

Universidad de las Ciencias Informáticas

Facultad 4



**Versión 2.0 del módulo Biblioteca del subsistema Bachiller de
la plataforma educativa Zera.**

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

Autores:

Lázara Dorys Rivero Aguilar.

Orestes Pérez Mena.

Tutor:

Ing. Osbel Montero Pérez.

Co-Tutores:

Ing. Miguel Medina Ramírez.

Ing. Irina Ivis Santiesteban Pérez.

La Habana, 2012

“Año 54 de la Revolución”

Pensamiento



Toda Biblioteca debe "... ser no sólo el laboratorio en que el investigador encuentre los elementos que requiere su faena cotidiana, sino, y sobre todo, centro difusor de la cultura capaz de cooperar en la elevación del nivel intelectual de nuestro medio..."

(Luis Gallo Porras)



Declaración de autoría

Declaramos que somos los únicos autores del presente trabajo y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente el ___ días del mes de _____ del año 2012.

Lázara Dorys Rivero Aguilar

Orestes Pérez Mena

Firma del Autor

Firma del Autor

Ing. Osbel Montero Pérez

Firma del Tutor

Ing. Miguel Medina Ramírez

Ing. Irina Ivis Santiesteban Pérez

Firma del Co-tutor

Firma del Co-tutor

Dedicatoria

Dorys

*A mi mamá, mi hermana, mi tía Lazarita, mi
abuela Nina, el tato, mi novio y mis primos.*

Orestes

*A mis padres por haberme dado una excelente
educación y apoyarme siempre.*

Agradecimientos

Dorys

Mi especial agradecimiento va:

A mi tutor Osbel (gran profesor): por ayudarnos tanto y ser tan buena persona, por ser tan preocupado, por ser un tutor ejemplar que desde el primer día hasta el último siempre estuvo ahí apoyándonos.

A Ore (gran compañero): gracias por haberme dado todo tu apoyo, por no permitir nunca que entre nosotros existieran problemas y por ser tan risueño, muchos pensaron que entre dos pinareños no iba a salir una tesis y ya vieron se equivocaron.

A mi mamá Coqui (Odalís): por traerme al mundo para ser hoy lo que soy, por haberme escogido como su hija, este logro es de las dos. TQMM.

A mi tía Lazarita (mi otra mamá): por darme cada día todo su amor sin medidas, por ser sencillamente la razón de mi existir, por darme ánimo para seguir adelante, por estar siempre que te necesité, por entenderme en todo momento y aunque no tenía la razón me dejabas tropezar para aprender de mis errores. TQMM.

A mi abuela Nina (mi vida): por ser mi primera escuela, por existir y darme la oportunidad de demostrarte lo que pudimos lograr juntas.

A mi hermana Madita: por ser tan especial, por quererme tanto, por estar siempre presente, por ser mi amiga, mi confidente y por apoyarme. Espero todo esto pueda servirte de ejemplo.

A mi tío Ramón (mi papá): por TODO, por quererme y sacrificarse por mí, por darme todo su amor y apoyo, esta tesis es lo primero de muchas cosas que quiero dedicarte.

A mis abuelos Tata, Reymundo, Guillermina y mi tío Leonardo: que me supieron guiar y educar siempre les estaré agradecida.

A mi tío Tony: por enseñarme que no hay nada imposible en el mundo y que las personas no se miden por las veces que se caen sino por las que se levantan.

Agradecimientos

A mis hermanos José y Josué: por ser mi ejemplo a seguir, por quererme tanto, por ser los hombres más importantes en mi vida, por siempre ayudarme y conversar conmigo cuando los necesito.

A Margarita: por darme la oportunidad de tener los mejores hermanos del mundo y por quererme tanto.

A Yamirys e Ismael: por acogerme en su casa como una hija y darme todo el cariño del mundo y más, por complacerme en mis malcriadeces y ser tan buenas personas, por todo lo que han hecho por mí y por quererme incondicionalmente

A mis tías Cachita y Gaby, y mis primas Yudeisy y Yudaisy por siempre estar ahí al pendiente de mí y estar entre las personas que más admiro de mi familia.

A Irina y Miguel: por el apoyo que nos brindaron en el desarrollo de la tesis.

Por último:

A mis mejores amigas de toda la universidad, no tengo orden, no me lo hagan más difícil. Maday, Bleydis, Yailín, Arelis, Lanneis, Qni, Dayanis y otras que no se encuentran como Regla, Greisy, Gretel, Delvis y Yoandra, las hermanitas de mi edad que siempre quise tener y saben que aunque tengo mi carácter las quiero mucho.

A mis mejores amigos Pablo, Leonel, Álvarez, Ramón, Alexis, Alejandro, Leo, Javier, Yasmany, Hector, Reinel y Erne que compartieron conmigo buenos y malos momentos.

A todos mis amigos y primas de pinar, que han sabido estar ahí en las buenas y malas, a Yarisleivys, Mirenyilis, Yany, Marised, Frank Luis, Montano, Cruz, Odeyan y Evaicel.

Y quise dejar para último a otra de las persona más importante en mi vida (sin que mi familia y amistades sientan celos, comprendan que este es otro tipo de amor al que siento por ustedes), a mi novio, amigo y compañero Yosley, por estar siempre presente, por tanto amor y dedicación en este tiempo juntos, por quererme tanto, por soportarme en las buenas y malas por ayudarme sin límites, gracias por enseñarme a ver la vida de otra manera. TQMM.

Agradecimientos

Orestes

A mi madre por ser la más cariñosa, dedicada, comprensiva, incansable y preocupada. Le agradezco por darme fuerzas para seguir adelante y por toda la confianza depositada en mí.

A mi padre por haberme inculcado sus valores, principios y convicciones; de los cuales me enorgullezco. Le agradezco por ser mi guía y ejemplo, por sus consejos y su apoyo incondicional.

A mi hermana Adilén; por ser tan cariñosa y estar siempre pendiente de mí, por su apoyo y por cuidarme cuando lo necesité.

A mi cuñado Alexis; por ser un hermano más, por todas sus enseñanzas, consejos y todo su apoyo.

A mi novia Claudia; por ser mi fuente de energía e inspiración. Le agradezco por apoyarme, por su amor, paciencia, comprensión y respeto. Gracias por estar en todo momento a mi lado.

A mis sobrinos; Alex y Martha por ser una razón más para seguir adelante.

A mis suegros León y Evis; por ser dos padres más para mí, por brindarme afecto, preocuparse y por apoyarme durante todos estos años.

A mi tutor Osbel; por ejercer una excelente labor como tutor, por estar siempre pendiente del cumplimiento de todas las tareas, por guiarme y educarme siempre que lo necesité y por velar que mis resultados fueran los mejores.

A mi compañera de tesis Dorys; por cumplir con todas las tareas en el tiempo previsto y mantener una actitud responsable y respetuosa en todo momento.

Agradecimientos

A mi cotutor Miguel; por estar siempre pendiente de la evolución del trabajo, por su disposición y por su ayuda.

A todos los miembros del equipo de trabajo del proyecto Alfaomega en especial los que aportaron su granito de arena en los talleres de tesis.

A todos los profesores que he tenido durante los 5 años de la carrera.

A la UCI y a la Revolución por haberme formado como profesional.

A todos mis compañeros; los del grupo, los del apartamento, los que en algún momento compartieron un instante de mi vida en la Universidad y los que de una forma u otra estuvieron cuando hacía falta. A todos muchísimas gracias.

Resumen

Desde su creación, las bibliotecas virtuales educativas han servido como software educativo en apoyo a los procesos de enseñanza-aprendizaje. Sin embargo, ante el avance vertiginoso de las tecnologías y el surgimiento de nuevos conceptos como redes sociales, trabajo colaborativo, enseñanza centrada en el aprendizaje, diseño centrado en el usuario, entre otros, se hace necesario adecuar el desarrollo de bibliotecas virtuales utilizadas por las instituciones educativas durante el proceso de enseñanza-aprendizaje para aprovechar estas ventajas y brindar un mejor apoyo al mismo. En el departamento de producción de herramientas educativas del centro Fortes de la facultad 4 de la Universidad de las Ciencias Informáticas (en lo adelante, UCI), se desarrolla una segunda versión de un módulo Biblioteca integrado a la plataforma educativa Zera que implementa los nuevos conceptos anteriormente señalados para incentivar el proceso de enseñanza-aprendizaje. Se persigue la idea que los estudiantes accedan a la biblioteca virtual educativa no solo para buscar información sino también para intercambiar experiencias e ideas, aclarar dudas, resolver problemas en equipo, crear sus propios contenidos, evaluar los contenidos diseñados para su aprendizaje y en respuesta a ello, proponer nuevos contenidos de interés del grupo, además de construir su propio conocimiento. Son empleadas un cúmulo de herramientas y tecnologías para el desarrollo entre las que se encuentran los framework Symfony 1.4 y JQuery 1.7, PostgreSQL 9.1, el lenguaje de programación PHP 5, XHTML, JavaScript, Ajax, entre otros. Se utilizó la metodología de desarrollo RUP para guiar el proceso y como métodos investigativos se emplearon la observación, la modelación, el analítico-sintético y el histórico-lógico.

Palabras claves: aprendizaje colaborativo, biblioteca virtual educativa, proceso centrado en el aprendizaje, intercambio de información.

Índice

Introducción.....	1
Capítulo I: Fundamentación teórica.....	6
1.1. Introducción.....	6
1.2. Sistemas E-learning.....	6
1.2.1. E-learning.....	6
1.2.2. La Web 2.0.....	7
1.2.3. Plataformas educativas.....	7
1.2.4. LMS	8
1.2.5. LCMS	8
1.3. Tendencias actuales en la educación a distancia.....	8
1.3.1. Teoría del Constructivismo.....	9
1.3.2. Aprendizaje colaborativo	9
1.4. Sistemas similares.....	10
1.4.1. Plataformas Educativas en el mundo	10
1.4.2. La plataforma educativa Zera.	11
1.4.3. Las bibliotecas digitales.....	12
1.4.4. Las bibliotecas virtuales educativas.....	13
1.4.5. El módulo Biblioteca en la plataforma educativa Zera	13
1.5. Objetos de aprendizaje	14
1.6. Tendencias y tecnologías actuales.....	15
1.6.1. Metodología de desarrollo de software.....	15
1.6.2. Lenguajes de Modelado.....	16
1.6.3. Herramienta CASE.....	16
1.6.4. Herramienta para el modelado de prototipos de interfaz de usuario	17

1.6.5.	Lenguajes de Desarrollo.....	17
1.6.5.1.	Lenguajes del lado del cliente.....	17
1.6.5.2.	Lenguajes del lado del servidor	18
1.6.6.	Frameworks de desarrollo.....	19
1.6.6.1.	Framework del lado del cliente	19
1.6.6.2.	Framework del lado del servidor	20
1.6.7.	Gestor de Bases de Datos.....	21
1.6.8.	Servidores Web.....	22
1.6.9.	Entornos de desarrollo IDE y editores para programación	22
1.7.	Conclusiones parciales.	23
Capítulo II:	Propuesta de solución.	24
2.1.	Introducción.....	24
2.2.	Modelo de Dominio.....	24
2.3.	Descripción de la solución propuesta.....	27
2.4.	Captura de Requerimientos del software	29
2.4.1.	Requerimientos Funcionales.....	29
2.4.2.	Requerimientos No Funcionales.....	32
2.5.	Modelo de Casos de Uso del sistema.	34
2.5.1.	Actores.....	34
2.5.2.	Patrones de casos de uso.....	35
2.5.3.	Diagrama de actores	37
2.5.4.	Diagrama de Casos de Uso del sistema.....	38
2.5.5.	Descripción de los casos de uso del sistema.....	38
2.6.	Conclusiones parciales.....	45
Capítulo III:	Análisis y Diseño	47

3.1.	Introducción.....	47
3.2.	Análisis.....	47
3.2.1.	Modelo de análisis.....	47
3.2.2.	Clases del análisis	47
3.2.3.	Diagramas de clases del análisis.....	48
3.2.4.	Diagramas de interacción.....	48
3.2.5.	Patrón arquitectónico Modelo – Vista – Controlador en Symfony.....	51
3.3.	Diseño.....	51
3.3.1.	Modelo de Diseño	51
3.3.2.	Diagrama de clases del diseño.....	52
3.4.	Modelo de datos.....	52
3.5.	Diagrama de Despliegue.....	53
3.6.	Conclusiones parciales	54
Capítulo IV: Implementación y Prueba.....		55
4.1.	Introducción.....	55
4.2.	Modelo de implementación.....	55
4.3.	Pruebas de software	56
4.3.1.	Niveles de Prueba.....	57
4.3.2.	Métodos de Prueba.....	57
4.3.3.	Pruebas de Caja Negra.....	58
4.3.4.	Pruebas de Caja Blanca.....	58
4.3.5.	Diseño de Casos de Pruebas.....	58
4.3.6.	Resultados Obtenidos.....	59
4.4.	Conclusiones parciales	60
Conclusiones.....		61

Recomendaciones	63
Referencias bibliográficas.....	64
Bibliografía.....	67
Glosario de términos	69

Introducción

El creciente desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC, por sus siglas en inglés) ha emergido con una nueva denominación y filosofía de trabajo en la web: la web 2.0, término comúnmente asociado con las aplicaciones que facilitan el intercambio de información interactivo, el diseño centrado en el usuario y la colaboración. Las posibilidades educativas y de aprendizaje se han multiplicado considerablemente debido a los cambios efectuados. Bajo estas circunstancias, toma auge el e-learning, conjunto de aplicaciones, tecnologías y servicios orientados a facilitar la enseñanza y el aprendizaje a través de internet o intranets que permitan el acceso y la comunicación con los participantes.

Los conceptos asociados al aprendizaje colaborativo, aprendizaje social, enseñanza centrada en el aprendizaje, y teorías del aprendizaje sobre el cognitivismo y constructivismo han ganado nuevos y enriquecedores espacios dentro del propio desarrollo de la Web 2.0. A esto se añade la posibilidad de realizar análisis de los recursos u objetos de aprendizaje con el objetivo de propiciar un mejor uso de los mismos, lo que favorece el aprendizaje.

Con el desarrollo de las nuevas tecnologías, se han creado alternativas para responder a las crecientes necesidades y demandas académicas-administrativas de las instituciones. De esta forma, surgen los sistemas de gestión del aprendizaje (LMS, Learning Management System, por sus siglas en inglés). En este nuevo entorno, los estudiantes asumen un rol esencial en la apropiación del conocimiento, y el papel de los docentes es favorecer estos ambientes de aprendizaje. Varias son las alternativas que han surgido en el mundo para cumplir con las expectativas académicas y administrativas, tanto privativas como de código abierto, por ejemplo: Moodle, Claroline, Dokeos y más recientemente Chamilo.

En el departamento de Producción de Herramientas Educativas de la UCI, se desarrolla una herramienta de gestión del aprendizaje, llamada Zera, la cual está basada en la concepción pedagógica de hiperentornos de aprendizaje. Esta herramienta está compuesta por varios subsistemas identificados como: Administración, Bachiller, Gestión del Aprendizaje, Materia, Recursos y Reportes. Dentro del subsistema Bachiller se encuentra el módulo “Biblioteca”, concebido como un espacio compartido que

preserva las funciones específicas de una colección de recursos multimedia y objetos de aprendizaje interactivos y las incrementa a través de la flexibilidad que ofrece el uso de las tecnologías de la información y las comunicaciones.

En una primera versión del módulo “Biblioteca”, se incorporaron las funcionalidades contenidas en los hiperentornos de aprendizaje desarrollados por el Ministerio de Educación en Cuba, estas se limitan a realizar búsquedas de los recursos que tiene el sistema almacenado mostrando un listado de coincidencias. Debido al gran cúmulo de información almacenada, que además puede crecer indefinidamente, las búsquedas actuales devuelven grandes listas de recursos. Esto dificulta al usuario acceder al elemento que realmente necesita para su aprendizaje y consume gran cantidad de recursos y tiempo, por lo que las consultas se tornan lentas.

Sería de gran utilidad obtener los estados de opinión que poseen los usuarios de la Biblioteca sobre los recursos que tienen a su disposición; propiciar el intercambio de opiniones e información entre ellos para un mejor aprendizaje permitiéndoles compartir sus conocimientos.

Por otra parte, la versión actual de la Biblioteca no permite el trabajo personalizado de los usuarios involucrados con los recursos, a través del cual, estos puedan agruparlos según sus preferencias o descargarlos para almacenarlos independientes de la plataforma.

Por todo lo anterior, se plantea como **problema a resolver** ¿Cómo lograr un fácil acceso, intercambio y personalización de la información contenida en el módulo Biblioteca perteneciente a la plataforma educativa Zera?

El **objeto de estudio** lo constituyen las bibliotecas virtuales que apoyan el proceso de enseñanza-aprendizaje y se centra el **campo de acción** en las bibliotecas virtuales educativas de los sistemas de gestión del aprendizaje.

Para dar respuesta a la interrogante anterior, se traza como **objetivo general** desarrollar una nueva versión del módulo Biblioteca perteneciente a la plataforma educativa Zera que facilite el acceso, intercambio y personalización de la información disponible; del mismo se derivan los siguientes **objetivos específicos**:

- Caracterizar las herramientas disponibles para el desarrollo de bibliotecas virtuales y su integración con los LMS.

- Efectuar el análisis y diseño para la nueva versión del módulo Biblioteca de la plataforma educativa Zera.
- Implementar las funcionalidades propuestas en los flujos de trabajo de Análisis y Diseño.
- Efectuar pruebas de calidad a lo largo del ciclo de desarrollo.

