



Universidad de las Ciencias Informáticas
Facultad 4

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

*Análisis y diseño de la versión 2.0 del módulo
Ejercicios de la colección El Navegante
en su versión multiplataforma*

Autora: Mileydis Concepción Silva

Tutores: Ing. Mailyn Cabrera Torres
Ing. Hector Luis Reyes Pupo

La Habana, junio de 2011
“Año 53 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor del trabajo “Análisis y diseño de la versión 2.0 del módulo Ejercicios de la colección El Navegante” y autorizo a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Autor:

Mileydis Concepción Silva

Tutores:

Ing. Mailyn Cabrera Torres

Ing. Hector Luis Reyes Pupo

RESUMEN

La introducción de las Tecnologías de la Información y las Comunicaciones en el sector de la educación en Cuba han provocado transformaciones en los métodos tradicionales de enseñanza, ya que se han desarrollado amplios programas de elaboración de software educativo, entre los que se encuentra la colección El Navegante, dedicada a la enseñanza secundaria. El desarrollo de una nueva versión de esta colección para Venezuela (multiplataforma) se lleva a cabo por la Universidad de las Ciencias Informáticas en conjunto con el Ministerio de Educación. Uno de los módulos con los que cuentan los productos de esta colección es Ejercicios, del cual se obtuvo una primera versión con un alto nivel de desarrollo, sin embargo, no se logró satisfacer completamente los requisitos del cliente. En aras de lograr implementar el módulo Ejercicios cumpliendo con las necesidades reales, se requiere del análisis y diseño del mismo, siendo este el objetivo del presente trabajo. Para dar cumplimiento al objetivo se realizó un estudio del estado del arte sobre los sistemas similares existentes y las tecnologías a utilizar, son definidos los requisitos funcionales y no funcionales, se elaboró el diagrama de casos de uso del sistema y la descripción textual de cada uno, se realizaron los diagramas de clases del análisis y del diseño, los cuales constituyen un punto de partida para las actividades de implementación.

PALABRAS CLAVES: software educativo, colección El Navegante, módulo Ejercicios

ÍNDICE DE CONTENIDO

INTRODUCCIÓN	1
FUNDAMENTACIÓN TEÓRICA	5
1.1 Software educativo	5
1.2 Análisis de soluciones similares existentes	10
1.3 Tecnologías a utilizar en el desarrollo de la versión 2.0 del módulo Ejercicios	13
1.3.1 Metodología de desarrollo de software	14
1.3.2 Lenguaje de modelado	15
1.3.3 Herramienta CASE de modelado	16
1.3.4 Lenguajes de desarrollo.....	16
1.3.5 Framework	20
1.4 Arquitectura de software	22
1.5 Conclusiones	23
CARACTERÍSTICAS DEL SISTEMA	24
2.1 Modelo de dominio	24
2.1.1 Descripción de los objetos del dominio.....	25
2.2 Solución propuesta.....	27
2.3 Especificación de requerimientos	28
2.3.1 Requisitos funcionales	28
2.3.2 Requisitos no funcionales	32
2.4 Modelo de sistema	33
2.4.1 Descripción de los casos de uso del sistema.....	35
2.5 Conclusiones	56

ANÁLISIS Y DISEÑO DEL SISTEMA	57
3.1 Modelo de análisis	57
3.2 Modelo de diseño	60
3.2.1 Patrón de diseño a utilizar	61
3.2.2 Diagrama de paquetes.....	62
3.2.3 Diagramas de clases del diseño.....	64
3.3 Diseño de la base de datos	71
3.4 Conclusiones	71
CONCLUSIONES GENERALES	72
RECOMENDACIONES	73
REFERENCIAS BIBLIOGRÁFICAS	74
GLOSARIO DE TÉRMINOS	¡Error! Marcador no definido.

INTRODUCCIÓN

Las Tecnologías de la Información y las Comunicaciones (TIC), se desarrollan a un ritmo acelerado y marcan notablemente la sociedad donde son utilizadas, pues los profundos cambios se manifiestan en todos los ámbitos de la actividad humana, mejorando la calidad de vida de las personas y garantizando mayor cultura y beneficios económicos.

El Estado cubano no está exento a esta era de la información y el conocimiento, el uso masivo de las TIC se ha fomentado en todas las esferas de la sociedad como estrategia para beneficiar a la mayor parte posible de la población y de las instituciones, además de facilitar el desarrollo sostenible del país.

Uno de los pilares fundamentales de la Revolución cubana es la educación. La introducción de las TIC en este sector, ha provocado un proceso de transformaciones en los métodos tradicionales de enseñanza, desarrollándose amplios programas de elaboración de software educativo, con el fin de elevar la calidad del proceso de enseñanza-aprendizaje. Actualmente las escuelas cubanas disponen de colecciones de software educativo que apoyan este proceso, siendo el caso de Multisaber para la enseñanza primaria, El Navegante para la enseñanza secundaria y Futuro para la enseñanza preuniversitaria.

Como parte del fortalecimiento de las relaciones de los gobiernos de Cuba y Venezuela, en la X Reunión de la Comisión Intergubernamental Cuba-Venezuela desarrollada en diciembre del 2009, se firmó un contrato con el Ministerio del Poder Popular para la Educación de Venezuela, para la realización del proyecto “Colecciones de Software Educativo Multisaber y El Navegante para Planteles de los Niveles Primaria y Secundaria del Sistema Educativo”, con el objetivo de utilizarlas en las escuelas venezolanas.

Las colecciones Multisaber y El Navegante instaladas en las escuelas cubanas se desarrollaron con herramientas propietarias, tales como Macromedia Director y Toolbook Instructor, por lo que solo pueden ser utilizadas en el sistema operativo Windows. Los software educativos presentes en estas colecciones utilizan aspectos genuinamente cubanos que no son identificados en el entorno venezolano, por lo que se hace necesario contextualizar los contenidos y el diseño y además migrarlas a software libre, para poder cumplir con las políticas del estado venezolano establecidas en el contrato firmado sobre el uso de tecnologías libres.

A la Universidad de las Ciencias Informáticas (UCI), en conjunto con el Ministerio de Educación (MINED), se le ha asignado la tarea de desarrollar ambas colecciones utilizando tecnologías libres. La colección El Navegante está constituida por 10 productos, en los que se abordan temas de varias asignaturas, tales como Matemática, Geografía e Informática. Cada uno de los productos está estructurado en 6 módulos: Contenidos, Ejercicios, Mediateca, Juegos, Resultados y Maestro(a), tres de los cuales comenzaron a desarrollarse en enero de 2010, obteniéndose una primera versión funcional de ellos.

Uno de los módulos iniciados en la primera etapa de desarrollo fue Ejercicios, el cual alcanzó un alto nivel de avance, sin embargo, no logró cumplir con todos los requisitos de la colección, debido a que aún no permite asignar ejercicios a un equipo de trabajo integrado por 2, 3 o 4 estudiantes; no registra la traza de la interacción de los estudiantes con el módulo; la herramienta que gestiona los cuestionarios no permite agregar, eliminar o modificar medias (imágenes, videos, sonidos) en los mismos. Además, la primera versión de este módulo, no permite realizar ejercicios de las tipologías enlazar columnas, ordenar textos, seleccionar textos, completar por escritura (con corrector ortográfico opcional), formar conjuntos, seleccionar palabras, acentuar palabras y dividir en sílabas.

A partir de las necesidades identificadas en la primera versión del módulo Ejercicios de la colección El Navegante surge el **problema a resolver**: ¿Cómo lograr que el módulo Ejercicios de la colección El Navegante en su versión multiplataforma gestione las tipologías de cuestionarios interactivos no desarrolladas en la primera versión, y permita la interacción del usuario con estos?

A partir del problema de investigación se decide **estudiar** el software educativo con tecnología hipertexto dedicado a la visualización y gestión de cuestionarios interactivos.

Para lograr que el módulo Ejercicios de la colección El Navegante en su versión multiplataforma gestione las tipologías de cuestionarios interactivos no desarrolladas en la primera versión, y permita la interacción del usuario con estos, se pretende desarrollar una nueva versión, teniendo en cuenta la magnitud que presenta la misma y el tiempo que se necesita para desarrollarla, la presente investigación tiene como **objetivo general** realizar el análisis y diseño de la versión 2.0 del módulo Ejercicios de la colección El Navegante en su versión multiplataforma.

El **campo de acción** de la investigación está enmarcado en el proceso de análisis y diseño de la versión 2.0 del módulo Ejercicios de la colección El Navegante en su versión multiplataforma.

Se plantean como **objetivos específicos** de la investigación los siguientes:

- Realizar un estudio del estado del arte sobre el objeto de estudio de la presente investigación.
- Realizar el análisis de la versión 2.0 del módulo Ejercicios de la colección El Navegante en su versión multiplataforma.
- Diseñar la versión 2.0 del módulo Ejercicios de la colección El Navegante en su versión multiplataforma.

Para dar cumplimiento a los objetivos trazados se definieron las siguientes **tareas investigativas**:

- Investigación acerca de soluciones similares existentes.
- Investigación sobre los lenguajes, metodología y herramientas a utilizar en el desarrollo de la nueva versión del módulo Ejercicios.
- Definición de las funcionalidades a incluir en el módulo Ejercicios.
- Realización del modelo de sistema de la versión 2.0 del módulo Ejercicios.
- Realización de los diagramas de clases del análisis de la segunda versión del módulo Ejercicios.
- Realización de los diagramas de clases del diseño de la segunda versión del módulo Ejercicios.

Con el desarrollo de la investigación se espera obtener una completa documentación del análisis y diseño de la nueva versión del módulo Ejercicios de la colección El Navegante en su versión multiplataforma, de manera que guíe su proceso de desarrollo y permita que el mismo cumpla con las especificaciones del cliente y las necesidades del usuario final.

Para abordar el contenido asociado al presente trabajo, se decidió estructurarlo en 3 capítulos:

Capítulo 1: Fundamentación teórica. Se tratan los elementos teóricos relacionados con el objeto de estudio. Se realiza un análisis de los sistemas similares existentes en el mundo, además se realiza un estudio la metodología, lenguajes y herramientas a utilizar en el desarrollo de la versión 2.0 del módulo Ejercicios de la colección El Navegante.

Capítulo 2: Características del sistema. Se identifican y describen los conceptos asociados al ámbito del sistema, así como el modelo de dominio correspondiente. Son definidos los requisitos funcionales y no funcionales que debe cumplir el sistema, se identifican los actores y casos de uso del sistema, son elaborados el diagrama de casos de uso y la descripción textual de cada uno de estos.

Capítulo 3: Análisis y diseño del sistema. Se representa una vista interna del sistema a través de la confección de los diagrama de clases del análisis y a partir de los mismos se crean los diagramas de clases del diseño, obteniéndose una visión más detallada del sistema y una arquitectura más sólida. Además, se modela la base de datos del sistema.

Capítulo 1

FUNDAMENTACIÓN TEÓRICA

El surgimiento del software educativo estuvo propiciado por el desarrollo de la tecnología educativa, un punto de partida fue la aparición de las máquinas de enseñanza en 1920, diseñadas para que los estudiantes ejercitaran y practicaran eligiendo una respuesta entre varias opciones. En la década del 50 se alcanzó un mayor desarrollo en estas máquinas, permitiendo a los estudiantes componer sus respuestas escribiendo, además las preguntas eran diseñadas en una secuencia programada de lo más fácil a lo más difícil. A partir del funcionamiento de estas máquinas surgió la enseñanza programada, método mediante el cual el estudiante aprendía progresivamente, siendo él el responsable de su aprendizaje, a su propio ritmo y según sus posibilidades, sin la mediación directa de un profesor, a través de máquinas puramente mecánicas y juegos educativos. Luego surgió la enseñanza asistida por computadora, la cual adoptó principios de la enseñanza programada para apoyar el proceso de enseñanza – aprendizaje. Un medio didáctico creado con la finalidad específica de ser utilizado en este proceso es el software educativo. (1) (2)

1.1 Software educativo

Luego de un análisis detallado de las definiciones de software educativo dadas por diferentes investigadores, la autora de la presente investigación asume el concepto elaborado por el Dr Pere Marqués, el cual utiliza los términos software educativo, programas didácticos y programas educativos como sinónimos para designar genéricamente los programas para computadoras creados con la finalidad específica de ser utilizados como medio didáctico, es decir, para facilitar los procesos de enseñanza y de aprendizaje. (3)

Los software educativos pueden tratar diferentes materias (matemática, idiomas, geografía...), presentan cinco características esenciales: (3)

- Son materiales elaborados con una **finalidad didáctica**.
- **Utilizan la computadora** como soporte en el que los alumnos realizan las actividades que ellos proponen.
- **Son interactivos**, contestan inmediatamente las acciones de los estudiantes y permiten un diálogo y un intercambio de informaciones entre la computadora y los estudiantes.
- **Individualizan el trabajo** de los estudiantes, ya que se adaptan al ritmo de trabajo de cada uno y pueden adaptar sus actividades según las actuaciones de los alumnos.
- **Son fáciles de usar**. Los conocimientos informáticos necesarios para utilizar la mayoría de estos programas son mínimos, aunque cada programa tiene unas reglas de funcionamiento que es necesario conocer.

El software educativo presenta distintas características, existiendo por ello varias clasificaciones para los mismos. Según Álvaro Galvis Panqueva en su publicación Ingeniería de Software Educativo teniendo en cuenta el enfoque educativo y las funciones educativas que asumen se clasifican en:

Algorítmico: Software educativo en el que predomina el aprendizaje vía transmisión de conocimiento, desde quien sabe hacia quien lo desea aprender y el diseño de las actividades de aprendizaje conduce al aprendiz desde donde está hasta donde desea llegar. El rol del alumno es asimilar el máximo de lo que se le transmite. Se consideran software educativos de tipo algorítmico los siguientes: (4)

- **Sistemas tutoriales:** Incluye las cuatro fases que deben formar parte de todo proceso de enseñanza - aprendizaje: la fase introductoria, en la que se genera la motivación y se centra la atención en lo que se desea que el alumno aprenda; la fase de orientación inicial, en la que los estudiantes retienen los conocimientos; la fase de aplicación, en la que se recuerda lo aprendido; y la fase de retroalimentación en la que se demuestran y se refuerzan los conocimientos. (4)
- **Sistemas de ejercitación y práctica:** Se parte de la base de que antes de interactuar con el software, el alumno ya adquirió los conceptos y destrezas que va a practicar, permite que los estudiantes resuelvan una gran variedad de ejercicios aplicando lo aprendido y que obtengan más conocimientos acerca de los contenidos ejercitados. (4)

Heurístico: Software educativo en el que predomina el aprendizaje por descubrimiento, donde el diseño crea ambientes ricos en situaciones que el alumno debe explorar. El alumno debe llegar al conocimiento a partir de la experiencia, creando sus propios modelos de pensamiento, sus propias interpretaciones del mundo, se pueden comprobar a partir del mismo software los conocimientos adquiridos por el alumno. Se consideran software educativos de tipo heurístico los siguientes: (4)

- **Simuladores y juegos educativos:** Ambos pretenden lograr el aprendizaje por descubrimiento. En una simulación, aunque el micromundo suele ser una simplificación del mundo real, el alumno resuelve problemas, aprende procedimientos, llega a entender las características de los fenómenos y cómo controlarlos, aprende qué acciones tomar en diferentes circunstancias. Los juegos pueden o no simular la realidad, se caracterizan por proveer situaciones excitantes (retos) o entretenidas. Los juegos educativos buscan que dicho entretenimiento sirva de contexto al aprendizaje de algo, dependiendo de la naturaleza del juego. (4)
- **Lenguajes sintónicos y micromundos exploratorios:** Una forma particular de interactuar con micromundos es haciéndolo con ayuda de un lenguaje sintónico, es aquel con el que uno está sintonizado con sus instrucciones y que se puede usar naturalmente para interactuar con un micromundo. La principal utilidad de los lenguajes sintónicos es servir para el desarrollo de estrategias de pensamiento basadas en el uso de heurísticas de solución de problemas. (4)
- **Sistemas expertos:** Son sistemas de computación con capacidades para representar y razonar acerca de algún dominio rico en conocimientos, para resolver problemas y dar consejo a quienes no son expertos en la materia. La capacidad de razonar como un experto es lo que hace a los sistemas expertos particularmente útiles para que los aprendices ganen experiencia en dominios en que es necesario obtenerla y hagan explícito el conocimiento que está detrás de ella. (4)

Son considerados además software educativos los **sistemas tutoriales inteligentes**, los cuales no se pueden ubicar en una sola de las dos grandes categorías de software educativo, ya que se caracterizan por mostrar un comportamiento inteligentemente adaptativo, es decir, adapta el tratamiento educativo en función de aquello que se desea aprender y de las características y desempeño del aprendiz. La idea

básica es la de ajustar la estrategia de enseñanza - aprendizaje, el contenido y forma de lo que se aprende, a los intereses, expectativas y características del aprendiz. (4)

Según Pere Marqués en su artículo *El software educativo* se clasifican como software educativo además de los analizados anteriormente, los siguientes: (3)

- **Bases de datos:** Proporcionan datos organizados en un entorno estático, según determinados criterios, y facilitan su exploración y consulta selectiva. Según la forma de acceder a la información se pueden distinguir dos tipos:
 - **Bases de datos convencionales:** Tienen la información almacenada en ficheros, mapas o gráficos, que el usuario puede recorrer según su criterio para recopilar información.
 - **Bases de datos tipo sistema experto:** Son bases de datos muy especializadas que recopilan toda la información existente de un tema concreto y además asesoran al usuario cuando accede buscando determinadas respuestas.

- **Constructores:** Son programas que tienen un entorno programable. Potencian el aprendizaje heurístico y facilitan a los alumnos la construcción de sus propios aprendizajes, a través de la reflexión que realizan al diseñar programas y comprobar los resultados obtenidos. Entre los constructores se pueden distinguir los lenguajes de programación¹.

Según José Antonio Gutiérrez Lam en su artículo *El proyecto de Software Educativo. Algunas consideraciones*, son considerados software educativo los **libros electrónicos**, estos presentan información al estudiante a partir del uso de textos, gráficos, animaciones, videos, con un nivel de interactividad y motivación que le facilite las acciones que realiza. (5)

En Cuba se han creado Centros de estudios de software educativo en todos los Institutos Superiores Pedagógicos y como consecuencia del desarrollo alcanzado el Ministerio de Educación ha desarrollado el

¹ Un lenguaje de programación es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes.

