

Universidad de las Ciencias Informáticas

Facultad 4



Desarrollo del módulo Ejercicios para la plataforma educativa ZERA basado en estándares para la interoperabilidad.

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

Autores:

Lizandra La Hoz Pavón

Adrián García Sánchez

Tutor:

Ing. Osmany Casas Benítez

Co-tutora:

Ing. Cindy Santos Salgado

La Habana, 2011

“Año 53 de la Revolución”

Declaración de autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) a que haga el uso que estimen pertinente con el mismo.

Para que así conste firmamos la presente a los ___ días del mes de Junio del 2011.

Lizandra La Hoz Pavón

Firma del autor

Adrián García Sánchez

Firma del autor

Ing. Osmany Casas Benítez

Firma del tutor

Ing. Cindy Santos Salgado

Firma de la co-tutora

Índice de contenido

Resumen.....	i
Introducción	1
Fundamentación Teórica.....	7
1.1 Introducción.....	7
1.2 Conceptos asociados al dominio del problema	7
1.2.1. Software educativo	7
1.2.2. Plataformas educativas	8
1.2.3. e-Learning	8
1.2.4. Sistema de Gestión de Aprendizaje	9
1.2.5. Objeto de aprendizaje	9
1.2.6. Estándares que permiten el intercambio de ejercicios	10
1.2.6.1. Estándares para la interoperabilidad	10
1.2.6.2. Estándares para los metadatos	11
1.2.6.3. Estándar para empaquetamiento.....	12
1.3 Sistemas similares.....	13
1.4 Metodologías de desarrollo de software	15
1.5 Lenguaje de Modelado	17
1.6 Herramientas CASE para el modelado UML	18
1.7 Herramientas y lenguajes de Desarrollo.....	20
1.8 Gestor de Bases de Datos	24
1.9 Servidores Web.....	26
1.10 Entorno de Desarrollo Integrado.....	27
1.11 Patrones	28
1.12 Conclusiones parciales	30
Características del sistema	31
2.1. Introducción	31
2.2. Modelo de Dominio.....	31
2.2.1. Análisis de los conceptos del Dominio.....	31
2.2.2. Diagrama del Modelo de Dominio	32
2.3. Descripción del Sistema Propuesto.....	33
2.4. Modelo de Casos de Uso del Sistema.....	38
2.4.1. Descripción de los actores del sistema.....	40
2.4.2. Diagrama de actores del sistema.....	40
2.4.3. Diagrama de casos de uso del sistema.....	41

ÍNDICE DE CONTENIDO

2.4.4. Descripción de los casos de uso del sistema.....	41
2.5. Conclusiones parciales	48
Análisis y diseño del sistema	49
3.1. Introducción	49
3.2. Modelo de análisis.....	49
3.3. Modelo de diseño	52
3.4. Diseño de la base de datos.....	55
3.5. Diagrama de despliegue	57
3.6. Conclusiones parciales	58
Implementación y pruebas	59
4.1. Introducción	59
4.2. Modelo de implementación	59
4.3. Diagrama de componentes	59
4.4. Pruebas de software	60
4.5. Diseño de casos de prueba	61
4.6. Resultado Obtenidos	70
4.7. Conclusiones parciales	70
Conclusiones generales	71
Recomendaciones.....	72
Bibliografía	73

Resumen

El avance de las tecnologías de la información y las comunicaciones ha cambiado la manera de gestionar los procesos que antes se realizaban de una forma tradicional. Actualmente son cada vez más las instituciones educativas que emigran hacia la modalidad a distancia en su afán de ofertar sus planes y programas de estudios a un mayor número de usuarios. Dentro de la Universidad de las Ciencias Informáticas, en la Facultad 4 está en desarrollo la plataforma de contenidos educativos ZERA. La presente investigación tiene como propósito el desarrollo de un módulo que gestione los ejercicios para la evaluación de los estudiantes en la plataforma ZERA, con un valor educativo reutilizable e interoperable en otros sistemas para la gestión del aprendizaje, realizándose un estudio de cómo otras plataformas y sistemas similares gestionan los ejercicios, así como de las herramientas y lenguajes de programación a emplear en el análisis, diseño, implementación y prueba del módulo.

Palabras clave: exámenes, interoperabilidad, estándares e-learning, plataforma web.

Introducción

Esta emergente sociedad de la información, impulsada por un vertiginoso avance científico en un marco socioeconómico neoliberal-globalizador y sustentado por el uso generalizado de las potentes y versátiles tecnologías de la información y las comunicaciones (TIC), conlleva cambios que alcanzan todos los ámbitos de la actividad humana. Sus efectos se manifiestan de manera muy especial en las actividades laborales y en el mundo educativo, donde todo debe ser revisado: desde la razón de ser de la escuela y demás instituciones educativas, hasta la formación básica que precisamos las personas, la forma de enseñar y de aprender, las infraestructuras y los medios que utilizamos para ello. (1)

En Cuba, la Universidad de las Ciencias Informáticas (UCI), es la primera universidad que surge al calor de la Batalla de Ideas. Su misión es formar profesionales, comprometidos con su patria, calificados en la rama de la Informática, a partir de un modelo pedagógico flexible, que vincula dinámicamente y coherentemente el estudio con la producción y la investigación, acorde con las necesidades sociales del país.

En el marco de los contratos surgidos a raíz del reconocimiento de la universidad como empresa productora de software, surge como proyecto, el desarrollo de una plataforma para la gestión del aprendizaje, nombrada ZERA. La misma tiene sus orígenes en una concepción pedagógica denominada “hiperentornos de aprendizaje”, propuesta y desarrollada por pedagogos y especialistas del Ministerio de Educación de Cuba (MINED), según las Licenciadas Dianelys Muguía Álvarez y Kethicer Castellanos Rodríguez en su artículo Software Educativo, los hiperentornos no son más que una mezcla de diferentes tipos de software educativos sustentado en tecnología hipertexto, estos constituyen un modelo de medio de enseñanza para el apoyo al proceso de enseñanza – aprendizaje concebidos además como fundamento teórico y conceptual del desarrollo de las colecciones cubanas de software educativos. (2)

Los software bajo esta concepción, son la base para el desarrollo de la plataforma para la gestión del aprendizaje (ZERA) que integra los principales conceptos de los hiperentornos antes citados, las mejores prácticas y elementos arquitectónicos de soluciones similares, las principales especificaciones y estándares educativos desarrollados y utilizados a nivel mundial en plataformas de aprendizaje colaborativo.

Cada día para los usuarios finales es más difícil adaptarse a los sistemas por las diferencias que existen entre sí, tanto en interfaz como en los formatos de fichero que

manejan, de igual modo para los desarrolladores es casi imposible lograr una compatibilidad con otro sistema, por las variaciones de implementación que existen para software con similares características.

Las grandes diferencias en la implementación de software hacen muy difícil mantener, e incrementar sistemas una vez creados, por los cambios vertiginosos en el mundo del desarrollo de software. La creación e implementación de estándares (tipo, modelo, norma, patrón o referencia) en el proceso de desarrollo de software fijó las bases para la existencia de normas o patrones reconocidos internacionalmente, logrando la interoperabilidad e intercambio entre aplicaciones similares o dependientes.

El uso de estándares de manera general logra una terminología común permitiendo a los diseñadores discutir los mismos conceptos; y hacer valoraciones comparativas, mantenimiento y evolución dado que todos los programas tienen la misma estructura y el mismo estilo e identidad común lo que hace que todos los sistemas sean fáciles de reconocer reducción en la formación, dada la facilidad de transmitir de un sistema a otro salud y seguridad por la cantidad de controles.

Estos estándares tienen 2 categorías principales, los estándares de jure, los cuales son respaldados por organizaciones oficiales tales como, la Organización Internacional de Normalización (ISO) y el Instituto Nacional de Normalización Estadounidense (ANSI¹) así como los estándares de facto o de popularidad, es decir que aún no han sido reconocidos por ninguna organización pero su popularidad hace de ellos una norma o estrategia seguida, adoptada y aprobada por muchos en el mercado.

En los entornos e-Learning² también se han implementado estándares con objetivos bien definidos favoreciendo este nuevo sistema de enseñanza entre ellos:

- LOM (Metadatos de Objetos de Aprendizaje) que bajo el principio de los metadatos como información sobre un objeto. Sea éste físico o digital define una estructura para la descripción interoperable de objetos educativos.(3)
- IMS³ QTI (Interoperabilidad de Ejercicios y Exámenes) propone las normas para el intercambio de ejercicios entre aplicaciones sin importar su autoría u

¹ Organización privada sin fines lucrativos que administra y coordina la normalización voluntaria y las actividades relacionadas a la evaluación de conformidad en los Estados Unidos.

² Procesos de enseñanza-aprendizaje que se llevan a cabo a través de Internet, caracterizados por una separación física entre profesorado y estudiantado, pero con el predominio de una comunicación tanto síncrona como asíncrona, a través de la cual se lleva a cabo una interacción didáctica continuada.

origen de los mismos, agrupando los ejercicios según el tipo de interacción necesaria para sus resoluciones.

Los exámenes son una de las opciones para la comprobación del conocimiento, y tienen vital importancia en cualquier sistema de enseñanza donde el profesor así como el estudiante necesitan comprobar su avance en la apropiación de conocimiento de una materia determinada. La gestión de estos ejercicios siempre ha sido muy variable en los distintos Sistemas para la Gestión de Aprendizaje (LMS⁴ por sus siglas en inglés) por los diferentes conceptos sobre esta rama, obligando a profesores a re-escribir los ejercicios una y otra vez sin posibilidades de uso en otras plataformas o exámenes incrementando el tiempo de creación de los mismos.

Actualmente ZERA como producto dirigido a la gestión de contenido educativo para el aprendizaje posee las siguientes deficiencias en el módulo Gestión de ejercicios:

- Una gestión diferente a la utilizada por herramientas profesionales.
- Escasa **interoperabilidad**: evitando intercambiar o mezclar ejercicios sin importar la fuente de los mismos de forma transparente, sencilla y directa.
- Insuficiente **reusabilidad**: imposibilitando el agrupamiento y reutilización de forma rápida y sencilla de los ejercicios para utilizarse incluso en un contexto distinto de aquel para el que fueron inicialmente diseñados.
- Mínima **durabilidad**: obligando a los consumidores a estar atrapados a la tecnología.
- Poca **confianza**: los usuarios confían más en aplicaciones que se rigen por estándares internacionales.

A raíz del problema existente y dado la necesidad de estandarizar la gestión de ejercicios, se plantea el **problema a resolver** en la siguiente interrogante: ¿Cómo gestionar los ejercicios en la plataforma educativa ZERA de manera que permita la interoperabilidad de los mismos?

³IMS Global Learning Consortium es un Consorcio formado por miembros provenientes de organizaciones educativas, así como de empresas públicas y privadas, cuya misión es desarrollar y promover especificaciones abiertas para facilitar las actividades del aprendizaje online.(32)

⁴ Los LMS se encargan de apoyar el modelo de formación e-Learning, y son sistemas basados en la web que permiten el acceso a la información de manera online, eliminando las barreras del tiempo.

Teniendo como **objetivo general**: Desarrollar un módulo para la gestión de ejercicios en la plataforma educativa ZERA que permita la interoperabilidad de los mismos.

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

El **campo de acción** se enmarca en el proceso de gestión de ejercicio bajo el estándar IMS-QTI en la plataforma ZERA.

Objetivos Específicos:

- Realizar el estudio del estado del arte de las herramientas que soportan la especificación IMS-QTI.
- Realizar el análisis y diseño del módulo propuesto.
- Re-implementar el módulo de gestión de cuestionarios interactivos actual, de manera que posibilite el intercambio de ejercicios entre los distintos LMS, para la plataforma educativa ZERA.
- Validar el módulo en la importación y exportación de ejercicios.

Para guiar la investigación se plantea la siguiente **idea a defender**: Si se estandariza el módulo para la gestión de ejercicios en la plataforma educativa ZERA, entonces se logrará una interoperabilidad de la información procesada para cada ejercicio, permitiendo la importación y exportación de los mismos.

En el proceso de investigación y elaboración se utilizaron **métodos científicos de investigación** tales como:

En el nivel teórico:

Entre los métodos teóricos que se identificaron necesarios aplicar durante la investigación está el **histórico-lógico** con el objetivo de conocer la evolución de sistemas similares y de los estándares existentes para la gestión de ejercicios en plataformas educativas. Para conformar el marco teórico se utilizó el método **analítico-sintético**, el cual permite estudiar y analizar la información sobre las de plataformas de gestión de aprendizaje. Además se utiliza la **modelación** para realizar un diseño simplificado de la realidad a través de diagramas de clases, diagramas de flujo y diagramas de componentes, y así poder llegar a tener una visión más precisa de lo que se desea realizar con la gestión de ejercicios.

En el nivel empírico:

Se hizo uso del método de **observación** para obtener información de las necesidades existentes a lo largo del desarrollo del proyecto. Para examinar las distintas fuentes de información como internet, libros, revistas, se utiliza el método de **análisis de fuentes de información**.

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

- Estudio e investigación de las herramientas que gestionan ejercicios.
- Investigación de las herramientas y metodologías existentes para realizar el análisis y diseño del módulo.
- Definición de los casos de uso arquitectónicamente significativos.
- Análisis y diseño del módulo gestión de ejercicios para la plataforma educativa ZERA.
- Estandarización de la gestión de ejercicios en la plataforma educativa ZERA.
- Validación del módulo ejercicio.
- Integración de la solución al resto de los módulos de la plataforma.
- Aplicación de los estándares de codificación establecidos.

El presente trabajo está estructurado en 4 capítulos

Capítulo 1: Fundamentación Teórica

Contiene la fundamentación teórica que soporta la investigación, enfatizando en el estudio de metodologías de desarrollo, en las tecnologías y lenguajes de programación que se utilizan en el desarrollo del módulo. Además, de realizar una exploración de soluciones existentes, semejantes al módulo en desarrollo para así tener una medida de las posibles automatizaciones y aportes a realizar.

Capítulo 2: Características del sistema

Contiene el levantamiento de requisitos y el modelado del negocio, obteniendo los principales artefactos que generan estos flujos de trabajo propuestos por la metodología a utilizar.

Capítulo 3: Análisis y Diseño del módulo Ejercicios.

Se realiza el análisis y diseño del módulo, así como los diagramas de clases web, que brindan una visión clara de cómo debe quedar el producto luego de realizada la investigación.

Capítulo 4: Implementación y pruebas del módulo Ejercicios.

INTRODUCCIÓN

Los aspectos del desarrollo son tratados en este capítulo. Se describe cómo está implementado el módulo, presentando los diagramas de componentes y despliegue. También se describen las pruebas realizadas al módulo para comprobar su funcionamiento.

Fundamentación Teórica

1.1 Introducción

Muchos sistemas de gestión de aprendizaje o Learning Management System (LMS) están proponiendo la estandarización de sus procesos para obtener una mayor interoperabilidad y reusabilidad de los contenidos generados por estos, teniendo en cuenta lo necesario que es para el avance de los sistemas e-learning. Obtener una visión general de los LMS estandarizados o en proceso, las herramientas y lenguajes adecuados para llevar a cabo un correcto proceso de estandarización, así como los estándares para recursos educativos (ejercicios) y la posibilidad de intercambio con otras herramientas o plataformas a través del estudio del estado de arte es el eslabón fundamental para dar inicio al proceso de investigación.

1.2 Conceptos asociados al dominio del problema

1.2.1. Software educativo

Según Jaime Sánchez (2008) el software educativo es cualquier programa computacional cuyas características estructurales y funcionales sirvan de apoyo al proceso de enseñar, aprender y administrar. Un ejemplo más restringido de software educativo lo define como aquel material de aprendizaje especialmente diseñado para ser utilizado por una computadora en los procesos de enseñar y aprender.(4)

Son programas para ordenadores creados con la finalidad específica de ser utilizados como medio didáctico, es decir, para facilitar los procesos de enseñanza y aprendizaje. Esta definición engloba todos los programas que han estado elaborados con fin didáctico, desde los tradicionales programados basados en los modelos conductistas de la enseñanza, los programas de Enseñanza Asistida por Ordenador (EAO), hasta los aún programas experimentales de Enseñanza Inteligente Asistida por Ordenador (EIAO), que utilizando técnicas propias del campo de los Sistemas Expertos y de la Inteligencia Artificial en general, pretenden imitar la labor tutorial personalizada que realizan los profesores y presentan modelos de representación del conocimiento en consonancia con los procesos cognitivos que desarrollan los alumnos. (5)

Se puede concluir que el software educativo es toda aplicación informática que puede servir de herramienta en el proceso de enseñar y aprender con el fin de lograr un avance en el desarrollo del conocimiento de los alumnos, a partir del empleo de

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

recursos multimedia, ejercicios y juegos instructivos que apoyan las funciones de evaluación, calificación y otros.

1.2.2. Plataformas educativas

Las plataformas educativas son Entornos Virtuales de Educación y Aprendizaje (EVEA), que surgen y se desarrollan con la intención de dar soporte a la docencia. Se apoyan en sistemas informáticos basados habitualmente en el protocolo http, con herramientas adaptadas a las necesidades educacionales requeridas. Esos entornos virtuales implican una renovación en los procedimientos de enseñanza y de aprendizaje, la manera de presentar la información de forma dinámica, que puede modificarse constantemente, de ahí el interés de trabajar con metodologías modulares, y la capacidad de integrar medios distintos, rompiendo con la noción de enseñanza centrada en el profesor para orientarse hacia una interdisciplinariedad basada en el alumno. (6)

Las plataformas educativas permiten estimular la idea de cooperación y de interacción, como aspectos centrales del proceso de aprendizaje y enseñanza, mediante el uso de herramientas colaborativas que favorecen la adquisición de aprendizajes significativos en los estudiantes y que al mismo tiempo afianzan en las docentes prácticas de enseñanza mediadas por las Tecnologías de la Información y la Comunicación (TIC).(7)

El software educativo se caracteriza por ser una herramienta altamente interactiva que permite que los procesos enseñar y aprender tengan un gran avance en el desarrollo del conocimiento de los alumnos, a partir del empleo de recursos multimedia, ejercicios y juegos instructivos que apoyan las funciones de evaluación y calificación. También permite el intercambio con los estudiantes, evalúa lo aprendido y posibilita el desarrollo de las habilidades a través de la ejercitación.

1.2.3. e-Learning

La definición del término e-learning hace referencia, por una parte, al uso de las tecnologías de internet (e -) y por otra, a una metodología de transmisión de conocimiento y desarrollo de habilidades centradas en el sujeto que aprende (learning); ya no se basan estos procesos en el profesor que enseña.(8)

El término e-learning nace con la incorporación de internet y las nuevas tecnologías en el área de la educación, permitiendo el aprendizaje a distancia y un nuevo modelo de

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

enseñanza donde el estudiante es el centro del proceso que se suministra de programas educacionales y sistemas de aprendizaje a través de medios electrónicos.