Para dar cumplimiento a los objetivos específicos, se proponen las siguientes **tareas de investigación**:

- Búsqueda de los principales LMS y su integración con las bibliotecas virtuales.
- Estudio de las tendencias internacionales en el desarrollo de plataformas educativas, bibliotecas virtuales, aprendizaje colaborativo y la Web 2.0.
- Estudio y selección de las herramientas disponibles para el análisis y diseño del módulo.
- Estudio de la propuesta inicial de requisitos funcionales y no funcionales para el módulo Biblioteca.
- Definición de los nuevos requisitos funcionales y no funcionales para el módulo Biblioteca.
- Elaboración de los artefactos del flujo de trabajo Análisis y Diseño.
- Análisis y selección de las herramientas para el desarrollo que se adapten a las condiciones de trabajo especificadas en el proyecto.
- Implementación de las funcionalidades propuestas, teniendo en cuenta los requisitos no funcionales especificados en el proyecto.
- Realización de pruebas de caja negra terminada la implementación.

Como **idea a defender** se plantea que la implementación de la nueva versión del módulo Biblioteca propiciará un mayor intercambio entre estudiantes y profesores, así como un fácil acceso y personalización de la información puesta a disposición de los usuarios.

En el desarrollo del proceso investigativo, se emplearon varios **métodos** que se mencionan y justifican a continuación:

Como **métodos teóricos** se utilizaron:

El **analítico-sintético**, en el análisis de documentos para la captura y levantamiento de requisitos de donde se obtienen los rasgos que caracterizan al negocio y que sirven de ayuda para procesar la información y elaborar conclusiones.

El **inductivo-deductivo**, en la descripción e implementación de las funcionalidades propuestas.

La **modelación**, para confeccionar los diagramas que permiten representar la propuesta de solución.

El **histórico-lógico**, para conocer la evolución y desarrollo del objeto de la investigación, las tecnologías, lenguajes, plataformas educativas y bibliotecas virtuales hasta la actualidad.

Como **método empírico** se utilizó:

La **observación**, para conocer la esencia del problema planteado, lo cual aporta un conocimiento más detallado de que se quiere, que hace falta hacer y cómo hay que hacerlo.

Aportes prácticos esperados:

Como resultado, se espera obtener una versión del módulo Biblioteca de la plataforma educativa Zera con funcionalidades especializadas que propicien el intercambio de información entre sus usuarios, el trabajo personalizado con los recursos y el rápido acceso a los mismos.

El documento queda estructurado en 4 capítulos:

El **Capítulo 1** contiene la fundamentación teórica de la investigación. En él, se realiza un análisis del estado del arte acerca del problema y las soluciones existentes en el desarrollo de herramientas con características similares. Se fundamenta además la selección en base al estudio realizado de las tecnologías, metodologías y lenguajes de programación que se ajustan al desarrollo del trabajo.

En el **Capítulo 2**, se realiza una descripción de la propuesta de solución y se plantean los requisitos funcionales y no funcionales que debe tener el sistema para obtener los

mejores resultados. Aquí se obtiene como artefacto fundamental el diagrama de casos de uso del sistema y las descripciones de estos últimos.

En el **Capítulo 3**, se realiza el análisis y diseño del módulo y se presentan los diagramas de clases con estereotipos web, como artefactos generados por la metodología seleccionada para el desarrollo del software, se genera el modelo de datos y diagrama de despliegue.

El **Capítulo 4** abarca los aspectos relacionados con la implementación. En él se describe cómo está implementado el sistema y se definen los tipos de pruebas, los casos de prueba y los resultados de las mismas.

Capítulo I: Fundamentación teórica.

1.1. Introducción

El presente capítulo expone el estado del arte y los conceptos vinculados al objeto de estudio, con el objetivo de lograr una mejor comprensión del problema de la investigación. Se presentan aspectos generales del e-learning, del trabajo colaborativo, así como otros conceptos relacionados con este como la web 2.0, las tendencias actuales en la concepción de los LMS, los hiperentornos y las tecnologías hipermedias. Se realiza además un estudio de diferentes metodologías, herramientas, tecnologías y lenguajes que darán soporte a la solución propuesta y se caracterizan los lenguajes de programación y sistemas para realizar el análisis y diseño, implementación y prueba de la aplicación así como las tendencias y tecnologías actuales. Finalmente, se presentan los elementos seleccionados para el desarrollo del producto informático con tecnologías web.

1.2. Sistemas E-learning

1.2.1. E-learning

Se han elaborado muchas definiciones de e-learning debido fundamentalmente a que los actores que hacen uso de él son muy diversos, cada uno con su idiosincrasia y su ámbito de aplicación. Existen definiciones tan amplias como la presentada por la Sociedad Americana de Capacitación y Desarrollo que lo precisa como “término que cubre un amplio grupo de aplicaciones y procesos, tales como aprendizaje basado en web, aprendizaje basado en ordenadores, aulas virtuales y colaboración digital.” (1) Otros autores como Rosenberg acotan más el alcance del e-learning y lo define como: “el uso de tecnologías Internet para la entrega de un amplio rango de soluciones que mejoran el conocimiento y el rendimiento.” (2)

En el desarrollo de la investigación, se tuvo más inclinación por la definición de García Peñalvo de la Universidad de Salamanca que, desde la perspectiva del desarrollo y explotación de plataformas educativas, define al e-learning como: “la 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 estudiante, además de garantizar ambientes de

aprendizaje colaborativos mediante el uso de herramientas de comunicación síncrona y asíncrona, potenciando en suma el proceso de gestión basado en competencias.” (1)

1.2.2. La Web 2.0

Es un concepto que se acuñó en el 2003 y se refiere al fenómeno social surgido a partir del desarrollo de diversas aplicaciones en Internet. Establece una distinción entre la primera época de la Web (donde el usuario era básicamente un sujeto pasivo que recibía la información o la publicaba, sin que existieran demasiadas posibilidades para generar la interacción) y la revolución que supuso el auge de los blogs, las redes sociales y otras herramientas. Está formada por las plataformas para publicación de contenidos, las redes sociales, los servicios conocidos como wikis y portales de alojamiento de fotos, audios o videos. La esencia de estas herramientas es la posibilidad de interactuar con el resto de los usuarios o aportar los contenidos. (3)

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, investigador canadiense líder en aprendizaje en línea. Es producto de la confluencia entre las teorías de la construcción social del conocimiento y de los avances tecnológicos aportados por la web 2.0 profundizando el camino de la resignificación del conocimiento por parte de comunidades de aprendizaje abiertas. (4)

Para llevar a cabo un programa de formación basado en e-learning, se hace uso de plataformas o sistemas de software que permiten la comunicación e interacción entre profesores, alumnos y contenidos. Se tienen principalmente dos tipos de plataformas: las que se utilizan para impartir y dar seguimiento administrativo a los cursos en línea o LMS y, por otro lado, las que se utilizan para la gestión de los contenidos digitales o LCMS (Learning Content Management Systems, por sus siglas en Inglés).

1.2.3. Plataformas educativas

Con el nombre de plataforma es como genéricamente se conoce a la herramienta tecnológica usada para distribuir el conocimiento. En contextos de formación, se refiere al conjunto de equipos y software básico sobre el cual va a funcionar un sistema que se desea diseñar, desarrollar o instalar para apoyar actividades de aprendizaje en línea. Una 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. (5)

Es una capacitación no presencial que, a través de la tecnología, posibilita y flexibiliza el acceso y el tiempo en el proceso de enseñanza-aprendizaje, adecuándolos a las habilidades, necesidades y disponibilidades de cada docente.

1.2.4. LMS

Una plataforma o software LMS es un sistema normalmente basado en tecnología web, destinado a la administración y gestión de la formación a distancia. Su finalidad es integrar las herramientas de gestión de usuarios (alumnos, tutores, administradores, etc.) con el material de apoyo para la formación (cursos, enlaces, archivos complementarios, etc.) y un sistema de seguimiento del progreso de los alumnos. (6)

Se podría decir que un software LMS permite la gestión mínima de un sistema de formación e-learning.

1.2.5. LCMS

Sistema basado en web que se utiliza para crear, aprobar, publicar, administrar y almacenar recursos educativos y cursos en línea. (7) Se refiere al término Learning Content Management System. Esta plataforma e-learning, agrupa todas las funciones del software LMS, pero además, añade la posibilidad de gestionar contenidos, lo que permite crear y gestionar los cursos online en la propia plataforma. Básicamente, se podría decir que la principal ventaja que aporta una plataforma LCMS frente a una LMS es la posibilidad de crear contenido sin herramientas externas a la solución e-learning.

1.3. Tendencias actuales en la educación a distancia

En el nuevo escenario educativo, las TIC deben ser miradas como un instrumento o un medio para mejorar los procesos de enseñanza-aprendizaje. Cuando se enfoca a Internet como un instrumento para este fin, existen teorías como el Proyecto Zero¹, Inteligencias Múltiples, las Teorías de la Conversación, las Teorías del Conocimiento situado y el Constructivismo que fundamentan este enfoque.

Como se conoce, para el desarrollo exitoso de un software educativo se deben tener en

¹ Proyecto de investigación desarrollado en la Universidad de Harvard para entender y mejorar el progreso del aprendizaje en niños y adultos.

cuenta varios elementos adicionales a los que normalmente se emplean en la construcción de otro tipo de software. Uno de estos elementos es la teoría de aprendizaje que se debe seguir en dependencia de las características del medio, de las necesidades educativas identificadas, de las características de los usuarios, entre otros. Por teorías de aprendizaje se entiende a las definiciones y explicaciones dadas a los procesos de asimilación del contenido, con sus características y peculiaridades, por parte de los usuarios. Estas tratan de explicar cómo ocurre el proceso de aprendizaje en el hombre. Una de las que se está poniendo en práctica en la actualidad debido al creciente uso de internet en la educación, es la teoría del Constructivismo.

1.3.1. Teoría del Constructivismo

John Abbott y Terence Ryan (8) identifican al constructivismo como: “cada alumno estructura su conocimiento del mundo a través de un patrón único, conectando cada nuevo hecho, experiencia o entendimiento en una estructura que crece de manera subjetiva y que lleva al estudiante a establecer relaciones racionales y significativas con el mundo”.

Según Wilson (9), “un entorno de aprendizaje constructivista es un lugar donde los alumnos deben trabajar juntos, apoyándose unos a otros, usando una variedad de instrumentos y recursos informativos que permitan la búsqueda de los objetivos de aprendizaje y actividades para la solución de problemas”.

Tomando como apoyo el concepto dado por Wilson se concluye que en el constructivismo los estudiantes construyen conocimientos en función a sus experiencias previas, estructuras mentales, creencias o ideas que usan para interpretar objetos y eventos. Además de que las acciones que realiza el estudiante sobre la realidad lo ayudan a construir su conocimiento. Uno de los elementos asociados a la teoría del constructivismo es el aprendizaje colaborativo.

1.3.2. Aprendizaje colaborativo

Kaye (10) define el aprendizaje colaborativo como la adquisición individual de conocimiento, destreza y actitudes que ocurre como resultado de la interacción en grupo.

Eggen y Kauchak señalan que los estudiantes que explican y elaboran, aprenden más que los que solamente escuchan explicaciones, quienes a su vez aprenden más que los

estudiantes que aprenden solos. “El aprendizaje colaborativo alienta la elaboración, pidiendo a los estudiantes que hablen acerca de sus nuevas ideas con otros estudiantes de su grupo”. (11)

De los criterios anteriores se toma el de Eggen y Kauchak debido a que mediante el aprendizaje colaborativo se puede estimular la comunicación interpersonal y facilitar las relaciones entre estudiantes al permitir que compartan información, trabajen con documentos conjuntos y participen en la solución de problemas y toma de decisiones.

1.4. Sistemas similares

La educación a distancia es una modalidad educativa en la que los estudiantes no necesitan asistir físicamente a un aula, el proceso de enseñanza-aprendizaje es dirigido y apoyado en la mayoría de los casos por plataformas educativas. Además estas plataformas educativas pueden incluir o no, espacios como bibliotecas virtuales, donde los usuarios consultan los recursos que tienen a su disposición para el aprendizaje.

1.4.1. Plataformas Educativas en el mundo

- Sakai

El Proyecto Sakai desarrolla software educativo de código abierto. Tiene su origen en la Universidad de Michigan y en la Universidad de Indiana, a las que se unieron el MIT y la Universidad de Stanford, junto a la Iniciativa de Conocimiento Abierto (OKI) y el consorcio uPortal.

Contiene un módulo denominado SCORM Cloud (Nube de Scorm) que permite al rol de profesor ver informes detallados sobre las interacciones de sus estudiantes con los recursos SCORM (Modelo de Referencias de Objetos de Contenido Compartido) que están siendo utilizados en sus tareas.

- Moodle

Es un software libre de código abierto llamado también entorno virtual de aprendizaje (EVA). Es un sistema de gestión de cursos, de distribución libre, que ayuda a los educadores a crear comunidades de aprendizaje en línea.

Promueve una pedagogía constructivista social (colaboración, actividades, reflexión crítica, etc.). Su arquitectura y herramientas son apropiadas para clases en línea, así como también para complementar el aprendizaje presencial. Tiene una interfaz de

navegador de tecnología sencilla, ligera, y compatible.

- **Dokeos**

Es un entorno de e-learning y una aplicación de administración de contenidos de cursos y también una herramienta de colaboración. Es un sistema flexible y de muy fácil uso mediante una interfaz de usuario sumamente amigable e intuitivo. Es una herramienta de aprendizaje especialmente recomendada a usuarios que tengan nociones mínimas de computación cuyo objetivo es la preocupación por el contenido.

Además cuenta con Doogie, una biblioteca digital. Todos los recursos multimedia que se importan a un curso de Dokeos serán indexados de acuerdo a su contenido textual (indexado de semántica de texto completo), o bien si el recurso es un video este se indexará por medio de palabras clave, además, se pueden incorporar materiales como exámenes y cursos SCORM.

- **Claroline**

Es una plataforma de aprendizaje y trabajo virtual de código abierto y software libre que permite a los formadores construir eficaces cursos *online* y gestionar las actividades de aprendizaje y colaboración en la web. Traducido a 35 idiomas, Claroline tiene una gran comunidad de desarrolladores y usuarios en todo el mundo.

Cuenta con un módulo denominado “Documentos” que muestra un listado de nombres de documentos referenciados a través de URLs, además de las imágenes relacionadas con un curso determinado. Posee un sistema de búsqueda rápida de sus recursos a través de un filtro por nombre.

1.4.2. La plataforma educativa Zera.

Zera es una plataforma para la gestión del aprendizaje que integra los principales conceptos de los hiperentornos. Un hiperentorno es una mezcla de diferentes tipos de software educativos sustentado en tecnología hipertexto que constituyen un modelo de medio de enseñanza para el apoyo al proceso de enseñanza-aprendizaje, concebidos además como fundamento teórico y conceptual del desarrollo de las colecciones cubanas de software educativo. Se desarrolló una primera versión de este software en la UCI que integra las mejores prácticas y elementos arquitectónicos de soluciones similares y las principales especificaciones y estándares educativos desarrollados y

utilizados a nivel mundial en plataformas educativas.

Permite la gestión de los hiperentornos de aprendizajes, la visualización de estos y la gestión del aprendizaje. Se divide en seis subsistemas, los cuales se encuentran estrechamente interrelacionados, estos son: la gestión de materias, la administración, la gestión de los recursos interactivos, la administración del aprendizaje, los reportes y el subsistema Bachiller.

1.4.3. Las bibliotecas digitales

Son instituciones culturales o educativas que no necesitan de una sede física como ya es conocido, también se les denomina bibliotecas virtuales. La biblioteca tradicional, al introducir el empleo de Sistemas Integrados de Automatización de Bibliotecas, se transforma en una biblioteca electrónica la cual gestiona documentos impresos y documentos electrónicos tangibles, por ej. CD-ROM. La biblioteca digital ya no sólo brinda información referencial sino también, acceso remoto a los textos completos. Los usuarios interactúan con la biblioteca digital, pero no entre ellos ni con los bibliotecarios.

Las posibilidades que brinda la Web 2.0 para el trabajo colaborativo entre los usuarios, bibliotecarios y editores define un nuevo modelo de biblioteca, la biblioteca digital personalizada. Los procesos que intervienen en este nuevo concepto son: el acceso y personalización de la información y el control estadístico del uso de los documentos, entre otros. La biblioteca digital personalizada además de las potencialidades disponibles en la biblioteca digital incluye la posibilidad de interacción e interactividad entre los usuarios.

Algunos ejemplos de bibliotecas digitales consultadas son: la galería online de la biblioteca británica, que muestra un catálogo de libros virtuales disponibles a consulta, una sala de visitas online a exhibiciones culturales, una guía por los principales libros consultados y una sección de galería personalizada. Otro ejemplo es la Biblioteca Virtual de Colorado con mucho contenido histórico, con catálogos de libros, artículos y revistas, con ayuda diferenciada a los usuarios, entre otros servicios. Por último la Biblioteca Digital Mundial es un inmenso cúmulo de información de todas partes del mundo que permite consultar los elementos por tipo, tema, lugar de procedencia, fecha de creación, institución, etc., además de filtrar los resultados por otros criterios.

