

Universidad de las Ciencias Informáticas

Facultad 4



**Título: Análisis y Diseño del módulo Biblioteca de
la Plataforma de Gestión del Aprendizaje ZERA.**

Trabajo de Diploma para optar por el título de
Ingeniero Informático

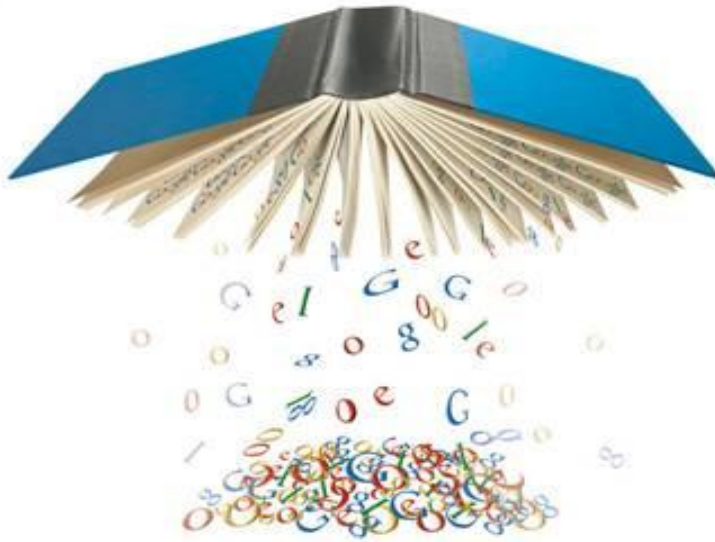
Autor: Marisniulkis Lescaille Cos

Tutor: MSc. Abel Ernesto Lorente Rodríguez

Consultantes: Dr. Alcides Cabrera Campos

MSc. Dora Enma Nicó Pérez

La Habana, Junio de 2011



Sin bibliotecas, ¿qué nos quedaría? No tendríamos pasado ni futuro.

Ray Bradbury

Declaración de Autoría

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Departamento de Producción de Herramientas Educativas (FORTES) de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Marisniulkis Lescaille Cos

Abel E. Lorente Rodríguez

Agradecimientos

AGRADECIMIENTOS

A mi familia que siempre me ha apoyado, a mis primos, abuelos y tíos, en especial a mi tía Liudys y mi tío Manolo.

A Dora, por ser como una madre para mí y brindarme otra familia, por los consejos, por los regaños, por el cariño y la comprensión.

A Alcides, Tito y Annia, los integrantes de mi otra familia, sin ellos todo hubiese sido más difícil.

A mi amigas: Dailenis, por estos cinco años de compartir aventuras, por la paciencia y el apoyo; Yari, que nunca deja de enseñarme cosas; Dairys, incondicional. A Tiburcia, que tanto me enseñó de la vida en tan poco tiempo. A mi amigo: Husse, desde primer año incondicional.

A Lisbet, Lisandra, Lis, Carlos y todos mis amigos del pre.

A todos mis compañeros de grupos con los que transité en la universidad.

A Osdalme.

A Yanirys, Jose, Piña, Mairelis, el otro Jose Antonio, Yoennis, Yolanda, Joe, Ismael, que colaboraron voluntariamente de una forma u otra para mejorar este trabajo.

A las chicas de DevGirls.

A todos los que no he puesto aquí, pero dieron su granito de arena en este trabajo.

Especialmente a mi mamá y mi abuela que me han guiado toda la vida.

Dedicatoria

DEDICATORIA

A mi abuela Reyna, que ha sido y es especial conmigo, por ser la mejor abuela que se puede tener.

A mi mamá por darme la libertad de buscar mi camino y abrirme otros, por creer en mí, se que estarás muy orgullosa.

A mi familia, con la siempre he podido contar.

A Dora, por todo.

A mis amigos, en especial a Daile y Yari por el recorrido, que vale más que llegar a la cima.

Resumen

RESUMEN

En la Universidad de las Ciencias Informáticas se desarrolla una Plataforma de Gestión del Aprendizaje, ZERA, destinada a estudiantes de la enseñanza media. Este trabajo se centra en el módulo Biblioteca del que se requiere una adecuación teniendo en cuenta las tendencias internacionales en el desarrollo de plataformas educativas, bibliotecas digitales e integración entre ambos conceptos. Atendiendo además a la evolución de la web hacia el uso de software de interacción social y el aprendizaje colaborativo.

Para dar solución a la problemática planteada se realizó el análisis y el diseño al módulo, partiendo de la realización de un estudio teórico de los conceptos asociados al problema y sistemas similares, lo que permitió capturar requerimientos así como la utilización de técnicas para su validación y el empleo de patrones de casos de uso en la estructuración de los mismos. Se agregaron opciones al módulo que permitirán una mejor visualización, configuración y uso de los recursos que se muestran. En función de esto se siguió el Proceso de Desarrollo Unificado (RUP) como metodología de software, el lenguaje unificado de modelado (UML) como lenguaje de modelado visual y la herramienta CASE Visual Paradigm para UML.

PALABRAS CLAVE

e-learning, bibliotecas digitales, recursos de aprendizaje, plataformas de gestión de aprendizaje.

Índice de figuras y tablas

ÍNDICE DE FIGURAS

Figura 1. Diagrama de clases del dominio.	43
Figura 2. Diagrama de actores.	50
Figura 3. Diagrama de Casos de Uso del Sistema.	51
Figura 4. Patrón de caso de uso CRUD Completo.	53
Figura 5. Patrón de caso de uso Rol común.	53
Figura 6. Diagrama de clases del análisis Caso de Uso Gestionar Búsqueda.	56
Figura 7. Diagrama de paquetes del análisis.	57
Figura 8. Patrón Modelo-Vista-Controlador.	58
Figura 9. Diagrama de clases del diseño para el caso de uso Gestionar búsqueda.	59
Figura 10 . Diagrama de paquetes del diseño.	60
Figura 11. Diagrama de secuencia caso de uso Gestionar Búsqueda, escenario Buscar.	60
Figura 12. Modelo de Datos.	61
Figura 13. Diagrama de Despliegue.	62

ÍNDICE DE TABLAS

Tabla 1. Diferencias entre metodologías ágiles y no ágiles.	35
Tabla 2. Cálculo del coeficiente de conocimiento.	65
Tabla 3. Ubicación de los expertos en las fuentes de argumentación.	65
Tabla 4. Matriz coeficiente de argumentación por experto.	66
Tabla 5. Coeficientes y nivel de competencia de los expertos.	66
Tabla 6. Matriz del criterio de experto por indicador con cálculos de variables.	67
Tabla 7 Grado de concordancia por indicador.	68
Tabla 8. Votos emitidos por los expertos.	69

Índice de contenido

ÍNDICE

INTRODUCCIÓN	12
1. CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	16
Introducción	16
1.1. Conceptos asociados al e-learning.....	16
1.1.1. Definición de e-learning.....	16
1.1.2. Web 2.0.....	17
1.1.3. E-learning 2.0.....	18
1.2. Plataformas de Gestión del Aprendizaje	18
1.2.1. Plataformas educativas a nivel internacional.	20
1.2.2. Plataformas Educativas en Cuba	21
1.3. Hiperentornos de aprendizaje.....	21
1.4. Concepción constructivista del aprendizaje	23
1.5. Metadatos.....	24
1.6. Objeto de aprendizaje.	25
1.6.1. Estándares para objetos de aprendizaje	26
1.7. Bibliotecas digitales.	26
1.7.1. Algunas bibliotecas digitales. Características.	28
1.7.2. Bibliotecas Virtuales Cubanas	29
1.7.3. Bibliotecas digitales y e-learning	29
1.8. Repositorio	32
1.9. Plataforma de Gestión del Aprendizaje ZERA	32
1.10. Metodologías de desarrollo de software, herramientas y lenguajes de modelado.....	33

Índice de contenido

1.10.1.	Metodologías de desarrollo de software. Metodologías ágiles y pesadas.....	33
1.10.2.	Metodologías ágiles.....	34
1.10.3.	Metodologías pesadas o tradicionales.....	37
1.10.4.	Selección de la metodología.....	39
1.11.	Lenguaje de Modelado.....	40
1.12.	Herramientas CASE.	40
1.12.1.	Visual Paradigm.....	40
1.12.2.	Rational Rose.....	41
1.12.3.	Selección de herramienta CASE a utilizar.....	41
	Conclusiones.....	41
2.	CAPÍTULO 2: DESCRIPCIÓN DE LA PROPUESTA.....	43
	Introducción.....	43
2.1.	Modelo de Dominio.....	43
2.2.	Captura de Requerimientos.....	45
2.3.	Requerimientos funcionales del sistema.....	46
2.4.	Requerimientos no funcionales del sistema.....	48
2.5.	Descripción del sistema propuesto.....	49
2.5.1.	Actores.....	49
2.5.2.	Casos de Uso.....	50
2.5.3.	Diagrama de Casos de Uso del sistema.	51
2.5.4.	Descripción textual de los Casos de Uso del Sistema.....	51
2.5.5.	Patrones de Casos de Uso.....	52
3.	CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA PROPUESTO.....	55

Índice de contenido

Introducción	55
3.1. Análisis	55
3.1.1. Modelo de Análisis.....	55
3.1.2. Clases del análisis	55
3.1.3. Diagrama de clases del análisis	56
3.1.4. Diagrama de paquetes del análisis	56
3.2. Arquitectura propuesta.....	57
3.3. Diseño	58
3.3.1. Modelo de Diseño.....	58
3.3.2. Diagrama de clases del diseño.	59
3.3.3. Subsistemas de diseño.	59
3.3.4. Realizaciones de Casos de Uso. Diagramas de interacción.....	60
3.3.5. Modelo de Datos.....	60
3.3.6. Diagrama Entidad-Relación	61
3.3.7. Diagrama de despliegue	62
Conclusiones.....	62
4. CAPÍTULO 4: VALIDACIÓN DE LA PROPUESTA.....	63
Introducción	63
4.1. Validación de requisitos	63
4.2. Técnicas de validación.....	63
4.2.1. Casos de prueba.....	64
4.2.2. Prototipo de interfaz no funcional.....	64
4.3. Validación de la propuesta mediante criterios de expertos	64

Índice de contenido

Conclusiones.....	69
CONCLUSIONES	70
RECOMENDACIONES	71
BIBLIOGRAFÍA	72
GLOSARIO.....	77

Introducción

INTRODUCCIÓN

La aparición, desarrollo y constante evolución de las Tecnologías de la Información y las Comunicaciones (TICs) han transformado muchas de las esferas de la vida del hombre y sus actividades, en lo que ha devenido la Sociedad de la Información. La educación, siendo la escuela su exponente más visible, tiene un papel preponderante en esta nueva sociedad: la de formar a las nuevas generaciones, propiciando la adquisición de conocimientos y valores que tributen a la integración plena del individuo a la sociedad. En este proceso, tradicionalmente se han visto envueltos factores disímiles como la aparición de la escritura, la invención de la imprenta y la aparición de la enseñanza por correspondencia. El desarrollo de las TICs ha supuesto una revolución en la manera en que se comunican las personas, desde la invención del telégrafo hasta la actual transmisión de voz e imágenes a través de redes de datos. Es debido a los cambios en la velocidad y el poder de las comunicaciones, unido al aumento de la capacidad para enviar, recibir y usar información y a menudo la desaparición de las restricciones espacio-temporales que se multiplican las posibilidades educativas.

En este contexto surge el e-learning, que es un conjunto de aplicaciones, tecnologías y servicios orientados a facilitar la enseñanza y el aprendizaje a través de internet o intranets que facilitan el acceso y la comunicación con otros participantes.

En la actualidad son muchas las herramientas disponibles que favorecen el e-learning y posibilitan la administración, documentación, seguimiento y reportes de sostén al aprendizaje, tanto de licencias de pago como gratuitos. A estas herramientas se les denomina Learning Management Systems (comúnmente LMS).

El creciente desarrollo de las TICs ha emergido con una denominación y filosofía de trabajo en la web: la web 2.0. Término comúnmente asociado a las aplicaciones que facilitan el intercambio de información interactivo, el diseño centrado en el usuario y la colaboración, en lo que difiere de la llamada web 1.0 como una mera fuente de información. La web 2.0 permite a los usuarios hacer más que sólo recuperar información, da la posibilidad de escoger si interactuar o colaborar en un medio social como creadores de contenidos generados por el usuario en una comunidad virtual. Ejemplos de web 2.0 constituyen sitios de redes sociales, blogs personalizados, wikis, servicios de alojamiento de videos e imágenes, entre otros.

Introducción

El e-learning, como parte de este fenómeno, históricamente constituía una forma más del tradicional estilo de educación autocrática, donde es concebido para transferir conocimiento. Sin embargo, más recientemente y desde la óptica de la web 2.0 se admite como sistemas de aprendizaje colaborativo donde se hace énfasis en el aprendizaje social y el uso de “software social”. Como consecuencia, la tendencia actual en la concepción de los LMS va en torno a la asunción de que el aprendizaje es socialmente construido abogando por el hecho de que una de las mejores maneras de aprender algo es enseñándolo a alguien más. Según Juan Rafael Fernández García en La Hora del e-aprendizaje¹ el diseño y desarrollo de herramientas para el e-learning se basan en la teoría del aprendizaje que se denomina *pedagogía construccionista social*. Para el construccionismo el aprendizaje es particularmente efectivo cuando se construye algo que debe llegar a otros (una frase, un artículo, un software).

En el departamento de producción de herramientas educativas de la Universidad de las Ciencias Informáticas (UCI) se desarrolla una herramienta de gestión del aprendizaje, llamada ZERA, la cual está basada en la concepción pedagógica de Hiperentornos de Aprendizaje, creada y desarrollada inicialmente por el Ministerio de Educación de Cuba. Los Hiperentornos son una mezcla sistémica de diversas tecnologías de software (libros electrónicos, evaluadores, juegos, simuladores, etc.) sustentadas en tecnologías denominadas hipermedia, término que denota la fusión entre tecnologías hipertexto y multimedia.

Entre los módulos con que cuentan los hiperentornos se encuentra “Biblioteca”, concebido como un repositorio de recursos multimedia y objetos de aprendizaje interactivos. La concepción actual y el desarrollo de este módulo tienen diversas limitaciones, no está acorde a las tendencias actuales en el desarrollo de LMS, no se implementan funcionalidades que permitan el uso de estos recursos interactivos desde una perspectiva actual de aprendizaje colaborativo. Está desarrollado de manera que dada una búsqueda por un criterio muestra un listado de los recursos que tiene el sistema almacenado sin brindar ninguna funcionalidad especializada sobre este contenido.

Sin embargo, se considera que el módulo debe evolucionar hacia un escenario colaborativo, más allá del acceso a la información, poder localizarla rápidamente y brindar la capacidad de adaptarla según los requerimientos cognitivos particulares de los individuos: evaluar, compartir entre otras opciones en las que

¹ http://people.ofset.org/jrfernandez/edu/n-c/moodle_1/index.html

Introducción

el usuario tiene un papel fundamental en la construcción del conocimiento que genera, no solo para sí, sino también para otros; que se aproximan a las tendencias actuales en el desarrollo de los LMS.

Por lo que se define como **problema de investigación** ¿Cómo modelar el módulo Biblioteca de la Plataforma de Gestión del Aprendizaje ZERA, de manera que facilite el uso y evaluación de los recursos educativos y le aporte un enfoque constructivista al proceso de enseñanza-aprendizaje? El **objeto de estudio** lo constituyen las tecnologías de la informática y las comunicaciones aplicadas en el proceso de enseñanza-aprendizaje, por lo que se define como **objetivo general**: Analizar y diseñar el módulo Biblioteca de la Plataforma de Gestión de Aprendizaje ZERA. El **campo de acción** está centrado en el proceso de visualización, configuración personalizada y uso colaborativo de los recursos educativos y objetos de aprendizaje de la plataforma de aprendizaje ZERA.

Como **idea a defender** se plantea que con el análisis y diseño del módulo Biblioteca se brindará una entrada para el flujo de implementación que proporcionará un entendimiento claro de las funcionalidades y facilitará las tareas durante este flujo de trabajo.

Los **objetivos específicos** son:

- Caracterizar las herramientas disponibles para repositorios de objetos de aprendizaje así como para bibliotecas digitales y su integración con sistemas e-learning.
- Especificar los requisitos funcionales y no funcionales del módulo Biblioteca de la plataforma ZERA.
- Realizar el análisis y diseño del módulo Biblioteca.
- Validar la propuesta presentada.

Aportes prácticos esperados:

Como resultado se espera obtener el análisis y diseño del módulo Biblioteca de la Plataforma de Gestión del Aprendizaje ZERA que sirva como entrada para las fases de implementación y pruebas y contribuya en la construcción del aprendizaje de los usuarios.

Estructura del documento:

El presente texto está estructurado por capítulos. En el capítulo 1 se abarca la Fundamentación Teórica, en el capítulo 2 Descripción de la Propuesta, en el capítulo 3, Análisis y el Diseño de la propuesta de

Introducción

solución, que constituye la parte fundamental de este trabajo y por último el capítulo 4 Validación de la Propuesta.

