

Universidad de las Ciencias Informáticas  
Facultad 4



# **Versión 2.0 del módulo para la gestión de ejercicios de la Plataforma Educativa Zera.**

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

**Autores:**

Leticia Martinez Rosales

José Antonio González Mazón

**Tutores:**

Ing. Cindy Santos Salgado

Ing. Adrián Sánchez García

La Habana, 2012  
“Año 54 de la Revolución”



## Declaración de Autoría

Declaramos que somos los únicos autores del trabajo “**Versión 2.0 del módulo para la gestión de ejercicios de la Plataforma Educativa Zera**” y autorizamos a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

\_\_\_\_\_  
**Autor: Leticia Martinez Rosales**

\_\_\_\_\_  
**Autor: José Antonio González  
Mazón**

\_\_\_\_\_  
**Tutor: Ing. Cindy Santos Salgado**

\_\_\_\_\_  
**Tutor: Ing. Adrián García Sánchez**



## *Agradecimientos*

Agradezco:

A mi mamita bella Dalmis, por ser pieza clave durante mi carrera, por ser madre y amiga, por su apoyo incondicional en todos los momentos difíciles.

Al amor de mi vida mi papá Eros, por el sacrificio de tantos años y por haber depositado en mí toda la confianza del mundo, pipo yo sé que soy uno de tus grandes orgullos, gracias por apoyarme y ser mi papito lindo.

A mis amores, mis nenes, la luz de mis ojos, mi hermanito Leonardo y mi sobrina Elizabeth, por ser la alegría de mi vida, a ustedes los quiero mucho.

A mis abuelos Milva, Raúl e Iلسis, por toda la sabiduría que han sabido trasmitirme, por malcriarme, cuidarme y por qué si ellos hoy no sería la persona que soy.

En especial a papi Moro aunque hoy no está conmigo sé que este era su gran sueño y siempre me apoyo, a ti que estás conmigo en este momento especial, tu sueño se ha hecho realidad.

A toda mi gran familia, por ser tan especial y existir, a mis tíos, tías, primos por apoyarme siempre.

A mi novio Wilson, por cuidarme y entenderme durante la carrera, gracias a ti también, hoy me hago ingeniera.

A mis compañeros y amigos, por los ratos compartidos, por haberme ayudado y soportado a lo largo de mi carrera, por apoyarme en las buenas y en las malas.

A mis tutores Cindy y Adrián, a mi compañero de tesis, si la ayuda de ustedes esto no sería posible.

A todos aquellos que pensaron que no lo lograría, pues me dieron las fuerzas suficientes para continuar.

A todos, muchísimas gracias, ustedes forman parte de este logro.

**Lety**



Agradezco:

A mis Compañeros en esta tarea Adrián, Cindy y Leticia sin los cuales hubiera sido imposible terminarla.

A mis hermanos Samy y Lester, por ser los mejores hermanos del mundo.

Al Mejor y a mi tía Lena por todo su amor y ánimo que han hecho de mí una mejor persona.

A mi flaca Cora por todo el cariño y el apoyo que me ha dado los últimos 8 años.

Al Ore por ser un gran amigo y estar presente en los momentos más difíciles.

A la gente del grupo 2 y 3.

A Mayi, Claudia, Willy, Dunet y Leo por ser tan buenos amigos.

A Yerandi y el Yoa por su ayuda y paciencia con todas mis dudas.

En fin a la gente del proyecto alfaomega.

**José**



## *Dedicatoria*

A mis padres por quererme y apoyarme siempre.

A mi hermano y a mi sobrina por exigirme buenas notas y ser mi alegría.

A la memoria de mi abuelo Moro, esta es una dedicatoria especial por el gran amor y apoyo  
que siempre tuve de él.

“Papá, sé que me escucharás porque no te has ido y nunca te irás. No has muerto, y nunca  
lo harás porque no se muere cuando el corazón deja de latir, se muere cuando en los  
recuerdos se deja de existir y tú estás presente, estas aquí, estás vivo, para todos y para  
mí.”

**Lety**

A mis padres, por ser los gigantes que me dejaron sentarme en sus hombros para poder ver  
más y mejor.

**Jose**



## Resumen

El progresivo avance tecnológico, facilita que la educación a distancia avance hacia un modelo de educación soportado por las tecnologías en Internet, por lo que ha sido necesaria la informatización de procesos educativos que durante años se realizaron de forma tradicional. Zera es una Plataforma Educativa desarrollada en la Universidad de las Ciencias Informáticas (UCI), que permite la gestión del aprendizaje. El objetivo de este trabajo es desarrollar una nueva versión del módulo para la gestión de ejercicios de Zera, permitiendo así una mejor interoperabilidad, durabilidad y portabilidad de la información gestionada. Para lograrlo, se realizó un estudio de las principales tendencias en el proceso de creación de ejercicios. El uso de la metodología RUP permitirá un proceso controlado durante las diferentes fases de desarrollo, obteniendo los artefactos propuestos por dicha metodología. Esta aplicación utiliza el framework de desarrollo Symfony. Se hará uso del servidor de base de datos PostgreSQL y se utiliza Apache como servidor web. Finalmente se obtiene el análisis, diseño e implementación de una nueva versión del módulo para la gestión de ejercicios que incluye la gestión de nuevas tipologías, creación de baterías de ejercicios, importación y exportación de paquetes de ejercicios, así como la exportación a formato pdf de los mismos.

**Palabras claves:** plataforma educativa, ejercicios, interoperabilidad



## Contenido

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>FUNDAMENTACIÓN TEÓRICA</b> .....	<b>5</b>
1.1 INTRODUCCIÓN.....	5
1.2 CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA.....	5
<b>1.2.1 Plataformas educativas</b> .....	5
<b>1.2.2 E-learning</b> .....	6
<b>1.2.3 Interoperabilidad</b> .....	6
<b>1.2.4 Portabilidad</b> .....	6
<b>1.2.5 Reusabilidad</b> .....	7
1.3 ESTÁNDARES QUE PERMITEN EL INTERCAMBIO DE EJERCICIOS.....	7
<b>1.3.1 Estándares para la interoperabilidad</b> .....	7
<b>1.3.2 Estándares para los metadatos</b> .....	11
<b>1.3.3 Estándar para empaquetamiento</b> .....	12
1.4 SISTEMAS SIMILARES.....	13
1.5 HERRAMIENTAS, METODOLOGÍAS Y TECNOLOGÍAS PARA EL DISEÑO E IMPLEMENTACIÓN DE LA VERSIÓN 2.0 DEL MÓDULO PARA LA GESTIÓN DE EJERCICIOS.....	14
<b>1.5.1 Metodología de desarrollo de software</b> .....	15
<b>1.5.2 Lenguaje Unificado de Modelado (UML)</b> .....	16
<b>1.5.3 Herramienta para el modelado</b> .....	17
<b>1.5.4 Herramientas para el diseño de interfaz de usuario (IU)</b> .....	17
<b>1.5.5 Herramientas y lenguajes de Desarrollo</b> .....	18
<b>1.5.6 Lenguajes de script del lado del cliente</b> .....	18
<b>1.5.7 Lenguajes del lado del servidor</b> .....	19
<b>1.5.8 Gestor de Bases de Datos</b> .....	20
<b>1.5.9 Servidor Web</b> .....	21
<b>1.5.10 Entorno de Desarrollo Integrado (IDE)</b> .....	21
1.6 CONCLUSIONES PARCIALES.....	22
<b>CARACTERÍSTICAS DEL SISTEMA</b> .....	<b>23</b>
2.1. INTRODUCCIÓN.....	23
2.2. MODELO DE DOMINIO.....	23
<b>2.2.1. Análisis de los conceptos del dominio</b> .....	23
<b>2.2.2. Diagrama del Modelo de Dominio</b> .....	24
2.3. DESCRIPCIÓN DEL SISTEMA PROPUESTO.....	25
<b>2.3.1. Especificación de requisitos</b> .....	26
2.4. MODELO DE CASOS DE USO DEL SISTEMA.....	30
<b>2.4.2. Diagrama de casos de uso del sistema</b> .....	32
2.5. CONCLUSIONES PARCIALES.....	39
<b>ANÁLISIS Y DISEÑO</b> .....	<b>40</b>
3.1 INTRODUCCIÓN.....	40
3.2 MODELO DE ANÁLISIS.....	40
<b>3.2.1 Diagramas de clases del análisis</b> .....	40
<b>3.2.2 Diagramas de colaboración del análisis</b> .....	41
3.3 PATRÓN ARQUITECTÓNICO MODELO – VISTA – CONTROLADOR EN SYMFONY.....	42
3.4 APLICACIÓN DE LOS PATRONES DE DISEÑO EN SYMFONY.....	43
3.5 MODELO DE DISEÑO.....	43
<b>3.5.1 Diagrama de clases del diseño</b> .....	43
3.6 DISEÑO DE LA BASE DE DATOS.....	45
<b>3.6.1 Descripción de las tablas</b> .....	46
3.7 CONCLUSIONES PARCIALES.....	50



<b>IMPLEMENTACIÓN Y PRUEBA .....</b>	<b>51</b>
4.1 INTRODUCCIÓN .....	51
4.2 MODELO DE IMPLEMENTACIÓN .....	51
<b>4.2.1 Diagrama de componentes.....</b>	<b>51</b>
4.3 PRUEBAS DE SOFTWARE.....	52
<b>4.3.1 Niveles de Prueba .....</b>	<b>52</b>
<b>4.3.2 Métodos de Prueba .....</b>	<b>53</b>
4.4 DISEÑO DE CASOS DE PRUEBA .....	53
<b>4.5 RESULTADOS OBTENIDOS .....</b>	<b>53</b>
4.6 CONCLUSIONES PARCIALES .....	55
<b>CONCLUSIONES GENERALES.....</b>	<b>56</b>
<b>RECOMENDACIONES .....</b>	<b>57</b>
<b>BIBLIOGRAFÍA .....</b>	<b>58</b>





## Índice de Figuras.

Figura 1 Estructura del modelo ASI (6) .....	8
Figura 2 Modelo de dominio.....	25
Figura 3 Patrón CRUD .....	30
Figura 4 Adición.....	31
Figura 5 Generalización/Especialización .....	31
Figura 6 Diagrama de casos de uso del sistema.....	32
Figura 7 Diagrama de clases del análisis CU Gestionar Batería .....	41
Figura 8 CU Gestionar Batería. Sección Induir .....	41
Figura 9 DC CU Gestionar Batería. Sección Modificar .....	42
Figura 10 DC CU Gestionar Batería. Sección Ver.....	42
Figura 11 DC CU Gestionar Batería. Sección Eliminar.....	42
Figura 12 DCD CU Gestionar Batería.....	45
Figura 13 Modelo de la Base de Datos.....	46
Figura 14 Diagrama de Componentes. Batería.....	52

## Índice de Tablas.

Tabla 1 Comparación de sistemas similares.....	14
Tabla 2 Descripción de actores del sistema.....	32
Tabla 3 Descripción del CU Gestionar Batería .....	39
Tabla 4 Estereotipos y sus características. ....	44
Tabla 5 Descripción: tabla tb_diaporam. ....	47
Tabla 6 Descripción: tabla tb_animation. ....	48
Tabla 7 Descripción: tabla tb_sound. ....	48
Tabla 8 Descripción: tabla tb_video. ....	49
Tabla 9 Descripción: tabla tb_picture. ....	50
Tabla 10 Resultados de las pruebas internas.....	54
Tabla 11 Resultados de las pruebas cruzadas.....	54



## Introducción

El desarrollo de las Tecnologías de la información y las Comunicaciones (TIC) en la segunda mitad del siglo XX, garantizó que se experimentasen cambios en diversos sectores de la sociedad. La esfera educacional no quedó exenta de estos cambios y surgieron nuevas formas de llevar, a un número mayor de personas, una educación de calidad.

La combinación de tres factores: profesor, alumno y computadora abrió nuevos caminos, métodos de trabajo e ideas de acción en la pedagogía. Una de ellas, es la educación a distancia mediada por el uso de las tecnologías. En ella se destaca el uso de la computadora como elemento importante en el perfeccionamiento de la enseñanza, su evolución ha sido favorecida por los avances tecnológicos y pedagógicos. En la actualidad muchas instituciones educativas y empresas buscan su adopción como una forma de satisfacer las demandas de los docentes y mantenerse en el liderazgo de las tecnologías. En este entorno es que surgen las plataformas educativas las cuales constituyen un espacio de aprendizaje personalizado para el estudiante.

Los exámenes y las evaluaciones son un factor fundamental en el proceso de enseñanza y, por ello, en las plataformas educativas, el proceso de gestión de ejercicios tiene gran importancia. Para garantizar su intercambio, varias plataformas han comenzado a estandarizar la manera de gestionar los mismos.

Una manera de lograr la estandarización del proceso de gestión de ejercicios es siguiendo la especificación IMS-QTI (*IMS Question and Test Interoperability*), esta describe la forma de representar preguntas individuales y gestionar evaluaciones o exámenes completos.

En el Centro de Tecnologías para la Formación (FORTES) de la Universidad de las Ciencias Informáticas (UCI), se desarrolla Zera, una plataforma educativa basada en las estructuras de las escuelas. Esta plataforma es capaz de adaptarse a los programas de estudio de los centros educacionales de la enseñanza media superior, siguiendo el concepto de hiperentornos de aprendizaje.

Zera posee un módulo para la gestión de ejercicios que posibilita la creación de exámenes para evaluar el aprendizaje de los estudiantes. Con el fin de lograr una interoperabilidad de los ejercicios gestionados por la plataforma se utiliza la especificación IMS-QTI versión 2.0.

En la actualidad, las prestaciones de la plataforma Zera como producto dirigido a la gestión de ejercicios para el aprendizaje, resultan insuficientes atendiendo a que:

- Brinda **soporte incompleto** a la especificación IMS-QTI en su versión 2.0 por no responder a todas las interacciones propuestas por esta, limitando el proceso de



importación de paquetes con ejercicios en este formato, que respondan a interacciones no tomadas en cuenta en la plataforma.

- Presentan insuficiente **reusabilidad** dada su incapacidad de agrupar y/o clasificar los ejercicios de acuerdo a características comunes.
- Reducida **durabilidad** causada por la dependencia de la plataforma de las tecnologías. La incapacidad del módulo para la gestión ejercicios de exportar los mismos independiente de la plataforma.
- Escasa **portabilidad** debido a la imposibilidad de gestionar ejercicios desde otros subsistemas y módulos del sistema que lo requieren.
- Limitado **nivel de búsqueda y selección** de ejercicios en el proceso de exportación. El criterio de búsqueda en la selección de ejercicios a exportar se reduce a la cantidad máxima de elementos en el paginado, impidiendo seleccionar un mayor número de ejercicios, o utilizar distintos criterios de búsqueda para la selección de ejercicios a exportar.
- Escasa **interoperabilidad** en la fase inicial de la creación del módulo no se tuvo en cuenta la existencia de texto enriquecido como elementos de los ejercicios y la asociación de sonidos, videos, diaporamas y animaciones como medias de ayuda.

Las deficiencias identificadas, se resumen en el **problema a resolver** mediante la siguiente interrogante: ¿Cómo permitir la interoperabilidad, reusabilidad y durabilidad de la información gestionada en la nueva versión del módulo para la gestión de ejercicios en la Plataforma Educativa Zera?

Se define como **Objetivo General**: Desarrollar una nueva versión del módulo para la gestión de ejercicios incorporando nuevas funcionalidades, de manera que se permita la interoperabilidad, reusabilidad y durabilidad de la información gestionada por el mismo en la Plataforma Educativa Zera.

El **objeto de estudio** de la problemática antes expuesta es el siguiente: Proceso de gestión de ejercicios en plataformas y software educativo bajo la especificación IMS-QTI v2.0.

El **campo de acción** se enmarca en el proceso de gestión de ejercicios bajo la especificación IMS-QTI v2.0, en la Plataforma Educativa Zera.



### Objetivos Específicos:

1. Elaborar el marco teórico conceptual a partir del estudio de las principales tendencias y herramientas existentes relacionadas con la gestión de ejercicios a nivel mundial.
2. Desarrollar el análisis y diseño del módulo para la gestión de ejercicios.
3. Realizar la implementación de la nueva versión del módulo para la gestión de ejercicios.
4. Validar la implementación de la nueva versión del módulo para la gestión de ejercicios de la Plataforma Educativa Zera.

Para guiar la investigación se plantea la siguiente **idea a defender**: Con el desarrollo de la nueva versión del módulo para la gestión de ejercicios de la Plataforma Educativa Zera se logrará la interoperabilidad, durabilidad y reusabilidad de la información gestionada.

