

Desarrollo del módulo iTopics de la plataforma educativa ZERA.

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

Autores

Mayliuvis Esquijarrova Valdés

Ernesto Vladimir Pereda Díaz

Tutora

Ing. Mairelis Gari Maribona

Cotutores

Ing. Osbel Montero Pérez

Ing. Yudanis Gago Martínez

Catálogo...

@ Sitios de Interés

¡! Anuncios...

Autenticarse

La Habana, junio 2011

“Año 53 de la Revolución”

Declaración de autoría

Declaramos que somos los autores de este trabajo y autorizamos a la Facultad 4 de la Universidad de las Ciencias Informáticas, así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los _____ días del mes de _____ del año 2011.

Autora

Mayliuvis Esquijarroza Valdés

Autor

Ernesto Vladimir Pereda Díaz

Tutora

Ing. Mairelis Gari Maribona

Cotutor

Ing. Osbel Montero Pérez

Cotutor

Ing. Yudanis Gago Martínez

Resumen

El avance de las Tecnologías de la Información y las Comunicaciones (TIC) ha tenido un impacto significativo en la esfera de la educación. Se han incorporado nuevas herramientas logrando un nuevo paradigma educativo más personalizado y centrado en la actividad de los estudiantes, entre las que se destacan los Sistemas de Gestión de Aprendizaje. La Universidad de las Ciencias Informáticas no está exenta a este desarrollo, por lo que se traza la tarea de implementar una plataforma para la gestión del aprendizaje. El objetivo de este trabajo es desarrollar el módulo iTopics para la plataforma educativa ZERA, permitiendo así una mejor visualización e interacción con sus contenidos. Para lograrlo, primeramente se realizó un estudio de las principales tendencias en la gestión y visualización de contenidos para el aprendizaje. Además, se investigaron las principales tecnologías y herramientas usadas a nivel mundial para el desarrollo de plataformas educativas. Seguidamente se transitó siguiendo la metodología de desarrollo RUP por los flujos de trabajo modelado del negocio, requisitos, análisis y diseño, implementación y prueba; obteniéndose en cada caso los artefactos propuestos por dicha metodología. Se realizaron diferentes pruebas aplicando el método de caja negra con el objetivo de examinar exhaustivamente el sistema. Finalmente, se logró un módulo que posee entre sus principales funcionalidades la gestión de notas, el resaltado de texto, el registro del avance en el contenido, un sistema de paginado para la navegación, un sistema de mensajería instantánea entre otras.

Palabras Clave: aprendizaje, contenido, hiperentorno, visualización de contenidos, resaltado de texto.

Índice

Introducción	1
Capítulo 1 Fundamento Teórico	7
1.1 Introducción.....	7
1.2 Sistemas similares.....	7
1.2.1 Sistemas a nivel internacional.....	7
1.2.2 Sistemas a nivel nacional.....	9
1.3 Proceso de desarrollo.....	10
1.3.1 Proceso Unificado de Desarrollo	11
1.3.2 Extreme Programming (XP)	12
1.3.3 SCRUM.....	12
1.4 Lenguaje de Modelado	13
1.5 Lenguajes de desarrollo	15
1.6 Frameworks.....	17
1.6.1 Capa de Presentación.....	18
1.6.2 Capa de Lógica de Negocio	19
1.6.3 Capa de Acceso a Datos	20
1.7 Sistema Gestor de Base de Datos.....	21
1.8 Servidor Web.....	21
1.9 Entorno de desarrollo	22
1.10 Herramientas CASE para el modelado UML.....	22
1.10.1 ArgoUML.....	22
1.10.2 Visual Paradigm.....	23
1.11 Conclusiones.....	24

Capítulo 2 Características del sistema.....	25
2.1 Introducción.....	25
2.2 Modelo de Dominio.....	25
2.2.1 Conceptos del Dominio	25
2.2.2 Diagrama del Modelo de Dominio	27
2.3 Descripción del Sistema Propuesto	28
2.4 Requisitos del software.....	28
2.5 Modelo de Casos de Uso del Sistema	34
2.5.1 Descripción de los actores del sistema	34
2.5.2 Patrones de casos de uso.....	35
2.5.3 Diagrama de actores del sistema	35
2.5.4 Descripción de los casos de uso del sistema	36
2.5.5 Diagrama de casos de uso del sistema.....	41
2.6 Conclusiones.....	42
Capítulo 3. Análisis y diseño de la propuesta de sistema.....	43
3.1 Introducción.....	43
3.2 Modelo de análisis.....	43
3.2.1 Diagramas de clases del análisis	43
3.3 Patrón arquitectónico Modelo – Vista – Controlador en symfony	46
3.4 Aplicación de los patrones de diseño en Symfony	47
3.5 Modelo de diseño	48
3.5.1 Diagrama de clases del diseño	48
3.6 Diagrama de despliegue.....	49
3.7 Diseño de la base de datos	49
3.8 Conclusiones.....	50

Capítulo 4. Implementación y Prueba.	51
4.1 Introducción.....	51
4.2 Modelo de implementación.....	51
4.2.1 Diagrama de componentes.....	51
4.3 Pruebas de software.....	52
4.3.1 Niveles de Prueba.....	53
4.3.2 Métodos de Prueba.....	54
4.3.3 Diseño de Casos de Prueba.....	55
4.3.4 Resultados Obtenidos.....	65
4.5 Conclusiones.....	67
Conclusiones.....	68
Recomendaciones.....	69
Referencias bibliográficas.....	70
Bibliografía.....	74

Introducción

La era de Internet exige cambios en el mundo educativo y los profesionales de la educación tienen múltiples razones para aprovechar las nuevas posibilidades que proporcionan las Tecnologías de la Información y las Comunicaciones (TIC). Para impulsar este cambio hacia un nuevo paradigma educativo más personalizado y centrado en la actividad de los estudiantes, se han incorporado herramientas entre las que se destacan los Sistemas de Gestión de Aprendizaje, en inglés Learning Management System (en lo adelante LMS).

Los LMS son plataformas que se encargan fundamentalmente de administrar, distribuir y controlar el proceso de enseñanza en línea, para brindar el soporte necesario al escenario de aprendizaje apoyado en las TIC. Constituyen aplicaciones Web que proveen las funciones administrativas y de seguimiento necesario para posibilitar y controlar el acceso a los contenidos, implementar recursos de comunicación y llevar a cabo el seguimiento de quienes utilizan la herramienta. En general, los LMS facilitan la interacción entre los docentes y los estudiantes, aportan herramientas para la gestión de contenidos académicos y permiten el seguimiento y la evaluación. Es decir, facilitan la “simulación” del modelo real en el mundo virtual, por lo que también se les conoce como Virtual Learning Environment (VLE). (1)

En sus inicios los LMS fueron diseñados para su uso en la educación superior o universitaria y en la capacitación profesional, esta es una de las razones por las cuales el grado de interactividad en ellos es aún, en su gran mayoría, limitado debido a la audiencia para la que fueron concebidos y dadas las tecnologías de desarrollo utilizadas en los inicios de su desarrollo.

Para el apoyo al proceso de enseñanza – aprendizaje en sistemas educacionales básicos en Cuba (primaria, secundaria y preuniversitaria) se desarrolló software con tecnologías multimedia, con el objetivo de utilizar las potencialidades de estas y de suplir las limitaciones de los LMS para el desarrollo de cursos en estos niveles educativos, propiciando además altos niveles de interactividad entre los usuarios y el software. Esta tecnología además de incorporar numerosas ventajas, pasó del ambiente en línea de los LMS al ambiente completamente desconectado del software multimedia. Los LMS comenzaron a implementar estándares educativos para la interoperabilidad entre ellos, se comenzaron a basar en teorías constructivas del conocimiento a través de instrumentos y herramientas que propician el

aprendizaje colaborativo. Finalmente, con el surgimiento de la Web 2.0 las principales limitaciones de los LMS fueron desapareciendo lo que dio lugar a la disminución gradual del software educativo multimedia.

Posteriormente surge una nueva tendencia, que es la de integrar en un mismo producto, todas o algunas de las tipologías de software educativos. A este nuevo modelo de software se le ha denominado Hiperentorno Educativo o de Aprendizaje, lo cual no es más que un sistema informático basado en tecnología hipertexto que contiene una mezcla de elementos representativos de diversas tipologías de software educativo tales como: tutorial, entrenador, evaluador, simulador, juego instructivo, libro electrónico entre otros. (2)

En este marco, la Universidad de las Ciencias Informáticas se traza la tarea de implementar una plataforma para la gestión del aprendizaje, que integre los principales conceptos de los hiperentornos y que en un futuro se convierta en una potente herramienta, capaz de adaptarse a los procesos del negocio de cualquier institución educativa.

Dentro del hiperentorno de aprendizaje se pretende incluir los Contenidos Educativos pertenecientes a la Colección Futuro. Los contenidos de esta colección son diversos, se destacan: Eureka (para Matemática), Redox (para Química), Sustancia y Campo (para Física), entre otros. Surge la necesidad de lograr una plataforma educativa, la cual debe contar con una navegación sencilla dentro de los capítulos, temas y subtemas, y que en la resolución de los ejercicios pertenecientes a los niveles anteriormente mencionados, no sea necesario volver a recorrer el árbol de navegación para situarse en la actividad correspondiente.

En la visualización de los contenidos de la plataforma ZERA se torna como una necesidad que los usuarios puedan acceder al último contenido visitado, lo cual evitará que estos pierdan tiempo en la lectura de contenidos ya vencidos para lograr ubicarse en el lugar de avance. Además, sería de vital importancia que puedan realizar comentarios asociados al contenido para que sean consultados posteriormente y resaltar dentro de este la información que sea de importancia, lo cual traerá consigo un mejor entendimiento por parte del usuario debido a la posibilidad de particularizar la visualización de los contenidos de la plataforma.

Aprovechando las nuevas características de la web 2.0, que son una combinación de las ventajas del software multimedia y de los LMS, se darán respuestas a las problemáticas planteadas anteriormente.

El **problema a resolver** quedaría resumido en la siguiente interrogante: ¿Cómo visualizar los contenidos y recursos interactivos, así como la gestión de apuntes y marcadores en la plataforma educativa ZERA?

El **objeto de estudio** fijado es el proceso de visualización de contenidos en los Sistemas de Gestión de Aprendizaje.

El **objetivo general** del presente trabajo es desarrollar el módulo iTopics para la plataforma educativa ZERA, que permita una mejor visualización e interacción de los contenidos.

Del objetivo general se derivan los siguientes **objetivos específicos**:

1. Realizar un análisis del estado del arte acerca de las tendencias tecnológicas actuales y estándares más adoptados en la visualización de los contenidos y recursos interactivos a nivel mundial.
2. Realizar el análisis y diseño del módulo iTopics a partir de los resultados obtenidos en la fase de requerimientos.
3. Implementar el módulo diseñado de acuerdo con la estructura de diseño definida.
4. Realizar pruebas de calidad a lo largo de todo el ciclo de desarrollo.

El **campo de acción** en que se enmarca el trabajo es el proceso de visualización de contenidos en la plataforma educativa ZERA.

Para guiar la investigación se plantea la siguiente **idea a defender**: Si se implementa un módulo que permita una visualización de recursos interactivos, así como la gestión de apuntes y marcadores, se podrá lograr una mejor interacción de los usuarios con el contenido.

Para dar cumplimiento a los objetivos específicos se planifican las siguientes tareas:

1. Investigación acerca de otras aplicaciones o soluciones similares.

2. Análisis y definición de los lenguajes de programación a utilizar, así como las herramientas de desarrollo.
3. Investigación sobre los diferentes frameworks de desarrollo más usados a nivel mundial, así como acerca de plugins y componentes con vista al desarrollo de elementos interactivos.
4. Análisis y definición de la metodología de desarrollo a utilizar, así como las herramientas de modelado.
5. Análisis de la propuesta de arquitectura desarrollada para la plataforma educativa ZERA.
6. Identificación y especificación de los casos de uso asociados al módulo iTopics de la plataforma educativa ZERA.
7. Realización del modelo de casos de uso que da cumplimiento a los requisitos funcionales y no funcionales asociados al módulo iTopics de la plataforma educativa ZERA.
8. Realización de los diagramas de clases de análisis y diseño para cada caso de uso del sistema.
9. Diseño del modelo de datos correspondiente al módulo propuesto.
10. Implementación de las clases definidas en el modelo de clases del diseño.
11. Implementación de la visualización de los recursos interactivos y elementos de navegabilidad.
12. Integración de la solución obtenida al resto de los módulos de la plataforma educativa ZERA.
13. Diseño de los casos de prueba.
14. Realización de las pruebas unitarias al código.
15. Realización de las pruebas funcionales al módulo iTopics.

Métodos investigativos

Para llevar a cabo la investigación, se emplean los métodos de nivel teórico y empírico.

Métodos teóricos:

- ✓ *Análisis Histórico-Lógico:* En la presente investigación se utiliza este método para realizar el estudio del estado del arte, o sea, para investigar acerca de otras aplicaciones o soluciones

similares y de los lenguajes y metodologías de desarrollo de software existentes, así como los frameworks (marcos de trabajo) y herramientas de desarrollo; describir la metodología, herramientas y lenguajes a utilizar en el análisis, diseño e implementación del módulo iTopics de la plataforma educativa ZERA y estudiar la propuesta de arquitectura para dicha plataforma.

- ✓ *Modelación:* El uso de este método permitirá: realizar el modelo de Casos de Uso que da cumplimiento a los requisitos funcionales y no funcionales asociados al módulo iTopics de la plataforma educativa ZERA, realizar los diagramas de clases de diseño y diagramas de interacción para cada caso de uso aplicando patrones de diseño, diseñar el modelo de datos y la distribución física del sistema.

Métodos empíricos:

- ✓ *Observación:* Este método es el instrumento que permite estudiar más de cerca el objeto de la investigación, las acciones, causas, consecuencias, etc. Se puede observar cómo funciona el proceso de visualización de contenidos y los principales problemas asociados a este.

Resultados esperados:

Con el desarrollo del presente trabajo de diploma se pretende lograr:

- ✓ La integración del módulo iTopics con el resto de los módulos de la plataforma educativa ZERA.
- ✓ Un diseño de los contenidos más instructivo e interactivo para los estudiantes.
- ✓ Mayor interacción entre los usuarios.
- ✓ Mejor navegabilidad entre los niveles del contenido y por las páginas dentro de estos.
- ✓ Un sistema de ubicación donde la plataforma pueda guardar el lugar del contenido donde se quedaron navegando los usuarios para una posterior consulta.
- ✓ Un sistema de resaltado de la información dentro del contenido.
- ✓ Una gestión de apuntes dentro del contenido en tiempo real y personalizable.
- ✓ Toda la documentación referente al desarrollo del módulo iTopics y de su integración al resto de los módulos de la plataforma educativa ZERA.

Estructura Capitular

Capítulo 1: Fundamento Teórico. Se exponen los elementos teóricos que sustentan el problema científico y los objetivos del trabajo. Se realiza un análisis de las metodologías y herramientas de desarrollo que se pueden utilizar y se justifica la selección de cada una de ellas.

Capítulo 2: Características del Sistema. Se especifican los requisitos que debe cumplir el sistema así como el diagrama de casos de uso y las descripciones de ellos.

Capítulo 3: Análisis y diseño del módulo iTopics. Se realiza el diseño del sistema, así como los diagramas de clases con estereotipos web que brindan una visión clara del producto.

Capítulo 4: Implementación y Pruebas. Los aspectos del desarrollo son tratados en este capítulo. Se describe cómo está implementado el sistema. Además, se definen los tipos de pruebas y los casos de prueba que se le realizarán al software.

Capítulo 1 Fundamento Teórico

1.1 Introducción

Al iniciarse el desarrollo de un producto de software los primeros pasos están en función de definir el tipo de aplicación y su ambiente de desarrollo. Por lo que se llevará a cabo el estudio de las soluciones similares ya existentes con respecto a la visualización de recursos educativos (videos, imágenes, applets, sonidos, palabras calientes, etc.), gestión de notas dentro del contenido, resaltado de texto, avance del contenido, y las principales tendencias de estas funcionalidades en la producción de aplicaciones educativas, que surgen como resultado del estudio analítico del estado del arte. También se estudiarán cuestiones referentes a las tecnologías y herramientas que se usan para dar cumplimiento al objetivo general de la investigación. Se señala que esta investigación se realiza con el fin de tomar ideas y conocer los patrones de conducta de estas herramientas, de forma que la investigación aporte una guía para el desarrollo del módulo iTopics perteneciente a la plataforma educativa ZERA.

1.2 Sistemas similares

1.2.1 Sistemas a nivel internacional

Entorno de Aprendizaje Dinámico Orientado a Objetos y Modular

Moodle (en inglés Modular Object-Oriented Dynamic Learning Environment) es un Sistema de Gestión de Cursos de Código Abierto (Open Source Course Management System), conocido también como Sistema de Gestión del Aprendizaje (Learning Management System, LMS) o como Entorno de Aprendizaje Virtual (Virtual Learning Environment, VLE). Integra estándares actuales como SCORM. Para utilizarlo, necesita ser instalado en un servidor web, puede ser instalado tanto en un ordenador personal como en un servidor proporcionado por una compañía de hospedaje de páginas web (3).

Esta plataforma presenta las siguientes funcionalidades:

- ✓ Permite al profesor activar los capítulos cuando lo necesite para que los alumnos accedan a ellos progresivamente a lo largo del curso.
- ✓ El profesor puede publicar archivos para consultarlos él mismo o para otros miembros del curso.

Plataforma Educativa Claroline

Constituye una plataforma de aprendizaje virtual (eLearning) y de trabajo virtual (eWorking). Claroline está basada en herramientas y lenguajes libres como PHP e integra estándares actuales como SCORM para intercambiar contenidos. Esta plataforma está publicada bajo una licencia Open Source (de código abierto o software libre), permite crear y administrar cursos a cientos de organizaciones de 93 países diferentes y la colaboración de espacios online. (4)

Claroline les permite a los profesores realizar operaciones tales como:

- ✓ Escribir un anuncio que pueda ser visualizado en la página de inicio del curso.
- ✓ Publicar archivos que puedan ser de interés para otros miembros del curso.
- ✓ Editar zona de texto fácilmente, añadir imágenes y enlaces, modificar y resaltar textos.

