

Universidad de las Ciencias Informáticas
Facultad 4



ESTRATEGIA PARA LA EVALUACIÓN DEL APRENDIZAJE POR OBJETIVOS Y POR COMPETENCIAS EN LA NUEVA VERSIÓN DE LA PLATAFORMA EDUCATIVA ZERA

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

AUTORES

Claudia Alvarez del Riego

Yordany Morejón Loyola

TUTORES

MSc. Yaillet Martínez Pérez

Ing. Miguel Medina Ramírez

La Habana, junio de 2014

“Año 56 del Triunfo de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Claudia Alvarez del Riego

Yordany Morejón Loyola

MSc. Yaillet Martínez Pérez

Ing. Miguel Medina Ramírez

Agradecimientos

A mis padres por su amor, su dedicación y esfuerzo durante estos cinco años y durante toda mi vida.

A mi familia gracias por todo el amor que me han brindado siempre, por su preocupación y apoyo constante.

A mi familia Pinera gracias por dejarme ser parte de ustedes.

A Cami, mil gracias por estar a mi lado siempre, en las buenas y en las no tan buenas.

A todos mis amigos.

A mi compañero Yordany por su esfuerzo.

A mis tutores por su ayuda.

A los profesores que contribuyeron en mi formación cada día.

A todas las personas que de una forma u otra contribuyeron con el desarrollo de esta investigación.

A todos ustedes, mil gracias

Claudia

A toda mi familia por su incondicional apoyo en todo momento. En especial a mi madre, mi padre, mi abuela por toda la confianza que depositaron en mí.

A los tutores Yaillet y Miguel los cuales guiaron este trabajo y siempre nos ayudaron cada vez que los necesitamos.

A mi compañera de tesis Claudia, sin la cual esta investigación no hubiera sido posible.

A los profesores Maily Cabrera, Jorge Luis Piña, Maria Caridad Váldez por toda la ayuda que nos brindaron.

A la UCI por ser uno de los lugares en el cual he tenido mis mejores experiencias y formar a la persona que soy hoy.

A todas las personas que de una forma u otra hicieron posible la realización de este trabajo.

Yordany

Dedicatoria

A mi abuela que es lo más lindo que tengo.

A mi hermana que ojala siga mis pasos.

A mis padres porque este también es su triunfo.

A Cami por su amor en todo momento.

A Tata por estar siempre ahí.

Claudia

A mis padres y mi abuela Orquidea por todo lo que se han sacrificado por mí y su apoyo incondicional a lo largo de toda esta larga etapa en la universidad.

A la memoria de mi abuelita Marta que ya no está junto a nosotros, a la cual recuerdo con mucho amor y cariño.

A todas las personas de la Iglesia Presbiteriana de San Nicolás por su apoyo cada fin de semana en especial a Jorge Vilarrubia.

A mis amigos tanto dentro de la universidad como fuera de la misma junto a los cuales pase momentos inolvidables.

Yordany

Resumen

La evaluación es uno de los momentos más importantes dentro del proceso de enseñanza y aprendizaje, esta es más compleja si se desarrolla en entornos virtuales. La presente investigación tiene como objetivo diseñar una base conceptual que contribuya al establecimiento de los modelos de evaluación por objetivos y por competencias mediante una estrategia como herramienta para la evaluación del aprendizaje en la nueva versión de la Plataforma Educativa ZERA. Esta fue concebida para ser adaptable a diferentes entornos educativos, por lo que surge la necesidad de incorporarle estos modelos de evaluación por objetivos y por competencias. El proceso fue concebido desde su diseño hasta la relación con una tarea. La evaluación por objetivos valora los contenidos según las metas definidas; mientras que la evaluación por competencias mide conocimientos específicos, además del desempeño de los estudiantes en las tareas asociadas a la práctica. Para dar cumplimiento a los objetivos planteados se definen los principales conceptos asociados a los modelos de evaluación del aprendizaje por objetivos y por competencias, para ello se investigó acerca de la evaluación en las plataformas educativas similares; además se describe la metodología y las herramientas apropiadas para el desarrollo de la propuesta de solución. Como resultado se obtuvo una estrategia para la evaluación del aprendizaje por objetivos y por competencias sostenida por las funciones que contempla la capa de abstracción desarrollada.

Índice de contenido

Introducción	9
Capítulo 1. Fundamentación teórica.....	14
1.1 Introducción	14
1.2 Conceptos asociados al dominio del problema	14
1.2.1 Evaluación del aprendizaje	14
1.2.2 Modelos de evaluación del aprendizaje	16
1.2.3 Plataforma e-Learning o plataforma educativa	20
1.2.4 Instrumento de evaluación del aprendizaje.....	20
1.2.5 Evaluación en entornos e-Learning.....	22
1.2.6 Modelos de evaluación del aprendizaje aplicados en entornos e-Learning.....	23
1.3 Sistemas similares.....	26
1.4 Entorno de desarrollo	28
1.4.1 Metodología de desarrollo de software	28
1.4.2 Lenguaje de modelado.....	30
1.4.3 Herramienta de modelado UML	30
1.4.4 Lenguajes de desarrollo	31
1.5 Marco de trabajo.....	31
1.5.1 Capa de lógica del negocio	32
Symfony 2.3.7.....	32
1.5.2 Capa de acceso a datos.....	32
Doctrine 2.1	32
1.6 Ambiente de desarrollo integrado (IDE según sus siglas en inglés)	33
1.7 Sistema gestor de bases de datos	33

1.8	Servidor Web.....	34
1.9	Conclusiones del capítulo.....	35
Capítulo 2.	Características de la propuesta de solución	36
2.1	Introducción	36
2.2	Modelo del dominio	36
2.2.1	Diagrama del modelo de dominio.....	38
2.3	Descripción del sistema propuesto.....	39
2.3.1	Requisitos del software	39
2.4	Modelo de casos de uso del sistema	43
2.4.1	Patrones de caso de uso.....	43
2.4.2	Descripción de los actores del sistema	45
2.4.3	Diagrama de actores	46
2.4.4	Diagrama de casos de uso del sistema	46
2.4.5	Descripciones de los casos de uso del sistema	49
2.5	Conclusiones del capítulo.....	51
Capítulo 3.	Diseño e implementación de la propuesta de solución	52
3.1	Introducción	52
3.2	Patrón arquitectónico Modelo Vista Controlador (MVC) en Symfony.....	52
3.3	Patrones de diseño.....	53
3.3.1	Patrones de diseño de Bases de datos.....	53
3.3.2	Patrones GRASP	56
3.3.3	Patrones GoF	57
3.4	Modelo de diseño	57
3.4.1	Diagramas de clases del diseño	58
3.4.2	Diagramas de interacción.....	58

3.5	Diseño de la base de datos	59
3.5.1	Descripciones de las tablas.....	61
3.6	Diagrama de despliegue.....	61
3.7	Modelo de implementación.....	62
3.7.1	Diagrama de componente	62
3.8	Validación del diseño de las clases de la capa de abstracción	63
3.8.1	Resultados obtenidos de la validación de la métrica TOC	64
3.8.2	Resultados obtenidos de la validación de la métrica RC	65
3.9	Pruebas de software.....	65
3.9.1	Niveles de prueba	66
3.9.2	Métodos de prueba	66
3.9.3	Automatización de las pruebas	67
3.9.4	Resultados obtenidos.....	68
3.10	Resumen comparativo.....	69
3.11	Conclusiones del capítulo.....	70
	Conclusiones generales.....	72
	Recomendaciones	73
	Referencias Bibliográficas	74
	Glosario de términos.....	78

Introducción

La evaluación es un proceso complejo pero inevitable. Es una fuerza positiva cuando "sirve al progreso y se utiliza para identificar los puntos débiles y fuertes, y para tender hacia una mejora" (1). En el proceso de enseñanza y aprendizaje esta juega un papel fundamental. El docente debe tener claro que la evaluación se da como un proceso de recolección de datos para emitir un juicio final. Además, la evaluación determina los logros: del sistema, de la mediación docente y del alumno; por lo tanto, debe ser lo más confiable posible y aplicada con criterios técnicos y razonados (2).

En los últimos años, con el auge de las tecnologías ha crecido la preocupación en torno a la aplicación de la evaluación en los ambientes virtuales con fines educativos (3), donde el proceso evaluativo se torna aún más complejo, siendo este un campo intrínsecamente interdisciplinar, que integra cuestiones relacionadas con la Psicología, la Pedagogía, las Tecnologías de la Información y las Comunicaciones (TIC) y otras disciplinas asociadas. Esta diversidad hace que sea un dominio difícil de tratar, cuyo entendimiento no siempre se logra desde aproximaciones teóricas parciales.

Cada vez son más las experiencias innovadoras en las que los medios tecnológicos sirven de soporte a los procesos educativos, como es el caso de las plataformas educativas o Sistemas de Administración del Aprendizaje (Learning Management System o LMS por sus siglas en inglés), los cuales brindan un apoyo invaluable a las instituciones educativas, facilitando una mejor administración y seguimiento del proceso de enseñanza y aprendizaje. Permiten estimular la idea de cooperación y de interacción como aspectos centrales del proceso de enseñanza y aprendizaje, mediante el uso de herramientas colaborativas que favorecen la adquisición de aprendizajes significativos en los estudiantes y al mismo tiempo afianzan en los docentes prácticas de enseñanza mediadas por las TIC.

En el plano internacional abunda la información sobre el desarrollo del aprendizaje electrónico, de las herramientas y de los entornos para esta clase de enseñanza. Existe una gran diversidad en la manera de enfocar los contenidos, de trabajar con el alumno, tanto en el orden pedagógico como el metodológico (4). Entre las soluciones más destacadas se encuentran: WebCT, BlackBoard, Moodle y Claroline (5).

Cuba, a pesar de ser un país en vías de desarrollo, dedica cuantiosos recursos para dotar a la educación cubana de estos beneficios. Se han encontrado evidencias de varios proyectos desarrollados en

condiciones similares y con propósitos afines relacionados con el tema de la educación a distancia. Se destaca el Sistema de Teleformación sobre Web, el Sistema Personalizado para el Aprendizaje a Distancia y las Plataformas ApreNDIST y la MedCampus.

En la Universidad de las Ciencias Informáticas (UCI) desde el año 2005 sirve de soporte al Entorno Virtual de Aprendizaje la plataforma de teleformación Moodle. En función de adaptarla al contexto universitario se creó un grupo para identificar, diseñar y desarrollar nuevos módulos, *plugins* (complementos) y funcionalidades. Se han realizado aportes significativos en el transcurso de estos años, siendo reconocidos, incluso por el creador de Moodle en la MoodleMoot 2009 en Buenos Aires, Argentina. Se han concretado múltiples proyectos nacionales e internacionales para la implementación de sistemas de teleformación basados en Moodle, muestra del prestigio alcanzado por los desarrolladores de la UCI (6).

En la UCI, además, se encuentra el Centro de Tecnologías para la Formación (FORTES), cuya misión es desarrollar tecnologías que permitan ofrecer servicios y productos para la implementación de soluciones de formación aplicando las TIC a instituciones con diferentes modelos de formación y condiciones tecnológicas, garantizando la calidad de las soluciones y la formación de los recursos humanos a partir de investigaciones que combinen los elementos pedagógicos y tecnológicos más avanzados, integrando así los procesos de formación, producción e investigación (7).

En el Centro FORTES de la Facultad 4 se ha desarrollado una Plataforma Educativa denominada ZERA que tiene como objetivo servir de apoyo al proceso de enseñanza y aprendizaje basada en una modalidad semipresencial, lo que flexibiliza la enseñanza, compartiendo la responsabilidad del aprendizaje entre el profesor y la Plataforma.

La Plataforma está estructurada por módulos, de manera que se encuentran informatizados varios procesos que son necesarios en las instituciones educativas; uno de estos módulos es: "Docente", donde se agrupa toda la información y los recursos educativos destinados al profesor, así como las acciones de administración que puede realizar. Permite la gestión de evaluaciones de los estudiantes, tanto en línea como presencial, así como llevar un control de su actitud ante el estudio y de la asistencia a clases (8).

La Plataforma Educativa ZERA fue creada con la premisa de que pudiera ser adaptada a diferentes entornos educativos. Fue diseñada en un primer momento para la Educación Media de México, enfocada a un sistema de evaluación del aprendizaje basado en competencias. Un análisis de sus requisitos

destacó que esta forma de evaluación del aprendizaje basada en competencias no abarca todos los instrumentos de evaluación existentes en la Plataforma; y presenta dificultades en su gestión y evaluación.

Se analizó además, que la Plataforma no cuenta con las funcionalidades para evaluar por objetivos, siendo esta la forma de evaluación que rige el modelo de enseñanza cubano.

Como parte de la definición de una arquitectura de referencia para el desarrollo de aplicaciones en el Centro FORTES, se ha iniciado la formalización de marcos de trabajo en cada Línea de Productos de Software, con el propósito de disminuir la diversidad tecnológica de las soluciones, arrojando como principal resultado un nuevo marco de trabajo denominado: Xalix.

En dicho Centro se desarrolla una nueva versión de la Plataforma Educativa ZERA sobre el marco de trabajo Xalix; la cual permitirá corregir las deficiencias del modelo de evaluación existente en las versiones anteriores. Dentro del desarrollo de esta nueva versión se enmarca la presente investigación.

Derivado de estas problemáticas se establece como **problema a resolver**: ¿Cómo contribuir al establecimiento de los modelos de evaluación por objetivos y por competencias en la nueva versión de la Plataforma Educativa ZERA?

Se plantea como **preguntas de investigación**:

1. ¿En qué fundamentos teóricos se apoyará la concepción de los modelos de evaluación a diseñar?
2. ¿Qué aspectos se deben tener presente para lograr el diseño de los modelos de evaluación que se proponen como solución al problema de investigación?
3. ¿Cómo garantizar las futuras implementaciones de los modelos de evaluación propuestos a partir del modelo conceptual diseñado?

Como **objeto de estudio** se define: los modelos de evaluación del aprendizaje para entornos de aprendizaje virtual, teniendo como **campo de acción**: el proceso de evaluación del aprendizaje por objetivo y por competencia para la nueva versión de la Plataforma Educativa ZERA.

Para sustentar la solución al problema referido se plantea el correspondiente **objetivo general**: diseñar una base conceptual para la evaluación del aprendizaje, que permita la adopción de los modelos de evaluación por objetivo y por competencias, en la nueva versión de la Plataforma Educativa ZERA.

Para dar cumplimiento al objetivo general se definen los siguientes **objetivos específicos**:

1. Analizar los conceptos asociados a la evaluación del aprendizaje por objetivos y por competencias, y su aplicación en plataformas educativas.
2. Diseñar un modelo conceptual que estandarice los datos asociados a la gestión de la evaluación del aprendizaje que contribuya a establecer los modelos de evaluación por objetivos y por competencias.
3. Desarrollar la capa de abstracción que responda al modelo propuesto, así como los métodos de comunicación entre los componentes de dicho modelo.

Desde el punto de vista metodológico se emplearon los siguientes **métodos científicos**:

Métodos Teóricos

Histórico-lógico: Se utilizó para estudiar la evolución y desarrollo de los modelos de evaluación tanto de los entornos tradicionales como de los entornos virtuales, específicamente lo referido a la evaluación por objetivos y por competencias. Este método resultó de gran utilidad sobre todo en la elaboración de los precedentes teóricos del tema trabajado.

Analítico-sintético: Se llevó a cabo el análisis de las particularidades que debe cumplir el modelo conceptual para la adopción de los modelos de evaluación del aprendizaje por objetivos y por competencias. Se examinaron las teorías, documentos, tendencias y procesos, permitiendo la extracción de los elementos esenciales que se relacionan con los modelos de evaluación del aprendizaje.

El presente trabajo consta de la siguiente estructura capitular:

Capítulo 1. Fundamentación teórica: Se hace un análisis del estado del arte del objeto de estudio, se reflejan todos los elementos teóricos que soportan la investigación: modelos e instrumentos de evaluación del aprendizaje, evaluación del aprendizaje por objetivos y por competencias. Además, se mencionan todas las herramientas, metodologías y lenguajes empleadas para dar solución al problema planteado.

Capítulo 2. Características de la propuesta de solución: Se detalla la propuesta de solución. Se realiza el modelo de dominio, se definen los requisitos funcionales y no funcionales que presentará el sistema, se generará la documentación correspondiente a la metodología de desarrollo de software seleccionada.

Capítulo 3. Diseño e implementación de la propuesta de solución: Se realiza el modelo de diseño y dentro de este serán generados los diagramas de clases del diseño y los diagramas de secuencia. Se define el patrón arquitectónico y los patrones de diseño a usar. En este capítulo se realizan además, los diagramas de componentes y las pruebas de software.

Además se presentan las conclusiones, recomendaciones, referencias bibliográficas y los anexos.

Capítulo 1. Fundamentación teórica

1.1 Introducción

En el presente capítulo se describen los principales conceptos relacionados con el problema, así como el estudio realizado a plataformas educativas similares con el fin de determinar las características fundamentales de las mismas y obtener resultados para apoyar el cumplimiento de los objetivos propuestos. Además se describe la metodología que guiará el proceso de desarrollo en conjunto con el resto de las herramientas que serán utilizadas.

1.2 Conceptos asociados al dominio del problema

1.2.1 Evaluación del aprendizaje

La evaluación es entendida como una etapa del proceso educacional que tiene por fin comprobar de modo sistemático en qué medida se han logrado los resultados previstos en los objetivos que se hubieran especificado con antelación (9).

La evaluación es un proceso permanente e interactivo orientado a recoger información sobre una realidad y valorar el nivel de logro alcanzado por el estudiante en el desarrollo de competencias, con la finalidad de tomar decisiones que lleven a mejorar la práctica educativa (10).

Según la Dra. Magalys Ruiz Iglesias: “La evaluación es útil como herramienta para el mejoramiento de la calidad educativa, ya que a través de la misma se puede obtener información para tomar decisiones efectivas. La evaluación del aprendizaje es una medida cuantitativa o cualitativa del grado con que ha sido adquirido un conocimiento” (11).

Los autores definen la evaluación del aprendizaje como un proceso sistemático que busca información acerca del desempeño, avance, rendimiento o logro del estudiante.

La evaluación del aprendizaje se clasifica según su:

Finalidad y función

- ❖ **Función formativa:** la evaluación se utiliza preferentemente como estrategia de mejora y para ajustar sobre la marcha, los procesos educativos de cara a conseguir las metas u objetivos previstos. Es la

más apropiada para la evaluación de procesos, aunque también es formativa la evaluación de productos educativos, siempre que sus resultados se empleen para la mejora de los mismos.

