un estudio de los estándares utilizados para la creación de objetos de aprendizaje en la actualidad.
- Analizar las principales tecnologías que permiten lograr la interoperabilidad en aplicaciones e-learning.
- Realizar el análisis y diseño de la Herramienta de Autor para la creación de OA reutilizables.

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 actualmente, sus características, ventajas y desventajas. También se utilizan los **métodos empíricos** como la **Observación** ya que mediante Este se puede observar el comportamiento de los sistemas existentes y permite extraer información de su manera de procesar a los OA.

El presente trabajo consta de una introducción, tres 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 OA, metadatos, LCMS y LMS, entre otros. Además se realiza un estudio sobre las principales Herramientas en la actualidad, sus ventajas, funcionalidades, estándares y metadatos que utilizan.

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, modelo de dominio, se determinan los requerimientos funcionales y no funcionales y se describen los casos de uso y los actores del sistema.

En el Capítulo 3 "Análisis y Diseño del Sistema", se detallan, mediante un grupo de artefactos, la solución del sistema hasta el diseño. Para su modelado se utilizan los diagramas de clases del análisis, de interacción, de clases del diseño y el diseño de la base de datos que se necesita para el almacenamiento de la información persistente.

## **Capítulo 1: Fundamentación Teórica**

### **Introducción**

En este capítulo se profundiza sobre la definición de los aspectos fundamentales para el desarrollo actual del e-learning. Se trata acerca de los estándares más difundidos utilizados en la creación de OA (SCORM e IMS CP). Además se abordan los diferentes estándares que hacen posible la comunicación entre las distintas aplicaciones que conforman un sistema e-learning.

Por otra parte se establece un análisis de las principales herramientas de autor empleadas en Cuba y el mundo con el objetivo de precisar sus ventajas, desventajas y funcionalidades que las diferencian. Además se fundamentan los lenguajes de programación Web, gestores de Bases de Datos y metodologías de desarrollo de software propuestos para el desarrollo de este trabajo.

### **1.1 El e-learning como variante educativa**

El desarrollo de la ciencia y la tecnología ha permitido que nuevos modelos de enseñanza-aprendizaje surjan, propiciando una educación a distancia donde el alumno es el centro de una formación autodidacta y flexible, al tener que gestionar su propio aprendizaje.

Siguiendo esta óptica surge el e-Learning, que como se menciona anteriormente es un “conjunto de tecnologías, aplicaciones y servicios orientados a facilitar la enseñanza y el aprendizaje a través de Internet/Intranet, que facilitan el acceso a la información y la comunicación con otros participantes.” (Red TTnet, 2005).

Los tres elementos fundamentales de una solución e-learning son: Plataforma, Contenidos y Herramientas comunicativas.

El elemento plataforma es un entorno de hardware y software diseñado para automatizar y gestionar el desarrollo de actividades formativas (ATICA\_SOCRATES 2007). Una de las dos plataformas más utilizadas en el ámbito universitario son la WebCT y la plataforma de licencia libre Moodle.

En los contenidos, la calidad es una condición necesaria para el éxito del programa formativo. El diseño de los contenidos debe de ser realizado con el objetivo de que respondan a las adecuadas

necesidades y posibilidades del alumno, calidad y cantidad de la información, interactividad y una estructura adecuada para su correcta asimilación.

Las herramientas de comunicación constituyen otro pilar de la solución e-learning, ya que posibilitan la interacción, es decir la capacidad de realizar trabajos en grupo, intercambiar experiencias, resolución de dudas, etc., entre los diferentes agentes del proceso de enseñanza-aprendizaje.

Entre las funcionalidades que tiene el e-learning se encuentran la autenticación al sistema, generación de contenidos, visualización de contenidos, realización de actividades como tareas, trabajos en grupo, reporte de las actividades realizadas por el alumno y herramientas de evaluación.

Una de las ventajas principales que ofrece el e-learning es la flexibilidad de horarios, factor de suma importancia pues permite al estudiante organizar el curso de la mejor forma posible y el interés del estudiante, ayudando a aquellos tímidos a ser de los más activos en clases por medio de foros de discusión y otros medios de participación.

### **1.1.1 Plataformas para el acceso a e-learning**

En este nuevo status que ha adquirido el modelo de enseñanza-aprendizaje, basado en la utilización de herramientas y recursos tecnológicos, se están utilizando, cada vez con más rapidez, las plataformas tecnológicas capaces de brindar soporte al aprendizaje, conocidas como “plataformas e-learning”. Por ejemplo los LMS (Sistemas de Gestión de Aprendizaje) y LCMS (Sistemas de Gestión de Contenido de Aprendizaje), de los cuales se hablará más adelante.

*LMS: Learning Management System o Sistema Gestor del Aprendizaje:*

El LMS es un sistema que está enfocado precisamente al área educativa, permitiendo llevar un control tanto sobre los contenidos como de los distintos usuarios que interactúan dentro de él. Su principal objetivo es la gestión del estudiante, la evolución de sus estudios y el rendimiento a través de las actividades que desarrolla.

Hacen más sencilla la interacción entre los profesores y los estudiantes, sin la necesidad forzosa de que estén presentes físicamente, aportan herramientas para la gestión de contenidos y permiten el seguimiento y la valoración de los estudiantes. Es decir, facilitan una simulación avanzada del modelo

real en el mundo virtual.

Las principales funciones del LMS son: gestionar usuarios, recursos y actividades de formación, administrar el acceso, controlar y hacer seguimiento del proceso de aprendizaje, realizar evaluaciones, generar informes, gestionar servicios de comunicación como foros de discusión, videoconferencias, entre otros.

Un LMS generalmente no incluye posibilidades de autoría (crear sus propios contenidos), pero se centra en gestionar contenidos creados por fuentes diferentes. Los contenidos cargados son creados de manera externa con alguna herramienta de autor.

Entre los LMS más conocidos está MOODLE, que incluye como características la administración de profesores y alumnos; aulas virtuales que contienen toda la información de un curso y permiten la comunicación mediante foros y chat; creación, mantenimiento y publicación del material de un curso, con soporte de varios formatos, incluidos audio y vídeo; exámenes y tests, trabajos con fecha de límite de entrega y gestionar servicios de comunicación como foros de discusión, videoconferencias, entre otros.

La gran mayoría de los LMS están basados en la Web con el fin de facilitar en cualquier momento y en cualquier lugar, acceso a los contenidos de aprendizaje.

A través de sus distintos tipos de herramientas, el LMS permite que:

- Los profesores coloquen a disposición de los alumnos los objetivos del curso, su contenido y su reglamentación.
- Los profesores supervisen el desarrollo del curso y el avance de cada alumno.
- Los alumnos accedan a los contenidos, realicen la ejercitación prevista, se comuniquen entre sí y con el profesor, para resolver dudas y realizar trabajos en grupo.
- Los administradores obtengan información del progreso del curso y de las acciones administrativas relacionadas, tales como inscripción de alumnos, historial de cursos, etc.

*LCMS: Learning Content Management System o Sistemas de Gestión de Contenido de Aprendizaje:*

Es un software para la gestión automatizada de cursos en línea, es decir crea los contenidos para dichos cursos y además incluye gestión de usuarios y de recursos.

Los LCMS integran las utilidades de los CMS<sup>6</sup> y los LMS, de ahí que proporcionen una mayor robustez; por lo general es un LMS al que se le ha agregado el módulo o funcionalidad de crear contenidos dentro de él.

Es un entorno multiusuario de aprendizaje donde, tanto los desarrolladores como los usuarios pueden llevar a cabo actividades como crear, almacenar, reutilizar, administrar y entregar contenidos digitales destinados al aprendizaje; es decir que un LCMS no es más que él que administra el proceso de creación, edición, almacenamiento y entrega de contenidos de aprendizaje electrónico.

Algunas de sus características:

1. Se basan en un modelo de objetos de contenido.
2. El contenido es reutilizable a lo largo de cursos.
3. El contenido no está ligado a un template único y se puede publicar en diversos formatos.
4. El contenido se almacena en una base de datos centralizada.
5. Son herramientas sencillas que facilitan la creación de contenidos, en forma de herramientas de autor en el sistema.
6. Sistemas flexibles de diseño y distribución de los cursos que permitan adaptarse a las necesidades y ritmos de aprendizaje del usuario.
7. La capacidad de reutilizar el contenido de aprendizaje, apoyándose en un modelo de objetos de aprendizaje.
8. Herramientas para evaluar durante el curso del aprendizaje del alumno.
9. Tienen la capacidad de interactuar con otros LCMS y LMS.
10. Aumentan la disponibilidad de materiales docentes y de su facilidad de adquisición.
11. La posibilidad de generación de bases de datos que posibiliten futuras investigaciones metodológicas, mediante técnicas estadísticas clásicas o procedimientos de extracción de conocimiento en bases de datos.
12. La actualización constante de los contenidos por el profesor.
13. La rápida creación de contenido para su utilización inmediata.
14. La posibilidad de evaluar el desempeño del estudiante de manera más precisa.

Después de explicar las principales características que definen las plataformas para el acceso a e-learning se hace necesario detallar de manera más profunda, sobre los diferentes aspectos de las unidades que las mismas manipulan: los Objetos de aprendizaje.

---

<sup>6</sup> CMS: Content Management System o Sistema de Manejo de Contenido.

## **1.2 Objetos de aprendizaje**

En el terreno de la enseñanza los objetos de aprendizaje son una tecnología instruccional, es decir, sirven para que los alumnos aprendan; dicha tecnología está basada en el paradigma de la programación orientado a objetos, el cual se refiere a crear componentes educativos reutilizables de tal manera que los objetos de aprendizaje puedan ser reutilizados en diferentes contextos de aprendizaje.

Observando los principales aspectos de los OA, estos, para su construcción, pueden ser seccionados en dos aspectos: el pedagógico y el tecnológico. La manera de diseñar el contenido de los OA, la forma en cómo va a interactuar el alumno con él, así como el establecimiento de un nivel de agregación y dificultad para obtener el impacto cognitivo adecuado, son comprendidos por el aspecto pedagógico. Por otra parte, el aspecto tecnológico, encapsula la construcción del OA, mediante la utilización de herramientas computacionales capaces de brindar características como su compatibilidad con otras plataformas, y aquellas que le permitan ser intercambiable e interoperable entre diversos LMS o repositorios (VELÁZQUEZ 2005). Otras características no menos importantes son las que posibilitan su fácil administración, localización y búsqueda.

Algunas de las ventajas de los OA son:

- Personalización: El diseño de materiales formativos en forma de OA hace más sencilla la personalización del contenido, permitiendo la recombinación de materiales, acorde con las necesidades formativas del colectivo o de los individuos a quienes van dirigidos.
- Flexibilidad: Un material creado para poder usarse en múltiples contextos puede ser reutilizado mucho más fácilmente que un material que ha de ser reelaborado para cada nuevo contexto. Este tipo de material formativo también puede ser actualizado, indexado y gestionado de forma mucho más sencilla.
- Incremento del valor del contenido: Con la reutilización de un contenido viene implícito el incremento de su valor, ya sea por el aumento del potencial de su explotación futura o por la amortización sucesiva de su coste de desarrollo.

Características obligatorias en todo OA:

- Han de ser una unidad duradera y capaz de soportar cambios tecnológicos sin necesidad de volver a ser rediseñada.
- Reutilizables y flexibles para incorporar componentes formativos desde diversas aplicaciones.

- Granular e indivisible, de forma que no se pueda subdividir en unidades más pequeñas que conserven un significado propio.
- Independiente de otras unidades de aprendizaje y tener sentido en sí mismo.
- Accesible dinámicamente a través de una base de datos.
- Interoperables, es decir, deben contar con una estructura basada en un lenguaje de programación XML, y contar con un estándar internacional de interoperabilidad de modo que los componentes formativos pueden ser utilizados en distintas plataformas y soportes.
- Compatibles con diversos ambientes y sistemas de administración de aprendizajes, fáciles de migrar de una plataforma a otra, fáciles de localizar, acceder, archivar y reutilizar.(VARAS 2002).

Para realizar ésta nueva forma de crear contenidos (consiste en diseñar los cursos como agregados de OA), y a causa de la heterogeneidad de plataformas educativas y de los sistemas de enseñanza en línea o LMS, se hace necesaria la existencia de estándares y recomendaciones de gran aceptación que brinden facilidades para la reutilización de los OA y su interoperabilidad. (MANJÓN 2006).

### 1.3 Estándares para la creación de objetos de aprendizaje

*Un estándar es un patrón, una tipificación o una norma de cómo realizar algo.* (AULAGLOBAL 2005). Gracias a estos se pueden llevar a cabo la interoperabilidad y la reutilización pues permiten el intercambio directo de OA y de cursos completos entre distintas plataformas y sistemas de enseñanza electrónica.

El conjunto de especificaciones y estándares de e-learning tienen por objetivo facilitar todos los procesos asociados para hacerlo de una forma más eficiente y sistemática. Con el fin de tratar de normalizar aspectos como la descripción (mediante metadatos) de los OA de modo que puedan ser gestionados, indexados y clasificados; su almacenamiento en bases de datos (denominados repositorios) o la descripción de un curso completo.(MANJÓN 2006)

Dichos estándares también permiten incorporar contenidos de distintos proveedores en un solo programa de estudios. Por otra parte estas reglas definen un modelo de empaquetamiento estándar para los contenidos. Los contenidos pueden ser empaquetados como OA, de tal forma que permitan a los desarrolladores crear contenidos que puedan ser fácilmente reutilizados e integrados en distintos cursos.

También se obtiene independencia de la plataforma de modo que se podrá cambiar de equipo informático o de sistema operativo sin problemas.(MANJÓN 2006) Por último, se obtiene una garantía en la durabilidad en los contenidos, ya que al seguir un estándar y no un formato propietario de una compañía se logra la seguridad de que perduren a lo largo del tiempo. (GUTIÉRREZ 2006).

Se desarrollan hoy en día para la industria del e-learning, diferentes estándares que se pueden clasificar en diversos parámetros, por ejemplo:

- Sobre el alumno
  - Almacenamiento e intercambio de información del alumno.
  - Habilidades del alumno.
  - Privacidad y seguridad.
  
- Sobre la interoperabilidad:
  - Integración de componentes del LMS.
  - Interoperabilidad entre múltiples LMS.
  
- Sobre el contenido o el curso:
  - Estructuras de contenidos.
  - Empaquetamiento de contenidos.
  - Seguimientos de resultados.

Finalmente, los estándares permiten crear tecnologías de aprendizaje más poderosas y personalizar el aprendizaje basándose en las necesidades individuales de los alumnos. Básicamente, lo que se persigue con la aplicación de un estándar para el e-learning es lo siguiente:

- Durabilidad: que la tecnología desarrollada con el estándar sea duradera y que evite que, de forma rápida, los cursos se vuelvan obsoletos.
- Interoperabilidad: que sea posible intercambiar información entre los diferentes LMS.
- Accesibilidad: que se permita un seguimiento de los progresos de los alumnos.
- Reusabilidad: que los diferentes cursos y objetos de aprendizaje puedan ser reutilizados con diferentes herramientas y en distintas plataformas.

### 1.3.1 Organizaciones que desarrollan estándares

Los responsables para desempeñar el papel de implementar estándares, deben tener un respaldo de la comunidad del medio en el que se desenvuelven, además de ofrecer seguridad y garantía de que los estándares que se dicten favorezcan a la mayor cantidad de usuarios y creadores del producto. Es así como surgen las asociaciones encargadas de analizar y publicar las normas que deben seguir los creadores de los productos para su óptima distribución y utilización por los destinatarios. Para el e-learning existen varias organizaciones responsables de desarrollar propuestas para la estandarización. (LÓPEZ 2005) identifica las siguientes:

- *AICC (Aviation Industry Computer-Based Training Comitee)*. Es una asociación de entrenamiento profesional y fue el primer organismo creado para elaborar un conjunto de normas que permitiese el intercambio de cursos entre diferentes sistemas, de este modo garantizan la armonización de los requerimientos de los cursos, así como la homogenización de los resultados obtenidos de los mismos. Las especificaciones del AICC cubren nueve áreas principales, que van desde los OA hasta los LMS. La AICC cuenta con un programa de certificación y dispone de un test suite que le permite a las compañías verificar que sus productos sean compatibles con otros sistemas que cumplen con las especificaciones AICC.

Se reconoce como una de las precursoras de la estandarización de materiales para entrenamiento profesional<sup>7</sup>.

- *ADL (Advanced Distributed Learning)*. La misión de ADL es desarrollar principios y guías de trabajo necesarias para el desarrollo y la implementación eficiente, efectiva y en gran escala, de formación educativa sobre nuevas tecnologías Web, además de proveer acceso de la más alta calidad en educación y entrenamiento, pues este organismo recogió lo mejor de las anteriores iniciativas (el sistema de descripción de cursos en XML de la IMS, y el mecanismo de intercambio de información mediante una API de la AICC), las refundió y mejoró en su propio estándar: SCORM (Shareable Content Object Referente Model)<sup>8</sup>.

- *IMS Global Consortium Inc.* es un Consorcio que está formado por miembros provenientes de organizaciones educacionales, empresas públicas y privadas dedicadas a definir y distribuir arquitecturas abiertas para actividades de educación en línea. Su misión es desarrollar y promover especificaciones abiertas para facilitar las actividades del aprendizaje online. Su objetivo fue la

---

<sup>7</sup> <http://www.aicc.org/>

<sup>8</sup> <http://www.adlnet.org/>

creación de un formato que pusiese en práctica las recomendaciones de la IEEE y la AICC. Lo que se hizo fue definir un tipo de fichero XML para la descripción de los contenidos de los cursos. De tal modo que cualquier LMS pueda, leyendo su fichero de configuración IMSMANIFEST.XML, cargar el curso. Uno de sus resultados es lo que se conoce como el estándar IMS<sup>9</sup>.

- *IEEE/LTSC (Institute of Electrical and Electronics Engineers/Learning Technology Standards Committee)*. Se trata de un organismo que promueve la creación de una norma ISO, una normativa estándar real de amplia aceptación. El LTSC se encarga de preparar normas técnicas, prácticas y guías recomendadas para el uso informático de componentes y sistemas de educación y de formación, en concreto, los componentes de software, las herramientas, las tecnologías y los métodos de diseño que facilitan su desarrollo, despliegue, mantenimiento e interoperación. Lo que hizo fue recoger el trabajo del comité de la AICC y mejorarlo<sup>10</sup>.

- *W3C (World Wide Web Consortium)*. Se encarga del desarrollo de tecnologías interoperables (especificaciones, normas, software y herramientas) para aprovechar todo el potencial del Web. Aunque no está directamente vinculado con el desarrollo del e-learning es importante mencionarlo, ya que de la interoperabilidad de la Web dependen muchas de las funciones de la educación en línea<sup>11</sup>.

Como resultado del trabajo de las organizaciones mencionadas anteriormente se han adoptado un grupo de estándares. Entre los más utilizados se encuentran el IMS y el SCORM, de los que se hablarán mas adelante.

### 1.3.2 Estándares para empaquetar OA

Los estándares e-learning son el vehículo a través del cual será posible dotar de flexibilidad a las soluciones e-learning tanto en contenido como en infraestructura.(MAURER 2004). Ellos han abierto una manera más coherente de empaquetar recursos y contenidos tanto para los estudiantes como para los desarrolladores.

Los estándares de empaquetamiento de contenidos de aprendizajes tienen gran importancia, en particular en las plataformas de código abierto con relación al diseño, creación y reusabilidad de contenido.

---

<sup>9</sup> <http://www.adlnet.org/>

<sup>10</sup> <http://www.ieee.org/>

<sup>11</sup> <http://www.w3.org/>

El empaquetado de contenidos señala la descripción, estructura y localización de los materiales de aprendizaje en la red y la definición de algunos tipos particulares de contenido.

### 1.3.2.1 Estándar IMS Content Packaging

El IMS CP es una variante de estándar para el empaquetamiento de contenidos promovida por *IMS Global Learning Consortium*.

Esta iniciativa provee la funcionalidad para describir y empaquetar material de aprendizaje, ya sea un curso individual o una colección de cursos, en paquetes portables e interoperables. El empaquetamiento de contenidos está vinculado a la descripción, estructura y ubicación de los materiales de aprendizaje online. El objetivo de IMS content packaging es permitir la distribución de contenidos reutilizables e intercambiables, es decir, describe el modo en el que se debe empaquetar el contenido educativo para que pueda ser procesado por otro sistema LMS diferente.

Se puede decir que un paquete representa una unidad de contenido utilizable y reutilizable que debe ser posible agregarlo o disgregarlo con otros paquetes, además debe contener toda la información necesaria para ser utilizado de forma autónoma, deduciendo así que esto es un objeto de aprendizaje.(LÓPEZ 2005). En SCORM los paquetes tienen una composición similar ya que están basados en esta especificación, pero se les llaman SCO.

Para IMS CP un paquete de contenido está compuesto por dos elementos: manifiesto y contenido. Esta especificación ofrece una forma de empaquetar (en un archivo comprimido tipo .Zip) los contenidos educativos tales como cursos individuales, conjuntos de cursos, o cualquier tipo de recurso necesario en el proceso educativo (por ejemplo, evaluaciones o exámenes). Este manifiesto es un fichero XML en el que se describe la estructura de los contenidos incluidos en el paquete y los organiza, añadiendo información adicional en forma de metadatos que pueden ser procesados y aprovechados en tareas de catalogar contenidos.(CUDI 2006).

La especificación IMS CP se enfoca sólo a la descripción y la estructura de los contenidos y es importante mencionar que no se involucra con los modelos pedagógicos, dejando con esto la libertad de que el profesor utilice el que a su consideración, sea más conveniente en cada aplicación.

### **1.3.2.2 Estándar SCORM**

SCORM (Shareable Content Object Reference Model) es un modelo de referencia para el desarrollo e integración de contenidos educativos para ser utilizados en un entorno virtual de formación; estableciendo un modo de desarrollar, empaquetar y gestionar la distribución de unidades formativas digitales, además de utilizar a los contenidos y a los sistemas para comunicarse con otros sistemas, obteniendo así interoperabilidad, reutilización, durabilidad y adaptabilidad.

La primera versión vio la luz en el año 2000 y se basa en el trabajo desarrollado por el AICC, el IEEE LTSC y el IMS e intenta recoger lo mejor de cada especificación.

Para el análisis de la tecnología del Aprendizaje a Distancia, SCORM lo divide en dos componentes fundamentales: la plataforma de formación (LMS) y los objetos de contenido intercambiables (SCO – *Sharable Content Objects*–). Los SCO constituyen su visión particular de los objetos de aprendizaje. La plataforma es cualquier elemento que almacena información sobre los estudiantes, es capaz de lanzar y de comunicarse con los SCO y puede interpretar las instrucciones que le indican la secuencia correcta entre un conjunto de SCO. (PEDRUELO 2004).

¿Cómo funciona SCORM?

A grandes rasgos, el funcionamiento de SCORM se basa en la agregación de un fichero XML a los ficheros contenedores de los recursos que serán utilizados para impartir la actividad docente. SCORM posibilita la asociación de metadatos a cada recurso utilizado o a un conjunto de recursos como pueda ser una lección, un tema o un curso completo.(PAULE 2006) Dichos metadatos son almacenados en un fichero XML que deberá acompañar al resto de los ficheros que contienen los recursos de la actividad docente.

O sea, la idea es que el usuario del sistema crea los objetos de aprendizaje, les da una estructura que piensa que facilita el aprendizaje y lo empaqueta en un único fichero. Este paquete se almacena en un repositorio para su posterior utilización o pueden ser distribuidos por la red y para que no se pierda la organización que le dio el autor, va acompañado del fichero XML en el cual queda reflejado el contenido y el orden o secuencia que se puede seguir para lograr los conocimientos. (GIL 2005).

La ventaja de utilizar estos metadatos consiste en que se puede almacenar casi cualquier información que describa el material docente. Esto permite llevar a cabo actividades tan importantes como buscar y seleccionar el material que se quiera para utilizarlo en posteriores actividades docentes, además de escoger que partes del mismo se quiere presentar a cada alumno.

Ofrece una flexibilidad elevada a la hora de adaptar la enseñanza al perfil de cada alumno. Por otra parte brinda gran facilidad en la reutilización de contenidos que hayan sido desarrollados anteriormente.

La especificación SCORM cuenta básicamente con tres componentes:(FOIX 2002)

1. El modelo de agregación de contenido (*Content Aggregation Model*), que asegura la coherencia en el formato y el conjunto de procedimientos en materia de almacenamiento, identificación, condicionamiento de intercambios y recuperación de contenidos; está compuesto por:
  - a) el modelo de contenido: formado por tres tipos de componentes: los elementos o “assets” (textos, imágenes, audio, etc.), los SCOs y el agrupamiento de contenidos.
  - b) los metadatos.
  - c) el empaquetado de contenidos: se emplea para agrupar los distintos contenidos educativos y la descripción de la estructura de estos contenidos.
2. Una especificación común para el ambiente en tiempo de ejecución o entorno de ejecución (*RTE-Run-Time Environment*) que describe los requisitos que debe implementar el sistema de gestión del aprendizaje para facilitar la comunicación entre LMS y los componentes de contenido de los cursos.
3. El modelo de secuenciación y de navegación, describe cómo debe producirse la secuencia de los contenidos almacenados en los SCO a través de una serie de eventos de navegación que pueden ser provocados por el estudiante o a iniciativa del propio sistema; las posibles ramificaciones de los contenidos y los flujos que describen los posibles recorridos se establecen habitualmente durante el diseño.

Por otra parte, aunque SCORM ha ido actualizando versiones y ha extendido sus funciones, su alcance es limitado y en su modelo solo cubre el empaquetamiento y la comunicación del recurso con el LMS, lo que hace su entendimiento e implementación mucho más sencilla que la de IMS, quizás este sea el motivo por el que es el más ampliamente utilizado hoy día para el intercambio de paquetes entre plataformas. (ADL, 2002).

### 1.3.3 Estándares para describir OA

La capacidad de intercambiar, distribuir y reutilizar los objetos de aprendizaje se hace posible gracias a la utilización de esquemas de metadatos. Los mismos son importantes ya que sirven para localizar,

identificar y describir el contenido en un documento Web y por ende para recuperar información a través de la red. No obstante los modelos de metadatos no son normas, mas bien responden a proyectos de desarrollo de normas o a estándares en determinadas comunidades de usuarios.(VEGA 2001).

Actualmente se han desarrollado distintos proyectos de normalización de la información tanto para la identificación de los recursos como para su acceso, estructuración y referencia, algunos de los cuales se explican a continuación.

### 1.3.3.1 Estándar Learning Object Metadata (LOM)

Como se menciona anteriormente, el hecho de apegarse a normas o estándares a la hora de asociar metadatos a OA implica la ventaja de su gran aceptación en una comunidad amplia de consumidores y productores de material educativo. Es por ello que, desde la comunidad de e-learning, se han realizado distintos esfuerzos para estandarizar los esquemas de metadatos que deben ser utilizados en la producción de contenidos educativos. El esfuerzo más prometedor ha desembocado en el estándar IEEE LOM, para el manejo de metadatos de OA; estándar que también ha sido adoptado, en una versión preliminar, como una especificación de descripción de metadatos por IMS; pero también se puede ver como una extensión del conocido estándar de propósito general Dublin Core, con elementos que describen propiedades adicionales de recursos educativos.(KASSANKE 2001).

Este estándar consta de 9 categorías de metadatos, en las que cada elemento es opcional. Cada categoría está conformada por subcategorías, formando así una jerarquía de más de 60 elementos distintos.

LOM distingue 9 categorías de metadatos diferentes:

- Categoría general. Los metadatos en esta categoría representan información general sobre el material educativo, que describe el mismo como un todo.
- Categoría lifecycle (*ciclo de vida*). Esta categoría agrupa metadatos referidos a la historia y estado actual del proceso de producción y mantenimiento del material educativo por parte de los autores.
- Categoría metametadatos (*meta-metadatos*). Agrupa información relativa a los metadatos en sí.
- Categoría technical (*técnica*). Agrupa metadatos relativos a las características y requisitos técnicos del material en sí.
- Categoría educational (*educativa*). Agrupa metadatos relativos a los usos educativos del material.

- Categoría rights (*derechos*). Categoría que agrupa metadatos relativos a los derechos de propiedad e intelectuales del material.
- Categoría relation (*relación*). Categoría de metadatos utilizados para establecer relaciones entre el material y otros materiales.
- Categoría annotation (*anotación*). Anotaciones y comentarios sobre el material educativo.
- Categoría classification (*clasificación*). Metadatos para la clasificación del material en taxonomías.

### Tipos de metadatos para la Categoría General

La categoría general agrupa 9 tipos de metadatos distintos

- identifier (*identificador*).
- title (*título*).
- catalogentry (*entrada en catálogo*).
- language (*idioma*).
- description (*descripción*).
- keyword (*palabra clave*).
- coverage (*cobertura*).
- structure (*estructura*).
- Aggregation level (*nivel de agregación*).

### Tipos de metadatos para la Categoría Lifecycle.

La categoría lifecycle incluye los siguientes 3 tipos de metadatos:

- version. (*versión*)
- status. (*estado*)
- contribute (*contribución*).

### Tipos de metadatos para la Categoría Metametadata

- identifier. (*identificador*)
- catalogentry. (*entrada en catálogo*).
- contribute. (*contribución*)
- metadatascheme (*esquema de metadatos*).
- language. (*lenguaje*)

### Tipos de metadatos para la Categoría Technical

Se muestran los distintos metadatos contemplados por la categoría technical:

- format (*formato*).
- size (*tamaño*).
- location (*localización*).
- requirement (*requisito*).

### Tipos de metadatos para la Categoría Educational

- interactivitytype (*tipo de interacción*).
- Learningresourcetype (*tipo de recurso educativo*).
- Interactivitylevel (*nivel de interacción*).
- semanticdensity (*densidad semántica*).
- intendeduserrole (*papel jugado por el supuesto usuario*).
- context (*contexto*).
- Typicalagerange (*segmento de edad típico*).
- difficulty (*dificultad*).
- typicallearningtime (*tiempo típico de aprendizaje*).
- description (*descripción*).
- language (*idioma*).

### Tipos de metadatos para la Categoría Rights

- cost (*coste*).
- copyrightandotherrestrictions (*derechos de copia y otras restricciones*).
- description (*descripción*).

### Tipos de metadatos para la Categoría Relation.

La categoría relation considera metadatos referidos a la relación entre el material y otros materiales. Cada una de estas relaciones exhibe las siguientes características:

- La clase de la relación.
- La caracterización del otro material con el que se establece la relación. Dicha caracterización puede darse en términos de:
  - El identificador único del otro material.
  - La descripción del otro material.
  - Una entrada en un catálogo para el otro material.

### Tipos de metadatos para la Categoría Annotation.

Los materiales pueden tener asociados múltiples anotaciones. Dichas anotaciones pueden caracterizarse por:

- El anotador que realiza la anotación.

- La fecha de la anotación.
- El texto en sí de la anotación.

### Tipos de metadatos para la Categoría Classification.

LOM permite someter a los materiales a múltiples clasificaciones. Cada clasificación puede tener asociada la siguiente información:

- El propósito de la clasificación.
- Una serie de rutas en distintas taxonomías.
- Una descripción textual del material relativa al propósito de clasificación establecido.
- Un conjunto de palabras clave relativas al propósito de clasificación establecido.

Se decide utilizar este estándar para la descripción de metadatos en el desarrollo de la propuesta actual.