Disponible en <http://www.definicion.org/lenguaje-de-programacion>

concepto de **hiperentorno de enseñanza - aprendizaje** (HEA), el cual consiste en una combinación de diferentes tipos de software educativo, basados en tecnología hipermedia. (6)

Como se mencionó anteriormente los hiperentornos de enseñanza - aprendizaje están basados en la tecnología hipermedia, para que se desarrollara esta tecnología en el mundo, surgió antes lo que se conoce como **multimedia**, este término es muy utilizado en el contexto de las tecnologías de la información. En el ámbito digital se refiere a “la integración en un sistema informático de textos, gráficos, imágenes, videos, animaciones, sonidos y cualquier otro medio que pueda ser tratado digitalmente”. (7) “Si la presentación multimedia permite al usuario actuar sobre la secuencia, velocidad o cualquier otro elemento de su desarrollo, o plantea preguntas, pruebas o alternativas que modifican su transcurso, entonces la calificamos como de multimedia interactiva”. (7)

Otra tecnología que propició el desarrollo de la hipermedia fue el **hipertexto**, que no es más que la organización de la información en diversos bloques de contenidos o nodos, conectados mediante enlaces en forma de red y permite que los usuarios, según sus preferencias puedan acceder a la información de forma no secuencial, es decir, no existe un orden único para acceder a cada nodo. (8)

En resumen, la tecnología **hipermedia** surge mediante la integración del hipertexto y la multimedia, es la organización de imágenes, videos, textos, sonidos y cualquier otro medio digitalizado en diversos nodos interconectados mediante enlaces, los nodos pueden ser accedidos de forma no lineal. La hipermedia facilita la interacción del usuario con la información de una manera amigable y sencilla. (8)

En la actualidad, la hipermedia constituye la base tecnológica en la creación de software educativo en Cuba, ya que provee una estructura de navegación a través de una gran variedad de medios, esta estructura puede tener componentes dinámicos y también puede ser controlada por el usuario, permitiendo una sincronización idónea de los medios durante la visualización del software. Además, esta tecnología es muy utilizada con fines docentes a nivel internacional.

La colección El navegante que se encuentra actualmente en desarrollo integra varias formas de presentación de la información: textos, imágenes, videos, sonidos y animaciones, las cuales permiten mantener la motivación e interés del alumno en el estudio. La colección tiene como objetivos que los estudiantes aprendan contenidos nuevos de una manera interactiva; contribuir al desarrollo de habilidades

mediante la ejercitación y práctica; evaluar los conocimientos de los estudiantes sobre determinados temas o contenidos; promover el aprendizaje de manera entretenida a través de juegos educativos; simular experimentos referentes a temáticas de las asignaturas mediante laboratorios virtuales que apoyen a los estudiantes a resolver problemas y entender fenómenos semejantes a la realidad. Los productos de dicha colección tienen una estructura de navegación que permite a los estudiantes interactuar con la información según sus necesidades y preferencias, es decir, en el orden que desean.

Teniendo en cuenta las funciones de la colección El Navegante se puede llegar a la conclusión de que los software educativos que la conforman son hiperentornos de enseñanza - aprendizaje, ya que están presenten en los mismos varios tipos de software como: libro electrónico, sistema de ejercitación y práctica, juegos educativos y simuladores.

1.2 Análisis de soluciones similares existentes

Actualmente en el mundo existen aplicaciones que permiten realizar cuestionarios interactivos y además gestionan los contenidos de los mismos, algunas de estas aplicaciones son:

iTest

Es una herramienta de evaluación online que permite a profesores configurar exámenes y a estudiantes realizarlos utilizando únicamente un navegador. Es un subconjunto de un Entorno Virtual de Aprendizaje. Cada uno de los exámenes se genera y corrige automáticamente de manera instantánea, desde una base de datos de preguntas creada y gestionada por cada profesor. Los alumnos pueden realizar exámenes y revisarlos corregidos. La herramienta muestra la calificación obtenida en pantalla una vez finalizado el examen. (9)

iTest posibilita también la configuración de exámenes que incluyan archivos multimedia como imágenes, sonidos, animaciones y partituras musicales en los enunciados de las preguntas y en las respuestas de manera muy rápida e intuitiva, además se pueden incluir fórmulas y gráficas de funciones matemáticas. Esto hace que iTest sea una herramienta muy versátil y adaptable a las distintas asignaturas y a todos los niveles educativos. (9)

Esta herramienta es de Licencia GNU/ Libre (sin costo), se puede utilizar en los sistemas operativos: Mac OS X, Linux/Unix, Microsoft Windows. (10)

QuizFaber

Permite generar diversos tipos de ejercicios interactivos y aplicar en ellos las ventajas de los materiales multimedia, pues se pueden insertar imágenes, sonidos y videos. Permite crear ejercicios de selección múltiple, respuesta abierta, verdadero/falso, rellenar espacios, unir elementos y se puede personalizar algunos detalles de su diseño, como colores y fuentes. (11)

Es un software gratuito para Windows, compatible con los navegadores actuales, se pueden crear exámenes en varios idiomas, el resultado de los mismos se puede guardar en un servidor web o enviar por correo electrónico. Tiene una versión portable. Se puede ejecutar en discos extraíbles. (12)

Hot potatoes

Es un programa gratuito, y puede ser utilizado para cualquier propósito o proyecto que se desee. No es de código abierto. Cuenta con seis componentes que se describen a continuación: (13)

- **JQuiz:** Permite elaborar ejercicios basados en preguntas. Cada test puede estar formado por un número ilimitado de preguntas. Se pueden formular cuatro tipos de preguntas:
 - **Elección múltiple:** Los alumnos han de elegir una respuesta, si es incorrecta el alumno podrá seguir seleccionando opciones hasta dar con la respuesta correcta.
 - **Respuestas cortas:** Los alumnos deberán escribir una respuesta en una casilla de texto.
 - **Pregunta híbrida:** Es una combinación de una pregunta de respuestas múltiples y una pregunta de respuesta corta. Al alumno se le presenta una casilla de texto, si no responde correctamente después de un número de intentos, la pregunta cambia a otra de respuestas múltiples para que sea más fácil para el alumno.
 - **Multi-selección:** Consiste en que el alumno ha de seleccionar varias respuestas de una lista definida previamente.
- **JCloze:** Se utiliza para elaborar ejercicios de completar espacios en blanco.

- **JMatch**: Se utiliza para crear ejercicios de emparejamiento. Se disponen una serie de elementos a un lado y cada uno de ellos ha de emparejarse con los elementos dispuestos en el lado opuesto.
- **JCross**: Se utiliza para realizar crucigramas.
- **The Masher**: Se utiliza para crear unidades completas de ejercicios de Hot Potatoes. Una unidad compartirá el mismo aspecto y configuración, y los ejercicios que la componen estarán enlazados automáticamente mediante botones de navegación. Para utilizar The Masher se necesita una clave de registro independiente, que solamente puede conseguir si se adquiere la versión comercial de Hot Potatoes. Sin dicha clave de registro, solamente se podrá elaborar pequeñas unidades de ejercicios.

Ardora

Es una aplicación informática para docentes, que permite crear actividades para los alumnos de un modo muy sencillo. Se pueden crear más de 45 tipos distintos de actividades, tales como crucigramas, sopas de letras, completar, paneles gráficos, relojes, entre otras, así como más de 10 tipos distintos de páginas multimedia: galerías, panorámicas o zooms de imágenes, reproductores mp3 o flv, etc. (14)

Ardora es gratuito, siempre y cuando sea usado de forma personal, sin carácter lucrativo y con fines estrictamente educativos. No es un programa de código abierto. (14)

Moodle

Es un Sistema de Gestión de Cursos de Código Abierto (Open Source Course Management System, CMS), 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. (15)

Moodle se ofrece libremente como software de código abierto (bajo la Licencia Pública GNU). Es multiplataforma, se puede ejecutar en Windows y sistemas operativos Mac y en las diferentes distribuciones de Linux (por ejemplo Red Hat, Debian o Ubuntu). (15)

Los tipos de preguntas que se pueden presentar en un cuestionario son seis: selección simple, selección múltiple (emparejamiento), verdadero o falso, numérico, respuesta corta o rellena los blancos. (16)

El Navegante

La colección El Navegante instalada en las escuelas secundarias de Cuba cuenta con 10 productos, en cada uno están presentes 6 módulos: Temas, Ejercicios, Juegos, Biblioteca, Registro y Maestro. El módulo Ejercicios permite realizar ejercicios de las tipologías: selección simple, selección múltiple, completamiento por desplazamiento y escritura, verdadero y falso, seleccionar textos, seleccionar palabras, pero no brinda la posibilidad de modificar los existentes ni agregar nuevos. Esta colección fue desarrollada con herramientas propietarias y solo puede ser usada en el sistema operativo Windows.

Análisis general de las herramientas educativas estudiadas

Luego de analizadas las características de las herramientas educativas consultadas, se observó que no gestionan todas las tipologías de ejercicios necesarias para el módulo: selección simple, selección múltiple, verdadero o falso, ordenar pasos, cálculo estimado, completar por desplazamiento (con o sin restitución), completar por escritura (con corrector ortográfico opcional), enlazar columnas, ordenar textos, seleccionar textos, formar conjuntos, dividir en sílabas, acentuar palabras y seleccionar palabras. iTest no permite crear las preguntas de los exámenes por tipología; QuizFaber, Ardora, Hot Potatoes, y El Navegante (versión instalada en las escuelas cubanas) no son de código abierto, por lo que no permiten agregar las tipologías de ejercicios que no gestionan. Debido a que ninguna cumple con las características que debe tener el módulo Ejercicios se decide desarrollar una versión.

1.3 Tecnologías a utilizar en el desarrollo de la versión 2.0 del módulo Ejercicios

Teniendo en cuenta que el objetivo de la presente investigación es realizar el análisis y diseño de las funcionalidades que se necesitan agregar en el módulo Ejercicios y que anteriormente fueron seleccionadas las tecnologías más apropiadas para el desarrollo de dicho módulo, se decide en esta investigación continuar utilizando las mismas tecnologías y herramientas para la documentación del desarrollo de la nueva versión, teniendo en cuenta además que estas se utilizaron en el desarrollo de la primera versión y en pruebas realizadas a la misma no presentó problemas de inestabilidad, ni de rendimiento. A continuación se describen algunas de sus características principales.

1.3.1 Metodología de desarrollo de software

“Un proceso define quién está haciendo qué, cuándo, y cómo alcanzar un determinado objetivo”. (17)

Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software, guía los esfuerzos de las personas implicadas en el proyecto, explicando los pasos necesarios para terminar el proyecto. (17)

Las metodologías según su filosofía de desarrollo se clasifican en dos grupos: en tradicionales o ágiles, las tradicionales se caracterizan por una fuerte planificación, una documentación exhaustiva y un control del proyecto durante todo el proceso de desarrollo, entre las más utilizadas se encuentran Rational Unified Process (RUP) y Microsoft Solutions Framework (MSF). Las metodologías ágiles están especialmente indicadas en el desarrollo de software donde los requisitos no estén bien definidos o cambien con mucha frecuencia, ya que es adaptable al cambio, están dedicadas a proyectos pequeños, de corta duración, en los que exista una estrecha comunicación entre el cliente y los desarrolladores. Entre estas metodologías se destacan Extreme Programming (XP) y Scrum. (18)

La metodología seleccionada para guiar el proceso de desarrollo del módulo Ejercicios en su segunda versión fue RUP, debido a que no existe una comunicación directa e inmediata con los clientes, toda la interacción es a través de terceros, esta metodología es adaptable a las necesidades de cada organización, permite obtener una completa documentación del proceso de desarrollo, aspecto muy importante, pues la documentación obtenida en la primera etapa de desarrollo del módulo Ejercicios sirve de apoyo al proceso de desarrollo de la nueva versión, ya que proporciona un mejor entendimiento de sus características y de su funcionamiento. Además, el desarrollo iterativo facilita la reusabilidad de partes implementadas, asegura que todos los requerimientos se implementen correctamente y se corrigen los errores en cada iteración. RUP ofrece una mejor comprensión del sistema debido a la gran cantidad de artefactos que se generan durante el proceso de desarrollo.

El Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones y diferentes niveles de aptitud; utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un software. (17)

Lo que hace único al Proceso Unificado de Desarrollo es que se caracteriza por ser **dirigido por caso de uso**, a partir de los mismos se detallan los requisitos funcionales del sistema y se construyen los artefactos resultantes del análisis, diseño, implementación y prueba de un software; es **centrado en la arquitectura**, la cual es una vista del diseño completo con las características más importantes resaltadas, permite que el equipo de trabajo se enfoque en los aspectos más importantes para su construcción, además RUP se distingue por ser **iterativo e incremental** ya que propone realizar un software en iteraciones, en las que se identifican, especifican, diseñan e implementan los casos de uso más relevantes, si se cumple con los objetivos de la iteración se obtiene un incremento o crecimiento en el producto y se continúa con la siguiente iteración.

1.3.2 Lenguaje de modelado

Un lenguaje proporciona un vocabulario y unas reglas para permitir la comunicación. Un lenguaje de modelado se centra en la representación gráfica de un sistema, capturando las partes esenciales del mismo. El **Lenguaje Unificado de Modelado (UML)** sirve para visualizar gráficamente sistemas complejos, tanto en el diseño del software como de la arquitectura hardware donde se ejecutan, siendo esta la razón fundamental por la que se decide utilizar UML como lenguaje para la representación gráfica y documentación de los artefactos generados en el desarrollo del módulo. (19)

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones: (19)

- Visualizar: Permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- Especificar: Permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado y pueden ser utilizados para su futura revisión.

El lenguaje UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático, desde el análisis con los casos de uso, el diseño con los diagramas de clases, hasta la implementación y configuración con los diagramas de despliegue. (19)

UML aporta beneficios al desarrollo del módulo Ejercicios, ya que permite realizar una representación gráfica del mismo con un gran rigor en la especificación, lo que proporciona un mayor entendimiento de sus características y facilita su proceso de desarrollo.

1.3.3 Herramienta CASE de modelado

Las herramientas CASE (por sus siglas en inglés: **C**omputer **A**ided **S**oftware **E**ngineering, o Ingeniería de Software Asistida por Computadora) son aquellas aplicaciones que brindan ayuda al desarrollo de software, disminuyendo su costo en términos de tiempo y de dinero. Garantizan una correcta documentación, generación de código y reutilización del software.

Visual Paradigm for UML 6.4 (VP-UML)

Es una herramienta CASE profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite realizar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. . Una de sus características más importantes es que es multiplataforma. (20)

Visual Paradigm es la herramienta CASE seleccionada para el modelado UML debido a que soporta el ciclo de vida completo del desarrollo de software, por lo que facilita la construcción de sistemas con calidad y de forma rápida; permite la generación de código y de bases de datos. Además es multiplataforma, soporta ingeniería inversa, genera documentación del sistema en formato HTML y PDF.

1.3.4 Lenguajes de desarrollo

Un lenguaje de programación es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes. (21)

Actualmente existen diversos lenguajes de programación para desarrollar en la web, ya que, con el avance de las tecnologías surgieron lenguajes que permiten interactuar con los usuarios haciendo uso de sistemas de base de datos.

Para el desarrollo del módulo Ejercicios se decidió utilizar los lenguajes del lado del cliente XHTML, CSS y JavaScript y como lenguaje del lado del servidor PHP. A continuación se describen las características de cada uno de estos y los beneficios que aportan al módulo Ejercicios.

XHTML

XHTML (Extensible Hypertext Markup Language) es un conjunto de documentos basados en XML que nace como extensión del HTML 4.0 y que permite la interoperabilidad entre aplicaciones basadas en XML. Se trata de un paso adelante en el campo del desarrollo Web, al aprovecharse las ventajas del XML, mantiene la compatibilidad de las aplicaciones de usuario realizadas con versiones distintas. (22)

Su empleo proporciona una serie de ventajas: (22)

- Basado en XML, lo que permite un sencillo manejo mediante herramientas XML Estándar.
- Compatibilidad con aplicaciones de usuario desarrolladas con HTML 4.0 y XHTML 1.0, obteniéndose un comportamiento idéntico en ambos tipos de aplicación.
- Posibilidad de crear contenidos Web y aplicaciones de usuario mediante técnicas de desarrollo modular. En este sentido, XHTML 1.0 se extiende y acota a través de un mecanismo que permite definir módulos (conjuntos de elementos) y combinarlos entre sí en función de las necesidades de cada ámbito.

Con el uso de este lenguaje se garantiza que el módulo Ejercicios tenga gran compatibilidad entre los navegadores que cumplen con estándares web, para lograr que en las nuevas versiones de los navegadores también el módulo funcione correctamente. Presenta buena facilidad de actualización, ya que al crear un XHTML correcto y darle los estilos desde un CSS (Cascading Style Sheets), es posible crear webs totalmente diferentes sin modificar absolutamente nada del XHTML. Además, aumenta la velocidad de desarrollo así como de navegación, pues permite crear y maquetar páginas webs correctamente, las cuales se cargan muchísimo más rápido en los navegadores.

CSS

Una hoja de estilo CSS (Cascading Style Sheets) es un lenguaje creado por el W3C que permite a los desarrolladores controlar los aspectos de estilo y formato de múltiples páginas (X)HTML simultáneamente, permitiendo separar el contenido de presentación en todo momento. (22)

El empleo de hojas de estilo CSS es una práctica que aporta una serie de beneficios como: (22)

- Aumento de la legibilidad y reducción del peso de las páginas Web.
- Mejora del mantenimiento y actualización de los sitios Web.
- Mejora de la accesibilidad: se pueden definir hojas de estilo locales en función de las necesidades o preferencias del usuario.
- Versatilidad: Se ofrecen diferentes hojas de estilo para los diferentes tipos de medio existentes (sintetizadores de voz, dispositivos braille, pantallas de computador a color, televisión).

Junto al lenguaje XHTML, el lenguaje CSS da la posibilidad de tener el control del diseño en la aplicación, ya que como se ha dicho, es muy útil para separar el contenido del diseño. Además, permite personalizar la aplicación en dependencia del dispositivo que se esté usando.

JavaScript

Es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Técnicamente, es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. (23)

JavaScript es seguro y fiable ya que los scripts tienen capacidades limitadas por razones de seguridad, es ejecutado en el cliente por lo que el servidor no es utilizado más de lo debido, aunque tiene la desventaja que el código del script debe descargarse completamente para poder ejecutarlo.

Este lenguaje brinda la posibilidad de establecer una conexión asincrónica con el servidor; lo que garantiza que no haya una sobrecarga de peticiones al mismo y el aumento de la velocidad de respuesta. Además, con JavaScript se puede tener el control total de las animaciones y la manipulación de los componentes en la página web.

PHP 5.2

PHP (acrónimo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular, especialmente adecuado para desarrollo Web y que puede ser incrustado en HTML. Con PHP se puede procesar la información de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies. (24)

Las características que distinguen a este lenguaje y que le hacen uno de los más usados son: (24)

- Puede ser utilizado en cualquiera de los principales sistemas operativos del mercado, incluyendo Linux, muchas variantes Unix, Microsoft Windows, Mac OS X, RISC OS, entre otros.
- Soporta la mayoría de servidores web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Netscape e iPlanet, O'Reilly Website Pro server, Caudium, Xitami, OmniHTTPd y muchos otros.
- Tiene módulos disponibles para la mayoría de los servidores, para aquellos otros que soporten el estándar CGI, PHP puede usarse como procesador CGI.
- Soporte para una gran cantidad de bases de datos: Adabas D, dBase, Empress, FilePro (solo lectura), Hyperwave, IBM DB2, Informix, Ingres, InterBase, FrontBase, mSQL, Direct MS-SQL, MySQL, ODBC, Oracle (OCI7 y OCI8), Ovrimos, PostgreSQL, SQLite, Solid, Sybase, Velocis, Unix dbm.
- Tiene soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.