1.2.4. Sistema de Gestión de Aprendizaje

Un Sistema de Gestión de Aprendizaje (LMS por sus siglas en inglés) permite crear o brindar contenidos, monitorear la participación de los estudiantes dentro del sistema, además de evaluar la actuación y desarrollo de los mismos, por otra parte, promueve en los estudiantes diferentes habilidades para el uso de herramientas interactivas como foros de discusión, videoconferencias, tele conferencias y correo electrónico. (9)

Los LMS se encargan de apoyar el modelo de formación e-Learning, y son sistemas basados en la web que permiten el acceso a la información de manera online, eliminando las barreras del tiempo y logrando una centralización y calidad de la información, que además, cuenta con las herramientas para una correcta asimilación de los conocimientos por parte de los estudiantes.

1.2.5. Objeto de aprendizaje

Desde los inicios de la aplicación de las tecnologías de la información y las telecomunicaciones al aprendizaje, ha existido gran controversia respecto a la definición del término Objeto de Aprendizaje (OA). Las diferentes visiones que cada autor o institución tienen de lo que es (o debe ser) son a veces tan diversas que resulta complicado unificar una definición compartida por todos.

Entre las múltiples definiciones de objeto de aprendizaje las más significativas son:

La IEEE⁵ define que un objeto de aprendizaje, es: “Cualquier entidad digital o no digital que puede ser usada, re-usada o referenciada para el aprendizaje soportado en tecnología”.(10)

Un objeto o conjunto de recursos que pueden ser utilizados para facilitar ciertos resultados educativos y ser extraídos y reutilizados en otros entornos educativos.(11)

De manera general un objeto de aprendizaje es una unidad didáctica en formato digital, preparado para su reutilización en varios contextos educativos por la inclusión de información auto descriptiva en forma de metadatos que apoya el proceso de enseñanza – aprendizaje y que puede ser unido a otros objetos creando uno nuevo.

⁵ Instituto de Ingenieros Eléctricos y Electrónicos

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

1.2.6. Estándares que permiten el intercambio de ejercicios

Internet apareció como medio de difusión y para facilitar el aprendizaje, permitiendo que disímiles instituciones hayan ingresado a la implementación y difusión de sistemas de enseñanza, propiciando una propagación de sistemas y recursos educativos siendo necesario establecer recomendaciones y estándares que permitan su uso eficaz.

La búsqueda de la homogeneidad en los sistemas y contenidos persigue:

- La reutilización efectiva de contenidos. Se trata de crear cursos con contenidos de alta calidad basados en el aprovechamiento de material previamente creado cuya calidad está contrastada.
- Capacidad de adaptabilidad de contenidos y adecuación del entorno de aprendizaje en función de los requisitos de conocimiento y preferencias del alumno, lo que permite aumentar el rendimiento del tiempo empleado en formación.(9)

1.2.6.1. Estándares para la interoperabilidad

La interoperabilidad facilita el intercambio de información, disponiendo de mecanismos que garanticen la integridad, consistencia y seguridad entre diferentes sistemas heterogéneos.

IMS QTI

La especificación de IMS para las Preguntas y Prueba de Interoperabilidad o Question & Test Interoperability (QTI) describe un modelo para la representación de preguntas, exámenes y sus correspondientes resultados, permitiendo el intercambio de éstos entre herramientas de autor, repositorios de objetos de aprendizaje y sistemas de gestión de contenidos educativos. La especificación se basa en un modelo de datos descrito en UML y con una traslación directa a XML para su intercambio.(12)

IMS QTI agrupa las tipologías de ejercicios en 20 interacciones atendiendo a como debe interactuar el usuario para dar respuesta a las mismas, proporciona un arsenal de herramientas que permiten crear métodos de evaluación con alto nivel de personalización, define un conjunto de plantillas de corrección en la que se han tenido en cuenta los casos típicos, permite presentar nueva información al alumno entre intentos haciendo uso del mecanismo de realimentación.(13)

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

SCORM

Es un estándar generado por la Distribución Avanzada de Aprendizaje (ADL por sus siglas en inglés). La propuesta de ADL dentro de su modelo SCORM persigue proporcionar un medio estándar para la interoperabilidad entre los objetos de contenidos y los sistemas de gestión de aprendizaje. Uno de sus requerimientos es que el contenido sea operativo en todos los entornos compatibles, sin tener en cuenta las herramientas que se utilicen para crear o usar los contenidos.

SCORM define un software que describe el modelo de agregación de contenidos, las interrelaciones establecidas entre los componentes de los cursos, los modelos de datos y los protocolos de comunicación, de manera que los objetos definidos en un LMS puedan compartirse entre diferentes LMS.

Selección de IMS QTI como estándar para el intercambio de ejercicios

IMS QTI como especificación destinada a la interoperabilidad de ejercicios, detalla las normas para la reutilización de los ejercicios como objeto de aprendizaje haciendo énfasis en los mensajes de retroalimentación. SCORM, por su parte, define una estructura genérica que permite el intercambio de objetos de aprendizaje, pero no tiene en cuenta el valor educativo de los mismos. IMS QTI es escalable mediante la interacción customInteraction para interacciones necesitadas en entornos específicos. La creación de plantillas para métodos de evaluación de las respuestas y la posibilidad de creación y personalización de estos métodos es otra de las características propuestas por IMS QTI para una evaluación estandarizada.

1.2.6.2. Estándares para los metadatos

Los metadatos son información añadida que enriquecen a los materiales digitales facilitando su clasificación y posterior recuperación. La especificación de los metadatos adecuados en materiales educativos es imprescindible a fin de añadir valor a los mismos, en vista de facilitar su reutilización.

Dublin Core

Es un modelo de metadatos elaborado y auspiciado por la Dublin Core Metadata Initiative (DCMI), una organización dedicada a fomentar la adopción extensa de los estándares interoperables de los metadatos que permitan una amplia gama de propósitos y modelos de negocio.

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

Este modelo fue diseñado específicamente para proporcionar un vocabulario de características "base", capaces de proporcionar la información descriptiva básica sobre cualquier recurso, sin que importe el formato de origen, el área de especialización o el origen cultural.

LOM

Metadatos para objetos de aprendizaje (LOM) es un modelo de datos, para describir objetos de aprendizaje y otros recursos digitales similares usados con el propósito de apoyar la reutilización de objetos de aprendizaje usualmente en el contexto de sistemas de aprendizaje online.

El estándar IEEE 1484.12.1:2002 sobre metadatos para objetos de aprendizaje es un estándar abierto internacionalmente reconocido (publicado por el Instituto de Ingenieros Eléctricos y Electrónicos) para la descripción de los "objetos de aprendizaje". Las cualidades relevantes de los objetos de aprendizaje que se describen incluyen: título, idioma, tipo de objeto, autor, propietario, términos de distribución, formato, copyright, y cualidades pedagógicas, tales como estilo de la enseñanza o de la interacción.

Los estándares del LOM se centran en el conjunto mínimo de propiedades que permiten que los objetos educacionales sean gestionados, ubicados y evaluados.(14)

Selección de LOM como estándar de metadatos

La principal ventaja de LOM sobre Dublin Core está en el mayor nivel de detalle en la especificación de los metadatos y la introducción de una categoría específica para caracterizar los propósitos educacionales de los contenidos, información clave en la reutilización de objetos de aprendizaje.

1.2.6.3. Estándar para empaquetamiento

La recolección y el empaquetado de los contenidos educativos en formato digital es un requisito básico para muchos de los procesos involucrados en el despliegue, gestión, distribución y agregación de dichos contenidos.

IMS CP

La especificación IMS CP empaquetamiento de contenido define un formato digital estándar para representar los paquetes de contenidos educativos facilitando la

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

interoperabilidad entre los sistemas de e-learning independientemente de la forma y el lugar en que dichos paquetes hayan sido producidos. Esta especificación proporciona una estructura donde se almacenan los metadatos del paquete (metadata), el orden y dependencia los objetos de aprendizajes (organizations), y los recursos que conforman los objetos de aprendizaje (resources).

1.3 Sistemas similares

El avance que se ha producido en los últimos tiempos en el área de las tecnologías de la información y las comunicaciones supone un reto y una oportunidad en el ámbito educativo. Uno de los aspectos más relevantes de este desarrollo está relacionado con la introducción e implementación de las llamadas plataformas de docencia virtual, basadas en Internet como medio de distribución. (15)

Las plataformas educativas permiten una mayor organización a los profesores mejorándole el trabajo en equipo en las diferentes materias, estructurar las clases de diferentes formas organizando grupos de trabajo colaborativo entre sus estudiantes y entre investigadores. A continuación se realizará una breve descripción de las más usadas destacando las principales dificultades de ellas.

Claroline

Es uno de los LMS usados en las universidades donde se aprecia su ambiente de aprendizaje colaborativo que permite a los enseñantes y a las instituciones educativas crear y administrar cursos en la Web. Las herramientas que ofrece el sistema son muchas (gestión de los grupos, fórum, repositorios de documentos, chat, administración del perfil de los usuarios, entre otras) y dan a los usuarios la posibilidad de establecer cualquier escenario deseado.

Las principales dificultades que presenta esta plataforma en el ámbito de la interoperabilidad de sus ejercicios y en el método utilizado para la gestión de los mismos son:

- La imposibilidad de utilizar imágenes en los ejercicios de enlazar.
- La inexistencia del uso de mensajes de retroalimentación, exceptuando en los ejercicios de selección simple.
- El soporte parcial que brinda a la especificación IMS-QTI v2.0, no tiene en cuenta interacciones necesarias en la plataforma (No cuenta con las interacciones de ordenar).

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

- La no existencia de la funcionalidad de importar un conjunto de ejercicios haciendo tedioso el trabajo de exportarlo uno a uno.(16)

Moodle

Es un sistema de gestión de cursos de código abierto, conocido también como Sistema de Gestión del Aprendizaje (Learning Management System, LMS) o como Entorno de Aprendizaje Virtual (Virtual Learning Environment, VLE). Es una aplicación web gratuita que los educadores pueden utilizar para crear sitios de aprendizaje efectivo en línea. Moodle dispone de características que le permiten escalar a grandes despliegues con cientos de miles de estudiantes, pero también puede ser utilizado en escuelas de educación infantil y primaria. Muchas instituciones lo utilizan como su plataforma para formación en línea, mientras que otras lo utilizan como apoyo a la formación presencial.

Principales dificultades presentes en la plataforma Moodle en la gestión e interoperabilidad de ejercicios:

- La no existencia de todas las interacciones requeridas en la plataforma propuestas por IMS QTI en su versión 2.0 (no cuenta con las interacciones de ordenar).
- No posee la funcionalidad que permita exportar ejercicios en formato IMS QTI v2.0.
- No gestiona ni almacena los metadatos relacionados con el ejercicio creado.(17)

Sakai

Típicamente la plataforma es utilizada para el manejo de cursos, es un ambiente de aprendizaje y colaboración que añade herramientas para el desarrollo de contenidos. Permite colocar presentaciones dentro de la plataforma, ofrece plantillas para diseñar lecciones de manera secuencial y para crear asignaciones y exámenes.

Principales dificultades presentes en la plataforma Sakai:

- El insuficiente uso de los mensajes de retroalimentación acorde a las necesidades de la plataforma y la especificación IMS-QTI.
- La no gestión y almacenamiento de metadatos relacionados con el ejercicio creado.

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

- El soporte parcial que brinda a la especificación IMS-QTI v2.0, no tiene en cuenta interacciones necesarias en la plataforma (No cuenta con las interacciones de ordenar). (18)

Las plataformas antes mencionadas han incursionado en la interoperabilidad de los ejercicios utilizando para ello la especificación IMS-QTI en su v2.0, pero aún de forma parcial y desestimando en ocasiones muchas de las características de la misma.

De manera general:

- Las plataformas analizadas implementan la importación de ejercicios utilizando el analizador sintáctico de PHP⁶ permitiéndole gestionar las etiquetas propias de la especificación, pero no la correcta estructura de las mismas siendo ésta una alternativa factible pero insuficiente para la implementación de la funcionalidad.
- La complejidad del estándar no permite que las plataformas lo implementen completamente, planteándose una arquitectura poco escalable.
- En resumen todas las plataformas analizadas poseen una base de datos relacional en la que cada tipología es almacenada en una o más tablas haciendo costosa e impracticable la escalabilidad de las mismas.

1.4 Metodologías de desarrollo de software

Se le denomina “Metodología” al proceso de desarrollo de software detallado y completo. Existen diversas propuestas las cuales inciden en cuan extenso sea el proceso. Se debe tener presente lo comprometido y difícil que es controlar el desarrollo de un software, por tanto, si deseamos obtener como resultado un producto de calidad es necesario aplicar una metodología adecuada. Una definición acertada, en correspondencia con el objetivo de la investigación es la abordada por Maddison en 1983 donde define metodología como “Conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de Sistemas de Información”. El proceso de desarrollo es guiado por las metodologías y la clave del éxito de un proyecto de software está en la correcta elección de la misma, conduciendo al desarrollador a lograr un buen sistema, incrementando la calidad y la productividad del producto.

⁶Hypertext Preprocessor

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

A continuación se realiza un estudio de las dos metodologías más utilizadas y documentadas con el objetivo de seleccionar una de ellas para el proceso que se desea desarrollar, teniendo en cuenta la magnitud del producto y peculiaridades del equipo de trabajo y de los clientes.

Proceso Unificado de Rational (RUP)

Es una metodología de desarrollo calificada dentro de las tradicionales o pesadas en la Ingeniería de Software, la cual va más allá del análisis y diseño orientado a objetos, proporcionando un conjunto de técnicas que soportan el ciclo completo de desarrollo de software, además de unificar completamente al equipo de desarrollo y optimizar el rendimiento de cada uno de sus miembros, favoreciendo a los jefes de proyecto a aumentar su experiencia en el desarrollo. Los aspectos determinantes RUP se resumen en que su ciclo de vida está caracterizado por ser: dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental. (19)

Extreme Programming (XP)

La programación extrema o Extreme Programming (XP) está caracterizada por ser una metodología ágil, la misma brinda al cliente el producto que requiere y en el momento requerido. Además de ser flexible y admitir cambios en el desarrollo del software siendo uno de los procesos ágiles más destacados, diferenciándose de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Tiene como meta la mengua de costes y su objetivo primordial es satisfacer a los clientes y potenciar al máximo el trabajo en grupo, además los jefes de proyecto, los desarrolladores y el cliente son parte del equipo de desarrollo. En el caso de los desarrolladores deben conocer que durante el ciclo de desarrollo de un proyecto de software los cambios aparecen, cambian los requisitos, las reglas del negocio, el personal y la tecnología, por lo tanto, éste no es el principal problema ya que va a suceder, el principal problema reside en la capacidad que tengan para adaptarse a esos cambios.

Selección de la metodología para solucionar el problema

La metodología XP está caracterizada principalmente por favorecer al máximo el trabajo en equipo. Uno de los inconvenientes para el equipo de trabajo es que tanto su líder de proyecto como sus clientes y desarrolladores forman parte de éste, encontrándose implicados en el desarrollo del producto final.

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

En cambio, la metodología RUP mantiene al equipo enfocado en la producción del software requerido permitiendo que se pueda cumplir con el tiempo estimado, con las características y calidad requerida, teniendo a su favor las mejor prácticas en la industria de software, incorporando tanto el conocimiento de sus líderes como la experiencia de una amplia gama de proyectos. La misma guía el proceso de desarrollo desde el inicio hasta el final, propiciando una documentación máxima en cada una de las fases. Cada artefacto generado en una fase sirven de entrada a la fase próxima, lo cual favorece la comunicación entre sus desarrolladores.

Se decide escoger la misma por todos los beneficios que favorecen su proceso, además se centra principalmente en la documentación del software exponiendo también un conjunto de actividades orientadas a visualizar, especificar, construir y documentar los artefactos necesarios para el desarrollo de software con la calidad que el cliente lo necesite, téngase en cuenta, que en el desarrollo de la plataforma ZERA los clientes no forman parte del equipo de desarrollo.

1.5 Lenguaje de Modelado

Lenguaje de Modelado Unificado (UML por sus siglas en inglés) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Actualmente es el más estandarizado por la industria, debido a que ha sido elaborado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh. Estos autores fueron contratados por la empresa Rational Software Co. para crear una notación unificada en la que base la construcción de sus herramientas CASE. En el proceso de creación de UML han participado, no obstante, otras empresas de gran peso en la industria como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupos de analistas y desarrolladores. (20)



Figura 1: UML

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

Este lenguaje es una notación que se fundamenta en principios de modelado, por lo que es importante para toda implementación de un sistema de información. El mismo se muestra mediante diagramas los cuales muestran múltiples vistas del sistema, esos conjuntos de vistas son conocidos como modelos. La utilización de los diagramas a utilizar en las diferentes etapas del desarrollo pueden cambiar teniendo en cuenta el tamaño y tipo de sistema, lo cual es necesario organizarlos según las fases del proceso unificado.

Características principales 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, etc.)
- 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.

UML proporciona a los desarrolladores un vocabulario que incluye tres categorías: elementos, relaciones y diagramas.(19)

1.6 Herramientas CASE para el modelado UML

Las herramientas de Ingeniería de Software Asistida por Computador (CASE), son recomendadas por la Ingeniería de Software para llevar a cabo el Análisis y Diseño de cualquier Sistema de Información, las cuales proporcionan un soporte automático para representar el proceso y los métodos de un sistema. Sus objetivos son:

- Permitir la aplicación práctica de metodologías, lo que resulta indispensable en el uso de estas herramientas.
- Facilitar la realización de prototipos y el desarrollo conjunto de aplicaciones Simplificar el mantenimiento del Software.
- Mejorar y estandarizar la documentación.
- Aumentar la portabilidad de las aplicaciones.
- Facilitar la Reutilización de componentes de software.

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

- Permitir un desarrollo y un refinamiento visual de las aplicaciones, mediante la utilización de controles gráficos (código reutilizable).

Dia

Dia, más que una herramienta de modelado UML es una aplicación informática para crear diagramas y mapas conceptuales. Se trata de una aplicación de código abierto. Es un programa intuitivo y ligero, que pone el acento en la simplicidad y la eficiencia. Para utilizarlo no se requiere ni un ordenador potente ni grandes conocimientos técnicos. En la parte negativa, se encuentra el soporte para generación de código, que aunque sí existe, es mediante programas externos que "parsean" el archivo que Dia genera. También presenta el inconveniente de que el código se ha de generar sólo cuando todo está correcto y completo, en la práctica esto no suele ser así.

Visual Paradigm

Es una herramienta de diseño profesional que soporta todos los diagramas del ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Visual Paradigm para UML ofrece amplias características de modelado de casos de uso incluyendo la función completa de UML Diagrama de casos de uso, flujos de evento editor, caso de uso y la generación de un diagrama de actividad. Permite producir la documentación del sistema con diseñador de plantilla, dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Los analistas del sistema pueden estimar las consecuencias de los cambios en los diagramas de análisis de impacto, tales como la matriz y el diagrama de análisis.

