



Universidad de las Ciencias Informáticas

Facultad 8

“Desarrollo del módulo Ejercicios de la Colección Multisaber en su versión multiplataforma”

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

Autores:

Ana María Álvarez Valdés

Osmany Montes de Oca Rodríguez

Tutores:

Ing. Fidel Alberto Curbelo Rosell

Ing. Ismael Armando Nodarse Mora

Ciudad de La Habana, junio 2010

“Año 52 de la Revolución”

Declaración de autoría

Declaramos ser los autores de la presente tesis y autorizamos a la Universidad de las Ciencias Informáticas para que haga uso de la misma, como estime pertinente, cediéndole de esta forma los derechos patrimoniales de la misma, con carácter exclusivo.

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

Ana María Álvarez Valdés

Firma del Autor

Osmany Montes de Oca Rodríguez

Firma del Autor

Ismael Armando Nodarse Mora

Firma del Tutor

Fidel Alberto Curbelo Rosell

Firma del Tutor

"Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber."

Albert Einstein

AGRADECIMIENTOS

De Ana:

A mis padres, por educarme e inculcarme buenos principios, por todo el amor y confianza que me han brindado durante toda mi vida. Gracias a ellos he logrado todo lo que tengo y lo que soy.

A mi hermanita querida, por todo su apoyo y su constante preocupación.

A mi abuela María, por su cariño y sus detalles de toda la vida.

A mis tíos Walter y Sonia, por ser como mis padres siempre que los he necesitado.

A los padres de Osmany y al resto de la familia, por su apoyo y preocupación.

A todos mis familiares y amigos que han estado siempre para mí.

A todos los que fueron de mi grupo alguna vez, desde el 8101 hasta el 8507.

A la Revolución Cubana, por facilitar que una joven humilde, hija de campesinos, se convierta hoy en Ingeniera en Ciencias Informáticas.

A Osmany, mi compañero de tesis y mi pareja, por toda su comprensión y cariño en estos 5 años.

De Osmany:

A mis padres Gina y Osvaldo por la educación y los principios que me han inculcado, por todo el apoyo que siempre he podido encontrar en ellos y por todo el sacrificio y esfuerzos que han tenido que hacer durante todos estos años en función de mi formación como profesional.

A Iván mi otro padre, al cual no tengo palabras para agradecerle todo lo que ha hecho por mí.



A mis hermanitas queridas por formar parte de mi inspiración.

A mis abuelos que constituyen una pieza fundamental en este proceso.

A todos mis tíos y primos, especialmente a mi tío Víctor y mi prima Ljudmila por todo lo que han hecho por mí y por toda mi familia en general.

A Norma, Serguey y Yusiley por estar siempre pendientes de mis cosas.

A todos mis compañeros de aula, especialmente a Ediel mi amigo y hermano, que siempre estaba para sacarme del apuro, a Ramiro y Danay, Yasniel, Jacobo, Héctor, Yeny, Adrián y Yaneisys, Ruly, Alejandro, en fin, a todos los que compartimos durante estos 5 años.

A mis compañeros, amigos y hermanos inseparables desde la primaria, Yasmany y Daimel por estar siempre conmigo en las buenas y las malas.

Un agradecimiento muy especial a Ana María mi compañera de tesis y de la vida durante estos 5 años, creo que sin su amor, apoyo y comprensión no hubiese podido llegar a la meta.

A toda la familia de Ana por lo que han hecho por mí y especialmente a sus padres y sus tíos Walter y Sonia.

Comunes:

A nuestros tutores Ismael y Fidel, por su apoyo, ayuda y amistad.

A la profesora Liana por todo lo que ha hecho por nosotros durante todos estos años.

A los profesores que nos impartieron clases en toda nuestra vida de estudiantes.

A Yasel y Yunior que aportaron su granito de arena a este trabajo.



DEDICATORIA

A nuestros padres, demás familiares y amigos que siempre han estado presentes, brindándonos su apoyo y ayuda incondicional.



RESUMEN

En la Universidad de las Ciencias Informáticas se desarrolla una nueva versión de la Colección Multisaber, conjunto de software educativos que se utiliza en las escuelas primarias cubanas para apoyar el proceso de enseñanza y aprendizaje.

En el presente trabajo se realizó el análisis, diseño e implementación de una aplicación web, multiplataforma y de código abierto, que posibilita a los guionistas o profesores insertar, modificar o eliminar los contenidos de los ejercicios, presentes en esta nueva versión de la Colección, directamente en la base de datos. Se determinó desarrollar dicha aplicación usando herramientas libres.

Inicialmente se documentaron los resultados del estudio realizado con respecto a las soluciones similares existentes y las herramientas y tecnologías utilizadas en distintos niveles, luego se describieron las características del sistema que se propone, a través de los artefactos propuestos por la metodología de desarrollo RUP. Al concluir se obtuvo una aplicación que posibilita que se realice de forma eficiente y dinámica el proceso de gestión de los cuestionarios interactivos presentes en el módulo Ejercicios de la Colección Multisaber en su versión multiplataforma.

ÍNDICE

INTRODUCCIÓN	1
FUNDAMENTACIÓN TEÓRICA.....	6
1.1 Introducción.....	6
1.2 Descripción del objeto de estudio	6
1.3 Análisis de soluciones similares existentes.....	7
1.4 Tendencias y tecnologías actuales	10
1.4.1 Metodologías de desarrollo.....	10
1.4.2 Lenguajes de desarrollo y de modelado	12
1.4.3 Herramientas CASE.....	16
1.4.4 Arquitectura y patrones.....	18
1.4.5 Entorno de Desarrollo Integrado.....	21
1.4.6 Sistemas Gestores de Bases de Datos	22
1.4.7 Servidores web.....	24
1.5 Propuesta de solución	24
1.6 Conclusiones.....	25
CARACTERÍSTICAS DEL SISTEMA.....	26
2.1 Introducción.....	26
2.2 Descripción del Modelo de Dominio.....	26
2.3 Especificación de requerimientos.....	28
2.3.1 Requisitos funcionales.....	28
2.3.2 Requisitos no funcionales	30
2.4 Modelo de Casos de Uso del Sistema.....	30
2.4.1 Actores del sistema	31
2.4.2 Diagrama de Casos de Uso del Sistema.....	31
2.4.3 Descripción de Casos de Uso del Sistema	31
2.5 Conclusiones.....	44
ANÁLISIS Y DISEÑO DEL SISTEMA	45
3.1 Introducción.....	45
3.2 Modelo de Análisis.....	45
3.2.1 Diagramas de Clases del Análisis.....	45
3.3 Modelo de Diseño.....	48
3.3.1 Diagramas de Clases del Diseño	48
3.4 Diagrama de Clases Persistentes	57



3.5 Modelo de Datos	58
3.6 Patrones de diseño	58
3.7 Conclusiones	59
IMPLEMENTACIÓN DEL SISTEMA	60
4.1 Introducción	60
4.2 Modelo de Implementación	60
4.2.1 Diagrama de Componentes	60
4.2.2 Diagrama de Despliegue	62
4.3 Interfaz de la aplicación	63
4.4 Conclusiones	70
CONCLUSIONES GENERALES	71
RECOMENDACIONES	72
REFERENCIAS BIBLIOGRÁFICAS	73
BIBLIOGRAFÍA	75
GLOSARIO DE TÉRMINOS	78
ANEXOS	80

INTRODUCCIÓN

En la actualidad existe un ritmo acelerado en el avance tecnológico desarrollado en el marco que rodea la actividad humana, estrechamente relacionado con el impacto producido por el empleo de las Tecnologías de la Información y las Comunicaciones (TICs). Debido a dicho desarrollo, las empresas, organizaciones y gobiernos están prestándole gran atención y han enfocado sus estrategias en aras de explotar los recursos que ofrece para su desempeño.

El estado revolucionario cubano, reconocido a nivel mundial como baluarte indiscutible en materia de educación, ha hecho énfasis en el proceso de informatización de la sociedad, el cual se encarga de poner todo el desarrollo tecnológico al alcance del pueblo. Como resultado, todos los niveles de enseñanza cuentan con la ayuda inestimable que brindan las TICs.

En medio de todo el proceso de revoluciones educacionales, la dirección del Ministerio de Educación (MINED) decidió crear grupos de docentes con motivaciones informáticas para elaborar software educativo, con el objetivo de elevar la calidad del proceso de enseñanza-aprendizaje. Con esta idea se crean varias colecciones para distintos niveles educacionales, como son:

- Multisaber para la educación primaria.
- El Navegante para la educación secundaria.
- Futuro para la educación preuniversitaria.

La Colección Multisaber constituye un conjunto de software educativos que se sustentan en la concepción de “Hiperentornos de aprendizaje”, definidos como: “Sistema informático basado en tecnología hipertexto que contiene una mezcla o elementos representativos de diversas tipologías de software educativo” (1), en los que están presentes contenidos relacionados con las materias de Matemáticas, Español, Ciencias Naturales e Informática. Esta Colección fue creada por diversos equipos de desarrollo, con diferentes tecnologías y lenguajes de programación, utilizando herramientas de software propietario como ToolBook Instructor, Adobe Flash, Adobe Director y otras.

La República Bolivariana de Venezuela, también enfrascada en un proceso de transformaciones, no está exenta al empleo de las TICs en el desarrollo educacional, por lo que el intercambio y colaboración con Cuba se ha extendido a la asistencia en métodos, programas y nuevas técnicas del proceso docente educativo, entre estos se encuentra Multisaber. En este país existen políticas, en lo que a informática se refiere, que determinan la migración a software libre de todos los organismos estatales.

A la Universidad de las Ciencias Informáticas (UCI), en conjunto con el MINED, se le ha asignado la misión de construir una nueva versión de la Colección Multisaber que sea multiplataforma y desarrollada con herramientas libres para satisfacer las necesidades del Ministerio del Poder Popular para la Educación, de la República de Venezuela. El proceso de construcción consiste en la implementación de una versión para la Web y la contextualización de los contenidos existentes en los diferentes módulos con que cuenta la Colección.

La Colección Multisaber presenta un prototipo estandarizado en todos los productos que la integran, en los cuales están presentes los siguientes módulos: Temas, Ejercicios, Juegos, Mediateca, Resultados y Maestro. Todos estos se han visto afectados en la fase de migración y construcción de la nueva versión y se han presentado diferentes **situaciones problemáticas** como es el caso del módulo Ejercicios donde:

La aplicación existente en la cual los guionistas introducen los contenidos de los cuestionarios interactivos, es capaz de generar ficheros de textos con la información introducida y brinda la posibilidad de modificar o eliminar según se desee dichos contenidos. La misma está desarrollada utilizando la herramienta de autor Toolbook Instructor 2004, un software propietario que implica ciertas restricciones en cuanto a su utilización, distribución y comercialización, por lo que se hace necesario el desarrollo de una nueva aplicación, utilizando software de libre uso y redistribución. Además que incluya mejoras en las tipologías de ejercicios que contengan imágenes, pues no se cuenta con una funcionalidad que permita una correcta gestión de estos recursos. Una deficiencia de esta aplicación es que no permite al profesor escoger la forma en la cual desea que se muestre el ejercicio, lo que implica que no se le dé utilidad a las plantillas implementadas en el módulo Ejercicios. Dicha aplicación utiliza una base de datos local que los guionistas tienen que copiar para su estación de trabajo, imposibilitando que trabajen varios en un mismo producto a la vez.

Otra dificultad es el proceso de insertar en la Base de Datos (BD) el contenido de los cuestionarios, lo cual se realiza de forma manual. Asimismo una vez que el contenido se encuentre en la BD no existe forma de que los guionistas puedan modificar o eliminar algún cuestionario interactivo.

El módulo Ejercicios cuenta con un proceso o lógica interna capaz de generar la información que será consultada posteriormente por la capa de presentación para visualizar el funcionamiento de los cuestionarios interactivos. Esta funcionalidad no se desarrolló utilizando el paradigma de Programación Orientada a Objeto y no se emplearon correctos

patrones arquitectónicos, por tanto, el código es poco reutilizable, no escalable y dificulta el mantenimiento y mejoras para posteriores versiones.

Problema a resolver

¿Cómo lograr que se realice de forma eficiente y dinámica el proceso de gestión y funcionamiento de los cuestionarios interactivos presentes en el módulo Ejercicios de la Colección Multisaber en su versión multiplataforma?

Objetivo general

Realizar de forma eficiente y dinámica el proceso de gestión y funcionamiento de los cuestionarios interactivos presentes en el módulo Ejercicios de la Colección Multisaber en su versión multiplataforma.

Objeto de estudio

El proceso de producción de software educativos.

Objetivos específicos

1. Realizar un estudio crítico valorativo de las diferentes tendencias y herramientas existentes relacionadas con la gestión de contenidos para ejercicios presentes en productos educativos.
2. Implementar el funcionamiento dinámico de los cuestionarios interactivos presentes en el módulo Ejercicios de la Colección Multisaber en su versión multiplataforma.
3. Efectuar el análisis y diseño de una aplicación que posibilite el proceso de inserción, modificación y eliminación de los contenidos presentes en cuestionarios interactivos para el módulo Ejercicios de la Colección Multisaber en su versión multiplataforma.
4. Implementar dicha aplicación para ser utilizada en la Colección Multisaber en su versión multiplataforma.

Campo de acción

El desarrollo de ejercicios para software educativos.

Resultados esperados

- Se podrá contar con una herramienta que facilite la gestión de los contenidos de los ejercicios de la Colección Multisaber, es decir, que permitirá a los especialistas modificar, agregar y eliminar el contenido de los ejercicios en la BD, según su

tipología.

- Se desarrollará el módulo Ejercicios de la nueva Colección Multisaber, posibilitando con ello que los estudiantes puedan contar con un medio interactivo que les permita comprobar los conocimientos adquiridos.
- Se contará con la documentación de todo el proceso de desarrollo, de modo que sirva de apoyo a futuras versiones de la colección.

Métodos científicos

Teóricos

Análisis Histórico Lógico: Se aplicará para realizar un estudio del estado del arte. Además para analizar la trayectoria histórica real de la Colección Multisaber en su proceso de migración a la versión multiplataforma y de los módulos que están presentes en ésta, principalmente el módulo Ejercicios en toda su evolución y desarrollo.

Empíricos

Observación: A través del registro visual, se determinará el comportamiento del objeto de investigación tal y como este se da en la realidad, es decir, es una forma de obtener información directa e inmediata sobre el fenómeno u objeto que está siendo investigado, en este caso, el módulo Ejercicios de la Colección Multisaber y la herramienta que permite la gestión de los ejercicios.

Estructura capitular

Capítulo 1: **Fundamentación teórica**, se hace referencia a los elementos teóricos que constituyen la base de la investigación realizada. Se describen los conceptos esenciales para comprender el problema, así como las tecnologías, herramientas y lenguajes de programación a utilizar durante el desarrollo de la solución.

Capítulo 2: **Características del sistema**, se describe el Modelo de Dominio donde se capturan los tipos de objetos más importantes o los principales eventos que suceden en el entorno donde se encuentra el sistema. Se especifican los requisitos esperados.

Capítulo 3: **Análisis y diseño del sistema**, se describe la solución que se propone, a partir de los diferentes modelos y diagramas presentes en el flujo de trabajo de Análisis y Diseño.

Capítulo 4: **Implementación del sistema**, se expone cómo está implementado el sistema en términos de componentes, a través que los diferentes diagramas.

Para dar cumplimiento a los objetivos planteados, se proponen realizar las **tareas de investigación**:

1. Realizar el diseño teórico de la investigación.
2. Estudiar la implementación del funcionamiento interno del módulo y la Base de Datos de los ejercicios. Así como la estructura de los ficheros JSON en los que se almacena el contenido.
3. Estudiar la herramienta existente para la gestión de los contenidos de los ejercicios.
4. Realizar un estudio del estado del arte.
5. Realizar un levantamiento de requisitos funcionales.
6. Reimplementar el proceso de funcionamiento interno del módulo Ejercicios.
7. Analizar y diseñar una herramienta que posibilite la gestión de los ejercicios en la versión multiplataforma de la Colección Multisaber.
8. Implementar una herramienta que posibilite la gestión de los ejercicios en la versión multiplataforma de la Colección Multisaber.
9. Conformar la documentación de todo el proceso de desarrollo.

FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En el presente capítulo se realiza un estudio acerca de las tecnologías y software usados en la actualidad, donde se abordan aspectos como características, ventajas y desventajas. Se efectúa un análisis de soluciones similares existentes en la UCI y a nivel internacional.

1.2 Descripción del objeto de estudio.

El uso de las TICs en la educación es cada vez mayor, y con ello, el proceso de producción y desarrollo del software educativo. Uno de los más populares, la multimedia: objeto que usa simultáneamente combinación de texto, color, gráficas, animaciones, video, sonido, en un mismo entorno, donde el estudiante interactúa con los recursos para mejorar el proceso de enseñanza-aprendizaje. (2)

Gracias a las facilidades de los recursos multimedia que permiten la interactividad, se logra motivar al estudiante a participar con más libertad y posibilidad de elección, en su propio proceso de enseñanza.

Los sistemas multimedia pueden presentar características diferentes en cuanto a su utilización en entornos de aprendizaje. Con relación a ello suelen distinguirse en multimedia interactiva, las cuales se clasifica fundamentalmente en:

Multimedia tutoriales: se caracterizan por ofrecer contenidos y ejercicios que promueven la autorregulación de los aprendizajes. En muchos casos estos materiales guían al estudiante a través de una serie de pasos para lograr un objetivo. (3)

Programas de ejercitación: ofrecen una serie de ejercicios a modo de prácticas para los estudiantes. En estos materiales no es necesaria la existencia de contenidos conceptuales de refuerzo. (3)

Multimedia de contenido: no solamente es la información en sí misma, sino que además está relacionado con la manera de exponer esa información. Ofrecen sus contenidos con una interfaz especialmente diseñada, desde un punto de vista educativo, para que los estudiantes tengan un acceso adecuado a la información y así favorecer los procesos de autorregulación de los aprendizajes. (3)

Simuladores: se pretende generar espacios que simulan la realidad pero que permiten un mejor control frente a los problemas. Por ejemplo, clases de laboratorio, video juegos, etc. (3)

Entre las principales ventajas de los sistemas multimedia educativos está lograr el interés y motivación del estudiante, lo cual le posibilita que esté permanentemente activo al interactuar con el ordenador y que además aprenda en menos tiempo. (4)

Uno de los principales componentes de estos software son los **cuestionarios interactivos:** que posibilitan un aprendizaje a partir de errores, pues con la retroalimentación inmediata que recibe el usuario le permite conocer el error justo en el momento en que se produce y generalmente el programa le ofrece la oportunidad de ensayar nuevas respuestas o formas de actuar para superarlos.

1.3 Análisis de soluciones similares existentes.

Existen varias aplicaciones que permiten la gestión de contenidos para cuestionarios interactivos o actividades escolares presentes en diferentes software educativos, capaces de crear ejercicios, administrarlos y gestionar sus contenidos, algunas de éstas son:

HotPotatoes

Es un programa en el cual se pueden crear actividades interactivas de carácter educativo, como ejercicios de selección simple y múltiple, ordenar frases, crucigramas, relacionar y ordenar, completar los espacios en blanco, todas para ser accesibles a través de la Web, no es de código abierto. (5)

Es fácilmente accesible en línea a través de Internet, es gratuito para uso individual o educativo sin ánimo de lucro, siempre y cuando el material producido sea accesible públicamente. Se puede descargar de Internet, pero debe ser registrado (gratuitamente),

para activar todas sus funcionalidades. Si el usuario lo desea puede comprar una versión para su libre distribución. (5)

Componentes:

- JBC crea ejercicios de selección múltiple.
- JCloze genera ejercicios de completar espacios en blanco.
- JCross crea crucigramas de cualquier tamaño.
- JMatch crea ejercicios de emparejamiento u ordenación.
- JMix crea ejercicios de reconstrucción de frases o párrafos a partir de palabras desordenadas.
- The Master, es una herramienta que permite compilar de forma automática ejercicios de Hot Potatoes en unidades didácticas. Permite enlazar todos los ejercicios de una unidad utilizando botones de navegación. (5)

Ardora

Ardora es una aplicación informática para docentes, que permite al usuario crear sus propias actividades, en formato HTML, para sus alumnos de un modo muy sencillo (6).

Con Ardora se pueden crear más de 45 tipos distintos de actividades, crucigramas, sopas de letras, completar, paneles gráficos, relojes, etcétera, así como más de 10 tipos distintos de páginas multimedia: galerías, panorámicas o zooms de imágenes, reproductores mp3 o flv, y otros (6).

El profesor sólo debe centrar su esfuerzo en los elementos de la actividad no en su tratamiento informático. Una vez introducidos los elementos de la actividad, mediante formularios muy sencillos, Ardora creará la página web y los archivos necesarios, luego sólo necesitará un navegador para visualizar el contenido (6).

eXelearning

Es un editor de recursos educativos Open Source, de gran utilidad para los docentes, ya que permite construir contenido web didáctico sin necesidad de ser experto en la edición y marcado con XML o HTML. Es multiplataforma, es decir, que puede hacerse funcionar en los habituales tipos de ordenador y sistemas operativos. (7)

Tiene diferentes módulos que permiten muchas utilidades a la hora de diseñar los recursos: plantillas para crear aplicaciones en Java, actividades de lectura, posibilidad de insertar vídeos, imágenes, archivos de sonido mp3, animaciones GIF, ficheros adjuntos, actividades de completar, pruebas de múltiples respuestas, actividades de verdadero falso, navegación por páginas web externas o exámenes para comprobar el nivel alcanzado por los alumnos al final de la unidad. (8)

CRODA

Es un proyecto de la UCI que se encarga del desarrollo de una herramienta de autor web que permita la creación de objetos de aprendizaje (OA) reutilizables, accesibles, duraderos e interoperables, de forma flexible, empleando los estándares SCORM y LOM. Entre los elementos que componen un OA creado a partir de esta herramienta de autor se encuentran: imágenes, audios, videos, links, ecuaciones matemáticas y variados ejercicios de autoevaluación, entre otros. Estos OA pueden ser almacenados en el repositorio de objetos de aprendizaje (RHODA) debido al uso de estándares que permiten la interoperabilidad entre sistemas.

