

Universidad de las Ciencias Informáticas

Facultad 8



Creación colaborativa de objetos de aprendizaje a partir de una herramienta de autor Web.

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Autores: Dianelys Preval Cobas

Jorge Iturria Pozo

Tutores: Ing. Dunia María Colomé Cedeño

Msc. Daymy Tamayo Avila

Ciudad de La Habana, junio 2010

“Año 52 de la Revolución”

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 2010.

Autores

Dianelys Preval Cobas

Jorge Iturria Pozo

Tutoras

Ing. Dunia Ma. Colomé Cedeño

MSc. Daymy Tamayo Avila

Agradecimientos

De Dianelys:

A mi madre por todo su apoyo, por sus consejos y preocupaciones.

A mis abuelos Reyna, Urbano, Angelia y mi tía Aracelis por preocuparse siempre por mí y apoyarme en todo lo que me hizo falta.

A mi hermanito Iván que aunque lejos supo darme fuerzas y consejos para que todo saliera adelante.

A mi novio César por estar a mi lado y brindarme su amor cuando más lo necesité.

A todos mis compañeros de proyecto por aportar un granito a este trabajo, sin ellos muchas cosas no hubieran sido posibles.

A mis amigas Leanet, Karina y Yuliet por compartir estos años de estudio y por todos los momentos felices y tristes que pasamos juntas.

A Liana que aunque no pudo terminar la carrera junto a mí, siempre ha estado presente para ayudarme en lo que he necesitado.

A todos mis amigos que a lo largo de la carrera he conocido, han sido súper especiales y nunca los olvidare: Ángela, Yisel, Marisol, Loty, Mayrel "Paco", Carlos Alberto, Jorge Luis, Merayo, Miguel Ángel, Yordania, Yilena.....

A mis tutores por servirme de guía.

A mi tutora Dunia en especial porque siempre estuvo presente para brindarme todo su apoyo, ayuda y dedicación al trabajo.

A mi compañero de tesis Jorge "Francés" por su dedicación, por ser tan buen compañero, por soportarme todo este tiempo a su lado.

A todas las personas que a lo largo de mi vida y carrera me han apoyado, gracias por estar ahí cuando los necesité.

Agradecimientos

De Jorge:

Primeramente doy gracias a Dios Padre, Hijo y Espíritu Santo por estos cinco años de carrera universitaria donde bajo su mirada y en sus brazos me ha llevado a recorrer este camino guiándome en todo momento aún sin percatarme de ello muchas veces. Gracias Dios mío porque como Padre me has hecho crecer en muchos aspectos de mi vida. Con Dios todo sin Él nada.

Gracias también a la virgen María bajo la advocación de la Caridad del Cobre porque ella es la madre de todos los cubanos. A ti también mamá las gracias por no dejarme solo nunca.

A mi familia en general gracias por el grano de arena que cada uno en mayor o menor medida han puesto en mi vida, en especial a mis padres por todo su amor incondicional, siempre prestos a darlo todo sin esperar nada a cambio siendo así fundamento de mi vida.

Gracias también a Trujillo quien ha sido otro padre para mí siempre enseñándome parte de su gran conocimiento y sabiduría.

Gracias a todos mis amigos y compañeros que han caminado conmigo juntos este camino.

Agradecimientos

Gracias a mi compañera de tesis por el trabajo que realizó y por siempre acompañarme durante todo su desarrollo.

Gracias a mis tutores por acompañarme igualmente durante todo el transcurso de la tesis.

¡Gracias a todos!

Dedicatoria

Dianelys:

Este trabajo se lo dedico a mi madre Lucia Cobas Vinent por ser mi fuerza y mi guía, por apoyarme durante todo este tiempo, por su amor, dedicación, por ser mi inspiración en cada paso que doy, por brindarme su confianza y por ser la persona que más quiero en el mundo.

Jorge:

Este trabajo se lo dedico a mi hermano Norbertico B. Pérez Pozo, mi prima hermana Elizabeth Oria Pozo, mi primo Angelito y Jorgito. A ustedes que aún se encuentran en sus años de estudio les va dedicado este trabajo para que les sirva de motivación por lograr Uds. también llegar al término de su vida estudiantil no así al término de los estudios y aprendizaje porque eso nunca se termina.

Resumen

Con anterioridad al surgimiento de las Tecnologías de la Información y las Comunicaciones (TICs) solo se conocía una única forma de transmitir los conocimientos de generación a generación y era el conocido modelo discípulos-maestros. Con la incorporación de las TICs a este modelo se evidenciaron muchísimos avances, entre ellos el surgimiento del e-learning, en el cual fue apareciendo de manera gradual la necesidad de compartir recursos y reutilizarlos en cualquier área del conocimiento, surgiendo la tecnología Objetos de Aprendizaje (OA).

Para la creación y edición de los OA, es necesaria la utilización de Herramientas de Autor (HA), estas deben permitir la interoperabilidad con los Repositorios de Objetos de Aprendizaje (ROA) para así posibilitar su almacenamiento y posterior uso.

En la Universidad de la Ciencias Informáticas (UCI) está en desarrollo una HA llamada CRODA que permite la creación de OA, la misma no ofrece la posibilidad de crear estos colaborativamente, ni permite la integración con el ROA denominado RHODA.

En el presente trabajo se desarrolla un módulo para CRODA que posibilite la creación colaborativa de OA y la interoperabilidad con el RHODA. El desarrollo del mismo conllevó a la obtención de los diferentes artefactos que se generan en cada flujo de trabajo de la metodología Rational Unified Process (RUP).

Palabras claves:

Creación colaborativa, Interoperabilidad, Objetos de Aprendizaje, Herramienta de Autor, Repositorio de Objetos de Aprendizaje.

Índice

Introducción	1
Capítulo I: Fundamentación teórica.....	6
1.1 Introducción.....	6
1.2 Objetos de Aprendizaje.	6
1.3 Estándares y especificaciones para la gestión de Objetos de Aprendizaje.....	7
1.3.1. Estándares para empaquetar OA.	8
1.3.2. Estándar para describir OA.	11
1.4 Herramientas de Autor.	12
1.4.1. ReLOAD (Reusable eLearning Object Authoring & delivery).	12
1.4.2. Autore.	13
1.4.3. eXe.....	13
1.4.4. Lectora.	13
1.4.5. CRODA.....	14
1.5 Herramientas para el trabajo colaborativo.	14
1.5.1. Edukalibre.	15
1.5.2. BSCW.	16
1.6 Repositorio de Objetos de Aprendizaje.	18
1.6.1. RHODA.....	18
1.7 Estándares de interoperabilidad.	18
1.8 Metodologías de desarrollo de software.	21
1.8.1. Herramientas CASE.	25
1.9 Servidores Web.....	26
1.10 Gestores de Base de Datos.	27
1.11 Framework de desarrollo.....	29
1.12 Conclusiones.....	31

Capítulo II: Características del sistema.	33
2.1. Introducción.....	33
2.2. Objeto de Automatización.	33
2.3. Propuesta del sistema.	33
2.3.1. Modelo de dominio.	34
2.4. Especificación de requisitos.	36
2.4.1. Requerimientos funcionales.	36
2.4.2. Requerimientos no funcionales.	38
2.5. Definición de Actores y Casos de Uso.....	41
2.5.1. Definición de actores del sistema.	41
2.5.2. Definición de casos de uso.....	41
2.5.3. Diagrama de Casos de Uso del Sistema.	45
2.6. Descripciones textuales de los casos de uso del sistema.....	46
2.7. Conclusiones.....	53
Capítulo III: Análisis y diseño del sistema.	55
3.1 Introducción.....	55
3.2 Análisis.....	55
3.2.1 Diagrama de clases de análisis.	55
3.3 Diseño.....	58
3.3.1 Diagrama de clases del diseño.....	58
Capítulo IV: Construcción de la solución propuesta.	64
4.1 Introducción.....	64
4.2 Diagrama de despliegue.....	64
4.3 Diagrama de componentes.	65
4.4 Conclusiones.....	67
Conclusiones generales.....	68
Recomendaciones	69
Bibliografía.....	70

Anexos.....	74
Glosario de términos.....	115

Índice de figuras

Figura 1 Diagrama del modelo del dominio	35
Figura 2 Diagrama de casos de uso del sistema.....	46
Figura 3 Diagrama de clases de análisis "Crear colaboración".	56
Figura 4 Diagrama de clases de análisis "Proponer a eliminar colaboración"......	56
Figura 5 Diagrama de clases de análisis "Asignar responsable de colaboración".	56
Figura 6 Diagrama de clases de análisis "Eliminar colaboración"......	56
Figura 7 Diagrama de clases de análisis "Visualizar área de colaboración".	57
Figura 8 Diagrama de clases de análisis "Relacionar recurso a la estructura de un OA".	57
Figura 9 Diagrama de clases de análisis "Crear tema de debate".	57
Figura 10 Diagrama de clases de análisis "Adicionar comentarios".	57
Figura 11 Diagrama de clases de diseño "Crear colaboración".	59
Figura 12 Diagrama de clases de diseño "Proponer a eliminar colaboración"......	60
Figura 13 Diagrama de clases de diseño "Asignar responsable de colaboración"......	61
Figura 14 Diagrama de clases de diseño "Eliminar colaboración".	62
Figura 15 Diagrama de clases de diseño "Visualizar área de colaboración".	63
Figura 20 Diagrama de despliegue.	64
Figura 21 Diagrama de componente.	66
Figura 22 Diagrama de clases de análisis "Editar colaboración".	91
Figura 23 Diagrama de clases de análisis "Configurar pizarra virtual"......	91
Figura 24 Diagrama de clases de análisis "Visualizar pizarra virtual"......	91
Figura 25 Diagrama de clases de análisis "Mostrar área de edición del OA".	92
Figura 26 Diagrama de clases de análisis "Adicionar recurso a un comentario"......	92
Figura 27 Diagrama de clases de análisis "Visualizar OA almacenado en RHODA".	92
Figura 28 Diagrama de clases de análisis "Editar OA almacenado en el RHODA".	93
Figura 29 Diagrama de clases de análisis "Gestionar Recurso Web"......	93
Figura 30 Diagrama de clases de análisis "Gestionar ejercicio de orden cronológico".	93
Figura 31 Diagrama de clases de análisis "Gestionar ejercicio de buscar en el texto".	94
Figura 32 Diagrama de clases de análisis "Gestionar ejercicio de enlazar columnas"......	94
Figura 33 Diagrama de clases de análisis "Gestionar cuestionario".	94

Figura 34 Diagrama de clases de diseño "Editar colaboración".	95
Figura 35 Diagrama de clases de diseño "Configurar pizarra virtual".	96
Figura 36 Diagrama de clases de diseño "Visualizar pizarra virtual".	97
Figura 37 Diagrama de clases de diseño "Mostrar área de edición del OA".	98
Figura 38 Diagrama de clases de diseño "Adicionar recurso a un comentario".	99
Figura 39 Diagrama de clases de diseño "Visualizar OA almacenado en el RHODA".	100
Figura 40 Diagrama de clases de diseño "Editar OA almacenado en el RHODA".	101
Figura 41 Diagrama de clases de diseño "Gestionar recurso web".	102
Figura 42 Diagrama de clases de diseño "Gestionar ejercicio de orden cronológico".	103
Figura 43 Diagrama de clases de diseño "Gestionar ejercicios de buscar en el texto".	104
Figura 44 Diagrama de clases de diseño "Gestionar ejercicio de enlazar columnas".	105
Figura 45 Diagrama de clases de diseño "Gestionar cuestionario".	106

Anexos

Anexo 1 Descripción textual CU_ Editar colaboración.....	74
Anexo 2 Descripción textual CU_ Visualizar pizarra virtual (extend).	75
Anexo 3 Descripción textual CU_ Configurar pizarra virtual (extend).	75
Anexo 4 Descripción textual CU_ Mostrar área de edición del OA.	76
Anexo 5 Descripción textual CU_ Adicionar recurso a un comentario (extend).	77
Anexo 6 Descripción textual CU_ Visualizar OA almacenado en el RHODA.	78
Anexo 7 Descripción textual CU_ Editar OA almacenado en el RHODA.	78
Anexo 8 Descripción textual CU_ Gestionar recurso web.	79
Anexo 9 Descripción textual CU_ Gestionar ejercicio de orden cronológico.....	80
Anexo 10 Descripción textual CU_ Gestionar ejercicio de buscar en el texto.	83
Anexo 11 Descripción textual CU_ Gestionar ejercicio de enlazar columnas.	86
Anexo 12 Descripción textual CU_ Gestionar cuestionario.	88
Anexo 13 Diagrama de clases de análisis CU_ Editar colaboración.....	91
Anexo 14 Diagrama de clases de análisis CU_ Configurar pizarra virtual.	91
Anexo 15 Diagrama de clases de análisis CU_ Visualizar pizarra virtual.	91
Anexo 16 Diagrama de clases de análisis CU_ Mostrar área de edición del OA.	92
Anexo 17 Diagrama de clases de análisis CU_ Adicionar recurso a un comentario.	92
Anexo 18 Diagrama de clases de análisis CU_ Visualizar OA almacenado en el RHODA.	92
Anexo 19 Diagrama de clases de análisis CU_ Editar OA almacenado en el RHODA.	93
Anexo 20 Diagrama de clases de análisis CU_ Gestionar recurso web.	93
Anexo 21 Diagrama de clases de análisis CU_ Gestionar ejercicio de orden cronológico.....	93
Anexo 22 Diagrama de clases de análisis CU_ Gestionar ejercicio de buscar en el texto.	94
Anexo 23 Diagrama de clases de análisis CU_ Gestionar ejercicio de enlazar columnas.	94
Anexo 24 Diagrama de clases de análisis CU_ Gestionar cuestionario.	94
Anexo 25 Diagrama de clases de diseño CU_ Editar colaboración.	95
Anexo 26 Diagrama de clases de diseño CU_ Configurar pizarra virtual.	96
Anexo 27 Diagrama de clases de diseño CU_ Visualizar pizarra virtual.....	97
Anexo 28 Diagrama de clases de diseño CU_ Mostrar área de edición del OA.	98
Anexo 29 Diagrama de clases de diseño CU_ Adicionar recurso a un comentario.	99

Anexo 30 Diagrama de clases de diseño CU_ Visualizar OA almacenado en el RHODA.	100
Anexo 31 Diagrama de clases de diseño CU_ Editar OA almacenado en el RHODA.	101
Anexo 32 Diagrama de clases de diseño CU_ Gestionar recurso web.....	102
Anexo 33 Diagrama de clases de diseño CU_ Gestionar ejercicio de orden cronológico.	103
Anexo 34 Diagrama de clases de diseño CU_ Gestionar ejercicio de buscar en el texto.	104
Anexo 35 Diagrama de clases de diseño CU_ Gestionar ejercicio de enlazar columnas.	105
Anexo 36 Diagrama de clases de diseño CU_ Gestionar cuestionario.....	106
Anexo 37 Descripción de las clases de la base de datos.	107

Introducción

En la actualidad se aplican con mayor fuerza las Tecnologías de la Información y las Comunicaciones (TICs) en la Educación, por lo que es difícil ignorar los cambios y novedades que esta aplicación ha provocado en el proceso de enseñanza-aprendizaje.

Una de esas novedades es la aparición del e-learning, el cual ha sido definido de muchas maneras, sin embargo para el desarrollo de esta investigación se tratará al mismo como *“un conjunto de aplicaciones y servicios orientados a facilitar la enseñanza a través de Internet/Intranet, que facilitan el acceso a la información y la comunicación con otros participantes”*. (Red TNet, 2005)

Dentro del e-learning fue apareciendo de manera gradual la necesidad de compartir recursos y reutilizarlos en cualquier área del conocimiento. Esto conllevó a que se comenzaran a buscar soluciones para ello y como resultado surgió la tecnología Objetos de Aprendizaje (OA).

En estos momentos no existe una definición única para esta tecnología, varias han sido las organizaciones, instituciones y personas que han emitido un concepto para los OA, algunos de ellos muy difíciles de llevar a la práctica, por lo que se toma como definición de OA para este trabajo el emitido por la doctora Clara López Guzmán, donde plantea que un OA es *“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)

Una característica notable en las diferentes definiciones existentes de los OA es la reutilización, la cual trae consigo muchas ventajas, entre ellas el ahorro de tiempo y de esfuerzo en la creación de OA. Para lograr dicha reutilización es necesario que el OA se encuentre descrito de forma correcta a través de descriptores o metadatos, los cuales deben abordar cuestiones relacionadas con la autoría, los derechos de la misma, la descripción del propio OA, su nombre, las versiones por las cuales ha transitado, la localización, entre otros (ADL, 2001) permitiendo de esta forma ser identificados. Entiéndase por metadatos al *“conjunto de atributos o elementos necesarios para describir un OA”*. (López, 2005)

A partir de la necesidad de gestionar los OA, varias son las herramientas existentes que permiten crear, almacenar y distribuir los mismos. Entre ellas se encuentran los repositorios de objetos de aprendizaje (ROA) y las herramientas de autor (HA).

El objetivo principal de los ROA es almacenar los OA para su búsqueda y localización, entre los más conocidos se encuentran: MOREA, SLOR, Merlot.

Otras de las herramientas mencionadas son las HA, las cuales permiten al profesor diseñar y crear los contenidos educativos de manera interactiva, sin poseer conocimientos de programación. Algunas de ellas son: ReLOAD, Manifest Generator Pro, Autore, eXeLearning, Lompad y CRODA.

Para garantizar la reutilización, durabilidad y accesibilidad de los OA a través de estas herramientas es necesario el uso de estándares. *“Un estándar es un modelo que se sigue para realizar un proceso, es un patrón, una tipificación o una norma de cómo realizar algo”*. (AULAGLOBAL, 2005) Los estándares van a contener las especificaciones técnicas y de calidad que deben reunir todos los productos y servicios para cumplir satisfactoriamente con las necesidades para las que han sido creados y para poder competir internacionalmente en condiciones de igualdad. Los estándares más utilizados actualmente con esa finalidad son: LOM, SCORM, Dublin Core.

Otra tendencia importante en la actualidad a tener en cuenta para la elaboración de OA es el trabajo colaborativo. Esta palabra aparece para indicar el entorno de trabajo donde los participantes en grupo colaboran y se ayudan en la realización de un objetivo, logrando el intercambio de experiencias, sugerencias y reflexiones entre los miembros de dicho grupo. Por lo que es necesario que surjan herramientas que permitan realizar este trabajo colaborativo y así lograr transformar el trabajo individual en un producto con más calidad, que contemple las observaciones efectuadas por los compañeros del equipo.

La Universidad de las Ciencias Informáticas (UCI), como institución de avanzada en el uso de las TICs en su proceso docente-educativo, ha introducido la tecnología OA en función de elevar la calidad de los recursos didácticos que se produzcan.

Actualmente la UCI cuenta con un repositorio de objetos de aprendizaje. Este repositorio brinda la posibilidad de empaquetar y catalogar los OA con una versión del estándar SCORM, pero no cuenta con posibilidades para crear los contenidos que componen los mismos, ya que ésta no es la finalidad de este tipo de herramienta. El mismo se encuentra en una fase piloto, y actualmente contiene un número pequeño de OA, resultando necesaria la disposición de una HA que contribuya a impulsar la producción de los mismos.

Por otro lado, en la UCI se ha desarrollado también una herramienta de autor llamada CRODA, que aún no se encuentra en explotación, la cual permite la incorporación de objetos de información a la estructura de un OA, así como el diseño de diversas estructuras de OA a partir de los elementos didácticos que determine el autor.

Esta herramienta no brinda algunas funcionalidades que son consideradas de gran importancia en las tecnologías vinculadas a los OA. Ejemplo de esto lo constituye la ausencia de un espacio de trabajo que permita a varios usuarios aportar durante el proceso de elaboración de un OA, desaprovechando experiencias de colectivos de especialistas en una materia específica.

Por otro lado, los especialistas en contenido, no son los únicos actores necesarios para la creación de OA, ya que en ocasiones resulta imprescindible la intervención de especialistas técnicos y de otros perfiles, con lo cual estos equipos de trabajo multidisciplinarios necesitan un espacio de intercambio, y no siempre es sencillo realizar esto en un espacio físico, ya que los mismos pueden encontrarse dispersos y en funciones diversas.

Finalmente, la herramienta CRODA no posibilita la comunicación con el repositorio de objetos de aprendizaje de la UCI, por lo que es imposible la importación y exportación de los OA.

Por todo lo anteriormente expuesto se plantea como **problema de investigación**: ¿Cómo garantizar en la herramienta de autor CRODA la creación de objetos de aprendizaje por parte de equipos de trabajo multidisciplinarios, facilitando la incorporación de los mismos en el repositorio de la Universidad de las Ciencias Informáticas?

Se define como **objeto de estudio** la gestión de OA y el **campo de acción** lo constituyen las tecnologías que permiten la creación colaborativa de OA y la interoperabilidad entre aplicaciones e-learning.

De acuerdo al problema planteado anteriormente se propone como **objetivo general** incorporar en la herramienta de autor CRODA funcionalidades para la creación colaborativa de OA y la interoperabilidad con el repositorio de objetos de aprendizaje de la Universidad de las Ciencias Informáticas.

Objetivos específicos:

- Realizar el análisis y diseño de un módulo para la herramienta de autor web CRODA que permita la creación colaborativa de OA.
- Realizar la implementación de un módulo para la herramienta de autor web CRODA que permita la creación colaborativa de OA y la interoperabilidad entre esta y el repositorio de objetos de aprendizaje.

Como **idea a defender** se plantea que con la incorporación de funcionalidades que permitan la creación colaborativa de OA en la herramienta de autor CRODA, así como la interoperabilidad de esta con el repositorio de objetos de aprendizaje de la Universidad de las Ciencias Informáticas, se enriquecerá la elaboración de dichos objetos a partir de la experiencia de sus autores, facilitando la gestión de los mismos.

Para el cumplimiento de los objetivos se plantean las siguientes **tareas**:

- Análisis del proceso de gestión de los OA.
- Análisis de las tecnologías y herramientas fundamentales para la creación colaborativa de contenidos.
- Realizar los modelos correspondientes al análisis y diseño del módulo para la creación colaborativa de OA.
- Realizar los modelos correspondientes al análisis y diseño de las funcionalidades que permitirán la interoperabilidad entre los sistemas CRODA y RHODA.
- Realizar la implementación del módulo para la creación colaborativa de OA.
- Realizar la implementación de las funcionalidades que permitirán la interoperabilidad entre los sistemas CRODA y RHODA.

Los **métodos teóricos** utilizados en la investigación son el **analítico-sintético**, al identificar y analizar los principales conceptos y definiciones relacionados con el proceso de creación de OA; el **histórico-lógico**, al estudiar cómo ha ido evolucionando la creación de OA, así como el trabajo con las herramientas colaborativas y la **modelación**, al confeccionar diagramas que permitan modelar aspectos del módulo que posibiliten su implementación.

El presente trabajo consta de una introducción, cuatro capítulos, conclusiones generales, recomendaciones, referencias bibliográficas y por último, los anexos.

Capítulo I Fundamentación teórica: Se tratan de forma general los principales conceptos a tener en cuenta en la investigación, tales como e-learning, OA, metadatos, estándares, entre otros.

Capítulo II Características del sistema: Se detalla el flujo existente de los procesos involucrados en el objeto de estudio planteado, analizando cómo se ejecutan actualmente dichos procesos, además se presenta la propuesta del sistema, modelo de dominio; se determinan los requerimientos funcionales y no funcionales de la aplicación y se realiza el modelo de caso de uso del sistema, describiendo los casos de uso y los actores de este.

Capítulo III Análisis y diseño del sistema: Se presentan los diagramas de clases de análisis, interacción, clases del sistema, además el diseño de la base de datos.

Capítulo IV Construcción de la solución propuesta: Este capítulo está dedicado a la fase de Implementación donde se generarán los diagramas de despliegue y componente.

Capítulo I: Fundamentación teórica.

1.1 Introducción.

En el presente capítulo se precisan elementos teóricos que sustentan la investigación y desarrollo del módulo. Se tratan las principales definiciones relacionadas con el diseño y creación de los OA, así como los estándares para su elaboración, descripción y para la interoperabilidad entre sistemas.

Además se realiza un estudio de un conjunto de HA que se utilizan en el mundo para la creación de OA y de otras herramientas que permiten lograr una colaboración en un contexto educativo. Se fundamenta la selección de la metodología de desarrollo de software, tecnologías a utilizar en la web, servidores web, gestores de base de datos (SGBD) y frameworks de desarrollo.

1.2 Objetos de Aprendizaje.

Con la incorporación de las TICs al proceso de enseñanza aprendizaje surgieron condiciones que posibilitaron la flexibilización de la enseñanza, dando paso a un nuevo modelo de aprendizaje en el que no era necesaria la presencia física del alumno, surgiendo de esta manera el e-learning.

Dentro de los entornos *e-learning* participan individuos con distintos intereses y objetivos, así como contenidos con características, objetivos y formatos de todo tipo. Referente a los contenidos, la tendencia actual es que éstos sean reutilizables e intercambiables entre aplicaciones, esto ha dado origen a los OA. (López, 2005)

Los OA deben cumplir con un conjunto de características entre las cuales se destaca, por su presencia en todas las definiciones, la capacidad para ser reutilizados. Otros autores plantean que los OA deben ser interoperables, accesibles, durables, independientes, autónomos, versátiles, flexibles y funcionales. (García, 2005)

Para garantizar que un OA sea reutilizable, el mismo debe contener un grupo de metadatos donde se recoge información referente a: nombre, autor, fecha de creación, descripción, versión, localización, entre otros.

La posibilidad de que los OA sean reutilizados en diferentes contextos y plataformas refleja que deben tener la capacidad de ser interoperables, es decir, importados y exportados sin presentar problemas de compatibilidad (López, 2005), implicando la necesidad de usar estándares para la creación de los OA.

1.3 Estándares y especificaciones para la gestión de Objetos de Aprendizaje.

En el campo de la educación inmerso en las TICs han surgido variadas organizaciones con múltiples productos. La aparición de estos ha traído como resultado que se busquen vías para lograr una estandarización entre ellos con el objetivo de lograr una comunicación, reutilización, accesibilidad, interoperabilidad y durabilidad.

El conjunto de especificaciones y estándares de e-learning buscan lograr una unidad y entendimiento entre los procesos asociados para hacerlos de una forma más eficiente. Varios de los estándares definen un modelo de empaquetamiento estándar para los contenidos donde se especifica cómo representar la estructura del contenido y como agregar distintos recursos educativos para su transporte entre distintos entornos. Los contenidos pueden ser empaquetados como OA, de tal forma que permitan a los programadores crearlos de modo que puedan ser fácilmente reutilizados e integrados en distintos cursos, además de garantizar la durabilidad de estos. Otros estándares normalizan aspectos como la descripción (mediante metadatos) de los OA de modo que puedan ser gestionados, indexados y clasificados y almacenados en repositorios.

El empleo de los estándares facilita la interoperabilidad entre los sistemas e-learning, ya que estos pueden intercambiar materiales empaquetados según los estándares definidos y adoptados, un sistema que los soporte (por ejemplo, una herramienta de autor, un sistema de gestión del aprendizaje o una biblioteca digital de recursos educativos) será capaz de *abrir* los paquetes, independientemente de la forma y el lugar en los que dichos paquetes hayan sido producidos.

1.3.1. Estándares para empaquetar OA.

Ante las múltiples propuestas provenientes de diversas organizaciones en cuanto a estándares de empaquetado, referentes a los OA, resaltan solamente dos de ellas por sus adopciones en varias herramientas dentro del campo del e-learning. Estas propuestas están vinculadas a los estándares: IMS Content Packaging y SCORM; encontrándose ambos con una estrecha relación debido a la integración que hace SCORM de IMS Content Packaging.

- **IMS Content Packaging.**

El IMS (Instruational Managment System) Content Packaging es una variante de estándar para el empaquetamiento de contenidos promovida por IMS Global Learning Consortium.

IMS Content Packaging describe estructuras de datos que pueden ser usadas para intercambiar información entre sistemas que deseen importar, exportar, agregar, y desagregar paquetes de contenido. La especificación habilita exportar contenido de un LMS o repositorio digital e importarlo hacia otro mientras se retiene la información que describe los medios en el paquete de contenido IMS y su estructura, tales como una tabla de contenidos o cual página HTML mostrar primero. (Ministerio de Educación y Ciencia, 2008)

La especificación IMS Content Packaging se enfoca en el empaquetamiento y distribución de recursos, pero no determina la naturaleza de esos recursos. Además permite a quienes lo empleen juntar, estructurar y agregar contenido en una variedad de formatos ilimitada. Un típico paquete de contenido consiste de páginas web y formatos comunes de fotos, tales como JPEG.