Una de las ventajas que brinda el lenguaje PHP al utilizarlo en el módulo Ejercicios es la posibilidad de programar orientado a objetos, lo cual proporciona mucha claridad al desarrollo. Es uno de los lenguajes de programación web que más documentación tiene, dispone de un sitio oficial que brinda toda la

documentación que los desarrolladores necesiten, además existe una gran comunidad de desarrollo que aporta sus conocimientos.

1.3.5 Framework

El término framework se refiere a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, se puede considerar como una aplicación genérica incompleta y configurable a la que se puede añadir las últimas piezas para construir una aplicación concreta. (25)

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. (25)

Un framework Web, por tanto, se puede definir como un conjunto de componentes que forman un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web.

El uso de un framework: (26)

- Simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes.
- Proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener.
- Facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

En el desarrollo de la segunda versión del módulo Ejercicios se utilizarán como framework de apoyo a la implementación en el lado del cliente jQuery y en el lado del servidor Symfony. Los beneficios que aportan al módulo y sus principales características se muestran a continuación.

Symfony 1.4.3

Symfony es un framework para construir aplicaciones web con PHP. En otras palabras, es un enorme conjunto de herramientas y utilidades que simplifican el desarrollo de las aplicaciones web. (27)

Es uno de los framework PHP más populares entre los usuarios y las empresas, ya que permite que los programadores sean mucho más productivos a la vez que crean código de más calidad y más fácil de mantener. Symfony es maduro, estable, profesional y está muy bien documentado. (27)

Algunas de las características más importantes de Symfony son: (28)

- Fácil de instalar y configurar en la mayoría de plataformas
- Independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Es utilizado Symfony como framework de apoyo a la implementación del módulo en el lado del servidor porque facilita el desarrollo de aplicaciones web basadas en lenguaje PHP. Sigue las mejores prácticas y patrones de desarrollo para la web, como la separación estricta de código y diseño; usa el patrón Modelo Vista Controlador (MVC). Es muy seguro ya que incluye por defecto protección contra ataques de tipo Cross Site Scripting (XSS) y Cross Site Request Forgery (CSRF). Permite el desarrollo de aplicaciones web de forma rápida. Además, posee una amplia documentación y es soportado por diversas plataformas.

jQuery 1.4

jQuery es una biblioteca JavaScript muy rápida y muy ligera que simplifica el desarrollo de la parte del cliente de las aplicaciones web. En otras palabras, incluye muchas utilidades para crear fácilmente las páginas web de las aplicaciones dinámicas complejas. (29)

El uso de jQuery permite a los desarrolladores el ahorro de muchas líneas de código ya que en JavaScript habría que hacer scripts compatibles con varios navegadores y jQuery implementa una serie de clases que permiten programar sin tener en cuenta el navegador del cliente. Esta biblioteca facilita la creación de

aplicaciones complejas del lado del cliente pues brinda ayuda en la creación de interfaces de usuario, para animar el contenido de las páginas de forma dinámica. Permite trabajar con aplicaciones que hacen uso de Ajax (**A**synchronous **J**avaScript **A**nd **X**ML). Provee un mecanismo para la captura de eventos.

jQuery tiene licencia para uso en cualquier tipo de plataforma, está soportado por una gran comunidad de creadores de plugins o componentes, lo que permite encontrar fácilmente soluciones ya creadas. Además, está bien documentado y es muy estable ya que cuenta con una gran comunidad de desarrolladores encargados de la actualización y mejora del mismo.

1.4 Arquitectura de software

El patrón de arquitectura modelo-vista-controlador (MVC) especifica que una aplicación consta de un modelo de datos, de información de presentación y de información de control. El patrón requiere que cada uno de estos elementos esté separado en distintos objetos: (30)

- El modelo (la información de datos) contiene únicamente los datos puros de aplicación; no contiene la lógica que describe cómo pueden presentarse los datos a un usuario.
- La vista (la información de presentación) presenta al usuario los datos del modelo. La vista sabe cómo acceder a los datos del modelo, pero no sabe el significado de estos datos ni lo que el usuario puede hacer para manipularlos.
- El controlador (la información de control) está entre la vista y el modelo. Escucha los sucesos desencadenados por la vista (u otro origen externo) y ejecuta la reacción apropiada a estos sucesos. En la mayoría de los casos, la reacción es llamar a un método del modelo. Puesto que la vista y el modelo están conectados a través de un mecanismo de notificación, el resultado de esta acción se reflejará automáticamente en la vista.

El uso del patrón MVC ofrece para el desarrollo de las aplicaciones múltiples ventajas, ya que define una separación entre datos y presentación, presenta gran sencillez para crear distintas representaciones de los mismos datos, permite a los desarrolladores la reutilización de los componentes, la tarea de mantenimiento de los sistemas se simplifica grandemente, brinda a los desarrolladores la facilidad de desarrollar prototipos rápidos y aplicaciones escalables.

1.5 Conclusiones

En este capítulo se analizaron los conceptos esenciales relacionados con el software educativo, teniendo en cuenta además las características de los sistemas similares que permiten realizar cuestionarios interactivos y gestionan los contenidos de los mismos, para conocer las tendencias actuales de su desarrollo, identificándose que ninguno de los consultados soporta el modelo pedagógico establecido para la colección El Navegante.

Además se realizó una investigación acerca de las tecnologías que fueron seleccionadas en previas investigaciones y que serán utilizadas en el desarrollo de la nueva versión del módulo Ejercicios. RUP será la metodología de desarrollo de software a emplear y UML será utilizado para modelar el sistema con la herramienta CASE Visual Paradigm. Para el desarrollo, que estará basado en una arquitectura Modelo Vista Controlador se utilizarán los lenguajes de desarrollo XHTML, CSS, JavaScript y PHP, además los frameworks Symfony y jQuery.

Capítulo 2

CARACTERÍSTICAS DEL SISTEMA

El primer paso en el proceso de desarrollo de software es comprender el contexto del sistema, es decir, obtener un cierto nivel de conocimiento sobre el problema en cuestión. Para lograr una correcta descripción del sistema a diseñar, se realiza un estudio profundo de la primera versión del módulo Ejercicios de la colección El Navegante (multiplataforma) y de la herramienta que gestiona los cuestionarios interactivos, para entender sus características esenciales.

En el presente capítulo se identifican y describen los conceptos más significativos presentes en el entorno del sistema a desarrollar, los cuales conforman el Modelo de dominio, con el objetivo de comprender el contexto del sistema y los requisitos funcionales y no funcionales que se desprenden de este. Además, se identifican los actores que interactúan con el sistema y los casos de uso que describen su comportamiento, se estructura el diagrama de casos de uso y se describen detalladamente cada uno.

2.1 Modelo de dominio

“Un Modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las "cosas" que existen o los eventos que suceden en el entorno en el que trabaja el sistema.” (17)

El objetivo del Modelo de dominio es ayudar a comprender el contexto del sistema y la descripción de los objetos o clases conceptuales más importantes que están presentes en el dominio del problema. Se representa mediante diagramas de clases UML, los cuales muestran las clases del dominio y la relación de asociación que existe entre estas; además ayudan a todos los implicados en el desarrollo de un sistema software a utilizar un vocabulario común, lo que permite un mejor entendimiento del problema existente.

Para obtener un mayor conocimiento del problema existente y concebir el funcionamiento del sistema, debido a que, no se pueden identificar los procesos del negocio claramente, las personas que lo inician ni

las que desarrollan cada una de las actividades que forman parte de los procesos, se hace necesaria la definición de diferentes conceptos que se agrupan en un Modelo de dominio.

2.1.1 Descripción de los objetos del dominio

Se identificaron los principales conceptos presentes en el contexto del problema, los cuales se describen a continuación:

Colección El Navegante: Es una colección de software educativos multiplataforma, creados con la finalidad de apoyar el proceso de enseñanza – aprendizaje de los estudiantes de la enseñanza secundaria en las escuelas venezolanas.

Producto: Sistema informático que combina diversos tipos de medias.

Módulo: Se le denomina al conjunto de elementos fundamentales que componen el producto.

Contenidos: Módulo en el que se presentan los contenidos de las asignaturas o temas a las cuales corresponde el producto.

Ejercicios: Módulo que permite a los estudiantes ejercitar y comprobar sus conocimientos.

Mediateca: Módulo que agrupa las medias presentes en el producto.

Juegos: Módulo que permite a los estudiantes divertirse mientras emplean sus habilidades y conocimientos.

Resultados: Módulo que visualiza la traza de la interacción del estudiante con los restantes módulos de la aplicación.

Maestro(a): Módulo que le permite al maestro(a) configurar algunas opciones del producto en cuestión.

Tipología: Clasificación de un ejercicio según la forma de proceder para su solución.

Tipología selección simple: A partir del enunciado de la pregunta se debe seleccionar solo una de las opciones de respuesta que se le presentan.

Tipología selección múltiple: A partir del enunciado de la pregunta se pueden seleccionar una o varias de las opciones de respuesta que se le presentan.

Tipología verdadero o falso: Se debe emitir un criterio de veracidad sobre cada una de las proposiciones que se presentan.

Tipología ordenar pasos: Se presenta de forma desordenada una serie de pasos o acciones a colocar en un orden establecido.

Tipología cálculo estimado: Se presenta un problema en el cual la respuesta se establece de forma aproximada y el estudiante en un rayo numérico debe desplazar un indicador hasta el lugar que estima que es correcto para la solución. Se establece un rango de solución correcta.

Tipología completar por desplazamiento (con o sin restitución): Se presenta un problema o texto base en el cual existen espacios en blanco a completar. Se dispone además de un área de opciones donde se encuentran dispersas de forma aleatoria las opciones de respuesta. El completamiento sin restitución es aquel en el que puede arrastrarse una opción de respuesta solo una vez al área a completar, mientras que el completamiento con restitución es en el que la opción de respuesta puede ser utilizada más de una vez para completar.

Media: Son textos, imágenes, sonidos o videos.

Texto: Es una composición de signos codificado en un sistema de escritura (como un alfabeto) que forma una unidad de sentido.

Sonido: Media que brinda información sonora sobre un contenido.

Imagen: Representación gráfica de un contenido.

Video: Producto audiovisual capturado de la realidad objetiva, mediante cámaras.

Retroalimentación: Es un proceso mediante el cual el sistema y el usuario interactúan por medio de mensajes, permiten al sistema expresar de forma afectiva la evaluación de la respuesta de un ejercicio, estos mensajes se visualizan acompañados de una locución; además permiten brindar una pista al

usuario para que el mismo reflexione y responda de forma correcta; también mostrar información acerca de la respuesta correcta del ejercicio.

A continuación se muestra el modelo de dominio que agrupa y relaciona los conceptos analizados anteriormente.

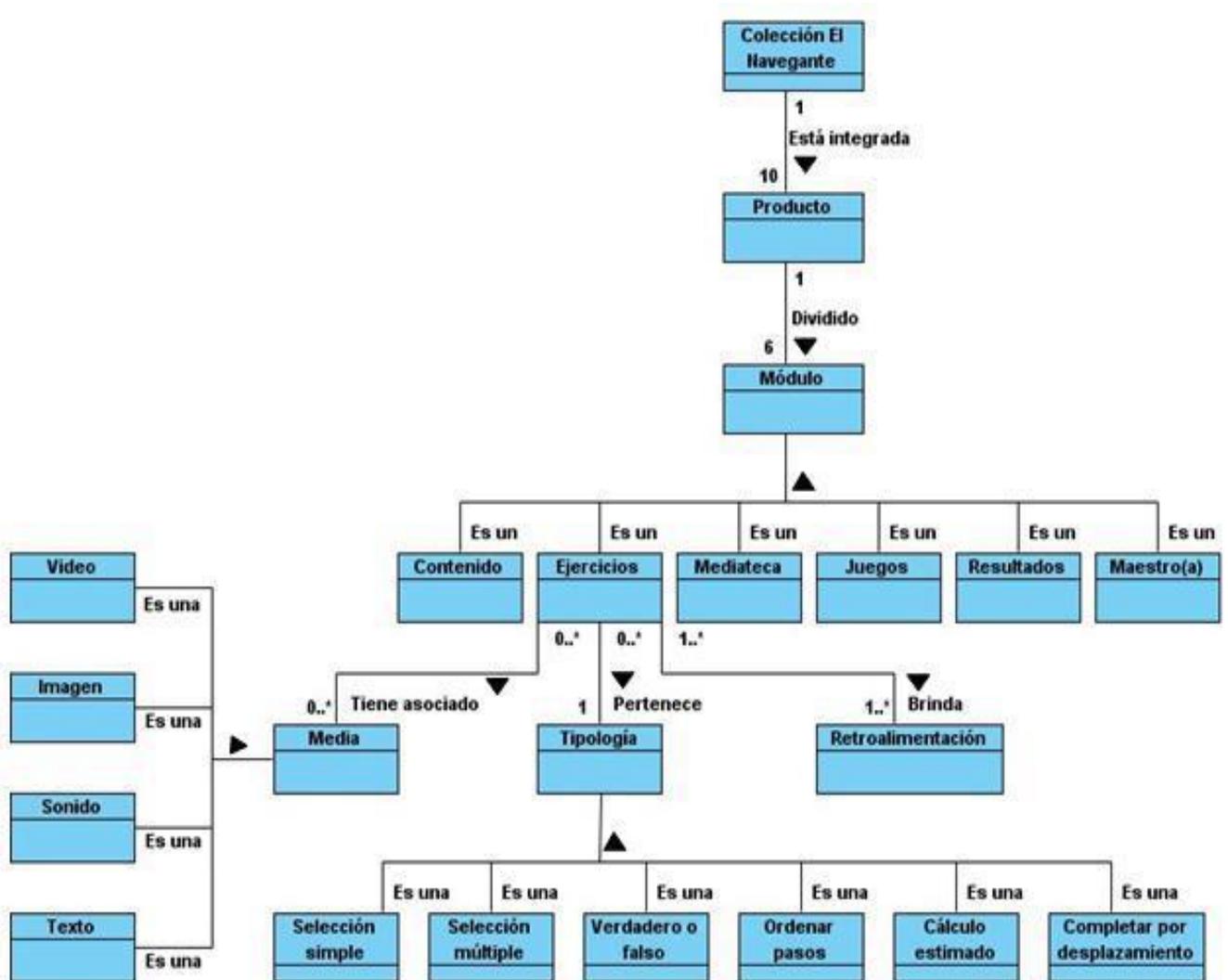


Figura 2.1. Modelo de dominio

2.2 Solución propuesta

A partir de un estudio realizado al módulo Ejercicios de la colección El Navegante y a la herramienta que gestiona los cuestionarios interactivos, se identificó la necesidad de añadir funcionalidades a ambas

aplicaciones. En vista de facilitar el proceso de desarrollo de las mismas, se realiza el análisis y diseño de una nueva versión del módulo Ejercicios y de la herramienta de administración; obteniéndose la documentación que describe sus características, su estructura de clases y relaciones entre ellas. Se elabora el diagrama de casos de uso del sistema, las descripciones textuales de cada uno de estos, así como los diagramas de clases del análisis y los diagramas de clases del diseño, artefactos que permiten obtener una visión general del sistema y constituyen la base para realizar las actividades de implementación y prueba.

2.3 Especificación de requerimientos

Una vez alcanzado una visión general de los conceptos que forman parte del contexto del sistema, se hace necesario comenzar a definir qué es lo que debe hacer este y cómo debe hacerlo, identificando los requisitos funcionales y no funcionales; los funcionales son las condiciones o capacidades que debe cumplir el sistema y los no funcionales son las propiedades o cualidades que debe tener.

A continuación se muestran los requisitos funcionales y no funcionales del módulo Ejercicios de la colección El Navegante y de la herramienta de administración.

2.3.1 Requisitos funcionales

RF 1: Mostrar la pantalla principal del módulo Ejercicios.

RF 2: Mostrar por niveles el menú de contenidos con ejercicios disponibles.

RF 3: Mostrar las opciones de selección de ejercicios (Secuencial, Al azar y Asignados).

RF 4: Permitir la selección de los contenidos de los que se desea realizar ejercicios.

RF 4.1: Mostrar la cantidad de ejercicios disponibles del contenido o los contenidos seleccionados según el tipo de selección de ejercicios elegido.

RF 4.1.1: Mostrar el total de ejercicios en el campo “Hasta” de la selección Secuencial o en el campo “Cantidad” de la selección Al azar o mostrar el número de cada ejercicio disponible por contenido en el campo “Disponibles” de la selección de ejercicios Asignados.

RF 5: Permitir especificar el tipo de selección de ejercicios.

RF 5.1: Secuencial: especificar en el campo “Desde” el número del ejercicio desde el que se desea comenzar a visualizar los ejercicios y en el campo “Hasta” el número del último ejercicio a mostrar.

RF 5.2: Al Azar: especificar en el campo “Cantidad” el número total de ejercicios que se desea visualizar.

RF 5.3: Asignados:

RF 5.3.1: Especificar el nombre del usuario al que se le asignarán los ejercicios, permitiendo también asignar ejercicios a un equipo de trabajo integrado por 2, 3 o 4 estudiantes.

RF 5.3.2: Mostrar en el campo “Disponibles” una lista con los títulos de los temas seleccionados y los números de los ejercicios disponibles por cada uno, permitiendo la selección de aquellos que se desean realizar.

RF 5.3.3: Especificar en el campo “Seleccionados” los números de los ejercicios asignados por tema a un usuario determinado.

RF 6: Mostrar orden de comienzo de ejercitación.

RF 6.1: En caso de haber más de un estudiante autenticado en el sistema, mostrar un mensaje informando el orden numérico en el que cada uno realizará los ejercicios.

RF 7: Reiniciar la configuración de selección de ejercicios, mostrando la opción secuencial marcada por defecto, el campo “desde” en 1 y en el campo “hasta” el total de ejercicios disponibles de los temas seleccionados.

RF 8: Realizar ejercicios.

RF 8.1: El sistema debe cargar una tipología de ejercicio.

RF 8.1.1: Cargar tipología completar por escritura (con control ortográfico opcional).

RF 8.1.2: Cargar tipología enlazar elementos de 2 conjuntos de naturaleza arbitraria.

RF 8.1.3: Cargar tipología seleccionar textos.

RF 8.1.4: Cargar tipología ordenar textos.

RF 8.1.5: Cargar tipología formar conjuntos.

RF 8.1.6: Cargar tipología dividir en sílabas.

RF 8.1.7: Cargar tipología acentuar palabras.

RF 8.1.8: Cargar tipología seleccionar palabras.

RF 8.2: Mostrar el número correspondiente al ejercicio que se está realizando y el número total de ejercicios seleccionados.

RF 8.3: Mostrar el número correspondiente al intento que se está realizando y el total de intentos permitidos para el ejercicio.

RF 8.4: Mostrar el tiempo que ha transcurrido durante la realización del ejercicio.

RF 8.5: Mostrar el número correspondiente al ejercicio visualizado según el tema al que pertenece.

RF 8.6: Mostrar el nombre del usuario que está realizando el ejercicio.