1.4.4. Las bibliotecas virtuales educativas

Tienen por lo general el mismo funcionamiento de una biblioteca virtual o digital pero con una orientación educativa en apoyo a los procesos de enseñanza-aprendizaje. Están asociadas a una institución educativa como universidades, escuelas politécnicas, etc. Muestran la información que la institución pone a disposición de sus estudiantes y usuarios en general para su aprendizaje, pueden mostrar además información sobre los proyectos de la escuela, los eventos futuros, noticias relacionadas, etc.

Se realizó un estudio sobre algunas bibliotecas virtuales educativas en el mundo donde se consultaron: la biblioteca de la Universidad de Zaragoza, la biblioteca virtual de la Universidad de Chile, la biblioteca de la Universidad Complutense de Madrid, la biblioteca de la Universidad Politécnica de Madrid, algunas bibliotecas de EBSCO², entre otras.

En todos los casos, la información que se muestra es de fácil acceso y se brinda una interfaz amigable e intuitiva al usuario, donde se presenta por cada recurso: el título, breve descripción, el autor, año de publicación, etc. Permiten un trabajo personalizado con los recursos, como por ejemplo: guardar en una carpeta personal, enviar por correo, imprimir la descripción del recurso, reservar el recurso para su posterior extracción, entre otros. Permiten además, de una forma fácil y sencilla, realizar una búsqueda avanzada de los recursos de la biblioteca donde incluyen elementos como la búsqueda booleana, ordenar el listado de coincidencias, realizar la búsqueda por una gran cantidad de criterios, brindar ayuda al usuario para realizar la búsqueda, devolver los resultados por coincidencia parcial o exacta, entre otros. En algunos casos las búsquedas pueden incluir elementos de otras bibliotecas virtuales educativas de distintas universidades o instituciones.

1.4.5. El módulo Biblioteca en la plataforma educativa Zera

La plataforma educativa Zera cuenta con el módulo Biblioteca que permite la consulta y visualización de todos los recursos de la misma puestos a disposición de los usuarios. Las consultas son distribuidas en tres grandes categorías: las medias, que incluyen imágenes, videos, animaciones, sonidos, recursos interactivos, entre otros; los recursos estructurales, que incluyen elementos de apoyo al contenido como curiosidades,

² Base de dato que ofrece textos completos, índices y publicaciones periódicas, académicas que cubren diferentes áreas de las ciencias y las humanidades.

definiciones de términos, teoremas, entre otros; y por último, la información de interés que incluye los artículos, archivos y enlaces de interés para los usuarios. Las consultas se realizan en dependencia del tipo de recurso y del elemento del contenido al que está asociado el recurso, dígame capítulos, temas o subtemas, en dependencia también de la materia en la que se encuentre el usuario. Posterior a la consulta, se permite al usuario filtrar el listado de coincidencias por el nombre del recurso. A partir del listado de recursos se accede al contenido del mismo al hacerle clic encima. Estas consultas se visualizan de forma diferente en dependencia del tipo de recurso, por lo que se hace necesario redefinir esta funcionalidad para poder agrupar todos los recursos de forma común, tanto para la búsqueda avanzada como para los demás listados que así lo requieran.

1.5. Objetos 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. El concepto de Objeto de Aprendizaje se está utilizando ampliamente en el mundo de la educación a distancia, David Wiley define objeto de aprendizaje como recurso digital que puede ser reusado para soportar el aprendizaje. (12)

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. Entre las más destacadas se encuentran:

IEEE, SCORM e IMS Global Learning Consortium. Estas especificaciones tienen su base en los metadatos y en el lenguaje XML.

El estándar de metadatos LOM (Learning Object Metadata por sus siglas en inglés), desarrollado por la IEEE es el primer esquema de metadatos acreditado para tecnología de aprendizaje (13). El objetivo de este estándar es simplificar las operaciones de búsqueda, gestión e intercambio de OA en la web.

SCORM 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.

1.6. Tendencias y tecnologías actuales

Para el desarrollo de la primera versión de la plataforma educativa Zera, se realizó un estudio de las tendencias y tecnologías actuales, donde se compararon las diferentes metodologías, herramientas CASE³, lenguajes y frameworks de desarrollo, sistemas gestores de bases de datos y servidores web existentes en el mundo del desarrollo del software. A partir de esta investigación quedaron definidas por el proyecto cuáles de estas herramientas y tecnologías se utilizarían en el proceso de desarrollo. Debido a que la solución que se va a desarrollar es una versión de lo antes realizado en el módulo Biblioteca, se trabajó sobre las tecnologías, lenguajes, metodologías etc. ya definidos. A continuación se precisan algunas características de las mismas.

1.6.1. Metodología de desarrollo de software.

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. En la presente investigación la metodología seleccionada para el desarrollo de la misma es una metodología pesada o tradicional.

Metodologías pesadas o tradicionales

Las metodologías tradicionales se enfocan en la 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.

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 un proceso formal: Provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto). Fue desarrollado por Rational Software

³ Ingeniería de Software Asistida por Computadoras.

y está integrado con toda la suite Rational de herramientas. Puede ser adaptado y extendido para satisfacer las necesidades de la organización que lo adopte. (14)

Además, esta metodología es adecuada tanto para el tipo de proyecto que se va a desarrollar como para las condiciones de trabajo creadas puesto que no existe una estrecha comunicación con el cliente.

1.6.2. Lenguajes de Modelado

Los lenguajes de modelado constituyen una guía muy importante para la representación de la estructura de los sistemas informáticos. Consiste en un conjunto de vistas, diagramas, símbolos y mecanismos generales o reglas que indican cómo utilizar los elementos.

- UML 2.0 (Lenguaje Unificado de desarrollo, Unified Modeling Language)

Es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos y describe la semántica esencial de lo que estos diagramas y símbolos significan. UML se puede usar para modelar distintos tipos de sistemas: sistemas de software, sistemas de hardware y organizaciones del mundo real.

Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas; es un lenguaje, por lo que cuenta con reglas para combinar tales elementos. UML es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso. Es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos. (15)

1.6.3. Herramienta CASE

- Visual Paradigm for UML 8.0

Es una herramienta utilizada para la Ingeniería de Software. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Constituye una herramienta de software libre de probada utilidad

para el analista.

Fue diseñada para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque Orientado a Objetos. Se caracteriza por ser multiplataforma, fácil de instalar y actualizar, es empleada para visualizar y diseñar elementos de software para lo que utiliza como lenguaje de modelado UML, proporciona a los desarrolladores una plataforma que les permite diseñar un producto con calidad de forma rápida y facilita la interoperabilidad con otras herramientas CASE como Rational Rose. (16)

1.6.4. Herramienta para el modelado de prototipos de interfaz de usuario

- Balsamiq Mockups 2.1

Balsamiq Mockups es una aplicación multiplataforma (por lo que es posible instalarlo en Windows, Mac y Linux) que sirve para crear de un modo dinámico, sencillo y práctico, bocetos o Mockups de interfaces de usuario, no solo para páginas web, sino también para aplicaciones de escritorio, o cualquier cosa que requiera interfaz gráfica. Usa componentes prefabricados, la tarea de incluirlos y si se requiere modificar su posición o proporciones se logra arrastrándolos al entorno de trabajo. (17)

Esta herramienta se utilizará para diseñar todas las interfaces gráficas de la solución propuesta ya que gracias a su sencillez y eficacia facilita que los diseños sean legibles y se desarrollen en tiempo récord.

1.6.5. Lenguajes de Desarrollo

1.6.5.1. Lenguajes del lado del cliente

- XHTML 1.0

El lenguaje eXtensible Hypertext Markup Language (XHTML) es el lenguaje de marcas de texto utilizado normalmente en la www (World Wide Web) para crear documentos con hipertexto (documentos formados por enlaces a otros documentos, imágenes, audios, videos y aplicaciones). El lenguaje XHTML puede ser creado y editado con cualquier editor de textos básico, como puede ser Gedit en Linux, el Bloc de notas de Windows, o cualquier otro que admita texto sin formato como GNU Emacs, Microsoft Wordpad, TextPad, Vim, Notepad++, entre otros. (18)

Este lenguaje nace precisamente con el objetivo de remplazar a HTML ante su

limitación de uso con las cada vez más abundantes herramientas basadas en XML y va a permitir una correcta interpretación de la información independientemente del dispositivo desde el que se accede.

- **CSS 2.1 (Cascading Style Sheets)**

Es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. Es la mejor forma de separar los contenidos y su presentación, pues obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). (19)

Además, su uso mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes. Es imprescindible para crear páginas web complejas.

- **Java Script**

Java Script es una de las múltiples maneras que han surgido para extender las capacidades del lenguaje HTML. No es un lenguaje de programación propiamente dicho como C, C++, Delphi, etc. Es un lenguaje script u orientado a documento, como pueden ser los lenguajes de macros que tienen muchos procesadores de texto y planillas de cálculo. No se puede desarrollar un programa con JavaScript que se ejecute fuera de un Navegador. Es un lenguaje interpretado (las instrucciones las analiza y procesa el navegador en el momento que deben ser ejecutadas) que se embebe en una página web HTML. (20)

Este lenguaje se puede probar directamente en cualquier navegador sin necesidad de procesos intermedios. Esto evita tener que compilar cada vez que se haga un cambio pero ralentiza el proceso ya que se debe interpretar cada una de las líneas cada vez que se ejecuta. No guarda ninguna relación directa con el lenguaje de programación Java.

1.6.5.2. Lenguajes del lado del servidor

- **PHP 5.3.3 (acrónimo de PHP: Hypertext Preprocessor)**

Es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web ya que puede ser incrustado en HTML.

Lo que distingue a PHP de lenguajes lado-cliente como JavaScript es que el código es

ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá los resultados de ejecutar el script, sin ninguna posibilidad de determinar qué código ha producido el resultado recibido. El servidor web puede ser incluso configurado para que procese todos los archivos HTML con PHP y entonces no hay manera de que los usuarios puedan conocer con exactitud el tipo de código implementado. (21)

Es un lenguaje multiplataforma (un software que trabaja en diversas plataformas); tiene soporte para la conexión con un gran número de bases de datos como MySQL, PgSQL, Oracle, Informix, entre otros; es capaz de expandirse potencialmente por su gran cantidad de módulos con bibliotecas externas de funciones; es libre (open source o código abierto) y por lo tanto asequible para todo el que lo desee.

1.6.6. Frameworks de desarrollo

En general los frameworks de desarrollo son soluciones completas que contemplan herramientas de apoyo a la construcción (ambiente de trabajo o desarrollo) y motores de ejecución (ambiente de ejecución), en otras palabras, un framework es un conjunto de librerías y componentes de probada capacidad, que incluye documentación y metodología de uso, que permite diseñar, construir e implantar de forma más uniforme, rápida y con mayor calidad.

Representan una arquitectura de software que modela las relaciones generales de las entidades del dominio y provee una estructura y una especial metodología de trabajo.

1.6.6.1. Framework del lado del cliente

- JQuery 1.8.16

Es una librería de JavaScript para acceder a los objetos del DOM (Modelo en Objetos para la Representación de Documentos) de un modo simplificado. Ofrece una infraestructura con la que se tendría mayor facilidad para la creación de aplicaciones complejas del lado del cliente.

Además, facilita la compatibilidad de la aplicación para los navegadores más conocidos, así como el trabajo con técnicas como AJAX⁴, implementando funcionalidades

⁴ Acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas. Permite realizar cambios sobre las páginas sin necesidad de recargarlas.

específicas con este objetivo. Permite agregar efectos a las páginas haciéndolas más interactivas y con una considerable disminución de código. Tiene una estructura que le da organización al proyecto y evita la implementación de funcionalidades comunes. (22)

Posee una versión compacta que recoge toda la implementación en un solo fichero, y lo hace más fácil de cargar. Puede acoplar librerías complementarias como JQuery UI que se encarga de la creación de elementos visuales con poderosas características como los acordeones, los calendarios. Además de permitir los eventos de arrastrar y soltar.

1.6.6.2. Framework del lado del servidor

- **Symfony 1.4.15**

Symfony es un framework PHP que facilita el desarrollo de las aplicaciones web. Se encarga de todos los aspectos comunes de las aplicaciones web y reserva al programador la tarea de desarrollar las características únicas de cada proyecto.

Funciona con todas las bases de datos comunes (MySQL, PostgreSQL, SQLite, Oracle, MSSQL Server); es compatible solamente con PHP5 desde hace años, para asegurar mayor rendimiento y acceso a las características más avanzadas de PHP; está basado en la premisa de *"convenir en vez de configurar"*, donde el desarrollador solo debe configurar aquello que no es convencional; está preparado para aplicaciones empresariales, ya que se puede adaptar con facilidad a las políticas y arquitecturas propias de cada empresa u organización; es flexible hasta cualquier límite y extensible mediante un completo mecanismo de plugins; está publicado bajo licencia MIT de software libre y apoyado por una empresa comprometida con su desarrollo; es fácil de instalar y configurar en sistemas Windows, Mac y Linux. (23)

Este framework ha sido probado con éxito durante años en varias aplicaciones gigantescas (Yahoo! Answers, Dailymotion, delicious) y en otros miles de sitios pequeños y medianos. Permite controlar hasta el último acceso a la información e incluye por defecto protección contra ataques XSS y CSRF⁵. Es un framework bien documentado, pues se han publicado cinco libros gratuitos de calidad y siempre actualizados. Además su código fuente incluye más de 9.000 pruebas unitarias y funcionales.

5 (Del inglés Cross-site request forgery o falsificación de petición en sitios cruzados)

1.6.7. Gestor de Bases de Datos

Un sistema gestor de base de datos se define como el conjunto de programas que administran y gestionan la información contenida en una base de datos. Ayuda a realizar las siguientes acciones:

- Definición de los datos.
- Mantenimiento de la integridad de los datos dentro de la base de datos.
- Control de la seguridad y privacidad de los datos.
- Manipulación de los datos. (24)

Proporciona una interfaz entre los datos, los programas que los manejan y los usuarios finales. Almacena una descripción de datos llamada diccionario de datos, así como los usuarios permitidos y los permisos y tiene que haber un usuario administrador encargado de centralizar todas estas tareas.

- PostgreSQL 9.1

La última serie de producción es la 9.1. Sus características técnicas la hacen una de las bases de datos más robustas a nivel internacional. Su desarrollo comenzó hace más de 16 años y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. (25)

Es un sistema de gestión de bases de datos objeto-relacional con código fuente disponible libremente. Sus últimas versiones están a la altura de otras bases de datos comerciales. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

Es un sistema que puede ser utilizado en plataformas UNIX y en Windows, además que tiene soporte para distintos tipos de datos como XML y JSON (formato ligero para el intercambio de datos) y permite añadir tipos de datos personalizados.

1.6.8. Servidores Web

Un servidor web puede ser tanto un ordenador de grandes dimensiones y capacidad como un programa informático que utiliza el protocolo de comunicaciones HTTP para recibir las peticiones de información de un programa cliente (navegador) en el ordenador del usuario. Una vez interpretada la solicitud que el servidor cliente le hace (la página web a la que se desea navegar), el servidor web recupera la información solicitada de sus carpetas y la envía al programa cliente (al ordenador del usuario por vía del navegador) para su visualización. Cuando los servidores web utilizan protocolos de seguridad para la transmisión de datos entre ellos y los clientes (los ordenadores de los usuarios que interactúan con ellos solicitando ver sus páginas), hablamos de servidores web seguros.

- Apache 2.2

Servidor web de código abierto. Su desarrollo comenzó en febrero de 1995 por Rob McCool. La primera versión apareció en enero de 1996, el Apache 1.0. Hacia el 2000, el servidor Web Apache era el más extendido en el mundo. El nombre «Apache» es un acrónimo de «a patchy server»-un servidor de remiendos-, es decir un servidor construido con código pre existente y piezas y parches de código. Es el auténtico software libre en el ámbito de los servidores y el ejemplo de mayor éxito, por delante incluso del kernel Linux. (26)

Desde hace años, más del 60% de los servidores web de Internet emplean Apache. Cuenta con soporte para CGI (Common Gateway Interface) ya porta características extendidas como personalización de las variables de entorno y soporte de reparación de errores o debugging, que son difíciles de encontrar en otros servidores web; es capaz de interpretar script PHP y permite crear fácilmente un sitio web.

1.6.9. Entornos de desarrollo IDE y editores para programación

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

- NetBeans IDE 7.0.1

El IDE NetBeans es un reconocido entorno de desarrollo integrado disponible para

Windows, Mac, Linux y Solaris. El proyecto NetBeans está formado por un IDE de código abierto y una plataforma de aplicación que permite a los desarrolladores crear con rapidez aplicaciones web, empresariales, de escritorio y móviles utilizando la plataforma Java, así como JavaFX, PHP, JavaScript, Ajax, y C/C++.

NetBeans IDE es un producto libre y gratuito sin restricciones de uso, basado en plugins, que permite ser ampliado por terceros y ser utilizado para tantos lenguajes como se desee. Tiene un amplio soporte para PHP brindando un sistema de completamiento de código inteligente y amplio. Brinda todas las opciones de un cliente de accesos a servidores de control de versiones lo que garantiza el trabajo permanente con la última versión del fichero en el servidor. Permite la ejecución de todos los comandos además de la creación de plantillas de código para agilizar la escritura de los elementos más comunes. (27)

1.7. Conclusiones parciales.

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 similares y se concluyó que en su mayoría necesitan 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 tengan la posibilidad de personalizar, acceder de manera rápida, valorar y referenciar recursos de aprendizaje.

