

**Universidad de las Ciencias Informáticas
Facultad 4**

**Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas Adaptación del
aprendizaje mediante recorridos dirigidos en la
Plataforma Educativa ZERA**



Autor:

Armando Avila Riverón

Tutora:

MSc. Mailin Carballosa Infante

La Habana 2018

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

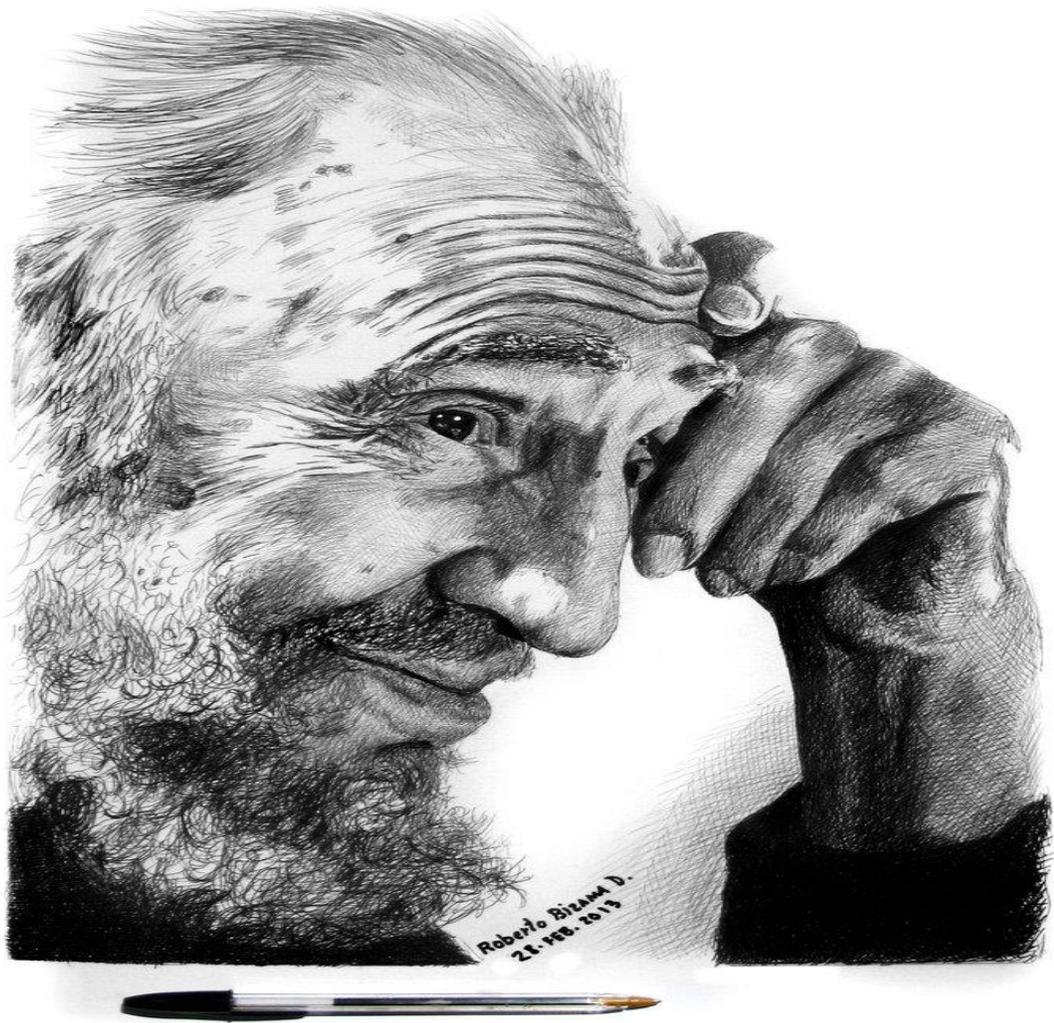
Para que así conste se firma la presente a los ____ días del mes de _____ del año _____.

Firma Autor

Armando Avila Riverón

Firma Tutor

Mailin Carballosa Infante



*"Usar la ciencia y la computación sin caer en lenguaje tecnicista
e ininteligible de élites especializadas (...)"*

Fidel Castro Ruz

Dedicatoria

A mis padres por haberse esforzado tanto por hacer de este momento realidad, a mi abuela querida por haberme malcriado tanto, a mi familia la que me tocó cuando nací y a la que escogí en la universidad.