Al concluir el análisis de algunas de las soluciones existentes se ha podido observar que a pesar de la gran utilidad que tienen estas herramientas no son capaces de dar solución al problema existente debido a que ninguna de ellas gestiona todas las tipologías de los cuestionarios presentes en el módulo Ejercicios de la Colección Multisaber en su versión multiplataforma, en el caso de HotPotatoes y Ardora, no son de código abierto, factor común con la aplicación existente desarrollada con Toolbook Instructor 2004, lo que impide la modificación y adaptación a las tipologías de ejercicios que no gestionan. En el caso de eXelearning y CRODA, no gestionan todas las tipologías presentes en el módulo mencionado, además, no brindan la posibilidad de almacenar en una BD los contenidos de modo que sean útiles a la Colección, sino que los exportan en paquetes SCORM los cuales no son interpretados por ésta.

Básicamente el funcionamiento de todas estas aplicaciones analizadas se centra en crear actividades educativas y exportarlas en páginas web autosuficientes, paquetes SCORM o proyectos específicos de cada aplicación, entre otros.

Lo que se necesita para solucionar el problema planteado no es eso, sino una aplicación capaz de gestionar los contenidos de los ejercicios ya existentes en el módulo de

acuerdo con las características específicas y funcionamiento de los mismos, así como al diseño y estructura de la BD.

Algo similar sucede con otras aplicaciones semejantes existentes en la UCI, que tampoco son adaptables al diseño de la BD utilizada por los productos de la Colección ni gestionan todas las tipologías presentes en el módulo.

1.4 Tendencias y tecnologías actuales.

1.4.1 Metodologías de desarrollo.

Una metodología de desarrollo de software es un conjunto de procedimientos, técnicas y pasos a seguir para construir un software. Define quién debe hacer qué, cuándo y cómo para alcanzar un determinado objetivo.

Existe un gran número de metodologías usadas en la actualidad como: **Programación Extrema (XP)**, que está especialmente orientada a proyectos cortos, aquellos en los cuales los equipos de desarrollo son pequeños, con plazos reducidos, requisitos volátiles, y basados en nuevas tecnologías, y en los cuales el cliente forma parte del equipo de desarrollo; **SCRUM**, especialmente indicada para proyectos con un rápido cambio de requisitos, donde el desarrollo del software se realiza mediante iteraciones, denominadas *sprints*, con una duración de 30 días, y otra característica distintiva es que se efectúan reuniones a lo largo del proyecto, entre las que se destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración; **Proceso Unificado de Desarrollo (RUP)** la cual se divide en 4 fases el desarrollo del software:

- Inicio: tiene como objetivo definir el alcance del proyecto.
- Elaboración: su objetivo es determinar la arquitectura óptima.
- Construcción: el objetivo es obtener la capacidad operacional inicial.
- Transmisión: el objetivo es llegar a obtener el *release* del proyecto. (9)

Define 9 disciplinas a realizar en cada fase del proyecto:

- Modelado del negocio: se pretende llegar a un mejor entendimiento de la organización donde se va a implantar el producto.
- Requisitos: es uno de los flujos de trabajo más importantes, porque en él se establece qué tiene que hacer exactamente el sistema que se construya.

- Análisis y Diseño: el objetivo de este flujo es traducir los requisitos a una especificación que describe cómo implementar el sistema.
- Implementación: se implementan las clases y objetos en ficheros fuente, binarios, ejecutables y otros. Además se deben hacer las pruebas de unidad.
- Pruebas: se evalúa la calidad del producto que se está desarrollando.
- Despliegue: el objetivo es producir con éxito distribuciones del producto y distribuirlo a los usuarios.
- Gestión de proyecto: el objetivo es lograr un balance al gestionar objetivos, riesgos y restricciones para desarrollar un producto que sea acorde a los requisitos de los clientes.
- Configuración de control y cambios: se pretende mantener la integridad de todos los artefactos que se crean en el proceso, así como de mantener información del proceso evolutivo que ha seguido.
- Entorno: se da soporte al proyecto con las adecuadas herramientas, procesos y métodos. (Ver Anexo 1)

En su modelación se definen como elementos principales los siguientes:

- Trabajadores (quién): Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Son los que realizan las actividades.
- Actividades (cómo): Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.
- Artefactos (qué): Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.
- Flujo de actividades (cuándo): Secuencia de actividades realizadas por trabajadores y que producen un resultado de valor observable.

El proceso de software propuesto por RUP tiene tres características esenciales: está dirigido por los casos de uso, está centrado en la arquitectura, y es iterativo e incremental.

Es considerada como una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software, debido a que en cada ciclo de iteración, se hace exigente el uso de artefactos. Además, se aplica a un proceso más controlado, con

numerosas políticas o normas, en el que existe un contrato prefijado, el grupo de desarrollo es significativamente grande y el cliente interactúa con el equipo de desarrollo mediante reuniones.

1.4.2 Lenguajes de desarrollo y de modelado.

Lenguajes de Programación

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. (10)

Lenguaje del lado del cliente:

HTML

HTML es el lenguaje utilizado para la creación de páginas web, significa "HyperText Mark-Up Language", y en español, "Lenguaje para el Formato de Documentos de Hipertexto". Los documentos HTML no son documentos de texto normal, sino documentos de hipertexto ya que en el propio documento aparecen enlaces a otros documentos. (11)

El lenguaje HTML es un estándar reconocido en todo el mundo y cuyas normas son definidas por el World Wide Web Consortium (W3C). Como se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de forma muy similar en cualquier navegador de cualquier sistema operativo. (11)

El propio W3C define el lenguaje HTML como "un lenguaje reconocido universalmente y que permite publicar información de forma global". Desde su creación, el lenguaje HTML ha pasado de ser un lenguaje utilizado exclusivamente para crear documentos electrónicos a ser un lenguaje que se utiliza en muchas aplicaciones electrónicas como buscadores, tiendas online y banca electrónica. Es un lenguaje utilizado únicamente para dar estructura a una página web. (11)

JavaScript

Es un lenguaje interpretado por la mayoría de los navegadores en sus últimas versiones. Se utiliza principalmente en un navegador web, permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas. No es un lenguaje orientado a objeto por lo que no dispone de herencia. Para evitar incompatibilidad el W3C diseñó un estándar nombrado DOM (en inglés Document Object Model, en español Modelo de Objetos del Documento). (12)

Puede incluirse en cualquier documento y es compatible con HTML. Puede ser utilizado por profesionales y para quienes se inician en el desarrollo y diseño de sitios web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos.

Lenguajes del lado del servidor:

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el lado del servidor pueden efectuar consultas a BD y otras tareas para crear la página final que es lo que verá el cliente, la cual contiene solamente código HTML, como resultado de la ejecución de la página PHP.

ASP

Es una tecnología del lado de servidor desarrollada por Microsoft para el desarrollo de sitios web dinámicos. ASP significa en inglés (Active Server Pages), fue liberado por Microsoft en 1996. Las páginas web desarrolladas bajo este lenguaje necesitan tener instalado Internet Information Server (IIS) (12).

ASP no necesita ser compilado para ejecutarse. Existen varios lenguajes que se pueden utilizar para crear páginas ASP. El más utilizado es VBScript, nativo de Microsoft. ASP se puede hacer también en Perl and JScript (no JavaScript). El código ASP puede ser insertado junto con el código HTML. (12)

Ventajas:

- Usa Visual Basic Script
- Comunicación óptima con SQL Server

- Soporta el lenguaje JavaScript

Desventajas:

- Código desorganizado
- Se necesita escribir mucho código para realizar funciones sencillas
- Tecnología propietaria
- Hospedaje de sitios web costosos (12)

Python

Es un lenguaje de programación que permite trabajar más rápido e integrar sus sistemas de manera más eficaz. Se puede aprender a usar Python y ver casi de inmediato las ganancias de la productividad y reducir los costos de mantenimiento. Python se ejecuta en Windows, Linux / Unix, Mac OS X, y ha sido portado a Java y .NET, máquinas virtuales. Python es libre de usar, incluso para productos comerciales. (13)

Es un lenguaje de scripting independiente de la plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, por lo que no se necesita compilar el código fuente para poder ejecutarlo, esto ofrece ventajas de rapidez de desarrollo y el inconveniente de una menor velocidad.

Ventajas:

- Libre y fuente abierta
- Lenguaje de propósito general
- Gran cantidad de funciones y librerías
- Sencillo y rápido de programar
- Multiplataforma
- Licencia de código abierto (Open Source)
- Orientado a Objetos
- Portable

Desventajas:

- Lentitud por ser un lenguaje interpretado

Ruby

Es un lenguaje de programación dinámico y de código abierto enfocado en la simplicidad y productividad. Su elegante sintaxis se siente natural al leerla y fácil al escribirla. Mezcla partes de Perl, Smalltalk, Eiffel, Ada, y Lisp, para formar un lenguaje que incorporara tanto la programación funcional como la programación imperativa. (14)

Características:

- Existe diferencia entre mayúsculas y minúsculas
- Múltiples expresiones por líneas, separadas por punto y coma “;”
- Dispone de manejo de excepciones
- Ruby puede cargar librerías de extensiones dinámicamente si el (Sistema Operativo) lo permite
- Portátil (12)

Ventajas:

- Permite desarrollar soluciones a bajo costo
- Software libre
- Multiplataforma (12)

PHP

Es un lenguaje de programación utilizado para la creación de sitios web. PHP es un acrónimo recursivo que significa “Hypertext Pre-processor”. (12)

Constituye un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión .php. (12)

Ventajas:

- Muy fácil de aprender.
- Se caracteriza por ser un lenguaje muy rápido.
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, etcétera.

- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables. (12)

Desventajas:

- Se necesita instalar un servidor web.
- Todo el trabajo lo realiza el servidor y no delega al cliente, por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- La programación orientada a objetos es aún muy deficiente para aplicaciones grandes. (12)

Lenguaje Unificado de Modelado

Lenguaje Unificado de Modelado (UML) es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. (15)

El UML es un lenguaje de representación visual en el cual se combinan diversos elementos gráficos para crear diagramas. Prescribe una notación estándar y semánticas esenciales para el modelado de un sistema orientado a objetos. Previamente, un diseño orientado a objetos podría haber sido modelado con cualquiera de la docena de metodologías populares, causando a los revisores tener que aprender las semánticas y notaciones de la metodología empleada antes que intentar entender el diseño en sí. Ahora con UML, diseñadores diferentes modelando sistemas diferentes pueden sobradamente entender cada uno los diseños de los otros. Aun así, no prescribe un proceso o método estándar para desarrollar un sistema. (16)

1.4.3 Herramientas CASE

CASE es una sigla, que corresponde a las iniciales de: Computer Aided Software Engineering; y en su traducción al Español significa Ingeniería de Software Asistida por Computación. Se puede definir a las Herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y

desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un Software.
(17)

Algunas de las herramientas CASE más populares son:

Rational Rose Enterprise

Considerado el producto más completo de la familia Rational Rose. Soporta UML. Genera código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic. Soporta Enterprise Java Beans 2.0. (18)

Algunas características:

- Generación de código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo- código configurables.
- Característica de control por separado de componentes, modelo que permite una administración más granular y el uso de modelos.
- Capacidad de análisis de calidad de código.
- Modelado UML para trabajar en diseños de base de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.
- Integración con otras herramientas de desarrollo de Rational.
- Capacidad para integrarse con cualquier sistema de control de versiones SCC-compliant, incluyendo a Rational ClearCase.
- Se puede usar en plataformas Windows, Linux y Unix.

Desventajas:

- No es un software libre.
- Su entorno gráfico no es muy amigable para el usuario.
- Necesidad de alta capacidad de procesamiento.

Visual Paradigm

Herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Soporta todos los diagramas UML y el Diagrama Entidad-Relación ayudando a una rápida construcción de aplicaciones con calidad y a un menor coste. Posee una interfaz amigable y se puede modelar en varios idiomas. Es una herramienta colaborativa, pues

soporta múltiples usuarios trabajando sobre el mismo proyecto. La documentación del proyecto puede ser generada automáticamente en varios formatos (Web o .pdf), y permite control de versiones.

Soporta la notación UML 2.1, ingeniería inversa, generador de informes, editor de figuras, integración con MS Visio, plug-in, integración IDE con Visual Studio, IntelliJ IDEA, Eclipse, NetBeans y otros.

Otras Ventajas:

- Apoya todo lo básico en cuanto a artefactos generados en las etapas de definición de requerimientos y de especificación de componentes.
- Tiene disponibilidad en múltiples plataformas: Microsoft Windows (98, 2000, XP, o Vista), Linux, Mac OS X, Solaris o Java.
- Brinda la posibilidad de intercambiar información mediante la importación y exportación de ficheros con aplicaciones como por ejemplo Visio y Rational Rose.
- Brinda la posibilidad de generar código a partir de los diagramas, para plataformas como .Net, Java y PHP, así como obtener diagramas a partir del código.
- Brinda la posibilidad de documentar todo el trabajo sin necesidad de utilizar herramientas externas.

Desventajas:

- Las imágenes y reportes generados, no son de alta calidad.

1.4.4 Arquitectura y patrones

La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución. (19)

Los patrones de arquitectura son un modelo que proporciona reglas y guías para organizar un sistema de software. Ejemplos de ellos son:

Arquitectura en Capas

La arquitectura basada en capas se enfoca en la distribución de roles y responsabilidades de forma jerárquica proporcionando una forma muy efectiva de separación de responsabilidades. El rol indica el modo y tipo de interacción con otras capas, y la responsabilidad indica la funcionalidad que está siendo desarrollada. (20)

Tiene como beneficios la mejora de la escalabilidad, disponibilidad, administración, y utilización de recursos.

La arquitectura en 3 capas es diferente de la tradicional arquitectura cliente/servidor de manera que la aplicación en lugar de residir en el equipo del cliente, reside en la capa-media, es decir, en el servidor de aplicación. Esto le permite a un usuario acceder a la aplicación desde el extremo del cliente, simplemente usando un navegador.

Las 3 capas son:

- Capa de Presentación (o interfaz del usuario): es la interfaz con el usuario y consiste de hardware como PC o estación de trabajo y un navegador. Puede constar de cualquier número de máquinas cliente.
- Capa Lógica de Funcionalidad / Negocio: proporciona la funcionalidad al extremo de los usuarios y contiene la lógica del negocio (aplicación). Proporciona el puente entre la primera y la tercera capa. Dependiendo del número de puntos y del número de usuarios, cualquier número de servidores de aplicación pueden colocarse en esta capa.
- Capa de Datos: incluye la base de datos que contiene todos los datos de la organización y ésta se encapsula desde los usuarios finales. Cualquier número de servidores de base de datos puede colocarse en esta capa, dependiendo del volumen de transacciones y de la cantidad de datos.

Algunas ventajas:

- Clientes livianos: la lógica del negocio reside en la capa media y en el cliente solo debe instalarse el navegador.
- Capacidad de migración de la BD: pueden realizarse reestructuración, migración de actualizaciones u otros cambios en la BD sin afectar sustancialmente los programas del cliente.
- Independencia de la plataforma: las máquinas del cliente pueden estar en plataformas diferentes. Así esto no exige tener una versión de plataforma específica de software a cada cliente.

- Facilidad de actualización: permite actualizaciones muy fáciles de las versiones más nuevas del producto.

Arquitectura Orientada a Servicios

Arquitectura Orientada a Servicios (SOA) es una estrategia tecnológica por la cual las aplicaciones hacen uso de los servicios disponibles en una red. Implementar una arquitectura orientada a servicios comprende el desarrollo de aplicaciones que usen los servicios, aplicaciones disponibles como servicios para otras o ambas situaciones. (21)

Es un patrón arquitectónico que debe su éxito a la alineación de los procesos de negocio con los sistemas de infraestructura. Separa la lógica de los procesos, de los sistemas base. Su implementación debe hacerse planeada.

Patrón Modelo Vista Controlador

El patrón Modelo Vista Controlador (MVC) propone separar los módulos de entrada, procesamiento y salida de datos en una aplicación, donde cada uno va a tener sus funcionalidades específicas.

Un modelo puede tener diversas vistas, cada una con su correspondiente controlador.

- El modelo es responsable de:
 - Acceder a la capa de almacenamiento de datos.
 - Define las reglas de negocio (la funcionalidad del sistema).
 - Lleva un registro de las vistas y controladores del sistema.
- Las vistas son responsables de:
 - Recibir datos del modelo.
 - Tienen un registro de su controlador asociado.
 - Pueden dar el servicio de actualización, para que sea invocado por el controlador o por el modelo.
- El controlador es responsable de:
 - Recibir los eventos de entrada (un clic, un cambio en un campo de texto, etcétera).
 - Contener reglas de gestión de eventos. (22)

1.4.5 Entorno de Desarrollo Integrado

Un Entorno de Desarrollo Integrado o en inglés Integrated Development Environment (IDE) es un programa compuesto por un conjunto de herramientas para un programador. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios. Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica GUI. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. El lenguaje Visual Basic por ejemplo, puede ser usado dentro de las aplicaciones de Microsoft Office, lo que hace posible escribir sentencias Visual Basic en forma de macros para Microsoft Word. (23)

NetBeans IDE

Es un entorno de desarrollo, una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeansIDE. (24)

Es un producto de código abierto, que dispone de forma gratuita todos los beneficios del software, lo cual ha sido examinado por una comunidad de desarrolladores y esto ha permitido una mayor capacidad de uso, con cada nueva versión, y ha proporcionado a los desarrolladores mayor flexibilidad, al modificar el IDE, si así lo desean.

Con NetBeans IDE se pueden desarrollar aplicaciones de escritorio, para la web y para teléfonos celulares, utilizando lenguajes de programación como Java, C/C++, Ruby on Rails, PHP, Python, JavaScript.

Características:

- Mejoras en el editor de código, completa el código más inteligentemente utilizando la técnica del resaltado.
- Instalación y actualización simple.
- Características visuales para el desarrollo web.
- Soporte para PHP.
- Mejoras para SOA y UML.

- Aplicaciones para celulares.

Permite ejecutar todas las tareas del framework Symfony y visualizar los resultados sin necesidad de una consola externa. Brinda un buen soporte a JavaScript y completamiento de código HTML y CSS.

1.4.6 Sistemas Gestores de Bases de Datos

Un Sistema Gestor de Bases de Datos (SGBD) es un intermediario entre los usuarios y las aplicaciones. Permite a los usuarios definir, crear y mantener la BD y proporciona un acceso controlado a la misma. Debe prestar los siguientes servicios:

- Definición y creación de la base de datos.
- Manipulación de los datos realizando consultas, inserciones y actualizaciones.
- Acceso controlado a los datos mediante mecanismos de seguridad de acceso a los usuarios.
- Mantener la integridad de los datos.
- Controlar la concurrencia a la base de datos.
- Mecanismos de copias de respaldo y recuperación para restablecer la información en caso de fallos en el sistema. (25)

Oracle

Es un Sistema de Gestión de Bases de Datos Relacional (SGBDR). Entre sus principales características se encuentran:

- Mecanismos de seguridad: acceso a los datos según privilegios concedidos por el administrador.
- Copia de seguridad y recuperación: sofisticados procedimientos para hacer copias de seguridad y recuperar datos.
- Gestión del espacio: se puede asignar espacio en disco para almacenar datos y controlar ese espacio.
- Conectividad abierta: es posible acceder a datos de Oracle usando software de otros fabricantes.

- Herramientas de desarrollo: el motor de la base de datos, Oracle Server, admite una amplia gama de herramientas de desarrollo de aplicaciones: de consulta para el usuario y de gestión de la información. (26)

Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que sólo se vea en empresas muy grandes y multinacionales, por norma general.

PostgreSQL

Es un poderoso SGBDR de código abierto. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación de confiabilidad, integridad de datos y corrección. Funciona en los principales sistemas operativos, incluyendo Linux, UNIX y Windows. Tiene soporte completo para claves foráneas, uniones, vistas, disparadores y procedimientos almacenados (en varios idiomas). Se incluye la mayoría de tipos de datos de SQL, incluyendo INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, y TIMESTAMP. También es compatible con el almacenamiento de objetos binarios, incluyendo imágenes, sonidos o vídeo. Tiene interfaces de programación nativo de C / C ++, Java, NET, Perl, Python, Ruby, Tcl, entre otros, y la documentación excepcional. (27)

MySQL

Es uno de los software de base de datos más populares, es de código abierto, cuenta con más de 100 millones de copias descargadas o distribuidas desde su surgimiento. (28)

Principales características:

- Probado con un amplio rango de compiladores diferentes.
- Funciona en diferentes plataformas.
- Usa GNU Automake y Libtool para portabilidad.
- Proporciona sistemas de almacenamiento transaccional y no transaccional.
- Usa tablas en disco B-tree (MyISAM) muy rápidas con compresión de índice.
- Un sistema de privilegios y contraseñas que es muy flexible y seguro, que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor.
- Soporte a grandes bases de datos. Es usado MySQL Server con bases de datos que contienen 50 millones de registros.

- El servidor puede proporcionar mensajes de error a los clientes en varios idiomas. MySQL server soporta comandos SQL para chequear, optimizar, y reparar tablas (29)

1.4.7 Servidores web

Un servidor es un tipo de software que realiza ciertas tareas en nombre de los usuarios. El término servidor ahora también se utiliza para referirse al ordenador físico en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos. (30)

Un servidor web es un equipo que ejecuta un software especial. Dicho software "sirve" páginas HTML y archivos asociados cuando un cliente, por ejemplo un explorador web, los solicita. (31)

En este caso el servidor web que se utilizará, después de haber planteado que el lenguaje de programación será PHP y el SGBD MySQL, no cabe duda que sea el **Apache** por la estrecha relación que existe entre ellos. El mismo constituye el más usado a nivel mundial (*Ver Anexo 2*) prácticamente desde su surgimiento hasta la actualidad.

Este servidor es uno de los mayores triunfos del software libre, que tanto gusta a los usuarios de Linux. Es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP 1.1). Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo y con la API de programación de módulos. Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para solución de los mismos.

1.5 Propuesta de solución

Teniendo en cuenta las necesidades existentes en el módulo Ejercicios de la Colección Multisaber en su versión multiplataforma, planteadas anteriormente, y el análisis de las soluciones similares, se propone el desarrollo de una aplicación web multiplataforma, de código abierto, mediante la cual los profesores podrán acceder a través de la red para gestionar los contenidos presentes en los cuestionarios interactivos directamente en la BD, así como la reimplementación del funcionamiento interno del módulo utilizando el

paradigma de Programación Orientada a Objetos. Para llevar a cabo el desarrollo de la aplicación mencionada se seleccionó la metodología RUP, que provee de una guía o pasos a seguir para efectuar un desarrollo exitoso, los lenguajes de programación HTML, JavaScript, por el lado del cliente, y PHP, por el lado del servidor, el lenguaje de modelado orientado a objetos UML, la herramienta CASE Visual Paradigm, el patrón de arquitectura 3 Capas como guía para estructurar la aplicación, el entorno de desarrollo NetBeans que soporta todos los lenguajes antes mencionados, proporcionando el uso de un mismo IDE para el desarrollo de la aplicación en su totalidad, el Sistema Gestor de Bases de Datos MySQL y el servidor web Apache. Esta selección responde en gran medida a las políticas establecidas por el proyecto Multisaber y está basada fundamentalmente en que sean herramientas libres.