El equipo de dirección del proyecto al que pertenece esta solución seleccionó un grupo de herramientas y tecnologías para su realización. Con el objetivo de apoyar esta selección, en este capítulo se exponen las principales características por las que se eligió RUP como metodología de desarrollo, JavaScript y PHP v5.3 como lenguajes de programación, UML v2.0 como lenguaje de modelado, Visual Paradigm para UML v8.0 como herramienta CASE, Symfony v1.4.15, JQuery v1.8.16 como frameworks de desarrollo y PostgreSQL v9.1 como gestor de base de datos.

Capítulo II: Propuesta de solución.

2.1. Introducción

En el desarrollo de software, es de vital importancia definir los procesos que intervienen en este, para lograr así un mejor entendimiento del sistema que se va a desarrollar por parte de los clientes y desarrolladores. Se tiene como propósito exponer el contexto en que se desarrolla el problema que da origen a esta investigación. Se identifican los requisitos funcionales, no funcionales y los casos de uso (en lo adelante CU) que guiarán el desarrollo del sistema propuesto. Además, se realiza el diagrama de casos de uso con el objetivo de mostrar las relaciones entre los actores del sistema y las secuencias de funcionalidades en las que están involucrados.

2.2. Modelo de Dominio

Al desarrollar la solución propuesta se encuentra la problemática que los procesos no están bien definidos, pues los cambios sobre la descripción de los mismos se producen con frecuencia. Debido a ello, se plantea la idea de modelar la solución mediante un modelo de dominio capaz de definir las instancias de las clases y las relaciones entre ellas, sin llegar a enunciar las clases del software en sí.

Descripción de las clases del diagrama de clases del dominio

Para lograr una mejor comprensión del diagrama, es necesario describir las clases relacionadas en el modelo.

Usuario bachiller: Usuario que generaliza a los actores Gestor de preguntas, Administrador local, Director y el Tutor por sus acciones comunes.

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

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: Persona que recibe contenidos educativos en uno o varios grupos de una escuela.

Tutor: Persona que representa legalmente a un estudiante. Un tutor puede tutorar a

varios estudiantes.

Administrador Local: Persona que 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 diferentes tipos como recursos estructurales, recursos multimedia y recursos interactivos.

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 y 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, Teoremas, Leyes y Problemas físicos.

Notas: Proceso por el cual el usuario emite sus criterios sobre los recursos que son puestos en la Biblioteca a su disposición, los cuales comparte con sus compañeros.

Preguntas y respuestas: Proceso por el cual el usuario realiza preguntas o emite respuestas relacionadas a algún recurso de los existentes en la Biblioteca.

Metadatos: Datos que describen los recursos, ejemplo: autor, fecha de creación, derechos de copia, etc. El estándar utilizado para la descripción de los metadatos de objetos de aprendizaje y que se usa para la descripción de los recursos en la plataforma educativa Zera es LOM (1.0). Los elementos de los datos según este estándar son agrupados en nueve categorías: General, Ciclo de vida, Meta-metadatos, Técnica, Educacional, 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.

Información de interés: Recurso que muestra la información contenida en determinado artículo que sea de interés para el usuario.

Descripción del modelo: El usuario bachiller consulta términos del glosario, información de interés y realiza búsquedas para consultar uno o varios recursos que poseen metadatos que los describen: estos son usados por las búsquedas para identificar coincidencias y mostrar los resultados. Los recursos pueden ser recursos multimedia, interactivos o estructurales. El estudiante y docente pueden consultar, evaluar, seleccionar recursos favoritos y realizar anotación sobre ellos, a diferencia del director, administrador local y tutor quienes solo pueden consultar los recursos y al tutor se agrega la posibilidad de evaluar un recurso.

Después de analizar el problema y sus principales conceptos, se hace posible el empleo de éstos en la elaboración del modelo de dominio, donde se presenta un marco conceptual y la relación existente entre las definiciones.

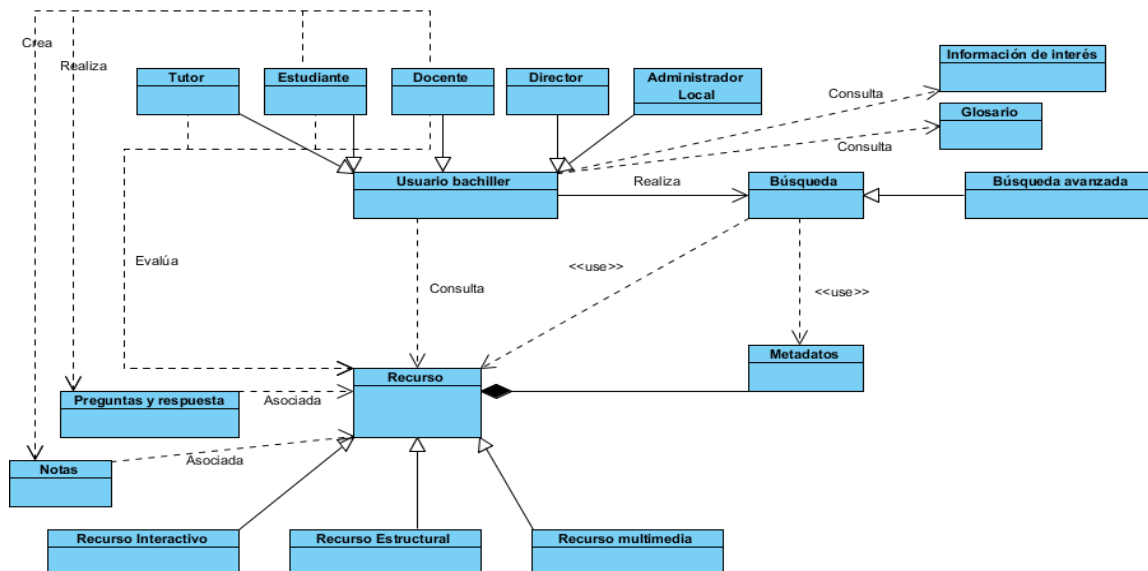


Figura 1. Diagrama de clases del dominio.

2.3. Descripción de la solución propuesta

A partir del análisis expuesto en el Capítulo 1, se definieron características que se deben considerar para el desarrollo de la nueva versión del módulo Biblioteca, que lo enriquecerán (al módulo) y aportarán mayor calidad a la plataforma educativa Zera. Se redefinirán funcionalidades ya implementadas en una versión anterior, específicamente las relacionadas con listar los recursos con vista a flexibilizar la integración con las nuevas, puesto que en la primera versión se listan en función del tipo y de forma diferente, lo que imposibilita agruparlos en un mismo escenario. Además de que se usan consultas con *Doctrine* (librería de acceso a datos) que devuelven *DoctrineCollections* (listado de objetos relacionados) un poco pesadas porque la base de datos almacena de forma física algunos tipos de recursos como imágenes, simuladores, animaciones y secuencias interactivas; lo que conlleva a que las mismas se tornen lentas. Además, se añadirán nuevas funcionalidades con el objetivo de obtener un software educativo capaz de propiciar espacios para el intercambio de ideas, opiniones e información entre sus usuarios, el trabajo personalizado con los recursos disponibles y el fácil acceso a la información.

Los espacios de intercambio se evidenciarán a través de la gestión de preguntas y respuestas directas entre los mismos y el proceso de compartir notas de estudio con sus compañeros.

La gestión de preguntas y respuestas, permitirá a los estudiantes lanzar una pregunta que esté asociada a un recurso aunque no necesariamente la duda puede ser sobre la comprensión del mismo. A través del uso de iconos intuitivos, al estilo de las redes sociales, los estudiantes podrán comprobar la existencia de preguntas lanzadas por sus compañeros al navegar por la Biblioteca. El proceso de respuesta será rápido, intuitivo y sin gasto excesivo de recursos. También la notificación de respuesta se realizará de forma rápida a través de íconos parpadeantes y sugerentes.

En el proceso de crear y compartir notas de estudios, primeramente los usuarios localizarán el recurso, posteriormente, si lo desean, podrán asociarle un texto a modo de nota de estudio o aclaración y contarán con la posibilidad de compartir este texto con sus compañeros de grupo. Aquí se podrán compartir ideas, conocimientos, hallazgos, en resumen, todo un cúmulo de información útil al estudiante para su aprendizaje. También brindará la posibilidad de que otros usuarios respondan las notas compartidas,

lo que quedará registrado en la plataforma.

Estos elementos permitirán que los estudiantes aprendan de las experiencias de sus compañeros y que interactúen entre sí de forma tal que construyan su propio conocimiento.

El trabajo personalizado con los recursos se evidencia al descargarlos, organizarlos o evaluarlos. Un ejemplo de ello será la posibilidad de marcar el recurso como favorito. Este recurso marcado se indexará para ser visualizado en una página personal del usuario a la que este podrá acceder desde cualquier parte de la Biblioteca. Ello facilitará que pueda localizar de una forma más rápida la información que él crea útil para su aprendizaje. El módulo propiciará que el propio estudiante intervenga en la dirección de su conocimiento permitiéndole que agrupe los recursos que considere necesarios para su aprendizaje. Por otra parte esta funcionalidad brindará la posibilidad a aquellos estudiantes que no cuentan con dispositivos de almacenamiento propio, de almacenar la información que les resulte relevante.

Otro ejemplo del trabajo personalizado con los recursos será la evaluación de un recurso por parte de los usuarios que lo consultan. En esta propuesta se brindará la posibilidad de realizar una evaluación a recursos desde dos vías o en dos momentos diferentes, la primera será una evaluación rápida sin tener que acceder a ninguna otra interfaz diferente de aquella donde se está navegando, la segunda será a través de una serie de criterios de evaluación ya sean pedagógicos, de diseño gráfico o aspectos generales que ya serán gestionados desde el subsistema Recursos de la plataforma. La evaluación de un recurso será individual de cada usuario, ya sea estudiante, profesor o tutor y este podrá modificarla tantas veces lo crea necesario.

También permitirá el fácil acceso al recurso que realmente se desee encontrar a través de una búsqueda avanzada, la cual contará con operadores booleanos; búsqueda por coincidencia parcial o frase exacta y búsquedas por varios términos a la vez, por uno o varios tipos de recursos.

Además, se implementarán otras funcionalidades para dar mayor vistosidad y competencia al software. Entre estas se pueden citar: la posibilidad de listar los recursos mejor valorados por los usuarios, listar los recursos más visitados y listar los recursos similares en dependencia de la coincidencia de palabras claves.

Esta constituye una breve descripción de la solución propuesta, la cual será desarrollada con una orientación educativa en apoyo a los procesos de enseñanza-aprendizaje.

2.4. Captura de Requerimientos del software

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.

2.4.1. Requerimientos Funcionales

Los requerimientos funcionales (en lo adelante RF) son capacidades o funciones que el sistema debe cumplir. Se mantienen invariables sin importar con qué propiedades o cualidades se relacionen.

RF 1 Lanzar preguntas directas: Los estudiantes lanzarán preguntas para aclarar dudas sobre el contenido correspondiente al recurso deseado, o el profesor, para ver cómo se asimiló el mismo. Tributa al aprendizaje colaborativo.

- **RF 1.1 Modificar pregunta:** Modificar una pregunta lanzada con anterioridad, solamente por el usuario que la lanzó.
- **RF 1.2 Eliminar pregunta:** Eliminar una pregunta lanzada con anterioridad, solamente por el usuario que la lanzó.
- **RF 1.3 Listar preguntas:** Listar preguntas asociadas a un recurso.

RF 2 Responder preguntas lanzadas: El profesor deberá responder las preguntas realizadas por los estudiantes. También se permite a los estudiantes responder las preguntas de sus compañeros. Tributa al aprendizaje colaborativo.

- **RF 2.1 Modificar respuesta:** Modificar una respuesta emitida con anterioridad, solamente por el usuario que la hizo.

- **RF 2.1 Eliminar respuesta:** Eliminar una respuesta emitida con anterioridad, solamente por el usuario que la hizo.
- **RF 2.3 Listar respuestas:** Listar respuestas asociadas a una pregunta.
- **RF 2.4 Valorar respuesta:** El estudiante o profesor hace una valoración en una escala de 1 a 5 sobre una respuesta de un compañero.

RF 3 Realizar búsqueda avanzada de los recursos: Realizar una búsqueda teniendo en cuenta criterios de búsqueda en base a los metadatos que describen los recursos: (título, palabras claves, autor) además de otros criterios como tipo y contenido al que pertenece el recurso.

RF 4 Listar recursos: Listar los recursos mostrando por cada uno algunos datos (título, breve descripción e imagen de referencia) de manera predeterminada. Se facilita así el acceso a los usuarios. Visualizar los recursos en una lista y de forma común para distintos tipos. (Este requisito está incluido en la primera versión y será redefinido)

RF 5 Mostrar recurso: Visualizar el contenido completo del recurso. (Este requisito está incluido en la primera versión, se redefinirá solo para los recursos de información de interés y se eliminarán para los elementos del glosario porque se visualizarán en otro escenario)

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

- **RF 6.1 Ver los recursos marcados del usuario:** Ver los recursos seleccionados por el usuario como sus favoritos.
- **RF 6.2 Eliminar un recurso previamente marcado:** Eliminar un recurso que ya no desee tener entre sus favoritos.

RF 7 Realizar anotación a recurso: Insertar un apunte a manera de recordatorio sobre ese recurso en particular. Facilita el trabajo personalizado con los recursos.

- **RF 7.1 Ver datos de una nota:** Visualizar los datos de los apuntes insertados por el usuario sobre un recurso en particular.
- **RF 7.2 Modificar datos de una nota:** Modificar los datos de un apunte creado por el usuario sobre un recurso en particular.
- **RF 7.3 Eliminar nota:** Eliminar un apunte creado por el usuario previamente.

RF 8 Compartir nota de estudio: Compartir un recurso con uno o varios usuarios de la

plataforma, enviándoles una notificación. En caso de ser Docente puede compartir sus notas con los demás profesores y con sus alumnos. En caso de ser Estudiante, puede compartirlas con su profesor y compañeros de grupo.

RF 9 Evaluar un recurso: 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 9.1 Modificar evaluación del recurso:** Modificar los datos de la evaluación dada por el usuario sobre un recurso en particular.

RF 10 Mostrar los recursos que han sido mejor valorados: Mostrar aquellos recursos que tengan mayores votaciones positivas. 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: La similitud se toma en base a temática específica y palabras claves.

RF 12 Mantener informado al usuario de las operaciones: Mostrar mensajes de información, alerta o confirmación al realizar las diferentes acciones que así lo requieran.

RF 13 Validar la integridad de los datos introducidos por el usuario: Verificar que los datos introducidos por el usuario son válidos para la acción que se solicita.

RF 14 Consultar los recursos históricos por una línea de tiempo: Filtrar los recursos históricos por la fecha en que estos ocurrieron.

RF 15 Consultar términos del glosario: Visualizar las palabras del glosario en un escenario diferente con filtros para acceder de una forma más intuitiva a los mismos. (Este requisito estaba contenido en la primera versión pero dentro de la consulta de los elementos estructurales)

RF 16 Sugerir criterios de evaluación: Sugerir nuevos criterios de evaluación con respecto a los recursos en determinados momentos, de acuerdo al objetivo que persiga el profesor con respecto al mismo.

- **RF 16.1 Modificar criterio de evaluación:** Modificar los datos de un criterio de evaluación.

- **RF 16.2 Eliminar criterio de evaluación:** Eliminar un criterio de evaluación.
- **RF 16.3 Mostrar criterio de evaluación:** Visualizar los datos de un criterio de evaluación.
- **RF 16.4 Filtrar criterio de evaluación:** Filtrar los criterios de evaluación de acuerdo al nombre y tipo.
- **RF 16.5 Listar criterios de evaluación:** Listar los criterios de evaluación mostrando por cada uno nombre, tipo y descripción.

RF 17 Filtrar los recursos mostrados de acuerdo a diversos criterios: Restringir los resultados de una búsqueda de acuerdo con el criterio que se especifique en el filtro, por ejemplo, todos los recursos de un tipo determinado o que pertenezcan a un capítulo tema o subtema determinado. (Este requisito está contenido en la primera versión y será redefinido)

RF 18 Salvar una copia del recurso de acuerdo con los requisitos de derecho de autor: 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.

2.4.2. Requerimientos No Funcionales

Los requerimientos no funcionales (en lo adelante RNF) son propiedades o cualidades que el sistema debe tener. Representan las características del producto que se va a desarrollar. A continuación se muestran los ya definidos por la arquitectura del proyecto para el desarrollo de la primera versión de la plataforma educativa Zera, los cuales no sufrirán cambios para la nueva propuesta.

RNF de Software:

Las computadoras deben tener instalado un navegador web:

- Mozilla Firefox 3.6 o superior.
- Internet Explorer 7.0 o superior.
- Chrome 10 o superior.

- Opera 10.6 o superior.
- Safari 5 o superior.

RNF de Hardware:

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

RNF de Diseño e Implementación:

- Lenguaje de programación: PHP 5.3.
- El marco de trabajo base de desarrollo que se utilizará es: Symfony 1.4.15.
- Como IDE se empleará NetBean 7.0.1.
- Como servidor Web se utilizará Apache 2.2
- El SGBD deberá ser PostgreSQL 9.1

RNF de Apariencia o Interfaz Externa:

- El diseño de las interfaces debe ser sugerente al usuario.
- El sistema proporcionará claridad y correcta organización de la información, permitiendo la interpretación correcta e inequívoca de ésta.
- Debe contener un diseño sencillo, con pocas imágenes y gráficos para acelerar la velocidad de respuesta del sistema.