Plataforma Helvia

Helvia es una plataforma educativa para los centros educativos TIC de Andalucía. Constituye una herramienta desarrollada en software libre con la cual los centros educativos pueden desarrollar todo un sistema telemático para la organización y funcionamiento de la comunidad escolar. (5)

Presenta funcionalidades como son:

- ✓ Edición de los contenidos, creación de hipertextos, imágenes, etc. Permite incluir recursos en general sin tener que recrearlos manualmente, así como incorporar por otros docentes aquellos recursos que estén incluidos en el repositorio local (6).
- ✓ Publicación libre de un diario personal, o comunicaciones públicas individuales o grupales (bitácora), desde una única entrada identificada.

Plataforma ELEVEN

Eleven es una plataforma educativa multieditorial, integral y universal que permite a los centros educativos el acceso a todo tipo de contenidos digitales, herramientas de gestión didáctica y funcionalidades docentes centralizadas en una única aplicación.

Con Eleven, tanto el centro como el profesor pueden crear contenidos propios, ejercicios, guías didácticas y repositorios personalizados de archivos, e integrarlos en los contenidos y recursos que ofrecen las diferentes editoriales, independientemente de cuál sea su formato, tipo, tecnología o sistema. El profesorado puede controlar y centralizar desde la plataforma todos los aspectos de gestión de sus alumnos, grupos y asignaturas de manera integral y unificada. No sólo integra contenidos en PDF o PDF enriquecido, sino también en SCORM y en los diferentes formatos propios de cada editorial. (7)

Esta plataforma brinda además:

- ✓ Un Blog para incorporar comentarios y hacer el seguimiento didáctico de la asignatura.
- ✓ Servicio interno de mensajería.
- ✓ Un repositorio de archivos para compartir con toda la clase.

1.2.2 Sistemas a nivel nacional

aprenDIST: Potente plataforma de educación a distancia, que comenzó a revolucionar el proceso de educación existente. Surgió a finales del año 2000 como parte de la firma de un contrato de la empresa española Imago Mundi, el Centro de Estudios de Ingeniería de Sistemas (CEIS) del Instituto Superior Politécnico José Antonio Echevarría (ISPJAE) de Cuba y el Centro de Tecnologías Avanzadas (CETA), para la creación de una nueva herramienta que fuese cómoda y flexible para impartir cursos a distancia en un entorno Web, favoreciendo en alto grado la interacción entre los participantes en el proceso de enseñanza y aprendizaje. Con el surgimiento de este sistema informático, en el Instituto Superior Politécnico “José Antonio Echevarría”, se han multiplicado las alternativas antes existentes, para desarrollar aquellos cursos (postgrados, pregrados semipresenciales, etc.) que se ajustan a las ventajas que ofrece la educación a distancia en cuanto a horario y lugares de concurrencia, que de manera abarcadora posee aprenDIST, haciendo uso de las TIC. (8)

Esta plataforma desarrollada en Cuba, cuenta con dos versiones; versión estática y versión online, presenta funcionalidades similares a las que se necesitan, por lo que es un punto de partida importante el tomarla de apoyo, debido a que tiene buenos resultados y ha sido puesta en explotación por un largo período de tiempo.

Valoración de las plataformas

Las funcionalidades analizadas en cada una de las plataformas estudiadas tienen características similares a las solicitadas en los requisitos del cliente, por esta razón fueron tomados como referencia para el desarrollo de la Plataforma ZERA. Dentro de las características que sirvieron de apoyo para el desarrollo del módulo iTopics están: la forma de visualizar los contenidos, los recursos y la información importante, que medios de comunicación utilizan y cómo lo hacen, la forma en que comparten información y se retroalimentan estudiantes y profesores. Algunas características que se detectaron ausentes en estas plataformas fueron: permitir resaltar dinámicamente un texto previamente seleccionado dentro del contenido que a juicio del usuario sea importante, así como dar la posibilidad al usuario profesor de adjuntarle a este texto un comentario y un recurso sin tener que editar el contenido, es decir, de forma dinámica mientras navega por este.

1.3 Proceso de desarrollo

En los últimos tiempos la cantidad y variedad de los procesos de desarrollo ha aumentado de forma impresionante, sobre todo teniendo en cuenta el tiempo que estuvo en vigor como ley única el famoso desarrollo en cascada. Se podría decir que en estos últimos años se han desarrollado dos corrientes en lo referente a los procesos de desarrollo, los llamados métodos pesados y los métodos ligeros. La diferencia fundamental entre ambos es que mientras los métodos pesados intentan conseguir el objetivo común por medio de orden y documentación, los métodos ligeros (también denominados métodos ágiles), tratan de mejorar la calidad del software por medio de una comunicación directa e inmediata entre las personas que intervienen en el proceso.

En la actualidad existen una gran variedad de metodologías orientadas al desarrollo de software, entre las que se pueden citar: Rational Unified Process (RUP), Extreme Programming (XP) y SCRUM las cuales constituyen en la actualidad metodologías líderes para guiar el proceso de desarrollo de software, razón por la cual serán estudiadas a continuación.

1.3.1 Proceso Unificado de Desarrollo

El Proceso Unificado de Desarrollo (RUP, por sus siglas en inglés) es una de las metodologías más generales que existen actualmente, su finalidad no está restringida a guiar el desarrollo de software, sino cualquier tipo de proyecto. La estrategia de este proceso es conseguir su objetivo por medio de orden y documentación, lo que lo convierte en un fiel exponente de los métodos pesados. RUP define cuatro fases (inicio, elaboración, construcción y transición) y dentro de cada una de ellas el equipo de trabajo pasa por todos los flujos que son transversales a las fases, inclusive en varias iteraciones.

RUP se caracteriza por ser **dirigido por casos de uso**, permitiendo a través de estos definir lo que el usuario desea a partir de la captura de requisitos. Es **centrado en la arquitectura**, lo cual brinda una visión completa del sistema, a través de la descripción de los procesos del negocio que son más importantes, para comprenderlo, desarrollarlo y producirlo de una forma eficaz. **Iterativo e Incremental** donde cada fase se desarrolla en iteraciones, de forma tal que se pueda dividir en pequeños proyectos mejorando su comprensión y desarrollo.

Un proyecto realizado siguiendo RUP se divide en cuatro fases:

1. Inicio: La cual brinda la posibilidad de definir el alcance y objetivo del proyecto.
2. Elaboración: Permite definir la arquitectura del sistema, lo cual provee una base estable para el diseño e implementación de la siguiente fase.
3. Construcción: En la que se obtiene la capacidad operativa del producto completando el desarrollo del sistema basado en la línea base de la arquitectura.
4. Transición: Se realiza una liberación del producto garantizando que esté listo para ser instalado y entregado a los usuarios

RUP es más adecuado para proyectos grandes, dado que requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas; en proyectos pequeños es posible que no se pueda cubrir los costos de dedicación del equipo de profesionales necesarios. Los requerimientos de los diversos

inversores pueden ser diferentes, contradictorios o disputarse recursos limitados, así que debe encontrarse un balance que satisfaga los deseos de todos y el control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción (9).

Por tal motivo esta metodología garantiza obtener los artefactos necesarios durante el desarrollo de la presente investigación que derivan en un producto con calidad, bien documentado y ejecutado en tiempo.

1.3.2 Extreme Programming (XP)

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software que promueve el trabajo en equipo, se preocupa por el aprendizaje de los desarrolladores y propicia un buen clima de trabajo. Se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.

XP se define como una metodología especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes donde existe un alto riesgo técnico. (10)

Una característica distintiva de XP es la programación en parejas, con el objetivo de que el código sea revisado y validado antes de ser escrito; la refactorización de código está presente durante todo el desarrollo, y no es más que escribir el mismo código fuente nuevamente buscando claridad, pero sin cambiar la funcionalidad resultante. Las parejas no serán siempre las mismas, sino que se pretende que cada desarrollador haya formado dupla al menos una vez con todos los demás, de donde se desprende que el código es de propiedad colectiva y cada uno es responsable por todo el proyecto. (10)

1.3.3 SCRUM

SCRUM es una metodología ágil de desarrollo de software que plantea que los ciclos de desarrollo de los proyectos sean lo más rápidos posible. Esta metodología se ajusta a los cambios que puedan presentarse y mantiene una estrecha relación con el cliente. Establece un desarrollo incremental del producto con entregas frecuentes al cliente, garantizando así que el mismo se encuentre al tanto del progreso del sistema.

SCRUM es muy fácil de aprender, ya que requiere de muy poco esfuerzo para utilizarlo, pero no se tiene una perspectiva global del proyecto más allá de una lista de tareas. Además, no genera toda la documentación que se obtiene con otras metodologías por lo que no es muy conveniente utilizarlo solo.

Selección de la metodología para guiar el desarrollo del software

XP es una metodología que establece que tanto cliente, como jefe de proyecto y desarrolladores son partes del equipo de trabajo y se encuentran implicados en el desarrollo del software. En este caso el cliente no forma parte del grupo de desarrollo, por lo que usar esta metodología sería incumplir con un requisito fundamental que esta exige para que el software tenga la calidad requerida. XP como SCRUM están destinadas a proyectos de corto plazo y se basan en iteraciones pequeñas a diferencia de RUP, ya que esta última metodología está concebida para la realización de proyectos y equipos de trabajo grandes, en cuanto a tamaño y duración. En el caso de RUP mantiene al equipo de trabajo enfocado en producir software en tiempo, con las características y calidad requerida. Además, basa su trabajo principalmente en la documentación del software y expone un conjunto de actividades orientadas a visualizar, especificar, construir y documentar los artefactos necesarios para el desarrollo de aplicaciones. Por todas estas razones se considera la elección de RUP como metodología que guíe el proceso de desarrollo del software.

1.4 Lenguaje de Modelado

Entre los lenguajes de Modelado estudiados se encuentran: Lenguaje para la Modelación de Aplicaciones Multimedia Educativas (**ApEM-L** por sus siglas en inglés), Lenguaje para la modelación Orientada a Objetos de Aplicaciones Multimedia (**OMMMA – L** por sus siglas en inglés) y el Lenguaje Unificado de Modelado (**UML** por sus siglas en inglés), siendo este último el empleado para modelar el sistema a desarrollar debido a que tanto ApEM-L como OMMMA-L se usan netamente para modelar aplicaciones con tecnologías multimedia. Razón por la cual no se detallan características de ambos en la presente investigación.

UML constituye el estándar internacionalmente más aceptado. Es un lenguaje de modelado visual que se usa para visualizar, especificar, construir y documentar artefactos de un sistema de software. (11)

Principales características de UML

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

El Lenguaje Unificado de Modelado proporciona a los desarrolladores un vocabulario que incluye tres categorías: elementos, relaciones y diagramas (11)

RUP propone usar UML para llevar la documentación del sistema, facilitar la etapa del diseño y posterior construcción o desarrollo, transmitir ideas y ayudar al equipo a comunicarlas lo cual constituyó un elemento de peso a la hora de su selección.

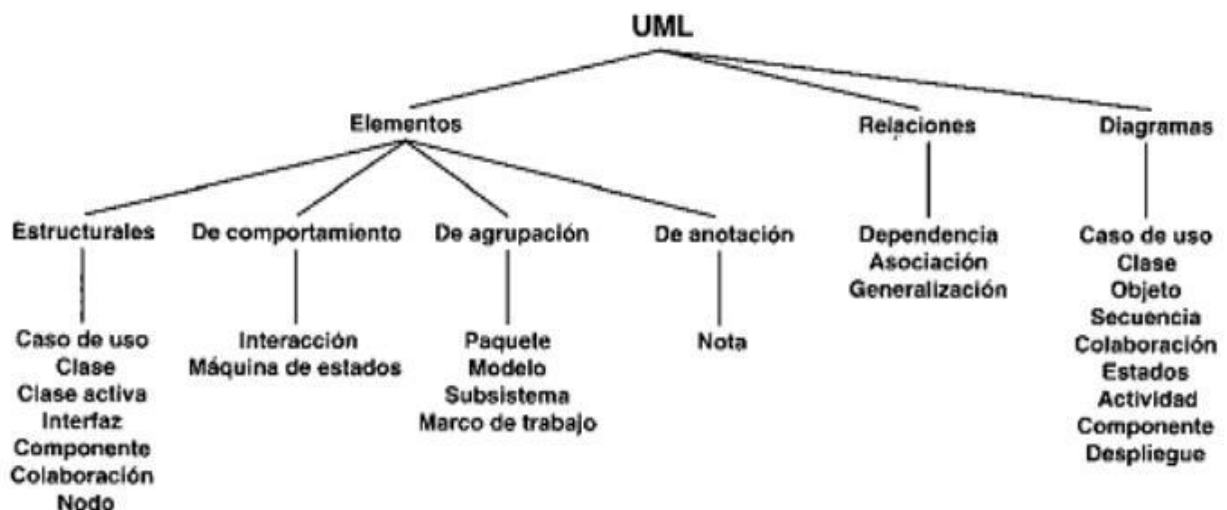


Figura 1. Vocabulario de UML.

1.5 Lenguajes de desarrollo

En la actualidad, el uso de Ajax (de Asynchronous JavaScript and XML) para el desarrollo de aplicaciones web se toma prácticamente como una necesidad debido a las ventajas que reporta, entre las cuales se encuentran: mayor dinamismo al software, mejoras en tiempo de respuesta, mejor interacción entre el sistema y el usuario, etc. Razón por la cual se tienen en cuenta para el desarrollo de la aplicación web un conjunto de tecnologías que al unirse forman a Ajax.

Ajax: En realidad el término es un acrónimo de Asynchronous JavaScript and XML, que se puede traducir como “JavaScript asíncrono y XML”. En el artículo publicado por Jesse James Garrett en el 2005 define Ajax como: “Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes”. Las tecnologías que forman Ajax son XHTML y CSS para crear una presentación basada en estándares, DOM para la interacción y manipulación dinámica de la presentación. XML, XSLT y JSON para el intercambio y la manipulación de información, XMLHttpRequest para el intercambio asíncrono de la información y JavaScript para unir todas las demás tecnologías. (12)

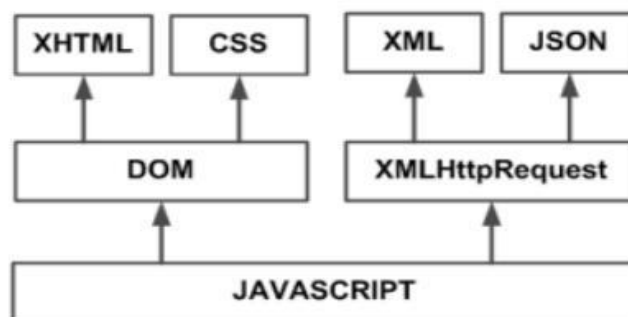


Figura 2. Tecnologías agrupadas bajo el concepto de Ajax.

Tecnologías del lado del cliente

XHTML (Lenguaje de Marcado de Hipertexto Extensible) **Versión 1.1:** El lenguaje XHTML es muy similar al lenguaje HTML (Lenguaje de Marcado de Hipertexto). De hecho, XHTML no es más que una adaptación de HTML al lenguaje XML, sólo añade pequeñas mejoras y modificaciones menores. Técnicamente, HTML es descendiente directo del lenguaje SGML (Lenguaje de Marcado Generalizado),

mientras que XHTML lo es de XML (que a su vez también es descendiente de SGML). (13) Este lenguaje proporciona una mejor velocidad de navegación ya que las páginas creadas y maquetadas correctamente se arman más rápido en los exploradores que si no lo están, ya sea porque hay errores de sintaxis o porque se usan elementos con un fin para el cual no fueron creados. Además, proporciona un mejor entendimiento del código principalmente para los diseñadores; pudiéndose actualizar fácilmente la web.

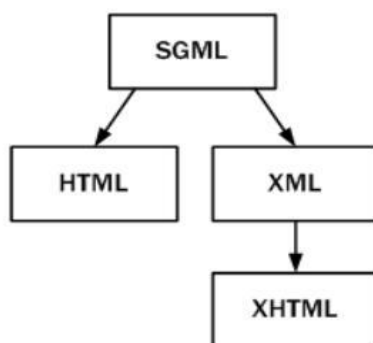


Figura 3. Esquema de evolución HTML y XHTML.

XML (Lenguaje de Etiquetado Extensible) **Versión 1.1:** Es un lenguaje muy simple pero estricto, representa un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML permite la lectura de datos a través de diferentes aplicaciones que usen este. (14)

CSS (Hojas de Estilo en Cascada) **Versión 2:** Es un lenguaje de hojas de estilo creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados “documentos semánticos”). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes. (15)

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

usuario. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java. Legalmente, JavaScript es una marca registrada de la empresa Sun Microsystems. (16)

Lenguaje del lado del servidor:

Luego de investigar acerca de los lenguajes del lado del servidor, específicamente de PHP **Versión 5.3.3** y ASP.NET, se han encontrado semejanzas entre estos como: son de uso gratis, se utilizan para construir páginas web dinámicas, presentan soporte para una gran cantidad de bases de datos (MySQL, Oracle, PostgreSQL entre otras, siendo esta última un requisito no funcional de la plataforma educativa ZERA), soportan el lenguaje XML y trabajan con programación OO. Se decide utilizar PHP debido a que es un lenguaje libre, permite integración con varias bibliotecas externas lo que posibilita agregar funcionalidades nuevas. Como producto de código abierto PHP goza de la ayuda de un gran grupo de programadores que permite que los fallos de funcionamiento se encuentren y reparen rápidamente, a diferencia de ASP.NET que solo puede ser asistido por Microsoft. El código PHP se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar sus capacidades. (17)

Otra característica que se tiene en cuenta para la selección es el servidor que debe dar soporte al lenguaje, siendo PHP soportado por varios servidores, a diferencia del ASP.NET que funciona solo sobre el servidor Microsoft IIS, lo cual es una ventaja sobre este último lenguaje.

1.6 Frameworks

El concepto framework se emplea en muchos ámbitos del desarrollo de sistemas de software. Se pueden encontrar frameworks para el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de juegos, y para cualquier otro ámbito.

En general, el término framework se refiere a una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras se puede considerar como una aplicación genérica incompleta y configurable a la que se le puede añadir las últimas

piezas para construir una aplicación concreta. Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar el código ya existente y promover buenas prácticas de desarrollo con el uso de patrones. Un framework Web, por tanto, se puede definir como un conjunto de componentes que forman parte de un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web. (18)

1.6.1 Capa de Presentación

Dentro de los frameworks de la capa de presentación relacionados con javascript, se estudiaron EXT y JQuery. Ambos tienen características similares como son: simplifican la manera de interactuar con los documentos HTML, manipulan el árbol de objetos dinámicos (DOM), manejan eventos, desarrollan animaciones y agregan interacción con la tecnología AJAX a páginas web, son libres, gratis y de código abierto, son compatibles con los navegadores Internet Explorer 7, Firefox 3.6, Opera 10.00, Safari 4.0 y Chrome 7.0 (19), siendo estos requisitos no funcionales de la plataforma ZERA.