RF 9: Realizar todos los intentos permitidos en un ejercicio para dar la solución correcta.

RF 10: Revisar la respuesta de un ejercicio.

RF 10.1: La mascota debe informar a través de un mensaje de retroalimentación afectiva si el ejercicio es correcto, parcialmente correcto o incorrecto.

RF 10.2: En caso de que la respuesta no sea correcta, la mascota debe mostrar un mensaje con una pista o ayuda en función del intento; los niveles de ayuda serán la cantidad de intentos menos uno. Si es el último intento la mascota debe preguntar al usuario si desea realizar un análisis de la respuesta correcta.

RF 10.3: Si la respuesta es correcta la mascota debe preguntar al usuario si desea realizar un análisis de la respuesta.

RF 11: Una vez agotados todos los intentos para responder un ejercicio, el sistema debe permitir ver la respuesta correcta y la respuesta dada por el usuario en cada uno de los intentos.

RF 12: Mostrar estadísticas de los ejercicios.

RF 12.1: Mostrar la efectividad de las respuestas a los ejercicios.

RF 12.2: Mostrar la cantidad de ejercicios sin responder.

RF 12.3: Mostrar en una gráfica de barras la cantidad de ejercicios por cada categoría de calificación.

RF 13: Mostrar los resultados obtenidos en la ejercitación al concluir la misma.

RF 13.1: Mostrar el área Resultados.

RF 13.1.1: Mostrar Contenido: el título del tema al que pertenecen los ejercicios.

RF 13.1.2: Mostrar Categoría de calificación máxima: la cantidad de ejercicios por tema evaluados en esa categoría.

RF 13.1.3: Mostrar Categoría de calificación media: la cantidad de ejercicios por tema evaluados en esa categoría.

RF 13.1.4: Mostrar Categoría de calificación mínima: la cantidad de ejercicios por tema evaluados en esa categoría.

RF 13.1.5: Mostrar Total: representa el total de ejercicios realizados de cada tema.

RF 13.2: Mostrar el área Ejercicios.

RF 13.2.1: Mostrar Contenido: el título del tema al que pertenece el ejercicio.

RF 13.2.2: Mostrar Pregunta: el número de cada ejercicio realizado.

RF 13.2.3: Mostrar Tiempo: el tiempo consumido por el usuario en la solución del ejercicio.

RF 13.2.4: Mostrar Evaluación: la evaluación otorgada a cada ejercicio.

RF 13.2.5: Mostrar Intentos: la cantidad de intentos utilizados del total disponible para la solución del ejercicio.

RF 13.3: Mostrar el área Gráfico

RF 13.3.1: Mostrar un gráfico de pastel en el cual cada porción muestra el porcentaje que representa la cantidad de ejercicios evaluados con cada categoría de calificación respecto al total.

RF 14: Navegar entre todos los ejercicios seleccionados a través de las opciones anterior y siguiente.

RF 15: Registrar la traza de la interacción de los estudiantes con el módulo Ejercicios.

RF 15.1: Registrar los nombres de los temas de los que el estudiante realizó ejercicios.

RF 15.2: Registrar la cantidad de ejercicios realizados en cada tema.

RF 15.3: Registrar calificación, es decir la cantidad de ejercicios por temas evaluados con cada una de las categorías de evaluación.

RF 15.4: Registrar el porcentaje de efectividad obtenido por un estudiante en cada tema.

RF 15.5: Registrar el nivel de desempeño al que pertenece el ejercicio (I: Reproducción, II: Aplicación, III: Creación).

RF 15.6: Registrar el tiempo empleado por el estudiante en cada ejercicio.

RF 15.7: Registrar el número de intentos utilizados por el estudiante en la solución del ejercicio.

RF 15.8: Registrar los nombres de los estudiantes que integraron un equipo de trabajo determinado para realizar ejercicios.

RF 16: Gestionar ejercicio.

RF 16.1: Adicionar ejercicio.

RF 16.2: Modificar datos de un ejercicio.

RF 16.3: Ver datos de un ejercicio.

RF 16.4: Eliminar ejercicio.

RF 16.5: Gestionar medias de un ejercicio.

RF 16.5.1: Adicionar medias a un ejercicio.

RF 16.5.2: Cambiar medias de un ejercicio.

RF 16.5.3: Eliminar medias de un ejercicio.

RF 17: Validar los datos introducidos por el usuario.

RF 18: Informar al usuario el resultado de las operaciones.

2.3.2 Requisitos no funcionales

RNF 1: Requisitos de hardware

Uso local

- Procesador Pentium 500 MHz.
- 256 MB de RAM.
- 9 GB de espacio en disco duro.
- Soporte de video que admita resolución de al menos 1024 x 600 píxeles.

Uso en red

- Velocidad de conexión mínima: 10 Mbit/s.
- Procesador Pentium 500 MHz.
- 512 MB de RAM.
- 9 GB de espacio en disco duro.

RNF 2: Requisitos de software

- Computadora personal con navegador Mozilla Firefox versión 3.5 o cualquier otra versión compatible con este.
- Sistema operativo Canaima versión 2.1 y Ubuntu en su versión 10.0.4.

RNF 3: Requisitos de restricciones en el diseño y la implementación

- Lenguaje de programación del lado del cliente JavaScript
- Framework de JavaScript para el cliente JQuery 1.4
- Lenguaje de programación PHP 5.2
- Framework de PHP para el servidor Symfony 1.4.3

- El estilo arquitectónico a emplear será el Modelo Vista Controlador

RNF 4: Requisitos de apariencia e interfaz externa

- El diseño de las interfaces debe ser amigable y sencillo.
- El diseño de la interfaz gráfica deberá garantizar la distinción visual entre los elementos del sistema.

RNF 5: Requisitos legales

- Cada una de las medias (imágenes, videos, sonidos) que se utilicen en el módulo deben tener el permiso legal de sus autores y su aprobación para hacer uso de ellas.

RNF 6: Requisitos de seguridad

- El sistema deberá tener el control de la información a la que accede cada usuario.
- No se denegará el acceso a la aplicación a ningún usuario.

RNF 7: Requisitos de rendimiento:

- El sistema debe permitir el acceso a la información en tiempos relativamente cortos, dependiendo de la conexión.
- La aplicación debe ser concebida para el consumo mínimo de recursos.

RNF 8: Requisitos de usabilidad:

- El sistema debe ser de fácil manejo para los usuarios que tengan niveles básicos sobre computación.

2.4 Modelo de sistema

El modelo de sistema es la integración del diagrama de casos de uso del sistema y la especificación de los casos de uso y actores relacionados en el mismo.

El diagrama de casos de uso muestra la relación entre los actores y los casos de uso del sistema, permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los requisitos que debe cumplir. Los actores son las personas o sistemas externos que interactúan con el sistema y los casos de uso especifican una secuencia de acciones, incluyendo variantes, que el sistema puede llevar a cabo, y que producen un resultado observable de valor para un actor concreto. (17)

Actores del sistema

La siguiente tabla define los actores del sistema y su justificación.

Actores	Justificación
Usuario	Persona que interactúa con el módulo Ejercicios, puede ser estudiante, profesor o invitado.
Gestor de contenido	Persona que se encarga de gestionar los cuestionarios interactivos en la herramienta de administración.

Tabla 2.1. Actores del sistema

Diagrama de casos de uso del sistema

A continuación se muestran el diagrama de casos de uso correspondiente al módulo Ejercicios y a la herramienta de administración.

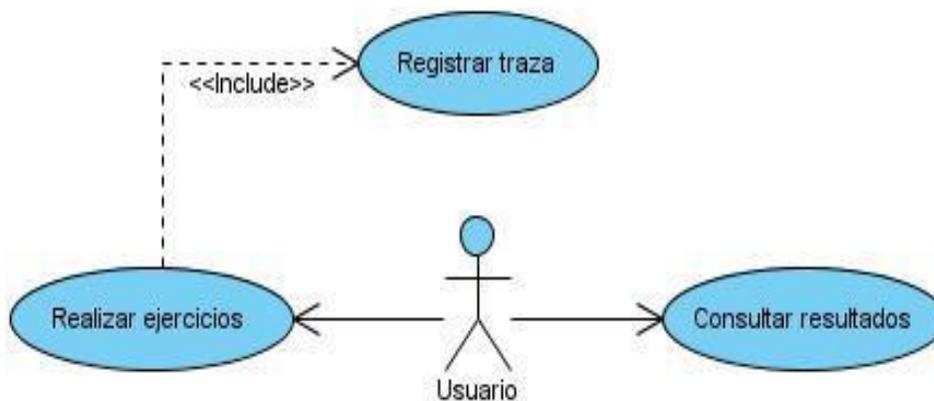


Figura 2.2. Diagrama de casos de uso del módulo Ejercicios.

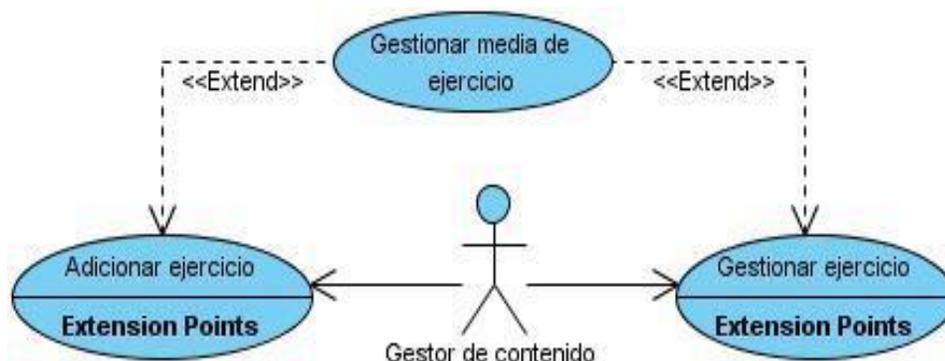


Figura 2.3. Diagrama de casos de uso de la herramienta de administración.

En el diagrama de casos de uso de la herramienta de administración se utilizó la alternativa parcial del patrón CRUD (Creating, Reading, Updating, Deleting en sus siglas en inglés) para separar la funcionalidad adicionar ejercicios del resto de las funcionalidades modificar, eliminar y ver, debido a que esta es más extensa y compleja que las demás. Además, se utiliza el patrón Concordancia específicamente Adición ya que el caso de uso Gestionar media de ejercicio es una subsecuencia común que extiende los casos de uso Adicionar ejercicio y Gestionar ejercicio, los cuales modelan el flujo que será expandido con la subsecuencia.

2.4.1 Descripción de los casos de uso del sistema

La descripción de un caso de uso detalla una secuencia de interacciones usuario-sistema, incluyendo cómo comienza y cómo termina. A continuación se describen los casos de uso Realizar ejercicio y Gestionar ejercicio seleccionados al azar, pues todos son significativos; las descripciones de los restantes casos de uso pueden ser consultadas en el Anexo 1.

Descripción textual del caso de uso “Realizar ejercicio”.

Caso de uso	Realizar ejercicio
Actor	Usuario
Propósito	Realizar ejercicios de los temas seleccionados.
Resumen	El caso de uso se inicia cuando el usuario accede al módulo Ejercicios. El sistema permite seleccionar los temas, el tipo de selección de ejercicios (Secuencial, Al azar o

	<p>Asignados) y la cantidad de ejercicios a realizar. El usuario selecciona los temas, el tipo de selección de ejercicios, la cantidad de ejercicios y la opción comenzar. El sistema muestra los ejercicios, el usuario realiza los mismos y selecciona la opción que permite revisar su respuesta; al concluir el último intento de cada ejercicio el sistema permite ver la respuesta correcta del ejercicio y la respuesta del usuario en todos los intentos. Además permite ver las estadísticas de la ejercitación en cualquier momento. El usuario selecciona una de estas opciones si así lo desea, el sistema muestra el resultado correspondiente a la opción seleccionada y finaliza así el caso de uso.</p>	
Referencias	<p>RF: 1, 2, 3, 4, 4.1, 4.1.1, 5, 5.1, 5.2, 5.3, 5.3.1, 5.3.2, 5.3.3, 6, 7, 8, 8.1, 8.1.1, 8.1.2, 8.1.3, 8.1.4, 8.1.5, 8.1.6, 8.1.7, 8.1.8, 8.2, 8.3, 8.4, 8.5, 8.6, 9, 10, 10.1, 10.2, 10.3, 11, 12, 2.1, 12.2, 12.3</p>	
Precondiciones	<p>Debe haberse mostrado el escritorio de trabajo del usuario autenticado.</p>	
Poscondiciones	<p>El usuario realizó ejercicios.</p>	
Flujo normal de eventos		
Acciones del actor	Respuesta del sistema	
1.El caso de uso se inicia cuando el usuario accede al módulo Ejercicios.	2. Muestra la pantalla principal del módulo Ejercicios.	
	<p>3. Muestra:</p> <ul style="list-style-type: none"> • Por niveles los temas con ejercicios disponibles. • El tipo de selección de ejercicios “Secuencial” de forma predeterminada. <p>Permite seleccionar uno o varios temas.</p>	
4. Selecciona uno o varios temas.	<p>5. Muestra en el campo desde del tipo de selección de ejercicios Secuencial el valor 1 y en el campo hasta la cantidad total de ejercicios disponibles en los temas seleccionados.</p> <p>Permite:</p> <ul style="list-style-type: none"> • Comenzar a visualizar ejercicios. 	

	<ul style="list-style-type: none"> • Seleccionar los valores desde y hasta. • Introducir los valores desde y hasta. • Seleccionar el tipo de selección de ejercicios “Al azar”. <i>Ver sección 1: “Al azar”.</i> • Seleccionar el tipo de selección de ejercicios “Asignados”. <i>Ver sección 2: “Asignados”.</i> • Reiniciar la configuración de selección de ejercicios.
6. Introduce el rango de ejercicios que desea realizar de forma secuencial, en el campo desde el ejercicio inicial y en el campo hasta el ejercicio final.	
7. Selecciona la opción Comenzar.	8. Registra la traza de la interacción del usuario con el módulo. <i>Ver CU Registrar traza.</i>
	9. Carga una tipología de ejercicio.
	<p>10. Muestra los siguientes datos:</p> <ul style="list-style-type: none"> • Enunciado. • Pregunta. • Intento. • Tiempo. • Número del ejercicio y el tema al que pertenece. • El nombre del usuario que está realizando el ejercicio. <p>Permite:</p> <ul style="list-style-type: none"> • Realizar el ejercicio. • Ver el ejercicio siguiente. • Ver el ejercicio anterior. • Revisar la respuesta. • Ver las estadísticas de la ejercitación. • Terminar la ejercitación.
11. Realiza el ejercicio visualizado.	

12. Selecciona la opción Revisa tu respuesta.	13. Registra la traza de la interacción del usuario con el sistema. <i>Ver CU Registrar traza.</i>
	14. Muestra un mensaje informando que la respuesta es incorrecta.
	15. Muestra un mensaje afectivo para alentar al usuario en la resolución del ejercicio. Reproduce en voz de la mascota el mensaje afectivo mostrado y permite interactuar con la mascota. <i>Ver CU Manipular mascota del módulo General.</i>
	16. Muestra un mensaje conclusivo sobre la respuesta del ejercicio.
	17. Permite ver la respuesta correcta del ejercicio realizado y la respuesta del usuario en cada uno de los intentos. Además permite terminar la ejercitación.
	18. Bloquea el permiso de edición sobre el ejercicio.
19. Selecciona la opción Terminar.	20. Registra la traza de la interacción del usuario con el módulo. <i>Ver CU Registrar traza.</i>
	21. Finaliza la ejercitación.
	22. Termina el caso de uso.
Flujo alternativo de los eventos.	
6. a El número introducido en el campo desde es mayor que la cantidad de ejercicios disponibles.	
	6. a.1 Señala el campo desde como incorrecto y muestra un tooltip con el siguiente mensaje: "Este valor está fuera del intervalo".
	6. a.2 Ejecuta el paso 6 del flujo básico.
6. b El campo desde está vacío.	
	6. b.1 Señala el campo desde como incorrecto y muestra un tooltip con el siguiente mensaje "Este valor está fuera del intervalo".

	6. b.2 Ejecuta el paso 6 del flujo básico.
6. c El número introducido en el campo hasta es mayor que la cantidad de ejercicios disponibles.	
	6. c.1 Señala el campo hasta como incorrecto y muestra un tooltip con el siguiente mensaje “Este valor está fuera del intervalo”.
	6. c.2 Ejecuta el paso 6 del flujo básico.
6. d El número introducido en el campo hasta es menor que el número introducido en el campo desde.	
	6. d.1 Señala el campo hasta como incorrecto y muestra un tooltip con el siguiente mensaje “Este valor está fuera del intervalo”.
	6. d.2 Ejecuta el paso 6 del flujo básico.
6. e Los valores introducidos en los campos desde y/o hasta no son números enteros.	
	6. e.1 Señala los campos hasta y/o desde como incorrectos y muestra un tooltip con el siguiente mensaje “Este valor está fuera del intervalo”.
	6. e.2 Ejecuta el punto 6 del flujo básico.
6. f No modifica los valores (desde - hasta) de la cantidad de ejercicios a realizar.	
	6. f.1 Ejecuta el punto 7 del flujo básico.
6. g Selecciona los valores del rango (desde – hasta) de ejercicios a visualizar.	
	6. g.1 Ejecuta el paso 7 del flujo básico.
7. a Selecciona la opción Configuración predeterminada.	
	7. a.1 Reinicia la configuración del módulo.
	7. a.2 Regresa al punto 3 del flujo básico.
8. a La cantidad de ejercicios seleccionada no es suficiente para la cantidad de estudiantes que iniciaron sesión.	
	8. a .1 Muestra el siguiente mensaje: “La cantidad de ejercicios seleccionada no es suficiente para todos los estudiantes autenticados en la sesión”.