- ❖ Función sumativa: se aplica más en la evaluación de productos, es decir, de procesos terminados, con realizaciones precisas y valorables. Con la evaluación no se pretende modificar, ajustar o mejorar el objeto de la evaluación, sino simplemente determinar su valía, en función del empleo que se desea hacer del mismo posteriormente.

Su extensión

- ❖ Evaluación global: abarca todos los componentes o dimensiones del alumno, del centro educativo o del programa. Se considera el objeto de la evaluación de un modo holístico, como una totalidad interactuante, en la que cualquier modificación en uno de sus componentes o dimensiones tiene consecuencias en el resto. Con este tipo de evaluación, la comprensión de la realidad evaluada aumenta, pero no siempre es necesaria o posible.
- ❖ Evaluación parcial: es el estudio o valoración de determinados componentes, dimensiones de un centro, de un programa educativo o de rendimiento de un alumno.

Los agentes evaluadores

- ❖ Evaluación interna: es aquella que es llevada a cabo y promovida por los propios integrantes de un centro o un programa educativo. A su vez, la evaluación interna ofrece diversas alternativas de realización, cuando el docente es quien evalúa: heteroevaluación, cuando el alumno se evalúa a sí mismo: autoevaluación y cuando el grupo es quien evalúa: coevaluación.
- ❖ Evaluación externa: se da cuando agentes no integrantes de un centro escolar o de un programa evalúan su funcionamiento. Suele ser el caso de la evaluación de expertos. Estos evaluadores pueden ser inspectores de evaluación, miembros de la administración, investigadores, equipos de apoyo a la escuela, entre otros.

El momento de aplicación

- ❖ Evaluación inicial: se realiza al comienzo del curso académico, de la implantación de un programa educativo o del funcionamiento de una institución escolar. Consiste en la recogida de datos en la situación de partida. Es útil para iniciar cualquier cambio educativo, para decidir los objetivos que se

pueden y deben conseguir y también para valorar si al final de un proceso, los resultados son satisfactorios o insatisfactorios.

- ❖ **Evaluación procesual:** consiste en la valoración a través de la recogida continua y sistemática de datos, del funcionamiento de un centro, de un programa educativo, del proceso de aprendizaje de un alumno o de la eficacia de un profesor, a lo largo del periodo de tiempo fijado para la consecución de unas metas u objetivos. La evaluación procesual es de gran importancia dentro de una concepción formativa de la evaluación, porque permite tomar decisiones de mejora sobre la marcha.
- ❖ **Evaluación final:** consiste en la recogida y valoración de unos datos al finalizar un período de tiempo previsto para la realización de un aprendizaje, un programa, un trabajo, un curso escolar, o para la consecución de unos objetivos (12).

1.2.2 Modelos de evaluación del aprendizaje

Para la evaluación del aprendizaje se toman como referencia una serie de modelos, de los cuales se abordarán en la presente investigación los modelos de evaluación por objetivos y por competencias.

Un modelo de evaluación debe verificar que los contenidos y los objetivos respondan a las necesidades, intereses y capacidades existentes. Y su finalidad consiste en ir creando una teoría sobre la evaluación y una gestión eficaz sobre su sistematización, ello garantizará la lógica del procedimiento y su justificación (13).

Los modelos de evaluación se adentran en distintas clasificaciones las mismas según Daniel L. Stufflebeam y Anthony J. Shinkfielden son:

La pseudoevaluación o evaluaciones en las que se intenta conducir a determinadas conclusiones mediante una evaluación. Distinguen dos tipos: las investigaciones encubiertas y los estudios basados en las relaciones públicas.

La cuasievaluación o evaluación en la que se trata de responder a ciertas cuestiones de interés, en lugar de determinar el valor de algo. Existen diferentes tipos de cuasievaluaciones tales como: la comprobación de programas, los sistemas de información administrativa, los estudios de responsabilidad, los estudios basados en objetivos y los estudios basados en la experimentación.

La evaluación verdadera: Su finalidad es enjuiciar o perfeccionar el valor o mérito de un objeto.

- ❖ Los estudios para la toma de decisiones.
- ❖ Lo estudios centrados en el cliente.
- ❖ Los estudios políticos.

El método holístico de la evaluación: Propone redefinir el concepto de evaluación como “proceso mediante el cual se proporciona información útil para la toma de decisiones”; diferenciándose de los estudios basados en objetivos, que se limita a determinar si los objetivos han sido alcanzados (14).

- ❖ Evaluación orientada hacia el perfeccionamiento.
- ❖ La evaluación iluminativa.
- ❖ La evaluación constructivista.
- ❖ La evaluación fundamentada en competencias.

Evaluación del aprendizaje por objetivos:

La evaluación por objetivos, de contenidos (formas de saber) y métodos (formas de hacer), recibe diversos nombres: evaluación por capacidades, evaluación formativa-sumativa, evaluación sumativa, evaluación criterial. Se trata de evaluar cuantitativamente los contenidos y los métodos en función de los objetivos (15).

Se parte de objetivos fijados por las personas interesadas o por el conjunto de participantes en el proceso. El propósito más común es determinar si los objetivos se han cumplido (14).

Según los autores, un objetivo es una expresión de lo que se quiere conseguir y debe permitir la articulación de acciones encaminadas a su logro.

En la literatura existen diversos criterios en torno a las clasificaciones de los objetivos, en esta investigación serán clasificados según su nivel de generalidad; estos se dividen en tres categorías: generales, específicos y operacionales. Los generales abarcan todo un nivel educativo o señalan las metas generales de un curso. Los específicos concretan las características de los cambios que se espera obtener, estos deben ser compatibles con los objetivos generales y que se desglosen de estos. Y los operacionales o de conducta indican con precisión lo que se espera que los alumnos aprendan.

Evaluación del aprendizaje por competencias

El concepto de competencia es complejo, ambiguo y difuso ejemplo de ello son las disímiles definiciones realizadas por diversos autores. Estos concuerdan en que las competencias son los conocimientos y destrezas que puede tener un individuo en un entorno determinado.

Las competencias son un conjunto de conocimientos, habilidades y valores que convergen y permiten llevar a cabo un desempeño de manera eficaz, es decir, que el alumno logre los objetivos de manera eficiente y que obtenga el efecto deseado en el tiempo estipulado y utilizando los mejores métodos y recursos para su realización (16).

La Junta de Acreditación de las Ciencias de la Computación (Computer Science Accreditation Board o CSAB según sus siglas en inglés) de los Estados Unidos en el año 2001 plantea: “la competencia es un concepto integrador porque consiste tanto en las aptitudes como en las actitudes, de modo que va más allá de los componentes técnicos, los cuales se complementan con los componentes metodológicos, participativos y personales. Supone no sólo saber lo que hay que hacer en una situación, sino también ser capaz de enfrentarse a ello en una situación real”.

Según el Dr. Sergio Tobón, existen diversas clasificaciones de las competencias, las más usadas son las competencias específicas y genéricas. Las competencias genéricas se refieren a las competencias que son comunes a una rama profesional (por ejemplo, salud, ingeniería, educación) o a todas las profesiones. Finalmente, las competencias específicas, a diferencia de las competencias genéricas, son propias de cada profesión y le dan identidad a una ocupación (17).

Pasos para evaluar el aprendizaje basado en competencias

Según el Dr. Sergio Tobón:

1. ¿Qué evaluar? (Las competencias y sus saberes).
2. ¿Para qué evaluar? (Para lograr una formación integral).
3. ¿Con qué criterios? (Resultados esperados de la competencia).
4. ¿Con qué evidencias? (Pruebas concretas que presentan los estudiantes).

5. ¿Cómo determinar el nivel de aprendizaje? (Mediante matrices de evaluación).
6. ¿En qué momento se evalúa? (Al inicio, durante y al final del proceso).
7. ¿Con qué estrategia evaluar? (Con diversas estrategias como pruebas, portafolios, etcétera).
8. ¿Cómo informar? (Se retroalimenta con base al nivel de aprendizaje alcanzado, los logros y los aspectos a mejorar).

Según el sistema de Educación Media Superior de Guadalajara (18):

1. Preparación: definir, qué se evalúa, qué tipo de evaluación, inicial, procesual, final.
2. Distinguir las competencias implícitas al logro de la actividad.
3. Definir los criterios de desempeño (cómo lo hace) e indicadores de logro (qué hace).
4. Diseñar el instrumento para evaluar; una mezcla de métodos y técnicas deberá ser usada para proveer evidencia suficiente de la cual inferir el logro de competencia.
5. Llevar a cabo la evaluación.
6. Interpretar los resultados (juicios y tomas de decisión del profesor sobre el progreso del estudiante).
7. Realizar la metaevaluación.

Según la Dra. Magaly Ruiz Iglesias (19):

1. Definir con exactitud las competencias a evaluar con sus respectivas dimensiones.
2. Construir los indicadores para evaluar las competencias de forma integral con criterios académicos y profesionales.
3. Definir el tipo de evidencias que se deben presentar para llevar a cabo la evaluación.
4. Establecer las estrategias e instrumentos con los cuales se llevará a cabo la evaluación.
5. Analizar la información con base en los indicadores, determinar fortalezas y aspectos a mejorar, retroalimentar de forma oportuna a los estudiantes.

6. Generar un espacio de reflexión en ellos tanto sobre el proceso como en torno a los resultados de la evaluación, con la posibilidad de cambiar los resultados de acuerdo a los argumentos que ellos presenten.

Como se pudo observar, los pasos descritos por los distintos autores para evaluar el aprendizaje basado en competencias coinciden en la mayoría de sus aspectos a pesar de que fueron concretados en períodos de tiempo distintos. Estos pasos guiarán la evaluación del aprendizaje por competencia en la nueva versión de la Plataforma Educativa ZERA.

1.2.3 Plataforma e-Learning o plataforma educativa

Una plataforma *e-Learning* (aprendizaje electrónico), plataforma educativa web o Entorno Virtual de Enseñanza y Aprendizaje es un entorno informático que integra un conjunto de herramientas para la enseñanza y aprendizaje en línea, permitiendo una enseñanza no presencial (20).

Algunas de las ventajas que tiene el uso de estos entornos es que permiten:

- ❖ La comunicación e interacción entre los usuarios.
- ❖ El desarrollo e implementación de contenidos.
- ❖ La creación de actividades interactivas.
- ❖ La implementación de estrategias colaborativas.
- ❖ La evaluación y el seguimiento de los estudiantes.
- ❖ Que cada estudiante pueda personalizar el entorno adaptándolo a sus necesidades y características (21).

Este concepto educativo es una revolucionaria modalidad que consiste en la educación a través de Internet, posibilitando la interacción del usuario con el contenido docente mediante el uso de diversas herramientas informáticas para simular un entorno real.

1.2.4 Instrumento de evaluación del aprendizaje

Es una herramienta destinada a documentar el desempeño de una persona, verificar los resultados obtenidos y evaluar los productos elaborados, de acuerdo con una norma o parámetro previamente definido en la que se establecen los mecanismos y criterios que permiten determinar si una persona posee competencias o no considerando las habilidades, destrezas, conocimientos, actitudes y valores puestos en juego en el ejercicio de una acción en un contexto determinado (22).

Los instrumentos, como herramientas utilizadas para recolectar información ayudan a la medición, la cual constituye una actividad presente en la práctica docente, ésta a su vez aproxima al monitoreo y evaluación del proceso educativo (23).

Algunos instrumentos de evaluación usados en plataformas educativas web y que serán usados en la nueva versión de la Plataforma Educativa ZERA:

Foro: Los foros son aplicaciones diseñadas para la comunicación, que permiten el intercambio de ideas, preguntas frecuentes, dudas y aclaraciones de estas, envío de mensajes, entre otras posibilidades comunicativas (3).

Cuestionarios o reactivos: Los cuestionarios son herramientas potentes y flexibles que permiten al profesor diseñar diversas estrategias de evaluación. Algunas de las aplicaciones que poseen son: en las evaluaciones iniciales, a través de exámenes diagnósticos para tener una idea del grado de conocimiento y las habilidades que posee el estudiante sobre un tema; los exámenes parciales en los que se incluyen varios temas de una asignatura y también ofrecen múltiples posibilidades de autoevaluación, para facilitar a los estudiantes el monitoreo de su propio rendimiento, como instrumento de refuerzo y repaso (3).

Evidencias: Las evidencias son orientaciones que se le asignan al estudiante para realizar una o varias actividades, que generan una respuesta (evidencia) a la tarea. La nomenclatura de la actividad tiende a confundir, pues la evidencia es lo que adjunta el estudiante y la actividad en sí sería una asignación de una tarea investigativa (23).

Softarea: La Softarea es una tarea que el estudiante debe hacer, apoyada con materiales que están en la plataforma, es la unión de un recorrido dirigido, uno o varios reactivos y la asignación de una o varias “evidencias” (23).

Los instrumentos de evaluación que serán integrados a la evaluación del aprendizaje por objetivos y por competencias en la nueva versión de la Plataforma Educativa ZERA son: Softarea, Evidencia, Reactivo, Foro y Recorridos dirigidos.

1.2.5 Evaluación en entornos e-Learning

Al igual que en la evaluación presencial, existen diversas definiciones acerca de la evaluación en ambientes virtuales de aprendizaje tales como: controlar, penalizar, calcular el valor de algo, calificar, acreditar, determinar o emitir un juicio sobre el grado de suficiencia.

El Dr. Ramón R. Abarca Fernández plantea que: “La evaluación debe permitir, al estudiante, elevar sus conocimientos al estimularlo a superar sus desaciertos, lo que garantiza de forma general su crecimiento personal. La evaluación debe ser desarrolladora, procesal, integral, contextualizada, democrática, formativa, cualitativa, investigativa, sistemática, que contemple la revalorización de errores, que tenga en cuenta indicadores que garanticen su objetividad, que promueva y transite por modelos de la autoevaluación, coevaluación y la heteroevaluación” (24).

Lorenzo García Aretio, propone, sobre la base de su experiencia en el trabajo docente en la Universidad Nacional de Educación a Distancia (UNED), tomar en cuenta las siguientes actividades, dentro de cualquier estrategia de evaluación usando LMS:

- ❖ Desarrollar pruebas de autoevaluación. El estudiante debe estar en posibilidad de verificar su progreso en el estudio.
- ❖ Evaluar, más que la cantidad, la calidad de la participación académica del estudiante, buscando verificar que él proponga ideas novedosas, cite fuentes bibliográficas significativas, plantee aportes originales.
- ❖ Desarrollar un sistema de evaluación continua apoyada en los LMS pero solo con fines formativos.
- ❖ Evaluar las actividades colaborativas, las cuales se nutren de alguna cuestión, pregunta, estudio de casos o propuestas abiertas que pueden ser respondidas desde diferentes ópticas. Estos trabajos se pueden desarrollar en equipo.
- ❖ Plantear una evaluación final para las áreas de conocimiento, la cual debe tener carácter discriminador y responder a la pregunta de si el participante ha aprendido o no los contenidos impartidos (24).

Los ambientes virtuales posibilitan el desarrollo de nuevas formas de evaluación, así como una retroalimentación inmediata de los resultados; permiten el desarrollo de la comunicación, el trabajo en equipo y la reflexión crítica; además, facilitan la práctica de las habilidades adquiridas. Los autores consideran que una desventaja de la evaluación en ambientes virtuales es el plagio, que puede ocurrir dada la libertad con que fluye el proceso.

1.2.6 Modelos de evaluación del aprendizaje aplicados en entornos e-Learning

Se definen los modelos de evaluación del aprendizaje que pueden ser utilizados en entornos *e-Learning*, algunos de estos modelos son adaptados de la formación tradicional. Escoger un modelo puede resultar una tarea difícil por las disímiles características que estos presentan; además de que no necesariamente se tiene que adoptar un único modelo.

Modelo de Ralph Tyler: Parte del principio de que educar consiste en modificar antiguos patrones o crear nuevos patrones de comportamiento, para Tyler, el currículum debe especificar las habilidades deseadas, expresadas en los objetivos que se deben alcanzar. De esta forma, la evaluación, a partir del modelo propuesto por Tyler, debe especificar los objetivos que deben ser alcanzados confrontando los resultados obtenidos.

El proceso contiene las siguientes etapas:

1. Reconocer las metas u objetivos que el programa espera promover y alcanzar.
2. Ordenar los objetivos de modo jerárquico.
3. Definir los objetivos en términos de comportamiento.
4. Establecer situaciones y condiciones en las cuales puede ser demostrada la consecución de los objetivos.
5. Explicar los propósitos de la estrategia al personal más importante (profesorado encargado de realizarla) y cuáles serán los momentos y situaciones más adecuadas para la evaluación.
6. Elegir o desarrollar las medidas técnicas de evaluación apropiada, capaces de medir de manera adecuada los indicadores del logro de los objetivos y utilizar los procedimientos estadísticos apropiados.
7. Recoger y analizar la información necesaria (que podrán referirse a los centros, a los programas desarrollados o al aprendizaje concreto de los estudiantes).
8. Contrastar los datos con los objetivos de comportamiento esperados, concluyendo si hubo o no un logro de estos y en qué medida.

Modelo de Stake: Recibe influencias de Tyler, a pesar de considerar que su propuesta tiene limitaciones para interpretar los resultados alcanzados. Defiende una evaluación centrada en el cliente y no solo en los resultados. Es partidario del empleo de métodos cualitativos y el estudio de casos, frente al uso exclusivo de metodologías cuantitativas de base experimental. Su evaluación responsiva busca dar respuesta a las preguntas básicas de un programa educativo. Por lo tanto, los objetivos, los métodos, la recogida de datos y las informaciones del proceso evaluativo deben ser continuamente ajustados y adaptados con la utilización de procedimientos y métodos analíticos tanto cuantitativos como cualitativos. Busca el ajuste constante del programa al contexto donde se lleva a cabo.

Modelo de Stufflebeam: Modelo conocido como Contexto, Input, Proceso y Producto (CIPP) revela cómo la evaluación puede ayudar a tomar decisiones en varias etapas del desarrollo de un programa para lo que distingue cuatro categorías en la toma de decisión: 1) Los fines a conseguir. 2) Los medios a utilizar conforme al diseño. 3) Los medios reales conforman los procedimientos y la propia estructura. 4) Los fines reales constituyen los resultados o logros del programa.