### **1.3.3.2 Estándar Dublin Core**

El estándar de metadatos Dublin Core (DC) es un simple pero eficaz conjunto de elementos para describir una amplia gama de recursos de red. La norma del Dublin Core conlleva dos niveles: Simple y Cualificado (KASSANKE 2001). El Dublin Core Simple tiene quince elementos; el Dublin Core Cualificado un elemento adicional, la audiencia [*Audience*], así como un grupo de elementos de matización que refinan la semántica de los elementos de tal forma que pueden ser útiles para la recuperación/localización de recursos en Internet.

Así pues, los metadatos Dublin Core han tenido un desarrollo espectacular en los últimos años y se han convertido en un estándar de metadatos entre los más extendidos para la recuperación de información en la World Wide Web y de alcance internacional en diferentes ámbitos y sectores. Desde sus primeras aplicaciones en la descripción por medio de etiquetas HTML hasta sus últimos desarrollos en lenguajes estructurados como XML y conjuntamente con otros lenguajes de descripción como RDF.

Los metadatos DC se han desarrollado como un método válido para describir distintas áreas del conocimiento y existen todavía numerosos proyectos en marcha<sup>12</sup>, en campos que no sólo tienen que ver con el área de la biblioteconomía y documentación, el arte y las humanidades, sino también con la

---

<sup>12</sup> <http://dublincore.org/projects/>

educación, el comercio, la ciencia y la tecnología, etc. Así pues, el conjunto de elementos DC es multidisciplinar e internacional y es extensible, ya que permite agregar modificaciones y revisiones de acuerdo con una materia específica o una necesidad concreta.

Usar metadatos Dublin Core, tiene como ventaja la facilidad de su uso. De hecho, se ha convertido en un modelo de descripción de datos semántico a través de RDF. Además, Dublin Core se usa junto con otros perfiles de aplicación. Por ejemplo, el protocolo Z39.50 para el intercambio entre sistemas bibliotecarios, acordó incluir los 15 elementos de los metadatos Dublin Core como formato aceptado y, de esta forma, se pueden recuperar los elementos DC especificándolos en las búsquedas.(LAMARCA 2005).

Cada elemento es opcional y puede repetirse. Además, los elementos pueden aparecer en cualquier orden. Aunque algunos entornos, como HTML, no diferencian entre mayúsculas y minúsculas, es recomendable escribir correctamente cada metadato, según su definición, para evitar conflictos con otros entornos, como SGML y XML.

### 1.4 Estándares para la interoperabilidad

Como es sabido el uso de estándares es esencial para alcanzar resultados en lo que a e-learning se refiere. Se puede decir que muchas veces se hace difícil el entendimiento entre los administradores de las diferentes aplicaciones ya que estas se desarrollan por separado, esto trae algunos problemas, principalmente a los usuarios, quienes tienen que utilizar distintas aplicaciones para tener una actividad e-learning completa. Es por eso que se hace necesario realizar una integración que posibilite que un solo sistema o interfaz pueda realizar todas las actividades operativas en torno a una experiencia de enseñanza-aprendizaje y la creación, búsqueda y transferencia de contenidos entre los sistemas sería más simple. Es aquí que surge el concepto de interoperabilidad, entendida como la capacidad de varios sistemas o aplicaciones de trabajar conjuntamente y compartir sus datos de forma transparente para el usuario.

La interoperabilidad da al usuario la posibilidad de acceder a diversas colecciones de recursos de información (datos y metadatos), gestionados por diferentes sistemas y localizadas en distintos lugares, sin dejar de utilizar en ningún momento la interfaz del sistema propio. Para ello, los sistemas interoperables deben implementar determinados estándares que permiten el intercambio y la gestión conjunta de la información.

Ejemplos de esos estándares están el **Z39.50** que es un estándar para la búsqueda y recuperación de

información, desarrollado y mantenido principalmente para bibliotecas; otro estándar es el **OAI-PMH** es un protocolo de recopilación para la transmisión de metadatos en Internet, tiene una interfaz sumamente sencilla para acceder a la información bibliográfica disponible en un archivo o repositorio, Este se creó con la misión de desarrollar y promover estándares de interoperabilidad para facilitar la difusión eficiente de contenidos en Internet; también se encuentran el **IMS-DRI**, **SQI** y **OKI**, los cuales se explican en las secciones siguientes.

### 1.4.1 OKI (Open Knowledge Initiative)

OKI<sup>13</sup> es un estándar de interoperabilidad orientado, principalmente, a proyectos e-learning dirigidos a la Educación Superior. Es una propuesta concreta de desarrollo que promueve especificaciones que describen cómo los componentes de un entorno de software se comunican entre sí y con otros sistemas empresariales. Al definir con claridad los puntos de interoperabilidad, la arquitectura permite que los componentes de un complejo entorno de aprendizaje sean desarrollados y actualizados de forma independiente, unos de otros.

Algunas de las ventajas de lo mencionado anteriormente son, por ejemplo, que las tecnologías de aprendizaje (proyectos e-learning) apropiadas para que una serie de requisitos de enseñanza y aprendizaje puedan ser integrados conjuntamente en un entorno común; que las tecnologías de aprendizaje y los contenidos puede ser más fácilmente compartida entre centros educacionales y departamentos. Esto proporciona un catalizador para la cooperación y el desarrollo comercial; que hay un menor costo a largo plazo de la propiedad del software porque los componentes pueden ser reemplazados y actualizados sin necesidad de que todos los demás componentes sean modificados y que la modularidad hace que las tecnologías de aprendizaje sean más estables, fiables, y capaces de crecer con el aumento de su uso y permite que los componentes sean actualizados sin desestabilizar otras partes del ambiente. OKI se basa en las tecnológicas que han demostrado ser escalables y fiables en gran escala en empresas de entornos informáticos.

La base que garantiza la interoperabilidad e integración de los componentes que conforman el sistema gestor del aprendizaje se basa en servicios; dichos servicios deben disponer de un conjunto de interfaces de software conocido como Open Service Interface Definiciones (OSIDs), las cuales se basan en una arquitectura constituida por un conjunto de módulos o herramientas, construidos sobre un sistema central de servicios comunes que a su vez comunica con un soporte persistente o repositorio de datos, generalmente constituido por una base de datos con lenguaje SQL.

---

<sup>13</sup> Iniciativa del Conocimiento Abierto (Traducción al español)

La utilización de una estructura de interfaces internas similar tiene como principal ventaja que su modularidad permite mantener y resolver por separado cada conjunto de tareas sin entorpecer al resto de la aplicación.

### **1.4.2 IMS-DRI (IMS Digital Repository Interoperability)**

IMS-DRI tiene como objetivo facilitar el acceso a los contenidos en los repositorios para contextos educativos, con los LMS y los LCMS, pero también con otros sistemas como los portales de búsqueda. Se ha implementado usando servicios Web constituyendo los servicios de una Arquitectura Orientada a Servicios (SOA). Esta especificación se propone para la interoperabilidad entre servicios o aplicaciones que tiene las funciones más comunes de un repositorio: buscar, exponer, coleccionar, enviar, almacenar, pedir, entregar y alertar. Entre estas funciones, se reconocen cinco combinaciones como actividades principales: Buscar/Exponer, Colectar/Exponer, Enviar/Almacenar, Pedir/Entregar y Alertar/Exponer.

### **1.4.3 SQI (Simple Query Interface)**

Es una especificación que pretende ser una capa que garantice la interoperabilidad entre redes o entornos educacionales heterogéneos. El objetivo es ser una parte del sistema que sea capaz de buscar en los distintos repositorios de objetos educativos existentes en las redes; además especifica los métodos que un repositorio puede tener disponibles para recibir y responder consultas de otros repositorios (AGUIRRE 2004). Se ha diseñado esta interfaz con el fin de proveer una especificación simple y fácil de implementar a través de servicios Web. Estos servicios incluyen servicios básicos como por ejemplo servicios de autenticación, gestión de la sesión y servicios de aplicación como gestión de las consultas. SQI es neutral en términos de lenguaje de consultas y modelo semántico usado. Permite la comunicación sincrónica y asincrónica de búsquedas y resultados entre repositorios.

SQI es una Interfaz de Programación de Aplicaciones (API), que es esencialmente una lista coherente de acuerdo en la programación de comandos. Lo único que hace falta es el envío de atención de consultas y la respuesta a las preguntas, pero no la estructura de las preguntas. Esto significa que SQI es mínimo, y, por tanto, relativamente fácil de implementar en una variedad de sistemas.

Los estándares que se seleccionan para la comunicación entre aplicaciones son el IMS-DRI y el SQI.

## **1.5 ¿Qué es la Arquitectura Orientada a Servicios?**

SOA es una arquitectura orientada a servicios, ésta define qué servicios componen un sistema, sus interacciones y con que tecnologías serán implementados. Es una visión de los sistemas de una organización como los servicios que proporcionan y no como los sistemas y funcionalidades en sí. (ENCINA 2006)

Esta arquitectura proporciona independencia total de la plataforma y del lenguaje, además de que permite generar servicios reutilizables y compartidos mediante el uso de lenguajes y protocolos totalmente estandarizados como HTTP, SOAP, XML, UDDI o WSDL.

Una de las funcionalidades que brinda la aplicación de SOA es la integración de funcionalidades y contenidos. El primer modo de integración, es soportado primordialmente por SOAP (Simple Object Access Protocol) y la integración de portales está soportada mediante WSRP (Web Services for Remote Portlets)(RICO 2007).

SOA representa la última tendencia como arquitectura de diseño de aplicaciones. Se basa en un modelo para construir componentes lo más básico posible, con ellos construir otros y al final diseñar aplicaciones basadas en el concepto de reutilización. Supone una evolución como respuesta a los requisitos del negocio, sobre todo requisitos que tengan que ver con la comunicación con terceros, y también en cuanto a la interoperabilidad de las propias aplicaciones.

## **1.6 Servicios Web**

Los Servicios Web son una colección de protocolos y estándares que sirven para comunicar aplicaciones, permitiendo que las aplicaciones compartan información y que además invoquen funciones de otras aplicaciones, independientemente de cómo se hayan creado las aplicaciones, cuál sea el sistema operativo o la plataforma en que se ejecutan y cuáles los dispositivos utilizados para obtener acceso a ellas.

Dicha comunicación se caracteriza por el intercambio de mensajes XML y por ser independientes del protocolo de comunicación. Para lograr esta independencia, el mensaje XML se envuelve de manera apropiada para cada protocolo gracias a la creación del protocolo de transporte SOAP. (ALVEZ 2006).

Algunas ventajas de los Servicios Web son:(BRAVO 2007)

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
- Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar y abiertos. Las especificaciones son gestionadas por una organización abierta, la W3C y se garantiza la plena interoperabilidad entre aplicaciones.

### 1.6.1 Lenguajes y protocolos asociadas.

Una vez definido el servicio Web se necesita darlo a conocer a la comunidad para que sepan de su existencia y ahí es donde juega su papel la **UDDI** (Universal Description Discovery and Integration), lugar en el cual se pueden buscar cuales son los Servicios Web disponibles, una especie de directorio en el cual se pueden encontrar los Servicios Web publicados y publicar los Servicios Web que se desarrollen.

Es una especificación para mantener directorios estandarizados de información acerca de los Servicios Web, sus capacidades, ubicación, y requerimientos en un formato reconocido universalmente.

UDDI utiliza **WSDL** para describir las interfaces de los Servicios Web. Este no es más que un lenguaje de descripción de servicios Web (Web Service Description Language) basado en XML sobre el esquema que describe un servicio Web.

Un documento WSDL proporciona la información necesaria al cliente para interactuar con el servicio Web, es decir describe como acceder al servicio, de qué funciones dispone, qué argumentos necesita y qué devuelve en cada uno. WSDL es extensible y se puede utilizar para describir, prácticamente, cualquier servicio de red, incluyendo SOAP sobre HTTP e incluso protocolos que no se basan en XML.

**XML** es un Lenguaje de Etiquetado Extensible desarrollado por el World Wide Web Consortium (W3C).

Es muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. Este es un formato que permite la lectura de datos a través de diferentes aplicaciones.

Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para estructurar, almacenar e intercambiar la información de una manera segura, fiable y fácil. Puede ser utilizado en bases de datos, editores de texto, hojas de cálculo, entre otras. XML realmente no es un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

**SOAP** (Simple Object Access Protocol)) adoptado por la W3C que permite la comunicación y la interoperabilidad entre diversas aplicaciones Web desarrolladas bajo tecnologías diferentes. Permite trabajar sobre múltiples protocolos de redes informáticas. Los procedimientos de llamadas remotas pueden ser modelados en la forma de varios mensajes SOAP interactuando entre sí.(CASTILLO 2002).

SOAP tiene como objetivos establecer un protocolo estándar de invocación de servicios remotos, basado en protocolos estándares de Internet: HTTP para la transmisión y XML para la codificación de datos y lograr independencia de plataforma, lenguaje de desarrollo e implementación.

Hoy en día tanto SOAP como los servicios Web, están siendo altamente utilizados en las grandes empresas del mundo pues le permiten a estas la cooperación e integridad entre ellas cuando trabajan en un proyecto en común, debido a que permite la interoperabilidad entre sus tecnologías.

En resumen, SOAP define un protocolo XML para la interoperabilidad básica entre servicios, WSDL introduce un lenguaje común para describir servicios y UDDI provee la infraestructura requerida para publicar y descubrir servicios. Juntas, estas especificaciones permiten a las aplicaciones interactuar siguiendo un modelo débilmente acoplado e independiente de la plataforma subyacente.(ALVEZ 2006).

### 1.7 ¿Qué es una herramienta de autor?

Las herramientas de autor son herramientas para la creación de materiales educativos y publicación de cursos; son aplicaciones que permiten un trabajo multimedia y constructivista para generar un entorno de aprendizaje dinámico, dentro de las funcionalidades que este tipo de herramientas presentan, se puede destacar la posibilidad de crear actividades o pequeñas aplicaciones desde la misma herramienta.

Estas herramientas pueden ser aplicadas al desarrollo de programas educativos ya que permite la creación de aplicaciones en las que, de forma sencilla y rápida se tiene la posibilidad de cambiar el flujo de la información según las necesidades del alumno, relacionar palabras, incluir cuestionarios y marcadores que evalúen los conocimientos alcanzados, activar animaciones y videos explicativos, incorporar sonidos y lenguaje hablado, entre otros.

Las herramientas de autor tienen como ventajas la posibilidad de realizar software sin apenas conocimientos de programación; son relativamente fáciles de usar, tienen una máxima contextualización, adaptable a los alumnos y una integración curricular.

Ejemplos de Herramientas de Autor:

### **Autore<sup>14</sup>**

Es una herramienta de autor para la gestión de contenidos basados en OAs. Se trata de una herramienta de código abierto que permite a los programadores adaptar su configuración a las necesidades específicas,(URIARTE 2005) destinada a ser utilizada remotamente por los usuarios finales a través de un navegador Web, ya sea el Mozilla FireFox o el Internet.

Autore es una aplicación de propósito general, que puede instalarse en servidores con distintos sistemas operativos, por lo que se ha desarrollado en Java. Esta herramienta soporta imágenes (ficheros JPEGs, GIFs), audio, videos y ecuaciones matemáticas.

Presenta un entorno intuitivo y amigable de sencillo manejo, basada en un editor What you See is What You Get (WYSIWYG), el usuario podrá gestionar sus recursos didácticos alojándolos en páginas de edición sencilla sin necesidad de conocer lenguajes de programación como HTML(URIARTE 2005). El estándar elegido para definir la forma de los OA es SCORM, por lo que esta herramienta se ajusta a criterios de interoperabilidad. Es multilingüe, el usuario puede elegir entre ingles, euskera y castellano. Sin embargo tiene como limitación que los OA no son recursos flexibles, estos constaran de tres partes (Idea-Descripción-Evaluación), donde la idea es obligatoria.

### **eXeLearning<sup>15</sup>**

Es un programa de edición de sitios Web educativos de código abierto, que permite a los profesores generar contenidos educativos sin necesidad de tener amplios conocimientos del lenguaje HTML o XML. Los contenidos educativos pueden exportarse como páginas Web, paquetes de contenido con

---

<sup>14</sup> <http://www.autore.ehu.es/AUTORe>

<sup>15</sup> <http://exelearning.org/>

formato estándar SCORM 1.2 e IMS Content Packaging 1.1.3; ambas soluciones le permitirán elaborar automáticamente el referido fichero de tipo ZIP que puede ser importado por cualquier campus virtual que cumpla estándares. Estos paquetes pueden utilizarse en las diferentes plataformas de enseñanza en línea (LMS). Está escrito en el lenguaje de programación Python y que se basa en el navegador Web de código abierto, Firefox.

La última versión es la 1.03, desarrollado completamente bajo el soporte del CORE Education. Puede descargarse para diferentes sistemas operativos: Windows, MAC OS X y Linux. También hay una versión que se puede ejecutar desde una memoria USB. (CUEVA 2007).

Como inconveniente, se debe señalar que el editor WYSIWYG que tiene el eXeLearning es considerado obsoleto, esto ocasiona que a veces se vuelva engorrosa la edición de textos.

Obviamente, el eXeLearning no es un gestor de contenidos; lo que permite es que un profesor se construyese una vista de sus materiales siguiendo un protocolo. Pero para una institución educativa, además se necesitaría un gestor de contenidos capaz de entender estos paquetes IMS y hacer con ellos una publicación Institucional.

### ***Reusable eLearning Object Authoring & Delivery (ReLOAD) <sup>16</sup>***

Constituye otra opción gratuita y de código abierto, se puede editar, pre-visualizar y empaquetar contenidos en diferentes tipos de estándares (SCORM, IMS) ó sea integrar SCOs. El editor ReLOAD, además de ser gratis, es de fácil instalación para cualquiera de los tres sistemas operativos - Windows, Macintosh y Linux - y su uso es relativamente fácil. La versión 2.5.5 incluye soporte para IEEE LOM, SCORM 2004, IMS CP. Un inconveniente que tiene dicha herramienta es que el editor ReLOAD, está en inglés, y aunque existe la opción de cambiarlo a otros idiomas, es insuficiente puesto que modifica sólo algunos términos. Además de necesitar de un navegador Web para ver el contenido de los paquetes.(GONZÁLEZ 2005).

Entre las funcionalidades contempladas por ReLOAD se encuentran crear, importar, editar y exportar paquetes de contenidos; empaquetar contenidos creados con otras herramientas; darle un propósito nuevo a los contenidos a través de la reorganización y recatalogación de los mismos; preparar contenidos para almacenar en sitios destinados a tales efectos y entregar contenido a usuarios finales usando la herramienta de guardado previo de contenidos.

---

<sup>16</sup> <http://www.reload.ac.uk/editor.html>

### **Lompad<sup>17</sup>**

Es un generador de metadatos para Windows, Mac y Linux que no permite crear objetos de aprendizaje, simplemente genera todos los metadatos que necesita, ya sea para el estándar IEEE LOM (Learning Object Model) o perfiles de SCORM. Es bastante recto y fácil de usar hasta la fecha. Es bilingüe (Francés, Inglés), pero su idioma por defecto es el francés. Esta herramienta ha sido elaborada en Java.

### **Weload<sup>18</sup>**

WELOAD es un proyecto alemán para imitar la funcionalidad de RELOAD, pero necesita para su funcionamiento un servidor Web apache portátil y del Sistema Gestor de Bases de Datos (SGDB) MySQL, los cuales vienen incluidos en el paquete de distribución de la aplicación. Weload puede ser instalado tanto en el disco duro como en una memoria USB, al igual que en una gran variedad de Sistemas Operativos. Esta se ha desarrollado con lenguaje PHP.(COSS 2006) Las funcionalidades de esta herramienta son subir OA al servidor, dibujar el árbol de contenidos y lanzar los contenidos. Soporta los estándares SCORM e IMS Content Packaging. Es de libre distribución y de código abierto.

### **Roxs<sup>19</sup>**

Es una aplicación Web que permite elaborar objetos de contenidos educativos basados en las especificaciones SCORM, utilizados por las diferentes plataformas de educación a distancia que tengan implementados estos estándares. Es una herramienta Web flexible, de fácil administración, interacción agradable y cómodo uso, que posibilita la creación de paquetes SCORM utilizados para la elaboración de cursos por los diferentes LMS. Tiene como inconvenientes que no describe todos los metadatos de LOM, solamente incorpora SCORM y además no implementa estándares para lograr la interoperabilidad a nivel de sistema con otras aplicaciones.

## **1.8 Lenguajes de Programación para la Web**

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una computadora. Consiste en un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Permite a uno o más programadores especificar de manera precisa sobre qué datos una computadora debe operar, cómo deben ser estos almacenados, transmitidos y qué acciones debe tomar bajo una variada gama de

---

<sup>17</sup> <http://sourceforge.net/projects/lompad>

<sup>18</sup> <http://weload.lernnetz.de>

<sup>19</sup> Trabajo de Diploma Herramienta de Autor para la creación y gestión de objetos de aprendizaje reutilizable.

circunstancias.

Estos lenguajes se pueden dividir para entender mejor su clasificación en dos partes fundamentales: los lenguajes del lado del Servidor y los lenguajes del lado del Cliente, donde ambos reconocen la filosofía de la arquitectura Cliente/Servidor para las plataformas Web.

Entre los lenguajes que se encuentran en el lado del servidor se pueden encontrar como los más importantes PERL, ASP, PHP, Java, JSP. Ellos se caracterizan por implementar la lógica del negocio dentro del servidor, además de encargarse de la tarea del acceso a las Bases de Datos y del tratamiento de la Información.

### ***PHP (Hypertext Preprocessor)***

Es un lenguaje de programación de alto nivel embebido en páginas HTML y ejecutado en el servidor, es de código abierto y resulta muy útil para diseñar de forma rápida y eficaz aplicaciones Web dinámicas dirigidas a bases de datos.

Es multiplataforma (GNU/Linux, Windows, Mac); es libre, convirtiéndose en una alternativa de fácil acceso para todos; tiene capacidad de conexión con la mayoría de los gestores de bases de datos utilizados en la actualidad, siendo ésta una de sus características más potentes; también permite las técnicas de programación orientadas a objetos y tiene soporte para diferentes protocolos.

PHP tiene asequible estructura de programación, la facilidad de llevar a cabo sentencias SQL, además de permitir la posibilidad de correr en diferentes tipos de servidores, entre ellos Apache. Además ofrece la integración con las varias bibliotecas externas, que ofrecen múltiples facilidades a los desarrolladores.

Este es el lenguaje de programación que se escoge para desarrollar la propuesta actual.

### ***PERL***

Es un lenguaje de scripting sencillo, flexible, poderoso y eficiente. En sus inicios era un lenguaje para facilitar la administración de ordenadores, y se ha convertido en una herramienta de propósito general, usada en la programación de sitios Web, bases de datos, procesamiento de grandes volúmenes de datos, etc. Puede utilizarse tanto para la creación de programas gráficos, como para el armado de páginas dinámicas en la Web, la automatización de tareas y muchas otras aplicaciones.

Perl puede funcionar en cualquier plataforma que tenga un intérprete para el lenguaje. Una de las

grandes ventajas del Perl es la comunidad de usuarios donde hay listas de correo, grupos de noticias, y foros en la Web, donde plantear y resolver las dudas que surjan en el desarrollo de una aplicación. Perl es además ideal para desarrollo rápido de aplicaciones, aunque su modularidad permite que se desarrollen también aplicaciones más complejas. Está licenciado bajo la GNU General Public License.

### **ASP**

ASP (Active Server Pages) es la tecnología para la creación de páginas dinámicas del lado del servidor desarrollada por Microsoft. Es en efecto el servidor quien se ocupa de ejecutarlo, interpretarlo y enviarlo al cliente (navegador) en forma de código HTML.

Con ASP se pueden realizar fácilmente páginas de consulta de bases de datos y funciones sencillas. El tipo de servidores que emplean este lenguaje son aquellos que funcionan con sistema operativo de la familia de Windows NT, pero entre sus desventajas está que es un poco lento, presenta problemas con servidores de terceros y es demasiado limitado.

### **Java**

Es un lenguaje de programación orientado a objetos, este lenguaje es muy parecido a C y C++, pero tiene un modelo de objetos más simple. Es multiplataforma y gratuito.

Java ha tenido un avance significativo en el mundo de los entornos software ya que cuenta con una sintaxis fácilmente accesible y un entorno robusto y agradable; pone al alcance de cualquier persona la utilización de aplicaciones que se pueden incluir directamente en páginas Web.

Una de las principales características que favoreció el crecimiento y difusión del lenguaje Java es su capacidad de que el código funcione sobre cualquier plataforma de software y hardware. Esto significa que el mismo programa escrito para Linux puede ser ejecutado en Windows sin ningún problema.

## **1.9 Tecnologías del lado del cliente**

En el lado del cliente se encuentran principalmente el Java Script (JScript), el Visual Basic Script (VBScript), CSS y HTML que son los encargados de aportar dinamismo y eficacia a la aplicación en los navegadores al ejecutarse ciertas funcionalidades que no recargan al servidor, brindando rapidez y optimización en los canales de comunicación.

### **Java script**

Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez, se utiliza para crear pequeños programas encargados de realizar acciones dentro de una página Web. Es un lenguaje del lado del cliente porque es el navegador el que soporta la carga de procesamiento. Fundamentalmente se basa en la creación de efectos especiales en las páginas, además de definir interactividades con los usuarios.

Es uno de los lenguajes más utilizados debido a su compatibilidad con la mayoría de los navegadores de la actualidad.

### **Visual Basic Script**

Es un lenguaje de programación de scripts, pero sólo es compatible con Internet Explorer. Está basado en Visual Basic, popular lenguaje para crear aplicaciones Windows. Su funcionamiento es muy similar al utilizado en Java Script y como recurso para acceder esta el navegador.

### **CSS**

Son las siglas de Cascading Style Sheets, en español Hojas de estilo en Cascada. Es una tecnología que permite crear páginas Web de una manera más exacta. Gracias a este lenguaje se pueden hacer mucho más cosas que antes no se podían hacer utilizando solamente HTML, como incluir márgenes, tipos de letra, fondos, colores, definir estilos en un archivo externo a nuestras páginas.

### **HTML**

Se traduce al español como Lenguaje de Etiquetas de Hipertexto. Es un lenguaje de marcado, diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web. HTML se ha convertido en uno de los formatos más populares y fáciles de aprender que existen para la elaboración de documentos para la Web. También permite códigos en lenguajes de programación extendiendo su capacidad y funcionalidad. Es el estándar usado en el World Wide Web.

HTML proporciona los medios para poder publicar online documentos con cabeceras, texto, tablas, listas, fotos; obtener información en línea vía enlaces de hipertexto; diseñar formularios para realizar transacciones con servicios remotos, que nos permitan búsqueda de información, además de incluir hojas de cálculo y otras aplicaciones directamente en los documentos.

Para el desarrollo de la Herramienta se decide utilizar Java script, HTML y CSS.

### 1.10 Servidores Web

Un servidor Web es un ordenador que usa el protocolo HTTP (hypertext transfer protocol), el cual define cómo son transmitidas las páginas Web al ordenador de un usuario cuando este las solicita. Estas páginas pueden estar compuestas por archivos multimedia (audio, imágenes o videos).

Los archivos para cada sitio de Internet se almacenan en el servidor, aunque algunos de ellos, como los de tipo CSS y Java Script son ejecutados en el ordenador cliente, por eso son llamados “lenguajes del lado del cliente” y otros, como los de tipo PHP son interpretados por el servidor. Hay muchos servidores en Internet y muchos tipos de servidores, pero comparten la función común de proporcionar el acceso a los archivos y servicios.

Uno de estos servidores es Internet Information Services (IIS), que constituye una serie de servicios para los ordenadores que funcionan con Windows. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Este servicio convierte a un ordenador en un servidor de Internet o Intranet es decir que en las computadoras que tienen este servicio instalado se pueden publicar páginas Web tanto local como remotamente (servidor Web).

IIS tiene como deficiencia que es propiedad de Microsoft Corporation, por lo tanto sólo funciona en sistemas operativos Windows, además de sus grandes problemas de seguridad.

Actualmente el servidor Web más utilizado es el Apache por ser gratuito, modular, open source, multiplataforma y extensible, además de ser compatible con una amplia variedad de sistemas operativos (Unix, Windows, Macintosh, entre otros). Está diseñado para ser un servidor Web flexible y potente que puede funcionar sobre varias plataformas y entornos, además de que se puede personalizar mejor para las necesidades de cada sitio Web.

El servidor Web que se escoge para el desarrollo de esta propuesta es el Apache.

### 1.11 Gestores de Bases de Datos

Los Sistemas Gestores de Base de Datos (SGBD) es un software de propósito general proporcionando una interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Su objetivo es el de

manejar de manera sencilla y ordenada un conjunto de datos que luego se convertirán en información relevante, para un buen manejo de datos. Además permite a los usuarios crear y mantener una BD asegurando la privacidad, integridad y seguridad de los datos.

### **MySQL**

Es uno de los sistemas de gestión de bases de datos SQL (Structured Query Language) más popular, pero actualmente es propiedad y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Es una implementación Cliente-Servidor que consta de un servidor y diferentes clientes (programas/librerías). Se puede agregar, acceder, y procesar datos grabados en una base de datos.

MySQL es increíblemente rápido, confiable, robusto y tiene gran facilidad de uso tanto para volúmenes de datos grandes como pequeños, ésta aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. Ofrece un rico y muy útil conjunto de funciones. La conectividad, velocidad y seguridad hace de MySQL altamente conveniente para acceder a bases de datos en Internet.

### **Oracle (Relational Data Base Management System):**

Es una potente herramienta cliente/servidor para la gestión de Bases de Datos, es uno de los sistemas de bases de datos más completos, donde se destaca su estabilidad, su escalabilidad, además de ser multiplataforma. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales. En el desarrollo de páginas Web pasa lo mismo al ser un sistema muy caro no está tan extendido como otras bases de datos, por ejemplo, Access, MySQL, SQL Server, etc.

### **Microsoft SQL Server**

Es un sistema de gestión de bases de datos relacionales (SGBD), capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea. Basado en el lenguaje SQL, este sistema provee de mayor escalabilidad, disponibilidad y seguridad a las aplicaciones y los datos empresariales al tiempo que simplifica su creación, implementación y gestión. Se destaca también por su amplio soporte de transacciones y procedimientos almacenados. Incorpora, además, un potente entorno gráfico de administración y una interfaz de acceso para las principales plataformas de desarrollo, incluyendo .NET.

Microsoft SQL Server se presenta como la alternativa de Microsoft a sistemas gestores de bases de datos tan famosos como Oracle, PostgreSQL o MySQL.

### **PostgreSQL**

Es un poderoso manejador de bases de datos de código abierto, sin costos de licencia, diseñado para administrar grandes cantidades de datos. Es robusto, confiable y mantiene la integridad de los datos. Se ejecuta en la mayoría de los Sistemas Operativos más utilizados en el mundo incluyendo, Linux, varias versiones de UNIX y en Windows. Es un magnífico gestor de BD, actualmente considerado uno de los sistemas gestor de BD más avanzado del mundo. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta.

Se decide utilizar este gestor de base de datos en la implementación de la Herramienta de autor modelada en el presente trabajo.

## **1.12 Metodologías de desarrollo de SW**

Las metodologías de desarrollo de software son un conjunto de pasos y procedimientos, técnicas y ayudas a la documentación que deben de seguir los desarrolladores para realizar nuevos productos de software de manera sistemática, indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben de tener. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.(ASENSIO 2005).

El desarrollo de software no es tarea fácil y prueba de esto es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proyecto de desarrollo, además de que su uso garantiza características como la calidad y la productividad, factores claves tanto para el cliente como para el desarrollador.