Estas características posibilitan que los contenidos se puedan mostrar y se interactúe con estos de forma dinámica, por lo que pueden usarse los frameworks anteriormente mencionados para apoyar el cumplimiento de las funcionalidades con las cuales debe contar el módulo a desarrollar.

Se decide utilizar JQuery debido a que es fácil de usar y aprender, el equipo de trabajo tiene experiencia en el uso de este framework, se integra con el lenguaje PHP y con el framework Symfony, presenta varios plugins que extienden sus funcionalidades, además de la amplia documentación y comunidad de usuarios, aspectos estos muy importantes en la retroalimentación y rectificación de errores con el uso de esta herramienta.

Se seleccionó también Blueprint como framework de la capa de presentación. Este es un complemento para desarrolladores de aplicaciones web que aumenta la productividad en las tareas de diseño y maquetación de páginas. Se trata de una serie de librerías de hojas de estilo en cascada que contienen código útil para maquetar una página web y aplicar otros tipos de estilos tipográficos o de impresión, algo que comúnmente se conoce como Framework CSS. Este sistema ofrece una serie de archivos con código CSS que se pueden incluir en las páginas web para aplicar estilos de una manera rápida y solucionar

también muchos de los problemas de compatibilidad entre navegadores que pueden surgir durante la etapa de maquetación. (20)

Entre las ventajas que aporta el anteriormente mencionado framework de presentación, se pueden destacar: una rejilla que permite crear cualquier estructura de página web, una definición de tipografía predeterminada de manera precisa para todos los elementos de la página que pueden tener texto, estilos para formularios con los que se pueden mejorar las interfaces de usuario, estilos para impresión con las definiciones CSS más útiles para que las páginas se lean bien cuando se imprimen en papel, estilos específicos para Internet Explorer con los que se pueden resolver algunos de los problemas más comunes de este navegador.

1.6.2 Capa de Lógica de Negocio

Es necesario definir el concepto de Modelo-Vista-Controlador (MVC): este es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web (21).

Modelo: Es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos; por ejemplo, no permite comprar un número de unidades negativo, calcula si hoy es el cumpleaños del usuario o los totales, impuestos o importes en un carrito de compra.

Vista: Presenta el modelo en un formato adecuado para interactuar con este, usualmente la interfaz de usuario.

Controlador: Este responde a eventos, usualmente acciones del usuario, e invoca cambios en el modelo y probablemente en la vista.

Luego de un análisis de los frameworks utilizados para optimizar el desarrollo de aplicaciones web, desarrollados en el lenguaje del lado del servidor PHP, y que separan la lógica de negocio, la lógica de servidor y la presentación de la aplicación web, como son Zend Framework, CakePHP y Symfony, se encontraron características similares como son: implementan el patrón MVC, proporcionan una estructura al código fuente, forzando al desarrollador a crear un código más legible para un futuro mantenimiento,

encapsulan operaciones complejas en instrucciones sencillas reutilizando código, se pueden ejecutar tanto en plataformas *nix (Unix, Linux, etc.) como en plataformas Windows, presentan CRUD (Crear, Obtener, Actualizar y Borrar o Create, Retrieve, Update y Delete en inglés) integrado para la interacción con la base de datos, usan programación OO, son independientes de base de datos soportando varios gestores como MySQL, Oracle y Postgres, siendo importante para si en un futuro se cambia el gestor, también soportan integración con el mapeador de objetos relacional (ORM por sus siglas en inglés) Doctrine y con los frameworks de la capa de presentación jQuery y Blueprint (22)

Se decide utilizar el framework Symfony 1.4.11 debido a que la comunidad de usuarios y documentación es muy amplia, siendo importante para la retroalimentación, superación y reparación de errores. La comunidad provee diversos plugins que pueden ser reutilizables. Se adapta a las necesidades de la plataforma educativa ZERA, ya que soporta proyectos grandes que usen metodologías pesadas, teniendo ZERA esta característica. El equipo de trabajo cuenta con vasta experiencia en el uso de este framework, el cual es actualizado constantemente con mejoras y soluciones a errores encontrados.

1.6.3 Capa de Acceso a Datos

Se analizaron los dos ORM más conocidos y utilizados, Doctrine y Propel. Ambos presentan características similares ya que soportan cualquier operación usual en un CRUD, pueden generar las clases PHP del modelo, Propel basado en XML y Doctrine basado en YAML (Ain't Markup Language), siendo este último más legible para las personas. Ambos soportan la generación de sus respectivos markups desde una base de datos existente, son para PHP, brindan la posibilidad de escribir (opcionalmente) consultas a la base de datos en un lenguaje orientado a objetos, posibilitando a los desarrolladores una alternativa al SQL, manteniendo una máxima flexibilidad sin necesidad de duplicar código. Se sitúan en la parte superior de la capa de abstracción de la base de datos lo que permite abstracción al gestor utilizado, tales como MySQL, Oracle, Postgres, etc. Soportan validación de datos en los modelos y relaciones entre modelos. (23)

Se decide usar Doctrine debido a que es muy rápido y eficaz, tanto es así que en la versión de Propel 2.0 se usarán muchas características de Doctrine 2.0. Doctrine posee una amplia documentación, presenta soporte para migraciones, lo que es un requisito no funcional de la Plataforma ZERA. Posee una característica que facilita aún más las consultas a la base de datos y es la presencia de métodos mágicos,

los cuales permiten obtener información por cualquier campo de la tabla en cuestión. Además, el equipo de trabajo tiene experiencia en el uso de este ORM.

1.7 Sistema Gestor de Base de Datos

Analizando tres de los gestores de bases de datos más utilizados a nivel mundial (MySQL, PostgreSQL y Oracle), se encontraron características similares como son: funcionan en los principales sistemas operativos, se incluyen Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows, son totalmente compatibles con ACID (Atomicity, Consistency, Isolation and Durability o Atomicidad, Consistencia, Aislamiento y Durabilidad en español), tienen soporte completo para claves foráneas, uniones, vistas, disparadores y procedimientos almacenados. También son compatibles con el almacenamiento de objetos binarios, incluyendo imágenes, sonidos y videos, siendo estos requisitos no funcionales de software de la plataforma educativa ZERA. Tienen una documentación de carácter excepcional, son altamente escalables tanto en la enorme cantidad de datos que pueden manejar como en el número de usuarios concurrentes que pueden permitir, siendo necesario para un mejor funcionamiento de la plataforma. Además brindan la posibilidad de replicar los datos de una base de datos a otra.

Se decide utilizar PostgreSQL, debido a que es un sistema de base de datos objeto-relacional libre, de código abierto y gratis, estas características lo convierten en un gestor más importante que otros. 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. (24)

Presenta una amplia y activa comunidad de usuarios y documentación, siendo estos dos aspectos importantes a la hora de dar solución tanto a errores del propio gestor como a errores del trabajo con las bases de datos, así como para una mejor retroalimentación y superación en el uso de esta herramienta. Además, el equipo de trabajo posee experiencia en el uso de este gestor de bases de datos.

1.8 Servidor Web

Se estudiaron los servidores más usados según estadísticas desde agosto de 1995 hasta febrero del 2011, Internet Information Services (IIS) y Apache. (25)

Se decide utilizar este último ya que presenta licencia de software libre y de código abierto siendo la de IIS licencia de software propietario. Apache es multiplataforma a diferencia de IIS que funciona solo sobre la plataforma Window, lo que es una limitante. Además, Apache soporta PHP a diferencia de IIS. (26)

Apache tiene una alta comunidad de usuarios lo que posibilita que se le dé solución tanto a errores del propio servidor como a errores de configuración, es seguro ya que permite protección de ficheros (27).

1.9 Entorno de desarrollo

Al estudiarse dos de los entornos de desarrollo integrado (IDE por sus siglas en inglés) más utilizados para la elaboración de aplicaciones web, como son Eclipse y NetBeans, se verificó que ambos IDEs son similares en cuanto a las funcionalidades y ventajas que ofrecen, como son: el soporte para PHP, son libres, gratuitos, sin restricciones de uso y de código abierto. Se decidió escoger NetBeans en su versión 6.9 como IDE debido a ventajas con respecto a Eclipse como: rapidez, lo que implica facilidad en cuanto a completamiento de código en los lenguajes de programación utilizados. Este IDE posee una característica que lo hace único, y es el completamiento de código para los métodos mágicos que usa el ORM Doctrine, característica que ningún otro editor posee. (28)

1.10 Herramientas CASE para el modelado UML

La utilización de una herramienta CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) mejora la comunicación entre usuarios y especialistas, permite la facilidad de revisión de aplicaciones instaladas, son capaces de generar automáticamente las instrucciones del programa fuente y da la posibilidad de la generación de documentación técnica.

Entre las herramientas CASE orientadas a UML se encuentran ArgoUML y Visual Paradigm, que constituyen dos de las herramientas de desarrollo de ingeniería de software para Linux más usadas en la actualidad.

1.10.1 ArgoUML

ArgoUML es una herramienta desarrollada en Java que permite crear modelos UML compatibles con los estándares de la versión 1.4 de este lenguaje. Incluye una interfaz muy intuitiva, estable y de sencillo

manejo. Los tipos de diagramas que se pueden crear son: diagramas de clases, de estados, de actividad, de casos de uso, de colaboración, de despliegue y de secuencia. (29)

Otras características presentes en ArgoUML son:

- ✓ Compatibilidad con el estándar XMI (XML Metadata Interchange, XML de Intercambio de Metadatos): es posible utilizar los modelos creados con UML en otros programas.
- ✓ Generación de código para los lenguajes Java, C++, C, PHP4 y PHP5.
- ✓ Framework modular para ingeniería inversa.
- ✓ Críticas de diseño o sugerencias realizadas por ArgoUML para mejorarlo. (29)

1.10.2 Visual Paradigm

Es una herramienta diseñada para desarrollar software con soporte a la programación orientada a objetos. Busca reducir la duración del ciclo de desarrollo, brindando ayuda a arquitectos, analistas, diseñadores y desarrolladores; permite el uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación y tiene capacidades de ingeniería directa e inversa. Usa UML como lenguaje de modelado. Presenta una interfaz de uso intuitiva y con muchas facilidades a la hora de modelar los diagramas que soportan la Ingeniería de Requerimientos (30).

Una de las características más importantes de su uso es que brinda la posibilidad de sincronización del modelo de diseño y el código en todo el ciclo de desarrollo una vez que se integra con el NetBeans IDE, permitiendo la facilidad de programar directamente sobre el código fuente generado y a su vez actualizar el diseño con cambios que se realicen en la programación. Tiene una alta disponibilidad de múltiples versiones para cada necesidad, al igual que en las plataformas *Linux*, *MacOS* y *Windows*.

Presenta un innovador analizador textual donde se introduce texto extraído de conversaciones con el cliente, se definen actores, entidades, casos de uso disponibles para la generación de artefactos posteriores. También posee una herramienta de generación de reportes en formato PDF o HTML configurable y selectiva, e importa o exporta formatos estándares de otras herramientas CASE como el Rational Rose.

Selección de la herramienta CASE

Por todas las potencialidades anteriormente citadas tales como: el innovador analizador textual, la constante sincronización del modelo de diseño y el código fuente durante todo el ciclo de desarrollo del software, además de su excelente generación de código y soporte para UML 2.1, se ha escogido Visual Paradigm en su versión **6.4** como herramienta CASE de desarrollo.

1.11 Conclusiones

Teniendo en cuenta el estudio realizado previamente sobre las principales tendencias en el campo de la informática, se toma como decisión desarrollar una aplicación web utilizando el lenguaje PHP, con la integración de los frameworks jQuery, Blueprint y Symfony, buscando ganar en velocidad de elaboración e incorporar buenas prácticas de desarrollo con el uso de patrones. Para la construcción del sistema se determina hacer uso de las herramientas NetBeans y Visual Paradigm debido a las facilidades que estas brindan. Todo este proceso será controlado y orientado por la metodología RUP, la cual más que un dogma constituye una guía de cómo se debe desarrollar un software.

Capítulo 2 Características del sistema.

2.1 Introducción

En el desarrollo de software es de vital importancia definir los procesos que intervienen en este, para lograr así un mejor entendimiento del sistema a desarrollar por parte de los clientes y desarrolladores.

Se tiene como propósito exponer el contexto en que se desarrolla el problema que da origen a esta investigación. Se identifican los requisitos funcionales, no funcionales y los casos de uso críticos que guiarán el desarrollo del sistema propuesto. Además, se realiza el diagrama de casos de uso con el objetivo de mostrar las relaciones entre los actores del sistema y las secuencias de funcionalidades en las que están involucrados.

2.2 Modelo de Dominio

El Modelo de Dominio ayuda a comprender los conceptos que utilizan los usuarios y están relacionados con los procesos que se desean desarrollar, por esta razón se hace necesaria su elaboración.

A continuación se muestran los conceptos identificados.

2.2.1 Conceptos del Dominio

- ✓ **ZERA:** Plataforma educativa destinada a apoyar el proceso de enseñanza – aprendizaje.
- ✓ **Hiperentorno de aprendizaje:** Sistema informático basado en tecnología hipermedia, en el que se mezclan diversas tipologías de software educativo con el mejor propósito de atender las necesidades didácticas.
- ✓ **Usuario de Hiperentorno:** Es el usuario que accede al hiperentorno de aprendizaje, puede comportarse como un estudiante, docente, tutor, administrador local o director.
- ✓ **Estudiante:** Persona que cursa estudios, tiene el propósito de aprender una materia y accede al hiperentorno en busca de conocimientos.
- ✓ **Docente (Profesor):** Persona responsable de educar, guiar y supervisar a los estudiantes en el proceso de enseñanza - aprendizaje.

- ✓ **Tutor:** Persona responsable de guiar y supervisar las actividades de los estudiantes.
- ✓ **Administrador Local:** Realiza las tareas de administración de la plataforma instalada en la escuela.
- ✓ **Director:** Persona encargada de dirigir una escuela, ya sea porque es el dueño o porque se le asignó la tarea.
- ✓ **Módulo:** Se denomina Módulo, a uno de los elementos fundamentales que componen el producto.
- ✓ **Prácticas:** Módulo que tiene como principal objetivo permitir a los estudiantes la ejercitación y autoevaluación de los contenidos estudiados mediante la resolución de ejercicios.
- ✓ **Biblioteca:** Módulo que tiene como principal objetivo agrupar y visualizar los recursos estructurales, multimedia e interactivos.
- ✓ **Docente:** Módulo que tiene como principal objetivo agrupar las acciones específicas que se le asignan al docente dentro del hiperentorno.
- ✓ **Contenidos:** Módulo que tiene como principal objetivo permitir a los docentes, estudiantes y tutores la navegación por cada uno de los contenidos de las materias a través de un índice.
- ✓ **Capítulo:** La más general de las estructuras en las que se divide el contenido de un producto.
- ✓ **Tema:** Estructura que agrupa el contenido dentro de un capítulo.
- ✓ **Subtema:** Estructura que agrupa el contenido dentro de un tema.
- ✓ **Recursos Multimedia:** Son las diferentes medias como: video, sonido, animación e imagen.
- ✓ **Recursos Estructurales:** Son los diferentes recursos como: saber más, sabías que, medita un instante, recuerda que, aplicación práctica, investiga y aprende e historia.
- ✓ **Recursos Interactivos:** Diferentes recursos, como son: secuencias interactivas, pregunta

interactiva, animaciones interactivas, simuladores y enfrenta retos.

- ✓ **Glosario de términos:** Definiciones o conceptos relacionados con la Materia.
- ✓ **Páginas:** Son los diferentes elementos por los que está compuesto un contenido. Estas pueden tener asociados diferentes recursos.
- ✓ **Palabra o Frase clave:** Palabra o frase resaltada que forma parte de un contenido y puede tener asociados diferentes recursos.
- ✓ **Índice:** Estructura del contenido, se compone de los capítulos, temas y subtemas.
- ✓ **Programa de Estudio:** Es la adecuación del macro índice al plan de estudios del sistema educativo y de la institución.

2.2.2 Diagrama del Modelo de Dominio

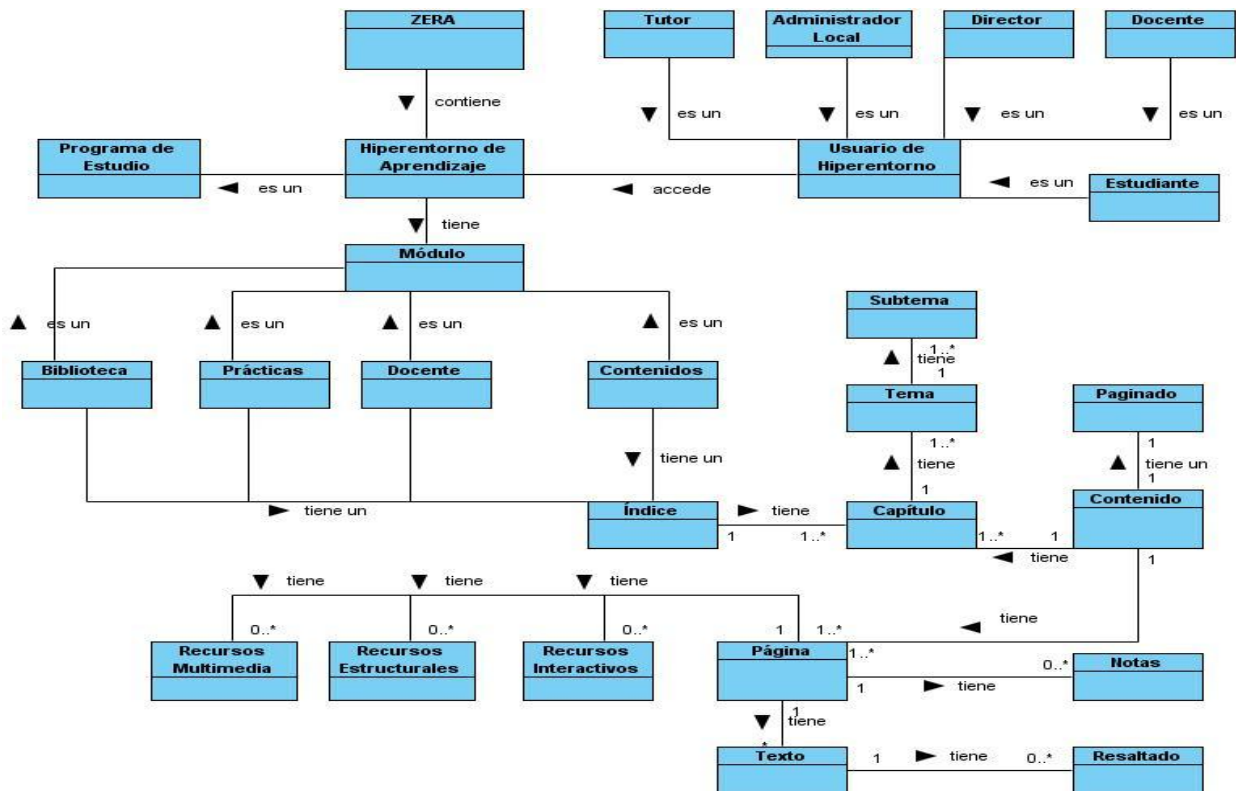


Figura 4. Representación del Modelo de Dominio.