Agradecimientos

Quiero agradecer a mi mamá y a mi abuela que aparte de ser las personas más importantes de mi vida y darme todos mis gustos lo único que han recibido a cambio ha sido dolores de cabeza, espero que este momento compense algo de todo lo que han hecho por mí, a mi papá que es el que siempre ha tenido que correr conmigo cuando pasa algo que necesita solución rápida, a toda mi familia en general porque de una manera u otra han tenido algo que ver con mi carrera. A mis grandes amigos y amigas de la universidad los cuales fueron muchos algunos no están otros sí, pero vivimos muchos momentos los cuales no merecen ser olvidados. A la universidad por darme la oportunidad de formar parte de sus filas y darme las oportunidades necesarias para lograr este momento.

Resumen

La industria del software actualmente se desarrolla y avanza de forma sorprendente. Una de las temáticas que ha tomado un gran auge es la gestión personalizada de actividades educativas, apoyándose en las ventajas que proporcionan las Tecnologías de la Información y las Comunicaciones. En el presente trabajo se realiza una propuesta de integración de funcionalidades que permiten la adaptación de contenidos de un curso en la Plataforma Educativa ZERA. Estas funcionalidades persiguen como objetivo lograr una atención diferenciada de los estudiantes que interactúan con la plataforma. Con el propósito de alcanzar el desarrollo de dicho trabajo se realiza una investigación de las principales soluciones que utilizan otros sistemas de gestión de aprendizaje para gestionar los procesos de secuenciación (1). Se llevó a cabo el proceso de desarrollo de software el cual fue guiado por los flujos de trabajo que propone la metodología de desarrollo AUP en su versión UCI y utilizando como lenguaje de programación PHP, integrando el marco de trabajo Symfony. Durante y después del desarrollo de la propuesta se realizaron diferentes pruebas aplicando el método de caja blanca con el objetivo de examinar exhaustivamente la eficiencia de dicha propuesta y una vez terminada se aplicó caja negra para comprobar su funcionamiento. Finalmente se obtuvo una herramienta que permite la personalización de un curso a uno o varios estudiantes de acuerdo al criterio del profesor.

Palabras clave: actividades, curso, secuenciación, subtema, recorrido dirigido.

Tabla de contenido

Introducción	1
Capítulo I Fundamentación Teórica.....	7
1.1 Necesidad de adaptar	7
1.2 Adaptación del contenido.....	8
1.2.1 Adaptación del secuenciamiento.....	8
1.3 Secuenciación de actividades y contenidos de aprendizaje	10
1.4 El e-learning como modalidad educativa.	11
1.5 Herramientas para el acceso al e-learning	12
1.6 Sistemas de gestión del aprendizaje	13
1.7 Plataforma educativa ZERA	14
1.7.1 Estructura del módulo docente	14
1.8 Recorridos dirigidos	15
1.9 Estudio del estado del arte de las herramientas	15
1.10 Metodología de desarrollo de software	17
1.10.1 Fundamentos de la selección.....	18
1.11 Herramientas para el desarrollo de software	19
1.11.1 Lenguajes del lado del servidor.....	19
1.11.2 Lenguajes del lado del cliente	19
1.11.1 Marcos de trabajo.....	21
1.11.4 Tecnologías del lado del cliente	22
1.11.5 Lenguaje y herramienta de modelado.....	22
1.11.6 Servidor de Base de Datos	23
1.11.7 Entorno de Desarrollo Integrado	24
1.11.8 Servidor web.....	24
1.12 Conclusiones	25

Capítulo II Propuesta de solución	26
2.1 Descripción de la solución propuesta	26
2.2 Modelo de dominio.....	27
2.2.1 Conceptos del dominio	27
2.2.2 Diagrama de modelado del dominio	27
2.3 Requerimientos del software.....	28
2.3.1 Requerimientos Funcionales	28
2.3.2 Requerimientos no funcionales	29
2.4 Actores del sistema.....	30
2.4.1 Descripción de los actores del sistema	30
2.5 Historias de usuario	30
2.6 Modelo de análisis	31
2.7 Diagrama de clases del análisis	32
2.8 Diagrama de colaboración del análisis	32
2.9 Diagrama de clases del diseño.....	33
2.10 Diagrama de secuencia del diseño.....	34
2.11 Diagrama de despliegue	34
2.12 Patrones de diseños aplicados.....	35
2.12.1 Patrones GRASP	35
2.13 Patrón arquitectónico Modelo Vista Controlador	37
2.14 Conclusiones	38
Capitulo III Implementación y prueba.....	40
3.1 Modelo de implementación.....	40
3.1.1 Diagrama de componentes	40
3.2 Modelado de datos	41
3.3 Pruebas de software.....	42
3.3.1 Niveles de Prueba	42
3.3.2 Diseño de casos de prueba	43