1.6 Conclusiones

El desarrollo del presente capítulo demuestra que a pesar de existir varias aplicaciones similares a la que se propone como posible solución, ninguna de ellas da respuestas a todos los problemas identificados y a la necesidad actual, por tanto se implementará dicha aplicación. Se exponen las tecnologías y herramientas seleccionadas para llevar a cabo el desarrollo exitoso de los objetivos trazados.

CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

En el presente capítulo se describe el Modelo de Dominio y sus principales conceptos. Se exponen los requerimientos funcionales y no funcionales, los actores involucrados en el sistema, el Diagrama de Casos de Uso del Sistema y la descripción de cada uno de ellos.

2.2 Descripción del Modelo de Dominio

Debido a la dificultad que existe para determinar el proceso del negocio con fronteras bien establecidas donde se logren ver claramente, quienes son las personas que lo inician, quienes son los beneficiados con cada uno de estos procesos, además quienes son las personas que desarrollan las actividades en cada uno de estos procesos, se propone el desarrollo de un Modelo de Dominio, en el cual se capturan los tipos de objetos más importantes que existen o los eventos que suceden en el entorno donde estará el sistema.

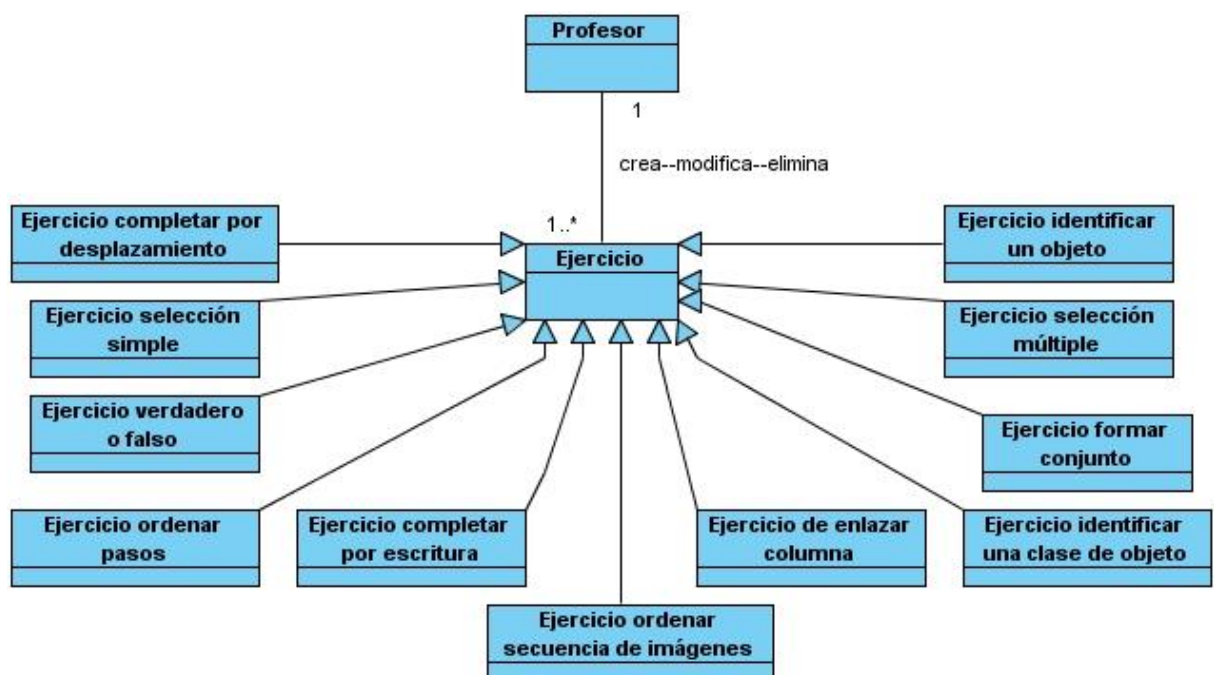


Figura 1. Modelo de Dominio

- Profesor: hace referencia a la persona que inserta, modifica o elimina los ejercicios.
- Ejercicio: es una forma de asignación de tareas. A través de un ejercicio el profesor le indica al estudiante qué actividades debe realizar para comprobar su conocimiento.
- Ejercicio de selección simple: se introducen varias opciones de respuestas, ya sean en forma de texto o en imágenes, de las que solo se debe seleccionar una.
- Ejercicio de selección múltiple: se introducen varias opciones de respuestas, ya sean en forma de texto o en imágenes, de las que se pueden seleccionar una o varias.
- Ejercicio de verdadero o falso: se introducen varias opciones de respuestas donde se debe emitir un criterio de veracidad sobre cada una de ellas.
- Ejercicio de completar por desplazamiento: se introduce un problema o texto base en el cual se seleccionan una o varias opciones a completar, las cuales van a ser excluidas del texto y se van a mostrar dispersas de forma aleatoria, permitiendo que sean desplazadas al espacio correspondiente.
- Ejercicio de completar por escritura: se introduce un problema o texto base en el cual se seleccionan una o varias opciones a completar, las cuales van a ser excluidas del texto, y que luego debe permitirse escribir directamente en el espacio correspondiente.
- Ejercicio de formar conjunto: se introduce el nombre de un grupo de elementos y 2 o 3 contenedores a los cuales deben pertenecer, para que luego se permita arrastrar el elemento a dicho contenedor.
- Ejercicio de ordenar secuencia de imágenes: se introduce el nombre de un grupo de imágenes en el orden correcto, las que luego van a ser mostradas desordenadamente para que el usuario las ordene.
- Ejercicio de enlazar columnas: se introducen dos columnas de texto, donde cada elemento de la columna 1 corresponde con uno y solo un elemento de la columna 2.
- Ejercicio de ordenar pasos: se introduce una serie de pasos o acciones en el orden correcto, las que luego van a ser mostradas desordenadamente para que el usuario las ordene.
- Ejercicio de identificar una clase de objeto: se introduce el nombre de un contenedor, el nombre de un conjunto de imágenes u objetos, y se indica cuáles de éstas pertenecen al contenedor.

- Ejercicio de identificar un objeto: se introduce el nombre de un contenedor, el nombre de un conjunto de imágenes u objetos, y se indica cuál de éstas pertenece al contenedor.

2.3 Especificación de requerimientos

2.3.1 Requisitos funcionales

RF1- Mostrar cantidad de ejercicios

RF1.1 Mostrar la cantidad de ejercicios dada la tipología

RF1.2 Mostrar la cantidad de ejercicios dado el nivel

RF1.3 Mostrar la cantidad de ejercicios dado el tema

RF2- Crear ejercicios de selección simple

RF2.1 Introducir opciones de respuesta

RF2.1.1 Introducir texto como opción de respuesta

RF2.1.2 Subir imagen como opción de respuesta

RF2.2 Introducir una retroalimentación para cada una de las opciones de respuesta

RF2.3 Indicar la opción correcta

RF3- Crear ejercicios de selección múltiple

RF3.1 Introducir opciones de respuestas

RF3.1.1 Introducir texto como opción de respuesta

RF3.1.2 Subir imagen como opción de respuesta

RF3.2 Introducir una retroalimentación para cada opción de respuestas

RF3.3 Indicar las opciones correctas

RF4- Crear ejercicios de verdadero o falso

RF4.1 Introducir opciones de respuestas

RF4.2 Introducir una retroalimentación para cada opción de respuesta

RF4.3 Indicar la opción verdadera

RF5- Crear ejercicios de enlazar columnas

RF5.1 Introducir opciones de respuestas columna 1

RF5.2 Introducir opciones de respuestas columna 2

RF5.3 Indicar la opción de la columna 1 que se corresponde con la columna 2

RF5.4 Introducir retroalimentación a las respuestas correctas

RF5.5 Introducir retroalimentación a las respuestas incorrectas

RF6- Crear ejercicios de formar conjuntos

RF6.1 Introducir nombre que identifica el conjunto

RF6.2 Subir imagen

RF6.3 Indicar las imágenes correctas

RF6.4 Introducir texto de retroalimentación

RF7- Crear ejercicios de completar por escritura

RF7.1 Introducir el texto del ejercicio

RF7.2 Seleccionar la palabra que se va a teclear

RF7.3 Modificar la palabra seleccionada

RF7.4 Eliminar la palabra seleccionada

RF7.5 Introducir texto de retroalimentación

RF8- Crear ejercicios de completar por desplazamiento

RF8.1 Introducir el texto del ejercicio

RF8.2 Seleccionar la palabra que se va a desplazar

RF8.3 Modificar la palabra seleccionada

RF8.4 Eliminar la palabra seleccionada

RF8.5 Introducir texto de retroalimentación

RF9- Crear ejercicios de ordenar secuencia de imágenes

RF9.1 Introducir título que identifica la imagen

RF9.2 Subir imagen en el orden correcto

RF9.3 Introducir texto de retroalimentación

RF10- Modificar ejercicios

RF11- Eliminar ejercicios

RF11.1- Eliminar todos los ejercicios a la vez

RF11.2- Eliminar uno o varios ejercicios seleccionados

RF12- Especificar datos generales de los ejercicios

RF12.1 Especificar el tema

RF12.2 Especificar el enunciado

RF12.3 Especificar el nivel de desempeño

RF12.4 Especificar la cantidad de intentos

RF12.5 Especificar niveles de ayuda

RF12.6 Especificar imágenes de referencia

RF13- Mostrar listado de ejercicios

RF13.1 Mostrar listado de ejercicios dada la tipología

RF13.2 Mostrar listado de ejercicios dado el nivel

RF13.3 Mostrar listado de ejercicios dado el tema

RF14- Mostrar ayuda

RF15- Eliminar selección de ejercicios

RF16- Guardar ejercicio

RF16.1 Se guarda en la BD el ejercicio creado

2.3.2 Requisitos no funcionales

- **Requisitos de hardware**
 - Procesador Pentium 233 MHz (recomendado 500 MHz o mayor).
 - 64 MB de RAM (recomendado 128 MB de RAM o mayor).
 - 1 GB de espacio en disco duro.
 - Soporte de video que admita resolución de al menos 800x600 y 24 bits.
 - Dispositivo de red de al menos 10 Mbits.

- **Requisitos de software**
 - Computadora personal con navegador Mozilla Firefox 3.x.

- **Requisitos de restricciones en el diseño y la implementación**
 - MySQL 5.0
 - Apache 2.x
 - NetBeans como herramienta de desarrollo
 - Lenguajes de programación: PHP 5 y JavaScript

- **Requisitos de apariencia e interfaz externa**
 - El diseño de las interfaces debe ser amigable y sencilla.
 - El sistema proporcionará claridad y correcta organización de la información, permitiendo la interpretación correcta e inequívoca de ésta.

- **Requisitos de soporte**
 - Se impartirán clases a los profesores para explicar el funcionamiento y utilidad del producto.

- **Requisitos de portabilidad**
 - El sistema podrá ser utilizado bajo cualquier Sistema Operativo.

2.4 Modelo de Casos de Uso del Sistema

2.4.1 Actores del sistema

Tabla1: Actores del sistema y su justificación

Actor	Justificación
Profesor	Representa a una persona que va a usar la aplicación para insertar, modificar o eliminar el contenido de los ejercicios.

2.4.2 Diagrama de Casos de Uso del Sistema



Figura 2. Diagrama de Casos de Uso del Sistema

2.4.3 Descripción de Casos de Uso del Sistema

Caso de Uso:	Mostrar ayuda
---------------------	---------------

Actores:	Profesor	
Resumen:	El caso de uso se inicia cuando el profesor solicita ver la ayuda correspondiente a la página en la que se encuentra.	
Precondiciones		
Referencias	RF14	
Prioridad	Secundario	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1.- El caso de uso inicia cuando el profesor selecciona el botón Ayuda.	2.- El sistema muestra la ayuda de la página en la que se encuentra.	
Poscondiciones		

Caso de Uso:	Mostrar listado de ejercicios	
Actores:	Profesor	
Resumen:	El caso de uso se inicia cuando el profesor entra a la aplicación.	
Precondiciones		
Referencias	RF13, RF13.1, RF13.2, RF13.3	
Prioridad	Secundario	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1.- El caso de uso inicia cuando el profesor entra a la aplicación.	2.- El sistema muestra un listado con todos los ejercicios y brinda la posibilidad de escoger el criterio por el cual mostrar el listado de ejercicios. <ul style="list-style-type: none"> - Mostrar listado de ejercicios por tipología: ver Sección 1 - Mostrar listado de ejercicios por tema: ver Sección 2 - Mostrar listado de ejercicios por módulo: ver Sección 3 	
Sección 1 “Mostrar listado de ejercicios por tipología”: Flujo Normal de los Eventos		
Acción del Actor	Respuesta del Sistema	

1.- El usuario selecciona la tipología de la cual desea ver los ejercicios	2.- Muestra los ejercicios de la tipología seleccionada
Sección 2 “Mostrar listado de ejercicios por tema”: Flujo Normal de los Eventos	
Acción del Actor	Respuesta del Sistema
1.- El usuario selecciona el tema del cual desea ver los ejercicios	2.- Muestra los ejercicios del tema seleccionado
Sección 3 “Mostrar listado de ejercicios por módulo”: Flujo Normal de los Eventos	
Acción del Actor	Respuesta del Sistema
1.- El usuario selecciona el módulo del cual desea ver los ejercicios	2.- Muestra los ejercicios del módulo seleccionado
Poscondiciones	

Caso de Uso:	Eliminar selección de ejercicios
Actores:	Profesor
Resumen:	El caso de uso se inicia cuando el profesor solicita eliminar los ejercicios seleccionados.
Precondiciones	
Referencias	RF15
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1.- El caso de uso inicia cuando el profesor selecciona el o los ejercicios que desea eliminar y escoge la opción eliminar.	2.-El sistema muestra un mensaje de confirmación.
3.-El usuario confirma eliminar el o los ejercicios.	4.- El sistema elimina el o los ejercicios.
Poscondiciones	

Caso de Uso:	Gestionar ejercicios de selección
Actores:	Profesor
Resumen:	El CU se inicia cuando el profesor desea realizar alguna de las

	siguientes acciones:	
	<ul style="list-style-type: none"> - Crear ejercicios de selección - Modificar ejercicio - Eliminar ejercicio 	
Precondiciones		
Referencias	RF2, RF2.1, RF2.2, RF2.3, RF3, RF3.1, RF3.2, RF3.3, RF4, RF4.1, RF4.2, RF4.3, RF10, RF11, RF11.1, RF11.2, RF14, RF16, RF16.1 CU Especificar datos generales (inclusión)	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1.- El caso de uso inicia cuando el profesor selecciona la acción que desea realizar: <ul style="list-style-type: none"> - Crear ejercicios de selección - Modificar ejercicio - Eliminar ejercicio 	2.- El sistema, en dependencia de la acción solicitada por el profesor, muestra la interfaz correspondiente: <ul style="list-style-type: none"> - Crear ejercicios de selección: ir a la Sección 1. - Modificar ejercicio: ir a la Sección 2. - Eliminar ejercicio: ir a la Sección 3. 	
Sección 1 “Crear ejercicio de selección”: Flujo Normal de los Eventos		
Acción del Actor	Respuesta del Sistema	
1.- El usuario selecciona la opción nuevo ejercicio de selección	2.- El sistema muestra una interfaz para crear un nuevo ejercicio de selección	
3.- El profesor introduce los datos del nuevo ejercicio: <ul style="list-style-type: none"> - Opciones de respuesta - Retroalimentación para cada una de las opciones de respuesta - Indica la opción correcta - Se instancia el caso de uso incluido Especificar datos generales		
4.- El profesor indica al sistema crear el nuevo ejercicio	5.- El sistema verifica que ningún campo haya quedado vacío.	
	6.- El sistema registra el nuevo ejercicio.	

Sección 1 “Crear ejercicio de selección”: Flujo Alterno	
Acción del Actor	Respuesta del Sistema
3.1- Si el ejercicio es de selección múltiple o de verdadero o falso, el profesor indica varias opciones correctas	4.1- Si quedó algún campo vacío el sistema emite un mensaje informándole al usuario que debe llenar los campos correctamente
Sección 2 “Modificar ejercicio”: Flujo Normal de los Eventos	
Acción del Actor	Respuesta del Sistema
1.- El usuario selecciona la opción modificar ejercicio	2.- El sistema muestra una interfaz para modificar el ejercicio de selección
3.- El profesor modifica los datos que desee	
4.- Indica al sistema guardar los datos	5.- El sistema verifica que ningún campo haya quedado vacío
	6.- El sistema actualiza el ejercicio.
Sección 2“Modificar ejercicio”: Flujo Alterno	
	5.1- Si quedó algún campo vacío el sistema emite un mensaje informándole al usuario que debe llenar los campos correctamente
Sección 3“Eliminar ejercicio”: Flujo Normal de los Eventos	
1.- El usuario selecciona la opción eliminar ejercicio	2.- El sistema muestra mensaje de confirmación
3.- Confirma eliminar el ejercicio	4.- El sistema elimina el ejercicio
Poscondiciones	

Caso de Uso:	Gestionar ejercicios de enlazar columnas
Actores:	Profesor
Resumen:	El CU se inicia cuando el profesor desea realizar alguna de las siguientes acciones: <ul style="list-style-type: none"> - Crear ejercicios de enlazar columnas - Modificar ejercicio - Eliminar ejercicio
Precondiciones	

Referencias	RF5, RF5.1, RF5.2, RF5.3, RF5.4, RF5.5, RF10, RF11, RF11.1, RF11.2, RF14, RF16, RF16.1 CU Especificar datos generales (inclusión)
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1.- El caso de uso inicia cuando el profesor selecciona la acción que desea realizar: <ul style="list-style-type: none"> - Crear ejercicios de enlazar columnas - Modificar ejercicio - Eliminar ejercicio 	2.- El sistema, en dependencia de la acción solicitada por el profesor, muestra la interfaz correspondiente: <ul style="list-style-type: none"> - Crear ejercicios de enlazar columnas: ir a la Sección 1. - Modificar ejercicio: ir a la Sección 2. - Eliminar ejercicio: ir a la Sección 3.
Sección 1 “Crear ejercicio de enlazar columnas”: Flujo Normal de los Eventos	
Acción del Actor	Respuesta del Sistema
1.- El usuario selecciona la opción nuevo ejercicio de enlazar columna	2.- El sistema muestra una interfaz para crear un nuevo ejercicio de enlazar columnas
3.- El profesor introduce los datos del nuevo ejercicio: <ul style="list-style-type: none"> - Opciones de respuestas columna 1 - Opciones de respuestas columna 2 - Retroalimentación a las respuestas correctas - Retroalimentación a las respuestas incorrectas - Se instancia el caso de uso incluido <p style="text-align: center;">Especificar datos generales</p>	
4.- El profesor indica al sistema crear el nuevo ejercicio	5.- El sistema verifica que ningún campo haya quedado vacío
	6.- El sistema registra el nuevo ejercicio
Sección 1 “Crear ejercicio de enlazar columnas”: Flujo Alterno	
Acción del Actor	Respuesta del Sistema

	5.1- Si quedó algún campo vacío el sistema emite un mensaje informándole al usuario que debe llenar los campos correctamente
Sección 2 “Modificar ejercicio”: Flujo Normal de los Eventos	
Acción del Actor	Respuesta del Sistema
1.- El usuario selecciona la opción modificar ejercicio	2.- El sistema muestra una interfaz para modificar el ejercicio de enlazar columna
3.- El profesor modifica los datos que desee	
4.- Indica al sistema guardar los datos	5.- El sistema verifica que ningún campo haya quedado vacío
	6.- El sistema actualiza el ejercicio
Sección 2 “Modificar ejercicio”: Flujo Alterno	
	5.1- Si quedó algún campo vacío el sistema emite un mensaje informándole al usuario que debe llenar los campos correctamente
Sección 3 “Eliminar ejercicio”: Flujo Normal de los Eventos	
1.- El usuario selecciona la opción eliminar ejercicio	2.- El sistema muestra mensaje de confirmación
3.- Confirma eliminar el ejercicio	4.- El sistema elimina el ejercicio
Poscondiciones	

Caso de Uso:	Gestionar ejercicios de formar conjuntos
Actores:	Profesor
Resumen:	El CU se inicia cuando el profesor desea realizar alguna de las siguientes acciones: <ul style="list-style-type: none"> - Crear ejercicios de formar conjuntos - Modificar ejercicio - Eliminar ejercicio
Precondiciones	
Referencias	RF6, RF6.1, RF6.2, RF6.3, RF6.4, RF10, RF11, RF11.1, RF11.2, RF14, RF16, RF16.1

	CU Especificar datos generales (inclusión)
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1.- El caso de uso inicia cuando el profesor selecciona la acción que desea realizar: <ul style="list-style-type: none"> - Crear ejercicios de formar conjuntos - Modificar ejercicio - Eliminar ejercicio 	2.- El sistema, en dependencia de la acción solicitada por el profesor, muestra la interfaz correspondiente: <ul style="list-style-type: none"> - Crear ejercicios de formar conjuntos: ir a la Sección 1. - Modificar ejercicio: ir a la Sección 2. - Eliminar ejercicio: ir a la Sección 3.
Sección 1 “Crear ejercicio de formar conjuntos”: Flujo Normal de los Eventos	
Acción del Actor	Respuesta del Sistema
1.- El usuario selecciona la opción nuevo ejercicio de formar conjunto	2.- El sistema muestra una interfaz para crear un nuevo ejercicio de formar conjuntos
3.- El profesor introduce los datos del nuevo ejercicio: <ul style="list-style-type: none"> - Nombre que identifica cada conjunto - Sube imagen - Texto de retroalimentación - Se instancia el caso de uso incluido Especificar datos generales	
4.- El profesor indica al sistema crear el nuevo ejercicio	5.- El sistema verifica que ningún campo haya quedado vacío
	6.- El sistema registra el nuevo ejercicio
Sección 1 “Crear ejercicio de formar conjuntos”: Flujo Alterno	
Acción del Actor	Respuesta del Sistema
	5.1- Si quedó algún campo vacío el sistema emite un mensaje informándole al usuario que debe llenar los campos correctamente
Sección 2 “Modificar ejercicio”: Flujo Normal de los Eventos	