2.3 Descripción del Sistema Propuesto

El sistema propuesto se corresponde con uno de los módulos de la plataforma educativa ZERA, el cual tiene como nombre iTopics. Este tiene como principal objetivo permitir a los docentes, estudiantes, tutores, director y administrador local la navegación por cada uno de los contenidos de las materias a través del índice.

El módulo a desarrollar proporcionará a los usuarios de la plataforma una cantidad considerable de funcionalidades que harán del contenido de esta, un ambiente agradable y cargado de facilidades, dentro de las cuales se encuentran:

- ✓ Registrar automáticamente el avance en el contenido, es decir, que el sistema memorice cualquier acción que constituya un cambio respecto a la vista actual del contenido.
- ✓ Llevar a cabo un sistema de apuntes que permitirá realizar comentarios asociados al contenido.
- ✓ Resaltar información que sea de importancia, lo cual traerá consigo un mejor entendimiento del contenido por parte del usuario debido a la posibilidad de personalizar la visualización de estos en la plataforma.

El módulo iTopics se integrará al resto de los módulos que conformarán la plataforma educativa ZERA, convirtiéndola en un producto que cumplirá con las necesidades del cliente.

2.4 Requisitos del software

La identificación de los requisitos del software permitirá definir las condiciones o capacidades que debe cumplir el sistema para que se ajuste a las necesidades del cliente.

Requisitos Funcionales

RF 1: Gestionar apuntes. Se permite realizar comentarios asociados al contenido. Tienen acceso a esta funcionalidad los usuarios con rol administrador local, director, estudiante y docente. Y el tutor solo tiene acceso a ver los apuntes de sus estudiantes.

- ✓ Para los docentes, se brinda la posibilidad de adjuntar a los apuntes recursos y compartirlos con sus estudiantes u otros profesores.

- ✓ Los administradores locales y directores podrán eliminar los apuntes creados por los docentes.
- ✓ Los estudiantes podrán crear apuntes de tipo texto y el sistema no les permitirá adjuntar recursos a los apuntes gestionados.

RF 1.1 Insertar los datos del apunte. Para ello se debe introducir la nota perteneciente al apunte. En caso de que el usuario sea un docente también podrá adjuntarle a este un recurso (ya sean recursos creados previamente en la Plataforma o gestionados por el propio usuario), además de la opción de compartir dichos apuntes con sus estudiantes y otros profesores.

RF 1.2 Ver los datos del apunte. Para ver los datos del apunte se accede a este, el sistema visualizará las opciones de modificar y eliminar dicho apunte en dependencia de los permisos del usuario autenticado.

RF 1.3 Modificar los datos de un apunte. Se podrá acceder a través del vínculo editar perteneciente al apunte que se desea modificar, luego se podrá cambiar la nota perteneciente a este y el recurso en dependencia de los permisos del usuario autenticado.

RF 1.4 Eliminar apunte. Se accede a través del vínculo eliminar perteneciente al apunte, permitiéndose confirmar la eliminación.

RF 2: Gestionar resaltado de textos.

RF 2.1 Resaltar textos en el contenido. Para ello se debe seleccionar la opción de resaltado en el panel de herramientas y seguidamente el texto que se desee marcar.

RF 2.2 Ver resaltado de textos. Se logra consultando una página del contenido que contenga textos resaltados.

RF 2.3 Eliminar resaltado de textos. Para desmarcar el texto es preciso seleccionar del panel de herramientas esta opción y seguidamente el texto a desmarcar.

RF 2.4 Al resaltar un texto se debe guardar automáticamente su ubicación en la página del contenido.

RF 2.5 Al acceder al contenido, el sistema permite al usuario consultar el texto que había resaltado anteriormente.

RF 3: Listar Índice.

RF 3.1 Mostrar el índice del hiperentorno y listar los capítulos, temas y subtemas asociados a este.

RF 3.2 Mostrar los subíndices asociados a un elemento seleccionado del índice.

RF 3.3 Mostrar las páginas asociadas a un elemento seleccionado. Para ello al seleccionar un elemento de los subíndices el sistema mostrará las páginas asociadas a este.

RF 4: Mostrar página.

RF 4.1 Mostrar la página con los recursos asociados.

RF 4.2 Mostrar las palabras o frases clave asociadas a la página en caso de existir.

RF 4.3: Mostrar paginado. Permite navegar por las páginas del contenido a través de las opciones: anterior, siguiente, inicio y fin, además permite insertar el número de la página que se desee consultar.

RF 6: Gestionar avance en el contenido.

RF 6.1 Registrar automáticamente el avance en el contenido, es decir, que el sistema memorice cualquier acción que constituya un cambio respecto a la vista actual del contenido.

RF 6.2 Mostrar avance en el contenido, al entrar al contenido se permite al usuario decidir si quiere o no entrar por el avance que llevaba en el árbol de contenidos (índice). En caso afirmativo se mostrará el último contenido visitado y en caso negativo se reiniciará el árbol de contenidos, mostrándose completamente cerrado.

RF 7: Consultar el chat.

RF 7.1 Activar servicio de chat. Al ser asignado un programa de estudios a una institución, a cada grupo académico de esta se le habilita el servicio de chat.

RF 7.2 Enviar mensajes al chat. Solo se permitirá enviar mensajes a los usuarios que estén conectados en el momento del envío.

RF 7.3 Acceder al chat. El chat estará activo en todos los módulos del hiperentorno. Tienen acceso al chat los siguientes roles: estudiante, docente, director y administrador local.

El comportamiento de cada rol en el chat, estará definido como sigue:

Estudiante

- ✓ Envía mensajes instantáneos solo a su docente y/o compañeros de grupo de la materia.
- ✓ No puede borrar comentarios.
- ✓ Siempre se genera traza de las acciones realizadas (sesiones (inicio y fin) y usuarios con los que interactúa).

Docente

- ✓ Envía mensajes instantáneos a los estudiantes asociados, otros docentes, director de la institución y administrador local.
- ✓ No puede borrar comentarios.
- ✓ Siempre se genera traza de los eventos (sesiones (inicio y fin) y usuarios con los que interactúa).
- ✓ Puede inhabilitar temporalmente el servicio de chat para sus estudiantes.

Director

- ✓ Envía mensajes instantáneos a docentes y al administrador local.
- ✓ Siempre se genera traza de los eventos (sesiones (inicio y fin) y usuarios con los que interactúa).
- ✓ Puede inhabilitar temporalmente el servicio de chat para los estudiantes y profesores de la escuela.

Administrador Local

- ✓ Envía mensajes instantáneos al director y a los docentes.
- ✓ Puede inhabilitar temporalmente el servicio de chat para los estudiantes y profesores de la escuela.

Requisitos No Funcionales

➤ Seguridad

RNF 1: Base de datos de usuarios y calificaciones: ofuscada.

RNF 2: Base de datos de elementos multimedia y contenidos: encriptado.

RNF 3: Comunicación web de transferencia de datos: encriptados.

RNF 4: Lograr en todo momento la seguridad de los contenidos:

- ✓ Negando la posibilidad de copiar los contenidos.
- ✓ Negando la posibilidad de descarga de los contenidos.
- ✓ Asegurando el encriptado de datos.

➤ **Soporte**

RNF 6: Ser generado en tecnología Web para ser accesible a través de Internet.

RNF 7: Correr sobre cualquier navegador, siendo como mínimo que sea compatible con:

- ✓ Explorer 7.0 y superior.
- ✓ Mozilla Firefox 3.6.
- ✓ Opera 10.00.
- ✓ Chrome 7.0.
- ✓ Safari 4.0.

➤ **Hardware**

Los servidores deberán cubrir las siguientes características o contar con una variante equivalente a:

RNF 8: Servidores centrales. Servidores web con procesador: Intel Xeon de 4 núcleos. La cantidad de núcleos depende de la cantidad de usuarios conectados. (Ejemplo: 80,000 usuarios – 7 núcleos). RAM: 8 GB o más, con posibilidades de expansión en caso de ser necesario. Servidor de Base de Datos con Procesador: Intel Xeon de 4 núcleos. La cantidad de núcleos depende de la cantidad de usuarios conectados. (Ejemplo: 80,000 usuarios – 7 núcleos). RAM: 8 GB o más, con posibilidades de expansión en caso de ser necesario. Servidor de Almacenamiento con 300 GB, inicialmente para el almacenamiento de todas las medias.

RNF 9: Servidores locales. En las escuelas en dependencia de la cantidad de usuarios pudiese elegirse usar un solo servidor para la web, Base de Datos y Almacenamiento, o de forma separada en dos: uno para la web y otro para la BD.

Pueden ser servidores normales, o máquinas mejoradas en caso de escuelas pequeñas de menos de 200 usuarios:

- ✓ 1 Procesador CoreDuo.
- ✓ 2GB de RAM.
- ✓ 120 GB de HDD mínimo, aunque puede ser configurable de acuerdo con la cantidad de contenidos que se vayan contratando.
- ✓ Banda ancha.

Equipos de usuario final

Los usuarios finales deberán contar como mínimo con:

- ✓ Procesador Pentium II o superior.
- ✓ 1GB de RAM.
- ✓ 20GB de HDD.
- ✓ Bocinas.
- ✓ Si no cuentan con servidor local: conexión de banda ancha 256 Kbps como mínimo.
- ✓ Si cuenta con servidor local: acceso a red interna.

➤ **Restricciones de diseño e implementación**

RNF 10: Lenguaje de programación: PHP 5.3.3

RNF 11: El marco de trabajo base de desarrollo que se utilizará es: Symfony 1.4.6

RNF 12: Como IDE se empleará NetBeans 6.9

RNF 13: Como servidor Web se explotará Apache 2.2.16

RNF 14: El SGDB deberá ser PostgreSQL 8.4.

➤ **Usabilidad**

RNF 15: Los elementos gráficos como los iconos deberán contar con un tooltip o mensaje flotante que señalen el tipo de recurso al que se refiere.

➤ **Eficiencia**

RNF 16: Cuando un usuario, sin importar su rol, permanece inactivo durante 20 minutos, se cierra la sesión automáticamente.

2.5 Modelo de Casos de Uso del Sistema

En esta sección se identifican los actores del sistema, se realiza el diagrama de casos de uso y una breve descripción de los mismos.

2.5.1 Descripción de los actores del sistema

Los actores identificados son personas que van a inicializar e interactuar con los casos de uso del sistema.

Tabla 1. Descripción de los actores del sistema.

Actor	Descripción
Estudiante	Actor que al acceder a una página del contenido, además de consultar la información, podrá reflejar el avance en la lectura de este, es decir, puede marcar el índice donde se quedó navegando, podrá realizar comentarios asociados a este para que sean consultados posteriormente y resaltar la información que sea de su interés.
Docente	Actor que podrá realizar apuntes y adjuntar a estos recursos creados previamente en la Plataforma o gestionados por este, además podrá compartirlos con sus estudiantes u otros docentes.
Tutor	Actor que podrá consultar todos los contenidos y recursos educativos, pudiendo resaltar textos y realizar comentarios asociados al contenido.

Administrador Local	Actor que podrá consultar todos los contenidos y recursos educativos, pudiendo además resaltar textos y realizar comentarios asociados al contenido.
Director	Actor que podrá consultar todos los contenidos y recursos educativos, pudiendo además resaltar textos y realizar comentarios asociados al contenido.
Usuario de Hiperentorno	Actor que representa la generalización de los actores: Estudiante, Docente, Tutor, Administrador Local y Director.
Usuario de mensajería instantánea	Actor que interactúa con las funcionalidades del chat y representa la generalización de los actores: Estudiante, Docente, Administrador Local y Director.

2.5.2 Patrones de casos de uso

La experiencia en la utilización de casos de uso ha evolucionado en un conjunto de patrones que permiten con más precisión reflejar los requisitos reales, haciendo más fácil el trabajo con los sistemas, y mucho más simple su mantenimiento. Dado un contexto y un problema a resolver, estas técnicas han mostrado ser la solución adoptada en las comunidades del desarrollo de software.

A continuación se enuncian los patrones presentes en el diagrama de casos de uso del sistema:

CRUD (Creating, Reading, Updating, Deleting): Este patrón se basa en la fusión de casos de uso simples para formar una unidad conceptual. Se puede apreciar el uso de este en los CU Gestionar apuntes y Gestionar resaltado de textos, siendo CRUD completo y parcial respectivamente.

Múltiples actores: El uso de este patrón se evidencia específicamente en la variante **roles comunes**, la cual establece que varios actores jueguen el mismo rol sobre el CU. Este rol es representado por otro actor, heredado por los actores que comparten este rol. Siendo reflejado a continuación en la **Figura 5**, la cual muestra una parte del diagrama de casos de uso del sistema.

2.5.3 Diagrama de actores del sistema

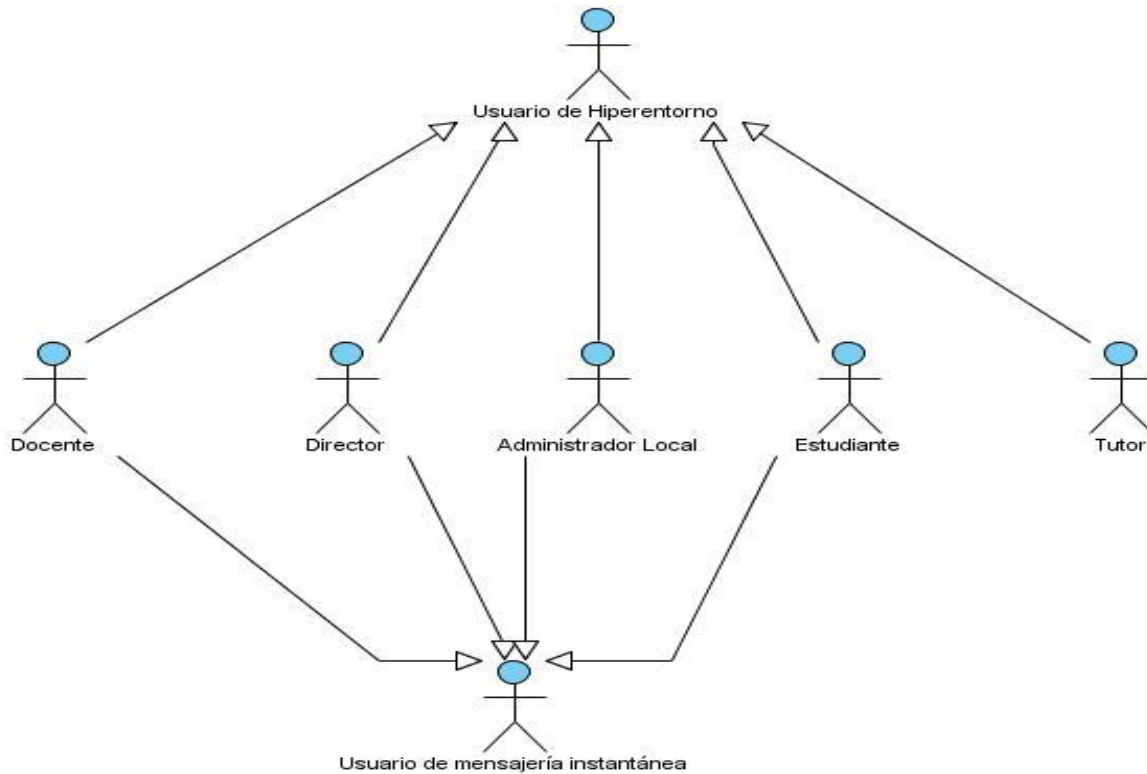


Figura 5. Diagrama de Actores del Sistema.

2.5.4 Descripción de los casos de uso del sistema

Tabla 2. Descripción del CU Gestionar Apuntes.

Caso de uso	CU Gestionar Apuntes
Actores	Usuario de Hiperentorno (Inicia), este actor generaliza a los actores: Docente, Director, Administrador Local, Estudiante y Tutor.
Resumen	El caso de uso se inicia cuando el Usuario de Hiperentorno selecciona la opción que le permite realizar una acción sobre el apunte. El actor puede incluir, ver, eliminar y modificar apuntes. En caso de que seleccione la opción de incluir un apunte, el sistema dará la posibilidad de insertar los datos que se necesitan en dependencia del usuario autenticado. Si el actor elige la opción de ver un apunte el sistema mostrará el contenido en cuestión. Si el actor elige la opción de modificar un apunte,

	<p>el sistema mostrará los datos que pueden ser editados dentro de este en dependencia del usuario autenticado, y una vez realizados los cambios, guardará las modificaciones. El sistema permite eliminar el apunte y muestra un mensaje de confirmación. En caso de que el usuario sea un docente también podrá adjuntarle a este un recurso (ya sean recursos creados previamente en la Plataforma o gestionados por el propio usuario), además de la opción de compartir dichos apuntes con sus estudiantes, terminando así el caso de uso.</p>
Precondiciones	<p>Debe haberse generado el escritorio de trabajo del usuario autenticado.</p> <p>Para incluir un apunte, este debe estar asociado a una parte del contenido, se selecciona previamente la opción de apunte perteneciente al panel de herramientas y seguidamente el texto al que pertenecerá el apunte.</p> <p>Se mostrarán todos los apuntes pertenecientes a la página del contenido, ya sea porque el usuario autenticado es el autor de dichos apuntes o porque se le ha asignado temporalmente este permiso.</p> <p>Para modificar un apunte, debe haber seleccionado previamente la opción editar y el actor debe ser el responsable temporal de este.</p> <p>Para eliminar un apunte, debe haber seleccionado previamente la opción eliminar y el actor debe tener el permiso de eliminar el elemento.</p>
Referencia	RF 1

Tabla 3. Descripción del CU Gestionar Resaltado de Textos.