Selección de la herramienta CASE para el modelado UML para solucionar el problema.

Después haber realizado el estudio de las herramientas CASE se establece que la más idónea a utilizar es Visual Paradigm. Se llega a esta conclusión por las ventajas que esta presenta, que permitirá realizar los diagramas y artefactos que se generan durante el desarrollo del software, con una interfaz de usuario fácil de usar por lo que considera como la herramienta más adecuada para el desarrollo del Módulo QTI de la plataforma ZERA, la misma posee capacidades similares a Dia pero cuenta con mayor integración con las demás herramientas y estándares escogidos. Se utilizará la versión 6.4.

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

1.7 Herramientas y lenguajes de Desarrollo

1.7.1. Lenguajes de desarrollo

XHTML

El lenguaje eXtensible Hypertext Markup Language (XHTML) v1.0 es un lenguaje de marcas para crear documentos con hipertexto (documentos formados por enlaces a otros documentos, imágenes, audios, videos y aplicaciones), base fundamental del intercambio de información en Internet y una de las herramientas utilizadas por Tim Berners Lee junto con el protocolo TCP para la creación de la World Wide Web. Este lenguaje nace precisamente con el objetivo de remplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML y que va a permitir una correcta interpretación de la información independientemente del dispositivo desde el que se accede. La versión actual XHTML 1.0 aun no elimina todas las fallas del html 4.1 se centra principalmente en el incremento de las restricciones semánticas de los documentos.(21)

CSS (Cascading Style Sheets)

Las hojas de estilo en cascada son un lenguaje centrado en la presentación de los contenidos de los documentos electrónicos, siendo así el mecanismo que permite separar los contenidos definidos mediante XHTML y el aspecto que deben presentar esos contenidos logrando que los XHTML creados sean más flexibles, fácil de mantener ya que se adaptan mejor a las diferentes plataformas: pantallas de ordenador, pantallas de dispositivos móviles, impresoras, dispositivos utilizados por personas discapacitadas.

1.7.2. Lenguajes de script del lado del cliente

JavaScript

Es un lenguaje de programación que se utiliza principalmente para dar dinamismo a las páginas web. 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. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos, así se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios, esto evita tener que compilar cada vez que se haga un cambio pero ralentiza el proceso ya que se debe interpretar cada una de las líneas cada vez que se ejecuta; JavaScript es el lenguaje de script por

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

defecto de la mayoría de los navegadores, es multiplataforma y el más popular de los lenguajes de script. A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java. Legalmente, JavaScript es una marca registrada de la empresa Sun Microsystems.

VBScript

También conocido como Visual Basic Scripting es similar a los lenguajes Visual Basic que utilizan los desarrolladores de Windows y tiene esencialmente la misma estructura que JavaScript. VBScript es el lenguaje de scripting desarrollado por Microsoft para páginas web con el que se pueden realizar efectos para el navegador Internet Explorer por lo que en principio es una ventaja para programadores experimentados en este sistema, pero una desventaja para los demás. El VBScript, como el Visual Basic del que deriva, incorpora la Programación Orientada a Objeto (POO), aunque en menor grado que otros lenguajes actuales. En realidad la estructura de este lenguaje es, hasta cierto punto, un poco anárquica, vestigio de las antiguas versiones de Basic, que eran totalmente procedimentales. Al contrario de otros lenguajes de programación, en VB solo existe un tipo general de datos que se conoce con el nombre de Variant. En otros lenguajes existen datos de tipo String (Cadena) para almacenar contenidos alfanuméricos, distintos tipos de datos numéricos enteros y en coma flotante, datos booleanos, etc. (22)

Fundamentación de la selección de JavaScript como lenguaje de script del lado del cliente.

- Al contrario de VBScript es apto para todos los navegadores convirtiéndolo en un lenguaje capaz de ser utilizado en varias plataformas.
- Puede ser utilizado el paradigma de programación orientado a objetos.
- Es un potente y popular lenguaje estandarizado.

1.7.3. Lenguajes del lado del servidor

Python

Es un lenguaje de programación, manejado como proyecto de software libre, de propósito general, o sea, puede ser usado tanto para aplicaciones de escritorio como para aplicaciones del lado del servidor. Es muy potente y compatible con muchos otros lenguajes de los cuales utiliza sus fortalezas, multiplataforma y multiparadigma, esto último permite a los programadores adoptar un estilo de programación particular.

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

Python es visto como un lenguaje exitoso debido a su facilidad de aprendizaje, su orientación a programadores promedio y la limpieza de su código.

Perl (Practical Extracting and Reporting Language)

Perl es libre y gratuito. Actualmente disponible para sistemas operativos de tipo Unix y Windows. Perl es un lenguaje eficiente y completo, está enfocado hacia un desarrollador con altos conocimientos del lenguaje y es ideal para el desarrollo rápido de aplicaciones. Soporta tanto la programación estructurada como la programación orientada a objetos y la programación funcional. Una de las grandes ventajas del Perl es la amplia comunidad de usuarios donde se hace muy sencillo plantear y resolver las dudas que surjan en el desarrollo de una aplicación.

PHP

Es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. Al ser un lenguaje libre dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas web dinámicas. (26) PHP es multiplataforma, permite la creación o uso de binarios ejecutables por parte del propio usuario, mantiene un bajo consumo de recursos hardware y software, tiene alto grado de seguridad, soporta en cierta medida la orientación a objeto (Clases y herencia).

Como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores que permite que los fallos de funcionamiento se encuentren y reparen rápidamente.

El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP. Su diseño modular le permite expandir su potencial utilizando una enorme cantidad de librerías.

Soporte nativo para una gran cantidad de bases de datos:

- MySQL
- Oracle
- PostgreSQL
- MSSQL Server
- DBM
- DBase

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

- Frontbase
- filePro
- SQLite

Fundamentación de la selección de PHP como lenguaje del lado del Servidor.

- Posee mayor velocidad de ejecución con respecto a Perl y Python.
- Es multiplataforma y soporta POO (Programación Orientada a Objetos).
- Es un lenguaje específicamente diseñado para realizar aplicaciones web, mientras que otros lenguajes son adaptaciones de lenguajes preexistentes, no pensados para la web.
- Trabaja en combinación con otras tecnologías: Perl, javaScripty DHTML.
- Excelente soporte de acceso a base de datos.

1.7.4. Frameworks

Los frameworks son arquitecturas genéricas integradas por un extensible conjunto de componentes. En otras palabras un framework es un conjunto de librerías y componentes de probada capacidad, junto con una documentación y metodología de uso, que permite diseñar, construir e implantar de forma más uniforme, rápida, y con mayor calidad. Además, simplifican el desarrollo de las aplicaciones mediante la automatización de muchas de las tareas comunes, proporcionan estructura al código fuente, forzando al programador a crear códigos más legibles y más fácil de mantener.

Jquery

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. 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.

Posee una versión compacta que recoge toda la implementación en un solo fichero, haciéndolo más fácil de cargar. Existe una librería complementaria JQuery UI⁷ que se encarga de la creación de elementos visuales con poderosas características como son

⁷ Interfaz de Usuario

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

los acordeones, los calendarios, además de permitir los eventos de arrastrar y soltar. Ambas librerías son completamente escalables por terceros. Posee una extensa documentación en idioma español y una gran comunidad tanto de uso como de desarrollo lo que permite que cada día se mejoren sus funcionalidades.

Symfony

Symfony es un framework PHP que facilita el desarrollo de las aplicaciones web. Se encarga de todos los aspectos comunes y aburridos de las aplicaciones web, dejando que el programador se dedique a aportar valor desarrollando las características únicas de cada proyecto.

Entre sus más relevantes características están:

- **Escalable:** Es infinitamente escalable si se disponen de los recursos necesarios.
- **Probado:** Ha sido probado con éxito durante años en varias aplicaciones gigantescas (Yahoo! Answers, Dailymotion, delicious) y en otros miles de sitios pequeños y medianos.
- **Licencia:** Se publica bajo licencia MIT, con la que puedes desarrollar aplicaciones web comerciales, gratuitas y/o de software libre.
- **Seguro:** Permite controlar hasta el último acceso a la información e incluye por defecto protección contra ataques XSS y CSRF.
- **Documentado:** Es un framework bien documentado, habiéndose publicado cinco libros gratuitos de calidad y siempre actualizados.
- **Calidad:** Su código fuente incluye más de 9.000 pruebas unitarias y funcionales.
- **Internacionalización:** Está traducido a más de cuarenta idiomas e incluye todas las herramientas necesarias para que traduzcas fácilmente tus aplicaciones.(23)
- **Fácil de instalar y configurar** en sistemas Windows, Mac y Linux
- **Flexible** hasta cualquier límite y **extensible** mediante un completo mecanismo de plugins.

1.8 Gestor de Bases de Datos

Un Sistema Gestor de Bases de Datos (SGBD) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos.(24)

1.8.1. PostgreSQL

PostgreSQL es el servidor de bases de datos de código abierto, liberado bajo la licencia BSD más potente que existe. Soporta características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores (triggers) y procedimientos almacenados, lo que significa que podría implementar la lógica del negocio en el servidor, tiene lenguaje procedimental (PL), de alta disponibilidad y recuperación de fallas. Además, soporta gran cantidad de información y se puede utilizar sobre cualquier sistema operativo.

Características

- Integridad referencial
- Replicación (soluciones comerciales y no comerciales) que permiten la duplicación de bases de datos maestras en múltiples sitios de réplica
- Interfaces nativas para ODBC, JDBC, C, C++, PHP, Perl, TCL, ECPG, Python y Ruby
- Procedimientos almacenados
- Soporte nativo SSL
- Herramientas para generar SQL portable para compartir con otros sistemas compatibles con SQL
- Sistema de tipos de datos extensibles para proveer tipos de datos definidos por el usuario, y rápido desarrollo de nuevos tipos
- Funciones de compatibilidad para ayudar en la transición desde otros sistemas menos compatibles con SQL
- También soporta el almacenamiento de grandes objetos binarios, incluyendo imágenes, sonidos o vídeo.
- Soporta trabajo con xml y json (Java Script Object Notation) nativamente. (25)

1.8.2. MySQL

Se trata de un sistema de gestión de base de datos relacional, multihilo y multiusuario. Constituye el servidor de bases de datos relacionales más popular y es desarrollado por la empresa MySQL AB, la cual desde enero del 2008 pertenece a Sun Micro

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

systems. MySQL es desarrollado como software libre siguiendo un esquema de licenciamiento dual, o sea, se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia. Sin embargo, si no se está de acuerdo con la misma o se necesita incorporar MySQL en una aplicación comercial es posible comprar una licencia específica que permita este uso.

Fundamentación de la selección de PostgreSQL como Sistema Gestor de Bases de Datos.

- Es un sistema que puede ser utilizada en plataformas UNIX y en Windows
- Tiene soporte para distintos tipos de datos como xml y json, además permite añadir tipos de datos personalizados.
- Es un sistema de gestión de base de datos de código abierto
- MySQL cuenta con dos licencias haciendo obligatorio la compra de licencia para su uso comercial.

1.9 Servidores Web

Un servidor web es un programa que se ejecuta continuamente en una computadora, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de Internet. Este se encarga de contestar y en algunos casos procesar estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo con los comandos solicitados.

Apache

Apache ha sido el servidor web más popular en Internet, que en Enero del 2011 llegó a ser el servidor preferido por más del 59% de los servidores de internet, encontrándose muy por encima de sus competidores, tanto gratuitos como comerciales. Apache cuenta con soporte para CGI (Common Gateway Interface) y aporta características extendidas como personalización de las variables de entorno y soporte de reparación de errores o debugging, que son difíciles de encontrar en otros servidores web, es capaz de interpretar script PHP, permite crear fácilmente un sitio Web SSL utilizando OpenSSL y el módulo mod-ssl..

Nginx

Nginx es un servidor HTTP y proxy inverso de alto rendimiento. Nginx fue desarrollado por Igor Sysoev para Rambler.ru, el segundo sitio web más visitado de Rusia, donde

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

ha estado funcionando en producción más de dos años y medio. Igor ha lanzado el código fuente bajo una licencia estilo BSD. Aunque aún se encuentra en una etapa beta, Nginx es conocido por su estabilidad, gran conjunto de características, configuración simple, y bajo consumo de recursos. (26)

Sitios Web que usan Nginx en la lista de Alexa Top 500 (27)

- WordPress
- FC2
- TorrentReactor
- Ultimate Guitar
- Scribd
- Badoo
- WatchMovies

Fundamentación de la selección de Apache como Servidor Web

Se realiza la selección de Apache como servidor web debido a algunas características, tales como: multiplataforma, contención de un diseño modular altamente configurable y expandible por cualquier persona con conocimiento de C o Perl, poseer tecnología gratuita de código fuente abierto, así como su popularidad, robustez y la obligación de pago por el servidor de Microsoft y la no venta directa de este tipo de software a Cuba.

Desarrollador	Diciembre 2010	Porcentaje	Enero 2011	Porcentaje	Cambio
Apache	151,516,152	59.35%	161,591,445	59.13%	-0.23
Microsoft	56,723,544	22.22%	57,392,351	21.00%	-1.22
nginx	16,910,205	6.62%	20,504,634	7.50%	0.88
Google	14,933,865	5.85%	15,112,532	5.53%	-0.32
lighttpd	1,308,935	0.51%	1,866,872	0.68%	0.17

Tabla 1: Estadísticas sobre uso de servidores web (28)

1.10 Entorno de Desarrollo Integrado

Un Entorno de Desarrollo Integrado (IDE) es una herramienta que permite escribir, compilar, depurar, ejecutar programas y que brinda herramientas para la disminución de la escritura de código y automatización de procesos comunes.

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

NetBeans IDE

NetBeans IDE es un producto libre y gratuito sin restricciones de uso, basado en plugins, permitiendo ser ampliado por terceros y ser utilizado para tantos lenguajes como se desee. Tiene un amplio soporte para PHP brindando un sistema de completamiento de código inteligente y amplio. Brinda todas las opciones de un cliente de accesos a servidores de control de versiones, permitiendo estar trabajando en todo momento con la última versión del fichero en el servidor. Permite la ejecución de todos los comandos propios del framework para PHP symfony convirtiéndose de esta manera en un sistema independiente de cualquier otra aplicación. Tiene la posibilidad de creación de plantillas de código que disminuye el tiempo de escritura de segmentos de código a una simple combinación de tecla, así como la configuración de un estándar de codificación incluyendo la aventación para su aplicación que aplica a todo el proyecto con un simple clic.(29)

1.11 Patrones

“Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo siquiera dos veces de la misma forma”(30)

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

Alta Cohesión: Symfony permite la organización del trabajo en cuanto a la estructura del proyecto y la asignación de responsabilidades con una alta cohesión. Un ejemplo de ello es la clase Actions, la cual está formada por varias funcionalidades que están estrechamente relacionadas, siendo la misma la responsable de definir las acciones para las plantillas y colaborar con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades.

Bajo Acoplamiento: Las clases Actions heredan únicamente de sfActions para alcanzar un bajo acoplamiento de clases. Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja.

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

Controlador: Todas las peticiones Web son manipuladas por el controlador frontal y las Actions, que es el punto de entrada único de toda la aplicación en un entorno determinado.

Creador: En la clase sfActions se encuentran las acciones definidas para el sistema y se ejecutan en cada una de ellas. En dichas acciones se crean los objetos de las clases que representan las entidades, lo que evidencia que la clase Actions es “creador” de dichas entidades.

Experto: Es uno de los patrones que más se utiliza cuando se trabaja con Symfony, con la inclusión de la librería Doctrine para mapear la Base de Datos. Symfony utiliza esta librería para realizar su capa de abstracción en el modelo, encapsular toda la lógica de los datos y generar las clases con todas las funcionalidades comunes de las entidades, las clases de abstracción de datos poseen un grupo de funcionalidades que están relacionadas directamente con la entidad que representan y contienen la información necesaria de la tabla que representan.

Patrones GOF (Banda de los Cuatro)

Patrón Decorator: Este método pertenece a la clase abstracta sfView, padre de todas las vistas, que contienen un decorador para permitir agregar funcionalidades dinámicamente. El archivo nombrado layout.php es el que contiene el layout de la página. Este archivo, conocido también como plantilla global, guarda el código HTML que es usual en todas las páginas del sistema, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el layout, o si se mira desde el otro punto de vista, el layout decora la plantilla. Este procedimiento es una implementación del patrón Decorator.

Patrón Singleton: Guarda una referencia a todos los objetos que constituyen el núcleo de Symfony y puede ser accedido desde cualquier parte de la aplicación.(31)

Patrón arquitectónico Modelo Vista Controlador en Symfony

Symfony está basado en el patrón arquitectónico Modelo Vista Controlador (MVC). Este patrón divide en tres niveles las funcionalidades de una aplicación con el objetivo de incrementar la usabilidad de las mismas. Estos niveles son:

Modelo: Esta es la representación específica de la información con la cual el sistema opera. En resumen, el modelo se limita a lo relativo de la vista y su controlador

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos no relativos a la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado.

Vista: Esta presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Controlador: Esta responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista.

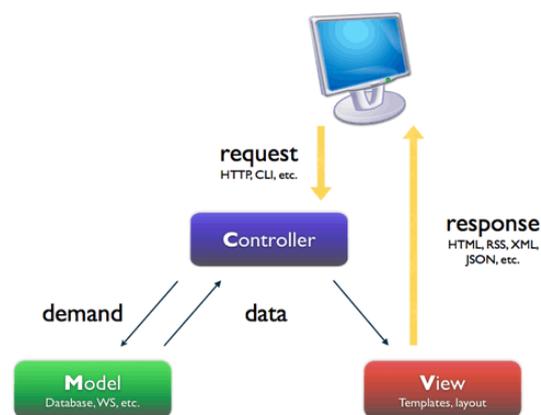


Figura 2: Representación gráfica del MVC.

En el módulo desarrollado el patrón MVC se manifiesta de la forma siguiente: las clase action del módulo QTI junto al controlador frontal son la representación de la capa controlador atendiendo todas las peticiones hechas a la aplicación y seleccionando las acciones y/o funciones a ejecutar, para actualizar la vista, conformada por las plantillas (templates) y los parciales (partial) archivos PHP cuyo objetivo es generar el código xhtml que será mostrado al usuario con datos que puede o no provenir de una solicitud a la capa de modelo que están conformadas por las clases de acceso a datos y contienen la lógica del negocio.

1.12 Conclusiones parciales

El proceso educativo de manera general está siendo apoyado por las nuevas tecnologías, creando nuevos conceptos como LMS, e-Learning, Hiperentornos de aprendizajes, así como a estándares, que incrementan la capacidad del mismo. Después de estudiar los conceptos anteriores y los nuevos paradigmas de desarrollo de software se identificaron los lenguajes y tecnologías que facilitarían la creación de la aplicación.