3.3.4 Resultados Obtenidos.....	45
3.4 Conclusiones	47
Conclusiones	48
Recomendaciones	49
Bibliografía	50
Anexos.....	55

Índice de tabla

Tabla 1 Descripción de los actores del sistema.	30
Tabla 2 Historia de usuario crear recorrido dirigido personalizado.	30
Tabla 3 Diseño de caso de prueba Incluir RDP.....	44
Tabla 4 Pruebas de Caja Negra.....	45
Tabla 5 HU Editar recorrido dirigido personalizado.	55
Tabla 6 HU Eliminar RDP.....	56
Tabla 7 HU Listar RDP.	58
Tabla 8 HU Realizar RDP.	60
Tabla 9 HU Seleccionar del índice.....	60
Tabla 10 HU Ver datos del RDP.	62
Tabla 11 Guarda los recursos.	67
Tabla 12 Guarda las materias creadas.....	67
Tabla 13 Guarda los programas de estudio creados.....	67
Tabla 14 Guarda los datos generales de los recursos guardados en tb_sco.....	68
Tabla 15 Guarda los datos generales de los recorridos dirigidos guardados.	68
Tabla 16 Guarda la relación entre estudiante, profesor grupo y programa de estudio.	68
Tabla 17 Guarda la relación entre la tabla tb_mater y los pares padre-hijo de tb_sco.	69
Tabla 18 Guarda la relación entre la tabla tb_study_program y la tabla tb_professor.....	69

Índice de figura

Figura 1 Diagrama de modelado del dominio	28
Figura 2 Diagrama de clases del análisis Crear RD.	32
Figura 3 Diagrama de colaboración del análisis crear RD.....	33
Figura 4 Diagrama de Clases del dominio Crear RD.....	34
Figura 5 Diagrama de secuencia del diseño crear RD.....	34
Figura 6 Diagrama de despliegue.	34
Figura 7 Patrón Experto.....	35
Figura 8 Patrón Controlador.	36
Figura 9 Patrón Creador.....	36
Figura 10 Patrón modelo vista controlador.....	38
Figura 11 Diagrama de componentes crear RD.....	41
Figura 12 Modelado de datos.	41
Figura 13 Total de no conformidades identificadas.	46
Figura 14 Diagramas de clases del diseño Editar RD.	64
Figura 15 Diagramas de clases del diseño Eliminar RD.....	64
Figura 16 Diagramas de clases del diseño Listar RD.....	65
Figura 17 Diagramas de Secuencia Editar RD.	65
Figura 18 Diagramas de Secuencia Eliminar RD.	66
Figura 19 Diagramas de Secuencia Listar RD.....	66

Introducción

La introducción y el desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) en la educación actual, plantean el desafío de aprovechar al máximo su potencial para impulsar nuevas y mejores formas de enseñar y de aprender. La conjunción del desarrollo del Proceso de Enseñanza y Aprendizaje (PEA) y la incorporación y extensión del uso de las TIC, debe facilitar la reformulación del papel y práctica pedagógica del docente, orientada hacia el pretendido desarrollo de las destrezas y potencialidades cognitivas del alumno. Por ello es imprescindible

establecer la relación de las TIC con el desarrollo en los alumnos de su capacidad para aprender

a aprender, para buscar información de forma selectiva. En este sentido, las TIC facilitan el desarrollo de una acción formativa flexible, centrada en el estudiante y adaptada a sus características y necesidades, con un seguimiento individualizado y continuo de los alumnos (1).

La utilización de estas tecnologías, proporciona que los docentes puedan adquirir mayor y mejor conocimiento dentro de su área, permitiendo la innovación, así como el intercambio de ideas y experiencias con otros usuarios, mejorando también la comunicación con los estudiantes. Además, estas fomentan la colaboración entre los alumnos, ayudan a centrarse en el aprendizaje, aumentan la motivación y el interés, favorecen la búsqueda, estimulan el desarrollo del razonamiento, la resolución de problemas, la creatividad y la capacidad de aprender a aprender (2).