Acción del Actor	Respuesta del Sistema
1.- El usuario selecciona la opción modificar ejercicio	2.- El sistema muestra una interfaz para modificar el ejercicio de formar columnas
3.- El profesor modifica los datos que desee	
4.- Indica al sistema guardar los datos	5.- El sistema verifica que ningún campo haya quedado vacío
	6.- El sistema actualiza el ejercicio
Sección 2 “Modificar ejercicio”: Flujo Alterno	
	5.1- Si quedó algún campo vacío el sistema emite un mensaje informándole al usuario que debe llenar los campos correctamente
Sección 3 “Eliminar ejercicio”: Flujo Normal de los Eventos	
1.- El usuario selecciona la opción eliminar ejercicio	2.- El sistema muestra mensaje de confirmación
3.- Confirma eliminar el ejercicio	4.- El sistema elimina el ejercicio
Poscondiciones	

Caso de Uso:	Gestionar ejercicios de completar
Actores:	Profesor
Resumen:	El CU se inicia cuando el profesor desea realizar alguna de las siguientes acciones: <ul style="list-style-type: none"> - Crear ejercicios de completar - Modificar ejercicio - Eliminar ejercicio
Precondiciones	
Referencias	RF7, RF7.1, RF7.2, RF7.3, RF7.4, RF7.5, RF10, RF11, RF11.1, RF11.2, RF14, RF16, RF16.1 CU Especificar datos generales (inclusión)
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1.- El caso de uso inicia cuando el profesor	2.- El sistema, en dependencia de la acción

selecciona la acción que desea realizar: <ul style="list-style-type: none"> - Crear ejercicios de completar - Modificar ejercicio - Eliminar ejercicio 	solicitada por el profesor, muestra la interfaz correspondiente: <ul style="list-style-type: none"> - Crear ejercicios de completar: ir a la Sección 1. - Modificar ejercicio: ir a la Sección 2. - Eliminar ejercicio: ir a la Sección 3.
---	--

Sección 1 “Crear ejercicio de completar”: Flujo Normal de los Eventos

Acción del Actor	Respuesta del Sistema
1.- El usuario selecciona la opción nuevo ejercicio de completar	2.- El sistema muestra una interfaz para crear un nuevo ejercicio de completar
3.- El profesor introduce los datos del nuevo ejercicio: <ul style="list-style-type: none"> - Texto del ejercicio - Selecciona la palabra que se va a teclear - Introducir texto de retroalimentación - Introduce una opción de respuesta incorrecta que pudiera dar el usuario - Introduce una retroalimentación para la palabra incorrecta - Se instancia el caso de uso incluido <p align="center">Especificar datos generales</p>	
4.- El profesor indica al sistema crear el nuevo ejercicio	5.- El sistema verifica que ningún campo haya quedado vacío
	6.- El sistema registra el nuevo ejercicio

Sección 1 “Crear ejercicio de completar”: Flujo Alterno

Acción del Actor	Respuesta del Sistema
	5.1- Si quedó algún campo vacío el sistema emite un mensaje informándole al usuario que debe llenar los campos correctamente

Sección 2 “Modificar ejercicio”: Flujo Normal de los Eventos

Acción del Actor	Respuesta del Sistema
-------------------------	------------------------------

1.- El usuario selecciona la opción modificar ejercicio	2.- El sistema muestra una interfaz para modificar el ejercicio de completar
3.- El profesor modifica los datos que desee	
4.- Indica al sistema guardar los datos	5.- El sistema verifica que ningún campo haya quedado vacío
	6.- El sistema actualiza el ejercicio
Sección 2 “Modificar ejercicio”: Flujo Alterno	
	5.1- Si quedó algún campo vacío el sistema emite un mensaje informándole al usuario que debe llenar los campos correctamente
Sección 3 “Eliminar ejercicio”: Flujo Normal de los Eventos	
1.- El usuario selecciona la opción eliminar ejercicio	2.- El sistema muestra mensaje de confirmación
3.- Confirma eliminar el ejercicio	4.- El sistema elimina el ejercicio
Poscondiciones	

Caso de Uso:	Gestionar ejercicios de ordenar secuencias
Actores:	Profesor
Resumen:	El CU se inicia cuando el profesor desea realizar alguna de las siguientes acciones: <ul style="list-style-type: none"> - Crear ejercicios de ordenar secuencias - Modificar ejercicio - Eliminar ejercicio
Precondiciones	
Referencias	RF9, RF9.1, RF9.2, RF9.3, RF10, RF11, RF11.1, RF11.2, RF14, RF16, RF16.1 CU Especificar datos generales (inclusión)
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1.- El caso de uso inicia cuando el profesor selecciona la acción que desea realizar:	2.- El sistema, en dependencia de la acción solicitada por el profesor, muestra la

<ul style="list-style-type: none"> - Crear ejercicios de ordenar secuencias - Modificar ejercicio - Eliminar ejercicio 	interfaz correspondiente: <ul style="list-style-type: none"> - Crear ejercicios de ordenar secuencias: ir a la Sección 1. - Modificar ejercicio: ir a la Sección 2. - Eliminar ejercicio: ir a la Sección 3.
Sección 1 “Crear ejercicio de ordenar secuencias”: Flujo Normal de los Eventos	
Acción del Actor	Respuesta del Sistema
1.- El usuario selecciona la opción nuevo ejercicio de ordenar secuencias	2.- El sistema muestra una interfaz para crear un nuevo ejercicio de ordenar secuencias
3.- El profesor introduce los datos del nuevo ejercicio: <ul style="list-style-type: none"> - Título que identifica la imagen - Subir imagen en el orden correcto - Texto de retroalimentación - Se instancia el caso de uso incluido <p style="text-align: center;">Especificar datos generales</p>	
4.- El profesor indica al sistema crear el nuevo ejercicio	5.- El sistema verifica que ningún campo haya quedado vacío
	6.- El sistema registra el nuevo ejercicio
Sección 1 “Crear ejercicio de ordenar secuencias”: Flujo Alterno	
Acción del Actor	Respuesta del Sistema
3.1.- Si el ejercicio es de ordenar pasos, el profesor introduce los datos del nuevo ejercicio: <ul style="list-style-type: none"> - Texto en el orden correcto - Texto de retroalimentación - Se instancia el caso de uso incluido <p style="text-align: center;">Especificar datos generales</p>	
	5.1- Si quedó algún campo vacío el sistema emite un mensaje informándole al usuario que debe llenar los campos correctamente
Sección 2 “Modificar ejercicio”: Flujo Normal de los Eventos	

Acción del Actor		Respuesta del Sistema
1.- El usuario selecciona la opción modificar ejercicio		2.- El sistema muestra una interfaz para modificar el ejercicio de ordenar secuencias
3.- El profesor modifica los datos que desee		
4.- Indica al sistema guardar los datos		5.- El sistema verifica que ningún campo haya quedado vacío
		6.- El sistema actualiza el ejercicio
Sección 2 “Modificar ejercicio”: Flujo Alterno		
		5.1- Si quedó algún campo vacío el sistema emite un mensaje informándole al usuario que debe llenar los campos correctamente
Sección 3 “Eliminar ejercicio”: Flujo Normal de los Eventos		
1.- El usuario selecciona la opción eliminar ejercicio		2.- El sistema muestra mensaje de confirmación
3.- Confirma eliminar el ejercicio		4.- El sistema elimina el ejercicio
Poscondiciones		

Caso de Uso:	Especificar datos generales
Actores:	Profesor
Resumen:	El proceso de especificación de datos inicia cuando el profesor escoge crear un ejercicio, entra los datos generales de dicho ejercicio, independientemente de la tipología.
Precondiciones	
Referencias	RF12, RF12.1, RF12.2, RF12.3, RF12.4, RF12.5, RF12.6
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1.- El caso de uso inicia cuando el profesor selecciona la acción de crear un nuevo ejercicio	2.- El sistema muestra la interfaz correspondiente al nuevo ejercicio
3.- El profesor introduce los datos: - Tema	4.- El sistema verifica que ningún campo haya quedado vacío

<ul style="list-style-type: none"> - Nivel de desempeño - Número de intentos - Enunciado - Módulo - Niveles de ayuda - Imágenes de referencia 	
	5.- El sistema registra el nuevo ejercicio
Flujo Alterno	
Acción del Actor	Respuesta del Sistema
3.1.- Si el ejercicio pertenece a las tipologías Selección Simple o Selección Múltiple, el profesor introduce los datos: <ul style="list-style-type: none"> - Tema - Nivel de desempeño - Número de intentos - Enunciado - Tipo de selección - Módulo - Niveles de ayuda - Imágenes de referencia 	4.1- Si quedó algún campo vacío el sistema emite un mensaje informándole al usuario que debe llenar los campos correctamente
Poscondiciones	

2.5 Conclusiones

A través del desarrollo del presente capítulo se tiene una visión más amplia sobre el entorno al cual se integrará el sistema a implementar, así como el comportamiento interno, propiedades o cualidades que debe tener dicho sistema, y la forma en que va a interactuar con los usuarios.

ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción

En el presente capítulo se expone el análisis y diseño del sistema propuesto, modelándose los artefactos necesarios que contribuyen a su implementación. Estos artefactos son Modelo de Análisis, Modelo de Diseño, Diagrama de Clases Persistentes y Modelo de Datos.

3.2 Modelo de Análisis

En el análisis se refinan y estructuran los requisitos obtenidos con anterioridad, de manera que sea más fácil su comprensión, su preparación, su modificación y en general su mantenimiento.

En el Modelo de Análisis están contenidas las clases de análisis, las cuales se centran en los requisitos funcionales. RUP propone clasificar dichas clases en:

- Clase Entidad: Modela información que posee larga vida.
- Clase Interfaz: Modela la interacción entre el sistema y sus actores.
- Clase Control: Coordina la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.

3.2.1 Diagramas de Clases del Análisis

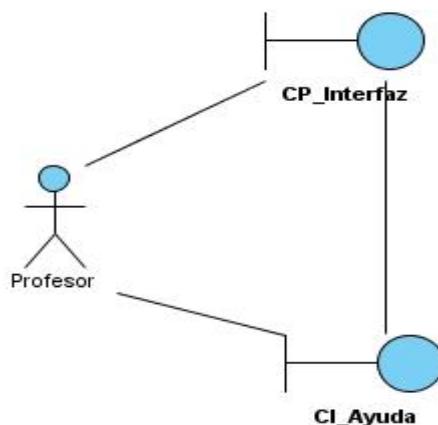


Figura 3. DCA_CU Mostrar ayuda

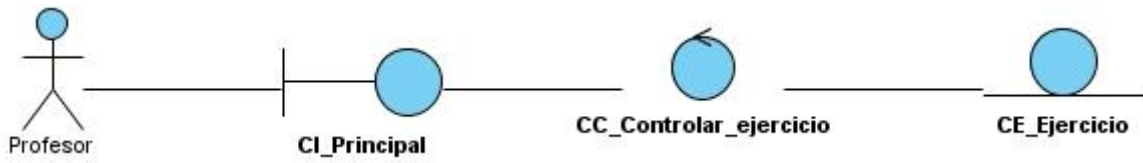


Figura 4. DCA_CU Mostrar listado de ejercicios



Figura 5. DCA_CU Eliminar selección de ejercicios

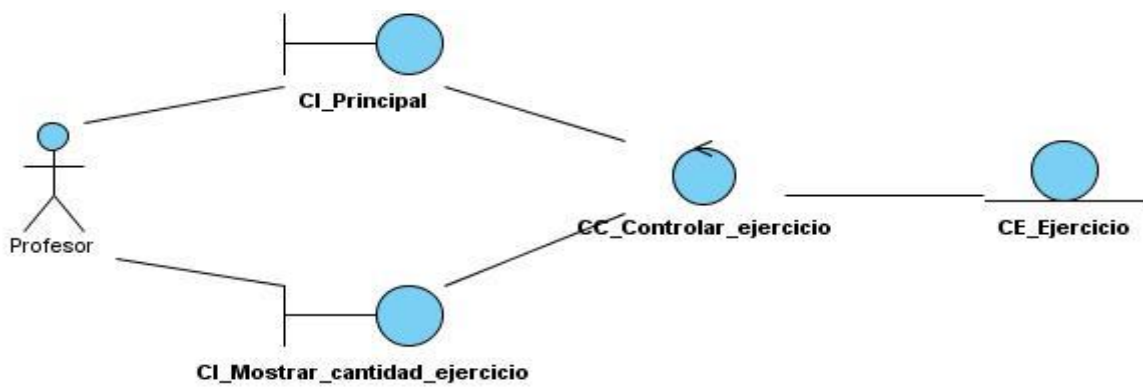


Figura 6. DCA_CU Mostrar cantidad de ejercicios

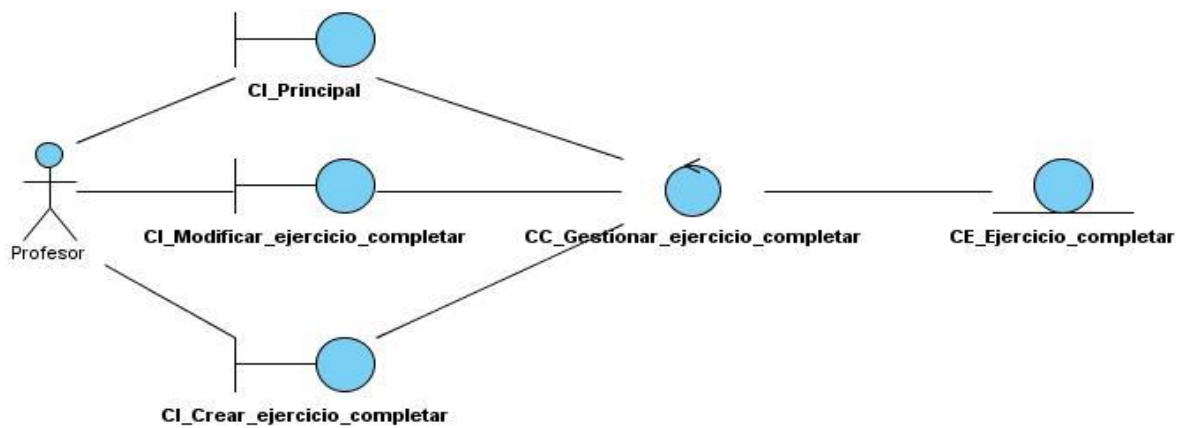


Figura 7. DCA_CU Gestionar ejercicios de completar

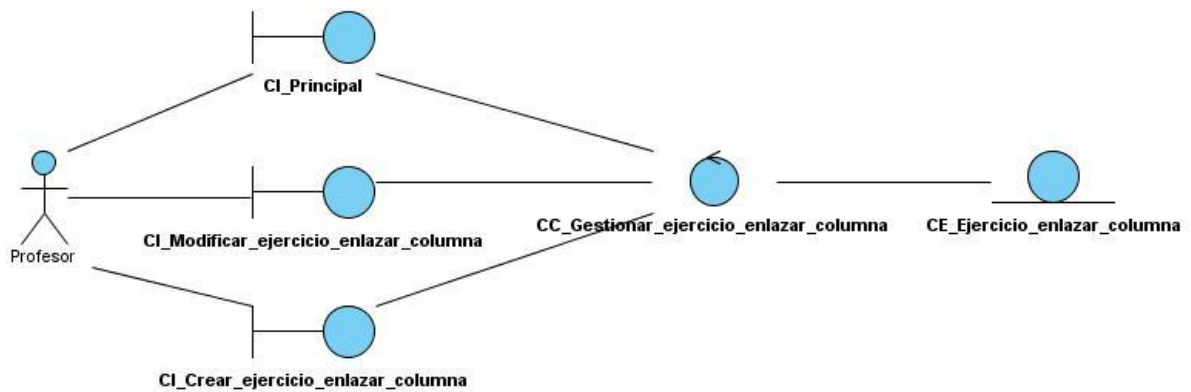


Figura 8. DCA_CU Gestionar ejercicios de enlazar columnas

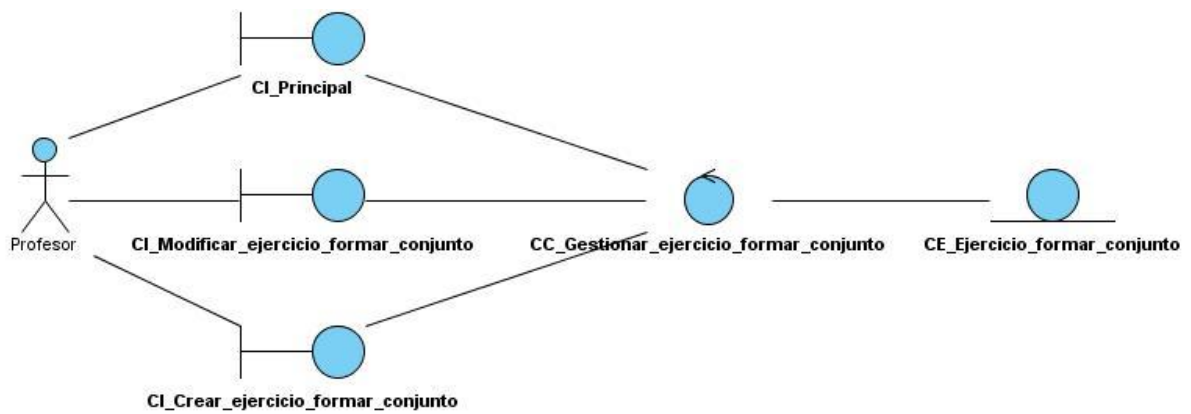


Figura 9. DCA_CU Gestionar ejercicios de formar conjuntos

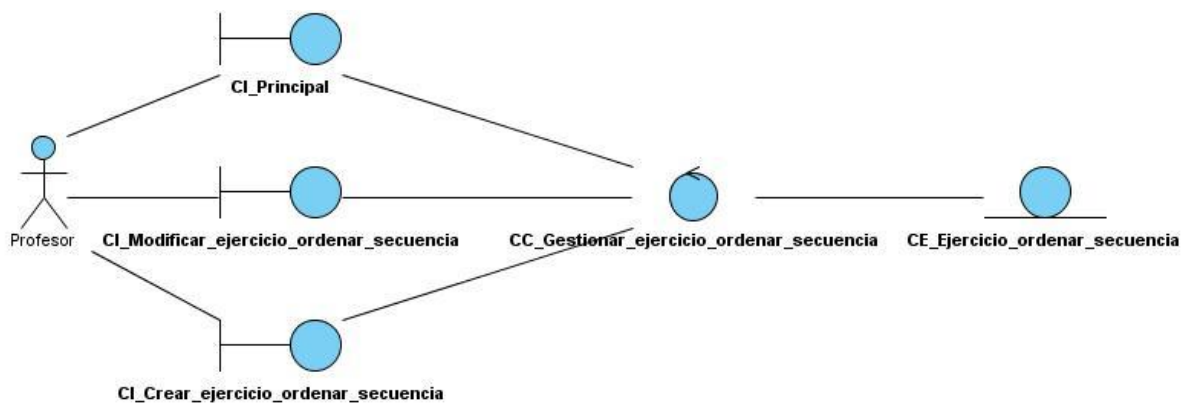


Figura 10. DCA_CU Gestionar ejercicios de ordenar secuencias

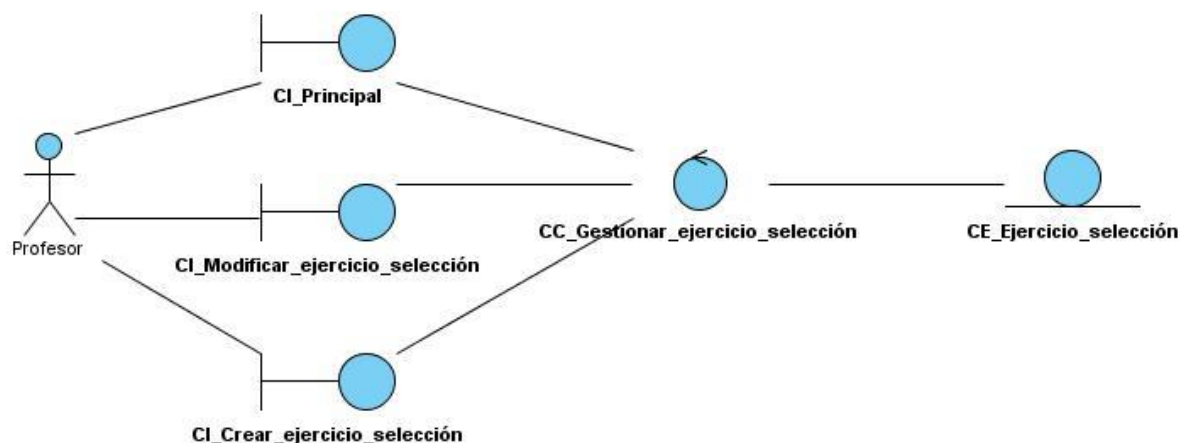


Figura 11. DCA_CU Gestionar ejercicios de selección

3.3 Modelo de Diseño

El diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, es decir, cómo cumple el sistema sus objetivos. Proporciona una completa idea de lo que es el Software, enfocando los dominios de datos, funcional y comportamiento desde el punto de vista de la implementación.

3.3.1 Diagramas de Clases del Diseño

Capas de Presentación y Lógica

La figura 12 muestra el Diagrama de Clases del Diseño correspondiente al caso de uso Mostrar ayuda, en el cual se modela una página cliente “Interfaz”, que representa a todas las páginas interfaz de la aplicación, una página cliente “Ayuda”, que representa a todas las páginas de ayuda correspondientes a cada una de las interfaces, y la relación entre ellas.