Caso de uso	CU Gestionar Resaltado de Textos
Actores	Usuario de Hiperentorno (Inicia), este actor generaliza a los actores: Docente, Director, Administrador Local, Estudiante y Tutor.
Resumen	El caso de uso se inicia cuando el Usuario de Hiperentorno selecciona la opción que le permite realizar una acción sobre el texto de la página. El actor puede incluir, ver y eliminar el resaltado de textos en la página. En caso de que seleccione la opción de

	resaltar el sistema dará la posibilidad de resaltar el texto deseado. Cuando el actor accede a una página del contenido, el sistema mostrará los textos resaltados en la página en cuestión. Si el actor elige la opción de desmarcar texto en el panel de herramientas, el sistema seguidamente permitirá desmarcar el texto resaltado. Terminando así el caso de uso.
Precondiciones	<p>Debe haberse generado el escritorio de trabajo del usuario autenticado.</p> <p>Para incluir un resaltado de texto, debe haberse seleccionado previamente el botón resaltar del panel de herramientas y luego señalar el texto a resaltar.</p> <p>Para ver el resaltado de texto el actor debe tener permiso de acceder al contenido, ya sea porque es el autor, un rol superior en jerarquía, o porque se le ha asignado temporalmente. Además, debe haber accedido a una página que contenga al menos un texto resaltado.</p> <p>Para eliminar el resaltado de texto, debe seleccionar la opción desmarcar del panel de herramientas previamente y luego seleccionar el texto a desmarcar.</p>
Referencia	RF 2

Tabla 4. Descripción del CU Listar Índice.

Caso de uso	CU Listar Índice
Actores	Usuario de Hiperentorno (Inicia), este actor generaliza a los actores: Docente, Director, Administrador Local, Estudiante y Tutor.
Resumen	El caso de uso se inicia cuando el Usuario de Hiperentorno accede al contenido dentro del hiperentorno. El sistema muestra el índice del hiperentorno y lista los capítulos, temas y subtemas asociados a este. El actor consulta los datos y selecciona un elemento de la lista mostrada. El sistema muestra los subíndices del elemento seleccionado y las páginas asociadas a él, en caso de no tener subíndices sólo muestra la página asociada al elemento seleccionado. El sistema permite salir de la vista actual y el caso de uso termina.

Precondiciones	<p>Haber generado el escritorio de trabajo del usuario autenticado.</p> <p>El actor debe haber accedido al hiperentorno.</p> <p>Para mostrar el listado de capítulos, debe haberse accedido al contenido dentro del hiperentorno.</p> <p>Para mostrar el listado de temas, debe haberse seleccionado el capítulo asociado.</p> <p>Para mostrar la lista de subtemas, debe haberse seleccionado el capítulo y el tema asociado.</p>
Referencia	RF 3

Tabla 5. Descripción del CU Mostrar Página.

Caso de uso	CU Mostrar Página
Actores	Usuario de Hiperentorno (Inicia), este actor generaliza a los actores: Docente, Director, Administrador Local, Estudiante y Tutor.
Resumen	El caso de uso se inicia cuando el Usuario de Hiperentorno accede a una página del contenido. El sistema permite mostrar todos los elementos pertenecientes a dicha página, donde el usuario puede consultarlos y navegar por el resto de las páginas asociadas al contenido a través del paginado. El caso de uso termina.
Precondiciones	<p>Debe haberse generado el escritorio de trabajo del usuario autenticado.</p> <p>El actor debe haber accedido al hiperentorno.</p> <p>Debe haberse seleccionado un nivel del índice.</p>
Referencia	RF 4

Tabla 6. Descripción del CU Gestionar Avance en el Contenido.

Caso de uso	CU Gestionar Avance en el Contenido
Actores	Usuario de Hiperentorno (Inicia), este actor generaliza a los actores: Docente,

	Director, Administrador Local, Estudiante y Tutor.
Resumen	El caso de uso se inicia cuando el Usuario de Hiperentorno realiza una acción sobre el contenido, como son: navegar en el módulo contenido o acceder a este. El sistema permite mostrar el avance en el contenido, además de registrar automáticamente el avance. En caso de que el actor realice una acción que constituye un cambio respecto a la configuración actual (acción sobre el sistema que provoque un cambio en la vista) en el módulo Contenido, el sistema registra automáticamente el avance en la información, memorizando el último contenido visitado por el usuario. Cuando el actor accede al contenido, el sistema muestra un mensaje de confirmación, de aceptarse dicha confirmación, el sistema muestra el último contenido guardado automáticamente, si no, entra al contenido con la navegación reiniciada, es decir, el árbol de contenidos recogido. Terminando así el caso de uso.
Precondiciones	Debe haberse generado el escritorio de trabajo del usuario autenticado. El actor debe haber accedido al hiperentorno. El actor debe realizar una acción sobre el sistema que provoque un cambio en la vista.
Referencia	RF 6

Tabla 7. Descripción del CU Mostrar Paginado.

Caso de uso	CU Mostrar Paginado
Actores	Usuario de Hiperentorno (Inicia), este actor generaliza a los actores: Docente, Director, Administrador Local, Estudiante y Tutor.
Resumen	El caso de uso se inicia cuando el Usuario de Hiperentorno accede a una página del contenido y selecciona una de las opciones del paginado para navegar por estas páginas. El sistema muestra la página del contenido con el paginado asociado y permite ir a la página siguiente, a la anterior, al inicio de estas páginas y a la final o

	insertar un número específico para ir a una página determinada. El actor puede realizar una de estas acciones y el caso de uso termina.
Precondiciones	Debe haberse generado el escritorio de trabajo del usuario autenticado. El actor debe haber accedido al hiperentorno. El actor debe haber accedido a una página del contenido para consultar el paginado.
Referencia	RF 5

Tabla 8. Descripción del CU Consultar el chat.

Caso de uso	CU Consultar el chat
Actores	Usuario de mensajería instantánea (Inicia), este actor generaliza a los actores: Docente, Director, Administrador Local y Estudiante.
Resumen	El caso de uso se inicia cuando el usuario de mensajería instantánea accede al hiperentorno para realizar una acción sobre la sección del chat. El sistema brinda la posibilidad de mostrar los usuarios en línea y consultar las opciones a realizar sobre el chat. Muestra además una vista para incluir los comentarios, dando la posibilidad de que sean enviados al usuario destino y permitiendo ser respondidos instantáneamente. El caso de uso termina.
Precondiciones	Debe haberse generado el escritorio de trabajo del usuario autenticado. Debe haberse asignado un programa de estudios a una institución para que se habilite el servicio chat a cada grupo académico de esta.
Referencia	RF 7

2.5.5 Diagrama de casos de uso del sistema

EL diagrama muestra la relación entre los CU y los usuarios que inicializan estos. El usuario de hiperentorno tiene la responsabilidad de gestionar el resaltado y los apuntes que se realizan a un texto determinado. De igual manera muestra las páginas, los paginados, gestiona el avance que ha tenido en el

contenido y lista el índice de temas asociados a la materia que estudia. Por otro lado, el usuario de mensajería instantánea es el encargado de consultar el chat, el cual le permite la comunicación entre los usuarios.

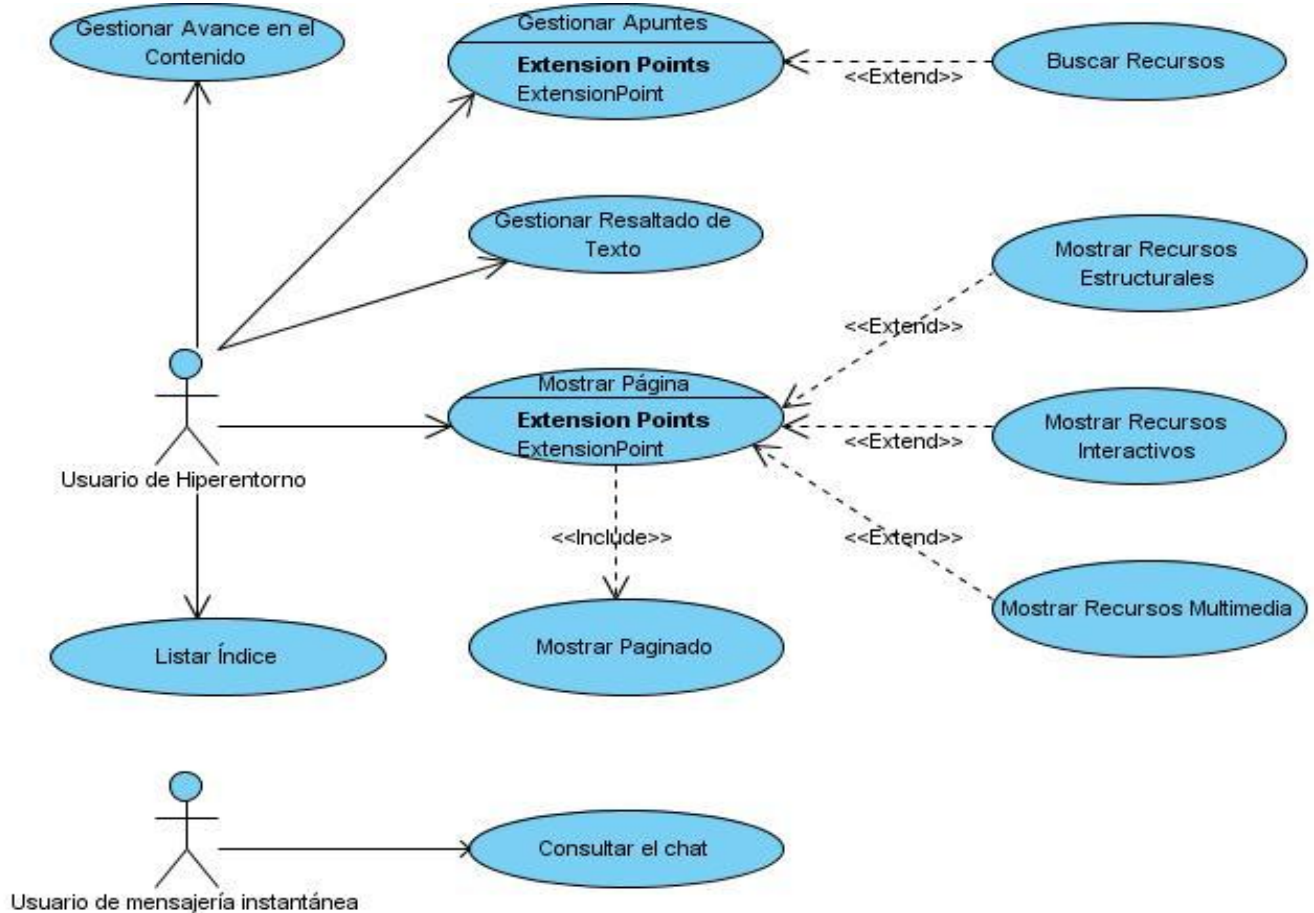


Figura 6. Diagrama de Casos de Uso del Sistema.

2.6 Conclusiones

El estudio realizado a los principales conceptos del negocio permite comenzar a desarrollar la propuesta de solución del sistema a partir de determinar los artefactos necesarios para iniciar el análisis y diseño de la propuesta de solución, donde se enmarcan todos los conceptos fundamentales del dominio. Se dará seguimiento a la investigación teniendo en cuenta el cumplimiento de cada requisito, para transformar estos en un diseño del sistema adecuado.

Capítulo 3. Análisis y diseño de la propuesta de sistema.

3.1 Introducción

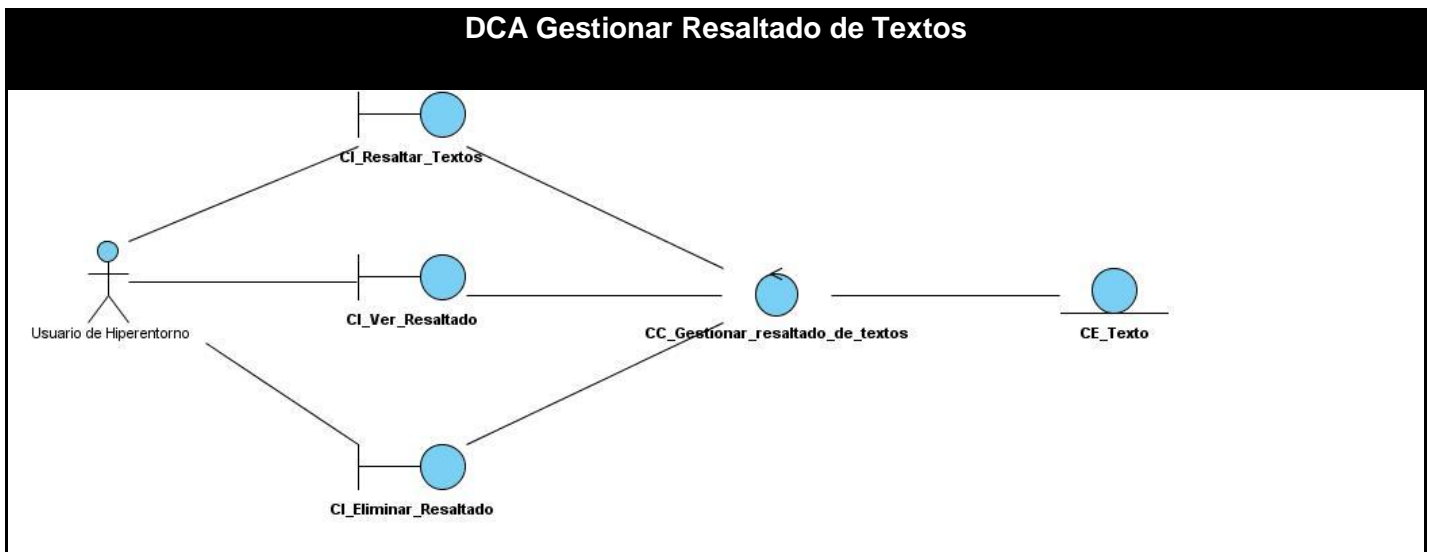
Análisis y diseño es el tercer flujo de trabajo que propone la metodología de desarrollo RUP, durante el cual se modelan los diagramas de clases del análisis y del diseño para cada caso de uso de la propuesta del sistema a desarrollar. Además, se define la arquitectura y los patrones de diseño que regirán el proceso. Tomando como base las clases persistentes para garantizar el almacenamiento de la información, se modela la estructura de la base de datos.

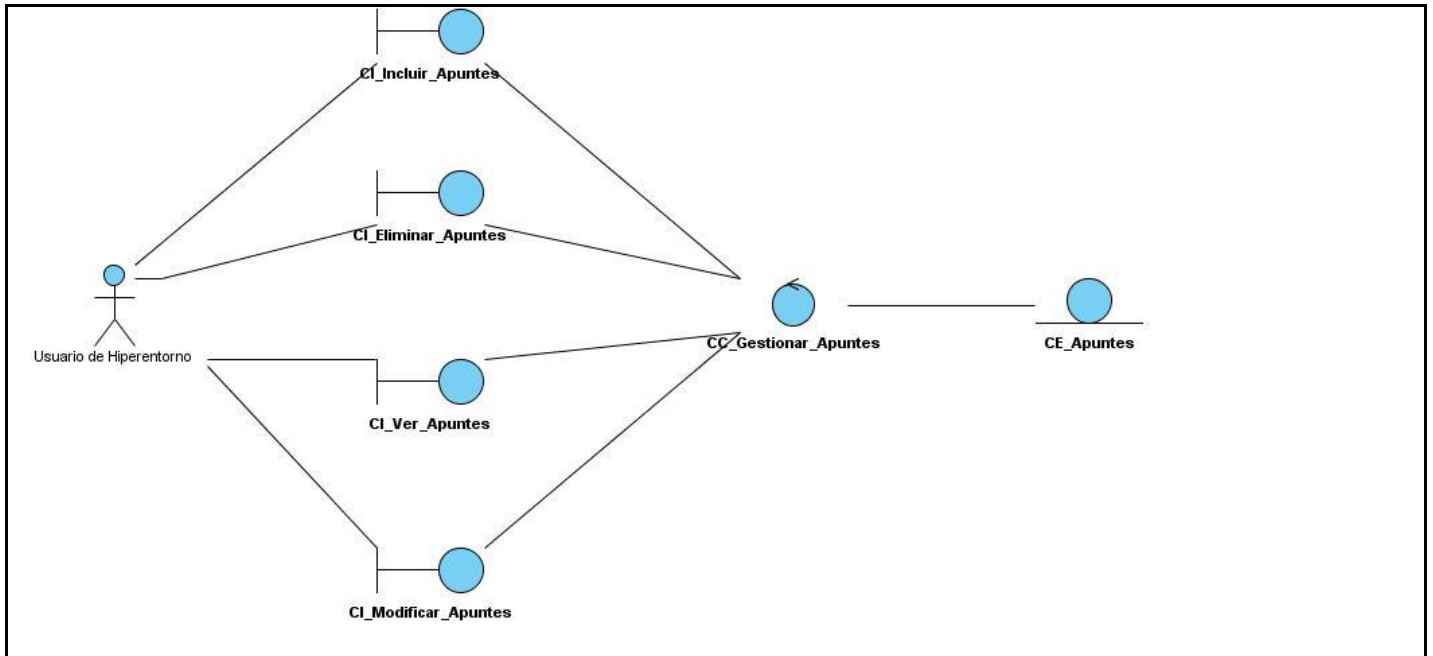
3.2 Modelo de análisis

El objetivo del análisis es traducir los requisitos a una especificación que describe cómo implementar el sistema. El análisis consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que sólo se interesa por los requisitos funcionales.

A continuación se muestran los diagramas de clase del análisis (DCA) para cada uno de los casos de uso del sistema.