Entre las más conocidas metodologías de desarrollo de software están: Rational Unified Process (RUP) adaptable para proyectos de largo plazo, Microsoft Solution Framework (MSF) la cual se vincula a proyectos de cualquier dimensión y tecnología, eXtreme Programming (XP) y Feature Driven Development (FDD), estas dos últimas para proyectos de corto plazo.(SANCHEZ 2004).

### **Microsoft Solution Framework (MSF)**

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. MSF tiene como características que es adaptable, siendo su uso limitado a un específico lugar; escalable, donde puede organizar equipos de 3 o 4 personas como también proyectos que requieren 50 personas o más; es flexible, utilizado en el ambiente de desarrollo de cualquier cliente; y su tecnología es agnóstica porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.(SANCHEZ 2004).

### **Extreme Programming (XP)**

Es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizadas para proyectos de corto plazo y corto equipo. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. Se encuadra dentro de la familia de metodologías ligeras, tratando de obtener métodos sencillos de obtener software de calidad.(CONTRERAS 2006) Uno de sus principios básicos es la mejora de la comunicación con los usuarios y los desarrolladores, la simplicidad, al desarrollar y codificar los módulos del sistema, la retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

XP es un proceso muy orientado a la implementación. Lo que es muy poco deseable en XP es el hecho de evitar cualquier tipo de documentación fuera del código fuente (UML juega un papel prácticamente nulo, por ejemplo).

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

Es una metodología de desarrollo ágil para el desarrollo de sistemas, que disminuye el riesgo de los proyectos, pues gracias a sus entregas tangibles y al constante monitoreo de su calidad, se asegura el firme avance del mismo. No hace énfasis en la obtención de los requerimientos sino en como se realizan las fases de diseño y construcción. Se basa en un proceso iterativo con iteraciones cortas que producen un software funcional que el cliente y la dirección de la empresa pueden ver y monitorear. También es adaptativo, pues permite realizar cambios de último momento debido a nuevos requerimientos y a las necesidades del negocio. Sin embargo presenta su talón de Aquiles en la necesidad de tener en el equipo miembros con experiencia que marquen el camino a seguir desde el principio, con la elaboración del modelo global, puesto que no es tan ágil como podría serlo XP. (ZAPATA 2006)

### **RUP**

Esta metodología es un proceso de desarrollo de software y constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. El ciclo de vida del RUP se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso.

Está basado en componentes y utiliza el lenguaje UML para visualizar, especificar y documentar los artefactos de un sistema de software orientado a objetos (OO).

En un proyecto de desarrollo de software la metodología define Quién (trabajadores) debe hacer Qué (artefactos), Cuándo (flujo de trabajo y fases) y Cómo (actividades) debe hacerlo. Los elementos principales de RUP son:

- **Trabajadores:** roles que definen el comportamiento y responsabilidades de los individuos.
- **Actividades:** tareas que tiene un propósito y se asignan a un rol.
- **Flujo de trabajo:** secuencia de actividades para producir determinados artefactos.
- **Artefactos:** son productos tangibles del proyecto que son producidos, modificados y usados por las actividades.

Divide el proceso de desarrollo en ciclos, teniendo un producto final al culminar cada una de ellos, estos a la vez se dividen en fases que finalizan con un hito donde se debe tomar una decisión importante:

- **Concepción o Inicio:** se hace un plan de fases, se identifican los principales casos de uso y se identifican los riesgos.
- **Elaboración:** se hace un plan de proyecto, se completan los casos de uso y se eliminan los riesgos.
- **Construcción:** se concentra en la elaboración de un producto totalmente operativo y eficiente y se realiza un manual de usuario.
- **Transición:** se instala el producto en el cliente y se entrena a los usuarios.

También define 9 flujos de trabajo:

- Modelado del negocio.
- Análisis de requisitos.

- Análisis y diseño.
- Implementación.
- Prueba.
- Distribución.
- Gestión de configuración y cambios.
- Gestión del proyecto.
- Gestión del entorno.

Se decide utilizar esta metodología, que está entre las más usadas actualmente en el mundo, dada las condiciones de trabajo, el tiempo de desarrollo y las características de la aplicación que se desarrollará.

### 1.12.1 Herramientas para la Modelación Visual del Sistema.

Las herramientas CASE (Computer Aided Software Engineering) son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática y documentación o detección de errores entre otras. Utilizan Unified Modeling Language (UML) y están destinadas a aumentar la productividad en el desarrollo reduciendo el coste de las mismas en términos de tiempo y de dinero.

#### **Rational Rose**

Herramienta de modelación visual desarrollada por Rational Corporación basada en UML permite crear los diagramas que se van generando durante el proceso de ingeniería en el desarrollo de un sistema informático.

Dicha herramienta es capaz de generar el código fuente de las clases definidas en el flujo de trabajo de diseño, pero tiene la limitación de que aún hay varios lenguajes de programación que no soporta. Por otra parte, una vez terminado el diagrama de clases persistentes a partir del cual se genera la base de datos del sistema, no existe la posibilidad de que el mismo exporte ese modelo hacia algún sistema gestor de bases de datos. Es la herramienta usada generalmente para el desarrollo de software propietario.

### **Visual Paradigm**

También utiliza UML como lenguaje de modelado. Soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objeto, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Es una herramienta que puede ser utilizada en la creación de software libre y es disponible en múltiples plataformas.

Permite realizar ingeniería tanto directa como inversa, ya que a partir de un modelo relacional en Sql Server, MySQL, etc., es capaz de desplegar todas las clases asociadas a las tablas. Además, la herramienta es colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto; genera la documentación del proyecto automáticamente en varios formatos como Web o Pdf, y permite control de versiones.(LEÓN 2007)

Se selecciona esta herramienta Case para realizar el modelado la propuesta actual.

### **1.13 Conclusiones**

En el presente capítulo se abordó de una manera más amplia los conceptos imprescindibles para la comprensión del campo de acción y los principales proyectos que desarrollan el e-learning actualmente. También se analizaron las principales herramientas de autor existentes en la actualidad, sus ventajas y desventajas. Por último se identificaron los estándares, metodología y tecnologías a utilizar para el desarrollo de esta herramienta.

## **Capítulo 2: Características del Sistema**

### **2.1 Introducción**

En el presente capítulo se describen las principales funciones y el flujo actual de los procesos relacionados en el campo de acción planteado anteriormente, donde se realiza un análisis crítico de dichos procesos, observando cómo se ejecutan actualmente.

También se desarrollan los flujos de requerimientos, análisis y diseño de la metodología propuesta en el capítulo anterior (RUP). Se incluye la elaboración del modelo de dominio, se plantean los requisitos funcionales y no funcionales que requiere el sistema, se definen los actores, sus relaciones y los casos de uso del sistema con sus descripciones textuales.

### **2.2 Propuesta actual**

La propuesta actual se trata de una herramienta de autor Web para la creación y edición de contenidos, diseñada para ayudar a usuarios, principalmente profesores, al desarrollo y publicación de materiales docentes y educativos sin necesidad de tener conocimientos profundos sobre XHTML y XML.

Los OA serán empaquetados utilizando los estándares IMS CP y SCORM, además se usará como esquema de metadatos LOM, para facilitar su reutilización. También se desarrollará una arquitectura orientada a servicios (SOA), lo que conllevará a que todas las funcionalidades de la HA sean concebidas como servicios Web, utilizando los estándares IMS-DRI y SQL. Del primero se tendrá en cuenta la funcionalidad SUBMIT, para la comunicación y transmisión de información entre sistemas, esta comunicación será principalmente con un repositorio de objetos de aprendizaje para almacenar los paquetes creados por la herramienta. Del segundo se utilizan los métodos `setQueryLanguage()`, `setResultsFormat()`, `setMaxQueryResults()`, `setResultsSetSize()`, `getTotalResultsCount()` y `queryResultsListener()` para la búsqueda y recuperación de los contenidos en el repositorio.

Esta aplicación se desarrollará con lenguaje PHP 5.0 por sus potencialidades y ventajas que presenta como software libre montado sobre un servidor Apache, el cual se decidió utilizar por su versatilidad, flexibilidad y cuestiones de compatibilidad con el lenguaje de desarrollo Web que se utiliza, además de utilizar lenguajes del lado cliente como java Script, HTML y CSS. El Gestor de Base de Datos utilizado

es PostgreSQL 8.1, uno de los más populares para aplicaciones Web, también muy asociado al lenguaje de programación escogido por su rapidez, robustez y seguridad.

Después de realizar un estudio de varias metodologías se decide utilizar RUP, la cual es una de las metodologías ágiles más usadas actualmente en el mundo, es un proceso de ingeniería de software para producir software de calidad, flexible y en plazos, además de que ayuda a alcanzar el Nivel de Madurez 2 del CMM. Como lenguaje de modelado se utiliza el UML y la herramienta para representar los artefactos es el Visual Paradigm en su versión 5.3 por ser de libre distribución, open source y brindar facilidades en materia de desarrollo colaborativo.

Esta herramienta además de subir recursos ya elaborados, permite crear ficheros textos, a los cuales se les puede dar formato, se le puede insertar imágenes, tablas, videos, expresiones matemáticas, etc. Brinda la posibilidad también de crear tipos de preguntas como Verdaderos y Falsos, rellenar espacios en blancos y preguntas de selección múltiple. Todos estos contenidos son empaquetados en un fichero .Zip para luego ser subidos a un repositorio de OA o a un LMS.

Hasta aquí se han brindado elementos generales del trabajo que se está llevando a cabo en la presente investigación, ahora se considera necesario abordar los aspectos técnicos de la misma.

### **2.3 Modelo de Dominio**

Después de haber realizado un profundo estudio de los procesos que se van a efectuar, se encontraron dos aproximaciones para expresar el contexto de un sistema en una forma utilizable para los desarrolladores de software: el modelado del dominio y el modelado del negocio, y se llegó a la conclusión de que el modelo de negocio presenta un bajo nivel de estructuración en esta aplicación, ya que no se pueden identificar quienes son los clientes con los cuales desarrollar un negocio y poder ver desde su punto de vista el desarrollo productivo; por lo que se decide realizar el modelo de dominio.

El Modelo de Dominio (o Modelo Conceptual) es una representación visual de los principales conceptos u objetos del mundo real significativos para un problema o área de interés. Esto es de gran ayuda entre desarrolladores y usuarios ya que de esta forma utilizan un vocabulario común y pueden entender el contexto en que se enmarca el sistema.

### **2.3.1 Descripción del Problema de Dominio.**

El desarrollo de una Herramienta de Autor Web interoperable para la creación de objetos de aprendizaje permite generar paquetes de contenidos ya sea SCORM e IMS CP, posibilitando la reutilización de estos contenidos educativos en diferentes plataformas que soporten dichos paquetes. El sistema debe ser accesible a través de un entorno Intranet/Internet.

Entre las principales funcionalidades que tiene el sistema se encuentran:

#### *Subir recurso:*

Permite a los usuarios subir objetos de aprendizaje, a los cuales se le deben llenar los campos de metadatos de LOM. Algunos ejemplo son título, identificador, descripción, palabras claves, lenguajes.

#### *Mostrar vista previa del paquete:*

Permite que los usuarios vean la estructura y organización del paquete, es decir, las secciones, temas y sub-temas que contienen cada uno de ellos, además de visualizar el recurso al acceder al tema con el que está relacionado.

#### *Generar paquete:*

Esta opción le permite al usuario generar el paquete IMS CP o SCORM que contendrá los objetos de aprendizaje subidos anteriormente, así como describir dicho paquete mediante los metadatos, dándole además la posibilidad de descargarlo para su máquina.

#### *Agregar metadatos:*

Esta opción permite, una vez subido los recursos a la aplicación, agregarle metadatos a dichos recursos por separado o al paquete en general.

#### *Exportar paquete:*

Esta funcionalidad me permite subir y guardar el paquete, ya una vez creado, en el repositorio.

### 2.3.2 Definición de las entidades y los conceptos principales.

**Objetos de Aprendizaje:** cualquier recurso con una intención formativa, compuesto de uno o varios elementos digitales, descrito con metadatos, que pueda ser utilizado y reutilizado dentro de un entorno e-Learning. (LÓPEZ 2005).

**Recurso:** Elemento necesario para llevar a cabo una tarea. Puede ser cualquier parte o componente de un sistema de información. Ejemplos de recursos están los archivos de texto, ilustraciones, videos, fotografías y animaciones.

**Metadatos:** Son datos que describen otros datos, es decir, información relativa a los propios datos que facilitan su catalogación.

**Paquete SCORM:** Conjunto de objetos de aprendizajes reutilizables empaquetados según el estándar SCORM.

**SCORM:** Shareable Content Object Reference Model (Modelo de Referencia de Objetos de Contenido Intercambiable).

**Organización:** Es una parte de la estructura del paquete. Es el título del tema a tratar de los ítems que contiene.

**Ítems:** Son los diferentes subtítulos que se encuentran dentro de la organización, donde cada uno de estos está relacionado con un recurso determinado.

### 2.3.3 Representación del modelo de dominio.

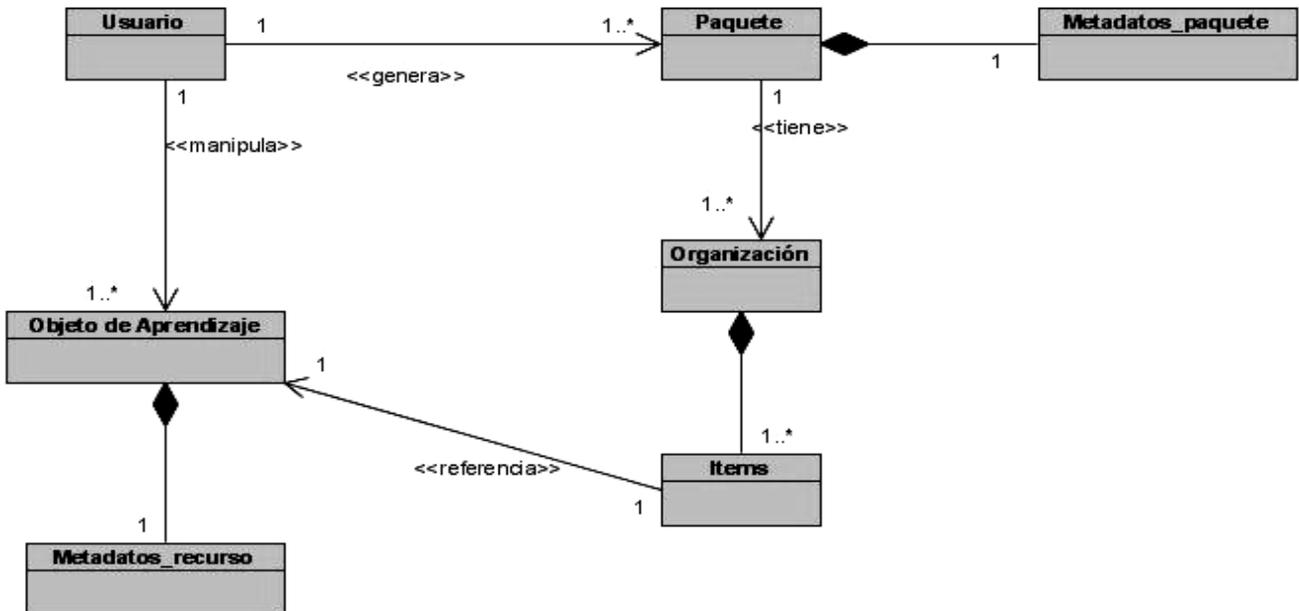


Figura 1. Diagrama del Modelo de Dominio

## 2.4 Especificación de los requisitos de software.

Los requerimientos son condiciones que tienen que ser alcanzada por un sistema o componente de un sistema para satisfacer un contrato, estándar u otro documento impuesto formalmente. Define qué es lo que el sistema debe hacer, identificando así las funcionalidades requeridas y las restricciones que se imponen.

Estos requerimientos se dividen en dos grupos: los requisitos funcionales y los requisitos no funcionales.

### 2.4.1 Requerimientos Funcionales

A continuación se especifican los requisitos funcionales propuestos para el sistema:

R1- Autenticar Usuario.

R2- Gestionar usuario.

R 2.1 Crear usuario.

R 2.2 Eliminar usuario.

R 2.3 Mostrar datos usuario.

R 2.4 Editar datos usuario.

R3-Gestionar rol.

R 3.1 Crear rol.

R 3.2 Modificar rol.

R4-Gestionar perfil.

R 4.1 Ver mi perfil.

R 4.2 Editar mi perfil.

R5-Mostrar reportes.

R 5.1 Mostrar lista de usuarios online.

R 5.2 Mostrar reportes de los usuarios conectados según un IP.

R 5.3 Mostrar reportes de los IP según un usuario.

R6-Configurar tipo de autenticación.

R 6.1 Configuración manual.

R 6.2 Configuración LDAP.

R7-Configurar HTTPs.

R8-Gestionar mensajes.

R 8.1 Crear mensajes.

R 8.2 Mostrar mensajes.

R 8.3 Eliminar mensajes.

R9-Administrar interfaz gráfica.

R 9.1 Configurar bloques.

R 9.1.1 Mostrar bloques.

R 9.1.2 Ocultar bloques.

R 9.1.3 Añadir bloques.

R 9.1.4 Eliminar bloques.

R 9.1.5 Mover bloques.

R 9.2 Configuración del texto inicio.

R10-Crear paquete de contenidos.

R 10.1 Subir recurso.

R 10.2 Eliminar recurso.

R 10.3 Añadir nueva organización.

R 10.4 Añadir nuevo ítem.

R 10.5 Relacionar ítem - recurso.

R11-Modificar paquete.

R12-Eliminar paquete.

R13-Agregar metadatos.

R 13.1 Adicionar metadatos al recurso.

R 13.2 Adicionar metadatos al paquete.

R14-Modificar metadatos.

R15-Mostrar vista previa del paquete.

R16-Generar paquete.

R17-Descargar paquete.

R18-Crear tipos de contenido.

R 18.1 Crear contenido para ficheros textos.

R 18.1.1 Insertar contenido para objetos multimedia (imágenes, animaciones, videos).

R 18.1.2 Insertar expresiones matemáticas.

R 18.1.3 Insertar tablas.

R 18.1.4 Insertar enlaces.

R 18.2 Crear tipo de preguntas.

R 18.2.1 Confeccionar preguntas de espacios en blanco.

R 18.2.2 Confeccionar preguntas de elección múltiple.

R 18.2.3 Confeccionar preguntas de verdadero y falso.

R19-Editar tipo de contenido.

R 19.1 Editar contenido para ficheros textos.

R 19.2 Editar contenido para objetos multimedia (imágenes, animaciones, videos).

R 19.3 Editar tablas.

R 19.4 Editar tipos de preguntas.

R20-Eliminar tipo de contenido.

R 20.1 Eliminar contenido para ficheros textos.

R 20.2 Eliminar tipos de preguntas.

R 21-Exportar paquete.

R 22-Recuperar contenido.

R 23-Configurar metadatos

R 23.1 Permitir que el administrador escoja los metadatos que van a hacer autocompletados.

R 23.2 Permitir que el administrador escoja cuales van a hacer los metadatos obligatorios.

R 24-Permitir que el administrador escoja el número máximo de paquetes que puede tener cada autor en su área de trabajo.

### 2.4.2 Requerimientos no Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Deben ser propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

#### Software:

- El sistema se implementará con tecnología PHP 5.0.
- Se requiere PostgreSQL 8.1 como servidor de bases de datos.
- Servidor Web Apache.
- El sistema utilizará los estándares IMS-DRI y SQL para lograr la interoperabilidad con otras aplicaciones.

#### Hardware:

- Microprocesador 200 MHz.
- 32 MB de memoria RAM.
- 4 GB de disco duro.

#### Apariencia o interfaz externa:

- La interfaz del sistema será amigable a los usuarios finales.
- La aplicación estará estructurada de forma clara y comprensible.
- El diseño responderá a la ejecución de acciones de una manera rápida, minimizando los pasos a dar en cada proceso.
- Todos los textos y mensajes de la aplicación aparecerán en idioma español.

#### Seguridad:

- Necesidad de autenticación para acceder a las funcionalidades de la aplicación y que no sean utilizadas por personas que no tienen permisos.
- Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos.

#### Soporte:

- Por parte del cliente se requiere un navegador capaz de interpretar JavaScript y CSS.
- El sistema debe dar la posibilidad de incorporarle nuevos servicios en caso de ser necesarios.
- Las pruebas realizadas al sistema deben permitir evaluar sus ventajas y funcionalidades, además de detectar los errores que presenta.

### Rendimiento

- La herramienta propuesta debe ser rápida y el tiempo de respuesta debe ser el mínimo posible, al igual que la velocidad de procesamiento de la información.

### Portabilidad:

- La aplicación debe funcionar en varias plataformas, siendo posible su acceso a través de cualquier navegador Web.

### Ayuda y documentación en línea

- Realizar un sistema de ayuda que le permita al usuario orientarse respecto a las opciones que le brinda el sistema, utilizando textos explicativos que describan la acción que se está realizando.

## 2.5 Definición de los casos de uso y actores

### 2.5.1 Definición de los actores del sistema

Tabla 1. Definición de los actores del sistema.

Actores	Justificación
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 contenidos y empaquetarlos.
Administrador	Persona encargada de monitorear el sistema así como gestionar paquetes (OAs), usuarios y otras funciones.

### 2.5.2 Casos de uso del sistema

CU-1	Autenticar usuario.
Actor	Autor
Descripción	El autor se identifica ante el sistema para recibir los permisos que hayan sido definidos para él.
Referencia	R1

---

<b>CU-2</b>	Gestionar Usuario.
<b>Actor</b>	Administrador
<b>Descripción</b>	El administrador accede al sistema para crear, editar o eliminar una cuenta de usuario.
<b>Referencia</b>	R2

<b>CU-3</b>	Gestionar roles.
<b>Actor</b>	Administrador
<b>Descripción</b>	El administrador accede al sistema para asignar privilegios a los usuarios.
<b>Referencia</b>	R3

<b>CU-4</b>	Gestionar perfil.
<b>Actor</b>	Autor
<b>Descripción</b>	El autor accede al sistema para observar su perfil y editarlo si necesita algún cambio.
<b>Referencia</b>	R4

<b>CU-5</b>	Mostrar reportes.
<b>Actor</b>	Administrador
<b>Descripción</b>	El administrador lleva el control de los diferentes reportes, por ejemplo los usuarios en línea.
<b>Referencia</b>	R5

<b>CU-6</b>	Configurar tipo de autenticación
<b>Actor</b>	Administrador
<b>Descripción</b>	El administrador accede al sistema para realizar configuraciones respecto al modo de autenticación.
<b>Referencia</b>	R6

<b>CU-7</b>	Configurar HTTPs
<b>Actor</b>	Administrador
<b>Descripción</b>	El administrador accede al sistema para realizar configuraciones respecto

---

	al protocolo de seguridad.
<b>Referencia</b>	R7

<b>CU-8</b>	Gestionar mensajes
<b>Actor</b>	Autor
<b>Descripción</b>	EL autor accede al sistema para hacer uso del servicio de mensajería, el cual permite enviar y recibir mensajes de los demás usuarios del sistema.
<b>Referencia</b>	R8

<b>CU-9</b>	Administrar interfaz gráfica.
<b>Actor</b>	Administrador
<b>Descripción</b>	El administrador lleva el control sobre la interfaz gráfica de la aplicación como añadir/eliminar bloques o configurar el texto de inicio.
<b>Referencia</b>	R9

<b>CU-10</b>	Crear paquetes de contenidos.
<b>Actor</b>	Autor
<b>Descripción</b>	El autor accede al sistema para crear paquetes, creándole una estructura al paquete.
<b>Referencia</b>	R10

<b>CU-11</b>	Gestionar paquetes de contenidos.
<b>Actor</b>	Autor
<b>Descripción</b>	El autor accede al sistema para modificar o eliminar algunos paquetes ya creados anteriormente.
<b>Referencia</b>	R11, R12

<b>CU-12</b>	Gestionar metadatos.
<b>Actor</b>	Autor
<b>Descripción</b>	El autor al crear paquetes debe incorporarle metadatos a dicho paquete o a los objetos de aprendizaje que lo constituyen, además de poder modificarlos para su reutilización.
<b>Referencia</b>	R13, R14

---

<b>CU-13</b>	Mostrar vista previa del paquete
<b>Actor</b>	Autor
<b>Descripción</b>	El autor después de haber conformado su paquete, puede observar la organización de dicho paquete.
<b>Referencia</b>	R15

<b>CU-14</b>	Generar paquete
<b>Actor</b>	Autor
<b>Descripción</b>	El autor al estructurar su paquete puede acceder a la opción generar paquete convirtiendo dicho paquete en un .Zip.
<b>Referencia</b>	R16

<b>CU-15</b>	Descargar paquete
<b>Actor</b>	Autor
<b>Descripción</b>	El autor puede descargar el paquete elaborado por él hacia la computadora.
<b>Referencia</b>	R17

<b>CU-16</b>	Crear contenido.
<b>Actor</b>	Autor
<b>Descripción</b>	El autor accede al sistema para crear contenido elaborando una página HTML, insertando imágenes, editando tablas y creando diferentes tipos de preguntas como verdaderos y falsos.
<b>Referencia</b>	R18

<b>CU-17</b>	Gestionar contenido.
<b>Actor</b>	Autor
<b>Descripción</b>	El autor accede al sistema para una vez creado el contenido poder modificarlo o eliminarlo cuando ya no haga falta.
<b>Referencia</b>	R19,R20

<b>CU-18</b>	Exportar paquete
<b>Actor</b>	Autor

---

<b>Descripción</b>	El autor al crear el paquete puede exportarlo al repositorio de objetos de aprendizaje (ROA).
<b>Referencia</b>	R21

<b>CU-19</b>	Recuperar contenido
<b>Actor</b>	Autor
<b>Descripción</b>	El autor puede realizar búsquedas en el repositorio de algún contenido en específico que le sea necesario para crear un paquete o un contenido.
<b>Referencia</b>	R22

<b>CU-20</b>	Configurar_ metadatos
<b>Actor</b>	Administrador
<b>Descripción</b>	El administrador accede al sistema para definir que metadatos deben ser de llenado obligatorio para los usuarios y cuales no van a tener que llenar.
<b>Referencia</b>	R23

<b>CU-21</b>	Parametrizar_ variables de configuración.
<b>Actor</b>	Administrador
<b>Descripción</b>	El administrador accede al sistema para definir cuántos paquetes se deben guardar en la herramienta por usuario.
<b>Referencia</b>	R24

### 2.5.3 Diagrama de casos de uso del sistema



Figura 2. Diagrama de Casos de Uso del Sistema.

### 2.5.4 Descripción de los Casos de Uso del Sistema

Tabla 2. Descripción Textual <CU Autenticar usuario>

Caso de uso	
CU-1	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 decide autenticarse en el sistema y finaliza cuando se autentica.
<b>Referencias</b>	R1

<b>Precondiciones</b>	
<b>Poscondiciones</b>	El usuario se debe haber autenticado con éxito.
<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>Acción del actor</b>	<b>Respuesta del sistema</b>
	1.1 El sistema chequea si el usuario se autentica de forma manual o LDAP. 1.2 El sistema verifica que los datos sean válidos. 1.3 El sistema permite el acceso al usuario asignándole sus privilegios.
Flujo alternativo	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	1.1 Si el usuario no existe muestra el mensaje "El usuario no existe". 1.2 Si los datos son incorrectos el sistema le muestra el mensaje de error "Usuario o clave no válida".

Tabla 3. Descripción Textual <CU Gestionar usuario>

<b>Caso de uso</b>	
CU-2	Gestionar usuario
<b>Propósito</b>	Tener el control de los usuarios que entran al sistema.
<b>Actores:</b> Administrador	
<b>Resumen:</b> El caso de uso se inicia cuando el administrador escoge la opción de gestionar usuario. Dentro del mismo puede crear, eliminar o editar un perfil de un usuario. Este finaliza cuando se han efectuado las acciones sobre el usuario.	
<b>Referencias</b>	R2
<b>Precondiciones</b>	El usuario debe haberse autenticado y tener privilegios de administrador.
<b>Poscondiciones</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

1. El administrador accede a la opción Usuario.	1.1 El sistema le brinda 4 opciones: a) Si elige la opción Crear usuario, ver sección - 1. b) Si elige la opción Eliminar usuario, ver sección - 2. c) Si elige la opción Mostrar datos de usuario, ver sección - 3. d) Si elige la opción Editar datos de usuario, ver sección - 4.
<b>Sección- 1. Crear usuario</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El administrador selecciona la opción de Crear un usuario.	1.1 El sistema le solicita al administrador los datos de la cuenta a crear.
2. El administrador entra los datos pedidos.	2.1. El sistema verifica que se hayan llenado todos los datos, en especial, aquellos que son obligatorios. 2.2. El sistema notifica el resultado de la operación con el mensaje “El usuario ha sido registrado satisfactoriamente”.
Flujo alternativo	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	2.1. Si en la verificación de completitud de los datos entrados por el administrador, no han sido completados los campos requeridos u obligatorios el sistema mostrará “Debe llenar todos los campos obligatorios (*)”.
<b>Sección- 2. Eliminar usuario</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El administrador escoge la opción Eliminar usuario.	1.1 El sistema le pide el usuario a eliminar.
2. El administrador introduce los datos pedidos.	2.1 El sistema busca el usuario a eliminar. 2.2 El sistema elimina al usuario y muestra el mensaje “El usuario ha sido eliminado correctamente”.

Flujo alternativo	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	2.1 Si el nombre de usuario a eliminar no existe el sistema lanza el mensaje “El usuario no se encuentra”.
<b>Sección- 3. Mostrar_ datos_ usuario</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El administrador accede a la opción Mostrar datos del usuario e introduce un usuario para ver su perfil.	1.1. El sistema busca el usuario. 1.2. El sistema visualiza el perfil del usuario encontrado.
Flujo alternativo	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	1.1. En caso de que no existe el usuario buscado el sistema muestra el mensaje "El usuario no se encuentra".
<b>Sección- 4. Editar_ datos_ usuario</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El administrador opta por la opción Editar datos del usuario.	1.1. El sistema solicita el nombre de usuario que van a modificar.
2. El administrador introduce el nombre de usuario solicitado.	2.1. El sistema busca al usuario. 2.2 El sistema muestra los campos de los datos del usuario.
3. El administrador modifica los campos deseados.	3.1 El sistema se actualiza con los nuevos cambios.
Flujo alternativo	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	2.1 Si el usuario no es encontrado el sistema muestra un mensaje de error “Usuario no valido”.
	3.1. Si el administrador no llena aquellos campos que son obligatorios entonces el sistema notifica el mensaje “Debe llenar todos los campos obligatorios (*)”.

Tabla 4. Descripción Textual <CU Gestionar rol >

Caso de uso	
CU-3	Gestionar rol
<b>Propósito</b>	Asignar permisos a determinados usuarios.
<b>Actores:</b> Administrador	
<b>Resumen:</b> El caso de uso se inicia cuando el administrador decide asignar privilegios a un determinado usuario y finaliza una vez asignado o modificados dichos permisos.	
<b>Referencias</b>	R3
<b>Precondiciones</b>	El usuario debe haberse autenticado y tener privilegios de administrador.
<b>Poscondiciones</b>	
Acción del actor	Respuesta del sistema
1. El administrador selecciona la opción Privilegios de usuario.	1.1 El sistema muestra 2 opciones: a) Si elige la opción Crear rol, ver sección -1. b) Si elige la opción Modificar rol, ver sección -2.
<b>Sección- 1. Crear rol</b>	
Acción del actor	Respuesta del sistema
1. El administrador selecciona la opción Crear rol.	1.1 El sistema solicita el nombre de usuario para asignarle un rol.
2. El administrador introduce el usuario solicitado.	2.1 El sistema busca al usuario. 2.2 El sistema muestra los roles que tiene la aplicación.
3. El administrador selecciona un rol para dicho usuario.	3.1 El sistema guarda las acciones realizadas.
Flujo alternativo	
Acción del actor	Respuesta del sistema
	2.1 Si el usuario no es encontrado el sistema muestra un mensaje de error "Usuario no valido".
<b>Sección- 2. Modificar rol</b>	
Acción del actor	Respuesta del sistema
1. El administrador accede a la opción Modificar rol.	1.1 El sistema le solicita el nombre de usuario que desea modificar.
2. El administrador introduce el nombre de usuario.	2.1 El sistema muestra el rol asignado a ese usuario anteriormente, además de otros roles.