Para dar cumplimiento a los objetivos se plantean un grupo de **tareas de investigación**:

- Identificación de las tendencias y herramientas usadas en el desarrollo de los sistemas de gestión de ejercicios.
- Análisis de las nuevas funcionalidades.
- Elaborar el análisis y diseño de la nueva versión del módulo para la gestión de ejercicios de la Plataforma Educativa Zera.
- Desarrollo de la nueva versión del módulo para la gestión de ejercicios.
- Implementación de las nuevas funcionalidades identificadas.
- Integrar la nueva versión del módulo para la gestión de ejercicios a la Plataforma Educativa Zera.
- Validación del módulo para la gestión de ejercicios.

**Los métodos de investigación** que soportan la siguiente investigación es una combinación de los **métodos teóricos y empíricos**. Los empleados son los que se muestran a continuación:

### Métodos Teóricos

**Histórico – Lógico**: para conocer el desarrollo y evolución de Plataformas Educativas que permiten la gestión del aprendizaje, y otros sistemas similares para la gestión de ejercicios.

**Analítico – Sintético**: para la revisión de bibliografías y materiales relacionados, así como la información de plataformas educativas.



**Modelación:** para la generación de los artefactos que permitan lograr una mejor comprensión entre el equipo de desarrollo y las personas relacionadas con el fenómeno en la realidad.

### **Métodos Empíricos**

**Observación:** para conocer la esencia del problema planteado y el objeto de investigación, así como las circunstancias o el ambiente que rodean los procesos de gestión de ejercicios en plataformas similares.

El presente trabajo está estructurado en 4 capítulos

#### ***Capítulo 1: Fundamentación Teórica***

Se aborda de manera general los aspectos teóricos relacionados con la especificación IMS-QTI, así como el estudio de aquellas plataformas que brindan soporte a dicha especificación, también se caracterizan las herramientas que serán utilizadas en el análisis, diseño e implementación de la versión 2.0 del módulo para la gestión de ejercicios.

#### ***Capítulo 2: Características del sistema***

En este capítulo se realiza un estudio de las necesidades del sistema. También se identificarán y se describirán las funcionalidades a implementar. Además, se hará una propuesta de solución, que responda a los requisitos identificados.

#### ***Capítulo 3: Análisis y Diseño del módulo para la gestión de ejercicios.***

Contiene, los flujos de análisis y diseño donde se verificará la solución a los objetivos propuestos inicialmente. Se presentan los diagramas de clases de análisis, interacción, clases del sistema y el diseño de la base de datos.

#### ***Capítulo 4: Implementación y pruebas del módulo para la gestión de ejercicios.***

Este capítulo está dedicado a la fase de Implementación donde se generará el diagrama de componente. Además, se realizarán las pruebas necesarias para validar el producto.



## Fundamentación Teórica

### 1.1 Introducción

El uso de la web como medio, ayuda a soportar el proceso de enseñanza-aprendizaje tanto en ámbitos presenciales como a distancia. Existen servicios y aplicaciones que son empleadas por profesores y alumnos para el intercambio de ideas y que permiten su acceso, desde cualquier lugar y momento, a contenidos y actividades de aprendizaje. Para posibilitar dicho intercambio existe una especial atención en anotar técnicas, métodos y recursos educativos. Así, surgen iniciativas encaminadas a proponer estándares que indican cómo marcar elementos educativos por medio de la definición de sus atributos.

Este capítulo pretende dar una visión general de los conceptos del dominio del problema. Después presenta las especificaciones que serán utilizadas como parte de la presente investigación. Se comparan algunas de las herramientas a nivel internacional, para definir las bases de la nueva versión del módulo para la gestión de ejercicios de la Plataforma Educativa Zera. Seguidamente, se mencionan las herramientas que serán utilizadas y algunas de sus características. Para finalizar, se esbozan las conclusiones sobre este tema.

### 1.2 Conceptos asociados al dominio del problema

#### 1.2.1 Plataformas educativas

La aplicación de las tecnologías en la educación, así como los cambios en los modelos pedagógicos, se han visto plasmados en los entornos virtuales de enseñanza y aprendizaje (EVEA). Los EVEA se apoyan en sistemas informáticos e incluyen herramientas adaptadas a las necesidades de la institución para la que se desarrollan. Estos sistemas reciben el nombre de plataformas educativas (1).

Las plataformas educativas posibilitan estimular la idea de cooperación e interacción, mediante el uso de herramientas colaborativas que aumentan el aprendizaje en los estudiantes. Además, fortalecen el uso de las Tecnologías de la Información y las Comunicaciones en los docentes.

La implementación de una plataforma educativa para la administración de cursos permite la autonomía de producción y publicación en la red de recursos y contenidos por parte de los participantes. El docente, en correspondencia con los objetivos del programa pone a disposición de sus estudiantes, las actividades, bibliografía y evaluaciones. De manera análoga, pero con otro nivel de autonomía, el estudiante puede acceder a los contenidos y al desarrollo de las actividades propuestas.



### 1.2.2 E-learning

Existen investigaciones científicas que abren el espectro del *e-learning* a cualquier proceso relacionado con la educación y las tecnologías, como la definición de la Sociedad Americana de Capacitación y Desarrollo que propone:

... “*Término que cubre un amplio grupo de aplicaciones y procesos, tales como aprendizaje basado en web, aprendizaje basado en ordenadores, aulas virtuales y colaboración digital. Incluye entrega de contenidos vía Internet, intranet/extranet, audio y vídeo grabaciones, transmisiones satelitales, TV interactiva, CD-ROM y más*”. (2)

Varios autores definen el uso de *e-learning* exclusivamente al ámbito de Internet, tal es el caso de la siguiente definición: “el uso de tecnologías Internet para la entrega de un amplio rango de soluciones que mejoran el conocimiento y el rendimiento. Está basado en tres criterios fundamentales:

1. El *e-learning* trabaja en red, lo que lo hace capaz de ser instantáneamente actualizado, almacenado, recuperado, distribuido y permite compartir instrucción o información.
2. Es entregado al usuario final a través del uso de ordenadores utilizando tecnología estándar de Internet.
3. Se enfoca en la visión más amplia del aprendizaje que van más allá de los paradigmas tradicionales de capacitación”. (2)

Por las facilidades de Internet, el *e-learning* se ha consolidado como un proceso de enseñanza-aprendizaje que garantiza un mayor grado de formalización gracias al uso de métodos y herramientas de calidad. Además, facilita la reutilización de materiales docentes ya desarrollados, dentro de las aplicaciones y también entre aplicaciones diferentes.

### 1.2.3 Interoperabilidad

La IEEE<sup>1</sup> define la interoperabilidad como la habilidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada. (3)

### 1.2.4 Portabilidad

Facilidad con que puede ser un sistema o componente transferido de un hardware o medio ambiente software a otro. (3)

---

<sup>1</sup> Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, por sus siglas en inglés), es la organización profesional a nivel mundial dedicada al avance tecnológico para el servicio de la humanidad. También está dedicada al avance teórico y práctico de la ingeniería eléctrica, electrónica e informática. (42)



### 1.2.5 Reusabilidad

Lluis Codina define la reusabilidad como: la propiedad que presenta la información digital de ser fácilmente seleccionable, recuperable, transmisible y transportable. (4)

### 1.3 Estándares que permiten el intercambio de ejercicios

En la actualidad existe un marcado interés con el motivo de asegurar que los ambientes e-learning sean accesibles, adaptables, durables y reutilizables. Esto ha propiciado que diferentes instituciones desarrollen propuestas de técnicas y métodos, dichas propuestas son conocidas como especificaciones.

Según el diccionario de la Real Academia de la Lengua Española un estándar es lo “que sirve como tipo, modelo, norma, patrón o referencia” (5). En el campo técnico, la estandarización es el proceso por el cual se establecen normas comúnmente aceptadas que permiten la cooperación de diferentes empresas o instituciones sin menoscabar su posibilidad de competir.

#### 1.3.1 Estándares para la interoperabilidad

Para beneficiarse del esfuerzo requerido durante la creación de recursos educativos, disímiles instituciones se han propuesto desarrollar especificaciones y estándares orientados hacia una forma común de identificar, definir y comunicar los recursos involucrados en un entorno *e-learning*. La especificación IMS-QTI ha sido uno de los intentos más reconocidos por lograr la interoperabilidad de ejercicios, entre sistemas gestores de aprendizaje.

Es cierto que las especificaciones IMS aún se encuentran en desarrollo y que son solo especificaciones, sin embargo permiten que los LMS empiecen a comunicarse entre sí, posibilitando que la portabilidad de contenidos entre plataformas sea una realidad.

#### IMS Question and Test Interoperability (IMS-QTI).

Con el surgimiento de herramientas y sistemas que contribuyeran al proceso de enseñanza-aprendizaje, mediante el uso de computadoras, se hizo cada vez más importante la posibilidad de intercambiar ejercicios entre aplicaciones informáticas. En 1999 el consorcio IMS comienza el desarrollo de un grupo de normas que estandarizarían el proceso de gestión de ejercicios dentro de los sistemas.

A lo largo de la historia de esta especificación han sido tres las versiones más significativas de IMS-QTI debido a los cambios revolucionarios realizados en ellas: v1.2, v2.0 y v2.1 (todavía en borrador). En el caso de la **versión 1.2** se introducen el modelo de datos





conocido como ASI (evaluación, sección y elemento, por sus siglas en inglés) en el cual se describen las evaluaciones utilizando estructuras de metadatos.

Las evaluaciones (*assessment*) están formados por elementos (*items*) que a su vez están contenidos en secciones (*sections*). Pueden ensamblarse por bloques de elementos que estén lógicamente relacionados. Una evaluación puede contener una o más secciones.

Los elementos contienen toda la información requerida para componer montar, evaluar y brindar una retroalimentación del elemento. La diferencia entre elementos y preguntas está dada en que los elementos contienen una pregunta, la plantilla de la información, el procesador de información de la respuesta, las pistas correspondientes y la retroalimentación dada a una solución.

Las secciones son elementos que permiten representar dentro de la evaluación diferentes tópicos o temas y a su vez permiten controlar la forma y el orden en que serán representados en la evaluación.

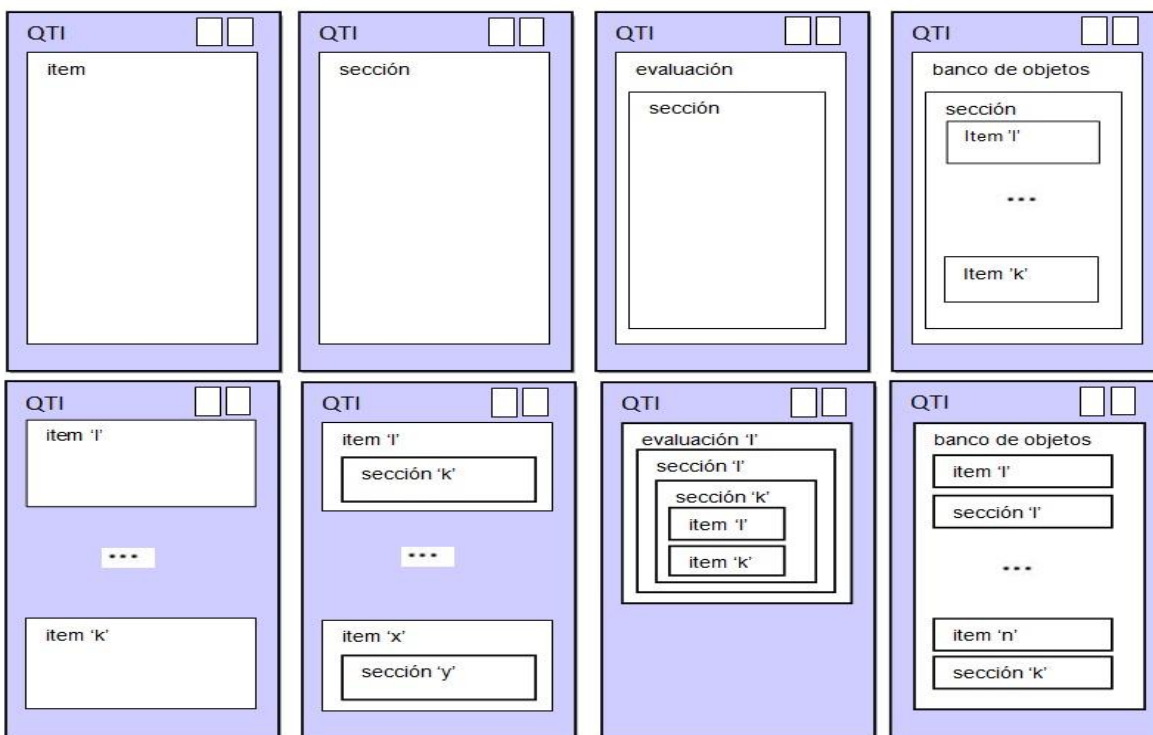


Figura 1 Estructura del modelo ASI (6)

Desde la aparición de IMS-QTI v1.2 y con el desarrollo de nuevas especificaciones como el *IMS Content Packaging (IMS CP)*, *IMS Simple Sequencing (IMS SS)* e *IMS Learning Design (IMS LD)* surgió la necesidad de hacer compatible la especificación IMS-QTI a las nuevas iniciativas. Con IMS-QTI v2.0 (2005) de la especificación, se inicia la resolución de los



problemas de la versión anterior y de compatibilidad con otras especificaciones. Esta versión se concentró solo en las preguntas individuales para simplificar el proceso de adopción y permitir un trabajo razonable, pero no actualizó aquellas partes referidas a la composición de dichas preguntas, es decir, la creación de exámenes completos. (6)

En la **versión 2.0** las preguntas individuales (*assessmentItem*) incluyen toda la información necesaria para su presentación al alumno y su corrección automática. Toda la información relativa a la presentación se agrupa en el cuerpo (*itemBody*) de las preguntas.

En el caso de IMS-QTI v2.0, no está contemplado el concepto de tipo de pregunta y existe, en su lugar, el concepto de interacción. Las interacciones permiten al profesor especificar las herramientas que tendrá el alumno disponible para poder construir la respuesta. A continuación se muestran los tipos de interacciones posibles que pueden utilizarse dentro de una evaluación.

Las interacciones simples son aquellas en las cuales la corrección de las mismas se realiza basándose en la selección de una o varias opciones disponibles. Las interacciones que pertenecen a esta categoría son: (7)

- ✓ *ChoiceInteraction*: muestra un conjunto de posibles opciones al estudiante, este podrá seleccionar una o varias respuestas (ejemplo: Selección Simple, Selección Múltiple)
- ✓ *OrderInteraction*: permite al alumno reordenar el conjunto de soluciones proporcionada (ejemplo: Ordenar Texto, Ordenar Media).
- ✓ *AssociateInteraction*: presenta un conjunto de opciones y permite crear asociaciones por parejas entre dichas opciones.
- ✓ *MatchInteraction*. muestra dos conjuntos de opciones y le permite crear pares de asociaciones entre ellas (ejemplo: Enlazar).

En la categoría interacciones de texto se encuentran aquellas donde la respuesta que construirá el alumno puede ser una única palabra, una frase corta o un párrafo de texto completo. Estas interacciones permiten que durante el proceso de corrección se tenga en cuenta la respuesta en forma de texto construida por el alumno. Las interacciones que pertenecen a esta categoría son:

- ✓ *InlineChoiceInteraction (interacción en línea)*: se define un hueco donde se permitirá al alumno escoger entre un conjunto de opciones, donde cada una de estas opciones es una palabra o frase corta (ejemplo: Completar por desplazamiento).
- ✓ *TextEntryInteraction (interacción en línea)*: tiene como objetivo crear un hueco donde se permitirá teclear una palabra o frase corta para poder construir la respuesta



(ejemplo: Completar texto).

- ✓ *ExtendedTextInteraction*: permite al alumno construir como respuesta un párrafo de texto (ejemplo: Redactar un párrafo).
- ✓ *HottextInteraction*: permite al alumno seleccionar partes del texto que estarán resaltadas en el enunciado de la pregunta. (8)

Las interacciones gráficas tienen como elemento principal una imagen que se utilizará como fondo del enunciado y sobre la que se realizarán todas las acciones permitidas para que el alumno construya la respuesta. Las interacciones que pertenecen a esta categoría son:

- ✓ *hotspotInteraction*: se presenta al alumno como un conjunto de “puntos calientes” (*hotspot*) sobre una imagen utilizada como fondo del enunciado (ejemplo: Interacción Gráfica Simple)
- ✓ *selectPointInteraction*: el alumno debe seleccionar uno o varios puntos de una imagen utilizada como fondo del enunciado. Al contrario que en la interacción anterior, no se le presentará al alumno ninguna zona resaltada.
- ✓ *graphicOrderInteraction*: muestra un conjunto de puntos sobre una imagen que será utilizada como fondo del enunciado. El alumno deberá ordenar dichos puntos (ejemplo: Ordenar Gráfica).
- ✓ *graphicAssociateInteraction*: muestra un conjunto de zonas seleccionables sobre una imagen que será utilizada como fondo del enunciado. Permite al alumno asociar pares de puntos.
- ✓ *graphicGapMatchInteraction*: muestra un conjunto de puntos calientes sobre una imagen que será utilizada como fondo del enunciado, se le proporciona al estudiante varias opciones (Interacción Gráfica Múltiple).
- ✓ *positionObjectInteraction*: el alumno colocará una imagen que se le proporcionó sobre alguna zona de otra imagen que se empleará como fondo del enunciado.