	8. a.2 Ejecuta el punto 5 del flujo básico.
9. a Se autenticó más de un estudiante en el sistema.	
	9. a.1 Muestra una pantalla con los nombres de los estudiantes autenticados en el orden en el que cada uno realizará ejercicios. Permite Aceptar y Cancelar.
9. a.2 Selecciona la opción Aceptar.	
	9. a.3 Ejecuta el punto 9 del flujo básico.
11. a Selecciona la opción de ver el ejercicio siguiente.	
	11. a.1 Muestra secuencialmente el ejercicio siguiente.
	11. a.2 Registra la traza de la interacción del usuario con el módulo. <i>Ver CU Registrar traza.</i>
	11. a.3 Regresa al punto 9 del flujo básico.
11. b Selecciona la opción de ver el ejercicio siguiente y este es el último.	
	11. b.1 Muestra secuencialmente el ejercicio siguiente.
	11. b.2 Registra la traza de la interacción del usuario con el módulo. <i>Ver CU Registrar traza.</i>
	11. b.3 Muestra la opción siguiente deshabilitada.
	11. a.4 Regresa al punto 9 del flujo básico.
11. c Selecciona la opción de ver el ejercicio anterior y este no ha sido resuelto.	
	11. c.1 Muestra secuencialmente el ejercicio anterior.
	11. c.2 Registra la traza de la interacción del usuario con el módulo. <i>Ver CU Registrar traza.</i>
	11. c.3 Regresa al punto 9 del flujo básico.
11. d Selecciona la opción de ver el ejercicio anterior y este es el primero.	
	11. d.1 Muestra de forma secuencial el ejercicio anterior correspondiente al total seleccionado.
	11. d.2 Registra la traza de la interacción del usuario con el módulo. <i>Ver CU Registrar traza.</i>

	11. d.3 Muestra la opción anterior deshabilitada.
	11. d.3 Regresa al punto 9 del flujo básico.
11. e Selecciona la opción de ver el ejercicio anterior y este ya fue resuelto.	
	11. e.1 Muestra el ejercicio anterior con las respuestas dadas por el usuario.
	11. e.2 Regresa al punto 9 del flujo básico.
11. f Selecciona la opción de ver el ejercicio siguiente y este ya fue resuelto.	
	11. f.1 Muestra el ejercicio siguiente con las respuestas dadas por el usuario.
	11. f.2 Regresa al punto 9 del flujo básico.
11. g Selecciona la opción de ver las estadísticas de la ejercitación.	
	11. g.1 Muestra: <ul style="list-style-type: none"> • El valor en por ciento de la efectividad obtenida en la realización de los ejercicios. • La cantidad de ejercicios sin responder. • Una gráfica de barras con la cantidad de ejercicios por cada categoría de calificación.
	11. g.2 Ejecuta el punto 11 del flujo básico.
11. h El enunciado del ejercicio tiene palabras clave.	
	11. h.1 Permite ver las palabras clave. <i>Ver CU Consultar palabra clave del módulo Temas.</i>
	11. h.2 Ejecuta el punto 11 del flujo básico.
14. a La respuesta dada al ejercicio es correcta.	
	14. a.1 Muestra un mensaje informando que la respuesta es correcta.
	14. a.2 Muestra un mensaje afectivo felicitando al usuario por su respuesta.
	14. a.3 Reproduce en voz de la mascota el mensaje afectivo mostrado.

	Permite interactuar con la mascota. <i>Ver CU Manipular mascota del módulo General.</i>
	14. a.4 Ejecuta el punto 16 del flujo básico.
13. a Quedan intentos disponibles para realizar el ejercicio.	
	13. a.1 Muestra un mensaje informando que la respuesta es incorrecta.
	13. a.2 Muestra un mensaje afectivo para alentar al usuario en la resolución del ejercicio. Reproduce en voz de la mascota el mensaje afectivo mostrado y permite interactuar con la mascota. <i>Ver CU Manipular mascota del módulo General.</i>
	13. a.3 Muestra un mensaje con una pista de ayuda. Permite realizar nuevamente el ejercicio.
	13. a.2 Regresa al punto 11 del flujo básico.
19. a Selecciona la opción Respuesta correcta.	
	19. a.1 Muestra la pantalla Comparación de resultados, en la columna izquierda se muestra la respuesta correcta del ejercicio y en la columna derecha la respuesta del usuario en cada uno de los intentos. Permite seleccionar la opción Aceptar para salir de la pantalla.
19. a.2 Selecciona la opción Aceptar.	
	19. a.3 Regresa a la vista anterior.
	19. a.4 Ejecuta el paso 19 del flujo básico.
Sección 1: “Al azar” Flujo normal de eventos.	
Acciones del actor	Respuesta del sistema
1. El usuario selecciona la opción Al azar.	2. Muestra la cantidad total de ejercicios disponibles de los temas seleccionados, en el campo cantidad. Permite: <ul style="list-style-type: none"> • Introducir el número de ejercicios que desea visualizar.

	<ul style="list-style-type: none"> • Seleccionar la cantidad de ejercicios a visualizar. • Comenzar la visualización de ejercicios.
3. Introduce la cantidad de ejercicios.	
4. Selecciona la opción Comenzar.	5. Registra la traza de la interacción del usuario con el módulo. <i>Ver CU Registrar traza.</i>
	6. Carga una tipología de ejercicio.
	<p>7. Muestra los siguientes datos:</p> <ul style="list-style-type: none"> • Enunciado. • Pregunta • Intento. • Tiempo. • Número del ejercicio y el tema al que pertenece. • El nombre del usuario que está realizando el ejercicio. <p>Permite:</p> <ul style="list-style-type: none"> • Realizar el ejercicio. • Ver el ejercicio siguiente. • Ver el ejercicio anterior. • Revisar la respuesta dada. • Ver las estadísticas de la ejercitación. • Terminar la ejercitación.
8. Realiza el ejercicio visualizado.	
9. Selecciona la opción Revisa tu respuesta.	10. Registra la traza de la interacción del usuario con el sistema. <i>Ver CU Registrar traza.</i>
	11. Muestra un mensaje informando que la respuesta es incorrecta.
	<p>12. Muestra un mensaje afectivo para alentar al usuario en la resolución del ejercicio.</p> <p>Reproduce en voz de la mascota el mensaje afectivo mostrado.</p>

	Permite interactuar con la mascota. Ver <i>CU Manipular mascota del módulo General</i> .
	13. Muestra un mensaje conclusivo sobre la respuesta del ejercicio.
	14. Permite ver la respuesta correcta del ejercicio realizado y la respuesta del usuario en cada uno de los intentos.
	15. Bloquea el permiso de edición sobre el ejercicio.
16. Selecciona la opción Terminar.	17. Registra la traza de la interacción del usuario con el módulo. Ver <i>CU Registrar traza</i> .
	18. Finaliza la ejercitación.
	19. Termina el caso de uso.
Flujo alternativo de los eventos.	
3. a El número introducido en el campo “cantidad” es mayor que la cantidad de ejercicios disponibles.	
	3. a.1 Señala el campo cantidad como incorrecto y muestra un tooltip con el siguiente mensaje “Este valor está fuera del intervalo”.
	3. a.2 Ejecuta el paso 8 del flujo básico.
3. b El campo cantidad está vacío.	
	3. b.1 Señala el campo cantidad como incorrecto y muestra un tooltip con el siguiente mensaje “Este valor está fuera del intervalo”.
	3. b.2 Ejecuta el paso 8 del flujo básico.
3. c Selecciona una cantidad específica del total mostrado.	
	3. c.1 Ejecuta el paso 9 del flujo básico.
5. a La cantidad de ejercicios seleccionada no es suficiente para la cantidad de estudiantes que iniciaron sesión.	
	5. a .1 Muestra el siguiente mensaje: “La cantidad de ejercicios seleccionada no es suficiente para todos los estudiantes

	autenticados en la sesión”.
	5. a.2 Ejecuta el punto 8 del flujo básico.
6. a Se autenticaron más de un estudiante en el sistema.	
	6. a.1 Muestra un mensaje informando el orden numérico en el que cada estudiante realizará ejercicios. Permite Aceptar y Cancelar.
6. a.2 Selecciona la opción Aceptar.	
	6. a.2 Ejecuta el punto 11 del flujo básico.
8. a Selecciona la opción de ver el ejercicio siguiente y este no ha sido resuelto.	
	8. a.1 Muestra de forma aleatoria el ejercicio siguiente correspondiente al total seleccionado.
	8. a.2 Registra la traza de la interacción del usuario con el módulo. <i>Ver CU Registrar traza.</i>
	8. a.3 Regresa al punto 11 del flujo básico.
8. b Selecciona la opción de ver el ejercicio siguiente y este es el último.	
	8. b.1 Muestra de forma aleatoria el ejercicio siguiente correspondiente al total seleccionado.
	8. b.2 Registra la traza de la interacción del usuario con el módulo. <i>Ver CU Registrar traza.</i>
	8. b.3 Muestra la opción siguiente deshabilitada.
	8. b.4 Regresa al punto 11 del flujo básico.
8. c Selecciona la opción de ver el ejercicio anterior y este no ha sido resuelto.	
	8. c.1 Muestra el ejercicio anterior visualizado por el usuario.
	8. c.2 Registra la traza de la interacción del usuario con el módulo. <i>Ver CU Registrar traza.</i>
	8. c.3 Regresa al punto 11 del flujo básico.
8. c Selecciona la opción de ver el ejercicio anterior y este es el primero.	
	8. c.1 Muestra el ejercicio anterior visualizado por el usuario.
	8. c.2 Registra la traza de la interacción del usuario con el

	módulo. <i>Ver CU Registrar traza.</i>
	8. c.3 Muestra la opción anterior deshabilitada.
	8. c.2 Regresa al punto 11 del flujo básico.
8. d Selecciona la opción de ver el ejercicio anterior y este ya fue resuelto.	
	8. d.1 Muestra el ejercicio anterior con las respuestas dadas por el usuario.
	8. d.2 Regresa al punto 11 del flujo básico.
8. e Selecciona la opción de ver el ejercicio siguiente y este ya fue resuelto.	
	8. e.1 Muestra el ejercicio anterior con las respuestas dadas por el usuario.
	8. e.2 Regresa al punto 11 del flujo básico.
8. f Selecciona la opción de ver las estadísticas de la ejercitación.	
	8. f.1 Muestra: <ul style="list-style-type: none"> • El valor en por ciento de la efectividad obtenida en la realización de los ejercicios. • La cantidad de ejercicios que faltan por responder. • En una gráfica de barras la cantidad de ejercicios por cada categoría de calificación.
	8. f.2 Regresa al punto 7 del flujo básico.
8. g El enunciado del ejercicio tiene palabras claves.	
	8. g.1 Permite ver las palabras clave. <i>Ver CU Consultar palabra clave del módulo Temas.</i>
	8. g.2 Ejecuta el punto 13 del flujo básico.
10. a Quedan intentos disponibles para realizar el ejercicio.	
	10. a.1 Muestra una mensaje informando que la respuesta es incorrecta.
	10. a.2 Muestra un mensaje afectivo para alentar al usuario en la resolución del ejercicio. Reproduce en voz de la mascota el mensaje afectivo mostrado

	y permite interactuar con la mascota. <i>Ver CU Manipular mascota del módulo General.</i>
	10. a.3 Muestra un mensaje con una pista de ayuda. Permite realizar nuevamente el ejercicio.
	10. a.2 Regresa al punto 13 del flujo básico.
11. h La respuesta dada al ejercicio es correcta.	
	11. h.1 Muestra un mensaje informando que la respuesta es correcta.
	11. h.2 Muestra un mensaje afectivo felicitando al usuario por su respuesta.
	11. h.3 Reproduce en voz de la mascota el mensaje afectivo mostrado. Permite interactuar con la mascota. <i>Ver CU Manipular mascota del módulo General.</i>
	11. a.4 Ejecuta el punto 18 del flujo básico de la sección.
16. a Selecciona la opción Respuesta correcta.	
	16. a.1 Muestra la pantalla Comparación de resultados, en la columna izquierda se muestra la respuesta correcta del ejercicio y en la columna derecha la respuesta del usuario en cada uno de los intentos. Permite seleccionar la opción Aceptar para salir de la pantalla.
16. a.2 Selecciona la opción Aceptar.	
	16. a.3 Regresa a la vista anterior.
	16 .a.4 Ejecuta el punto 21 del flujo básico.
Sección 2: “Asignados” Flujo normal de eventos.	
Acciones del actor	Respuesta del sistema
1. El usuario selecciona la opción Asignados.	2. Permite seleccionar la forma de asignar los ejercicios: Individual y En equipo.
3. Selecciona la opción En equipo.	4. Muestra el número de cada tema seleccionado y los

	<p>números de los ejercicios disponibles por tema.</p> <p>Permite seleccionar de cada tema el ejercicio o los ejercicios que serán asignados.</p>
5. Selecciona los ejercicios.	6. Permite comenzar la ejercitación.
7. Selecciona la opción Comenzar.	8. Registra la traza del módulo. <i>Ver CU Registrar traza.</i>
	<p>9. Muestra una pantalla con los nombres de los estudiantes autenticados en el orden en el que cada uno realizará ejercicios.</p> <p>Permite Aceptar.</p>
10. Selecciona la opción Aceptar.	11. Carga una tipología de ejercicio.
	<p>12. Muestra los siguientes datos:</p> <ul style="list-style-type: none"> • Enunciado. • Pregunta. • Intento. • Tiempo. • Número del ejercicio y el tema al que pertenece. • El nombre del usuario que está realizando el ejercicio. <p>Permite:</p> <ul style="list-style-type: none"> • Realizar el ejercicio. • Ver el ejercicio siguiente. • Ver el ejercicio anterior. • Revisar la respuesta dada. • Ver las estadísticas de la ejercitación. • Terminar la ejercitación.
13. Realiza el ejercicio visualizado.	
14. Selecciona la opción Revisa tu respuesta.	
	15. Registra la traza del módulo. <i>Ver CU Registrar traza.</i>
	16. Muestra un mensaje informando que la respuesta es

	incorrecta.
	17. Muestra un mensaje afectivo para alentar al usuario en la resolución del ejercicio. Reproduce en voz de la mascota el mensaje afectivo mostrado. Permite interactuar con la mascota. <i>Ver CU Manipular mascota del módulo General.</i>
	18. Muestra un mensaje conclusivo sobre la respuesta del ejercicio.
	19. Permite ver la respuesta correcta del ejercicio realizado y la respuesta del usuario en cada uno de los intentos.
	20. Bloquea el permiso de edición sobre el ejercicio.
21. Selecciona la opción Terminar.	22. Registra la traza del módulo. <i>Ver CU Registrar traza.</i>
	23. Finaliza la ejercitación.
	24. Termina el caso de uso.
Flujo alternativo de los eventos.	
3. a Selecciona la opción Individual.	
	3. a.1 Permite seleccionar el nombre del usuario al que se asignarán los ejercicios.
3. a.2 Selecciona el nombre del usuario al que se asignarán ejercicios.	
	3. a.3 Ejecuta el punto 9 del flujo básico.
9. a Solo hay un estudiante autenticado en el sistema.	
	4. a.1 Muestra el siguiente mensaje: “No es posible crear un equipo de trabajo con un solo estudiante”.
	4. a.2 Regresa al punto 8 del flujo básico.
8. a La cantidad de ejercicios seleccionada es menor que la cantidad de estudiantes que forman el equipo de trabajo.	
	8. a.1 Muestra el siguiente mensaje “La cantidad de ejercicios

	seleccionada no es suficiente para todos los estudiantes del equipo de trabajo”.
	8. a.2 Regresa al punto 10 del flujo básico.
13. a Selecciona la opción de ver el ejercicio siguiente y este no ha sido resuelto.	
	13 a.1 Muestra el ejercicio siguiente correspondiente al total seleccionado.
	13. a.2 Registra la traza del módulo. <i>Ver CU Registrar traza.</i>
13. b Selecciona la opción de ver el ejercicio siguiente y este es el último.	
	13 b.1 Muestra el ejercicio siguiente correspondiente al total seleccionado.
	13. b.2 Registra la traza del módulo. <i>Ver CU Registrar traza.</i>
	13. b.3 Muestra la opción siguiente deshabilitada.
	13. b.4 Regresa al punto 16 del flujo básico.
13. c Selecciona la opción de ver el ejercicio anterior y este no ha sido resuelto.	
	13. c.1 Muestra el ejercicio anterior visualizado por el usuario.
	13. c.2 Registra la traza del módulo Ejercicios. <i>Ver CU Registrar traza.</i>
13. d Selecciona la opción de ver el ejercicio anterior y este es el primero.	
	13. d.1 Muestra el ejercicio anterior visualizado por el usuario.
	13. d.2 Registra la traza del módulo Ejercicios. <i>Ver CU Registrar traza.</i>
	13. d.3 Muestra la opción anterior deshabilitada.
	13. d.4 Regresa al punto 16 del flujo básico.
13. e Selecciona la opción de ver el ejercicio anterior y este ya fue resuelto.	
	13. e.1 Muestra el ejercicio anterior con las respuestas dadas por el usuario.
	13. e.2 Regresa al punto 16 del flujo básico.
13. f Selecciona la opción de ver el ejercicio siguiente y este ya fue resuelto.	
	13. f.1 Muestra el ejercicio siguiente con las respuestas dadas

	por el usuario.
	13. f.2 Regresa al punto 16 del flujo básico.
13. g Selecciona la opción de ver las estadísticas de la ejercitación.	
	13. g.1 Muestra: <ul style="list-style-type: none"> • El valor en por ciento de la efectividad obtenida en la realización de los ejercicios. • La cantidad de ejercicios que faltan por responder. • En una gráfica de barras la cantidad de ejercicios por cada categoría de calificación.
	13. g.2 Ejecuta el punto 18 del flujo básico.
13. h El enunciado del ejercicio tiene palabras clave.	
	13. h.1 Permite ver las palabras clave. <i>Ver CU Consultar palabra clave del módulo Temas.</i>
	13. h.2 Ejecuta el punto 18 del flujo básico.
15. a Quedan intentos disponibles para realizar el ejercicio.	
	15. a.1 Muestra un mensaje informando que la respuesta es incorrecta.
	15. a.2 Muestra un mensaje afectivo para alentar al usuario en la resolución del ejercicio. Reproduce en voz de la mascota el mensaje afectivo mostrado y permite interactuar con la mascota. <i>Ver CU Manipular mascota del módulo General.</i>
	15. a.3 Muestra un mensaje con una pista de ayuda. Permite realizar nuevamente el ejercicio.
	15. a.2 Regresa al punto 18 del flujo básico.
16. a La respuesta dada al ejercicio es correcta.	
	16. a.1 Muestra un mensaje informando que la respuesta es correcta.
	16. a.2 Muestra un mensaje afectivo felicitando al usuario por

	su respuesta.
	16. a.3 Reproduce en voz de la mascota el mensaje afectivo mostrado. Permite interactuar con la mascota. <i>Ver CU Manipular mascota del módulo General.</i>
	16. a.9 Ejecuta el punto 24 del flujo básico de la sección.
21. a Selecciona la opción Respuesta correcta.	
	21. a.1 Muestra la pantalla Comparación de resultados, en la columna izquierda se muestra la respuesta correcta del ejercicio y en la columna derecha la respuesta del usuario en cada uno de los intentos. Permite seleccionar la opción Aceptar para salir de la pantalla.
21. a.2 Selecciona la opción Aceptar.	
	21. a.3 Regresa a la vista anterior.
	21 .a.4 Ejecuta el punto 26 del flujo básico

Tabla 2.3. Descripción textual del caso de uso “Realizar ejercicios”.

Descripción textual del caso de uso “Gestionar Ejercicio”.

Caso de uso	Gestionar ejercicio
Actores	Gestor de contenido
Propósito	Ver, modificar o eliminar ejercicios.
Resumen	El caso de uso se inicia cuando el actor selecciona la opción de ver, modificar o eliminar un ejercicio. Si selecciona la opción de ver un ejercicio el sistema mostrará los datos del ejercicio en cuestión. Si el actor selecciona la opción de modificar un ejercicio, el sistema mostrará los datos del ejercicio que se pueden modificar, una vez realizados los cambios, guardará las modificaciones. Si selecciona la opción de eliminar un ejercicio el sistema eliminará el ejercicio en cuestión, finalizando así el caso de uso.