3. El administrador escoge otro rol, cambiándole los privilegios al usuario.	3.1 El sistema muestra el mensaje "Se han cambiado los privilegios".
Flujo alternativo	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	1.1. En caso de que no existe el usuario buscado, el sistema muestra el mensaje "El usuario no se encuentra".

Tabla 5. Descripción Textual <CU Gestionar perfil >

Caso de uso	
CU-4	Gestionar perfil
<b>Propósito</b>	Permite al autor poder observar su perfil y editarlo.
<b>Actores:</b> Autor	
<b>Resumen:</b> El caso de uso se inicia cuando el autor escoge la opción Mi perfil y decide ver su perfil o editarlo y finaliza una vez realizado alguna de estas acciones.	
<b>Referencias</b>	R4
<b>Precondiciones</b>	El usuario debe haberse autenticado.
<b>Poscondiciones</b>	
<b>Sección "Ver _ mi perfil"</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El autor selecciona la opción Mi perfil.	1.1. El sistema visualiza el perfil del usuario.
<b>Sección "Editar _ mi perfil"</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El autor selecciona la opción Editar _ mi perfil.	1.1. El sistema visualiza el perfil del usuario con los campos disponibles para modificarlos.
2. El autor modifica los campos deseados.	2.2. El sistema se actualiza con los nuevos cambios.

Tabla 6. Descripción Textual <CU Mostrar reportes>

Caso de uso	
CU-5	Mostrar reportes.
<b>Propósito</b>	Permitir al administrador llevar un control de los usuarios que acceden al sitio.
<b>Actores:</b> Administrador	

<b>Resumen:</b> El caso de uso se inicia cuando el administrador escoge la opción Mostrar reportes, dentro del cual se puede mostrar los usuarios online, reportes según un usuario y según un IP. Este finaliza cuando se han efectuado las acciones de control de reportes.	
<b>Referencias</b>	R5
<b>Precondiciones</b>	El usuario debe haberse autenticado y poseer privilegios de administrador.
<b>Poscondiciones</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El administrador escoge la opción Reportes.	1.1 El sistema brinda 3 opciones: a) Si elige la opción Usuarios online, ver sección- 1. b) Si elige la opción Reportes por IP, ver sección- 2. c) Si elige la opción Reportes por usuario, ver sección- 3.
<b>Sección- 1. Usuarios online</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El administrador escoge la opción Usuarios Online.	1.1. El sistema busca todos los usuarios que están conectados a la aplicación en ese momento. 1.2 El sistema muestra los datos de los usuarios online conectados (nombre del usuario, correo, privilegio, último módulo).
<b>Sección- 2. Reportes por IP</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El administrador accede a la opción Reportes por IP.	1.1. El sistema le pide la dirección de IP que desea controlar.
2. El administrador entra la solicitud del sistema.	2.1. El sistema busca el IP especificado. 2.2 El sistema muestra los datos correspondientes a dicha dirección (usuario, fecha y módulo).
Flujo alternativo	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

	1.1 Si la dirección IP no está correcta o no exista se emite el mensaje de error “El IP no es válido”.
<b>Sección- 3. Reportes por Usuario</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El administrador accede a la opción Reportes por Usuario.	1.1. El sistema le pide el nombre de usuario que desea verificar.
2. El administrador entra la solicitud del sistema.	2.1. El sistema busca el usuario que desea controlar. 2.2. El sistema muestra los datos correspondientes a dicho usuario (IP, fecha y módulo).
Flujo alternativo	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	1.1 Si el usuario no existe se emite el mensaje de error “El usuario no es válido”.

**Tabla 7. Descripción Textual <CU Configurar tipo de autenticación>**

<b>Caso de uso</b>	
CU-6	Configurar tipo de autenticación.
<b>Propósito</b>	Le permite a un administrador configurar el sistema en cuanto a la forma de entrada al sitio.
<b>Actores:</b> Administrador	
<b>Resumen:</b> El caso de uso se inicia cuando el administrador escoge la opción configurar tipo de autenticación para escoger por LDAP o manual. Este finaliza cuando se han efectuado las acciones de configuración.	
<b>Referencias</b>	R6
<b>Precondiciones</b>	Este usuario debe poseer privilegios de Administrador.
<b>Poscondiciones</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El administrador accede a la opción Configurar autenticación.	1.1 El sistema brinda 2 opciones: a) Si elige la opción LDAP, ver sección – 1. b) Si elige la opción Manual, ver sección – 2.
<b>Sección- 1. LDAP</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

1. El administrador accede a un formulario de configuración de LDAP.	1.1 El sistema le muestra los campos que debe llenar.
2. El administrador llena los campos y activa la opción de configuración por LDAP.	2.1 El sistema guarda la información y notifica el resultado con el mensaje “Se configuró correctamente”.
<b>Sección- 2. Manual</b>	
1. El administrador accede a la opción Manual.	1.1 El sistema le muestra los campos que debe llenar.
2. El administrador llena los campos.	2.1 El sistema guarda la información y notifica el resultado con el mensaje “Se configuró correctamente”.

**Tabla 8. Descripción Textual <CU Configurar HTTPs >**

<b>Caso de uso</b>	
CU-7	Configurar HTTPs
<b>Propósito</b>	Le permite a un administrador configurar la versión segura del protocolo HTTP.
<b>Actores:</b> Administrador	
<b>Resumen:</b> El caso de uso se inicia cuando el administrador escoge la opción Configuración HTTPs y finaliza una vez que se hayan efectuado las acciones de configuración.	
<b>Referencias</b>	R7
<b>Precondiciones</b>	Este usuario debe poseer privilegios de Administrador.
<b>Poscondiciones</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El administrador accede a la opción HTTPs para activar el protocolo seguro.	1.1 El sistema le muestra un cuadro de selección para que el administrador decida si desea utilizar el servicio HTTPs o no.
2. El administrador escoge dicha opción.	2.1 El sistema activa el protocolo seguro.

**Tabla 9. Descripción Textual <CU Gestionar mensajes >**

<b>Caso de uso</b>	
CU-8	Gestionar mensajes
<b>Propósito</b>	Permite al autor usar el servicio de mensajería interna para comunicarse con otros usuarios.
<b>Actores:</b> Autor	

<b>Resumen:</b> El caso de uso se inicia cuando el autor escoge la opción Mensajes, dentro del cual se puede crear nuevos mensajes, eliminarlos, entre otras cosas y finaliza el caso de uso cuando se han efectuado las acciones del servicio.	
<b>Referencias</b>	R8
<b>Precondiciones</b>	El usuario debe haberse autenticado. El autor debe haber entrado a Mensajes.
<b>Poscondiciones</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El autor escoge la opción Mensajes.	1.1 El sistema brinda 3 opciones: a) Si elige la opción Crear mensajes, ver sección-1. b) Si elige la opción Mostrar mensajes, ver sección- 2. c) Si elige la opción Eliminar mensajes, ver sección- 3.
<b>Sección- 1. Crear mensajes</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El autor selecciona la opción Nuevo mensaje.	1.1. El sistema muestra los campos que deben llenarse para enviar el mensaje.
2. El autor llena los campos solicitados.	2.1 El sistema envía el mensaje.
Flujo alternativo	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	2.1 Si no se especifica el campo del destinatario en el caso de enviar mensaje el sistema muestra “Debe seleccionar un usuario válido”.
<b>Sección- 2. Mostrar mensajes</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El autor escoge la opción Mostrar mensajes recibidos.	1.1 El sistema muestra una interfaz mostrando todos los mensajes recibidos de dicho usuario.
Flujo alternativo	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

1. El autor escoge la opción Mostrar mensajes enviados.	1.1 El sistema muestra una interfaz mostrando todos los mensajes enviados por dicho usuario.
<b>Sección- 3. Eliminar mensajes</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El autor selecciona la opción Mostrar mensajes.	1.1. El sistema muestra una lista con todos los mensajes.
2. El autor escoge los mensajes a eliminar.	2.1 El sistema elimina todos los mensajes marcados.

Tabla 10. Descripción Textual <CU Administrar interfaz gráfica >

<b>Caso de uso</b>	
CU-9	Administrar interfaz gráfica
<b>Propósito</b>	Permite al administrador configurar la interfaz gráfica de la aplicación ya sea para ocultar, eliminar o agregar bloques o para configurar el texto de bienvenida de la herramienta.
<b>Actores:</b> Administrador	
<b>Resumen:</b> El caso de uso se inicia cuando el administrador escoge la opción Interfaz gráfica, dentro del cual se puede añadir o eliminar bloques y hacer cambios en la configuración del texto de inicio y finaliza el caso de uso cuando se han efectuado las acciones que conllevan a la administración de la interfaz gráfica.	
<b>Referencias</b>	R9
<b>Precondiciones</b>	El usuario debe haberse autenticado. El administrador debe haber entrado a la opción Activar edición.
<b>Poscondiciones</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

1. El administrador accede a la opción Interfaz gráfica.	1.1 El sistema le muestra varias opciones: a) Si elige la opción Mostrar bloques, ver sección-1. b) Si elige la opción Ocultar bloques, ver sección-2. c) Si elige la opción Añadir bloques, ver sección-3. d) Si elige la opción Eliminar bloques, ver sección-4. e) Si elige la opción Mover bloques, ver sección-5. f) Si elige la opción Configurar texto inicio, ver sección-6.
<b>Sección- 1. Mostrar _ bloques</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El administrador accede a la opción Mostrar de un bloque en específico.	1.1 El sistema muestra el bloque escogido.
<b>Sección- 2. Ocultar _ bloques</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El administrador accede a la opción Ocultar del bloque que él considere.	1.1 El sistema esconde el bloque escogido para que los usuarios no tengan acceso a él.
<b>Sección- 3. Añadir _ bloques</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El administrador accede a la opción Añadir bloques.	1.1 El sistema muestra los nombres de los diferentes bloques que existen en la aplicación.
2. El administrador escoge uno de los bloques.	2.1 El sistema guarda los resultados realizados y muestra dicho bloque.
<b>Sección- 4. Eliminar _ bloques</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El administrador accede a la opción Eliminar bloques.	1.1 El sistema elimina el bloque deseado.
<b>Sección- 5. Mover _ bloques</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

1. El administrador escoge la opción de mover bloques, ya sea para subir, bajar, mover a la derecha o a la izquierda.	1.1 El sistema cambia de posición el bloque escogido y actualiza los cambios.
<b>Sección- 6. Configurar texto de inicio</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El administrador accede a la opción Editar bienvenida.	1.1 El sistema muestra una página HTML para crear texto de bienvenida y un browser para subir imágenes, si es necesario.
2. El administrador crea el texto de bienvenida y/o inserta imágenes.	2.1 El sistema actualiza la aplicación con el texto de bienvenida creado.

**Tabla 11. Descripción Textual <CU crear paquete de contenidos >**

<b>Caso de uso</b>	
CU-10	Crear paquete de contenidos.
<b>Propósito</b>	Permite al autor crear sus propios paquetes y tenerlos así de una forma mejor organizada para una posterior reutilización.
<b>Actores:</b> Autor	
<b>Resumen:</b> El caso de uso se inicia cuando el autor escoge la opción Crear paquetes, dentro del cual se pueden realizar otras funciones como subir recursos o eliminarlos y finaliza el caso de uso cuando se han efectuado las acciones que conllevan a la creación de un paquete.	
<b>Referencias</b>	R10
<b>Precondiciones</b>	El usuario debe haberse autenticado.
<b>Poscondiciones</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El autor accede a la opción Crear paquete.	<p>1.1 El sistema verifica que la cantidad de paquetes que tiene dicho autor en su área de trabajo no exceda de la cantidad que el administrador definió que debía tener cada autor.</p> <p>1.2 El sistema brinda 4 opciones:</p> <p>a) Si desea subir un recurso, ver sección-1.</p> <p>b) Si desea eliminar algún recurso ir a la sección - 2.</p> <p>b) Si desea crear una organización ir a la sección - 3.</p>

	c) Si desea crear un ítem ir a la sección - 4. d) Si desea relacionar un ítem con un recurso ir a la sección – 5.
Flujo alternativo	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	1.1 Si el área de trabajo del autor esta lleno, el sistema envía el mensaje “No puede crear ningún paquete porque ha excedido el número de paquetes que debe tener”.
<b>Sección- 1. Subir recurso</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El autor selecciona la opción Importar.	1.1. El sistema pide la ubicación del recurso a subir.
2. El autor introduce la dirección del recurso.	2.1 El sistema sube el recurso y muestra el mensaje “El archivo fue subido con éxito”.
<b>Sección- 2. Eliminar recurso</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El autor escoge el recurso que desea eliminar y selecciona la opción de eliminar.	1.1 El sistema lo elimina y se actualiza el sistema.
<b>Sección- 3. Añadir nueva organización</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El autor escoge la opción Añadir organización.	1.1 El sistema solicita un nombre adecuado para la organización.
2. El autor introduce el nombre que va a tener la organización.	2.1 El sistema crea la nueva organización.
<b>Sección- 4. Añadir nuevo ítem</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El autor marca la organización a la que quiere insertarle un ítem y escoge la opción Añadir ítem.	1.1 El sistema muestra un campo para introducir el nombre al nuevo ítem.
2. El autor le pone un nombre adecuado al ítem.	2.1 El sistema crea el nuevo ítem.
<b>Sección- 5. Relacionar ítem-recurso.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

1. El autor escoge un ítem para asignarle un recurso.	1.1 El sistema muestra la lista de recursos subidos para escoger el que se le va a asignar al ítem seleccionado.
2. El autor escoge el recurso.	2.1 El sistema realiza el vínculo del ítem al recurso.

**Tabla 12. Descripción Textual <CU gestionar paquete >**

<b>Caso de uso</b>	
CU-11	Gestionar paquete
<b>Propósito</b>	Permite al autor modificar o eliminar el paquete una vez creado.
<b>Actores:</b> Autor	
<b>Resumen:</b> El caso de uso se inicia cuando el autor escoge la opción Mis Paquetes, y selecciona un paquete para realizarle modificaciones o eliminarlo (cuando el paquete haya sido descargado o subido al repositorio). Este finaliza cuando se han efectuado las acciones realizadas sobre el paquete.	
<b>Referencias</b>	R11, R12
<b>Precondiciones</b>	El usuario debe haberse autenticado. El autor debe haber creado un paquete.
<b>Poscondiciones</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El autor accede a la opción Mis paquetes.	1.1 El sistema le muestra todos los paquetes creados por dicho autor. a) Si desea modificar un paquete, ver sección - 1. b) Si desea eliminar un paquete, ver sección - 2.
<b>Sección- 1. Modificar_ paquete</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El autor escoge un paquete para hacerle modificaciones.	1.1 El sistema muestra la estructura interna del paquete con todos sus recursos.
2. El autor realiza los cambios ya sea a los recursos, a la estructura o a los metadatos.	2.1 El sistema guarda los cambios.
<b>Sección- 2. Eliminar_ paquete</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El autor escoge el paquete que desea	1.1 El sistema elimina el paquete.

eliminar.	
-----------	--

**Tabla 13. Descripción Textual <CU Gestionar metadatos >**

<b>Caso de uso</b>	
CU-12	Gestionar metadatos
<b>Propósito</b>	Permite al usuario ponerle metadatos al paquete y a cada recurso para su posterior reutilización.
<b>Actores:</b> Autor	
<b>Resumen:</b> El caso de uso se inicia cuando el autor escoge la opción Metadatos, ya sea del paquete o del recurso para realizar el llenado de metadatos; Este finaliza una vez se haya llenado los campos, especialmente los obligatorios.	
<b>Referencias</b>	R13, R14
<b>Precondiciones</b>	El usuario debe haberse autenticado. El autor debe haber subido un recurso o creado un paquete.
<b>Poscondiciones</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El autor accede a la opción Metadatos.	1.1 El sistema le brinda 2 opciones: a) Si desea agregarle metadatos a un paquete, ver sección - 1. b) Si desea modificar los metadatos a un paquete, ver sección - 2.
<b>Sección- 1. Agregar metadatos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El autor escoge la opción Metadatos ya sea del recurso o del paquete.	1.1 El sistema le muestra todos los campos que debe llenar el autor.
2. El autor introduce los datos.	2.1 El sistema guarda los metadatos llenados por el autor.
Flujo alternativo	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	1.1 Si el autor no llena aquellos campos que son obligatorios entonces el sistema notifica el mensaje "Debe llenar todos los campos obligatorios (*)".
<b>Sección- 2. Modificar metadatos</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

1. El autor accede a la opción Metadatos del recurso o del paquete.	1.1 El sistema le muestra todas las categorías de metadatos.
2. El autor modifica algunos campos de metadatos.	2.1 El sistema actualiza los cambios realizados.
Flujo alternativo	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	1.1 Si el autor no llena aquellos campos que son obligatorios entonces el sistema notifica el mensaje “Debe llenar todos los campos obligatorios (*)”.

Tabla 14. Descripción Textual <CU Mostrar vista previa del paquete >

<b>Caso de uso</b>	
CU-13	Mostrar vista previa del paquete.
<b>Propósito</b>	Permite que el autor pueda observar la estructura interna del paquete.
<b>Actores:</b> Autor	
<b>Resumen:</b> El caso de uso se inicia cuando el autor escoge la opción Visualizar paquete para ver el contenido interno del mismo y finaliza el caso de uso cuando se concluye la acción de visualizar.	
<b>Referencias</b>	R15
<b>Precondiciones</b>	El usuario debe haberse autenticado. El autor debe haber creado un paquete.
<b>Poscondiciones</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El autor accede a la opción Mis paquetes.	1.1 El sistema muestra todos los paquetes realizados por dicho usuario.
2. El autor selecciona la opción visualizar, del paquete que Este desea observar.	2.1 El sistema muestra la organización interna del paquete.

Tabla 15. Descripción Textual <CU Generar paquete >

<b>Caso de uso</b>	
CU-14	Generar paquete.
<b>Propósito</b>	Permitir al usuario generar el paquete en un fichero .Zip.
<b>Actores:</b> Autor	

<b>Resumen:</b> El caso de uso se inicia cuando el autor termina de conformar su paquete y desea compactarlo, accediendo a la opción Generar paquete y Este finaliza una vez el autor haya compactado el paquete en un fichero .Zip.	
<b>Referencias</b>	R16
<b>Precondiciones</b>	El usuario debe haberse autenticado.
<b>Poscondiciones</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El autor accede a la opción Generar paquete.	1.1 El sistema convierte el paquete en un fichero .Zip.

Tabla 16. Descripción Textual <CU Descargar paquete >

<b>Caso de uso</b>	
CU-15	Descargar paquete
<b>Propósito</b>	Permitir al autor una vez creado el paquete poder descargarlo para su máquina.
<b>Actores:</b> Autor	
<b>Resumen:</b> El caso de uso se inicia cuando el autor escoge la opción Descargar paquete dándole la posibilidad de llevar el mismo a su máquina y finaliza luego de haber descargado el paquete.	
<b>Referencias</b>	R17
<b>Precondiciones</b>	El usuario debe haberse autenticado.
<b>Poscondiciones</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El autor escoge la opción Descargar paquete.	1.1 El sistema le solicita al autor la dirección del lugar donde desea guardar el paquete.
2. El autor introduce la dirección.	2.1 El sistema notifica el resultado de la operación.

Tabla 17. Descripción Textual <CU Crear contenido >

<b>Caso de uso</b>	
CU-16	Crear contenido
<b>Propósito</b>	Permite al autor crear sus propios contenidos, ya sea crear ficheros de textos como elaborar preguntas de verdaderos y falsos, de elección múltiple, entre otras cosas.
<b>Actores:</b> Autor	

<b>Resumen:</b> El caso de uso se inicia cuando el autor escoge la opción Crear contenido, dentro del cual se elaboran ficheros de textos, se insertan y editan tablas, imágenes; además de poder crear sus propias evaluaciones y finaliza luego de haber creado el contenido que va dentro del paquete.	
<b>Referencias</b>	R18
<b>Precondiciones</b>	El usuario debe haberse autenticado.
<b>Poscondiciones</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El autor accede a la opción Crear contenido.	1.1 El sistema muestra 2 opciones: a) Si elige la opción Crear ficheros textos, ver sección-1. b) Si elige la opción Crear preguntas, ver sección-2.
<b>Sección- 1. Crear ficheros textos.</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El autor accede a la opción Crear ficheros textos.	1.1 El sistema muestra una interfaz para elaborar el contenido con algunas opciones como para insertar objetos multimedia (imágenes, videos, animaciones), tablas, expresiones matemáticas y enlaces (externos o internos).
2. El autor crea el contenido con las opciones necesarias.	2.1 El sistema guarda el contenido creado.
<b>Sección-2. Crear preguntas</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El autor accede a la opción Crear Preguntas.	1.1 El sistema le muestra los diferentes tipos de preguntas que existen: - Confeccionar preguntas de espacios en blanco. -Confeccionar preguntas de elección múltiple. -Confeccionar preguntas de verdadero y falso.
2. El autor escoge uno de los 3 tipos de preguntas que existen.	2.1 El sistema muestra la interfaz para elaborar las preguntas.
3. El autor elabora las preguntas.	3.1 El sistema guarda los cambios y se actualiza.

Tabla 18. Descripción Textual <CU Gestionar contenido >

Caso de uso	
CU-17	Gestionar contenido
<b>Propósito</b>	Permite al autor una vez creado los contenidos poder modificar o eliminar algunos.
<b>Actores:</b> Autor	
<b>Resumen:</b> El caso de uso se inicia cuando el autor escoge la opción Mis contenidos para modificar o eliminar algunos contenidos y finaliza luego de haber realizado la acción.	
<b>Referencias</b>	R19, R20
<b>Precondiciones</b>	El usuario se debe haber autenticado con éxito.
<b>Poscondiciones</b>	
Acción del actor	Respuesta del sistema
1. El autor accede a la opción Mis contenidos.	1.1 El sistema le muestra los contenidos que ya existen. a) Si elige la opción modificar un contenido específico, ver sección-1. b) Si elige la opción eliminar un contenido específico, ver sección-2.
<b>Sección- 1. Modificar_ contenido</b>	
Acción del actor	Respuesta del sistema
1. El autor escoge uno de los contenidos ya creados y le realiza algunos cambios.	1.1 El sistema guarda los cambios realizados.
<b>Sección- 2. Eliminar_ contenido</b>	
Acción del actor	Respuesta del sistema
1. El autor escoge el contenido que desea eliminar.	1.1 El sistema elimina dicho contenido.

Tabla 19. Descripción Textual <CU Exportar paquete >

Caso de uso	
CU-18	Exportar paquete.
<b>Propósito</b>	Exportar el paquete directo al repositorio sin necesidad de descargarlo a la máquina local.
<b>Actores:</b> Autor.	
<b>Resumen:</b> El caso de uso se inicia cuando el autor decide exportar el paquete al repositorio y finaliza luego de haber realizado la acción.	

<b>Referencias</b>	R21
<b>Precondiciones</b>	El usuario se debe haber autenticado con éxito.
<b>Poscondiciones</b>	El paquete es exportado.
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El autor selecciona el paquete a exportar.	
2. El autor accede a la opción Exportar paquete.	2.1 El sistema envía el paquete al repositorio. 2.2 El sistema almacena el paquete en el repositorio. 2.3 El sistema muestra el mensaje “El paquete fue exportado con éxito”.

Tabla 20. Descripción Textual <CU Recuperar contenido >

<b>Caso de uso</b>	
CU-19	Recuperar contenido
<b>Propósito</b>	Permite al autor buscar en el repositorio algún contenido para reutilizarlo en un nuevo paquete.
<b>Actores:</b> Autor	
<b>Resumen:</b> El caso de uso se inicia cuando el autor escoge la opción Buscar contenido para utilizar ese contenido en algún nuevo paquete y finaliza luego de haber recuperado el contenido.	
<b>Referencias</b>	R22
<b>Precondiciones</b>	El usuario se debe haber autenticado con éxito.
<b>Poscondiciones</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El autor accede a opción Buscar contenido.	1.1 El sistema visualiza una ventana con varios criterios de búsqueda.
2. El autor completa los campos según los criterios de búsqueda por los cuales desea recuperar los contenidos.	2.1 El sistema se comunica con el repositorio para realizar búsquedas según los criterios.
	3. El sistema visualiza el resultado de la búsqueda.
Flujo alternativo	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>

	2.1 Si no encontró nada muestra un mensaje “No se ha encontrado contenido con ese criterio”.
--	--

Tabla 21. Descripción Textual <CU Configurar metadatos >

Caso de uso	
CU-20	Configurar metadatos
<b>Propósito</b>	Permite al administrador definir los metadatos que van a ser obligatorios o los que se van a autocompletar.
<b>Actores:</b> Administrador	
<b>Resumen:</b> El caso de uso se inicia cuando el administrador escoge la opción Configurar_ metadatos para definir qué metadatos el usuario debe llenar obligatoriamente y cuáles son los que se van a generar. Este finaliza luego de haber definido dichos metadatos.	
<b>Referencias</b>	R23
<b>Precondiciones</b>	El usuario se debe haber autenticado con éxito y tener el rol de administrador.
<b>Poscondiciones</b>	
Acción del actor	Respuesta del sistema
1. El administrador accede a la opción Configurar Metadatos.	1.1 El sistema brinda dos opciones: a) Definir metadatos a autocompletar, ver sección - 1. b) Definir metadatos obligatorios, ver sección – 2.
<b>Sección- 1.</b> Definir metadatos a autocompletar.	
Acción del actor	Respuesta del sistema
1. El administrador selecciona la opción “Definir metadatos a autocompletar”.	1.1 El sistema muestra una interfaz con todos los metadatos.
2. El administrador selecciona los metadatos y realiza el llenado de aquellos metadatos que él considere que el usuario no deba llenar.	2.1 El sistema guarda los cambios.
<b>Sección- 2.</b> Definir metadatos obligatorios.	
Acción del actor	Respuesta del sistema
1. El administrador selecciona la opción “Definir metadatos obligatorios”.	1.1 El sistema muestra una interfaz con todos los metadatos.

2. El administrador selecciona los metadatos obligatorios a llenar por el usuario.	2.1 El sistema guarda los cambios.
--	------------------------------------

**Tabla 22. Descripción Textual <CU Parametrizar variables de configuración >**

<b>Caso de uso</b>	
CU-21	Parametrizar variables de configuración.
<b>Propósito</b>	Permite al administrador seleccionar la cantidad de paquetes que puede tener cada usuario de la herramienta.
<b>Actores:</b> Administrador.	
<b>Resumen:</b> El caso de uso se inicia cuando el administrador escoge la opción Variables de configuración para definir la cantidad de paquetes que se deben tener por usuario y finaliza luego de haber definido la cantidad.	
<b>Referencias</b>	R24
<b>Precondiciones</b>	El usuario se debe haber autenticado con éxito y tener rol de administrador.
<b>Poscondiciones</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
1. El administrador accede a la opción Variables de configuración.	1.1 El sistema muestra un campo de texto para introducir la cantidad máxima de paquetes que un usuario puede tener en su área de trabajo. La cantidad máxima debe escribirse en formato numérico.
2. El administrador inserta la cantidad de paquetes que debe tener un usuario.	2.1 El sistema guarda la cantidad de paquetes por usuario.

## 2.6 Conclusiones

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 análisis y diseño de la aplicación teniendo en cuenta los requerimientos especificados en el capítulo.

## Capítulo 3: Análisis y Diseño

### 3.1 Introducción

El presente capítulo se centra en el flujo de trabajo Análisis y Diseño, representando mediante un grupo de artefactos la descripción del sistema que sirve de entrada para la implementación del mismo. Se muestran los diagramas de clases del análisis, los diagramas de clases del diseño por extensiones Web y los diagramas de interacción. Además se realiza la descripción de las clases y se diseña la base de datos, necesaria para el almacenamiento de la información.