La parte central de un paquete de contenido es el manifiesto. Un manifiesto describe el paquete lógico y las relaciones entre todos sus componentes. El manifiesto es a la vez un documento XML y más abstractamente, la información estructural en el documento. Dentro del manifiesto, la sección de los recursos funciona como una cuenta de materiales. Lista todos los archivos que son contenidos dentro del paquete y todas las referencias a recursos que residen en otros lugares. En ciertos casos puede contener además contenido estructural especializado en la sección de los recursos del propio documento

manifiesto. Finalmente la sección de organización del manifiesto estructura todos los componentes del paquete de contenido en una pieza de contenido educacional. (Ministerio de Educación y Ciencia, 2008)

La especificación IMS Content Packaging se enfoca solo 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.

- **SCORM.**

SCORM es un conjunto de estándares y especificaciones técnicos para productos software e-learning producido por Advanced Distributed Learning (ADL) un grupo de investigaciones patrocinado por el Departamento de Defensa de los Estados Unidos. SCORM presenta a los programadores una forma de cómo escribir sus códigos de manera que estos sean compatibles en otro software e-learning. Específicamente, SCORM gobierna cómo el contenido de aprendizaje online y las plataformas se comunican entre sí. SCORM no se involucra con modelos pedagógicos, es simplemente un estándar técnico.

SCORM significa “modelo referenciado de objetos de contenido compatible”. “Objeto de contenido compatible” indica que SCORM define como crear materiales de manera tal que puedan ser reutilizados y compartidos a través de diferentes sistemas y contextos. “Modelo de referencia” refleja que SCORM no es realmente un estándar sino que referencia estándares existentes y guía a los desarrolladores a emplearlos adecuadamente en su conjunto. Esto es debido a que ADL no partió desde cero en su elaboración sino que la industria poseía un grupo de estándares existentes que resolvían una parte del problema.

SCORM está compuesto de tres libros técnicos (SCORM, 2008):

- El empaquetamiento de contenido (Content Packaging): Este especifica como el contenido debe ser empaquetado y descrito. Está basado fundamentalmente en XML.
- El ambiente en tiempo de ejecución (Run-Time Environment): Este especifica cómo el contenido debe ser distribuido y cómo se comunica con el LMS. Está basado principalmente en ECMAScript (JavaScript).

- La secuencia y navegación (Sequencing and Navigation): Este especifica como el aprendiz puede navegar entre las diversas partes del curso. Está definido por un conjunto de reglas y atributos escritos en XML.

SCORM propone básicamente cuatro cualidades necesarias para lograr OA reutilizables efectivamente. (Fernández, et al., 2007)

- Reutilización: el contenido debe ser independiente del contexto de aprendizaje, apto para su uso en diferentes situaciones, públicos, plataformas de entrega con diversas herramientas de desarrollo.
- Interoperabilidad: el contenido debe funcionar en múltiples programas de aplicación, ambientes, hardware y software, independientemente de las herramientas usadas para crearlo o entregarlo.
- Durabilidad: el contenido debe continuar operando sin requerir modificaciones ante cambios o actualizaciones en el hardware o el software del sistema.
- Accesibilidad: el contenido debe ser identificable y ubicable cuando se necesite, para los requerimientos formativos necesarios. Debe poder conocerse su adecuación a los objetivos sin necesidad de obtener el propio contenido o pagar derechos por él, mediante la provisión de información suficiente acerca de cada OA.

Las características mencionadas de SCORM hacen factible su aplicación en el módulo a desarrollar teniéndose en cuenta además su amplio empleo a nivel mundial, su soporte por una reconocida organización y ser un modelo de referencia que integra varios estándares y especificaciones logrando una interoperabilidad entre los múltiples ambientes gestores del aprendizaje.

SCORM hace referencia también a otros estándares vinculados con la descripción de los OA. Estos especifican y detallan los metadatos que son admisibles en dependencia del seleccionado, pues SCORM no impone el empleo único de uno de ellos sino que recomienda y deja abierta la opción.

1.3.2. Estándar para describir OA.

Los metadatos permiten adicionar cierta información a los materiales digitales de forma tal que describan algunos aspectos concernientes a diversas características de los mismos. Estos a su vez facilitan su clasificación, posterior recuperación y reutilización. Además enriquecidos de esta manera pueden ser almacenados en diversos sistemas como los ROA, los cuales proveerán de consultas que permitan recuperar aquellos materiales almacenados que cubran una necesidad pedagógica en específico.

En la comunidad e-learning uno de los mayores logros alcanzados en cuanto a estándares de metadatos lo constituye el IEEE LOM.

LOM distingue 9 categorías de metadatos diferentes: (IEEE, 2002)

- Categoría general (general): Representa información general sobre el material educativo, que describe el mismo como un todo.
- Categoría ciclo de vida (lifecycle): Agrupa metadatos referidos a la historia y estado actual del proceso de producción y mantenimiento del material educativo por parte de los autores.
- Categoría meta-metadatos (meta-metadata): Agrupa información relativa a los metadatos en sí.
- Categoría técnica (technical): Agrupa metadatos relativos a las características y requisitos técnicos del material en sí.
- Categoría educativa (educational): Agrupa metadatos relativos a los usos educativos del material.
- Categoría derechos (rights): Agrupa metadatos relativos a los derechos de propiedad e intelectuales del material.
- Categoría relación (relation): Agrupa los metadatos utilizados para establecer relaciones entre el material y otros materiales.
- Categoría anotación (annotation): Anotaciones y comentarios sobre el material educativo.

- Categoría clasificación (classification): Metadatos para la clasificación del material en taxonomías.

Existe además otro esquema: Dublin Core (enfocado al ámbito bibliotecario), el cual no es tan amplio como LOM en cuanto a posesión de elementos, pues este último posee más de sesenta elementos a describir mientras que Dublin Core solo posee quince. LOM será el esquema a utilizar debido al gran factor descriptivo que contiene y a que es recomendado por SCORM siendo incluso referenciado por este.

Todos los estándares definidos anteriormente encuentran una aplicación práctica en las llamadas herramientas de autor las cuales son capaces de seguir las normas establecidas dando así surgimiento a los OA.

1.4 Herramientas de Autor.

Las HA fueron desplegadas para la creación de materiales educativos; son aplicaciones que permiten un trabajo constructivista para generar un entorno de aprendizaje dinámico, dentro de las funcionalidades que este tipo de herramienta presenta se puede destacar la posibilidad de crear actividades o pequeñas aplicaciones desde la misma herramienta. (CARRODEGUAS, 2008)

Existen muchas y diversas herramientas de autor que permiten la creación de contenidos, todas con características y funcionalidades diferentes, ejemplo de ellas son ReLOAD, Autore, eXe, Lectora y CRODA. A continuación se analizarán las mencionadas destacando sus principales características y funcionalidades.

1.4.1. ReLOAD (Reusable eLearning Object Authoring & delivery).

Reload es una herramienta gratuita y de código abierto, que puede editar, pre-visualizar y empaquetar contenidos en diferentes tipos de estándares (SCORM, IMS). Entre las funcionalidades que ofrece 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.

La versión 2.5.5 incluye soporte para IEEE LOM, SCORM 2004, IMS CP últimas versiones. Un inconveniente de dicha herramienta es su editor, el cual está en inglés y aunque admite la elección de cambiar a otros idiomas, es limitado, modifica sólo algunos términos...También posee una organización un tanto compleja para aquellos profesores sin conocimiento previo de informática. (González, 2005)

1.4.2. *Autore.*

AUTORE es una herramienta para la creación y visualización de materiales docentes. Permite tanto producir OA como buscar objetos de otros autores, importarlos, exportarlos y publicarlos en varios formatos...Ofrece la posibilidad de aprender de una manera interactiva al usuario final. (Romo, 2002)

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 inglés, euskera y castellano. Sin embargo tiene como limitación que los OA no son recursos flexibles pues son limitados en su creación a una estructura aportada por la herramienta que no está sujeta a modificaciones.

1.4.3. *eXe.*

Es un programa de código abierto, que permite a los profesores generar contenidos educativos sin necesidad de tener amplios conocimientos del lenguaje HTML o XML. A través de editores HTML, posibilita la creación de los contenidos. Permite exportar los cursos a SCORM. No brinda plantillas para la generación de cursos. No posibilita importar objetos SCORM, solamente objetos creados en eXe y que hayan sido guardados con su extensión (elp).

1.4.4. *Lectora.*

Lectora es una aplicación utilizada para la creación de cursos, incluye herramientas para capturar, grabar y editar videos y crear presentaciones multimedia. Brinda variadas opciones de autoría (ej. texto, imágenes, audios, videos, documentos). La creación de algunos contenidos requieren de conocimientos de programación (ej. menús.). Permite exportar a SCORM pero no brinda la posibilidad de importar objetos SCORM. Y no posibilita generar cursos a partir de plantillas.

1.4.5. CRODA

Aplicación Web que proporciona y facilita la creación de OA empleando el estándar SCORM. Al incorporar a los OA este estándar se garantiza la interoperabilidad y reusabilidad de los mismos en diferentes plataformas que utilicen dichos estándares.

Esta aplicación presenta una interfaz agradable y de cómodo uso. Facilita el acceso a los diferentes servicios manteniendo una simplicidad en su arquitectura de información.

Dicha herramienta brinda además la posibilidad de crear plantillas, que definen una estructura inicial, que posteriormente podrán ser empleadas en la elaboración de algún OA, haciendo posible además que las mismas se encuentren disponibles para todos los usuarios después de que hayan pasado por un proceso de revisión que garantice una correcta estructura de diseño.

En el proceso de elaboración de un OA, la herramienta da la posibilidad de modificar la estructura inicial aportada por una plantilla y permite incorporar información y contenido de diversas características. A los objetos que pasen por este proceso de creación se les incorporan metadatos que describen muchos de sus elementos facilitando la reutilización e identificación en otras herramientas que realicen búsquedas. La incorporación de estos metadatos se implementa con el uso del estándar LOM.

Un inconveniente de CRODA es el no brindar una comunicación con un ROA perdiendo la oportunidad de guardar los OA finalizados en un espacio donde podrán ser accedidos y reutilizados con mayor facilidad, así como importar OA existentes en el repositorio hacia el área de desarrollo para continuar el trabajo e incorporar mejoras.

1.5 Herramientas para el trabajo colaborativo.

Las herramientas para el trabajo colaborativo utilizadas en la educación, garantizan que un grupo de profesores se involucren en la creación de los recursos educativos. En esta colaboración cada miembro del grupo podrá contribuir aportando sus experiencias, sugerencias y reflexiones. Van a poder colaborar y ayudar en la realización de un objetivo determinado. Pudiendo así diseñar materiales y recursos

adaptados a las características de los estudiantes, garantizando el intercambio de experiencias y transformar el trabajo individual a un producto con mayor calidad.

Algunas de las herramientas que desarrollan el trabajo colaborativo son las que se presentan a continuación: Edukalibre y BSCW.

1.5.1. Edukalibre.

Edukalibre es un proyecto financiado por la Comisión Europea bajo el programa Sócrates/Minerva (destinado a la promoción de las tecnologías de la información y comunicación en el campo de la educación). El principal objetivo permitir a los usuarios crear material educativo (documentación en general) de forma colaborativa, añadiendo funcionalidad extra a este proceso, como sistemas de control de versiones, conversiones automáticas de documentos (html, PDF) o diferentes vías de editar y actualizar los documentos (con un editor, con un navegador web).

El componente básico del sistema son los documentos, los cuales tienen un formato base que permite al sistema realizar operaciones interesantes sobre ellos, en este momento estos formatos base son DocBook y LaTeX. Estos documentos son almacenados en un sistema de control de versiones (Subversion en este momento) que permite a los usuarios tener disponible la historia de cada uno de ellos. Además el sistema tiene la funcionalidad de crear conversiones automáticas de los documentos a diversos formatos (html, pdf, ps, texto). (González-Barahona, 2002)

Presenta las siguientes ventajas:

- 1 Permite a diferentes autores colaborar en la elaboración de un documento.
- 2 Los autores pueden revisar la historia de un documento y tienen acceso a cada una de las diferentes versiones del mismo.
- 3 Los usuarios tienen libertad para elegir la herramienta de edición de documentos que prefieran, además de poder modificar un documento usando un editor incluido en una página web mediante un simple navegador.
- 4 Los usuarios pueden seleccionar la herramienta para interactuar con el sistema que prefieran entre diferentes alternativas que van desde un simple navegador web o un cliente de WebDAV.

- 5 El sistema genera conversiones automáticas de los documentos introducidos en el sistema. Conversiones de Ficheros DocBook/XML, LaTeX y OpenOffice a formatos listos para ser impresos (como PDF o PostScript) o a formato HTML listo para ser publicado en Internet.

1.5.2. BSCW.

BSCW, Basic Support for Cooperative Work, es una plataforma o aplicación informática que permite el trabajo en grupo de manera sencilla. Es multiplataforma y su interfaz está disponible en múltiples idiomas. Su uso es gratuito para instituciones educativas mientras que para el resto existe una tabla específica de costes.

Con BSCW, cada persona dispone de un espacio de trabajo en el que puede crear otros y reunir o invitar a que participen en cada uno de ellos a otras personas junto a las que llevarán a cabo proyectos de manera compartida. En cada espacio se puede trabajar en colaboración sobre información y documentos, de forma que cuando se invita a alguien a un espacio de trabajo, a la persona invitada le aparecerá ese espacio al que ha sido invitada en forma de carpeta entre sus propios espacios de trabajo.

Una importante función de BSCW para la asistencia al trabajo colaborativo es la de informar a los miembros de un espacio de trabajo compartido de que algo ha sucedido en él. Para ello, BSCW registra cada acción realizada sobre un objeto como un suceso o evento.

Un derecho de acceso de BSCW permite a un usuario realizar determinadas acciones sobre un objeto. La idea básica es que este pueda ser (R. Bentley, 1997):

- Privado: solo el usuario que lo ha creado en una carpeta privada pueda tener acceso al mismo.
- Compartido: los miembros del espacio de trabajo compartido presentan diversos derechos de acceso.

En BSCW los derechos de acceso se determinan mediante el rol o roles que un usuario desempeña. Los predefinidos son los siguientes (R. Bentley, 1997):

- Miembro.

- Miembro restringido.
- Gestor de un espacio de trabajo o documento.
- Propietario de un objeto.

Cuando se invita a un usuario a participar en un espacio de trabajo, se debe especificar qué rol desempeñará como nuevo miembro del mismo. El usuario invitado recibe entonces los derechos de acceso asociados a su rol sobre la carpeta y los objetos que esta contiene.

Las herramientas colaborativas analizadas permiten la elaboración de contenidos educativos por múltiples autores pero no conforman los llamados OA, pues no soportan la tecnología requerida ni siguen los estándares existentes que dan cumplimiento a la definición de OA. Las otras, aunque siguen y emplean los estándares existentes para el desarrollo de estos, no brindan una creación colaborativa, desaprovechando la experiencia de un conjunto de autores con la cual se pueden enriquecer, al incorporar un espacio de trabajo en el que se definan, compartan criterios y construyan OA en conjunto. Por lo tanto con la incorporación a CRODA del módulo a desarrollar se piensa dar solución al déficit presentado.

Este módulo contendrá una interfaz agradable y sencilla para los profesores que lo empleen, mantendrá la flexibilidad en la creación de los OA, empleará los estándares descritos y permitirá la creación a partir de plantillas definidas.

Otro de los inconvenientes encontrado en general es que no establecen comunicación con algún medio de almacenamiento de objetos de aprendizaje como por ejemplo un Repositorio de Objetos de Aprendizaje (ROA). De existir esta comunicación se garantizaría un almacenamiento en un medio seguro, confiable, duradero y extensible de los OA; además de que se podrían importar hacia las herramientas los que ya se encuentren almacenados permitiéndoles la modificación. Dentro de la solución de este trabajo también se incluye beneficiar a CRODA estableciendo comunicación con un repositorio garantizando que se puedan importar y exportar los OA.

1.6 Repositorio de Objetos de Aprendizaje.

En la actualidad la gran mayoría de los contenidos educativos se están construyendo como OA y se están recopilando en contenedores, conocidos como Repositorios de Objetos de Aprendizaje. Estos organizan y hacen accesibles para diferentes aplicaciones y perfiles de usuarios los objetos almacenados. Esto se logra fundamentalmente a través de la comunicación de aplicaciones interoperables y del intercambio de catálogos de metadatos compatibles. Como resultado de las propuestas de importantes grupos de estandarización, se ha dado paso a diferentes tecnologías y métodos de catalogación de recursos digitales, por lo que los proyectos de Repositorios de Objetos de Aprendizaje han ido en aumento. (López, 2005)

1.6.1. RHODA.

RHODA es un ROA desarrollado en la UCI, el cual se encuentra en fase piloto y ha sido desplegado en la UCI y el Ministerio del Poder Popular para la Educación Universitaria (MPPEU) en Venezuela. Dentro de sus principales funcionalidades se encuentran las siguientes: almacenamiento y descarga de OA; creación de paquetes SCORM; revisión de paquetes SCORM y exportación de OA al estándar SCORM 1.2. El éxito de un repositorio consiste en la cantidad de OA que contenga y la calidad de los mismos, sin embargo RHODA aún posee muy pocos, por lo que la disponibilidad de una herramienta de autor que genere los OA que necesita y sea capaz de establecer una comunicación con él que permita el almacenamiento de los OA creados, así como la reutilización de los existentes en la creación de nuevos, es inminente.

Para lograr la interoperabilidad entre el repositorio de la UCI y la herramienta de autor web CRODA es necesario el empleo por ambos de varios estándares que normalicen la comunicación y permitan el intercambio sin problemas de compatibilidad.

1.7 Estándares de interoperabilidad.

La mayoría de las aplicaciones se desarrollan por separado, lo cual trae consigo la dificultad de la interoperabilidad una vez terminadas. Entiéndase por interoperabilidad la capacidad de varios sistemas o aplicaciones de trabajar conjuntamente y compartir sus datos de forma transparente para el usuario. En el campo del e-learning este factor también está presente. De cierta forma constituye un freno para el

avance y desarrollo de la educación al no lograr y hacer más engorroso el empleo de distintos productos elaborados en diferentes plataformas.

La interoperabilidad brinda al usuario la posibilidad de acceder a diversas colecciones de recursos de información (datos y metadatos), gestionados por diferentes sistemas y en distintas localizaciones, sin dejar de utilizar la interfaz del sistema propio.

Existe un conjunto de estándares que son empleados a nivel de aplicación que logran vincularlas permitiendo el flujo de información; dándole de esta forma cumplimiento a la definición de interoperabilidad. En la solución a desarrollar se emplearán los estándares que se mencionarán a continuación por el empleo que hace de ellos el ROA con el cual establecerá vínculo la aplicación.

Open Knowledge Initiative (O. K. I.)

O.K.I define una arquitectura abierta y extensible para la tecnología de aprendizaje específicamente para las necesidades de la educación superior. Provee especificaciones detalladas para interfaces entre componentes de un ambiente gestor del aprendizaje y ejemplos en código abierto de cómo estas interfaces trabajan.

Esta arquitectura especifica como los componentes de un entorno de aprendizaje se comunican entre sí y con otros sistemas y permite que estos sean desarrollados y actualizados independientemente de cada uno. Esto trae consigo varios beneficios: (OKI, 2002)

- Las tecnologías de aprendizaje y los contenidos pueden ser fácilmente compartidos entre centros educacionales. Esto provee un catalizador para el desarrollo cooperativo y comercial.
- 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.
- La modularidad hace que las tecnologías de aprendizaje sean más estables, fiables, capaces de crecer con el aumento de su uso y permite que los componentes sean actualizados sin desestabilizar otras partes del ambiente.

- La arquitectura ofrece una base estándar para el desarrollo de software basado en tecnologías para el aprendizaje. Esto reduce el esfuerzo de desarrollo y alienta el desarrollo de componentes especializados que se integren en sistemas más grandes.

OKI proporciona un conjunto de interfaces para la definición de servicios denominados OSID's (Open Service Interface Definitions) y proporciona una implementación de referencia para cada uno de ellos.

OSID's son contratos de software y por lo tanto son compatibles con muchas otras tecnologías y especificaciones, tales como, SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language). Pueden ser usados con tecnología existente en código abierto o cerrado.

IMS Digital Repositories Interoperability (IMS-DRI)

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 búsquedas. Esta especificación se propone para la interoperabilidad entre servicios o aplicaciones que tienen 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. (López, 2005)

Esta especificación define a los repositorios digitales como una colección de recursos que son accesibles desde la red sin conocimiento previo de la estructura de la colección. Los repositorios pueden manejar recursos o metadatos que describen esos recursos, los cuales pueden o no estar en un mismo repositorio albergado.

Simple Query Interface (SQI)

Se trata de una especificación, enmarcada en el entorno de los ROA, que define una capa para facilitar las búsquedas de contenidos. Especifica un estándar y un API para resolver la problemática de las búsquedas de contenidos digitales en entornos heterogéneos como Internet, donde existen múltiples repositorios con recursos orientados a la educación que residen en entornos dispares y hacen complicado su acceso para usuarios externos. (Nguyen, et al., 2007)

El API define una serie de primitivas que permiten a los usuarios considerar los repositorios digitales como pares (peers) de una comunicación en la que un repositorio puede actuar como fuente de consultas y otro como objetivo de las mismas.

El interfaz SQL no establece la forma en la que los repositorios realizan la catalogación ni almacenan los metadatos de sus contenidos, ni define la manera en la que resuelven las consultas. Pero detalla, en el API, el mecanismo necesario para configurar tanto el lenguaje utilizado en la búsqueda, como el lenguaje de respuesta.

1.8 Metodologías de desarrollo de software.

El éxito de un proyecto de software involucra, entre otros elementos, la utilización de metodologías de desarrollo, entiéndase por estas, al conjunto de pasos y procedimientos a seguir que permiten estructurar, planear y controlar el proceso de desarrollo de software. El uso de una metodología define qué hacer, cómo y cuándo durante el planeamiento, desarrollo y mantenimiento de un proyecto. Entre las metodologías más conocidas se pueden mencionar a eXtreme Programming (XP), Microsoft Solution Framework (MSF) y Rational Unified Process (RUP).

Extreme Programming (XP)

XP es una metodología ágil que surge como resultado de un proyecto desarrollado por Kent Beck en DaimlerChrysler. Dicha metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo al usuario final, siendo este uno de los requisitos para llegar al éxito del proyecto. La mejora de la comunicación con los usuarios y los desarrolladores, así como la simplicidad al desarrollar y codificar los módulos del sistema es uno de sus principios básicos. XP presenta cuatro fases a tener en cuenta en el desarrollo del software: planificación, diseño, desarrollo y pruebas. (Fernández, 2002)

En la fase de planificación se realiza un permanente intercambio entre la parte empresarial y técnica del proyecto, en la que se definirá el alcance, la composición de las versiones y las fechas de las mismas, entre otros aspectos. Además en esta fase se redactan las historias de los usuarios, las que servirán para crear el plan estimado de entrega.

En la fase de diseño se elige una metáfora para el sistema, o sea una historia que todo el mundo puede contar acerca de cómo funciona el mismo. Para poderlo diseñar se deben usar las tarjetas CRC (Cargo o clase, Responsabilidad y Colaboración) en las reuniones del diseño para garantizar que todo el equipo trabaje en las tareas de este tipo. El nombre de la clase se coloca a modo de título en la tarjeta, las responsabilidades se colocan a la izquierda, y las clases que se implican en cada responsabilidad a la derecha. No se añadirá funcionalidad en las primeras etapas, debe centrarse en la tarea que se ha fijado para ese día y hacerlas lo mejor posible.

La fase de desarrollo debe reunir las siguientes características: el cliente deberá estar siempre disponible, no solo para ayudar al equipo de desarrollo sino formando parte del mismo. Todo el código debe programarse por parejas, de esta manera se incrementará la calidad del software desarrollado sin afectar el tiempo de entrega, mientras uno de ellos se encarga de pensar la táctica con la que se va a abordar el problema, el otro se encargará de pensar las estrategias para llevar dichas tácticas a la práctica. El código ha de ser desarrollado siguiendo los estándares de desarrollo para facilitar su lectura y modificación por cualquier miembro del equipo de desarrollo.

La última fase y no por eso menos importante: pruebas, son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Los clientes escriben las pruebas funcionales para cada historia de usuario que deba validarse. En este contexto de desarrollo evolutivo y de énfasis en pruebas constantes, la automatización para apoyar esta actividad es crucial.

Microsoft Solution Framework (MSF)

El MSF proporciona un sistema de modelos, principios y pautas que permiten que todos los elementos de un proyecto, tales como personas, procesos y herramientas, puedan ser manejados con éxito. Esta metodología se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. Entre sus principales características se encuentra la adaptabilidad, escalabilidad y flexibilidad. Es válido resaltar que esta metodología puede ser utilizada para desarrollar soluciones basadas sobre cualquier tecnología. (Sánchez, 2004)

MSF está compuesta por seis fases: visión, en la que se pretende dar un enfoque general del proyecto, se identifican las tareas y los entregables que permiten al equipo cumplir con los requerimientos y objetivos

del mismo; planeación, en la que se realiza la preparación de la especificación funcional, diseño de la solución, planes de trabajo, costes estimados y calendarios para los entregables, además de identificar y analizar los requerimientos del negocio, de usuario, operacionales y de sistema; desarrollo, en la cual se realiza la construcción de los elementos y entregables de la solución, incluidos los códigos de los componentes, infraestructura (software, hardware, red) y la documentación para el uso de las operaciones; estabilización, fase en la que se determina que la solución cumple los criterios de calidad necesarios y el equipo se centra en detectar y priorizar errores. Además de estas fases se encuentran la de instalación y soporte.

Rational Unified Process (RUP)

RUP constituye la metodología más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Su ciclo de vida se caracteriza por ser iterativo e incremental, centrado en la arquitectura y dirigido por casos de uso. A continuación se explican estas características:

- **Iterativo e incremental:** pues cada fase se desarrolla en iteraciones, las cuales involucran actividades de todos los flujos de trabajo, aunque es válido aclarar que algunos flujos se ven más representados que otros en cada iteración.
- **Centrado en la arquitectura:** en RUP la arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.
- **Dirigido por casos de uso:** los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los mismos.

Esta metodología define “quién” (trabajadores) debe hacer “qué” (artefactos), “cuándo” (flujo de trabajo y fases) y “cómo” (actividades) debe hacerlo. Por lo que sus principales elementos son:

- **Trabajadores:** define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.
- **Actividades:** es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.
- **Artefactos:** productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.
- **Flujo de actividades:** secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

El proceso de desarrollo del software llevado a cabo utilizando RUP, se divide en cuatro fases: inicio, en la que se definen la visión, los objetivos y el alcance del proyecto; elaboración, la cual tiene como principal finalidad completar el análisis de los casos de uso y definir la arquitectura del sistema; construcción, fase que se encarga de la elaboración de un producto totalmente operativo y eficiente; transición, en la cual se realiza el despliegue del producto y se capacita a los usuarios finales.

RUP consta de nueve flujos de trabajo: modelado del negocio, análisis de requisitos, análisis y diseño, implementación, prueba, distribución, gestión de configuración y cambios, gestión del proyecto y gestión del entorno.

Luego del estudio de las metodologías mencionadas anteriormente, se decide utilizar RUP a pesar de conocerse como una metodología “pesada”, por las siguientes razones. Está enfocada en el desarrollo de software orientado a objetos, tipo de programación a utilizar en la implementación del sistema. Es adaptable, lo que permitirá que se realicen los cambios que sean necesarios. El lenguaje recomendado para la modelación del sistema es UML, el cual brinda amplias posibilidades en la representación, es de fácil uso y conocido por el equipo de proyecto. RUP genera gran cantidad de artefactos que permiten tener una amplia documentación del producto, aspecto importante teniendo en cuenta que el personal del

proyecto varía de año en año debido a que en su mayoría son estudiantes, por lo que la tenencia de dicha documentación permitirá que no se pierdan elementos del producto que pueden ser de utilidad para su continuidad. Otra de las características que hacen de RUP una buena elección para utilizar como metodología de desarrollo de software es el ser iterativo lo que permite reducir riesgos y dividir los proyectos en pequeños ciclos o iteraciones a través de cada una de las fases.