Características del sistema

2.1. Introducción

Para realizar el diseño de un sistema informático es necesario aclarar todos los conceptos y sus relaciones utilizando un lenguaje perceptible, mediante la descripción de la propuesta de solución facilitando la comprensión de la misma por todos los involucrados. La poca claridad de los procesos en el entorno donde estará el sistema es uno de los motivos para la realización del modelo del dominio en el cual se capturarán los objetos existentes y eventos que suceden en el mismo, permitiendo esto la identificación de los requisitos funcionales y no funcionales así como los casos de uso críticos que regirán el desarrollo del sistema.

2.2. Modelo de Dominio

Una vez analizado el problema e identificados los principales conceptos presentes en el mismo, se hace posible el empleo de éstos en la elaboración del Modelo de Dominio donde se presenta un marco conceptual y la relación existente entre las definiciones.

2.2.1. Análisis de los conceptos del Dominio

Para lograr una mayor comprensión del diagrama de Modelo de Dominio que se expondrá a continuación, es necesario realizar previamente la definición de los conceptos que involucra dicho modelo:

Gestor de Ejercicio: Asigna actividades al grupo o individualmente, hacer seguimiento a las actividades de todos los estudiantes que estén a su cargo.

Ejercicio: Actividad mental que consiste en la repetición de ciertas rutinas, con el fin de desarrollar una determinada habilidad. Los ejercicios permiten comprobar el nivel asimilación de un contenido por parte de un estudiante

Tipología de ejercicio: Es el formato en el que se puede presentar un determinado ejercicio. Las mismas pueden ser: Selección simple, Selección múltiple, Completar por escritura, Completar por desplazamiento, Ordenar Procedimiento, Ordenar Medias, Enlazar, Clasificar Texto, Clasificar Media.

Imagen: Representación gráfica que sirve de apoyo al ejercicio.

Materia: Son las asignaturas que forman un programa de estudio en la plataforma.

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

Metadatos: Información adicional añadida al Ejercicio para describir los datos del mismo, con el objetivo de localizar su ubicación en bases de datos globales y locales facilitando reutilización e interoperabilidad.

Retroalimentación: Mensaje interactivo del Ejercicio que brindará al estudiante información adicional para mejor comprensión del mismo.

Mensajes: Mensajes de interacción para un Ejercicio, el cual le servirá de guía al estudiante para la realización del mismo.

2.2.2. Diagrama del Modelo de Dominio

El Modelo de Dominio es una representación visual estática del entorno real objeto del proyecto. Es decir, un diagrama con los objetos que existen reales relacionados con el módulo que está en construcción y las relaciones que existen en él. Pero no son clases de software, aunque algunos objetos del Modelo de Dominio pueden terminar siéndolo. Se dice que es estática porque no representa la interacción en el tiempo de los objetos, sino que representa una visión "parada" de las clases y sus interacciones.

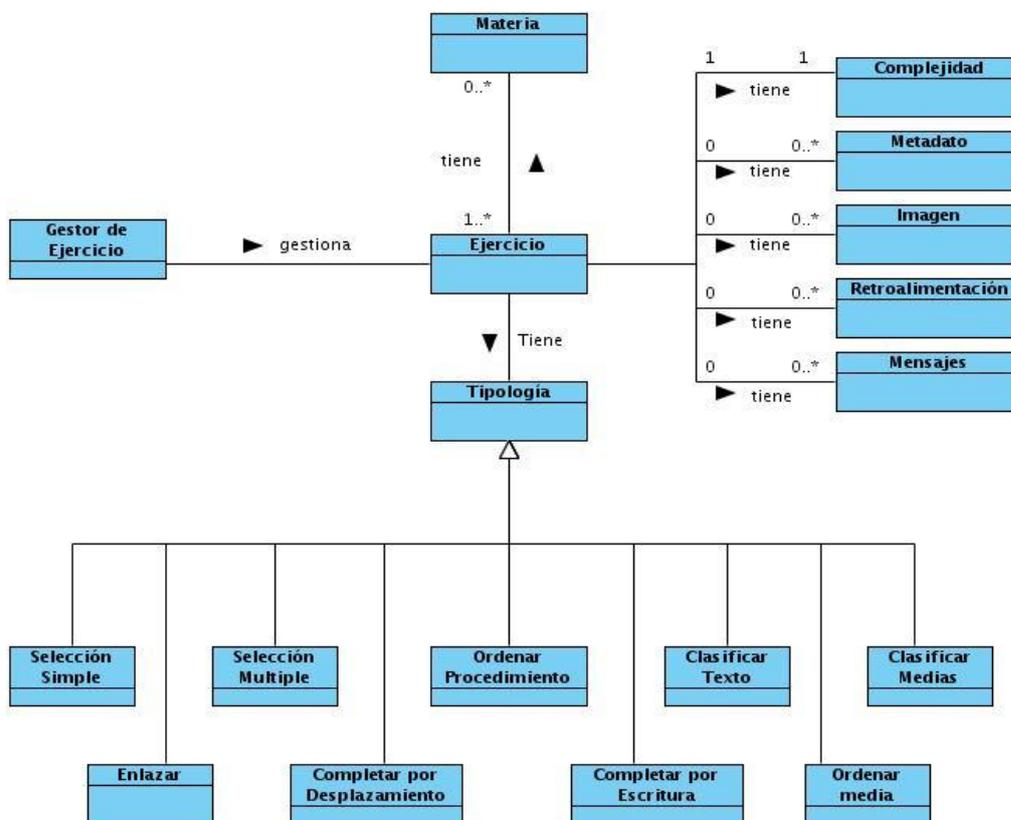


Figura 3. Representación del Modelo de Dominio.

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

2.3. Descripción del Sistema Propuesto

El objetivo del módulo propuesto es la gestión de los ejercicios, permitiendo a los docentes realizarles evaluaciones digitales a los estudiantes para tener conocimiento de su avance en los diferentes temas impartidos, así como adicionarle la interoperabilidad mediante la especificación IMS QTI v2.0.

Requerimientos del software

Los requerimientos de software son las condiciones o capacidades que debe cumplir el sistema según a las necesidades del cliente.

Requerimientos Funcionales

Los requerimientos funcionales (RF) son aquellos requisitos que, desde el punto de vista de las necesidades del usuario, debe cumplir el sistema, los cuales no varían por sus propiedades o cualidades relacionadas.

Para cumplir con los objetivos propuestos se prevé que el sistema tenga las siguientes funcionalidades:

RF 1. Gestionar Ejercicios

RF 1.1. Insertar un Ejercicio: Se le brinda al usuario la posibilidad de introducir o seleccionar los siguientes datos:

- Título.
- Descripción.
- Enunciado.
- Complejidad.
 - Muy Fácil.
 - Fácil.
 - Media.
 - Difícil.
 - Muy Difícil.
- Tipología.
 - Selección simple.
 - Selección múltiple.
 - Completar por escritura.
 - Completar por desplazamiento.
 - Ordenar media.
 - Ordenar procedimiento.
 - Enlazar.
 - Clasificar texto
 - Clasificar media

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

RF 1.2. Ver los datos del Ejercicio: Se le permite ver los datos del Ejercicio seleccionado.

RF 1.3. Modificar los datos de un Ejercicio: Se le permite al usuario en dependencia de sus permisos modificar los datos del Ejercicio.

RF 1.4. Eliminar un Ejercicio: Se le permite al usuario en dependencia de sus permisos eliminar un Ejercicio que haya sido previamente seleccionado.

RF 2. Listar Ejercicios

RF 2.1. Mostrar un listado de Ejercicios: Se muestra un listado de los Ejercicios que ya han sido creados.

RF 2.2. Filtrar Ejercicio: Se le permite al usuario la posibilidad de realizar un filtro de los ejercicios

RF 2.3. Exportar Ejercicio: Se le brinda al usuario la posibilidad de exportar uno o varios ejercicios en forma de paquete.

RF 2.4. Edición Rápida del Ejercicio: Se le permite al usuario en dependencia de sus permisos modificar los siguientes datos del Ejercicio:

- Título.
- Descripción.
- Enunciado.
- Complejidad.
- Imagen de Ayuda.

RF 3. Gestionar Retroalimentación⁸

RF 3.1. Insertar Retroalimentación: Se le brinda al usuario la posibilidad de introducir la retroalimentación que ayudará a la realización del ejercicio.

RF 3.2. Modificar datos de Retroalimentación: Se le permite al usuario en dependencia de sus permisos modificar los datos de la retroalimentación.

RF 3.3. Eliminar la Retroalimentación: Se le permite al usuario en dependencia de sus permisos eliminar una retroalimentación.

⁸ La retroalimentación no es más que mensajes afectivos que se le muestran al estudiante combinado con lo cognitivo siendo positivos o negativos dependiendo del éxito alcanzado en la pregunta, que revelen la intencionalidad didáctica que determinó la formulación de la misma.

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

RF 4. Gestionar Mensaje

RF 4.1. Insertar un Mensaje: Se le brinda al usuario la posibilidad de introducir los mensajes de apoyo al ejercicio.

RF 4.2. Modificar datos del Mensaje: Se le permite al usuario en dependencia de sus permisos modificar los datos del mensaje.

RF 4.3. Eliminar el Mensaje: Se le permite al usuario en dependencia de sus permisos eliminar un mensaje.

RF 5. Gestionar Tipología Selección Simple

RF 5.1. Insertar Tipología Selección Simple: Se le brinda al usuario la posibilidad de introducir los datos de la tipología selección simple.

RF 5.2. Modificar Tipología Selección Simple: Se le permite al usuario en dependencia de sus permisos modificar los datos de la tipología selección simple.

RF 6. Gestionar Tipología Selección Múltiple

RF 6.1. Insertar Tipología Selección Múltiple Se le brinda al usuario la posibilidad de introducir los datos de la tipología selección múltiple.

RF 6.2. Modificar Tipología Selección Múltiple: Se le permite al usuario en dependencia de sus permisos modificar los datos de la tipología selección múltiple.

RF 7. Gestionar Tipología Clasificar Media

RF 7.1. Insertar Tipología Clasificar Media: Se le brinda al usuario la posibilidad de introducir los datos de la tipología clasificar media.

RF 7.2. Modificar Tipología Clasificar Media: Se le permite al usuario en dependencia de sus permisos modificar los datos de la tipología clasificar media.

RF 8. Gestionar Tipología Clasificar texto

RF 8.1. Insertar Tipología Clasificar texto: Se le brinda al usuario la posibilidad de introducir los datos de la tipología clasificar texto.

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

RF 8.2. Modificar Tipología Clasificar texto: Se le permite al usuario en dependencia de sus permisos modificar los datos de la tipología clasificar texto.

RF 9. Gestionar Tipología Completar por desplazamiento

RF 9.1. Insertar Completar por desplazamiento: Se le brinda al usuario la posibilidad de introducir los datos de la tipología completar por desplazamiento.

RF 9.2. Modificar Completar por desplazamiento: Se le permite al usuario en dependencia de sus permisos modificar los datos de la tipología completar por desplazamiento.

RF 10. Gestionar Tipología Completar por escritura

RF 10.1. Insertar Tipología Completar por escritura Se le brinda al usuario la posibilidad de introducir los datos de la tipología completar por escritura.

RF 10.2. Modificar Tipología Completar por escritura Se le permite al usuario en dependencia de sus permisos modificar los datos de la tipología completar por escritura.

RF 11. Gestionar Tipología Enlazar

RF 11.1. Insertar Tipología Enlazar: Se le brinda al usuario la posibilidad de introducir los datos de la tipología enlazar.

RF 11.2. Modificar Tipología Enlazar: Se le permite al usuario en dependencia de sus permisos modificar los datos de la tipología enlazar.

RF 12. Gestionar Tipología Ordenar Media

RF 12.1. Insertar Tipología Ordenar Media: Se le brinda al usuario la posibilidad de introducir los datos de la tipología ordenar media.

RF 12.2. Modificar Tipología Ordenar Media: Se le permite al usuario en dependencia de sus permisos modificar los datos de la tipología ordenar media.

RF 13. Gestionar Tipología Ordenar Procedimiento

RF 13.1. Insertar Tipología Ordenar Procedimiento: Se le brinda al usuario la posibilidad de introducir los datos de la tipología ordenar procedimiento.

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

RF 13.2. Modificar Tipología Ordenar Procedimiento: Se le permite al usuario en dependencia de sus permisos modificar los datos de la tipología ordenar procedimiento.

RF 14. Importar Ejercicio: Se le permite al usuario la posibilidad de importar un paquete de ejercicios.

Requerimientos No Funcionales

Los requisitos no funcionales son propiedades o cualidades que el sistema debe tener, para hacer al sistema atractivo, usable, rápido y confiable.

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 con al menos 512 MB de RAM y 1 GB de espacio en disco duro.

Restricciones en el diseño y la implementación

- Lenguaje de programación a usar debe ser PHP 5.3.3.
- El marco de trabajo base de desarrollo que se utilizará es Symfony 1.4.11.
- Como IDE se empleará NetBeans 6.8.
- Como Servidor Web se explotará Apache 2.2.16.
- El SGDB deberá ser PostgreSQL 8.4.

Apariencia o interfaz externa

- Debe contener un diseño sencillo, permitiendo que no sea necesario mucho entrenamiento 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 de ésta.

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.

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

- 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 para minimizar los pasos a dar en cada proceso.

Legales, Derecho de Autor y otros

- Una vez terminado el producto, el sistema debe ser sometido a una evaluación y certificación por parte del cliente del mismo.

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

Describe las funcionalidades del sistema propuesto. Un Caso de Uso (CU) representa los flujos de eventos que ocurren al interactuar un usuario y el sistema; puede "incluir" o "extender" la funcionalidad de otro con su propio comportamiento.

Patrones de casos de uso

Son comportamientos que deben existir en el sistema, facilitan la descripción de lo que el sistema debe hacer, es decir, describen el uso del sistema y su interacción con los usuarios. Se utilizan generalmente como plantillas para describir la estructura y

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

organización que deberían tener los casos de uso. Los mismos capturan mejores prácticas para la modelación de los casos de uso.

Patrón CRUD

El patrón CRUD Completo consiste en un caso de uso para administrar la información, permite modelar las diferentes operaciones para administrar una entidad de información, tales como incluir, ver, modificar y eliminar.

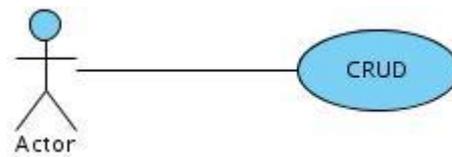


Figura 4: Patrón CRUD

Concordancia

Extrae una subsecuencia de acciones que aparecen en diferentes lugares del flujo de casos de uso y es expresado por separado. La variante que está presente en el modelo de CU es:

Reuso

Consiste en dos casos de uso y una relación extendida entre ellos. Puede ser instalado en sí mismo, así como extendido en el caso de uso base. El referente puede ser concreto o abstracto. Este patrón se aplica cuando un flujo puede extender el flujo de otro caso de uso así como ser realizado en sí mismo. Un ejemplo de su uso se evidencia en la siguiente figura:

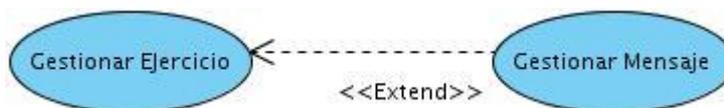


Figura 5: Extensión

Múltiples actores

Se utiliza cuando dos actores jueguen el mismo rol sobre el CU. Este rol se representa por otro actor, heredado por los actores que comparten este rol. Es aplicable cuando, desde el punto de vista del caso de uso, solo exista una entidad externa interactuando con cada una de las instancias del caso de uso.

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

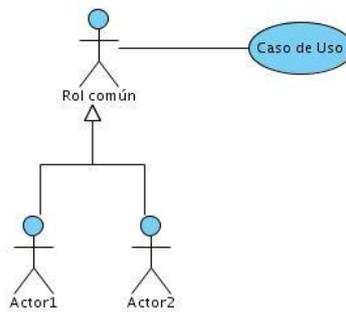


Figura 6: Múltipleactores

2.4.1. Descripción de los actores del sistema

Un actor del sistema está definido como el individuo, grupo de individuos, sistema automatizado o máquina que interactúa con el mismo intercambiando información.

Actor	Descripción
Gestor de Ejercicio	Es el encargado de crear y/o actualizar los distintos tipos de ejercicios así como incluirlos e importarlos. Es una generalización de los actores Administrador central y Editor Digital.
Administrador Central	Se encarga de la administración global del sitio y tiene permiso sobre todos los componentes de la plataforma.
Editor Digital	Carga y edita los ejercicios.

Tabla 2: Descripción de los actores del sistema.

2.4.2. Diagrama de actores del sistema

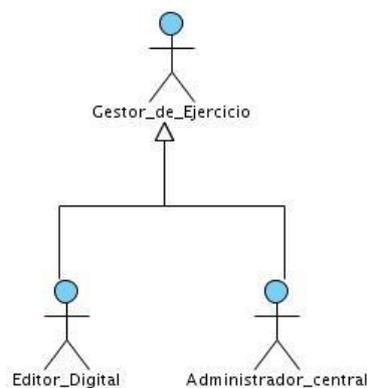


Figura 7: Diagrama de Actores.

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

2.4.3. Diagrama de casos de uso del sistema

Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas. Es decir, un diagrama de casos de uso del sistema representa gráficamente a los procesos y su interacción con los actores.