Capítulo 1 Fundamentación teórica

1. CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

En este capítulo se exponen los basamentos teóricos necesarios para comprender la solución propuesta. Se presentan aspectos generales del e-learning así como otros conceptos relacionados con este como metadatos y web 2.0; se definen características de las bibliotecas digitales y se analiza la integración entre estos conceptos. Se abordan además, las tendencias actuales en la concepción de los LMS (Learning Management Systems), los hiperentornos y las tecnologías hipermedias. Se realiza el estudio de diferentes metodologías, herramientas y lenguajes que darán soporte a la solución del problema de investigación.

1.1. Conceptos asociados al e-learning.

1.1.1. Definición de e-learning

El e-learning al ser un fenómeno asociado al cambiante mundo de las redes digitales es un concepto no definido del todo y que evoluciona constantemente debido a la evolución de las distintas tecnologías que le sirven de soporte. Su traducción literal, tomando la e como acrónimo de electrónico, es aprendizaje electrónico o por medios electrónicos.

Sin embargo pueden encontrarse varias definiciones desde diferentes puntos de vista con mayor o menor alcance. Básicamente es la instrucción o adquisición de conocimientos con el uso total o parcial de Tecnologías de la Informática y las Comunicaciones (TICs).

El e-learning es la transferencia de habilidades y conocimientos a través de una computadora y una red habilitada. Puede ser autónomo o guiado por un instructor en forma de texto, imágenes, animaciones, video streaming y audio. Las fuentes de contenido son diversas: internet, intranet/extranet, audio o cintas de video, televisión satelital y CD-ROMs [19].

Capítulo 1 Fundamentación teórica

Debido a su naturaleza digital el e-learning puede ser adaptado, con bajos gastos en tiempo y costo, a individuos o pequeños grupos. El mismo contenido puede ser ofrecido en multitud de formatos: facilitado, autónomo, usado en las aulas o una combinación de los anteriores.

La vía más común de aplicar el concepto de e-learning es a través de la implementación de Sistemas Administradores de Aprendizaje, también se les denomina Plataformas de Gestión del Aprendizaje (LMS, Learning Management Systems) y puede estar asociado a Course Management Systems (CMS) y Learning Content Management Systems (LCMS). El contenido puede ser obtenido listo para su presentación a los usuarios o confeccionado de acuerdo con las necesidades educativas de la organización.

1.1.2. Web 2.0

Recientemente, con la revolución que trajo consigo el uso de internet y sus sucesivos cambios, un término ha generado diferentes acrónimos: web 2.0. La web 2.0 tiene muchas definiciones: es un fenómeno más que una tecnología en sí que tiene su base en la interacción de diferentes aplicaciones que facilitan el acceso y uso de la información, siendo la interoperabilidad, el diseño centrado en el usuario y la colaboración, los factores fundamentales detrás de este concepto. Ejemplos de web 2.0 son las comunidades en la red, las aplicaciones web, los servicios de red social, los servicios de alojamiento, las wikis y los blogs.

La web 2.0 es la red como plataforma, abarcando todas las computadoras conectadas; las aplicaciones asociadas a este fenómeno son aquellas que entregan software como un servicio constantemente actualizado, que es mejor cuanto más las personas lo usen, consumiendo y mezclando datos de múltiples fuentes, incluyendo usuarios individuales, mientras proveen sus propios datos en una forma que a la vez puede ser mezclada por otros, creando efectos en la red y arquitectura de participación, más allá de la metáfora de la web 1.0, para crear experiencias de usuario enriquecedoras. [20]

El uso de la web 2.0 permite además de localizar y obtener información, que era el paradigma de la web antes de la aparición de este, el interactuar con otros usuarios o contribuir en la creación de los contenidos

Capítulo 1 Fundamentación teórica

de los sitios web. La web 2.0 es una forma relativamente nueva de ver la web y de usarla, no incluye formalidades técnicas. Le proporciona al usuario mayor interacción, control de sus datos y proyección en la web e inserta nuevos modelos de negocio en la red.

1.1.3. E-learning 2.0

Dentro de esta evolución de la red, en el e-learning ha aparecido un concepto que inserta la web 2.0 en este fenómeno: el e-learning 2.0. El término se deriva de todas las tendencias del e-learning en combinación con la web 2.0. Fue establecido por Stephen Downes, un investigador canadiense líder en aprendizaje en línea. [21].

Desde la perspectiva del e-learning 2.0 los sistemas de gestión del proceso de enseñanza-aprendizaje en línea convencionales estaban basados en paquetes instruccionales que eran entregados a los estudiantes a través de internet. El rol del estudiante consistía en aprender de las lecturas y preparar una tarea, las que eran evaluadas por el profesor.

El e-learning 2.0, por otra parte, asume que el conocimiento es socialmente construido. El aprendizaje tiene lugar a través de conversaciones sobre el contenido, la idea es un grupo construyendo cosas entre sí, creando colaborativamente una cultura de artefactos compartidos con diferentes semánticas. Además de las clases virtuales, el aprendizaje social se ha convertido en una parte importante del e-learning 2.0. Las redes sociales son usadas para fomentar comunidades de aprendizaje en línea alrededor de materias tan diversas como preparación para tests y aprendizaje de idiomas.

1.2. Plataformas de Gestión del Aprendizaje

Las plataformas educativas o didácticas son aquellos sistemas que, apoyados en las TICs, representan un entorno de enseñanza-aprendizaje que facilita la adquisición de conocimientos. Pueden ser en modo parcial, es decir de apoyo a la enseñanza tradicional, o completo, denominado aprendizaje en línea, donde las interacciones estudiantes-profesor, estudiante-estudiante son a través de la red y pueden no ocurrir en un espacio o tiempo concurrente.

Capítulo 1 Fundamentación teórica

Una plataforma de gestión del aprendizaje o como también suele denominarse plataforma educativa, es un sistema de formación interactivo para desarrollar programas de enseñanza, que hace uso masivo de los medios electrónicos para llegar a un alumnado generalmente remoto. O sea es una “capacitación no presencial que, a través de plataformas tecnológicas, posibilita y flexibiliza el acceso y el tiempo en el proceso de enseñanza-aprendizaje, adecuándolos a las habilidades, necesidades y disponibilidades de cada docente. [11] Esta definición es asumida en el resto de este documento, añadiéndole la necesidad de gestionar los recursos en diferentes formatos y de todas las actividades que de una u otra manera influyan en el proceso de enseñanza-aprendizaje, pueden adaptarse a necesidades específicas del proceso en una localidad o país.

Son denominadas también LMS por su acrónimo Learning Management Systems (Sistemas de Gestión del Aprendizaje). Las principales características que debe poseer un LMS son: [7]

- Soporte de estándares.
- Administración de usuarios, roles, cursos, profesores o instructores.
- Generación de reportes y trazas de las actividades de los usuarios.
- Calendario de los cursos.
- Administración del aprendizaje de los usuarios.
- Herramientas para la comunicación (chat, fórums).
- Creación, edición y mantenimiento de cursos.
- Materiales educativos en diferentes formatos.
- Camino del Aprendizaje (Learning pathway).

Capítulo 1 Fundamentación teórica

1.2.1. Plataformas educativas a nivel internacional.

Entre las plataformas educativas de código abierto más difundidas se encuentran las que se analizarán a continuación:

- Moodle (Modular Object-Oriented Dynamic Learning Environment). Es un software diseñado para cursos basados en internet y sitios web. Creado para soportar un entorno educativo constructor social, algunas características: gestión de contenidos, administración de cursos, reporte de trazas de la actividad de los usuarios en los cursos, foros de discusión, asignación de tareas, evaluación a través de escalas de calificaciones. No cuenta en su instalación básica con un módulo que gestione una biblioteca. Sin embargo puede encontrarse `vt4moodle` un bloque que provee las funcionalidades básicas para una biblioteca para la plataforma Moodle. Este componente permite crear y mantener un catálogo de material bibliográfico en formato digital (documentos PDF) utilizando el estándar MARC 21. Las funcionalidades que brinda este bloque son las siguientes: [43]
 - Catalogación de ítems.
 - Búsquedas dentro del catálogo por Título, Autor e ISBN.
 - Visualización del registro catalográfico en formato MARC 21.
 - Exportación e Importación de ítems catalográficos en formato MARC 21.
 - Lectura Online de los ítems.
 - Actualización de los registros catalográficos.

Existe además un bloque que gestiona archivos no digitales de bibliotecas: `moodle-library-reserve`. [44]

- **Claroline** es una herramienta educativa basada en cursos que permite crear, administrar y llenar sus cursos a través de la web. Tiene una aproximación distinta a otros LMS: los cursos son divididos en sus elementos componentes y luego publicados en el sitio en áreas separadas (anuncios, ejercicios, chats, recursos). No se encontró ningún componente que gestionara una biblioteca virtual para esta herramienta.
- **Dokeos**. Suite de aprendizaje en línea basada en software libre. Las principales funcionalidades son: autoría, reportes, administración de usuarios, grupos y cursos, portal multisitios y calificaciones. Tiene versiones libre, para la educación, la medicina y profesional. Puede señalarse como desventaja que en su versión libre no ofrece garantía de uso y solo implementa las funciones

Capítulo 1 Fundamentación teórica

de autor, reporte, administración e interacción. [46] Posee un módulo denominado Doogie, una biblioteca digital para este sistema que a partir de los recursos que pueden encontrarse en los cursos son indexados de acuerdo a su contenido o palabras claves y luego pueden ser obtenidos a través de un motor de búsqueda, aunque no es de código abierto. [45]

- **Sakai** es una comunidad que desarrolla un Ambiente de Aprendizaje y Colaboración (del inglés Collaboration and Learning Environment, CLE). Es software libre y plataforma educativa diseñada para la enseñanza, la investigación y la colaboración. Ofrece características comunes a sistemas de gestión de aprendizaje y cursos en línea como distribución de documentos, administración de cursos y exámenes en línea. [47] Entre las características listadas en [48] aparece la de postear, almacenar y organizar los recursos, sin embargo no incluye funcionalidades más abarcadoras de una biblioteca digital.

1.2.2. Plataformas Educativas en Cuba

- **aprenDist**: Plataforma educativa que implementa un campus virtual desarrollada en del Instituto Superior Politécnico José Antonio Echeverría (CUJAE) de Cuba. Es un sistema para implantar e impartir cursos de formación a distancia. Diseñado para un alumnado con determinadas características, que toman diferente el proceso de enseñanza-aprendizaje tradicional (o sea presencial), además de ser una plataforma flexible y ajustable a toda una variedad de cursos, por lo que el usuario de este sistema, puede dominarlo, con sólo tener algún conocimiento de navegación sobre Web. [49]

1.3. Hiperentornos de aprendizaje

Con la intensificación en las últimas décadas del uso de tecnologías para el apoyo a los procesos docente-educativo y de enseñanza-aprendizaje, en Cuba se han desarrollado por parte del MINED (Ministerio de Educación de Cuba) varios software educativos. Particularmente este avance se refleja durante la década del 90 con la creación de varios Centros de Estudios de Software Educativo, con lo cual se inició la producción de estos sistemas. [6]

Sin embargo, no es hasta el año 2001 cuando se toma con carácter curricular extensivo y se crea la colección de software educativo Multisaber [6], para el cual se creó el concepto de hiperentorno de

Capítulo 1 Fundamentación teórica

aprendizaje (también conocido por las siglas HEA), el cual consiste en “una combinación armoniosa de diferentes tipologías de software educativo, basados en tecnología hipermedia” [1]. El concepto hipermedia deviene de la unión de los conceptos de hipertexto y de tecnología multimedia. Un documento hipermedia contiene información a través de la cual se puede navegar. La información puede aparecer en múltiples formatos: textos, gráficos, sonidos, videos, etc.

Por hipertexto se entiende aquel texto que no está diseñado para ser lineal y que puede verse en una computadora u otro dispositivo electrónico. Contiene vínculos o referencias a otros documentos que pueden ser accedidos inmediatamente a través de un click del mouse o al presionar una tecla del teclado. Es el concepto subyacente en la estructura de la red de redes: la World Wide Web. El término fue propuesto por Ted Nelson en 1965 [3] [4] y su significado literal es más allá del texto.

A partir del concepto de hiperentorno de aprendizaje se han desarrollado en Cuba diversas colecciones de software educativo entre las que se incluye El Navegante y la colección Futuro. Estos fueron desarrollados inicialmente para su distribución en CD-ROM, pero la tendencia actual de uso de espacios virtuales en línea por los conocidos beneficios que brinda para muchas esferas de la sociedad y entre ellas la educación, ha visto su evolución al ciberespacio. Las principales características que presentan los hiperentornos de aprendizaje para web (Cuba como escenario) son: [6]

- Creados y montados a través de la herramienta de autor SAdHEA-Web.
- Productos multiplataforma, por lo que se ejecutan desde diversos sistemas operativos.
- Desarrollados con herramientas de software libre.
- Materializa de manera plena lo mejor del concepto de HEA generalizado en las colecciones de software educativos del MINED.
- Incorpora conceptos y herramientas propios de la Web 2.0 (Foros, Chat, etc.)
- Pueden ser instalados de acuerdo con las posibilidades tecnológicas de la institución: en una red interna, externa o en una PC local.

Los HEA para la web están compuestos por un grupo de módulos y diversos servicios informáticos en los que se reflejan los principales elementos del contenido. Los principales son: Ejercicios, Juegos, Profesor, Resultados y Mediateca o Biblioteca Virtual.

Capítulo 1 Fundamentación teórica

1.4. Concepción constructivista del aprendizaje

En el diseño de los modelos educativos de la actualidad, cada vez más las organizaciones e instituciones están adoptando el enfoque constructivista como síntesis recogedora de los mejores postulados de las escuelas pedagógicas. Representa una alternativa cualitativamente superior a los enfoques tradicionales, tecnocráticos y reproducionistas de la educación, capaz de inspirar estrategias docentes innovadoras y la formación de estudiantes activos, autónomos, que posean la principal habilidad de aprender a aprender. [14]

El constructivismo tiene esencialmente un enfoque epistemológico, que sostiene que todo conocimiento es construido como resultado de procesos cognitivos dentro de la mente humana. Se opone a la idea de que el conocimiento sea una representación fuera de la mente del que aprende.

El constructivismo es una “visión global del conocimiento humano como un proceso de construcción y reconstrucción cognoscitiva llevada a cabo por los individuos que tratan de entender los procesos, objetos y fenómenos del mundo que los rodea, sobre la base de lo que ellos conocen”. [15]

Entre los principales teóricos del constructivismo se encuentra Jean Piaget (1896-1980), que fundamenta que el desarrollo cognoscitivo es un proceso adaptativo (de asimilación-acomodación) [15]. Por otra parte un discípulo suyo Seymour Papert partiendo de las nociones de Piaget da un enfoque pedagógico-didáctico y define el construccionismo como:

“Tomamos de las teorías constructivistas de la psicología el enfoque de que el aprendizaje es mucho más una reconstrucción que una transmisión de conocimientos. A continuación extendemos la idea de materiales manipulables a la idea de que el aprendizaje es más eficaz cuando es parte de una actividad que el sujeto experimenta como la construcción de un producto significativo” [16].

Muchos sistemas de aprendizaje que se desarrollan en la actualidad se basan en esta teoría pedagógica, entre ellos de los más usados como Moodle, Claroline, así como Blackboard, WebCT y Desire2Learn [17]. También se identifican herramientas de la web 2.0 como herramientas que soportan el aprendizaje desde una perspectiva constructivista: mashups, blogs, marcadores sociales (social bookmarking), wikis, VoiceThread. [17]

Capítulo 1 Fundamentación teórica

1.5. Metadatos

Una definición primaria de metadatos es la de “datos acerca de los datos” o información acerca de la información.

Un metadato no es más que un dato estructurado que describe las características de un recurso. El término meta proviene del griego que denota una naturaleza de orden mayor o de carácter fundamental. Un registro de metadatos consiste en un número de elementos predefinidos que representan atributos específicos de un recurso donde cada elemento puede tener uno o más valores. Las características usualmente observadas en un esquema de metadatos son [8]:

- Un número finito de elementos.
- Cada elemento posee un nombre.
- Cada elemento tiene un significado particular.

Aspectos claves de los elementos que dan acceso a recursos publicados incluyen el originador del trabajo, el título, lugar y fecha de publicación y el área del conocimiento a la que se aplica el recurso. Sin embargo se proveen otros metadatos para asistir en la búsqueda y localización de información. Entre los esquemas de metadatos más populares se encuentran [8]:

- Dublin Core
- AACR2 (Anglo-American Cataloging Rules)
- GILS (Government Information Locator Service)
- EAD (Encoded Archives Description)
- IMS (IMS Global Learning Consortium)
- AGLS (Australian Government Locator Service)

Los metadatos son datos asociados con objetos, que delegan a sus usuarios potenciales conocimiento avanzado completo de su existencia y características [8]. Son un método sistemático de describir un recurso y en consecuencia mejora el acceso al mismo. Provee el vínculo necesario entre el proveedor del recurso y el usuario final [8]. Aunque el objetivo primario de los metadatos es el descubrimiento de los recursos, tienen otras funciones como administración y control de los recursos, administración del contenido, gestión de derechos de autor y preservación.

Los metadatos constituyen un elemento clave para los recursos de la web, entre ellos por supuesto los recursos para el aprendizaje y la capacitación, debido a que facilitan la búsqueda, acceso y catalogación del contenido.

Capítulo 1 Fundamentación teórica

1.6. Objeto de aprendizaje.

El Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, *por sus siglas en inglés*), define objeto de aprendizaje (en lo adelante OA) como una entidad, digital o no, que puede ser empleada en el aprendizaje, la educación y la capacitación soportada por tecnología. David Wiley define objeto de aprendizaje como recurso digital que puede ser reusado para soportar el aprendizaje [10]. Otra definición de OA es un paquete con uno o más recursos que tienen utilidad educacional.