1.8.1. Herramientas CASE.

Las herramientas CASE (Computer Aided Assisted Automated Software Systems Engineering) son utilizadas para automatizar o apoyar una o más fases del proceso de desarrollo de software. Entre las más utilizadas se encuentran Rational Rose y Visual Paradigm, a continuación las principales características de cada una de ellas.

Rational Rose

Rational Rose es un entorno de modelado que permite generar código a partir de modelos Ada, ANSI C++, C++, CORBA, Java/J2EE, Visual C++ y Visual Basic. Ofrece un lenguaje de modelado común que agiliza la creación del software. Entre sus principales funciones se encuentra la generación de código en lenguaje Ada, ANSI C++, C++, CORBA, Java y Visual Basic, con funciones configurables de sincronización entre los modelos y el código, (IBM, 2002) siendo notable que aún existen varios lenguajes de programación que no soporta.

A pesar de permitir la correcta modelación del diagrama de clases persistentes, no es posible que esta herramienta exporte ese modelo hacia algún sistema gestor de bases de datos. Es la herramienta usada generalmente para el desarrollo de software propietario.

Visual Paradigm

Visual Paradigm al igual que Rational Rose también utiliza UML como lenguaje de modelado. Dicha herramienta soporta todos los diagramas UML, siendo esta la primera razón que justifica la selección de la misma para la modelación del sistema. Además genera documentación del sistema en formato PDF, HTML y Word (anónimo) y permite la generación de código a partir de diagramas. Esta herramienta CASE

puede ser utilizada para la modelación de procesos de desarrollo de software que sigan la filosofía de software libre, otra de las razones que justifican su selección.

Visual Paradigm permite realizar ingeniería tanto directa como inversa, pues a partir de un modelo relacional es capaz de desplegar todas las clases asociadas a las tablas. Además, esta herramienta es colaborativa, ya que soporta múltiples usuarios trabajando sobre el mismo proyecto. Es multiplataforma. Permite el control de versiones. (León, 2007)

1.9 Servidores Web.

Un servidor web es *“un programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo HTTPS”*. (Cibernetia, 2009)

Entre los servidores más conocidos se encuentran Internet Information Services (IIS) y Apache. A continuación las principales características de estos servidores web.

IIS

IIS ofrece una serie de servicios para los ordenadores como: FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), NNTP (Network News Transport Protocol) y HTTP/HTTPS. Se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. (Microsoft,2000) También pueden ser incluidos los de otros fabricantes, como PHP o Perl, aunque este servidor solo funciona con el sistema operativo Windows, por lo que es propiedad de Microsoft Corporation, lo que constituye una gran limitante para emplearlo en la solución.

Apache

Apache es gratuito, modular, código abierto, extensible y el más utilizado. Está diseñado para ser un servidor Web flexible y potente que puede funcionar sobre varias plataformas y entornos, además de

poder ser personalizado con el objetivo de mejorar las necesidades de cada sitio Web. En la solución se empleará el servidor Web Apache por las características antes mencionadas.

1.10 Gestores de Base de Datos.

Un Sistema Gestor de Base de Datos (SGBD) es un conjunto de programas que permite a los usuarios crear y mantener una base de datos (BD). Es un software de propósito general que facilita el proceso de definir, construir y manipular la BD para diversas aplicaciones. Los más conocidos a nivel mundial son Oracle, MySQL y PostgreSQL, a continuación se enuncian las principales características de estos.

Oracle (Relational Data Base Management System)

Oracle es uno de los sistemas de bases de datos más completos, donde se destaca su estabilidad y 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 utilice en empresas muy grandes y multinacionales, razón por la cual no es seleccionado para formar parte de la solución.

MySQL

Es uno de los sistemas de gestión de bases de datos SQL (Structured Query Language) más populares, es propietario y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código, por lo que no es posible su empleo para la solución.

Es increíblemente rápido, confiable, robusto y tiene gran facilidad de uso tanto para volúmenes de datos grandes como pequeños, esta aceptación es debida, en parte, a que existe una 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. La conectividad, velocidad y seguridad hacen que MySQL sea altamente conveniente para acceder a bases de datos en Internet.

PostgreSQL

Es un poderoso gestor 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 más avanzado en el mundo. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, soportando de esta manera una mayor cantidad de peticiones simultáneas de forma correcta. Este será el empleado en el desarrollo de la solución por los elementos planteados.

eXist-db

Para el logro de la solución se empleará un gestor de base de datos xml. El mismo posibilita guardar un xml en su estructura original y realizarle consultas para la obtención de información o modificación de su contenido. Debido a que las estructuras de un OA están expresadas en ficheros xml, llevarlas a una base de datos relacional traería varios inconvenientes como garantizar que se mantenga la integridad entre el esquema y la estructura física del xml.

Exist-db es un SGBD de código abierto construido usando tecnología XML. Almacena datos en XML de acuerdo al modelo de datos de ese mismo formato.

Soporta múltiples estándares de tecnologías (web principalmente) haciéndolo una excelente plataforma para el desarrollo de aplicaciones basadas en la web: (exist, 2008)

1. Xquery 1.0 / Xpath 2.0 / XSLT 1.0 ó XSLT 2.0.
2. Interfaces HTTP: REST, WebDav, SOAP, XMLRPC, Atom Publishing Protocol.

Exist-db es altamente compatible con el estándar Xquery. El motor de consulta es extensible y proporciona una larga colección de funciones Xquery.

Características de eXist-db: (exist, 2008)

Requerimientos de sistema.

Plataforma soportada. Es independiente de cualquier plataforma.

Software requerido. Ninguno. Un servidor web (Jetty) y Cocoon son incluidos en la distribución, pero eXist-db puede correr sin ellos.

Características del núcleo: (exist, 2008)

- **Almacenamiento de datos.** Almacenamiento de dato XML nativo basado en B+- árboles y archivos paginados. Nodos documentos son almacenados en un DOM persistente.
- **Colecciones.** Los documentos son gestionados en colecciones jerárquicas, similares al almacenamiento en un sistema de archivos.
- **Índice.** Basado en un esquema de indexado numérico el cual soporta una rápida identificación de relaciones estructurales entre nodos, tales como padre/hijo, ancestro/descendiente o anterior/posterior.
- **Mecanismos de autorización.** Permisos de acceso como los sistemas Unix para usuarios/grupos que accedan a colecciones y documentos.
- **Seguridad.** eXtensible Access Control Markup Language (XACML) para control de acceso Xquery.

EXist-db satisface las necesidades para integrarlo dentro de la solución a desarrollar al emplear lenguajes de consulta como Xquery y Xpath que trabajan directamente sobre xml; por su capacidad de almacenamiento y soporte de varios protocolos que permiten la interoperabilidad entre las aplicaciones.

1.11 Framework de desarrollo.

A partir del desarrollo alcanzado en el mundo de las tecnologías ha sido necesaria la creación de marcos de trabajo (framework) con el objetivo de facilitar el desarrollo de software, permitiendo a los diseñadores y programadores pasar más tiempo identificando requerimientos de software que tratando con los tediosos detalles de bajo nivel para proveer un sistema funcional.

Los framework proporcionan además una estructura al código fuente, forzando al desarrollador a crear código más legible y fácil de mantener. Facilitan la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

Entre los más conocidos están: Akelos, Zend Framework, CodeIgniter y Symfony.

Akelos

Akelos es una plataforma de desarrollo de aplicaciones web basada en el patrón de diseño MVC (Modelo - Vista - Controlador), que se enfoca en buenas prácticas que permiten programar Vistas que emplean AJAX de forma fácil, controlar las peticiones y las respuestas a través de las Controladoras, manejar aplicaciones estandarizadas internacionalmente, modelos de comunicación y de base de datos utilizando convenciones sencillas. (Akelos, 2006)

Zend Framework

Se trata de un framework para el desarrollo de aplicaciones y servicios Web con PHP, brinda soluciones para construir sitios web modernos, robustos y seguros. Además es código abierto. Se centra en el desarrollo de aplicaciones más robustas, confiables, modernas para la Web 2.0 y los servicios web, consumiendo una amplia gama de APIs disponibles de los principales proveedores Google, Amazon, Yahoo!, Flickr, Strikelron and ProgrammableWeb. (Zend, 2001)

CodeIgniter

Es un programa o aplicación web desarrollada en PHP para la creación de cualquier tipo de aplicación web bajo PHP. Es un producto de código abierto. Contiene una serie de librerías que sirven para el desarrollo de aplicaciones web. Marca una manera específica de codificar las páginas web y clasificar sus diferentes scripts, que sirve para que el código esté organizado y sea más fácil de crear y mantener. Se basa en el patrón de diseño MVC. (CodeIgniter. 2008)

Symfony

Facilita el desarrollo de las aplicaciones web, aumenta exponencialmente la productividad y ayuda a mejorar la calidad de las aplicaciones web, emplea el tradicional patrón de diseño MVC para separar las distintas partes que forman una aplicación web, la lógica de negocio, la lógica de servidor y la presentación. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Cuenta con miles de páginas de documentación distribuidas en varios libros gratuitos y decenas de tutoriales. (Symfony, 2003)

Symfony constituye un framework lo suficientemente maduro en el desarrollo de aplicaciones para empresas y posee además una gran cantidad de documentación, lo que propicia un mejor aprendizaje por parte de los desarrolladores. Además el sistema que se beneficiará con la solución está desarrollado sobre este framework por lo que continuar su empleo mantendrá un mejor desempeño y no romperá con la arquitectura.

ExtJs

ExtJs es una librería desarrollada en JavaScript que permite el desarrollo de aplicaciones web interactivas empleando tecnologías como Ajax, DHTML, DOM.

ExtJs proporciona una interfaz de usuario enriquecida, muy parecida a las que se encuentran en las aplicaciones de escritorios. Esto permite a los desarrolladores web concentrarse en las funcionalidades de las aplicaciones y no en los detalles técnicos. Puede trabajar en conjunto con otras librerías si se definen adaptadores para las mismas. Son estas características más la compatibilidad con diferentes navegadores y los múltiples manejos de eventos que posee las que impulsan a integrarlo en la solución, específicamente en el empleo y desarrollo de la interfaz de usuario.

1.12 Conclusiones.

Como resultado de la investigación y el análisis bibliográfico realizado, a lo largo de este capítulo, han sido expuestos los principales puntos de interés relacionados con los OA, los estándares utilizados durante su creación y descripción, así como los relacionados con la interoperabilidad entre sistemas. Se plantearon

además los objetivos de la HA, haciendo alusión a la HA “CRODA”. Fue analizado cómo es el proceso de creación colaborativa de algunas herramientas.

En este capítulo además se definió la metodología de desarrollo de software, herramientas, tecnologías, gestores de base de datos y frameworks a utilizar.

Capítulo II: Características del sistema.

2.1. Introducción.

En el presente capítulo se describirán las principales funciones y el flujo actual de los procesos relacionados en el campo de acción planteado anteriormente.

También se desarrolla el flujo de requerimientos 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 y las relaciones entre ellos, así como los casos de uso del sistema con sus descripciones textuales.

2.2. Objeto de Automatización.

En la herramienta de autor CRODA se incluirá un módulo que permitirá automatizar el proceso de creación colaborativa de un OA. Dicha automatización estará basada en la incorporación de recursos digitales para su discusión y empleo en la conformación de un OA. Además se brindará un área en la cual se expondrán los criterios de cada uno de los miembros del proceso colaborativo que servirán al creador de la colaboración para la edición del OA objeto de discusión. Se mantendrá el empleo del estándar SCORM, siendo responsable la aplicación de conformar un OA que cumpla con dicho estándar.

2.3. Propuesta del sistema.

El sistema propuesto constituirá parte de una aplicación web desarrollada en PHP basado en una versión mayor que la 5.0 y empleando el framework de desarrollo Symfony. Se hará uso del servidor de base de dato PostgreSQL y una base de datos nativa de XML. Por su ubicación en la web el sistema será de fácil acceso a los usuarios sin necesidad de emplear espacio en disco duro para su instalación y empleo. La mayoría de las acciones estarán accesibles. El sistema proporcionará una sencilla interfaz, facilitando el encuentro de las acciones a ejecutar. Básicamente el sistema debe permitir desarrollar un proceso colaborativo de creación de OA entre varios autores. Durante este proceso los mismos podrán incorporar

sus opiniones, recursos e ideas que en conjunto irán constituyendo el aporte individual que será traducido en un aporte colaborativo final. El sistema mantendrá el empleo del concepto de OA por lo que el proceso colaborativo estará sujeto a esta definición, así como a la utilización del estándar SCORM. Estas últimas características lo ponen por encima de otras soluciones para la creación colaborativa mencionados en el capítulo anterior pues las herramientas analizadas aunque permiten un desarrollo colaborativo sobre materiales educativos no están sujetas al empleo del concepto OA y SCORM, no aprovechando los beneficios brindados por estas tecnologías dentro de los que se encuentran la reutilización en diferentes entornos educativos de los materiales educativos creados. El otro grupo de herramientas analizadas si trabaja estos conceptos pero su deficiencia radica en la no incorporación de un espacio donde se pueda aprovechar la experiencia de varios autores para crear un OA con un mayor enriquecimiento en sus distintos aspectos.

2.3.1. Modelo de dominio.

No se realiza la modelación del negocio porque no fue posible determinar el proceso del negocio con fronteras bien establecidas donde se logren ver claramente las personas que lo inician, los beneficiados y los que desarrollan las actividades en cada uno de estos procesos, por lo que se decide realizar un modelo de dominio o conceptual.

El modelo de dominio es una representación visual de los principales conceptos u objetos del mundo real, significativos para un problema o área de interés. Este es de gran ayuda para desarrolladores y usuarios, ya que de esta forma utilizan un vocabulario común y pueden entender el contexto en que se enmarca el sistema (*Ver figura 1*).

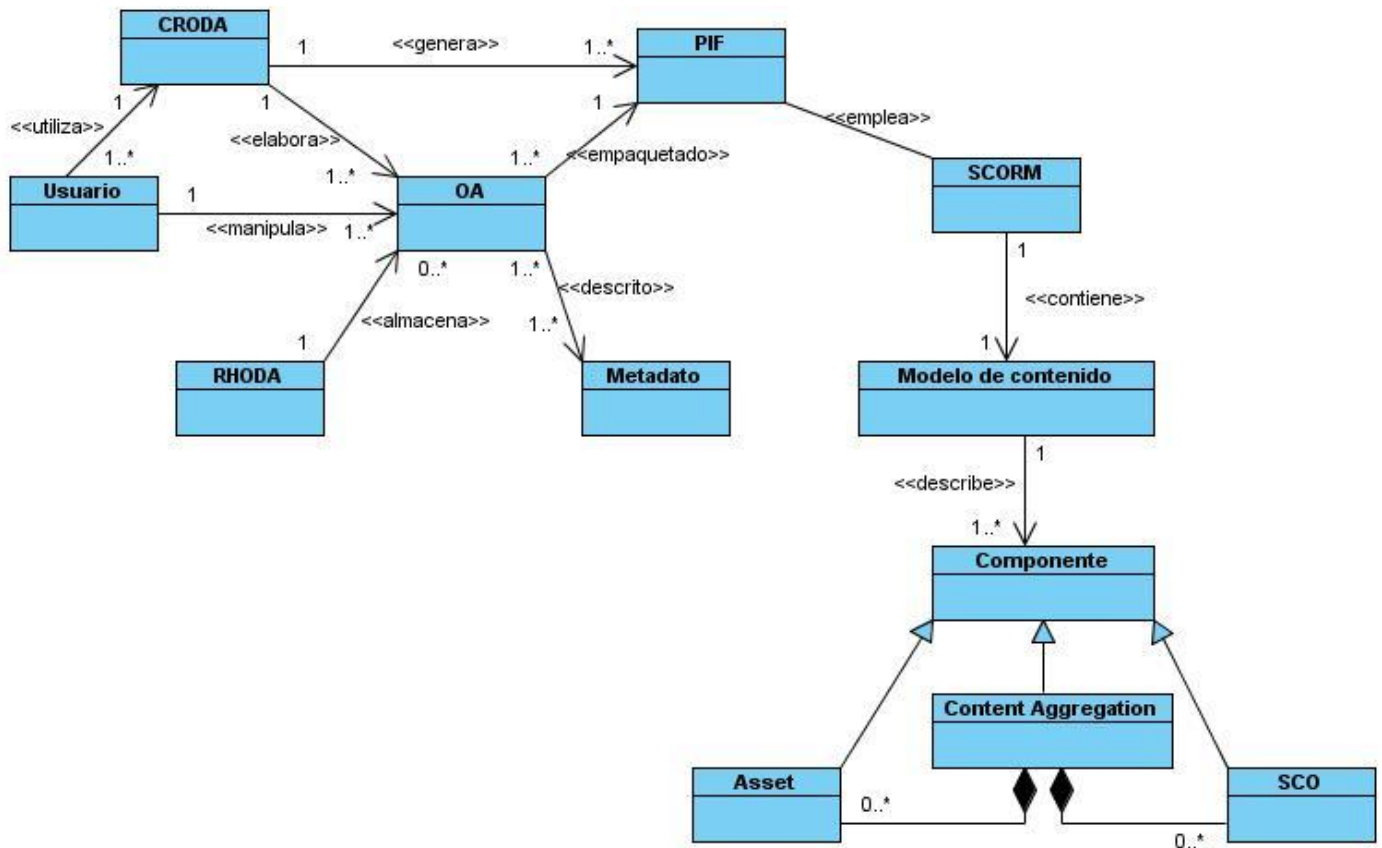


Figura 1 Diagrama del modelo del dominio

Definición de las clases del modelo de dominio

Usuario: persona que interactúa con la herramienta y que posee el rol de autor.

Objeto de Aprendizaje (OA): cualquier recurso con una intención formativa, compuesto de uno o varios elementos digitales, descrito con metadatos, que pueda ser utilizado y reutilizado para apoyar el aprendizaje.

CRODA: herramienta de autor web que permite la creación de OA.

RHODA: repositorio de objetos de aprendizaje que permite el almacenamiento de OA.

Package Interchange File (PIF): es la unión de los componentes del paquete de contenidos en forma de un fichero comprimido.

Metadato: son datos que describen otros datos, es decir, información relativa a los propios datos que facilitan su catalogación.

SCORM: es un modelo que integra un conjunto de estándares, especificaciones y guías diseñadas para lograr la accesibilidad, durabilidad, interoperabilidad y reusabilidad de contenidos y sistemas.

Modelo de contenido: describe los componentes SCORM utilizados para construir una experiencia de aprendizaje desde recursos de aprendizaje.

Componente: son utilizados para construir una experiencia de aprendizaje desde recursos de aprendizaje reutilizables.

Asset: son contenidos de aprendizaje en su forma más básica. Son la representación electrónica de textos, medias, imágenes, sonidos u otra pieza de información que puede ser entregada a un cliente Web.

SCO: representa una colección de uno o más asset, que incluye un asset específico que utiliza el Run-time Environment de SCORM para la comunicación con los LMSs.

Content Aggregation: es el proceso de agregación de recursos dentro de una estructura definida (estructura de contenido) para construir una experiencia de aprendizaje particular.

2.4. Especificación de requisitos.

2.4.1. Requerimientos funcionales.

R1. Seleccionar forma de creación del OA (individual o colaborativa).

R2. Definir datos generales de la colaboración.

R3. Adicionar miembro al equipo de trabajo colaborativo.

R4. Eliminar miembro del equipo de trabajo colaborativo.

- R5. Seleccionar plantilla a utilizar en el trabajo colaborativo.
- R6. Visualizar lista de colaboraciones.
- R7. Editar colaboración.
- R8. Enviar mensaje automático al colaborador eliminado de una contribución.
- R9. Proponer a eliminar una colaboración.
- R10. Asignar responsable de colaboración.
- R11. Eliminar colaboración.
- R12. Visualizar temas de debate.
- R13. Visualizar pizarra virtual.
- R14. Configurar pizarra virtual.
- R15. Crear tema de debate.
- R16. Añadir comentario a un tema de debate.
- R17. Subir recurso a un comentario.
- R18. Relacionar recurso con elemento de la estructura del OA.
- R19. Mostrar área de edición del OA.
- R20. Crear recurso web.
- R21. Editar recurso web.
- R22. Eliminar recurso web.
- R23. Crear ejercicio de orden cronológico.

- R24. Agregar elemento al ejercicio de orden cronológico.
- R25. Eliminar elemento del ejercicio de orden cronológico.
- R26. Crear ejercicio de búsqueda en el texto.
- R27. Agregar elemento del ejercicio de búsqueda en el texto.
- R28. Eliminar elemento del ejercicio de búsqueda en el texto.
- R29. Crear ejercicio de enlazar columnas.
- R30. Agregar elemento a una columna del ejercicio de enlazar columnas.
- R31. Eliminar elemento de una columna del ejercicio de enlazar columnas.
- R32. Crear ejercicio de selección simple.
- R33. Adicionar opción al ejercicio de selección simple.
- R34. Adicionar pregunta de selección simple.
- R35. Visualizar OA almacenado en RHODA.
- R36. Editar OA almacenado en RHODA.

2.4.2. Requerimientos no funcionales.

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

Usabilidad:

- Facilidad de aprendizaje (Asimilación del sistema por parte del cliente).

La capacidad por parte del usuario final para trabajar con el software constituye un requisito no funcional de usabilidad. Se considera que dadas las características de la aplicación el usuario puede entender completamente las funcionalidades en un tiempo máximo de 20 días, estableciendo como valor crítico 30 días.

- Intercambio o comunicación con el usuario.

El sistema poseerá una comunicación con el usuario. Durante el desarrollo del software se tendrá en cuenta el intercambio con el usuario para la ejecución de las diferentes funcionalidades a través de mensajes que logren un entendimiento común, dejando la toma de decisiones críticas en manos del cliente.

La totalidad de las funcionalidades se deben implementar siguiendo un mismo patrón de comportamiento para facilitar su comprensión.

Las funcionalidades estarán correctamente relacionadas de manera que se asegure la navegación y un acercamiento al trabajo normal del usuario.

Accesibilidad:

La información y las funcionalidades estarán disponibles y el usuario podrá acceder a ellas en todo momento.

Las funcionalidades visibles serán las correspondientes a cada usuario según su rol dentro de la aplicación.

Fiabilidad:

- Disponibilidad.

Una vez que el sistema esté publicado estará disponible durante las 24 horas del día.

Eficiencia:

- El sistema debe demorar como promedio en una transición de uno (1) a cinco (5) segundos, siendo este tiempo el correspondiente a los procesos que realizan consultas a la base de datos.
- Recursos tecnológicos requeridos:
 - Software para servidores.

- Servidores de base de datos: PostgreSQL 8.4 o inferior.
- Base de datos XML: Exist 1.4.
- Apache 2.
- PHP 5.
- Hardware mínimo para el servidor.
 - 1 GB de memoria RAM.
 - 80 GB de disco duro.

Restricciones del diseño:

- **Herramienta de modelado Visual Paradigm:**

Se empleará la herramienta CASE “Visual Paradigm” por sus múltiples beneficios en la modelación de los diferentes artefactos que se obtienen en los distintos flujos de trabajo.

- **Lenguaje de programación:**

Por el empleo del framework Symfony el lenguaje de programación a utilizar es PHP 5 el cual constituye la base del framework.

- **Entorno de desarrollo integrado (IDE):**

El software se desarrollará sobre Zend Studio.

Interfaz:

- **Interfaces de usuario:**

El sistema debe ofrecer una interfaz fácil de operar.

- **Interfaces de software:**

El sistema debe interactuar con el repositorio de objetos de aprendizaje (RHODA). Para ello se ofrecerá una interfaz de comunicación haciendo uso del estándar SQI e IMS-DRI. Dicha interfaz contará con la vía para acceder a las funcionalidades o información que forman parte de la integración.

2.5. Definición de Actores y Casos de Uso.

2.5.1. Definición de actores del sistema.

Autor	Justificación
Autor	Es la persona con permisos para la creación de los OA.
Creador de la colaboración	Es el autor con permisos para modificar una colaboración, así como proponerla a eliminar. Además podrá editar el OA y relacionar los recursos incluidos en los comentarios con los elementos de la estructura de dicho OA.

2.5.2. Definición de casos de uso.

CU-1	Crear colaboración.
Actor	Autor.
Descripción	Permite al autor crear una colaboración, especificando sus datos generales, miembros y la plantilla que se utilizará.
Referencia	R1, R2, R3, R4, R5.

CU-2	Editar colaboración.
Actor	Creador de colaboración.
Descripción	Permite al autor editar una colaboración, modificando algunos de sus datos generales, así como gestionando los miembros del equipo.
Referencia	R6, R7, R8.

CU-3	Proponer a eliminar colaboración.
Actor	Creador de colaboración.
Descripción	Permite al creador de la colaboración proponer a eliminar una colaboración.
Referencia	R6, R9.

CU-4	Asignar responsable de colaboración (extend).
Actor	Creador de colaboración.
Descripción	Permite al creador de la colaboración asignar un responsable a una colaboración.
Referencia	R10.

CU-5	Eliminar colaboración (extend).
Actor	Creador de colaboración.
Descripción	Permite al creador de la colaboración eliminar dicha colaboración.
Referencia	R11.

CU-6	Visualizar área de colaboración.
Actor	Autor.
Descripción	Permite al autor observar el área de colaboración, donde se encuentran todos los temas discutidos pertenecientes a una colaboración.
Referencia	R12.

CU-7	Configurar pizarra virtual (extend).
Actor	Autor.
Descripción	Permite al autor configurar una pizarra virtual.
Referencia	R14.

CU-8	Visualizar pizarra virtual (extend).
Actor	Autor.
Descripción	Permite al autor visualizar la pizarra virtual perteneciente a una determinada colaboración.

Referencia	R13.
------------	------

CU-9	Mostrar área de edición del OA.
Actor	Creador de colaboración.
Descripción	Permite al creador de colaboración acceder al área de edición del OA desde el área de trabajo colaborativo.
Referencia	R19.

CU-10	Crear tema de debate.
Actor	Autor.
Descripción	Permite al autor crear un tema de debate sobre el cual se aportarán ideas para la creación del OA.
Referencia	R15.

CU-11	Adicionar comentarios.
Actor	Autor.
Descripción	Permite al autor añadir un comentario en un tema de debate.
Referencia	R16.

CU-12	Adicionar recurso a un comentario.
Actor	Autor.
Descripción	Permite al autor adicionar un recurso a un comentario.
Referencia	R17.

CU-13	Relacionar recursos a la estructura de un OA.
Actor	Creador de colaboración.

Descripción	Permite al creador de colaboración relacionar un recurso con un elemento de la estructura del OA.
Referencia	R18.

CU-14	Gestionar recurso web.
Actor	Autor.
Descripción	Permite al autor crear, editar o eliminar un recurso web.
Referencia	R20, R21, R22.

CU-15	Gestionar ejercicio de orden cronológico.
Actor	Autor.
Descripción	Permite al autor crear, editar o eliminar un ejercicio de orden cronológico.
Referencia	R23, R24, R25.

CU-16	Gestionar ejercicios de buscar en el texto.
Actor	Autor.
Descripción	Permite al autor crear, editar o eliminar ejercicios de buscar en el texto.
Referencia	R26, R27, R28.

CU-17	Gestionar ejercicios de enlazar columnas.
Actor	Autor.
Descripción	Permite al autor crear, editar o eliminar ejercicios de enlazar columnas.
Referencia	R29, R30, R31.

CU-18	Gestionar cuestionario.
Actor	Autor.

Descripción	Permite al autor crear, editar o eliminar ejercicios de selección simple.
Referencia	R32, R33, R34.

CU-19	Visualizar OA almacenado en el RHODA.
Actor	Autor.
Descripción	Permite al autor visualizar un OA almacenado en el repositorio de objetos de aprendizaje.
Referencia	R35.

CU-20	Editar OA almacenado en el RHODA.
Actor	Autor.
Descripción	Permite al autor editar un OA almacenado en el repositorio de objetos de aprendizaje, para ello se realiza una copia del mismo en el área de objetos en edición del mismo.
Referencia	R36.

2.5.3. Diagrama de Casos de Uso del Sistema.

El diagrama de casos de uso del sistema brinda las funcionalidades que el sistema debe ofrecer para aportar un resultado de valor para sus actores.