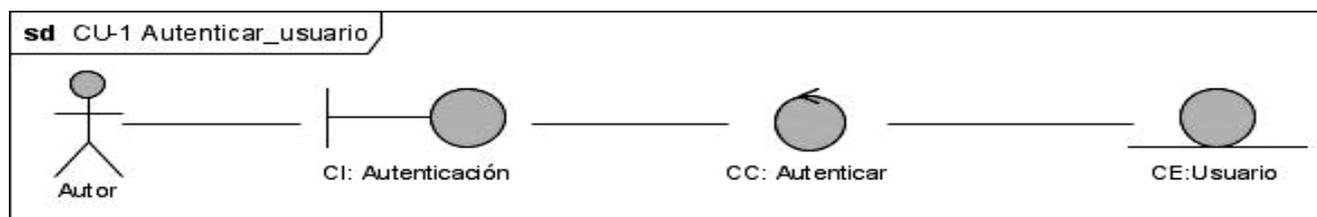
### 3.2 Análisis

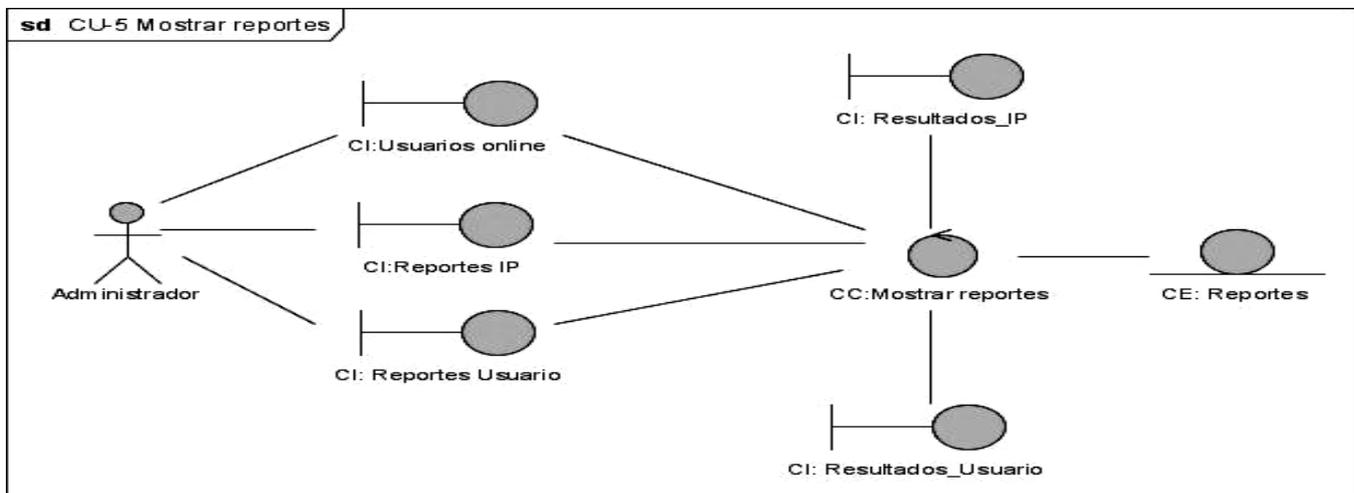
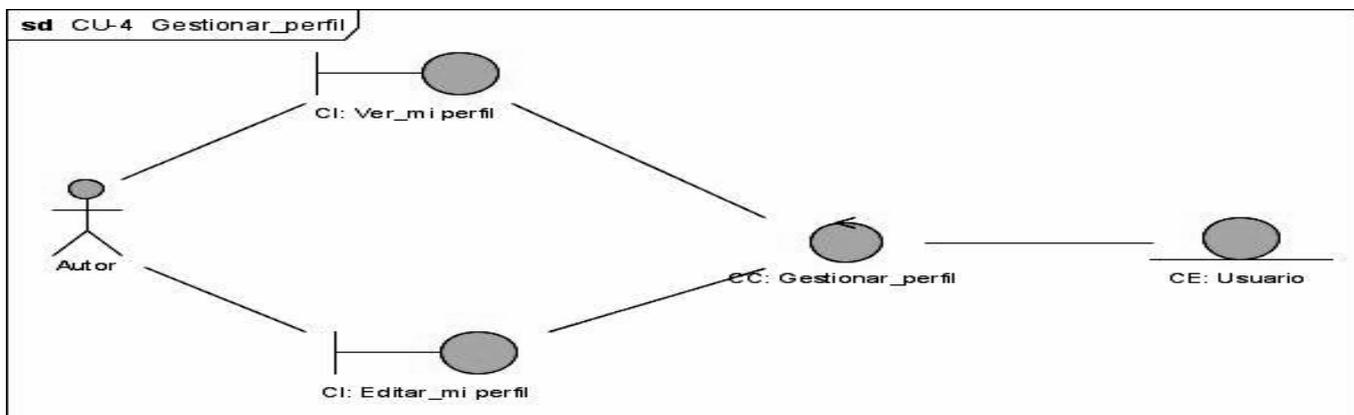
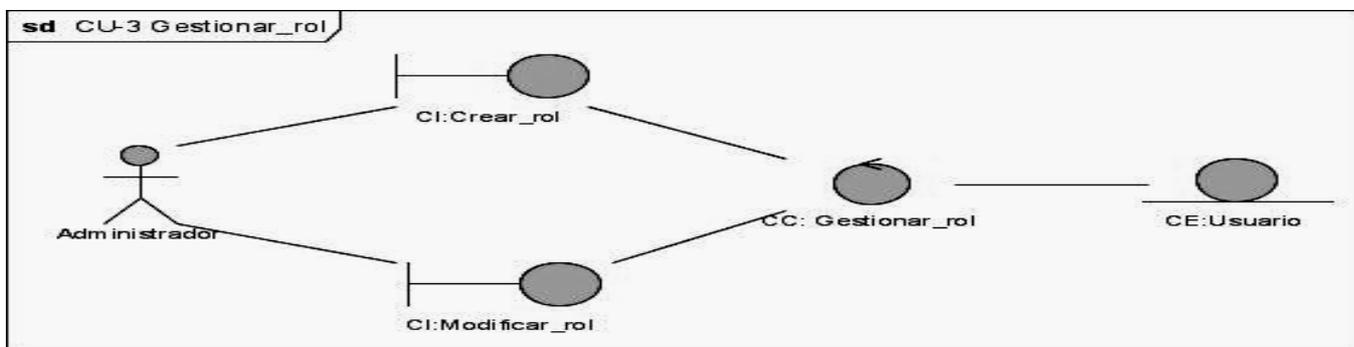
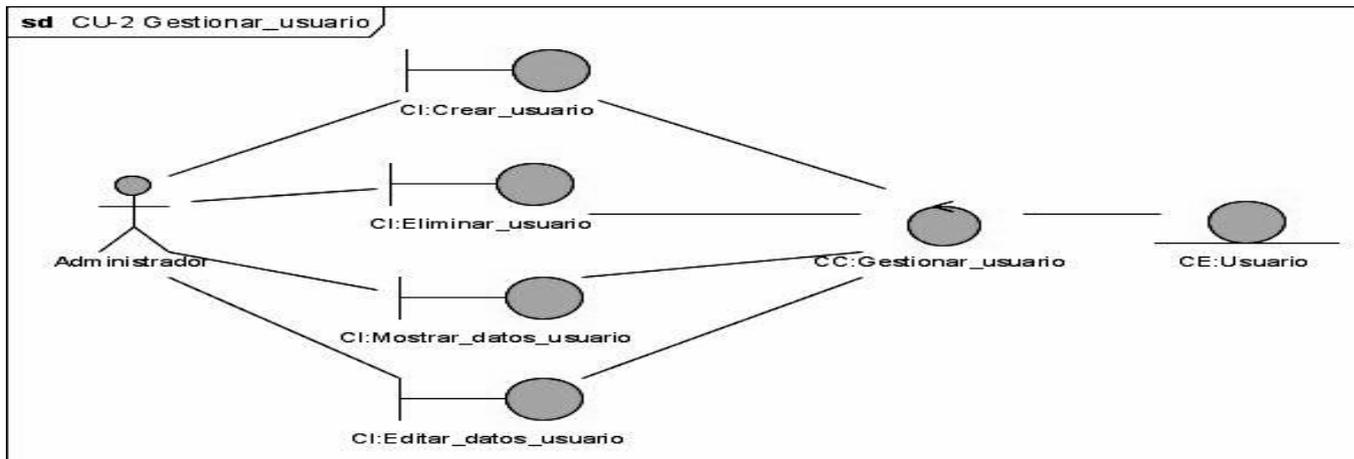
El análisis es uno de los flujos de trabajos desarrollado en la fase de elaboración. Consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que sólo se interesa por los requisitos funcionales.

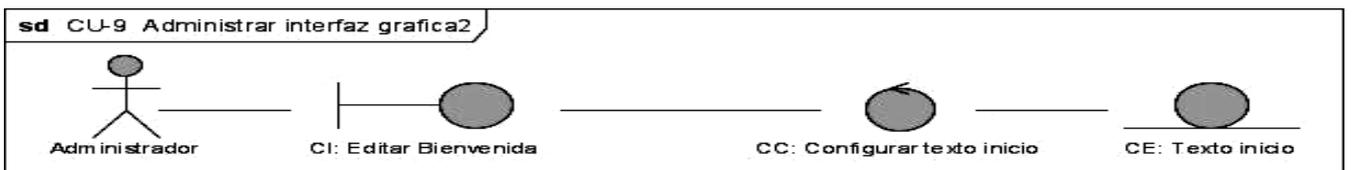
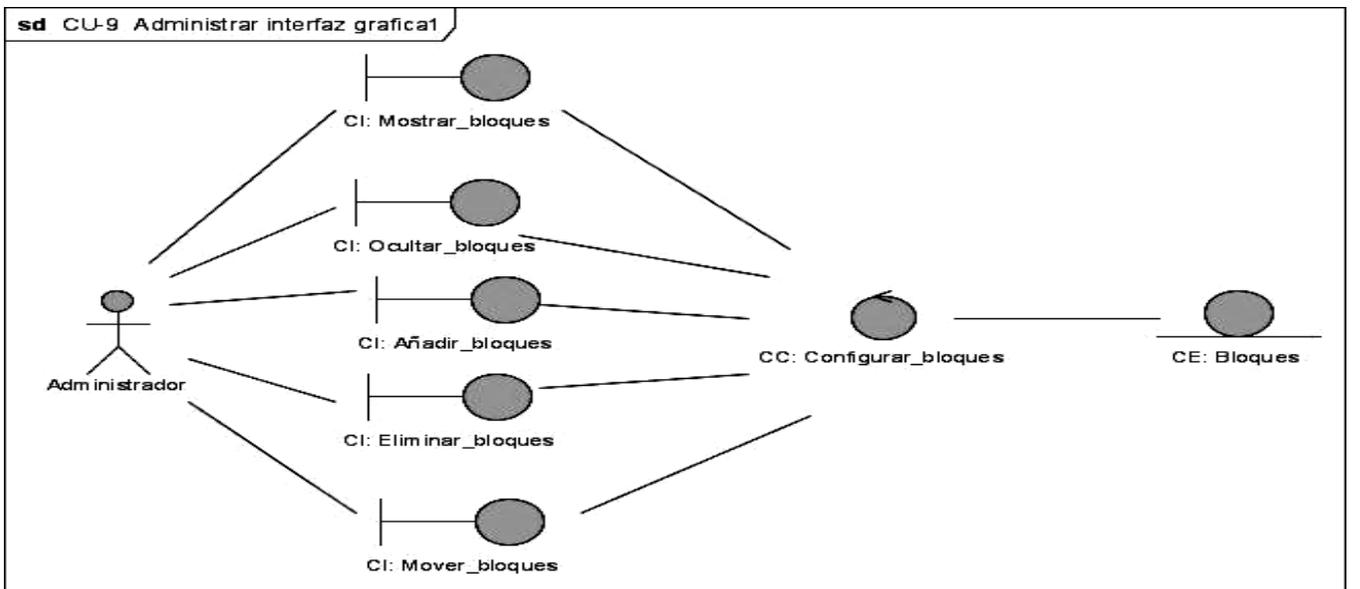
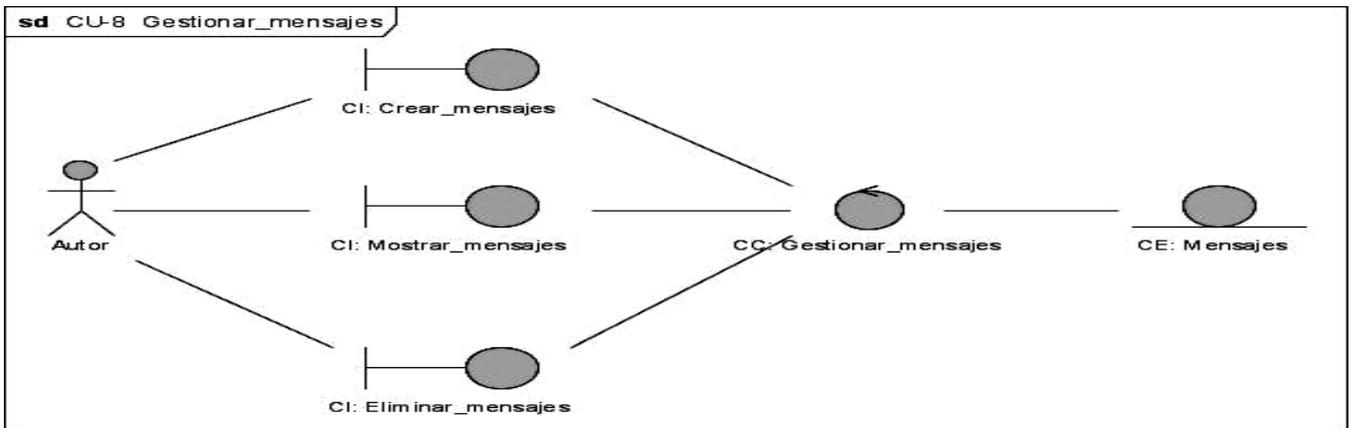
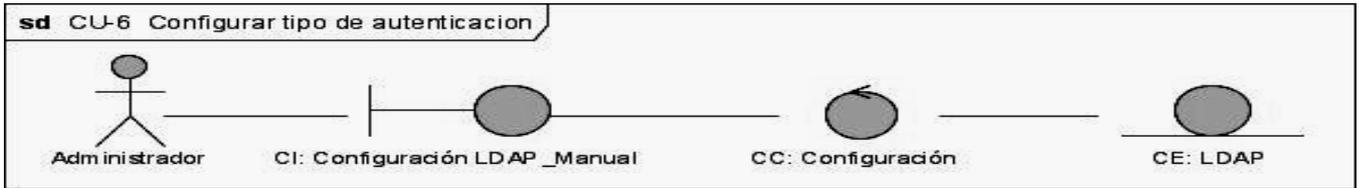
Un diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada de estas cosas.

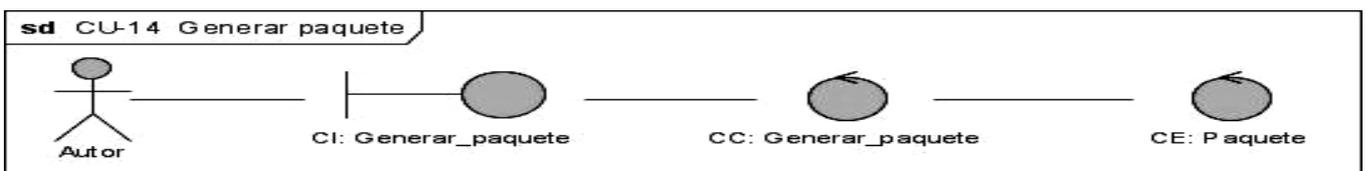
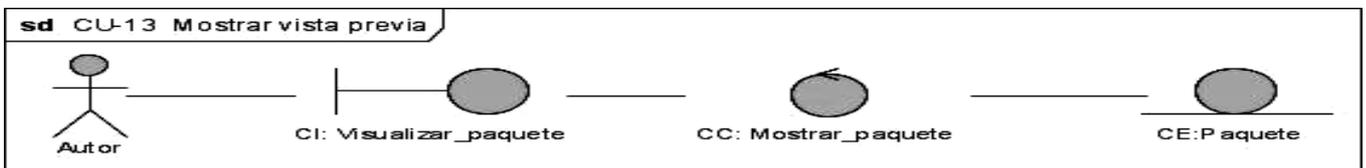
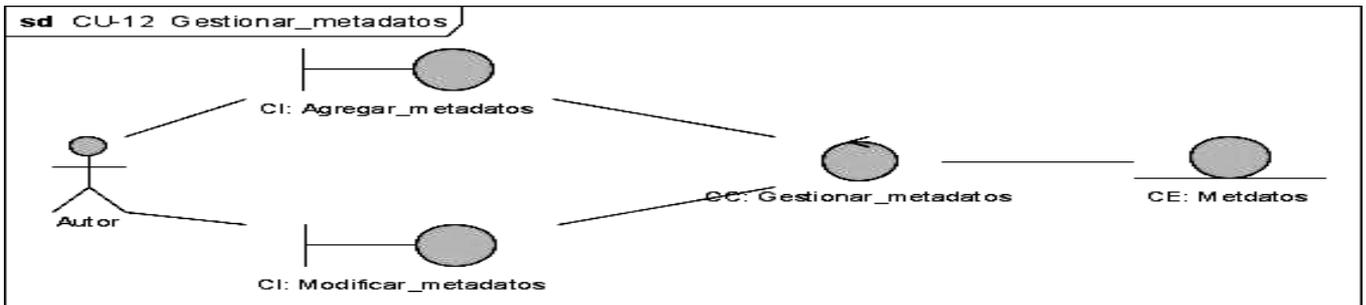
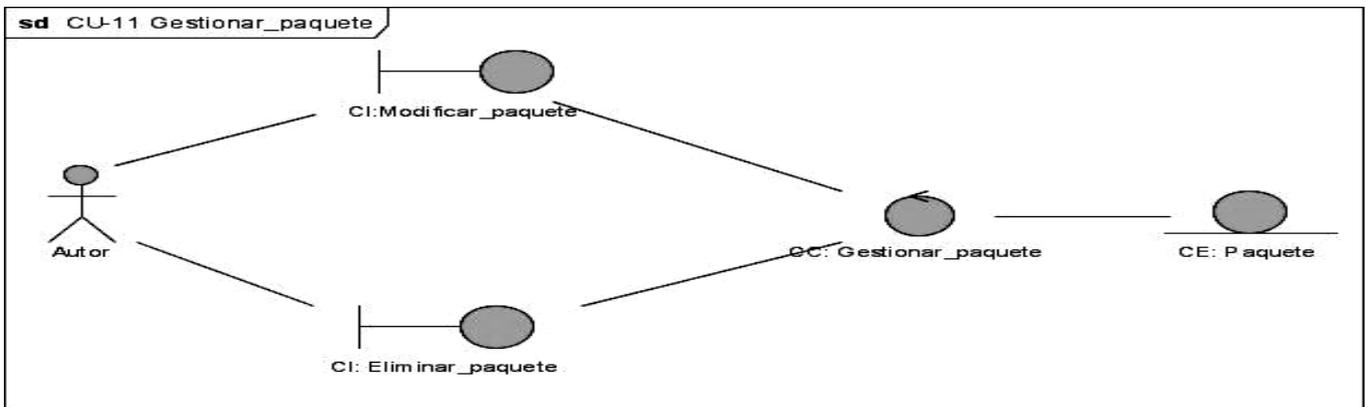
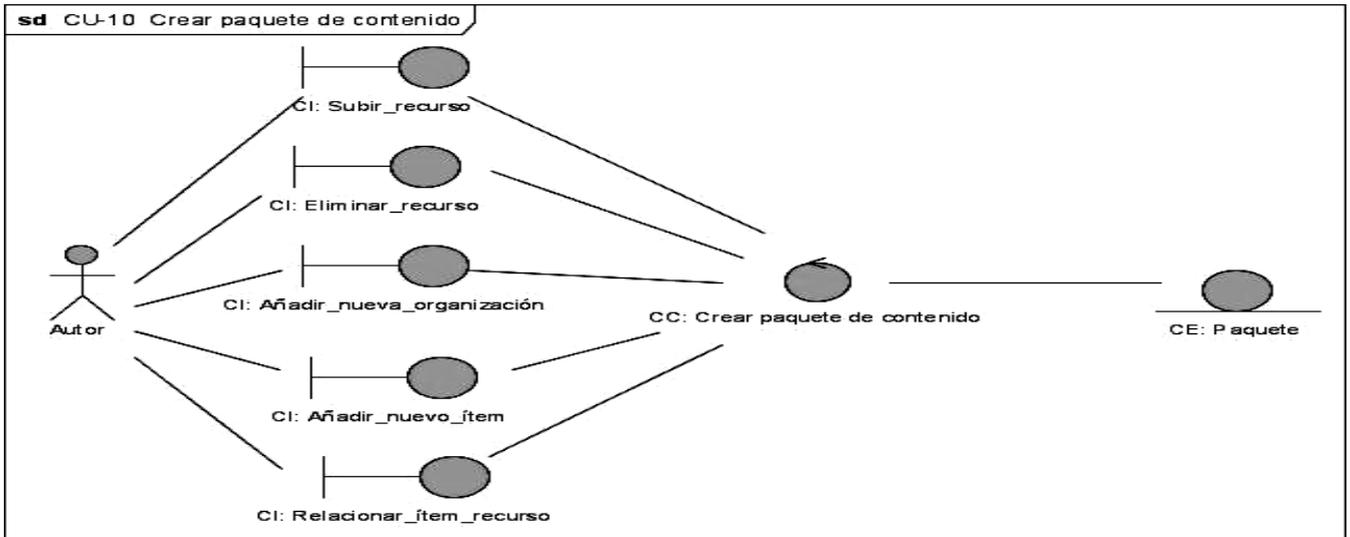
A continuación se refleja el diagrama de clases de análisis correspondiente a los casos de usos descritos.

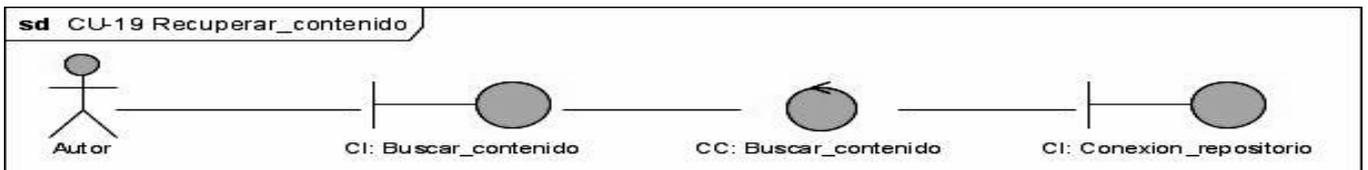
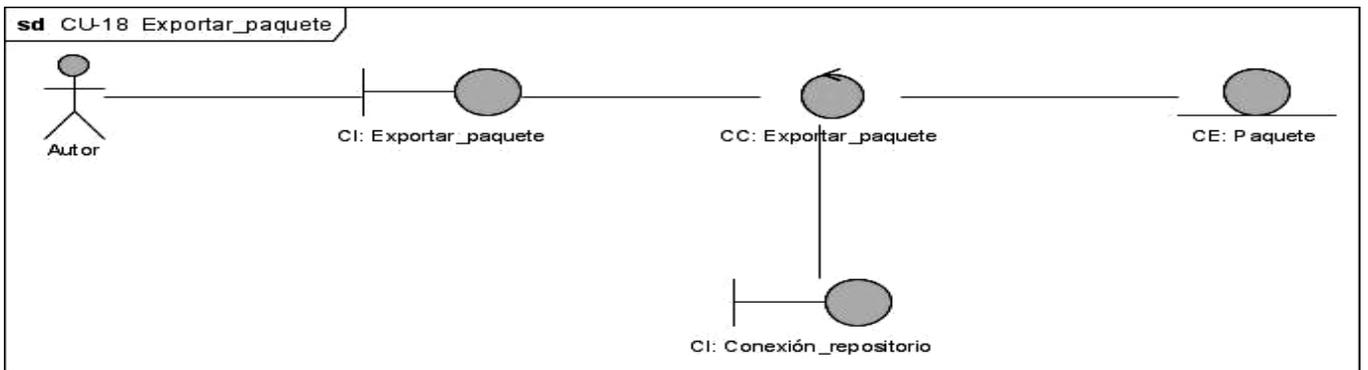
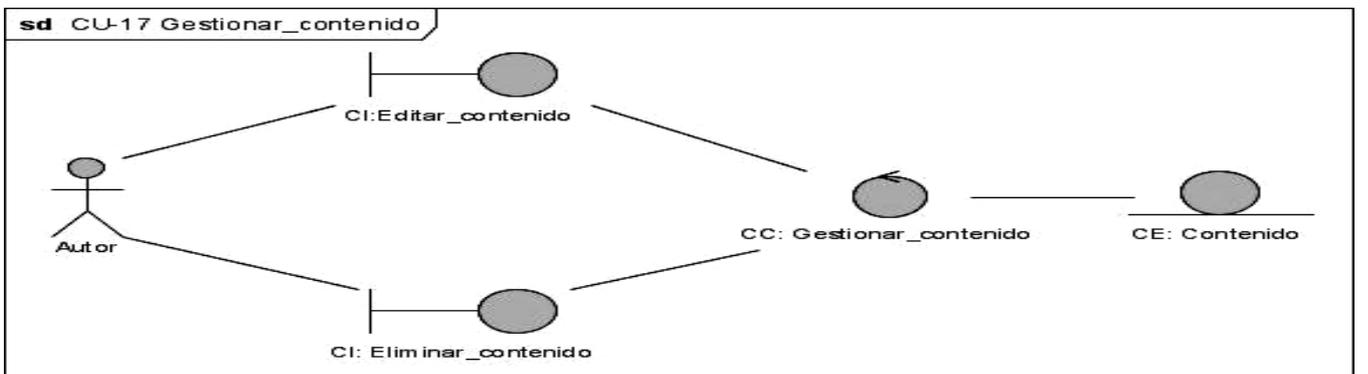
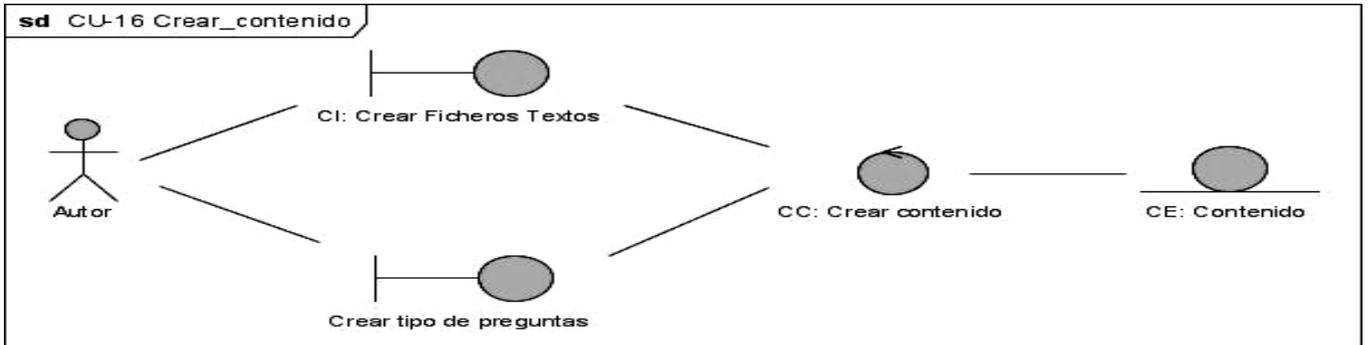
#### 3.2.1 Diagramas de clases del análisis

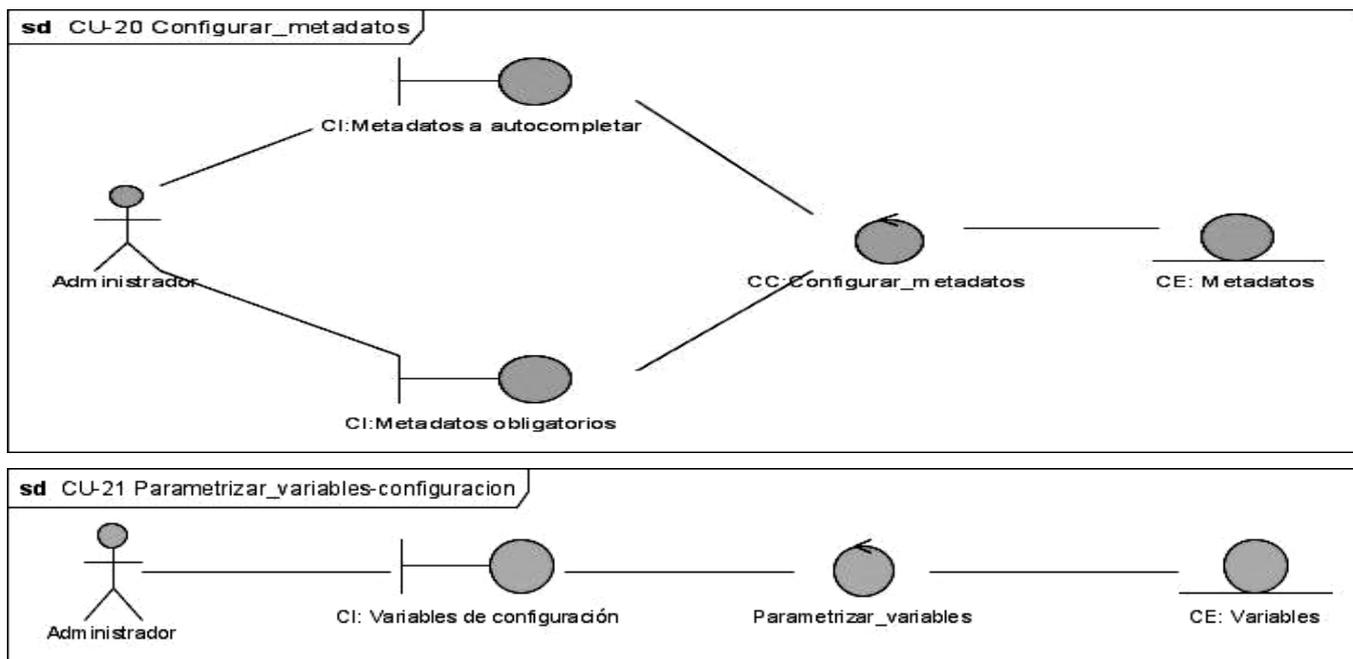












### 3.3 Diseño

El Diseño es otro de los flujos de trabajo que se realiza al final de la fase de elaboración después del análisis. Este es un refinamiento del análisis, es decir es una representación más concreta que el diagrama de clases del análisis, el cual tiene en cuenta los requisitos no funcionales, en definitiva cómo cumple el sistema sus objetivos. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades.

La arquitectura propuesta para el sistema es la Arquitectura Orientada a Servicios (SOA), la cual se compone de 4 capas.

A continuación se mencionan cada una de las clases que son utilizadas en los diagramas del modelo de diseño.

#### Capa de Presentación:

- Página Cliente (CP): Son las páginas que van a funcionar como interfaz a los usuarios. Una instancia de Página Cliente es una página Web, con formato HTML. Mezcla de datos, presentación y lógica. Son interpretadas por el navegador.

- Formulario (FR): Colección de elementos de entrada que son parte de una página cliente. Sus atributos son los elementos de entrada del formulario (input boxes, text areas, radio buttons, check boxes).

- Página Servidora (SP): Representa la página Web que tiene código que se ejecuta en el servidor. Este código interactúa con recursos en el servidor. Son las páginas que construyen a las páginas clientes.

### **Capa de Servicios:**

- Clase de Servicios (CS): Son las clases que brindan los servicios necesarios a la capa de presentación. Contienen la lógica de la aplicación, en forma de transacciones de negocio. Realiza una interfaz de servicio donde se exponen todos los servicios necesarios para la primera capa referidos a la lógica, lo que permite separar físicamente ambas capas.

### **Capa de Dominio**

- Entidad del Negocio (BE): Son las entidades del negocio, clases que contienen todos los datos, no tienen lógica de negocio. Pueden ser objetos (individuales o colecciones).

- Clase de Negocio (BC): Son las clases que contienen las reglas del negocio de la aplicación relacionadas con una entidad del negocio.

### **Capas de Persistencia:**

- Las clases de datos se representan a través de un paquete Acceso a Datos, el cual realiza una interfaz de persistencia. Las clases de datos se explican más adelante en el modelo de datos.