Impulsado por estos beneficios, han ido surgiendo nuevas modalidades educativas más personalizadas y centradas en la actividad de los estudiantes. Un ejemplo de ello, lo evidencian los entornos *e-learning*, los cuales brindan gran flexibilidad y dinamismo a la hora de crear, distribuir, recibir y evaluar los contenidos, además de integrar herramientas que proporcionan apoyo al proceso de aprendizaje, facilitando el seguimiento y la evaluación de los estudiantes. En muy pocos años este término ha pasado del vocabulario utilizado por una minoría de expertos en las aplicaciones de la tecnología en la enseñanza a ser empleado por múltiples instituciones.

Algunos autores como Manuel Area Moreira y Jordi Adell Segura definen el concepto de *e-learning* como “una modalidad de enseñanza-aprendizaje que consiste en el diseño, puesta

en práctica y evaluación de un curso o plan formativo desarrollado a través de redes de ordenadores y puede definirse como una educación o formación ofrecida a individuos que están geográficamente dispersos o separados o que interactúan en tiempos diferidos del docente empleando los recursos informáticos y de telecomunicaciones” (3).

En esta nueva era del PEA han jugado un papel importante los *Learning Management System* (LMS) o Sistemas de Administración del Aprendizaje las cuales se encargan de administrar la educación a distancia, tales como Moodle, Edmodo, Blackboard, Chamilo, Claroline, entre otras. Aunque las plataformas mencionadas anteriormente presentan como principal ventaja la administración de la educación a distancia, no realizan un correcto uso a los contenidos de aprendizaje a través de la secuenciación de los mismos, dado que muestran el contenido de manera regular para todos los estudiantes sin dar atención diferenciada a ninguno de ellos de acuerdo a las necesidades de su aprendizaje.

La secuencia permite organizar las actividades en forma progresiva, ganando en complejidad a medida que los alumnos avanzan. Este secuenciamiento es lo que se ha denominado flujo de aprendizaje. De esta forma se pueden considerar, por ejemplo, distintos recorridos dirigidos, tareas que puedan ser realizadas en paralelo o trabajos que deben completarse en subgrupos antes de continuar con el desarrollo del curso.

La finalidad de la secuenciación es establecer un ordenamiento de los contenidos de enseñanza que asegure el enlace entre los objetivos educativos y las actividades de aprendizaje de los alumnos, de tal manera que la organización de la actividad desarrollada proporcione garantías suficientes para la consecución de las intenciones formativas propias del programa de formación, de la comunidad educativa o de la institución. Asimismo, la secuenciación de contenidos, de tareas y de actividades ha de ir propiciando un acercamiento progresivo desde la situación inicial de aprendizaje de los alumnos hasta los objetivos propuestos para el programa formativo en el que se encuentre inmerso. Este elemento es diferente para cada colectivo de destinatarios de la formación e incluso para cada individuo, para cada situación de aprendizaje, para cada programa formativo y para cada institución (4).

Entre los LMS que organizan las actividades de aprendizaje se encuentran: Chamilo, que mediante diseño instruccional y colaborativo permite al docente escoger entre una serie de metodologías pedagógicas, siendo una de ellas el constructivismo social; Claroline, sigue las especificaciones de *Sharable Content Object Reference Model* (SCORM) e *Instruction Management Systems* (IMS); .LRN, cuenta con soporte a diversos estándares como: IMS Content Packaging (IMS-CP), IMS Meta Data (IMS-MD), IMS Question & Test

Interoperability Specification (IMS-QTI), IMS-Learning Design (IMS-LD), IMS Enterprise, SCORM, obteniendo de esta forma un fácil camino hacia la interoperabilidad de sistemas; Moodle, permite combinar las actividades en secuencias y grupos, ayuda al docente a guiar a los participantes y es compatible con estándares y especificaciones tales como: SCORM e IMS; Blackboard, cuenta con herramientas de seguimiento y gestión de aprendizaje, con soporte para especificaciones de contenido tales como: IMS, AICC y SCORM.

La adaptación de contenidos permite que los diferentes contenidos que se requieran en un proceso de aprendizaje sean adaptados para obtener su máximo rendimiento desde el punto de vista pedagógico (5).