Modelo de Kirkpatrick: Este modelo ha sido y es ampliamente utilizado en la evaluación de acciones formativas tradicionales, y en la actualidad son varios los autores que recomiendan su adaptación y uso en el *e-Learning*. El modelo está orientado a evaluar el impacto de una determinada acción formativa a través de cuatro niveles: la reacción de los participantes, el aprendizaje conseguido, el nivel de transferencia alcanzado y finalmente el impacto resultante. 1) Evaluación de la Reacción: evalúa la reacción de los participantes en diferentes aspectos de acciones formativas. 2) Evaluación del Aprendizaje: busca información que pueda comprobar si los “participantes mejoran o modifican sus conocimientos, habilidades o actitudes”. 3) Evaluación del Comportamiento: evalúa los cambios de conducta que tienen lugar después de participar en un programa. 4) Evaluación de los Resultados: tiene por objetivo medir los resultados del programa (24).

Modelo de Miller: Con la pirámide de Miller, se puede acometer la evaluación de la competencia a partir de los distintos tipos de saberes. Siguiendo la referida pirámide, se puede aludir a cuatro niveles de evaluación en relación directa del tipo de saber. En la base estaría el saber (*know*) o conjunto de conocimientos teóricos que todo profesional debe dominar como fundamentadores de la práctica profesional, a sabiendas como indica el propio Miller, de que no son suficientes, y por tanto no se puede quedar en este nivel de evaluación para atrapar la competencia, sencillamente estaríamos en su

dimensión más cognitiva. En el segundo nivel estaría el saber cómo (*know how*) usaría los saberes teóricos adquiridos si los tuviera que poner en práctica en un contexto particular. En este nivel evaluativo se puede obtener una visión previa de la práctica profesional. Con todo tampoco se puede conformar con este nivel evaluativo, puesto que el mejor de los casos puede llevar a la capacidad de las personas pero no a su competencia, al faltar la dimensión contexto y acción profesional. En el tercer nivel se encuentra el demuestra cómo (*show how*) lo haría, es una demostración con hechos, cercana a la realidad pero aún no es la práctica profesional directa. Finalmente, en la cúspide de la pirámide, Miller incardina el hacer (*doing*) en la práctica profesional. Se estaría ante una actuación real, en un contexto o situación profesional de desempeño y en el momento de evidenciar la competencia (25).

Instrumentos de evaluación que se relacionan en cada una de las etapas de la pirámide de Miller y que además son usados en entornos *e-Learning*:

Etapas	Sabe y sabe cómo	Demuestra cómo	Hace
Instrumentos	Pruebas orales Pruebas de ensayo Pruebas de libro abierto Pruebas objetivas Mapas conceptuales	Pruebas de desempeño (proyectos, cuestionarios) Rúbricas Entrevistas	La observación Evaluación 360° Portafolios

Tabla 1. Instrumentos de evaluación usados en las fases de la pirámide de Miller, fuente: elaboración propia a partir de (25)

Los modelos de Stake, Stufflebeam y Kirkpatrick no esclarecen con nitidez los indicadores de evaluación, los procesos y las formas de obtención de evidencias de cada uno de los elementos evaluados. Por ello para la realización de la evaluación del aprendizaje basado en objetivos se decide utilizar el Modelo Tyleriano, siendo este el pionero de estos estudios centrados en objetivos, pudiéndose determinar satisfactoriamente el cumplimiento de los objetivos propuestos, incluyéndole a este Modelo que exista una retroalimentación entre los actores del proceso evaluativo. Además, se decide utilizar de manera conceptual para la evaluación del aprendizaje basado en competencias el Modelo de Miller, pues este define que el estudiante debe tener conocimientos teóricos pero también debe saber cómo aplicarlos en la práctica; este modelo se utilizará junto a los pasos descritos en el epígrafe 1.2.2 para evaluar el aprendizaje por competencias.

La selección del modelo de evaluación del aprendizaje (evaluación por objetivos o evaluación por competencia) que se empleará se define cuando son creados los programas de estudio, pues puede que

para algunos de estos programas de estudio sea conveniente utilizar uno u otro modelo de evaluación del aprendizaje.

1.3 Sistemas similares

Con el objetivo de identificar tendencias en cuanto a la evaluación del aprendizaje en entornos *e-Learning* se realizó un estudio de plataformas educativas con características similares a la Plataforma Educativa ZERA; entre las que se encuentran: BlackBoard, Sakai y Moodle.

La Plataforma BlackBoard plasma objetivos claros y congruentes los cuáles logran tener una clara coherencia con la forma de evaluar, esta organización permite que el alumno logre obtener una evaluación de sus resultados. Permite además, realizar la heteroevaluación, la coevaluación y la autoevaluación como tipos de evaluación. La Plataforma Sakai se basa en el aprendizaje colaborativo y permite realizar la evaluación del aprendizaje por competencias; Sakai incluye herramientas como: Tareas, Exámenes, Encuestas, Informes y Foro. Permite además realizar, el seguimiento del proceso educativo y estadísticas de los alumnos

En el caso de la Plataforma Moodle se pudo constatar que permite realizar la evaluación del aprendizaje por competencias. En su versión 2.2, incorpora un sistema avanzado de calificaciones diseñado exclusivamente para el módulo “Tareas”, que consiste en la implementación de un sistema de evaluación basado en rúbricas, con la definición de una serie de “elementos a evaluar” de forma diferenciada dentro de la tarea y de la asignación de un valor a cada uno de esos aspectos. La calificación final se obtiene mediante el cálculo previo del porcentaje de los puntos obtenidos en todas las rúbricas sobre los puntos posibles a obtener, y este porcentaje se trasladará a la calificación final según el baremo establecido. Además, este sistema permite una retroalimentación no solo global de las competencias sino para cada una de las rúbricas establecidas, de manera que el estudiante podrá conocer en todo momento cómo ha sido evaluado en cada uno de los indicadores que conforman la calificación final de la competencia. Para adaptar el modelo a las necesidades del Plan de Evaluación de Competencias (PEC). El diseño en el aula virtual se estructura del siguiente modo:

- ❖ Fichas de evaluación por competencias. Se ofrecen al alumnado las fichas de cada una de las competencias que serán objeto de evaluación en la asignatura.
- ❖ Plan de actividades de evaluación. Este documento contiene información sobre las actividades que han de realizar los estudiantes. Una tabla resumen global que contiene los siguientes datos:

denominación de la actividad, competencia/s implicada/s, subcompetencia/s implicada/s, indicador/es implicado/s y temporalización. Y para cada una de las actividades: indicador/es, tareas, recursos y criterios de evaluación.

- ❖ Elaboración de escala personalizada. Las rúbricas que se utilizarán para cada uno de los indicadores de la evaluación por competencias exigen, en nuestro caso, la creación de una escala numérica específica (0; 0,25; 0,5; 0,75 y 1) para cada uno de los siguientes valores: muy deficiente, insuficiente, bueno, muy bueno y excelente respectivamente.
- ❖ Diseño del sistema de calificación. En primer lugar, se definen las “categorías” que se corresponden con cada una de las competencias a evaluar. Posteriormente, los “ítems de calificación”, que son cada uno de los indicadores, agrupados en su competencia correspondiente. Para cada categoría (competencia) se selecciona la opción “Media ponderada de calificaciones”, se determina cuáles serán las calificaciones máxima y mínima (sobre 10) y se establece el peso (porcentaje) con relación a la calificación final. Para cada uno de los ítems de calificación (indicadores) también se marca su peso y su calificación máxima y mínima (sobre 5), así como la escala personalizada, previamente definida.
- ❖ Actividades de evaluación. Se incorporan al aula virtual tantas “Tareas” como actividades de evaluación, y se utilizan diferentes modalidades: “Actividad no en línea”, “Subir un solo archivo” y “Subida avanzada de archivos”. Ninguna de estas tareas incorpora “calificación”, puesto que la evaluación no se realiza directamente sobre la actividad sino sobre los indicadores asociados pertenecientes a una determinada competencia. También se incorporan “Foros”, evaluados en la modalidad de “debate simple”, que sí son calificados directamente por estar asociados con un indicador específico.
- ❖ Rúbricas para cada uno de los indicadores de todas las competencias a evaluar. Se incorpora una “Actividad no en línea” para cada indicador, en la que se selecciona la escala de evaluación personalizada, la “rúbrica” como método de calificación y la categoría de calificación, es decir, la competencia a la que pertenece el indicador (27).

El uso de un sistema de gestión de aprendizaje para el diseño y gestión de la evaluación por competencias facilita la incorporación del PEC, puesto que simplifica el diseño de la evaluación por parte del profesorado y ofrece una herramienta accesible para estudiantes y docentes. Añade al PEC dos elementos muy importantes: la recogida centralizada de información sobre evaluación por competencias, y

la comunicación directa e inmediata de los resultados de aprendizaje a los estudiantes a través del calificador individualizado que incluye retroalimentación (26).

De manera general las plataformas estudiadas sirvieron de apoyo para la presente investigación ya que fueron identificados los tipos de evaluación, los instrumentos de evaluación, así como las estrategias de aprendizaje que emplean. En el caso específico de la Plataforma Moodle, se pudo evidenciar la forma en que se realiza el proceso de evaluación del aprendizaje por competencias, haciendo uso de la rúbrica como instrumento de evaluación que favorece el proceso evaluativo.

1.4 Entorno de desarrollo

Para obtener un software con la calidad requerida se hace necesaria la utilización de metodologías y herramientas que permitan conformar el área de trabajo. Con la definición de la arquitectura de referencia para el desarrollo de aplicaciones en el Centro FORTES la selección de las herramientas para la implementación de la solución está definida dentro de las determinaciones de dicha arquitectura. Las herramientas que no estén definidas dentro del marco de referencia serán utilizadas según las necesidades de la propuesta.

1.4.1 Metodología de desarrollo de software

Las metodologías de desarrollo de software surgen para facilitar y mejorar la construcción y desarrollo de un producto de software, mediante el uso de técnicas, métodos, prácticas, actividades, etapas, procedimientos, herramientas, y gestión documental (27). Existen dos tipos de metodologías de desarrollo de software que engloban el gran número de estas: las ágiles y las pesadas o tradicionales.

Las metodologías ágiles son aquellas donde los individuos y las interacciones entre ellos son más importantes que las herramientas y los procesos empleados, la funcionabilidad de un producto de software es más importante que generar documentación exhaustiva y para estas metodologías dar respuesta a un cambio es más importante que el seguimiento estricto de un plan. Mientras que las metodologías pesadas se centran en la definición detallada de los procesos y tareas a realizar, herramientas a utilizar, y requiere una extensa documentación, ya que pretende prever todo de antemano. Este tipo de metodologías son más eficaces y necesarias cuanto mayor es el proyecto que se pretende realizar respecto a tiempo y recursos que son necesarios emplear, donde una gran organización es requerida (28). Con lo anteriormente mencionado y teniendo en cuenta que las metodologías ágiles llevan a cabo un proceso menos controlado, están concebidas para grupos de trabajo pequeños, generan pocos artefactos, define

pocos roles y hace menos énfasis en la arquitectura de *software*, se ha decidido seleccionar una metodología pesada ya que el resultado de la presente investigación será incorporado a un proyecto grande en cuanto a tamaño y duración, donde el cliente no forma parte del equipo de desarrollo y donde la arquitectura del *software* es un elemento esencial. La comparación se realizará entre las metodologías pesadas: Proceso Unificado de "Rational" (Rational Unified Process o RUP según sus siglas en inglés) y MSF (Microsoft Solution Framework).

RUP: Es una metodología pesada que 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. Puede ser adaptado y extendido para satisfacer las necesidades de la organización que lo adopte. Es guiado por casos de uso (CU), iterativo e incremental y centrado en la arquitectura, utiliza el Lenguaje Unificado de Modelado (Unified Model Language o UML según sus siglas en inglés) como lenguaje de notación (28).

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. Durante la fase de inicio las iteraciones ponen mayor énfasis en las actividades de modelado del negocio y de requisitos. En la fase de elaboración, las iteraciones se orientan al desarrollo de la línea base de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la línea base de la arquitectura. En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones. En la fase de transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios (29).

MSF: Es un compendio de las mejores prácticas en cuanto a administración de proyectos se refiere. Más que una metodología rígida de administración de proyectos, MSF es una serie de modelos que puede adaptarse a cualquier proyecto de tecnología de información (28).

MSF está separado en cinco fases: la fase de Visión y Alcances trata uno de los requisitos más fundamentales para el éxito del proyecto, la unificación del equipo detrás de una visión común; se definen los líderes y responsables del proyecto, adicionalmente se identifican las metas y objetivos a alcanzar. La segunda fase es la Planeación donde el equipo prepara las especificaciones funcionales, realiza el proceso de diseño de la solución, y prepara los planes de trabajo, estimaciones de costos y cronogramas

de los diferentes entregables del proyecto. La tercera fase es: Desarrollo, durante esta fase el equipo realice la mayor parte de la construcción de los componentes. Estabilización es el nombre de la cuarta fase en la que se conducen pruebas sobre la solución. La quinta y última fase es la de Implantación donde el equipo implanta la tecnología base y los componentes relacionados, estabiliza la instalación y obtiene la aprobación final del cliente.

Selección de la metodología a utilizar

Con el estudio realizado acerca de las metodologías de desarrollo se llegó a la conclusión que se debe usar RUP, ya que define bien los procesos, las tareas a realizar, herramientas a utilizar y logra una detallada documentación, adaptándose al contexto y a las necesidades de cada organización. Conjuntamente es una metodología robusta y bien definida que posibilita la programación orientada a objetos y permite llevar a cabo un proceso de desarrollo práctico, brindando amplias guías, plantillas y ejemplos para todas las actividades. Al inicio de cada capítulo son mencionadas las actividades, artefactos y flujos de trabajo que se realizarán de la metodología de desarrollo para la presente investigación.

1.4.2 Lenguaje de modelado

UML 2.0

UML se ha convertido en ese estándar tan ansiado para representar y modelar la información con la que se trabaja en las fases de análisis y, especialmente, de diseño. El lenguaje UML tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, hasta la implementación y configuración con los diagramas de despliegue (30).

1.4.3 Herramienta de modelado UML

Visual Paradigm 8.0

Se puede definir a las Herramientas de Ingeniería de Software Asistida por Computadora (Computer Aided Software Engineering o CASE según sus siglas en inglés) como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida del desarrollo de un software. Para modelar los artefactos de la propuesta de solución se utilizan las potencialidades que brinda Visual Paradigm (31).

Visual Paradigm es una herramienta CASE multiplataforma de modelado visual, que facilita la construcción de artefactos en un proceso de desarrollo de software mediante UML. Es fácil de instalar y actualizar, además de ser compatible entre ediciones. Soporta una amplia gestión de casos de uso, diseño de base de datos relacionales y proporciona medidas más eficaces en el análisis y diseño de sistemas. Algunas de las funcionalidades que brinda esta herramienta CASE son la generación de código y de base de datos a partir de los diagramas UML realizados, la realización de ingeniería inversa, generación automática de informes en formato PDF, Word o HTML. Admite exportar e importar ficheros de proyectos realizados, así como exportar los artefactos que genera en los formatos JPG, PNG y SVG (32).

1.4.4 Lenguajes de desarrollo

Un lenguaje de programación o desarrollo es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente. En el caso del desarrollo de aplicaciones web, se dividen en dos grupos: lenguajes del lado del cliente y lenguajes del lado del servidor (33).

Lenguaje del lado del servidor

La programación del lado del servidor es una tecnología que consiste en el procesamiento de una petición de un usuario mediante la interpretación de un *script* (código) en el servidor web para generar páginas HTML dinámicamente como respuesta (34).

PHP 5

Es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. PHP está enfocado principalmente a la programación de *scripts* del lado del servidor, y se utiliza para generar páginas web dinámicas. Con PHP puedes procesar la información de formularios, generar páginas con contenidos dinámicos, o enviar y recibir *cookies*. Puede usarse en todos los principales sistemas operativos, incluyendo Linux. Una de las características más potentes y destacables de PHP es su soporte para un amplio abanico de bases de datos como mysql y postgresql (35).

1.5 Marco de trabajo

Un *framework* (marco de trabajo) simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, un *framework* proporciona

estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, un *framework* facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas (36).

1.5.1 Capa de lógica del negocio

Symfony 2.3.7

Symfony es un completo *framework* diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web (36).

Symfony se diseñó para que se ajustara a los siguientes requisitos:

- ❖ Fácil de instalar y configurar en la mayoría de plataformas.
- ❖ Independiente del sistema gestor de bases de datos.
- ❖ Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- ❖ Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional.
- ❖ Sigue la mayoría de las mejores prácticas y patrones de diseño para la web.
- ❖ Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros (36).

1.5.2 Capa de acceso a datos

Doctrine 2.1

Doctrine 2.1 es un Mapeador de Objetos Relacionales (ORM según sus siglas en inglés). Doctrine 2 es un ORM para PHP 5.3.0+ que proporciona persistencia transparente de objetos PHP. Se sitúa en la parte superior de una poderosa Capa de Abstracción de Base de Datos (DBAL según sus siglas en inglés). Una de las características clave de Doctrine es la opción de escribir las consultas de base de datos en un dialecto de Lenguaje de Consulta Estructurado (SQL según sus siglas en inglés) propio orientado a

objetos llamado Lenguaje de Consulta Doctrine (DQL según sus siglas en inglés), inspirado en Hibernate HQL. Además, DQL difiere ligeramente de SQL en que abstrae considerablemente la asignación entre las filas de la base de datos y objetos, permitiendo a los desarrolladores escribir poderosas consultas de una manera sencilla y flexible. Doctrine 2.1 viene integrado a Symfony 2.x, este framework permite usarlo como una librería, para facilitar a los desarrolladores el acceso a datos (37).

1.6 Ambiente de desarrollo integrado (IDE según sus siglas en inglés)

Un Entorno de Desarrollo Integrado es un conjunto de herramientas que facilitan el trabajo de los desarrolladores, puede estar orientado a uno o varios lenguajes de programación y brinda facilidades como: resaltado de sintaxis y completamiento de código.