### 3.3.1 Diagrama de Clases del Diseño.

Diagrama de clases del diseño < CU Autenticar\_usuario >

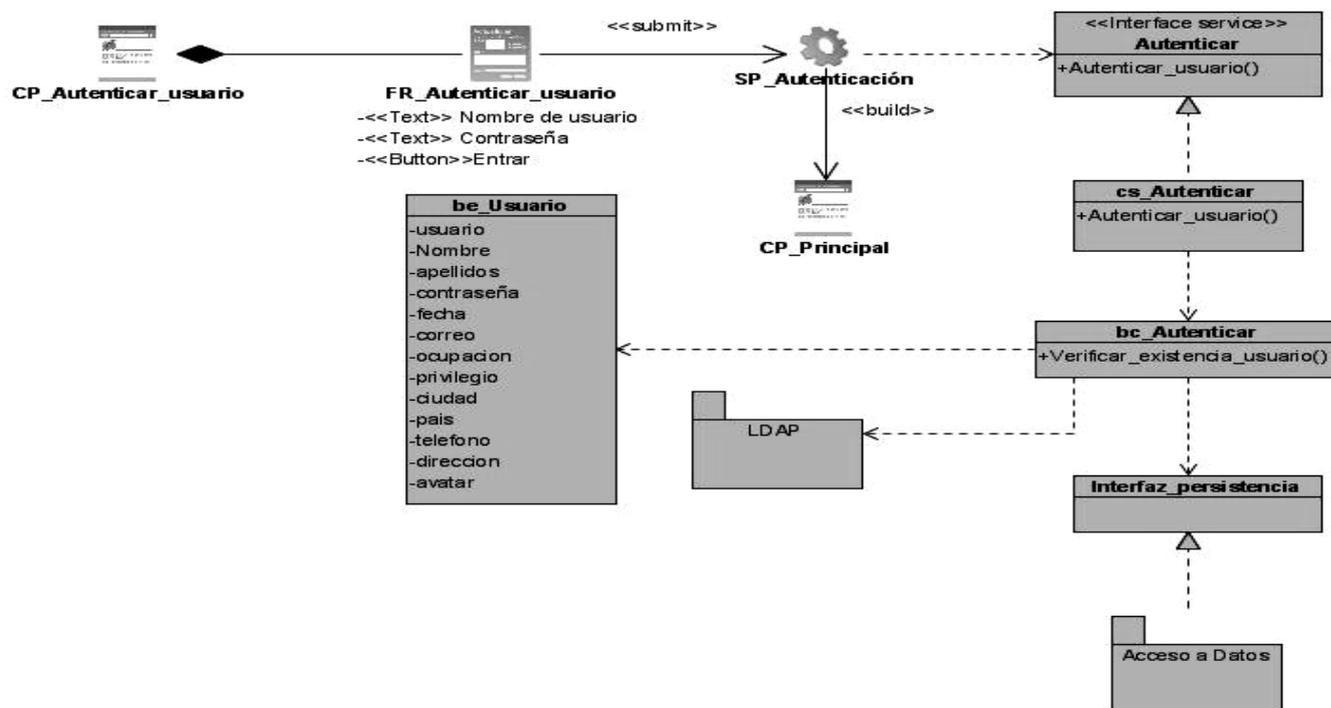


Diagrama de clases del diseño < CU Gestionar\_perfil >

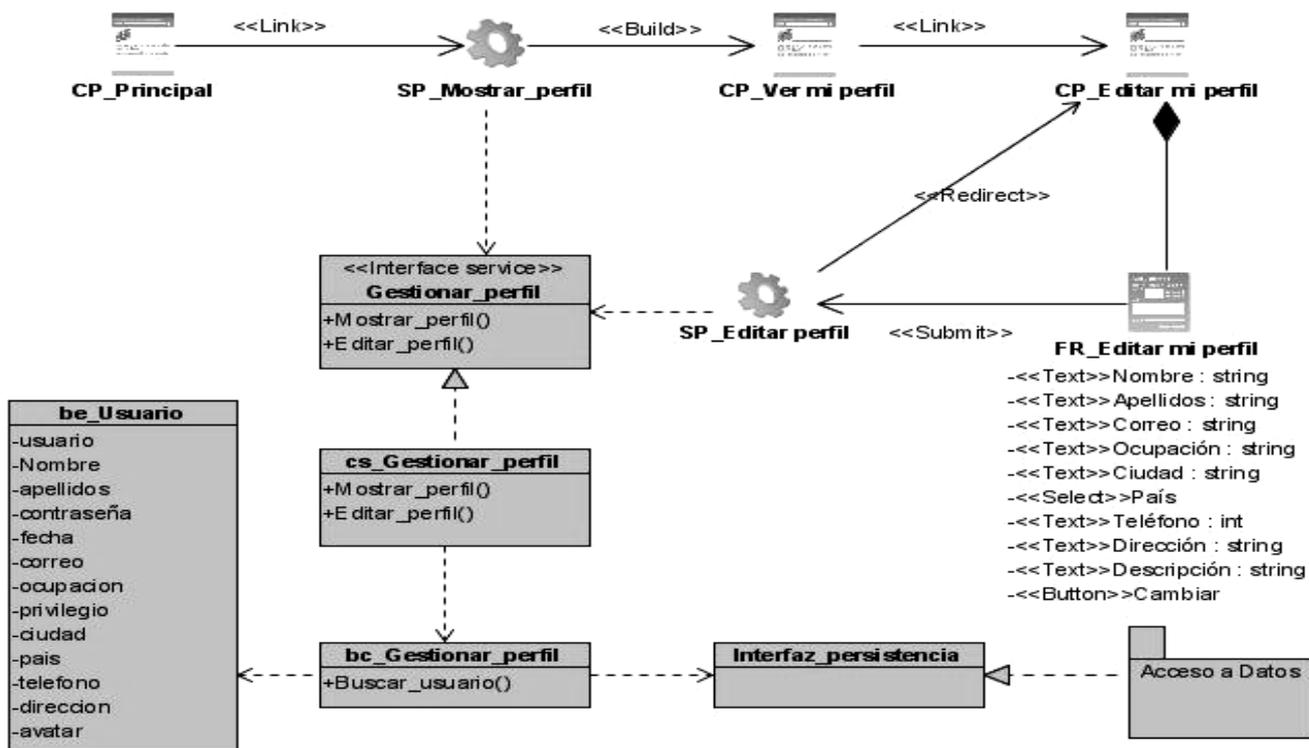


Diagrama de clases del diseño <CU Configurar\_tipo\_autenticación>

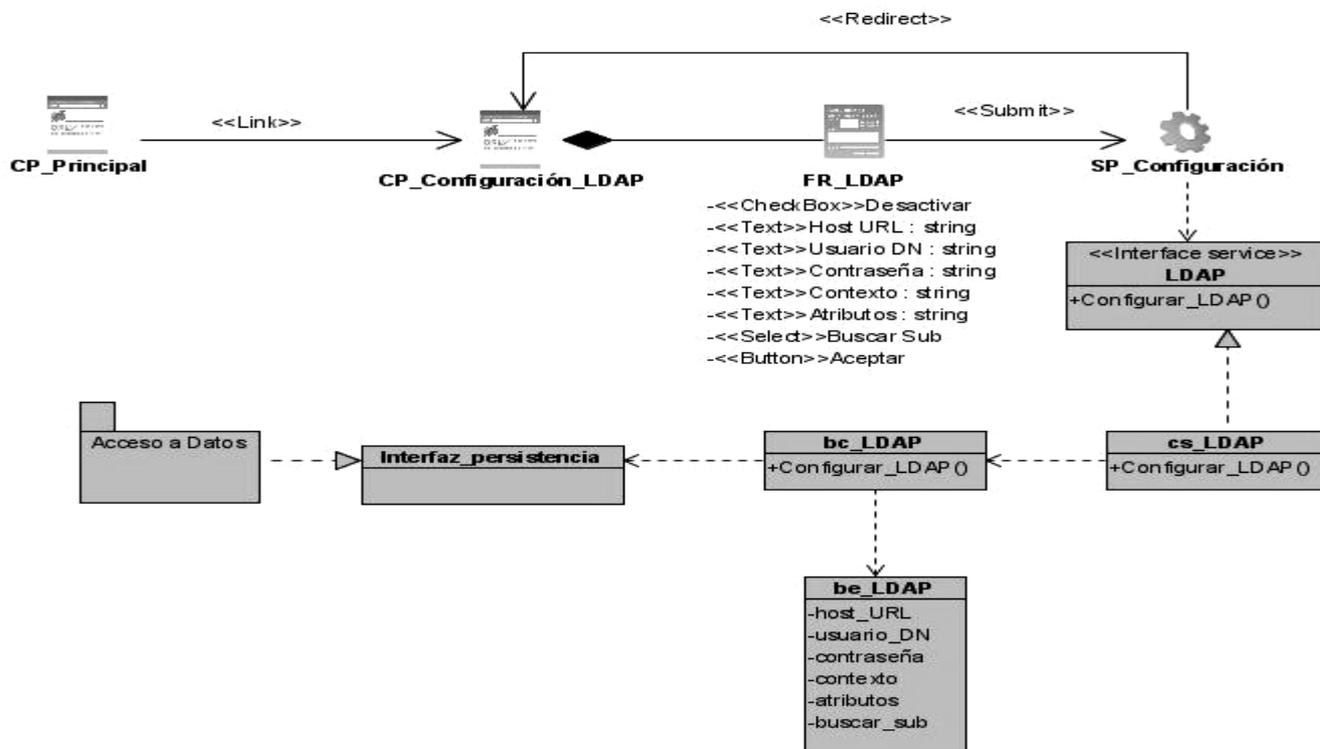


Diagrama de clases del diseño <CU Configurar\_HTTPs>

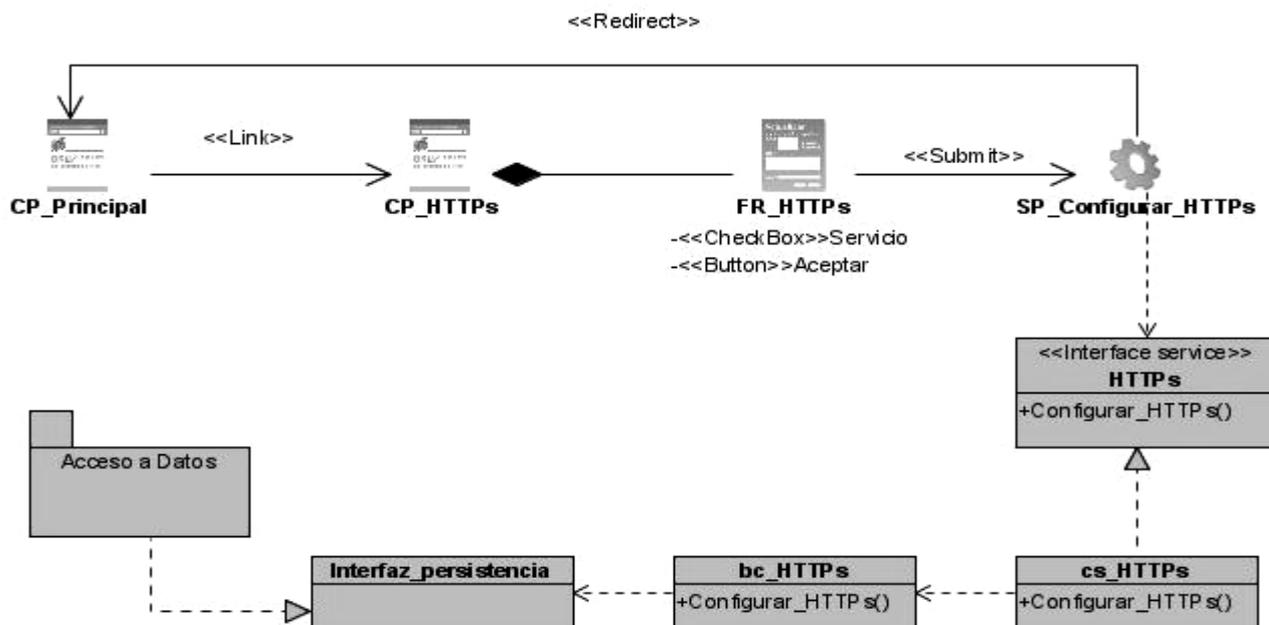


Diagrama de clases del diseño <CU Gestionar\_rol>

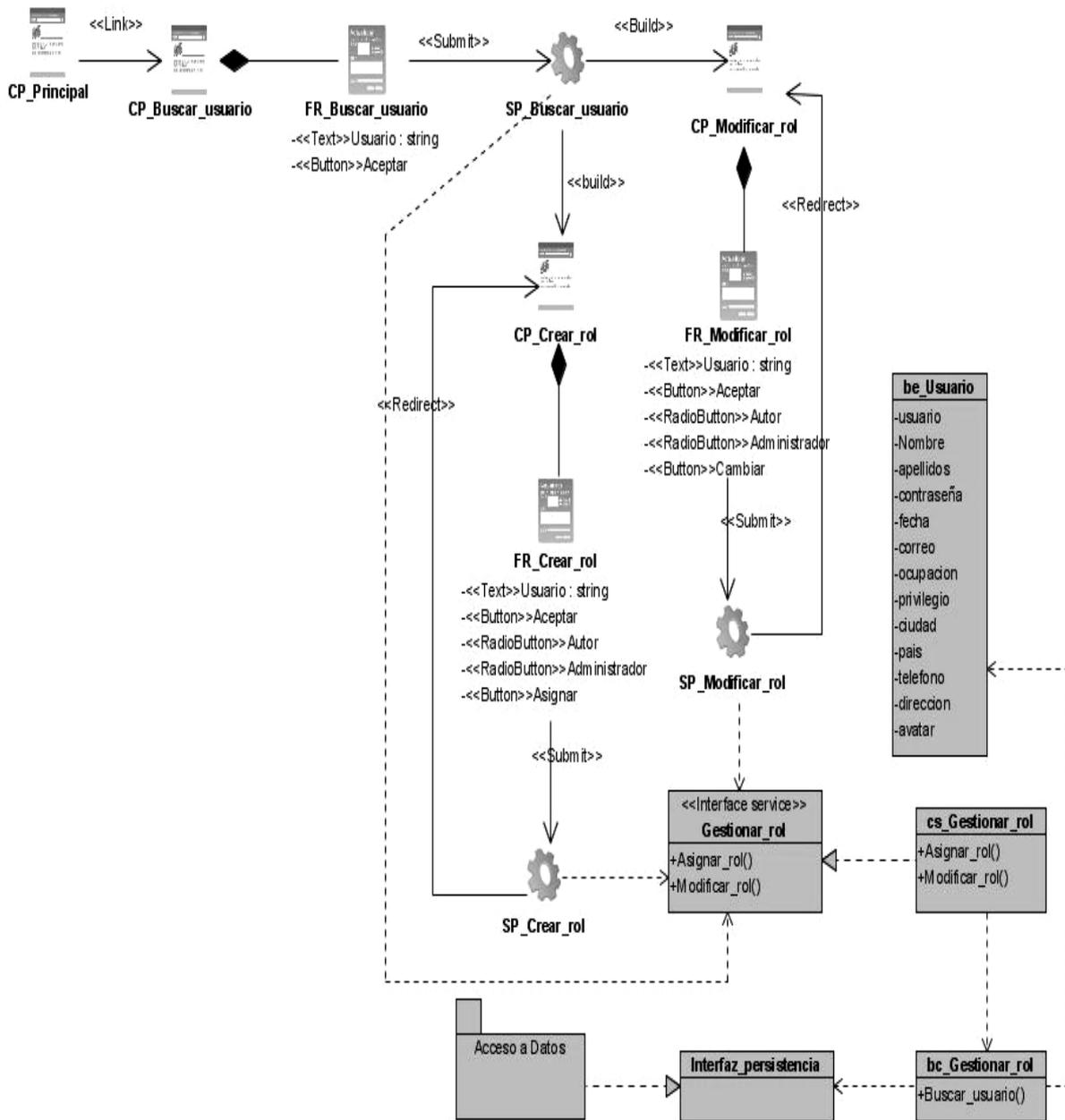


Diagrama de clases del diseño <CU Mostrar vista previa del paquete>

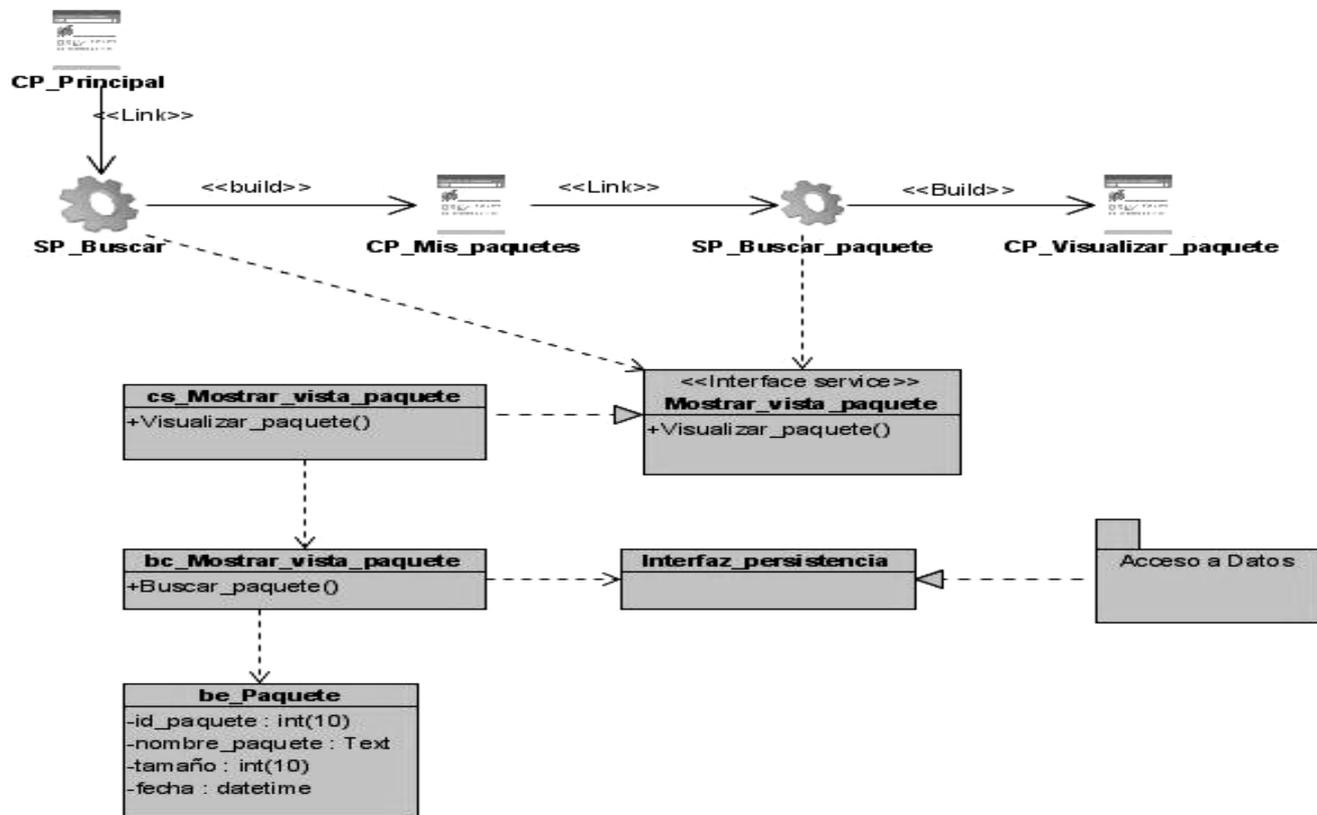


Diagrama de clases del diseño <CU Parametrizar variables de configuración>

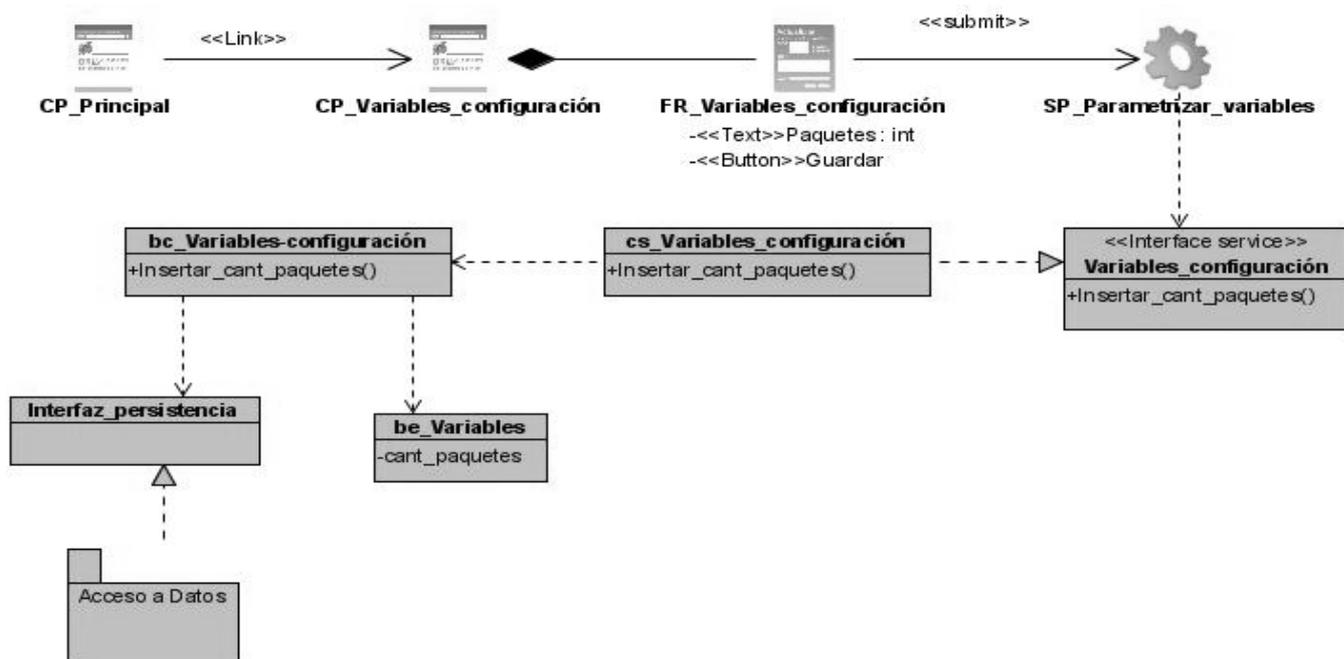


Diagrama de clases del diseño <CU Gestionar\_usuario>

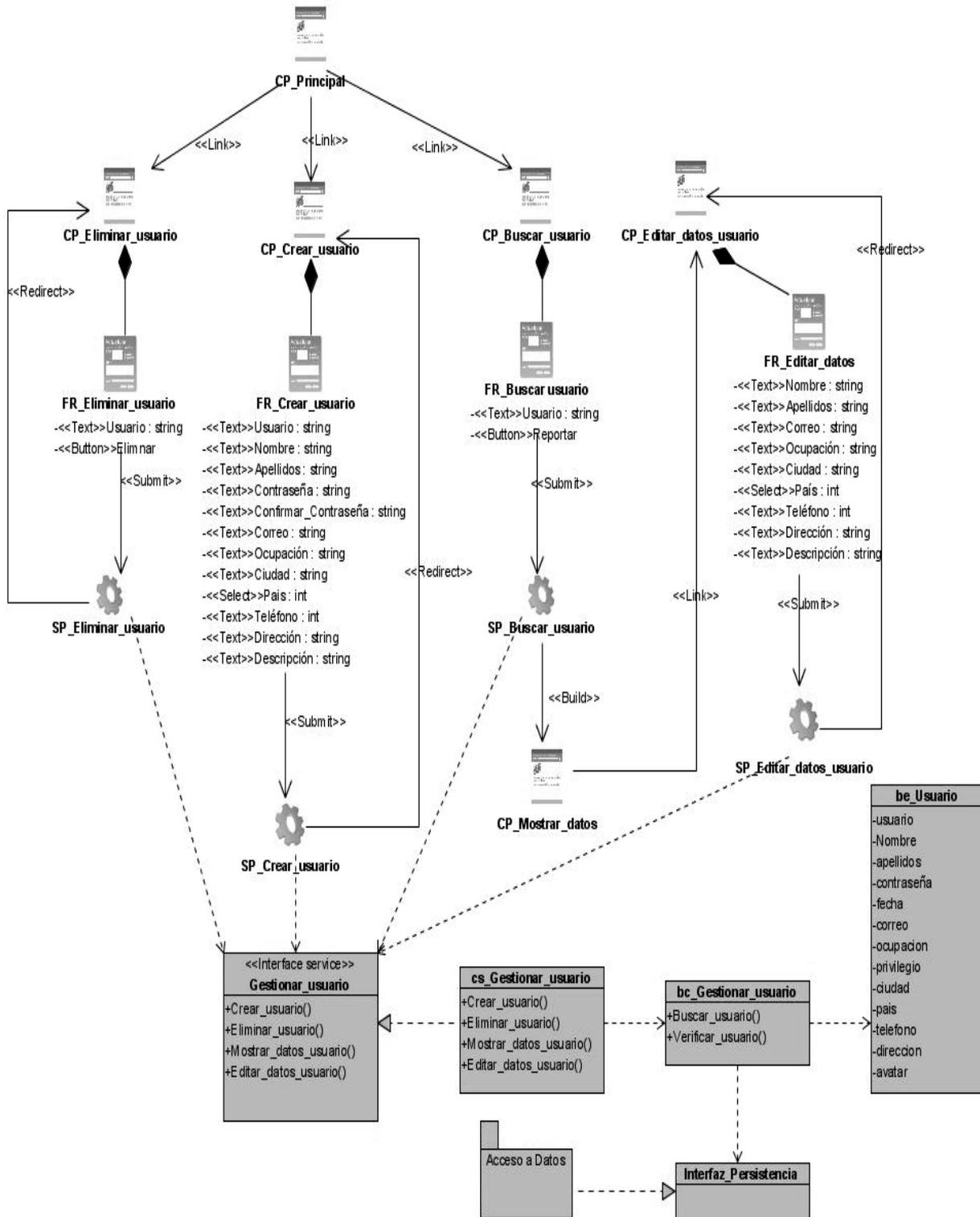


Diagrama de clases del diseño <CU Mostrar\_reportes>

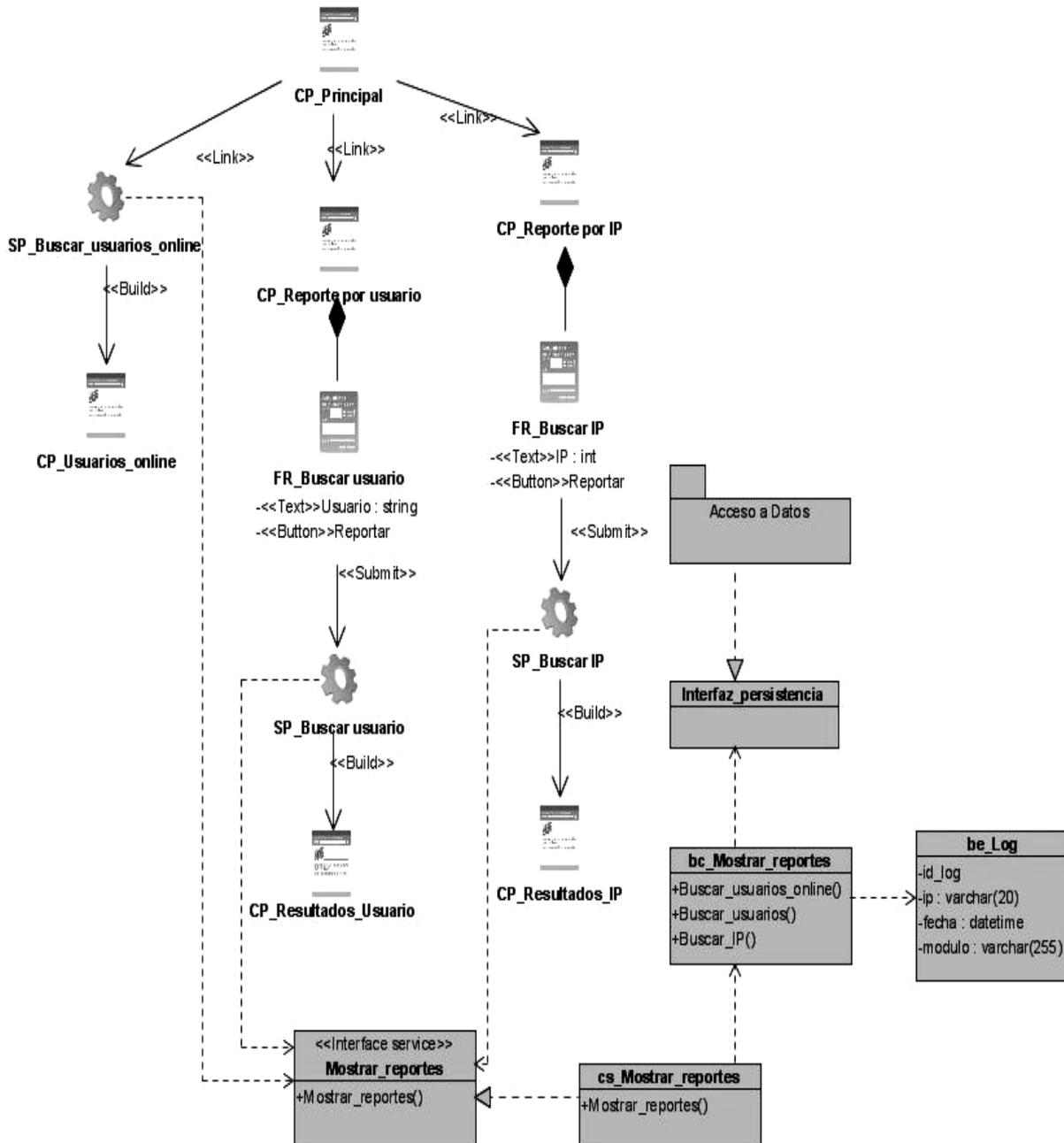


Diagrama de clases del diseño <CU Gestionar\_mensajes>

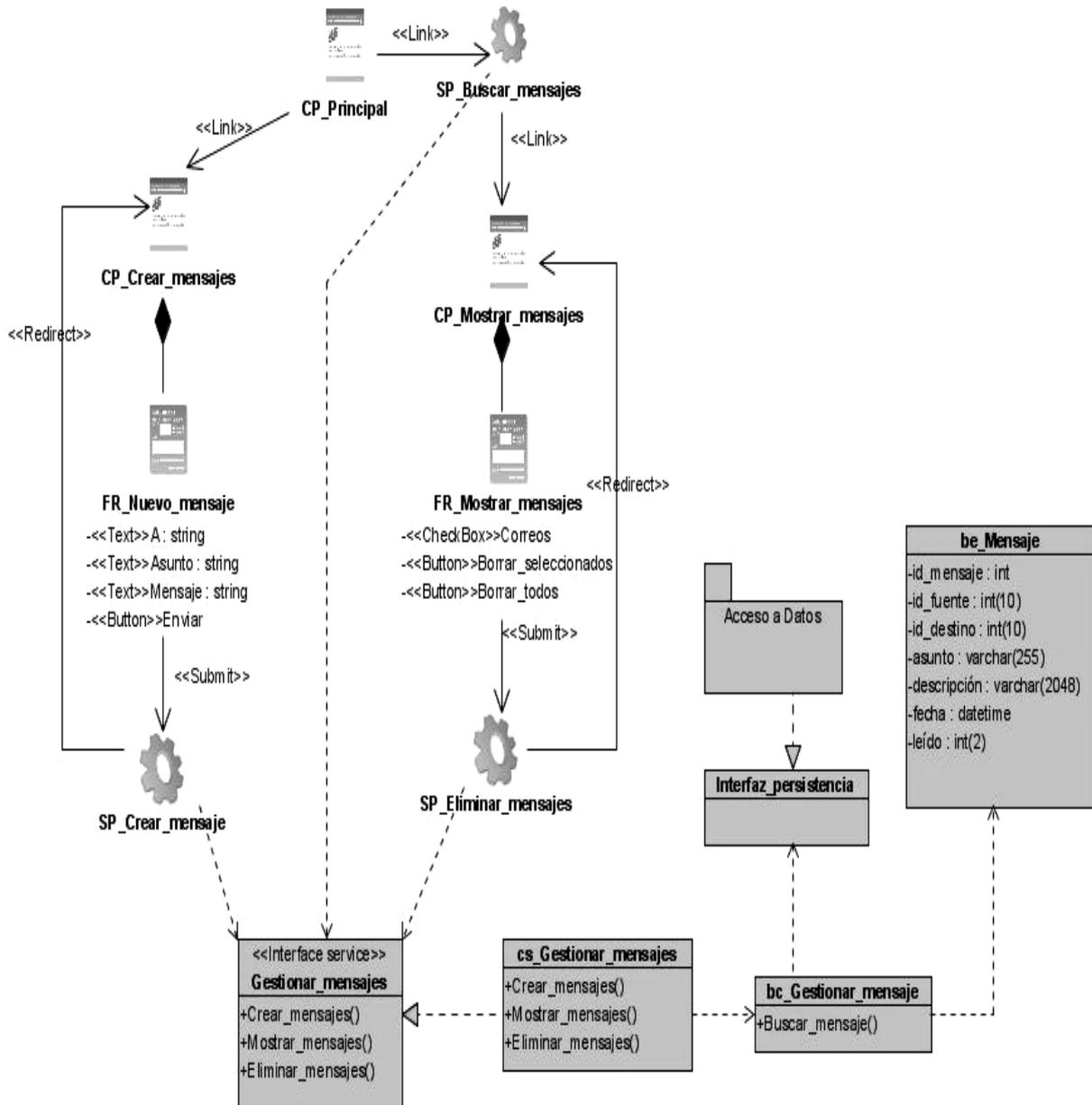


Diagrama de clases del diseño <CU Administrar\_interfaz\_gráfica>

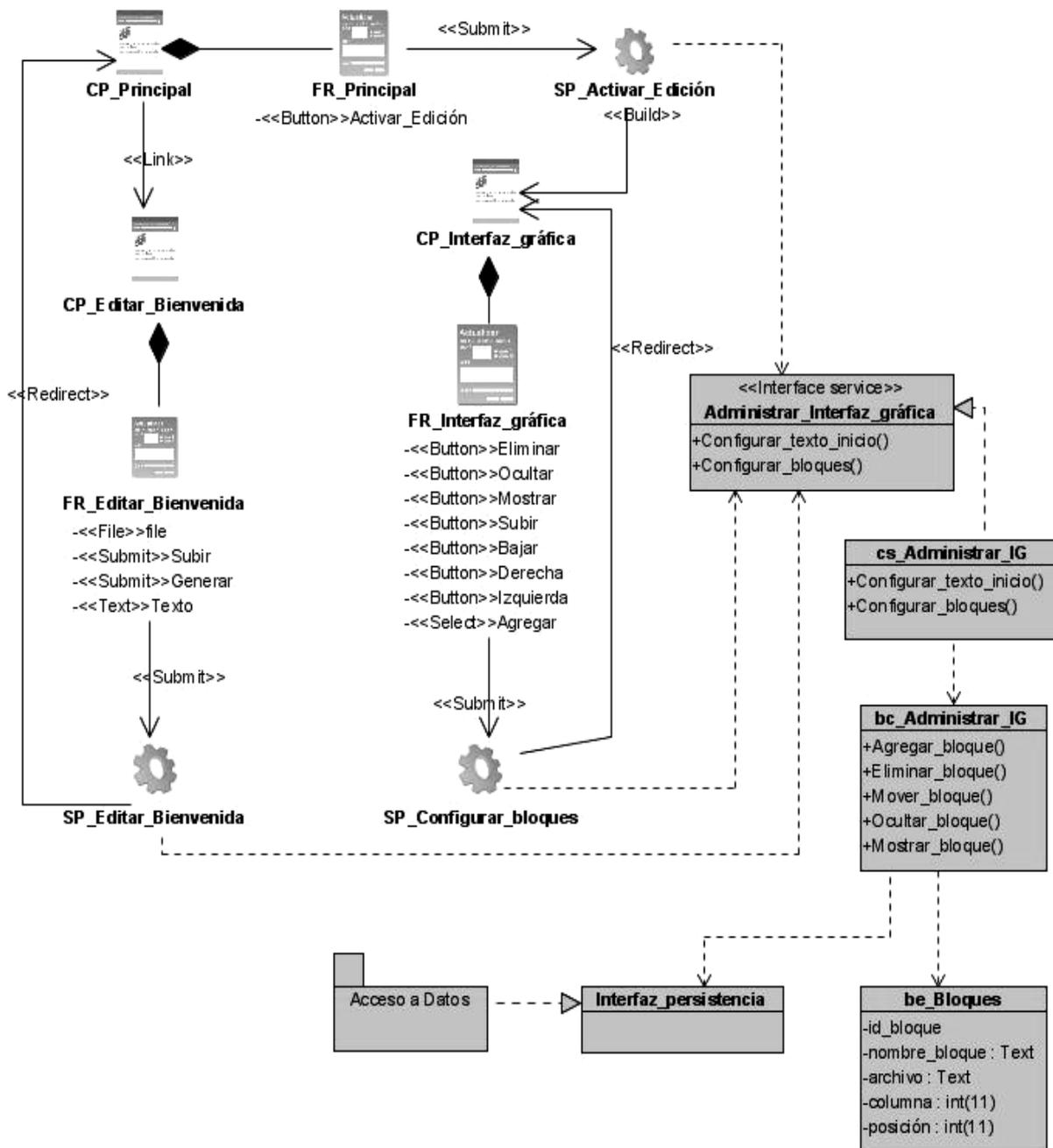


Diagrama de clases del diseño <CU Generar\_paquete>

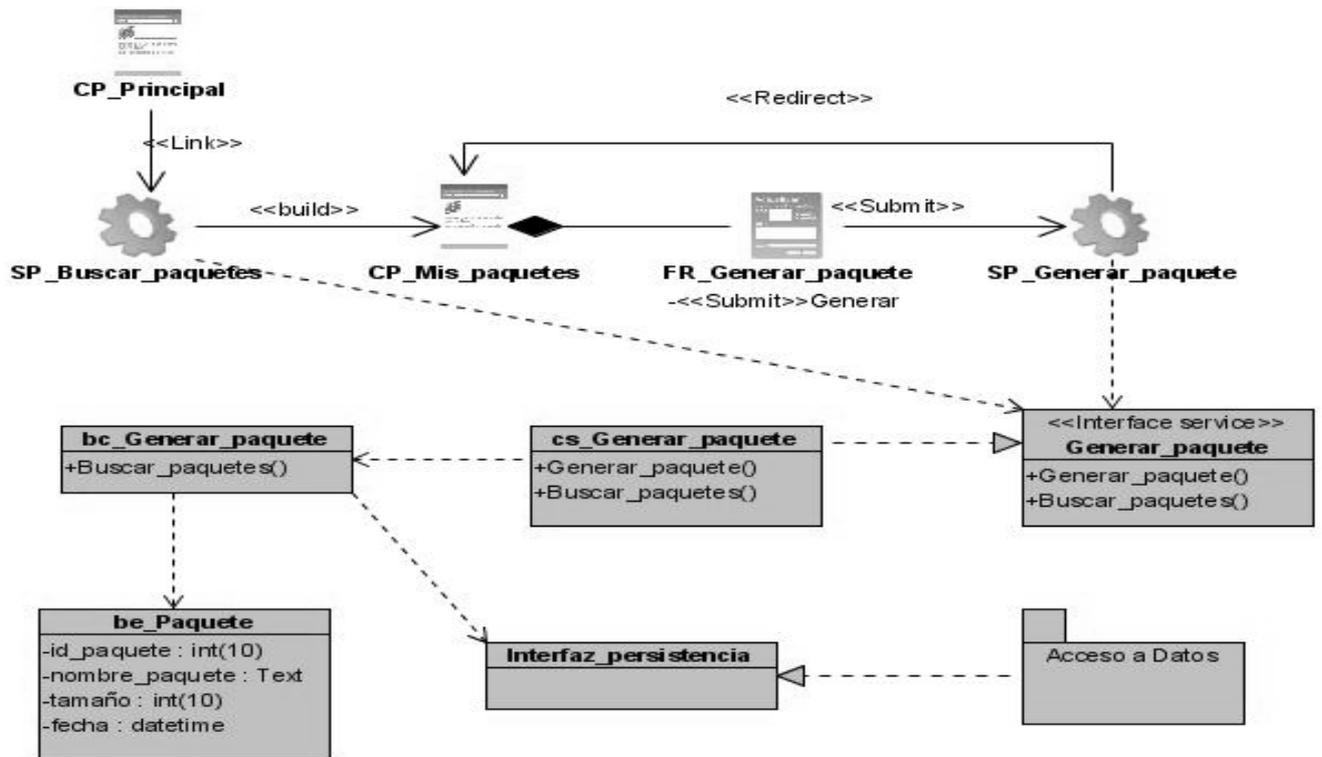


Diagrama de clases del diseño <CU Descargar\_paquete>

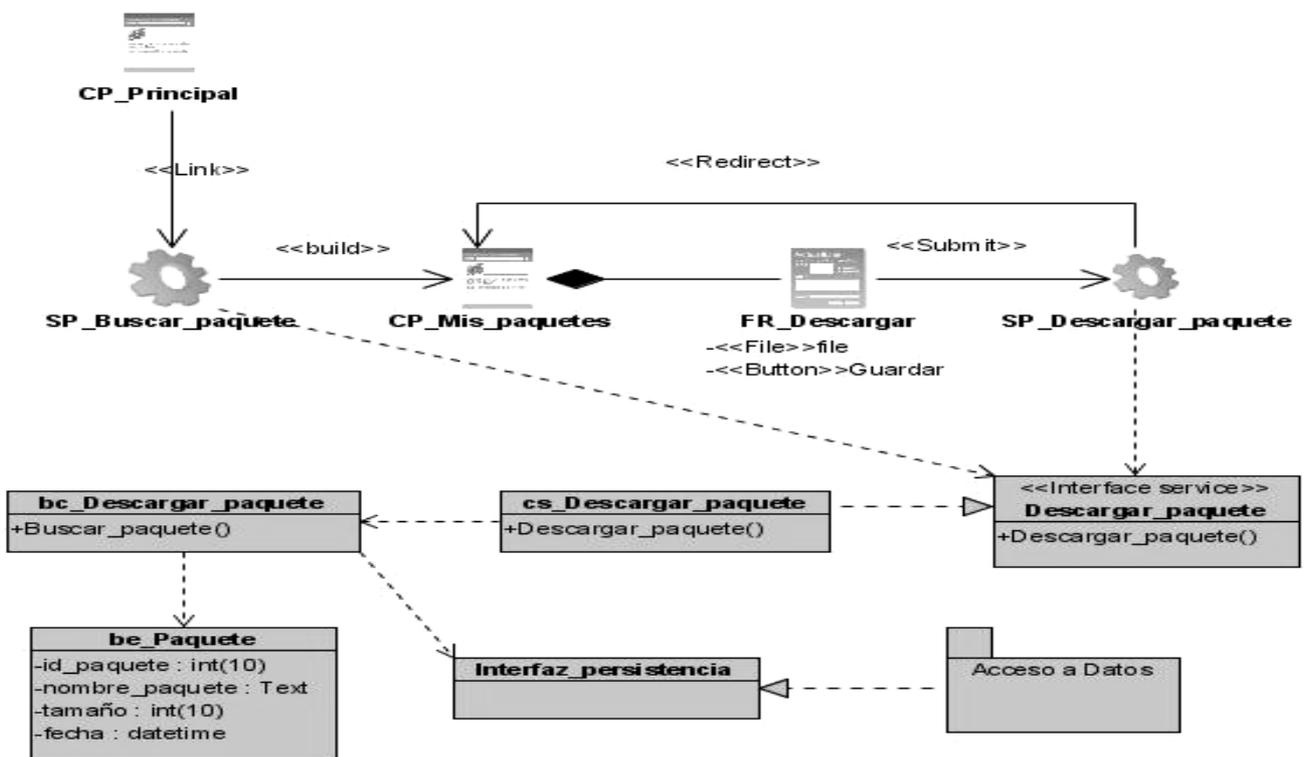


Diagrama de clases del diseño <CU Gestionar\_paquete>

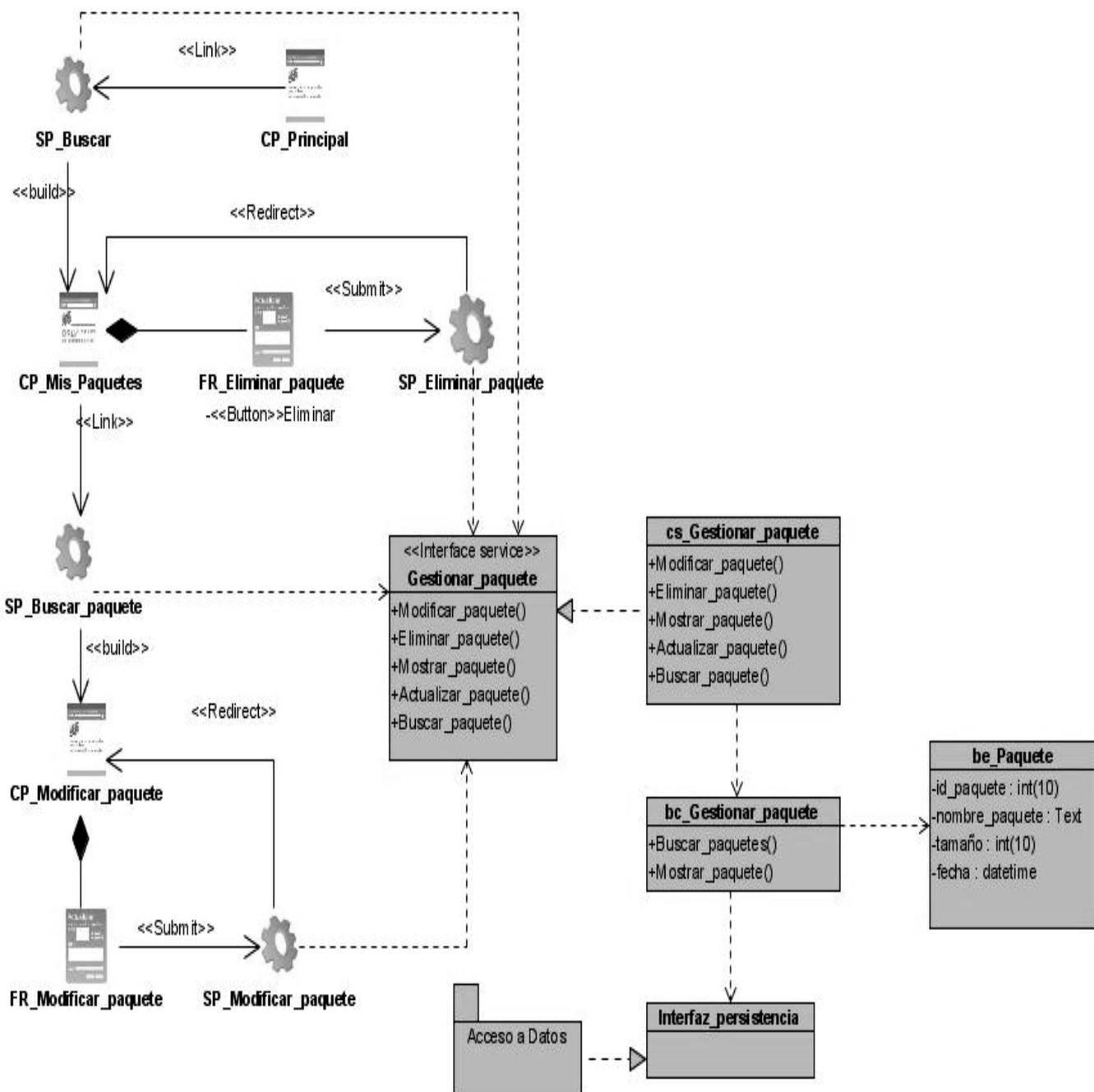


Diagrama de clases del diseño <CU Configurar\_metadatos>

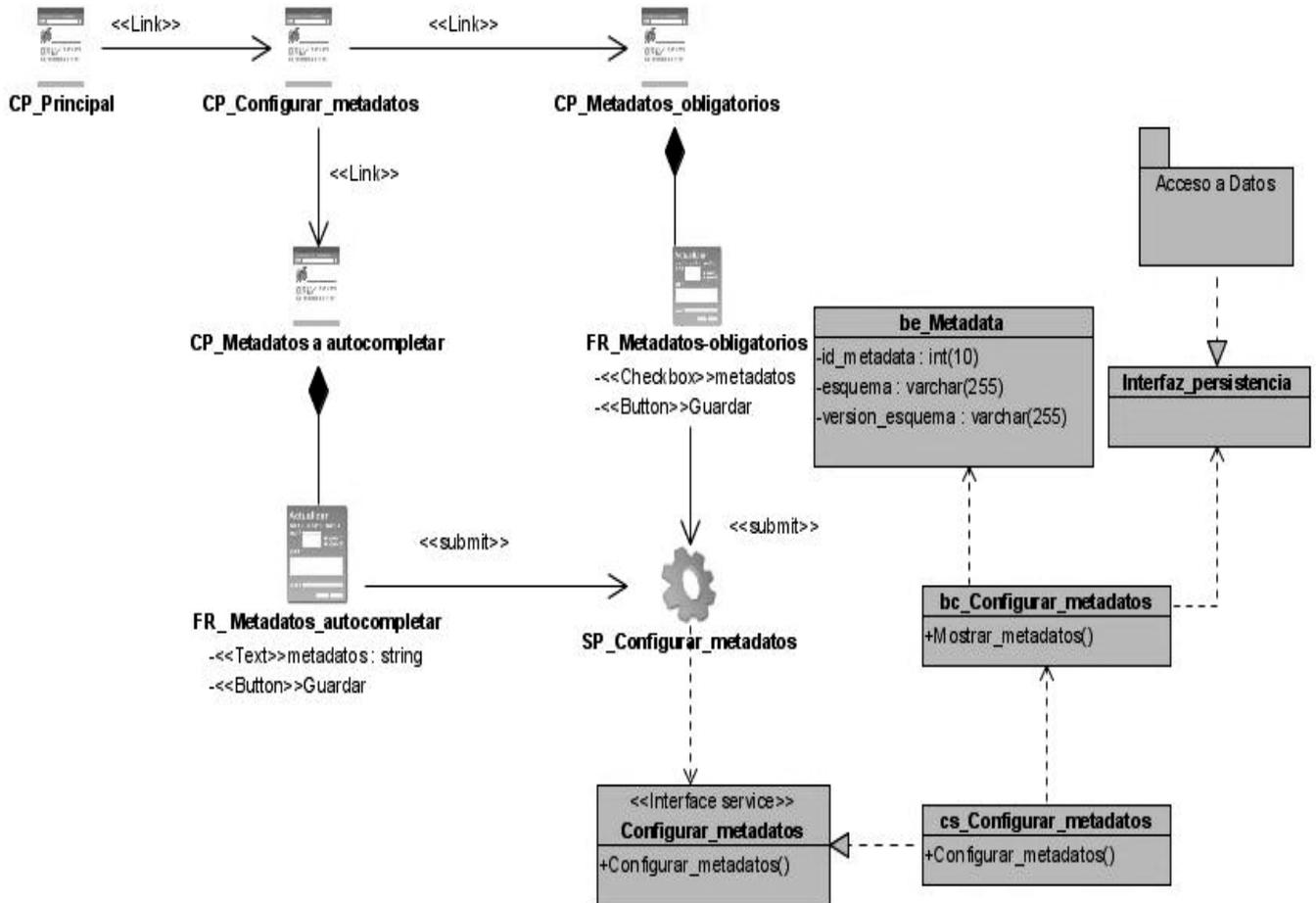


Diagrama de clases del diseño <CU Crear\_paquete>

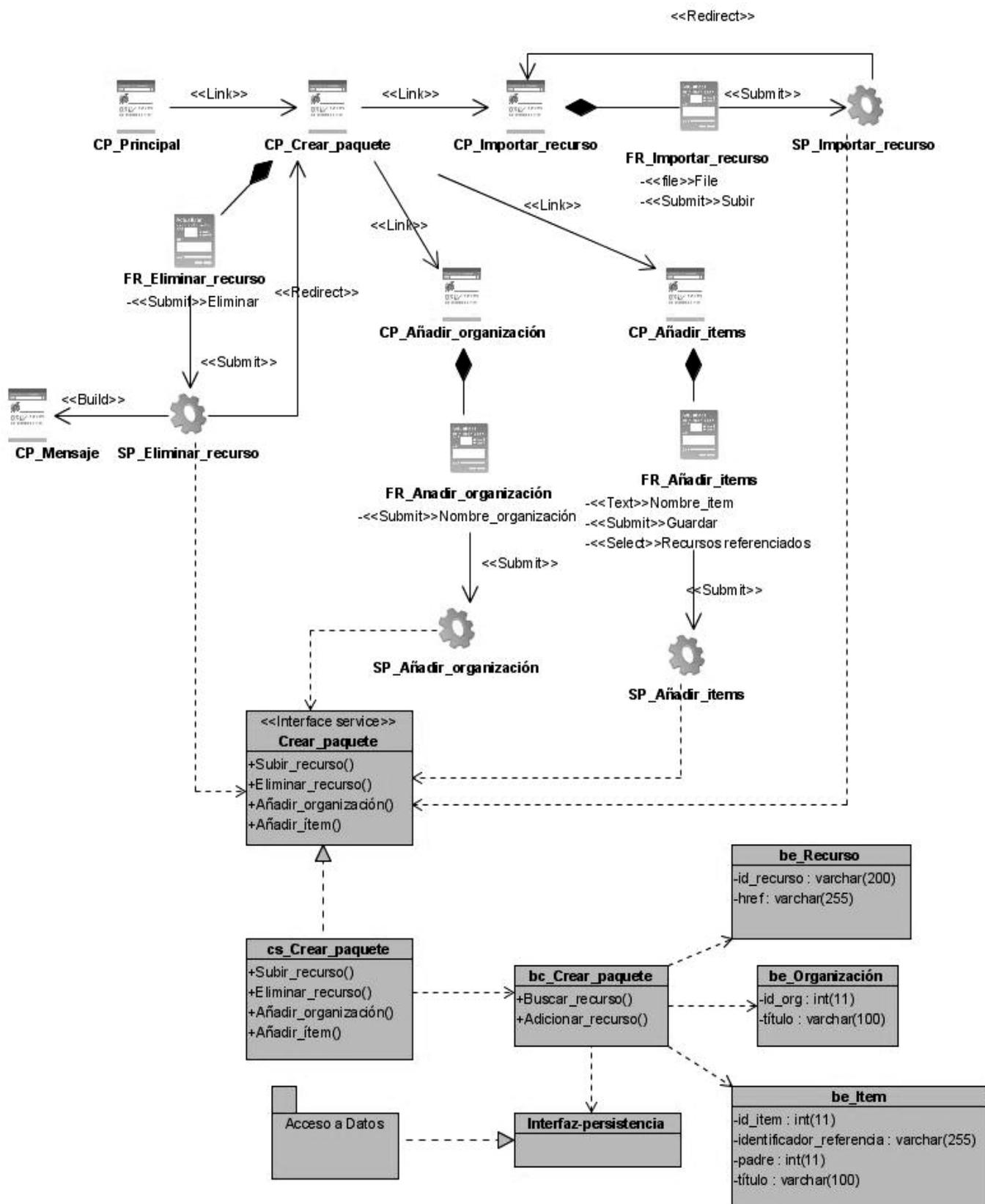


Diagrama de clases del diseño <CU Crear\_ contenido>

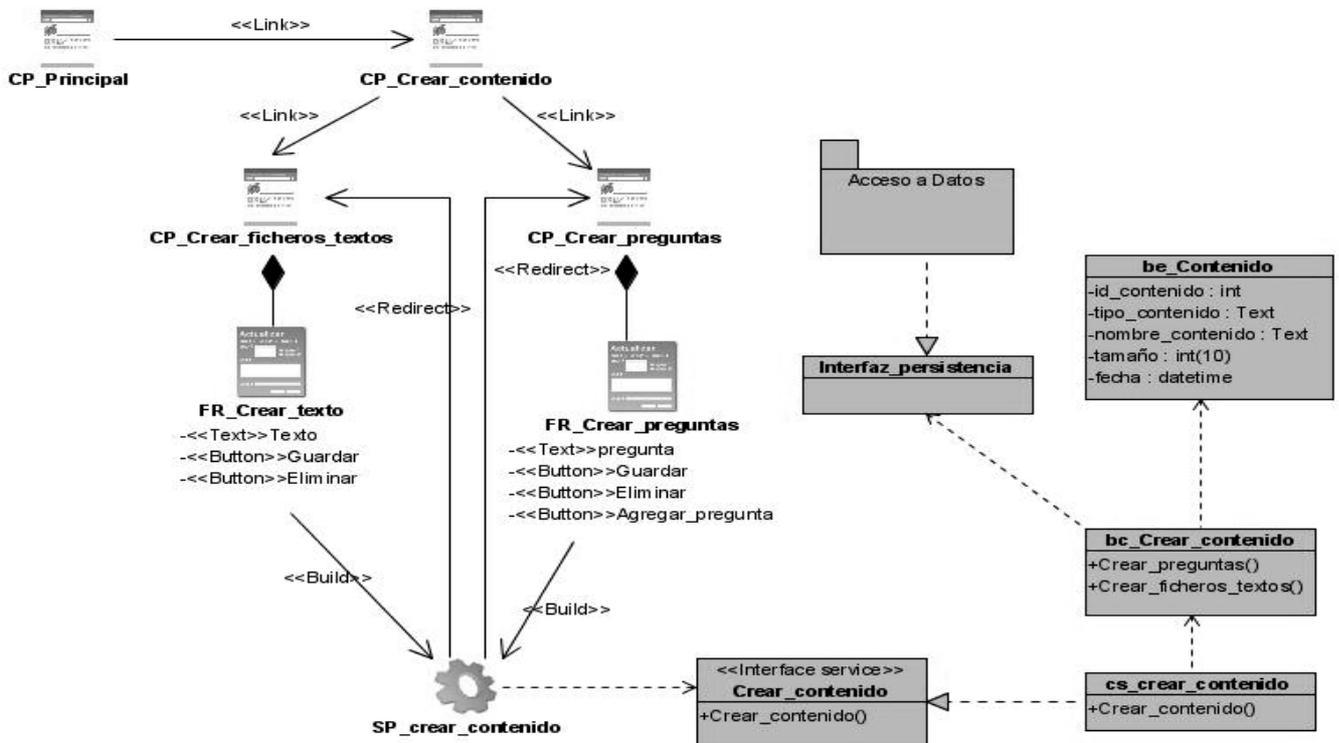


Diagrama de clases del diseño <CU Recuperar\_ contenido>

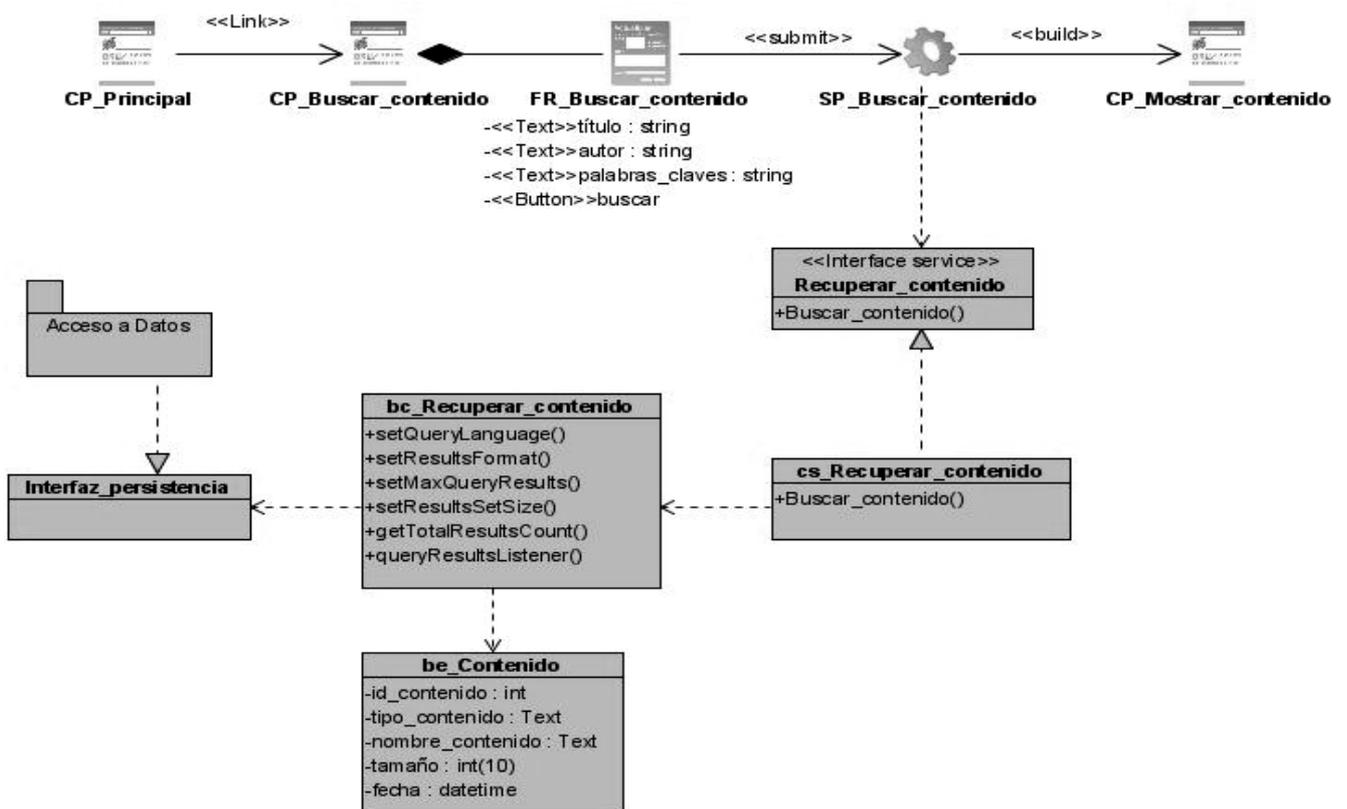


Diagrama de clases del diseño <CU Gestionar\_metadatos>

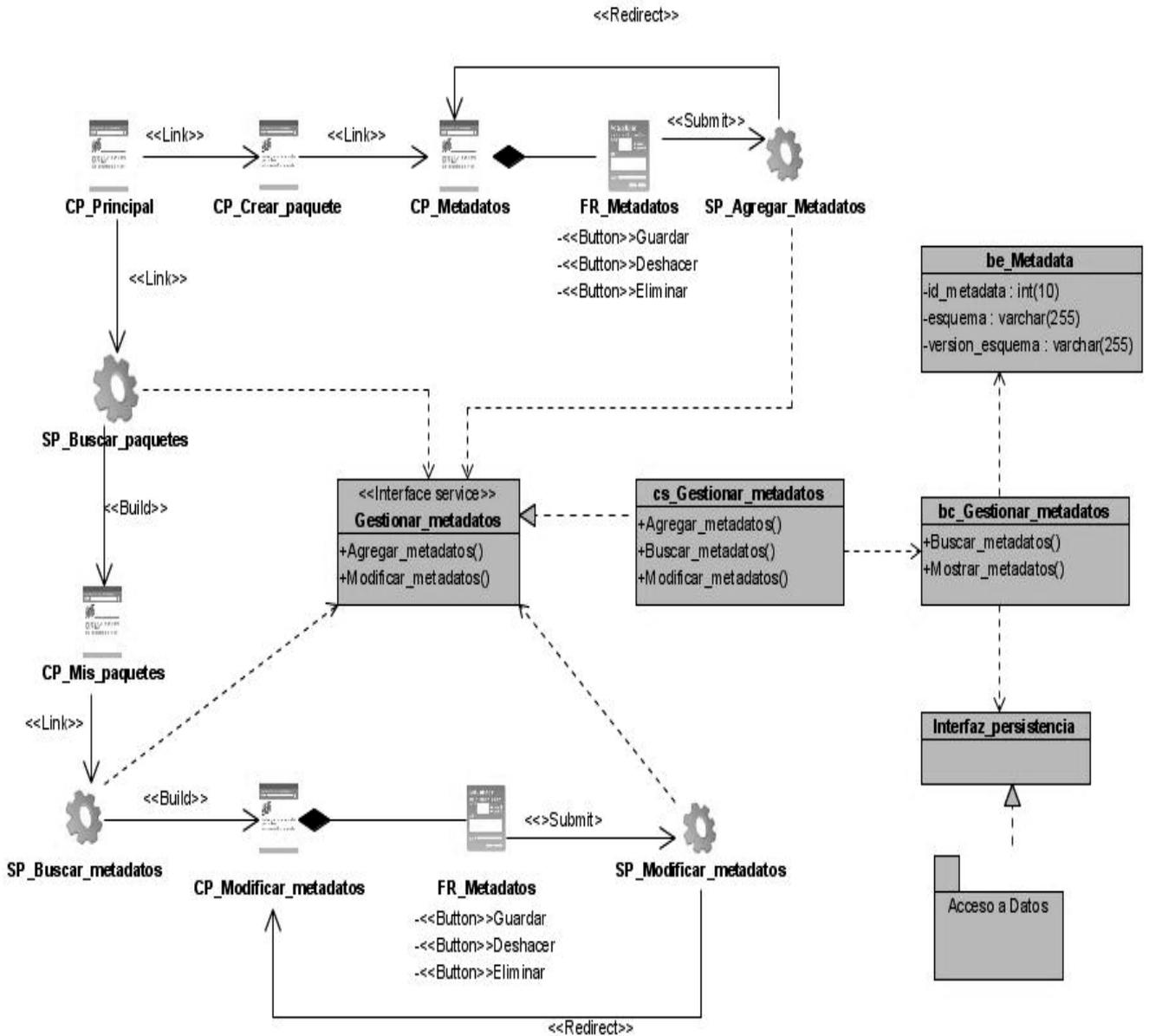


Diagrama de clases del diseño <CU Exportar\_paquete>

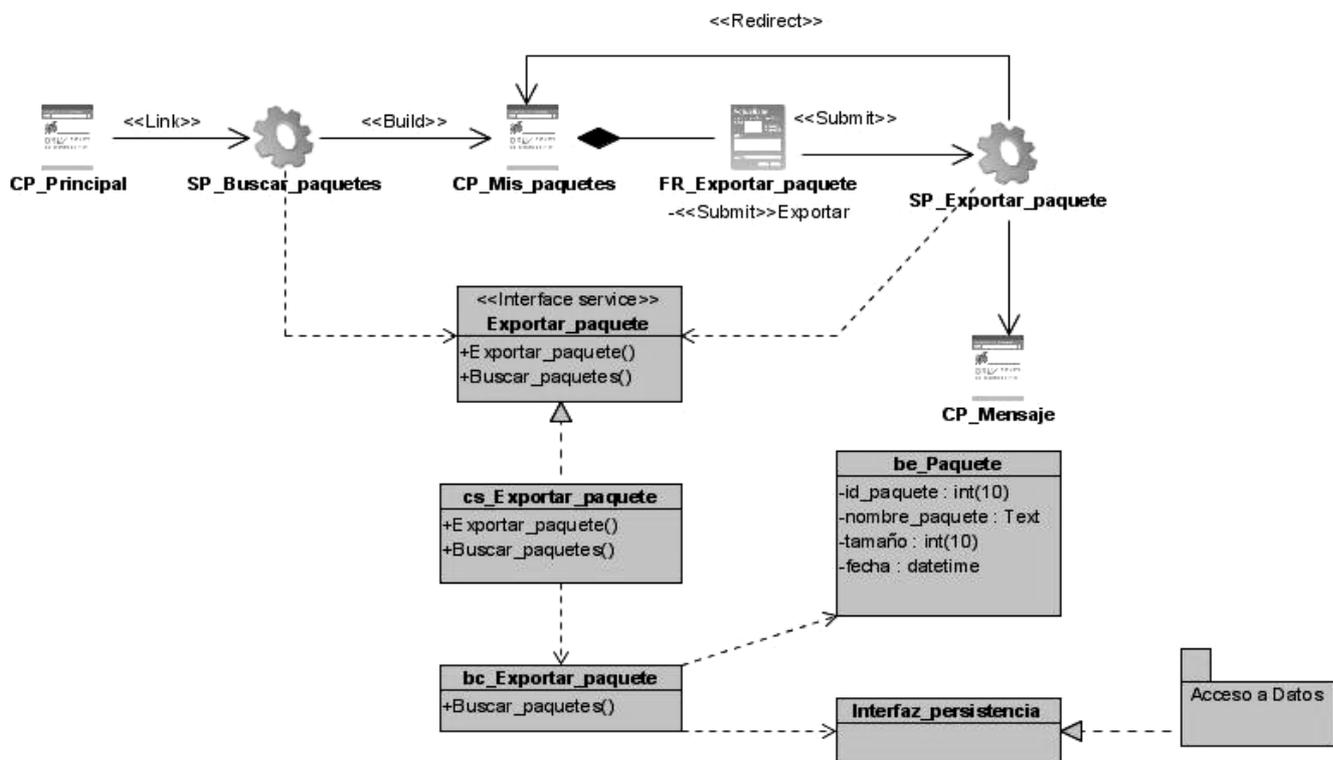
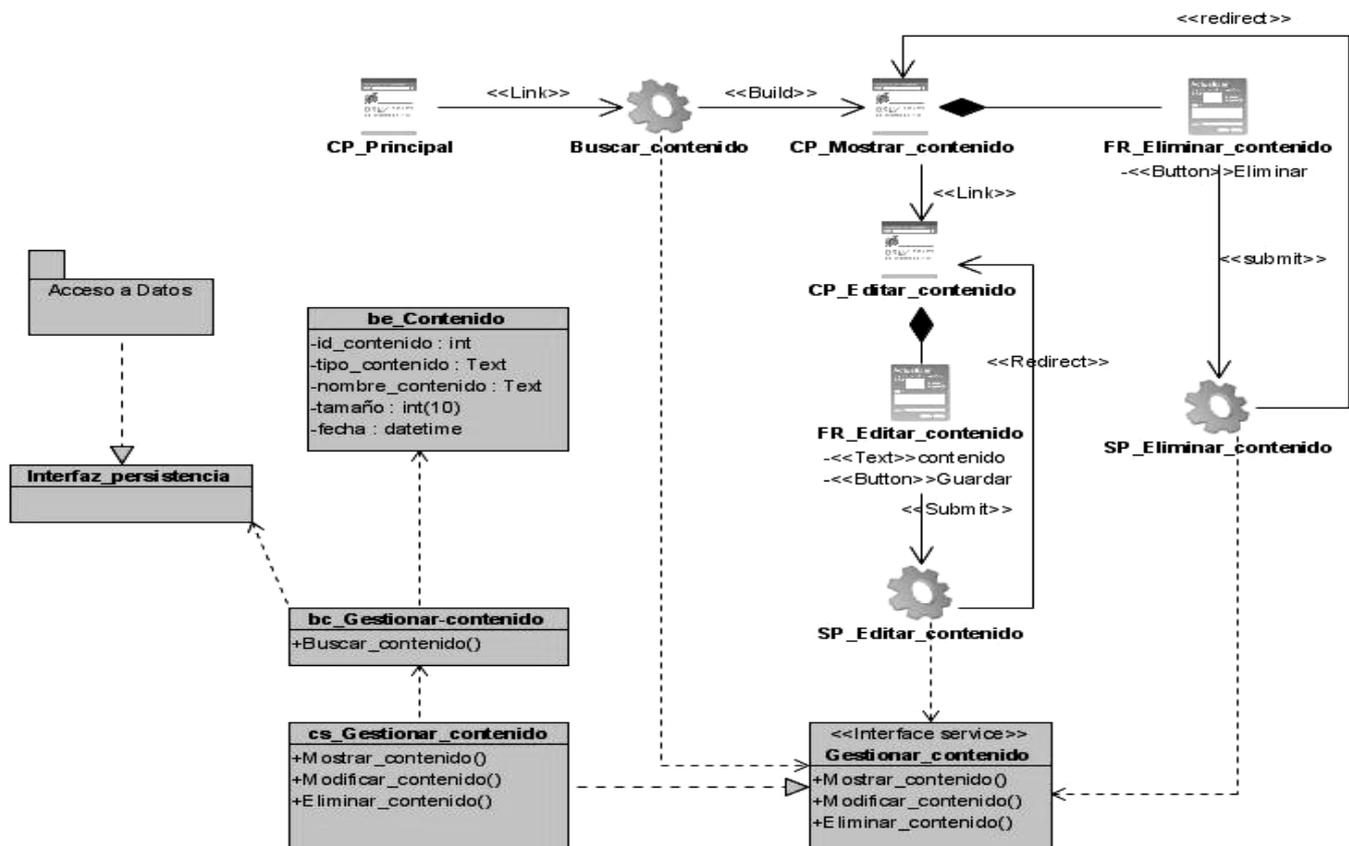


Diagrama de clases del diseño <CU Gestionar\_contenido>



### 3.4 Diseño de la Base de Datos

A partir de las clases del diseño se realiza el diseño de la Base de Datos con el propósito de asegurar que los datos persistentes sean almacenados consistente y eficientemente, además de definir el comportamiento que debe ser implementado en la base de datos, como resultado de esto surge el artefacto Modelo de Datos, que describe la representación lógica y física de los datos persistentes.

#### 3.4.1 Modelo de Datos

##### Modelo de Datos (Parte I)

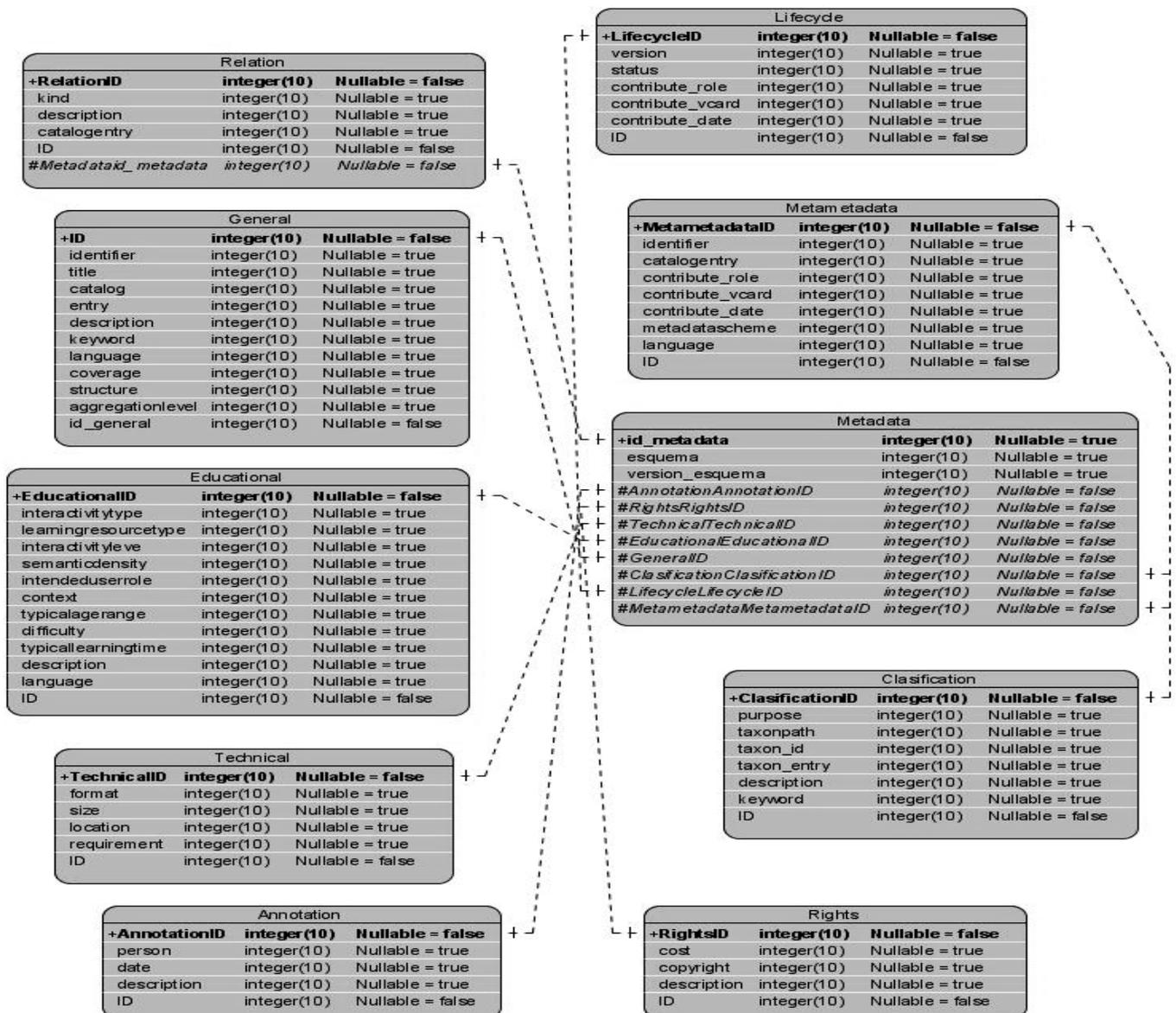


Figura 3. Modelo de datos (Entidad-Relación)

Modelo de Datos (Parte II)

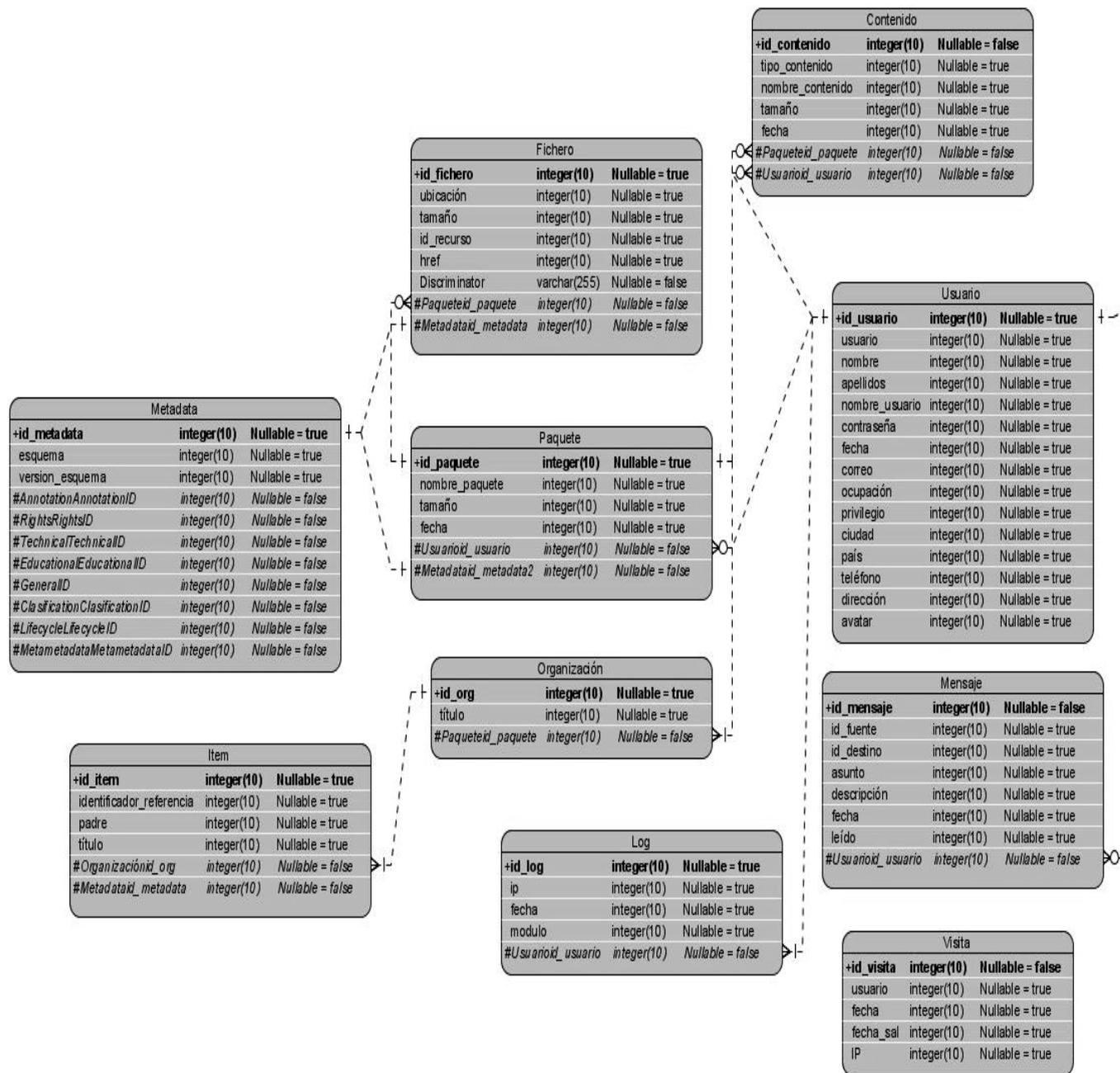


Figura 4. Modelo de datos (Entidad-Relación)

3.4.2 Descripción de las tablas de la Base de Datos

<b>Nombre: Fichero</b>		
<b>Descripción:</b> Esta tabla contiene la información de los ficheros del paquete.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
ID _ fichero	Integer	Identificador de la tabla

Ubicación	varchar	Ubicación del fichero.
tamaño	varchar	Tamaño del fichero.
<b>Nombre: Ítem</b>		
<b>Descripción:</b> Esta tabla contiene la información de los ítem del paquete		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
ID_ ítem	Integer	Identificador de la tabla.
Identificador_ referencia	Varchar	Identificador del recurso asociado el ítem
padre	Integer	Ítem padre.
título	Varchar	Título que se le da al ítem.

<b>Nombre: Log</b>		
<b>Descripción:</b> Esta tabla contiene el control de los reportes para la aplicación.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
ID_ Log	Integer	Identificador de la tabla.
IP	varchar	IP de la máquina donde está sentado el usuario que accede a la aplicación
fecha	datetime	Fecha de acceso a la aplicación.
módulo	varchar	Módulo al que el usuario accedió.

<b>Nombre: Metadata</b>		
<b>Descripción:</b> Esta tabla contiene algunos metadatos de LOM.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id metadata	Integer	Identificador
esquema	varchar	Un elemento opcional cuyo contenido proporciona una descripción textual del esquema que norma la estructura gramatical de la descripción de los metadatos.
Versión _ esquema	varchar	Elemento opcional que describe textualmente la versión del esquema utilizado.

<b>Nombre: Organización</b>		
<b>Descripción:</b> Esta tabla contiene datos sobre la organización del paquete.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id org	Integer	Identificador de la tabla
título	varchar	Título de la organización.

<b>Nombre: Paquete</b>		
<b>Descripción:</b> Esta tabla contiene la información del paquete.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_ paquete	Integer	Identificador de la tabla
nombre	text	Nombre del paquete
tamaño	Integer	Tamaño del paquete
fecha	datetime	Fecha de la creación del paquete

<b>Nombre: Recurso</b>		
<b>Descripción:</b> Esta tabla contiene datos sobre los recursos de un paquete.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Identificador	Integer	Identificador de la tabla
Id_ recurso	Varchar	Identificador del recurso
href	Varchar	La referencia del recurso.

<b>Nombre: Usuario</b>		
<b>Descripción:</b> Esta tabla contiene la información de todos los usuarios registrados en la aplicación.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_ usuario	Integer	Identificador de la tabla

Nombre	varchar	Nombre Real del usuario.
Apellidos	varchar	Apellidos del usuario.
usuario	varchar	Nombre del usuario (nick).
contraseña	varchar	Contraseña del usuario.
fecha	datetime	Fecha y hora de creado el usuario.
correo	varchar	Correo del usuario registrado.
ocupación	varchar	Ocupación del usuario.