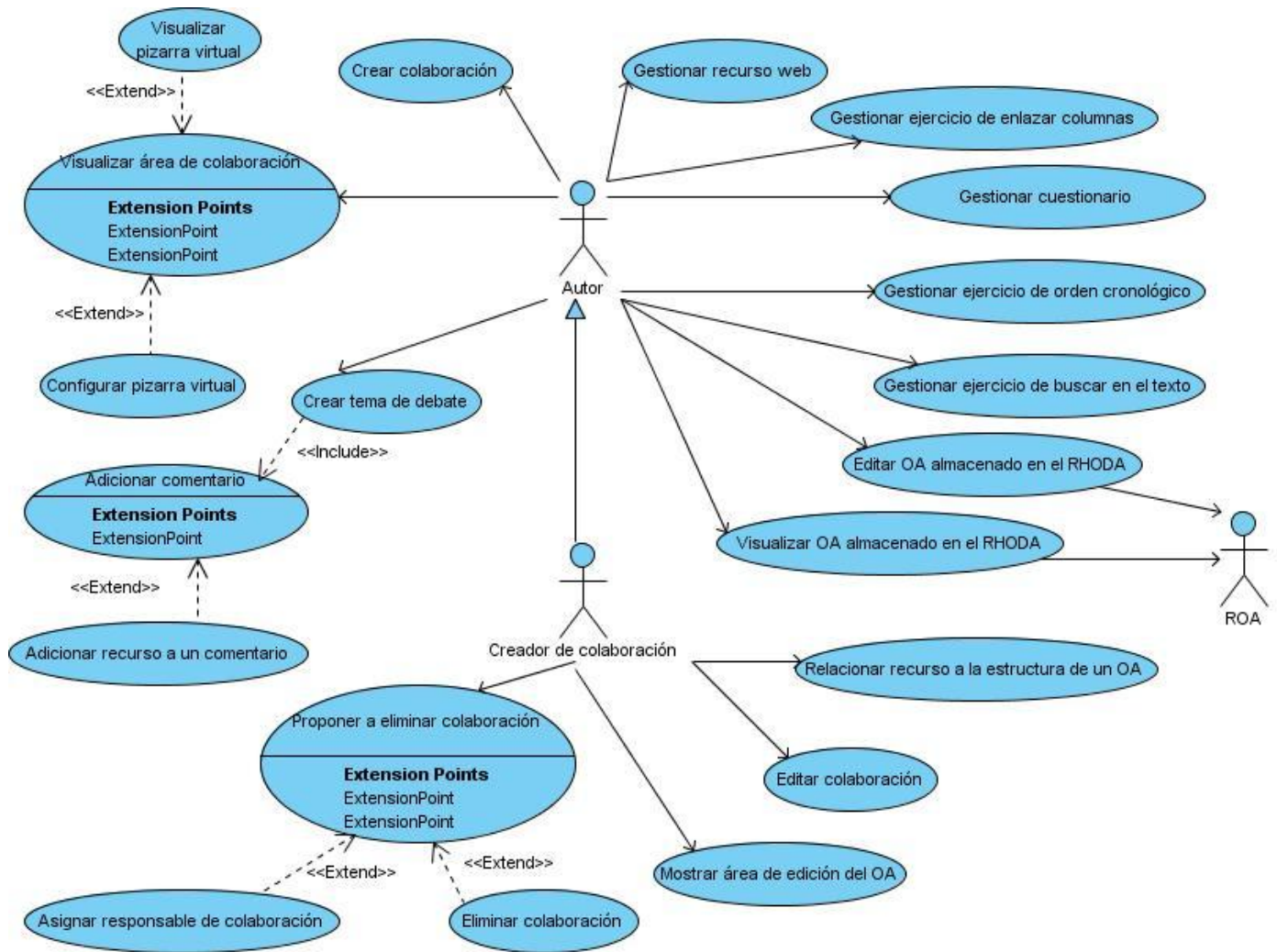


Figura 2 Diagrama de casos de uso del sistema.

2.6. Descripciones textuales de los casos de uso del sistema.

Caso de uso	Crear colaboración.
Actores	Autor.
Resumen	El caso de uso comienza cuando el autor decide crear una colaboración y finaliza cuando la misma es creada.

Referencias	R1, R2, R3, R4, R5.
Precondiciones	El autor debe estar autenticado en el sistema.
Poscondiciones	Se crea la colaboración.
Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El autor selecciona la opción “ <i>Crear OA colaborativamente</i> ”.	2. El sistema muestra una pantalla con los datos generales de la colaboración (<i>Nombre, Objetivos, Descripción</i>), así como un espacio para definir el equipo que participará en la misma (una lista con todos los autores del sistema, para que el autor seleccione los que formarán parte del equipo de colaboración) y la plantilla que se utilizará (cuadro de selección de la plantilla a utilizar para la creación del OA). Además de los botones Crear y Cancelar.
3. El autor introduce el nombre, los objetivos y la descripción de la colaboración. Además selecciona los autores que formarán parte del equipo, arrastrando del listado de autores del sistema al listado de autores de la colaboración. Selecciona también la plantilla que desea utilizar en la colaboración.	
4. El autor selecciona la opción Crear.	5. El sistema muestra el siguiente mensaje “Colaboración creada” y finaliza el caso de uso.
Flujo Alternativo	
Acción del actor	Respuesta del sistema

4. El autor selecciona la opción “ <i>Cancelar</i> ”.	5. El sistema limpia todos los campos.
Prioridad	Crítico.

Caso de uso	Proponer a eliminar colaboración.
Actores	Creador de colaboración.
Resumen	El caso de uso comienza cuando el creador de una colaboración desea eliminar una colaboración y concluye cuando el sistema elimina la colaboración o es reasignado el privilegio de colaboración.
Referencia	R6, R9.
Precondiciones	El autor debe haber creado al menos una colaboración.
Poscondiciones	La colaboración es eliminada o reasignado el privilegio de propietario de colaboración.

Flujo Normal de Eventos

Acción del Actor	Respuesta del Sistema
1. El autor accede a la pantalla “ <i>Mis trabajos colaborativos</i> ”.	2. El sistema muestra todas las colaboraciones en las que está registrado como miembro o creador, brindando la opción “Proponer a eliminar” solo para las colaboraciones en las que es el creador.
3. El autor selecciona la opción “ <i>Proponer a eliminar</i> ”.	4. El sistema envía un mensaje a cada uno de los miembros de la colaboración explicando que la colaboración <i>nombre_de_la_colaboración</i> será eliminada, que es necesario que acceda a “ <i>Mis trabajos colaborativos</i> ” y vote si está de acuerdo o no, en que sea realizada esa acción.” Y finaliza el caso de uso.

Prioridad	Crítico
------------------	---------

Caso de uso	Asignar responsable de colaboración (extend).
Actores	Creador de colaboración.

Resumen	El caso de uso comienza cuando el creador de la colaboración desea asignar un responsable de colaboración y finaliza cuando es asignado el responsable.	
Referencias	R10.	
Precondiciones	El autor debe ser el creador de la colaboración.	
Poscondiciones	Se asigna un responsable de colaboración.	
Flujo Normal de Eventos		
Acción del actor	Respuesta del sistema	
	1. El sistema muestra en el espacio que ocupa la colaboración que será eliminada en la pantalla “ <i>Mis trabajos colaborativos</i> ” de cada miembro de la colaboración el siguiente mensaje: “ <i>¿Está de acuerdo con que sea eliminada esta colaboración?</i> ” y los botones Sí y No.	
2. Al menos un miembro del equipo selecciona el botón No.	3. El sistema brinda una interfaz para que el creador de la colaboración asigne como propietario de la colaboración a uno de los miembros que no ha estado de acuerdo con que se elimine la colaboración. Y finaliza el caso de uso.	
Prioridad	Crítico.	

Caso de uso	Eliminar colaboración (extend).
Actores	Creador de colaboración.
Resumen	El caso de uso comienza cuando el creador de colaboración desea eliminar una colaboración y finaliza cuando la misma es eliminada.
Referencias	R11.
Precondiciones	El autor debe estar autenticado en el sistema.
Poscondiciones	Se elimina la colaboración.
Flujo Normal de Eventos	

Acción del actor	Respuesta del sistema
	1. El sistema muestra en el espacio que ocupa la colaboración que será eliminada en la pantalla “Mis colaboraciones” de cada miembro de la colaboración el siguiente mensaje: “¿Está de acuerdo con que sea eliminada esta colaboración?” y los botones Sí y No.
2. Todos los miembros seleccionan la opción Sí.	3. El sistema mantendrá dicha colaboración por un espacio de 72 horas, para que los miembros puedan recuperar alguna información relacionada con la colaboración, luego será eliminada de manera definitiva. Y finaliza el caso de uso.
Prioridad	Crítico.

Caso de uso	Visualizar área de colaboración.
Actores	Autor.
Resumen	El caso de uso comienza cuando el autor desea visualizar el área de colaboración y finaliza cuando la misma es mostrada.
Referencias	R11.
Precondiciones	El autor debe estar autenticado en el sistema.
Poscondiciones	Se visualiza el área de colaboración.

Flujo Normal de Eventos

Acción del actor	Respuesta del sistema
1. El autor selecciona la opción “ <i>Mis trabajos colaborativos</i> ”.	2. El sistema muestra una lista de los trabajos colaborativos en los que el autor participa.
3. El autor selecciona el trabajo del que desea visualizar el área de colaboración.	4. El sistema muestra todos los temas discutidos pertenecientes a la colaboración contando cada uno

	<p>con las siguientes opciones:</p> <ul style="list-style-type: none"> • <i>Pizarra virtual</i>, ver caso de uso “<i>Visualizar pizarra virtual</i>”. • <i>Configurar pizarra virtual</i>, ver caso de uso “<i>Configurar pizarra virtual</i>”. • <i>Ir a edición</i>, ver caso de uso “<i>Mostrar área de edición del OA</i>”. • <i>Visualizar OA</i>, ir a sección 1.
Sección 1 “Visualizar OA”	
1. El autor selecciona la opción “ <i>Visualizar OA</i> ”.	2. El sistema muestra la estructura del OA con sus contenidos. Y finaliza la sección.
Prioridad	Crítico.

Caso de uso	Crear tema de debate.
Actores	Autor.
Resumen	El caso de uso inicia cuando el autor desea crear un tema de debate y finaliza cuando el mismo es creado.
Referencias	R15.
Precondiciones	El autor debe estar autenticado en el sistema.
Poscondiciones	Es creado un tema de debate.

Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El autor selecciona la opción “ <i>Nuevo tema</i> ”.	2. El sistema solicita al autor los siguientes datos: (*) <i>Título, Comentario, Subir recurso</i> . Además muestra el botón “ <i>Aportar</i> ”.
3. El autor inserta los datos solicitados.	
4. El autor accede al botón “ <i>Aportar</i> ”.	5. El sistema verifica que se hayan escrito todos los

	datos (*) y finaliza el caso de uso
Flujo Alternativo	
Acción del Actor	Respuesta del sistema
3. El autor no llenó todos los datos (*).	4. El sistema mostrará el siguiente mensaje: <i>“Debe llenar todos los campos obligatorios (*)”</i> .
Prioridad	Secundario.

Caso de uso	Adicionar comentario.
Actores	Autor.
Resumen	El caso de uso inicia cuando el autor decide añadir un comentario a un tema de debate y finaliza cuando se realiza la acción solicitada.
Referencias	R16.
Precondiciones	El autor debe haber seleccionado un tema de debate.
Poscondiciones	Se añade un comentario a un tema de debate.

Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El autor selecciona el tema de debate sobre el cual incluir un comentario.	2. El sistema muestra una interfaz con todos los comentarios existentes sobre el tema seleccionado. Muestra un cuadro de texto en el que el autor podrá insertar la opinión acerca de la creación del OA. Además del botón: <ul style="list-style-type: none"> • <i>Aportar.</i>
3. El autor añade el comentario que desea incorporarle al tema de debate y da clic en el botón <i>“Aportar”</i> .	4. El sistema guarda el comentario que el autor ha creado. Y finaliza el caso de uso.
Prioridad	Crítico.

Caso de uso	Relacionar un recurso a la estructura de un OA.	
Actores	Creador de colaboración.	
Resumen	El caso de uso comienza cuando el creador de la colaboración desea relacionar un recurso con un elemento de la estructura del OA y finaliza cuando es realizada la acción solicitada por el autor.	
Referencias	R18.	
Precondiciones	El autor debe estar autenticado en el sistema. Sólo podrá ser incorporado un recurso por el creador de la colaboración.	
Poscondiciones	Es relacionado un recurso con un elemento de la estructura del OA.	
Flujo Normal de Eventos		
	Acción del actor	Respuesta del sistema
	1. El creador de la colaboración selecciona el recurso a relacionar con un elemento de la estructura del OA y selecciona la opción "Guardar".	2. El sistema almacena dicho recurso en una carpeta de local de ficheros.
	3. El autor accede a la vista de árbol de la estructura e incorpora sobre un elemento especificado el recurso deseado. Luego da clic en Guardar.	4. El sistema relaciona el recurso seleccionado con el elemento de la estructura del OA que se determinó. Y finaliza el caso de uso.
Prioridad	Crítico	

2.7. Conclusiones.

En este capítulo se mostraron las principales clases del dominio para una mayor comprensión del negocio, se justificó la selección de los actores que interactúa con el sistema, además de analizar las características y funciones fundamentales del sistema para la creación colaborativa de OA y su comunicación con el RHODA, 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 III: Análisis y diseño del sistema.

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, además de realizar las descripciones de estas 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 solo se interesa por los requisitos funcionales. Representa las cosas del mundo real, no de la implementación automatizada de estas cosas.

3.2.1 Diagrama de clases de análisis.

Los diagramas de clases de análisis representan un modelo conceptual temprano que describe las características y comportamiento comunes de un conjunto de cosas que existen en el sistema. Se expresa que es conceptual pues pospone todos los elementos de diseño ya que no considera posibles tecnologías a emplear en el desarrollo del software; constituyen un prototipo de las futuras clases que darán vida al mismo. Las clases del análisis están siempre identificadas con uno de los tres estereotipos existentes, los cuales son:

- **Interfaz:** Se encargan de la modelación de toda la interacción que puede existir entre los actores y el sistema; constituyen las fronteras del mismo.

- **Control:** Representan la coordinación, secuenciación, transacciones y a veces la lógica del negocio; se emplean a menudo para encapsular el control referido a un CU.
- **Entidad:** representa la información de larga duración y a menudo persistente que se maneja en el sistema.

Resumiendo, a continuación se muestran los diagramas de clases de análisis correspondientes a los casos de usos descritos anteriormente.

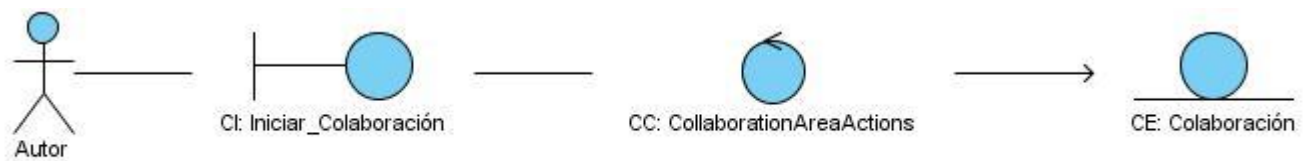


Figura 3 Diagrama de clases de análisis "Crear colaboración".

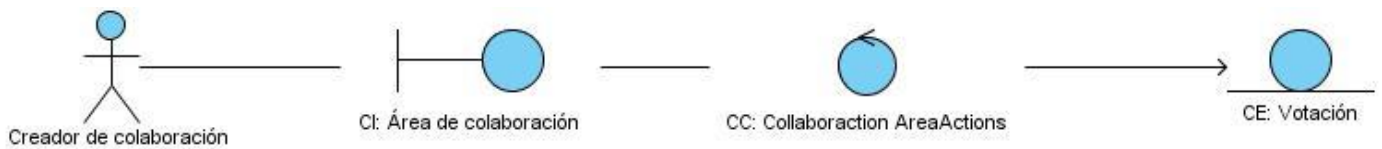


Figura 4 Diagrama de clases de análisis "Proponer a eliminar colaboración".



Figura 5 Diagrama de clases de análisis "Asignar responsable de colaboración".



Figura 6 Diagrama de clases de análisis "Eliminar colaboración".

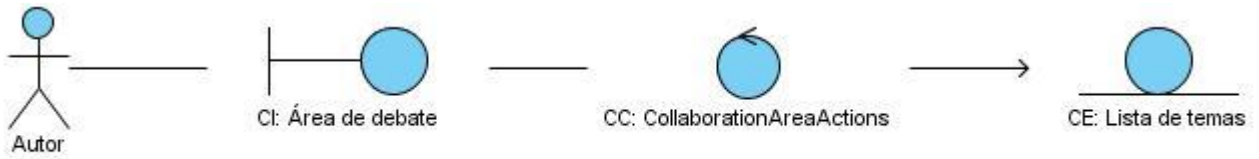


Figura 7 Diagrama de clases de análisis "Visualizar área de colaboración".



Figura 8 Diagrama de clases de análisis "Relacionar recurso a la estructura de un OA".



Figura 9 Diagrama de clases de análisis "Crear tema de debate".

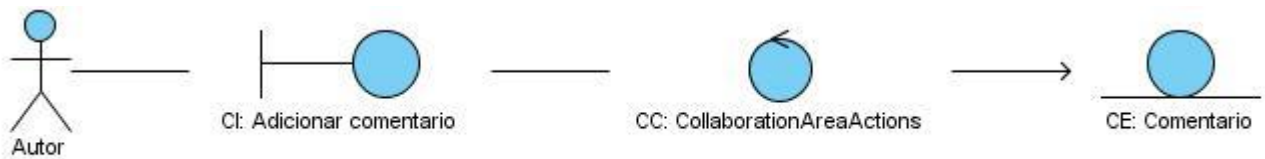


Figura 10 Diagrama de clases de análisis "Adicionar comentarios".

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.

3.3.1 Diagrama de clases del diseño.

En la fase de elaboración de la metodología RUP se comienza con un análisis de los elementos significativos de la arquitectura como parte de la primera iteración de elaboración y en las siguientes iteraciones se refina la arquitectura hasta diseñar todos sus elementos. El diseño es el centro de atención al final de la fase de elaboración y el comienzo de las iteraciones de construcción. Esto contribuye a una arquitectura estable, sólida y crea un plano al modelo de implementación.

En el diseño se confeccionan los diagramas de clases del diseño. Los elementos básicos que se pueden encontrar en este diagrama son clases y relaciones entre estas. A continuación se representan los diagramas de clases de diseño correspondiente a los casos de usos descritos anteriormente.

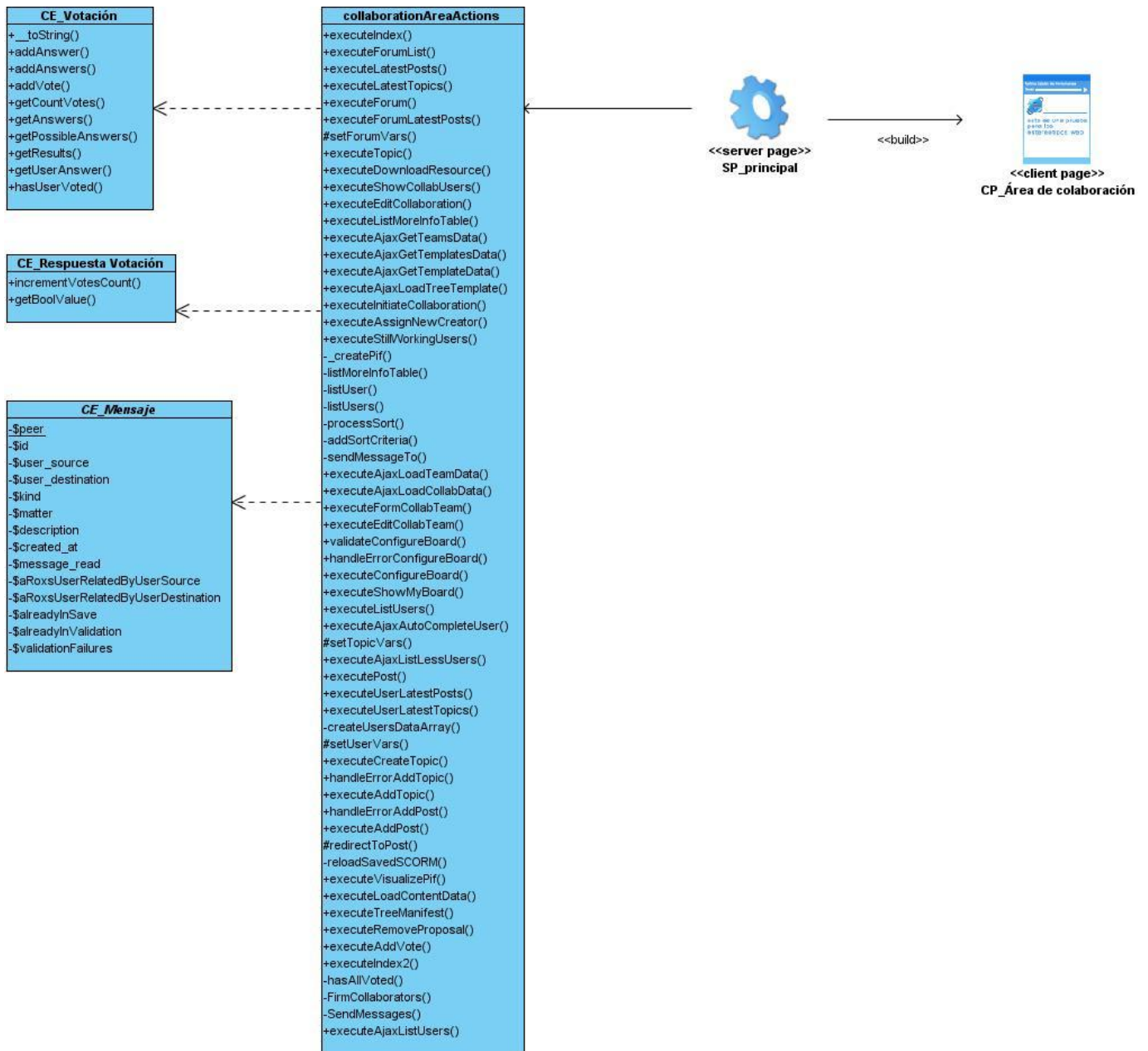


Figura 12 Diagrama de clases de diseño "Proponer a eliminar colaboración".

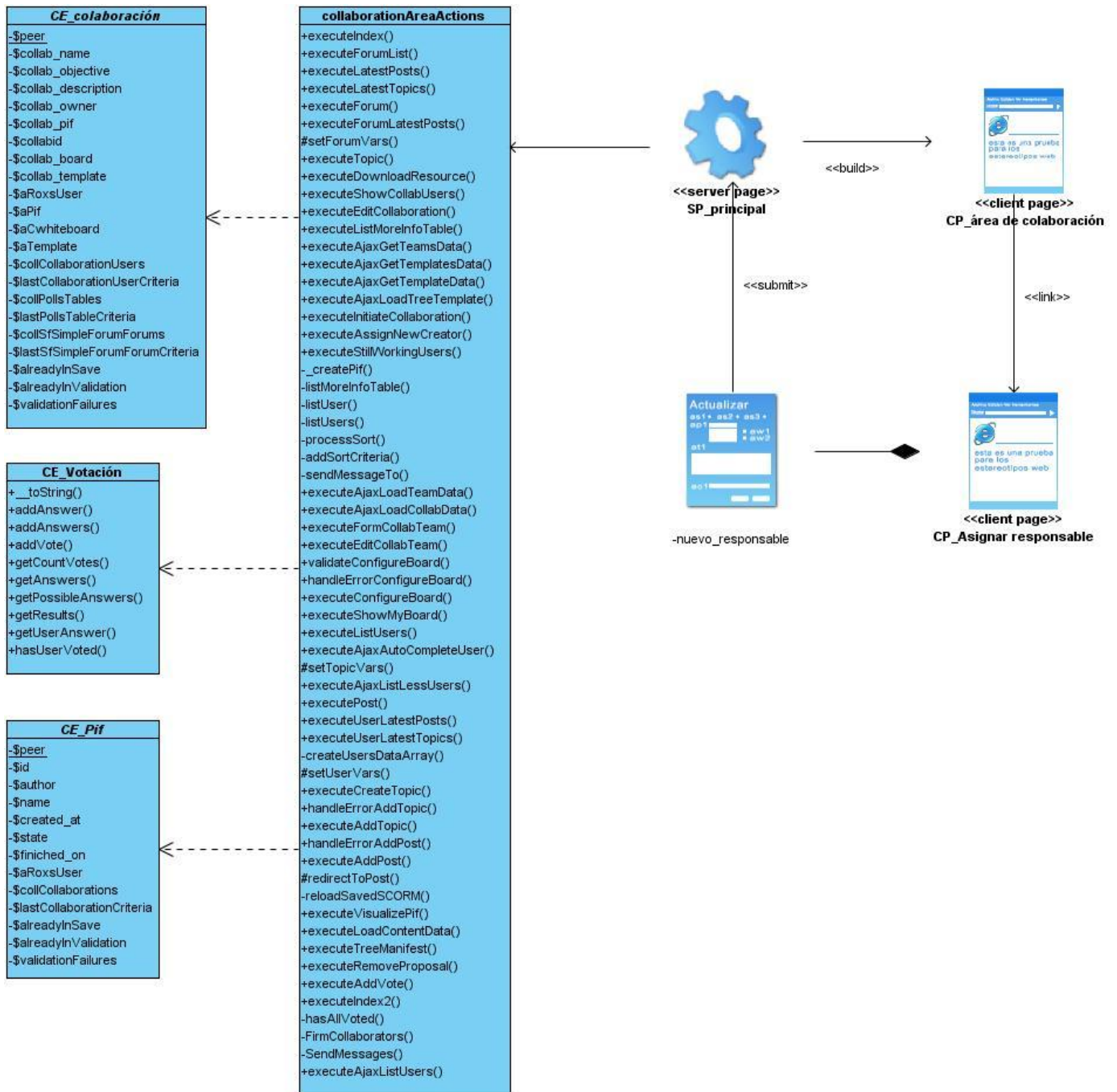


Figura 13 Diagrama de clases de diseño "Asignar responsable de colaboración".

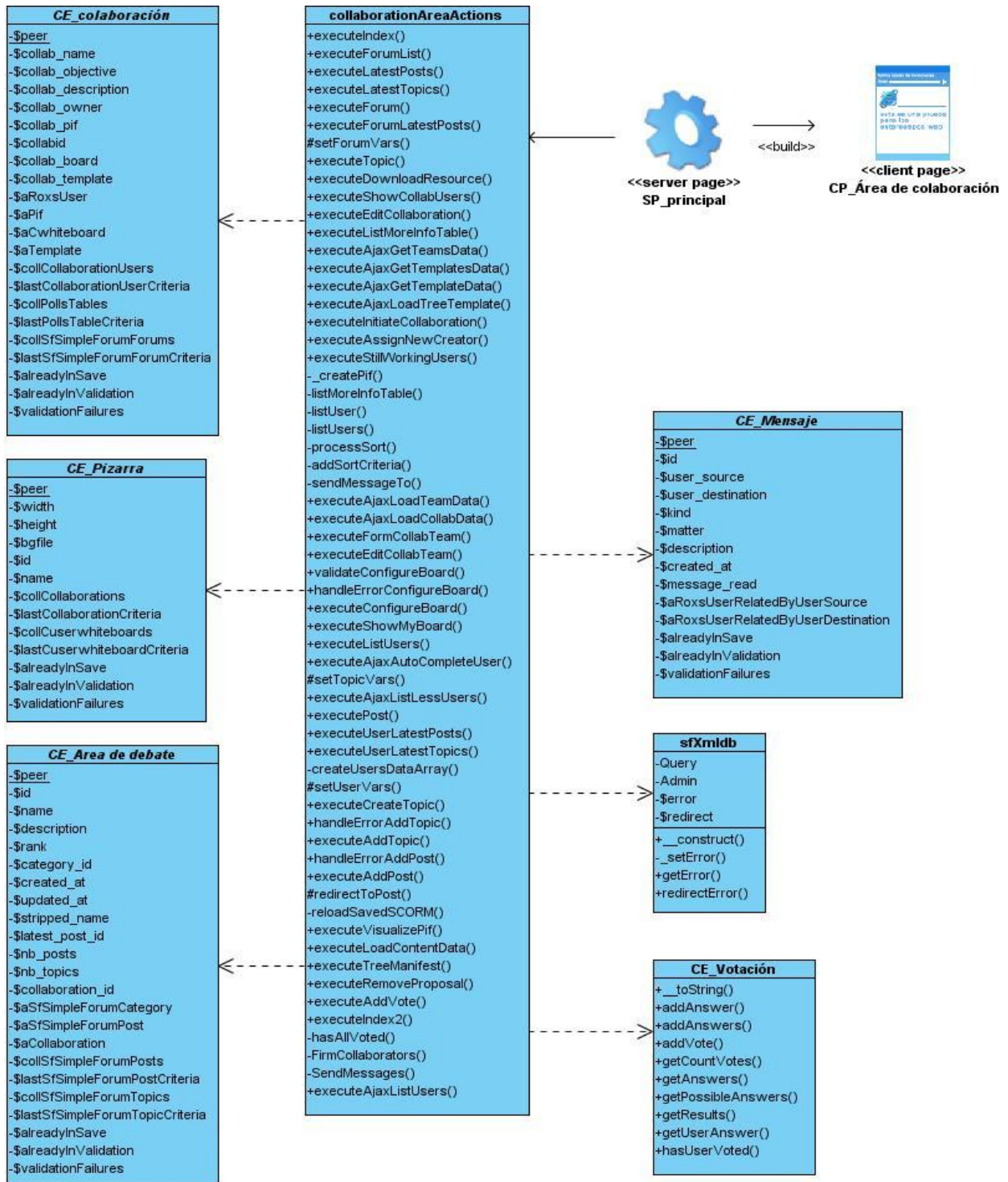


Figura 14 Diagrama de clases de diseño "Eliminar colaboración".