Netbeans IDE 8.0

NetBeans es un reconocido IDE de código abierto y de distribución gratuita, apoyado por una amplia comunidad de desarrolladores. Permite que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación como: PHP. Brinda depuración y un potente completamiento de código. Además de soportar e integrar diferentes tipos de frameworks como Symfony, permitiendo la creación sencilla de aplicaciones web, agilizando así el desarrollo de la propuesta de solución. La integración con Symfony posibilita la ejecución de tareas tales como crear bundles, clases, limpiar caché, entre otras. Este IDE utiliza asistentes para crear servicios web y generar código para invocar los mismos (38).

1.7 Sistema gestor de bases de datos

Un Sistema Gestor de Base de Datos (SGBD) es un software que ofrece una interfaz entre la capa de datos, el usuario y las aplicaciones que la utilizan. Además de estar compuesto por un lenguaje de definición de datos, uno de manipulación de datos y uno de consulta.

PostgreSQL 9.1

Es un Sistema de Gestión de Bases de Datos basado en el proyecto POSTGRES, de la Universidad de Berkeley. Es multiplataforma y altamente escalable. Permite acomodar una enorme cantidad de datos de usuarios concurrentes. Se adapta para trabajar a alta velocidad en escenarios con gran carga de trabajo, además de constituir el SGBD familiarizado con el equipo de desarrollo (39).

Características principales:

- ❖ Implementación del estándar SQL92/SQL99.
- ❖ Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP), cadenas de bits, etc. También permite la creación de tipos propios.
- ❖ Incorpora una estructura de datos array.
- ❖ Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etcétera.
- ❖ Incluye herencia entre tablas, por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- ❖ Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos (39).

1.8 Servidor Web

Un Servidor Web es un programa informático para procesar las aplicaciones del lado del servidor, mediante conexiones que generan una respuesta desde las aplicaciones del lado del cliente.

Apache 2

Es un servidor web de código libre robusto cuya implementación se realiza de forma colaborativa, con prestaciones y funcionalidades equivalentes a las de los servidores comerciales. El proyecto está dirigido y controlado por un grupo de voluntarios de todo el mundo que, usando Internet y la web para comunicarse, planifican y desarrollan el servidor y la documentación relacionada (40).

Características:

- ❖ Corre en una multitud de sistemas operativos, lo que lo hace prácticamente universal.
- ❖ Apache es una tecnología gratuita de código fuente abierto.
- ❖ Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor web Apache. Actualmente existen muchos módulos para Apache que son adaptables a este.

- ❖ Apache te permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado *script* cuando ocurra un error en concreto (41).

1.9 Conclusiones del capítulo

- ❖ Se adoptaron los modelos de evaluación de aprendizaje de Tyler y Miller para una correcta ejecución de los procesos de evaluación por objetivos y competencias respectivamente.
- ❖ Se generalizó la utilización de la rúbrica como instrumento de evaluación para ambos modelos.
- ❖ Se determinó la necesidad de utilizar la herramienta Visual Paradigm 8.0 para garantizar la automatización del proceso de desarrollo.
- ❖ Se seleccionó el marco de trabajo Symfony 2 para hacer más sencilla la implementación de las funcionalidades gracias a las potencialidades que ofrece, y por otro lado el haber identificado el ORM Doctrine 2.1 permitirá manejar de una forma más simple la persistencia a datos.

Capítulo 2. Características de la propuesta de solución

2.1 Introducción

Con el objetivo de tener una mayor claridad del sistema que se desea desarrollar se realiza un Modelado del negocio, siendo este el primer flujo de trabajo propuesto por la metodología seleccionada, como resultado al problema planteado se obtiene el modelo de dominio, definiendo así los principales conceptos que intervienen en la investigación y sus relaciones. Posteriormente en el segundo flujo de trabajo: Requerimientos, uno de los más importantes pues define que tiene que hacer el sistema que se construya, en este flujo se realiza la definición de los requisitos funcionales y no funcionales permitiendo establecer las funciones y propiedades que el producto debe cumplir, además se identifican los actores que intervienen, los casos de uso y las relaciones que existen entre ellos a través del diagrama de casos de uso de la propuesta.

2.2 Modelo del dominio

Una vez analizado el negocio se comprueba que los procesos existentes no se encuentran bien definidos, se identificaron los conceptos presentes y se decide realizar un modelo de dominio, permitiendo mostrar al usuario los conceptos asociados al negocio, lo cual ayudará a los clientes, desarrolladores e interesados, a emplear un vocabulario común para entender el contexto en que se encuentra el sistema. Para la correcta definición de los requisitos se hace necesario tener bien definidos los conceptos asociados al negocio:

Objetivo: Es el resultado que se espera logre el alumno al finalizar un determinado proceso de aprendizaje.

Objetivo genérico: Brújula que guía el trabajo de los alumnos.

Objetivo específico: Concretan las características de los cambios que se espera obtener.

Objetivo operacional: Indica con precisión lo que se espera que los alumnos aprendan.

Competencia: Es el conjunto de los conocimientos, habilidades y actitudes que el estudiante va adquiriendo durante toda su vida.

Competencia genérica: Las competencias que debe tener vencidas el estudiante cuando finalice el programa de estudio.

Competencia específica: Estas competencias se refieren al saber hacer en una situación y contexto concreto.

Criterios de desempeño: Son el conjunto de habilidades que una persona debe cumplir para cumplir una competencia determinada.

Indicadores de logro: Actividades concretas que deben hacerse para cumplir con una habilidad determinada.

Conocimientos específicos: Contenidos que el estudiante debe saber para vencer una competencia específica.

Índice: Árbol de contenido donde están asociados los capítulos, temas y subtemas.

Docente: Usuario encargado de educar a los estudiantes en el proceso de enseñanza-aprendizaje.

Administrador central: Persona responsable de gestionar competencias y objetivos, asociar al índice de contenido objetivos específicos y competencias específicas; así como de crear el sistema educativo.

Tutor: Responsable legal de un determinado estudiante.

Editor: Persona que gestiona objetivos y competencias y asocia al índice objetivos específicos y competencias específicas.

Estudiante: Persona que cursa estudios para obtener el grado de bachillerato.

Evaluación: Proceso para medir el desarrollo docente de un estudiante.

Rúbrica: Estructura que se utiliza para evaluar una actividad determinada según una serie de aspectos evaluativos (42).

Materia: Conjunto de contenidos específicos comunes a un área específica del conocimiento, agrupados bajo una denominación genérica. En un plan de estudios una materia puede ser objeto del tratamiento de uno o varios cursos o asignaturas.

2.2.1 Diagrama del modelo de dominio

El modelo de dominio se describe mediante diagramas UML, específicamente con un diagrama de clases conceptuales con el objetivo de comprender y describir las clases más importantes dentro del contexto del sistema (43).

Para una mejor comprensión y legibilidad del modelo de dominio se decidió separarlo en dos partes. La primera parte del modelo de dominio comprende el proceso de gestión de las competencias u objetivos y la segunda parte comprende el proceso de evaluación, donde la clase “Evaluación” es el elemento vinculante de ambos modelos, quedando como sigue:

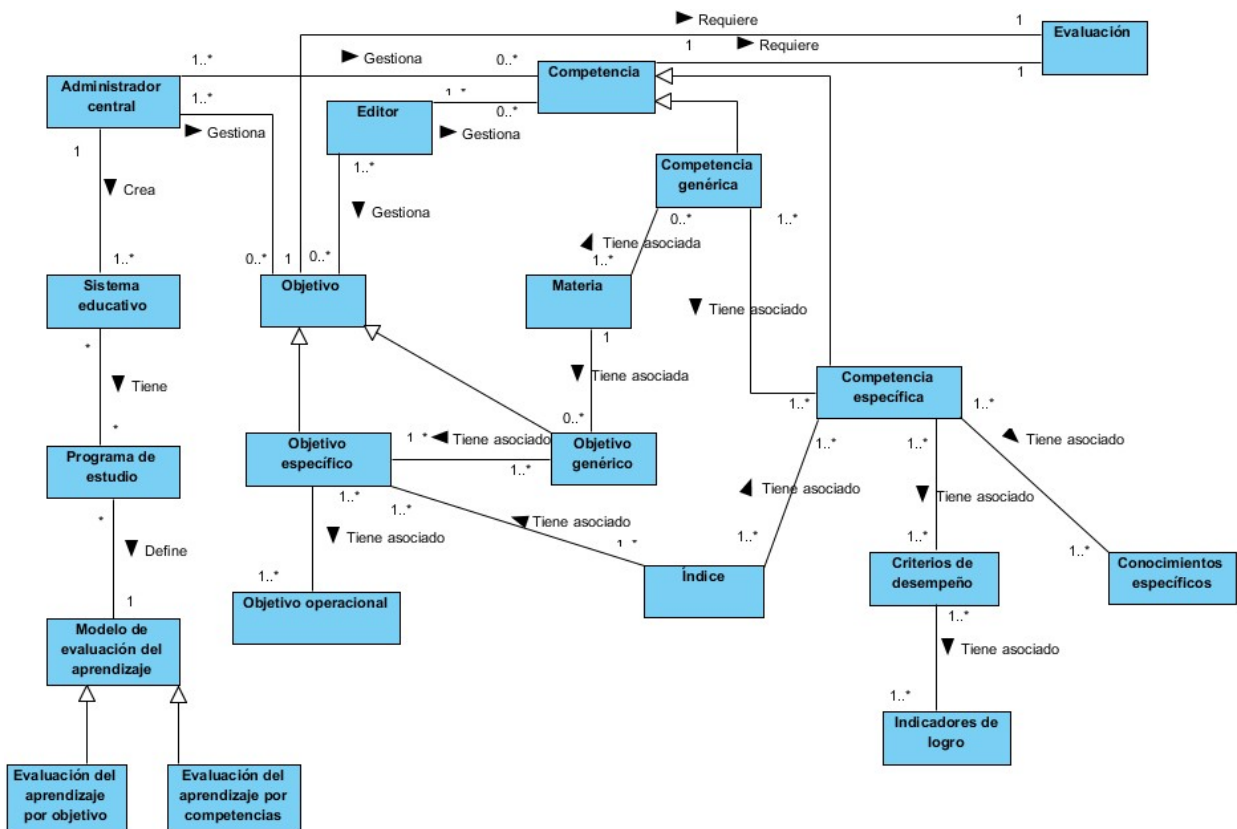


Figura 1. Modelo de dominio (primera parte)

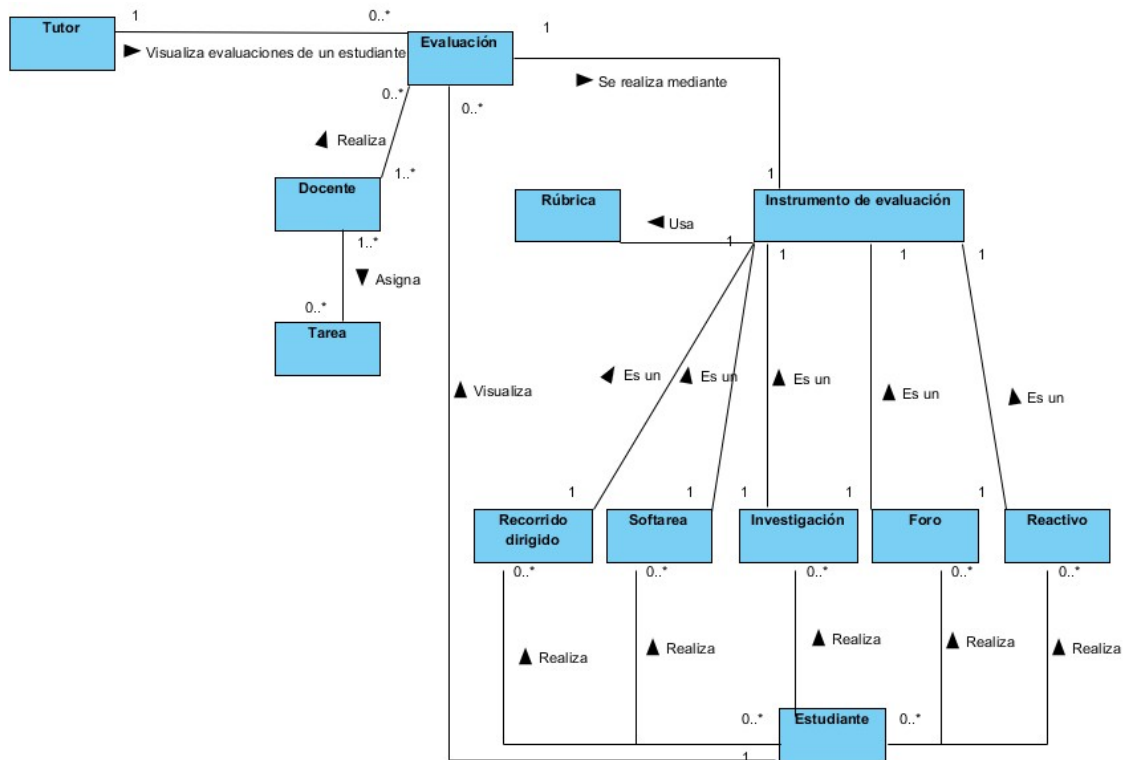


Figura 2. Modelo de dominio (segunda parte)

2.3 Descripción del sistema propuesto

2.3.1 Requisitos del software

En este apartado se especifican los requisitos funcionales y no funcionales existentes; representando las condiciones o capacidades que debe cumplir el sistema para lograr un entendimiento entre el cliente y los desarrolladores.

Requisitos funcionales

Los requisitos funcionales (RF) son funciones que el sistema debe cumplir:

RF1: Gestionar programa de estudio.

RF1.1: Seleccionar modelo de evaluación (competencia u objetivo).

RF2: Gestionar objetivo.

RF2.1: Incluir objetivo.

RF2.2: Ver objetivo.

RF2.3: Modificar objetivo.

RF2.4: Eliminar objetivo.

RF2.5: Asociar objetivo al índice de contenido.

RF2.6: Incluir datos para conformar una rúbrica.

RF3: Gestionar objetivo operacional.

RF3.1: Incluir objetivo operacional.

RF3.2: Modificar objetivo operacional.

RF3.3: Ver objetivo operacional.

RF3.4: Eliminar objetivo operacional.

RF3.5: Asociar objetivo operacional al índice de contenido.

RF4: Listar objetivos.

RF4.1: Mostrar un listado con todos los objetivos

RF4.2: Filtrar objetivos por el nombre

RF5: Listar los objetivos operacionales.

RF5.1: Mostrar un listado con todos los objetivos operacionales.

RF6: Asociar competencia específica u objetivo específico (Foro, Reactivo, Softarea, Recorrido dirigido, Evidencia).

RF7: Gestionar competencia.

RF7.1: Incluir competencia.

RF7.2: Modificar competencia.

RF7.3: Ver competencia.

RF7.4: Eliminar competencia.

RF7.5: Asociar competencia al índice de contenido.

RF8: Gestionar criterio de desempeño.

RF8.1: Incluir criterio.

RF8.2: Modificar criterio.

RF8.3: Ver datos del criterio.

RF8.4: Eliminar criterio.

RF8.5: Asociar criterio al índice de contenido.

RF8.6: Incluir datos para conformar una rúbrica.

RF9: Gestionar indicador de logro.

RF9.1: Incluir indicador de logro.

RF9.2: Modificar indicador de logro.

RF9.3: Ver datos del indicador de logro.

RF9.4: Eliminar indicador de logro.

RF9.5: Asociar indicador de logro al índice de contenido.

RF10: Listar las competencias.

RF10.1: Mostrar un listado con todas las competencias.

RF10.2: Filtrar competencias por el nombre.

RF11: Listar los criterios de desempeño.

RF11.1: Mostrar un listado con todos los criterios de desempeño.

RF12: Listar los indicadores de logro.

RF12.1: Mostrar un listado con todos los indicadores de logro.

RF13: Evaluar competencia específica u objetivo específico (Foro, Reactivo, Softarea, Recorrido dirigido, Evidencia).

RF13.1: Evaluar mediante la rúbrica.

RF14: Evaluar competencia específica u objetivo específico de forma general.

RF15: Evaluar competencias genéricas u objetivos genéricos.

RF16: Listar competencias específicas u objetivos específicos para el docente.

RF16.1: Personalizar competencias específicas u objetivos específicos.

RF17: Visualizar estado de las competencias u objetivos para el docente.

RF18: Visualizar competencias u objetivos del estudiante.

RF19: Visualizar competencias específicas u objetivos específicos (Foro, Reactivo, Softarea, Recorrido dirigido, Evidencia).

Requisitos no funcionales

Los requisitos no funcionales (RNF) especifican propiedades que debe cumplir el sistema.

RNF de software

Contar con uno de los siguientes sistemas operativos:

- ❖ Windows 7 o superior.
- ❖ Ubuntu 10.4 o superior (recomendado).

RNF de soporte:

- ❖ Compatibilidad con los navegadores: Explorer 7.0 o superior, Mozilla Firefox 14.0 o superior, Opera 10.6 o superior, Chrome 10.0 o superior.

RNF de hardware:

- ❖ Tarjeta de red.
- ❖ 512MB de RAM (o superior).
- ❖ 1GB de espacio en disco duro.
- ❖ Procesador Pentium 4 (o similar).
- ❖ Periférico: Mouse y teclado.

RNF de diseño e implementación:

- ❖ Los lenguajes utilizados para la integración de las funcionalidades implementadas en la máquina del servidor son los siguientes: Symfony2.3.7, Postgres SQL 9.x, PHP 5.3, Ubuntu 13.10 LTS.

RNF de rendimiento:

- ❖ Se debe garantizar que la velocidad de procesamiento de la información sea rápida.

2.4 Modelo de casos de uso del sistema

Se identifican y describen los actores del sistema, se realiza el diagrama de actores y el diagrama de casos de uso del sistema, así como la descripción de los casos de uso definidos.

2.4.1 Patrones de caso de uso

Son guías que describen como deben ser estructurados y organizados los casos de uso; permitiendo un mejor entendimiento de la interacción entre el actor y el sistema. Seguidamente se explican los patrones que se tuvieron en cuenta en la realización de los casos de uso y su representación en el diagrama de casos de uso.

CRUD (Create, Read, Update, Delete o Crear, Leer, Modificar, Eliminar)

Este patrón se basa en la fusión de casos de uso simples para formar una unidad conceptual.

- ❖ **Completo:** Este patrón consta de un caso de uso, llamado Información CRUD o gestionar información, modela todas las operaciones que pueden ser realizadas sobre una parte de la información de un tipo específico, tales como creación, lectura, actualización y eliminación. Suele ser utilizado cuando todos los flujos contribuyen al mismo valor del negocio, y estos a su vez son cortos y simples (44). Ejemplo de la utilización de este patrón son los: CU Gestionar competencia, CU Gestionar objetivo y CU Gestionar indicador de logro.