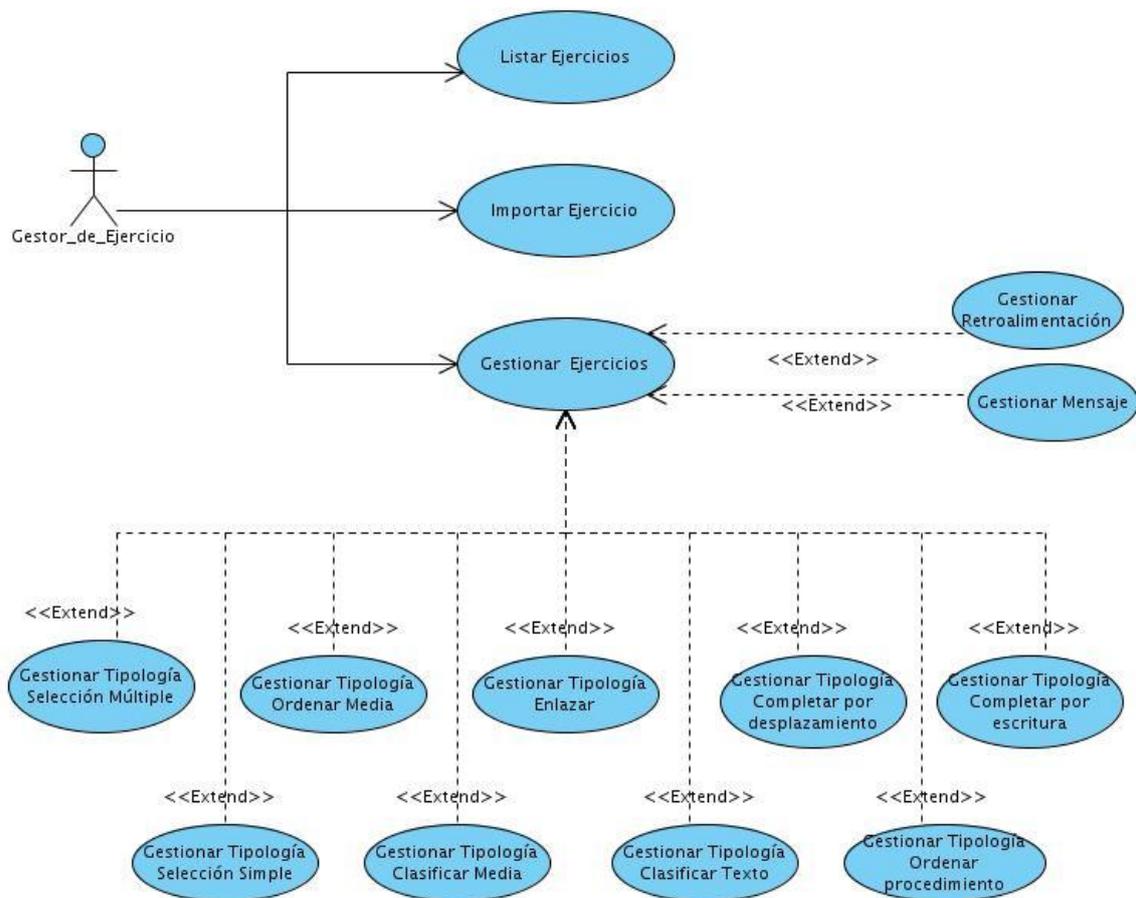


Figura 8: Diagrama de Casos de Usos.

2.4.4. Descripción de los casos de uso del sistema

La descripción de los casos de uso del sistema no son más que las acciones que se realizan durante la interacción entre el actor y el sistema, o sea describe el flujo de actividades que realiza un actor al hacer uso del sistema y las respuestas que emite el mismo.

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

Se muestra la descripción de los casos de usos (DCU) del CU Gestionar Ejercicios. El resto se encuentran en [Anexo 1](#).

CU Gestionar Ejercicio

Especificación del Caso de Uso

Objetivo

Incluir, ver, modificar o eliminar Ejercicio.

Actores

Gestor de Ejercicio (Inicia), generaliza a Administrador central y Editor Digital: Incluye, ve, modifica o elimina un Ejercicio.

Resumen

El caso de uso se inicia cuando el actor selecciona la opción que le permite realizar una acción sobre los Ejercicios. El actor puede incluir, ver, eliminar y modificar un Ejercicio. En caso de que seleccione la opción de incluir Ejercicio, el sistema dará la posibilidad de insertar los datos que se necesitan para llenar este formulario. Si el actor elige la opción de ver Ejercicio el sistema mostrará el contenido del Ejercicio en cuestión. Si el actor elige la opción de modificar Ejercicio, el sistema mostrará los datos que pueden ser editables dentro del Ejercicio, y una vez realizados los cambios, guardará las modificaciones. Si el actor elige la opción de eliminar Ejercicio, el sistema permitirá eliminar el ejercicio mostrando un mensaje de confirmación. El sistema permitirá ver una vista previa del Ejercicio, terminando así el caso de uso.

Precondiciones

Debe haberse generado el escritorio de trabajo del usuario autenticado.

Para incluir, modificar o eliminar debe haber generado el Caso de Uso Listar Ejercicio.

Para ver Ejercicio 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.

Para modificar Ejercicio el actor debe ser el responsable temporal del mismo.

Para eliminar Ejercicio y el actor debe tener el permiso de eliminar el elemento.

Poscondiciones

Se incluyó, vio, modificó o eliminó Ejercicio por el actor.

Referencia

RF 1, RF 1.1, RF 1.2, RF 1.3 y RF 1.4.

Flujo de eventos

Flujo básico

Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona la opción de realizar una acción	

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

sobre un Ejercicio.

2. Brinda la posibilidad de realizar las acciones:
 - Incluir una nueva entidad.
 - Ver los datos de la entidad. Ver Sección 1: "Ver datos del Ejercicio".
 - Modificar los datos de la entidad. Ver Sección 2: "Modificar datos del Ejercicio".
 - Eliminar la entidad. Ver Sección 3: "Eliminar Ejercicio".
-

3. Selecciona la opción de incluir una nueva entidad.

4. Brinda la posibilidad de introducir o seleccionar los datos generales del ejercicio:
 - Título.
 - Descripción.
 - Enunciado.
 - Complejidad
 - Muy Fácil.
 - Fácil.
 - Media.
 - Difícil.
 - Muy difícil.
 - Tipología.
 - Selección simple.
 - Selección múltiple.
 - Completar por escritura.
 - Completar por desplazamiento.
 - Ordenar procedimiento.
 - Ordenar media.
 - Enlazar.
 - Clasificar texto.
 - Clasificar media.
-

5. Introduce y selecciona los datos del ejercicio.

6. Selecciona la tipología selección simple

7. Brinda la posibilidad de introducir o seleccionar los datos de la tipología ver CU Gestionar tipología Selección Simple.
-

8. Permite adicionar opcionalmente:
 - Otros Datos.
-

9. Además, permite seleccionar la opción:
 - Guardar
 - Guardar sin publicar
 - Guardar y continuar
 - Cancelar.
-

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

10. Selecciona la opción de Guardar.

11. Valida los datos.

12. Crea el Ejercicio

13. Muestra un mensaje de información

14. El caso de uso termina.

Flujo alterno

*.a El actor selecciona la opción de Cancelar.

Acciones del actor

Respuesta del sistema

*. a.1 Elimina los datos creados.

*.a.2 Muestra un mensaje de información.

*.a.3 Regresa a la vista anterior.

*.b.4 El caso de uso termina.

6.a. El actor selecciona la tipología de Selección múltiple.

Acciones del actor

Respuesta del sistema

6.a.1. Brinda la posibilidad de introducir o seleccionar los datos de la tipología ver CU Gestionar tipología selección múltiple.

6.a.2. Regresa al paso 8 del flujo básico.

6.b. El actor selecciona la tipología Completar por escritura.

Acciones del actor

Respuesta del sistema

6.b.1. Brinda la posibilidad de introducir o seleccionar los datos de la tipología ver CU Gestionar tipología Completar por escritura

6.b.2. Regresa al paso 8 del flujo básico.

6.c.El actor selecciona la Tipología completar por desplazamiento.

Acciones del actor

Respuesta del sistema

6.c.1. Brinda la posibilidad de introducir o seleccionar los datos de la tipología ver CU Gestionar tipología Completar por desplazamiento

6.c.2. Regresa al paso 8 del flujo básico

6.d. El actor selecciona la Tipología Ordenar Media.

Acciones del actor

Respuesta del sistema

6.d.1. Brinda la posibilidad de introducir o seleccionar los datos de la tipología ver CU Gestionar tipología Ordenar Media.

6.d.2. Regresa al paso 8 del flujo básico.

6.e. El actor selecciona la Tipología Ordenar Procedimiento.

Acciones del actor

Respuesta del sistema

6.e.1. Brinda la posibilidad de introducir o seleccionar los datos de la tipología ver CU Gestionar tipología Ordenar Procedimiento.

6.e.2. Regresa al paso 8 del flujo básico.

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

6.f. El actor selecciona la Tipología Enlazar	
Acciones del actor	Respuesta del sistema
	6.f.1. Brinda la posibilidad de introducir o seleccionar los datos de la tipología ver <u>CU Gestionar tipología Enlazar elementos.</u>
	6.f.2. Regresa al paso 8 del flujo básico.
6.g. El actor selecciona la Tipología Clasificar texto.	
Acciones del actor	Respuesta del sistema
	6.g.1. Brinda la posibilidad de introducir o seleccionar los datos de la tipología ver <u>CU Gestionar tipología Clasificar texto.</u>
	6.g.2. Regresa al paso 8 del flujo básico.
6.h. El actor selecciona la Tipología Clasificar media.	
Acciones del actor	Respuesta del sistema
	6.h.1. Brinda la posibilidad de introducir o seleccionar los datos de la tipología ver <u>CU Gestionar tipología Clasificar media.</u>
	6.h.2. Regresa al paso 8 del flujo básico.
9.a El actor selecciona la opción de adicionar otros datos	
Acciones del actor	Respuesta del sistema
	9.a.1. Brinda la posibilidad de adicionar opcionalmente los datos siguientes: <ul style="list-style-type: none"> • Imagen de apoyo • Autor • Lenguaje • Dependiente de tiempo • Tiempo • Intento • Palabras claves Y permite adicionar los Mensajes ver <u>CU Gestionar Mensaje</u>
9.a.2. Introduce y/o selecciona los datos deseados	9.a.3. Regresa al paso 9 del Flujo Básico.
10.a El actor selecciona la opción de guardar sin publicar	
Acciones del actor	Respuesta del sistema
	1. Valida los datos.
	2. Crea el ejercicio
	3. Muestra el listado de Ejercicio
	4. Muestra un mensaje de información.
	5. El caso de uso termina.
10.b El actor selecciona la opción de guardar y continuar	

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

Acciones del actor	Respuesta del sistema
	10.b.1. Valida los datos.
	10.b.2. Crea el ejercicio
	10.b.3. Muestra un mensaje de información.
	10.b.4. Regresa al paso 4 del flujo básico.
11. a Existen datos incompletos.	
Acciones del actor	Respuesta del sistema
	11. a.1 Sobre cada campo incompleto muestra el mensaje de error.
	11. a.2 Regresa al paso 7 del Flujo Básico.
11. b Existen datos incorrectos.	
Acciones del actor	Respuesta del sistema
	11. b.1 Sobre cada campo incorrecto muestra el mensaje de error.
	11. b.2 Regresa al paso 7 del Flujo Básico.

Sección 1: “Ver datos del Ejercicio”

Flujo básico

Acciones del actor	Respuesta del sistema
1. Selecciona la opción de ver los datos de la Entidad	
	2. Muestra los datos del Ejercicio <ul style="list-style-type: none">• Título.• Descripción• Enunciado.• Tipología.• Complejidad. De acuerdo a los permisos del usuario permite: <ul style="list-style-type: none">• Modificar.• Eliminar.• Cancelar• Exportar.• Publicar.
3. Selecciona la opción de cancelar.	
	4. Regresa a la vista anterior.
	5. El caso de uso termina.

Flujo alterno

3.a El actor selecciona la opción de Modificar los datos de la entidad.	
Acciones del actor	Respuesta del sistema
	3.a.1 Brinda la posibilidad de modificar los datos de la entidad. Ver Sección 3: “ <i>Modificar datos del Ejercicio.</i> ”

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

3. a.2 El caso de uso termina.	
3. b El actor selecciona la opción de eliminar la entidad.	
Acciones del actor	Respuesta del sistema
	3. b.1. Brinda la posibilidad de eliminar la entidad. Ver Sección 4: " <u>Eliminar Ejercicio</u> ".
	3. b.2. El caso de uso termina.
3. c El actor selecciona la opción de Exportar	
Acciones del actor	Respuesta del sistema
	3. c.1. Brinda la posibilidad de exportar el ejercicio. Ver <u>CU Listar Ejercicio</u> . Sección 1: " <u>Exportar Ejercicio</u> ".
	3. c.2. El caso de uso termina.
3. d El actor selecciona la opción de Publicar	
Acciones del actor	Respuesta del sistema
	3. d.1. Publica el ejercicio
	3. d.2. El caso de uso termina.

Sección 2: "Modificar datos del Ejercicio"

Flujo básico

Acciones del actor	Respuesta del sistema
1. Selecciona la opción de modificar los datos del Ejercicio.	
	2. Muestra los datos de Ejercicio y brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos o seleccionando diferentes. Permite además: <ul style="list-style-type: none">• Actualizar los datos.• Cancelar la operación en cualquier momento.
3. Modifica los datos que necesite y selecciona la opción de Actualizar los mismos.	
	4. Valida los datos.
	5. <i>Actualiza los datos del Ejercicio.</i>
	6. Muestra un mensaje de información.
	7. El caso de uso termina.

Flujo alterno

*.a El actor selecciona la opción de Cancelar	
Acciones del actor	Respuesta del sistema
	*.a.1 Elimina los datos creados.
	*.a.2 Muestra un mensaje de información.

CAPÍTULO II. CARACTERÍSTICAS DEL SISTEMA

	*.a.3 Regresa a la vista anterior.
	*.a.4 El caso de uso termina.
4. a Existen datos incompletos.	
Acciones del actor	Respuesta del sistema
	4. a.1 Sobre cada campo incompleto muestra el mensaje de información.
	4. a.2 Regresa al paso 2 del Flujo Básico.
4. b Existen datos incorrectos.	
Acciones del actor	Respuesta del sistema
	4. b.1 Sobre cada campo incorrecto muestra el mensaje de información.
	4. b.2 Regresa al paso 2 del Flujo Básico.

Sección 3: "Eliminar Ejercicio"

Flujo básico

Acciones del actor	Respuesta del sistema
1. Selecciona la opción de eliminar el Ejercicio.	
	2. Muestra el mensaje de advertencia y permite: <ul style="list-style-type: none"> • Aceptar. • Cancelar la operación.
3. Selecciona la opción de Aceptar	
	4. <i>Ocultar el Ejercicio.</i>
	5. Muestra un mensaje de información.
	6. Regresa a la vista anterior.
	7. El caso de uso termina.

Flujo alternativo

*.a El actor selecciona la opción de Cancelar	
Acciones del actor	Respuesta del sistema
	*.a.1 Regresa a la vista anterior.
	*.a.2 El caso de uso termina.

2.5. Conclusiones parciales

El análisis de las características del sistema propicia una documentación que mantiene al equipo de desarrollo centrado en los requerimientos funcionales y no funcionales de la aplicación a través de los casos de uso arquitectónicamente significativos, después de obtener los eventos y procesos que ocurren en el entorno donde será implantado el mismo, logrando un lenguaje común entre todos los involucrados mediante la descripción de la propuesta.

Análisis y diseño del sistema

3.1. Introducción

El análisis de requisitos genera la especificación de las características operacionales de software; indica la interfaz del software con otros elementos del sistema y establece las restricciones que debe tener el software. El análisis se centra en el qué, y no en el cómo, construyendo modelos que representan escenarios del usuario, actividades funcionales así como el comportamiento de las clases y el sistema, siendo estos la principal entrada al diseño, el cual tiene el propósito de formular los modelos que se centran en los requisitos no funcionales y en el dominio de la solución y que prepara para la implementación y prueba del sistema. Creando un plano del modelo de implementación.

3.2. Modelo de análisis

El Modelo de Análisis es una representación conceptual, en él se modelan una serie de clases y relaciones. Se crea con el objetivo de refinar los casos de uso de una forma más detallada, posibilitando un conocimiento razonable del sistema o producto, ayuda a refinar y estructurar los requisitos, permite entender mejor los aspectos internos del sistema, a través de sus diagramas ofrece una mayor formalización.

A continuación una breve descripción de los estereotipos presentes en los diagramas de clases de análisis.

Clase de interfaz: Es utilizada para modificar la interacción entre el sistema y sus actores, es la encargada de recibir y representar informaciones y peticiones de usuarios y de sistemas externos. Representa la abstracción de ventanas, formularios, paneles, interfaz de comunicación.

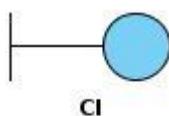


Figura 9: Clase interfaz

Clase control: Se encarga de presentar coordinación secuencial, transacciones y control de otros objetos. Es usada para encapsular el control de un caso de uso en concreto. Además se usa para representar derivaciones y cálculos complejos.



Figura 10: Clase controladora

Clase de entidad: Se usa para modelar información que posee larga vida, y que es a menudo persistente. Modela la información y comportamiento asociado de algún fenómeno o concepto (persona, objeto del mundo real o suceso). Se derivan de una clase de entidad del negocio o del dominio.

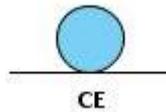


Figura 11: Clase entidad

3.2.1. Diagramas de clases del análisis

Un Diagrama de clases del análisis es un artefacto en el que se representan los conceptos con el contexto del dominio del problema. Representa el funcionamiento del mundo real, no de la implementación automatizada del mismo.

Se muestra una representación del diagrama de clases del análisis (DCA) del CU Gestionar Ejercicios. El resto se encuentran en [Anexos 2.](#)

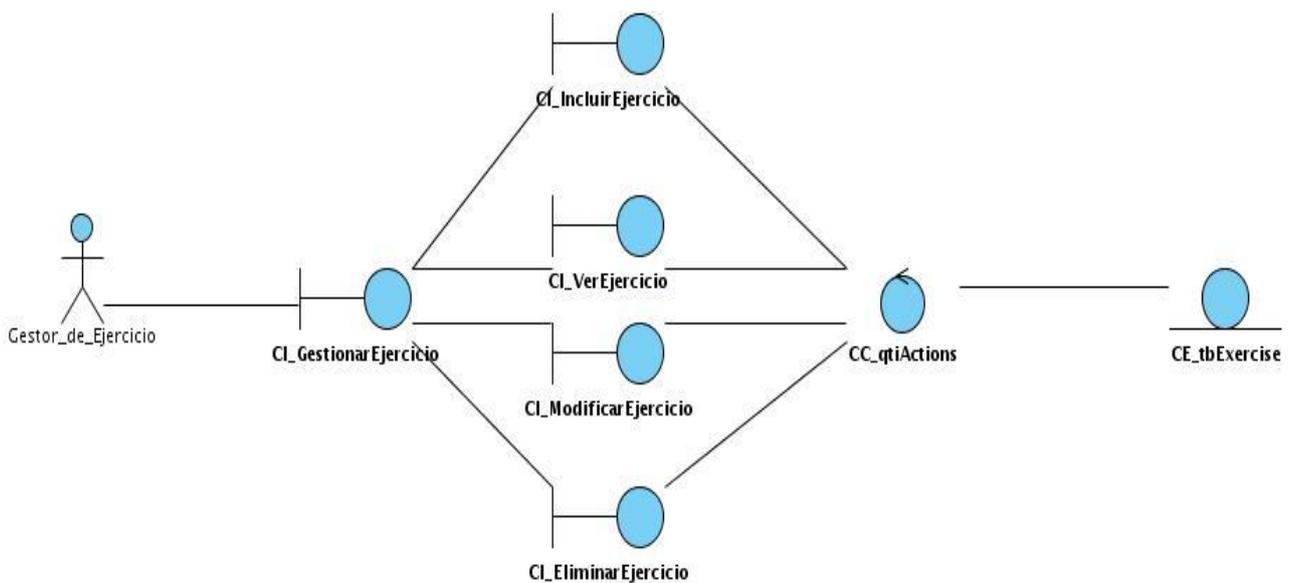


Figura 12: Diagrama de Clases de Análisis CU Gestionar Ejercicio

3.2.2. Diagrama de Colaboración del análisis

Los Diagramas de Colaboración se prestan más al descubrimiento de abstracciones pues permite representar los objetos en una disposición próxima a la realidad. Son útiles en la fase exploratoria para identificar objetos. Permiten representar una disposición espacial de la estructura de los objetos en ejecución. La estructura estática viene dada por los enlaces; la dinámica por el envío de mensajes por los enlaces.