Otros tipos de interacciones:

- ✓ *SliderInteraction*: muestra al alumno una barra deslizante que permitirá al alumno seleccionar la respuesta correcta.(ejemplo: Barra deslizante)
- ✓ *MediaInteraction*: esta interacción está diseñada para ser utilizada en conjunto con otra interacción. Tiene como objetivo controlar el número de veces que el alumno visualiza un material multimedia.
- ✓ *DrawingInteraction*: permite al alumno pintar sobre una imagen proporcionada en el enunciado. El resultado de la pregunta será la propia imagen modificada.(ejemplo: Pintar)



- ✓ *UploadInteraction*: posibilita crear una respuesta y subir un fichero desde el escritorio del estudiante hacia la herramienta (ejemplo: Subir Archivo).
- ✓ *CustomInteraction*: permite crear interacciones particulares de cada herramienta y que no puedan ser descritas mediante ninguna de las interacciones explicadas anteriormente.

IMS-QTI proporciona un conjunto de herramientas que permiten crear métodos de evaluación personalizados, incluyendo la posibilidad de forzar una determinada presentación. Sin embargo, esto propicia que el creador de la pregunta necesite conocer en detalle la especificación IMS-QTI e incluso tener conocimientos de programación.

Además, pueden crearse preguntas y no indicar como corregirlas automáticamente, en este caso, se necesita una intervención externa (ejemplo: el profesor), para realizar las correcciones adecuadas, es decir, para evaluar las preguntas. Habitualmente solo es necesario señalar cuales opciones son correctas.

IMS-QTI define una estructura de datos XML para almacenar las preguntas independientemente del sistema o la herramienta de autoría utilizada para crearlas. Esto permite, la integración de las preguntas o exámenes desarrollados con distintas herramientas en un único LMS. (6)

**IMS-QTI versión 2.1** (aparece en 2006) está actualmente en proceso de evolución en modo borrador y tanto la comunidad educativa como técnica, puede opinar sobre ella. El objetivo de esta nueva versión es seguir con el proceso de simplificación y evolución de la especificación, esta vez dando soporte a los exámenes completos y al intercambio de los resultados de los mismos. Además, también se incluye información para clarificar la compatibilidad y el uso de IMS-QTI con algunas otras de las especificaciones (ejemplo: IMS LD, IMS SS e IMS CP) ya existentes (6).

Debido a que la versión 1.2 presentó problemas a la hora de su implementación dando así paso a la versión 1.2.1 la cual experimentó grandes cambios provocando la pérdida de compatibilidad con las versiones anteriores. La 2.1 aún se encuentra en proceso de evolución en modo borrador por lo cual no se considera estable, se considera que la versión más adecuada para usar, por tanto, sería la 2.0 con el inconveniente de que no permite la creación de exámenes completos.

### 1.3.2 Estándares para los metadatos

Debido al crecimiento del uso de los metadatos en Internet, como conjunto de atributos o elementos necesarios para describir un recurso; y la necesidad de reutilizar estos en



sistemas diferentes, fue necesario estandarizar la forma de describirlos en la red.

Varias organizaciones aprovecharon las posibilidades que ofrecen los metadatos, para conocer las principales características de los recursos, y han diseñado sus propios estándares para definir la estructura de un registro de metadatos.

### **Learning Object Metadata (LOM)**

En 1998, el proyecto Ariadne desarrolló, en colaboración con IMS, la propuesta inicial de metadatos que posteriormente se convertiría en el estándar IEEE LOM. Su objetivo era desarrollar una representación de metadatos que pudiera ser empleado en un entorno multicultural y multilingüe, neutro respecto al idioma empleado, tanto en el recurso educativo como en la propia instancia de metadatos. Establece un esquema de datos conceptual que define la estructura de un registro de metadatos (denominado instancia de metadatos) para un objeto de aprendizaje. Dicha instancia describe características del objeto agrupadas en nueve categorías: general, ciclo de vida, meta-metadatos, educativas, técnicas, derechos, relaciones, anotaciones y clasificaciones. (9)

La utilización de esta estructura facilita la búsqueda, evaluación, adquisición y uso de objetos de aprendizaje por los alumnos, instructores o sistemas automatizados. Permite intercambiar y compartir dichos recursos, posibilitando la creación de catálogos e inventarios. (11)

Como se ha podido apreciar IEEE LOM especifica la semántica y la sintáctica de un conjunto mínimo de metadatos necesarios para completar, identificar, administrar, localizar y evaluar los objetos de aprendizaje. Su propósito es facilitar a profesores, alumnos y a sistemas automáticos la tarea de buscar, compartir e intercambiar objetos de aprendizaje, permitiendo el desarrollo de catálogos que contemplan la diversidad cultural e idiomática de los contextos en los que se puedan utilizar los objetos y sus metadatos.

### **1.3.3 Estándar para empaquetamiento**

#### **IMS Content Packaging (CP)**

IMS CP v1.2 describe las estructuras de datos que pueden utilizarse para intercambiar recursos entre los sistemas que desean importar, exportar, agregar y desagregar los paquetes de contenido. Permite exportar e importar los contenidos entre sistemas de aprendizaje de gestión de contenido y/o de repositorios y a su vez retener la información que describen los medios de comunicación en el paquete de contenido y la forma en que se encuentra estructurado.



La especificación IMS Content Packaging se enfoca en el empaquetamiento y distribución de recursos, pero no determina la naturaleza de los mismos. 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.

### Valoración de los estándares y especificaciones

En el presente acápite se mencionaron algunas de las características de las especificaciones y estándares que serán utilizados para el desarrollo de la nueva versión del módulo. Se emplea IMS-QTI como especificación para la interoperabilidad pues describe la forma de representación de los ejercicios. IMS CP para el empaquetamiento de contenidos y LOM para la descripción de los metadatos.

### 1.4 Sistemas similares

La educación a distancia, en los últimos años, presenta un avance progresivo hacia modelos de formación en línea soportados por la tecnología de Internet. El uso de las redes de comunicación ha evolucionado desde distintos enfoques, hasta definir modelos que son adoptados masivamente por distintas instituciones. Entre los aspectos más significativos relacionados con el desarrollo de las tecnologías en la esfera educativa se encuentra la introducción e implementación de las llamadas plataformas virtuales que tiene a Internet como medio de distribución. A continuación se muestra una tabla con los resultados obtenidos del estudio realizado a sistemas similares, centrándose en la gestión de ejercicios.

Funcionalidades	Plataformas Educativas					
	Chamillo	Atutor	Claroline	Moodle	Sakai	Dokeos
Utiliza IMS-QTI.	Si	Si	Si	Si	Si	No
Tipologías representadas.	7	8	5	11	8	8
Utiliza QTI para la exportación.	Si	Si	Si	No	Si	No
Exporta varios ejercicios.	No	No	No	Si	Si	No
Exporta a .pdf.	No	No	Si	No	No	No



Importa utilizando QTI.	Si	No	Si	No	Si	No
Banco de preguntas.	Si	Si	Si	Si	Si	Si
Niveles de retroalimentación.	Si	-	No	Parcial	Si	Si
Especificación para el empaquetado de ejercicios (IMS CP).	No	No	No	No	No	No

Tabla 1 Comparación de sistemas similares

### Valoración de las plataformas

Se elaboró una guía de observación relacionada con el proceso de gestión de ejercicios. Las plataformas estudiadas poseen funcionalidades similares a las que se necesitan implementar y que serán utilizadas como referencia en el desarrollo de la nueva versión del módulo para la gestión de ejercicios de la Plataforma Educativa Zera. Los sistemas que se tuvieron en cuenta son altamente reconocidos a nivel mundial, aún así, se detectaron ciertas deficiencias durante el proceso de creación de ejercicios: cuatro de las plataformas estudiadas no permiten exportar un conjunto de ejercicios, ninguna incluye todas las interacciones propuestas por la especificación IMS-QTI v2.0, solo una permite exportar ejercicios a formato pdf. Se observó que, de las plataformas estudiadas ninguna hace uso de la especificación IMS CP para la exportación de contenidos, limitando el intercambio de ejercicios para ser reutilizados.

### 1.5 Herramientas, metodologías y tecnologías para el diseño e implementación de la versión 2.0 del módulo para la gestión de ejercicios

En el desarrollo de la versión 2.0 del módulo para la gestión de ejercicios los cambios desde el punto de vista arquitectónico no son considerables, se ha decidido continuar con las metodologías, herramientas y lenguajes de Zera, destacando que dichas herramientas son las definidas por el proyecto debido a las características que ofrecen, siendo en su mayoría tecnologías y herramientas libres. A continuación algunas de las características de dichas herramientas.



### 1.5.1 Metodología de desarrollo de software

El desarrollo de software educativo resulta realmente complicado. El mismo incluye un conjunto de bases, herramientas y métodos que necesitan de forma absoluta el uso de una metodología, con el fin de lograr procesar a cada uno de estos elementos en lo que sería un proceso coordinado y conforme al resultado que se persigue.

Se define como metodología de desarrollo de software “un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevo software”. (10)

A continuación se definen las características que hacen de las Metodologías fuertes o establecidas, la mejor opción para el desarrollo de la solución propuesta.

#### Metodologías Fuertes o Establecidas

Las metodologías fuertes o establecidas son aquellas que orientan el proceso de desarrollo de software para grandes equipos de trabajo y se caracterizan por su rigidez. El proceso de mejora presenta cierta resistencia a los cambios, es más controlado, existe un contrato prefijado y el cliente actúa con el equipo de desarrollo mediante reuniones o entrevistas en donde se precisan los detalles del producto.

Están guiadas por una estricta planificación durante el proceso de desarrollo, donde se definen las actividades, los roles, los artefactos que serán generados, las herramientas que se utilizarán, así como el modelado y una documentación detallada.

El Proceso Unificado de Rational (RUP, *Rational Unified Process*), es una metodología fuerte que se usa en proyectos nuevos o actualizaciones de sistemas existentes. RUP es un proceso de desarrollo de software configurable que se adapta a los proyectos variados en tamaños y complejidad. Además, es tolerado por herramientas que automatizan el modelado visual, la administración de cambios y las pruebas.

#### Proceso Unificado de Rational (RUP)

“El Proceso Unificado de Rational (RUP), es un proceso de ingeniería de software cuyo objetivo es producir software de alta calidad, es decir, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecido. Cubre el ciclo de vida y desarrollo de software”. (11)

RUP es un proceso para el desarrollo de un proyecto de software que define claramente quien, cómo, cuándo y qué debe hacerse en el proyecto. Como 3 características esenciales se reconoce que está **dirigido por Casos de Uso** (que orientan el proyecto según la





importancia para el usuario y lo que este requiere), está **centrado en la arquitectura** (relaciona la toma de decisiones que indican cómo tiene que ser construido el sistema y en qué orden), y es **iterativo e incremental** (divide el proyecto en mini proyectos donde los casos de uso y la arquitectura cumplen sus objetivos de manera más depurada). (12)

RUP es una metodología pensada para la Ingeniería de Software que va más allá del análisis y diseño orientado a objetos, proporciona un conjunto de técnicas que soportan el ciclo completo de desarrollo de software. Unifica los equipos de desarrollo, optimiza la productividad de cada uno de sus miembros y ayuda a los líderes de proyecto a incrementar su experiencia en el desarrollo. Describe los diversos pasos involucrados en la captura de los requerimientos y en el establecimiento de una guía arquitectónica para diseñar y probar el sistema hecho de acuerdo a los requerimientos y a la arquitectura.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. Las primeras iteraciones (en las fases de *Inicio* y *Elaboración*) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos y al establecimiento de una buena arquitectura.

### 1.5.2 Lenguaje Unificado de Modelado (UML)

El Lenguaje Unificado de Modelado (UML) no constituye una guía para realizar el análisis y diseño orientado a objetos. Es un lenguaje estándar para visualizar, especificar, construir y documentar los artefactos de sistemas en los que el software juega un papel importante. Permite a los desarrolladores visualizar los resultados de su trabajo en esquemas o diagramas estandarizados (13).

#### Principales características de UML

- ✓ Permite modelar sistemas utilizando técnicas orientadas a objetos (OO).
- ✓ Mediante UML se pueden especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
- ✓ Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, entre otros)
- ✓ Cubre las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos.
- ✓ Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.



RUP propone el uso de UML para generar la documentación y los artefactos del sistema durante la etapa inicial de creación y posteriormente en el desarrollo del producto.

### **1.5.3 Herramienta para el modelado**

Las herramientas de Ingeniería de Software Asistida por Computador (CASE, por sus siglas en inglés), están destinadas a aumentar la productividad del software permitiendo la aplicación práctica de metodologías, permiten la reutilización de los diferentes componentes de software y facilitan la estandarización de la documentación.

#### **Visual Paradigm**

Visual Paradigm es una herramienta profesional multiplataforma, que utiliza como lenguaje de modelado UML. Soporta el ciclo completo del desarrollo de software. Posibilita la construcción de aplicaciones con calidad a un menor coste. Permite crear los artefactos utilizados en las diferentes fases de desarrollo: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Incluye todas las funciones de UML: diagrama de casos de uso, diagramas de actividades, diagramas de componentes, entre otros. Proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. (14)

### **1.5.4 Herramientas para el diseño de interfaz de usuario (IU)**

Durante la fase de inicio y como parte del flujo de Requisitos, definidos por la metodología RUP, se confeccionan los prototipos de interfaz de usuario. La elaboración de prototipos posibilita probar el diseño del sistema antes de comenzar su desarrollo real, lo cual garantiza que se está construyendo el sistema que se quiere antes de dedicarle tiempo y recurso al desarrollo. Además, permite a los clientes y al equipo de trabajo tener una visión general, desde los momentos iniciales, del aspecto que tendrá la aplicación una vez terminada.

Un prototipo es una versión inicial compacta de la solución o parte de la solución de un sistema construido en un breve período de tiempo y mejorado en varias iteraciones para probar y evaluar la eficacia del diseño general que se utiliza para resolver un problema determinado. (15)

#### **Balsamiq Mockups**

Es una herramienta que permite crear prototipos de diseños para la web de forma ágil. Está desarrollada en Adobe Air, facilitando su uso en todos los sistemas operativos, además existe una versión para escritorio multiplataforma (Linux, Windows, Mac). Dispone de más de 75 modelos predefinidos con los diferentes elementos de las interfaces de usuario para



montar prototipos. Posibilita importar los elementos propios que no se encuentran representados en la aplicación.

### 1.5.5 Herramientas y lenguajes de Desarrollo

#### Lenguajes de desarrollo

##### XHTML (Lenguaje de Marcado de Hipertexto Extensible)

El lenguaje *eXtensible Hypertext Markup Language* (XHTML) v1.0 se usa para describir la estructura y el contenido en forma de hipertexto, así como para complementar al mismo con objetos tales como medias y otras aplicaciones. El lenguaje XHTML es muy similar al lenguaje HTML, de hecho, XHTML no es más que una adaptación de HTML al lenguaje XML dada la necesidad de dar soporte al cada vez más creciente número de herramientas basadas en XML. (16)

##### CSS

CSS (Hojas de Estilo en Cascada) es un lenguaje de hojas de estilos creado para controlar la presentación de los documentos electrónicos definidos con HTML y XHTML. Constituye la mejor forma de separar los contenidos y su presentación. Es imprescindible para crear páginas web complejas. Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, entre otros. Una vez creados los contenidos, se utiliza este lenguaje para definir el aspecto de cada elemento: color, tamaño, tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página. (17)

### 1.5.6 Lenguajes de script del lado del cliente

#### JavaScript

JavaScript es un lenguaje de programación que se utiliza esencialmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. (18)

Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM<sup>2</sup>. Es un lenguaje de programación interpretado, por lo que no es

---

<sup>2</sup> El Document Object Model (DOM) es una interfaz de la plataforma y del lenguaje neutro, que permitirá a los programas y scripts acceder y actualizar dinámicamente el contenido, estructura y



necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

### 1.5.7 Lenguajes del lado del servidor

#### PHP

PHP es un acrónimo recursivo que significa *PHP Hypertext Pre-processor* (inicialmente *PHP Tools*, o, *Personal Home Page Tools*) y constituye una de las tecnologías utilizadas en el desarrollo de aplicaciones web, siendo uno de los lenguajes de mayor aceptación para este tipo de desarrollo. Es utilizado en conjunto con el lenguaje HTML del lado del cliente, estando PHP del lado del servidor, formando así una arquitectura cliente/servidor. Admite la incorporación de diferentes librerías externas, se puede integrar con varios gestores de bases de datos como MySQL, PostgreSQL, Oracle, entre otros. Permite realizar procesamiento de información en formularios, foros de discusión, manipulación de cookies<sup>3</sup> y página dinámicas. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable. Es libre y no es necesario pagar licenciar para utilizarlo.