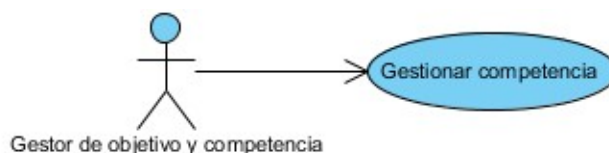


Figura 3. Patrón CRUD – Completo.

Múltiples actores

- ❖ **Roles comunes:** Puede suceder que los dos actores jueguen el mismo rol sobre el CU. Este rol es representado por otro actor, heredado por los actores que comparten este rol. Es aplicable cuando, desde el punto de vista del caso de uso, solo exista una entidad externa interactuando con cada una de las instancias del caso de uso (44). Ejemplo de la utilización de este patrón son los: CU Visualizar competencias u objetivos del estudiante y CU Listar indicador de logro.

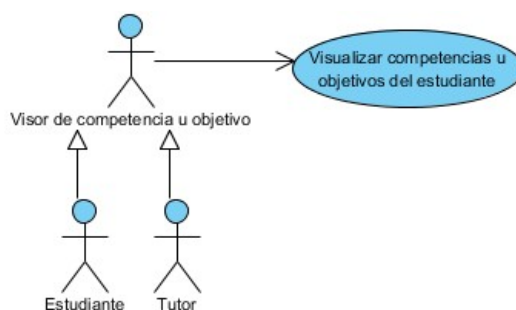


Figura 4. Patrón Múltiples actores – Roles comunes

Concordancia

Extrae una subsecuencia de acciones que aparecen en diferentes lugares del flujo de casos de uso y es expresado por separado.

- ❖ Adición: En el caso de este patrón alternativo, la subsecuencia común de casos de uso, extiende los casos de uso compartiendo la subsecuencia de acciones. Los otros casos de uso modelan el flujo que será expandido con la subsecuencia (44). Ejemplo de la utilización de este patrón es el: CU Visualizar competencias específicas u objetivos específicos en instrumentos de evaluación.



Figura 5. Patrón Concordancia– Adición

- ❖ Reusabilidad: Consta de 3 casos de uso. El primero llamado subsecuencia común, modela una secuencia de acciones que aparecerán en múltiples casos de uso en el modelo. Los otros casos de uso modelan el uso del sistema que comparte la subsecuencia común de acciones. De manera que deben existir al menos dos de ellos (44). Ejemplo de la utilización de este patrón es el: CU Evaluar competencia específica u objetivo específico asociado a los instrumentos.

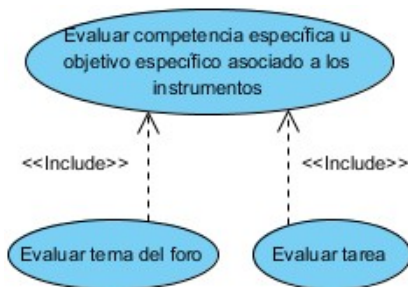


Figura 6. Patrón Concordancia-Reusabilidad

2.4.2 Descripción de los actores del sistema

Actor	Descripción
Docente	Actor que asocia los objetivos específicos y las competencias específicas a los instrumentos de evaluación, evalúa y personaliza los objetivos específicos y las competencias específicas, evalúa además, las competencias genéricas u objetivos genéricos, puede también consultar estado de las mismas.
Estudiante	Actor que puede ver los objetivos específicos y las competencias específicas

	asociados a cada instrumento de evaluación, así como ver las competencias u objetivos (genéricas y específicas) que tiene asociado, y sus respectivas notas.
Administrador central	Actor que gestiona las competencias y objetivos, crea los programa de estudio y define la estrategia de evaluación del aprendizaje a usar (evaluación por objetivo o evaluación por competencia),
Editor	Actor que gestiona las competencias y objetivos.
Director	Actor que podrá visualizar los objetivos y las competencias tanto genéricas como específicas.
Administrador local	Actor que podrá visualizar los objetivos y las competencias tanto genéricas como específicas.
Tutor	Actor que puede ver las competencias u objetivos (genéricas y específicas) que tiene asociado un determinado estudiante.
Visor de competencia y objetivo	Actor que representa la generalización de los actores estudiante y tutor, puede visualizar las evaluaciones de las competencias y objetivos desde el portafolio.
Gestor de objetivo y competencia	Actor que representa la generalización de los actores administrador central y editor, puede gestionar competencias u objetivos.
Usuario hiperentorno	Actor que representa la generalización de los actores: administrador local, tutor, estudiante, director y docente.

Tabla 2. Descripción de los actores del sistema

2.4.3 Diagrama de actores

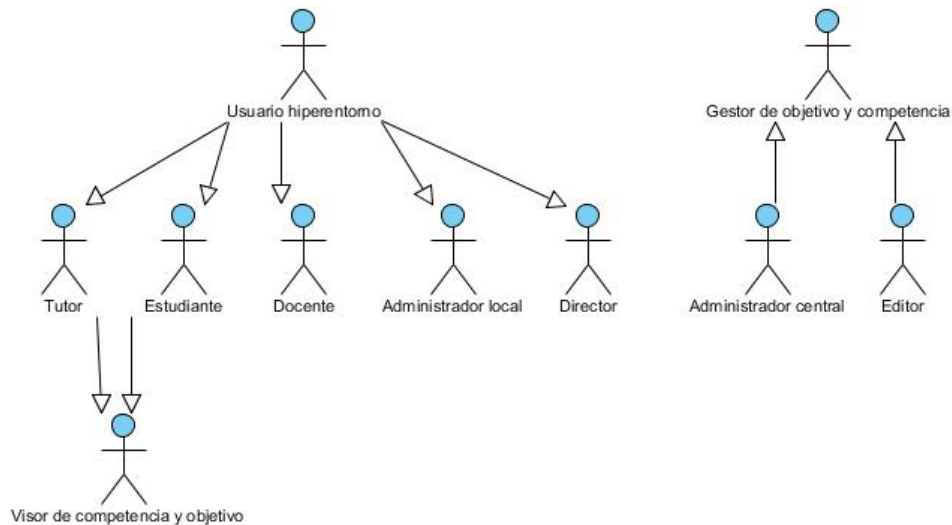


Figura 7. Diagrama de actores

2.4.4 Diagrama de casos de uso del sistema

El diagrama de casos de uso fue dividido en dos partes para lograr un mejor entendimiento. Estos están formados por los casos de uso definidos a partir de los RF identificados anteriormente, se muestran

además las relaciones existentes entre ellos y los actores que intervienen en cada uno. Es necesario destacar que los casos de uso identificados son 19, el resto de los mostrados pertenecen a la Plataforma Educativa ZERA y se relacionan con la presente investigación. Los casos de uso identificados son:

- ❖ Gestionar competencia.
- ❖ Gestionar objetivo.
- ❖ Gestionar Indicador de logro.
- ❖ Gestionar criterio de desempeño.
- ❖ Gestionar objetivo operacional.
- ❖ Listar objetivo.
- ❖ Listar competencia.
- ❖ Listar indicador de logro.
- ❖ Listar criterio de desempeño.
- ❖ Listar objetivo operacional.
- ❖ Seleccionar modelo de evaluación.
- ❖ Asociar competencia específica u objetivo específico.
- ❖ Evaluar competencia u objetivo específico asociado a los instrumentos.
- ❖ Evaluar competencia específica u objetivo específico de forma general.
- ❖ Evaluar competencia genérica u objetivo genérico.
- ❖ Listar competencia específica u objetivo específico para el docente.
- ❖ Visualizar competencias específicas u objetivos específicos en instrumentos de evaluación.
- ❖ Visualizar estado de las competencias u objetivos para el docente.
- ❖ Visualizar competencias u objetivos del estudiante.

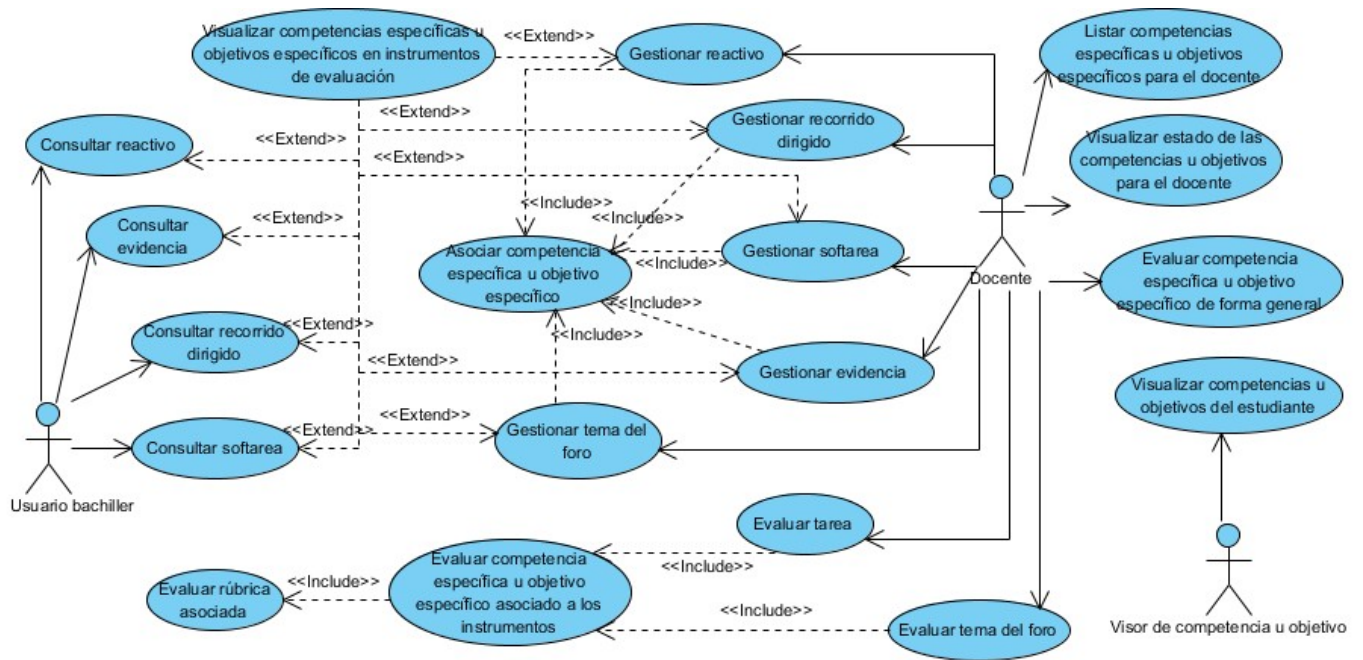


Figura 8. Diagrama de casos de uso del sistema (primera parte)

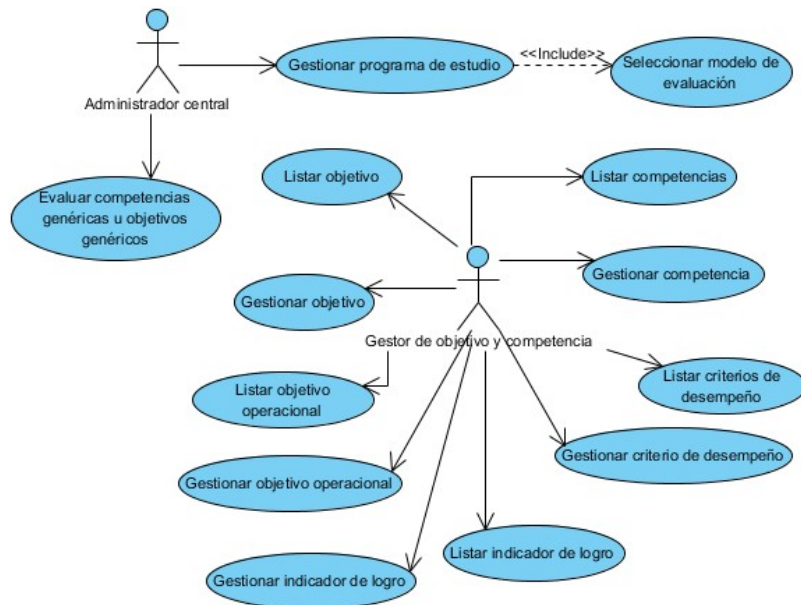


Figura 9. Diagrama de casos de uso (segunda parte)

2.4.5 Descripciones de los casos de uso del sistema

Los casos de uso son una guía para el trabajo (29). La descripción que se muestra inmediatamente detalla la relación entre un actor y el sistema del CU Evaluar competencia u objetivo específico asociado a los instrumentos. Para consultar las restantes de las descripciones de casos de uso dirigirse al Anexo 1: Descripciones de los casos de uso.

Objetivo	Evaluar las competencias específicas u objetivos específicos asociados a los diferentes instrumentos de evaluación (Foro, Reactivo, Evidencia, Recorrido dirigido, Softarea).	
Actores	Docente (inicia)	
Resumen	El caso de uso se inicia cuando el actor evalúa un instrumento y llega a la parte en que evaluará las competencias específicas u objetivos específicos que tiene asociada la tarea.	
Complejidad	Alta	
Prioridad	Alta	
Precondiciones	El escritorio de trabajo del usuario autenticado ha sido generado. El usuario autenticado se ha registrado con el rol docente.	
Postcondiciones	Se evaluaron las competencias específicas u objetivos específicos asociados a los instrumentos.	
Flujo de eventos		
Flujo básico Evaluar competencias específicas		
	Actor	Sistema
1.		Dependiendo del modelo de evaluación que tiene definido el programa de estudio: Permite: <ul style="list-style-type: none"> • Evaluar competencias específicas. • Evaluar objetivos específicos. <i>Ver sección 1 “Evaluar objetivos específicos”.</i>
2.	Accede a la opción de evaluar competencias específicas asociadas a un instrumento de evaluación (Reactivo, Recorrido dirigido, Softarea, Foro, Evidencia).	
3.		Muestra listado de las competencias específicas asociadas a la tarea, con los siguientes datos: <ul style="list-style-type: none"> • Nombre • Descripción • Criterios de desempeño Y permite: <ul style="list-style-type: none"> • Seleccionar competencia específica para evaluación

4.	Selecciona competencia específica para evaluarla.	
5.		Muestra los criterios de desempeño asociados a la competencia seleccionada y para cada criterio de desempeño su rúbrica asociada. Y permite: <ul style="list-style-type: none"> • Evaluar la rúbrica correspondiente a cada criterio de desempeño. • Guardar
6.	Evalúa cada una de las rúbricas asociadas a los criterios de desempeño de la competencia específica. <u>Ver CU Evaluar rúbrica asociada.</u>	
7.		Muestra evaluación calculada por la rúbrica de manera automática para cada criterio de desempeño.
8.		Calcula mediante la media geométrica las evaluaciones obtenidas de los criterios de desempeño de una competencia específica. Y muestra: <ul style="list-style-type: none"> • Nota (en porciento) de la competencia específica.
9.	Emite comentario (retroalimentación) de la nota obtenida para cada una de las competencias específicas.	
10.	Selecciona la opción de guardar.	
11.		Guarda los datos.
12.		Termina el caso de uso.
Sección 1 "Evaluar objetivos específicos"		
Flujo básico Evaluar objetivos específicos		
	Actor	Sistema
1.	Selecciona la opción de evaluar objetivos específicos asociados a un instrumento (Foro, Reactivo, Recorrido dirigido, Softarea, Evidencia).	
2.		Muestra listado de los objetivos específicos asociados a la tarea, con los siguientes datos: <ul style="list-style-type: none"> • Nombre • Descripción • Rúbrica asociada a cada objetivo específico

		Y permite: <ul style="list-style-type: none"> • Evaluar la rúbrica correspondiente a cada objetivo específico. <u>Ver CU Evaluar rúbrica asociada.</u> • Guardar.
3.	Evalúa la rúbrica asociada al objetivo específico. <u>Ver CU Evaluar rúbrica asociada</u>	
4.		Muestra evaluación calculada por la rúbrica de manera automática para el objetivo específico.
5.	Emite comentario (retroalimentación) de la nota obtenida para cada uno de los objetivos específicos.	
6.	Selecciona la opción de guardar.	
7.		Guarda los datos.
8.		Termina el caso de uso.
Relaciones		CU Incluidos
		CU Evaluar rúbrica asociada

Tabla 3. Descripción del CU Evaluar competencias específicas y objetivos específicos asociados a los instrumentos

2.5 Conclusiones del capítulo

- ❖ Basada en la metodología RUP se precisa la propuesta de solución que servirá como premisa para las actividades a desarrollar durante el transcurso de la investigación.
- ❖ Se lograron confeccionar los requisitos funcionales y no funcionales, los cuales fueron traducidos a un conjunto de casos de uso, representando así las principales funcionalidades del sistema.

Capítulo 3. Diseño e implementación de la propuesta de solución

3.1 Introducción

Análisis y diseño es el tercer flujo de trabajo que propone la metodología RUP. Dado que los resultados de esta investigación se incorporarán a la nueva versión de la Plataforma Educativa ZERA y la misma ya cuenta con una arquitectura, además de estar bien claros los resultados que se desean obtener, los requisitos son simples, bien conocidos y los desarrolladores cuentan con una correcta comprensión de los mismos; por lo anteriormente mencionado y dado que el costo de tiempo y recursos que consume la realización del modelo de análisis es alto, se dará prioridad al modelo de diseño siendo este la salida más importante del flujo de trabajo analizado, por ello se describió con mayor formalismo el modelo de casos de uso ya que es la entrada para realizar el diseño de la propuesta. Dentro de los artefactos que se generan en esta etapa se encuentran los diagramas de clases con estereotipos web y los diagramas de secuencia, ambos se realizan para cada uno de los casos de uso identificados. Además se define el patrón arquitectónico y los patrones de diseño a aplicar, el modelo de datos y se realiza el diagrama de despliegue. En este capítulo se realiza también el flujo de trabajo implementación, siendo este el centro durante la fase de construcción. Los resultados obtenidos en el diseño constituyen la entrada principal de este flujo de trabajo; describiendo cómo los elementos del modelo del diseño se implementan en términos de componentes, esto es materializado con el diagrama de componentes. En este capítulo también son realizadas las pruebas funcionales, verificando así la calidad del resultado de la implementación.