Un Objeto de Aprendizaje es una estructura (distribución, organización) autónoma que contiene un objetivo general, objetivos específicos, una actividad de aprendizaje, un metadato (estructura de información externa) y por ende, mecanismos de evaluación y ponderación, el cual puede ser desarrollado con elementos multimedia con el fin de posibilitar su reutilización, interoperabilidad, accesibilidad y duración en el tiempo. Un OA puede estar constituido al menos con los siguientes componentes: Contenido(s), actividad (es) de aprendizaje y un contexto. Un OA puede ser montado (incorporado, subido, instalado, configurado) en una plataforma de Gestión de Aprendizaje o LMS (Learning Management System). Es el producto de un diseño instruccional donde convergen procesos de educomunicación y objetivos instruccionales. [12]

El tamaño de un OA depende del dominio de aplicación y de las competencias planteadas en el programa educativo. [12]

Los OA están basados en Programación Orientada a Objetos (POO), definidos en base a estándares basados en XML (eXtensible Markup Language), metadatos y empaquetamiento para ser catalogados, indexados y localizados [12].

Los OA poseen una dualidad didáctica y técnica. Como instrumento del aprendizaje apoyan el proceso de enseñanza-aprendizaje, promueven la construcción y distribución del conocimiento e impulsan la apropiación de tecnología educativa.

Como objetos están basados en etiquetas, el uso de estándares y buscan alcanzar la interoperabilidad y reusabilidad y requieren de sistemas para crear, editar y administrar OA.

Capítulo 1 Fundamentación teórica

Ejemplos de OA son un simulador, un juego, una unidad de ejercicio con contenido instruccional, diferenciándose de otros objetos mediáticos, como imágenes o videos, independientemente de que estos puedan cumplir una función educativa en un contexto determinado.

1.6.1. Estándares para objetos de aprendizaje

Debido a la necesidad de interoperabilidad y reusabilidad de los OA algunas organizaciones se dieron a la tarea de desarrollar estándares, especificaciones y modelos de referencia que cubran estas necesidades. Son diversas las iniciativas, entre las más destacadas se encuentran: IEEE, SCORM e IMS Global Learning Consortium [18]. Estas especificaciones tienen su base en los metadatos y en el lenguaje XML.

El estándar de metadatos LOM (Learning Object Metadata), desarrollado por la IEEE es el primer esquema de metadatos acreditado para tecnología de aprendizaje. [12] El objetivo de este estándar es simplificar las operaciones de búsqueda, gestión e intercambio de OA en la web.

SCORM (Sharable Content Object Reference Model) es un modelo de software que define la relación intrínseca entre los recursos, modelos de datos y protocolos para que los objetos de los cursos sean compatibles entre sistemas con el mismo modelo [18].

El IMS (Instructional Management System) Global Learning Consortium desarrolla y promueve especificaciones abiertas para facilitar la distribución de actividades académicas en línea. Entre sus especificaciones se encuentra el IMS Learning Resource Metadata Especification, que crea una representación uniforme de describir los recursos de aprendizaje para que puedan ser encontrados de manera más fácil. [18]

1.7. Bibliotecas digitales.

Las bibliotecas digitales son colecciones organizadas de información digitalizada. Combinan estructura y recolección de información, lo que las bibliotecas han hecho tradicionalmente, con la representación digital que las TICs han hecho posible. [22]

Sin pretender dar una definición formal de biblioteca digital, Arms [24] expone que es una colección gestionada de información con servicios asociados, donde la información es almacenada en formato

Capítulo 1 Fundamentación teórica

digital y accesible a través de una red. El aspecto clave de esta definición es que la información es gestionada.

Las bibliotecas digitales contienen diversas colecciones de información que tendrán diferentes usuarios. El talante unificador es que la información es organizada en computadoras y está disponible en una red, con procedimientos para seleccionar el material en las colecciones, organizarlo, hacerlo accesible a los usuarios y archivarlo. [24]

DELOS (Digital Library Reference Model) establece el término biblioteca digital como una organización, que puede ser virtual, que exhaustivamente colecciona, gestiona y preserva a largo plazo contenido digital y ofrece a sus comunidades de usuarios funcionalidad especializada de ese contenido de calidad medible y de acuerdo con políticas codificadas. [25]

Debido a que no es un campo totalmente establecido, otras definiciones pueden aparecer: conjunto de recursos electrónicos y sus capacidades técnicas asociadas para la creación, búsqueda y uso de información. Constituyen una extensión y mejora del almacenamiento de información y de los sistemas de recuperación de datos digitales en cualquier medio. El contenido de las bibliotecas digitales incluye no sólo los datos, sino también los metadatos. [22]

La Federación de Bibliotecas Digitales (DLF, Digital Library Federation) ofrece la siguiente definición:

Organizaciones que proveen los recursos, incluyendo el personal especializado para seleccionar, estructurar, ofrecer acceso intelectual a, interpretar, distribuir, preservar la integridad de y asegurar la preservación en el tiempo de colecciones de trabajos digitales de manera que puedan ser leídos y económicamente accesibles para su uso por una comunidad definida o por un conjunto de comunidades. [26]

Estas definiciones muestran que el concepto de biblioteca digital tiene varios significados a los que se puede recurrir en diferentes contextos. Sin embargo y para los objetivos de este trabajo los elementos de estas definiciones sugieren una serie de atributos que le dan coherencia al concepto de biblioteca digital.

Estos atributos incluyen funciones de colección, organización, preservación, acceso y economía. [26]

Capítulo 1 Fundamentación teórica

Los términos: Biblioteca Virtual (Virtual library), Biblioteca Electrónica (Electronic library), Biblioteca en Línea (Online library), Biblioteca Digital (Digital library), Biblioteca sin Fronteras (Library without walls), Biblioteca en Red (Networked library) y Bibliotecas-E (E-library), son todos utilizados para designar una biblioteca que tiene como soporte a las TICs. [31]

1.7.1. Algunas bibliotecas digitales. Características.

El desarrollo de las bibliotecas virtuales viene dado fundamentalmente para y por instituciones académicas, librerías y museos. Algunos ejemplos de bibliotecas virtuales que dan libre acceso a sus contenidos son:

Biblioteca Digital Mundial (World Digital Library, www.wdl.org/es). Contiene importantes materiales de culturas de todo el mundo. Entre sus objetivos se encuentra promover el entendimiento internacional e intercultural. [29]

Biblioteca Digital Berkeley (Berkeley Digital Library SUNsite, Universidad de Berkeley, sunsite.berkeley.edu) Iniciativa de la Universidad de Berkeley y de la empresa de informática Sun Microsystems, constituye una de las mejores bibliotecas digitales del mundo, con varias colecciones de interés. [30]

Internet Classics (classics.mit.edu) Obras clásicas de autores grecolatinos, de China y de Persia en versión inglesa. Este sitio está elaborado por el MIT (Instituto de Tecnología de Massachussets) y contiene enlaces de interés y comentarios de los lectores.

Entre los sistemas comerciales de automatización de bibliotecas se encuentran [29]:

- Digital Library Toolkit de Sun Microsystems (Sitio web oficial: www.sun.com/edu/libraries).
- ENCompass, de Endeavor Systems (Disponible en: www.endinfosys.com).
- Digitool de Ex Libris (Disponible en: www.exlibris.co.il o www.exlibris-usa.com).
- TEAMSTM de Artesia Technologies (Sitio web artesia.com, es el software usado por la Universidad de Stanford como base para su biblioteca digital)

Las características comunes que ofrecen estos sistemas incluyen opciones de búsqueda y localización, gestión de derechos de autor y licencias, uso de sintaxis estándares como XML (Extensible Markup

Capítulo 1 Fundamentación teórica

Language), SGML (Standard Generalized Markup Language) y RDF (Resource Description Framework), además de funciones para compartir, manipular y distribuir la información como recurso. Estas están destinadas al público en general y aunque también pueden ser útiles para intereses académicos y educativos aunque no están diseñadas con ese objetivo.

1.7.2. Bibliotecas Virtuales Cubanas

En Cuba existen varios sitios que hospedan bibliotecas digitales fundamentalmente de universidades e instituciones, se caracterizan por ofrecer sus materiales de interés para la comunidad a la que sirven y algunas son principalmente un apoyo a las tradicionales, puede consultarse el material de la biblioteca que no está digitalizado, así como conocer servicios que ofrece y solicitud de material. Estas no ofrecen ninguna funcionalidad especializada sobre los recursos que potencie la filosofía social de la web actual, se limitan al tradicional proceso de búsqueda y acceso.

Biblioteca Virtual de las Ciencias en Cuba. www.bibliociencias.cu

Biblioteca Digital de Cuba. Biblioteca virtual de la Biblioteca Nacional José Martí. bdigital.bnjm.cu/

Biblioteca Virtual de Salud de Cuba. Ofrece acceso a las revistas médicas cubanas, libros de autores nacionales, catálogos, noticias y enlaces de interés. www.bvscuba.sld.cu. La más importante experiencia cubana en materia de bibliotecas virtuales. [31]

En la Universidad de las Ciencias Informáticas (UCI) está disponible a través de la web una biblioteca digital (BiUCI) que funciona como sistema de apoyo a los servicios de una biblioteca tradicional como reservación de documentos y libros a través de la web y la consulta de la disponibilidad de los materiales. Ofrece además opciones de acceso a sus materiales digitalizados y un servicio de búsqueda. (Sitio accesible sólo desde la UCI, biblioteca.uci.cu).

1.7.3. Bibliotecas digitales y e-learning

El crecimiento en el e-learning y con ello la necesidad de localización y acceso en un universo de información digital ha planteado la premisa de integración entre sistemas con soluciones e-learning y aquellos que implementan servicios de las bibliotecas digitales.

Capítulo 1 Fundamentación teórica

Las bibliotecas digitales tienen el potencial para cambiar significativamente elementos fundamentales de la educación con un alto impacto en el proceso de enseñanza-aprendizaje. Por tanto no pueden ser vistas como una mera colección de materiales digitales y sus herramientas de gestión asociadas, sino como un entorno que reúne las colecciones, servicios y las personas que dan soporte al ciclo completo de creación, diseminación, uso y preservación de los datos, la información y el conocimiento. [27]

Entre los objetivos intermedios del papel que pueden desempeñar las bibliotecas digitales para dar soporte al e-learning se encuentran: [27]

- Mejorar el desempeño de los estudiantes.
- Incrementar la cantidad, calidad y exhaustividad de los recursos educacionales basados en internet.
- Hacer estos recursos fáciles de descubrir y recuperar para los estudiantes, padres y educadores.
- Asegurarse de la preservación de estos recursos en el tiempo.

Las bibliotecas digitales deben proveer además servicios para los autores e instructores o profesores como anotación, evaluación y capacidades de gestionar derechos de autor y diseminación selectiva de información. Los estudiantes, profesores y las instituciones podrán participar de la información dentro de los propios sistemas que gestionan el aprendizaje en línea de modo que sean capaces de acceder, adaptar, evaluar y referenciar apropiadamente material didáctico.

Los LMS permiten a sus usuarios interactuar con materiales digitales didácticos, matricularse en cursos, tomar exámenes y comunicarse con otros usuarios (tanto estudiantes como profesores). Consisten en varios componentes representando diferentes servicios que son usados en un entorno e-learning como la presentación y administración de cursos y asignación y gestión de tareas, ejercicios y evaluaciones. Un elemento importante dentro de estos es el contenido que es gestionado en los mismos, que para los objetivos de visualización, acceso y apropiación de conocimientos, que es en definitiva la función principal de todo sistema e-learning, es de importancia su conservación, localización y acceso, todas estas características naturales de las bibliotecas digitales.

Sin embargo integración entre sistemas e-learning y bibliotecas digitales no es una esfera totalmente establecida en el mundo académico actual debido a los retos técnicos que plantea. Algunos de estos retos son: [28]

- Cómo controlar el acceso a contenido protegido por las leyes de derecho de autor (copyright).

Capítulo 1 Fundamentación teórica

- Creación de los metadatos adecuados que describan más explícitamente un recurso.
- Preservación digital. Asegurar que medios digitales y contenido sea interpretable en un futuro indefinido.
- Referenciar contenidos a través de URLs (OpenURL database, servidores de enlaces, capaces de resolver enlaces de metadatos a recursos en línea o físicos).

La integración entre LMS y bibliotecas digitales está actualmente desarrollada sobre todo en los campus virtuales de las universidades que intentan integrar sus sistemas e-learning, bibliotecas y otros sistemas como de gestión de cursos.

Algunos ejemplos de bibliotecas virtuales en entornos educativos son las de los campus virtuales de universidades y escuelas.

Minnesota Digital Library (www.mndigital.org).

McLennan Community College Library. Biblioteca acoplada con los cursos de Blackboard, un LMS comercial. (www.mclennan.edu/library/index.html).

Lake Forest High School ²

De estas destacar que no son bibliotecas en el sentido tradicional, es decir no se limitan a la imitación digital de las bibliotecas tradicionales, sino que además de los servicios de búsqueda y visualización incorporan servicios digitales como:

- Referencia virtual (consulta a expertos bibliotecarios de la institución)
- Recomendación de libros (automática y personalizada)
- Revisión y anotación de materiales digitales

Además pueden identificarse características de las interfaces de recuperación de información, dado que eso son las bibliotecas digitales, como útiles para los objetivos de esta investigación. [52]:

- Puntos de partida
- Formulación de consultas
- Visualización de resultados, puede ser a través de:
 - Visualización simple o detallada
 - Resaltado de los términos de búsqueda
 - Casilla de selección

² <http://lfhs.lib.overdrive.com/50E8F748-7E07-4D64-8EEC-A306FD7B1858/10/404/en/Default.htm>

Capítulo 1 Fundamentación teórica

- Búsquedas similares
- Historia de la búsqueda (grabado de la estrategia de búsqueda para su uso posterior)
- Navegación entre registros del sistema
- Acceso a texto completo
- Acciones posteriores
 - Alertas de nuevos documentos que citen al indicado
 - Alertas a partir de la misma estrategia de búsqueda
 - Marcar documentos
 - Imprimir o grabar resultados
 - Exportar citas
 - Suscribirse a alertas
 - Enviar por email
 - Escribir al autor
 - Otras

1.8. Repositorio

Asociado con el concepto de bibliotecas digitales se encuentra el de repositorios. Un repositorio es cualquier sistema por computadora que tiene como función primordial la de almacenar material digital para su uso en una biblioteca. Los repositorios son el estante de libros de las bibliotecas. La mayoría de los repositorios son sistemas que almacenan información en un sistema de archivos o en una base de datos y es presentada con una interfaz bien definida. [23]

Los repositorios recopilan tanto los datos como los metadatos que los describen. Los repositorios son parte una parte importante de las bibliotecas digitales al ser el almacén de sus recursos.

1.9. Plataforma de Gestión del Aprendizaje ZERA

En el departamento de herramientas educativas de la Universidad de Ciencias Informáticas se desarrolla ZERA, una Plataforma de Gestión de Aprendizaje, que inicialmente se basó en el software educativo colección Futuro desarrollado por el Ministerio de Educación de Cuba para la enseñanza media, actualmente es más una adaptación para la web de los hiperentornos de dicha colección, suponiendo

Capítulo 1 Fundamentación teórica

tareas no sólo de manejo de estos contenidos sino de administración y gestión de todo el proceso de enseñanza-aprendizaje virtual.

La colección Futuro tiene una concepción curricular pues cubre los programas de estudio del nivel medio superior y sus análogos en la enseñanza técnica superior. Cuenta con los módulos definidos para los HEA: Temas, Ejercicios, Biblioteca, Juegos, Resultados y Profesor. El módulo Biblioteca o Mediateca tiene como objetivo ser un repositorio de recursos multimedia en el que aparecen galerías de diversos tipos de recursos (imágenes, diaporamas, videos, sonidos, etc.), un glosario de términos afines a la asignatura, información de interés, elementos de conectividad y efemérides. [42]

La plataforma ZERA está estructurada por sistemas y subsistemas. Entre los sistemas se encuentra el Sistema de materias o hiperentornos que posee varios módulos incluido Biblioteca, módulo de los objetos o elementos que forman parte del contenido como: recursos estructurales (Saber más, Sabías que, Medita un instante, Recuerda que, aplicación práctica, Investiga y aprende, Historia), recursos multimedia (imágenes, videos, sonido, animaciones, diaporama) y recursos interactivos (Secuencia interactiva, Pregunta interactiva, Simuladores, Enfrenta retos).

En la captura inicial de requisitos se determinaron los siguientes requisitos:

- Consultar los recursos en dependencia del índice de la materia donde fue creado el recurso.
- Filtrar los elementos mostrados de acuerdo a diversos criterios.

1.10. Metodologías de desarrollo de software, herramientas y lenguajes de modelado.

1.10.1. Metodologías de desarrollo de software. Metodologías ágiles y pesadas.

Una metodología de desarrollo de software es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo de un software. Consiste en una filosofía de desarrollo de software y en el uso de herramientas, modelos y métodos para asistir al proceso de desarrollo del software. Existen dos enfoques fundamentales para las metodologías: las pesadas o tradicionales y las ágiles.