Los contenidos pueden definirse como lo que los estudiantes deberían saber o comprender como resultado del proceso de aprendizaje. Lo que tiene que ser enseñado y aprendido en función de los programas determinados en el plan de estudio oficial y en las adaptaciones con necesidades especiales. Una decisión clave en el proceso de enseñanza es determinar lo que se tiene que aprender. Para ello, deberá plantearse qué contenidos son fundamentales y tienen que aprender todos, la base que deben adquirir, y a la que luego se van añadiendo contenidos en extensión o profundidad, en función de las posibilidades de los estudiantes (6).

En la Universidad de las Ciencias Informáticas (UCI) se desarrolló un LMS denominado plataforma educativa ZERA, en la cual se ha logrado fomentar un entorno que sirve de apoyo en el proceso de enseñanza-aprendizaje. Entre las principales características se pueden destacar:

- ✓ Basada en hiper-entornos de aprendizajes.
- ✓ Creación de cursos con una estructura capitular donde el contenido se muestra como un libro: capítulo, temas y subtemas, avance del contenido (marcador de libro), resaltado y apuntes al contenido.
- ✓ Soporte para las especificaciones IMS-QTI y el estándar SCORM.
- ✓ Sugerencias de uso y registro de avance.
- ✓ Ideal para el trabajo semi-presencial *b-learning* y a distancia *e-learning*.
- ✓ Atención diferenciada: recorridos dirigidos, softareas, orientaciones de trabajos.

La plataforma educativa ZERA cuenta con un subsistema basado en los hiper-entornos de aprendizaje, en el que se integran módulos como: Docente, Contenido, Biblioteca, Prácticas y Tareas. Las funcionalidades implementadas en el módulo Docente presentan limitaciones,

no cumpliendo así con las necesidades de los profesores. Estos últimos no pueden gestionar en tiempo real la creación de las secuencias didácticas o recorridos dirigidos (RD), en las que pueda seleccionar la información del contenido que realmente le interesa, o simplemente personalizar el aprendizaje a sus estudiantes para atender diferencias individuales.

Los RD constituyen rutas de aprendizaje creadas por el docente a partir del contenido oficial de la materia, ejercicios y recursos mostrados en la Biblioteca. Estos RD brindan la posibilidad de estructurar el contenido seleccionado mediante plantillas previamente establecidas.

La plataforma no proporciona un sistema que le permita al profesor crear actividades atendiendo a la diversidad de conocimientos y habilidades de sus estudiantes dentro de un curso. Esto trae como consecuencia que los estudiantes participen en las mismas actividades y vean el mismo contenido independientemente de las diferencias que existan entre ellos. Partiendo del criterio de que, *“la adaptación del proceso educativo a cada individuo tiene un efecto muy significativo sobre su aprendizaje. Estos efectos pueden ser evidenciados en la diferencia de los resultados de los alumnos que tienen acceso a una tutoría individualizada”* (7), pues se hace necesario que los contenidos puedan ser adaptados mediante RD.

Otra de las dificultades que evidencian la carencia de este sistema es la incorrecta incursión de los estudiantes por las diferentes actividades, asociado a la independencia y libertad para navegar por los contenidos, lo que en principio y para determinado tipo de estudiante podría ser favorable, sin embargo, para otros podría ser un distractor. Al mismo tiempo, la forma de presentar los contenidos a los estudiantes en ocasiones se ve limitada, pues a pesar de trabajar en la estructuración de los mismos y presentarlos atendiendo a los currículos, en muchas ocasiones, se hace necesario mostrarlos de acuerdo a las características de cada estudiante, sus resultados y desempeño.

Por todo lo antes planteado se establece como **problema científico**: ¿Cómo adaptar los contenidos y las actividades de aprendizaje a las diferencias individuales de los estudiantes mediante recorridos dirigidos en la plataforma educativa ZERA? De lo planteado anteriormente se deriva como **objeto de estudio**: el proceso de adaptación de los contenidos y las actividades de aprendizaje a las diferencias individuales de los estudiantes en los LMS. Determinando para la investigación como, **campo de acción**: el proceso de

adaptación de los contenidos y las actividades de aprendizaje a las diferencias individuales de los estudiantes en la plataforma educativa ZERA.

Objetivo general: desarrollar una adaptación de los contenidos y las actividades de aprendizaje a las diferencias individuales de los estudiantes mediante recorridos dirigidos en la plataforma educativa ZERA.