3.2 Patrón arquitectónico Modelo Vista Controlador (MVC) en Symfony

El uso de un framework que utiliza MVC obliga a dividir y organizar el código de acuerdo a las convenciones establecidas por el framework. El código de la presentación se guarda en la vista, el código de manipulación de datos se guarda en el modelo y la lógica de procesamiento de las peticiones constituye el controlador. Aplicar el patrón MVC a una aplicación resulta bastante útil además de restrictivo. La implementación que realiza Symfony de la arquitectura MVC incluye varias clases como son:

- ❖ Controlador (Controller): Es la clase del controlador y se encarga de decodificar la petición y transferirla a la acción correspondiente.
- ❖ Petición (Request): Guarda todos los elementos que integran la petición (parámetros, cookies, cabeceras)
- ❖ Respuesta (Response): Posee las cabeceras de la respuesta y los contenidos. El contenido de este objeto se convierte en la respuesta HTML que se remite al usuario.

Symfony toma lo mejor de la arquitectura MVC y la realiza de modo que el desarrollo de aplicaciones sea rápido y sencillo. En el controlador se encuentran las acciones, las cuales son el núcleo de la aplicación, pues contienen toda la lógica de la aplicación. Estas acciones utilizan el modelo y precisan las variables para la vista. Al realizarse una petición web en una aplicación Symfony, la URL define una acción y los parámetros de la petición. La vista es la encargada de originar las páginas que son mostradas como resultado de las acciones, donde se encuentra el *layout*, que es común para todas las páginas de la aplicación. La vista en Symfony está conformada por varias partes, preparadas cada una de ellas especialmente para ser fácilmente transformable por la persona que normalmente trabaja con cada aspecto del diseño de las aplicaciones. En el modelo se encuentran las clases, que son generadas de forma automática según la estructura de la base de datos. En Symfony, el acceso y la modificación de los datos que se almacenan en la base de datos, se realiza mediante objetos. Doctrine es el motor generador que se encarga de esta generación automática para construir sus clases, creando la estructura y generando el código de las mismas. Para el desarrollo de la propuesta de solución fueron utilizadas dos capas de esta arquitectura; el modelo, para crear las entidades que conforman el modelo de datos y el controlador, para publicar las funcionalidades que facilitan la implementación del proceso de evaluación por objetivos y competencias haciendo uso de servicios web (36).

3.3 Patrones de diseño

Se define un patrón como el fragmento de un modelo que es recurrente. Un patrón es una solución a un problema específico que se ha mantenido a pesar del tiempo (45).

3.3.1 Patrones de diseño de Bases de datos

Los patrones de diseño de base de datos (BD) permiten al usuario crear una BD más fortalecida, ya que constituyen una guía que especifica cómo debe ser la misma. El diseño y construcción de una BD requiere

del mayor esfuerzo y análisis posible, ya que a partir de este diseño es que esta se crea y la calidad con que se obtenga determinará su comportamiento futuro (45).

Árboles

Un árbol es un término de la teoría de grafos. Un árbol es un conjunto de nodos conectados en la estructura de hijo a padre y que se asemeja a la forma de un árbol. Un nodo es la unidad sobre la que se construye el árbol y puede tener cero o más nodos hijos conectados a él. Se dice que un nodo *a* es padre de un nodo *b* si existe un enlace desde *a* hasta *b* (en ese caso, también decimos que *b* es hijo de *a*). En un árbol no existen los ciclos, por lo que un camino sólo conecta a dos nodos.

Los árboles son sumamente utilizados en los diseños actuales, por lo que existen varios patrones de diseño de BD asociados a los mismos:

- ❖ Árboles fuertemente codificados (Hardcoded tree): En este caso a cada nivel del árbol se le asocia una entidad. Normalmente constituyen relaciones de 1 a muchos (n). Es comúnmente utilizado para representar jerarquías donde es bien conocida la estructura y es importante representar la correspondencia, por ejemplo las estructuras organizacionales (45).

Correspondencia entre las competencias:

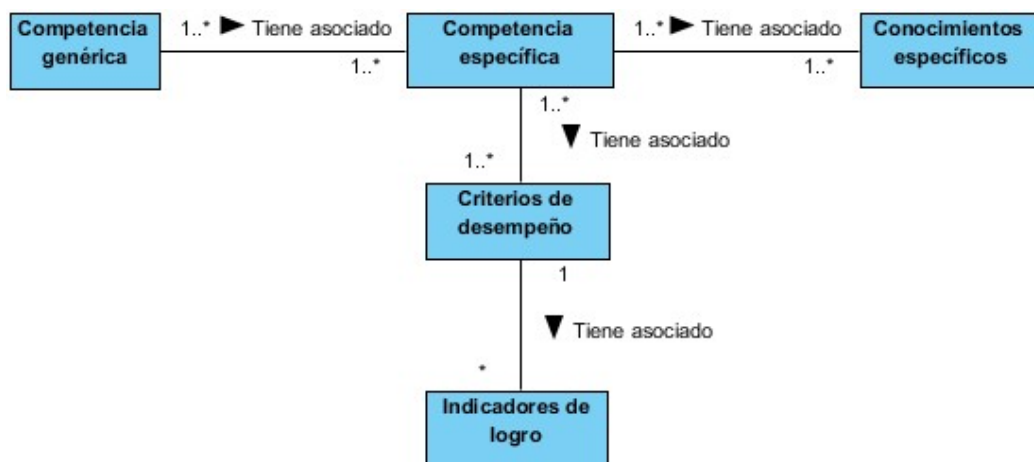


Figura 10. Ejemplo de árbol fuertemente codificado

Correspondencia entre los objetivos:

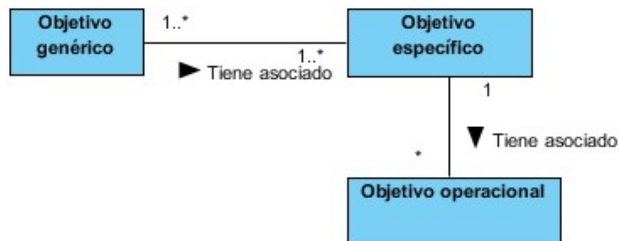


Figura 11. Ejemplo de árbol fuertemente codificado

- ❖ Árboles simples (Simple tree): Patrón normalmente utilizado cuando el árbol es la representación de una estructura de datos. Los elementos a almacenar son del mismo tipo, es decir, pueden ser almacenados en la misma entidad. No pueden existir ciclos, es decir, un hijo no puede ser su propio padre (45).

Representación de las competencias:

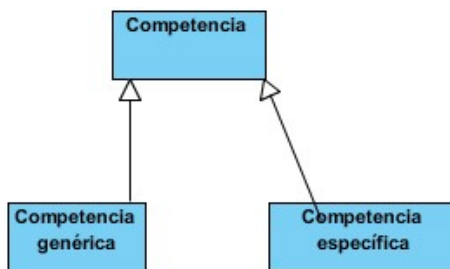


Tabla 4. Ejemplo de Árbol simple

Representación de los objetivos:

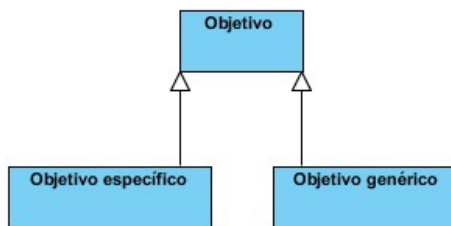


Tabla 5. Ejemplo de Árbol simple

3.3.2 Patrones GRASP

GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns, traducido al español como: Patrones Generales de Software para Asignar Responsabilidades; estos como su nombre lo indica, describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones.

Para la asignación de responsabilidades en la implementación de la capa de abstracción se emplean los siguientes patrones GRASP:

- ❖ Experto: Es uno de los patrones que más se utiliza cuando se trabaja con Symfony, con la inclusión de la librería Doctrine para mapear la BD y así realizar su capa de abstracción en el modelo, donde se generarían las clases con todas las funcionalidades comunes de las entidades (36). En las clases *CompetenceEval*, *ObjetivesEval*, *InstrumentEval* se utiliza esta librería, con el propósito de encapsular todo el acceso a los datos, a través de un grupo de funcionalidades. Estas funcionalidades serán las que estarán relacionadas directamente con las entidades que representan el modelo de datos.
- ❖ Creador: En las clases *CompetenceEval*, *ObjetivesEval*, *InstrumentEval* se encuentran las acciones definidas para el sistema y se ejecutan en cada una de ellas. En dichas acciones se crean los objetos de las clases que representan las entidades, lo que evidencia que las clases *CompetenceEval*, *ObjetivesEval*, *InstrumentEval* son “creador” de dichas entidades.
- ❖ Alta Cohesión: Symfony permite la organización del trabajo en cuanto a la estructura del proyecto y la asignación de responsabilidades con una alta cohesión. Un ejemplo de ello es la clase *EvaluationController*, la cual está formada por varias funcionalidades que están estrechamente relacionadas, siendo la misma la responsable de definir las acciones para las plantillas y colaborar con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades.
- ❖ Bajo Acoplamiento: Las clases *EvidenceEval*, *ForumEval*, *SoftTaskEval*, *DirectedPathEval* representan los instrumentos de evaluación y las conforman los métodos encargados de la gestión de los mismos. La clase *InstrumentEval* es la encargada de utilizar estas clases para gestionar los instrumentos de evaluación, alcanzando un bajo acoplamiento de clases y mayor escalabilidad, lo que proporciona que la dependencia en este caso sea baja.
- ❖ Controlador: En Symfony 2 todas las peticiones web son manipuladas por un solo controlador frontal (*Controller*), que es el punto de entrada único de toda la aplicación en un entorno determinado. Los

eventos se separan en varias acciones (*Actions*) para que aumente la cohesión y disminuya el acoplamiento.

3.3.3 Patrones GoF

GoF es el acrónimo de Gang of Four, traducido al español como Grupo de cuatro. Seguidamente se muestra la clasificación de estos patrones según su propósito:

- ❖ De creación: conciernen al proceso de creación de objetos.
- ❖ De estructura: tratan la composición de clases y objetos.
- ❖ De comportamiento: caracterizan las formas en las que interactúan y reparten responsabilidades las distintas clases u objetos (46).

Dentro de los estructurales se utiliza el patrón **Facade** (fachada) que simplifica el acceso a un conjunto de clases proporcionando una única clase que es utilizada por todos para comunicarse con dicho conjunto de clases. Con el uso de este patrón los clientes no necesitan conocer las clases que hay tras la clase Facade y además se pueden cambiar las clases “ocultadas” sin necesidad de cambiar los clientes. Solo hay que realizar los cambios necesarios en Facade (46). Ejemplo de clases que usan este patrón son CompetencesEval, ObjectivesEval, InstrumentsEval que son las que contienen las diferentes funcionalidades que se comunican con las demás clases del sistema.

En los patrones de comportamiento se encuentra el patrón **Strategy** (estrategia) cuya intención es encapsular algoritmos relacionados en clases y hacerlos intercambiables, además, permite que la selección del algoritmo se haga según el objeto que se trate (46). Ejemplo de esto es cuando se selecciona el tipo de instrumento de evaluación a gestionar, se define la clase conveniente de la capa de abstracción para ese proceso: *InstrumentsEval*.

3.4 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 requisitos funcionales y no funcionales tienen impacto en el sistema a considerar. Además, el modelo de diseño sirve de abstracción de la implementación del sistema y es de ese modo, utilizado como una entrada fundamental de las actividades de implementación (43).

3.4.1 Diagramas de clases del diseño

Muestran la estructura estática del modelo, en particular, los elementos que existen como clases, su estructura interna y sus relaciones con otras clases. A continuación se muestra el diagrama de clase del diseño (DCD) del CU Evaluar competencia u objetivo específico asociado a los instrumentos, los restantes DCD correspondientes al resto de los CU se encuentran en el Anexo 2: Diagramas de clases del diseño.

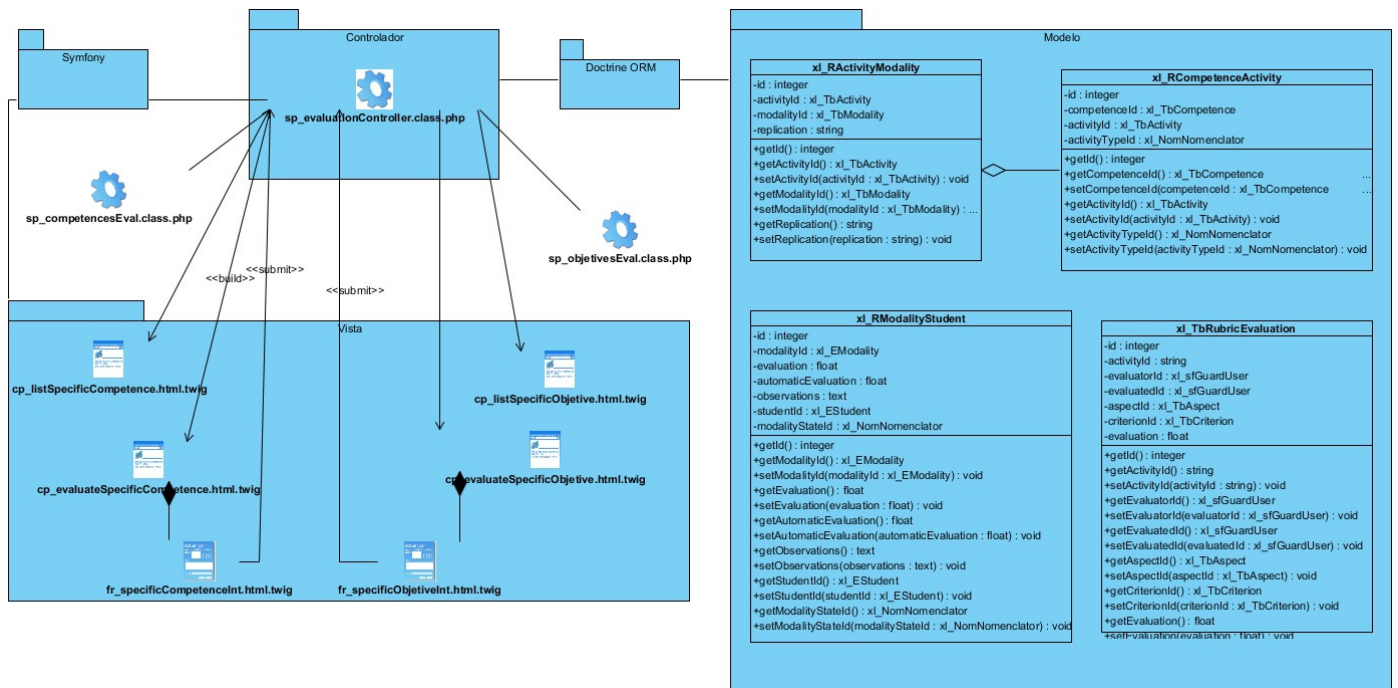


Figura 12. Diagrama de clases del diseño del CU Evaluar competencia u objetivo específico asociado a los instrumentos

3.4.2 Diagramas de interacción

Un diagrama de interacción consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos. El UML define dos tipos de diagramas de interacción (47):

- ❖ Diagramas de colaboración
- ❖ Diagramas de secuencia

En el diseño, es preferible representar estas relaciones con diagramas de secuencia ya que el centro de atención principal es encontrar secuencias de interacciones detalladas y ordenadas en el tiempo. Seguidamente se muestra el diagrama de secuencia (DS) correspondiente al CU Evaluar competencia u

objetivo específico asociado a los instrumentos, sección Objetivo, los restantes DS se encuentran en el Anexo 3: Diagramas de secuencia.

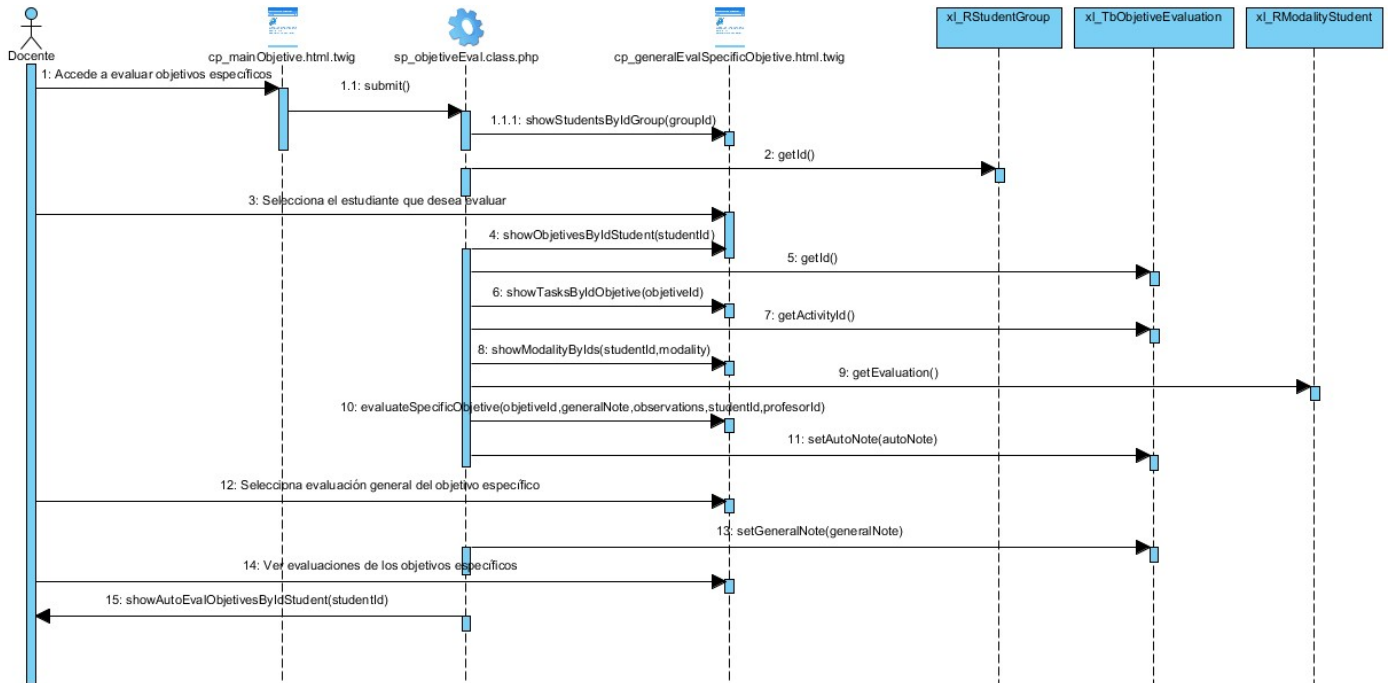


Figura 13. DS del CU Evaluar competencia específica u objetivo específico asociado a los instrumentos: sección Objetivo

3.5 Diseño de la base de datos

El diseño del modelo de datos consiste en describir la estructura de la base de datos representando la porción que se quiere almacenar en la BD. El modelo de datos que se muestra seguidamente contiene las relaciones que existen entre las entidades críticas que permiten realizar la evaluación del aprendizaje por objetivo y por competencia. Para consultar el modelo de datos íntegro, ver el Anexo 4: Modelo de datos.