Las características deseables que debe tener una metodología son: [38]

- Existencia de reglas predefinidas
- Cobertura total del ciclo de desarrollo

Capítulo 1 Fundamentación teórica

- Verificaciones intermedias
- Planificación y control
- Comunicación efectiva
- Utilización sobre un número amplio de proyectos
- Fácil formación
- Soportada por herramientas CASE
- Actividades que mejoren el proceso de desarrollo
- Soporte al mantenimiento
- Soporte de la reutilización de software

1.10.2. Metodologías ágiles

Con el objetivo de esbozar los valores y principios que deberían seguir los equipos para desarrollar software rápidamente y respondiendo a los cambios que puedan surgir en los proyectos se creó el término ágil en una reunión de expertos en metodologías de software en Utah, Estados Unidos en el año 2001.

Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas.

Así quedó constituida The Agile Alliance (Alianza Ágil), organización sin ánimo de lucro, dedicada a promover los conceptos relacionados con este enfoque para desarrollar software. El punto de partida fue el Manifiesto Ágil, que resume la filosofía ágil. [32]

Los valores que quedaron recogidos en el Manifiesto son: [33]

- Las interacciones entre el equipo de desarrollo sobre el proceso y las herramientas.
- Desarrollar software que funcione más que una buena documentación.
- La colaboración con el cliente más que la negociación de un contrato.
- Responder a los cambios más que seguir estrictamente un plan.

Capítulo 1 Fundamentación teórica

Al analizar una metodología ágil se hace necesario resaltar las diferencias con las denominadas metodologías tradicionales o pesadas ya que ambos enfoques no se limitan al proceso en sí, sino también al contexto de equipo y organización. [33]

Tabla 1. Diferencias entre metodologías ágiles y no ágiles.

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código.	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo.
Especialmente preparadas para cambios durante el proyecto.	Cierta resistencia a los cambios.
Impuestas internamente (por el equipo de desarrollo).	Impuestas externamente.
Proceso menos controlado con pocos principios.	Proceso mucho más controlado, con numerosas políticas/normas.
No existe contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio.	Grupos grandes y posiblemente distribuidos.
Pocos artefactos.	Más artefactos
Pocos roles.	Más roles.
Menos énfasis en la arquitectura del software.	La arquitectura del software es esencial y se expresa mediante modelos.

Entre los principales métodos ágiles tenemos Scrum, Iconix, Cristal Methods y AUP (Agile Unified Process, Proceso Unificado Ágil). De las más destacadas entre los procesos ágiles se encuentra XP (eXtreme Programming).

Programación Extrema XP (eXtreme Programming)

Capítulo 1 Fundamentación teórica

XP es una metodología ágil que propone como clave para el éxito las relaciones interpersonales, promoviendo el trabajo en equipo, el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. Está definida específicamente para proyectos con requisitos imprecisos y muy cambiantes y donde existe un alto riesgo técnico. [33] Dirigida a proyectos de corto plazo y tiene la particularidad de tener al usuario final como parte del equipo. [34] Sus principales características son:

- Pruebas unitarias: pruebas realizadas a los principales procesos, de tal manera que adelantándose en algo hacia el futuro, se puedan hacer pruebas de las fallas que pudieran ocurrir.
- Refabricación: se basa en la reutilización de código para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- Programación en pares: propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento.

¿Qué propone XP? [34]

- Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- El manejo del cambio se convierte en parte sustantiva del proceso.
- El costo del cambio no depende de la fase o etapa.
- No introduce funcionalidades antes que sean necesarias.
- El cliente o el usuario se convierte en miembro del equipo.

El ciclo de vida ideal de XP consta de seis fases:

- Exploración
- Planificación de la entrega
- Iteraciones
- Producción
- Mantenimiento
- Muerte del proyecto

Capítulo 1 Fundamentación teórica

1.10.3. Metodologías pesadas o tradicionales

Las metodologías tradicionales se enfocan en documentación y procesos (plantillas, técnicas de administración, revisiones, etc.). Son aquellas con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado. [40]

Se centran en el control del proceso mediante la definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada.

RUP (Proceso Unificado de Desarrollo, Rational Unified Process)

RUP es una metodología preparada para grandes y complejos proyectos, orientada a objetos y que utiliza UML (Unified Modeling Language) como lenguaje de modelado. Provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización. Su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales respetando cronograma y presupuesto.

Los principales elementos de RUP son: [35]

Trabajadores (“quién”): Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto.

Actividades (“cómo”): Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.

Artefactos (“qué”): Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.

Flujo de actividades (“cuándo”): Secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo, los cuales son:

Capítulo 1 Fundamentación teórica

Modelación del negocio: Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.

Requerimientos: Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

Análisis y diseño: Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.

Implementación: Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.

Prueba (Testeo): Busca los defectos a lo largo del ciclo de vida.

Instalación: Produce *release* del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.

Administración del proyecto: Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.

Administración de configuración y cambios: Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.

Ambiente: Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

Las fases de RUP son:

Conceptualización (Concepción o Inicio): Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.

Capítulo 1 Fundamentación teórica

Elaboración: Se define la arquitectura del sistema. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.

Construcción: Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene uno o varios *release* del producto que han pasado las pruebas. Se ponen estos *release* a consideración de un subconjunto de usuarios.

Transición: El *release* ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

En el Anexo 6 se representa a RUP en dos dimensiones donde se relacionan las fases y los flujos de trabajo.

El ciclo de vida de RUP se caracteriza por:

Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.

Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura.

Iterativo e Incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros.

1.10.4. Selección de la metodología

Existen varios criterios para seleccionar una metodología [41]: que se ajuste a los objetivos, la realización de validaciones en el ciclo de desarrollo, así como la posibilidad de emplearse en un entorno amplio de proyectos de software y soportar la eventual evolución del sistema. Se escogieron para este análisis las metodologías RUP y XP por ser las principales representantes de los dos enfoques en el desarrollo de metodologías.

Capítulo 1 Fundamentación teórica

En la metodología XP el desarrollo va acompañado con la opinión del cliente que es un integrante más del equipo de desarrollo. XP además prácticamente no genera documentación fuera del código fuente, lo que dificulta la representación de los sistemas del negocio o la utilización de experiencias anteriores y está pensado para equipos de desarrollo relativamente pequeños y para proyectos con requisitos muy cambiantes. En el caso de RUP intenta reducir la complejidad a través de la planeación intensiva, produce pequeños resultados incrementales, está basado en roles, puede adaptarse tanto a proyectos grandes como a pequeños y no necesita de la interacción física del cliente para entregar un producto con calidad, costes y en el tiempo requerido. Debido a que en este caso el cliente no forma parte del equipo de desarrollo e interesa por tanto generar la mayor documentación posible y es un proyecto basado en roles, se escoge como metodología a utilizar RUP.

1.11. Lenguaje de Modelado.

UML (Unified Modeling Language)

UML es un lenguaje de modelado visual de propósito general que se usa para visualizar, especificar, construir y documentar artefactos de un sistema de software. Está diseñado para ser usado con todos los métodos y ciclos de desarrollo, dominios de aplicación y en procesos orientados a objetos. [36]

Un modelo UML está compuesto por tres clases de bloques de construcción: [37]

- Elementos. Los elementos son abstracciones de cosas reales o ficticias. (Ej. Objetos, acciones).
- Relaciones. Relacionan elementos entre sí.
- Diagramas. Colecciones de elementos con sus relaciones.

1.12. Herramientas CASE.

Las herramientas de la ingeniería del software suministran un soporte automático o semiautomático para los métodos. Hoy existen herramientas para soportar cada uno de los métodos mencionados. Cuando se integran las herramientas de forma que la información creada por una herramienta pueda ser usada por otra, se establece un sistema para el soporte de desarrollo del software, llamado ingeniería del software asistida por computadora (en inglés, CASE, *Computer Aided Software Engineering*).

1.12.1. Visual Paradigm

Es una herramienta de diseño que soporta todos los diagramas UML y diagrama entidad – relación. Visual Paradigm para UML ofrece amplias características de modelado de casos de uso incluyendo la función

Capítulo 1 Fundamentación teórica

completa de UML hasta la versión 2.2, diagrama de casos de uso, flujos de evento editor, caso de uso/red de actor y la generación de un diagrama de actividad. Los analistas del sistema pueden estimar las consecuencias de los cambios en los diagramas de análisis de impacto, tales como la matriz y el diagrama de análisis. Esta herramienta permite además la ingeniería inversa de código Java, .NET, PHP5, Python, C. Características:

- Modelado UML
- Modelado de Casos de Uso
- Generación de reportes
- Análisis de Impacto

1.12.2. Rational Rose

Rational Rose es la herramienta CASE que comercializan los desarrolladores de UML (Booch, Rumbaugh y Jacobson) a través de IBM y soporta de forma completa la especificación del UML. Es una herramienta que propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar vistas creando un modelo completo que representa el dominio del problema y el sistema de software. Dentro de sus características está el desarrollo iterativo pues utiliza un proceso de desarrollo iterativo controlado, donde el desarrollo se lleva a cabo en una secuencia de iteraciones.

Genera código en distintos lenguajes de programación a partir de un diseño en UML y proporciona mecanismos para realizar ingeniería inversa. [38]

1.12.3. Selección de herramienta CASE a utilizar

Se escoge como herramienta de modelado el Visual Paradigm para UML 6.4 debido fundamentalmente a que es multiplataforma (incluyendo Linux) y soporta modelado tanto de todas las fases como del modelo de datos. Soporta además ingeniería directa e inversa y todos los diagramas UML y entidad-relación y el ciclo de vida del desarrollo de software completo: análisis, diseño, implementación, pruebas y despliegue.

Conclusiones

En este capítulo se realizó un estudio de las tendencias y estado del arte relacionado con los conceptos asociados al dominio del problema de investigación. Se describieron características de los sistemas

Capítulo 1 Fundamentación teórica

similares determinando que las tendencias actuales en la construcción de estos sistemas siguen el camino de desarrollo y evolución de los servicios en la red a una mayor participación de los usuarios en la creación de contenidos. Así como que los sistemas de administración del aprendizaje necesitan para ofrecer un mejor servicio de las funciones de conservación, localización y acceso que proveen las bibliotecas digitales para los recursos que son generados por los cursos, de manera que estudiantes, profesores y usuarios en general sean capaces de personalizar, acceder, evaluar y referenciar recursos de aprendizaje. Se analizaron las herramientas que serán usadas en la confección de la solución propuesta. Se escogió como metodología de desarrollo RUP, lenguaje de modelado UML y herramienta CASE Visual Paradigm para UML 6.4. Se argumentó el objeto de estudio con el objetivo de lograr un mejor entendimiento del mismo.

Capítulo 2 Descripción de la propuesta

2. CAPÍTULO 2: DESCRIPCIÓN DE LA PROPUESTA

Introducción

En este capítulo se presenta la propuesta de solución que incluye el análisis del sistema, la captura de requisitos y la elaboración del modelo de casos de uso del sistema. Se incluyen además las descripciones textuales de los casos de uso y se presentan los conceptos fundamentales del problema en un modelo de dominio.

2.1. Modelo de Dominio

En el desarrollo de las actividades del flujo de trabajo Modelamiento del Negocio no se identificaron los procesos de negocio de manera clara, por lo que se realiza un modelo de dominio, que no es más que un modelo de objetos que captura las clases más importantes del sistema. Se procederá a mostrar el diagrama del dominio y a explicar los conceptos que forman parte del mismo. Lo anterior con el objetivo de establecer un vocabulario y comprensión del contexto del sistema común a usuarios, desarrolladores y clientes. En la figura 1 se muestra el diagrama del dominio.

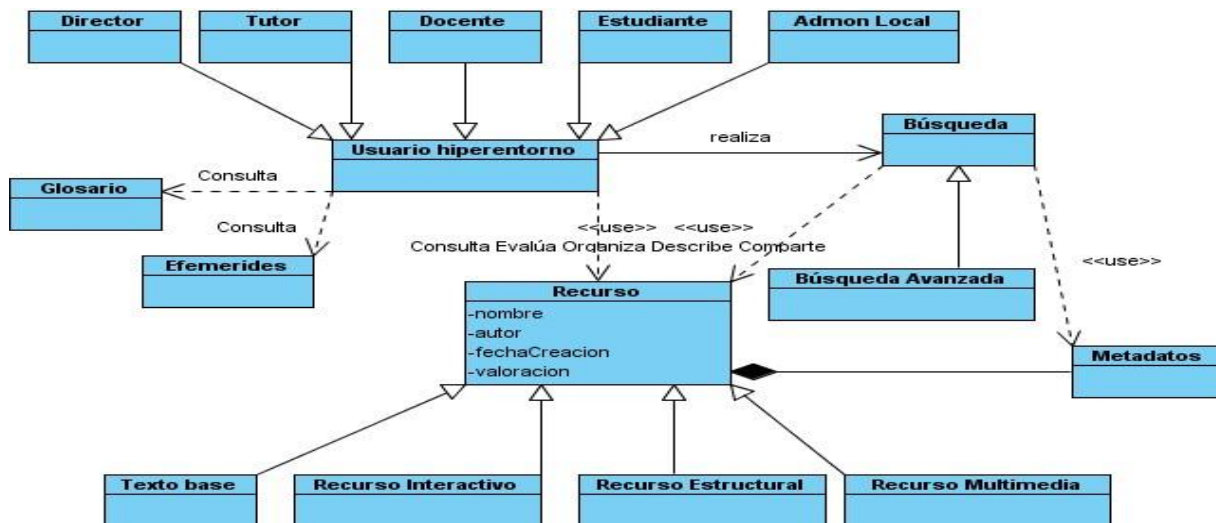


Figura 1. Diagrama de clases del dominio.

Descripción de las clases del diagrama de clases del dominio.

Capítulo 2 Descripción de la propuesta

Usuario hiperentorno: Usuario que generaliza a todos los actores por sus acciones comunes.

Director: Persona encargada de dirigir una escuela. Puede revisar y analizar la información a nivel de estudiante, curso, docente y escuela.

Tutor: Es la persona que representa legalmente a un estudiante. Un tutor puede tutorar a varios estudiantes.

Docente: Persona que puede impartir una o varias materias en uno o muchos grupos. Puede asignar actividades al grupo o individualmente, hace el seguimiento a las actividades de todos los estudiantes que estén a su cargo y evalúa a los estudiantes de su (s) grupo (s) en los contenidos que imparte.

Estudiante: Es la persona que recibe contenidos educativos en uno o varios grupos de una escuela.

Administrador Local: Realiza las tareas de administración en la plataforma instalada en la escuela.

Recursos: Objetos o elementos que forman parte del contenido. Se encuentran clasificados por tipo.

Texto base: Constituye la fuente de información fundamental en la que se codifica la parte de la cultura que se propone en calidad de contenido y cuyo alcance debe estar en correspondencia con las exigencias mínimas del currículo y estar escrito en un lenguaje acorde con el público al que va dirigido el producto.

Recursos Multimedia: Recursos que muestran el contenido en formato multimedia o sus combinaciones: Imágenes, Sonidos, Animaciones, Videos, Diaporamas.

Recursos Interactivos: Recursos que materializan la interactividad hombre-máquina, donde existe un emisor, un receptor y una retroalimentación continua. Pueden ser: Secuencias Interactivas, Pregunta interactiva, Animación interactiva, Simuladores, Enfrenta Retos.

Recursos Estructurales: Recursos con el principal objetivo de contribuir a la estructuración didáctica de la información, ofrecen una exposición diferenciada del contenido del texto base. Tienen una marcada intencionalidad pedagógica. Estos son: Saber más, Sabías que, Recuerda que, Medita un instante, Aplicación Práctica, Investiga y Aprende, Historia.

Capítulo 2 Descripción de la propuesta

Metadatos: Datos que describen los recursos, ejemplo: autor, fecha de creación, derechos de copia, etc. El estándar de facto para la descripción de los metadatos de objetos de aprendizaje y que será usado para la descripción de los recursos en la plataforma ZERA, es LOM. Los elementos de los datos según este estándar son agrupados en nueve categorías: General, Ciclo de vida, Meta-metadatos, Técnica, Educativa, Derechos de copia, Anotación, Relación y Clasificación. Cada una de estas categorías incluye elementos de dato como identificador, título, idioma y descripción para la categoría General.

Búsqueda: Proceso por el cual se localizan archivos o recursos de acuerdo con ciertos criterios predefinidos.

Búsqueda avanzada: Proceso por el cual se localizan archivos o recursos de acuerdo a más parámetros establecidos restringiendo el rango de la búsqueda.

Glosario: Vocabulario con términos y su significado, relativo a las materias.

Efeméride: Sucesos notables ocurridos en la fecha en que se está o de la que se trata, pero en años anteriores.

Descripción del modelo: El usuario hiperentorno realiza búsqueda general o avanzada, para consultar uno o varios recursos que poseen metadatos que los describen y que son usados por las búsquedas para identificar coincidencias y mostrar los resultados. Los recursos pueden ser el texto base, recursos multimedia, interactivos o estructurales. El usuario hiperentorno puede consultar además términos del glosario o efemérides.

2.2. Captura de Requerimientos

Durante el flujo de trabajo Requerimientos, de acuerdo con RUP, el objetivo que se persigue es desarrollar un modelo del sistema que se va a construir, es decir, guiar el desarrollo hacia un sistema correcto. Se definen requerimientos funcionales y no funcionales, así como casos de uso, actores y relaciones entre estos. Durante este flujo de trabajo se aborda qué será lo que el sistema debe hacer, una visión general de las operaciones que podrán realizar los usuarios de la Biblioteca, teniendo en cuenta las nuevas tendencias en la construcción de sistemas e-learning a partir del aprendizaje colaborativo y el software social.