3.2.1 Diagramas de clases del análisis

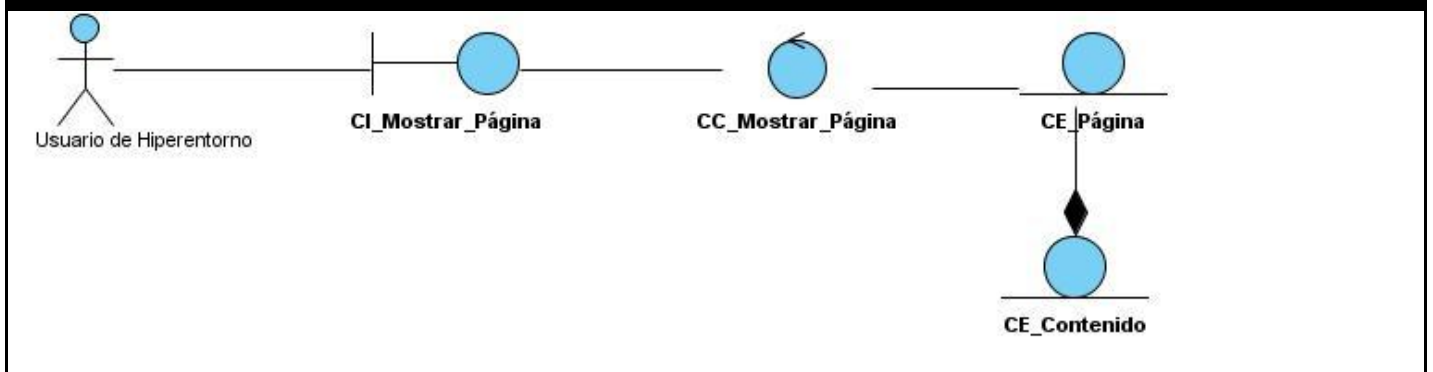




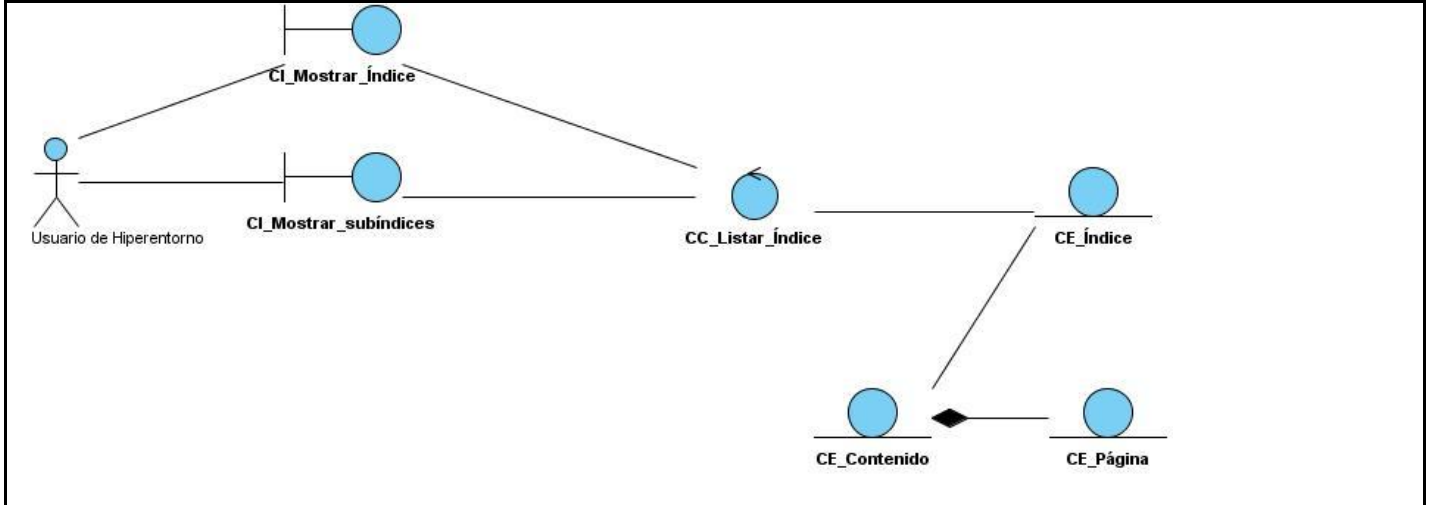
DCA Gestionar Avance en el Contenido



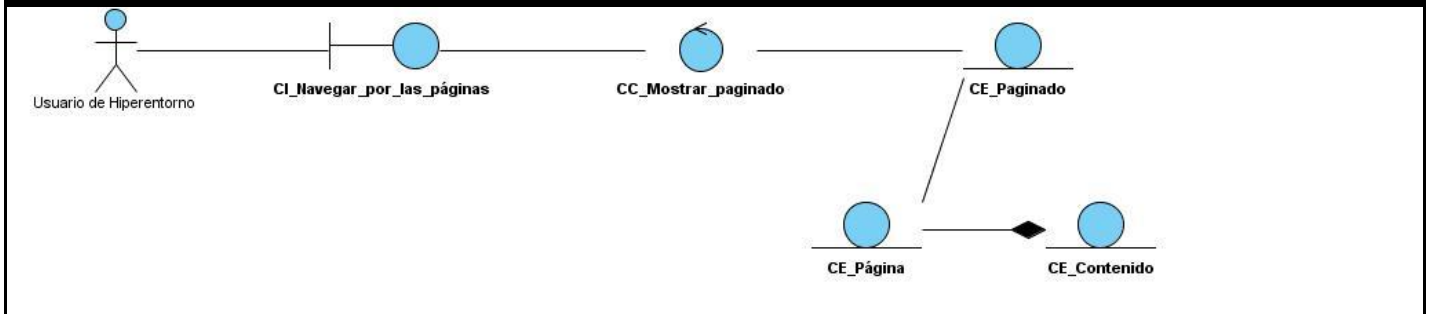
DCA Mostrar Página



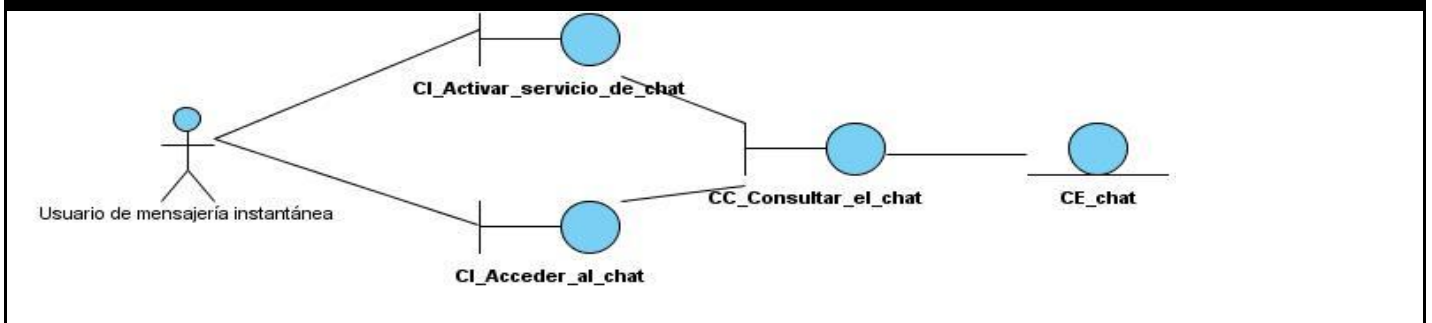
DCA Listar Índice



DCA Mostrar Paginado



DCA Consultar el chat



3.3 Patrón arquitectónico Modelo – Vista – Controlador en symfony

La arquitectura MVC separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes de, por ejemplo, el tipo de gestor de bases de datos utilizado por la aplicación. (22)

El framework Symfony está basado en este patrón arquitectónico, el cual permite que los cambios en la vista no afecten el resto de la aplicación. Proporcionando como ventajas la reutilización de los componentes y simplicidad en el mantenimiento del sistema.

En Symfony la estructura de la base de datos es mapeada a clases del sistema mediante un ORM (Object-Relational Mapping). Utilizando el patrón de diseño Active Record el desarrollador puede interactuar con el contenido de la base de datos a través de las instancias de estas clases. En el módulo iTopics se usan algunas clases del modelo generadas por Doctrine, ejemplos de estas son: TbSco y TbSelection, las cuales permiten acceso a los datos de las tablas correspondientes a estas clases.

La vista está representada por ficheros escritos en PHP que se encargan de construir la página HTML con la que interactúa el usuario. En el módulo iTopics, estos archivos están ubicados en el directorio templates/, los cuales son plantillas (nombreSuccess.php) y parciales (_nombre.php). Estos son actualizados por las acciones de dicho módulo.

En symfony, el nivel controlador está constituido por varios componentes, como son:

- ✓ El controlador frontal, que es el único punto de entrada a la aplicación el cual carga la configuración y determina la acción a ejecutarse.
- ✓ Las acciones, que contienen la lógica de la aplicación y verifican la integridad de las peticiones y preparan los datos requeridos para actualizar la capa de presentación. En el módulo iTopis, estas se encuentran en la clase iTopicsActions.

- ✓ Los objetos request, response y sesión, que dan acceso a los parámetros de la petición, las cabeceras de las respuestas y a los datos persistentes del usuario.
- ✓ Los filtros, los cuales son trozos de código ejecutados para cada petición, antes o después de una acción, como es el caso del filtro de seguridad, ejecutado antes de cada acción.

3.4 Aplicación de los patrones de diseño en Symfony

El framework Symfony utiliza en su implementación una serie de patrones de diseño, los cuales clasifican y describen formas de solucionar problemas específicos y comunes del diseño orientado a objetos. A continuación se explicarán algunos de los patrones utilizados directamente en la solución.

Observer (Observador): Este patrón permite definir una dependencia entre objetos de forma que cuando un objeto cambia de estado, todos sus objetos dependientes son notificados y actualizados. Este patrón se utiliza cuando se envían mensajes a través del chat, modificando el estado de las conversaciones. También es usado cuando se conectan usuarios nuevos, actualizando la lista de usuarios conectados.

Singleton: Este patrón garantiza la existencia única de una instancia de una clase y un mecanismo global de acceso a esta instancia u objeto. En la aplicación, el objeto sfContext, guarda una referencia a todos los objetos del núcleo de Symfony, mecanismo mediante el cual se interactúa con los objetos únicos.

Strategy (Estrategia): El objetivo de este patrón es definir un grupo de clases que representan un conjunto de posibles comportamientos. Estos comportamientos pueden ser fácilmente intercambiados en una aplicación, modificando la funcionalidad en cualquier instante. Este patrón se emplea en la gestión de notas, adicionando un comportamiento específico dependiendo del tipo de usuario que haga uso de las funcionalidades incluir, modificar, eliminar y ver notas.

Decorator (Decorador): El objetivo de este patrón es extender la funcionalidad de un objeto dinámicamente. Se utiliza a la hora de cargar el contenido dinámico de las plantillas del módulo iTopics, dentro de otra plantilla más general, la que extiende su funcionamiento a la primera.

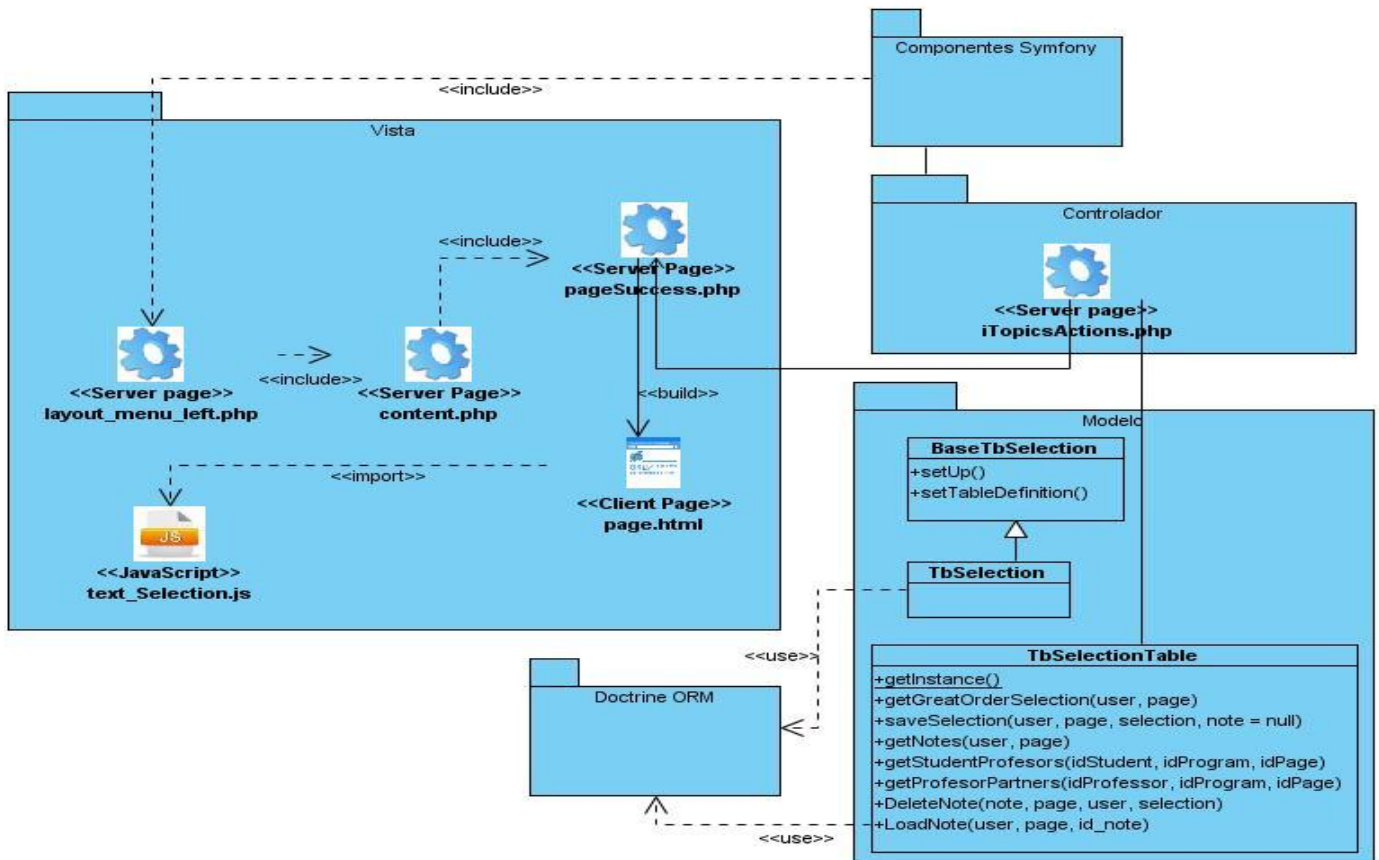
Creador: Este patrón se usa para establecer la responsabilidad de quién es el encargado de crear objetos de una clase determinada. Es usado en la mayoría de las acciones del módulo iTopics, creando objetos entidades, de formularios y de clases de abstracción a la base de datos.

3.5 Modelo de diseño

3.5.1 Diagrama de clases del diseño

Los diagramas de clases del diseño son una representación más concreta y detallada que los diagramas de clases del análisis, aunque también representan la parte estática del sistema conteniendo las clases y sus relaciones. Son empleados para representar las relaciones que se establecen entre las clases.

A continuación se presenta el diagrama de clases del diseño correspondiente al caso de uso Gestionar Resaltado de Textos. Para el estudio de los demás diagramas de clase del diseño remitirse al **Anexo 3: Diagramas de clases del diseño**.



Los diagramas de interacción del diseño realizados pueden encontrarse en el **Anexo 4**: Diagramas de interacción del diseño.

3.6 Diagrama de despliegue

El modelo de despliegue define la arquitectura física del sistema. Se usa para modelar de manera detallada los nodos físicos y las asociaciones de comunicación que existen entre ellos. Del mismo modo, queda especificado qué hardware, sistemas operativos, software de interfaces y soporte conformarán el nuevo sistema.

A continuación se presenta el diagrama de despliegue propuesto para el sistema:

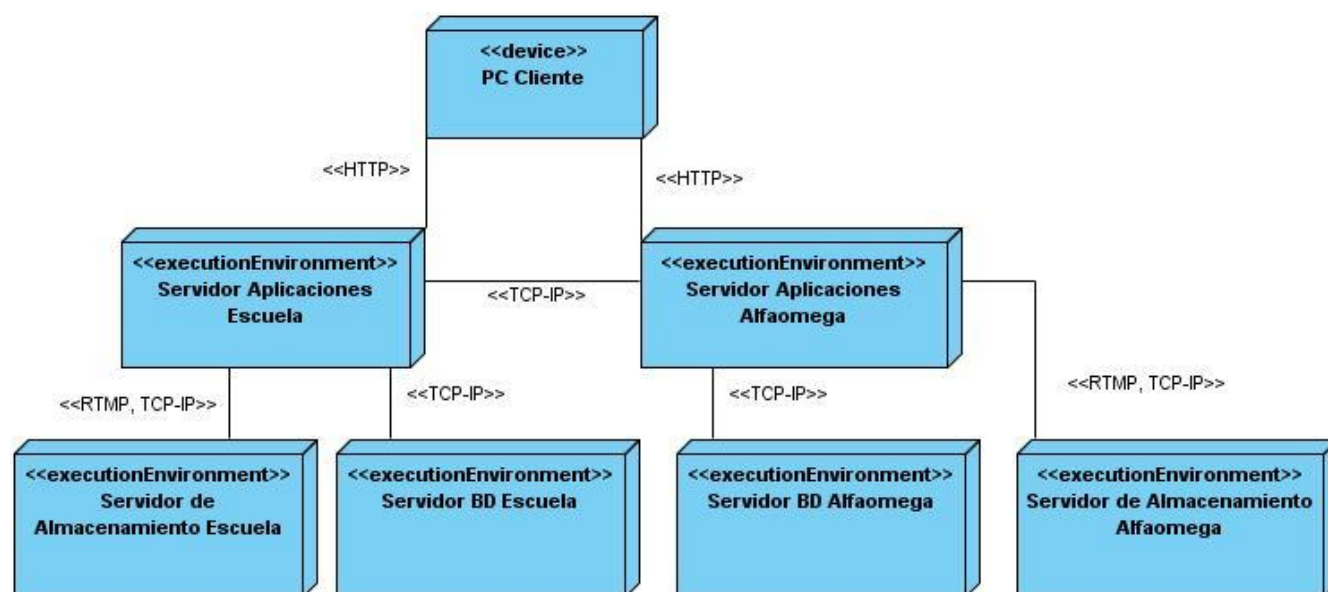


Figura 7. Modelo de despliegue.

3.7 Diseño de la base de datos

En esta sección se presenta el modelo de datos que estará compuesto por las entidades que pasarán a ser las tablas de la base de datos, para ser utilizadas por las funcionalidades a desarrollar. Éste se encuentra simplificado para una mejor visualización, además se describen las tablas más importantes que la conforman. Ver **Anexo # 7**: Descripciones de las Tablas de la Base de Datos.