Referencias	RF: 16, 16.1, 16.2, 16.3, 16.4, 17, 18
Precondiciones	Deben haberse listado los ejercicios disponibles en el software.
Poscondiciones	Se vio, modificó o eliminó un ejercicio.
Flujo normal de eventos.	
Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona la opción de ver, modificar o eliminar un ejercicio.	2. Permite: <ul style="list-style-type: none"> • Ver los datos de un ejercicio. • Modificar los datos de un ejercicio. Ver sección 1: "Modificar datos de un ejercicio". • Eliminar un ejercicio. Ver sección 2: "Eliminar un ejercicio".
3. Selecciona la opción que permite ver los datos de un ejercicio.	4. Muestra los datos del Ejercicio. <ul style="list-style-type: none"> • Número del ejercicio. • Enunciado. • Complejidad. • Visibilidad. • Cantidad de intentos. • Opciones de respuesta. • Retroalimentación. Permite salir de la vista actual mediante la opción Aceptar.
5. Selecciona la opción Aceptar.	6. Muestra la vista anterior.
	7. El caso de uso termina.
Sección 1: "Modificar datos de un ejercicio". Flujo normal de eventos.	
Acciones del actor	Respuesta del sistema
1. El usuario selecciona la opción que permite modificar los datos de un ejercicio.	2. Muestra los datos del ejercicio y brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos o seleccionando diferentes: <ul style="list-style-type: none"> • Número. • Descripción.

	<ul style="list-style-type: none"> • Nombre del autor. • Enunciado. • Complejidad. • Visibilidad. • Cantidad de intentos. • Módulo al que pertenece. • Tipología. • Formato de la tipología. • Nivel de ayuda. <p>Permite además:</p> <ul style="list-style-type: none"> • Guardar los datos. • Cancelar.
3. Modifica los datos que desee.	
4. Selecciona la opción de guardar los datos.	5. Valida los datos.
	6. Actualiza los datos del ejercicio.
	7. Muestra un mensaje de información “Se han actualizado los datos del ejercicio.”
	8. El caso de uso termina.
Flujo alterno de los eventos.	
3. a Selecciona la opción de cambiar una media del ejercicio.	
	3. a.1 Permite cambiar la media. <i>Ver caso de uso “Gestionar medias de un ejercicio”.</i>
	3. a.2 Ejecuta el punto 5 del flujo básico.
3. b Selecciona la opción de eliminar una media del ejercicio.	
	3. a.1 Permite eliminar la media. <i>Ver caso de uso “Gestionar medias de un ejercicio”.</i>
	3. a.2 Ejecuta el punto 5 del flujo básico.

4.a El actor selecciona la opción de Cancelar.	
	4. a.1 Regresa a la vista anterior.
	4. a.2 El caso de uso termina.
5. a Existen datos incompletos.	
	5. a.1 Muestra el mensaje de información “Existen campos vacíos que son obligatorios, por favor, complete estos datos”.
	5. a.2 Muestra un indicador sobre los campos vacíos.
	5. a.3 Regresa al paso 3 del flujo básico.
Sección 2: “Eliminar un ejercicio”. Flujo normal de eventos.	
Acciones del actor	Respuesta del sistema
1. El usuario selecciona la opción que permite eliminar un ejercicio.	2. Muestra el mensaje de información “Se eliminará el ejercicio seleccionado. ¿Desea continuar?”. Permite: <ul style="list-style-type: none"> • Aceptar • Cancelar
3. Selecciona la opción de Aceptar.	4. Elimina el ejercicio.
	5. Muestra el mensaje de información “El ejercicio seleccionado ha sido eliminado”.
	6. Regresa a la vista anterior actualizando los datos.
	7. El caso de uso termina.
Flujo alternativo de los eventos.	
3.a El actor selecciona la opción de Cancelar.	
	3. a.1 Regresa a la vista anterior.
	3. a.2 El caso de uso termina.

Tabla 2.4. Descripción textual del caso de uso “Gestionar ejercicio”.

2.5 Conclusiones

Con el desarrollo del presente capítulo se logró especificar las características del sistema a desarrollar mediante la realización del modelo de dominio, la identificación de los requisitos funcionales y no funcionales que debe cumplir el mismo, la elaboración del diagrama de casos de uso del sistema y la descripción textual de cada uno de estos.

Capítulo 3

ANÁLISIS Y DISEÑO DEL SISTEMA

Las actividades que propone el flujo de trabajo Análisis y Diseño, constituyen la base para alcanzar un producto final exitoso, pues su objetivo es traducir los requisitos a una especificación que describe cómo implementar el sistema, obteniéndose en el análisis una visión abstracta de lo que debe hacer el mismo, enfocada en los requisitos funcionales. El diseño es un refinamiento del análisis que tiene en cuenta también los requisitos no funcionales para representar en un lenguaje técnico la estructura de clases de la aplicación y las relaciones entre ellas.

En el presente capítulo se modela la solución propuesta mediante los diagramas de clases del análisis que permiten estructurar el sistema, obtener una vista general y comprender cómo deberá diseñarse el mismo; además los diagramas de clases del diseño con estereotipos web, los cuales muestran la interacción entre las diferentes clases de la aplicación y son un punto de partida para la implementación.

3.1 Modelo de análisis

El modelo de análisis está estructurado por clases y paquetes estereotipados, esboza cómo llevar a cabo la funcionalidad dentro del sistema, incluida aquella significativa para la arquitectura; sirve como una primera aproximación del diseño. (17)

Este modelo ayuda a entender los requisitos funcionales, pues representa una vista interna del sistema descrita en un lenguaje más técnico; estructura el funcionamiento interno del sistema mediante las clases del análisis y las asociaciones entre las mismas; las clases se centran en los requisitos funcionales y se identifican con los siguientes estereotipos:



Clase Interfaz

Modelan la interacción entre el sistema y sus actores.


Clase Control

Controlan la realización de uno o unos pocos casos de uso, coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.


Clase Entidad

Modelan información manejada en el caso de uso que posee larga vida.

A continuación se muestran los diagramas de clases del análisis de los casos de uso del sistema.

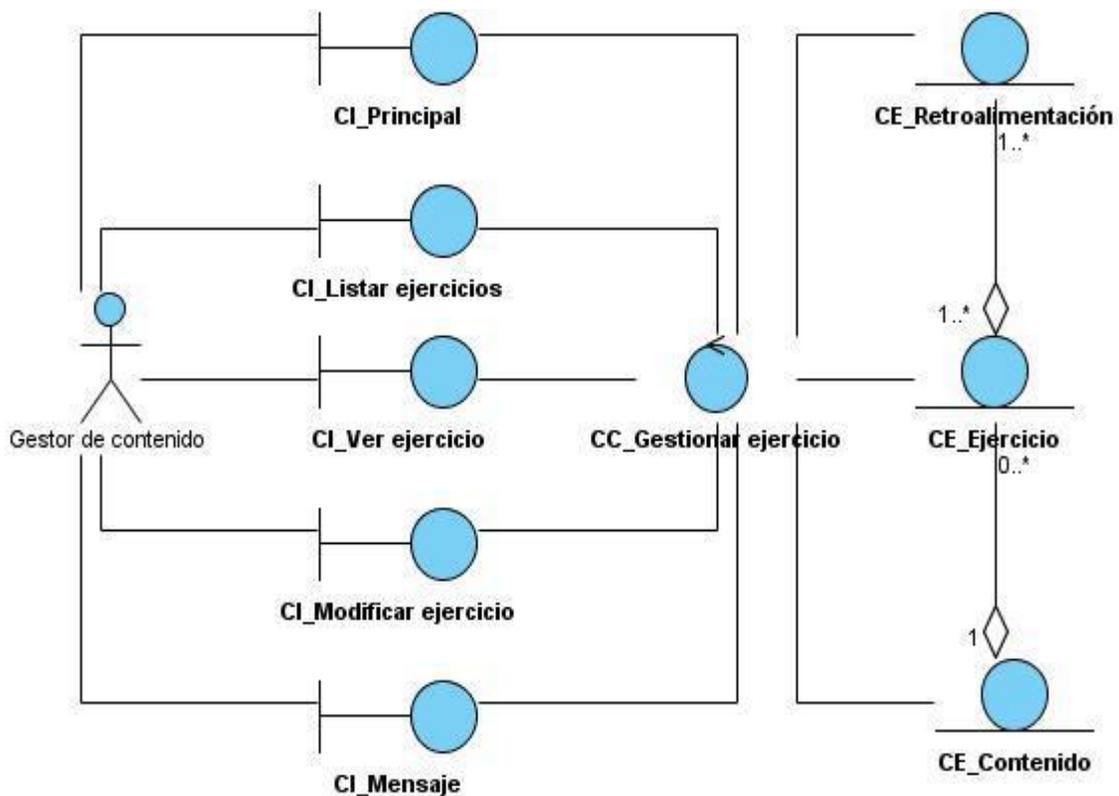


Figura 3.1. Diagrama de clases del análisis del caso de uso “Gestionar ejercicio”.

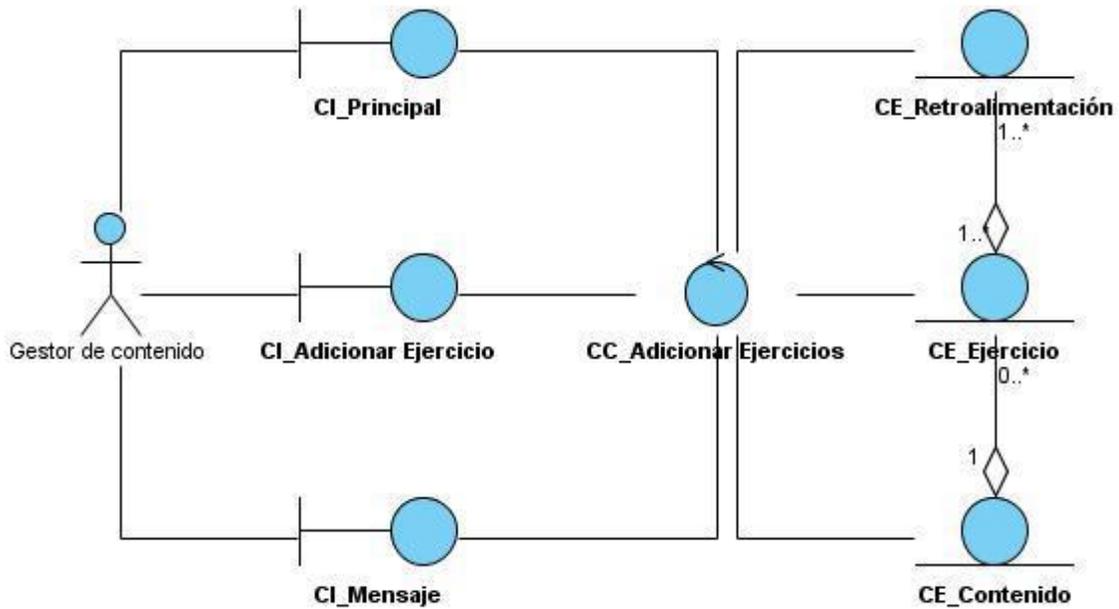


Figura 3.2. Diagrama de clases del análisis del caso de uso “Adicionar ejercicio”.

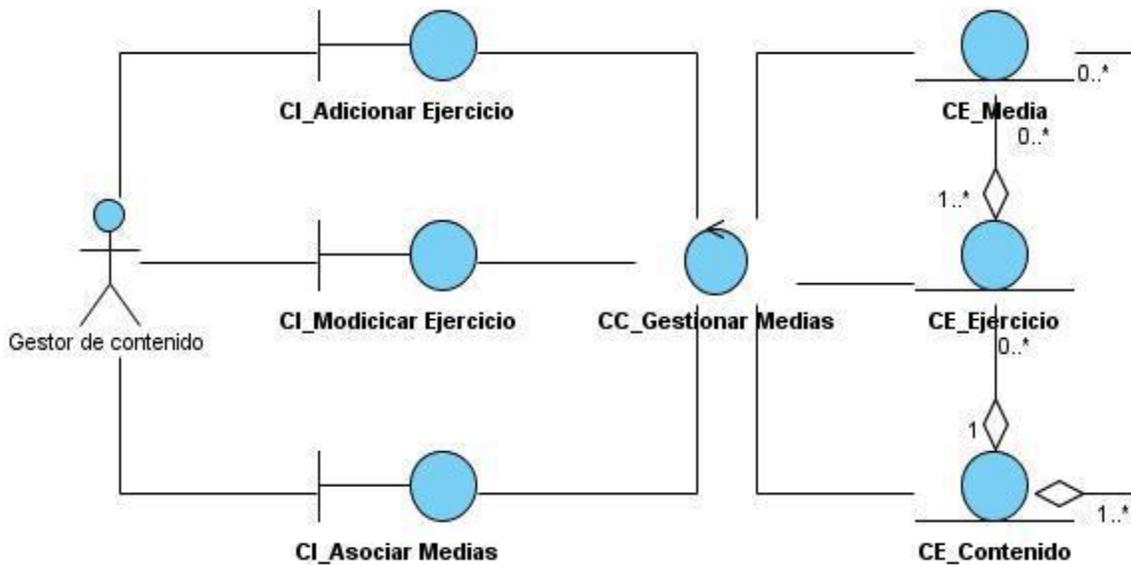


Figura 3.3. Diagrama de clases del análisis del caso de uso “Gestionar media de ejercicio”.

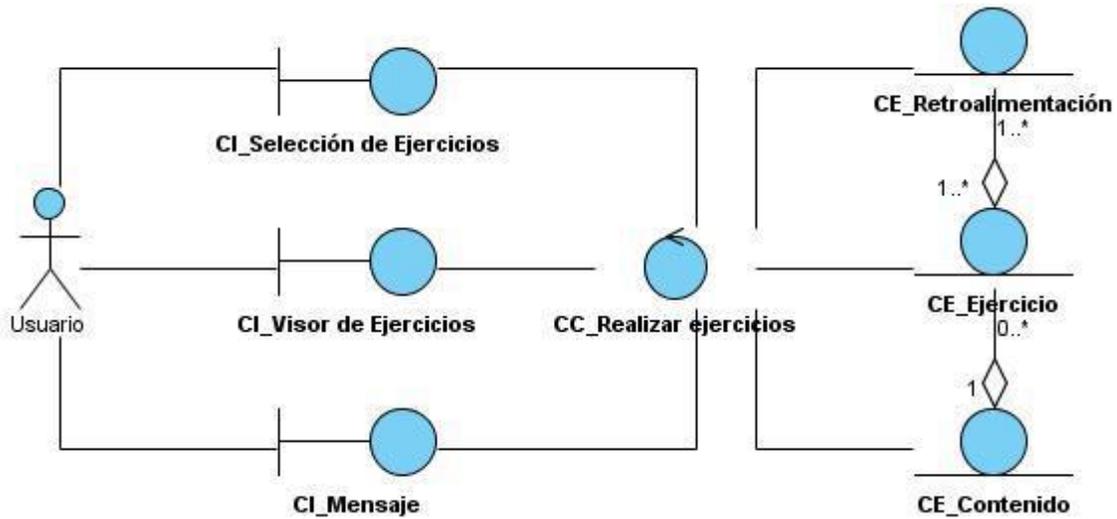


Figura 3.4. Diagrama de clases del análisis del caso de uso “Realizar ejercicio”.

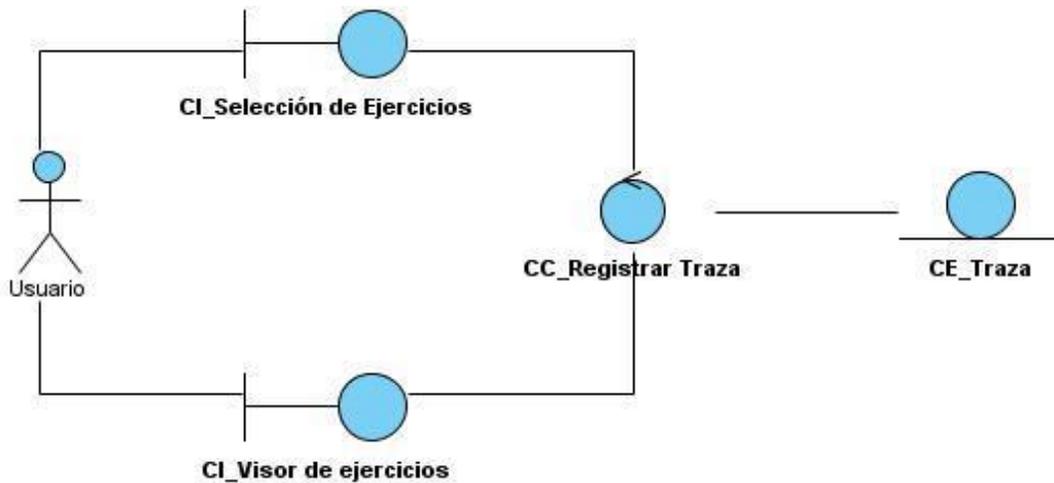


Figura 3.5. Diagrama de clases del análisis del caso de uso “Registrar traza”.

3.2 Modelo de diseño

“El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación. Además, el modelo de diseño sirve de abstracción de la implementación del sistema y es, de ese modo, utilizado como una entrada fundamental de las actividades de implementación”. (17)

El modelo de diseño tiene como propósito transformar los requisitos tanto funcionales como no funcionales en un diseño de clases que describa las relaciones entre estas, en un lenguaje técnico lo más cercano posible al lenguaje de programación. Una entrada esencial en el diseño es el resultado del análisis, o sea, el modelo de análisis.

El módulo Ejercicios cuenta con dos aplicaciones, la herramienta encargada de gestionar los contenidos educativos de los cuestionarios interactivos (backend) y la aplicación que visualiza los cuestionarios y permite la interacción con los mismos (frontend).

Cada una de estas aplicaciones cuenta con un único punto de entrada que es el controlador frontal, el cual recibe todas las peticiones y es el encargado de delegar a las acciones las tareas a ejecutar. Las acciones consultan el modelo de datos, llevan a cabo la lógica de negocio y preparan los datos a mostrar en las vistas correspondientes a la acción. En el backend las páginas servidoras construyen las páginas clientes enviadas a los usuarios y en el frontend la vista se construye en las páginas clientes mediante peticiones asíncronas al servidor. Las páginas clientes envían al controlador frontal la acción a ejecutar. Para el acceso a los datos es utilizado el ORM (de sus siglas en inglés “object-relational mapping”) Doctrine, el cual brinda un conjunto de clases mapeadas que permiten la interacción con la base de datos a través de objetos.

3.2.1 Patrón de diseño a utilizar

“Un patrón de diseño describe una estructura de diseño que resuelve un problema de diseño particular dentro de un contexto específico”. (31).

El patrón Mediador define un objeto que encapsula cómo interactúan un conjunto de objetos. El mediador estimula la pérdida de acoplamiento ocultando las referencias explícitas entre los objetos, permitiendo variar su interacción de forma independientemente. (32)

El mediador es un objeto que encapsula la interacción entre un conjunto de objetos, promoviendo el bajo acoplamiento, un objeto no tiene que conocer en detalles cómo interactuar con otros objetos, sino que lo hace a través del mediador.