3.5.1 Descripciones de las tablas

XIECompetenceEvaluation		
Descripción: Esta tabla almacena los datos de la evaluación de una competencia.		
Tipo	Atributo	Descripción
Integer	id	Es el identificador de cada registro de la tabla.
float	autoNote	Es la nota automática de la competencia.
float	generalNote	Es la nota general de la competencia.
string	observation	Es la observación de la competencia.
XIECompetence	competenceld	Es el identificador de la competencia.
XIEActivity	activityId	Es el identificador de la actividad.
XIEStudent	studentId	Es el identificador del estudiante.
XIsfGuardUser	profesorId	Es el identificador del profesor.

Tabla 6: Descripción de la tabla XIECompetenceEvaluation

XIEObjectiveEvaluation		
Descripción: Esta tabla almacena los datos de la evaluación de un objetivo.		
Tipo	Atributo	Descripción
integer	id	Es el identificador de cada registro de la tabla.
float	autoNote	Es la evaluación automática del objetivo.
float	generalNote	Es la evaluación general del objetivo.
string	observation	Es la observación del objetivo.
XIEObjetives	objetiveld	Es el identificador del objetivo.
XIEActivity	activityId	Es el identificador de la actividad a la que está asociado el objetivo.
XIEStudent	<u>studentId</u>	Es el identificador del estudiante.
XIsfGuardUser	profesorId	Es el identificador del profesor.

Tabla 7: Descripción de la tabla XIEObjectiveEvaluation

Para consultar las restantes descripciones remitirse al Anexo 5: Descripciones de las tablas de la base de datos.

3.6 Diagrama de despliegue

El diagrama de despliegue representa el hardware utilizado en la implementación del sistema y las relaciones entre sus componentes.

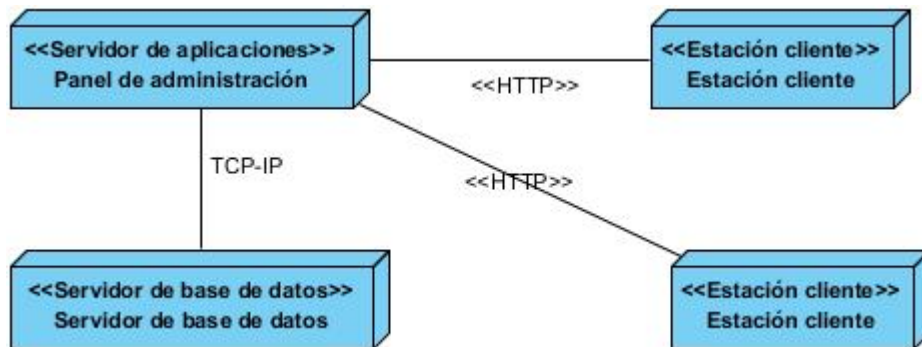


Figura 15. Diagrama de despliegue

3.7 Modelo de implementación

El modelo de implementación describe cómo los elementos del modelo de diseño, como las clases, se implementan en términos de componentes, como ficheros de código fuente, ejecutables etcétera. El modelo de implementación describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y como dependen los componentes unos de otros (43).

3.7.1 Diagrama de componente

Los diagramas de componente (DC) describen la organización de los elementos físicos que implementan el sistema, representando los componentes y las dependencias entre ellos. Se muestra el DC del CU Evaluar competencia u objetivo específico asociado a los instrumentos. Para la consulta de los restantes diagramas de componente, consultar el Anexo 6: Diagramas de componente.

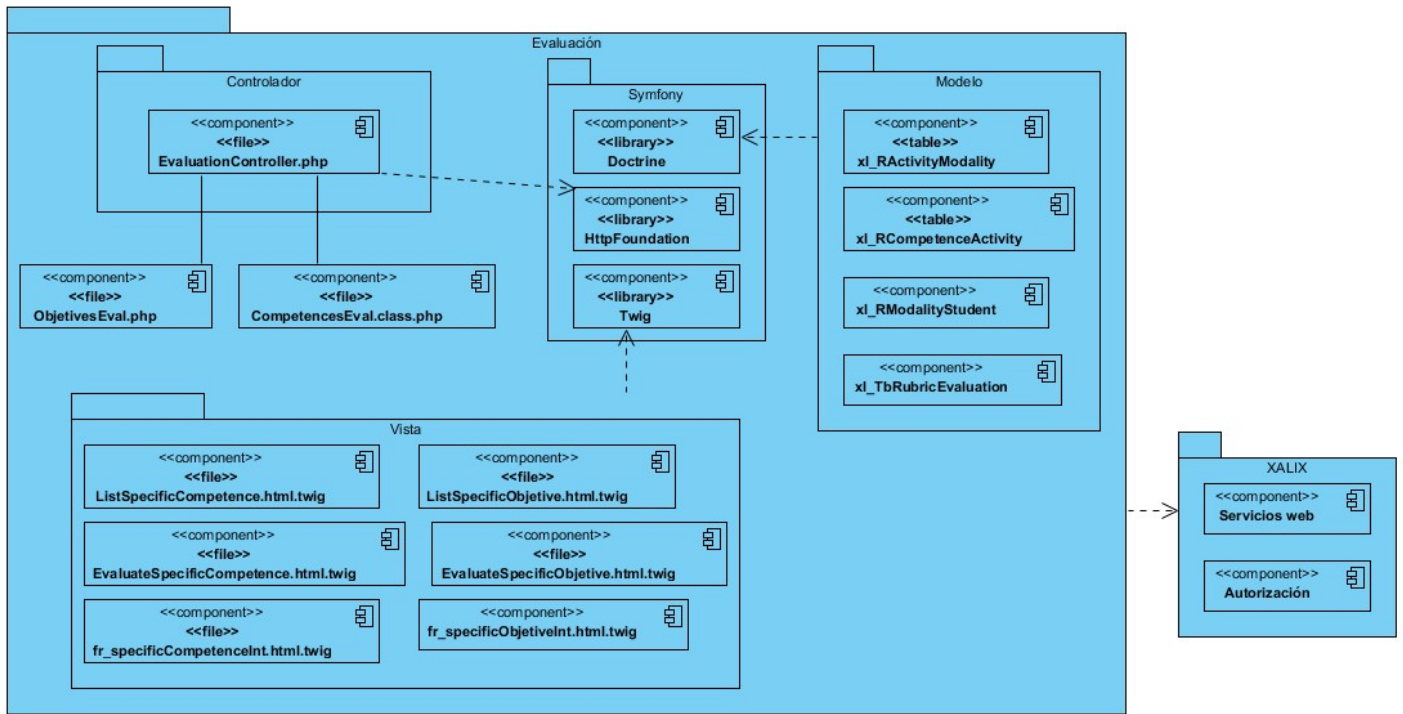


Figura 16. DC del CU CU Evaluar competencia específica u objetivo específico asociado a los instrumentos

3.8 Validación del diseño de las clases de la capa de abstracción

Con el objetivo de medir de forma cuantitativa la calidad del diseño de la capa de abstracción se emplean las métricas: Tamaño Operacional de la Clase (TOC) y Relación entre clase (RC). Las métricas empleadas están diseñadas para evaluar la calidad de los siguientes atributos.

La métrica TOC está dada por el número de métodos asignados a una clase y permite medir la responsabilidad asignada a una clase, la complejidad de implementación y el grado de reutilización de las clases del diseño. Es válido destacar que para esta métrica, la responsabilidad y la complejidad son inversamente proporcionales a la reutilización, por lo que a mayor responsabilidad y complejidad de implementación de una clase, menor será su nivel de reutilización.

3.8.1 Resultados obtenidos de la validación de la métrica TOC

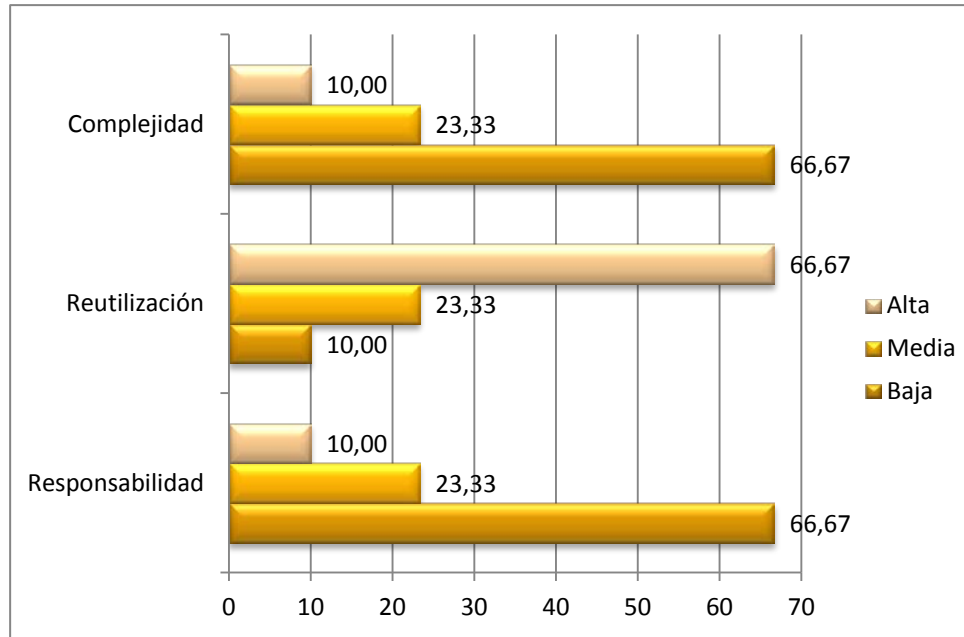


Figura 17. Representación de los resultados obtenidos de la validación de la métrica TOC

Luego de aplicada la métrica se pudo constatar que las clases de la capa de abstracción no son complejas, ni presentan un alto nivel de responsabilidad beneficiando esto a la reutilización de las mismas.

La métrica RC está dada por el número de relaciones de uso de una clase con otra y mide el acoplamiento, la complejidad de mantenimiento, la reutilización y la cantidad de pruebas unitarias necesarias para probar una clase.

3.8.2 Resultados obtenidos de la validación de la métrica RC

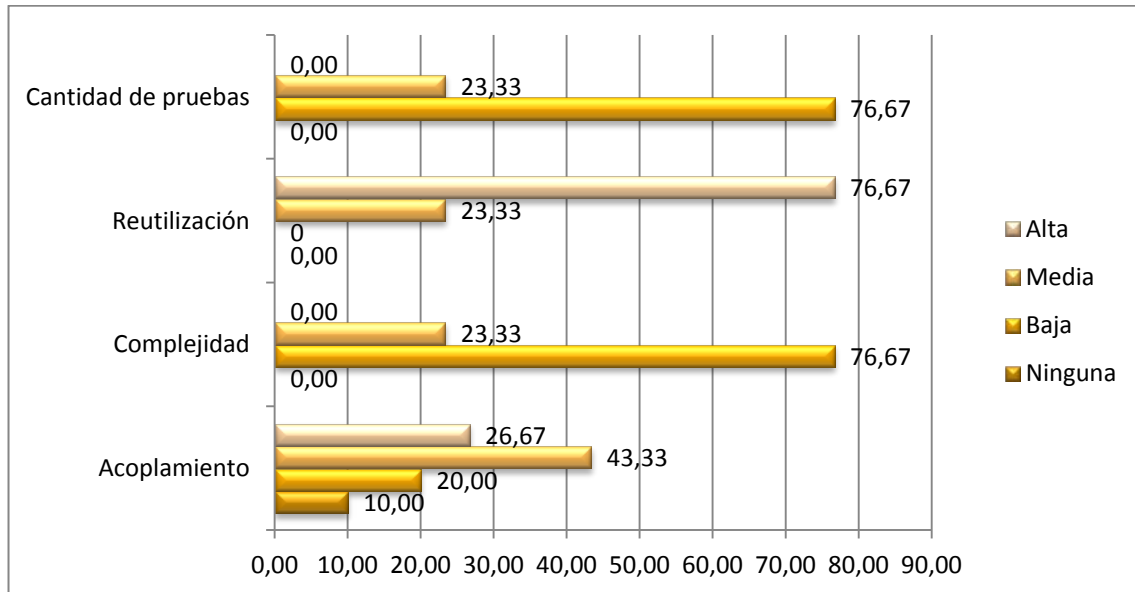


Figura 18. Representación de los resultados obtenidos de la evaluación de la métrica RC

Al aplicar esta métrica se obtiene que las clases del diseño presentan un grado de reutilización alto, la cantidad de pruebas y la complejidad es baja, y el acoplamiento es medio.

De manera general para conocer si los resultados obtenidos de las relaciones atributo/métrica es positivo o no, llevando estos resultados a una escala numérica (Bien [0.1; 0.3], Regular [0.4; 0.7] y Mal [0.8; 1]), donde si los resultados son positivos se le asigna el valor 1, si son negativos toma valor 0 y si no existe relación es considerada como nula y es representada con un guión simple (-). Obteniéndose como resultado de aplicar las métricas TOC y RC que el diseño de la capa de abstracción no es complejo, presenta un alto grado de reutilización y un bajo acoplamiento.

3.9 Pruebas de software

La realización de pruebas se centra en las fases de elaboración, cuando se prueba la línea base ejecutable de la arquitectura, y de construcción cuando el grueso del sistema está implementado (43). Las pruebas de software se realizan con la intención de descubrir errores previos a la entrega al usuario final. En esta actividad se ejecutan las funcionalidades implementadas bajo una serie de condiciones o requerimientos específicos, los resultados son observados y registrados para realizar una evaluación de las funcionalidades ejecutadas.

3.9.1 Niveles de prueba

La prueba es aplicada para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo. Existen diferentes niveles de pruebas entre los que se encuentran: Prueba de unidad, Prueba de integración, Prueba de sistema y Prueba de aceptación; para probar las funcionalidades implementadas fueron realizadas Pruebas de unidad.

Las Pruebas de unidad se centran en la verificación de los elementos más pequeños del software que se puedan probar. Normalmente, las Pruebas de unidad se aplican a componentes representados en el modelo de implementación para verificar que se cubren los flujos de control y los flujos de datos y que funcionan como se esperaba. El implementador realiza la Prueba de unidad mientras se desarrolla la unidad (48).

3.9.2 Métodos de prueba

Cualquier producto de ingeniería puede ser probado de una de estas formas:

- ❖ Conociendo la funcionalidad específica para la cual fue diseñado el producto, se pueden llevar a cabo pruebas que demuestren que cada función es completamente operativa.
- ❖ Conociendo el funcionamiento del producto se pueden desarrollar pruebas que aseguren que “todas las piezas encajen”, o sea, que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada.

El primer enfoque se denomina prueba de Caja negra y el segundo prueba de Caja blanca. La prueba de Caja blanca se basa en el minucioso examen de los detalles procedimentales. Se comprueban los caminos lógicos del software proponiendo casos de prueba que examinen que están correctas todas las condiciones o bucles para determinar si el estado real coincide con el esperado. Esto genera gran cantidad de caminos posibles por lo que hay que dedicar esfuerzos a la determinación de las condiciones de prueba que se van a verificar (49).

La prueba de Caja blanca se basa en el diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivarlos. Mediante esta se pueden obtener casos de prueba que:

- ❖ Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
- ❖ Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.

- ❖ Ejecuten todos los bucles en sus límites operacionales.
- ❖ Ejerciten las estructuras internas de datos para asegurar su validez.

Es por ello que se considera a la prueba de Caja blanca como uno de los tipos de pruebas más importantes que se le aplican a los software, logrando como resultado que disminuya en un gran porcentaje el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad (49).

3.9.3 Automatización de las pruebas

Cualquier programador con experiencia en el desarrollo de aplicaciones web conoce el esfuerzo que supone probar correctamente la aplicación. Crear casos de prueba, ejecutarlos y analizar sus resultados es una tarea compleja. Este es el motivo por el que la automatización de pruebas es una recomendación, aunque no una obligación, útil para crear un entorno de desarrollo satisfactorio. En ocasiones, las pruebas automatizadas pueden reemplazar la documentación técnica de la aplicación, ya que ilustran de forma clara el funcionamiento de la aplicación. Un buen conjunto de pruebas muestra la salida que produce la aplicación para una serie de entradas de prueba, por lo que es suficiente para entender el propósito de cada método (36).

Symfony 2 ha optado por utilizar la librería PHPUnit, que prácticamente se ha convertido en un estándar en el mundo PHP. De esta forma, los *tests* (pruebas) unitarios de Symfony 2 combinan la potencia de PHPUnit con las utilidades y facilidades proporcionadas por Symfony 2. Por convención, cada test unitario y funcional de Symfony 2 se define en una clase cuyo nombre acaba en *Test* y se encuentra dentro del directorio *Tests/* del bundle. Además, se recomienda utilizar dentro de *Tests/* la misma estructura de directorios del elemento que se quiere probar. PHPUnit utiliza *assertions* (aserciones) para verificar que el comportamiento de una unidad de código es el esperado. Cuando se produce un error, PHPUnit muestra el texto *"Failures"* (Fallo) como resumen de la ejecución. Antes muestra el listado de todos los *tests* que han fallado, indicando para cada error la clase y método erróneos, el mensaje propio que se incluyó en el *test* e información adicional como el valor esperado y el valor obtenido. Un test que no pasa satisfactoriamente es la mejor señal de que algo no funciona bien en la aplicación. Esta es la gran ventaja de los *tests* unitarios, que te avisan cada que vez hay problemas en la aplicación. El mensaje *"Ok"* (Bien) indica que todos los *tests* se han ejecutado correctamente (50).