Se muestra una representación del diagrama de colaboración del análisis (DCA) del CU Gestionar Ejercicios. El resto se encuentran en [Anexo 3](#).

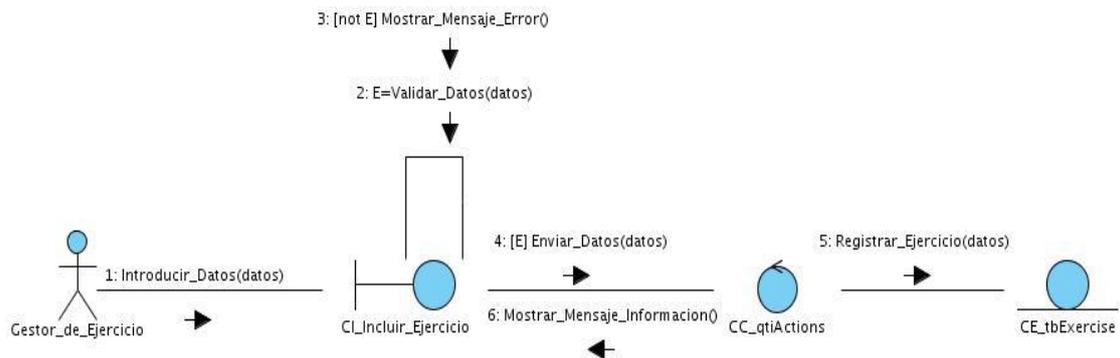


Figura 13: DC CU Gestionar Ejercicio. Sección Incluir.



Figura 14: DC CU Gestionar Ejercicio. Sección Ver.

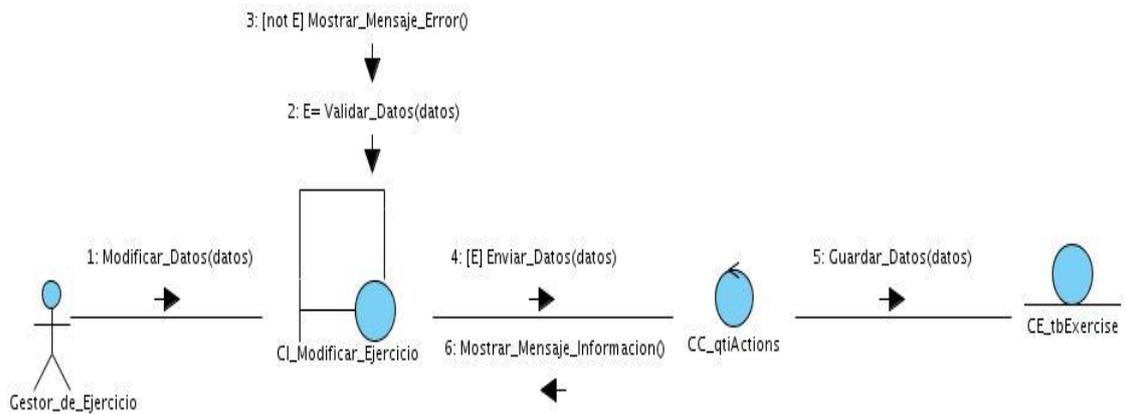


Figura 15: DC CU Gestionar Ejercicio. Sección Modificar.

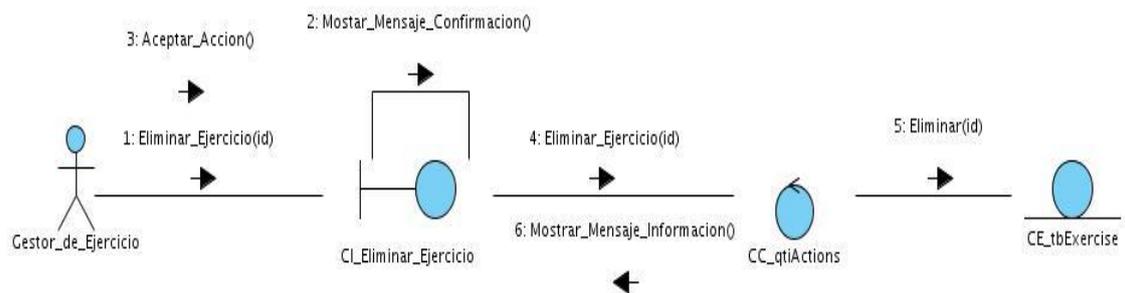


Figura 16: DC CU Gestionar Ejercicio. Sección Eliminar.

3.3. Modelo de diseño

Es un modelo de objetos que representa las colaboraciones que ocurren entre las páginas, donde cada página lógica puede ser representada como una clase. Facilitando una abstracción de la implementación del sistema siendo el artefacto fundamental de entrada a la implementación.

3.3.1. Diagrama de clases del diseño

Es una representación visual del interior del módulo. Modela las páginas, sus enlaces y todo el contenido dinámico de crear las páginas del lado del servidor y el contenido dinámico en el lado del cliente. Además de las capas del MVC y su relación con el componente de symfony.

Se muestra una representación del diagrama de clases del diseño (DCD) del CU Gestionar Ejercicios. El resto se encuentran en [Anexos 4](#).

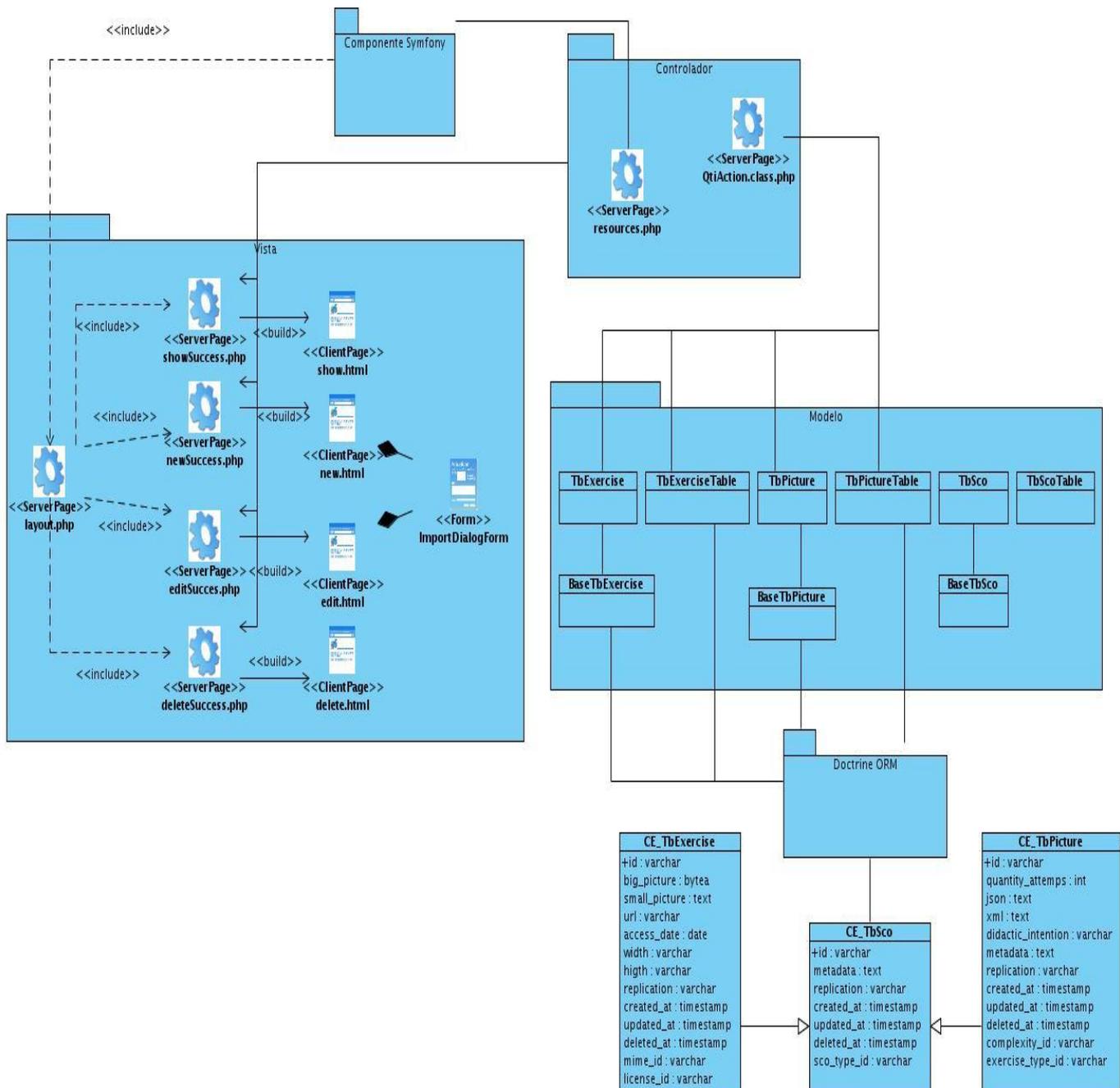


Figura 17: DCD CU Gestionar Ejercicio

3.3.2. Diagrama de secuencia del diseño

Los diagramas de secuencia son útiles en la fase exploratoria para identificar objetos. Muestran gráficamente las interacciones del actor y de las operaciones a que dan origen. Su creación depende de la formulación previa de los casos de uso. Muestra un determinado escenario de un caso de uso, los eventos generados por actores externos, su orden y los eventos internos del sistema.

CAPÍTULO III. ANÁLISIS Y DISEÑO DEL SISTEMA

Se muestra una representación del diagrama de secuencia del diseño (DSD) del CU Gestionar Ejercicios. El resto se encuentran en [Anexo 5](#).

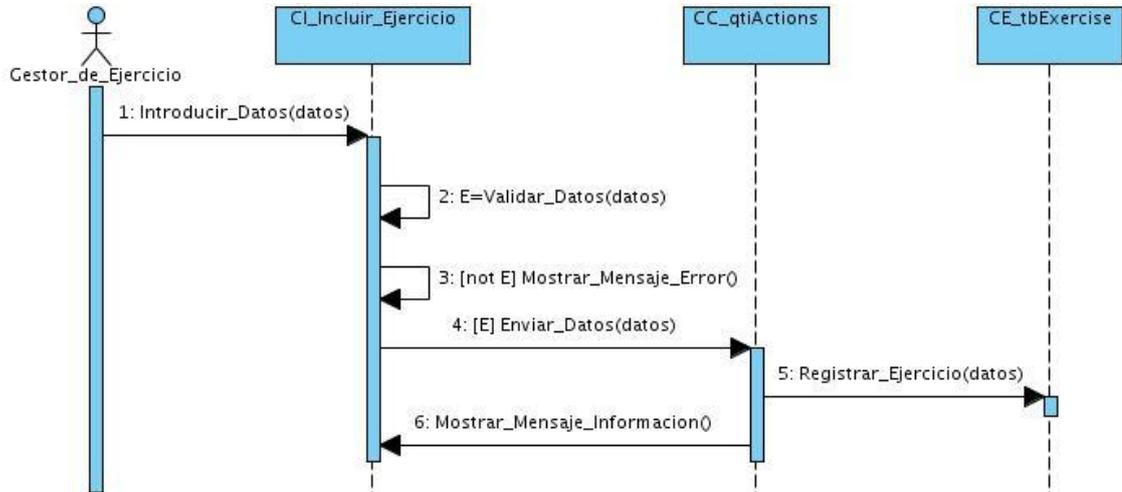


Figura 18: DS CU Gestionar Ejercicio. Sección Incluir.

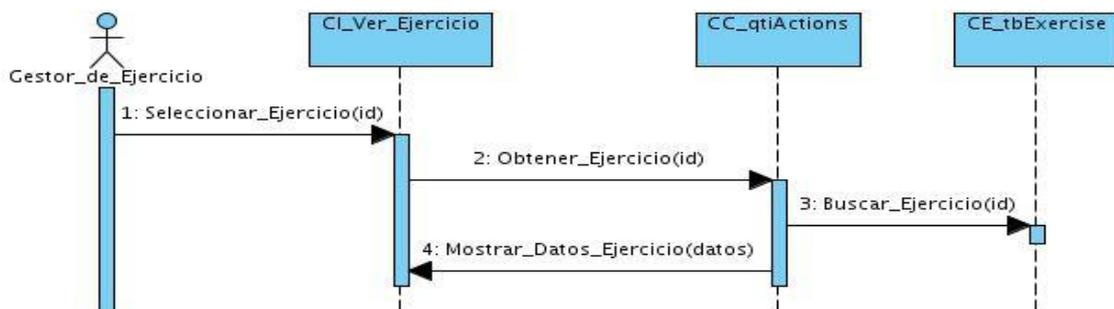


Figura 19: DS CU Gestionar Ejercicio. Sección Ver.

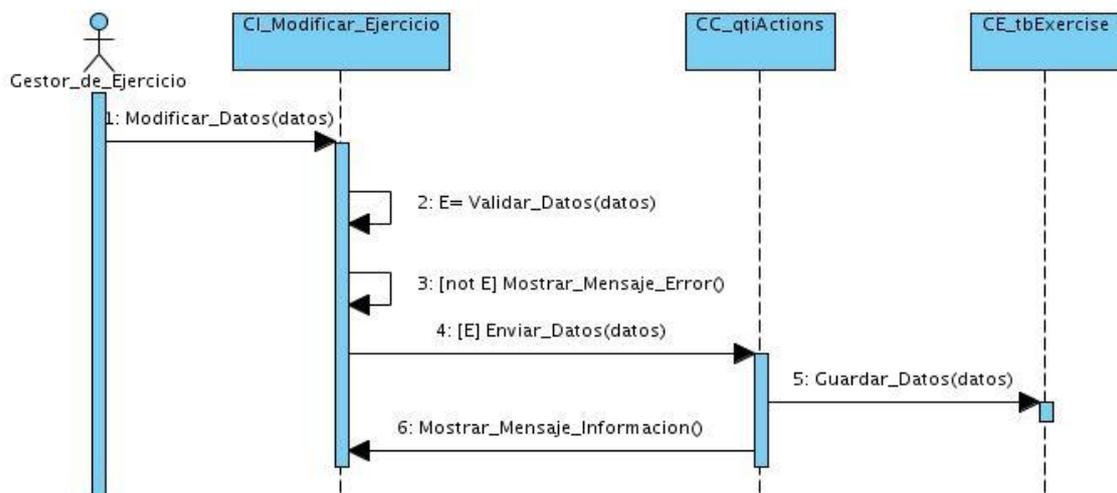


Figura 20: DS CU Gestionar Ejercicio. Sección Modificar.

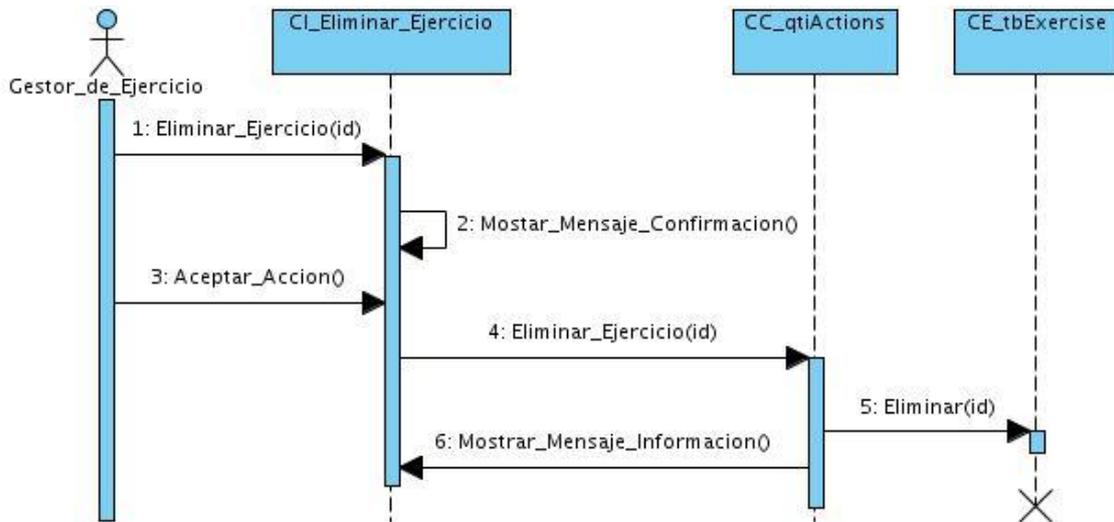


Figura 21: DS CU Gestionar Ejercicio. Sección Eliminar.

3.4. Diseño de la base de datos

Una base de datos es un almacén que permite guardar 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 de forma organizada para que luego podamos encontrar y utilizar fácilmente.