Este patrón es utilizado en el lado del cliente de la aplicación frontend, la cual permite a los usuarios realizar cuestionarios interactivos de diferentes tipologías. Para cada una de estas se crea una clase JavaScript; además son utilizados widgets de jQuery que facilitan el desarrollo de las funcionalidades y proveen componentes visuales. El uso del patrón se puede observar en las relaciones de las clases JavaScript, donde cada clase es dirigida por el objeto ejmediador, el cual es encargado de la comunicación de las demás clases con el propósito de que exista un bajo acoplamiento, es decir, la mínima dependencia posible entre los objetos, además facilitar la implementación y mantenimiento del sistema.

3.2.2 Diagrama de paquetes

Un paquete de diseño agrupa clases relacionadas, realizaciones de casos de uso y otros paquetes, es usado para estructurar el modelo de diseño mediante su división en partes más pequeñas. El diagrama de paquete es una herramienta organizacional de los elementos del modelo de diseño. A continuación se muestra el diagrama de paquetes del sistema.

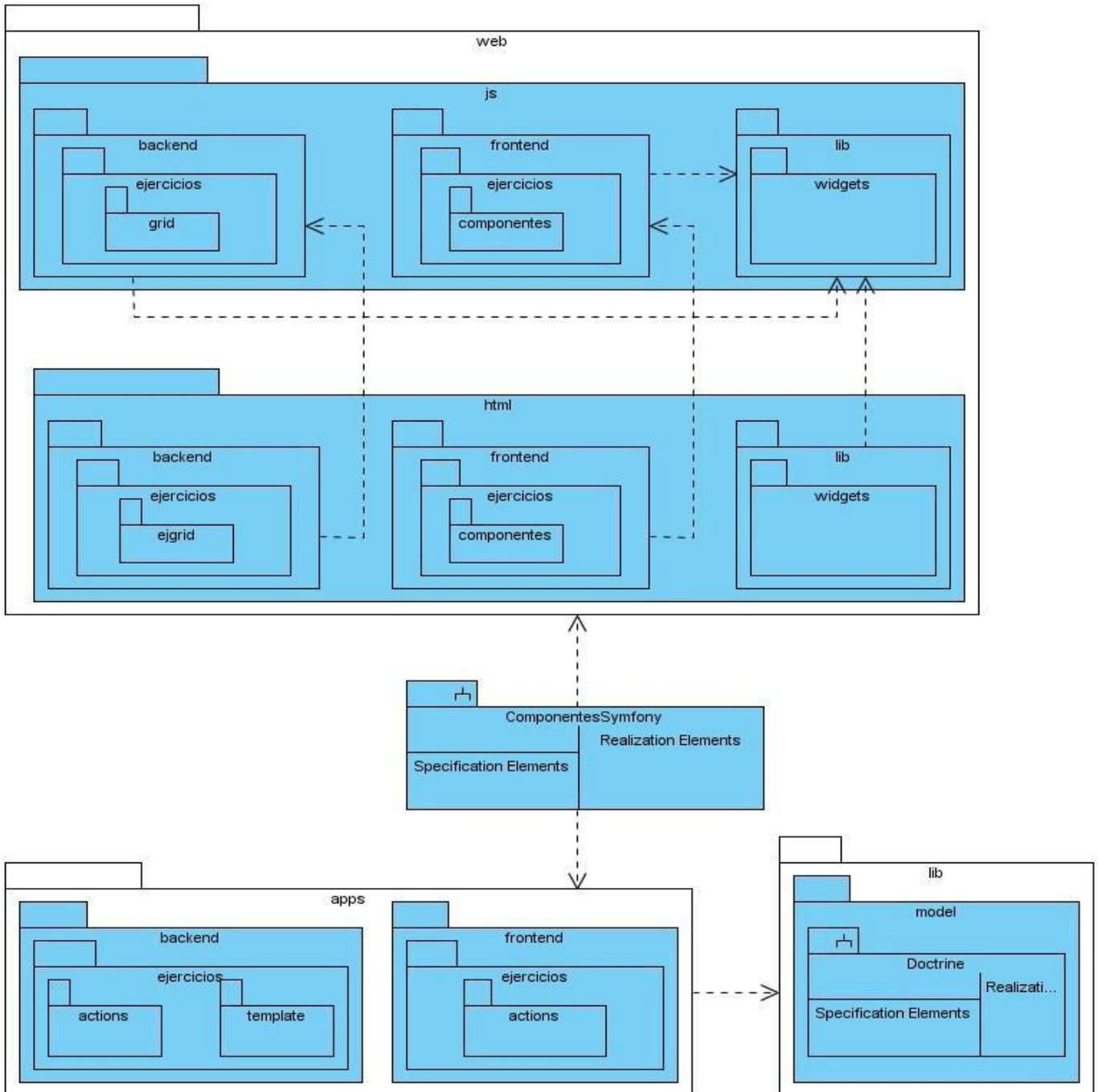


Figura 3.7 Diagrama de paquete

3.2.3 Diagramas de clases del diseño

Para un mejor entendimiento de los diagramas de clases del diseño es necesario aclarar los siguientes puntos:

- Se usan paquetes para distribuir y mostrar las responsabilidades de las clases en el patrón Modelo Vista Controlador.
- El subsistema Componentes Symfony contiene todos los componentes (núcleo del framework, clases internas y ficheros) que brinda Symfony para su uso.
- El subsistema Doctrine representa todos los componentes que utiliza el ORM Doctrine.

A continuación se muestran los diagramas de clases del diseño de los casos de uso del sistema.

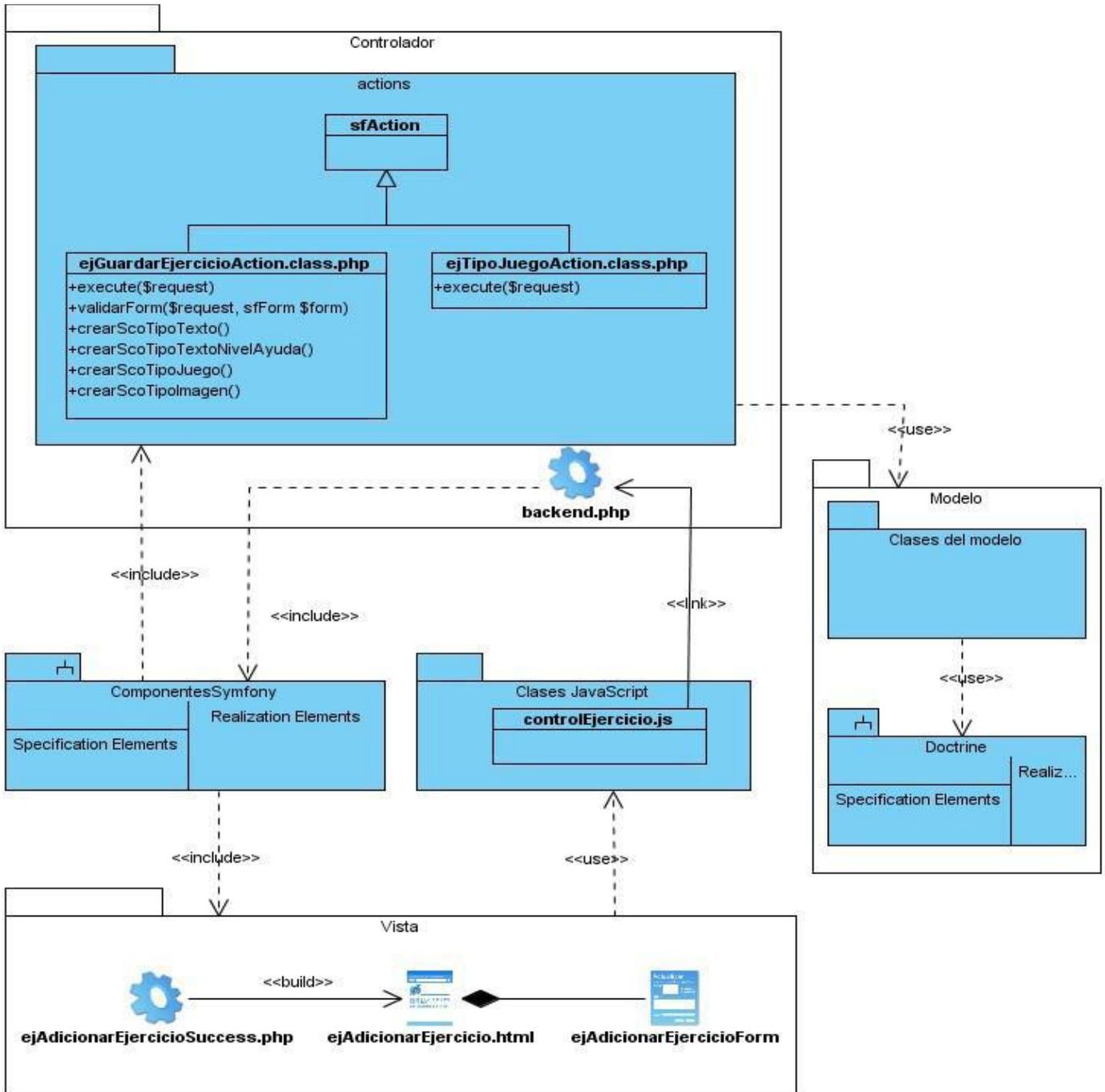


Figura 3.8. Diagrama de clases del diseño del caso de uso “Agregar ejercicio”.

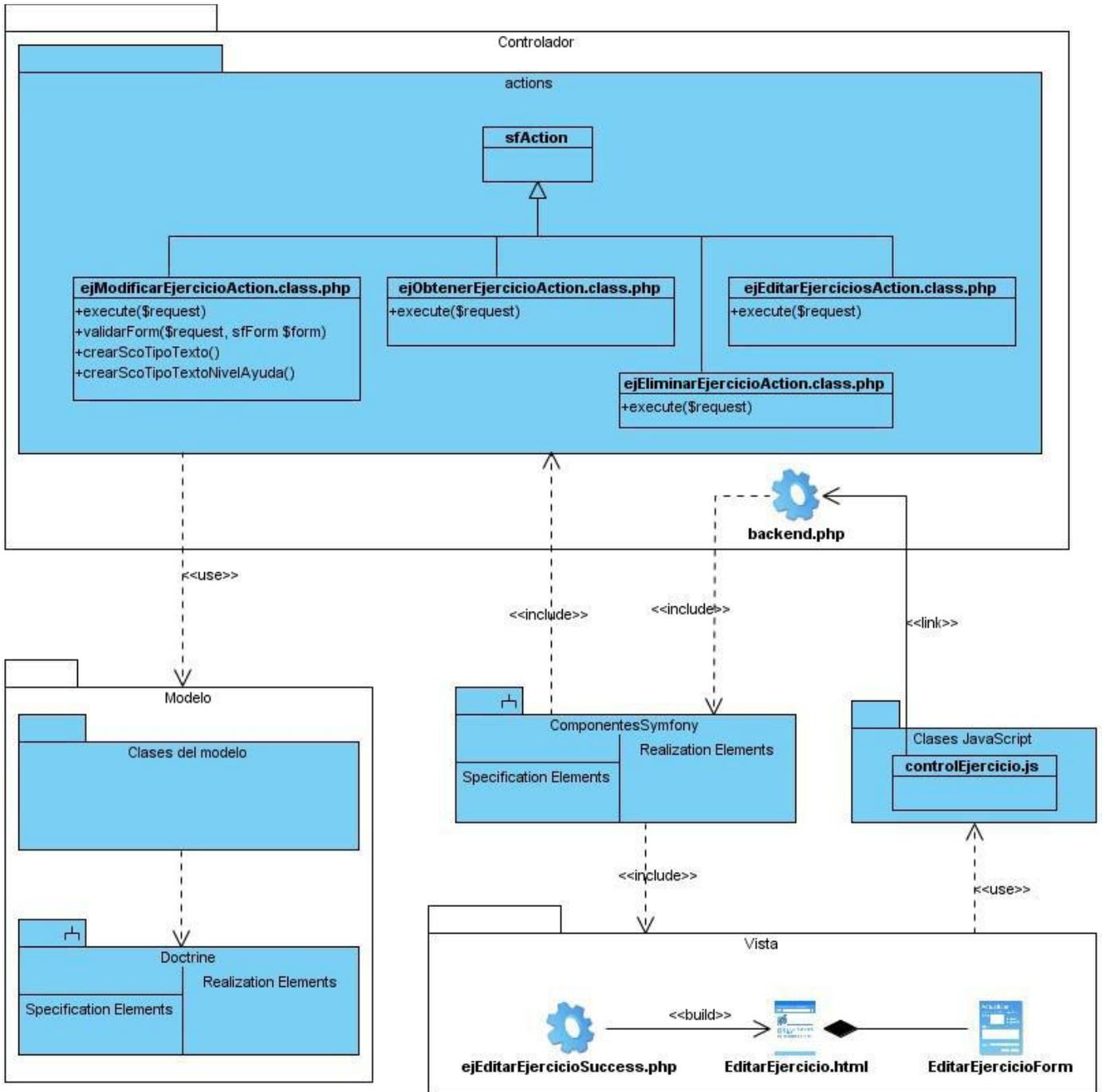


Figura 3.9. Diagrama de clases del diseño del caso de uso “Gestionar ejercicio”.

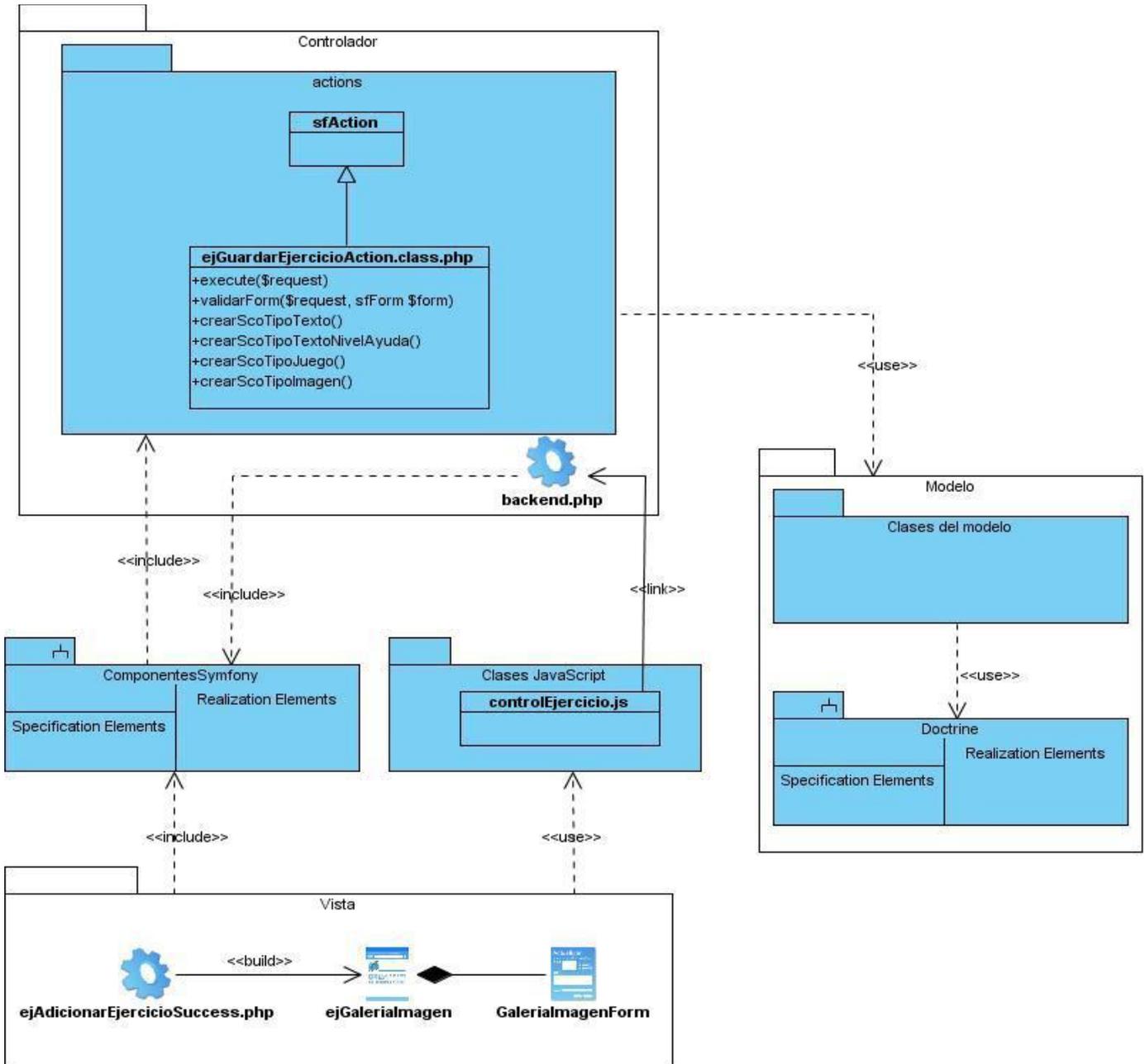


Diagrama de clases del diseño del caso de uso “Gestionar media de ejercicio”.

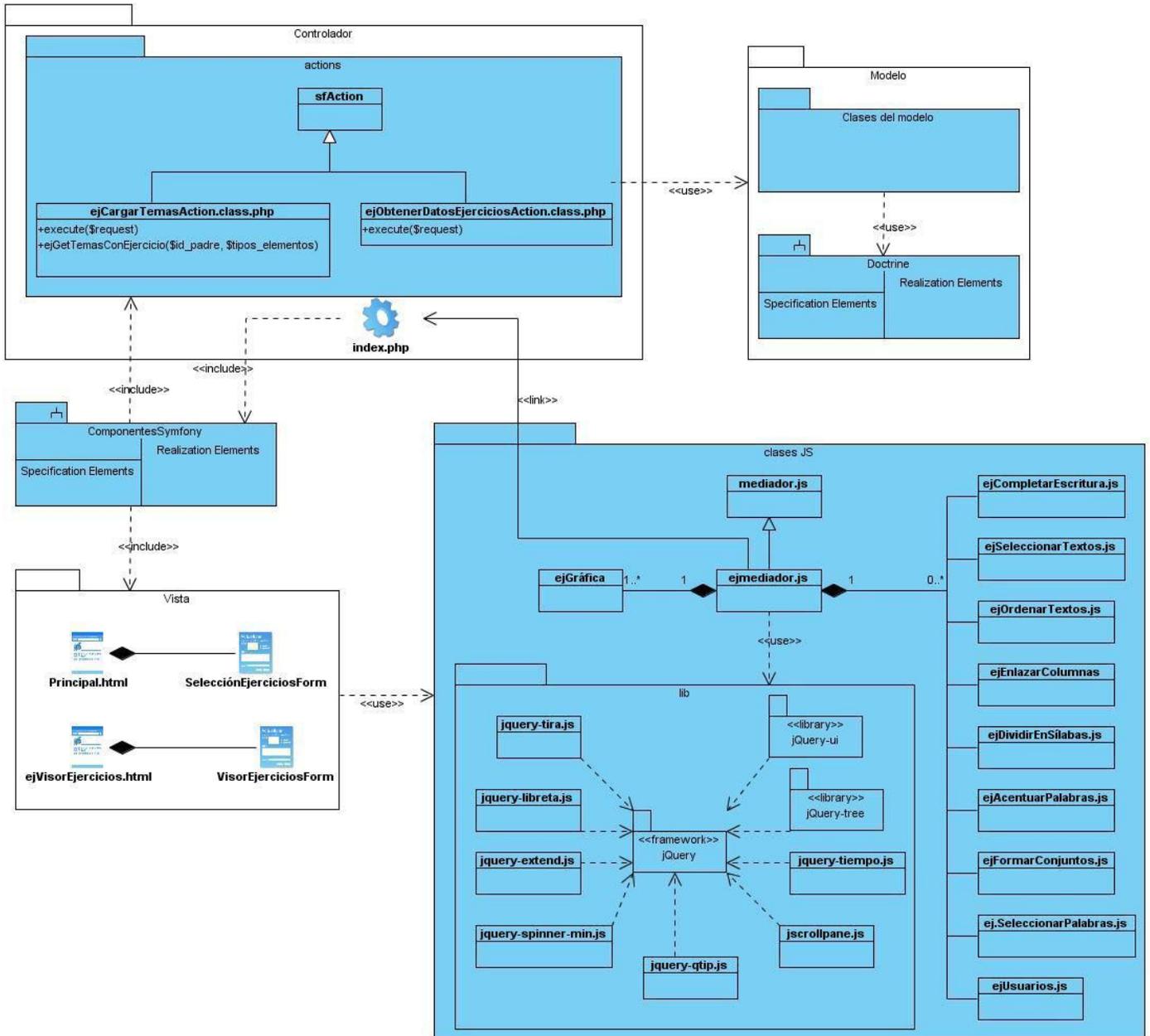


Figura 3.10. Diagrama de clases del diseño del caso de uso "Realizar ejercicio".