Figura 12. DCD_CU Mostrar ayuda

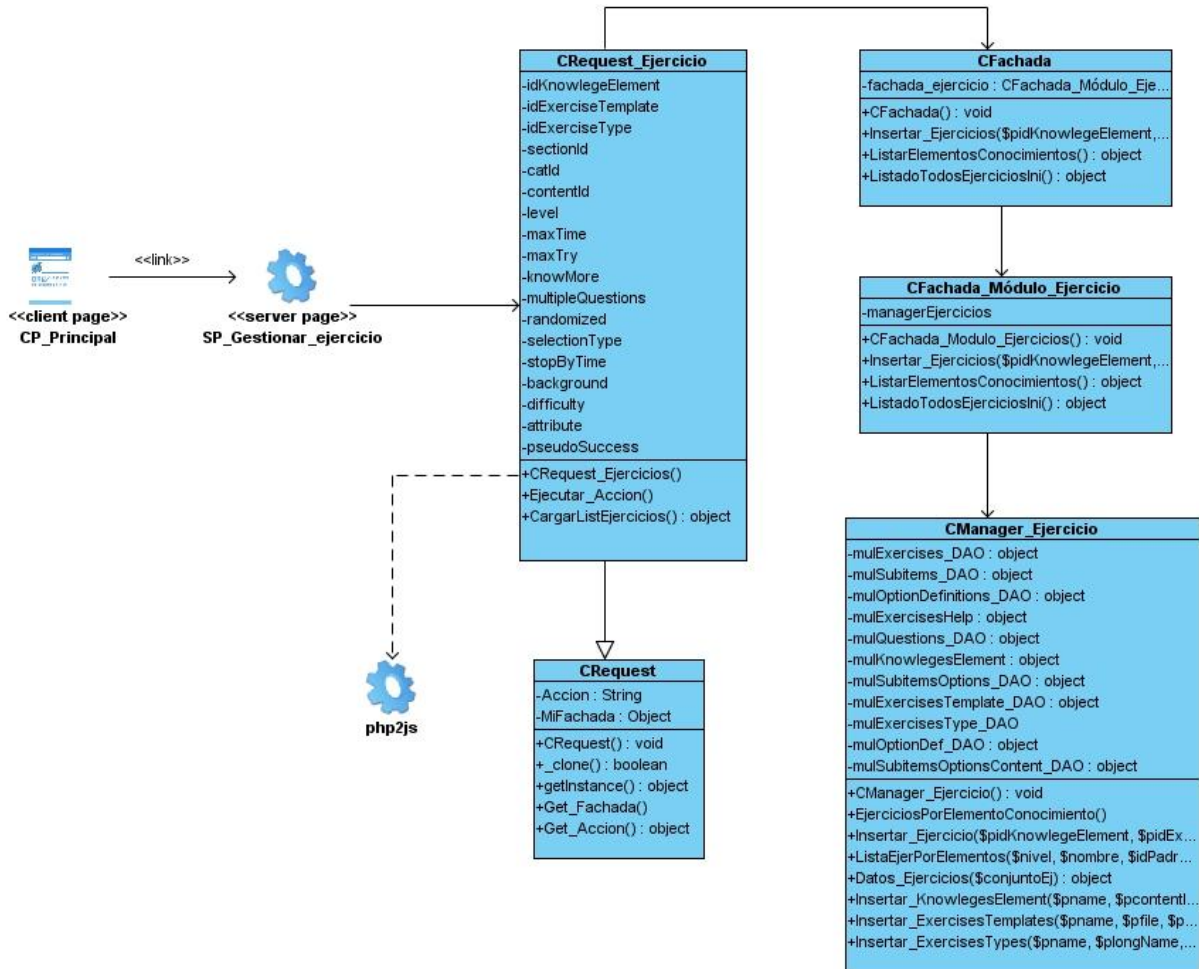


Figura 13. DCD_CU Mostrar listado de ejercicios

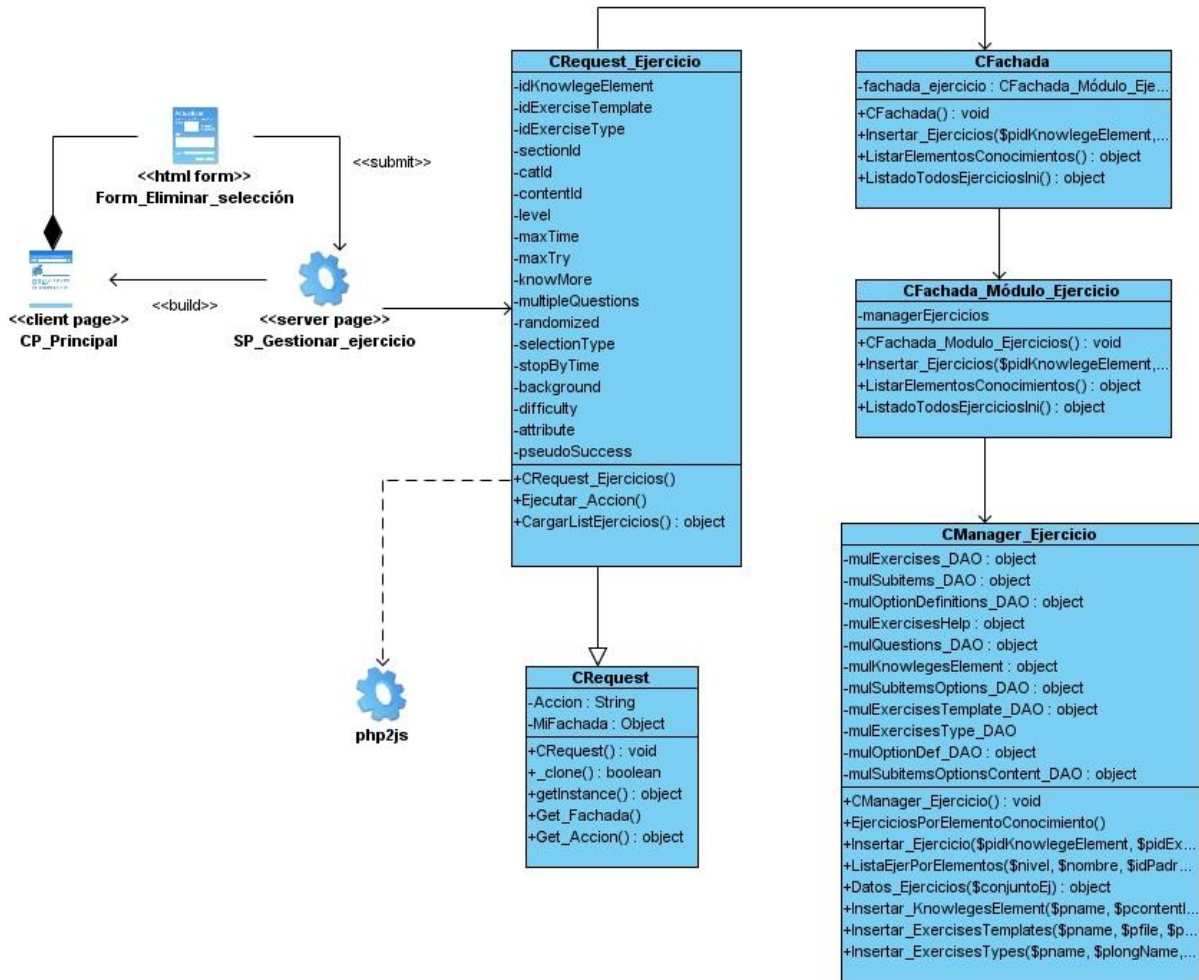


Figura 14. DCD_CU Eliminar selección de ejercicios

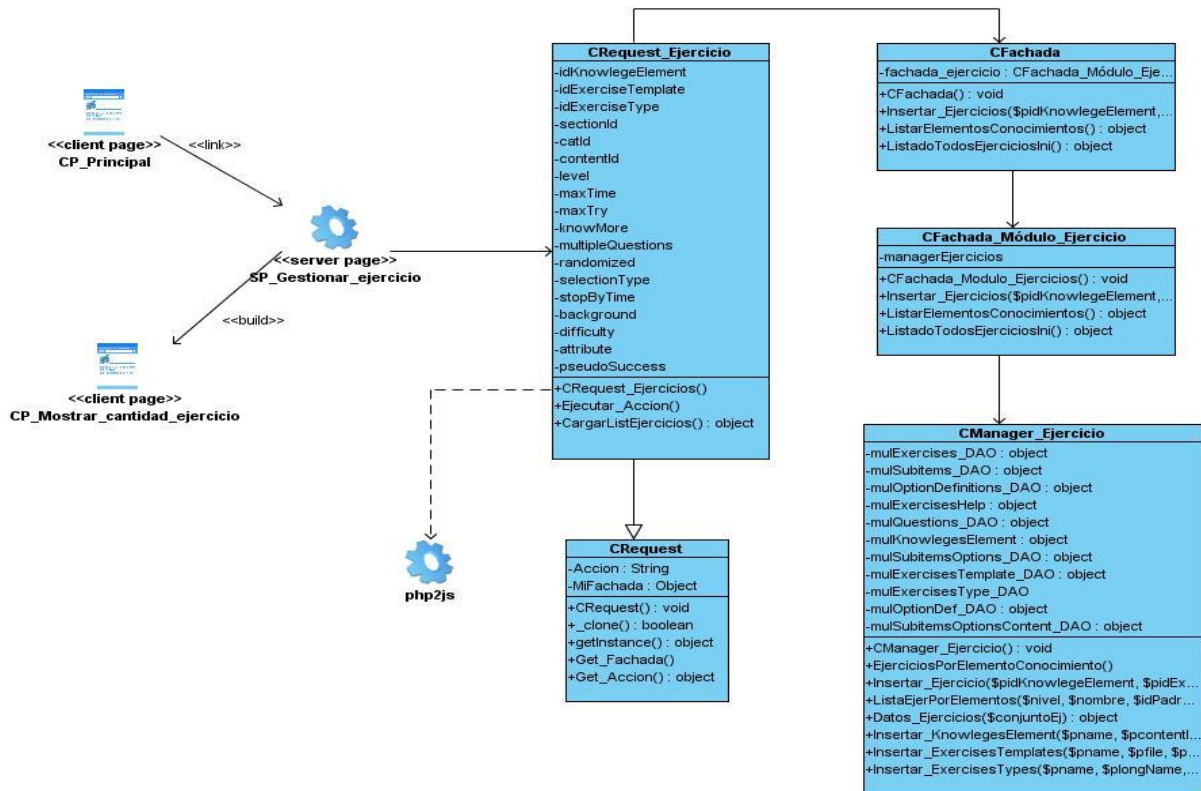


Figura 15. DCD_CU Mostrar cantidad de ejercicio

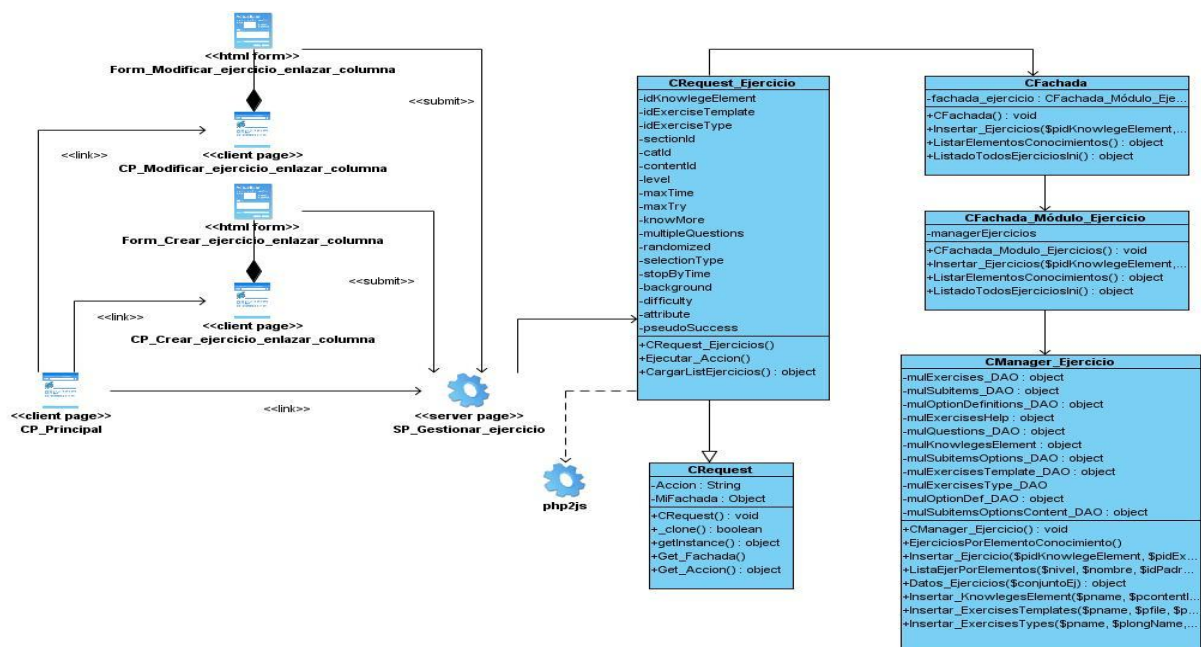


Figura 16. DCD_CU Gestionar ejercicio de enlazar columna

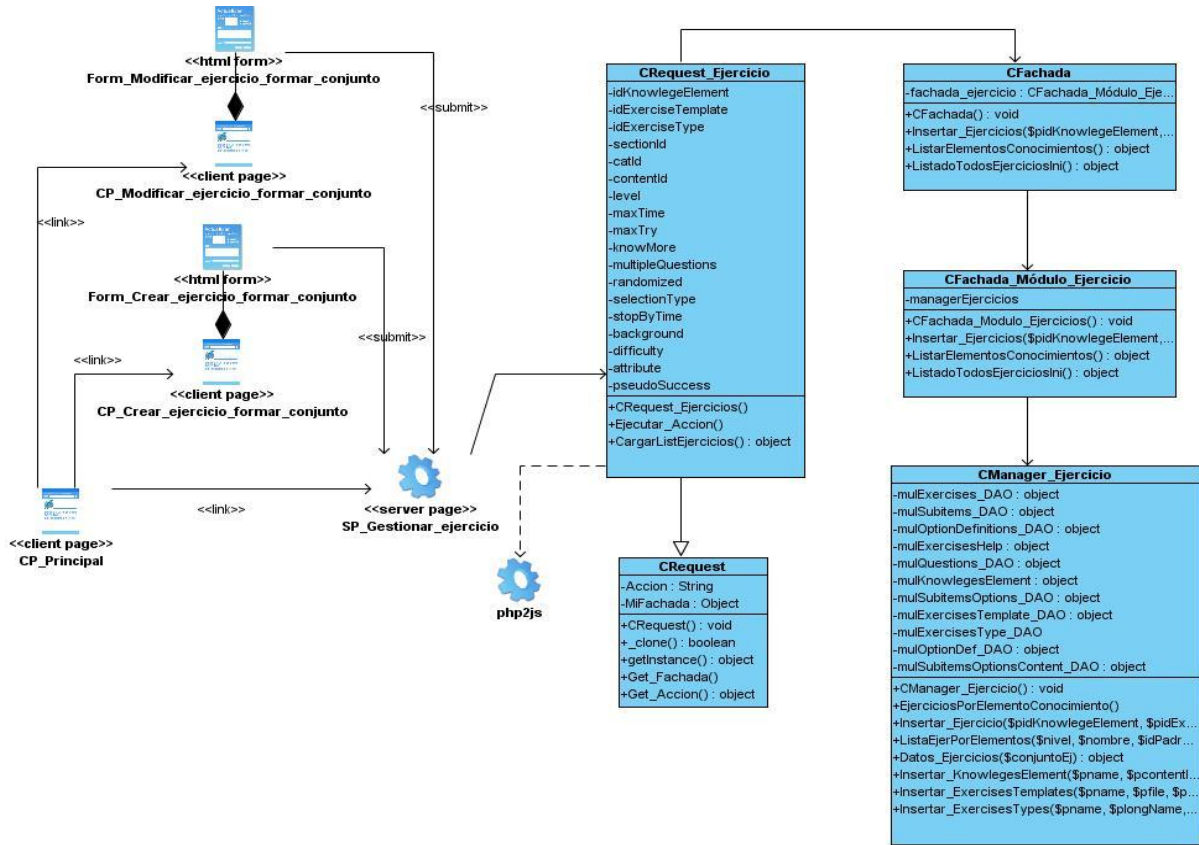


Figura 17. DCD_CU Gestionar ejercicio de formar conjunto

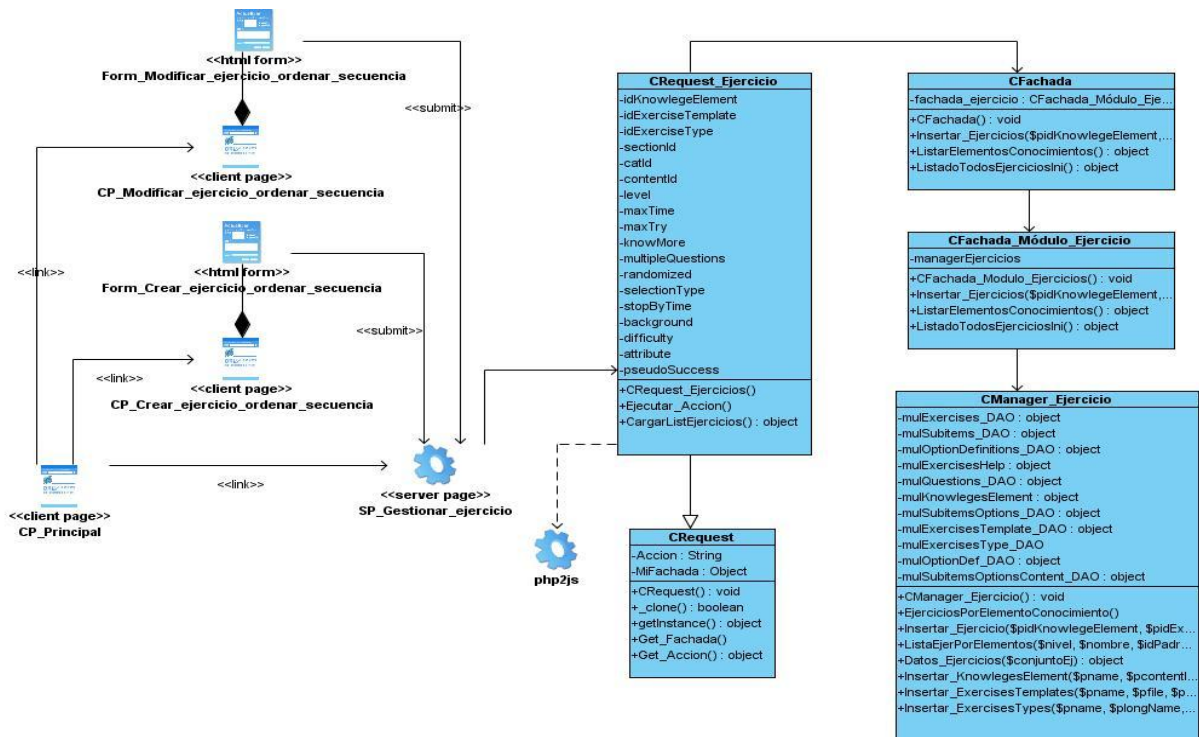


Figura 18. DCD_CU Gestionar ejercicio de ordenar secuencia

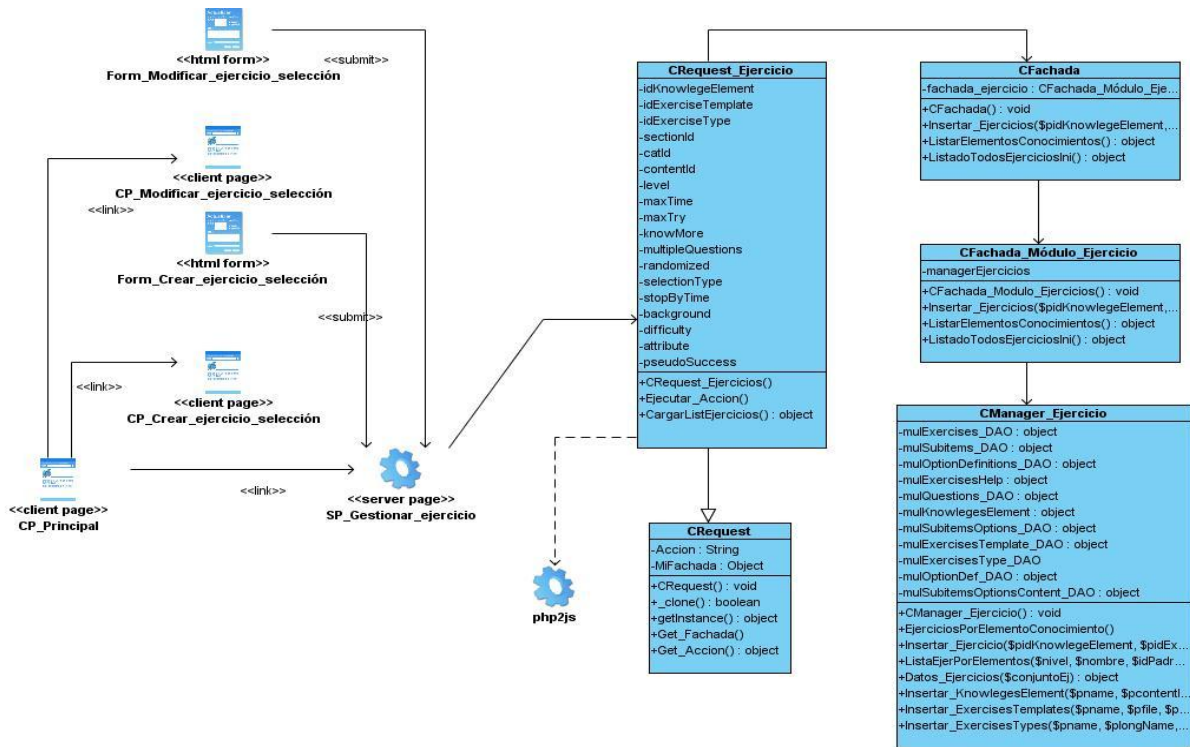


Figura 19. DCD_CU Gestionar ejercicio de selección

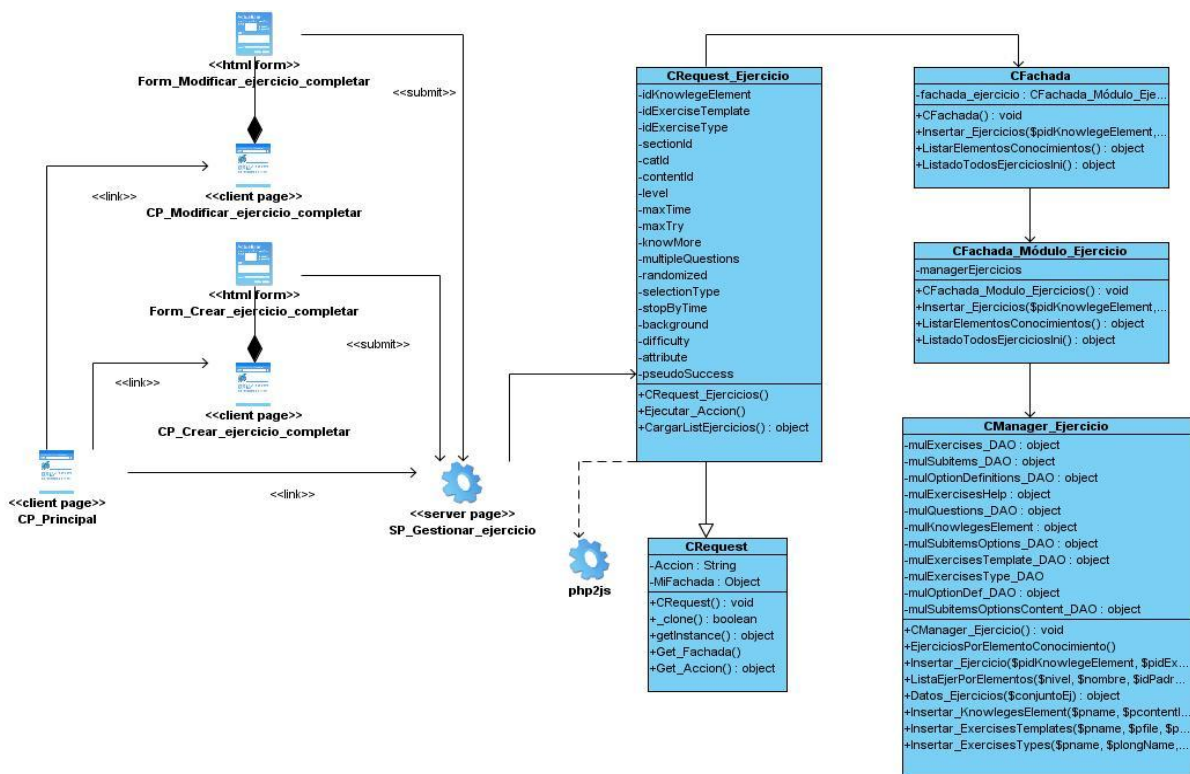


Figura 20. DCD_CU Gestionar ejercicio de completar

Capas de Lógica y Acceso a Datos

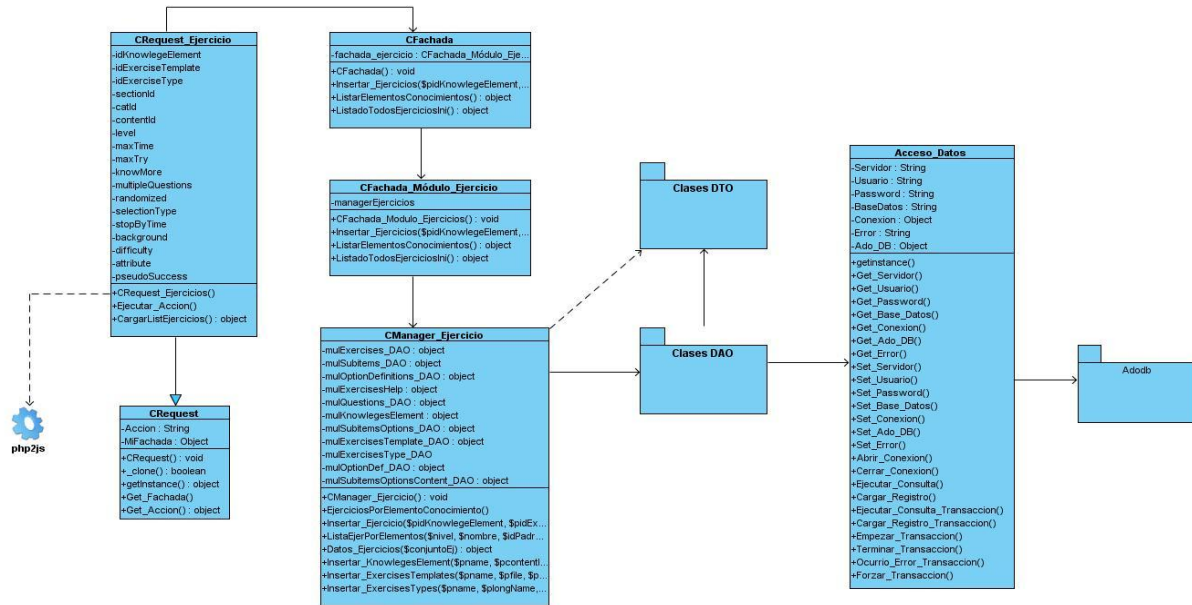


Figura 21. DCD_CU Gestionar ejercicio

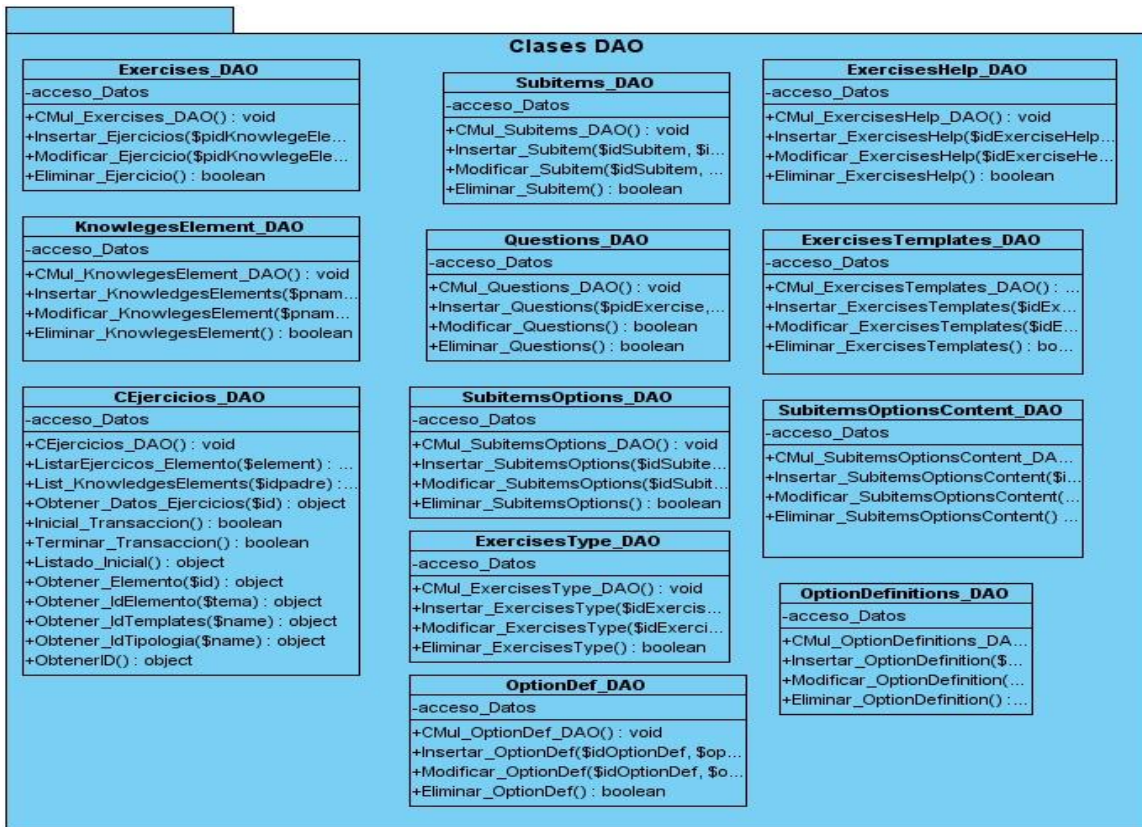


Figura 22. Paquete de clases DAO

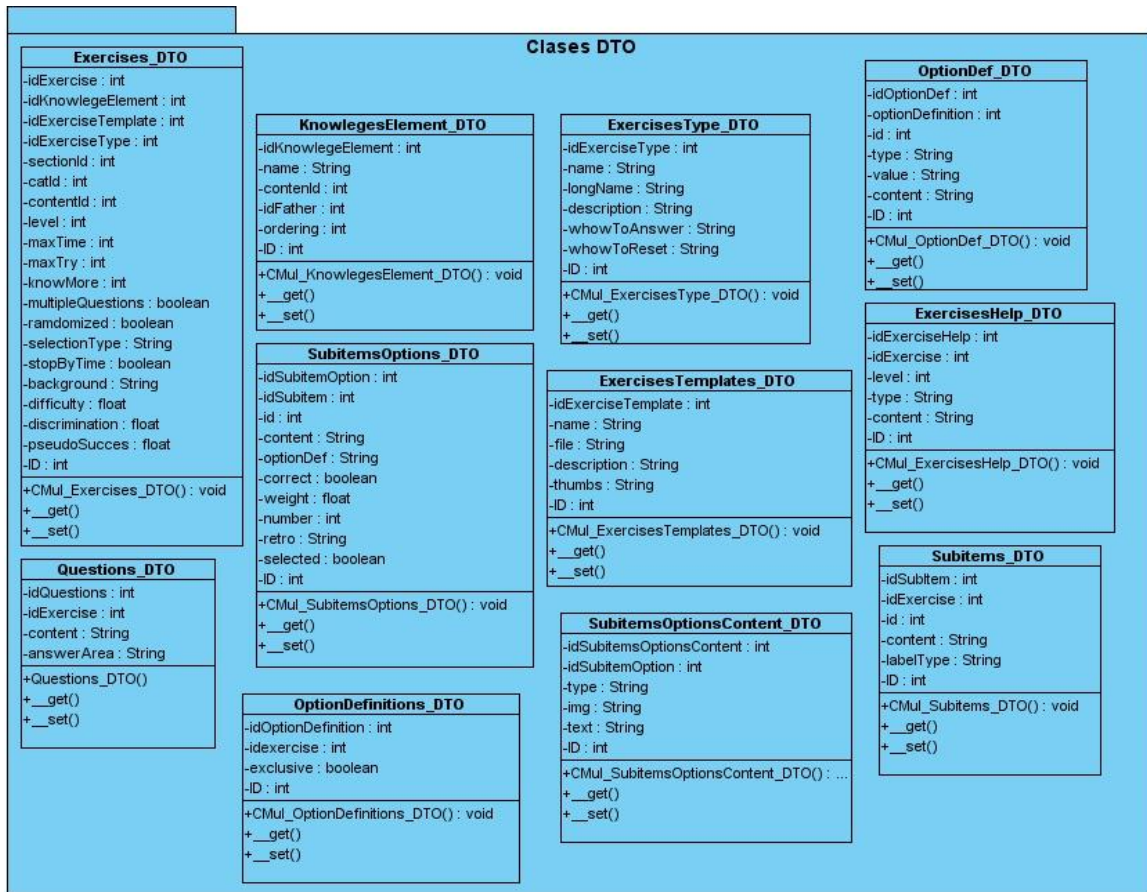


Figura 23. Paquete de clases DTO

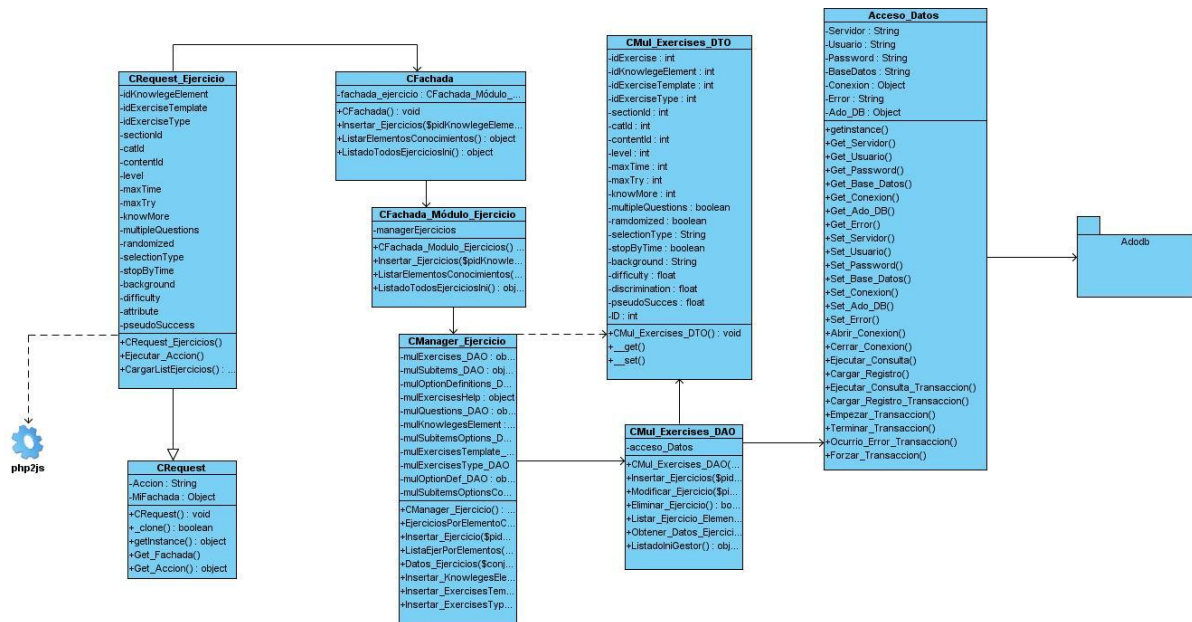


Figura 24. DCD_CU Mostrar cantidad de ejercicio

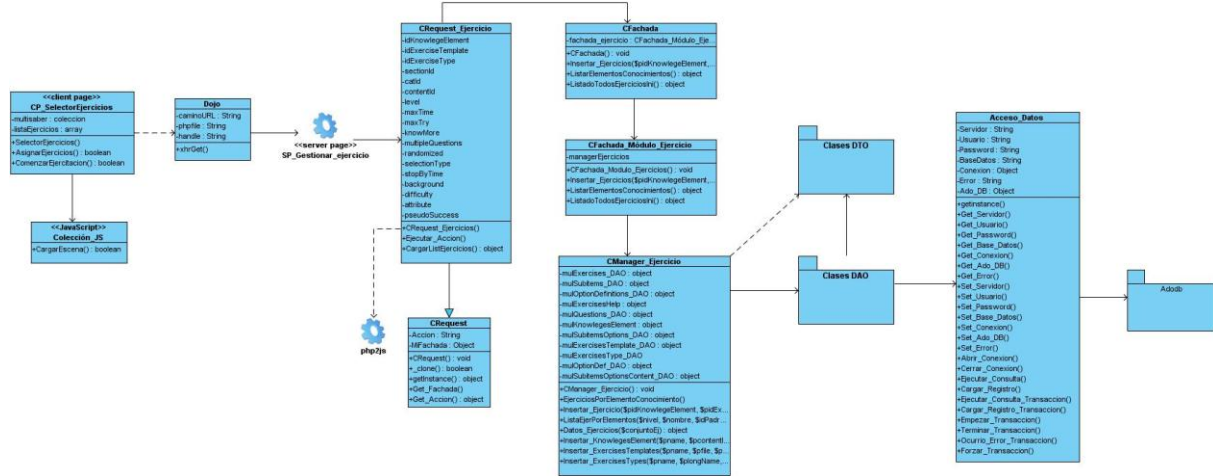


Figura 25. DCD Módulo Ejercicios

3.4 Diagrama de Clases Persistentes

Todas las clases identificadas durante el flujo de trabajo de Análisis y Diseño no son persistentes. La persistencia es la capacidad de un objeto de mantener su valor en el espacio y en el tiempo. Es responsabilidad del diseñador definir cuáles clases son las que deben ser persistentes.

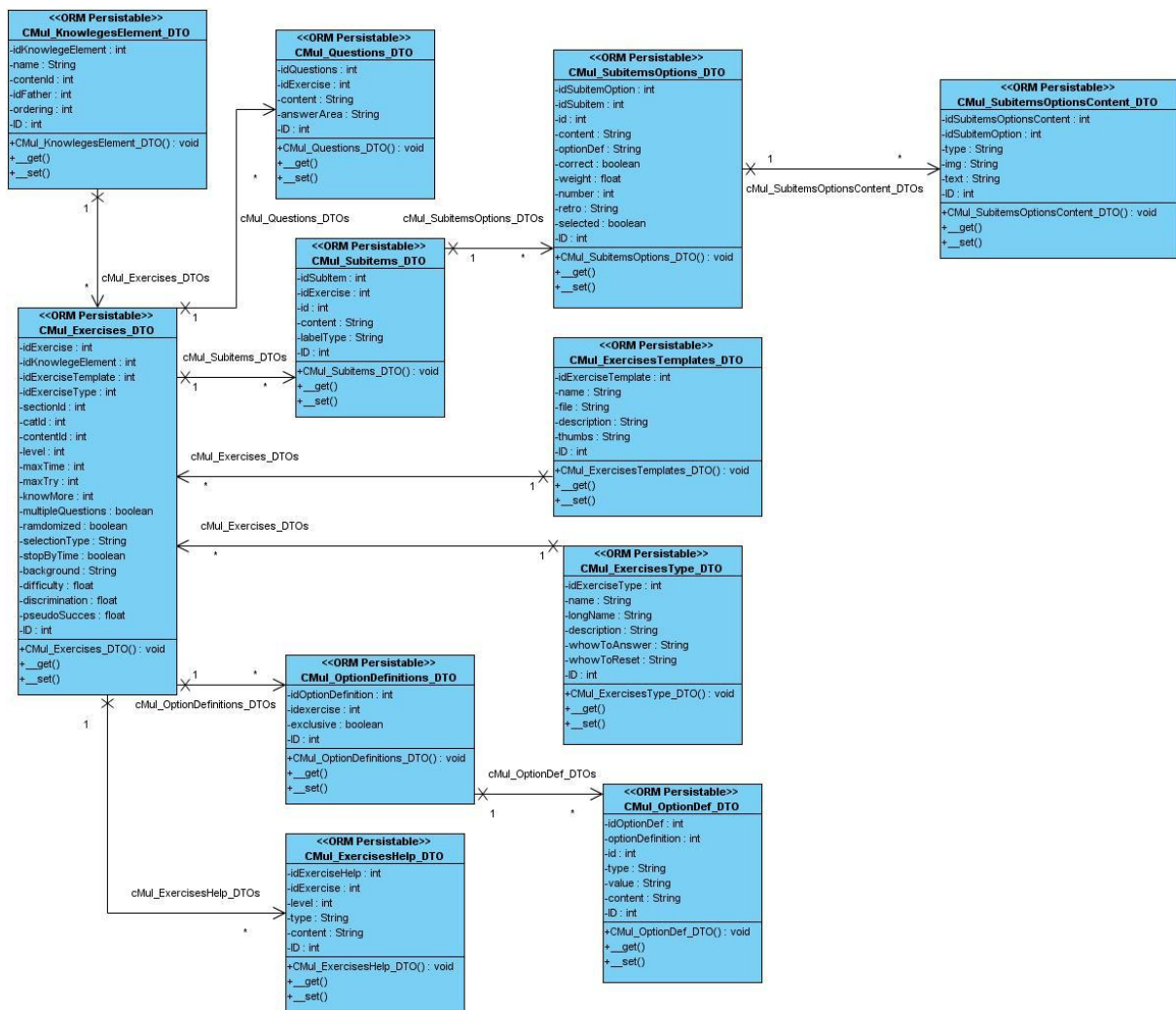