CAPÍTULO III. ANÁLISIS Y DISEÑO DEL SISTEMA

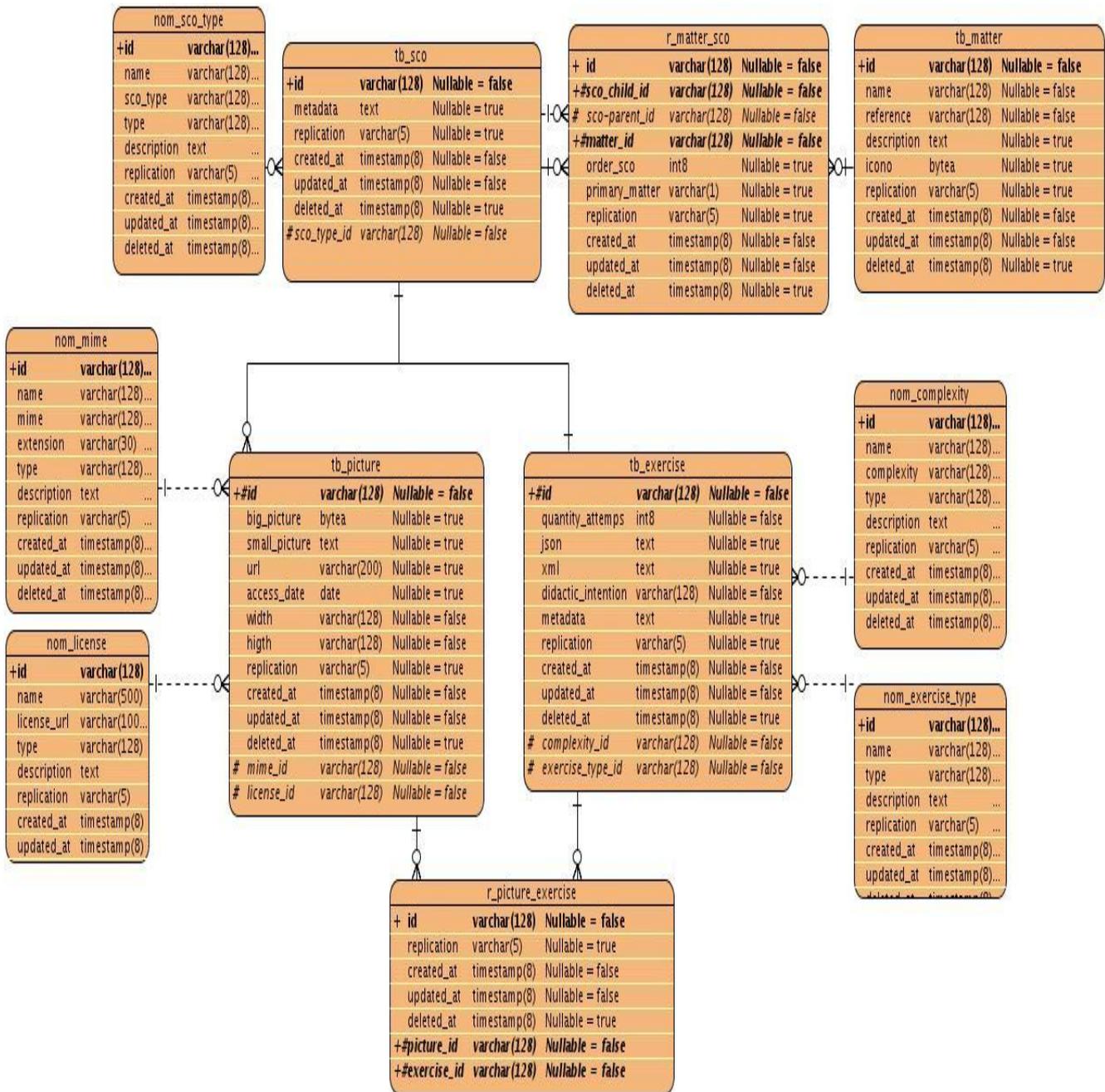


Figura 22: Diseño de Base Datos.

Descripción de las tablas

Se muestra la descripción de la tabla tb_exercise. El resto se encuentran en [Anexo 6](#).

tb_exercise		
Descripción: En la siguiente tabla se agrupa la información existente de un ejercicio.		
Atributo	Tipo	Descripción

CAPÍTULO III. ANÁLISIS Y DISEÑO DEL SISTEMA

id	VARCHAR(128)	Etiqueta única que identifica al objeto (tb_exercise) en la tabla.
exercise_type_id	VARCHAR(128)	Etiqueta extranjera que identifica al objeto (nom_tipologia_ejercicio) en la tabla.
quantity_attemps	bigint	Almacena la cantidad de intentos que tendrá el ejercicio
complexity_id	VARCHAR(128)	Etiqueta extranjera que identifica al objeto (nom_complejidad) en la tabla.
json	TEXT	Representación en JSON del ejercicio.
xml	TEXT	Representación en XML del ejercicio acorde a IMS-QTI 2.0
didactic_intention	VARCHAR(128)	Almacena la intención didáctica del ejercicio
metadata	TEXT	Representación en XML de los metadatos de la especificación QTI para el ejercicio
replication	VARCHAR(5)	Almacena las opciones de replicación del ejercicio en la plataforma.
created_at	Timestamp	Almacena la fecha de creación del ejercicio.
updated_at	Timestamp	Almacena la última fecha de actualización del ejercicio.
deleted_at	Timestamp	Almacena la fecha de eliminación del ejercicio.

Tabla 3: Descripción de la tabla tb_exercise

Se muestra la descripción de la tabla tb_exercise. El resto se encuentran en [Anexo 6](#).

3.5. Diagrama de despliegue

El diagrama de despliegue consiente en mostrar la arquitectura en tiempo de ejecución del sistema respecto al hardware y al software. Los nodos representan los objetos físicos existentes en tiempo de ejecución, sirven para modelar recursos que tienen memoria y capacidad de proceso, y pueden ser computadores, dispositivos o personas. Los componentes participan en los procesos mediante los nodos.

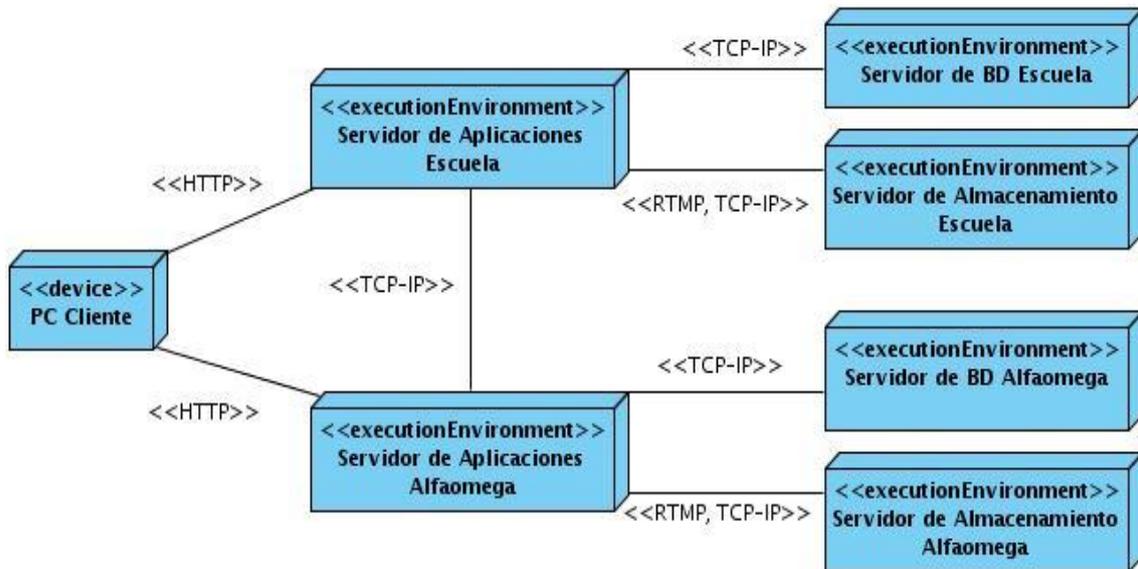


Figura 23: Diagrama de despliegue.

3.6. Conclusiones parciales

El modelado de los diagramas de clases del análisis ayuda a tener un mayor entendimiento y refinamiento del sistema, el cual da paso a la realización del diagrama de diseño que es el artefacto principal de salida de la fase del análisis y diseño. Al iniciar la fase de implementación éste le muestra al programador una visión más clara de lo que se desea realizar en el módulo y estar más enfocado al cumplimiento de los requerimientos funcionales del sistema. El diseño de la base de datos permitirá que la información persistente de larga o corta vida tenga un almacén donde se pueda mantener intacta.

Implementación y pruebas

4.1. Introducción

El proceso de implementación será guiado por el análisis y diseño realizado al módulo, utilizándose como artefacto principal el diagrama de clases del diseño para darle al desarrollador una visión clara del sistema. Durante este proceso el programador realiza todos los requerimientos funcionales teniendo en cuenta en su desarrollo los no funcionales también para así llegar al producto final. Una vez terminado software se procede a realizarle diferentes pruebas las cuales arrojarán su nivel de calidad permitiendo hacerle mejoras al mismo.

4.2. Modelo de implementación

La implementación es la actividad fundamental durante la fase de construcción, partiendo principalmente del resultado obtenido durante el flujo de trabajo de análisis y diseño. Durante esta etapa se desarrolla la arquitectura, se logra estructurar el sistema final, se organiza el código, se implementan los elementos del diseño, y se integran todos estos elementos para obtener un resultado satisfactorio que responda a lo definido durante las fases anteriores. Al llevar a cabo el modelo de implementación se establece la estructura de los elementos de implementación, basándose en las responsabilidades asignadas a los subsistemas de implementación y su contenido.

4.3. Diagrama de componentes

El diagrama de componentes describe la descomposición física del sistema en componentes, a efectos de construcción y funcionamiento. La descomposición del diagrama de componentes se realiza en términos de componentes y de relaciones entre los mismos. Los componentes identifican objetos físicos que hay en tiempos de ejecución, de compilación o de desarrollo, y tienen identidad propia y una interfaz bien definida. Cada componente incorpora la implementación de ciertas clases del diseño del sistema.

CAPÍTULO IV. IMPLEMENTACIÓN Y PRUEBAS

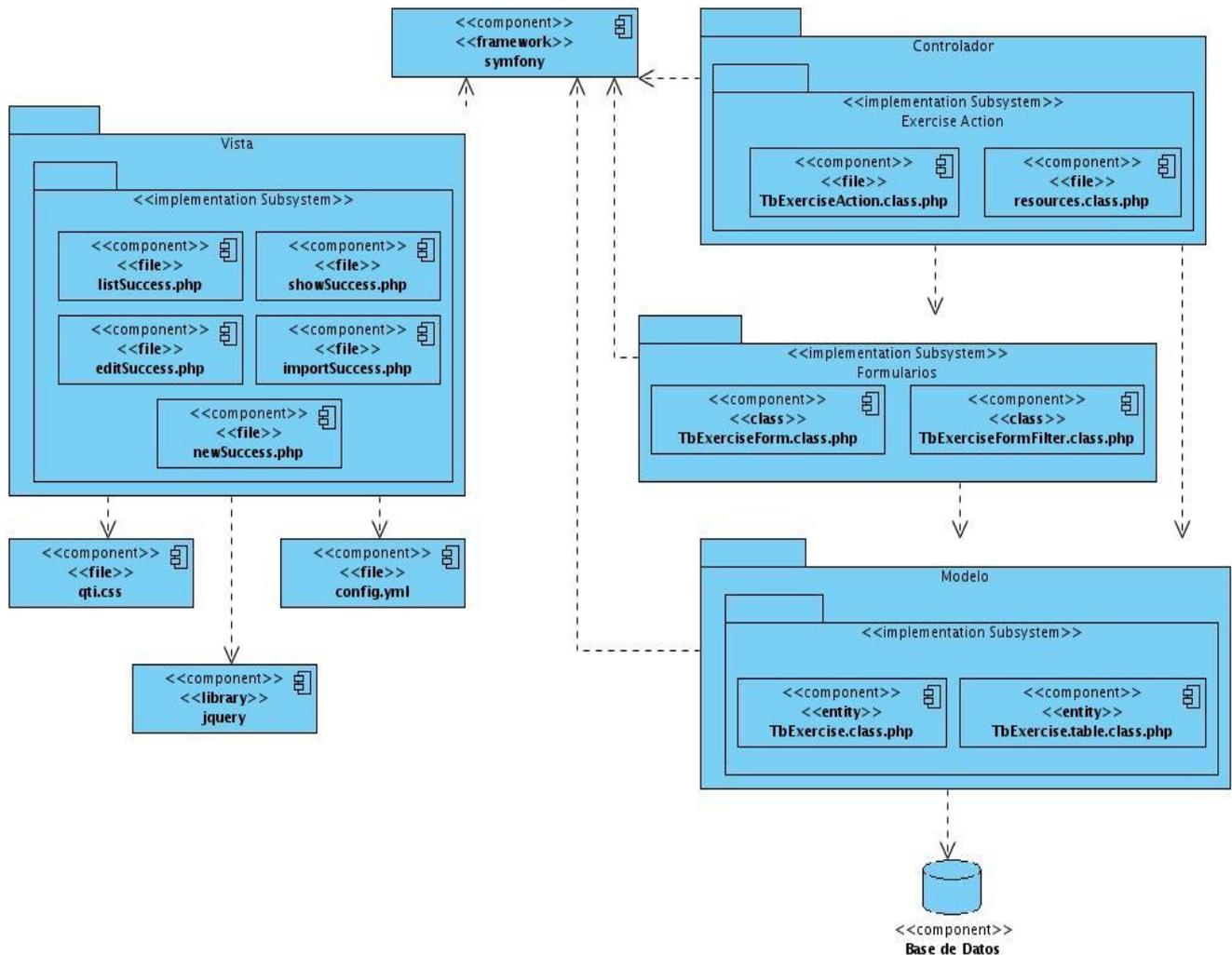


Figura 24: Diagrama de componentes

4.4. Pruebas de software

El proceso de pruebas es el instrumento apropiado para determinar el nivel de la calidad de un producto software. Durante este proceso se realizan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos. Existen diferentes niveles de pruebas aplicada para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo entre las usadas para probar el módulo desarrollado están:

Prueba de Desarrollador: Es la prueba diseñada e implementada por el equipo de desarrollo. Tradicionalmente estas pruebas han sido consideradas solo para la prueba de unidad, aunque en la actualidad en algunos casos pueden ejecutar pruebas de integración.

Prueba independiente: Es la prueba que es diseñada e implementada por alguien independiente del grupo de desarrolladores. El objetivo de estas pruebas es proporcionar una perspectiva diferente y en un ambiente más rico que los desarrolladores. Son pruebas basadas en las necesidades y preocupaciones de los stakeholders⁹.

Pruebas funcionales: Como su propio nombre indican, prueban una funcionalidad completa, donde pueden estar implicadas una o varias clases, la propia interfaz de usuario.

Pruebas de aceptación: Son pruebas funcionales, pero vistas directamente desde el cliente. Digamos que son aquellas pruebas que demuestran al cliente que la funcionalidad está terminada y funciona correctamente.

El tipo de prueba usado para probar el software fue el de funcionalidad del cual se realizaron pruebas de:

Función: Pruebas fijando su atención en la validación de las funciones, métodos, servicios, caso de uso.

Seguridad: Asegurar que los datos o el sistema solamente es accedido por los actores deseados.

Pruebas de caja negra

El método de la caja negra se centra en los requisitos fundamentales del software y permite obtener entradas que prueben todos los requisitos funcionales del sistema. Con este tipo de pruebas se intenta encontrar:

- Funciones incorrecta o ausente.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

4.5. Diseño de casos de prueba

Los Casos de Prueba (CP) se realizan para determinar que un caso de uso está implementado teniendo en cuenta cada uno de sus requerimientos. Debe haber un CP

⁹Cualquier grupo o individuo que pueda afectar o ser afectado por el logro de los propósitos de una corporación. (33)

CAPÍTULO IV. IMPLEMENTACIÓN Y PRUEBAS

para cada caso de uso. En este caso, el sistema desarrollado constaba con 14 caso de uso por lo tanto se realizó el diseño de 14 CP.

A continuación se muestra una representación del diseño de Gestionar Ejercicio. Los restantes están en el [Anexo 7](#).

SC Incluir datos de los Ejercicios

ID del escenario	Escenario	Título	Descripción	Enunciado	Imagen de Apoyo	Autor	Dependiente del	Palabras claves	Respuesta del sistema	Flujo central
EC 1	Selecciona la opción de incluir una nueva entidad.								<p>Brinda la posibilidad de introducir o seleccionar los datos generales del ejercicio:</p> <ul style="list-style-type: none"> • Título. • Descripción. • Enunciado. • Complejidad <ul style="list-style-type: none"> • Muy Fácil. • Fácil. • Media. • Difícil. • Muy difícil. • Tipología. <ul style="list-style-type: none"> • Selección simple. • Selección múltiple. • Completar por escritura. • Completar por desplazamiento. • Ordenar procedimiento. • Ordenar media. • Enlazar. • Clasificar texto. • Clasificar media. 	Recursos Interactivos/Cuestionarios/Listar todos/Clic derecho sobre un Índice/Incluir Ejercicio
EC 2	Introduce y selecciona los datos del ejercicio. Selecciona la tipología selección simple								<p>Brinda la posibilidad de introducir o seleccionar los datos de la tipología ver <i>CP Gestionar tipología Selección Simple</i>. Permite adicionar opcionalmente:</p> <ul style="list-style-type: none"> • Otros Datos. <p>Además, permite seleccionar la opción:</p> <ul style="list-style-type: none"> • Guardar • Guardar sin publicar 	Recursos Interactivos/Cuestionarios/Listar todos/Clic derecho sobre un Índice/Incluir Ejercicio

CAPÍTULO IV. IMPLEMENTACIÓN Y PRUEBAS

									<ul style="list-style-type: none"> • Guardar y continuar • Cancelar. 	
EC 3	Selecciona la opción de Guardar.	V	V	V	N/A	N/A	N/A	N/A	Valida los datos. Crea el Ejercicio. Muestra un mensaje de información. El caso de uso termina.	Recursos Interactivos/Cuestionarios/Listar todos/Clic derecho sobre un Índice/Incluir Ejercicio/ Botón Guardar
EC 4	Selecciona la opción de Cancelar.								Elimina los datos creados. Muestra un mensaje de información. Regresa a la vista anterior. El caso de uso termina.	Recursos Interactivos/Cuestionarios/Listar todos/Clic derecho sobre un Índice/Incluir Ejercicio/ Botón Cancelar
EC 5	Selecciona la tipología de Selección múltiple.								Brinda la posibilidad de introducir o seleccionar los datos de la tipología ver <u>CP Gestionar tipología selección múltiple</u> . Regresa al EC 2	Recursos Interactivos/Cuestionarios/Listar todos/Clic derecho sobre un Índice/Incluir Ejercicio
EC 6	Selecciona la tipología de Completar por escritura.								Brinda la posibilidad de introducir o seleccionar los datos de la tipología ver <u>CP Gestionar tipología Completar por escritura</u> . Regresa al EC 2	Recursos Interactivos/Cuestionarios/Listar todos/Clic derecho sobre un Índice/Incluir Ejercicio
EC 7	Selecciona la Tipología completar por desplazamiento.								Brinda la posibilidad de introducir o seleccionar los datos de la tipología ver <u>CP Gestionar tipología Completar por desplazamiento</u> . Regresa al EC 2	Recursos Interactivos/Cuestionarios/Listar todos/Clic derecho sobre un Índice/Incluir Ejercicio