A continuación se muestra el diagrama de las clases que agrupa el paquete JavaScript de las aplicaciones frontend y backend.

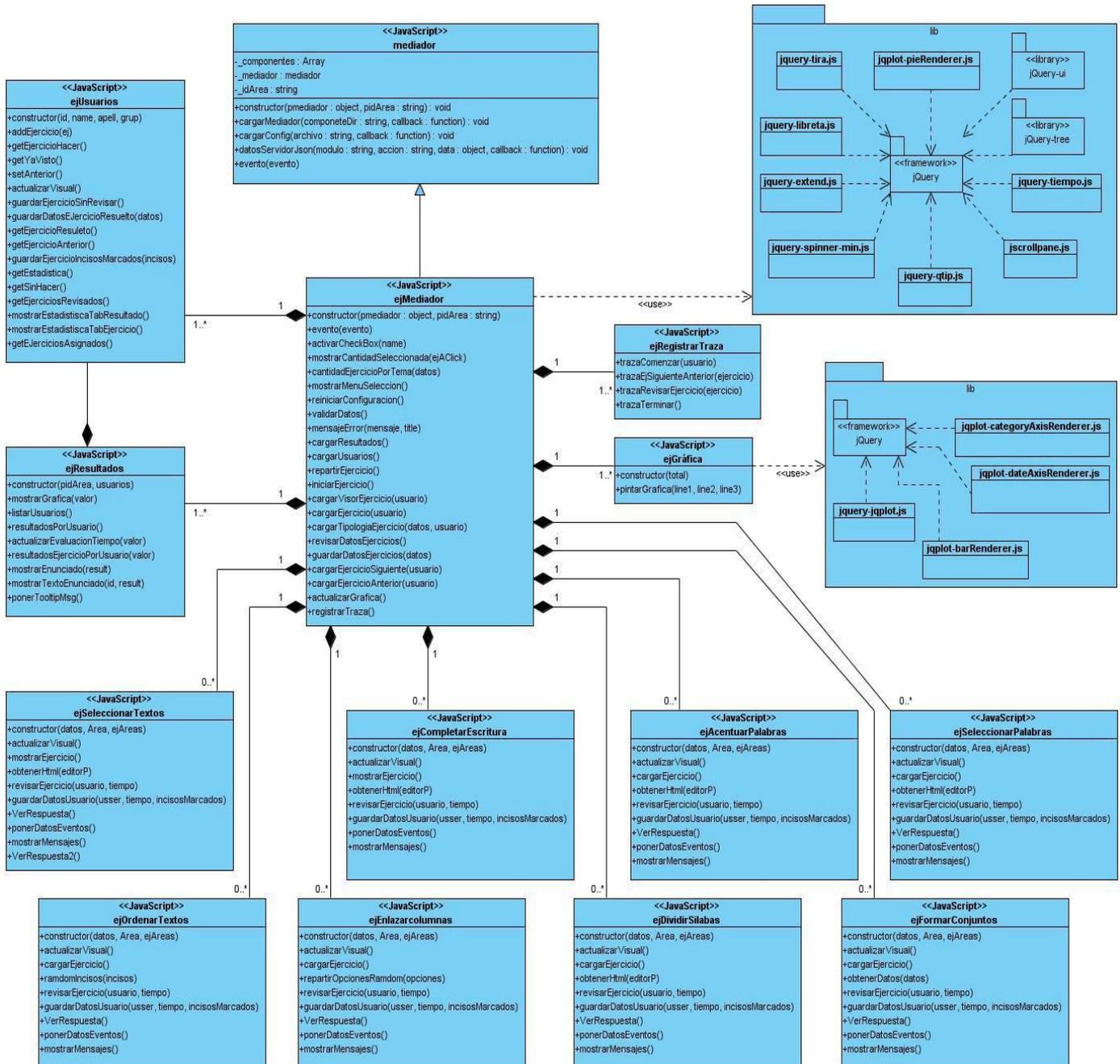


Figura 3.13 Paquete de clases JavaScript del frontend

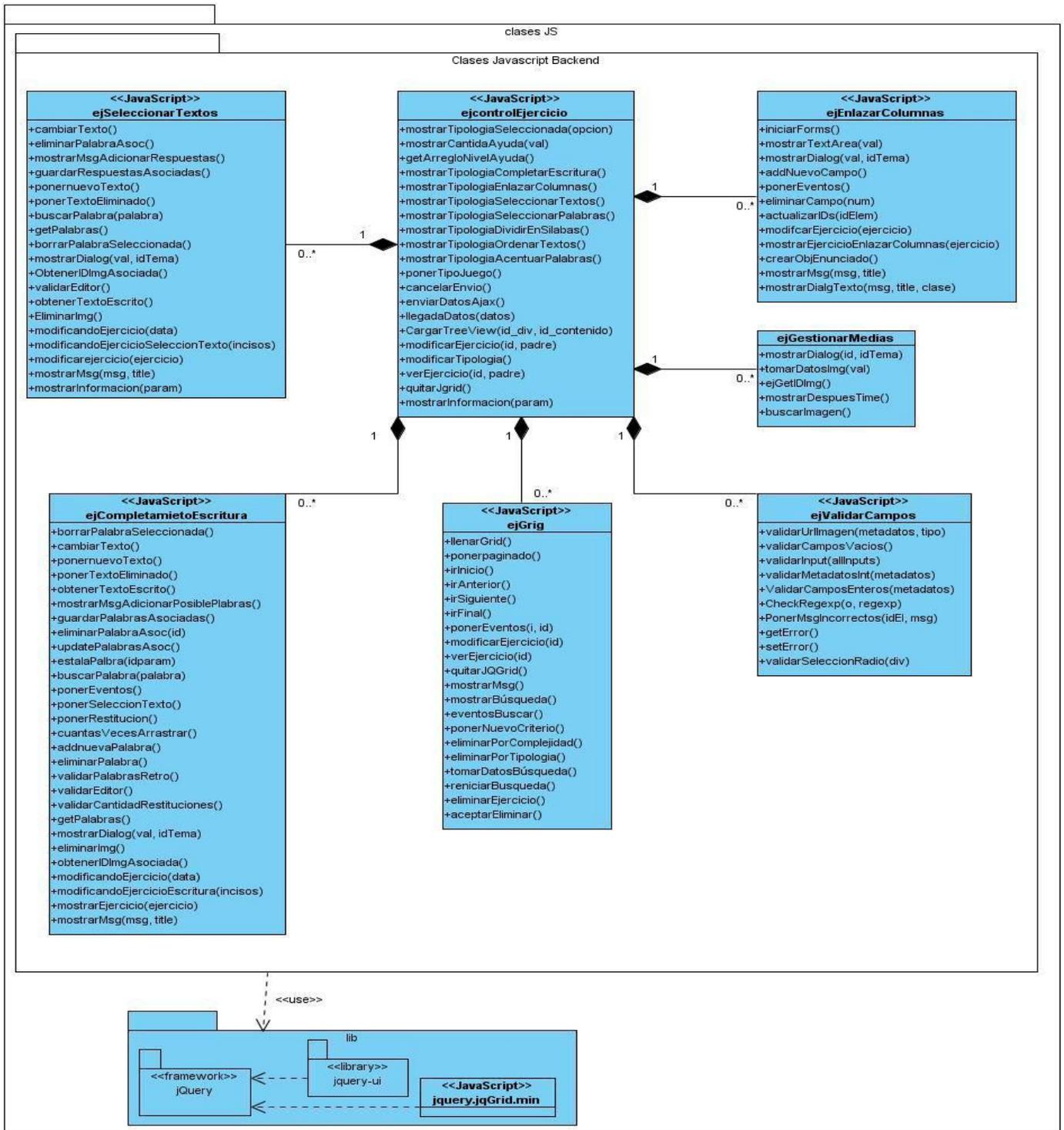


Figura 3.14 Paquete de clases JavaScript del backend

3.3 Diseño de la base de datos

Las bases de datos siguen una estructura relacional. PHP 5 y Symfony por el contrario son orientados a objetos. Por este motivo, para acceder a la base de datos como si fuera orientada a objetos, es necesaria una interfaz que traduzca la lógica de los objetos a la lógica relacional. Esta interfaz se denomina “mapeo de objetos a bases de datos” (ORM, de sus siglas en inglés “object-relational mapping”). Un ORM consiste en una serie de objetos que permiten acceder a los datos y que contienen en su interior cierta lógica de negocio. (28)

Doctrine es un ORM para PHP 5.2.3 y posterior, brinda una capa de abstracción de la base de datos muy completa. La característica más importante es que da la posibilidad de escribir consultas de base de datos en un lenguaje propio llamado Doctrine Query Language (DQL), es un lenguaje creado para ayudar al programador a extraer objetos de la base de datos. (33)

Diagrama de clases persistentes

El diagrama de clases persistentes muestra todas las clases capaces de guardar su estado en un medio permanente, más allá del tiempo de ejecución de la aplicación, lo cual está dado por el almacenamiento físico de la información de la clase, para la copia de seguridad en caso del fracaso del sistema, o para el intercambio de información.

El diagrama de clases persistentes del módulo Ejercicios se realizó a través del modelo de datos de la colección El Navegante del proyecto Multisaber-Navegante, puede ser consultado en el Anexo 2. A partir del diagrama de clases persistentes se obtuvo el **Modelo físico de datos**, el cual puede ser consultado en el Anexo 3.

3.4 Conclusiones

A través del análisis y diseño del sistema se ha logrado traducir los requisitos a una especificación que describe cómo implementar el sistema, de modo que se ha creado el punto de partida para las actividades de implementación. Se analizó el patrón de diseño seleccionado para facilitar el desarrollo y el mantenimiento del código. Además, se realizó el diseño de la base de datos, generando el modelo físico de datos a partir del diagrama de clases persistentes.

CONCLUSIONES GENERALES

Luego de la realización de este trabajo se puede arribar a las siguientes conclusiones:

- Se alcanzó un mejor entendimiento de las características del sistema, mediante la realización del modelo de dominio, la identificación de los requisitos funcionales y no funcionales, así como la elaboración del diagrama de casos de uso del sistema y la descripción detallada de cada uno de estos.
- A través del análisis y diseño del sistema se obtuvo la documentación necesaria para llevar a cabo las actividades de implementación.

RECOMENDACIONES

Se recomienda al proyecto Multisaber – Navegante realizar los flujos de trabajo Implementación y Prueba que propone la metodología RUP al sistema diseñado.

REFERENCIAS BIBLIOGRÁFICAS

1. **Verdecia Carballo, Enrique.** Edutec. Revista Electrónica de Tecnología Educativa. *Algunos Fundamentos Filosóficos y Psicológicos de la Tecnología Educativa.* [En línea] [Citado el: 6 de diciembre de 2010.] <http://edutec.rediris.es/Revelec2/revelec23/everdecia/everdecia.html>.
2. **Valero Aguayo, Luis.** MÁQUINAS DE ENSEÑANZA DE SKINNER. [En línea] [Citado el: 6 de diciembre de 2010.] http://www.robertexto.com/archivo12/maquinas_enseniar.htm.
3. **Marquès Graells, Pere.** El software educativo. *Universidad Autónoma de Barcelona.* [En línea] [Citado el: 2 de diciembre de 2010.] http://www.lmi.ub.es/te/any96/marques_software/.
4. **Galvis Panqueva, Álvaro.** [En línea] [Citado el: 2 de diciembre de 2010.] <http://www.slideshare.net/algalvis50/ingeniera-de-software-educativo-1992-parte-1-fundamentos>.
5. **Gutiérrez Lam, José Antonio.** [En línea] [Citado el: 3 de diciembre de 2010.] http://www.ucp.pr.rimed.cu/sitios/revistamendive/nfuentes/num32/pdf/Art_1_Jose.pdf.
6. **Coloma Rodríguez, Orestes.** Portal Nacional del Fórum de Ciencia y Técnica. *HIPERENTORNO DE APRENDIZAJE “EUREKA”: UN SOFTWARE EDUCATIVO PARA LA ENSEÑANZA DE LA MATEMÁTICA.* [En línea] [Citado el: 3 de diciembre de 2010.] www.forumcyt.cu/UserFiles/forum/Textos/1101687.pdf.
7. **Fernández Breis, Jesualdo Tomás.** *Universidad de Murcia.* [En línea] [Citado el: 3 de diciembre de 2010.] <http://dis.um.es/~jfernand/0405/tsm/tema1.pdf>.
8. **Lamarca Lapuente, María Jesús.** *Hipertexto: El nuevo concepto de documento en la cultura de la imagen.* [En línea] [Citado el: 5 de diciembre de 2010.] <http://www.hipertexto.info/documentos/hipermedia.htm>.
9. Grupo de Diseño y Optimización de Sistemas Informáticos CES “Felipe II” – Aranjuez. Universidad Complutense de Madrid. *itest.* [En línea] [Citado el: 4 de diciembre de 2010.] http://www.itest.es/docs/Manual_de_Profesor_iTest.pdf.
10. **Noguera Rojas, Asdrúbal.** [En línea] [Citado el: 4 de diciembre de 2010.] <http://unimet.blog.com.es/>.
11. **González Verdejo, Noelia.** [En línea] [Citado el: 4 de diciembre de 2010.] http://www3.unileon.es/dp/dfh/noelia/met/trab2/trabajo2_sistautor.htm.
12. **Galli, Luca.** Luca Galli's Site. [En línea] [Citado el: 5 de diciembre de 2010.] <http://www.lucagalli.net/en/>.
13. Hot Potatoes. [En línea] [Citado el: 5 de diciembre de 2010.] <http://hotpot.uvic.ca/>.
14. webArdora.net. [En línea] [Citado el: 5 de diciembre de 2010.] http://www.webardora.net/index_cas.htm.
15. Moodle. [En línea] [Citado el: 5 de diciembre de 2010.] <http://moodle.org>.
16. Entorno Virtual de Aprendizaje. [En línea] [Citado el: 5 de diciembre de 2010.] <http://eva.uci.cu/mod/resource/view.php?id=200>.
17. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El proceso Unificado de Desarrollo de Software.* Madrid : Addison Wesley, 2000. 84-7829-036-2.
18. **G. Figueroa, Roberth, J. Solís, Camilo y A. Cabrera, Armando.** *METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES.* *Universidad Técnica Particular de Loja, Escuela de*

- Ciencias en Computación*. [En línea] [Citado el: 10 de diciembre de 2010.]
<http://www.scribd.com/doc/42572078/Articulo-Metodologia-de-Sw-Formato>.
19. **Hernández Orallo, Enrique**. Universidad Politécnica de Valencia DISCA. [En línea] [Citado el: 8 de diciembre de 2010.] <http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>.
20. Sitio de descargas de software. [En línea] [Citado el: 8 de diciembre de 2010.]
http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/.
21. Definición de lenguaje de programación. [En línea] [Citado el: 7 de diciembre de 2010.]
<http://www.definicion.org/lenguaje-de-programacion>.
22. Centro de Referencia en Accesibilidad y Estándares Web. *INTECO - Instituto Nacional de Tecnologías de la comunicación*. [En línea] [Citado el: 7 de diciembre de 2010.]
http://www.inteco.es/file/Kilat7eU_6hCnkU_DZrdhA.
23. librosweb.es. [En línea] [Citado el: 7 de diciembre de 2010.]
<http://www.librosweb.es/javascript/capitulo1.html>.
24. PHP. [En línea] [Citado el: 7 de diciembre de 2010.] <http://www.php.net/manual/es/intro-what-is.php>.
25. **J. Gutiérrez, Javier**. Departamento de Lenguajes y Sistemas Informáticos Universidad de Sevilla . [En línea] [Citado el: 8 de diciembre de 2010.]
http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf .
26. librosweb.es. [En línea] [Citado el: 8 de diciembre de 2010.] http://www.librosweb.es/symfony_1_2/.
27. **Eguiluz, Javier**. Maestros del Web. [En línea] [Citado el: 8 de diciembre de 2010.]
<http://www.maestrosdelweb.com/editorial/el-framework-symfony-una-introduccion-practica-i-parte/>.
28. librosweb.es. [En línea] [Citado el: 8 de diciembre de 2010.] http://www.librosweb.es/symfony_1_2/.
29. **Eguiluz, Javier**. Maestros del Web. [En línea] [Citado el: 8 de diciembre de 2010.]
<http://www.maestrosdelweb.com/editorial/javascript-facil-y-rapido-con-jquery/>.
30. IBM WebSphere Commerce. *Guías de programación y aprendizaje*. [En línea] [Citado el: 9 de diciembre de 2010.]
<ftp://ftp.software.ibm.com/software/websphere/commerce/55/es/WC55ProgrammingGuide.pdf>.
31. **S Pressman, Roger**. *Ingeniería del software. Un enfoque práctico*. España : McGraw.Hill/Interamericana, 2002.
32. **Kramek, Andy**. *portalfox*. [En línea] [Citado el: 4 de marzo de 2010.]
<http://www.portalfox.com/modules.php?op=modload&name=Sections&file=index&req=viewarticle&artic=159>.
33. **Guardado, Iván**. *web.Ontuts Tutoriales y recursos web*. [En línea] [Citado el: 4 de marzo de 2011.]
<http://web.ontuts.com/tutoriales/utilizando-doctrine-como-orm-en-php/>.