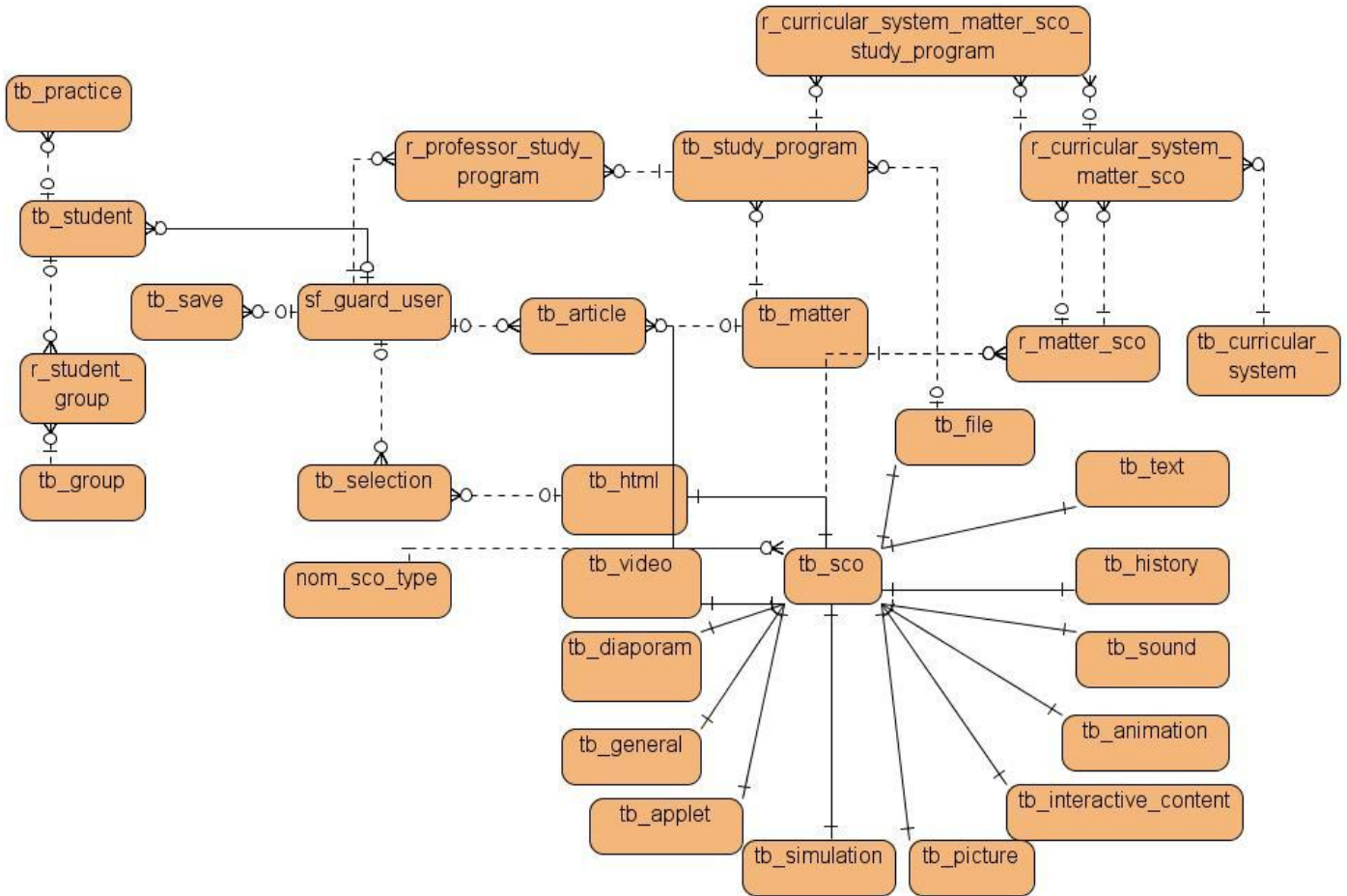


Figura 8. Modelo de datos.

3.8 Conclusiones

Durante el flujo de trabajo análisis y diseño llevado a cabo en el módulo iTopics de la plataforma educativa ZERA, se tuvo en cuenta las descripciones de los casos de uso como principal artefacto de entrada. Se obtuvo como resultado el diagrama entidad relación que define el diseño de la base de datos, los diagramas de clases del análisis y los diagramas de clases del diseño. Para estos últimos se tuvo en cuenta principalmente las facilidades que los frameworks aportaban y los patrones de diseño que sin forzar al programador ayudan a la solución propuesta. Una vez descrito cómo el sistema será realizado a partir de las funcionalidades previstas y las restricciones impuestas (requerimientos), se indica con precisión lo que se debe programar.

Capítulo 4. Implementación y Prueba.

4.1 Introducción

Los artefactos generados durante el análisis y el diseño son la entrada fundamental al flujo de trabajo de implementación, el cual tiene como propósito: definir la organización del código, implementar clases y objetos en forma de componentes, probar los componentes desarrollados e integrarlos a un sistema ejecutable. Para poner a prueba todos estos componentes desarrollados, es necesario entrar principalmente en la evaluación de la calidad del producto, para lo cual se hace necesario que el producto de software funcione como está previsto y que la validación de los requisitos se aplique correctamente.

4.2 Modelo de implementación

En la implementación se parte del resultado del diseño y se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. El propósito principal de la implementación es desarrollar la arquitectura y el sistema como un todo.

El modelo de implementación es la entrada principal de las etapas de prueba que siguen a la implementación. Más concretamente, durante la etapa de prueba cada construcción generada durante la implementación es sometida a pruebas de integración, y posiblemente también a pruebas de sistema. (31)

4.2.1 Diagrama de componentes

Los diagramas de componentes son usados para estructurar el modelo de implementación en término de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación.

El uso más importante de estos diagramas es mostrar la estructura de alto nivel del modelo de implementación, especificando que:

- ✓ Muestra las organizaciones y las dependencias entre tipos de componentes.
- ✓ Organiza los subsistemas de implementación en capas.

Se utilizan para modelar la vista estática de un sistema. Muestran la organización y las dependencias lógicas entre un conjunto de componentes de software, sean estos componentes de código fuente, librerías, binarios o ejecutables. (31)

A continuación se presenta el diagrama de componentes propuesto para el CU Gestionar Apuntes:

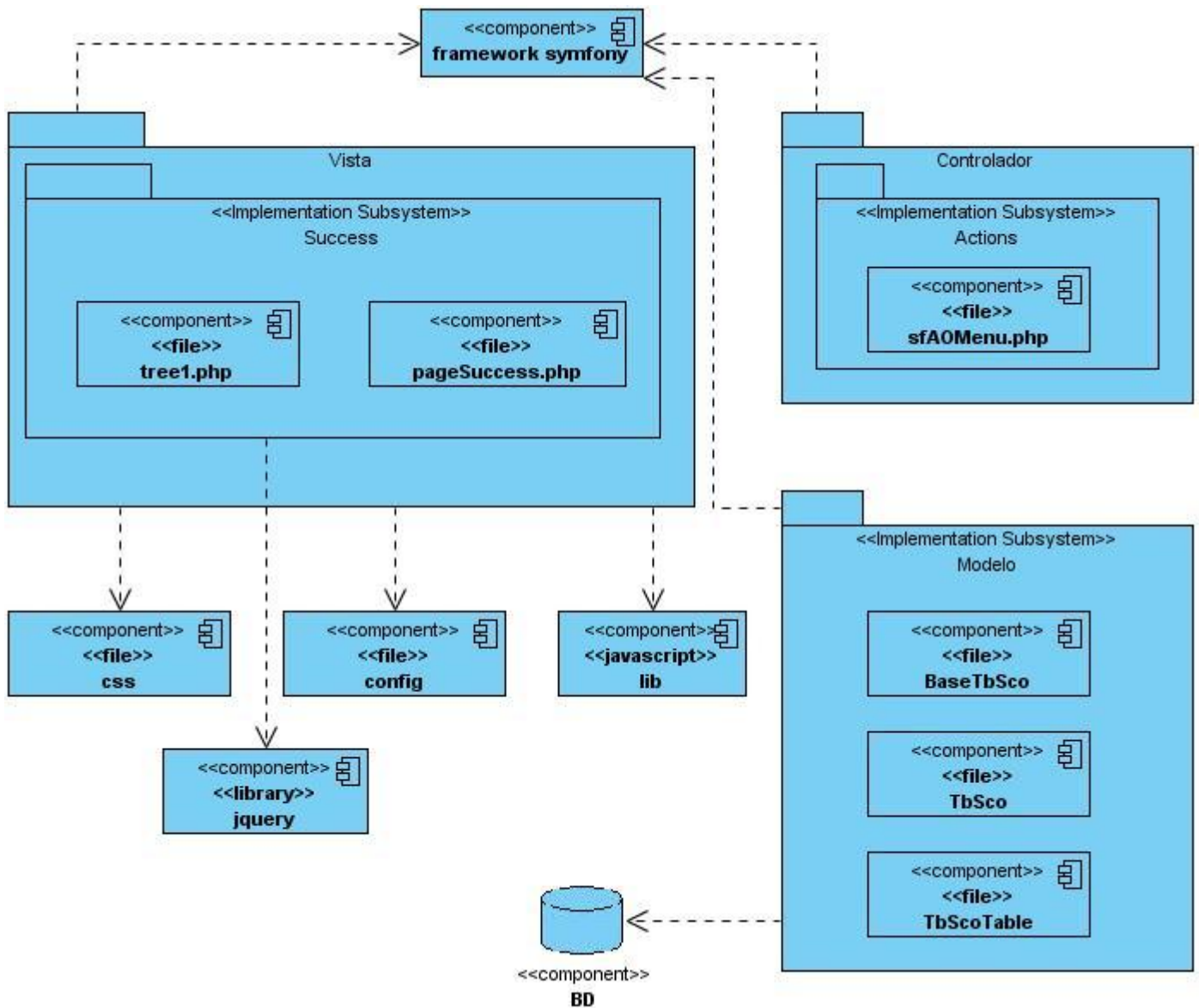


Figura 9. Diagrama de componentes CU Gestionar Apuntes.

El resto de los diagramas de componentes realizados pueden encontrarse en el **Anexo # 8: Diagramas de componentes por CU.**

4.3 Pruebas de software

La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación. (32)

El objetivo de las pruebas al sistema es comprobar la integración del sistema de información globalmente, verificando el funcionamiento correcto de las interfaces entre los distintos subsistemas que lo componen y con el resto de los sistemas de información con los que se comunica. En la realización de estas pruebas es importante comprobar la cobertura de los requisitos, dado que su incumplimiento puede comprometer la aceptación del sistema por el equipo de operación responsable de realizar las pruebas de implantación del sistema, que se llevarán a cabo en el proceso Implantación y Aceptación del Sistema. (33)

4.3.1 Niveles de Prueba

Entre los niveles de prueba se encuentran el nivel de pruebas unitarias, nivel de pruebas de integración, nivel de pruebas del sistema y nivel de pruebas de aceptación. Una vez implementado el sistema fue sometido a los niveles de pruebas que a continuación se detallan, los cuales ayudaron a la detección de los errores existentes.

Pruebas Unitarias: Comienzan con la prueba de cada módulo. Una prueba unitaria es una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de estos funcione correctamente por separado. El objetivo de las pruebas unitarias es aislar cada parte del programa y mostrar que las partes individuales son correctas. Proporcionan un contrato escrito que el fragmento de código debe satisfacer. Estas pruebas aisladas proporcionan cinco ventajas básicas; fomentan el cambio, simplifican la integración, documentan el código, separan la interfaz del código y hacen que los errores estén más acotados y sean fáciles de localizar. (11)

Pruebas de Integración: A partir del esquema del diseño, los módulos probados se vuelven a probar combinados para probar sus interfaces. Pruebas integrales o pruebas de integración son aquellas que se realizan en el ámbito del desarrollo de software una vez que se han aprobado las pruebas unitarias. Únicamente se refieren a la prueba o pruebas de todos los elementos unitarios que componen un proceso, hecha en conjunto, de una sola vez. Consiste en realizar pruebas para verificar que un gran conjunto de partes del software funcionan juntos. (11)

Pruebas del Sistema: El software ensamblado totalmente con cualquier componente hardware que requiera, se prueba para comprobar que se cumplen los requisitos funcionales. Cualquier pieza de software completo, desarrollado o adquirido, puede verse como un sistema que debe probarse, ya sea para decidir acerca de su aceptación, para analizar defectos globales o para estudiar aspectos específicos de su comportamiento, tales como seguridad o rendimiento. Este tipo de pruebas estudia el producto completo. (11)

4.3.2 Métodos de Prueba

RUP propone dos métodos fundamentales: caja blanca y caja negra. A continuación se describen ambos métodos, haciéndose mayor énfasis en las pruebas de Caja Negra ya que serán las más utilizadas en la comprobación de la solución.

Pruebas de Caja Negra

Las pruebas de caja negra, también denominadas pruebas de comportamiento, se centran en los requisitos funcionales del software. O sea, permiten al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Estas pruebas no son una alternativa a las técnicas de pruebas de caja blanca, más bien se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores que se llevan a cabo sobre la interfaz del software. El objetivo es demostrar que las funciones del software son operativas, que las entradas se aceptan de forma adecuada y se produce un resultado correcto, y que la integridad de la información externa se mantiene, saber qué es lo que hace el software pero sin entrar en detalles de código, es decir, que es lo que hace, y no cómo lo hace (no se ve el código). Estas pruebas se centran principalmente en los requisitos funcionales del software y permiten encontrar: (34)

- ✓ Funciones incorrectas o ausentes.
- ✓ Errores de interfaz.
- ✓ Errores en estructuras de datos o en accesos a las bases de datos externas.
- ✓ Errores de rendimiento.
- ✓ Errores de inicialización y terminación.

Pruebas de Caja Blanca

La prueba de caja blanca, denominada a veces prueba de caja de cristal es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante los métodos de prueba de caja blanca, el ingeniero del software puede obtener casos de prueba que garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo, ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa; ejecuten todos los ciclos en sus límites y con sus límites operacionales, y ejerciten las estructuras internas de datos para asegurar su validez. (35)

4.3.3 Diseño de Casos de Prueba

A continuación se presenta el diseño de casos de prueba propuesto para el CU Gestionar Apuntes:

Diseño de Casos de Prueba: CU Gestionar Apuntes

Descripción general

El caso de uso se inicia cuando el Usuario de Hiperentorno selecciona la opción que le permite realizar una acción sobre el apunte. El actor puede incluir, ver, eliminar y modificar apuntes. En caso de que seleccione la opción de incluir un apunte, el sistema dará la posibilidad de insertar los datos que se necesitan en dependencia del usuario autenticado. Si el actor elige la opción de ver un apunte el sistema mostrará el contenido de este. Si el actor elige la opción de modificar un apunte, el sistema mostrará los datos que pueden ser editables dentro del apunte en dependencia del usuario autenticado, y una vez realizados los cambios, guardará las modificaciones. El sistema permite eliminar el apunte mostrando un mensaje de confirmación, terminando así el caso de uso.

Condiciones de ejecución

Debe haberse generado el escritorio de trabajo del usuario autenticado.

Para incluir un apunte, este debe estar asociado a una parte del contenido, se selecciona previamente la opción de apunte perteneciente al panel de herramientas y seguidamente el texto al que pertenecerá el apunte.

Se mostrarán todos los apuntes pertenecientes a la página del contenido, ya sea porque el usuario

autenticado es el autor de dichos apuntes o porque se le ha asignado temporalmente este permiso.

El tutor solo puede ver los apuntes de sus estudiantes, sin tener permiso a modificar ni eliminar estos.

Para modificar un apunte, debe haber seleccionado previamente la opción editar y el actor debe ser el responsable temporal de este.

Para eliminar un apunte, debe haber seleccionado previamente la opción eliminar y el actor debe tener el permiso de eliminar el elemento.

SC Incluir apunte

ID del escenario	Escenario	Comentario del apunte	Compartir a mis estudiantes	Respuesta del sistema	Flujo central
EC 1	Selecciona la opción de incluir una nueva entidad.	V	V	Brinda la posibilidad de introducir los datos del apunte.	Hiperentorno/ Menú Principal/ Contenidos/ Índice/ Selecciona una página para agregar apuntes dentro de ella/ Panel de herramientas/ Clic en el botón de apuntes/ Selecciona el
				Si es un profesor el usuario autenticado permite introducir: Comentario del apunte. Seleccionar: Compartir a mis estudiantes y/o Compartir a profesores y/o Incluir recurso existente. <u>Muestra el CP Asociar Recurso.</u> Incluir recurso nuevo. Muestra el <u>CU Gestionar Recursos Multimedia</u> o el <u>CU Gestionar Recursos Interactivos</u> o el <u>CU</u>	

				<p><u>Gestionar Recursos Estructurales.</u> texto al que pertenecerá el apunte/</p> <p>Si es un estudiante o director o administrador local el usuario autenticado, permite introducir:</p> <p>Comentario del apunte.</p> <p>Y permite:</p> <p>Guardar los datos.</p> <p>Cancelar la operación en cualquier momento.</p> <p>Valida los datos.</p> <p>Crea el apunte.</p>	<p>Introduce los datos para crear el apunte.</p>
EC 2	Selecciona la opción de cancelar			<p>Elimina los datos creados.</p> <p>Regresa a la vista anterior.</p>	<p>Hiperentorno/ Menú Principal/ Contenidos/ Índice/ Selecciona una página para agregar apuntes dentro de ella/</p>

					Panel de herramientas/ Clic en el botón de apuntes/ Selecciona el texto al que pertenecerá el apunte/ Introduce los datos para crear el apunte/ Cancelar
EC 3	Existen datos incompletos	/	V	Muestra el mensaje de información.	Hiperentorno/ Menú Principal/ Contenidos/ Índice/ Selecciona una página para agregar apuntes dentro de ella/
		V	/	Muestra un indicador sobre los campos vacíos.	