RNF de Seguridad y Confiabilidad:

- Restringir el acceso a usuarios no autorizados para que no se pueda realizar cualquier acción sobre el contenido de la plataforma.
- Seguridad de acceso y administración de usuarios: otorgamiento de privilegios y roles, asignación de perfiles. Los niveles de acceso están determinados por los diferentes roles válidos dentro de la plataforma.

- Garantizar que la información sea editada únicamente por quien está autorizado y posea permisos para ello.
- Verificación sobre acciones irreversibles (mensajes de eliminaciones).

RNF de Usabilidad:

- El sistema podrá ser usado por cualquier persona que posea conocimientos básicos del manejo de la computadora y de un ambiente web en sentido general.
- El sistema debe tener acceso al menú general desde cualquiera de sus páginas.
- Se deben mostrar las rutas de acceso según la navegación que tenga el usuario.
- Las rutas de acceso deben tener vínculos a las secciones que muestran.
- Los elementos gráficos como los íconos deberán contar con un mensaje flotante que señalen el tipo de recurso al que se refiere.

RNF de Portabilidad:

- El sistema podrá ser utilizado bajo cualquier Sistema Operativo.

RNF Funcionalidad:

- Reducir al mínimo el tiempo en que carga la plataforma.

RNF Rendimiento:

- Tiempos de respuestas rápidas al igual que la velocidad de procesamiento de la información.

RNF de Disponibilidad:

- Los usuarios del sistema deben tener acceso (según sus permisos) en todo momento a la información solicitada.

2.5. Modelo de Casos de Uso del sistema.

2.5.1. Actores

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

Docente: Consulta materiales, incluyendo los mejor valorados por los estudiantes. Puede valorar, descargar recurso, gestionar recursos favoritos, realizar anotaciones a un recurso y gestionar preguntas y respuestas directas.

Estudiante: Consulta materiales. Puede valorar, descargar recursos, gestionar recurso favorito, realizar anotación a un recurso así como gestionar preguntas y respuestas directas.

Administrador Local: Consulta materiales.

Tutor: Consulta y evalúa los materiales.

Gestor biblioteca: Gestiona las preguntas y respuestas que pueden realizar estudiantes y profesores referentes a un tema, los recursos favoritos de los usuarios y las notas de los mismos sobre un recurso. Usuario que generaliza al Estudiante y Docente.

Evaluador: Evalúa los recursos que son puestos a su disposición. Usuario que generaliza al Gestor biblioteca y el Tutor.

Usuario bachiller: Consulta los materiales existentes en la Biblioteca y el glosario de términos. Rol que generaliza las acciones del Evaluador, el Administrador local y el Director.

Editor: Encargado de gestionar el contenido de las materias que se imparten en la escuela.

2.5.2. Patrones de casos de uso

Durante la estructuración del modelo de casos de uso, se tuvo en cuenta la aplicación de patrones de casos de uso, buenas prácticas recomendadas por varios autores. La aplicación de patrones a un modelo de casos de uso permite una descripción más profunda de la interacción actor-sistema, así como la creación de una estructura fácilmente entendible por los clientes.

Su aplicación trae los siguientes beneficios:

- Aumentar la productividad.
- Reutilizar elementos existentes.
- No invertir tiempo en resolver problemas ya resueltos.

- Aplicar la teoría al trabajo práctico.
- Habilitar las herramientas de soporte para modelar el desarrollo.

Una vez obtenido el modelo de casos de uso inicial, se decidió aplicar sobre el mismo algunos patrones de caso de uso. Estos permitieron definir nuevos casos de uso en función de crear un modelo más sencillo y estructurado.

Entre los patrones aplicados se encuentran los siguientes:

CRUD(Create, Read, Update, Delete) Completo: Consiste en un caso de uso para administrar la información, permite modelar las diferentes operaciones para administrar una entidad de información, tales como crear, leer, modificar y eliminar o dar de baja. (28)

Ejemplo de su uso, se muestra en la siguiente figura:

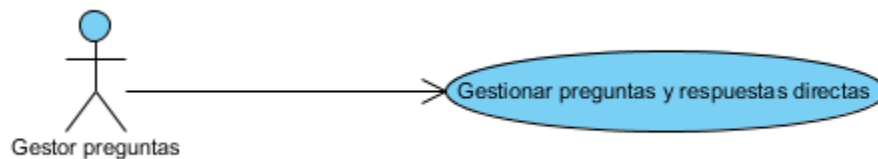


Figura 2. CRUD Completo

Múltiples actores

Roles comunes: Puede suceder que los dos actores jueguen el mismo rol sobre el caso de uso. 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.

Ejemplo de su uso, se muestra en la siguiente figura:

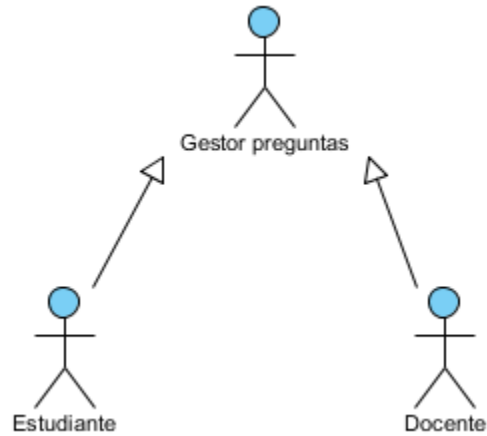


Figura 3. Múltiples actores_Roles comunes

Extensión Concreta: Es un patrón que involucra dos CU y una relación de extensión, donde el CU extendido puede ser instanciado por su cuenta o por el CU base.

Ejemplo de su uso, se muestra en la siguiente figura:

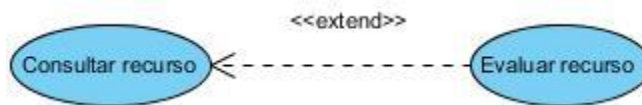


Figura 4. Patrón Extensión Concreta

2.5.3. Diagrama de actores

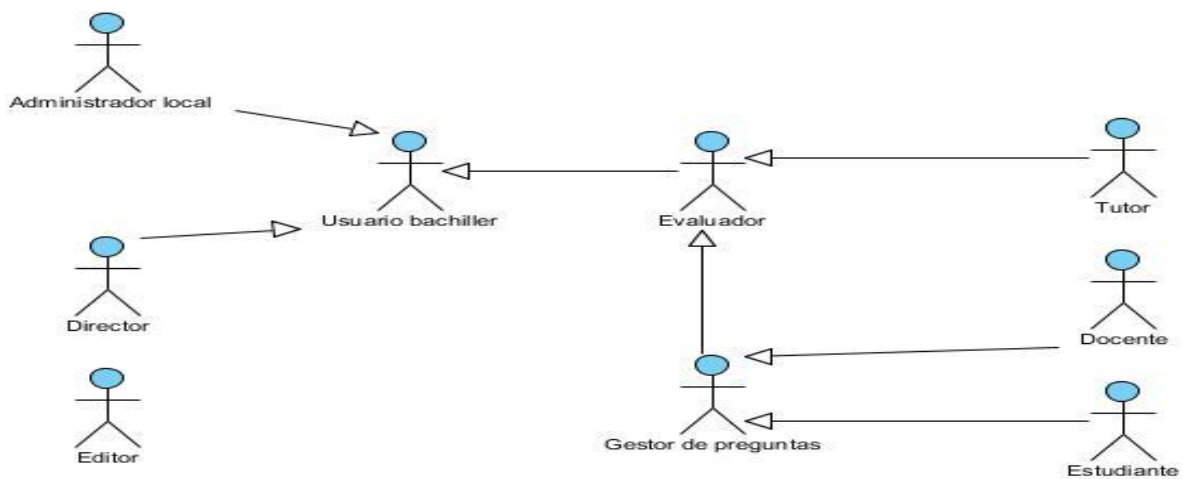


Figura 5. Diagrama de actores.

2.5.4. Diagrama de Casos de Uso del sistema

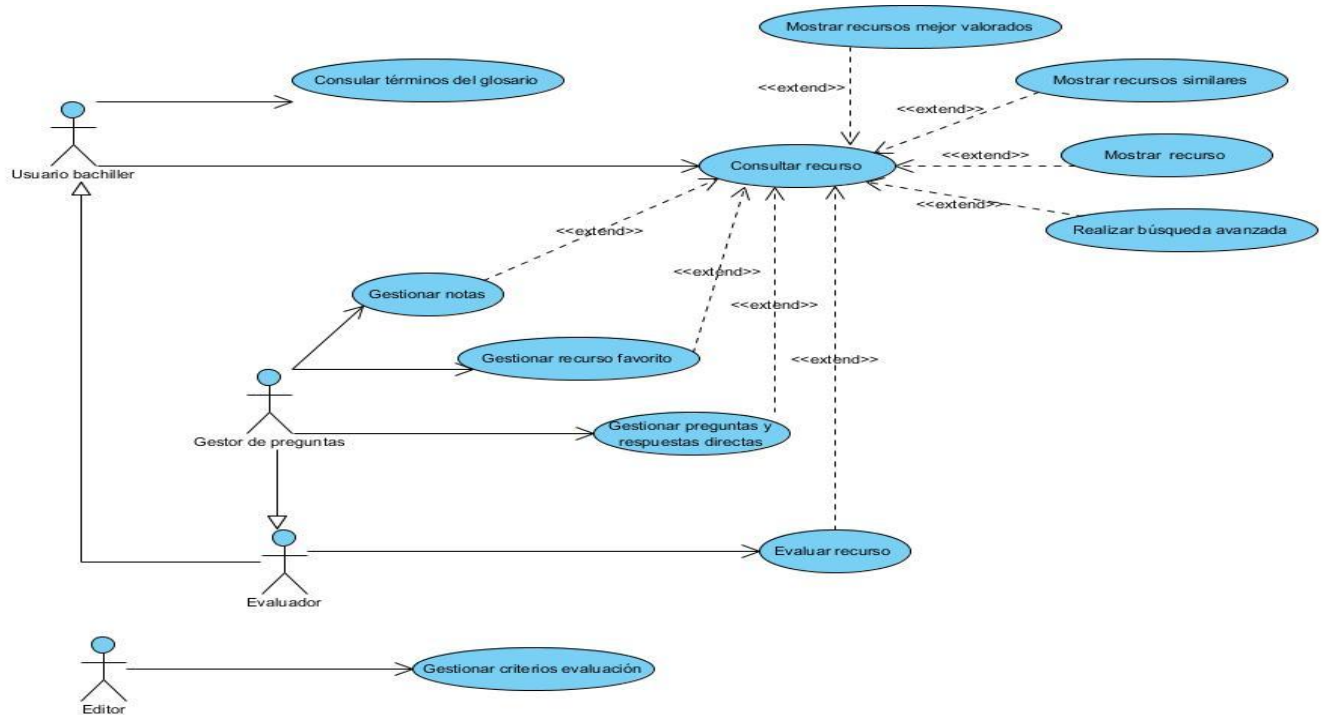


Figura 6. Diagrama de casos de uso del sistema.

2.5.5. Descripción de los casos de uso del sistema

Se muestra la descripción del CU Evaluar Recurso.

Nombre	Evaluar recurso
Objetivo	Evaluar un recurso de acuerdo a una serie de criterios de evaluación cualitativos a través de un peso cuantitativo.
Actor	Evaluador (Inicia): Rol que generaliza al Gestor de preguntas y al tutor, los cuales evalúan los recursos puestos a su disposición en la biblioteca.
Resumen	El caso de uso se inicia cuando el Evaluador ha consultado un recurso y selecciona la opción que le permite evaluar el recurso. El sistema permite evaluar el recurso de acuerdo a una escala numérica del uno al

Capítulo II: Propuesta de solución

	cinco. El Evaluador selecciona un elemento que corresponde a un número. El sistema actualiza la evaluación del recurso. El caso de uso termina.
Complejidad	Media.
Nivel	Usuario
Precondición	Debe haberse generado el escritorio de trabajo del usuario autenticado. Debe haber seleccionado una materia. Debe haber accedido a la biblioteca. Debe haber consultado un recurso.
Poscondición	Se evaluó un recurso.
Flujo de eventos	

Flujo básico

Acciones del actor	Respuesta del sistema
1. Accede a la opción que le permite realizar una acción sobre la evaluación de un recurso.	2. Brinda la posibilidad de : <ul style="list-style-type: none"> • Evaluar recurso. <p>Por cada recurso brinda la posibilidad de emitir una evaluación, esta puede realizarse de 2 formas la primera mediante una evaluación rápida y la segunda mediante diferentes criterios. Estas se registran en una tabla para diferenciar las evaluaciones positivas de las negativas</p> <ul style="list-style-type: none"> • 0 -3 • 1 -2 • 2 -1 • 3 - 0 • 4 - 1 • 5 - 2 <p>Cada vez que se hace una evaluación se actualiza la ponderación total del recurso, que consiste en sumar todos los registros de evaluaciones de todos los usuarios por todos los criterios lo que va a dar positivo o negativo en función de lo que hayan evaluado, a esto</p>

Capítulo II: Propuesta de solución

además se le procesa y se le suma el promedio de evaluaciones simples. Cuando se habla de procesar se refiere a pasar por la tablita antes descrita

- Modificar evaluaciones previas del recurso. Ver Sección 1: “Modificar datos del recurso evaluado”

Y permite:

- Seleccionar la opción de evaluar el recurso en cuanto a un criterio propuesto.
- Seleccionar la opción de evaluar el recurso por diferentes criterios, pedagógicos, diseño gráfico o aspectos generales
- Cancelar la operación en cualquier momento.

3. Seleccionar la opción de evaluar el recurso en cuanto a un criterio propuesto.

4. Muestra el criterio propuesto para que el usuario emita su opinión, la cual va a sombreadarse de color rojo, mediante la puntuación dada por el mismo

Y permite:

- Evaluar el recurso por el criterio propuesto
- Cancelar la operación en cualquier momento.

5. Evaluar el recurso por el criterio propuesto

6. Muestra la evaluación emitida por el usuario sobre el recurso por un criterio propuesto, de color rojo la estrella de la puntuación dada.

Y permite:

En caso de que desee continuar evaluando el recurso en dependencia de los permisos del usuario.

- Seleccionar la opción de evaluar el recurso por diferentes criterios, pedagógicos, diseño gráfico o aspectos generales.

Capítulo II: Propuesta de solución

	<ul style="list-style-type: none"> • Seleccionar la opción de consultar los datos del recurso. <u>Ver CU Consultar recurso.</u>
7. Seleccionar la opción de evaluar el recurso por diferentes criterios, pedagógicos, generales etc.	
	<p>8. Muestra los criterios de evaluación para que el usuario realice la continuación de la misma, no necesariamente tiene que marcar todos los criterios solo evaluará por el o los criterios que estime conveniente.</p> <p>Y permite</p> <ul style="list-style-type: none"> • Evaluar el recurso. • Cancelar la operación en cualquier momento.
9. Evalúa el recurso y guarda la evaluación dada.	
	<p>10. Muestra un mensaje de información con la evaluación resultante.</p> <p>Y permite</p> <ul style="list-style-type: none"> • Aceptar la evaluación resultante. • Cancelar evaluación del recurso.
11. Acepta la evaluación resultante.	
	12. Actualizar la evaluación del recurso y muestra el resultado de la misma.
	13. El caso de uso termina.

Flujo alterno

2. a El actor selecciona la opción de modificar datos del recurso evaluado.	
Acciones del actor	Respuesta del sistema
	2.a.1 Modificar evaluaciones previas del recurso Ver Sección 1: “Modificar datos de recurso evaluado”
	2. a.2 El caso de uso termina.
2. b El actor seleccionar la opción de evaluar el recurso por diferentes criterios de evaluación	
Acciones del actor	Respuesta del sistema
	2. b.1 Regresa al punto 8 del Flujo Básico

Capítulo II: Propuesta de solución

2.c El actor selecciona la opción de cancelar la operación en cualquier momento	
Acciones del actor	Respuesta del sistema
	2. c.1 Elimina los datos creados.
	2. c.2 Muestra un mensaje de información “La acción ha sido cancelada”.
	2. c.3 Regresa a la vista anterior.
	2. c.4 El caso de uso termina.

4.a El actor selecciona la opción de cancelar la operación en cualquier momento	
Acciones del actor	Respuesta del sistema
	4. a.1 Elimina los datos creados.
	4. a.2 Muestra un mensaje de información “La acción ha sido cancelada”.
	4. a.3 Regresa a la vista anterior.
	4. a.4 El caso de uso termina.

6.a El actor selecciona la opción de consultar los datos del recurso	
Acciones del actor	Respuesta del sistema
	8. a.1 <u>Ver CU Consultar recurso.</u>
	8. a.2 El caso de uso termina.

8. a El actor selecciona la opción de consultar los datos del recurso.	
Acciones del actor	Respuesta del sistema
	10. a.1 Elimina los datos creados.
	10. a.2 Muestra un mensaje de información “La acción ha sido cancelada”.
	10. a.3 Regresa a la vista anterior.
	10. a.4 El caso de uso termina.

10.a El actor selecciona la opción de cancelar la evaluación del recurso	
Acciones del actor	Respuesta del sistema
	10. a.1 Elimina los datos creados.
	10. a.2 Muestra un mensaje de información “La acción ha sido cancelada”.
	10. a.3 Regresa a la vista anterior.