CAPÍTULO IV. IMPLEMENTACIÓN Y PRUEBAS

EC 8	Selecciona la Tipología Ordenar Media.							Brinda la posibilidad de introducir o seleccionar los datos de la tipología ver <u>CP Gestionar tipología Ordenar Media</u> . Regresa al EC 2	Recursos Interactivos/Cuestionarios/Listar todos/Clic derecho sobre un Índice/Incluir Ejercicio
EC 9	Selecciona la Tipología Ordenar Procedimiento.							Brinda la posibilidad de introducir o seleccionar los datos de la tipología ver <u>CP Gestionar tipología Ordenar Procedimiento</u> . Regresa al EC 2	Recursos Interactivos/Cuestionarios/Listar todos/Clic derecho sobre un Índice/Incluir Ejercicio
EC 10	Selecciona la Tipología Enlazar							Brinda la posibilidad de introducir o seleccionar los datos de la tipología ver <u>CP Gestionar tipología Enlazar elementos</u> . Regresa al EC 2	Recursos Interactivos/Cuestionarios/Listar todos/Clic derecho sobre un Índice/Incluir Ejercicio
EC 11	Selecciona la Tipología Clasificar texto.							Brinda la posibilidad de introducir o seleccionar los datos de la tipología ver <u>CP Gestionar tipología Clasificar texto</u> . Regresa al EC 2	Recursos Interactivos/Cuestionarios/Listar todos/Clic derecho sobre un Índice/Incluir Ejercicio
EC 12	Selecciona la Tipología Clasificar media.							Brinda la posibilidad de introducir o seleccionar los datos de la tipología ver <u>CP Gestionar tipología Clasificar media</u> . Regresa al EC 2	Recursos Interactivos/Cuestionarios/Listar todos/Clic derecho sobre un Índice/Incluir Ejercicio
EC 13	Selecciona la opción de Adicionar más datos							Brinda la posibilidad de adicionar opcionalmente los datos siguientes: <ul style="list-style-type: none"> • Imagen de apoyo • Autor • Lenguaje • Dependiente de tiempo • Tiempo 	Recursos Interactivos/Cuestionarios/Listar todos/Clic derecho sobre un Índice/Incluir

CAPÍTULO IV. IMPLEMENTACIÓN Y PRUEBAS

									<ul style="list-style-type: none"> • Intento • Palabra claves. Y permite adicionar los Mensajes ver <u>CP Gestionar Mensaje</u>	Ejercicio
EC 14	Se introduce y/o seleccionan los datos deseados.								Regresa al EC 2	Recursos Interactivos/Cuestionarios/Listar todos/Clic derecho sobre un Índice/Incluir Ejercicio
EC 15	Selecciona la opción de guardar sin publicar								Valida los datos. Crea el ejercicio Muestra el listado de Ejercicio Muestra un mensaje de información. El caso de uso termina.	Recursos Interactivos/Cuestionarios/Listar todos/Clic derecho sobre un Índice/Incluir Ejercicio/ Botón Guardar sin publicar
EC16	Selecciona la opción de guardar y continuar								Valida los datos. Crea el ejercicio. Muestra un mensaje de información. Regresa al EC 1	Recursos Interactivos/Cuestionarios/Listar todos/Clic derecho sobre un Índice/Incluir Ejercicio/ Botón Guardar y continuar
EC 17 EC 18	Existen Datos incompletos.	/	V	V	N/A	N/A	N/A	N/A	Sobre cada campo incompleto muestra el mensaje de error. Regresa al EC1	Recursos Interactivos/Cuestionarios/Listar todos/Clic derecho sobre un Índice/Incluir Ejercicio
		V	/	V	N/A	N/A	N/A	N/A		
		V	V	/	N/A	N/A	N/A	N/A		
EC	Existen datos	/	V	V	N/A	N/A	N/A	N/A	Sobre cada campo incompleto	Recursos

CAPÍTULO IV. IMPLEMENTACIÓN Y PRUEBAS

19	incorrectos.	V	I	V	N/A	N/A	N/A	N/A	muestra el mensaje de error. Regresa al EC 1	Interactivos/Cuestionarios/Listar todos/Clic derecho sobre un Índice/Incluir Ejercicio
		V	V	I	N/A	N/A	N/A	N/A		
		V	V	V	I	N/A	N/A	N/A		
		V	V	V	N/A	I	N/A	N/A		
		V	V	V	N/A	N/A	I	N/A		
		V	V	V	N/A	N/A	N/A	I		

SC Ver datos del Ejercicio

ID del escenario	Escenario	Respuesta del sistema	Flujo central
EC 1	Selecciona la opción de ver los datos de la entidad.	<p>Muestra los datos del Ejercicio</p> <ul style="list-style-type: none"> • Título. • Descripción • Enunciado. • Tipología. • Complejidad. <p>De acuerdo a los permisos del usuario permite:</p> <ul style="list-style-type: none"> • Modificar. • Eliminar. • Cancelar • Exportar. • Publicar. 	Recursos Interactivos/Cuestionarios/Listar todos/Clic izquierdo sobre el título de un ejercicio.
EC 2	Selecciona la opción de cancelar.	Regresa a la vista anterior. El caso de uso termina.	Recursos Interactivos/Cuestionarios/Listar todos/Clic izquierdo sobre el título de un ejercicio/Botón cancelar
EC 3	Selecciona la opción de Modificar los datos de la entidad.	Brinda la posibilidad de modificar los datos de la entidad. <u>Ver Sección "Modificar datos del Ejercicio"</u> .	Recursos Interactivos/Cuestionarios/Listar todos/Clic izquierdo sobre el título de un

CAPÍTULO IV. IMPLEMENTACIÓN Y PRUEBAS

			ejercicio/ Botón Modificar
EC 4	Selecciona la opción de Eliminar la entidad.	Brinda la posibilidad de eliminar la entidad. <u>Ver Sección “Eliminar Ejercicio”.</u>	Recursos Interactivos/Cuestionarios/Listar todos/Clic izquierdo sobre el título de un ejercicio/ Botón Eliminar
EC 5	Selecciona la opción de Exportar	Brinda la posibilidad de exportar el ejercicio. <u>Ver CP del CU Listar Ejercicio.</u> El caso de uso termina.	Recursos Interactivos/Cuestionarios/Listar todos/Clic izquierdo sobre el título de un ejercicio/ Botón Exportar
EC 6	Selecciona la opción de Publicar	Pública el ejercicio. El caso de uso termina.	Recursos Interactivos/Cuestionarios/Listar todos/Clic izquierdo sobre el título de un ejercicio/ Botón Publicar

SC Modificar datos del Ejercicio

ID del escenario	Escenario	Título	Descripción	Enunciado	Imagen de Apoyo	Autor	Dependiente del tiempo	Palabras claves	Respuesta del sistema	Flujo central
EC 1	Selecciona a la opción de modificar los datos del Ejercicio.								Muestra los datos de Ejercicio y brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos o seleccionando diferentes. Permite además: <ul style="list-style-type: none"> • Actualizar los datos. • Cancelar la operación en cualquier momento. 	Recursos Interactivos/Cuestionarios/Listar todos/ Seleccionar título del Ejercicio/ Modificar
EC 2	Modifica los datos que necesite y selecciona a la	V	V	V	N/A	N/A	N/A	N/A	Valida los datos. Actualiza los datos del Ejercicio. Muestra un mensaje de información. El caso de uso termina	Recursos Interactivos/Cuestionarios/Listar todos/ Seleccionar título del Ejercicio/ Modificar /

CAPÍTULO IV. IMPLEMENTACIÓN Y PRUEBAS

	opción de Actualizar los mismos.										Botón Actualizar
EC 3	Selecciona a la opción de Cancelar									Elimina los datos creados. Muestra un mensaje de información. Regresa a la vista anterior.	Recursos Interactivos/Cuestionarios/Listar todos/ Seleccionar título del Ejercicio/ Modificar/Botón Cancelar
EC 4	Existen datos incompletos.	V	V	V	N/A	N/A	N/A	N/A	Muestra un mensaje de información. Muestra un indicador sobre cada campo incompleto.	Recursos Interactivos/Cuestionarios/Listar todos/ Seleccionar título del Ejercicio/ Botón Modificar	
		I	V	V	N/A	N/A	N/A	N/A			
		V	I	V	N/A	N/A	N/A	N/A			
		V	V	I	N/A	N/A	N/A	N/A			
EC 5	Existen datos incorrectos.	I	V	V	N/A	N/A	N/A	N/A	Muestra un mensaje de información. Muestra un indicador sobre cada campo incorrecto.	Recursos Interactivos/Cuestionarios/Listar todos/ Seleccionar título del Ejercicio/ Botón Modificar	
		V	I	V	N/A	N/A	N/A	N/A			
		V	V	I	N/A	N/A	N/A	N/A			
		V	V	V	I	N/A	N/A	N/A			
		V	V	V	N/A	I	N/A	N/A			
		V	V	V	N/A	N/A	I	N/A			
		V	V	V	N/A	N/A	N/A	I			

SC Eliminar Ejercicio

ID del escenario	Escenario	Respuesta del sistema	Flujo central
EC 1	Selecciona la opción de eliminar el Ejercicio.	Muestra el mensaje de advertencia y permite: <ul style="list-style-type: none"> • Aceptar. • Cancelar la operación. 	Recursos Interactivos/Cuestionarios/Listar todos/ Seleccionar título del Ejercicio/ Botón Eliminar

CAPÍTULO IV. IMPLEMENTACIÓN Y PRUEBAS

EC 2	Selecciona la opción de Aceptar.	Oculto el Ejercicio. Muestra un mensaje de información. Regresa a la vista anterior.	
EC 3	Selecciona la opción de Cancelar	Regresa a la vista anterior.	Recursos Interactivos/Cuestionarios/Listar todos/ Seleccionar título del Ejercicio / Botón Eliminar/ Botón Aceptar
			Recursos Interactivos/Cuestionarios/Listar todos/ Seleccionar título del Ejercicio / Botón Eliminar/ Botón Cancelar

Descripción de las variables.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Título	Campo de texto	No	Título del ejercicio.
2	Descripción	Campo de texto	No	Es la explicación de forma detallada y ordenada, de en qué consiste el tema y todo lo relacionado con el mismo. Es de tipo editable, acepta caracteres alfanuméricos.
3	Enunciado	Campo de texto	No	Enunciado referido al ejercicio es de tipo editable, acepta caracteres alfanuméricos.
4	Tipología	Campo de selección	No	Tipo de ejercicio a desarrollar
5	Intento	Campo de texto	No	Cantidad de intentos del ejercicio, es de tipo seleccionable.
6	Complejidad	Campo de selección	No	Complejidad del ejercicio, de tipo seleccionable
7	Autor	Campo de texto	Si	Autor del ejercicio
8	Palabras claves	Campo de texto	Si	Palabras claves o frase de búsqueda.
9	Dependiente del tiempo	Campo de selección	Si	Si el ejercicio va a depender de un tiempo para su realización o no.
10	Tiempo	Campo de texto	Si	Tiempo que tendrá el estudiante para resolver el ejercicio.

CAPÍTULO IV. IMPLEMENTACIÓN Y PRUEBAS

11	Lenguaje	Campo de selección	Si	El lenguaje que puede tener asociado el recurso. Ejemplo: Inglés, Español.
12	Imagen de apoyo	Campo de selección	Si	Imagen de apoyo relacionada con el ejercicio.

4.6. Resultado Obtenidos

Al módulo desarrollado se le han ejecutado 2 iteraciones de pruebas de caja negra guiadas en los diseños de CP las cuales han arrojado las siguientes No Conformidad (NC) que propiciaron un aumento de su nivel de calidad.

Tipo de No Conformidad	Iteración 1	Iteración 2
Formato (A y D)		
Error Técnico (D)		
Correspondencia con otro artefacto (A y D)	2	1
Ortografía (A y D)	4	1
Redacción (A y D)		
Validación (A)	1	1
Opciones que no funcionan (A)		
Error de Interfaz (A)		
Error Idioma (A)		
Funcionalidad (A)	6	4
Excepciones (A)		
Otros (A y D)		
Seguridad (A)		
Diseño CP (D)	2	1
Recomendación (A y D)		1
Total	15	9

Tabla 4: Resultado de las NC

4.7. Conclusiones parciales

Mediante la construcción del diagrama de componente se incrementó entendimiento de la descomposición física del sistema en componentes, a efectos de construcción y funcionamiento. El proceso de implementación respondió las necesidades de la plataforma lo cual fue comprobado mediante las pruebas de liberación realizadas al sistema.

Conclusiones generales

El presente trabajo ha dado cumplimiento a los objetivos trazados de la investigación científica, obteniéndose:

- El estudio del estado de arte sobre herramientas que gestionan ejercicios arrojó un punto de partida a la solución definiendo características y métodos a utilizar.
- El análisis y diseño de la solución propuesta utilizando la metodología RUP generó documentación detallada para futuros cambios y centró al equipo de desarrollo en los requerimientos de esta.
- Se obtuvo el módulo para la gestión de ejercicios que posibilita el intercambio estandarizado de los mismos con otros LMS.
- El proceso de validación del módulo en la importación y exportación estandarizada de ejercicios incrementó la calidad del software eliminando fallas del mismo, haciéndolo más confiable y seguro.
- El uso de estándares en el desarrollo de la solución, posibilitó la reutilización, interoperabilidad, retroalimentación, búsqueda y localización de los ejercicios entre los distintos LMS.

Recomendaciones

- Crear una arquitectura escalable que permita la adición de nuevas interacciones con un bajo costo de tiempo y pocos cambios.
- Crear un framework que facilite el interoperabilidad de ejercicios utilizando la especificación IMS-QTI en los sistemas para la gestión de aprendizaje.
- Generalizar las experiencias alcanzadas en la estandarización del módulo bajo el estándar IMS-QTI para la gestión de ejercicios y las alternativas utilizadas para su desarrollo.

Bibliografía

1. **Graells, Dr. Pere Marquès.** EL IMPACTO DE LA SOCIEDAD DE LA INFORMACIÓN EN EL MUNDO EDUCATIVO. [En línea] 30 de 04 de 2004. [Citado el: 20 de 06 de 2010.] <http://peremarques.pangea.org/impacto.htm>.
2. **Rodríguez, Kethicer Castellanos.** Monografias.com. [En línea] [Citado el: 2011 de Abril de 29.] <http://www.monografias.com/trabajos31/software-educativo-cuba/software-educativo-cuba.shtml>.
3. **IEEE.** *Comité de Estandarización de Tecnologías Educativas del Estándar para Metadatos de Objetos Educativos.* New York : s.n., 2002.
4. **Pérez, Lourdes Ramos.** ¿Software educativo, hipermedia o entorno educativo? [En línea] 28 de Septiembre de 2008. [Citado el: 27 de 06 de 2010.] http://bvs.sld.cu/revistas/aci/vol18_4_08/aci61008.htm.
5. **Márquez., P.** El software educativo. [En línea] http://www.lmi.ub.es/te/any96/marques_software/#capitol1.
6. **Alonso, Covadonga López.** Entornos Formativos en el ciberespacio: las plataformas educativas. [En línea] 2004. [Citado el: 25 de 06 de 2010.] <http://dialnet.unirioja.es/servlet/articulo?codigo=2353487>. ISSN 1135-867X.
7. **Agudelo, M.** Plataformas educativas. [En línea] 2006. <http://aprendeenlinea.udea.edu.co/banco/html/plataformaseducativas>.
8. *Estandares en los Sistemas de Gestión de Aprendizaje.* **Solis, Elizabeth Castro.** 2007 .
9. *Estudio de la propuesta IMS de estandarización de enseñanza asistida por computadora.* **Manero, I. B.** 2003.
10. **IEEE.** IEEE. [En línea] [Citado el: 3 de Junio de 2011.] <http://www.ieee.org/portal/site>.
11. *Uso y abuso de los objetos de aprendizaje reutilizables.* **Polsani.**
12. **LLOP, ROBERTO ROMERO.** *ESPECIFICACIÓN OWL DE UNA ONTOLOGÍA PARA TELEEDUCACIÓN EN LA WEB SEMÁNTICA.* VALENCIA : s.n., 2007.
13. **Cambridge), Steve Lay (University of.** *IMS Question and Test Interoperability Implementation Guide.* 2005 .
14. Recuperación y Acceso a la Información. [En línea] Universidad Carlos III de Madrid. [Citado el: 5 de Junio de 2011.] <http://www.metadatos-xmlrdf.com/metadatos/LOM>.
15. **Tejada, María del Mar Ramos.** e-revistas. [En línea] 30 de Abril de 2010. [Citado el: 11 de Enero de 2011.]

http://www.erevistas.csic.es/ficha_articulo.php?url=oai_revista67:616&oai_iden=oai_revista67#.

16. **Mercenier, Philippe.** [En línea] [Citado el: 14 de Enero de 2011.] <http://lab.claroline.net/>.

17. **Moodle.** [En línea] [Citado el: 14 de Enero de 2011.] <http://demo.moodle.net/>.

18. **Group.** Longsight Sakai. [En línea] [Citado el: 14 de Enero de 2011.] <http://trysakai.longsight.com/portal/>.

19. **Jacobson, I.** *El proceso unificado de Desarrollo de Software.* 2000.

20. **Cornejo, José Enrique González.** [En línea] [Citado el: 11 de Enero de 2011.] <http://www.docirs.cl/uml.htm>.

21. **(W3C, World Wide Web Consortium.** World Wide Web Consortium. [En línea] 07 de Febreo de 2008. [Citado el: 12 de Enero de 2011.] <http://www.w3c.es/divulgacion/guiasbreves/XHTML>.

22. **Martinez, Ronaldo.** [En línea] 2010. [Citado el: 14 de Enero de 2011.] http://www.programacion.com/articulo/vbscript_54/2.

23. **Eguiluz, Javier.** Symfony.es. [En línea] 2011. [Citado el: 12 de Enero de 2011.] <http://www.symfony.es/10-razones-para-utilizar-symfony/>.

24. **developmen, Web design.** [En línea] 2011. [Citado el: 12 de Enero de 2011.] [.http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/](http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/).

25. **Group, PostgreSQL Global Development.** PostgreSQL: The world's most advanced open source database. [En línea] 2011. [Citado el: 12 de Enero de 2011.] <http://www.postgresql.org/about/>.

26. NginxEs. [En línea] [Citado el: 20 de 05 de 2011.] <http://wiki.nginx.org/NginxEs>.

27. Alexa. [En línea] [Citado el: 20 de 05 de 2011.] <http://www.alexa.com/topsites>.

28. **Netcraft. Internet Research, http://news.netcraft.com/.** Anti-Phishing and PCI Security Services | Netcraft. [En línea] 2011. [Citado el: 14 de Enero de 2011.] <http://news.netcraft.com/>.

29. **Corporation, Oracle.** netbeans.org. [En línea] 2011. [Citado el: 12 de Enero de 2011.] http://netbeans.org/index_es.html.

30. **Alexander, Christopher.** Patrones y Antipatrones: una Introducción. [En línea] [Citado el: 01 de Marzo de 2010.] <http://msdn.microsoft.com/es-es/library/bb972242.aspx#mainSection>.

31. EcuRed. [En línea] [Citado el: 16 de Mayo de 2011.] http://www.ecured.cu/index.php/Aplicaci%C3%B3n_de_patrones_en_Symfony.

32. Instituto de Tecnologías Educativas. [En línea] [Citado el: 20 de 05 de 2011.]
<http://ares.cnice.mec.es/informes/11/contenido/25.htm>.

33. **Freeman, Edward y McVea, John.** *A Stakeholder Approach to Strategic Management*. 2001.