Figura 26. Diagrama de Clases Persistentes

3.5 Modelo de Datos

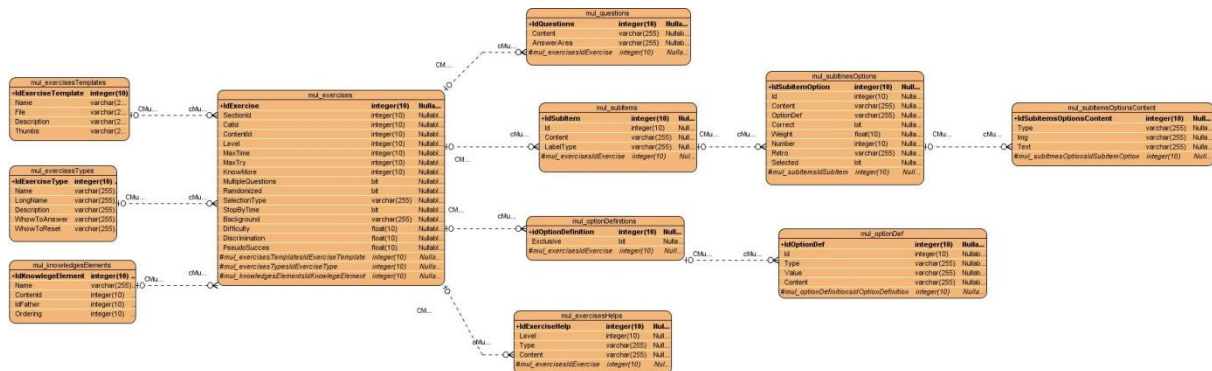


Figura 27. Modelo de Datos

3.6 Patrones de diseño

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular. Es una manera más práctica de describir ciertos aspectos de la organización de un programa.

Según el libro “Design Patterns: Elements of Reusable Object-Oriented Software”, también conocido como GOF existen 3 tipos de patrones:

- De Creación: abstraen el proceso de creación de instancias.
- Estructurales: se ocupan de cómo clases y objetos son utilizados para componer estructuras de mayor tamaño.
- De Comportamiento: atañen a los algoritmos y a la asignación de responsabilidades entre objetos.

Para la implementación de la solución propuesta se usarán los patrones:

Facade (Fachada): Es de tipo estructural, provee de una interfaz unificada simple para acceder a una interfaz o grupo de interfaces de un subsistema.

Entre las principales ventajas que provee están las siguientes:

- Al separar al cliente de los componentes del subsistema, se reduce el número de objetos con los que el cliente trata, facilitando así el uso del subsistema.

- Se promueve un acoplamiento débil entre el subsistema y sus clientes, eliminándose o reduciéndose las dependencias.
- No existen obstáculos para que las aplicaciones usen las clases del subsistema que necesiten. De esta forma se puede elegir entre facilidad de uso y generalidad.
- Ofrece un punto de acceso al resto de las clases, si éstas cambian o se sustituyen por otras solo hay que actualizar la clase Fachada sin que el cambio afecte a las aplicaciones cliente.