privilegio	tinyint	Rol que le es asignado al usuario (admin, autor).
ciudad	varchar	Ciudad donde vive el usuario.
país	varchar	País donde reside el usuario.
teléfono	varchar	Teléfono del usuario.
dirección	varchar	Dirección particular del usuario
avatar	varchar	Imagen definida por el usuario que lo identifica

<b>Nombre: Visita</b>		
<b>Descripción:</b> Esta tabla contiene datos sobre la entrada y salida de un usuario a la aplicación.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
Id_visita	Integer	Identificador de la visita
usuario	Integer	Usuario que entra a la aplicación.
fecha	datetime	Fecha de entrada.
Fecha_sal	datetime	Fecha de salida.
IP	varchar	IP con el cual entro a la aplicación.

<b>Nombre: Mensaje</b>		
<b>Descripción:</b> Esta tabla contiene la información de cada mensaje		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
ID_mensaje	Integer	Identificador del mensaje
ID_destino	Integer	Identificador del usuario destino.
ID_fuente	Integer	Identificador del usuario fuente.
asunto	varchar	Asunto del mensaje.
descripción	varchar	Cuerpo del mensaje.
fecha	datetime	Fecha de envío del mensaje.
leído	tinyint	Estado del mensaje, leído o no.

<b>Nombre: Contenido</b>		
<b>Descripción:</b> Esta tabla contiene la información que describe los contenidos creados.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
ID_contenido	Integer	Identificador del contenido
tipo_contenido	Text	Descripción del tipo de contenido que se trata.
nombre_contenido	Text	Nombre del contenido.
fecha	datetime	Fecha de creación del contenido.
tamaño	Integer	Tamaño del contenido.

<b>Nombre: Bloque</b>		
<b>Descripción:</b> Esta tabla contiene la información que describe cada bloque.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
ID_bloque	Integer	Identificador del bloque
nombre_bloque	Text	Nombre del bloque.
archivo	Text	Ubicación del script que implementa la funcionalidad del bloque.
columna	Integer	Número de la columna en que se encuentra el bloque.
posición	Integer	Posición en que se encuentra el bloque dentro de la columna.

### **3.5 Patrones de diseño**

Con el objetivo de lograr un sistema robusto y reutilizable el diseño se realizó utilizando patrones de diseño, ya que estos son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software. Los patrones de diseño utilizados son:

**Domain Model:** Es el diseño orientado a objetos de cada componente del sistema, además permite el aislamiento tanto con sus capas superiores como sus inferiores. Provee un modelado transparente de la presentación y la persistencia.

**Service Layer:** Utiliza los objetos del modelo del dominio para resolver los casos de uso de la aplicación. Se utiliza para definir, con una capa de servicios, un límite de la aplicación que expone un conjunto de operaciones. Una Service Layer encapsula toda la lógica de negocio, control de transacciones y coordinación de la respuesta en la implementación de las operaciones que expone.

**DataTransferObject (DTO):** Estos objetos DTO cumplen el papel de contenedores de datos y se utiliza con el objetivo de facilitar el transporte de información entre las capas de presentación/servicio.

### **3.6 Conclusiones**

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

## **Conclusiones**

Una vez culminada la investigación se puede afirmar que se cumplieron los objetivos propuestos, arribando a las siguientes conclusiones:

- El análisis y diseño propuesto permite la implementación de una Herramienta de Autor Web capaz de crear Objetos de Aprendizaje.
- El uso de los estándares IMS-DRI y SQI le confieren interoperabilidad a la Herramienta de Autor propuesta.
- La utilización de la Herramienta de Autor Web propuesta y su integración con otras aplicaciones potenciará las facilidades para la gestión de contenidos en un sistema e-learning.

## **Recomendaciones**

Para dar continuidad a la presente investigación se recomienda:

- Implementar la Herramienta de Autor propuesta.
- Extender la herramienta con facilidades para la creación de contenidos.
- Incorporar a la herramienta el estándar para el empaquetamiento SCORM 2004.
- Extender la herramienta con facilidades para la exportación o importación de contenidos en formatos soportados por otras herramientas de autor de amplio uso.
- Utilizar el estándar OKI para ampliar los mecanismos de interoperabilidad.

## Bibliografía

### Referencias Bibliográficas

- 1- AGUIRRE, S. Mediadores e Interoperabilidad en Elearning 2004.
- 2- ALVEZ, P. Estado del arte del proyecto Batuta, 2006.
- 3- ASENSIO, R. M.-B. Metodología de desarrollo de software, 2005.
- 4- ATICA\_SOCRATES. E-Learning, 2007.
- 5- AULAGLOBAL. Estándares - SCORM, 2005. [Disponible en:  
<http://www.aulaglobal.net.ve/observatorio/index.php?lng=es>
- 6- BRAVO, P. Web Service, 2007.
- 7- CASTILLO, A. B. Construcción de Servicios Web con SOAP, 2002, 2.
- 8- CONTRERAS, M. V. M. SISTEMATIZACIÓN DE LA HISTORIA MÉDICA DEPORTIVA PARA EL NIÑO Y EL ADOLESCENTE LÍNEA DE INVESTIGACIÓN EN INFORMÁTICA MÉDICA. FACULTAD DE INGENIERÍA DE SISTEMAS. BOGOTÁ D.C., EL BOSQUE 2006. p.
- 9- COSS, A. C. J. Herramienta de autor para la creación y gestión de objetos de aprendizaje reutilizables. Ciudad de la Habana, Universidad de las Ciencias Informáticas, 2006. 92. p.
- 10- CUDI, D. O. D. A. EMPAQUETAMIENTO DE CONTENIDOS SCORM, 2006. p.
- 11- CUEVA, J. eXeLearning 1.02 2007. [Disponible en: <http://e-profes.net/blog/exelearning-102/>
- 12- ENCINA, E. E. ARQUITECTURA ORIENTADA A SERVICIOS (SOA) EN LA JUNTA DE ANDALUCÍA, 2006.
- 13- FOIX, C. Estándares e-learning, 2002. 1.
- 14- GIL, J. Q. Tutorial para crear paquetes SCORM y usarlos en Moodle, 2005.
- 15- GONZÁLEZ, L. A. Á. EMPAQUETAMIENTO DE OBJETOS DE APRENDIZAJE BAJO EL ESTANDAR SCORM., 2005. 12.