Sección 1: “Modificar datos del recurso evaluado”

Flujo básico

Acciones del actor	Respuesta del sistema
1. Accede a modificar los datos de un recurso evaluado.	2. Brinda la posibilidad de modificar las evaluaciones dadas sobre el recurso, de la forma deseada si decide cambiar la evaluación del criterio propuesto, no necesariamente tiene que cambiar la evaluación de los demás criterios y así viceversa. Y permite: <ul style="list-style-type: none">• Seleccionar la opción de modificar evaluación mediante un criterio propuesto.• Seleccionar la opción de modificar evaluación mediante diferentes criterios.• Cancelar la operación en cualquier momento.
3. Seleccionar la opción de modificar evaluación mediante un criterio propuesto.	4. Muestra la evaluación emitida por el usuario sobre el recurso por un criterio propuesto, de color rojo la estrella de la puntuación dada. Y permite: <ul style="list-style-type: none">• Modificar evaluación del recurso mediante el criterio propuesto.• Cancelar la operación en cualquier momento.
5. Modifica evaluación del recurso	6. Muestra la evaluación emitida por el usuario sobre el recurso Y permite: En caso de que desee continuar modificando la evaluación del recurso en dependencia de los permisos del usuario <ul style="list-style-type: none">• Seleccionar la opción de modificar evaluación del recurso por diferentes criterios de evaluación.• Seleccionar la opción de consultar los datos

Capítulo II: Propuesta de solución

	del recurso. <u>Ver CU Consultar recurso.</u>
7. Seleccionar la opción de modificar evaluación del recurso por diferentes criterios de evaluación.	8 Muestra los criterios de evaluación para que el usuario realice la modificación de la misma, no necesariamente tiene que marcar todos los criterios solo modificará por el o los criterios que estime conveniente. <ul style="list-style-type: none"> • Pedagógicos • Diseño grafico • Aspectos generales Y permite <ul style="list-style-type: none"> • Modificar evaluación del recurso. • Cancelar la operación en cualquier momento.
9. Modifica la evaluación del recurso.	10. <i>Muestra un mensaje de información con la evaluación resultante.</i> Y permite <ul style="list-style-type: none"> • Aceptar la evaluación resultante. • Cancelar evaluación del recurso.
11. Acepta la evaluación resultante.	12 <i>Actualizar la nueva evaluación del recurso y muestra el resultado de la misma.</i> 13 El caso de uso termina.

Flujo alterno

2. a El actor seleccionar la opción de modificar evaluación mediante diferentes criterios.	
Acciones del actor	Respuesta del sistema
	2.a.1 Regresa al paso 8 del flujo básico
2. b El actor selecciona la opción de cancelar la operación en cualquier momento.	
Acciones del actor	Respuesta del sistema
	2. b.1 <i>Elimina los datos modificados.</i>
	2.b.2 Muestra un mensaje de información
	2. b.3 Regresa a la vista anterior.
	2. b.4 El caso de uso termina.
4. a El actor selecciona la opción de cancelar la operación en cualquier momento.	

Capítulo II: Propuesta de solución

Acciones del actor	Respuesta del sistema
	4. a.1 <i>Elimina los datos modificados.</i>
	4. a.2 Muestra un mensaje de información
	4. a.3 Regresa a la vista anterior.
	4. a.4 El caso de uso termina.
6. a El actor selecciona la opción de consultar los datos del recurso.	
Acciones del actor	Respuesta del sistema
	6. a.1 <u>Ver CU Consultar recurso.</u>
	6. a.2 El caso de uso termina.
8. a El actor selecciona la opción de cancelar la operación en cualquier momento.	
Acciones del actor	Respuesta del sistema
	8. a.1 Elimina los datos creados.
	8. a.2 Muestra un mensaje de información “La acción ha sido cancelada”.
	8. a.3 Regresa a la vista anterior.
	8. a.4 El caso de uso termina.
10.a El actor selecciona la opción de cancelar la evaluación del recurso	
Acciones del actor	Respuesta del sistema
	10. a.1 Elimina los datos creados.
	10. a.2 Muestra un mensaje de información “La acción ha sido cancelada”.
	10. a.3 Regresa a la vista anterior.
	10. a.4 El caso de uso termina.

Tabla 1: Descripción del CU Evaluar recurso.

Para ver el resto de las descripciones de CU realizadas consultar Anexos.

2.6. Conclusiones parciales

Con la realización de este capítulo se logra desarrollar un análisis crítico de los procesos involucrados en el campo de acción. Se identificaron los principales conceptos asociados al dominio del problema para conformar el modelo de objetos del dominio. Se definió la propuesta de solución del sistema. Se obtiene un listado de 36 RF y 10 RNF

Capítulo II: Propuesta de solución

que serán las capacidades y condiciones que el sistema debe tener, donde se recogen las principales necesidades del sistema a modelar. Estas necesidades fueron traducidas a un conjunto de 11 CU, los cuales representan las principales funcionalidades del sistema. Además se definieron los patrones de caso de uso aplicados a la solución propuesta. Partiendo de este punto, se puede comenzar la construcción de la propuesta de solución, teniendo en cuenta el cumplimiento de cada requerimiento.

Capítulo III: Análisis y Diseño

3.1. Introducción

Tomando el diagrama de casos de uso como punto de partida en este capítulo se mostrará la realización a través de clases del análisis, del diseño y colaboración del módulo Biblioteca. Se describen los artefactos generados a partir del seguimiento de las actividades del flujo de trabajo Análisis y Diseño, para el CU Evaluar recurso tomado como muestra. 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.

3.2. Análisis

En la disciplina de Análisis y Diseño, el principal objetivo 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.

3.2.1. Modelo de análisis

El modelo de análisis es un modelo conceptual, ya que en él se identifican una serie de clases y relaciones. Tiene la finalidad de refinar los casos de uso más detalladamente. Posibilita un conocimiento razonable del sistema o producto, ayuda a refinar y estructurar los requisitos, permite entender mejor los aspectos internos del sistema, sus diagramas ofrecen una mayor formalización y poder expresivo. Es un Modelo de Objetos que describe la realización de CU y sirve como idealización para el Modelo del Diseño.

3.2.2. 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. Los no funcionales se reservan para las actividades del diseño e implementación.

Seguidamente se explica de forma breve cada uno de los estereotipos presentes en los diagramas de clases del análisis (en lo adelante DCA).

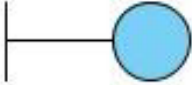

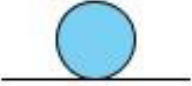
Nombre	Características	Estereotipos
Interfaz	Modelan la interacción entre el sistema y sus actores.	 CI_Nombre
Control	Coordinan la realización de uno o unos pocos casos de uso relacionando las actividades de los objetos que implementa la funcionalidad del caso de uso.	 CC_Nombre
Entidad	Modelan información que posee larga vida y que es a menudo persistente.	 CE_Nombre

Tabla 2. Descripción de estereotipos presentes en los DCA.

3.2.3. Diagramas de clases del análisis

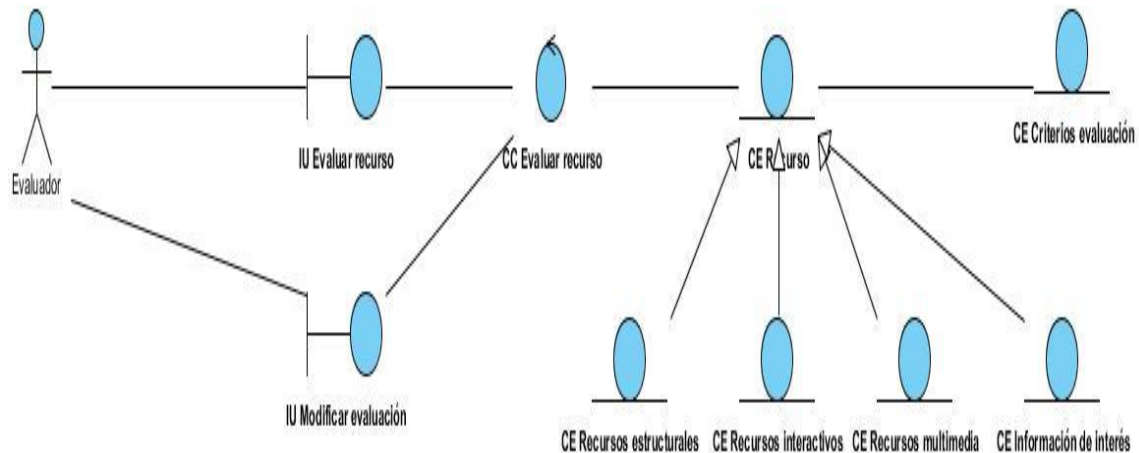


Figura 7. Diagrama de clases del análisis CU Evaluar recurso.

Para ver el resto de los diagramas de clases del análisis realizados consultar Anexos.

3.2.4. Diagramas de interacción.

- Diagramas de colaboración.

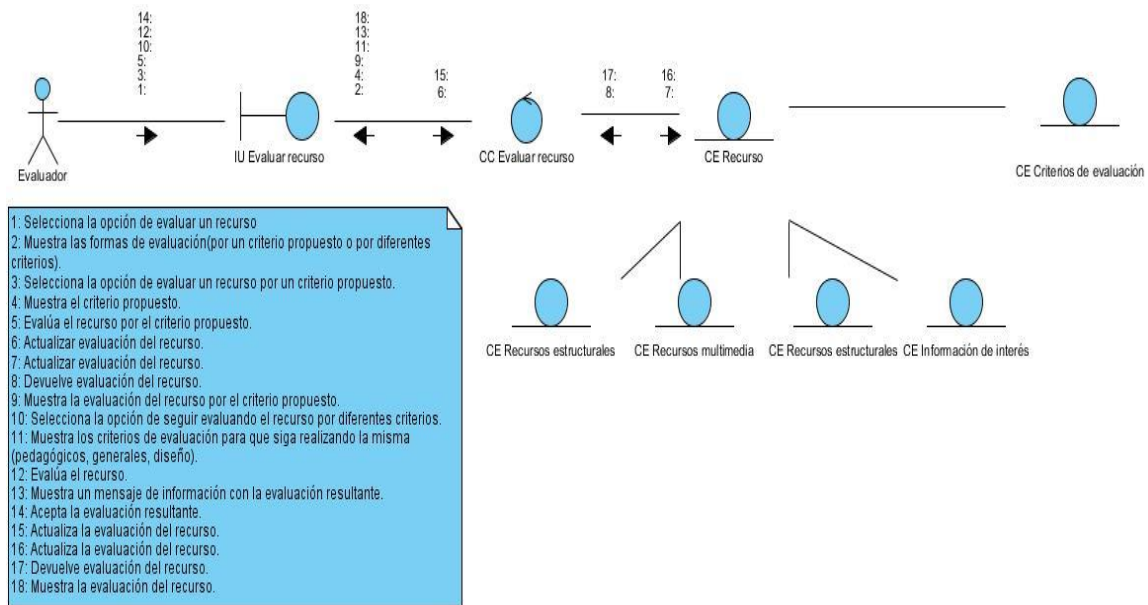


Figura 8. Diagrama de colaboración CU Evaluar recurso: “Evaluar recurso”.

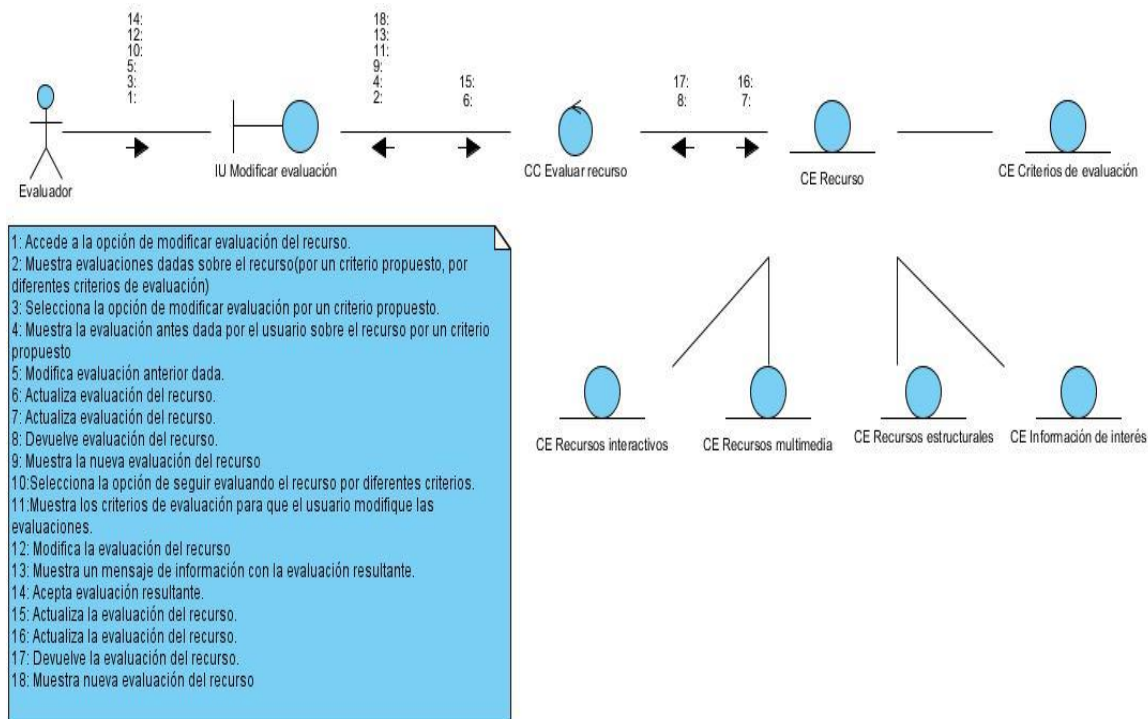
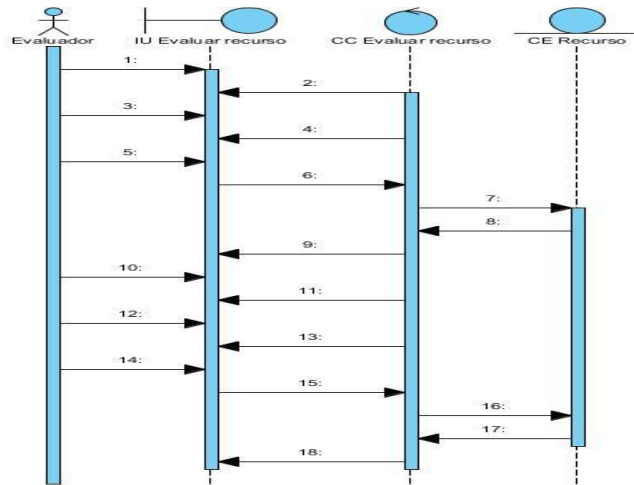


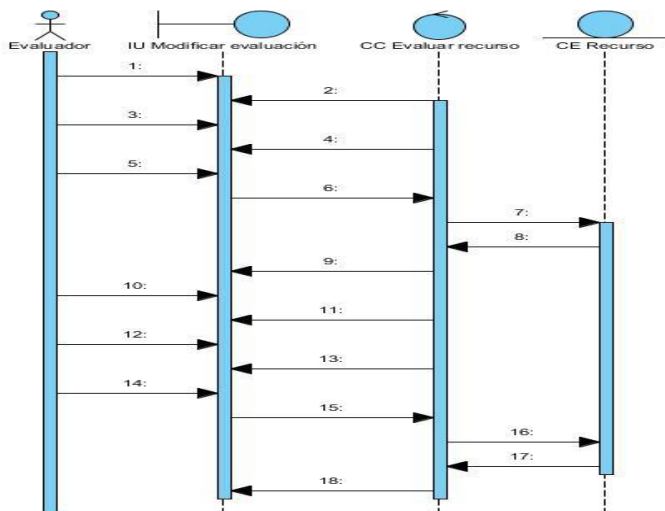
Figura 9. Diagrama de colaboración CU Evaluar recurso Sección 1: “Modificar evaluación del recurso”

- Diagramas de secuencia



- 1: Selecciona la opción de evaluar un recurso
- 2: Muestra las formas de evaluación(por un criterio propuesto o por diferentes criterios).
- 3: Selecciona la opción de evaluar un recurso por un criterio propuesto.
- 4: Muestra el criterio propuesto.
- 5: Evalúa el recurso por el criterio propuesto.
- 6: Actualizar evaluación del recurso.
- 7: Actualizar evaluación del recurso.
- 8: Devuelve evaluación del recurso.
- 9: Muestra la evaluación del recurso por el criterio propuesto.
- 10: Selecciona la opción de seguir evaluando el recurso por diferentes criterios.
- 11: Muestra los criterios de evaluación para que siga realizando la misma(pedagógicos, generales, diseño).
- 12: Evalúa el recurso.
- 13: Muestra un mensaje de información con la evaluación resultante.
- 14: Acepta la evaluación resultante.
- 15: Actualiza la evaluación del recurso.
- 16: Actualiza la evaluación del recurso.
- 17: Devuelve evaluación del recurso.
- 18: Muestra la evaluación del recurso.

Figura 10. Diagrama de secuencia CU Evaluar recurso: “Evaluar recurso”.



- 1: Accede a la opción de modificar evaluación del recurso.
- 2: Muestra evaluaciones dadas sobre el recurso(por un criterio propuesto, por diferentes criterios de evaluación)
- 3: Selecciona la opción de modificar evaluación por un criterio propuesto.
- 4: Muestra la evaluación antes dada por el usuario sobre el recurso por un criterio propuesto
- 5: Modifica evaluación anterior dada.
- 6: Actualiza evaluación del recurso.
- 7: Actualiza evaluación del recurso.
- 8: Devuelve evaluación del recurso.
- 9: Muestra la nueva evaluación del recurso
- 10: Selecciona la opción de seguir evaluando el recurso por diferentes criterios.
- 11: Muestra los criterios de evaluación para que el usuario modifique las evaluaciones.
- 12: Modifica la evaluación del recurso
- 13: Muestra un mensaje de información con la evaluación resultante.
- 14: Acepta evaluación resultante.
- 15: Actualiza la evaluación del recurso.
- 16: Actualiza la evaluación del recurso.
- 17: Devuelve la evaluación del recurso.
- 18: Muestra nueva evaluación del recurso

Figura 11. Diagrama de secuencia CU Evaluar recurso Sección 1: “Modificar evaluación del recurso”.