#### Frameworks

En el desarrollo de software, un framework constituye una estructura conceptual y tecnológica de soporte definida, incluye soporte de programas, bibliotecas y un lenguaje interpretado. Ayuda a desarrollar y unir los diferentes componentes de un proyecto.

Se define frameworks, en términos generales, como un conjunto estandarizado de conceptos, prácticas y criterios para enfocar una problemática particular, sirve como referencia para enfrentar y resolver nuevos problemas de índole similar. (19)

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio. (20)

#### jQuery

jQuery es una biblioteca de JavaScript rápida y concisa que simplifica los documentos HTML a través del manejo de eventos, animaciones y las interacciones Ajax para el desarrollo web

---

estilo de los documentos. El documento puede ser procesado y los resultados de que el procesamiento se pueden incorporar de nuevo en la página presentada. (63)

<sup>3</sup> Archivo del ordenador que graba información tal como dónde ha estado el usuario en la red. Algunos sitios web usan cookies para identificar a los visitantes en ese sitio, por lo que el usuario puede ser reconocido cuando vuelve a visitar el sitio, posibilitando funciones como la personalización. (64)



rápido. Es un conjunto de librerías o framework de JavaScript que disminuyen el tiempo de desarrollo para incorporar aspectos de la web 2.0 a la aplicación. Además, facilita la compatibilidad de la aplicación para los navegadores más conocidos, así como el trabajo con técnicas como Ajax, implementando más de cinco funcionalidades específicas con este objetivo.

jQuery permite agregar efectos a las páginas haciéndola más interactiva y con una considerable disminución de código. Tiene una estructura que le da organización al proyecto y evita la implementación de funcionalidades comunes.

### **Symfony**

Symfony es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web. Hace uso del patrón Modelo Vista Controlador, lo cual permite separar la lógica del negocio, la lógica del servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Es compatible con la mayoría de gestores de bases de datos (incluido PostgreSQL), además puede ser ejecutado en diferentes plataformas como: Unix, Linux y Windows.

### **1.5.8 Gestor de Bases de Datos**

Los sistemas de gestión de bases de datos (DBMS, por sus siglas en inglés) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan (21). Se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos. Ayuda a realizar las siguientes acciones:

1. Definición de los datos.
2. Mantenimiento de la integridad de los datos dentro de la base de datos.
3. Control de la seguridad y privacidad de los datos.
4. Manipulación de los datos.

Además debe proporcionar una serie de lenguajes para la definición y manipulación de la base de datos. Estos lenguajes son los siguientes:

1. Lenguaje de definición de datos (DDL): para definir los esquemas de la base de datos.
2. Lenguaje de manipulación de datos (DML): para manipular los datos de la base de datos.



3. Lenguaje de control de datos (DCL): para la administración de usuarios y seguridad en la base de datos.

## **PostgreSQL**

PostgreSQL es un sistema gestor de base de datos objeto-relacional libre. Es robusto, confiable y mantiene la integridad de los datos, puede ejecutarse en la mayoría de los sistemas operativos incluyendo Linux, Unix y Windows. Posee más de 15 años de desarrollo activo y constituye una arquitectura probada que se ha ganado una sólida reputación de confiabilidad, integridad de datos y corrección (22).

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

### **1.5.9 Servidor Web**

Se denomina servidor HTTP tanto al programa como el ordenador que recibe peticiones HTTP desde un ordenador cliente con un visor HTTP (navegador). (23)

## **Apache**

El Servidor Apache HTTP es un servidor Web de tecnología de código abierto y para uso comercial, desarrollado por la *Apache Software Foundation*. Posee una excelente configurabilidad, robustez y es multiplataforma. Apache es gratuito, extensible y el más utilizado, permite ser personalizado con la finalidad de mejorar las necesidades de cada sitio web. Por su diseño modular es muy fácil ampliar las capacidades del servidor.

### **1.5.10 Entorno de Desarrollo Integrado (IDE)**

#### **NetBeans IDE**

NetBeans es un proyecto de código abierto de éxito con una gran base de usuarios y una comunidad en constante crecimiento. Constituye una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación incluyendo PHP. Permite la ejecución de los comandos propios del framework symfony. Existe además un número importante de módulos para extenderlo. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso. (24)



## 1.6 Conclusiones parciales

Este capítulo permitió identificar y definir los conceptos que se utilizarán en la concepción de la nueva versión del módulo para la gestión de ejercicios de la Plataforma Educativa Zera. Para cumplir los objetivos planteados, se realizó una investigación de plataformas y sistemas similares a la propuesta de solución, teniendo en cuenta las funcionalidades comunes e ideas que servirán de apoyo en el proceso de implementación de la solución. Se utilizará la especificación IMS QTI v2.0 para lograr la interoperabilidad de los ejercicios, pues constituye la versión más acabada actualmente. Además, se caracterizaron brevemente las herramientas que serán empleadas y que fueron definidas como parte de la arquitectura del proyecto.



## Características del sistema

### 2.1. Introducción

En el desarrollo de sistemas de información existe interés por mejorar la productividad y crear productos de software de alta calidad. Para ello, es indispensable establecer un entorno disciplinado que permita entenderlo, desde los inicios del software. Con este objetivo se realiza la descripción de la propuesta de solución.

En el presente capítulo y como parte del flujo de trabajo de requisitos se describirán las principales funciones del sistema, realizando una breve descripción de lo que se quiere obtener y aquellos elementos que serán utilizados como base en el proceso de desarrollo, proporcionando requisitos detallados sobre lo que el sistema debe hacer.

### 2.2. Modelo de Dominio

Varios autores coinciden en que el modelo de dominio es una representación visual de las clases conceptuales, que representa los eventos reales más importantes dentro del sistema y cuyo objetivo se basa en la comprensión de los conceptos que utilizan los usuarios, los conceptos con que trabajan y con los que deberá trabajar la aplicación. (25)

#### 2.2.1. Análisis de los conceptos del dominio

Con el objetivo de posibilitar la comprensión del dominio del problema, a continuación se realiza una descripción de aquellos conceptos relacionados.

**Editor Digital:** es el encargado de gestionar los recursos en la plataforma.

**Docente:** es el encargado de la educación del estudiantado, puede crear ejercicios para la evaluación de los mismos.

**Batería:** es un conjunto de ejercicios que pueden ser de diferentes tipos y que estarán relacionados a una materia.

**Conocimientos específicos:** se definen como el conjunto de habilidades que deberá tener el estudiante para vencer un contenido determinado.

**Evaluación:** es el eslabón que expresa, mediante su desarrollo, el grado de acercamiento al objetivo formulado y que posibilita, en última instancia, caracterizar la eficiencia del proceso. (26)

- ✓ **Formativa:** se refiere a todo el proceso de aprendizaje de los alumnos, desde la fase de detección de las necesidades hasta el momento de la evaluación final o sumativa.





- ✓ **Diagnóstico:** comprueba hasta qué punto se han conseguido los objetivos propuestos. Determina el grado de identificación o discrepancia entre éstos y los resultados.
- ✓ **Evaluativo:** son asignados al estudiante para medir el conocimiento influyendo de esta forma en su evaluación final. Estas evaluaciones pueden o no estar asociadas a contenidos específicos.

**Retroalimentación:** mensaje interactivo del ejercicio que brindará al estudiante información adicional para mejor comprensión del mismo (27).

- ✓ **Positivo:** mensaje que se le muestra al estudiante cuando la respuesta es correcta, esta constituye una explicación de la solución del ejercicio.
- ✓ **Negativo:** mensaje que se le muestra al estudiante cuando la respuesta es incorrecta, incluye una pista de la respuesta correcta.
- ✓ **Por Intento:** mensaje que la indica al estudiante detalles de la respuesta del ejercicio.

**Mensaje:** mensajes de interacción para un ejercicio, el cual le servirá de guía al estudiante para la realización del mismo (27).

- ✓ **Conclusivo:** es la información brindada al estudiante cuando la respuesta del ejercicio no es totalmente correcta.
- ✓ **General:** es la información ofrecida al estudiante, sin importar la respuesta del ejercicio.
- ✓ **Correcto:** es la información brindada al estudiante cuando la respuesta del ejercicio es la correcta.
- ✓ **Por intento:** es el mensaje ofrecido al estudiante cada vez que resuelve el ejercicio y la respuesta no es la correcta (mensajes reflexivos).

### 2.2.2. Diagrama del Modelo de Dominio

El diagrama del modelo de dominio representa los conceptos de la realidad, que serán modelados por el sistema a construir. Está constituido por entidades o clases conceptuales, así como las relaciones entre ellas y los atributos que contienen.

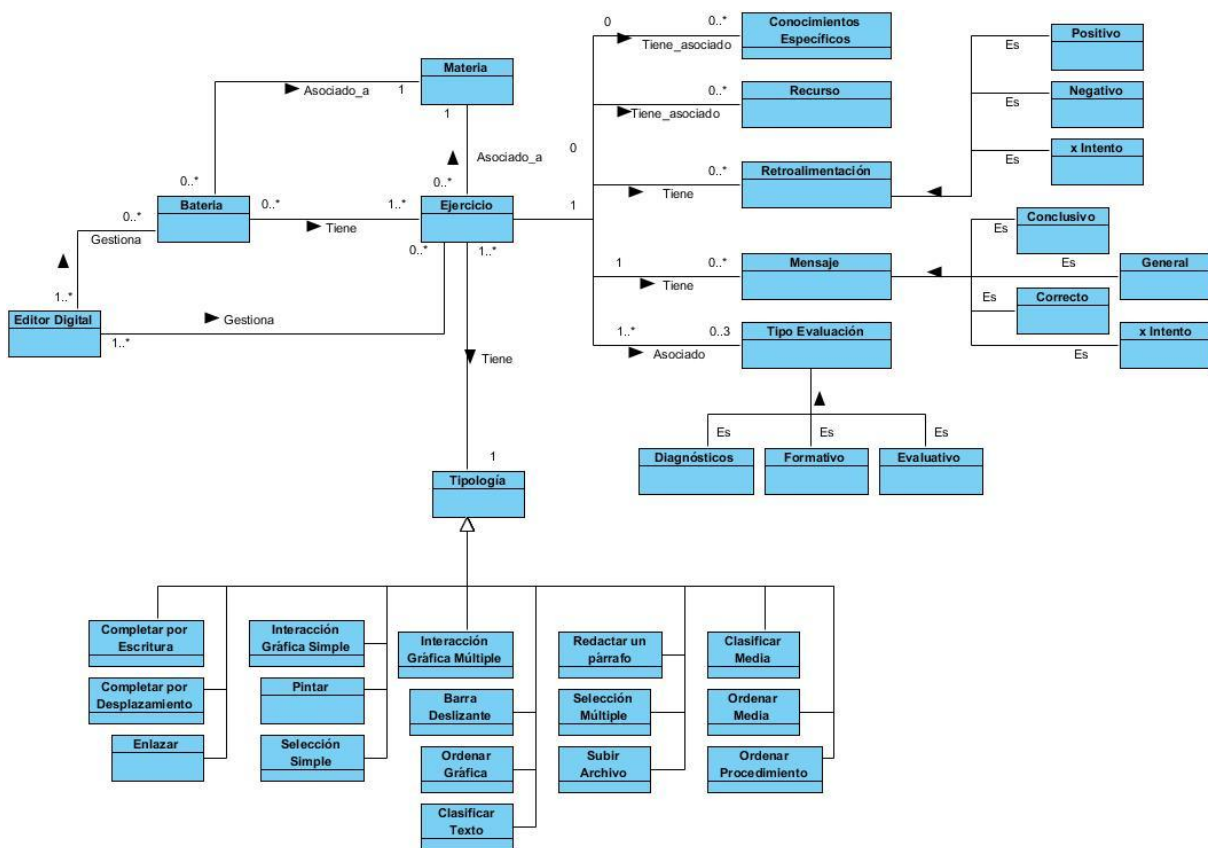


Figura 2 Modelo de dominio.

En la imagen se pueden apreciar los distintos conceptos que se manejan actualmente y la relación entre ellos. A través de su interrelación podemos establecer que en la plataforma Zera, el editor digital puede elaborar ejercicios, que podrán estar asociados o no a una batería, pero estarán asociados a conocimientos específicos para relacionarlos a aquellas habilidades que el estudiante debe vencer. Los ejercicios tendrán retroalimentación y mensajes con el objetivo de facilitar la autoevaluación del estudiante. Estos ejercicios lo componen recursos que facilitarán su comprensión, y a su vez, tienen asociado un tipo de evaluación que puede ser evaluativa, de diagnóstico o formativa.

### 2.3. Descripción del Sistema Propuesto

El sistema formará parte de la Plataforma Educativa Zera, el cual tiene como objetivo facilitarle al profesor la creación de evaluaciones. Los profesores podrán gestionar los ejercicios e incluirlos en las baterías, lo cual permitirá evaluar integralmente al estudiante, pues los ejercicios asociados pueden o no responder a un mismo contenido. El módulo para la gestión de ejercicios contará con nuevas funcionalidades tales como:

- ✓ Crear ejercicios utilizando cualquiera de las interacciones planteadas por IMS-QTI v2.0.



- ✓ Crear baterías de ejercicios.
- ✓ La creación de ejercicios desde cualquier módulo de la plataforma que así lo requiera, y exportar los ejercicios a formato IMS-QTI v2.0 y a .pdf.
- ✓ Exportar e importar recursos tales como: texto enriquecido, sonidos, videos, diaporamas, animaciones e imágenes.

Durante el proceso de gestión de ejercicios se le informará al usuario de los cambios que se realicen. Con la nueva versión del módulo se permitirá intercambiar ejercicios con otras plataformas, los ejercicios podrán ser reutilizados, agrupándolos por características comunes y permitirá la exportación de los mismos independientes de la plataforma (formatos .pdf e IMS-QTI v2.0), logrando así la interoperabilidad, reusabilidad y durabilidad de la información gestionada por el módulo.

### 2.3.1. Especificación de requisitos

Los requerimientos de software determinan con precisión las propiedades y/o restricciones que un producto debe tener para cumplir con las necesidades del usuario. Estos pueden ser funcionales y no funcionales.

Los requisitos funcionales (RF) definen las capacidades o funciones que el sistema debe cumplir, especifican el comportamiento interno de entrada y salida del sistema, sin tener en cuenta las restricciones físicas. Los requisitos no funcionales (RNF) son restricciones de servicios o funciones ofrecidas por el sistema. (13)

El objetivo de este acápite es la identificación de los requisitos funcionales que serán utilizados en la presente propuesta de solución. Los RNF que se relacionan son aquellos requerimientos establecidos por el proyecto Alfaomega perteneciente a la colección Futuro y que responden a las restricciones de la plataforma.

Según lo establecido anteriormente los requisitos que debe tener el sistema son:

#### Requerimientos Funcionales

##### Proceso Gestionar ejercicio

**RF 1.** Ver vista previa: visualiza el ejercicio en la forma que será presentado al estudiante. Muestra la solución del ejercicio, brinda la posibilidad al usuario de deshacer la respuesta, ver la respuesta correcta y comprobar si la respuesta dada es la acertada.

##### Proceso Listar ejercicio.

**RF 2.** Listar Ejercicios: se le muestra al usuario un listado con los ejercicios. Además, se podrá acceder a las opciones de eliminar, editar, edición rápida, publicar y filtrar. También



permitirá acceder a las acciones en lote: eliminar, publicar y exportar un conjunto de ejercicios, seleccionados previamente.

**RF 3.** Filtrar ejercicios: se le permite al usuario realizar una búsqueda de los ejercicios teniendo en cuenta los siguientes criterios: título, descripción, intentos, complejidad, tipología, materia, capítulo, tema, subtema, evaluación, creado, actualizado.

#### **Proceso Exportar ejercicio**

**RF 4.** Exportar ejercicio: se le brinda al usuario la posibilidad de exportar uno o varios ejercicios, podrá acceder a esta opción desde listar ejercicio y Macro Índice, seleccionando previamente los ejercicios que desea exportar. Además, podrá realizar una búsqueda de ejercicios, según diferentes criterios (ver criterios de filtrado en RF 3).

#### **Proceso Importar ejercicio**

**RF 5.** Importar ejercicio: se le ofrece al usuario la posibilidad de importar ejercicios en forma de paquete para ser utilizados en la plataforma, siguiendo la especificación IMS-QTI v2.0. Permite importar los recursos de los ejercicios: animación, imagen, sonido, video y diaporama.