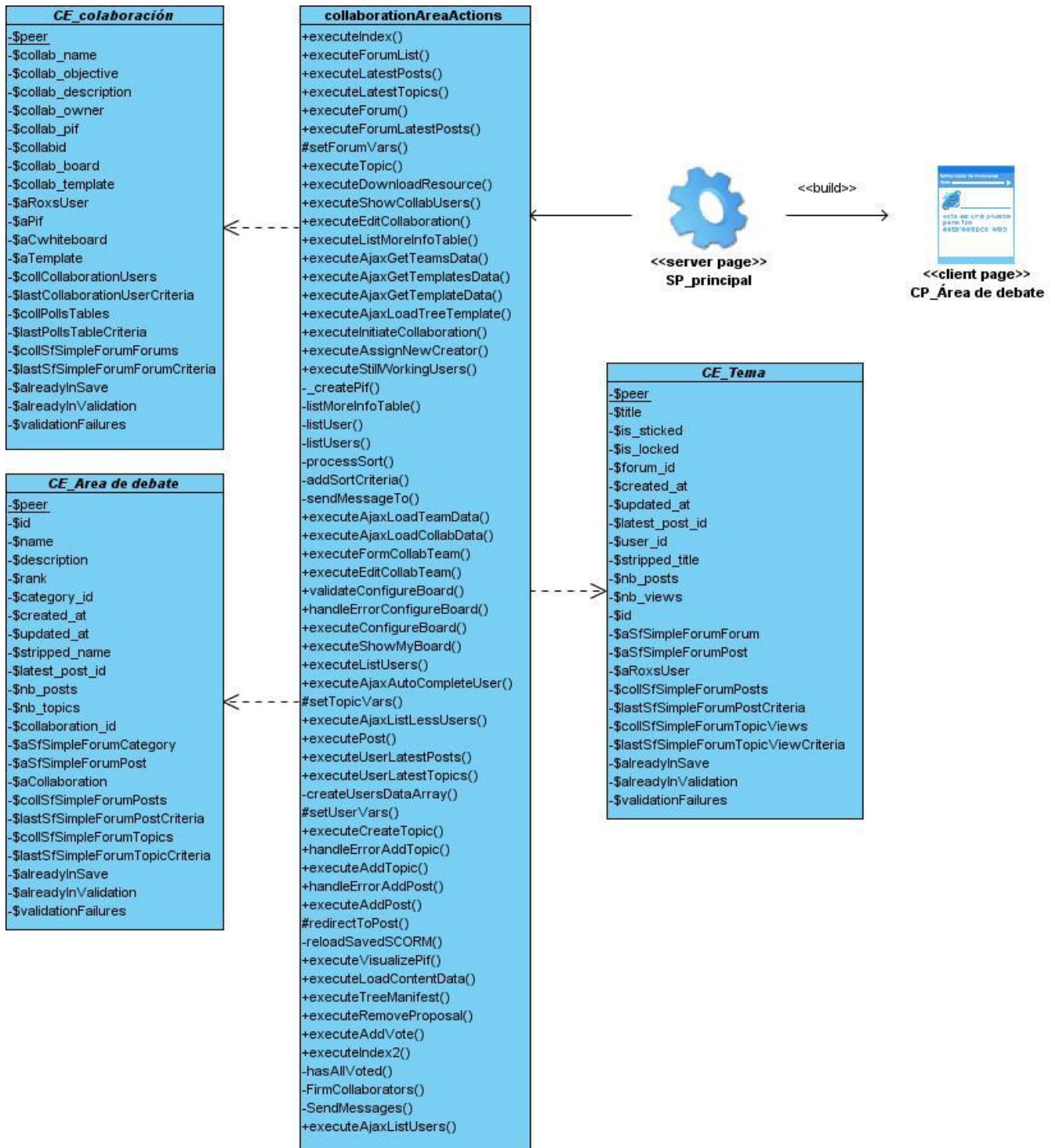


Figura 15 Diagrama de clases de diseño "Visualizar área de colaboración".

Capítulo IV: Construcción de la solución propuesta.

4.1 Introducción.

El flujo de trabajo de Implementación tiene como objetivo definir la organización del código teniendo en cuenta los subsistemas de implementación organizados por capas, la implementación de los elementos de diseño en términos de ficheros fuentes, binarios, ejecutables y para poder integrar los diferentes componentes de desarrolladores o equipos y generar un ejecutable entregable o producto final.

El presente capítulo se centra en el flujo de trabajo Implementación, el cual está fuertemente regido por el flujo de Análisis y Diseño. Se muestran el diagrama de despliegue y el de componentes.

4.2 Diagrama de despliegue.

El diagrama de despliegue muestra la configuración de los nodos de procesamiento en tiempo de ejecución, los links de comunicación entre ellos y las instancias de los componentes y objetos que residen en ellos. El propósito del diagrama de despliegue es capturar la configuración de los elementos de procesamiento y las conexiones entre estos elementos en el sistema. El diagrama consiste en uno o más nodos, dispositivos, conectores entre nodos y conectores entre nodos y dispositivos.

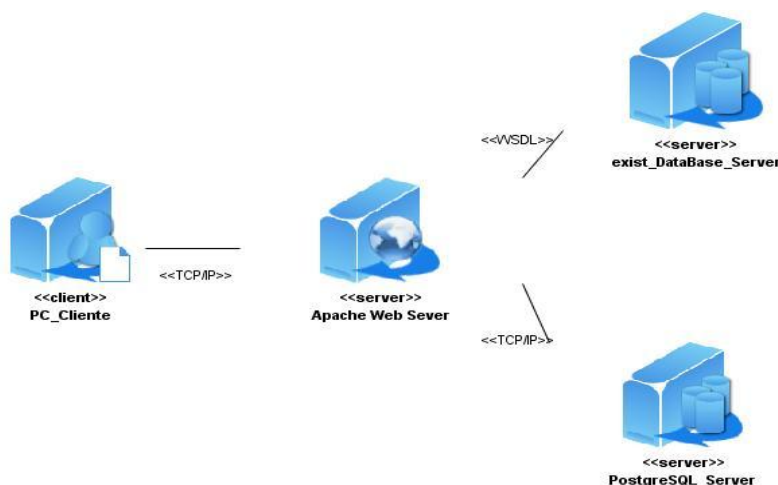


Figura 16 Diagrama de despliegue.

El diagrama de despliegue está compuesto por los nodos necesarios para acceder a la aplicación. El usuario que desee acceder al sistema, lo podrá hacer mediante una computadora que se conectará a través del protocolo TCP/IP a un servidor web Apache donde se montarán las funcionalidades implementadas, el mismo estará conectado a un servidor de base de datos PostgreSQL usando el protocolo antes mencionado y a otro servidor eXist-database donde se guardará el xml del OA en su estructura original, este último servidor se conectará a través del protocolo WSDL al servidor web Apache.

4.3 Diagrama de componentes.

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistema de implementación y mostrar las relaciones entre los elementos de implementación. El uso más importante de estos diagramas es mostrar la estructura de alto nivel del modelo de implementación, especificando:

- Los subsistemas de implementación y sus dependencias al importar un código.
- Organizar los subsistemas de implementación en capas.

Son utilizados para:

- Mostrar las dependencias de compilación de los ficheros de código.
- Mostrar relaciones de derivación entre ficheros de código fuente y ficheros que son resultados de compilación.

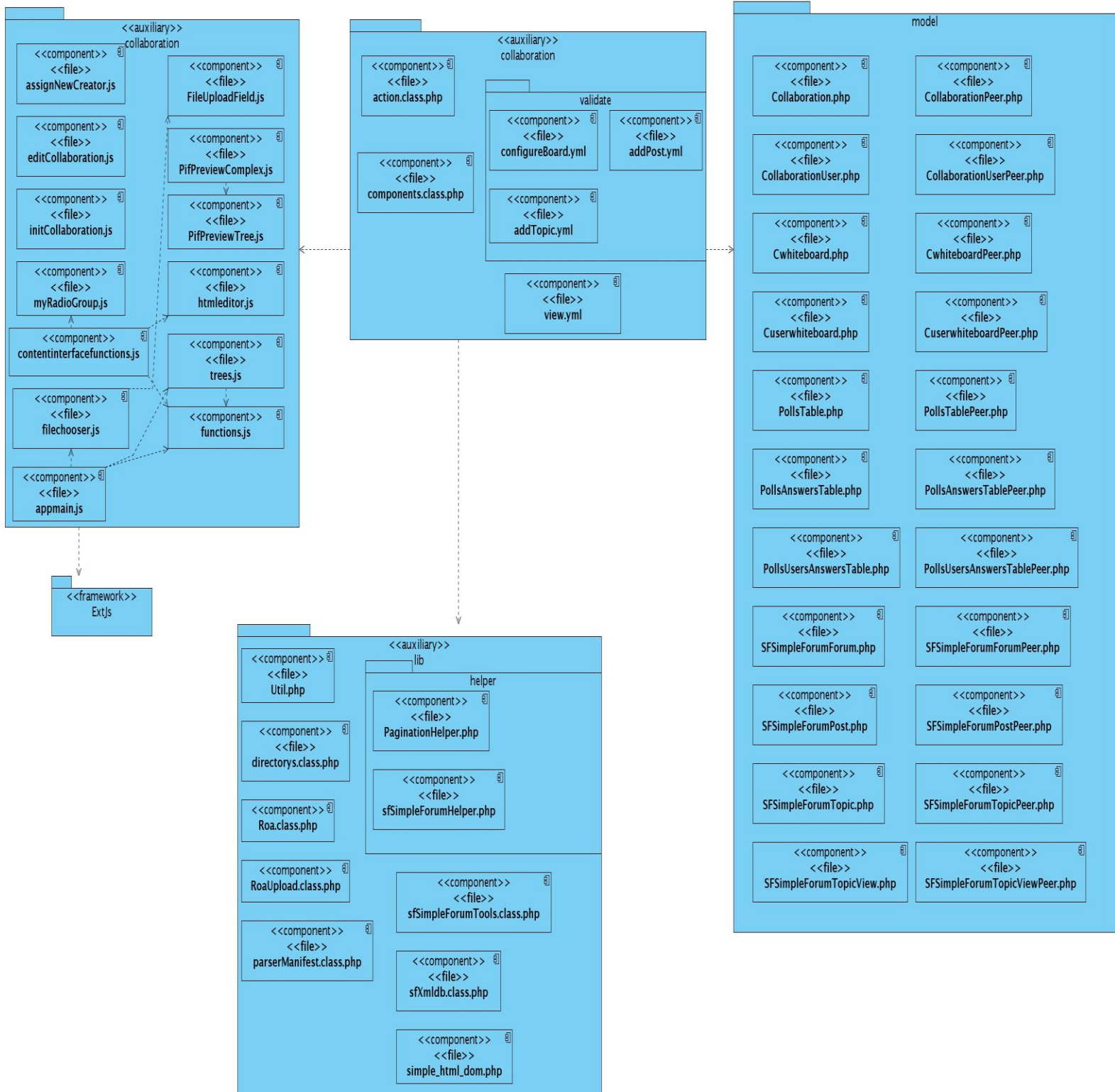


Figura 17 Diagrama de componente.

El diagrama mostrado anteriormente muestra la estructura física presente en la solución obtenida. Esta estructura sigue las orientaciones del framework empleado para su acoplamiento y correcto funcionamiento. Existe un paquete “colaboración” en el cual se encuentran los ficheros principales que permiten que la solución se comporte como un módulo.

Este paquete principal se apoya en otros tres. Uno de ellos es el que contiene las clases que modelan los datos persistentes del sistema. Otro contiene los ficheros auxiliares en los que se apoya el paquete principal para otorgar la solución. Por último otro que contiene ficheros que son empleados para las interfaces y para el manejo de datos en la parte del cliente el cual se apoya en un framework de javascript llamado ExtJs.

4.4 Conclusiones.

En este capítulo se ha desarrollado el flujo de trabajo Implementación en el proceso de desarrollo del módulo propuesto, modelándose el diagrama de despliegue el cual es una configuración de nodos de procesamiento en tiempo de ejecución y el diagrama de componentes que no es más que una organización y las dependencias entre un conjunto de componentes.

Conclusiones generales

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

- El análisis y diseño de las funcionalidades expuestas propició una vía de solución que guió el proceso de desarrollo del módulo para la herramienta de autor web CRODA.
- Con la realización de este trabajo en la UCI se logró desarrollar un módulo de la herramienta de autor web CRODA que permite la creación colaborativa de OA aprovechando las experiencias de cada uno de los autores que participan en la elaboración del mismo.
- Se logró la interoperabilidad entre la herramienta de autor web CRODA y RHODA.

Recomendaciones

Para dar continuidad al presente trabajo se le recomienda al Departamento de Producción "*Herramientas Educativas*" del Centro FORTES:

- Realizar las actividades propuestas por la metodología de desarrollo RUP en el flujo de trabajo Pruebas con vista a garantizar la calidad del producto.
- Confeccionar un Manual de Usuario para mejor entendimiento del módulo implementado.

Bibliografía

Anónimo. Visual Paradigm. [Online] <http://www.visual-paradigm.com/product/vpuml/>.

AULAGLOBAL. 2005 Estándares-SCORM, 2005. [En línea]
<http://www.aulaglobal.net.ve/observatorio/articles.php?lng=es&pg=157>.

Akelos. 2006. Akelos Framework. [Online] 2010. <http://www.akelos.org/>.

ATICA_SOCRATES. 2007. *E-learning*. 2007.

Canarie. 2001. White Paper for a Learning Object Repository. [Online] 2001.
http://oknl.edu.gov.on.ca/eng/pdf/1_3_13_1.pdf.

CARRODEGUAS, Y. 2008. *Análisis y Diseño de una herramienta de autor Web Interoperable*. 2008.

Cibernetia. 2009. Servidores Web. [En línea] 2009.
http://www.cibernetia.com/tesis_es/PEDAGOGIA/TEORIA_Y_METODOS_EDUCATIVOS/1.

Codeigniter. 2008. Codeigniter. [Online] 2010. <http://codeigniter.com/>.

Daniel, G. 2004. Learning Object Repositories. *WWWtools for Education*. [Online] 2004.
<http://magazines.fasfind.com/wwwtools/m/1030.cfm>.

de la Torre, Aníbal. 2004. Análisis de los portales en Internet de Centros educativos. [En línea] 2004.
http://www.adelat.org/media/docum/nuke_publico/01impr.pdf.

eXist. 2008. eXist. [Online] 2010. <http://exist.sourceforge.net/>.

Fernández, B, Moreno, B y Sierra, J. 2007. *Uso de estándares aplicados a las TICs en educación*. 2007.

Fernández, Gerardo. 2002. Introducción a Extreme Programming. 2002.

García, F. J. 2000. *Modelo de Reutilización Soportado por Estructuras Complejas*. Salamanca: s.n., 2000.

García, Lorenzo. 2005. Objetos de Aprendizajes. Características y repositorios. [Online] 2005. [Cited: 02 14, 2010.] <http://ares.unimet.edu.ve/programacion/cursos/tobj/articObj2.pdf>.

González-Barahona, Jesús M. 2002. Edukalibre: una herramienta para la creación colaborativa. [En línea] 2002. http://www.educacionenvalores.org/IMG/pdf/edukalibre_ctsl.pdf.

González, D. 2005. Los repositorios de objetos de aprendizaje como soporte para los entornos e-learning. [Online] 2005. http://www.biblioweb.dgsca.unam.mx/libros/repositorios/objetos_aprendizaje.htm.

Gutiérrez, Sarah. FDD: Feature Driven Development Desarrollo Basado en Funcionalidades.

IBM. 2002. IBM. [Online] 2002. <http://www-142.ibm.com/software/products/es/es/enterprise/>.

IEEE. 2001. [En línea] 2001. [Citado el: 6 de 12 de 2009.] <http://ltsc.ieee.org/wg12/index.html>.

IEEE. 2002. Draft Standard for Learning Object Metadata. [Online] 2002. http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf.

JORUM+Project. 2004. The JISC Online Repository for [learning and teaching] Materials. [En línea] 2004. http://www.jorum.ac.uk/docs/Vol1_Fin.pdf.

Latorre, Miguel, et al. 2009. Objetos de aprendizaje: Buenas prácticas y trabajo colaborativo. [Online] 2009. <http://www.ieec.uned.es/Investigacion/RedOber/archivos/ccita-v3-mlatorre.pdf>.

León, E. 2007. Visual Paradigm, una herramienta de lo más útil. [En línea] 2007. <http://slion2000.blogspot.com/2007/04/visual-paradigm-una-herramienta-de-lo.html>.

López, Clara. 2005. *Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning*. España: s.n., 2005.

López, Martín Luis. 2006. Microsoft Solutions Framework. 2006.

Lowerison, G; Gallant, G; Boyd, G. 2003. Learning Objects in Distance Education: Addressing issues of Quality, Learner Control and Accessibility. [En línea] 2003. http://www.cade-aced2003.ca/conference_proceedings/Gallant.pdf.

Majón, B F. 2006. Especificaciones y estándares en e-learning Revista de Tecnologías de la Información y Comunicación Educativas. [Online] 2006.

Martínez, Susana Naharro. 2006. Los objetos de aprendizaje como recurso de calidad para la docencia: criterios de validación de objetos en la Universidad Politécnica de Valencia. [Online] 2006.

Mentores.net. [En línea] http://www.mentores.net/articulos/intro_microsoft_sol_frame.htm.

Microsoft. 2000. Microsoft.com. [Online] 2010.

<http://www.microsoft.com/spain/windowsserver2003/technologies/webapp/iis.msp>.

Ministerio de Educación y Ciencia. 2008. Educación. [Online] 2010.

<http://ares.cnice.mec.es/informes/16/contenido/13.htm>.

Nguyen, Nhu Van and Massart, David. 2007. Binding the Simple Query Interface. [Online] Nhu Van Nguyen and David Massart, 2007. <http://fire.eun.org/lode2007/lode05.pdf>.

OKI. 2002. Open Knowledge Initiative. [Online] 2010. <http://www.okiproject.org/>.

ORACLE. 2003. Oracle. [Online] 2003. <http://www.oracle.com>.

OSSANDÓN NÚÑEZ, YANKO. 2005. *OBJETOS DE APRENDIZAJE: Un recurso pedagógico para e-learning*. Chile: Arica, 2005.

R. Bentley, W. Appelt, U. Busbach, E. Hinrichs, D. Kerr, K. Sikkell, J. Trevor, G. Woetzel. 1997. Basic Support for Cooperative Work on the World Wide Web. [Online] 1997.

<http://jonathantrevor.net/trevor/papers/IJHCS.pdf>.

Red TTnet. 2005. *La formación sin distancia. Estudio realizado por el Grupo de Trabajo de e-learning de la*

red TNet. 2005.

Rengarajan, R. 2001. LCMS and LMS: Taking Advantage of Tight Integration. [Online] 2001. http://www.e-learn.cz/soubory/lcms_and_lms.pdf.

Romo, Jesús. 2002. AUTORE: herramienta de autor para la generación de Objetos de Aprendizaje. [En línea] 2002. <http://www.uoc.edu/symposia/spdece05/pdf/ID13.pdf>.

Sánchez, E. 2004. *Microsoft Solutions Framework*. 2004.

SCORM. 2008. SCORM. [Online] 2010. <http://www.scorm.com/scorm-explained/technical-scorm/>.

Sicilia, Miguel Angel. 2003. *On the Concepts of Usability and Reusability of Learning Objects*. 2003.

Symfony. 2003. Symfony. [Online] 2010. <http://www.symfony.es/>.

Wiley, David A. 2000. *Connecting learning objects to instructional design theory: a definition, a metaphor, and a taxonomy*. s.l: <http://reusability.org/read/chapters/wiley.doc>, 2000.

Zend. 2001. Zend framework. [Online] 2010. <http://framework.zend.com>

Anexos

Anexo 1 Descripción textual CU_ Editar colaboración.

Caso de uso	Editar colaboración.	
Actores	Creador de colaboración.	
Resumen	El caso de uso comienza cuando el autor (creador de una colaboración) desea editar una colaboración y concluye cuando el sistema actualiza los datos de la colaboración.	
Referencias	R6, R7, R8.	
Precondiciones	El autor debe haber creado al menos una colaboración.	
Poscondiciones	Los datos de la colaboración son actualizados.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El autor accede a la interfaz “ <i>Mis trabajos colaborativos</i> ”.	2. El sistema muestra todas las colaboraciones en las que está registrado como miembro o creador, brindando la opción “Editar” solo para las colaboraciones en las que es el creador.	
3. El autor selecciona la opción de “Editar” una colaboración.	4. El sistema muestra la interfaz “Datos de la colaboración”, en la cual los campos “Nombre de la colaboración” y la sección “Plantilla” no son editables. Ofrece la opción Modificar y Cancelar.	
5. El autor modifica algunos o todos los elementos siguientes: Objetivos de la colaboración. Descripción de la colaboración. Miembros del equipo.		
6. El autor da clic en Modificar.	7. El sistema actualiza los datos de la colaboración	

	editada y finaliza el caso de uso.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
6. El autor da clic en Cancelar.	7. El sistema limpia todos los campos.
Prioridad	Secundario.

Anexo 2 Descripción textual CU_ Visualizar pizarra virtual (extend).

Caso de uso	Visualizar pizarra virtual (extend).	
Actores	Autor.	
Resumen	El caso de uso comienza cuando el autor desea visualizar la pizarra virtual y finaliza cuando la misma es visualizada.	
Referencias	R13.	
Precondiciones	El autor debe estar autenticado en el sistema.	
Poscondiciones	Se visualiza la pizarra virtual.	
Flujo Normal de Eventos		
Acción del actor	Respuesta del sistema	
1. El autor selecciona la opción " <i>Pizarra virtual</i> ".	2. El sistema muestra una pizarra virtual donde los autores podrán dejar anotaciones acerca de la creación del OA. Y finaliza el caso de uso.	
Prioridad	Opcional.	

Anexo 3 Descripción textual CU_ Configurar pizarra virtual (extend).

Caso de uso	Configurar pizarra virtual (extend).	
Actores	Autor.	
Resumen	El caso de uso comienza cuando el autor desea configurar la pizarra virtual y finaliza cuando esta es configurada.	
Referencias	R14.	

Precondiciones	El autor debe estar autenticado en el sistema.	
Poscondiciones	Se configura la pizarra virtual.	
Flujo Normal de Eventos		
Acción del actor		Respuesta del sistema
1. El autor selecciona la opción “ <i>Configurar Pizarra Virtual</i> ”.		2. El sistema muestra los siguientes campos para configurar la pizarra: <i>ancho de la pizarra, alto de la pizarra, un botón de selección para escoger si se utiliza imagen de fondo o no y la localización de la misma.</i>
3. El autor proporciona los datos solicitados por el sistema.		4. El sistema ofrece dos opciones: <ul style="list-style-type: none"> • <i>Guardar.</i> • <i>Entrar.</i>
5. El autor selecciona la opción “ <i>Guardar</i> ”.		6. El sistema configura la pizarra virtual por los datos insertados por el autor y finaliza el caso de uso.
Flujo Alternativo		
Acción del Actor		Respuesta del sistema
5. El autor selecciona la opción “ <i>Entrar</i> ”.		6. El sistema muestra una pizarra virtual donde los autores podrán dejar anotaciones acerca de la creación del OA y finaliza el caso de uso.
Prioridad	Opcional.	

Anexo 4 Descripción textual CU_ Mostrar área de edición del OA.

Caso de uso	Mostrar área de edición del OA.
Actores	Creador de colaboración.
Resumen	El caso de uso comienza cuando el creador de colaboración desea ir al área de edición del OA desde el área de trabajo colaborativo y finaliza cuando la misma es mostrada.
Referencias	R19.

Precondiciones	El autor debe ser el creador de la colaboración. Debe encontrarse en el área de trabajo colaborativo.
Poscondiciones	Se muestra el área de edición del OA.
Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El autor selecciona la opción “Ir a edición”.	2. El sistema muestra el área de edición del OA y finaliza el caso de uso.
Prioridad	Secundario.

Anexo 5 Descripción textual CU_ Adicionar recurso a un comentario (extend).

Caso de uso	Adicionar recurso a un comentario (extend).
Actores	Autor.
Resumen	El caso de uso comienza cuando el autor desea subir un recurso y finaliza cuando es realizada la acción solicitada por el autor.
Referencias	R17.
Precondiciones	El autor debe estar autenticado en el sistema.
Poscondiciones	Es incorporado el recurso al comentario.
Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El autor selecciona el comentario.	2. El sistema muestra el contenido del comentario seleccionado y ofrece la posibilidad de: <i>Subir recurso</i> . Muestra un campo llamado “ <i>examinar</i> ” donde se especificará la dirección donde se encuentra el recurso que se desea subir, además de brinda la opción “ <i>Aportar</i> ”.
3. El autor introduce la dirección donde se encuentra el recurso que desea	4. El sistema le incorpora el recurso al comentario seleccionado y finaliza el caso de uso.

incorporarle al comentario y da clic en “Aportar”.	Y finaliza el caso de uso.
Prioridad	Secundario.

Anexo 6 Descripción textual CU_ Visualizar OA almacenado en el RHODA.

Caso de uso	Visualizar OA almacenado en el RHODA.	
Actores	Autor.	
Resumen	El caso de uso inicia cuando el autor decide visualizar un OA que se encuentra almacenado en el RHODA y finaliza cuando se realiza la acción solicitada.	
Referencias	R35.	
Precondiciones	El autor debe haber realizado una búsqueda de OA.	
Poscondiciones	Se visualiza el OA almacenado en el RHODA.	
Flujo Normal de Eventos		
	Acción del actor	Respuesta del sistema
	1. El autor selecciona la opción “Visualizar”.	2. El sistema muestra la estructura del OA seleccionado, así como sus contenidos. Y finaliza el caso de uso.
Prioridad	Opcional.	

Anexo 7 Descripción textual CU_ Editar OA almacenado en el RHODA.

Caso de uso	Editar OA almacenado en el RHODA.	
Actores	Autor.	
Resumen	El caso de uso inicia cuando el autor decide editar un OA que se encuentra almacenado en el RHODA y finaliza cuando se realiza la acción solicitada.	
Referencias	R36.	
Precondiciones	El autor debe haber realizado una búsqueda de OA.	
Poscondiciones	Es realizada una copia del OA en la sección de OA en edición.	

Flujo Normal de Eventos	
Acción del actor	Respuesta del sistema
1. El autor selecciona la opción “ <i>Editar</i> ”.	2. El sistema crea una copia del OA en el área de trabajo del autor para que pueda trabajar con él.
3. El autor modifica los elementos pertenecientes al OA, como su estructura y contenidos, y da clic en <i>Guardar</i> .	4. El sistema guarda los cambios realizados sobre el OA.
Prioridad	Opcional.