Para ver el resto de los diagramas de interacción realizados consultar Anexos.

3.2.5. Patrón arquitectónico Modelo – Vista – Controlador en Symfony.

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.

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

La elección de la arquitectura determina la definición de los artefactos del modelo de diseño, que es 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 que se utilizará.

3.3.2. Diagrama de clases del diseño.

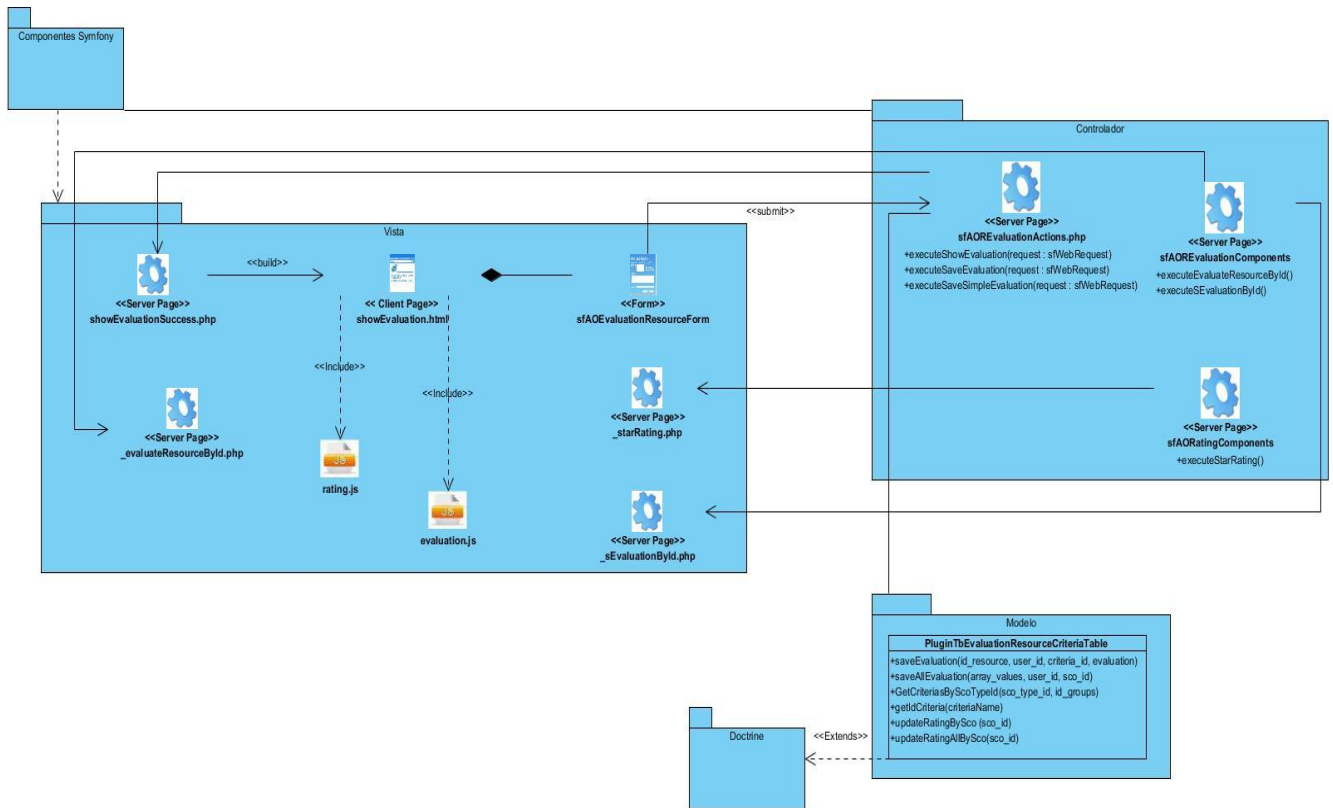


Figura 12. Diagrama de clases del diseño CU Evaluar recurso

Para ver el resto de los diagramas de clases del diseño realizados consultar Anexos.

3.4. 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 pero del CU Evaluar recurso tomado como ejemplo.

- Diagrama Entidad-Relación.

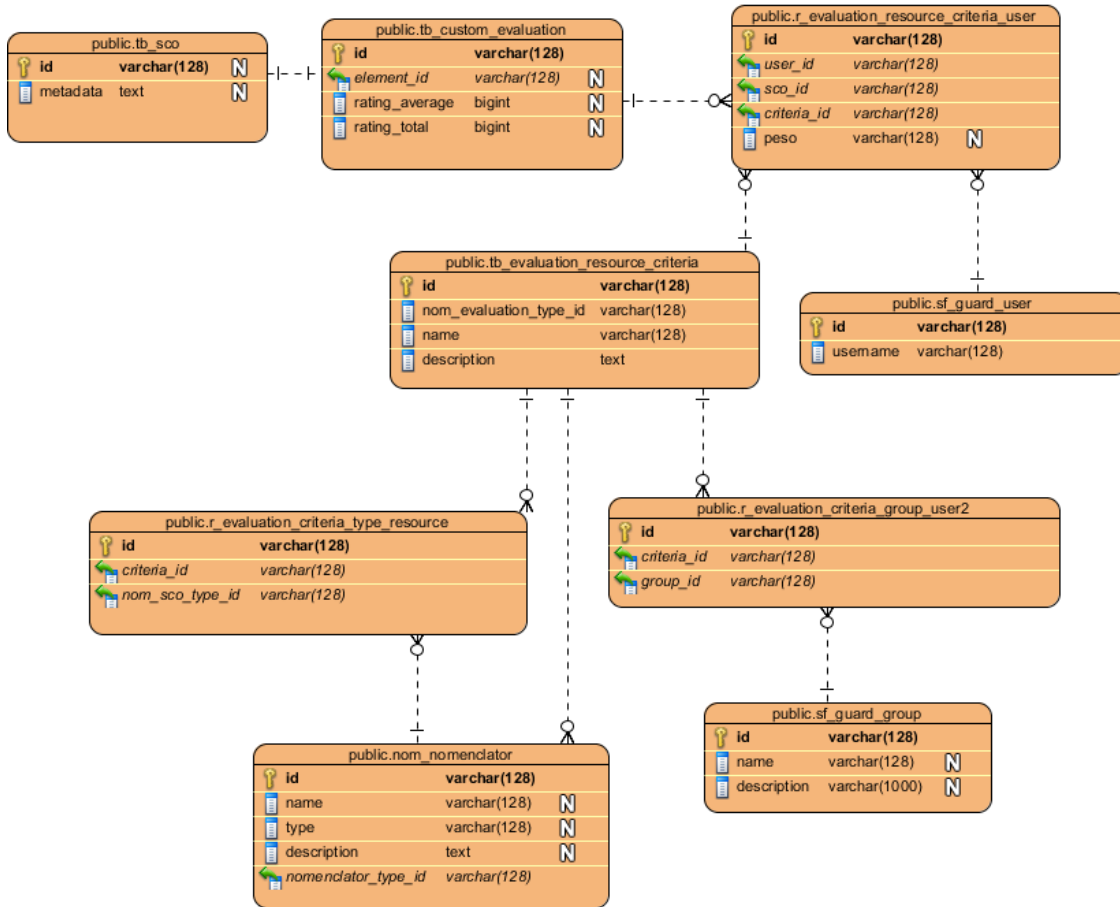


Figura 13. Diagrama Entidad-Relación Evaluar recurso

Para ver el resto de los diagramas de Entidad-Relación realizados consultar Anexos.

3.5. Diagrama de Despliegue

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

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

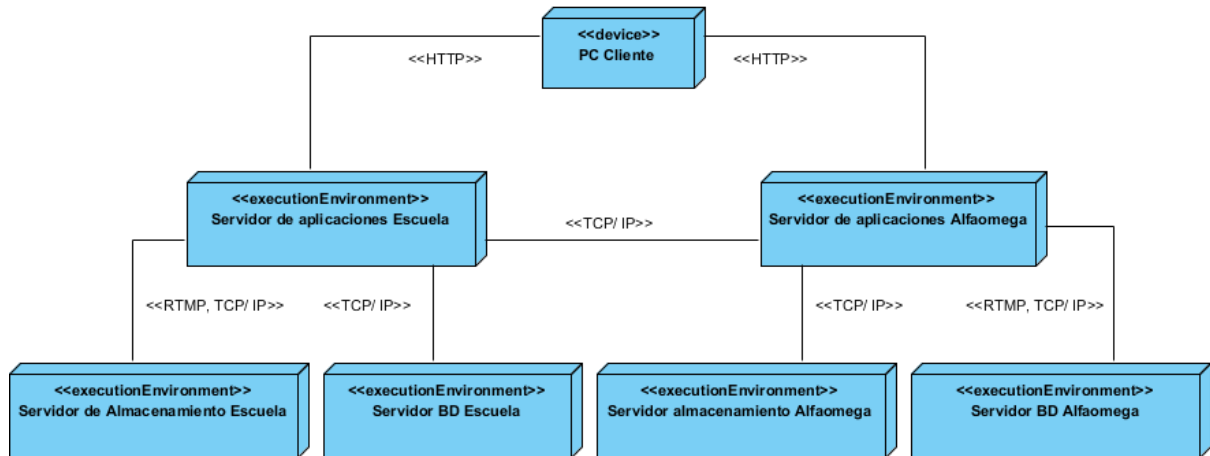


Figura 14. Diagrama de Despliegue.

3.6. Conclusiones parciales

En este capítulo quedó definido el análisis y diseño del sistema así como sus relaciones, describiendo las mismas en términos de diagramas de clase, enfocados al cumplimiento de los RF del sistema. Se realizó una valoración de los patrones de diseño teniendo en cuenta las características del sistema. Se modeló el diseño de la base de datos.

Todo esto proporciona las bases o esquema para la implementación del sistema propuesto.

Capítulo IV: Implementación y Prueba

4.1. Introducción

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

4.2. Modelo de implementación

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

Constituye la entrada principal de las etapas de prueba que siguen a la implementación. Concretamente, durante la etapa de prueba cada construcción generada en la implementación es sometida a pruebas de integración, y posiblemente también a pruebas de sistema.

- Diagrama de componentes

Los diagramas de componentes son usados para estructurar el modelo de implementación en término de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación. El uso más importante de estos diagramas es mostrar la estructura de alto nivel del modelo de implementación, especificando que:

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

Se utilizan para modelar la vista estática de un sistema. Muestran la organización y las dependencias lógicas entre un conjunto de componentes de software, sean estos componentes de código fuente, librerías, binarios o ejecutables. (29)

A continuación se presenta el diagrama de componentes propuesto para el CU Evaluar recurso:

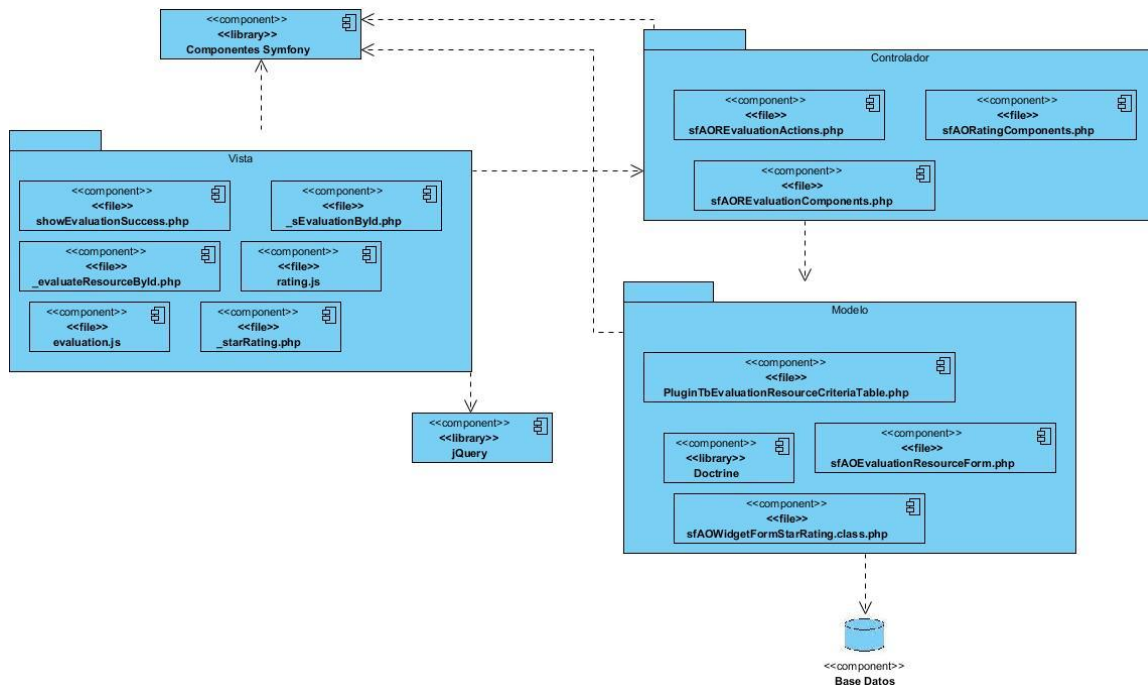


Figura 15. Diagrama de Componentes CU Evaluar recursos.

Para ver el resto de los diagramas de Componentes realizados consultar Anexos.

4.3. Pruebas de software

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

El objetivo de las pruebas al sistema es comprobar la integración del sistema de información globalmente, verificando el funcionamiento correcto de las interfaces entre los distintos subsistemas que lo componen y con el resto de los sistemas de información con los que se comunica. En la realización de estas pruebas, es importante comprobar la cobertura de los requisitos, dado que su incumplimiento puede comprometer la aceptación del sistema por el equipo de operación responsable de realizar las pruebas de implantación del sistema, que se llevarán a cabo en el proceso Implantación y Aceptación del Sistema. (30)

4.3.1. Niveles de Prueba

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

- **Pruebas Unitarias:** Comienzan con la prueba de cada módulo. Una prueba unitaria es una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de estos funcione correctamente por separado. El objetivo de las pruebas unitarias es aislar cada parte del programa y mostrar que las partes individuales son correctas. Proporcionan un contrato escrito que el fragmento de código debe satisfacer. Estas pruebas aisladas proporcionan cinco ventajas básicas: fomentan el cambio, simplifican la integración, documentan el código, separan la interfaz del código y hacen que los errores estén más acotados y sean fáciles de localizar. (15)
- **Pruebas de Integración:** A partir del esquema del diseño, los módulos probados se vuelven a probar combinados para probar sus interfaces. Pruebas integrales o pruebas de integración son aquellas que se realizan en el ámbito del desarrollo de software una vez que se han aprobado las pruebas unitarias. Únicamente se refieren a la prueba o pruebas de todos los elementos unitarios que componen un proceso, hecha en conjunto, de una sola vez. Consiste en realizar pruebas para verificar que un gran conjunto de partes del software funcionan juntos. (15)
- **Pruebas del Sistema:** El software ensamblado totalmente con cualquier componente hardware que requiera, se prueba para comprobar que se cumplen los requisitos funcionales. Cualquier pieza de software completo, desarrollado o adquirido, puede verse como un sistema que debe probarse, ya sea para decidir acerca de su aceptación, para analizar defectos globales o para estudiar aspectos específicos de su comportamiento, tales como seguridad o rendimiento. Este tipo de pruebas estudia el producto completo. (15)

4.3.2. Métodos de Prueba

RUP propone dos métodos fundamentales: caja blanca y caja negra. A continuación, se describen ambos métodos, haciéndose mayor énfasis en las pruebas de caja negra ya

que fueron las más utilizadas en la comprobación de la solución.

4.3.3. Pruebas de Caja Negra

Las pruebas de caja negra, también denominadas pruebas de comportamiento, se centran en los requisitos funcionales del software. O sea, permiten al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Estas pruebas no son una alternativa a las técnicas de pruebas de caja blanca, más bien se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores que se llevan a cabo sobre la interfaz del software. El objetivo es demostrar que las funciones del software son operativas, que las entradas se aceptan de forma adecuada y se produce un resultado correcto, y que la integridad de la información externa se mantiene, saber qué es lo que hace el software pero sin entrar en detalles de código, es decir, que es lo que hace, y no cómo lo hace (no se ve el código). Estas pruebas se centran principalmente en los requisitos funcionales del software y permiten encontrar: (31)

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las bases de datos externas.
- Errores de rendimiento
- Errores de inicialización y terminación. (30)

4.3.4. Pruebas de Caja Blanca

La prueba de caja blanca, denominada a veces prueba de caja de cristal es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante los métodos de prueba de caja blanca, el ingeniero del software puede obtener casos de prueba que garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo, ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa; ejecuten todos los ciclos en sus límites y con sus límites operacionales, y ejerciten las estructuras internas de datos para asegurar su validez. (31)

4.3.5. Diseño de Casos de Pruebas

A partir de la descripción de los 11 casos de usos de la investigación se generaron los

diseños de casos de pruebas (CP) correspondientes, que constituyen la guía principal del probador. Se confeccionó cada caso de prueba con la identificación de los principales escenarios con sus posibles entradas. Para ver los diseños de casos de pruebas consultar Anexos.