#### **Proceso Gestionar batería**

**RF 6.** Incluir batería: el usuario podrá introducir y/o seleccionar los datos del ejercicio: título, descripción y permitirá seleccionar los ejercicios para ser asociados a la batería.

**RF 7.** Ver datos de la batería: se le permite al usuario observar los datos de la batería seleccionada.

**RF 8.** Modificar datos de la batería: se le permite al usuario cambiar los datos de la batería previamente seleccionada.

**RF 9.** Eliminar una batería: el usuario podrá eliminar una batería que haya sido previamente seleccionada. Al eliminar la batería esta se oculta para el usuario, pero continúa almacenada en la base de datos.

#### **Proceso Listar baterías**

**RF 10.** Listar baterías: se muestra un listado de las baterías que han sido creadas.

**RF 11.** Filtrar batería: se le permite al usuario realizar la búsqueda de baterías, teniendo en cuenta los siguientes criterios: título y descripción.

#### **Proceso Gestionar Tipología Interacción Gráfica Múltiple**

**RF 12.** Incluir datos requeridos por la Tipología Interacción Gráfica Múltiple: se le permite al usuario introducir y/o seleccionar los datos de la tipología. Esta tipología incluye como elemento esencial una imagen que será utilizada como fondo del enunciado.



**RF 13.** Modificar datos de la Tipología Interacción Gráfica Múltiple: se le permite al usuario cambiar los datos de la tipología.

#### **Proceso Gestionar Tipología Interacción Gráfica Simple**

**RF 14.** Incluir datos requeridos por la Tipología Interacción Gráfica Simple: se le permite al usuario introducir y/o seleccionar los datos de la tipología. Incluye como elemento esencial una imagen que será empleada como fondo del enunciado.

**RF 15.** Modificar datos de la Tipología Interacción Gráfica Simple: se le permite al usuario cambiar los datos de la tipología.

#### **Proceso Gestionar Tipología Interacción Ordenar Gráfica**

**RF 16.** Incluir datos requeridos por la Tipología Ordenar Gráfica: se le permite al usuario introducir y/o seleccionar los datos de la tipología. Incluye como elemento esencial una imagen que será utilizada como fondo del enunciado.

**RF 17.** Modificar datos de la Tipología Interacción Ordenar Gráfica: se le permite al usuario cambiar los datos de la tipología.

#### **Proceso Gestionar Tipología Subir Archivo**

**RF 18.** Incluir datos requeridos por la Tipología Subir Archivo: se le permite al usuario introducir y/o seleccionar los datos de la tipología.

**RF 19.** Modificar datos de la Tipología Subir Archivo: se le permite al usuario cambiar los datos de la tipología.

#### **Proceso Gestionar Tipología Pintar**

**RF 20.** Incluir datos requeridos por la Tipología Pintar: se le permite al usuario introducir y/o seleccionar los datos de la tipología.

**RF 21.** Modificar datos de la Tipología Pintar: se le permite al usuario cambiar los datos de la tipología.

#### **Proceso Gestionar Tipología Redactar un párrafo**

**RF 22.** Incluir datos requeridos por la Tipología Redactar un párrafo: se le permite al usuario introducir y/o seleccionar los datos de la tipología.

**RF 23.** Modificar datos de la Tipología Redactar un párrafo: se le permite al usuario cambiar los datos de la tipología.

#### **Proceso Gestionar Tipología Barra deslizante**

**RF 24.** Incluir datos de la Tipología Barra deslizante: se le permite al usuario introducir y/o seleccionar los datos de la tipología.

**RF 25.** Modificar datos de la Tipología Barra deslizante: se le permite al usuario cambiar los datos de la tipología.



## Requerimientos No Funcionales

### Software

Las computadoras deben tener instalado el navegador web Mozilla Firefox 3, Internet Explorer 7, Chrome 8, Opera 10 o superior en cada uno de los navegadores.

### Hardware

Las computadoras locales que brindarán el servicio cliente del sistema no deberán presentar potencias menores a las brindadas por una Pentium 4, al menos, 512 MB de RAM.

### Restricciones en el diseño y la implementación

Como herramienta de modelado se empleará Visual Paradigm v8.0

Lenguaje de programación a usar debe ser PHP v5.3.6

El marco de trabajo base de desarrollo que se utilizará es Symfony 1.4.15.

Como IDE se empleará NetBeans 7.0.1

Como Servidor Web se explotará Apache 2.2.20

El SGDB deberá ser PostgreSQL 9.1

### Apariencia o interfaz externa

Debe contener un diseño sencillo, permitiendo que no sea necesario tener conocimientos informáticos para utilizar el sistema.

El sistema deberá ser intuitivo y requerir de información con claridad y buena organización de la misma, que permita la interpretación correcta e inequívoca.

### Seguridad y Confiabilidad

La plataforma debe proteger el contenido de los productos educativos que gestione, de forma que limite el acceso a usuarios no autorizados.

La seguridad de acceso y administración de usuarios será mediante el otorgamiento de privilegios y roles, asignación de perfiles. Los niveles de acceso están determinados por los diferentes roles válidos dentro de la misma.

### Usabilidad

El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente web en sentido general.

### Portabilidad

El sistema deberá ser multiplataforma.



### Funcionalidad

Reducir al mínimo el tiempo en que carga la plataforma.

### Rendimiento

La ejecución de acciones debe ser de manera rápida y minimizar los pasos a dar en cada proceso.

### Disponibilidad

La plataforma estará disponible para los usuarios registrados en ella, desde la institución en la que estudian o laboran, así como desde cualquier punto con conexión a internet, manteniendo la integridad de los datos y la traza presentados a través de la sincronización de los servidores.

## 2.4. Modelo de Casos de Uso del Sistema

El modelo de casos de uso permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre las condiciones que el sistema debe cumplir.

### Patrones de casos de uso

Un patrón es la pareja problema/solución que estandariza buenos principios y sugerencias relacionados comúnmente con la asignación de responsabilidades. De ahí que un patrón de caso de uso va a codificar el comportamiento del sistema y su interacción con los usuarios. A continuación se enuncian los patrones presentes en el diagrama de casos de uso del sistema:

### Patrón CRUD

El patrón CRUD (crear, ver, modificar, eliminar, por sus siglas en inglés) completo consiste en un caso de uso para administrar la información, permite modelar las diferentes operaciones para administrar una entidad de información.

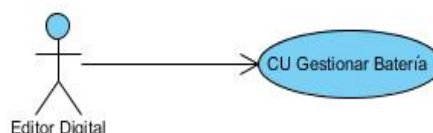


Figura 3 Patrón CRUD

### Concordancia



Extrae una subsecuencia de acciones que aparecen en diferentes lugares del flujo de casos de uso y es expresado por separado.

✓ **Adición:** En el caso de este patrón alternativo, la subsecuencia común de casos de uso, extiende los casos de uso compartiendo la subsecuencia de acciones. Los otros casos de uso modelan el flujo que será expandido con la subsecuencia. Este patrón es preferible usarlo cuando otros casos de uso se encuentran propiamente completos, o sea, que no requieren de una subsecuencia común de acciones para modelar los usos completos del sistema.

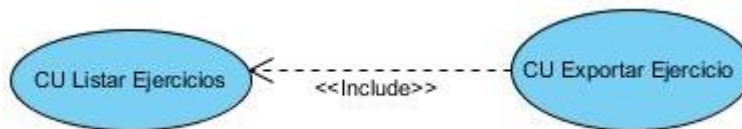


Figura 4 Adición.

### Generalización/Especialización

Consiste en dos casos de usos y una relación de padre e hijo entre los elementos, se identifican los conceptos comunes y se define un concepto general (padre) y un concepto especializado (hijo). Este patrón se aplica cuando un flujo puede heredar de otro caso de uso, obteniendo las características comunes.

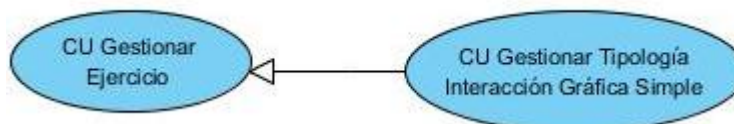


Figura 5 Generalización/Especialización

#### 2.4.1. Descripción de los actores del sistema

Los actores representan aquellas personas u otros sistemas que van a interactuar con la aplicación. A continuación se mencionan los actores del sistema, así como una breve descripción de los mismos.

Actor	Descripción
Administrador Central	Se encarga de la administración global del sitio y tiene permisos sobre todos los componentes de la plataforma.
Editor Digital	Es el encargado, dependiendo de la(s) materia(s) que





tenga asociada, de gestionar ejercicios y baterías.

**Docente**

Usuario que podrá gestionar ejercicios desde los diferentes hiperentornos.

Tabla 2 Descripción de actores del sistema.

**2.4.2. Diagrama de casos de uso del sistema**

El diagrama de casos de uso del sistema especifica las funcionalidades y el comportamiento del sistema a través de la interacción con los actores, representados gráficamente. Permite que los clientes y usuarios validen que el sistema se convierta en lo que esperaban y los desarrolladores del sistema construyan lo que se espera. (13)

Para la realización del diagrama de casos de uso y con el objetivo de obtener una visión general de la gestión de ejercicios, se definen la relación de los actores con los casos de uso que serán inicializados por cada uno de ellos. En el inicio se modela un caso de uso por requisito funcional, luego con la aplicación de los patrones de casos de uso se van agrupando funcionalidades como sucede con el patrón CRUD que incluye: crear, ver, modificar y eliminar ejercicios y baterías para los casos de uso Gestionar Ejercicio y Gestionar Batería, respectivamente. También se utiliza el patrón concordancia en su variante adición. Con la creación de este diagrama se obtendrá una visión general de la nueva versión del módulo para la gestión de ejercicios de la Plataforma educativa Zera.

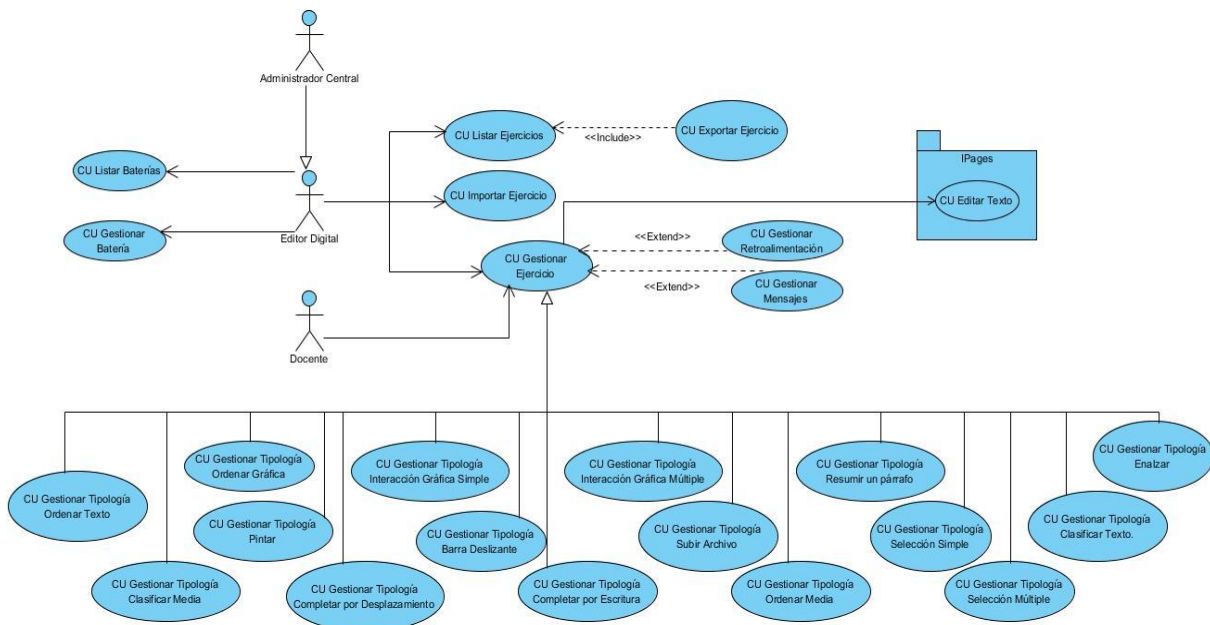


Figura 6 Diagrama de casos de uso del sistema

Los Casos de uso representan las actividades que deben ser realizadas en cada escenario, teniendo en cuenta las funcionalidades que debe cumplir en la interacción con el usuario. A



continuación se describe el CU Gestionar Batería, los restantes CU se pueden ver en el Anexo 1.

<b>Objetivo</b>	Incluir, ver, modificar o eliminar batería.	
<b>Actores</b>	Editor Digital (Inicia), generaliza a Administrador central: incluye, ve, modifica o elimina una batería.	
<b>Resumen</b>	El caso de uso se inicia cuando el actor selecciona la opción que le permite realizar una acción sobre las baterías.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	<p>Debe haberse generado el escritorio de trabajo del usuario autenticado.</p> <p>Para incluir, modificar o eliminar debe haber generado el Caso de Uso Listar batería.</p> <p>Para ver batería el actor debe tener permiso de acceder al contenido, ya sea porque es el autor, un rol superior en jerarquía, o porque se le ha asignado temporalmente.</p> <p>Para modificar batería el actor debe ser el responsable temporal del mismo.</p> <p>Para eliminar batería el actor debe tener el permiso de eliminar el elemento.</p>	
<b>Poscondiciones</b>	Se incluyó, vio, modificó o eliminó una batería por el actor.	
<b>Flujo de eventos</b>		
<b>Flujo básico Gestionar Batería.</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Inicia al seleccionar la opción de realizar una acción sobre una batería.	
2.		<p>Brinda la posibilidad de realizar las acciones:</p> <ul style="list-style-type: none"><li>• Incluir una nueva batería.</li><li>• Ver los datos de la batería. Ver Sección 1: "<u>Ver datos de la batería</u>".</li><li>• Modificar los datos de la batería. Ver Sección 2: "<u>Modificar</u></li></ul>



		<p><u>datos de la batería</u>.</p> <ul style="list-style-type: none"><li>• Eliminar la batería. Ver Sección 3: "<u>Eliminar batería</u>".</li></ul>
3.	Selecciona la opción de incluir una nueva batería.	
4.		<p>Brinda la posibilidad de introducir y/o seleccionar los datos generales de la batería:</p> <ul style="list-style-type: none"><li>✓ Título.</li><li>✓ Descripción.</li><li>✓ Tipo.</li></ul> <p>Permite</p> <ul style="list-style-type: none"><li>• Asociar ejercicios.</li></ul> <p>Además, permite:</p> <ul style="list-style-type: none"><li>• Guardar.</li><li>• Guardar y crear otro.</li><li>• Cancelar.</li></ul>
5.	Introduce y/o selecciona los datos de la batería.	
6.	Selecciona la opción <i>Asociar Ejercicio</i> .	
7.		<p>Brinda la posibilidad de seleccionar la materia donde desea realizar la búsqueda de ejercicios para ser asociados.</p> <p>Permite:</p> <p>Realizar la búsqueda de ejercicios por criterios (Nombre, Palabras claves, intención).</p>
8.	Selecciona la materia para realizar la búsqueda.	
9.		<p>Muestra el listado de ejercicios asociados a la materia seleccionada.</p>



10.	Selecciona los ejercicios que desea asociar a la batería.	
11.		Muestra los datos de la batería y un listado con los ejercicios seleccionados.
12.	Selecciona la opción de Guardar.	
13.		Valida los datos.
14.		<i>Crea la batería.</i>
15.		Muestra un mensaje de información.
16.		Muestra un listado con las baterías creadas.
17.		Termina el caso de uso.
<b>Flujos alternos</b>		
*.a Selecciona la opción de Guardar y crear otro.		
	<b>Actor</b>	<b>Sistema</b>
		*.a.1 Valida los datos.
		*.a.2 <i>Crea la batería.</i>
		*.a.3 Muestra un mensaje de información.
		*.a.4 Regresa al flujo Básico.
*.b Selecciona la opción de Cancelar.		
	<b>Actor</b>	<b>Sistema</b>
		*. b.1 <i>Elimina los datos creados.</i>
		*.b.2 Muestra un mensaje de información.
		*.b.3 Regresa a la vista anterior.
		*.b.4 Termina el caso de uso.
7. a Selecciona la opción de realizar la búsqueda de ejercicios por criterios.		
	<b>Actor</b>	<b>Sistema</b>
		7. a.1 Brinda la posibilidad de introducir y/o seleccionar los



		criterios de búsqueda.
		7. a.2 Termina el caso de uso.
<b>9. a</b> Existen datos incompletos.		
	<b>Actor</b>	<b>Sistema</b>
		9. a.1. Sobre cada campo incompleto muestra un mensaje de error.
		9. a.2 Regresa al paso 4 del Flujo Básico.
<b>9. b</b> Existen datos incorrectos.		
	<b>Actor</b>	<b>Sistema</b>
		9. b.1 Sobre cada campo incorrecto muestra un mensaje de error.
		9. b.2 Regresa al paso 4 del Flujo Básico.
<b>Sección 1: “Ver datos de la batería”</b>		
<b>Flujo básico Gestionar Batería.</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción de ver los datos de la batería.	
2.		Brinda la posibilidad de modificar los datos de la batería. Ver Sección 3: “ <i>Modificar datos de la batería.</i> ”
3.		Termina el caso de uso.
<b>Flujos alternos</b>		
<b>2. a</b> Selecciona la opción de eliminar la batería.		
	<b>Actor</b>	<b>Sistema</b>
		2. a.1. Brinda la posibilidad de eliminar la batería. Ver Sección 4: “ <i>Eliminar batería.</i> ”