Anexo 8 Descripción textual CU_ Gestionar recurso web.

Caso de uso	Gestionar recurso web.
Actores	Autor
Resumen	El caso de uso comienza cuando el autor desea crear, editar o eliminar un recurso web y finaliza cuando el autor realiza la acción.
Referencias	R20, R21, R22.
Precondiciones	El autor debe estar autenticado en el sistema y encontrarse editando un OA.
Poscondiciones	El recurso web es creado, editado o eliminado.
Flujo Normal de Eventos	
Sección #1 “ <i>Crear recurso web</i> ”	
Acción del actor	Respuesta del sistema
1. El autor selecciona la opción “ <i>Crear recurso web</i> ”.	2. El sistema ofrecerá un editor HTML con diferentes opciones de edición y un botón para guardar.
3. El autor inserta los elementos que desea.	

4. El autor accede al botón “ <i>Guardar</i> ”.	5. El sistema crea el recurso web. Y finaliza la sección.
Sección #2 “<i>Editar recurso web</i>”	
1. El autor selecciona la opción “ <i>Editar recurso web</i> ”.	2. El sistema muestra el recurso web previamente creado con todos sus elementos para editarlo. Y finaliza la sección.
Sección #3 “<i>Eliminar recurso web</i>”	
1. El autor selecciona la opción “ <i>Eliminar recurso web</i> ”.	2. El sistema muestra un mensaje de verificación con opciones de confirmación: “ <i>Desea eliminar el recurso</i> ”. Y finaliza la sección.
3. El autor confirma el mensaje.	4. El sistema elimina el recurso y muestra el mensaje: “ <i>El recurso ha sido eliminado con éxito</i> ”. Y finaliza la sección.
Flujo Alternativo	
Acción del actor	Respuesta del sistema
3. El autor no acepta el mensaje.	4. El sistema no elimina el recurso HTML y regresa a la interfaz de creación del curso.
Prioridad	Opcional.

Anexo 9 Descripción textual CU_ Gestionar ejercicio de orden cronológico.

Caso de Uso:	Gestionar ejercicio de orden cronológico.
Actores:	Autor.
Resumen:	El CU inicia cuando el autor desea gestionar un ejercicio de orden cronológico, y finaliza después de haber realizado esta acción.
Referencia:	R23, R24, R25.
Precondiciones:	El autor debe estar autenticado en el sistema.
Poscondiciones:	Es creado, editado o eliminado el ejercicio de ordenar cronológicamente.

Flujo Normal de Eventos	
Sección #1 “Crear ejercicio de orden cronológico”.	
Acción del Actor	Respuesta del Sistema
1. El autor selecciona la opción de Gestionar ejercicio de orden cronológico.	2. El sistema muestra una interfaz para crear la pregunta mostrando un espacio para introducir el enunciado, así como campos de texto para editar las órdenes a enlazar y especificar el orden.
	3. El sistema muestra dos campos de texto para introducir la información de retroalimentación para la respuesta correcta e incorrecta.
4. El autor introduce la orden y las preguntas, que pueden ser texto, imágenes, etc. Y especifica las respuestas, así como la información de retroalimentación.	
5. El autor presiona la opción Aceptar.	6. El sistema crea la pregunta y muestra el mensaje de confirmación “ <i>La pregunta ha sido creada con éxito</i> ”. Y finaliza la sección.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
5. El autor presiona la opción Cancelar.	6. El sistema regresa a la interfaz de creación del curso.
Sección #2 “Eliminar elemento de la columna”	
Acción del Actor	Respuesta del Sistema
1. El autor presiona la opción Eliminar en un elemento de la columna.	2. El sistema elimina el elemento de la columna. Y finaliza la sección.

Sección #3 “Eliminar ejercicio de orden cronológico”	
Acción del Actor	Respuesta del Sistema
1. El autor presiona la opción Eliminar ejercicio de orden cronológico.	2. El sistema el mensaje: “¿Está seguro que desea eliminar el ejercicio?”, y dos opciones. a) <i>Aceptar</i> . b) <i>Cancelar</i> .
3. El autor selecciona la opción “ <i>Aceptar</i> ”.	4. El sistema elimina el ejercicio y muestra el mensaje de confirmación “ <i>El ejercicio ha sido eliminado con éxito</i> ”. Y finaliza la sección.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
3. El autor presiona la opción “ <i>Cancelar</i> ”.	4. El sistema regresa a la interfaz de creación del curso.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
1. El autor selecciona la opción Agregar elemento al ejercicio de orden cronológico.	2. El sistema agrega un elemento al ejercicio y brinda la posibilidad de introducir la orden y especificar la respuesta.
Sección #5 “Editar ejercicio de orden cronológico”	
Acción del Actor	Respuesta del Sistema
1. El autor selecciona la opción Editar ejercicio de orden cronológico.	2. El sistema muestra una interfaz permitiendo modificar los elementos de las columnas y las opciones: a) <i>Aceptar</i> . b) <i>Cancelar</i> .
3. El autor realiza los cambios que desea y presiona la opción <i>Aceptar</i> .	4. El sistema guarda los cambios. Y finaliza la sección.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema

3. El autor presiona la opción Cancelar.	4. El sistema regresa a la interfaz de creación del curso.
Prioridad:	Opcional.

Anexo 10 Descripción textual CU_ Gestionar ejercicio de buscar en el texto.

Caso de Uso:	Gestionar ejercicios de buscar en el texto.	
Actores:	Autor	
Resumen:	El caso de uso se inicia cuando el Autor decide crear, editar o eliminar un ejercicio de Buscar en el texto y concluye cuando se ejecuta la acción.	
Referencia:	R26, R27, R28.	
Precondiciones:	El Autor debe previamente autenticarse en el sistema.	
Poscondiciones:	.	
Flujo Normal de Eventos		
Sección “Crear ejercicio”		
Acción del Actor	Respuesta del Sistema	
1. El autor escoge la opción de “Crear un ejercicio” de buscar en el texto.	2. El sistema muestra una interfaz donde le permitirá al Autor insertar: <ul style="list-style-type: none"> • La orden del ejercicio. • El texto en el que deberá buscarse la palabra. • La descripción de la palabra a buscar. • La palabra correcta. • Información de retroalimentación, tanto para cuando la respuesta es correcta como para cuando es incorrecta. 	
	3. Mostrará un botón “Acepta” y otro “Cancelar”.	
4. El autor introduce en los campos correspondientes:		

<ul style="list-style-type: none"> • La orden del ejercicio. • El texto en que se debe buscar. • Describe la palabra que desea que el estudiante busque. • Especifica cuál es la palabra correcta. • Ofrece una información de retroalimentación tanto para cuando la respuesta del estudiante es correcta como para cuando no lo es. 	
5. Escoge la opción “Aceptar”.	6. El sistema guarda los cambios creando un nuevo ejercicio de búsqueda en el texto. Y finaliza la sección.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
5. Escoge la opción “Cancelar”.	6. El sistema no guarda los cambios y regresa a la vista de creación de un ejercicio.
Sección “Eliminar elemento”	
Acción del Actor	Respuesta del Sistema
1. El autor escoge la opción “Eliminar”.	2. El sistema elimina los campos correspondientes a la descripción de las palabras, las palabras correctas y las informaciones de retroalimentación. Y finaliza la sección.
Flujo Alternativo “Agregar elemento”	
Acción del Actor	Respuesta del Sistema
1. El autor escoge la opción “Agregar elemento”.	2. El sistema agrega los campos correspondientes a la descripción de las palabras, las palabras correctas y las informaciones de retroalimentación.
Sección “Editar ejercicio”	

Acción del Actor	Respuesta del Sistema
1. El autor escoge la opción de “Editar un ejercicio” de buscar en el texto.	2. El sistema muestra una interfaz donde le permitirá al Autor modificar: <ul style="list-style-type: none"> • La orden del ejercicio. • El texto en el que deberá buscarse la palabra. • La descripción de la palabra a buscar. • La palabra correcta. • Información de retroalimentación, tanto para cuando la respuesta es correcta como para cuando es incorrecta. Mostrará un botón “Aceptar” y otro “Cancelar”.
3. El autor escoge el campo o los campos que desea modificar e introduce los nuevos valores.	
4. Escoge la opción “Aceptar”.	5. El sistema guarda los cambios realizados al ejercicio editado y muestra el mensaje de confirmación “ <i>Los cambios han sido guardados</i> ”. Y finaliza la sección.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3. Escoge la opción “Agregar elemento”	4. Ver flujo alternativo “Agregar elemento”
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
4. Escoge la opción “Cancelar”.	5. El sistema no guarda los cambios.
	6. Regresa a la vista de creación de un ejercicio.
Sección “Eliminar ejercicio”	
Acción del Actor	Respuesta del Sistema

1. El autor escoge la opción de “ <i>Eliminar un ejercicio</i> ” de buscar en el texto.	2. El sistema muestra el mensaje “ <i>¿Está seguro que desea eliminar el ejercicio de buscar en el texto?</i> ” y dos botones “ <i>Aceptar</i> ” y “ <i>Cancelar</i> ” respectivamente.
3. El autor escoge la opción “ <i>Aceptar</i> ”.	4. El sistema elimina el ejercicio. 5. Concluye el caso de uso. Y finaliza la sección.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
3. El autor escoge la opción “ <i>Cancelar</i> ”.	4. Es cancelada la operación y el sistema visualiza la vista de creación del curso.
Prioridad:	Opcional.

Anexo 11 Descripción textual CU_ Gestionar ejercicio de enlazar columnas.

Caso de Uso:	Gestionar ejercicios de enlazar columnas.
Actores:	Autor.
Resumen:	El CU inicia cuando el Autor desea gestionar un ejercicio de enlazar columnas, y finaliza después de haber realizado esta acción.
Referencia:	R29, R30, R31.
Precondiciones:	El Autor debe estar autenticado en el sistema.
Poscondiciones:	Es creado un ejercicio de enlazar preguntas.
Flujo Normal de Eventos	
Sección #1 “Crear ejercicio de enlazar columnas”	
Acción del Actor	Respuesta del Sistema
1. El autor selecciona la opción de Crear ejercicio de enlazar columnas.	2. El sistema muestra una interfaz para crear la pregunta mostrando un espacio para introducir el enunciado, así como campos de texto para editar las órdenes a enlazar y especificar las respuestas.

	3. El sistema muestra dos campos de texto para introducir la información de retroalimentación para la respuesta correcta e incorrecta.
4. El autor introduce el enunciado y los elementos de las columnas, que pueden ser texto, imágenes, etc. Además especifica las respuestas, así como la información de retroalimentación.	
5. El autor presiona la opción <i>Aceptar</i> .	6. El sistema crea la pregunta y muestra el mensaje de confirmación “La pregunta ha sido creada con éxito”. Y finaliza la sección.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
5. El autor presiona la opción <i>Cancelar</i> .	6. El sistema regresa a la interfaz de creación del curso.
Sección #2 “Eliminar elemento de la columna”	
Acción del Actor	Respuesta del Sistema
1. El autor presiona la opción Eliminar en un elemento de la columna.	2. El sistema elimina el elemento de la columna. Y finaliza la sección.
Sección #3 “Eliminar ejercicio de enlazar columnas”	
Acción del Actor	Respuesta del Sistema
1. El autor presiona la opción Eliminar ejercicio de enlazar columnas.	2. El sistema el mensaje: “¿Está seguro que desea eliminar el ejercicio?”, y dos opciones. a) <i>Aceptar</i> . b) <i>Cancelar</i> .
3. El autor selecciona la opción <i>Aceptar</i> .	4. El sistema elimina el ejercicio y regresa a la interfaz de creación del curso. Y finaliza la sección.

Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
3. El autor selecciona la opción <i>Cancelar</i> .	4. El sistema regresa a la interfaz de creación del curso.
Flujo Alternativo “Agregar elemento al ejercicio de enlazar columnas”	
Acción del Actor	Respuesta del Sistema
1. El Autor selecciona la opción Agregar elemento al ejercicio de enlazar columnas.	2. El sistema agrega un elemento al ejercicio y brinda la posibilidad de introducir la orden y especificar la respuesta.
Sección #5 “Editar ejercicio de enlazar columnas”	
Acción del Actor	Respuesta del Sistema
1. El Autor selecciona la opción Editar ejercicio de enlazar columnas.	2. El sistema muestra una interfaz permitiendo modificar los elementos de las columnas y las opciones: b) Aceptar. c) Cancelar.
3. El autor realiza los cambios que desea y presiona la opción <i>Aceptar</i> .	4. El sistema guarda los cambios. Y finaliza la sección.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
3. El Autor presiona la opción <i>Cancelar</i> .	4. El sistema regresa a la interfaz de creación del curso.
Prioridad	Opcional.

Anexo 12 Descripción textual CU_ Gestionar cuestionario.

Caso de Uso:	Gestionar cuestionario.
Actores:	Autor.
Resumen:	El CU inicia cuando el autor selecciona la opción Gestionar ejercicios de selección simple y finaliza cuando se haya realizado esta acción.
Referencia:	R32, R33, R34.
Precondiciones:	El Autor debe estar autenticado en el sistema.

Poscondiciones:	Es creado el ejercicio de selección simple	
Flujo Normal de Eventos		
Sección # 1 “Crear ejercicio de selección simple”		
Acción del Actor	Respuesta del Sistema	
1. El autor selecciona la opción “ <i>Crear ejercicio de selección simple</i> ”.	2. El sistema muestra una interfaz para crear el ejercicio mostrando un espacio para introducir la orientación, así como campos de texto para insertar las preguntas y las opciones de respuesta, especificando cuál es la correcta. El sistema muestra dos campos de texto para introducir la información de retroalimentación para la respuesta correcta e incorrecta. <i>Además de las opciones de Aceptar y Cancelar.</i>	
3. El autor introduce la orientación y la pregunta. Además especifica la opción correcta, así como la información de retroalimentación.		
4. El autor presiona la opción Aceptar.	5. El sistema crea el ejercicio y muestra el mensaje de confirmación “ <i>El ejercicio ha sido creado con éxito</i> ”. Y finaliza la sección.	
Flujo Alternativo		
Acción del Actor	Respuesta del Sistema	
4. El autor presiona la opción “ <i>Cancelar</i> ”.	5. El sistema regresa a la interfaz de creación del curso.	
Flujo Alternativo “Adicionar opción al ejercicio de selección simple”		
Acción del Actor	Respuesta del Sistema	
1. El autor selecciona la opción “ <i>Adicionar opción</i> ” al ejercicio de selección simple.	2. El sistema adiciona la opción, permitiéndole introducir el texto de la misma.	
3. El autor adiciona la opción y especifica que es correcta.	4. El sistema almacena esa opción como correcta.	

Flujo Alternativo	
3. El autor adiciona la opción y no especifica que es correcta.	4. El sistema almacena esa opción como incorrecta
Flujo Alternativo “Adicionar pregunta de selección simple”	
Acción del Actor	Respuesta del Sistema
1. El autor selecciona la opción “Adicionar pregunta” al ejercicio de selección simple.	2. El sistema muestra una interfaz para adicionar la pregunta. Ver sección 1.1. Y finaliza la sección.
Sección # 4 “Editar ejercicio de selección simple”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El autor selecciona la opción “Editar” del ejercicio de selección simple.	2. El sistema muestra una interfaz en la que permite al autor realizar cambios en la pregunta y las opciones: a) Aceptar. b) Cancelar.
3. El Autor realiza los cambios que desea y presiona la opción Aceptar.	4. El sistema guarda los cambios. Y finaliza la sección.
Sección # 5 “Eliminar opción del ejercicio de selección simple”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Autor selecciona la opción “Eliminar opción” del ejercicio de selección simple.	2. El sistema elimina la opción seleccionada. Y finaliza la sección.
Sección # 6 “Eliminar ejercicio de selección simple”	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El autor selecciona la opción “Eliminar” del ejercicio de selección simple.	2. El sistema el mensaje: “¿Está seguro que desea eliminar el ejercicio?”, y dos opciones. a) Aceptar. b) Cancelar.

3. El autor selecciona la opción Aceptar.	4. El sistema elimina el ejercicio y muestra el mensaje de confirmación "El ejercicio ha sido eliminado con éxito". Y finaliza la sección.
Flujo Alternativo	
Acción del Actor	Respuesta del Sistema
3. El autor selecciona la opción Cancelar.	4. El sistema regresa a la interfaz de creación del curso.
Prioridad	Opcional.

Anexo 13 Diagrama de clases de análisis CU_ Editar colaboración.

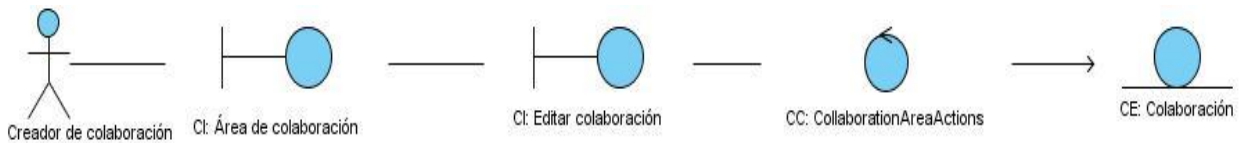


Figura 18 Diagrama de clases de análisis "Editar colaboración".

Anexo 14 Diagrama de clases de análisis CU_ Configurar pizarra virtual.



Figura 19 Diagrama de clases de análisis "Configurar pizarra virtual".

Anexo 15 Diagrama de clases de análisis CU_ Visualizar pizarra virtual.

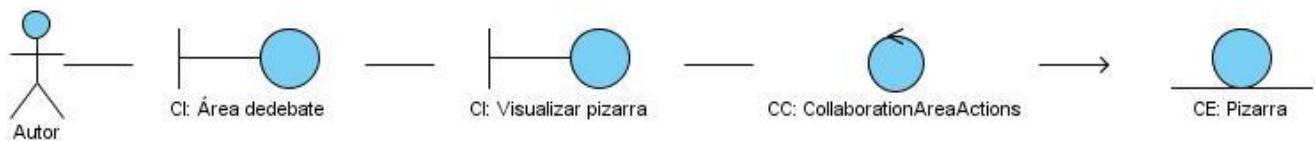


Figura 20 Diagrama de clases de análisis "Visualizar pizarra virtual".

Anexo 16 Diagrama de clases de análisis CU_ Mostrar área de edición del OA.



Figura 21 Diagrama de clases de análisis "*Mostrar área de edición del OA*".

Anexo 17 Diagrama de clases de análisis CU_ Adicionar recurso a un comentario.



Figura 22 Diagrama de clases de análisis "*Adicionar recurso a un comentario*".

Anexo 18 Diagrama de clases de análisis CU_ Visualizar OA almacenado en el RHODA.

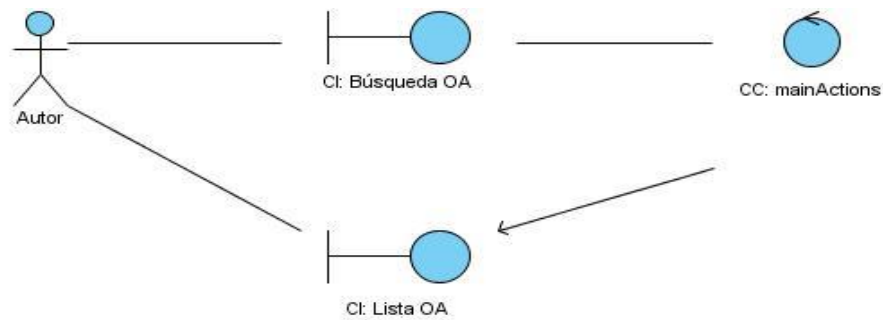


Figura 23 Diagrama de clases de análisis "*Visualizar OA almacenado en RHODA*".

Anexo 19 Diagrama de clases de análisis CU_ Editar OA almacenado en el RHODA.

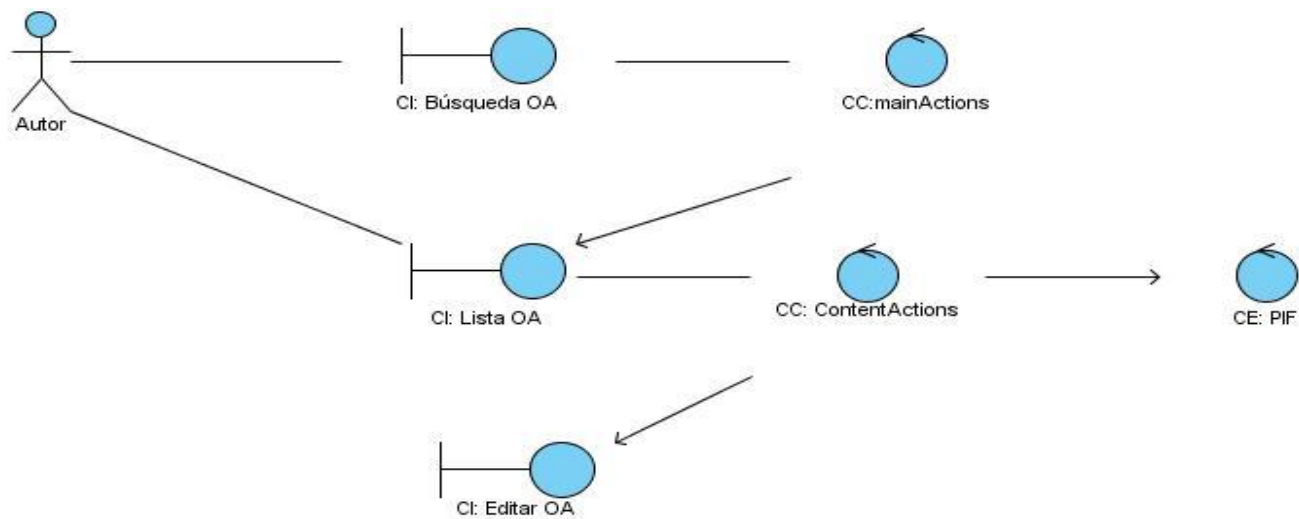


Figura 24 Diagrama de clases de análisis "Editar OA almacenado en el RHODA".

Anexo 20 Diagrama de clases de análisis CU_ Gestionar recurso web.



Figura 25 Diagrama de clases de análisis "Gestionar Recurso Web".

Anexo 21 Diagrama de clases de análisis CU_ Gestionar ejercicio de orden cronológico.

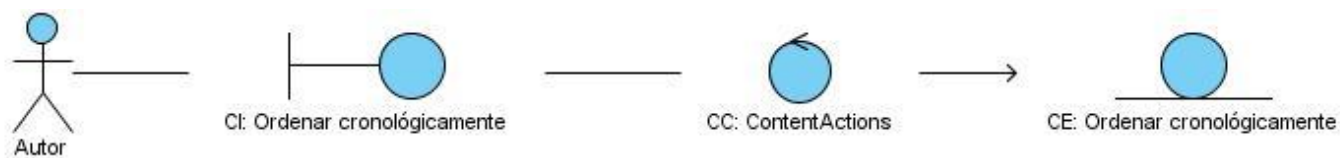


Figura 26 Diagrama de clases de análisis "Gestionar ejercicio de orden cronológico".

Anexo 22 Diagrama de clases de análisis CU_ Gestionar ejercicio de buscar en el texto.

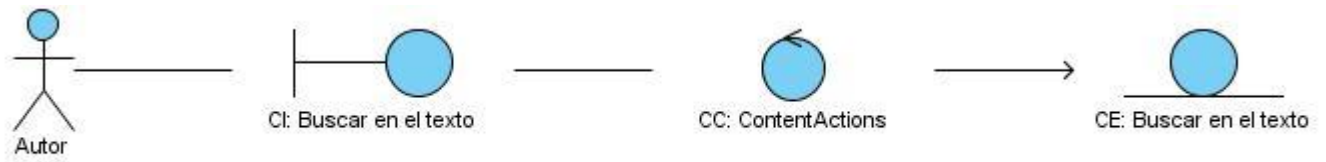


Figura 27 Diagrama de clases de análisis "Gestionar ejercicio de buscar en el texto".

Anexo 23 Diagrama de clases de análisis CU_ Gestionar ejercicio de enlazar columnas.

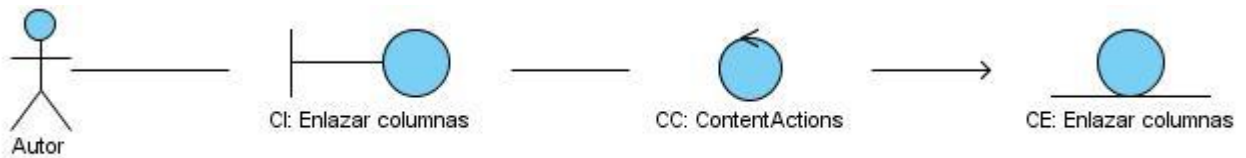


Figura 28 Diagrama de clases de análisis "Gestionar ejercicio de enlazar columnas".

Anexo 24 Diagrama de clases de análisis CU_ Gestionar cuestionario.

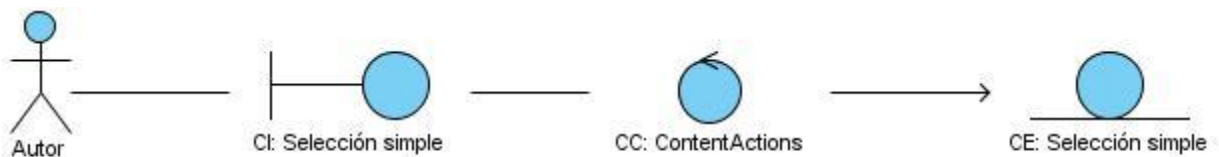


Figura 29 Diagrama de clases de análisis "Gestionar cuestionario".

Anexo 25 Diagrama de clases de diseño CU_Editar colaboración.

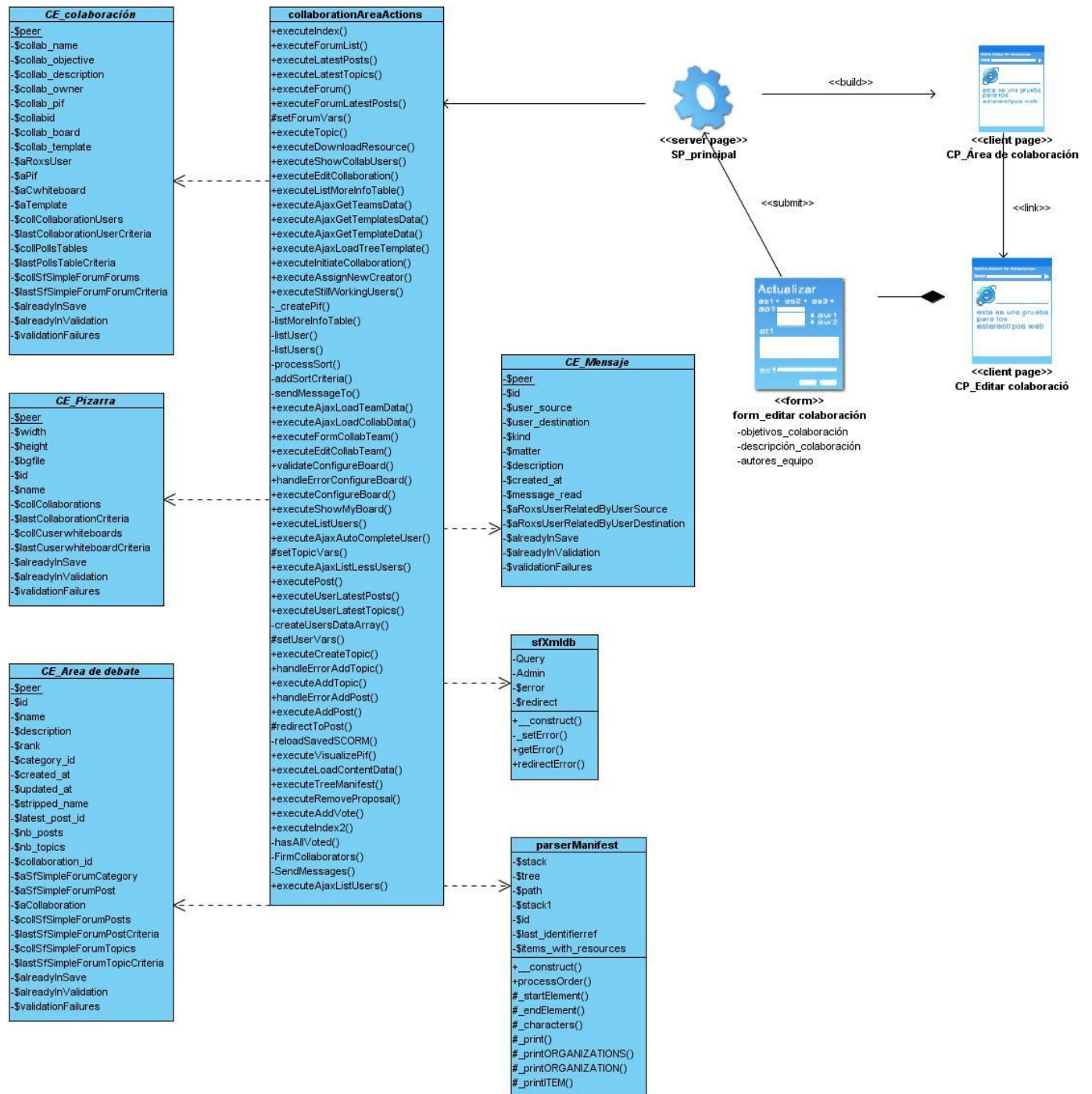


Figura 30 Diagrama de clases de diseño "Editar colaboración".