Capítulo 2 Descripción de la propuesta

2.3. Requerimientos funcionales del sistema

Los requerimientos para un sistema son la descripción de los servicios proporcionados por este y sus restricciones operativas. Son condiciones o capacidades que debe cumplir el sistema. A partir de la Colección Futuro se capturaron algunos requerimientos funcionales iniciales que se adaptaron o incorporaron a la plataforma ZERA, de acuerdo a las necesidades del cliente, el resto de los requerimientos son los que se determinaron a partir del estudio realizado.

Los requerimientos que fueron tomados del software educativo Colección Futuro y que se consideraron inicialmente para ZERA, se limitan a la visualización de los objetos sin tener en cuenta las interacciones que la evolución de la web permite en la configuración y personalización de los recursos. A continuación se listan los requerimientos funcionales analizados para el módulo Biblioteca y una descripción de los mismos.

El sistema debe permitir:

RF 1 Realizar búsqueda general.

RF 1.2 Consultar recurso.

RF 1.3 Mostrar recursos asociados a términos de búsqueda introducidos por el usuario.

Luego del estudio teórico realizado fueron capturados los siguientes requerimientos:

RF 2 Realizar búsqueda avanzada de los recursos. Descripción: Teniendo en cuenta criterios de búsqueda en base a los metadatos que describen los recursos.

RF 3 Filtrar los elementos mostrados de acuerdo a diversos criterios. Descripción: Restringe los resultados de una búsqueda de acuerdo con el criterio que se especifique en el filtro, por ejemplo todos los recursos de un autor determinado.

RF 4 Salvar una copia del recurso de acuerdo con los requisitos de derecho de autor. Descripción: Permitirá guardar al escritorio de trabajo del usuario una copia del recurso si las restricciones de licencia y derecho de autor así lo permiten. Esta posibilidad debe existir en pos de evitar los problemas legales y éticos que puede conllevar la limitación de libertades de uso que puedan tener algunos recursos, así como la limitación a los usuarios de su libre uso y distribución.

RF 5 Mostrar los datos relacionados con el recurso como título, autor, tipo, licencia, copyright. Descripción: Mostrar algunos datos de manera predeterminada facilitando su acceso a los usuarios.

Capítulo 2 Descripción de la propuesta

RF 6 Marcar un recurso. Descripción: Agregar el recurso a los recursos favoritos del usuario. Permite el tratamiento personalizado de los recursos, su configuración y uso.

RF 6.1 Organizar los recursos marcados.

RF 6.2 Editar un recurso marcado.

RF 6.3 Eliminar un recurso previamente marcado.

RF 7 Realizar anotación a recurso. Descripción: Permitirá insertar una nota a manera de recordatorio sobre ese recurso en particular. Facilita la personalización de los recursos.

RF 7.1 Ver datos de nota.

RF 7.2 Incluir nota asociada a un recurso.

RF 7.3 Modificar datos de una nota.

RF 7.4 Eliminar nota.

RF 8 Compartir recurso. Descripción: Permite compartir un recurso con uno o varios usuarios de la plataforma, enviándoles una notificación.

RF 9 Evaluar un recurso. Descripción: Dar una puntuación en un rango de valores de menor a mayor (comenzando por 1 hasta 5) a un recurso. Permitirá una mejor gestión de los recursos que se muestran en la Biblioteca.

RF 10 Mostrar los recursos que han sido mejor valorados. Descripción: Deberá mostrar aquellos recursos que tengan mayores votaciones en un tema que se esté consultando. Permitirá valorar la utilidad de un recurso de acuerdo al objetivo con que fue creado, mejorando la posibilidad de uso y la visualización por parte de los usuarios.

RF 11 Mostrar recursos similares a un recurso consultado. Descripción: La similitud se toma en base a temática específica y palabras claves.

RF 12 Consultar artículos de interés que hayan sido compartidos a un usuario por el docente. Descripción: El docente puede compartir artículos o archivos que considere puede servir de ayuda al estudiante en una tarea o en la profundización de un contenido. Estos artículos no forman parte del texto base.

RF 13 Consultar términos del glosario. Descripción: Consultar glosario de la materia desde donde se consulta de manera predeterminada o de otras materias si se desea.

RF 14 Consultar efemérides. Descripción: Se consultan las efemérides del día actual por defecto o de otro día seleccionado por el usuario.

Capítulo 2 Descripción de la propuesta

RF 15 Mantener informado al usuario de las operaciones. Descripción: Mostrar mensajes de información, alerta o confirmación al realizar las diferentes acciones en el sistema que así lo requieran.

RF 16 Validar la integridad de los datos introducidos por el usuario. Descripción: Verificar que los datos introducidos por el usuario son válidos para la acción que se solicita.

2.4. Requerimientos no funcionales del sistema

Los requerimientos no funcionales son propiedades o cualidades que debe tener el sistema. Establecen los servicios que se espera que el sistema proporcione y las restricciones bajo las que funcionará. Son aquellas características que permiten que el producto final sea más o menos usable, rápido o que software o hardware es necesario para que funcione. Pueden agruparse de acuerdo a diversas clases como rendimiento, usabilidad, seguridad, software o hardware.

Se detallan los requisitos no funcionales del sistema recogidos del análisis con el cliente que se había realizado con anterioridad al comienzo de esta investigación y a partir de las necesidades específicas del entorno seleccionado de programación.

Usabilidad

Los usuarios requerirán preparación mínima para interactuar con el módulo.

Fiabilidad

Debe estar disponible las 24 horas del día todos los días de la semana.

Soporte

El soporte o mantenimiento del sistema no debe provocar la caída del servicio.

Restricciones de diseño

El lenguaje de programación a emplear debe ser PHP 5.3.3 o superior. El marco de trabajo base de desarrollo que se utilizará es: Symfony 1.4.6. Como entorno de desarrollo (IDE) se empleará NetBeans 6.9. Como servidor Web deberá usarse Apache 2.2.16. El Sistema Gestor de Base de Datos deberá ser PostgreSQL versión 8.4.

Capítulo 2 Descripción de la propuesta

Requisitos para la documentación de usuarios en línea y ayuda del sistema

El sistema deberá contar con una ayuda que guíe al usuario en la interacción con el sistema.

Interfaz

Interfaces de usuario

Interfaz amigable para la navegación.

Su funcionamiento deberá ser intuitivo y requerir de información mínima.

La información debe estructurarse de manera clara y comprensible.

Interfaces Hardware

Las computadoras que brindarán el servicio cliente del sistema no deberán de presentar prestaciones menores a las brindadas por una Pentium 4, con al menos 512 MB de RAM y 200 MB de espacio en el disco. La conexión de red deberá ser de 128 Kbps de velocidad mínima.

Interfaces Software

Sistema Operativo tanto Windows (win9.x o versión superior) como Linux (cualquiera de sus distribuciones).

Correr sobre cualquier navegador, como mínimo compatible con Mozilla Firefox.

2.5. Descripción del sistema propuesto

2.5.1. Actores

Director: Consulta la información a nivel de estudiante, curso, docente y escuela. Consulta materiales más valorados, los recomendados por el profesor a sus estudiantes, contenidos de las materias y puede valorar un recurso.

Docente: Consulta materiales, incluyendo los más valorados por los estudiantes. Puede valorar, compartir y gestionar recursos favoritos, así como realizar anotaciones a un recurso.

Tutor: Consulta materiales, comparte y descarga recurso.

Capítulo 2 Descripción de la propuesta

Estudiante: Consulta materiales. Puede valorar, compartir, anotar, gestionar marcadores a un recurso, así como consultar los artículos que le sean compartidos por los docentes.

Administrador Local: Consulta materiales.

Consultor biblioteca: Evalúa, comparte, crea notas, gestiona sus recursos favoritos. Rol que generaliza las acciones del docente y el director en la Biblioteca.

Usuario hiperentorno: Consulta materiales.

En la figura 2 se muestra el diagrama de actores.

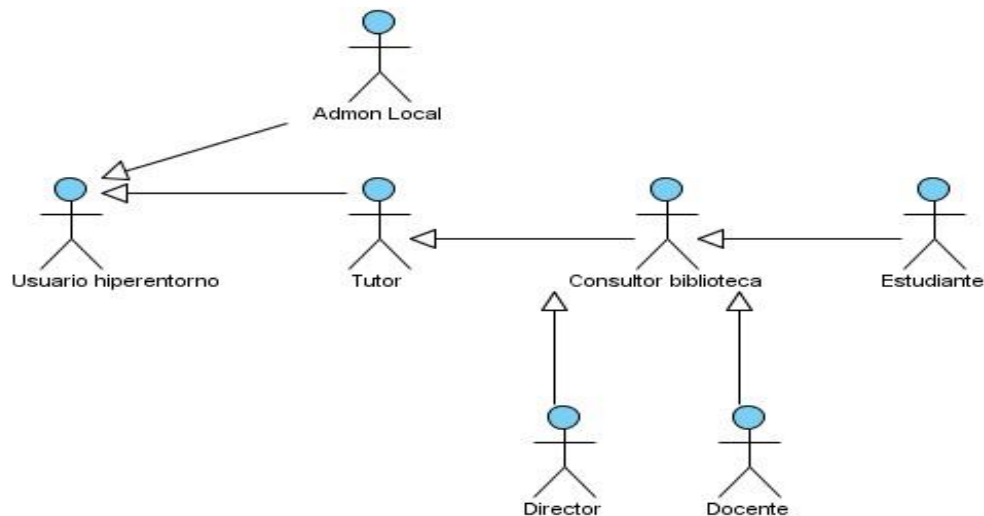


Figura 2. Diagrama de actores.

2.5.2. Casos de Uso

- Gestionar búsqueda.
- Descargar recurso.
- Evaluar recurso.
- Compartir recurso.
- Realizar anotación a recurso.
- Gestionar recursos favoritos
- Mostrar recurso similar.
- Consultar artículos compartidos.
- Consultar glosario.
- Consultar efemérides.

2.5.3. Diagrama de Casos de Uso del sistema.

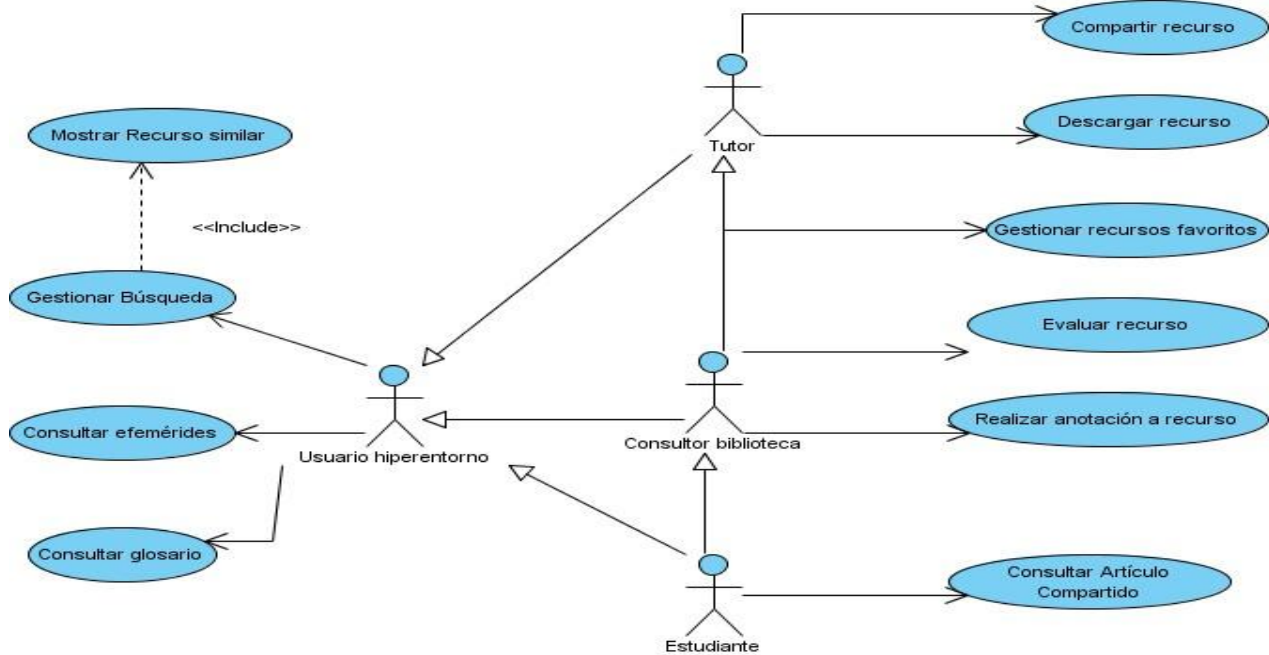


Figura 3. Diagrama de Casos de Uso del Sistema.

2.5.4. Descripción textual de los Casos de Uso del Sistema

Se muestra la descripción del caso de uso Gestionar Búsqueda, el resto puede consultarse en el expediente del proyecto.

Precondiciones

Debe haberse generado el escritorio de trabajo del usuario autenticado. Debe haber seleccionado una materia. Debe haber accedido a la biblioteca.

Postcondiciones

Se realizó una búsqueda general o avanzada, se filtraron elementos de la búsqueda.

Flujo de eventos

Flujo básico

Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona la opción de realizar búsqueda.	
	2. Brinda la posibilidad de realizar las acciones:

Capítulo 2 Descripción de la propuesta

	<ul style="list-style-type: none"> • Introducir un criterio de búsqueda. • Seleccionar un índice de la materia como criterio de búsqueda. • Realizar búsqueda avanzada. • Salir de la vista actual en cualquier momento.
3. Selecciona la opción de introducir un criterio de búsqueda y accede a la opción Buscar .	
	<p>4. Consulta los datos y muestra una lista de coincidencias. Y permite:</p> <ul style="list-style-type: none"> • Seleccionar la cantidad de resultados a mostrarse por página. • Navegar entre las páginas de resultados. • Filtrar los resultados de acuerdo a: <ul style="list-style-type: none"> ○ Autor ○ Materia ○ Fecha de publicación • Salir de la vista actual en cualquier momento.
5. Consulta el recurso. <u>Ver CU Consultar Recurso</u> .	
	<p>6. Muestra los datos del recurso. Y permite:</p> <ul style="list-style-type: none"> • Compartir recurso. • Evaluar recurso. • Crear una nota asociada al recurso.
	7. El caso de uso termina.

2.5.5. Patrones de Casos de Uso.

Debido al papel clave que juegan los casos de uso en el modelado del sistema se hace necesaria su estructuración de la mejor manera posible. Para ello se puede aprovechar las experiencias anteriores codificadas a través de patrones de casos de uso. Estos patrones son un par problema/solución que han demostrado ser la solución adoptada en la comunidad de desarrollo de software. Existen varios tipos de patrones de caso de uso, aquí se mostrarán los que se reflejan en el diagrama de casos de uso del sistema.

- **CRUD** (Creating, Reading, Updating, Deleting, Creación, Lectura, Actualización y Eliminación *por sus siglas en inglés*)

Capítulo 2 Descripción de la propuesta

Se basa en la fusión de casos de uso simples para formar una unidad conceptual.

Completo

Este patrón consta de un caso de uso, llamado Información CRUD o Gestionar información, modela todas las operaciones que pueden ser realizadas sobre una parte de la información de un tipo específico, tales como creación, lectura, actualización y eliminación. Suele ser utilizado cuando todos los flujos contribuyen al mismo valor del negocio, y estos a su vez son cortos y simples. Este patrón se refleja en el modelo en la relación del actor con el caso de uso Gestionar favoritos.

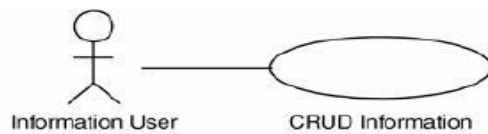


Figura 4. Patrón de caso de uso CRUD Completo.

- Múltiples actores

Rol común

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

Este patrón se utilizó para unir las acciones comunes del docente y el director como Consultor biblioteca.

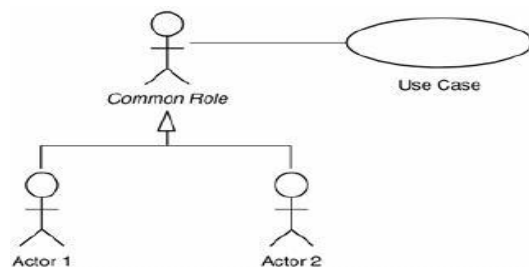


Figura 5. Patrón de caso de uso Rol común.

Capítulo 2 Descripción de la propuesta

Conclusiones

En este capítulo se identificaron y definieron los principales artefactos del flujo de trabajo Requerimientos como lo propone la metodología de desarrollo escogida RUP: actores, casos de uso y los requerimientos funcionales y no funcionales del sistema así como la relación entre estos elementos a través de la estructuración del modelo de Casos de Uso del sistema. Se identificaron los principales conceptos asociados al dominio del problema para conformar el modelo de objetos del dominio. Se definieron los patrones de caso de uso aplicados a la solución propuesta. Todo lo anterior permitirá una mejor comprensión del sistema por parte de usuarios y desarrolladores y servirá de entrada para la generación de los artefactos del análisis y el diseño.