		2. a.2. Termina el caso de uso.
<b>2. b</b> Selecciona la opción de salir.		
	<b>Actor</b>	<b>Sistema</b>
		2. b.1 Regresa a la vista anterior.
		2. b.2 Termina el caso de uso.
<b>Sección 1: “Modificar datos de la batería”</b>		
<b>Flujo básico Gestionar Batería.</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción de modificar los datos de la batería.	
2.		Muestra los datos de la batería y brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos y/o seleccionando diferentes. Permite: <ul style="list-style-type: none"><li>• Actualizar los datos.</li><li>• Cancelar la operación en cualquier momento.</li><li>• Eliminar.</li></ul>
3.	Modifica los datos que necesite y selecciona la opción de actualizar los mismos	
		Valida los datos
4.		<i>Actualiza los datos de la batería</i>
5.		Muestra un mensaje de información.
6.		Regresa a la vista anterior.
7.		Termina el caso de uso.
<b>Flujos alternos</b>		
<b>*.a</b> El actor selecciona la opción de Cancelar.		
	<b>Actor</b>	<b>Sistema</b>
		*.a.1 <i>Elimina los datos creados.</i>



		*.a.2 Muestra un mensaje de información.
		*.a.3 Regresa a la vista anterior.
		*.a.4 Termina el caso de uso.
*.b El actor selecciona la opción de Eliminar.		
	<b>Actor</b>	<b>Sistema</b>
		*.a.1 <i>Ocultar la batería.</i>
		*.a.2 Muestra un mensaje de información.
		*.a.3 Regresa a la vista anterior.
		*.a.4 Termina el caso de uso.
4. a Existen datos incompletos.		
	<b>Actor</b>	<b>Sistema</b>
		4. a.1 Sobre cada campo incompleto muestra un mensaje de información.
		4. a.2 Regresa al paso 2 del Flujo Básico.
4. b Existen datos incorrectos.		
	<b>Actor</b>	<b>Sistema</b>
		4. b.1 Sobre cada campo incorrecto muestra el mensaje de información.
		4. b.2 Regresa al paso 2 del Flujo Básico.
<b>Sección 3: "Eliminar batería"</b>		
<b>Flujo básico Gestionar Batería</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción de eliminar batería.	
2.		Muestra un mensaje de advertencia y permite: <ul style="list-style-type: none"><li>• Aceptar.</li></ul>



		<ul style="list-style-type: none"> <li>• Cancelar la operación.</li> </ul>
3.	Selecciona la opción de Aceptar.	
4.		<i>Ocultar la batería.</i>
5.		Muestra un mensaje de información.
6.		Muestra el listado actualizado de baterías.
7.		Termina el caso de uso.
<b>Flujos alternos</b>		
<b>3.a</b> El actor selecciona la opción de Cancelar.		
	<b>Actor</b>	<b>Sistema</b>
		3. a.1 Regresa a la vista anterior.
		3. a.2 Termina el caso de uso.
<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	
<b>Requisitos funcionales</b>	<b>no</b>	La información en el sistema no puede ser eliminada.
<b>Asuntos pendientes</b>		Ver datos de la batería y Ver vista previa de la batería.

Tabla 3 Descripción del CU Gestionar Batería

## 2.5. Conclusiones parciales

Analizados los conceptos asociados al dominio del problema, se presentó el listado de RF que definen las necesidades de las nuevas funciones del módulo para la gestión de ejercicios de la Plataforma Educativa Zera. Se explicó la selección del administrador central, editor digital y docente como actores del sistema, representados mediante los casos de uso, para alcanzar descripciones detalladas de los mismos en el sistema a modelar.





## Análisis y diseño

### 3.1 Introducción

En el flujo de trabajo análisis y diseño se elaboran los elementos necesarios que darán paso a la implementación. Tiene como propósito esencial transformar los requisitos tanto funcionales como no funcionales en un diseño de clases, observando las relaciones e interacción entre ellas. Durante el proceso se tiene en cuenta una arquitectura robusta que permitirá adaptar el sistema al entorno de implementación que se está trabajando y al mismo tiempo posibilitara reutilizar y actualizar los cambios que vayan surgiendo.

### 3.2 Modelo de análisis

Un Modelo conceptual es un diagrama que describe las relaciones entre determinados elementos utilizados como directrices para obtener una condición de interés. Se considera que el modelo de análisis es un modelo conceptual, pues en él se identifican las clases y relaciones que vinculadas entre si van a permitir la interpretación y funcionamiento del sistema que se modela.

#### 3.2.1 Diagramas de clases del análisis

El diagrama de clases del análisis describe el funcionamiento y características comunes del sistema, constituye un artefacto centrado en los requisitos funcionales pues no incluye los elementos del diseño, al no considerar posibles tecnologías que serán utilizadas.

Las clases del análisis están siempre identificadas con uno de los tres estereotipos existentes, los cuales son:



Clases Interfaz: son utilizadas para modelar la interacción actor-sistema (es decir, usuarios y sistemas externos).



Clases Control: representan la coordinación, secuencia, transacciones y control de otros objetos, con frecuencia son utilizadas para coordinar el trabajo de las clases y encapsulan el comportamiento de un caso de uso concreto.



Clases Entidad: modelan la información y el comportamiento del sistema. Se derivan de una clase de entidad del negocio o del dominio.

A continuación se muestra una representación del diagrama de clases del análisis del CU Gestionar Ejercicio. El resto de los diagramas se encuentran en el **Anexo 2**.

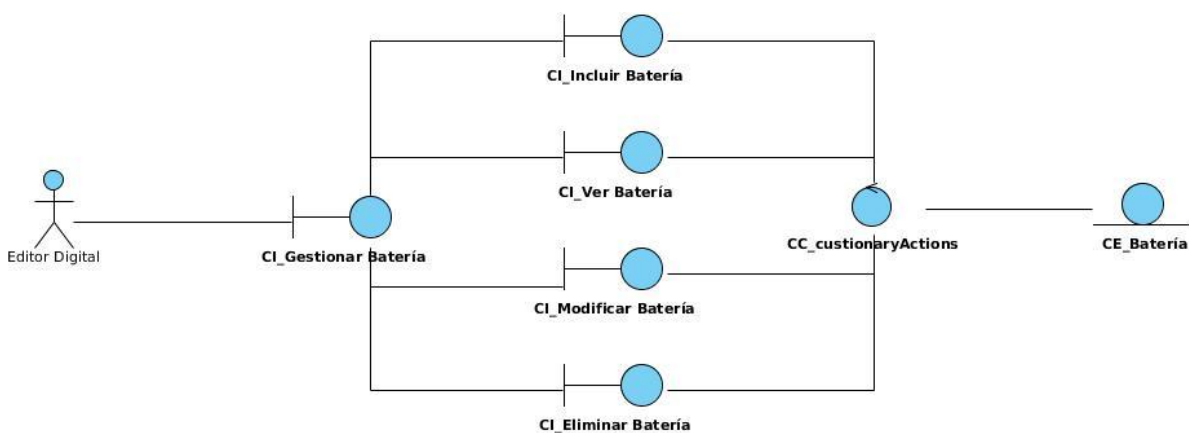


Figura 7 Diagrama de clases del análisis CU Gestionar Bateria

### 3.2.2 Diagramas de colaboración del análisis

Un diagrama de colaboración muestra una interacción organizada en torno a los objetos que efectúan operaciones. Es parecido a un diagrama de objetos que muestra los objetos y los enlaces existentes entre ellos, necesarios para implementar una operación de nivel más elevado. Los diagramas de colaboración describen el comportamiento dinámico del sistema mostrando de forma sencilla los elementos entre sí.

A continuación se muestra una representación del diagrama de colaboración del análisis (DCA) del CU Gestionar Ejercicio. El resto de los diagramas se encuentran en el **Anexo 3**.

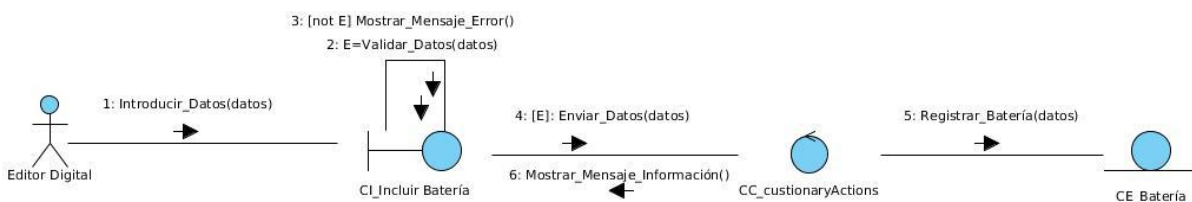


Figura 8 CU Gestionar Bateria. Sección Incluir

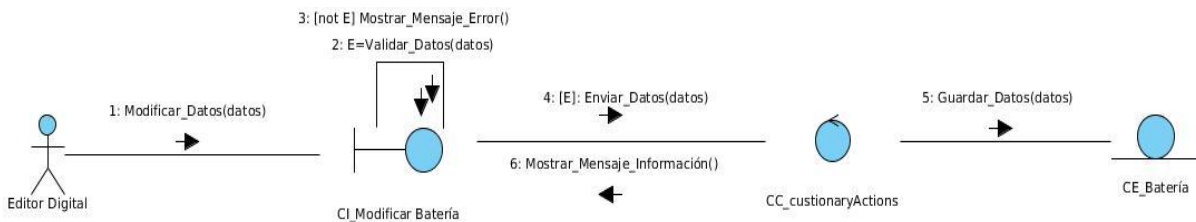


Figura 9 DC CU Gestionar Bateria. Sección Modificar

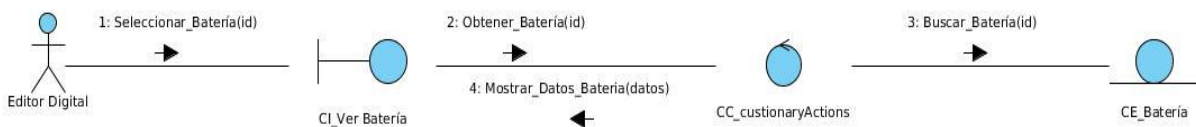


Figura 10 DC CU Gestionar Bateria. Sección Ver

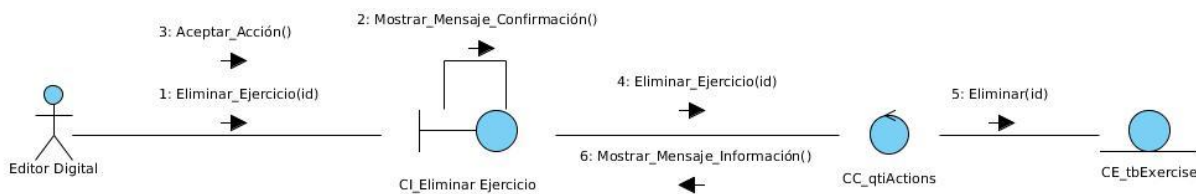


Figura 11 DC CU Gestionar Bateria. Sección Eliminar

### 3.3 Patrón arquitectónico Modelo – Vista – Controlador en Symfony

Symfony está basado en un patrón clásico del diseño web conocido como arquitectura MVC. Está formado por tres niveles:

**Modelo (Model):** encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.

**Vista (View):** muestra la información al usuario y obtiene los datos del modelo. Pueden existir múltiples vistas del modelo, cada vista tiene asociado un componente controlador.

**Controlador (Controller):** reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son



traducidos a solicitudes de servicio para el modelo o la vista. El usuario interactúa con el sistema a través de los controladores.

### 3.4 Aplicación de los patrones de diseño en Symfony

Para la realización de la versión 2.0 del módulo para la gestión de ejercicios de la Plataforma Educativa Zera se emplean aquellos patrones que fueron definidos en la versión 1.0 de dicho módulo. (27)

#### Patrones GoF (Grupo de los cuatro)

- ✓ Singleton
- ✓ Decorator

#### Patrones GRASP (Patrones generales de software para asignación de responsabilidades)

- ✓ Experto
- ✓ Creador
- ✓ Alta Cohesión
- ✓ Bajo Acoplamiento
- ✓ Controlador

### 3.5 Modelo de diseño

El modelo de diseño es un modelo de objetos que describe la realización de los CU centrándose en los RF y RNF. Sirve de abstracción y es utilizado como entrada principal en las actividades de implementación, estas abstracciones son directas y representan una sencilla correspondencia entre el diseño y la implementación. (13)

#### 3.5.1 Diagrama de clases del diseño

El diseño es la entrada a la fase de implementación del sistema por lo que poseen una construcción similar, este flujo constituye el centro de atención en la fase de elaboración y contribuirá a lograr una arquitectura estable y sólida.

Con el desarrollo de aplicaciones web, a finales del 90, se hizo necesario buscar formas que permitiesen su modelación. Debido a que UML es un lenguaje extensible, Jim Conallen haciendo uso de estas facilidades modela aplicaciones web. De esta forma su extensión utiliza tres estereotipos esenciales. A continuación se muestra una descripción detallada de cada uno:

---

**Estereotipo**

**Características**

---



<<Server Page>>

Representa las páginas web que tienen código interpretable e interactúa con los recursos en el servidor, las operaciones representan las funciones del código y los atributos las variables visibles dentro del alcance de la página.



<<Client Page>>

Constituye una página web, con formato HTML, es una mezcla de datos, presentación y lógica que son interpretadas por los navegadores y sus atributos son las variables declaradas dentro del script que son accesibles para páginas cualquier función dentro de la página.



<Form>

Son aquellos elementos de entrada que forman parte de una página cliente. Sus atributos son los elementos de entrada del formulario (input boxes, text áreas, radio buttons, check boxes y hidden fields).

Tabla 4 Estereotipos y sus características.

A continuación se muestra una representación del diagrama de clases del diseño (DCD) del CU Gestionar Ejercicio. El resto de los diagramas y los atributos de las tablas representadas en el diagrama se encuentra en **Anexo 4**.