- 16- GUTIÉRREZ, S. M. Tendencias en la Nueva Educación Electrónica, 2006.
- 17- KASSANKE. Estándares relacionados con Objetos de Aprendizajes, 2001.
- 18-LAMARCA, M. J. Metadatos Dublin Core, 2005.
- 19- LEÓN, E. Visual Paradigm, una herramienta de lo más útil., 2007. [Disponible en:  
<http://slion2000.blogspot.com/2007/04/visual-paradigm-una-herramienta-de-lo.html>
- 20- LÓPEZ, C. Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning. España, Salamanca, 2005. p.
- 21- MANJÓN, B. F. Especificaciones y estándares en e-learning Revista de Tecnologías de la Información y Comunicación Educativas, 2006, Nº 6: 2.
- 22- MAURER, W. Estándares elearning, 2004.
- 23- PAULE, M. D. P. Herramienta para la creación de contenidos de aprendizaje, 2006: 6.

- 24- PEDRUELO, M. R. El estándar SCORM para EaD, 2004.
- 25- RICO, J. A. Documento del estado del arte en SOA y Cálculo Pi, 2007.
- 26- SANCHEZ, M. A. M. Metodologías De Desarrollo De Software, 2004.
- 27- TAMAYO, D. Herramientas para la reutilización de contenidos educativos, Universidad de las Ciencias Informáticas, 2007. p.
- 28- URIARTE, J. R. AUTORE: herramienta de autor para la generación de Objetos de Aprendizaje, 2005.
- 29- VARAS, L. REPOSITORIOS DE OBJETOS DE APRENDIZAJE 2002.
- 30- VEGA, J. A. M. Localización, identificación y descripción de recursos web: tentativas hacia la normalización, 2001.
- 31- VELÁZQUEZ, C. La Determinación de la Calidad del Contenido de un Objeto de Aprendizaje, 2005.
- 32- ZAPATA, H. Desarrollo Basado en Funcionalidades, 2006.

## **Bibliografía consultada**

- 1- ALICANTE, U. D. Web Services, 2006.
- 2- AULAGLOBAL. Estándares - SCORM, 2005. [Disponible en:  
<http://www.aulaglobal.net.ve/observatorio/index.php?lng=es>
- 3- BARCO, A. SOA y los Servicios Web., 2006.
- 4- BONET, J. C. A. Metadatos y documentos XML/RDF para recuperación, 2007.
- 5- CASTILLO, A. B. Construcción de Servicios Web con SOAP, 2002, 2.
- 6- COSS, A. C. J. Herramienta de autor para la creación y gestión de objetos de aprendizaje reutilizables. Ciudad de la Habana, Universidad de las Ciencias Informáticas, 2006. 92. p.
- 7- CUDI, D. O. D. A. EMPAQUETAMIENTO DE CONTENIDOS SCORM, 2006. p.
- 8- CUEVA, J. eXeLearning 1.02 2007. [Disponible en: <http://e-profes.net/blog/exelearning-102/>
- 9- DAVIES, D. WeLOAD RELOAD 2005.
- 10- DÁVILA, O. Creación de SCORM, 2007: 21.
- 11- ENCINA, E. E. ARQUITECTURA ORIENTADA A SERVICIOS (SOA) EN LA JUNTA DE ANDALUCÍA, 2006.
- 12- Estudio comparativo internacional sobre modelos estandarizados... Disponible en:  
<http://ares.cnice.mec.es/informes/11/contenido/25.htm> g
- 13- GIL, J. Q. Tutorial para crear paquetes SCORM y usarlos en Moodle, 2005.
- 14- GONZÁLEZ, L. A. Á. EMPAQUETAMIENTO DE OBJETOS DE APRENDIZAJE BAJO EL ESTANDAR SCORM., 2005. 12.
- 15- GUTIÉRREZ, S. M. Tendencias en la Nueva Educación Electrónica, 2006.
- 16- JACOBSON, I. El proceso Unificado de Desarrollo de Software. 2004. 438 p.
- 17- KASSANKE. Estándares relacionados con Objetos de Aprendizajes, 2001.
- 18- LAMARCA, M. J. Metadatos Dublin Core, 2005.
- 19- LÓPEZ, C. Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning. España, Salamanca, 2005. p.
- 20- MAURER, W. Estándares elearning, 2004.
- 21- PAULE, M. D. P. Herramienta para la creación de contenidos de aprendizaje, 2006: 6.
- 22- PEDRUELO, M. R. El estándar SCORM para EaD, 2004.
- 23- PRESSMAN, R. Ingeniería de Software. 2002. p.
- 24- Protocolo Z39.50. Recuperación estructurada de la información distribuida.
- 25- RICO, J. A. Documento del estado del arte en SOA y Cálculo Pi, 2007.
- 26- SANCHEZ, M. A. M. Metodologías De Desarrollo De Software, 2004.

- 27- TAMAYO, D. Herramientas para la reutilización de contenidos educativos, Universidad de las Ciencias Informáticas, 2007. p.
- 28- TUNIS, E. H. Las Herramientas de Autor en el proceso de producción de materiales educativos en formato digital.
- 29- URIARTE, J. R. AUTORE: herramienta de autor para la generación de Objetos de Aprendizaje, 2005.
- 30- VARAS, L. REPOSITORIOS DE OBJETOS DE APRENDIZAJE 2002.
- 31- VEGA, J. A. M. Localización, identificación y descripción de recursos web: tentativas hacia la normalización, 2001.
- 32- ZAPATA, H. Desarrollo Basado en Funcionalidades, 2006.

## **Glosario de Términos:**

**LDAP:** Protocolo a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

**WYSIWYG:** What You See Is What You Get (en español: “lo que ves es lo que obtienes”). Para describir un editor de texto que permite escribir un documento viendo directamente el resultado final, frecuentemente el resultado impreso.

**UML:** Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modelling Language) es el lenguaje de modelado de sistemas de software más conocido en la actualidad.

**SCORM** (Sharable Content Object Reference Model): Modelo de referencia para el desarrollo e integración de contenidos educativos.

**FTP** (File Transfer Protocol): Es un protocolo de transferencia de archivos entre sistemas conectados a una red TCP basado en la arquitectura cliente-servidor.

**Programación Orientada a Objetos:** Es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y programas de computadora.

**XML:** Un metalenguaje extensible de etiquetas desarrollado.

**IMS:** Es un consorcio internacional que ha propuesto un conjunto de especificaciones sobre distintos aspectos que intervienen en el modelado del aprendizaje en *e-learning*.

**HTTP** (HyperText Transfer Protocol): El protocolo de transferencia de hipertexto usado en cada transacción de la Web.

**HTTPS** (Protocolo seguro de transferencia de hipertexto): es un protocolo de red basado en el protocolo HTTP, destinado a la transferencia segura de datos de hipertexto.

**SQL:** Lenguaje de consulta estructurado.