Data Access Object (DAO, Objeto de Acceso a Datos): es un “adaptador” o un nexo entre la lógica de negocio y la capa de persistencia (generalmente, una Base de Datos). Permite que se encapsule la forma de acceder a la fuente de datos. Surge de la necesidad de gestionar una diversidad de fuentes de datos, aunque su uso se extiende al problema de encapsular no sólo la fuente de datos, sino además ocultar la forma de acceder a los datos. Se trata de que el software cliente se centre en los datos que necesita y se olvide de cómo se realiza el acceso a los mismos o de cuál es la fuente de almacenamiento.

Singleton (Solitario): Es de tipo creacional, garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia.

Algunas de sus características son:

- El acceso a la instancia única está más controlado. Puede tener un control estricto sobre cómo y cuando acceden los clientes a la instancia.
- Permite el refinamiento de operaciones y la representación. Se puede crear una subclase de Singleton.
- Permite un número variable de instancias.

3.7 Conclusiones

A través del desarrollo del capítulo 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 han expuesto los patrones de diseño que ofrecen un código con ventajas de reutilización y facilidad de comprensión y modificación.

IMPLEMENTACIÓN DEL SISTEMA

4.1 Introducción

En el presente capítulo se exponen los artefactos generados durante el flujo de trabajo de Implementación, se describe cómo los elementos del Modelo de Diseño son implementados en términos de componentes y la forma en que va a estar distribuido el sistema, mediante los Diagramas de Componentes y de Despliegue. Se presenta además, cómo está diseñada la interfaz de la aplicación.

4.2 Modelo de Implementación

La implementación comienza con los resultados del diseño. En el Modelo de Implementación se describen los componentes a construir, así como la organización y dependencia entre nodos físicos en los que funcionará la aplicación, a través de los Diagramas de Componentes y de Despliegue.

4.2.1 Diagrama de Componentes

Los Diagramas de Componentes muestran los componentes de software (ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares) y las relaciones que existen entre ellos. Además de las dependencias de compilación que existen entre los ficheros de código.

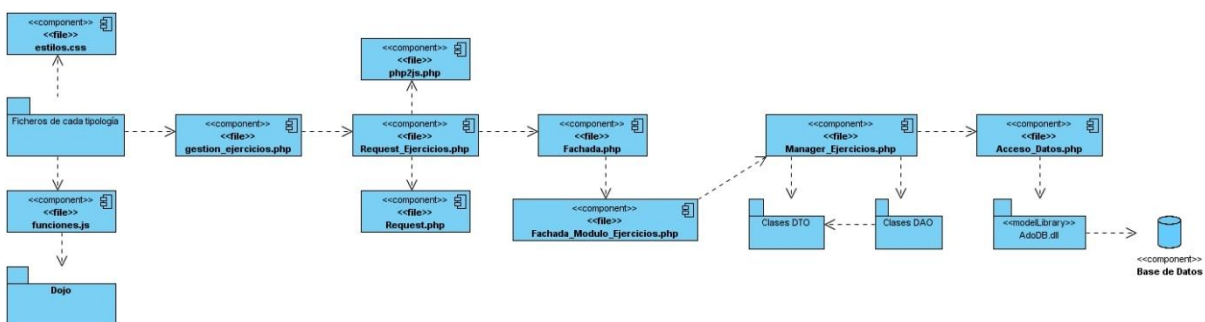


Figura 28. Diagrama de Componentes General

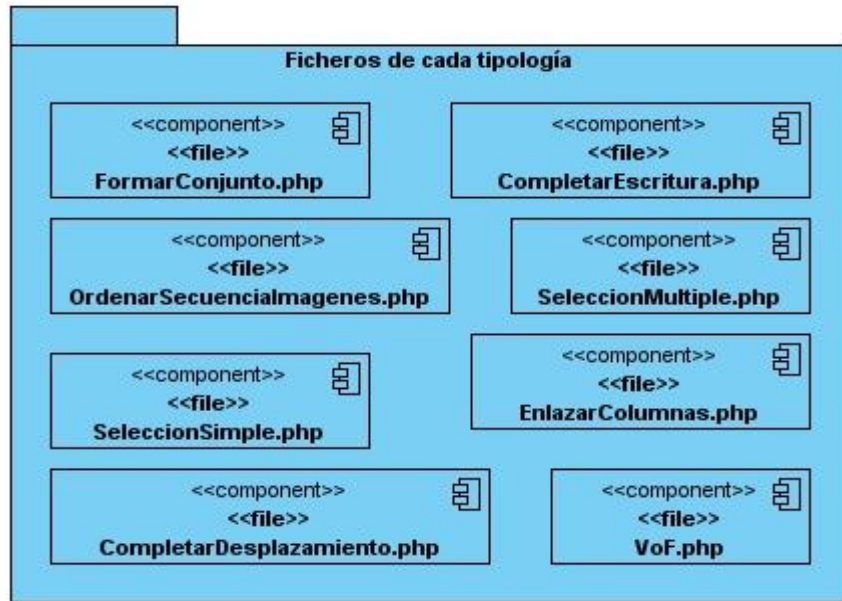


Figura 29. Diagrama de Componentes, paquete de ficheros de cada tipología

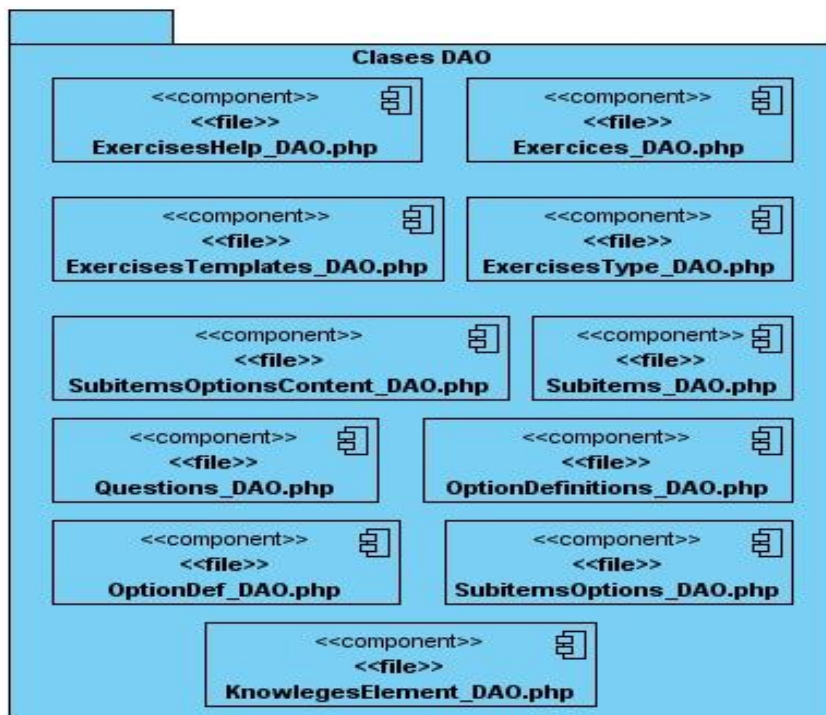


Figura 30. Diagrama de Componentes, paquete Clases DAO

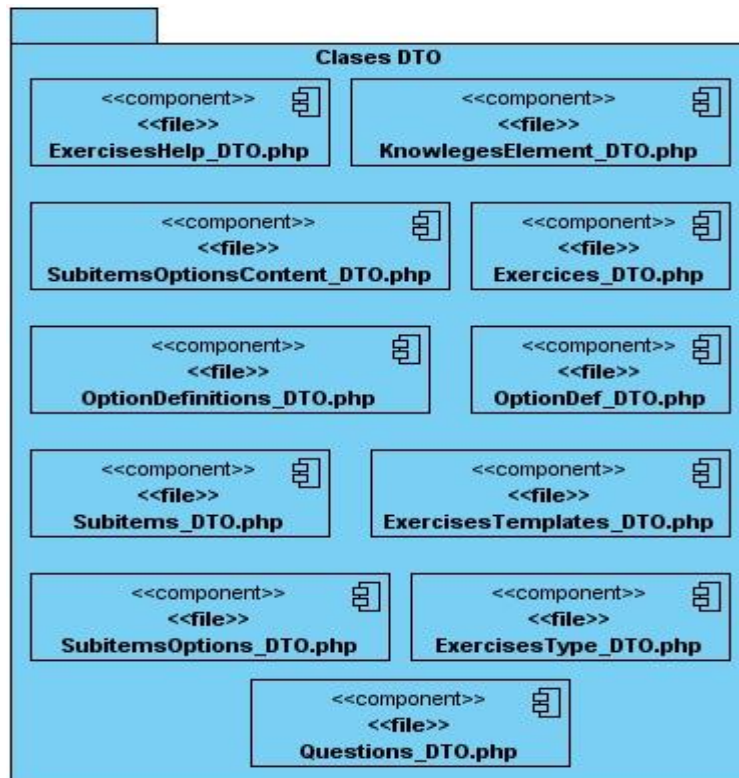


Figura 31. Diagrama de Componentes, paquete Clases DTO

4.2.2 Diagrama de Despliegue

El Diagrama de Despliegue muestra la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos.

A continuación se exponen dos Diagramas de Despliegue, ya que existen dos formas en las que se puede desplegar la aplicación, atendiendo a las necesidades del cliente. El primero está compuesto por dos nodos, uno representa la computadora cliente, donde el usuario hará uso de la aplicación, y el otro representa los servidores web y de bases de datos. El segundo diagrama muestra el servidor web y de base de datos montados en la misma PC cliente.

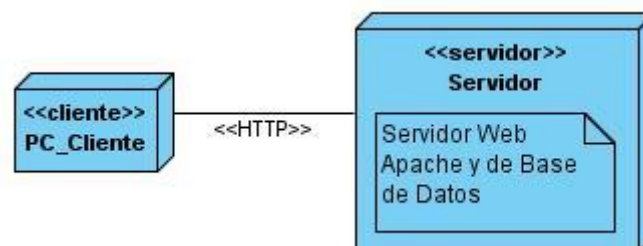


Figura 32. Diagrama de Despliegue 1

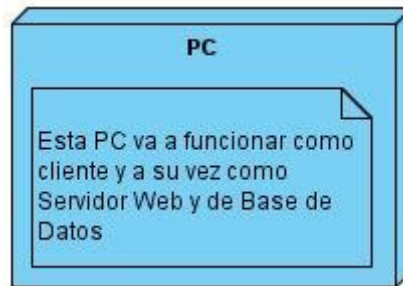


Figura 33. Diagrama de Despliegue 2

4.3 Interfaz de la aplicación

La interfaz es la parte que permite al usuario interactuar con una aplicación. El diseño de interfaces es una actividad que ha adquirido gran importancia en el desarrollo de software. La calidad con que sea diseñada la interfaz de usuario determina el éxito o el fracaso del sistema, por tanto es uno de los aspectos más relevantes de la usabilidad de la aplicación.

Enfocando los pasos a lograr interfaces agradables, sencillas y atractivas a la vista del usuario, que en este caso será un profesor o maestro, y buscando la reducción de errores posibles a cometer por dicho usuario, se obtienen las siguientes interfaces de la aplicación.

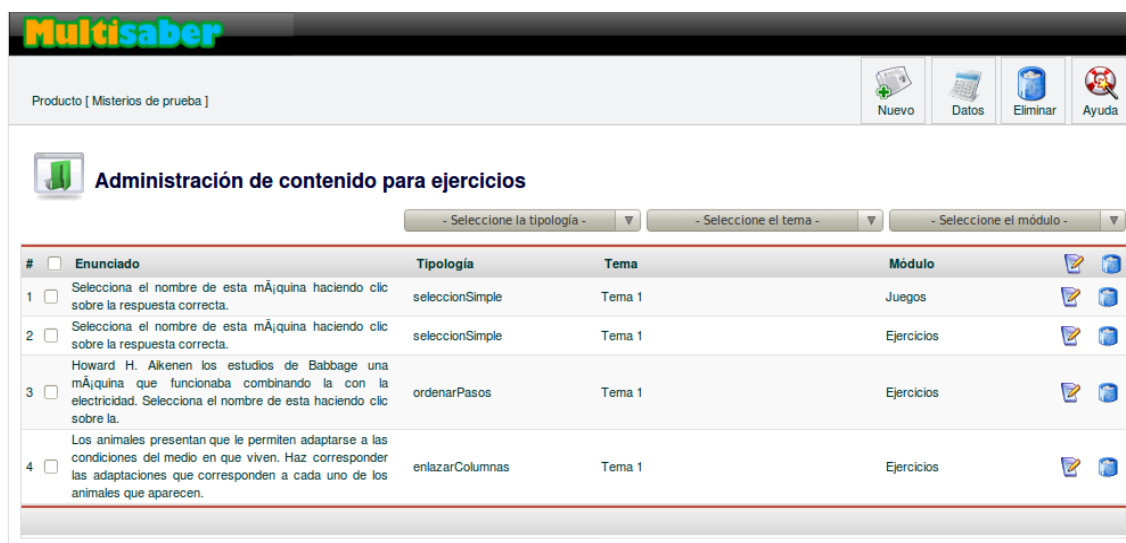


Figura 34. Interfaz página principal

Producto [Misterios de prueba]

Aplicar Guardar Cerrar Ayuda

Ejercicios: Nuevo [Tipología: Selección simple]

Detalles

Tema: -- Seleccione -- Nivel de desempeño: -- Seleccione -- Número de intentos: -- Seleccione --

Enunciado:

Tipo de selección:
 Texto
 Imagen

Propósito:
 Módulo Ejercicios
 Módulo Juegos

Posicionamiento:
 Texto en dos columnas

Microsoft Publisher Microsoft Excel

Correcto	Opciones	Retroalimentación	Eliminar
<input type="radio"/>			
<input type="radio"/>			
<input type="radio"/>			

Añadir Opciones

Niveles de ayuda

Imagen de referencia

Examinar...

Figura 35. Interfaz Insertar ejercicios de selección simple

The screenshot shows the 'Ejercicios: Nuevo' interface for 'Selección múltiple' (multiple choice) exercises. At the top, there's a header with the 'Multisaber' logo and a navigation bar with icons for 'Aplicar', 'Guardar', 'Cerrar', and 'Ayuda'. Below the header, the product name 'Producto [Misterios de prueba]' is displayed. The main title is 'Ejercicios: Nuevo [Tipología: Selección múltiple]'. Under the 'Detalles' section, there are dropdown menus for 'Tema', 'Nivel de desempeño', and 'Número de intentos'. The 'Enunciado' field is a large text area. To its right, there are radio buttons for 'Tipo de selección' (Text and Imagen, with 'Imagen' selected) and checkboxes for 'Propósito' (Módulo Ejercicios and Módulo Juegos). Below this is a table with columns: 'Correcto', 'Opciones', 'Texto', 'Retroalimentación', and 'Eliminar'. The table has three rows, each with a checkbox, an 'Opciones' input field with an 'Examinar...' button, a 'Texto' input field, a 'Retroalimentación' input field, and an 'Eliminar' button. At the bottom, there is an 'Añadir Opciones' button and a 'Niveles de ayuda' input field.

Figura 36. Interfaz Insertar ejercicios de selección múltiple

The screenshot shows the 'Ejercicios: Nuevo' interface for 'Verdadero o falso' (true or false) exercises. The layout is similar to Figure 36, with the same header and navigation bar. The main title is 'Ejercicios: Nuevo [Tipología: Verdadero o falso]'. The 'Detalles' section includes dropdown menus for 'Tema', 'Nivel de desempeño', and 'Número de intentos'. The 'Enunciado' field is a large text area. To its right, there are checkboxes for 'Propósito' (Módulo Ejercicios and Módulo Juegos). Below this is a table with columns: 'Verdadero', 'Opciones', 'Retroalimentación', and 'Eliminar'. The table has three rows, each with a checkbox, an 'Opciones' input field, a 'Retroalimentación' input field, and an 'Eliminar' button. At the bottom, there is an 'Añadir Opciones' button, a 'Niveles de ayuda' input field, an 'Imagen de referencia' input field, and an 'Examinar...' button.

Figura 37. Interfaz Insertar ejercicios de verdadero o falso

Producto [Misterios de prueba]

Aplicar Guardar Cerrar Ayuda

Ejercicios: Nuevo [Tipología: Enlazar columnas]

Detalles

Tema: -- Seleccione -- Nivel de desempeño: -- Seleccione -- Número de intentos: -- Seleccione --

Enunciado:

Propósito

Módulo Ejercicios

Módulo Juegos

Columna A	Corres	Columna B	Retroalimentación (Correcto)	Retroalimentación (Incorrecto)
<input type="text"/>	➔	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	➔	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	➔	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	➔	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	➔	<input type="text"/>	<input type="text"/>	<input type="text"/>

Niveles de ayuda

Figura 38. Interfaz Insertar ejercicios de enlazar columnas

Producto [Misterios de prueba]

Aplicar Guardar Cerrar Ayuda

Ejercicios: Nuevo [Tipología: Ordenar imágenes]

Detalles

Tema: -- Seleccione -- Nivel de desempeño: -- Seleccione -- Número de intentos: -- Seleccione --

Enunciado:

Módulo

Módulo Ejercicios

Módulo Juegos

Orden	Imagen	Retroalimentación
1ra	<input type="text"/> Examinar...	<input type="text"/>
2da	<input type="text"/> Examinar...	<input type="text"/>
3ra	<input type="text"/> Examinar...	<input type="text"/>
4ta	<input type="text"/> Examinar...	<input type="text"/>
5ta	<input type="text"/> Examinar...	<input type="text"/>

Nivel de ayuda

Figura 39. Interfaz Insertar ejercicios de ordenar secuencia de imágenes

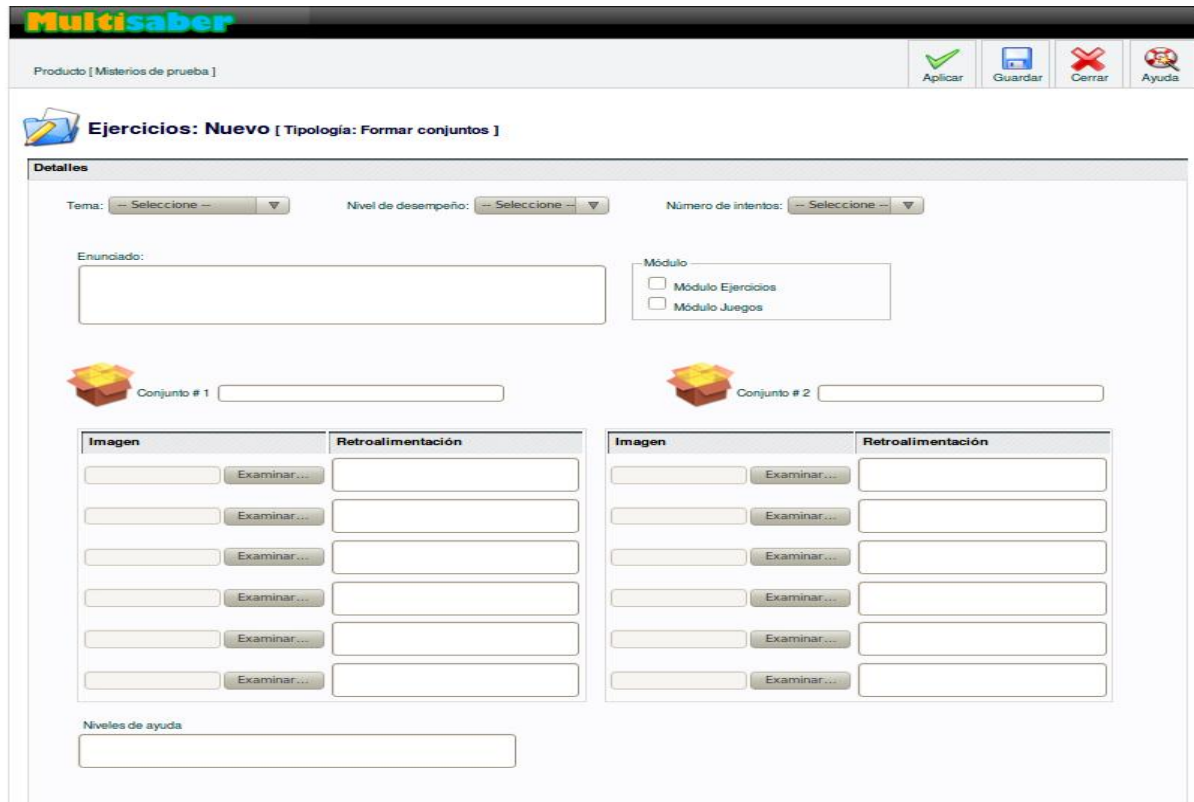


Figura 40. Interfaz Insertar ejercicios de formar conjunto

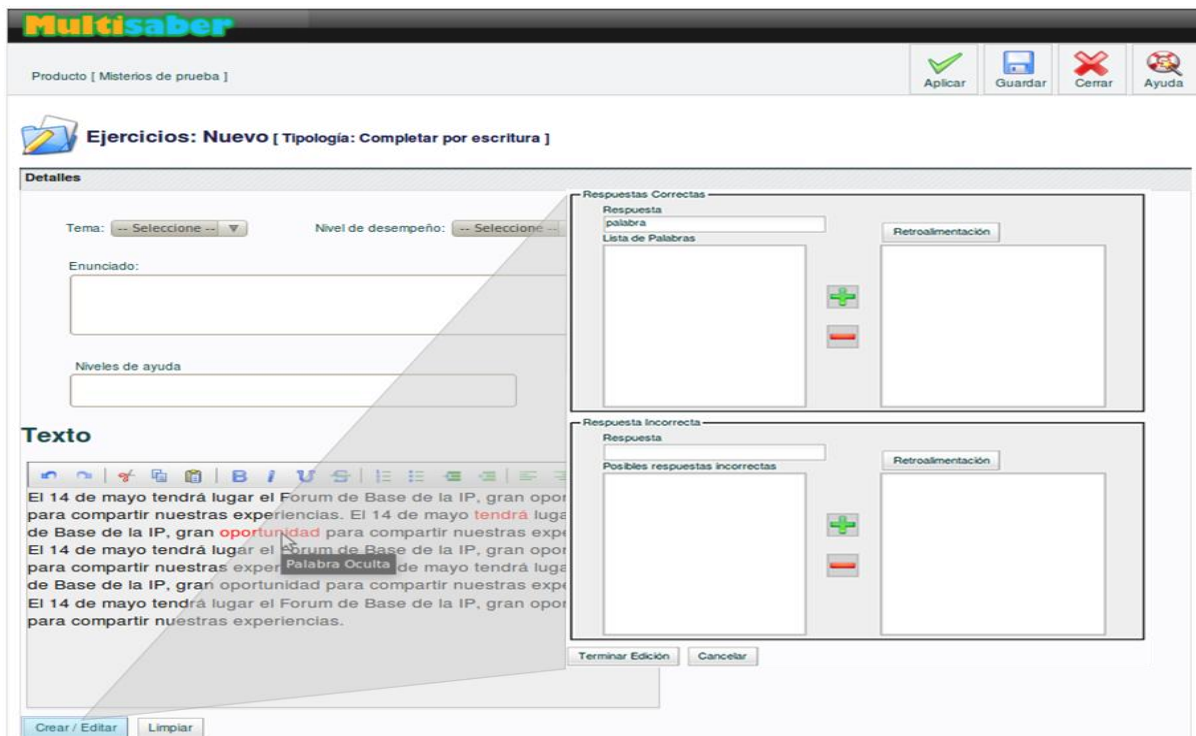


Figura 41. Interfaz Insertar ejercicios de completar por escritura

The screenshot shows the 'Ejercicios: Nuevo' interface for the 'Completar por desplazamiento' (fill-in-the-blanks) exercise type. At the top, the 'Multisaber' logo is on the left, and navigation icons for 'Aplicar', 'Guardar', 'Cerrar', and 'Ayuda' are on the right. Below the header, the title 'Ejercicios: Nuevo [Tipología: Completar por desplazamiento]' is displayed. The 'Detalles' section contains several form fields: 'Tema' and 'Contenido' are text boxes; 'Nivel de desempeño' and 'Número de intentos' are dropdown menus; 'Enunciado' is a large text area; and 'Niveles de ayuda' is a text box. To the right of the 'Enunciado' field is a 'Propósito' section with two checkboxes: 'Módulo Ejercicios' and 'Módulo Juegos'. Below the 'Enunciado' field is a 'Texto' section with a rich text editor toolbar and a text area containing a paragraph about Blaise Pascal's calculator. At the bottom left of the text area are 'Crear / Editar' and 'Limpiar' buttons.