					Panel de herramientas/ Clic en el botón de apuntes/ Selecciona el texto al que pertenecerá el apunte/ Introduce datos incompletos.
--	--	--	--	--	---

SC Ver datos del apunte

ID del escenario	Escenario	Respuesta del sistema	Flujo central
EC 1	Acceder a una página del contenido para ver los datos del apunte.	En dependencia del usuario autenticado y de los permisos que tenga este, se muestran los apuntes pertenecientes a la página. Links: Mis Apuntes o Apuntes del Profesor en caso de ser un estudiante el usuario autenticado.	Hiperentorno/ Menú Principal/ Contenidos/ Índice/ Selecciona una página para ver apuntes dentro

		<p>Links: Mis Apuntes o Los de misde ella/ compañeros en caso de ser un Profesor.</p> <p>Links: Mis Apuntes o Los de los profesores en caso de ser un Director o Administrador Local.</p> <p>Links: Apuntes de mis estudiantes en caso de ser un Tutor.</p> <p>Y permite:</p> <p>Salir de la vista actual.</p> <p>De acuerdo con los permisos del usuario permite:</p> <p>Editar los datos del apunte.</p> <p>Eliminar el apunte.</p>	
EC 2	<p>Selecciona la opción de editar los datos del apunte.</p>	<p>Brinda la posibilidad de modificar los datos de la entidad. Ver SC Modificar datos del apunte.</p>	<p>Hiperentorno/ Menú Principal/ Contenidos/ Índice/ Selecciona una página para ver apuntes dentro de ella/ Seleccionar Link Editar del apunte a modificar.</p>
EC 3	<p>Selecciona la opción de eliminar el apunte.</p>	<p>Brinda la posibilidad de eliminar el apunte. Ver SC Eliminar apunte.</p>	<p>Hiperentorno/</p>

			Menú Principal/ Contenidos/ Índice/ Selecciona una página para ver apuntes dentro de ella/ Seleccionar Link Eliminar del apunte a eliminar.
--	--	--	---

SC Modificar datos del apunte

ID del escenario	Escenario	Comentario del apunte	Compartir a mis estudiantes	Respuesta del sistema	Flujo central
EC 1	Selecciona la opción de editar los datos de la entidad	V	V	Muestra los datos de la entidad a modificar (excepto los datos del resaltado, es decir, solo se podrá modificar los datos del apunte, no el texto resaltado al que pertenece este) en dependencia del usuario autenticado y brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos	Hiperentorno/ Menú Principal/ Contenidos/ Índice/ Selecciona una página para agregar apuntes dentro de ella/ Panel de herramientas/

			<p>seleccionando diferentes.</p> <p>Permite además:</p> <p>Actualizar los datos.</p> <p>Cancelar la operación en cualquier momento.</p> <p>Al modificar los datos que necesite y seleccionar la opción de actualizar, se valida los datos y se actualiza el apunte mostrando los datos de este. Ver SC Ver datos del apunte.</p>	<p>Clic en el botón de apuntes/</p> <p>Selecciona el texto al que pertenecerá el apunte/</p> <p>Introduce los datos para modificar el apunte/</p> <p>Actualizar</p>
EC 2	<p>Selecciona la opción de cancelar</p>		<p>Elimina los datos creados.</p> <p>Muestra un mensaje de información. Regresa a la vista anterior.</p>	<p>Hiperentorno/</p> <p>Menú Principal/</p> <p>Contenidos/</p> <p>Índice/</p> <p>Selecciona una página para agregar apuntes dentro de ella/</p> <p>Panel de herramientas/</p> <p>Clic en el botón de</p>

					<p>apuntes/ Selecciona el texto al que pertenecerá el apunte/ Introduce los datos para modificar el apunte/ Cancelar</p>
EC 3	Existen datos/incompletos		✓	<p>Muestra el mensaje de información. Muestra un indicador sobre los campos vacíos.</p>	<p>Hiperentorno/ Menú Principal/ Contenidos/ Índice/ Selecciona una página para agregar apuntes dentro de ella/ Panel de herramientas/ Clic en el botón de apuntes/</p>

					<p>Selecciona el texto al que pertenecerá el apunte/</p> <p>Introduce datos incompletos.</p>
--	--	--	--	--	--

SC Eliminar apunte

ID del escenario	Escenario	Respuesta del sistema	Flujo central
EC 1	Selecciona la opción de eliminar.	Muestra el mensaje de confirmación "El apunte será eliminado. ¿Desea continuar?" y permite: Aceptar. Cancelar la operación.	Hiperentorno/ Menú Principal/ Contenidos/ Índice/ Selecciona una página para ver apuntes dentro de ella/ Clic en el Link Eliminar del apunte que desea eliminar/ Aceptar.
EC 2	Selecciona la opción Cancelar.	Brinda la posibilidad de regresar a la vista anterior.	Hiperentorno/ Menú Principal/

			Contenidos/ Índice/ Selecciona una página para ver apuntes dentro de ella/ Clic en el Link Eliminar del apunte que desea eliminar/ Cancelar.
--	--	--	--

El resto de los casos de prueba realizados pueden encontrarse en el **Anexo # 9: Diseño de casos de pruebas por CU.**

4.3.4 Resultados Obtenidos

Durante el desarrollo del módulo iTopics se realizaron pruebas unitarias para ir comprobando el funcionamiento del software. Fueron realizadas por el desarrollador aprovechando las ventajas de compilación paso a paso que brinda el IDE en conjunto con el servidor Apache. Estas no se planificaron ni se registraron sus resultados, fueron haciéndose a medida que la solución se desarrollaba.

Para evaluar la solución se realizaron varias iteraciones donde se probó el software íntegramente. Prestando gran atención a las pruebas de integración con vista a probar funcionalidades que relacionan al módulo iTopics con el resto de los módulos de la plataforma educativa ZERA.

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

Pruebas Internas: Son realizadas por el equipo de calidad interna del proyecto (en este caso las analistas) con el fin de entregar un producto con la menor cantidad de errores posibles. Se centraron en el

cumplimiento de las funcionalidades descritas en el listado de requerimientos y de casos de uso del sistema.

Pruebas Cruzadas: Fueron realizadas al sistema por el equipo de desarrollo del proyecto (analistas de diferentes módulos), con el fin de encontrar la mayor cantidad de errores posibles en término de validaciones, pautas definidas por arquitectura de información, formato de los campos, entre otras.

Pruebas de Liberación: Pruebas realizadas por un tercero, en este caso Calisoft, institución encargada de validar que el software cuente con la calidad requerida para ser entregado a los clientes finales.

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

Pruebas Internas:

Módulo	Casos de Uso	No Conformidades			
		Alta	Media	Baja	Total
iTopics	7	1	2	5	8

Pruebas Cruzadas:

Módulo	Casos de Uso	No Conformidades			
		Alta	Media	Baja	Total
iTopics	7	1	4	6	11

Pruebas de Liberación 1era Iteración:

Módulo	Casos de Uso	No Conformidades			
		Alta	Media	Baja	Total
iTopics	7	2	4	8	14

4.5 Conclusiones

Los artefactos generados en la implementación del módulo, describen las especificidades de las técnicas de programación que se utilizaron. Además, las pruebas realizadas validaron que las funcionalidades desarrolladas satisfacen los requisitos especificados, dejando listo el software con vista a su implantación.

Conclusiones

Con la culminación del presente trabajo se ha dado cumplimiento a los objetivos trazados en la investigación, obteniéndose como resultado principal, un módulo funcional integrado a la Plataforma educativa ZERA, con toda la documentación generada durante el proceso de desarrollo de software, esta es la prueba más fehaciente del cumplimiento de los objetivos a grandes rasgos.

- ✓ El estudio realizado como parte de la investigación sirvió de apoyo en la toma de decisiones con vista a un desarrollo eficiente del módulo.
- ✓ Se demostró lo beneficioso que resulta la utilización de una metodología tan sólida como RUP paralelamente a la utilización del framework Symfony, lo que propició una implementación correcta de la aplicación.
- ✓ Se diseñó e implementó el conjunto de clases que dan cumplimiento a los requisitos funcionales y no funcionales asociados a los casos de uso del módulo. Otra tarea muy importante fue la integración al sistema, que se logró llevando a cabo buenas prácticas de diseño e implementación, como son patrones, convenciones de código y documentación de este, llevando a correcto término todos los objetivos propuestos como cumplimiento de las actividades planificadas.
- ✓ Se demostró a partir de las diferentes iteraciones de pruebas practicadas al software, que este satisface los requisitos funcionales y no funcionales obtenidos durante el flujo de trabajo Requerimientos.

Recomendaciones

- ✓ Continuar potenciando el uso del framework Symfony en la facultad ya que reporta un conjunto de ventajas el uso de este.
- ✓ Mejorar el sistema de mensajería instantánea (chat) montando un servidor jabber para el funcionamiento de este, relacionando su gestión con la plataforma, ya que la mayoría de los servidores jabber soportan múltiples conexiones y mediante conexiones bidireccionales se disminuye el intercambio entre cliente y servidor, para evitar sobrecargar al servidor de aplicaciones con las peticiones de los usuarios del chat.
- ✓ Mejorar el sistema de carga de las páginas para disminuir el tiempo en que estas se muestran y al mismo tiempo no sobrecargar la memoria de los navegadores.
- ✓ Redefinir un modelo de datos para la gestión de resaltado de texto dentro del contenido, así como de la gestión de apuntes, para lograr una mejor estructuración de la información y se pueda dar soporte a los reportes de las acciones realizadas sobre las funcionalidades mencionadas anteriormente.
- ✓ Evitar que las selecciones y las notas sean eliminadas cuando la página donde estas se encuentran sea editada.
- ✓ Transmitir experiencias a otros proyectos que utilicen metodología, herramientas y tecnologías similares.
- ✓ Tener en cuenta este trabajo para proyectos futuros, siempre y cuando se cumpla con las normas de confidencialidad establecidas.

Referencias bibliográficas

1. **Agudelo, Mónica María.** Plataformas Educativas. *Plataformas Educativas*. [En línea] 2006. [Citado el: 20 de junio de 2010.] <http://aprendeonline.udea.edu.co/banco/html/plataformaseducativas/>.
2. Pedagogía a tu alcance. [En línea] [Citado el: 10 de febrero de 2011.] http://patalcance.rimed.cu/module/biblioteca/datos_esp.php?pag=glo&num=244&Tip=ssb&idMod=7
3. moodle. *moodle*. [En línea] moodle, septiembre 30, 2009. [Citado el: 26 septiembre de 2010.] <http://moodle.org>.
4. **Claroline.** Claroline. *Claroline*. [En línea] Claroline, enero 1, 2008. [Citado el: 26 septiembre de 2010.] <http://www.claroline.net/about-claroline.html>.
5. **Andalucía, Junta de.** Plataforma Educativa Helvia. [En línea] Junta de Andalucía, diciembre 2010. [Citado el: 27 junio de 2010.] <http://www.juntadeandalucia.es/averroes/helvia/sitio/index.cgi>.
6. Junta de Andalucía, Consejería de Educación. [En línea] [Citado el: 5 de Febrero de 2011.] http://www.juntadeandalucia.es/averroes/helvia/sitio/index.cgi?wid_seccion=2&wid_item=24#.
7. Eleven. *e-Lerning Virtual Envaironment*. [En línea] 2010. [Citado el: 11 de Septiembre de 2010.] <http://www.plataformaeleven.com>.
8. **Peña, J. C. S** Plataforma de Teleformación aprenDIST. [En línea] 2005. [Citado el: 12 de octubre de 2010.] <http://www.forumcyt.cu/UserFiles/forum/Textos/0301877.pdf>
9. **Kruchten, Philippe.** *Rational Unified Process, An Introduction, Third Edition*. 2003.
10. **Penadés, Patricio Letelier y M^a Carmen.** Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). España: Universidad Politécnica de Valencia, 2006.
11. **James Rumbaugh, Ivar Jacobson, Grady Booch.** *El lenguaje Unificado de Modelado. Manual de Referencia*. s.l.: Addison Wesley.

12. **Pérez, J. E.** Introducción a Ajax. *Introducción a Ajax*. [En línea] 2008. [Citado el: 3 de enero de 2011.] <http://librosweb.es>.
13. **Pérez, J. E.** Introducción a XHTML. [En línea] 2008. [Citado el: 2 de enero de 2011.] <http://librosweb.es/>.
14. **Tidwell, Doug.** Introducción a XML. [En línea] [Citado el: 5 de Febrero de 2011.] <http://www.ibm.com/developerWorks>.
15. **Pérez, J. E.** Introducción al CSS. *Introducción al CSS*. [En línea] 2008. [Citado el: 2 de enero de 2011.] <http://librosweb.es>.
16. **Pérez, J. E.** Introducción a JavaScript. *Introducción a JavaScript*. [En línea] 2009. [Citado el: 3 de enero de 2011.] <http://librosweb.es>.
17. **Hinostroza, R. R.** Características de PHP. [En línea] 2005. [Citado el: 10 de enero de 2011.] <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.
18. **Gutiérrez, J. J.** ¿Qué es un framework web? [En línea] 2009. [Citado el: 2 de enero de 2011.] http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf.
19. SlideShare [En línea] [Citado el: 11 de Diciembre de 2010.] <http://www.slideshare.net/tutorialsruby/d422guiaparaeldesarrollodegadgets10-2620805>
20. **Alvarez, Miguel Angel.** [desarrolloweb.com](http://www.desarrolloweb.com). [En línea] 2010. [Citado el: 10 de diciembre de 2010.] www.desarrolloweb.com.
21. java.sun.com. [En línea] [Citado el: 21 de Febrero de 2008.] <http://java.sun.com>.
22. **Potencier, F.** *Symfony la guía definitiva*. 2009.
23. Doctrine. [En línea] 2010. [Citado el: 15 de Septiembre de 2010.] <http://www.doctrine-project.org/>.
24. **PostgreSQL.** PostgreSQL - Sitio Oficial. [En línea] 2010. [Citado el: 10 de diciembre de 2010.] <http://www.postgresql.org>.

25. **Ltd, Netcraft.** Netcraft Ltd. [En línea] [Citado el: 5 de Febrero de 2011.] <http://news.netcraft.com/archives/2011>.
26. GNU Operating System [En línea] [Citado el: 17 de Enero de 2010] <http://www.gnu.org/licenses/license-list.es.html>
27. <http://www.abadiadigital.com/articulo/google-confirma-que-apache-es-el-doble-de-seguro-que-iis/>. [En línea] [Citado el: 5 de Febrero de 2011.]
28. **Guerrero, S.** Netbeans 6.8 con soporte para PHP 5.3 y Symfony. [En línea] 2009. [Citado el: 12 de diciembre de 2010.] <http://www.sergioguerrero.es/php/netbeans-soporte-php53-symfony>.
29. UPtoDOWN. [En línea] [Citado el: 12 de 2 de 2011.] <http://argouml.uptodown.com/>.
30. Sitio Web oficial Visual-Paradigm. *Sitio Web oficial Visual-Paradigm*. [En línea] 2011. [Citado el: 12 de enero de 2011.] <http://www.visual-paradigm.com/product/vpuml/>.
31. **James Rumbaugh, Ivar Jacobson, Grady Booch.** *El Proceso Unificado de Desarrollo de Software. Manual de Referencia*. s.l.: Addison Wesley.
32. Software, "*Fase de Elaboración. FT Prueba (procedimientos genéricos y aplicación de algunos tipos de pruebas simples)*". [En línea] 2005. [Citado el: 17 marzo de 2011.]
33. Métrica, M. "Construcción del Sistema de Información." **Versión 3**. [En línea] 2007. [Citado el: 17 marzo de 2011.]
34. **Pressman, Roger S.** *Ingeniería del Software. Un enfoque práctico*. España : McGraw-Hill, 2002. [En línea] 2002. [Citado el: 20 febrero de 2011.]
35. **Wiley, John.** *Software Engineering Standards*. s.l. : IEEE Press, 2001. ANSI/IEEE Standard 829-1983
36. Sitio Web ISSUU. *Guía de Consejos para un centro TIC ideal*. [En línea] 2011. [Citado el: 10 de febrero de 2011.] <http://issuu.com/redtic/docs/consejos09>

37. **Vera, Roberto Garduño.** Contenido educativo en el aprendizaje virtual. [En línea] 2008. [Citado el: 2 de enero de 2011.] <http://www.ejournal.unam.mx/ibi/vol23-47/IBI002304702.pdf>
38. **Alvarez, Miguel Angel.** desarrolloweb.com. [En línea] 2010. [Citado el: 10 de diciembre de 2010.] www.desarrolloweb.com.
39. Wizhosting. Alojamiento web en Argentina. [En línea] 2010. [Citado el: 10 de Septiembre de 2010.] <http://www.wizhosting.com/e-learning>.
40. **SCORM, A.** Sharable Course Object Reference Model. *Sharable Course Object Reference Model*. [En línea] 2002. [Citado el: 20 de septiembre de 2010.] <http://ares.cnice.mec.es/informes/16/contenido/39.htm>.
41. **Torre, Aníbal de la.** Bitácora de Aníbal de la Torre. *Bitácora de Aníbal de la Torre*. [En línea] Bitácora de Aníbal de la Torre, octubre 1, 2005. [Citado el: 17 diciembre de 2010.] <http://www.adelat.org/media/gadgetopolvo/LO/index.htm>.

Bibliografía

Conferencias de ISW. UCI, departamento de Ingeniería de Software. 2010-2011.

Collado, M. (2003) "Pruebas de software. Técnicas de prueba del software. Estrategias de prueba del software."

Jalon, Javier Garcia de. *Aprenda Java como si estuviera en primero.* San Sebastian : Escuela Superior de Ingenieros Industriales, 2000.

James Rumbaugh, Ivar Jacobson, Grady Booch. *El lenguaje Unificado de Modelado. Manual de Referencia.* s.l. : Addison Wesley.

Introducción a CSS. [En línea] 2008. www.librosweb.es.

Introducción a JavaScript. [En línea] 2008. www.librosweb.es.

Introducción a XHTML. [En línea] 2008. www.librosweb.es.

Kruchten, Philippe. *Rational Unified Process, An Introduction, Third Edition.* 2003.

Labs, Sensio. 2009. More with symfony. [En línea] 2009. www.librosweb.com.

María A. Mendoza Sanchez. Ing. Informático, Microsoft Certified Professional ,Analísta y Desarrolladora - TeamSoft. *Metodologías De Desarrollo De Software.* Peru : s.n., 2004.

Molpeceres, Alberto. Procesos de desarrollo: RUP, XP y FDD. *Willi.net.* [En línea] 2002. [Citado el: 10 de enero de 2011.] <http://www.willydev.net/descargas/articulos/general/cualxpfdrrup.PDF>.

NetBeans. NetBeans. *NetBeans.* [En línea] 2008. [Citado el: 12 de febrero de 2011.] <http://www.netbeans.org>.

Penadés, Patricio Letelier y M^a Carmen. *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP).* España : Universidad Politécnica de Valencia, 2006.

Pérez, Javier Eguiluz. 2008. Introducción a Ajax. [En línea] 2008. www.librosweb.es.