Capítulo 3 Análisis y Diseño del sistema propuesto

3. CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA PROPUESTO

Introducción

Tomando el diagrama de casos de uso como punto de inicio en este capítulo se mostrará la realización a través de clases del análisis, del diseño y colaboraciones del módulo Biblioteca. Se describen los artefactos generados a partir del seguimiento de las actividades del flujo de trabajo Análisis y Diseño, para el caso de uso Gestionar búsqueda tomado como muestra. El resto de los artefactos se encuentran en los anexos. Se obtienen los requisitos refinados a través del modelo de análisis y un plano de la implementación a través del modelo de Diseño. Se presenta además el modelo de los datos y el modelo de despliegue de la aplicación.

3.1. Análisis

En la disciplina de Análisis y Diseño el principal objetivo a alcanzar es transformar los requerimientos en el diseño del sistema, por lo que la función del análisis es proveer una aproximación conceptual al diseño como un paso inicial. Durante el análisis los requerimientos son refinados y se ven en forma de paquetes y clases de análisis. Se establece la trazabilidad entre los requerimientos, casos de uso, clases del análisis y realización de los casos de uso. El análisis se realizó con el objetivo de comprender mejor los requisitos y para proporcionar una entrada mejor elaborada al diseño.

3.1.1. Modelo de Análisis

El Modelo de Análisis es un modelo de objetos conceptual que está compuesto por clases del análisis y sus objetos organizados en paquetes que colaboran. RUP propone que para cada caso de uso se elabore un diagrama de clases con las clases del análisis participantes. El modelo de análisis provee un mecanismo de transformación de comportamiento deseado del sistema a cómo trabajará.

3.1.2. Clases del análisis

Las clases del análisis se modelan a través de tres estereotipos conceptuales básicos: interfaz, entidad y control. Estos estereotipos tienen asociado una semántica y se centran en los requerimientos funcionales delegando los no funcionales a las actividades del Diseño y la Implementación.

Las clases interfaz se encargan de la modelación de la interacción del sistema con los actores. Modelan la parte del sistema que depende de sus actores encapsulando los cambios de la interfaz. Representan

Capítulo 3 Análisis y Diseño del sistema propuesto

abstracciones de formularios, ventanas, paneles e interfaces de aplicación programables (APIs).

Las clases entidad modelan la información que posee una larga vida y que a menudo es persistente. Modelan la información y el comportamiento de algún concepto o fenómeno como un objeto del mundo real o una persona.

Las clases de control son usadas para encapsular el control de un caso de uso en concreto. Representan coordinación, secuencia, transacciones y control de otros objetos. Modelan los aspectos dinámicos del sistema y delegan trabajo a otros objetos de interfaz o entidad.

3.1.3. Diagrama de clases del análisis

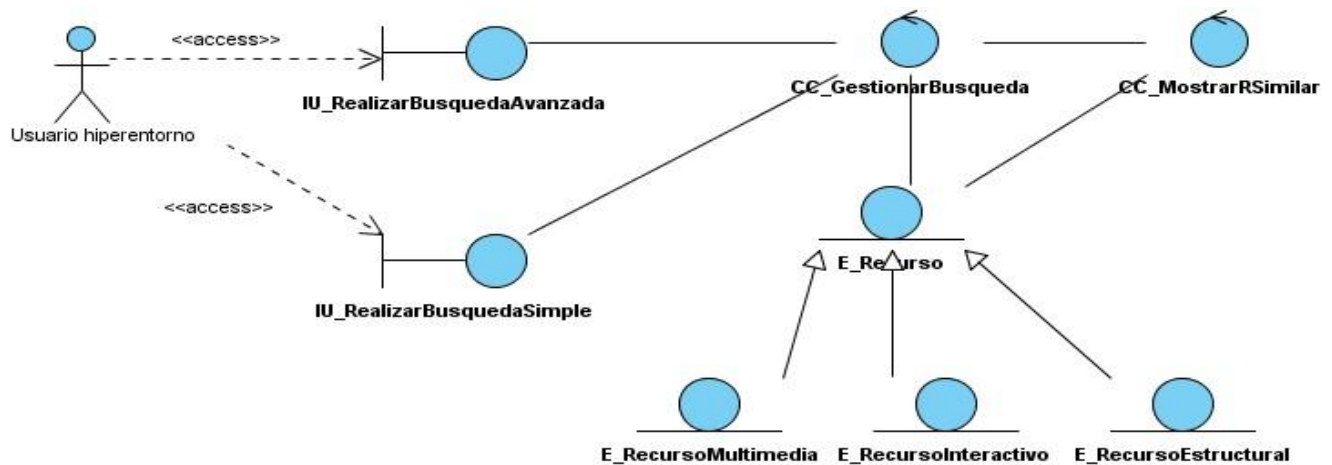


Figura 6. Diagrama de clases del análisis Caso de Uso Gestionar Búsqueda.

3.1.4. Diagrama de paquetes del análisis

Los paquetes del análisis proporcionan un medio para organizar el modelo de análisis en piezas más pequeñas y manejables. La identificación de los paquetes se realiza basándose en los requisitos funcionales y en el dominio del problema. Debido a que los requerimientos son capturados en forma de casos de uso una forma para ser identificados es a través de la asignación de cierto número de casos de uso a un paquete en concreto teniendo en cuenta uno de los aspectos siguientes:

- Casos de uso requeridos para dar soporte a un determinado proceso del negocio.
- Casos de uso para dar soporte a un determinado actor del sistema.
- Casos de uso con relaciones de generalización y de extensión.

Capítulo 3 Análisis y Diseño del sistema propuesto

Para la elaboración del diagrama de paquetes se tomó la primera aproximación teniendo en cuenta las características del sistema.

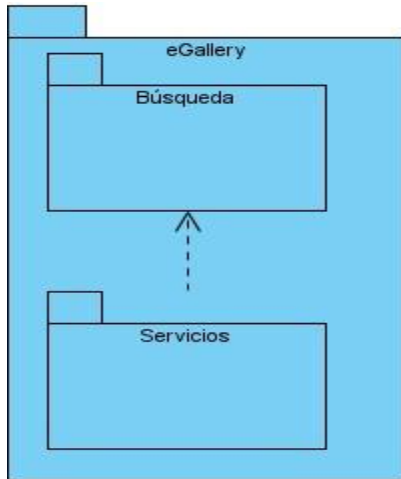


Figura 7. Diagrama de paquetes del análisis.

Dentro del paquete de Búsqueda se relacionan los casos de uso que soportan el proceso de búsqueda y visualización de recursos, a saber: Gestionar búsqueda, Mostrar recurso similar, Descargar recurso y Consultar artículos asignados, se incluyen en este paquete además los casos de uso Consultar glosario y Consultar efemérides. Dentro de Servicios son asociados los que recogen funcionalidades sobre los recursos: Evaluar recurso, Compartir recurso, Realizar anotación a recurso y Gestionar recursos favoritos.

3.2. Arquitectura propuesta

La arquitectura del sistema estará acotada por la del marco de trabajo (framework) utilizado para su construcción: Symfony. La arquitectura de Symfony está basada en el patrón de diseño web MVC (Modelo-Vista-Controlador), estructurado por niveles:

- El Modelo, representa los datos de la aplicación.
- La Vista, maneja la presentación visual de los datos del modelo.
- El Controlador, interactúa con el modelo actualizando la vista.

MVC separa la lógica de la presentación de los datos por lo que las aplicaciones son más fáciles de mantener.

Capítulo 3 Análisis y Diseño del sistema propuesto

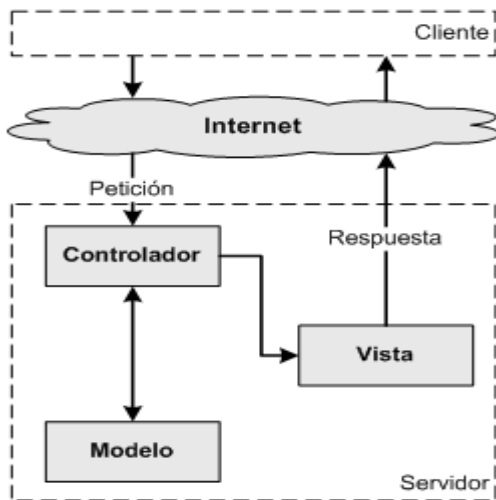


Figura 8. Patrón Modelo-Vista-Controlador.

Symfony realiza su propia implementación de este patrón aprovechando las ventajas de otros patrones de diseño como el decorator para la construcción de la vista. El flujo de trabajo de Symfony puede verse en el Anexo 5.

La elección de la arquitectura determina la definición de los artefactos del modelo de Diseño, que está dado a ser un plano de la implementación, por lo que influye en los artefactos generados en este flujo de trabajo.

3.3. Diseño

3.3.1. Modelo de Diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso, centrándose en cómo los requerimientos funcionales y no funcionales, así como otras restricciones del entorno de implementación, tienen impacto en el sistema. Es usado como abstracción del modelo de implementación del sistema y es por tanto una entrada fundamental a las actividades de Implementación.

En el modelo de diseño los casos de uso son realizados por las clases del diseño y sus objetos. Las clases del diseño representan abstracciones de las clases en el lenguaje de programación a utilizarse.

Capítulo 3 Análisis y Diseño del sistema propuesto

3.3.2. Diagrama de clases del diseño.

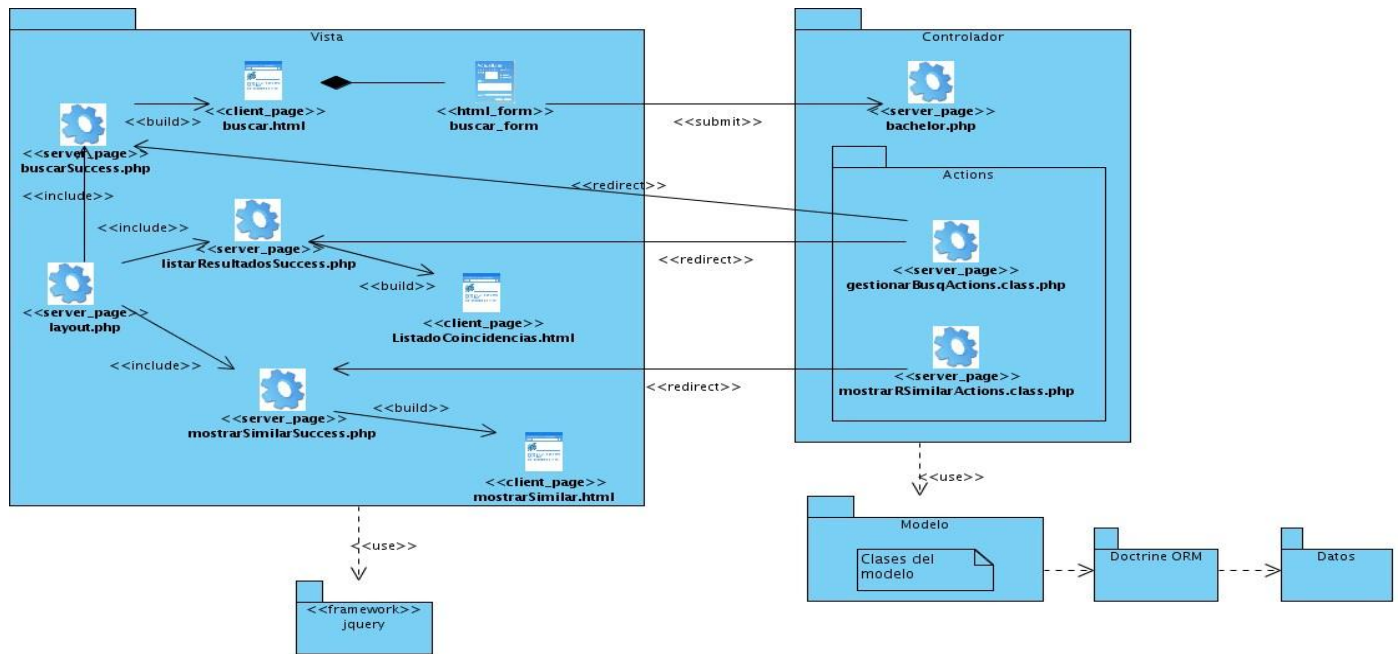


Figura 9. Diagrama de clases del diseño para el caso de uso Gestionar búsqueda.

3.3.3. Subsistemas de diseño.

Los subsistemas de diseño son una forma de organizar los artefactos del modelo de diseño en piezas más manejables. Un subsistema puede contar con clases del diseño, realizaciones de casos de uso, interfaces y otros subsistemas. Como el diseño es un plano de la implementación estos subsistemas tienen relación directa con el entorno de implementación. Los subsistemas de diseño van a estar determinados por la estructura que Symfony le otorga a las aplicaciones web.

El diagrama con los subsistemas identificados para el módulo se muestra a continuación:

Capítulo 3 Análisis y Diseño del sistema propuesto

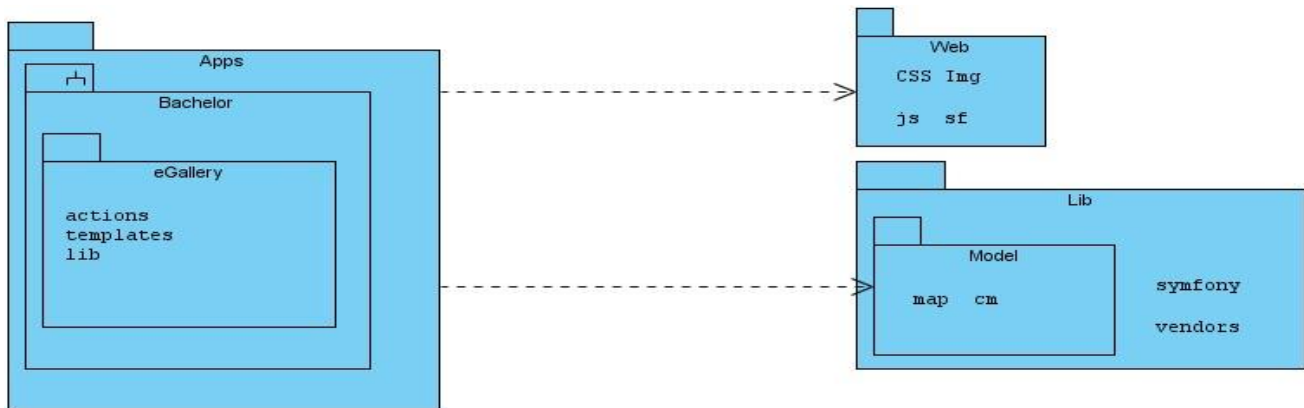


Figura 10 . Diagrama de paquetes del diseño.

3.3.4. Realizaciones de Casos de Uso. Diagramas de interacción.

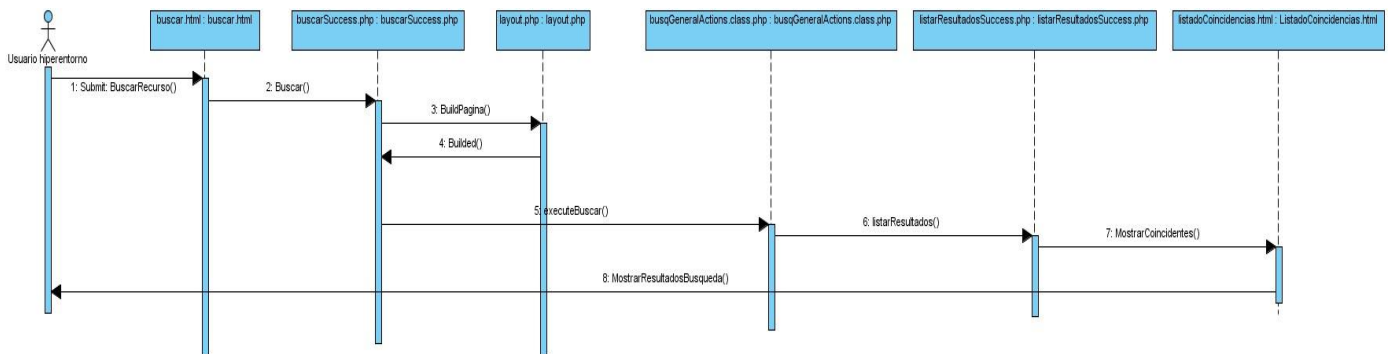


Figura 11. Diagrama de secuencia caso de uso Gestionar Búsqueda, escenario Buscar.

3.3.5. Modelo de Datos

En Symfony la estructura de la base de datos es mapeada a clases del sistema a través de un ORM (Object-Relational Mapping), en el caso de ZERA se seleccionó Doctrine. Esto le permite al desarrollador interactuar con la base de datos a través de objetos y no de sentencias estructuradas de consulta (SQL). A continuación se muestra el modelo de datos para el módulo Biblioteca.