4.3.6. Resultados Obtenidos

Durante el desarrollo del módulo Biblioteca se realizaron pruebas unitarias para comprobar el funcionamiento del software. Estas no se planificaron ni se registraron sus resultados, se realizaron de forma simultánea a la implementación. Para evaluar la solución, se efectuaron varias iteraciones donde se probó el software íntegramente. Se prestó gran atención a las pruebas de integración con vista a probar funcionalidades que relacionan al módulo Biblioteca con el resto de los módulos de la plataforma educativa Zera. Se especifican las principales pruebas realizadas a la aplicación haciendo uso del método de caja negra, estas pertenecen al nivel de prueba de sistema y se detallan a continuación.

- **Pruebas Internas:** Fueron realizadas por el equipo de desarrollo del proyecto (en este caso los analistas) con el fin de entregar un producto con la menor cantidad de errores posibles. Se centraron en el cumplimiento de las funcionalidades descritas en el listado de requerimientos y de casos de uso del sistema, además, se revisó toda la documentación.
- **Pruebas Cruzadas:** Fueron realizadas al sistema por el equipo de desarrollo del proyecto (analistas y desarrolladores de los diferentes módulos) con el fin de encontrar la mayor cantidad de errores posibles en término de validaciones, pautas definidas por arquitectura de información, formato de los campos, entre otras.
- **Pruebas de Liberación:** Pruebas realizadas por el equipo de calidad interno del proyecto (en este caso los analistas), encargados de validar que el software cuenta con la calidad requerida para ser liberado.

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

Pruebas Internas:

Capítulo IV: Implementación y Prueba

Módulo	Casos de Uso	Iteración	NC	Cerrada(S)	No procede
Biblioteca	11	1ra	13	13	0
		2da	2	2	0
De estas NC obtenidas:					
Alta		Media		Baja	
0		2		13	

Tabla 5: Resultados de las pruebas internas.

Pruebas Cruzadas:

Módulo	Casos de Uso	Iteración	NC	Cerrada(S)	No procede
Biblioteca	11	1ra	4	4	0
De estas NC obtenidas:					
Alta		Media		Baja	
0		0		4	

Tabla 6: Resultados de las pruebas cruzadas.

Pruebas de Liberación:

Módulo	Casos de Uso	Iteración	NC	Cerrada(S)	No procede
Biblioteca	11	1ra	5	5	0
		2da	18	16	2
Alta		Media		Baja	
		2		19	

Tabla 7: Resultados de las pruebas de liberación.

4.4. Conclusiones parciales

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

Conclusiones

Con la culminación del presente trabajo, se obtuvo como resultado principal, una segunda versión del módulo Biblioteca funcional, integrada a la plataforma educativa Zera y con toda la documentación generada durante el proceso de desarrollo de software.

- El estudio realizado como parte de la investigación sirvió de apoyo en la toma de decisiones con vista a un desarrollo eficiente del módulo.
- Se 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, frameworks de desarrollo, lenguajes de desarrollo etc. lo que permitió la selección de las óptimas, para realizar el análisis, diseño e implementación de la nueva versión del módulo Biblioteca de la plataforma Zera.
- Se demostró lo beneficioso que resulta la utilización de una metodología tan sólida como RUP paralelamente a la utilización del framework Symfony, lo que propició una implementación correcta de la aplicación.
- Se 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 propicien el intercambio de información y aprendizaje colaborativo.
- Se capturaron, 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, colaboraciones y secuencias.

- Se aplicaron patrones de caso de uso, que generan mejores resultados en el desarrollo de las actividades.
- Se demostró, a partir de las diferentes iteraciones de pruebas practicadas al software, que este satisface los requisitos funcionales definidos durante el flujo de trabajo Requerimientos.

Recomendaciones

Se recomienda:

- 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.
- Añadir una funcionalidad al módulo Biblioteca en la que los estudiantes puedan utilizar un recurso colaborativo para elaborar su propio contenido y proponerlo al profesor, a través de una Wiki.
- Brindar herramienta de ayuda a los usuarios de la Biblioteca la cual fungirá como un “Bibliotecario”, dando sugerencias de búsquedas a los usuarios, indicando cómo realizar la búsqueda, mostrando los recursos mejor valorados, los más visitados, etc. similar a un sistema de recomendaciones.
- Continuar potenciando el uso del framework Symfony en la facultad ya que su uso reporta un conjunto de ventajas.
- Transmitir experiencias a otros proyectos que utilicen metodología, herramientas y tecnologías similares.
- Tener en cuenta este trabajo para proyectos futuros, siempre y cuando se cumpla con las normas de confidencialidad establecidas.

Referencias bibliográficas

1. **García Peñalvo, F. J.** Estado actual de los sistemas e-learning. [En línea] [Citado el: 12 de 1 de 2012.] http://campus.usal.es/~teoriaeducacion/rev_numero_06_2/n6_02_art_garcia_pe_nalvo.htm.
2. **Rosenberg, Marc J.** *E-Learning: strategies for delivering knowledge*. New York : s.n., 2001. pág. 343. ISBN: 0-07-136268-1.
3. Definicion de Web 2.0. [En línea] Copyright © 2008-2012 - Definicion.de, 2008-2012. [Citado el: 17 de 1 de 2012.] <http://definicion.de/web-2-0/>.
4. **Karrer, Tony.** Understanding E-learning 2.0. [En línea] 2007. [Citado el: 7 de 12 de 2011.] https://www.astd.org/LC/2007/0707_karrer.htm..
5. **Colectivo de Autores.** Ecured. *Enciclopedia colaborativa en la red cubana*. [En línea] [Citado el: 15 de 01 de 2012.] http://www.ecured.cu/index.php/Plataformas_Educativas.
6. **Colectivo de Autores** . ¿Plataforma LMS o LCMS? *e-learning, Blog de Computer Aided*. [En línea] computer aided eLearning S.A., 2009. [Citado el: 17 de 1 de 2012.] www.cae.net/blog-es/plataforma-lms-y-lcms.
7. **Rengarajan, Raghavan.** Taking Advantage of Tight Integration. [En línea] 08 de 2001. [Citado el: 18 de 01 de 2012.] http://www.e-learn.cz/soubory/lcms_and_lms.pdf.
8. **Ryan, John Abbott y Terence.** Constructing knowledge and shaping brains. [En línea] 1999. [Citado el: 05 de 02 de 2012.] <http://www.21learn.org>.
9. **Wilson.** *Cómo valorar la calidad de la enseñanza*. Madrid, Paidós : Ediciones Paidós Ibérica, S.A., 1992. pág. 143. ISBN: 8475098223.
10. **Franky, María Carolina Díaz.** El blog de Carolina Díaz: una herramienta de aprendizaje virtual. *Aprendizaje colaborativo*. [En línea] [Citado el: 15 de 02 de 2012.] http://caroldiazf.blogspot.com/p/aprendizaje-colaborativo_7397.html.
11. **Kauchak, Eggen y.** *Estrategias docentes. En señanza de contenidos curriculares y desarrollo de habilidades de pensamiento*. MEXICO : FONDO DE CULTURA ECONOMICA DE ESPAÑA, S.L., 2006. ISBN 9789681664688.

12. **Wiley, David A.** Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. [En línea] 2002. [Citado el: 14 de 12 de 2011.] <http://www.det.wa.edu.au/>.
13. **Núñez, Yanko.Ossandón.** Objetos de Aprendizaje: Un recurso pedagógico para e-learning. [En línea] 2005. [Citado el: 14 de 12 de 2011.] <http://www.ualpovirtual.cl/archivos/simposio2005/YankoOssandon-ObjetosDeAprendizaje.pdf>.
14. **Figuroa, Roberth G., Camilo J. Solís y Armando A. Cabrera.** METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES. [En línea] [Citado el: 15 de 02 de 2012.] <http://www.google.com.cu/url?sa=t&rct=j&q=RUP+es+un+proceso+formal:+Prov+ee+un+acercamiento+disciplinado+para+asignar+tareas+y+responsabilidades+d+entro+de+una+organizaci%C3%B3n+de+desarrollo.+Su+objetivo+es+asegurar+l+a+producci%C3%B3n+de+software+de+alta+c>.
15. **James Rumbaugh, Ivar Jacobson, Grady Booch.** *El proceso unificado de desarrollo de software*. Madrid : Pearson Educación S.A, 2000. ISBN: 84-7829-036-2.
16. **Colectivo de Autores.** Ecured. *Enciclopedia colaborativa en la red cubana*. [En línea] [Citado el: 07 de 02 de 2012.] http://www.ecured.cu/index.php/Visual_Paradigm.
17. Sitio oficial de Balsamiq Mockups. [En línea] [Citado el: 05 de 02 de 2012.] <http://www.balsamiq.com/products/mockups>.
18. **Ravioli, Pablo.** Lenguaje de programación para paginas web. [En línea] 2011. [Citado el: 3 de 12 de 2011.] <http://www.monografias.com/trabajos7/html/html.shtml> .
19. **Pérez, Javier Eguíluz.** Introducción a CSS. [En línea] 2009. [Citado el: 3 de 12 de 2011.] <http://www.librosweb.es/css/capitulo1.html> .
20. —. Introducción a JavaScript. [En línea] 2009. [Citado el: 3 de 12 de 2011.] <http://www.librosweb.es/javascript/capitulo1.html>.

21. **Colectivo de Autores.** Manual de PHP. [En línea] The PHP Documentation Group, 01 de 06 de 2012. [Citado el: 3 de 12 de 2011.] <http://www.php.net/manual/es/index.php>.
22. **Alvarez, Miguel Angel.** Manual de JQuery. [En línea] 2010. [Citado el: 15 de 12 de 2011.] <http://www.biblioteca-digital.net.ve/wordpress/wp-content/uploads/2010/07/manual-jquery.pdf> .
23. Página Oficial de Symfony. [En línea] [Citado el: 3 de 12 de 2011.] <http://www.symfony.es/que-es-symfony/>.
24. **Alvarez, Sara.** Sistemas gestores de bases de datos. [En línea] 2007. [Citado el: 3 de 12 de 2011.] <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>.
25. Página Oficial de Postgres SQL. [En línea] [Citado el: 3 de 12 de 2011.] http://www.postgresql.org.es/sobre_postgresql.
26. Sitio Oficial de Apache Server. [En línea] [Citado el: 3 de 12 de 2011.] http://httpd.apache.org/ABOUT_APACHE.html .
27. Sitio Oficial de Netbeans IDE. [En línea] [Citado el: 3 de 12 de 2011.] http://netbeans.org/index_es.html .
28. **Colectivo de Autores.** Ecured. *Enciclopedia colaborativa en la red cubana.* [En línea] [Citado el: 10 de 03 de 2012.] http://www.ecured.cu/index.php/Patrones_de_Casos_de_Uso.
29. **James Rumbaugh, Ivar Jacobson, Grady Booch.** *El Proceso Unificado de Desarrollo de Software. Manual de Referencia.* s.l.: Addison Wesley. Madrid : ADDISON-WESLEY, 2000. pág. 464. ISBN 9788478290369.
30. **Colectivo de Autores Universidad de Granada.** Departamento de Lenguajes y sistemas informáticos. *Pruebas de Software.* [En línea] [Citado el: 30 de 04 de 2012.] <http://lsi.ugr.es/~ig1/docis/pruso.pdf>.
31. **Mañas, José A.** Prueba de Programas. *Departamento de Ingeniería de Sistemas telemáticos.* [En línea] Universidad Politécnica de Madrid, 16 de 3 de 1994. [Citado el: 20 de mayo de 2012.] <http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/testing.htm>.

Bibliografía

Alvarez, J. Bibliotecas virtuales. [En línea] 2001. [Citado el: 3 de 12 de 2011.] <http://es.scribd.com/doc/51572763/bibliotecas-virtuales>.

Marquès, Pere. El software educativo. [En línea] 2006. [Citado el: 7 de 12 de 2011.] http://www.lmi.ub.es/te/any96/marques_software/.

Consortium. [En línea] [Citado el: 2011 de 12 de 14.] <http://www.w3.org/Consortium/>.

Mendoza, María Teresa Montero. Dialogando con el constructivismo: Visiones y versiones. [En línea] [Citado el: 14 de 12 de 2011.] http://www.uacj.mx/DINNOVA/Documents/modelo_pedagogico.pdf.

Dr. Mazarío Triana, Israel, Lic. Mazarío Triana, Ana C. El constructivismo: paradigma de la escuela. [En línea] [Citado el: 14 de 12 de 2011.] <http://www.bibliociencias.cu/gsd/collect/libros/index/assoc/HASH2243.dir/doc.pdf>

Papert, Seymour. Constructionism: A New Opportunity for Elementary Science Education . [En línea] 1989. [Citado el: 14 de 12 de 2011.] <http://nsf.gov/awardsearch/showAward.do?AwardNumber=8751190>.

Castellanos Coutiño, Carlos Alberto. Panorama general de la Educación a distancia. [En línea] [Citado el: 14 de 12 de 2011.] http://www.ateneonline.net/datos/19_01_Castellanos_Carlos.pdf.

Cardona, G. Tendencias educativas para el siglo XXI. Educación virtual, online y @Learning. Elementos para la discusión. [En línea] 5 de 2002. [Citado el: 18 de 1 de 2012.] http://www.uib.es/depart/gte/edutec-e/revelec15/car.htm#_Toc519250761.

Agudelo, M. M. Plataformas educativas. *Programa de integración de Tecnologías de la información y la Comunicación a la Docencia*. [En línea] 2008. <http://aprendeonline.udea.edu.co/banco/html/plataformaseducativas/>.

slideshare. [En línea] [Citado el: 5 de 04 de 2012.] <http://www.slideshare.net/guest0a6e49/sem-8-modelo-de-analisis-1362411>.

Francisco José García Peñalvo, Miguel Ángel Conde González y Sergio Bravo Martín. Ingeniería del Software. *Tema 6: Diseño orientado a objetos.* [En línea] 16 de 10 de 2008. [Citado el: 11 de 04 de 2012.] <http://ocw.usal.es/enseñanzas-tecnicas/ingenieria-del-software/contenidos/Tema6-DOO-1pp.pdf>.

Aprendizaje colaborativo y tecnologías de la información y la comunicación.
Calzadilla, María Eugenia. Venezuela: OEI-Revista Iberoamericana de Educación. ISSN: 1681-5653.

García Rubio, Félix Oscar, Bravo Santos, Crescencio. Metodología de Desarrollo de Software. *Universidad Autónoma de México.* [En línea] [Citado el: 15 de 1 de 2012.] http://tabasco.torreingenieria.unam.mx/GCH/Disenio%20y%20gestion/tema3_1xh.pdf.

Sitio oficial de Moodle. [En línea] [Citado el: 3 de 12 de 2011.] <http://moodle.org>.

Colectivo de Autores Cátedra UNESCO de Educación a Distancia. Taller sobre Objetos de Aprendizaje. [En línea] [Citado el: 14 de 12 de 2011.] http://www.google.com.cu/url?sa=t&rct=j&q=objeto+de+aprendizaje+definicion&source=web&cd=11&ved=0CB4QFjAAOAO&url=http%3A%2F%2Fwww.sceu.frba.utn.edu.ar%2Fdav%2Fcongreso%2Ftaller.doc&ei=5O_oTt_bNuXn0QGzWLjVCQ&usg=AFQjCNH2_4gTz8SyTi5UYwix65M-yD7w5Q&cad=rja.

51. **Colectivo de Autores.** Ecured. *Enciclopedia colaborativa en la red cubana.* [En línea] [Citado el: 11 de 04 de 2012.] http://www.ecured.cu/index.php/Patrones_en_Symfony.

52. —. Ecured. *Enciclopedia colaborativa en la red cubana.* [En línea] [Citado el: 12 de 04 de 2012.] http://www.ecured.cu/index.php/Diagrama_de_despliegue.

53. *Hiperentornos educativos.* **Ledo, María Vidal, Freddy Gómez Martínez y Alina María Ruiz Piedra.** 1, Ciudad de La Habana : Editorial Ciencias Médicas, 2011, Vol. 25. ISSN 1561-2902.

Glosario de términos

Capítulo: La más general de las estructuras en las que se divide el contenido de un producto.

CASE: Computer Aided Software Engineering / Ingeniería de Software Asistida por Ordenador.

Calidad de Software: Conjunto de propiedades inherentes a un objeto que le confieren capacidad para satisfacer necesidades implícitas o explícitas.

Caso de Prueba (CP): Conjunto de condiciones o variables bajo las cuáles el analista determinará si el requisito de una aplicación es parcial o completamente satisfactorio.

Framework: Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Herramienta CASE: Aplicación informática destinada a aumentar la productividad en el desarrollo de software reduciendo el coste de estas en términos de tiempo y de dinero, se utiliza para la modelación del sistema.

IDE: Integrated Development Environment (Entorno integrado de desarrollo). Aplicación compuesta por un conjunto de herramientas útiles para un programador a la hora de desarrollar un software.

Learning Management System (LMS): Un LMS es una aplicación de software instalado en un servidor, que se emplea para administrar, distribuir y controlar las actividades de formación no presencial o e-Learning (Educación a distancia) de una institución u organización.(39)

PHP: Hypertext Pre-processor, lenguaje de programación, diseñado originalmente para la creación de páginas web dinámicas. Es utilizado fundamentalmente como lenguaje interpretado del lado del servidor.

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