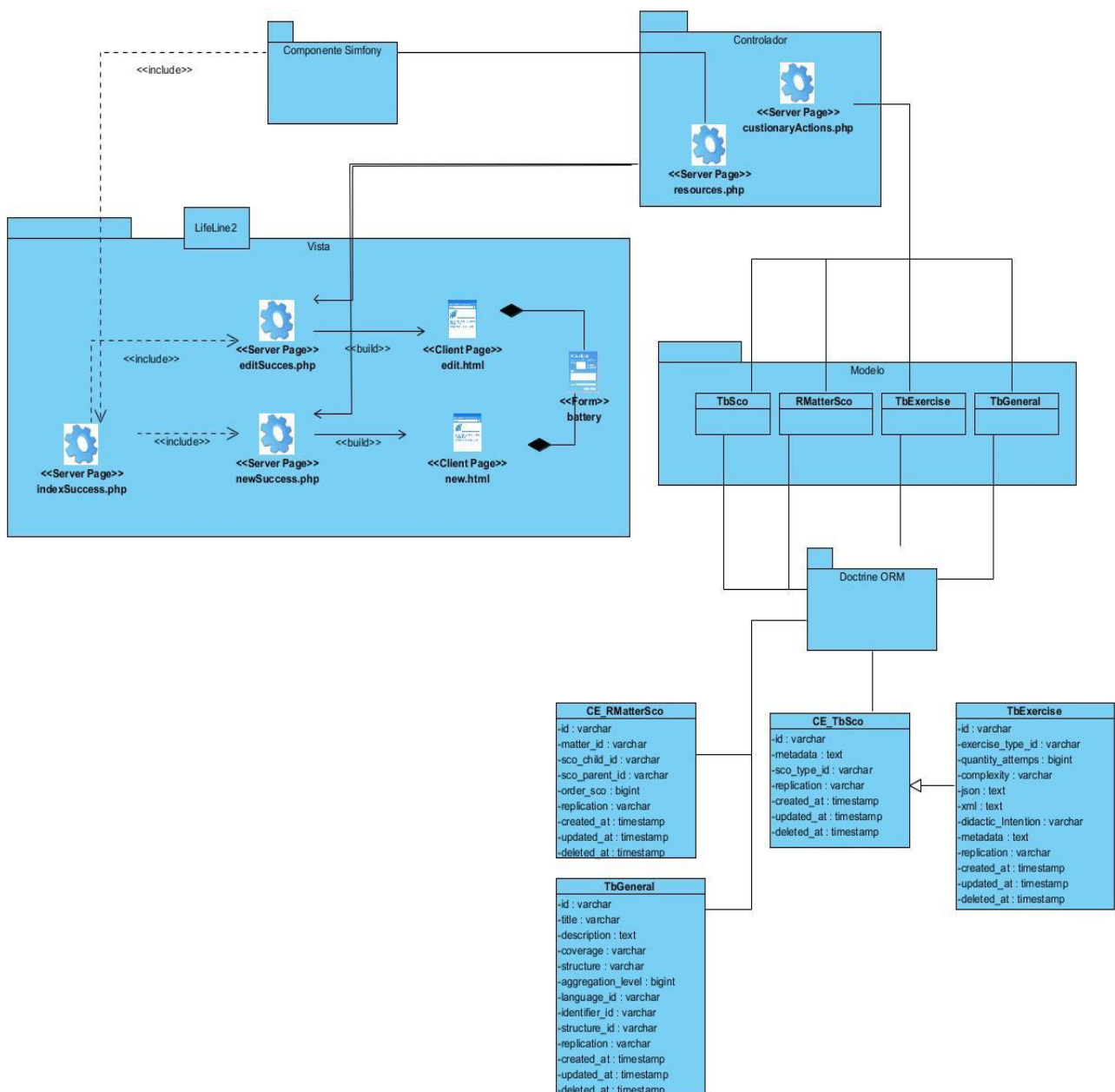


Figura 12 DCD CU Gestionar Bateria.

### 3.6 Diseño de la base de datos

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular. A continuación se representa el modelo de bases de datos del módulo para la gestión de ejercicios de la Plataforma Educativa Zera:

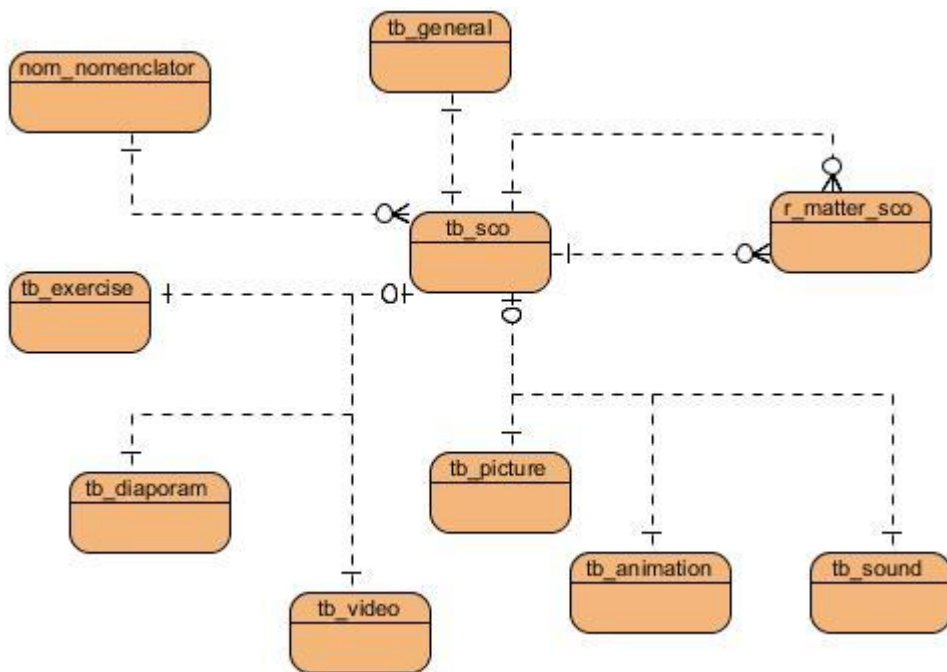


Figura 13 Modelo de la Base de Datos.

### 3.6.1 Descripción de las tablas

A continuación se muestra la descripción de las tablas `tb_sound`, `tb_video`, `tb_diaporam`, `tb_picture`, `tb_animation` que fueron adicionadas como parte de la nueva versión del módulo. El resto se encuentran en **Anexo 6**.

`tb_diaporam`

**Descripción:** en la siguiente tabla se agrupa la información existente de un diaporama.

Atributo	Tipo	Descripción
id	varchar(128)	Etiqueta única que identifica al objeto ( <code>tb_diaporam</code> ) en la tabla.
diaporam	varchar(500)	Almacena la ubicación del recurso en la plataforma.
url	varchar(128)	Almacena la dirección de donde se obtuvo el recurso.
access_date	timestamp	Almacena la fecha en la que fue accedido el recurso.
reference_picture	varchar(500)	Almacena la imagen de referencia del recurso en la plataforma.
license_id	varchar(128)	Almacena el identificador del tipo de



		licencia del recurso.
mime_id	varchar(128)	Almacena el identificador de la extensión del recurso.
replication	varchar(5)	Almacena las opciones de replicación del diaporama en la plataforma.
created_at	timestamp	Almacena la fecha de creación del diaporama.
updated_at	timestamp	Almacena la última fecha de actualización del diaporama.
deleted_at	timestamp	Almacena la fecha de eliminación del diaporama.

Tabla 5 Descripción: tabla tb\_diaporam.

tb\_animation

**Descripción:** en la siguiente tabla se agrupa la información existente de una animación.

Atributo	Tipo	Descripción
id	varchar(128)	Etiqueta única que identifica al objeto (tb_animation) en la tabla.
animation	varchar	Almacena la ubicación del recurso en la plataforma.
url	varchar	Almacena la dirección de donde se obtuvo la animación.
access_date	timestamp	Almacena la fecha en la que fue accedido la animación.
reference_picture	varchar	Almacena la imagen de referencia del recurso en la plataforma.
license_id	varchar(128)	Almacena el identificador del tipo de licencia del recurso.
mime_id	varchar(128)	Almacena el identificador de la extensión del recurso.
replication	varchar(5)	Almacena las opciones de replicación de la animación en la plataforma.
created_at	timestamp	Almacena la fecha de creación del ejercicio.





updated_at	timestamp	Almacena la última fecha de actualización de la animación.
deleted_at	timestamp	Almacena la fecha de eliminación de la animación.

Tabla 6 Descripción: tabla tb\_animation.

tb\_sound

**Descripción:** en la siguiente tabla se agrupa la información existente de un sonido.

Atributo	Tipo	Descripción
id	varchar(128)	Etiqueta única que identifica al objeto (tb_sound) en la tabla.
sound	varchar	Almacena la ubicación del recurso en la plataforma.
url	varchar	Almacena la dirección de donde se obtuvo el sonido.
access_date	timestamp	Almacena la fecha en la que fue accedido el sonido.
license_id	varchar(128)	Almacena el identificador del tipo de licencia del recurso.
mime_id	varchar(128)	Almacena el identificador de la extensión del recurso.
replication	varchar(5)	Almacena las opciones de replicación del sonido en la plataforma.
created_at	timestamp	Almacena la fecha de creación del sonido.
updated_at	timestamp	Almacena la última fecha de actualización del sonido.
deleted_at	timestamp	Almacena la fecha de eliminación del sonido.

Tabla 7 Descripción: tabla tb\_sound.

tb\_video

**Descripción:** en la siguiente tabla se agrupa la información existente de un video.



Atributo	Tipo	Descripción
id	varchar(128)	Etiqueta única que identifica al objeto (tb_video) en la tabla.
video	varchar	Almacena la ubicación del recurso en la plataforma.
url	varchar	Almacena la dirección de donde se obtuvo el video.
access_date	timestamp	Almacena la fecha en la que fue accedido el video.
reference_picture	char	Almacena la imagen de referencia del recurso en la plataforma.
license_id	varchar(128)	Almacena el identificador del tipo de licencia del recurso.
mime_id	varchar(128)	Almacena el identificador de la extensión del recurso.
replication	varchar(5)	Almacena las opciones de replicación del video en la plataforma.
created_at	timestamp	Almacena la fecha de creación del video.
updated_at	timestamp	Almacena la última fecha de actualización del video.
deleted_at	timestamp	Almacena la fecha de eliminación del video.

Tabla 8 Descripción: tabla tb\_video.

tb\_picture

**Descripción:** en la siguiente tabla se agrupa la información existente de una imagen.

Atributo	Tipo	Descripción
id	varchar(128)	Etiqueta única que identifica al objeto (tb_picture) en la tabla.
big_picture	bytea	Almacena la imagen en código binario en tamaño real.
small_picture	text	Almacena la referencia de la imagen pequeña en la plataforma.



url	varchar	Almacena la dirección de donde se obtuvo la imagen.
access_date	date	Almacena la fecha en la que fue accedido la imagen.
license_id	varchar(128)	Almacena el identificador del tipo de licencia del recurso.
mime_id	varchar(128)	Almacena el identificador de la extensión del recurso.
width	varchar(128)	Almacena el ancho de la imagen.
high	varchar	Almacena el largo de la imagen.
replication	varchar(5)	Almacena las opciones de replicación de la imagen en la plataforma.
created_at	timestamp	Almacena la fecha de creación de la imagen.
updated_at	timestamp	Almacena la última fecha de actualización de la imagen.
deleted_at	timestamp	Almacena la fecha de eliminación de la imagen.

Tabla 9 Descripción: tabla tb\_picture.

### 3.7 Conclusiones parciales

Luego de analizar y diseñar el sistema, se definieron aquellos elementos que guiarán el desarrollo de la nueva versión del módulo para la gestión de ejercicios, se describieron las clases y sus relaciones, se pulieron los requisitos funcionales y se determinaron las nuevas tablas que formarán parte de la base de datos. Se realizó el diagrama de clases del análisis de los casos de uso del sistema, se confeccionó los diagramas de colaboración y diagramas de clases del diseño y se obtuvo, de esta forma, los artefactos necesarios que servirán como entrada a la implementación.



## Implementación y prueba

### 4.1 Introducción

La fase de construcción se centra esencialmente en la implementación del sistema, se inicia a partir de los artefactos generados como resultado del análisis y el diseño. La idea esencial de este flujo es lograr desarrollar la arquitectura y el sistema como un todo, y realizar una descripción de los elementos del diseño, implementados en términos de componentes. Mientras que en la etapa de prueba se verificarán las funcionalidades del sistema obtenido como resultado de la implementación. Las pruebas se realizan para detectar errores durante el desarrollo, garantizando de modo eficiente, el rendimiento y fiabilidad del software.

Es necesario mencionar que se pueden ir realizando pruebas a lo largo del desarrollo, a los artefactos que se van generando como resultado de cada una de las fases. En ocasiones el éxito de un producto de software depende de calidad con que se realicen las pruebas.

### 4.2 Modelo de implementación

El modelo de implementación describe como los elementos del diseño y las clases serán implementadas en términos de componentes, describe la organización considerando los mecanismos de estructuración utilizados y la dependencia entre componentes.

#### 4.2.1 Diagrama de componentes

Un componente es el empaquetamiento físico de los elementos de un modelo (13). Por tanto, un diagrama de componentes refleja todos los archivos y elementos que son utilizados en la implementación para hacer disponible el sistema físicamente. A continuación se muestra una representación del diagrama de componentes del CU Gestionar Batería. El resto de los diagramas aparece en el **Anexo 5**.

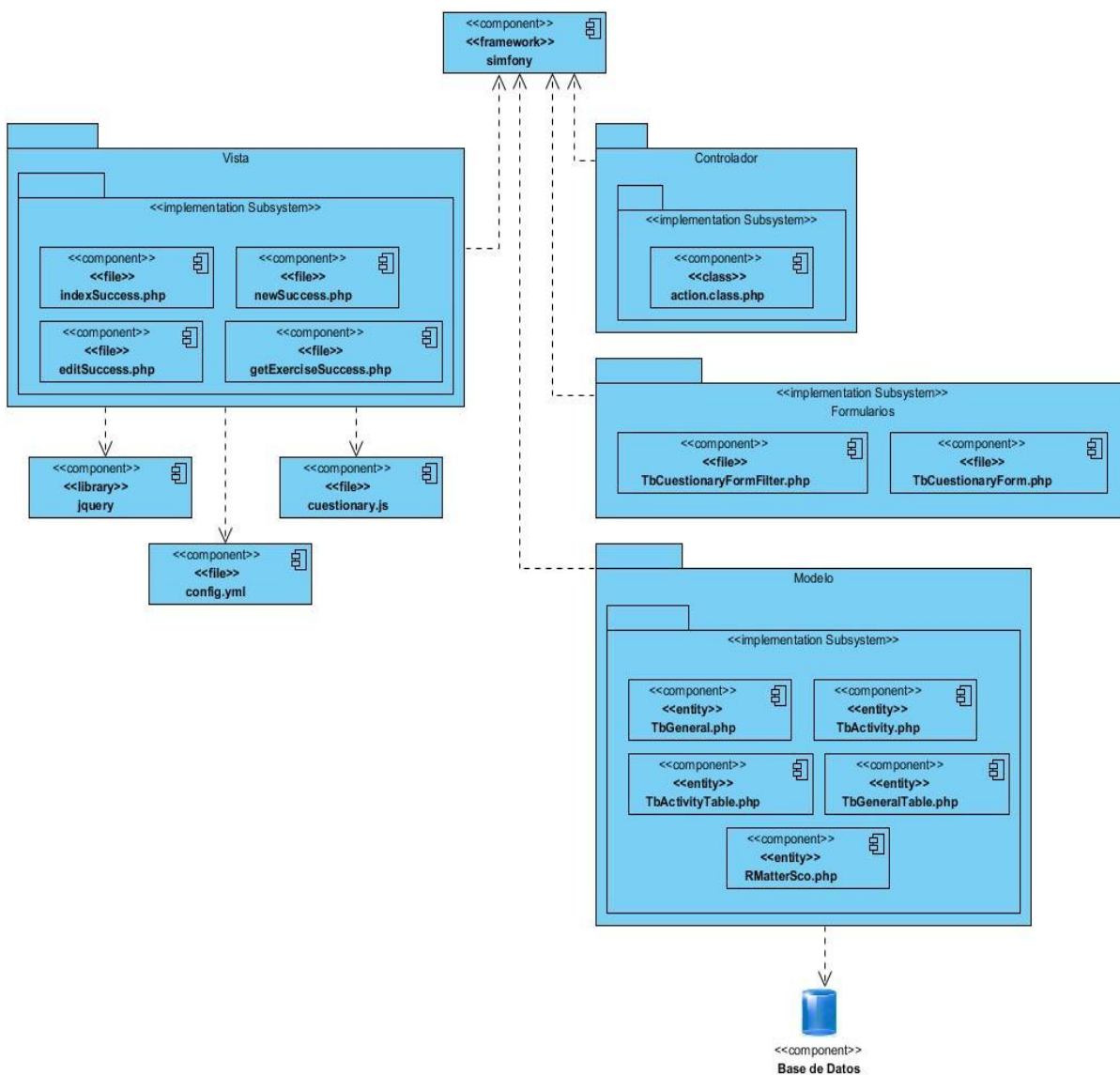


Figura 14 Diagrama de Componentes. Batería.

## 4.3 Pruebas de software

### 4.3.1 Niveles de Prueba

**Prueba de Unidad:** es la prueba enfocada en aquellos elementos más pequeños del diseño de software, el componente o módulo (28). Es aplicable a componentes representados en el modelo de implementación para verificar que los flujos de control y de datos están cubiertos, y que ellos funcionen como se espera.

**Prueba de Integración:** es la prueba ejecutada para asegurar que los componentes en el modelo de implementación operen correctamente cuando son combinados para ejecutar un caso de uso. Se prueba un paquete o un conjunto de paquetes del modelo de implementación. Las pruebas de integración descubren errores en las especificaciones de



las interfaces de los paquetes constituye responsabilidad de desarrolladores y de independientes, sin solaparse las pruebas.