Figura 42. Interfaz Insertar ejercicios de completar por desplazamiento

The screenshot shows the 'Ejercicios: Nuevo' interface for the 'Ordenar pasos' (order steps) exercise type. The layout is similar to Figure 42, with the 'Multisaber' logo and navigation icons at the top. The title is 'Ejercicios: Nuevo [Tipología: Ordenar pasos]'. The 'Detalles' section features three dropdown menus for 'Tema', 'Nivel de desempeño', and 'Número de intentos'. Below these is the 'Enunciado' text area and the 'Propósito' checkboxes. The main part of the interface is a table with three columns: 'Orden', 'Pasos', and 'Retroalimentación'. The 'Orden' column lists '1ro', '2do', '3ro', and '4to'. Each row has a corresponding text box in the 'Pasos' column and another in the 'Retroalimentación' column. At the bottom, there is a 'Niveles de ayuda' text box.

Orden	Pasos	Retroalimentación
1ro		
2do		
3ro		
4to		

Figura 43. Interfaz Insertar ejercicios de ordenar pasos

The screenshot shows the 'Ejercicios: Nuevo' interface for 'Tipología: Identificar objeto'. It includes a header with the Multisaber logo and navigation icons (Aplicar, Guardar, Cerrar, Ayuda). The main area is titled 'Detalles' and contains several form elements:

- Dropdown menus for 'Tema', 'Nivel de desempeño', and 'Número de intentos'.
- An 'Enunciado' text area.
- A 'Módulo' section with checkboxes for 'Módulo Ejercicios' and 'Módulo Juegos'.
- A 'CONJUNTO' label with a small cube icon and an associated text input field.
- A table with 5 columns: 'No.', 'Correcto', 'Objeto', 'Titulo del objeto', and 'Retroalimentación'. It contains 6 rows for exercise entries.
- A 'Niveles de ayuda' text input field at the bottom.

Figura 44. Interfaz Insertar ejercicios de identificar un objeto

The screenshot shows the 'Ejercicios: Nuevo' interface for 'Tipología: Identificar una clase de objeto'. It has the same layout as Figure 44 but with a few key differences:

- The title is 'Ejercicios: Nuevo [Tipología: Identificar una clase de objeto]'.
- The 'CONJUNTO' label is replaced by 'Clase de Objeto' with a small cube icon.
- The table structure remains identical to the previous interface.

Figura 45. Interfaz Insertar ejercicios de identificar una clase de objeto

4.4 Conclusiones

En el presente capítulo se presentaron los Diagramas de Componentes y de Despliegue, a través de los cuales se describen los componentes desarrollados y su organización, así como la dependencia entre nodos físicos en los que funcionará la aplicación. Se expusieron las interfaces de la aplicación, las cuales han sido diseñadas con el propósito de satisfacer las expectativas del usuario.

CONCLUSIONES GENERALES

Con el desarrollo del presente trabajo

- Se realizó una investigación sobre aplicaciones existentes en la UCI y a nivel internacional, que posibilitan la gestión de contenidos para ejercicios, de las cuales se concluyó, que ninguna resolvía la situación problemática planteada por lo que era necesario desarrollar una nueva aplicación.
- Se estudiaron diferentes tecnologías y herramientas usadas en la actualidad para desarrollar aplicaciones web, logrando de esta manera, escoger las más adecuadas para llevar a cabo la implementación de la solución propuesta.
- Se generaron los artefactos fundamentales propuestos por la metodología RUP para los flujos de trabajo de Análisis, Diseño e Implementación, obteniendo la documentación necesaria para garantizar el futuro mantenimiento del sistema.
- Se desarrolló una aplicación para la gestión de los contenidos de los ejercicios de la Colección Multisaber en su versión multiplataforma, con una interfaz amigable y fácil de usar, la cual facilita que dicho proceso de gestión se realice de forma eficiente y dinámica.

RECOMENDACIONES

Se recomienda:

- Continuar desarrollando la aplicación, brindándole soporte al resto de las tipologías presentes en la Colección Multisaber.
- Incorporar las funcionalidades de exportar el contenido de los ejercicios en formato PDF e importar los ficheros existentes con los contenidos actuales.
- Realizar pruebas a la aplicación.
- Integrar la aplicación como parte de la Colección Multisaber en su versión multiplataforma.

REFERENCIAS BIBLIOGRÁFICAS

1. **Ms C Labañino, César.** *Maestría Ciencias de la Educación.* [CD] Ciudad de La Habana : s.n., 2006.
2. **Jumbo, Fabián.** slideshare. [En línea] 2008. <http://www.slideshare.net/jumbofabian/multimedia-educativa-presentation-715361>.
3. **Comunidad de aprendizaje sobre multimedia educativo.** Prometeo. Comunidad de aprendizaje sobre multimedia educativo. [En línea] 18 de octubre de 2008. [Citado el: 3 de febrero de 2010.] http://prometeo.co.cr/index.php?option=com_content&view=article&id=61:i-que-tipos-de-multimedia-educativo-hay&catid=41:multimedia-educativa&Itemid=91.
4. **Dr Marqués, Pere.** MULTIMEDIA EDUCATIVO: CLASIFICACIÓN, FUNCIONES, VENTAJAS E INCONVENIENTES. [En línea] 12 de enero de 2009. [Citado el: 3 de febrero de 2010.] <http://www.pangea.org/peremarques/funcion.htm>.
5. **Hot Potatoes.** Hot Potatoes. [En línea] 2009. [Citado el: 15 de enero de 2010.] <http://hotpot.uvic.ca/>.
6. **Matanza, José Manuel Bouzán.** webArdora.net. [En línea] http://www.webardora.net/index_cas.htm.
7. **Fundación Asmoz.** asmoz. [En línea] [Citado el: 13 de enero de 2010.] http://www.asmoz.org/index.php?option=com_content&view=article&id=131&Itemid=101&lang=es.
8. *@tic : revista d'innovació educativa.* **García, N, José, F y Piqueras, C.** 3, 2009.
9. **Díaz, Mirian Milagros.** Universidad de San Martín Porres. [En línea] [Citado el: 15 de enero de 2010.] <http://www.usmp.edu.pe/publicaciones/boletin/fia/info49/articulos/RUP%20vs.%20XP.pdf>.
10. **definicion.org.** definicion.org. [En línea] [Citado el: 15 de enero de 2010.] <http://www.definicion.org/lenguaje-de-programacion>.
11. **Prado Ajona, Guillermo.** Universidad de la Rioja. [En línea] [Citado el: 10 de febrero de 2010.] <https://belenus.unirioja.es/~guprado/pagweb/historiacss.html>.
12. **Pérez, Damián.** maestros del web. [En línea] 2 de noviembre de 2007. [Citado el: 15 de febrero de 2010.] <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.
13. **Python.** Python. [En línea] 2009. [Citado el: 18 de enero de 2010.] <http://www.python.org>.
14. **Ruby.** Ruby A Programmer's Best Friend. [En línea] [Citado el: 18 de enero de 2010.] <http://www.ruby-lang.org/es/about>.
15. **Booch, G, Rumbaugh, J y Jacobson, I.** "El Lenguaje Unificado de Modelado". 2000.

16. **Popkin Software and Systems.** Popkin Software and Systems. [En línea]
<http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/doc-modelado-sistemas-uml.pdf>.
17. **Tuesta, Martín.** El prisma. [En línea] 2005. [Citado el: 10 de febrero de 2010.]
<http://www.elprisma.com/apuntes/curso.asp?id=13324>.
18. **Grupo soluciones GSInnova.** Grupo soluciones GSInnova. [En línea] 2007. [Citado el: 10 de febrero de 2010.] <http://www.rational.com.ar/index.html>.
19. **IEEE.** Std 1471-2000.
20. **Peláez, Juan.** Blog. [En línea] 29 de mayo de 2009. [Citado el: 19 de enero de 2010.]
<http://geeks.ms/blogs/jkpelaez/archive/2009/05/29/arquitectura-basada-en-capas.aspx>.
21. **Canales, Roberto.** Arquitectura SOA e Integración de aplicaciones. *Adictos al trabajo*. [En línea] 2008. <http://www.adictosaltrabajo.com/tutoriales/pdfs/soa.pdf>.
22. **Lago, Ramiro.** Patrones de diseño de software. [En línea] 2007. [Citado el: 20 de enero de 2010.]
<http://www.proactiva-calidad.com/java/patrones/mvc.html>.
23. **babylon.** Dictionary & Translation by Babylon. [En línea] 2007. [Citado el: 15 de enero de 2010.]
http://www.babylon.com/definition/entorno_de_desarrollo/Spanish.
24. **NetBeans Community.** NetBeans. [En línea] [Citado el: 16 de enero de 2010.]
http://www.netbeans.org/index_es.html.
25. *Innovación y experiencias educativas.* **Zambrano, Raquel.** 14, Cordoba : s.n., 2008. ISSN 1988-6047.
26. **Comunidad emagister.** emagister.com. [En línea] [Citado el: 16 de enero de 2010.]
<http://www.emagister.com/que-es-oracle-cursos-1053949.htm#programa>.
27. **PosgreSQL.** PostgreSQL. [En línea] 2010. [Citado el: 17 de enero de 2010.]
<http://www.postgresql.org/about/>.
28. **MySQL.** MySQL. [En línea] 2010. [Citado el: 12 de enero de 2010.]
<http://www.mysql.com/about/>.
29. —. MySQL. [En línea] 2010. [Citado el: 16 de enero de 2010.]
<http://dev.mysql.com/doc/refman/5.0/es/features.html>.
30. **masadelante.com.** masadelante.com. [En línea] 2010. [Citado el: 16 de enero de 2010.]
<http://www.masadelante.com/faqs/servidor>.
31. **Microsoft Office.** Microsoft Office Online. [En línea] 2010. [Citado el: 16 de enero de 2010.]
<http://office.microsoft.com/training/training.aspx?AssetID=RP061790473082&CTT=6&Origin=RP061790353082>.

BIBLIOGRAFÍA

1. Ms C Labañino, César. **Maestría Ciencias de la Educación**. [CD] Ciudad de La Habana : s.n., 2006.
2. Jumbo, Fabián. **Multimedia educativa**. [Online] 2008. Disponible en web: <http://www.slideshare.net/jumbofabian/multimedia-educativa-presentation-715361>.
3. **Comunidad de aprendizaje sobre multimedia educativo**. Prometeo. Comunidad de aprendizaje sobre multimedia educativo. [Online] octubre 18, 2008. [Cited: febrero 3, 2010.] Disponible en web: http://prometeo.co.cr/index.php?option=com_content&view=article&id=61:ique-tipos-de-multimedia-educativo-hay&catid=41:multimedia-educativa&Itemid=91.
4. Dr Marquéz, Pere. **MULTIMEDIA EDUCATIVO: CLASIFICACIÓN, FUNCIONES, VENTAJAS E INCONVENIENTES**. [Online] enero 12, 2009. [Cited: febrero 3, 2010.] Disponible en web: <http://www.pangea.org/peremarques/funcion.htm>.
5. **Sitio oficial de Hot Potatoes**. [Online] Disponible en web: <http://hotpot.uvic.ca/>.
6. **Sitio oficial de Ardora** [Online] Disponible en web: <http://www.webardora.net>.
7. García, N, José, F and Piqueras. **Rrevista d'innovació educativa**. C. 3, 2009.
8. Publicación Universidad de San Martín Porres. [Online] [Cited: enero 15, 2010.] Disponible en web: <http://www.usmp.edu.pe/publicaciones/boletin/fia/info49/articulos/RUP%20vs.%20XP.pdf>.
9. Pérez, Damián. **Maestros del web** [Online] noviembre 2, 2007. [Cited: febrero 15, 2010.] Disponible en web: <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.
10. **Sitio oficial de Python**. [Online] 2009. [Cited: enero 18, 2010.] Disponible en web: <http://www.python.org>.
11. **Ruby A Programmer's Best Friend**. [Online] [Cited: enero 18, 2010.] Disponible en web: <http://www.ruby-lang.org/es/about>.
12. Booch, G, Rumbaugh, J and Jacobson, I. **"El Lenguaje Unificado de Modelado"**. 2000.
13. **Popkin Software and Systems**. [Online] Disponible en web: <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/doc-modelado-sistemas-uml.pdf>.
14. Tuesta, Martín. **El prisma**. [Online] 2005. [Cited: febrero 10, 2010.] Disponible en web: <http://www.elprisma.com/apuntes/curso.asp?id=13324>
15. **Grupo soluciones GSInnova**. [Online] 2007. [Cited: febrero 10, 2010.] Disponible en web: <http://www.rational.com.ar/index.html>.

16. IEEE. Std 1471-2000.
17. Peláez, Juan. **Blog**. [Online] mayo 29, 2009. [Cited: enero 19, 2010.] Disponible en web: <http://geeks.ms/blogs/jkpelaez/archive/2009/05/29/arquitectura-basada-en-capas.aspx>.
18. Canales, Roberto. **Arquitectura SOA e Integración de aplicaciones**. *Adictos al trabajo*. [Online] 2008. Disponible en web: <http://www.adictosaltrabajo.com/tutoriales/pdfs/soa.pdf>.
19. Lago, Ramiro. **Patrones de diseño de software**. [Online] 2007. [Cited: enero 20, 2010.] Disponible en web: <http://www.proactiva-calidad.com/java/patrones/mvc.html>.
20. **Sitio oficial de NetBeans**. [Online] [Cited: enero 16, 2010.] Disponible en web: http://www.netbeans.org/index_es.html.
21. Zambrano, Raquel. **Innovación y experiencias educativas**. 14, Cordoba : s.n., 2008. ISSN 1988-6047.
22. **Comunidad emagister**. [Online] [Cited: enero 16, 2010.] Disponible en web: <http://www.emagister.com/que-es-oracle-cursos-1053949.htm#programa>.
23. **Sitio oficial de PostgreSQL**. [Online] 2010. [Cited: enero 17, 2010.] Disponible en web: <http://www.postgresql.org/about/>.
24. **Sitio oficial de MySQL**. [Online] 2010. [Cited: enero 12, 2010.] Disponible en web: <http://www.mysql.com/about/>.
25. **Masadelante.com**. [Online] 2010. [Cited: enero 16, 2010.] Disponible en web: <http://www.masadelante.com/faqs/servidor>.
26. **Sitio oficial de Microsoft Office Online**. [Online] 2010. [Cited: enero 16, 2010.] Disponible en web: <http://office.microsoft.com>.
27. Visconti, Marcello y Astudillo, Hernán. **Fundamentos de Ingeniería de Software. Esquemas, Patrones y Persistencia**.
28. LARMAN, Craig. **“UML y Patrones”**. s.l.: Prentice Hall, 2000. ISBN-84-205-3438-2.
29. Gamma, Erich; Helm, Richard; Johnson, Ralph y Vlissides, John. **“Design Patterns”**.
30. **Sitio oficial de Visual Paradigm**. [Online] 2010. [Cited: enero 20, 2010.] Disponible en web: <http://www.visual-paradigm.com/>
31. **Sitio oficial de PHP**. [Online] 2010. [Cited: enero 10, 2010.] Disponible en web: <http://php.net/>.
32. **Sitio oficial de Mozilla**. [Online] 2010. [Cited: febrero 10, 2010.] Disponible en web: <http://www.mozilla-europe.org/es/firefox/>
33. **La Web del Programador: Comunidad de Programadores**. [Online] 2010. [Cited: febrero 10, 2010.] Disponible en web: <http://www.lawebdelprogramador.com/>

34. Smaldone, Javier. **Desarrollo de Software** [Online] 2010. [Cited: febrero 11, 2010.]
Disponible en web: <http://www.smaldone.com.ar/profesional/programacion.shtml>
35. PRESSMAN, Roger. **Ingeniería del Software. Un enfoque práctico**. s.l.: Interamericana, 2002.
36. Álvarez, Miguel Ángel. **Qué es Python**. [Online] 2010. [Cited: febrero 10, 2010.] Disponible en: <http://www.desarrolloweb.com/articulos/1325.php>
37. White, Alexei. **“JavaScript. Programmer’s Reference”**. Wiley Publishing, Inc. 2009. ISBN: 978-0-470-34472-9.
38. Pollock, John. **“JavaScript. A Beginner’s Guide”**. The McGraw-Hill Companies, Inc.2010. ISBN: 978-0-07-163296-6.

GLOSARIO DE TÉRMINOS

API: Application Programming Interface, en español Interfaz de Programación de Aplicaciones, es un conjunto de funciones residentes en bibliotecas que permiten que una aplicación corra bajo un determinado sistema operativo.

Automake: herramienta de programación que produce programas portables usados en la compilación de SW.

DCA_CU: Diagrama de Clases del Análisis del Caso de Uso.

DCD_CU: Diagrama de Clases del Diseño del Caso de Uso.

FLV: Flash Video, es el formato de video de Macromedia para transmisión de video digital por Internet.

JSON: JavaScript Object Notation, es un formato ligero para el intercambio de datos.

Libtool: herramienta libre para crear librerías de SW portable.

LOM: Learning Object Metadata, en español Metadatos para Objetos de Aprendizaje, es el estándar de metadatos para objetos de aprendizaje.

LTS: Long Term Support, en español Soporte de Tiempo Prolongado, quiere decir que se podrá tener soporte y actualizaciones durante un largo tiempo.

MIT: es una licencia libre que permite cualquier uso del framework para fines comerciales, y no obliga a que el código resultante se tenga que publicar bajo esta misma licencia *open source*.

Open Source: en español código abierto, es el término con el que se conoce al software desarrollado y distribuido libremente.

OS: Operating System, en español Sistema Operativo, programa informático que actúa de interfaz entre los dispositivos de hardware y los programas usados por el usuario para manejar la computadora.

PDF: Portable Document Format, formato de almacenamiento de documentos.

SCORM: Shareable Content Object Reference Model, es un conjunto de especificaciones para desarrollo, empaquetamiento y distribución de material educativo.

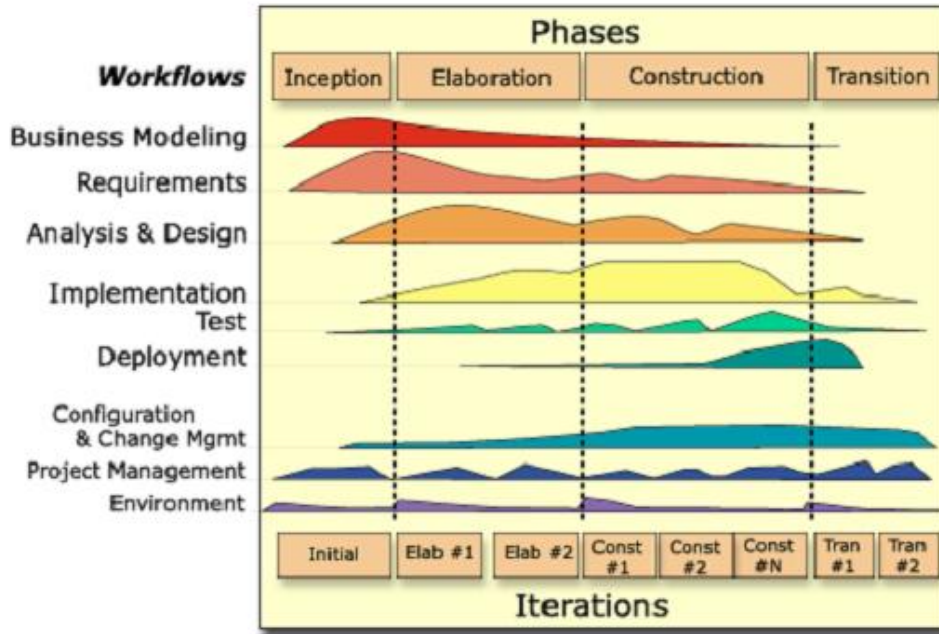
SW: Software, programas de sistema, utilerías o aplicaciones expresados en un lenguaje de máquina.

XML: Extensible Markup Language, en español Lenguaje de Etiquetado Extensible, es un conjunto de reglas para definir etiquetas semánticas que organizan un documento en diferentes partes.

XSS: en inglés Cross-site Scripting es un tipo de inseguridad informática o agujero de seguridad basado en la explotación de vulnerabilidades del sistema de validación de HTML incrustado. Las vulnerabilidades de XSS ocurren cuando un atacante utiliza una aplicación web para mandar código malicioso, generalmente en la forma de un script, a un usuario diferente.

ANEXOS

Anexo 1: Rational Unified Process (RUP)



Anexo 2: Gráfica de los servidores web más usados en el mercado del 95 hasta el 2010