Capítulo 3 Análisis y Diseño del sistema propuesto

3.3.6. Diagrama Entidad-Relación

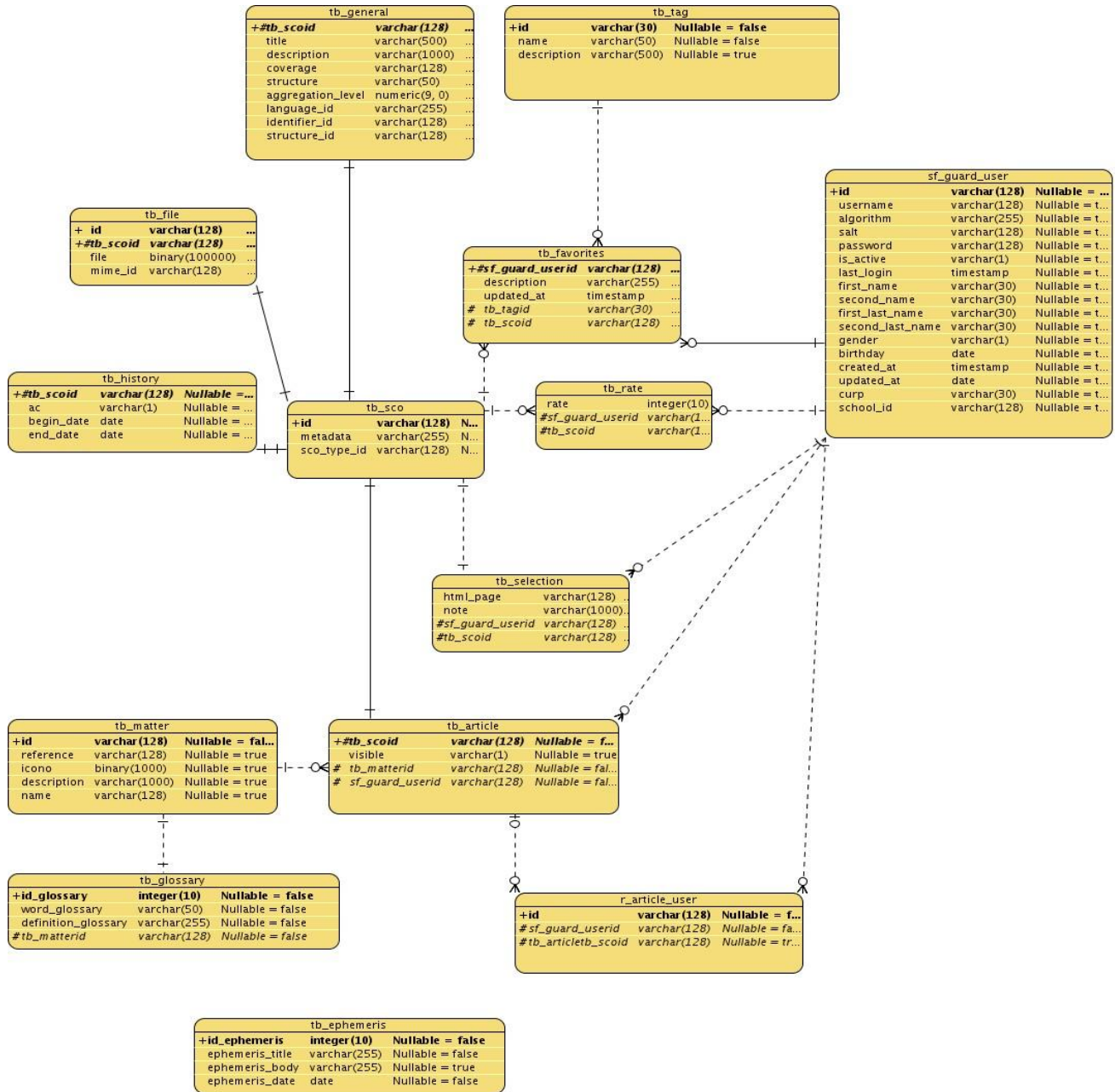


Figura 12. Modelo de Datos.

Capítulo 3 Análisis y Diseño del sistema propuesto

3.3.7. Diagrama de despliegue

El diagrama de despliegue muestra cómo interactúan las interfaces hardware para capturar la configuración de los elementos de procesamiento, y las conexiones entre estos elementos en el sistema. Para su elaboración se tiene en cuenta la arquitectura que exige el cliente de acuerdo a los nodos físicos que forman parte de la solución y como interactúan con distintos protocolos.

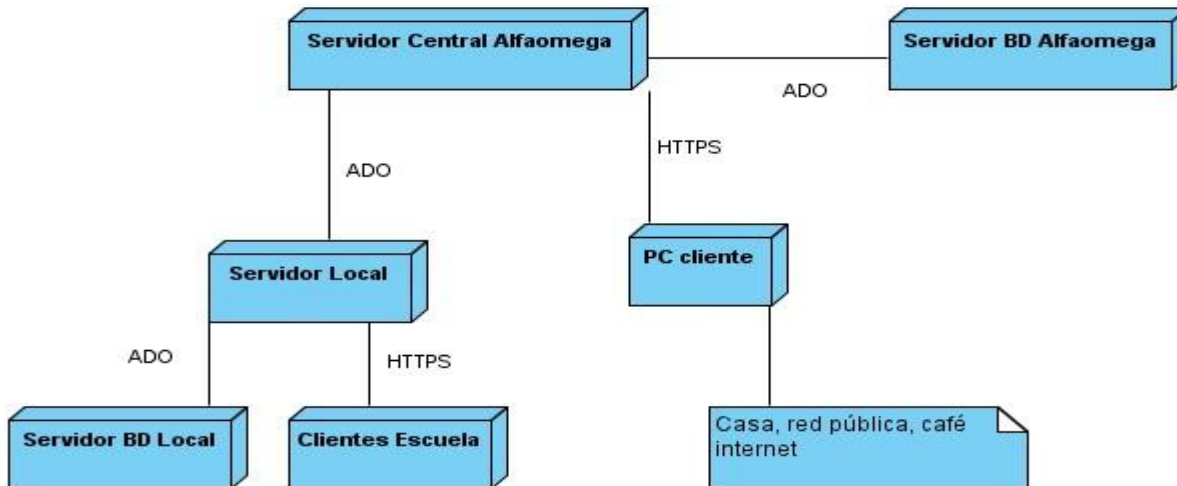


Figura 13. Diagrama de Despliegue.

Descripción del diagrama:

El cliente solicita la existencia de un Servidor central denominado Alfaomega al que se conectarán a través del protocolo de conexión segura HTTPS los servidores locales por institución y las PC clientes externas a dichas instituciones. Al servidor local de la aplicación se conectarán las PC clientes de la institución. Cada servidor, tanto el local como el central tendrán una conexión ADO a sus respectivos servidores de BD.

Conclusiones

A través del capítulo se han ilustrado los artefactos generados a partir del desarrollo de las actividades de Análisis y Diseño. Se refinaron los requisitos a través del modelo de Análisis y se realizó un modelo más cercano al entorno de implementación a través de los distintos artefactos obtenidos del modelo de Diseño, obteniéndose así los artefactos de modelación de la propuesta y una descripción de cómo debe trabajar el sistema para devolver un resultado de valor a sus usuarios.

Capítulo 4 Validación de la propuesta

4. CAPÍTULO 4: VALIDACIÓN DE LA PROPUESTA

Introducción

En este capítulo se presenta la validación a través de métodos de Ingeniería de Requisitos para validar los requerimientos y el método de consulta a expertos Delphi para validar de manera general la propuesta.

4.1. Validación de requisitos

Durante el desarrollo de las actividades de análisis y diseño se capturaron los requerimientos para el sistema, funcionales y no funcionales, se definieron los requerimientos a través de casos de uso, finalmente para completar este ciclo deben validarse estos requerimientos de manera que se pueda comprobar que los requisitos cumplen con las necesidades planteadas inicialmente y que reflejan el sistema que se quiere construir. Al validar los requerimientos de deben efectuar verificaciones de: [50]

- Validez. Usuarios, stakeholders y desarrolladores.
- Consistencia. No deben contradecirse.
- Completitud. Deben definir en conjunto todas las funciones y restricciones del usuario.
- Realismo. Que están de acuerdo con el presupuesto y pueden ser desarrollados en el tiempo estimado.
- Verificabilidad. Deben poder escribirse un conjunto de pruebas que demuestren que el sistema a entregar cumple cada uno de los requerimientos especificados.

4.2. Técnicas de validación

Las técnicas de validación de requerimientos pueden usarse individualmente o en conjunto: [50]

- Revisiones de requerimientos.
- Construcción de prototipos.
- Generación de casos de prueba.
- Análisis de consistencia automático (uso de herramientas CASE).

Para la validación de los requerimientos obtenidos para el módulo Biblioteca se escogió una combinación de las técnicas de generación de casos de prueba y construcción de prototipos, en este caso no funcionales.

Capítulo 4 Validación de la propuesta

4.2.1. Casos de prueba

En la elaboración de los casos de prueba se utilizó un método que es una variante del de particiones equivalentes propuesto por RUP basado totalmente en los casos de uso y consta para su elaboración de tres pasos fundamentales: identificación de escenarios de prueba, identificación de los casos de prueba (identificación de las variables) y por último identificación de valores de datos para las pruebas.

4.2.2. Prototipo de interfaz no funcional

Otra técnica de validación es la de prototipos. Los prototipos de interfaz no funcional permiten, a partir de los requisitos, obtener una interfaz del sistema que puede ser validada por usuarios finales y desarrolladores [50], siendo importante porque permite verificar y corregir las funcionalidades principales interactuando con el equipo de desarrollo ante deficiencias y este último puede aclarar aquellos aspectos que no sean lo suficientemente explícitos y corregir así también la definición de las funcionalidades.

Pueden identificarse 2 tipos de prototipos: [51]

Prototipo evolutivo: Son empleados para lograr una mejor comprensión de los requisitos. Debe ser usado en sistemas donde la especificación no se pueda desarrollar inicialmente.

Prototipo desechable: Se desarrolla a partir de una especificación inicial, para experimentación, después es desechado.

En el análisis del módulo Biblioteca se utilizó el prototipo desechable, una vez que se habían definido los requisitos y se tenía una versión inicial de su especificación.

4.3. Validación de la propuesta mediante criterios de expertos

Para validar la factibilidad y pertinencia de la propuesta se decidió aplicar el método de consulta a expertos Delphi. Este es un método de pronóstico cualitativo que utiliza un grupo de expertos que se mantienen aislados para el análisis. No existe una estructura rígida para aplicar el método pero es usual que se siga una determinada secuencia. Presupone dos etapas: una de selección de indicadores y otra de medición.

Selección de expertos

Se listaron siete personas que cumplirían los requisitos de expertos, a criterio del investigador y que cumplieran unos requisitos mínimos. Estos son:

- ✓ Graduado de nivel superior.

Capítulo 4 Validación de la propuesta

- ✓ Conocimientos sobre los contenidos que sustentan esta investigación: Metodología RUP, e-learning, plataformas de aprendizaje, teorías pedagógicas.
- ✓ Experiencia de al menos 2 años en proyectos de software educativo.
- a) Determinación del coeficiente de conocimiento (Kc).

Se le planteó que evaluaran en una escala del 1 al 10 ascendentemente su nivel de conocimiento sobre el tema específico que se aborda. Ver [Anexo 4](#).

Tabla 2. Cálculo del coeficiente de conocimiento

Escala No. de experto	1	2	3	4	5	6	7	8	9	10	Kc
1								X			0.80
2								X			0.80
3								X			0.80
4								X			0.80
5								X			0.80
6							X				0.70
7								X			0.80

De acuerdo a los valores seleccionados por los expertos en la Tabla 2 se calcula el coeficiente de conocimiento (Kc) dividiendo el valor marcado entre 10.

- b) Calcular el coeficiente de argumentación.

Se solicitó a los expertos una autoevaluación a través del cuestionario sobre fuentes de argumentación, registrado en la tabla 3.

Tabla 3. Ubicación de los expertos en las fuentes de argumentación.

No.	Fuentes de Argumentación	Alto	Medio	Bajo
1	Análisis teóricos realizados por usted	0.30	0.20	0.10
2	Experiencia obtenida	0.50	0.40	0.30
3	Conocimiento de trabajos de autores nacionales	0.05	0.04	0.03
4	Conocimiento de trabajos de autores extranjeros	0.05	0.04	0.03
5	Conocimiento del estado del problema en el extranjero	0.05	0.04	0.03
6	Su intuición	0.05	0.04	0.03

Capítulo 4 Validación de la propuesta

El nivel de argumentación de los expertos fue determinado por autoevaluaciones. Los resultados se muestran en la tabla 4.

Tabla 4. Matriz coeficiente de argumentación por experto.

F.A. Exp.	1	2	3	4	5	6	Suma
1	0,2	0,5	0,04	0,03	0,03	0,04	0,84
2	0,3	0,5	0,03	0,04	0,05	0,04	0,96
3	0,3	0,5	0,04	0,04	0,04	0,03	0,95
4	0,2	0,5	0,03	0,05	0,04	0,03	0,85
5	0,3	0,4	0,05	0,05	0,04	0,04	0,88
6	0,2	0,4	0,03	0,04	0,04	0,04	0,75
7	0,3	0,5	0,05	0,04	0,05	0,05	0,99

Finalmente se calcularon los coeficientes de competencias (K) a partir de los coeficientes de conocimiento (Kc) y argumentación (Ka), los cuales se reflejan a continuación:

Coefficiente de competencia de los expertos (K): $K = 0.5 (Kc + Ka)$

Donde los Niveles de Competencia vienen dados por:

Alta: $0.8 < K < 1.0$; Media: $0.5 < K < 0.8$; Baja: $K < 0.5$.

Los resultados se muestran a continuación:

Tabla 5. Coeficientes y nivel de competencia de los expertos.

No. de experto	Coeficiente de competencia (K)	Nivel de competencia
1	0,8	Alta
2	0,9	Alta
3	0,9	Alta
4	0,8	Alta
5	0,8	Alta
6	0,8	Alta
7	0,9	Alta

Capítulo 4 Validación de la propuesta

Hasta aquí se puede concluir que de los siete expertos seleccionados, los siete obtuvieron calificación Alta, el de mayor puntuación fue de 0.99. Los expertos que obtuvieron menor calificación alcanzaron 0.78, es decir, ninguno obtuvo menos de 0.50, que es la calificación baja dentro del nivel de competencia.

Al hacer una caracterización de la muestra de expertos elegidos, se presentaron los siguientes rasgos obtenidos para los mismos:

- Todos los consultados presentaron niveles de competencia alta.
- El 14.3 % posee la categoría de Máster en Ciencia.
- La media de años de experiencia en el desarrollo de software educativo de los expertos es de 6 años.
- El 85% de los encuestados son graduados de Ingeniería Informática y el 15% es graduado de Ingeniería Automática.
- El 100% de los encuestados está vinculado a proyectos productivos de software educativo.

c) Coeficiente de concordancia

Para la determinación del coeficiente de concordancia de los indicadores en evaluación se parte de los resultados de la encuesta de indicadores ([Anexo 4](#)), donde se presentan los 10 indicadores, cada uno tiene un número del 1 al 5. Donde: 5: Manifestación **MUY ADECUADO**; 4: Manifestación **BASTANTE ADECUADO**; 3: Manifestación **ADECUADO**; 2: Manifestación **POCO ADECUADO**; 1: Manifestación **INADECUADO**. Además se presenta un indicador de evaluación final de la propuesta. El procesamiento de la información se hizo a través de una matriz por experto y se cuantificaron los resultados.

Se realizaron los cálculos correspondientes como se muestra en la tabla 6.

Tabla 6. Matriz del criterio de experto por indicador con cálculos de variables.

<i>EXP. / IND.</i>	1	2	3	4	5	6	7	8	9	10	<i>Evaluación Final</i>
1	5	4	5	5	5	5	5	5	5	3	4
2	5	3	5	5	5	4	5	5	5	4	4
3	5	3	5	5	5	4	5	5	5	3	4
4	4	3	5	5	5	5	5	5	5	4	3
5	5	3	4	4	5	5	5	5	4	4	5
6	3	3	4	4	4	3	4	4	3	4	4
7	5	4	5	5	5	5	5	5	5	5	5

Capítulo 4 Validación de la propuesta

Suma	32	23	33	33	34	31	34	34	32	27	29
Xm	3,2	2,3	3,3	3,3	3,4	3,1	3,4	3,4	3,2	2,7	2,9
Ds	0,8	0,5	0,5	0,5	0,4	0,8	0,4	0,4	0,8	0,7	0,7
Ds/Xm	0,25	0,21	0,15	0,15	0,11	0,25	0,11	0,11	0,25	0,26	0,24
$C=(1-Ds/Xm)*100$	75,4	78,8	85,2	85,2	88,9	74,6	88,9	88,9	75,4	74,4	76,2

El nivel de consenso C, se determina, según el Método Delphi, aplicando la expresión: $C = (1 - Ds/Xm) * 100$; donde: Concordancia = (C); Xm = Valor medio o promedio de los expertos por indicador; Ds=Desviación estándar. Luego se determinó el grado de concordancia por pregunta (indicador) para lo que según el método Delphi el grado de concordancia debe ser de 75% o más ($C > 75$.)

Criterios de los expertos sobre las preguntas

Tabla 7 Grado de concordancia por indicador

No.	Indicador	Grado de concordancia (en %)
1	Calidad de la investigación	75.4
2	Novedad científica	78.8
3	Posibilidad de aplicación de la propuesta	85.2
4	Posibilidad de implementación de la propuesta (en entorno de programación).	85.2
5	Influencia de la propuesta en el mejoramiento de la plataforma ZERA.	88.9
6	Apoyo para la evaluación de los recursos educativos.	74.6
7	Contribuye a la construcción del aprendizaje de los estudiantes.	88.9
8	Integración de la propuesta a la plataforma.	88.9
9	Aceptación de la propuesta por el encuestado.	75.4
10	Posibilidad de adaptarse a otros proyectos de plataformas de gestión de aprendizaje en la UCI.	74.7
11	Evaluación final de la propuesta	76.2

Capítulo 4 Validación de la propuesta

En todos los indicadores se obtuvo un grado de concordancia superior al 75% por lo que se considera que los resultados obtenidos son válidos y fundamentan los criterios dados por los expertos.