**Prueba de regresión:** cada vez que se agrega un nuevo módulo como parte de la prueba de integración, el software cambia. Se establecen nuevas rutas de flujo de trabajo, ocurren nuevas entradas y salidas. Estos cambios pueden causar problemas con funciones que antes trabajaban bien. Una prueba de regresión consiste en ejecutar nuevamente el subconjunto de pruebas que ya se han aplicado para asegurarse de que los cambios no han propagado elementos colaterales indeseables. (28)

#### 4.3.2 Métodos de Prueba

Hay una de dos maneras de probar cualquier producto construido: 1) si se conoce la función específica para la que se diseñó el producto, se aplican las pruebas, que demuestren que cada función es simplemente operacional, mientras se buscan errores de cada función. 2) si se conoce el funcionamiento interno del producto, se aplican pruebas para asegurarse de que todas las piezas encajan, es decir que todas las operaciones internas se realizan de acuerdo con las especificaciones y que se han probado todos los componentes internos de manera adecuada. (28) Las primeras constituyen las pruebas de caja negra y la segunda posibilidad las pruebas de caja blanca.

**Pruebas de caja negra:** es la prueba que se realiza teniendo en cuenta la interfaz del software. Su objetivo es verificar las especificaciones funcionales del producto, sin considerar la estructura interna, no es necesario tener dominio sobre el funcionamiento interno del software. No validan funciones ocultas (por ejemplo funciones implementadas pero no descritas en las especificaciones funcionales del diseño) por tanto los errores asociados a ellas no serán encontrados.

#### 4.4 Diseño de Casos de Prueba

Un caso de prueba (CP) consiste en un conjunto de elementos de entradas, condiciones de ejecución y resultados desarrollados para cumplir un objetivo particular o una función esperada. Por tanto, con los casos de prueba se pretende: demostrar que las funciones del software son operativas, que las entradas se aceptan de la forma adecuada y que se produce el resultado correcto, así como la entrada de la información externa (por ejemplo archivos de datos) se mantiene. Los CP se encuentran en el **Anexo 7**.

#### 4.5 Resultados obtenidos

Durante el desarrollo de la nueva versión del módulo para la gestión de ejercicios se



realizaron pruebas de unidad para comprobar el funcionamiento del software, las cuales fueron realizadas por el desarrollador durante la implementación, aprovechando las ventajas de compilación brindadas por el IDE utilizado. Dichas pruebas no fueron registradas pues se realizaron durante el proceso de mejora de la solución.

Se especifican las principales pruebas realizadas a la aplicación haciendo uso del método de caja negra, estas pertenecen al nivel de prueba de sistema y se detallan a continuación.

**Pruebas Internas:** fueron implementadas por el equipo de desarrollo del proyecto (analistas) con el fin de corregir y entregar una solución con la mínima cantidad de errores. Se centraron en el cumplimiento de los requisitos y casos de uso del sistema.

**Pruebas Cruzadas:** fueron realizadas por personal ajeno al módulo (analistas de otros módulos) con la finalidad de identificar posibles errores relacionados con las validaciones, formato de campos, así como pautas definidas por la arquitectura.

A continuación se presentan los resultados arrojados durante las diferentes pruebas aplicadas:

#### Pruebas Internas:

Módulo	Casos de Uso	Iteración	NC	Cerrada(S)	No procede
Ejercicios	13	1ra	15	15	0
		2da	10	10	3

Tabla 10 Resultados de las pruebas internas.

#### Pruebas Cruzadas:

Módulo	Casos de Uso	Iteración	NC	Cerrada(S)	No procede
Ejercicios	13	1ra	10	10	0

Tabla 11 Resultados de las pruebas cruzadas.



#### 4.6 Conclusiones parciales

Al terminar este capítulo se obtuvo el diagrama de componentes para lograr una mejor comprensión física del sistema. Para la incorporación de la nueva versión del módulo para la gestión de ejercicios en Zera se desarrollaron funcionalidades como: la gestión de baterías, la inclusión de nuevas tipologías de ejercicios y la exportación e importación de ejercicios. Los resultados de las diferentes pruebas posibilitaron detectar y corregir no conformidades relacionadas con el diseño de los casos de prueba, así como errores funcionales de la aplicación.





## Conclusiones Generales

Durante la investigación desarrollada se generaron los artefactos y una documentación detallada que facilitará la realización de posibles mejoras al módulo. Además, se llegó a las siguientes conclusiones:

- Para la gestión de ejercicios, la especificación más utilizada es IMS-QTI, la cual resulta ser la más recurrida a nivel internacional, pues garantiza el intercambio de ejercicios entre plataformas.
- Se elaboró un marco teórico conceptual a partir del estudio de las principales tendencias y herramientas existentes relacionadas con la gestión de ejercicios.
- La confección de los artefactos correspondientes al flujo de trabajo análisis y diseño, fueron empleados como apoyo en la implementación de la nueva versión del módulo para la gestión de ejercicios.
- Se agregaron nuevas funcionalidades al proceso de gestión de ejercicios.
- Se evidenció mediante las pruebas realizadas al producto el correcto funcionamiento del mismo, pudiendo apreciar el cumplimiento de los Requisitos Funcionales identificados.

Dándose cumplimiento a todos los objetivos específicos y garantizando la creación de un módulo para la gestión de ejercicios interoperables que cuente además con la portabilidad y usabilidad deseada.



## Recomendaciones

Debido a que el ejercicio o la batería al ser eliminados se pierde todo enlace de los mismo con cualquier asociación que tenga, dígase página, asignación al estudiante, los autores recomiendan tener dicha deficiencia en cuenta y relacionar los ejercicios o baterías, guardando su relación con cualquier elemento, para imposibilitar la eliminación de los mismos.

También es recomendable implementar todas las interacciones propuestas por la especificación IMS-QTI v2.0 para incrementar el intercambio de ejercicios con otros sistemas.

Permitir la importación y exportación de baterías de ejercicios en formato IMS-QTI v2.0, así como la exportación de las mismas en .pdf.



## Bibliografía

1. Plataformas educativas. [En línea] [Citado el: 9 de enero de 2012.] <http://agora.ucv.cl/manual/plataformas/plataformas.html>..
2. **García, Peñalvo Francisco José.** Teoría de la Educación: Educación y Cultura en la Sociedad de la Información. Estado actual de los sistemas e-learning. [En línea] 2005. [http://campus.usal.es/~teoriaeducacion/rev\\_numero\\_06\\_2/n6\\_02\\_art\\_garcia\\_penalvo.htm](http://campus.usal.es/~teoriaeducacion/rev_numero_06_2/n6_02_art_garcia_penalvo.htm)..
3. **IEEE.** IEEE Standard Glossary of Software Engineering Terminology. 1990. ISBN 1-55937-067-X.
4. **Codina, Lluís.** El libro digital y la www. Madrid : s.n., 2000. p. 274. ISBN 84 88605 544.
5. Diccionario de la Real Academia Española. Versión en línea.
6. **Fernández Manjón, Baltasar, y otros, y otros.** Cnice. Uso de estándares aplicados a las TIC en Educación. [En línea] [Citado el: 11 de enero de 2012.] <http://ares.cnice.mec.es/informes/16/index.htm>.
7. **Consortium, IMS Global Learning.** IMS Question and Test Interoperability Implementation Guide V 2.0. 2005.
8. **Baltasar Fernández Manjón, Pablo Moreno Ger, José Luis Sierra Rodríguez, Iván.** Uso de estándares aplicados a Tic. 2006.
9. **Guzmán, Clara López.** Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning. Salamanca, España : s.n., 2005. Tesina doctoral.
10. **Cataldi, Lage, y García Martínez,.** Ingeniería de Software Educativo.
11. **Díaz-Antón, María.** Propuesta de una metodología de desarrollo de software educativo bajo un enfoque de calidad sistémica. Caracas,Venezuela : s.n.
12. **Gallego, Juan Pablo Gómez.** Scribd. Fundamentos de la metodología RUP. [En línea] [Citado el: 10 de enero de 2012.] <http://es.scribd.com/doc/297224/RUP>.
13. **Rumbaugh, J. and Jacobson, I. y Booch, G.** El proceso unificado de desarrollo de software. 2000.
14. **Manager, Free Download.** Visual Paradigm for UML (ME) - (Paradigma Visual para UML (ME)) (Visual Paradigm for UML (ME)) 6.0. [En línea] 2007.



[http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_%28M%C3%8D%29\\_14720\\_p/..](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/)

15. Ayuda de Rational.

16. **Pérez, Javier Eguíluz.** Introducción a XHTML. 2008.

17. —. Introducción a CSS. 2008.

18. —. Introducción a JavaScript. 2008.

19. [En línea] [Citado el: 16 de noviembre de 2011.] <http://x-web.es/2012/04/07/comparador-de-frameworks/>].

20. [En línea] [Citado el: 16 de noviembre de 2011.] <http://www.buenastareas.com/ensayos/Tecnologia-Framework-Net-De-Microsoft/3525138.html>].

21. [En línea] [Citado el: 16 de noviembre de 2011.] <http://www.buenastareas.com/ensayos/Base-De-Datos/52096.html>.

22. PostgreSQL - Sitio Oficial. . PostgreSQL. [Online] [Cited: diciembre 15, 2011.] <http://www.postgresql.org>.

23. Servidores HTTP-Cursos-Programación Web. [En línea] [Citado el: 10 de enero de 2012.] <http://www.programacionweb.net/articulos/articulo/?num=402>. [Citado el: 10 de enero de 2012].

24. [En línea] [Citado el: 16 de noviembre de 2011.] <http://netjava.bligoo.com/netbeans>.

25. **Rumbaugh, James y Jacobson, Ivar. y Booch Grady.** El lenguaje Unificado de Modelado.Manual de Referencia.

26. **Zayas, Dr. Cs. Carlos M. Alvarez de.** La pedagogía como ciencia(Epistemología de la educación).

27. **García Sanchez, Adrián and La Hoz Pavón, Lizandra.** Desarrollo del módulo Ejercicios para la plataforma educativa ZERA basado en estándares para la interoperabilidad. Universidad de las Ciencias Informáticas. 2011.

28. **Pressman, Roger S.** Ingeniería de Software.Un enfoque práctico. Sexta Edición.



29. **Labañino Rizzo, César y del Toro Rodríguez, Mario.** Ecured. [En línea] [Citado el: 22 de mayo de 2012.] [http://www.ecured.cu/index.php/Hiperentorno\\_de\\_aprendizaje](http://www.ecured.cu/index.php/Hiperentorno_de_aprendizaje).
30. Wizhosting. Alojamiento web en Argentina. [En línea] [Citado el: 16 de febrero de 2012.] <http://www.wizhosting.com/e-learning..>
31. **Lapuente, María Jesús Lamarca.** Hipertexto, el nuevo concepto de documento en la cultura de la imagen. [En línea] noviembre de 2011. [Citado el: 10 de enero de 2012.] <http://www.hipertexto.info/documentos/metadatos.htm>.
32. **Ortiz, Kadir Hector.** BIBLIOTECA VIRTUAL de Derecho, Economía y Ciencias Sociales. Plataforma para el control del uso de softwares educativos. [En línea] 2004. [Citado el: 15 de mayo de 2012.] <http://www.eumed.net/libros/2009c/583/indice.htm>.
33. **Marqués, Pere.** El software educativo. [En línea] [Citado el: 10 de enero de 2012.] [http://www.lmi.ub.es/te/any96/marques\\_software/..](http://www.lmi.ub.es/te/any96/marques_software/)
34. **Wiley, David A.** Connecting learning objects to instructional design theory: A definition, a metaphor, and taxonomy. [En línea] 2000. <http://reusability.org/read/chapters/wiley.doc>. .
35. IEEE. Learning Object Metadata. [En línea] [Citado el: 14 de octubre de 2011.] <http://ltsc.ieee.org/wg12/index.html>.
36. MoodleDocs. [En línea] [Citado el: 23 de abril de 2012.] [http://docs.moodle.org/all/es/Agregar\\_un\\_Wiki..](http://docs.moodle.org/all/es/Agregar_un_Wiki..)
37. LMS Definition from PC Magazine Enciclopedia. [En línea] [Citado el: 9 de enero de 2012.] [http://www.pcmag.com/encyclopedia\\_term/0,2542,t=LMS&i=46205,00.asp..](http://www.pcmag.com/encyclopedia_term/0,2542,t=LMS&i=46205,00.asp..)
38. IMS GLC. Content Packaging Specification. [En línea] [Citado el: 10 de enero de 2012.] [http://www.imsglobal.org/content/packaging/..](http://www.imsglobal.org/content/packaging/)
39. **Etxeberría, Ana Landeta.** Capítulo 16 - Estándares de e-learning. Libro de Buenas Prácticas de e-learnig. [En línea] <http://www.buenaspracticass-elearning.com/capitulo-16-estandares-e-learning.html..>
40. **Larman, Craig.** UML y Patrones.Una introducción al análisis y diseño orientado a objetos y al proceso unificado.
41. **Potencier, Fabien.** Symfony la guía definitiva. 2009.



42. IEEE Sección El Salvador. [Online] [Cited: mayo 22, 2012.] [http://ewh.ieee.org/r9/el\\_salvador/](http://ewh.ieee.org/r9/el_salvador/).
43. [En línea] <http://keibee.com/es/marcos-php/>].
44. [Online] [Cited: noviembre 16, 2011.] [http://www.librosweb.es/symfony\\_1\\_0/capitulo1/symfony\\_en\\_pocas\\_palabras.html](http://www.librosweb.es/symfony_1_0/capitulo1/symfony_en_pocas_palabras.html)].
45. [En línea] [Citado el: 20 de noviembre de 2011.] <http://postgresql.uci.cu/node/63>.
46. [En línea] [Citado el: 16 de noviembre de 2011.] <http://programaciondesarrollo.es/ventajas-del-servidor-apache/>.
47. **Cano, Cristina Alonso**. Los recursos informáticos y los contextos de enseñanza y aprendizaje en Sancho, Juana Ma (Coord.): Para una tecnología educativa. Barcelona : Horsori.
48. **del Castillo Ventosa, Hector, García Fernández, Gonzalo y Sanz Sierra, Rafael**. Implementación del módulo para el sistema <e-AUla> siguiendo especificaciones IMS QTI. Madrid : s.n.
49. **García Molina, Jesús y José Ortín, M**. De los procesos del negocio a los Casos de Uso. Facultad de Informática, Universidad de Murcia.
50. **Berlaga Flores, Adriana y García Peñalvo, Francisco**. Introducción a los Estándares y Especificaciones en Ambientes e-learning. Salamanca, España : s.n.
51. **Génova Fuster, Gonzalo, Fuentes Torres, José Miguel y Cruz, Valiente Blázquez María**. Evaluación Comparativa de herramientas CASE para UML desde el punto de vista notacional. Dpto. de Informática, Universidad Carlos III. Madrid : s.n.
52. **Manual, The Unified Modeling Language Reference**. Rumbaugh, James; Jacobson, Ivar; Booch, Grady. ISBN 0-201-30998-X .
53. Un estudio comparativo de herramientas para el modelado UML. **Bernardo Quintero, Juan, y otros, y otros**. 137, Colombia : s.n., Revista Universidad Eafit, Vol. 41. ISSN (Versión Impresa): 0120-341X.
54. **del Moral, María Esther, Cernea, Doina Ana y Villalustre, Lourdes**. Contributions of the Web 2.0 to collaborative work. Universidad de Oviedo.



55. **MARQUÉS GRAELLS, P.** Software Educativo. Guía de uso y metodología de diseño. Barcelona : Editorial Estel.
56. Repositorios de objetos de aprendizaje: Bibliotecas para compartir y reutilizar recursos en los entornos e-learning. **López Gúzman, Clara y García Peñalvo, Francisco.** 002, México : s.n., julio-diciembre de 2006, Biblioteca Universitaria, Vol. 9. ISSN (Versión impresa): 0187-750X.
57. **Álvarez González, Luis Alberto.** Sistema de Gestión del Aprendizaje. [En línea] [http://www.gita.cl/files/3\\_Sistemas\\_de\\_Gestion\\_de\\_Aprendizaje\\_v21.pdf](http://www.gita.cl/files/3_Sistemas_de_Gestion_de_Aprendizaje_v21.pdf). .
58. Plataformas abiertas de e-learning para el soporte de contenidos. **Boneu, Joseph M.** España : s.n., Revista de Universidad y Sociedad del Conocimiento (RUSC) de la Universidad Oberta de Catalunya., Vol. 4.
59. Las plataformas en la educación en línea. **Robles Peñaloza, Alberto.** Revista Electrónica e-formadores.
60. **Wills, G.B, y otros, y otros.** A Delivery Engine for QTI Assessments. School of Electronics and Computer Science, University of Southampton, Southampton, UK .
61. **Secilia, Miguel Angel.** Libro de Buenas Prácticas de e-learning. Estándares de e-Learning. [En línea] <http://www.buenaspracticas-elearning.com/Colaboradores.html>.
62. Interoperabilidad; Estándares. **Castañeda de León, Ing. Luz María.** 10, Revista Digital Universitaria, Vol. 5. ISSN:1067-6079.
63. W3C Architecture domain. Document Object Model. [Online] <http://www.w3.org/DOM>.
64. <http://diccionario.babylon.com/cookie/>. [Online]