3.9.4 Resultados obtenidos

A lo largo de la implementación de la capa de abstracción fueron realizadas por los desarrolladores Pruebas unitarias, haciendo uso de la librería PHPUnit, con el objetivo de obtener un producto con la calidad requerida; utilizando para la realización de las mismas el método de Caja blanca. Estas pruebas fueron realizadas a 30 de las clases de la capa de abstracción. A continuación se muestran ejemplos de las pruebas realizadas haciendo uso de la librería PHPUnit.

```
PHPUnit 3.6.10 by Sebastian Bergmann.
Configuration read from /var/www/tesis/app/phpunit.xml.dist
FFF.
Time: 0 seconds, Memory: 3.25Mb
There were 3 failures:
1) XLECompetenceEvaluationTest::testValidatAutoNote
La nota automatica no puede ser un número negativo
Failed asserting that 0 is greater than 0.
/var/www/tesis/src/Tesis/EvaluationBundle/Tests/Entity/XLECompetenceEvaluationTest.php:47
2) XLECompetenceEvaluationTest::testValidatGeneralNote
La nota general no puede ser un número negativo
Failed asserting that 0 is greater than 0.
/var/www/tesis/src/Tesis/EvaluationBundle/Tests/Entity/XLECompetenceEvaluationTest.php:61
3) XLECompetenceEvaluationTest::testValidateObservation
La observacion no puede dejarse en blanco
Failed asserting that 0 is greater than 0.
/var/www/tesis/src/Tesis/EvaluationBundle/Tests/Entity/XLECompetenceEvaluationTest.php:74
FAILURES!
Tests: 4, Assertions: 6, Failures: 3.
```

Figura 19. Prueba unitaria realizada a la entidad XLECompetenceEvaluationTest

```
PHPUnit 3.6.10 by Sebastian Bergmann.
Configuration read from /var/www/tesis/app/phpunit.xml.dist
FF
Time: 0 seconds, Memory: 3.00Mb
There were 2 failures:

1) XLEOperationalObjectiveTest::testValidateName
El nombre no puede dejarse en blanco
Failed asserting that 0 is greater than 0.

/var/www/tesis/src/Tesis/EvaluationBundle/Tests/Entity/XLEOperationalObjectiveTest.php:20

2) XLEOperationalObjectiveTest::testValidateDescription
La descripcion no puede dejarse en blanco
Failed asserting that 0 is greater than 0.

/var/www/tesis/src/Tesis/EvaluationBundle/Tests/Entity/XLEOperationalObjectiveTest.php:41

FAILURES!
Tests: 2. Assertions: 2. Failures: 2.
```

Figura 20. Prueba unitaria realizada a la entidad XLEOperationalObjectiveTest

Fueron realizadas dos iteraciones de las Pruebas unitarias, en la primera iteración se detectaron como errores que:

- ❖ Algunas funcionalidades no validaban los datos de entrada.
- ❖ No se validaba que todos los campos deben ser llenados, es decir que no se dejen campos vacíos.

Estos errores fueron corregidos de manera inmediata, y en la segunda iteración no se encontraron errores, por tanto las funcionalidades probadas funcionan correctamente.

En total fueron probadas 50 funcionalidades correspondientes a la capa de abstracción, los errores fueron encontrados en 20 de estas, lo que representa el 40% de las funcionalidades probadas.

3.10 Resumen comparativo

A continuación son evaluados un grupo de indicadores, seleccionados a partir de las deficiencias del modelo de evaluación que presenta la actual Plataforma Educativa ZERA. Estos indicadores son evaluados en las versiones anteriores de la Plataforma y en la nueva versión que se desarrolla, con el fin de verificar que se cumple con los objetivos de la investigación en un plano conceptual. Es importante destacar que no es posible validar, en la práctica, la aplicación de los modelos de evaluación propuestos debido a que la nueva versión de la Plataforma Educativa ZERA se encuentra aún en desarrollo.

Indicador	Versiones anteriores de la Plataforma	Nueva versión de la Plataforma
Modelo de evaluación por objetivos	No se realiza la gestión de los objetivos, y por tanto no existe la forma de evaluarlos.	Se garantiza la gestión de objetivos, su relación a las actividades de la Plataforma y su evaluación utilizando el modelo Tyleriano que define una serie de pasos que guían el proceso evaluativo. Los objetivos son divididos en objetivos genéricos y objetivos específicos definiendo una jerarquía entre ellos.
Modelo de evaluación por competencias	Presenta dificultades en la gestión y evaluación de las competencias; las competencias genéricas y específicas se encuentran en un mismo nivel jerárquico. Las competencias son evaluadas mediante una fórmula matemática.	Las competencias genéricas pueden abarcar varias materias y las competencias específicas son asociadas a un capítulo, tema o subtema de un programa de estudio. Para su evaluación se propone el Modelo de Miller y un conjunto de pasos definidos por especialistas en el tema, garantizando la correcta ejecución del modelo de evaluación por competencias.
Instrumentos de evaluación	Solo se emplea para la evaluación por competencias los instrumentos de evaluación: Reactivo, Evidencia y Foro. La rúbrica es usada solo para evaluar las tareas.	Se empleará para la evaluación por competencias y por objetivos los instrumentos de evaluación: Recorrido dirigido, Reactivo, Softarea, Evidencia y Foro. Se plantea el uso de la rúbrica para evaluar los objetivos y competencias, posibilitando grandes ventajas tanto para docentes como para estudiantes.

Tabla 8. Resumen comparativo

3.11 Conclusiones del capítulo

- ❖ Con el uso de los patrones de diseño GRASP y GoF se contribuyó a que el sistema contara con un diseño orientado al bajo acoplamiento, la reutilización y la alta cohesión; el logro de la independencia del sistema, de cómo sus objetos son creados, integrados y representados, garantiza que la modificación en los requisitos del sistema ocasione la menor cantidad posible de cambios en las relaciones entre objetos existentes.
- ❖ A través del modelo de diseño se logró la elaboración del sistema en pro de favorecer una arquitectura estable y sólida.
- ❖ Se realizó el modelo de datos que muestra las relaciones entre las entidades que conforman la base de datos; así como el modelo de implementación que representa las relaciones de los

componentes físicos donde se encuentran implementadas las funcionalidades y sus dependencias con la base de datos.

- ❖ Las funcionalidades implementadas fueron publicadas haciendo uso de los servicios web para aumentar interoperabilidad entre los diferentes componentes que se definan en la Plataforma.
- ❖ Fue generada la documentación correspondiente inherente a todo el proceso de implementación de la capa de abstracción, facilitando de su comprensión a los futuros desarrolladores.
- ❖ Se obtuvieron resultados satisfactorios que garantizaron la calidad de las funcionalidades implementadas después de haber realizado las correspondientes Pruebas unitarias.

Conclusiones generales

Luego del desarrollo de la presente investigación y el análisis de los resultados logrados se arribaron a las siguientes conclusiones:

- ❖ El estudio de los fundamentos teóricos permitió identificar los conceptos asociados a los modelos de evaluación por objetivos y por competencias, obteniendo de esta forma un modelo conceptual que posibilitó realizar el proceso de evaluación en ambientes virtuales de aprendizaje en diferentes materias y niveles de enseñanza, el cual da respuesta a la necesidad planteada en la problemática de la investigación.
- ❖ Se documentó e implementó, guiado por la metodología RUP, la base conceptual de los modelos de evaluación seleccionados para la nueva versión de la Plataforma Educativa ZERA, alcanzándose como resultando una capa de abstracción que proporciona un conjunto de funcionalidades para los futuros desarrollos.
- ❖ El resultado de las pruebas funcionales facilitó la verificación y la validación de la correspondencia entre los requisitos identificados y la implementación realizada, con lo que se obtuvo una capa de abstracción para estandarizar el proceso de evaluación por objetivos y por competencias, lográndose con ello una mejor eficiencia en el accionar evaluativo, tanto por parte de los estudiantes como de los profesores.

Recomendaciones

A partir de las conclusiones formuladas, se recomienda:

- ❖ Agregar nuevos instrumentos de evaluación en la Plataforma Educativa ZERA que incluya como tipo la coevaluación para favorecer el aprendizaje colaborativo y estimular el aprendizaje grupal entre los estudiantes.
- ❖ Incorporar a los modelos de evaluación del aprendizaje en la Plataforma Educativa ZERA el proceso de metaevaluación para espacios de autorreflexión y de reflexión colectiva acerca de cómo se desarrolla el proceso de aprendizaje.

Referencias Bibliográficas

1. STUFFLEBEAM, Daniel and SHINKFIELD, Anthony J. *Evaluación sistémica: Guía Teórica y práctica*. PAIDOS IBERICA, 1995. ISBN 9788475094458.
2. CAMPOS, Isabela S. *Nociones Básicas sobre evaluación*. Universidad Autónoma de Centro América.
3. MARIBONA, Mairelis G., ALMAGUER, Yusdel M. and RODRIGUEZ, Abel Ernesto L. Propuesta de instrumentos de evaluación para entornos virtuales de aprendizaje: una experiencia en la universidad de las ciencias informáticas. 2011. Vol. 8.
4. Cátedra de la UNESCO para la Educación a Distancia. *Centro Iberoamericano de Recursos para la Educación a Distancia (CIREAD)* [en línea]. [Accedido el 26 de Abril 2014]. Disponible en: <http://www.uned.es/catedraunesco-ead/equipo.html#16,7>
5. Plataformas Disponibles para el Desarrollo de Cursos Virtuales UNESCO. [en línea]. [Accedido el 26 de Abril de 2014]. Disponible en: <http://www.uned.es/catedraunesco-ead/cursos.htm>
6. BOZA, Msc. Leonardo H. *Moodle*. 12 de Mayo de 2014.
7. BOZA, Msc. Leonardo H., PIÑA, Juenlis Enrique C. and MONTEJO, Carlos A. Personalización, extensión y desarrollo de módulos para Moodle. Universidad de las Ciencias Informáticas. Cuba.
8. PÉREZ, Yuleisy G., VALDÉS, Harold O., PONS, Yaismel M. and SOBERATS, Daniel R. Gestión de los Contenidos del Microdiseño Curricular desde la Plataforma Educativa Zera. *Serie Científica de la Universidad de las Ciencias Informáticas*. 15 de Junio de 2012. Vol. 5, no. 6.
9. ZÚÑIGA, Ing. Oscar H. *Evaluación y Modelos de Enseñanza-Aprendizaje*. Universidad de Santander. 2007.
10. REYES, Virgilio H. and DOMÍNGUEZ, Manuel B. *Evaluación de los aprendizajes*. Marzo 2001. Universidad Peruana Cayetano Heredia.
11. IGLESIAS, Magalys R. *La evaluación basada en competencias*.
12. FUNDACIÓN INSTITUTO DE CIENCIAS DEL HOMBRE. *La evaluación educativa: conceptos, funciones y tipos*.
13. LLAMAS, José Luis G. *Evaluación de programas educativos: modelos, técnicas e instrumentos, análisis de datos y elaboración de informes*. Universidad Nacional de Educación a Distancia.
14. VARGAS, Ana Isabel M. *La Evaluación educativa: conceptos, períodos y modelos*. Universidad de Costa Rica, 13 de Diciembre de 2004.

15. MARTINIANO, Román P. Evaluación por objetivos en el proyecto Internacional PISA. *Portal Daniela Rojas* [en línea]. [Accedido el 29 de Enero de 2014]. Disponible en: http://www.rmm.cl/index_sub.php?id_seccion=1044&id_portal=180&id_contenido=8244
16. GONZÁLEZ, Teresa L. and REYES, María del Rosario G. *El proceso de aprendizaje por competencias*.
17. TOBÓN, Sergio. *Aspectos básicos de la formación basada en competencias*. 2006. Proyecto Mesesup.
18. Sistemas de Educación Media Superior. Evaluación del aprendizaje. Enfoque centrado en competencias. Universidad de Guadalajara. Agosto de 2009.
19. IGLESIAS, Dra. Magalys R. *MAESTRIA INTERNACIONAL DE COMPETENCIAS PROFESIONALES*. Junio de 2008.
20. FERNÁNDEZ-PAMPILLÓN CESTEROS, Ana. Las plataformas e-learning para la enseñanza y el aprendizaje universitario en Internet. [en línea]. 2009. [Accedido el 4 de diciembre de 2013]. Disponible en: <http://eprints.ucm.es/10682/>
21. BELLOCH, Consuelo. *Entornos Virtuales de Aprendizaje*. Unidad de Tecnología Educativa (UTE). Universidad de Valencia.
22. AÑORVE, Gladys A., MARÍN, Francisco G. and GARMENDIA, Esmeralda V. Instrumentos de evaluación. Universidad Pedagógica Nacional de México. Febrero de 2010.
23. CABRERA, Leonardo P. and GIL, Dunet C. *Integración de los instrumentos de evaluación para la gestión de las evidencias en la plataforma educativa Zera*. La Habana : Universidad de las Ciencias Informáticas, 2012.
24. FERNÁNDEZ, Ramón A. *Propuesta para evaluar aprendizajes virtuales*. 2009.
25. FERNANDEZ, José T. La evaluación de las competencias en contextos no formales: dispositivos e instrumentos de evaluación. Abril de 2011.
26. SÁNCHEZ, María Rosa F., DOMÍNGUEZ, Francisco Ignacio R. and BERROCOS, Jesús V. Modelos de evaluación por competencias a través de un sistema de gestión de aprendizaje. Experiencias en la formación inicial del profesorado. 2012. No. 60.
27. GONZÁLEZ, Rodrigo P. and PÉREZ, Isaías C. *Metodologías de desarrollo de software*. 2008.
28. FIGUEROA, Roberth G., SOLÍS, Camilo J. and CABRERA, Armando A. *Metodologías Tradicionales Vs. Metodologías Ágiles*. Universidad Técnica Particular de Loja.
29. DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN. *Rational Unified Process*. Universidad Politécnica de Valencia.

30. ORALLO, Enrique H. *El Lenguaje Unificado de Modelado (UML)*.
31. SUB-JEFATURA DE INFORMÁTICA. *Herramientas CASE*. Instituto Nacional de Estadística e Informática, 1999.
32. Sitio oficial de la herramienta Visual Paradigm. [en línea]. [Accedido el 13 de Febrero de 2014]. Disponible en: <http://www.visual-paradigm.com/>
33. LENGUAJES DE PROGRAMACION | El Mundo Informático. [en línea]. [Accedido el 13 de Febrero de 2014]. Disponible en: <http://jorgesaavedra.wordpress.com/2007/05/05/lenguajes-de-programacion/>
34. Programación del lado del servidor. [en línea]. [Accedido el 13 de Febrero de 2014]. Disponible en: <http://www.paginasprodigy.com.mx/evaristopacheco/pweb/clienteServidor.html>
35. PHP: Conceptos básicos - Manual. [en línea]. [Accedido el 13 de Febrero de 2014]. Disponible en: <http://www.php.net>
36. POTENCIER, Fabien and ZANINOTTO, François. *Symfony, la guía definitiva*. 2008.
37. DOCTRINE TEAM. Doctrine. [en línea]. 2014. [Accedido el 15 de Mayo de 2014]. Disponible en: <http://www.doctrine-project.org/>
38. Bienvenido a NetBeans:Portal del IDE Java de Código Abierto. [en línea]. [Accedido el 13 de Febrero de 2014]. Disponible en: www.netbeans.org
39. Sitio oficial de Postgres. [en línea]. [Accedido el 13 de Febrero de 2014]. Disponible en: <http://www.postgresql.org.es/>
40. MATEU, Carles. *Desarrollo de aplicaciones web*. 2014. ISBN 84-9788-118-4.
41. Una Introducción a Apache. [en línea]. [Accedido el 13 de Febrero de 2014]. Disponible en: http://linux.ciberaula.com/articulo/linux_apache_intro
42. MOLINA, Sandra R. *Desarrollo de un componente para la gestión de rúbricas en la Plataforma Educativa ZERA*. La Habana. Cuba : Universidad de las Ciencias Informática, 2013.
43. JACOBSON, Ivar, BOOCH, Grady and RUMBAUGH, James. *El proceso unificado de desarrollo de software*. Addison Wesley, 2000. ISBN 84-7829-036-2.
44. ÖVERGAARD, Gunnar and PALMKVIST, Karin. *Use Cases Patterns and Blueprints*. Addison Wesley Professional, 2004. ISBN 0-13-145134-0.
45. BLAHA, Michael. *Patterns of data modeling*. Taylor and Francis Group, 2010. ISBN 978-1-4398-1989-0.
46. *Patrones de Gang of Four*. Facultad de informática - Universidad Politécnica de Madrid.

47. LARMAN, Craig. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. México: PRENTICE HALL, 1999. ISBN 970-17-0261-1.
48. IBM CORPORATION. RUP. 2007.
49. PRESSMAN, Roger S. *Ingeniería de software: Un enfoque práctico*. Quinta Edición. Mc Graw Hill.
50. EGUILUZ, Javier. *Desarrollo web ágil con Symfony 2*. 2013.
51. PÉREZ, Yuleisy G. *Análisis y Diseño de un módulo para la gestión de los Sistemas Educativos*. La Habana. Cuba : Universidad de las Ciencias Informática, [sin fecha].

Glosario de términos

Metaevaluación: Reflexión sobre el proceso de aprendizaje (18).

Autoevaluación: Los evaluadores evalúan su propio. Los roles de evaluador y evaluado coinciden en las mismas personas (12).

Heteroevaluación: Evalúan una actividad, objeto o producto, evaluadores distintos a las personas evaluadas (12).

Coevaluación: Es aquella en la que unos sujetos o grupos se evalúan mutuamente (alumnos y profesores mutuamente, unos y otros equipos docentes, el equipo directivo al Consejo Escolar y viceversa). Evaluadores y evaluados intercambian su papel alternativamente (12).

Metacognición: Manera de aprender a razonar sobre el propio razonamiento, aplicación del pensamiento al acto de pensar, aprender a aprender, es mejorar las actividades y las tareas intelectuales que uno lleva a cabo usando la reflexión para orientarlas y asegurarse una buena ejecución.

Recorrido dirigido: Constituyen rutas de aprendizaje creados por el profesor a partir del contenido oficial de la materia, ejercicios y recursos mostrados en la biblioteca. Permite estructurar el contenido seleccionado mediante el uso de plantillas previamente establecidas.

Sistema educativo: Es un concepto inserto en el contexto de las prácticas pedagógicas, de la organización política, social y económica de un país. Su estructura y organización interna está fuertemente vinculada a otros conceptos (51).

Programa de estudio: Son creados a partir del sistema curricular. Es la adecuación del macro índice al plan de estudios del sistema educativo y la institución que adquiere el software (51).