Anexo 26 Diagrama de clases de diseño CU_ Configurar pizarra virtual.

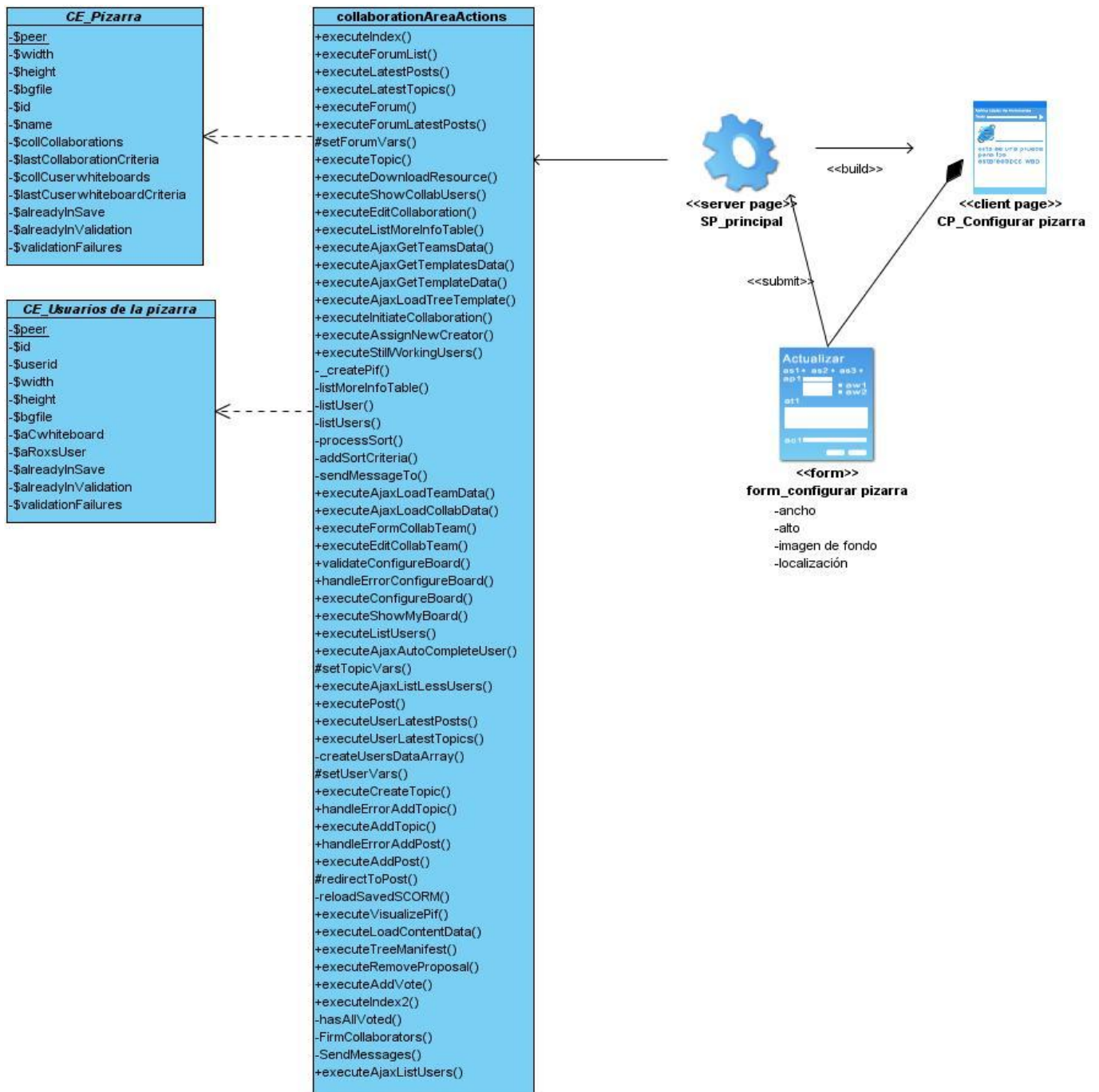


Figura 31 Diagrama de clases de diseño "Configurar pizarra virtual".

Anexo 27 Diagrama de clases de diseño CU_ Visualizar pizarra virtual.

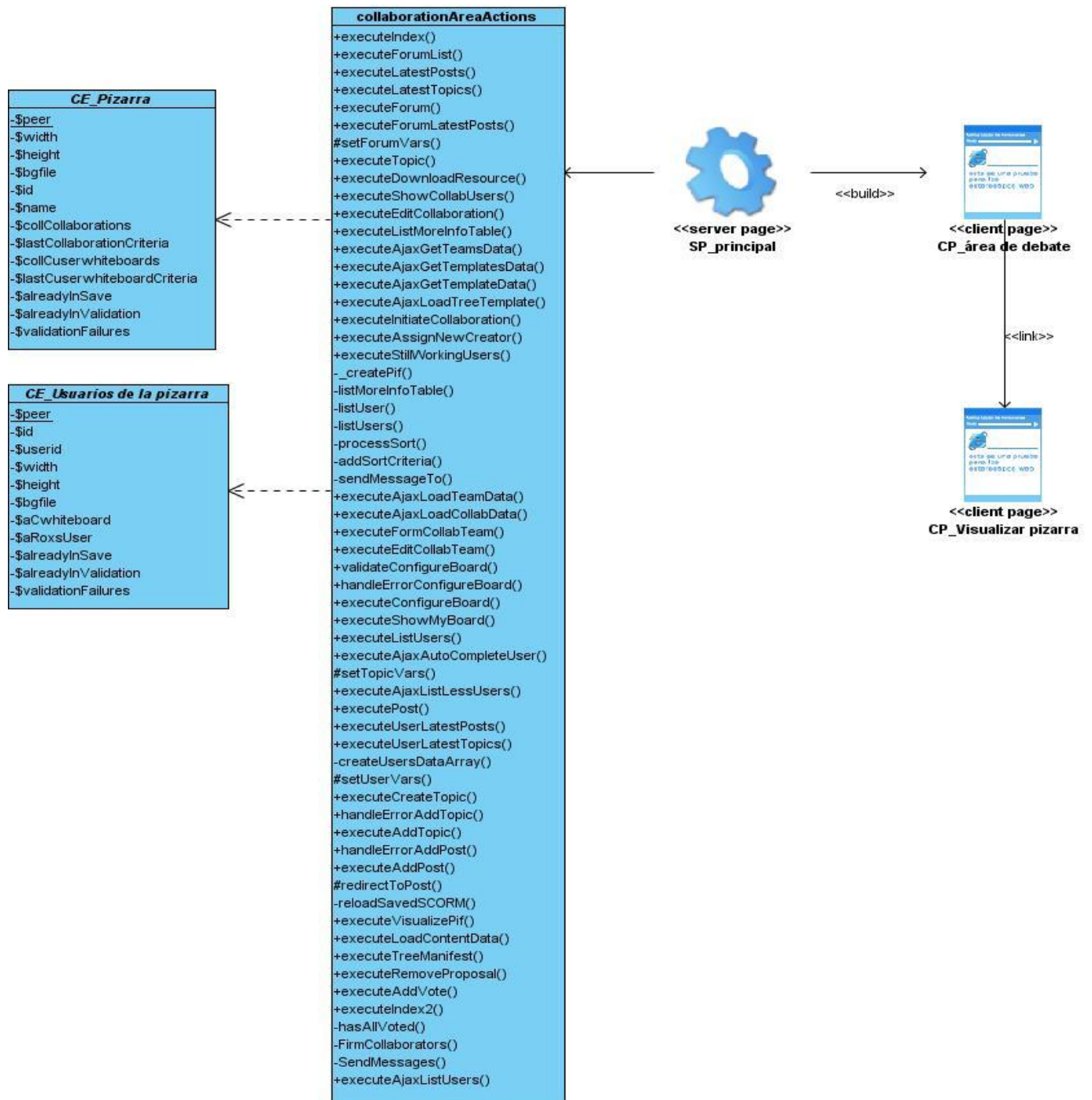


Figura 32 Diagrama de clases de diseño "Visualizar pizarra virtual".

Anexo 28 Diagrama de clases de diseño CU_Mostrar área de edición del OA.

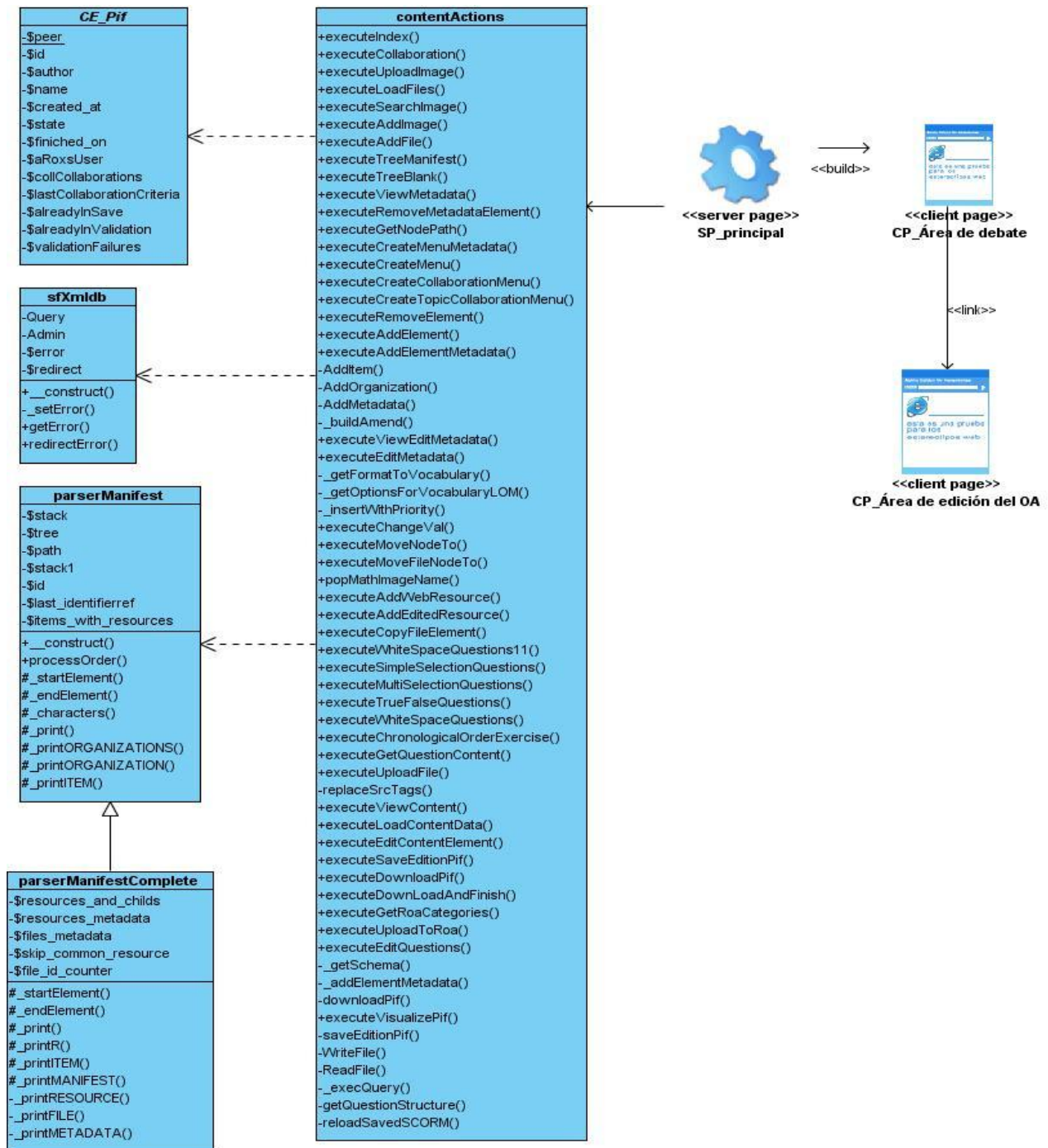


Figura 33 Diagrama de clases de diseño "Mostrar área de edición del OA".

Anexo 29 Diagrama de clases de diseño CU_ Adicionar recurso a un comentario.

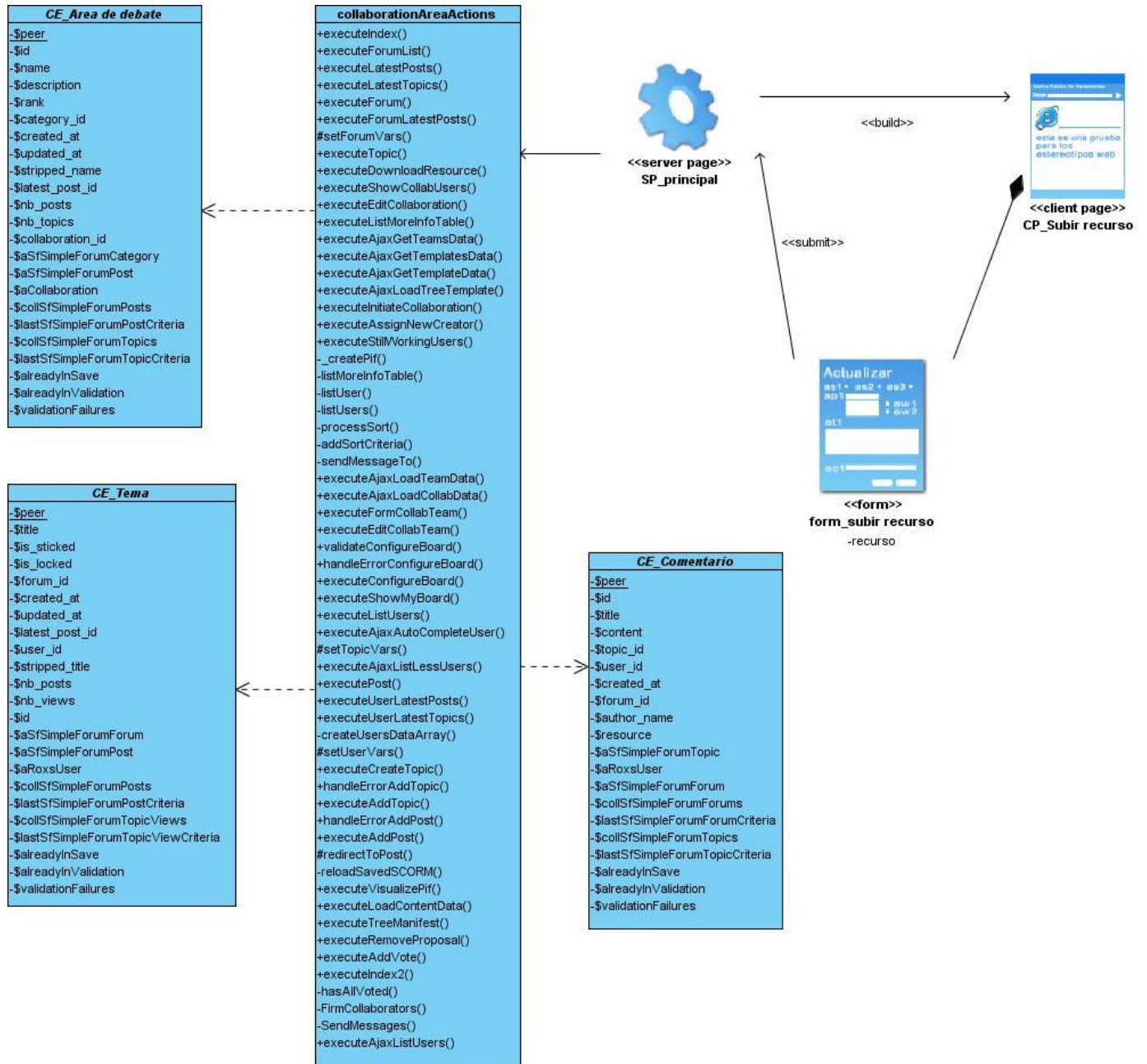


Figura 34 Diagrama de clases de diseño "Adicionar recurso a un comentario".

Anexo 30 Diagrama de clases de diseño CU_Visualizar OA almacenado en el RHODA.

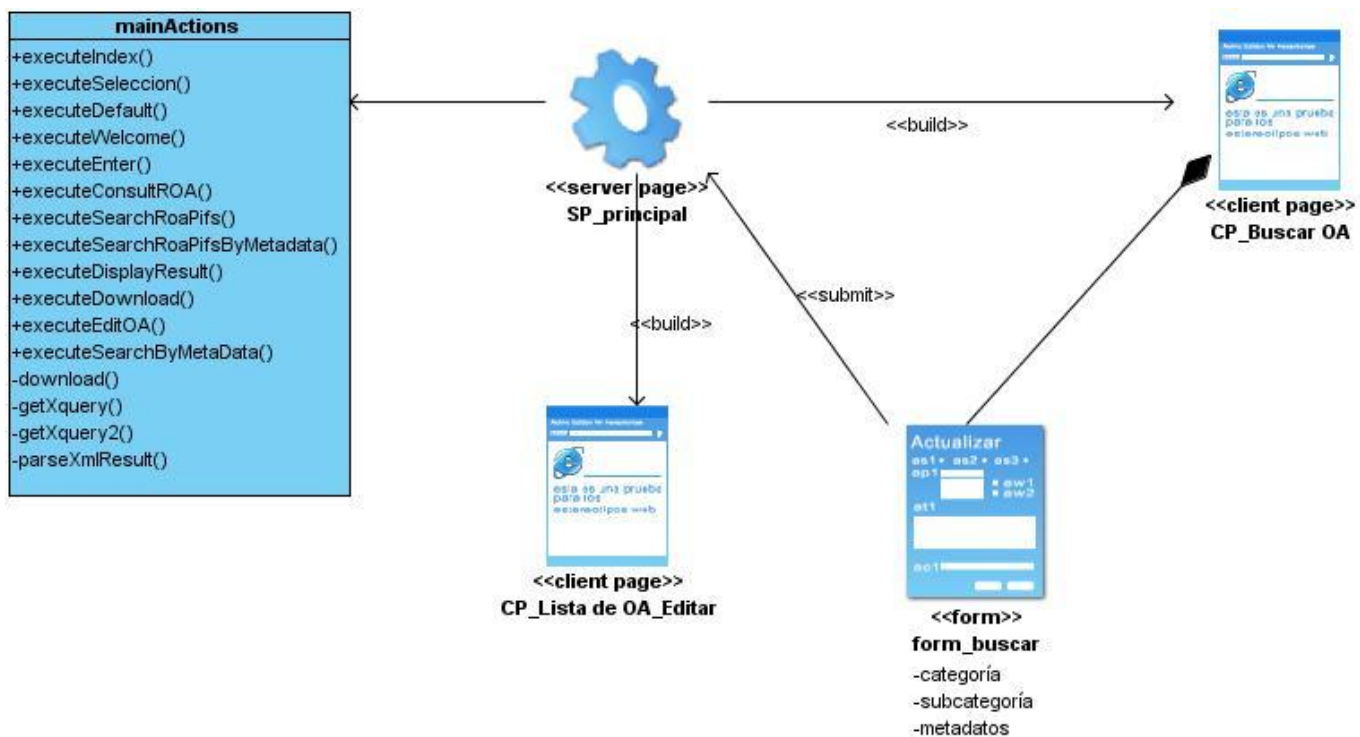


Figura 35 Diagrama de clases de diseño "Visualizar OA almacenado en el RHODA".

Anexo 31 Diagrama de clases de diseño CU_Editar OA almacenado en el RHODA.

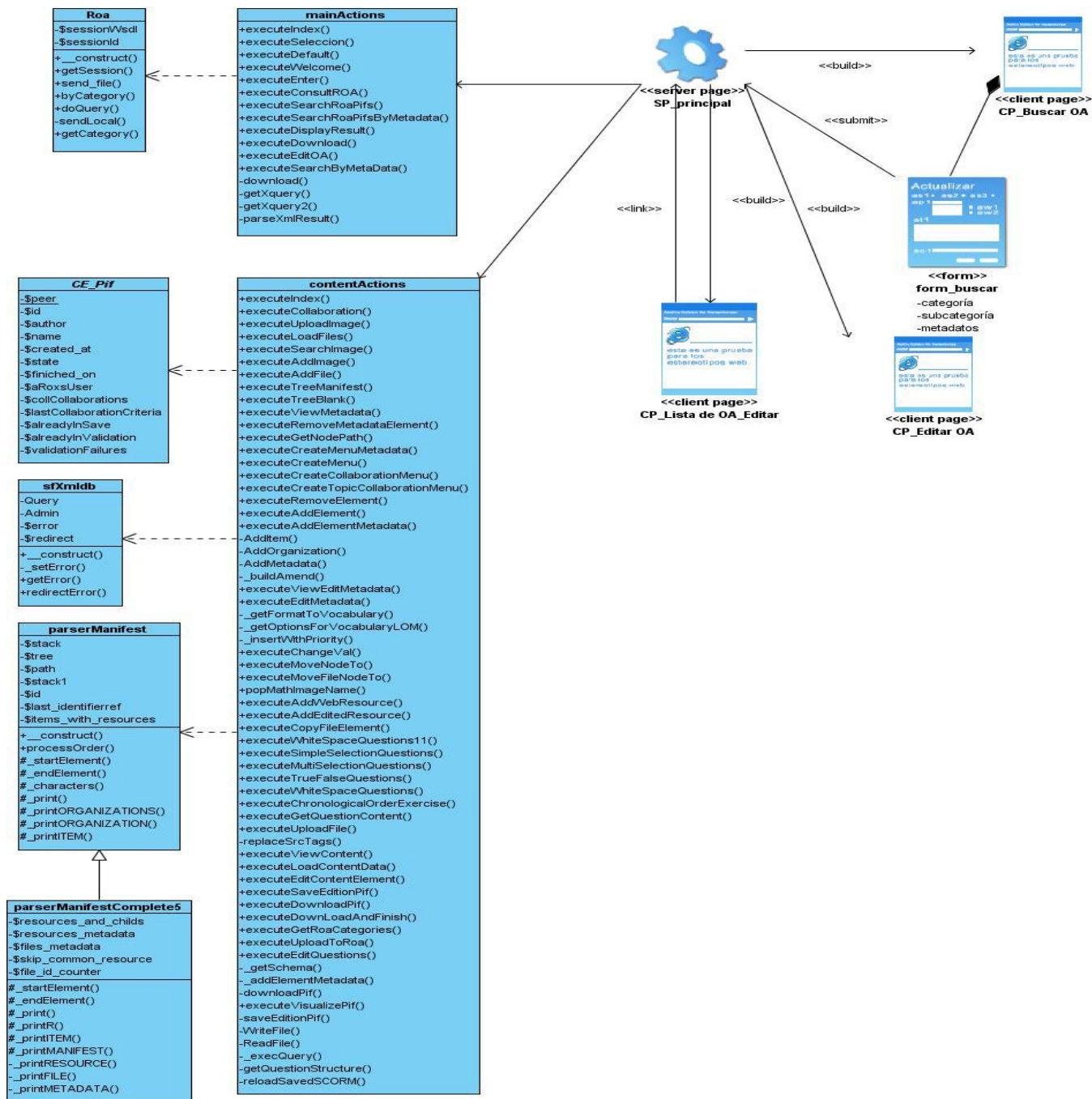


Figura 36 Diagrama de clases de diseño "Editar OA almacenado en el RHODA".

Anexo 32 Diagrama de clases de diseño CU_ Gestionar recurso web.

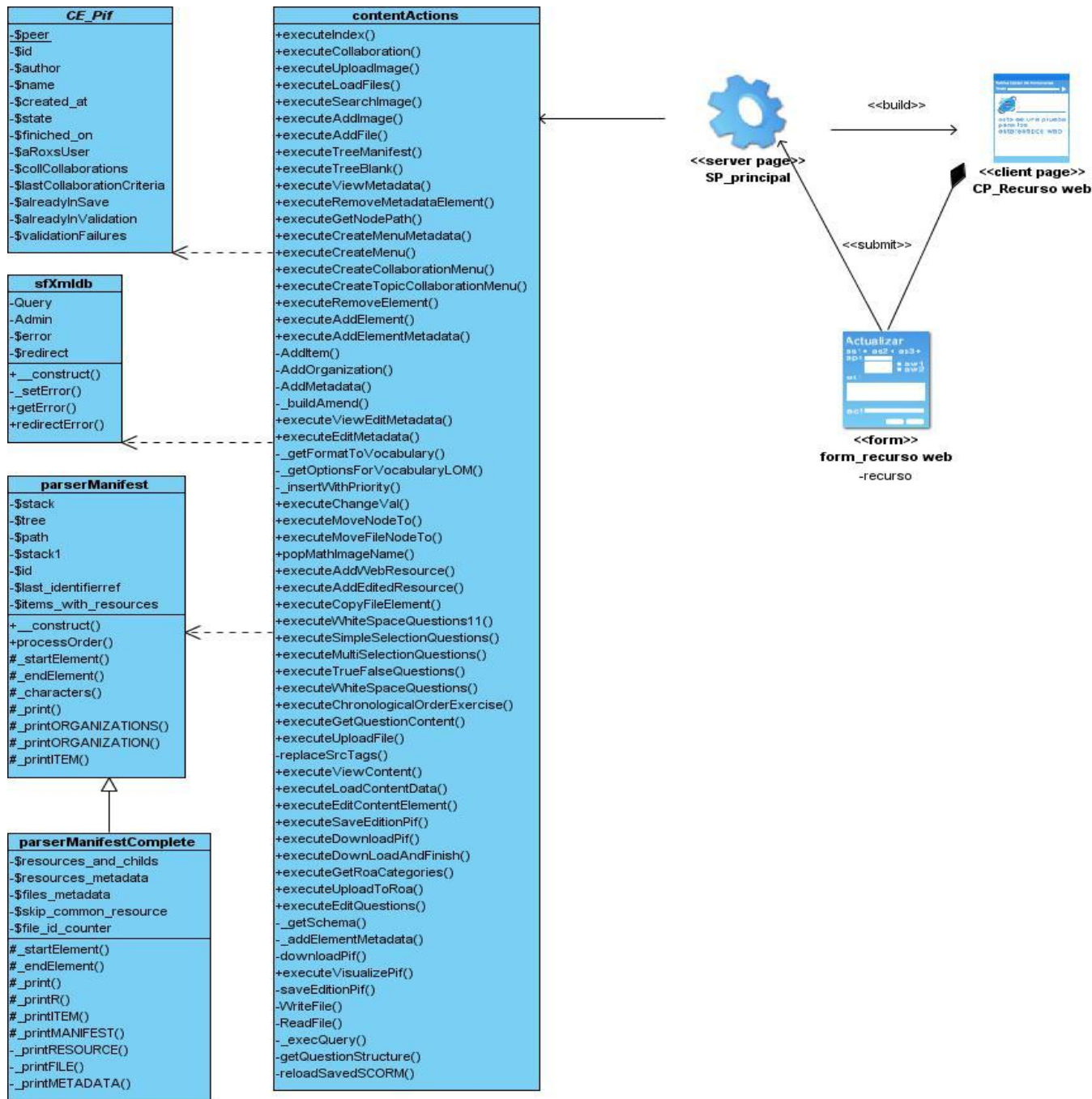


Figura 37 Diagrama de clases de diseño "Gestionar recurso web".

Anexo 33 Diagrama de clases de diseño CU_ Gestionar ejercicio de orden cronológico.