Cálculo del coeficiente de consenso total de la propuesta

El nivel de consenso total se establece según las fórmulas siguientes:

$C_c = (1 - V_d / V_t) * 100$; donde: C_c = Coeficiente de concordancia; V_d = Votos negativos, son los de puntuación 2 ó 1; V_t = Votos totales.

CT = 100 porque no existen puntuaciones de 2, ni de 1 en la evaluación de los indicadores de la propuesta. El Coeficiente de concordancia total de la propuesta fue de 100 puntos.

Tabla 8. Votos emitidos por los expertos

	Vt	Vd
	Total de votos	Votos negativos
Votos indicadores	77	0
Votos argumentación	42	0
Votos de coef. de conocimientos	7	0
Total	126	0

Conclusiones

En este capítulo se validaron los requisitos a través de casos de prueba y prototipos de interfaz no funcionales, se validó además de manera general la propuesta por medio del método Delphi, concluyendo que es viable la implementación de la propuesta para el módulo Biblioteca, que permitirá añadir a la plataforma ZERA integración con bibliotecas digitales y mejor visualización y evaluación de los recursos que se gestionan en las materias.

Conclusiones

CONCLUSIONES

Al finalizar la investigación puede afirmarse que se cumplieron los objetivos planteados:

- Se realizó un estudio de las herramientas de gestión de aprendizaje, bibliotecas digitales y su integración con sistemas e-learning, lo que permitió identificar requerimientos y definir características esenciales de una biblioteca en entornos e-learning.
- Se analizaron metodologías de desarrollo de software, lenguajes de modelado y herramientas CASE, lo que posibilitó la selección de las óptimas para realizar el análisis y diseño del módulo Biblioteca de la plataforma ZERA.
- Se demostró la necesidad de modificar el módulo Biblioteca para adaptarlo a las concepciones actuales en el desarrollo tanto de bibliotecas digitales como de sistemas e-learning.
- Se señaló que ninguno de los sistemas de las plataformas de gestión de aprendizaje de código abierto analizados implementaban las funciones de integración de bibliotecas digitales que favoreciera el aprendizaje colaborativo y uso personalizado de los recursos.
- Se capturaron y especificaron requerimientos y se describieron los casos de uso en un modelo de casos de uso del sistema.
- Se modeló a partir del diagrama de casos de uso la propuesta de requerimientos en base a clases del análisis, clases del diseño, paquetes y colaboraciones.
- Se aplicaron patrones de caso de uso, potenciando más eficiencia en el desarrollo de las actividades.
- A través de técnicas de validación de requerimientos se validaron los requisitos, elevando la calidad de la especificación y del diagrama de casos de uso.
- Se validó la propuesta a través del método de experto Delphi, con lo que se demuestra la viabilidad de implementar la propuesta y la calidad de la investigación.

Recomendaciones

RECOMENDACIONES

Se recomienda:

- Dar seguimiento al presente trabajo con la implementación y subsiguientes flujos de trabajo que propone la metodología seleccionada.
- Controlar los cambios que puedan surgir en los requisitos así como los riesgos que de ello se deriven a través de la Gestión de los requisitos.
- Continuar con el estudio de la integración de bibliotecas digitales y plataformas de gestión del aprendizaje en particular en el acceso, uso y administración de recursos que no son generados en los cursos y que se encuentran disponibles en repositorios de objetos de aprendizaje de libre uso.
- Añadir una funcionalidad al módulo en la que se permita referenciar de acuerdo a las normas internacionales de referencias bibliográficas los recursos que se visualizan en la biblioteca.

Bibliografía

BIBLIOGRAFÍA

1. Labañino Rizzo, César y otros. (2001). Primer Seminario Nacional de Guionistas. Cojímar, Oct, 2001. La Habana.
2. Definición de hipermedia. [En línea]. Consultado el 14 de Enero de 2011. Disponible en: http://www.wikilearning.com/curso_gratis/internet_y_la_web_definicionesque_es_hipermedia/6065-1.
3. World Wide Web Consortium. [En línea]. Consultado el 14 de Enero de 2011. Disponible en: <http://www.w3.org/WhatIs.html>.
4. Wikipedia en inglés. Hipertexto. En línea. Consultado el 14 de Enero de 2011. Disponible en: <http://en.wikipedia.org/wiki/Hypertext>.
5. Márquez, Pere. Glosario de Tecnología Educativa. [En línea]. Consultado el 14 de Enero de 2011. Disponible en: www.pangea.org/peremarques/glosario.htm
6. Prof. Inst. Mariño Blanco, Dagoberto y otros. Los hiperentornos de aprendizaje para la web: Una alternativa para el trabajo colaborativo en las universidades pedagógicas. 2009.
7. Wikipedia en inglés. Learning Management Systems. [En línea]. Consultado el 14 de Enero de 2011. Disponible en: <http://en.wikipedia.org/LMS>
8. Taylor Manager, Chris. Information Access Service University of Queensland Library. What is Metadata. [En línea]. Consultado el 15 de Enero de 2011. Disponible en: <http://www.library.uq.edu.au/iad/ctmeta4.html>.
9. Taller en Tecnología de Redes Internet para América Latina y el Caribe, INET'98, programa "Bibliotecarios en Tecnologías Internet", realizado del 13 al 18 de julio de 1998 en Rio de Janeiro, Brasil. [En línea]. Consultado el 14 Enero de 2011. Disponible en: <http://caribe.udea.edu.co/~hlopera/metadata.html>.
10. Wiley, David A. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. 2002 [En línea]. Consultado el 14 Enero de 2011. Disponible en: <http://www.det.wa.edu.au>.
11. Ecured. Plataformas Educativas. [En línea]. Consultado el 14 Enero de 2011. Disponible en: http://www.ecured.cu/index.php/Plataformas_Educativas.

Bibliografía

12. Ossandón Núñez, Yanko. Dirección de Formación Mediatizada Universidad de Tarapacá Arica, Chile. Objetos de Aprendizaje: Un recurso pedagógico para e-learning. [En línea]. Consultado el 18 de Enero de 2011. Disponible en <http://www.uvalpovirtual.cl/archivos/simposio2005/YankoOssandon-ObjetosDeAprendizaje.pdf>.
13. Cano Zárate, José Carlos. Apuntes de Tecnología Educativa para las NT Cátedras de Comunicación Educativa Universidad Marista y Apuntes de Diseño Instruccional (2007). Universidad de las Californias, BC, México.
14. Montero Mendoza, María Teresa y varios autores. Dialogando con el constructivismo: Visiones y versiones. Universidad Autónoma de Ciudad Juárez, México. Colección DINNOVA. [En línea]. Consultado el 18 de Enero de 2011. Disponible en: http://www.uacj.mx/DINNOVA/Documents/modelo_pedagogico.pdf.
15. Dr. Mazarío Triana, Israel, Lic. Mazarío Triana, Ana C. El constructivismo: paradigma de la escuela contemporánea. Universidad de Matanzas, Cuba. Monografía. [En línea]. Consultado el 18 de Enero de 2011. Disponible en: <http://www.bibliociencias.cu/gsd/collect/libros/index/assoc/HASH2243.dir/doc.pdf>.
16. Papert, Seymour. Abstract Constructionism: A New Opportunity for Elementary Science Education. Massachusetts Institute of Technology (MIT). [En línea]. Consultado el 18 de Enero de 2011. Disponible en: <http://nsf.gov/awardsearch/showAward.do?AwardNumber=8751190>.
17. Bell, Jarrod. Constructivist Learning Environments. [En línea]. Consultado el 18 de Enero de 2011. Disponible en: http://sites.wiki.ubc.ca/etec510/Constructivist_Learning_Environments.
18. Castellanos Coutiño, Carlos Alberto. Panorama general de la Educación a distancia. UPM, España. [En línea]. Consultado el 15 de Enero de 2011. Disponible en: http://www.ateneonline.net/datos/19_01_Castellanos_Carlos.pdf.
19. E-learning definition. [En línea]. Consultado el 15 de Enero de 2011. Disponible en: <http://www.about-elearning.com/definition-of-e-learning.html>.

Bibliografía

20. O'Reilly, Tim. What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. Communications & Strategies, No. 1, p. 17, First Quarter 2007. [En línea]. Consultado el 18 de Enero de 2011. Disponible: <http://ssrn.com/abstract=1008839>.
21. Karrer, Tony. Understanding E-learning 2.0. [En línea] 2007. [Citado el: 15 de Enero de 2011.] Disponible en: https://www.astd.org/LC/2007/0707_karrer.htm.
22. Lesk. Understanding digital libraries, 1997. [En línea]. Consultado el 18 de Enero de 2011. Disponible en: https://www.ideals.illinois.edu/bitstream/handle/2142/8343/librarytrendsv49i2i_opt.pdf?sequence=1.
23. Arms, William. Digital Libraries. Chapter Repositories and archives. [En línea] 2000. [Consultado el: 16 de Enero de 2011.] <http://www.cs.cornell.edu/wya/DigLib/MS1999/Chapter13.html>.
24. Arms, William. Digital Libraries. Chapter: An Introduction to digital libraries. [En línea] 2000. [Citado el: 16 de Enero de 2011.] <http://www.cs.cornell.edu/wya/DigLib/MS1999/Chapter1.html>.
25. Candela, L. y otros. The DELOS, Digital Library Reference Model Foundations for Digital Libraries. Versión 0.98. [En línea]. Febrero 2008. [Consultado el: 16 de Enero de 2011.] http://www.delos.info/files/pdf/ReferenceModel/DELOS_DLReferenceModel_0.98.pdf.
26. Digital Library Federation. A working definition of digital library. [En línea] 1998. [Citado el: 16 de 01 de 2011.] <http://www.diglib.org/about/dldefinition.htm>.
27. Rezaei Sharifabadi, Saeed. How digital libraries can support e-learning. Department of Library and Information Science, Alzahra University, Tehran, Irán. [En línea] 2006. [Consultado el: 16 de Enero de 2011.] <http://www.srlst.com/ijst/ijst-Vol4N1/ijst41-39-56.pdf>.
28. Flecker, Dale, Associate Director, Harvard University Library, and McLean, Neil, Director IMS Australia. Digital Library Content and Course Management Systems: Issues of Interoperation. [En línea]. [Consultado en 19 de Enero de 2011]. Disponible en: <http://www.diglib.org/pubs/dlf100/>.
29. Pasquinelli, Art. Digital Library Technologies trends. [En línea] 2002. [Citado el: 18 de 01 de 2011.] Disponible en: http://www.ncsi.iisc.ernet.in/raja/is214/is214-2005-01-04/digital_library_trends-020923.pdf

Bibliografía

30. Referencia Electrónica Bibliotecas digitales y virtuales. Bibliotecas digitales y virtuales con acceso libre a documentos a texto completo. [En línea] 2010 .Consultado el 18 de enero de 2011. Disponible en: <http://www.ua.es/es/bibliotecas/referencia/electronica/bibdigi.html>.
31. Dr. C Torricella Morales, Raúl G. y Stolik Novigrod, Daniel. Biblioteca Virtual de la Educación Superior. Centro de Nuevas Tecnologías de la Información y las Comunicaciones del Ministerio de Educación Superior de Cuba. [En línea].Consultado el 18 de Enero de 2011. Disponible en: <http://www.bibliociencias.cu/gsd/collect/eventos/index/assoc/HASH3d44.dir/doc.pdf>.
32. Agile Alliance. [En línea].Consultado el 20 de Enero de 2011.Disponible en: www.agilealliance.com.
33. Letelier, Patricio y Penadés, María Carmen. Metodologías ágiles para el desarrollo de software: eXtreme Programming. Universidad Politécnica de Valencia 2004 [En línea]. Consultado el 20 de Enero de 2011.Disponible en: <http://www.informatize.net>.
34. Ayuda de Rational RUP. [En línea]. Consultado el 18 de Enero de 2011. Disponible en: <http://www.buenastareas.com/ensayos/Metodolog%C3%ADa-Rup/473434.html>.
35. Jacobson I, Booch. G., Rumbaugh J. The Unified Modeling Language Reference Manual. Editorial Addison-Wesley 1998. 568pp. P 3.
36. Hernández Orallo, Enrique. El lenguaje Unificado de Modelado (UML).
37. Jacobson I, Booch. G., Rumbaugh J. El Proceso Unificado de Desarrollo de Software.
38. García Rubio, Félix Oscar, Bravo Santos, Crescencio. Metodología de Desarrollo de Software. Universidad Autónoma de México. [En línea]. Consultado el 20 de Enero de 2011. Disponible en: http://tabasco.torreingenieria.unam.mx/GCH/Disenio%20y%20gestion/tema3_1xh.pdf.
39. Brito Acuña, Karenny. Selección de metodologías de desarrollo para aplicaciones web en la facultad de Informática de la Universidad de Cienfuegos. [En línea]. Consultado el 20 de Enero de 2011.Disponible en: <http://www.eumed.net/libros/2009c/584/Metodologias%20tradicionales%20y%20metodologias%20agiles.htm>.
40. Wikilearning. Metodologías usadas en ingeniería del software - ¿Qué hay que saber para construir o elegir una metodología? 2007 [En línea]._Consultado el 21 de Enero de 2011.Disponible en:

Bibliografía

- http://www.wikilearning.com/curso_gratis/metodologias_usadas_en_ingenieria_del_software-que_hay_que_saber_para_construir_o_elegir_una_metodologia/3618-1.
41. Departamento Nacional de Software Educativo. Manual de Usuario de la Colección Futuro. Programa REDOX. [En línea]. Consultado el 24 de Enero de 2011. Disponible en: <http://mediateca.rimed.cu/getmediafile.php?key=43733367ad9f1d6340f5>.
 42. Biblioteca Virtual para Moodle. [En línea]. Consultado en 24 de Enero de 2011. Disponible en: <http://vl4moodle.sourceforge.net/index.php>.
 43. Moodle-library-reserve. A Moodle block that handles library reserves. [En línea]. Consultado el 24 de Enero de 2011. Disponible en: <http://code.google.com/p/moodle-library-reserve/>.
 44. Dokeos launches Doogie, the Digital Library. [En línea]. Consultado el 24 de enero de 2011. Disponible en: <http://www.dokeos.com/en/node/840>.
 45. Productos Dokeos libre. [En línea]. Consultado el 25 de enero de 2011. Disponible en: <http://www.dokeos.com/es/soluciones>.
 46. Sakai Project. Wikipedia en inglés. [En línea]. Consultado el 26 de enero de 2011. Disponible en: http://en.wikipedia.org/wiki/Sakai_Project.
 47. Características de Sakai. [En línea]. Consultado el 26 de Enero de 2011. Disponible en: <http://computoeducativo-mega.blogspot.com/2009/09/potencialidades-tecnicas-y-pedagogicas.html>.
 48. Selección de plataforma de Ingeniería de Gestión. 2005. [En línea]. Consultado el 25 de Enero de 2011. Disponible en: <http://www.forumcyt.cu/UserFiles/forum/Textos/0301877.pdf>.
 49. Sommerville, Ian. Ingeniería de Software 2005, séptima edición. Cap 7.
 50. Animación de requerimientos. [En línea]. Consultado el 12 de Mayo de 2011. Disponible en: <http://delta.cs.investav.mx/~pmejia/softeng/Cap8.ppt>.
 51. Morales Arabel. El método Delphi. 2006. [En línea]. Consultado el 16 de Mayo de 2011. Disponible en: <http://www.gestiopolis.com/canales6/eco/metodo-delphi-estadistica-de-investigacion-cientifica.htm>.
 52. Recuperación de información. [En línea]. Consultado el 27 de Mayo de 2011. Disponible en: <http://www.scribd.com/doc/7565063/Bibliotecas-Digitales-UNLA-Modulo-4>

Glosario

GLOSARIO

ADO (ActiveX Data Object): Interfaz programable para acceder a datos en una base de datos.

Doctrine: Es un framework ORM para PHP.

Framework: Esqueleto de un programa de software que sirve como patrón para el desarrollo o implementación de una aplicación.

HTML: HyperText Markup Language (Lenguaje de Marcado de Hipertexto), describe la estructura y el contenido de una página web.

Método de particiones equivalentes: método de prueba de caja blanca. Una partición equivalente es un subdominio de una entrada cuyos miembros se procesan de forma equivalente.

Object Relational Mapping (ORM): Mapeo objeto-relacional, técnica de programación que permite convertir datos entre una base de datos relacional y un lenguaje de programación orientado a objetos utilizando un motor de persistencia.

POO: Programación Orientada a Objetos. Paradigma de programación que se basa en las interacciones de objetos para el diseño de programas. Se basa en tres conceptos: encapsulación, herencia y polimorfismo.

RDF: Resource Description framework. Marco de descripción de recursos para metadatos en la web.

SGML: Estándar de lenguaje de marcado generalizado. Permite que la estructura de un documento puede ser definida en base a la relación lógica de sus partes. Sistema para la organización y etiquetado de documentos.

SQL: Structured Query Language, Lenguaje de Consulta Estructurado, lenguaje declarativo de acceso a base de datos relacionales que permite especificar diferentes operaciones en estas.

XML: eXtensible Markup Language. Lenguaje de etiquetado simple cuyo objetivo es describir datos.