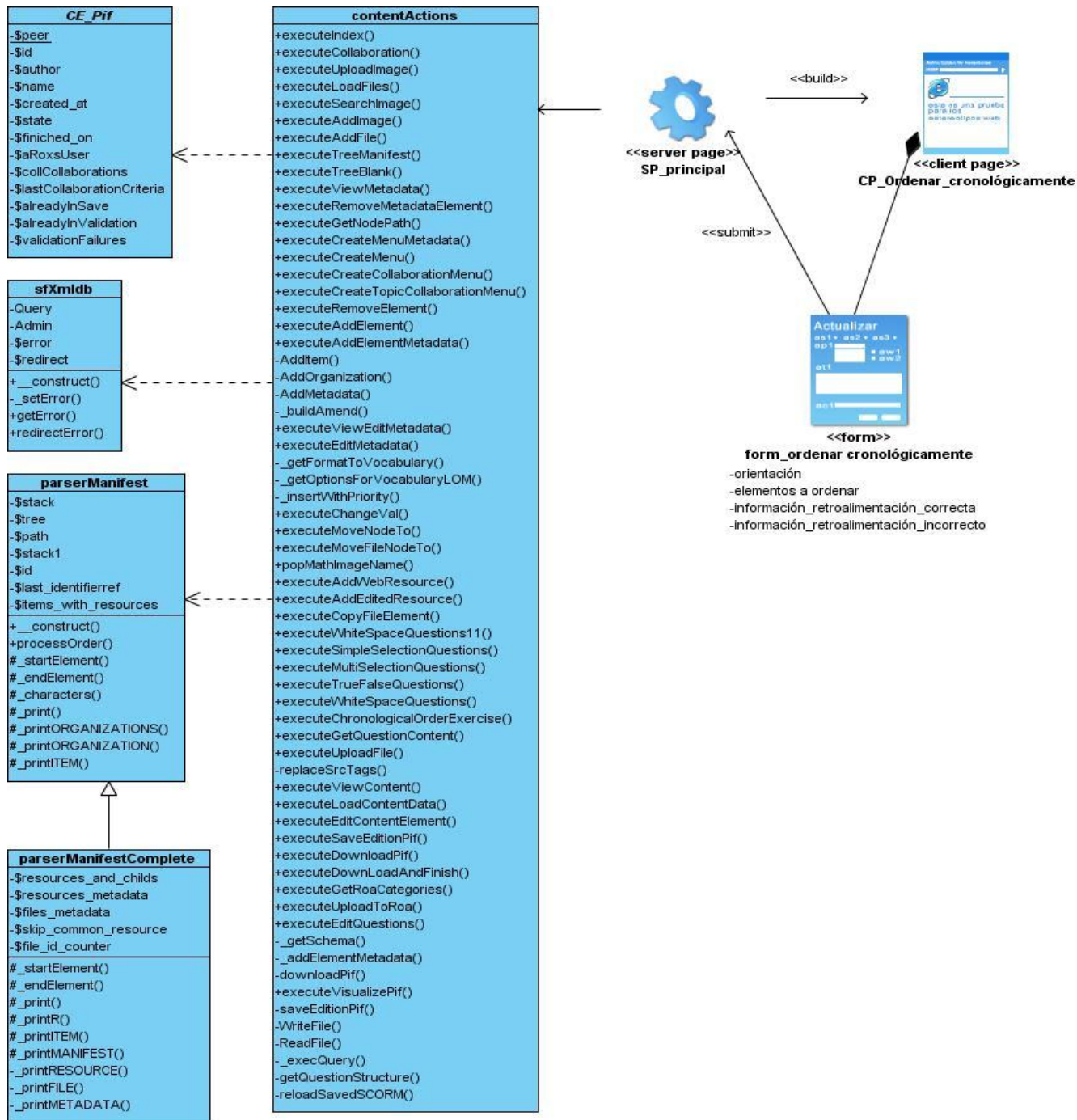


Figura 38 Diagrama de clases de diseño "Gestionar ejercicio de orden cronológico".

Anexo 34 Diagrama de clases de diseño CU_ Gestionar ejercicio de buscar en el texto.

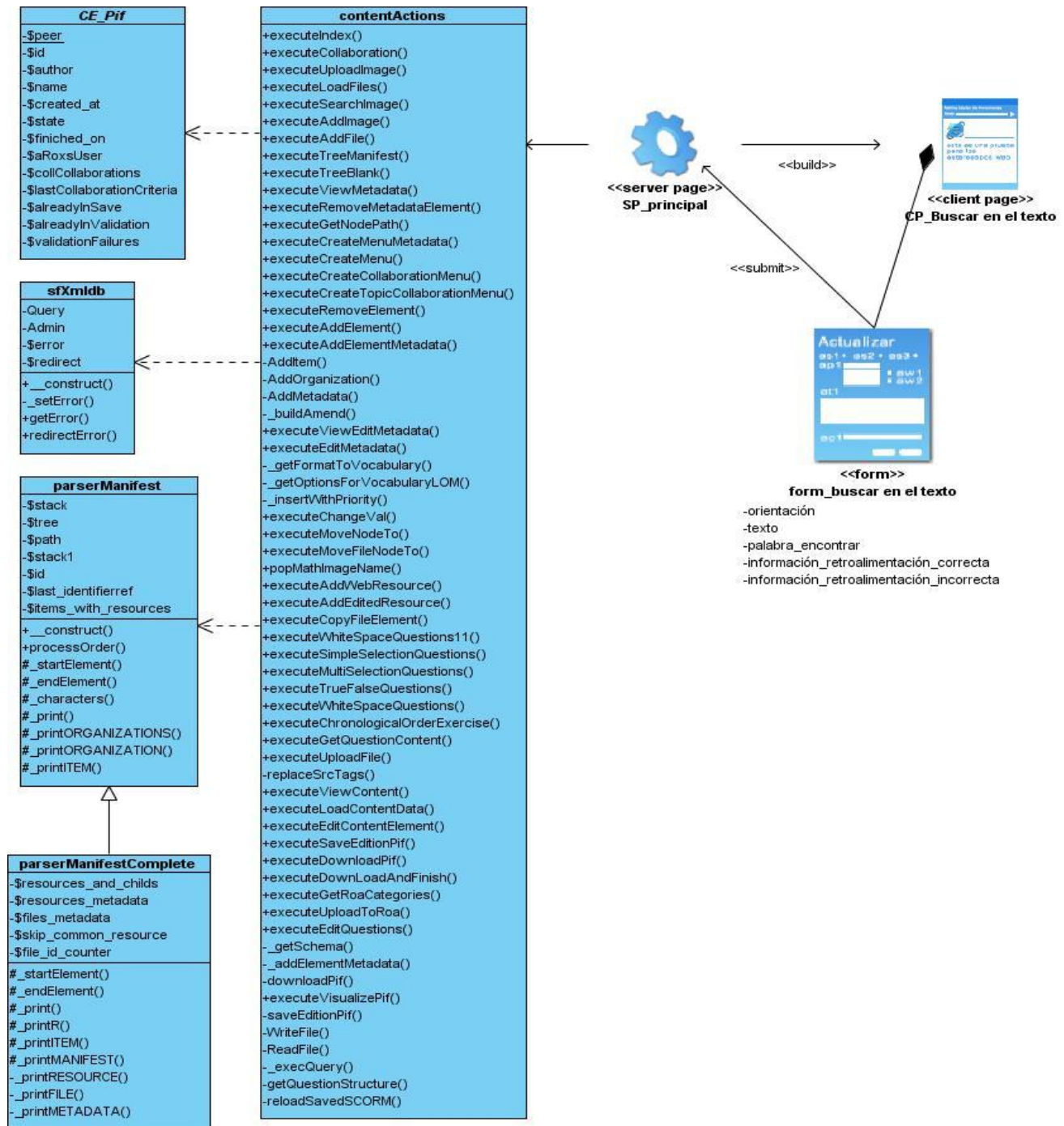


Figura 39 Diagrama de clases de diseño "Gestionar ejercicios de buscar en el texto".

Anexo 35 Diagrama de clases de diseño CU_ Gestionar ejercicio de enlazar columnas.

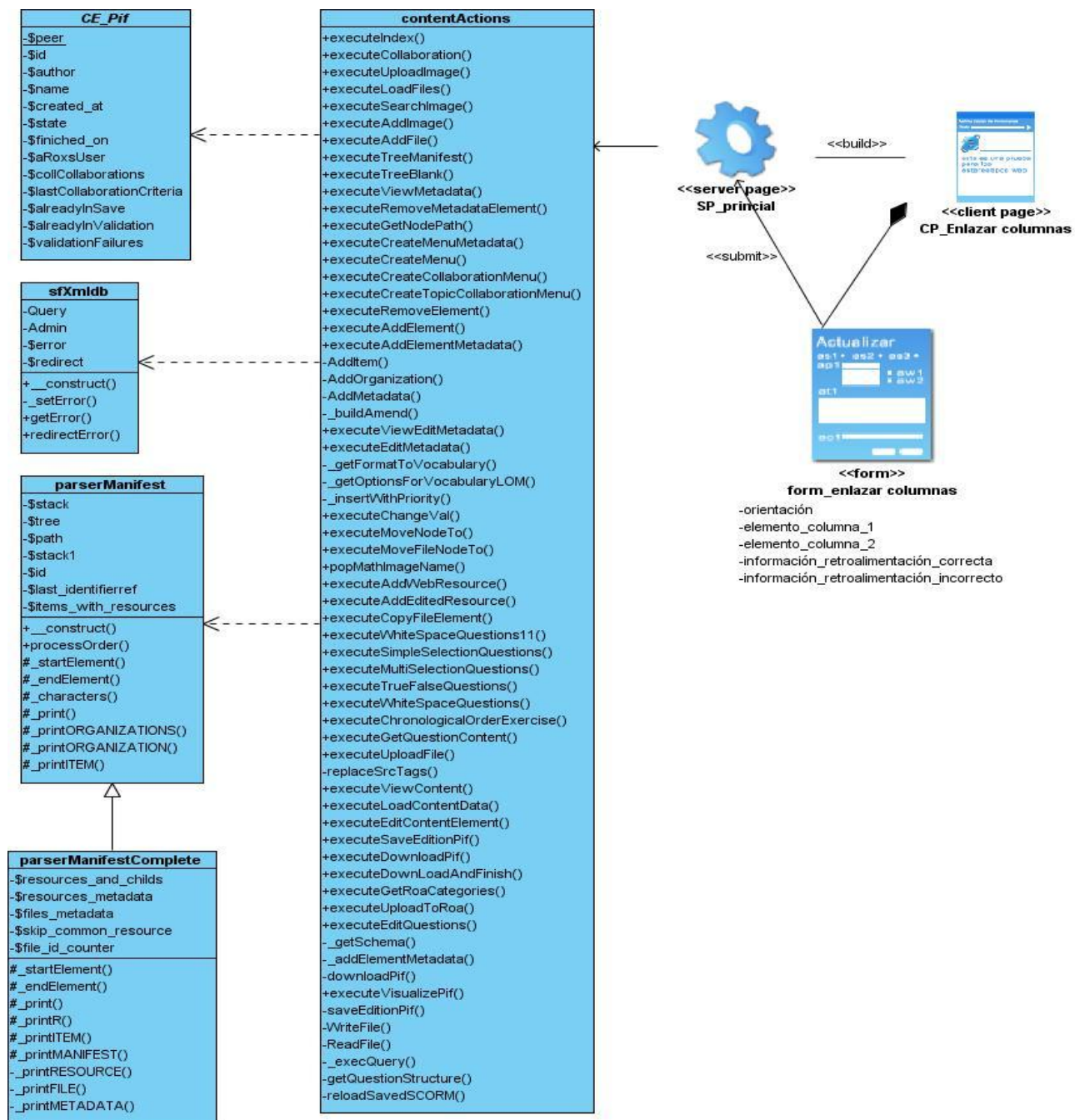


Figura 40 Diagrama de clases de diseño "Gestionar ejercicio de enlazar columnas".

Anexo 36 Diagrama de clases de diseño CU_ Gestionar cuestionario.

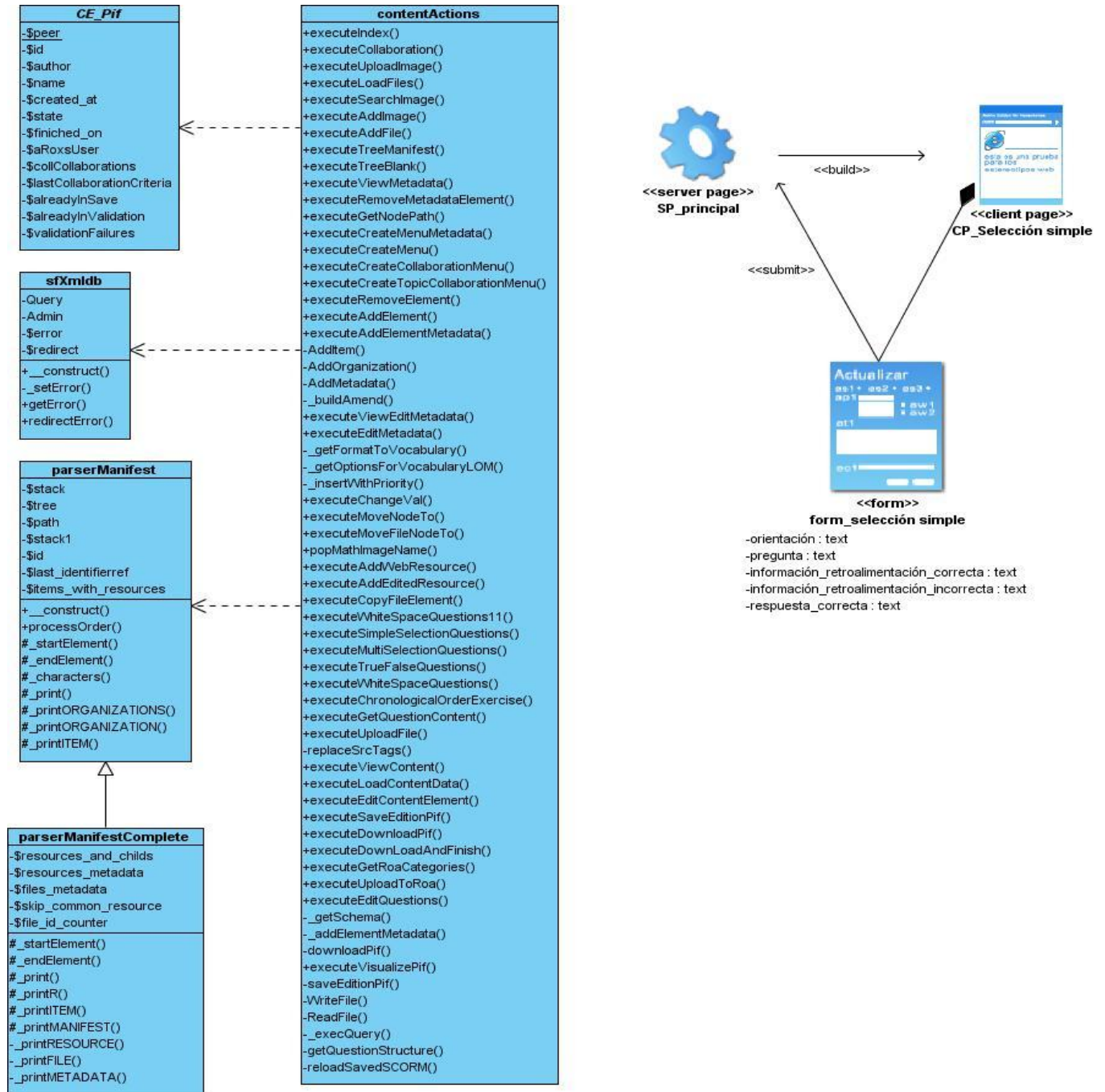


Figura 41 Diagrama de clases de diseño "Gestionar cuestionario".

Anexo 37 Descripción de las clases de la base de datos.

Nombre:	roxs_user.	
Descripción:	Tabla referente a los usuarios de la aplicación, la cual contiene los datos generales y específicos de los mismos, nombre de usuario, contraseña, nombre y apellidos, rol que ocupará en la aplicación (autor, creador de colaboración), el tipo de acceso a la herramienta, a través de LDAP o manual así como identificador del propio usuario.	
Atributo:	Tipo:	Descripción:
<ul style="list-style-type: none"> • Id (llave primaria). • User_name. • Pass. • Name. • Surname. • E-mail. • Rol_id. • Occupation. • Acc_type. • Created_at. • Updated_at. • Blocked. • Deleted. • User_templates. 	<ul style="list-style-type: none"> • Serial • Text. • Text. • Text. • Text. • Text. • Integer. • Text. • Text. • Timestamp. • Timestamp. • Smalling. • Boolean. • Smalling. 	<ul style="list-style-type: none"> • Identificador del usuario. • Nombre de usuario. • Contraseña para acceder al sistema. • Nombre. • Apellidos. • Correo electrónico. • Identificador del rol. • Ocupación. • Tipo de acceso al sistema (LDAP-Manual). • Fecha de registro del usuario. • Fecha de actualización de la cuenta del usuario. • Bloqueo en caso de contraseña no válida. • Atributo para la eliminación de usuario. • Máximo de plantillas en posesión.

Nombre:	pif.	
Descripción:	Tabla referente a los OA conformados en la aplicación, la cual contiene los datos generales y específicos de los mismos, nombre del OA, autor, fecha de creado, fecha de culminación, estado del OA, así como el identificador del mismo.	
Atributo:	Tipo:	Descripción:
<ul style="list-style-type: none"> • Id (llave primaria). • Author. • State. • Name. • Created_at. • Finish_On. 	<ul style="list-style-type: none"> • Serial. • Integer. • Integer. • Text. • Timestamp. • Timestamp. 	<ul style="list-style-type: none"> • Identificador del OA. • Autor que conformó el OA. • Estado del OA (en edición). • Nombre del OA. • Fecha de creación del OA. • Fecha de finalización de la creación del OA.

Nombre:	template.	
Descripción:	Tabla referente a las plantillas creadas en la aplicación, la cual contiene nombre, autor y descripción de la misma, condición, fecha de creada y el identificador de la plantilla.	
Atributo:	Tipo:	Descripción:
<ul style="list-style-type: none"> • Id (llave primaria). • Name • Author. • Description. • Created_at. • Condition. 	<ul style="list-style-type: none"> • Serial. • Text. • Integer. • Text. • Timestamp. • Char. 	<ul style="list-style-type: none"> • Identificador de la plantilla. • Nombre de la plantilla. • Autor que conformó la plantilla. • Breve descripción de la plantilla. • Fecha de creación de la plantilla. • Estado de plantilla (privada, propuesta, pública).

Nombre:	message.	
Descripción:	Tabla referente a los mensajes gestionados en el proceso de revisión de la plantilla y en la creación colaborativa de un OA principalmente, la cual contiene asunto, remitente y	

destinatario del mismo además de la fecha de enviado y leído.		
Atributo:	Tipo:	Descripción:
<ul style="list-style-type: none"> • Id (llave primaria). • User_source. • User_destination. • Matter. • Message_read. • Created_at. • Description. • Kind. 	<ul style="list-style-type: none"> • Serial. • Integer. • Integer. • Text. • Smalling. • Timestamp. • Text. • Smalling. 	<ul style="list-style-type: none"> • Identificador del mensaje. • Remitente del mensaje. • Destinatario del mensaje. • Asunto del mensaje. • Estado del mensaje (leído o no leído). • Fecha de creación del mensaje. • Cuerpo del mensaje. • Bandeja de entrada o salida.

Nombre:	collaboration.	
Descripción:	Tabla referente a la creación de las colaboraciones la cual contiene el identificador de la colaboración, nombre, objetivos, descripción, propietario, OA en creación, pizarra virtual y la plantilla utilizada.	
Atributo:	Tipo:	Descripción:
<ul style="list-style-type: none"> • Collabid (llave primaria). • Collab_name. • Collab_objective. • Collab_description. • Collab_owner. • Collab_pif. • Collab_board. • Collab_template. 	<ul style="list-style-type: none"> • Smalling. • Text. • Text. • Text. • Smalling. • Smalling. • Smalling. • Smalling. 	<ul style="list-style-type: none"> • Identificador de la colaboración. • Nombre de la colaboración. • Objetivo de la colaboración. • Descripción de la colaboración. • Propietario de la colaboración. • OA creado por la colaboración. • Pizarra de la colaboración. • Plantilla empleada en la colaboración.

Nombre:	collaboration_user.
Descripción:	Tabla referente a los usuarios pertenecientes a la colaboración, la cual contiene el

	identificador de la colaboración y del usuario que forma parte de la misma.	
Atributo:	Tipo:	Descripción:
<ul style="list-style-type: none"> • Collab_id (llave primaria). • User_id (llave primaria). 	<ul style="list-style-type: none"> • Smalling. • Smalling. 	<ul style="list-style-type: none"> • Identificador de la colaboración. • Identificador del usuario.

Nombre:	cwhiteboard.	
Descripción:	Tabla referente a la pizarra virtual asociada a cada colaboración la cual contiene un identificador, ancho, alto, nombre e imagen de fondo.	
Atributo:	Tipo:	Descripción:
<ul style="list-style-type: none"> • Id (llave primaria). • Width. • Height. • name. • Bgfile. 	<ul style="list-style-type: none"> • Smalling. • Smalling. • Smalling. • Text. • Text. 	<ul style="list-style-type: none"> • Identificador de la pizarra. • Ancho de la pizarra. • Alto de la pizarra. • Nombre de la pizarra. • Dirección de la imagen de fondo.

Nombre:	cuserwhiteboard.	
Descripción:	Tabla referente a la configuración personal de cada uno de los usuarios que poseen acceso a la pizarra de una colaboración.	
Atributo:	Tipo:	Descripción:
<ul style="list-style-type: none"> • id (llave primaria). • Userid (llave primaria). • Width. • Height. 	<ul style="list-style-type: none"> • Smalling. • Smalling. • Smalling. 	<ul style="list-style-type: none"> • Identificador de la pizarra virtual. • Identificador del usuario. • Ancho de la pizarra. • Alto de la pizarra.

Nombre:	sf_simple_forum_forum.
----------------	-------------------------------

Descripción:	Tabla referente al área de discusión para la creación colaborativa de los OA la cual contiene un identificador, nombre, descripción, fecha de creación, fecha de actualización, último comentario realizado, número de comentarios, número de tópicos y la colaboración asociada.	
Atributo:	Tipo:	Descripción:
<ul style="list-style-type: none"> • Id (llave primaria). • Name. • Description. • Created_at. • Update_at. • Latest_post_id. • Nb_posts. • Nb_topics. • Collaboration_id. 	<ul style="list-style-type: none"> • Smalling. • Text. • Text. • Timestamp. • Timestamp. • Smalling. • Smalling. • Smalling. • Smalling. 	<ul style="list-style-type: none"> • Identificador del área de discusión. • Nombre del área de discusión. • Descripción del área de discusión. • Fecha de creación. • Fecha de actualización. • Último comentario realizado. • Número de comentarios. • Número de tópicos. • Colaboración asociada.

Nombre:	sf_simple_forum_topic.	
Descripción:	Tabla referente a los tópicos creados en el área de discusión la cual contiene un identificador, título, área de discusión, fecha de creación, fecha de actualización, último comentario realizado, número de comentarios, número de tópicos y el usuario creador del tópico.	
Atributo:	Tipo:	Descripción:
<ul style="list-style-type: none"> • Id (llave primaria). • Título. • Forum_id. • Created_at. • Update_at. • Latest_post_id. • Nb_posts. • Nb_views. • User_id. 	<ul style="list-style-type: none"> • Smalling. • Text. • Smalling. • Timestamp. • Timestamp. • Smalling. • Smalling. • Smalling. • Smalling. 	<ul style="list-style-type: none"> • Identificador del tópico. • Título del tópico. • Identificador del área de discusión. • Fecha de creación. • Fecha de actualización. • Último comentario realizado. • Número de comentarios. • Cantidad de veces que se ha visto el tópico.

		<ul style="list-style-type: none"> • Usuario creador del tópico.
--	--	---

Nombre:	sf_simple_forum_post.	
Descripción:	Tabla referente a los comentarios creados en cada tópico del área de discusión para la creación colaborativa de los OA la cual contiene un identificador, título, contenido, fecha de creación, tópico asociado, usuario del comentario, área de discusión asociada, nombre del autor y recurso que puede subir.	
Atributo:	Tipo:	Descripción:
<ul style="list-style-type: none"> • Id (llave primaria). • Title. • Content. • Created_at. • Topic_id. • User_id. • Forum_id. • Author_name. • Resource. 	<ul style="list-style-type: none"> • Smalling. • Text. • Text. • Timestamp. • Smalling. • Smalling. • Smalling. • Text. • Text. 	<ul style="list-style-type: none"> • Identificador del comentario. • Título del comentario. • Contenido del comentario. • Fecha de creación. • Tópico asociado. • Usuario del comentario. • Área de discusión asociada. • Nombre del autor. • Recurso asociado.

Nombre:	sf_simple_forum_topic_view.	
Descripción:	Tabla referente a la cantidad de veces que un usuario visita un tópico de discusión.	
Atributo:	Tipo:	Descripción:
<ul style="list-style-type: none"> • User_id (llave primaria). • Topic_id. • Created_at. 	<ul style="list-style-type: none"> • Smalling. • Smalling. • Timestamp. 	<ul style="list-style-type: none"> • Identificador. • Tópico visitado. • Fecha de creación.

Nombre:	polls_table.	
Descripción:	Tabla referente a las votaciones a realizar para la eliminación de una colaboración la cual contiene un identificador, título, fecha de creación, propietario, colaboración asociada y	

	fecha de terminación.	
Atributo:	Tipo:	Descripción:
<ul style="list-style-type: none"> • Id (llave primaria). • Title. • Created_at. • Propietario. • Collab. • Finished_on. 	<ul style="list-style-type: none"> • Smalling. • Text. • Timestamp. • Smalling. • Smalling. • Timestamp. 	<ul style="list-style-type: none"> • Identificador de la votación. • Título de la votación. • Fecha de creación. • Propietario de la colaboración. • Colaboración asociada. • Fecha de terminación.

Nombre:	polls_answers_table.	
Descripción:	Tabla referente a las respuestas posibles de las votaciones a realizar para la eliminación de una colaboración la cual contiene un identificador, votación asociada, nombre, votos y fecha de creación.	
Atributo:	Tipo:	Descripción:
<ul style="list-style-type: none"> • Id (llave primaria). • Poll_id. • Created_at. • Name. • Votes. 	<ul style="list-style-type: none"> • Smalling. • Smalling. • Timestamp. • Text. • Smalling. 	<ul style="list-style-type: none"> • Identificador de la respuesta. • Votación asociada. • Fecha de creación. • Cuerpo de la respuesta. • Número de votos.

Nombre:	polls_users_answers_table.	
Descripción:	Tabla referente a las votaciones efectuadas por los usuarios de las posibles respuestas de las votaciones a realizar para la eliminación de una colaboración la cual contiene un identificador, votación asociada, identificador de la respuesta, usuario asociado y fecha de creación.	
Atributo:	Tipo:	Descripción:
<ul style="list-style-type: none"> • Id (llave primaria). • Poll_id. • Created_at. 	<ul style="list-style-type: none"> • Smalling. • Smalling. • Timestamp. 	<ul style="list-style-type: none"> • Identificador del voto. • Votación asociada. • Fecha de creación.

<ul style="list-style-type: none">• Answer_id.• User_id.	<ul style="list-style-type: none">• Smalling.• Smalling.	<ul style="list-style-type: none">• Respuesta asociada.• Usuario asociado.
---	---	---

Glosario de términos.

Apache: servidor HTTP de páginas Web.

Aplicación: es el programa que el usuario activa para trabajar en el ordenador. Existen muchos programas de ordenador que pueden clasificarse como aplicación. Generalmente se les conoce como software.

Especificación: es un documento técnico que describe los componentes (parte estática) y el comportamiento (parte dinámica) de un determinado sistema.

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

HTTP (HyperText Transfer Protocol): es 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.

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*.

Metadatos: conjunto de atributos o elementos necesarios para describir un recurso en cuestión.

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.

Open Source: *código abierto*. Manera de nombrar también a las aplicaciones desarrolladas bajo el amparo de un producto con licencia GPL.

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

SQL: Lenguaje de consulta estructurado.

TICs: Tecnologías de la Información y las Comunicaciones.

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.