Para darle respuesta al objetivo general propuesto se definen los siguientes **objetivos específicos:**

- Construir los referentes teóricos relacionando los aspectos teóricos fundamentales que sustentan la investigación, basándose en los resultados al analizar procesos de gestión de contenidos en sistemas especializados en esta actividad y las principales tendencias en cuanto a la gestión de contenidos para el aprendizaje.
- Desarrollar una adaptación del aprendizaje a las diferencias individuales de los estudiantes mediante recorridos dirigidos en la plataforma educativa ZERA.
- Realizar pruebas de satisfacción a la solución obtenida para validar la correspondencia entre los resultados obtenido y los objetivos planteados.

Los métodos científicos utilizados en la investigación estuvieron determinados por el objetivo general y los objetivos específicos. A nivel teórico fueron utilizados los métodos: analítico – sintético: Análisis: permite la división mental del fenómeno en sus múltiples relaciones y componentes para facilitar su estudio. Síntesis: establece mentalmente la unión entre las partes previamente analizadas, posibilita descubrir sus características generales y las relaciones esenciales entre ellas (8). Es utilizado para realizar un estudio de la bibliografía referente a los diferentes conceptos asociados al objeto de estudio permitiendo la extracción de los elementos más importantes que se relacionan con este.

Métodos empíricos:

La Observación Científica, permite valorar los avances realizados en el estudio de las diferentes herramientas y tecnologías a usar.

El método de Consulta de la Información es utilizado para elaborar el marco teórico o el estado del arte de la investigación, permitiendo el conocimiento y el acceso a las referencias bibliográficas de los múltiples criterios que han sido citados para comprender mejor el problema de investigación planteado.

El presente trabajo posee la siguiente estructura:

Capítulo I: Fundamentación teórica. En este capítulo se investigan y analizan las tendencias actuales del proceso de gestión de contenidos para plataformas *e-learning*. Además, se analizan las distintas herramientas, tecnologías y metodología que dará solución al problema planteado.

Capítulo II: Se especifica la propuesta de solución para el problema planteado. Se presenta el Modelo de dominio y se exponen los requisitos funcionales y no funcionales que debe cumplir el sistema. Además, se obtienen como artefactos fundamentales las historias de usuarios.

Capítulo III: Se realiza el análisis y diseño de la propuesta de solución planteada. Se generan los diagramas de clases del análisis y el diseño, el modelo de bases de datos y el modelo de despliegue. Además, se realiza un estudio de los patrones de arquitectura y diseño empleados.

Se obtiene la descripción del proceso de implementación de la herramienta a través de los

diagramas de componentes. Por último, se realiza la descripción de los casos de prueba y se describen los resultados obtenidos una vez aplicadas las pruebas.

Capítulo 1 Fundamentación Teórica

En este capítulo se especifica la fundamentación teórica que soporta la investigación, basándose en los resultados al analizar procesos de gestión de contenidos en sistemas especializados en esta actividad y las principales tendencias en cuanto a la gestión de contenidos para el aprendizaje. Además, se hace especial referencia a la utilización de plataformas de *e-learning*, explicando en qué consisten y para qué se utilizan. Para dar cumplimiento a la correcta definición de estos puntos se realiza el estudio de soluciones similares y de las principales tendencias que se llevan a cabo en la producción de aplicaciones educativas relacionadas con la secuenciación de contenidos, con el fin de determinar la necesidad, el nivel de aceptación y la correcta forma de aplicación de estos elementos en la Plataforma Educativa ZERA.

1.1 Necesidad de adaptar

Una de las ventajas de los sistemas educativos basados en web es la posibilidad de ofrecer una gran cantidad de recursos al usuario. A cambio, se corre el riesgo de quedar “perdido en el hiperespacio”. Para evitar que esto ocurra, es conveniente ajustar el material educativo mostrado a las necesidades de cada estudiante. De esta forma, no se le muestran los mismos contenidos a todo el mundo, sino que el material es ajustado (9).

La adaptación de contenidos tiene gran importancia porque ciertamente todo grupo de alumnos se caracteriza por ser heterogéneo. Los alumnos difieren los unos de los otros en términos de sus capacidades personales, su ritmo y estilo de aprendizaje, sus intereses y motivaciones, sus rasgos de personalidad y su historia socio familiar. De ahí que, el mayor desafío que enfrentan los profesores en su práctica cotidiana tiene que ver con descubrir los modos de enseñanza y estrategias que aseguren el éxito de aprendizaje de todos sus alumnos. Cuando a esta realidad se suma un alumno con necesidades educativas especiales, es decir que presenta dificultades mayores que el resto de sus compañeros para comprender el contenido, se encuentra que el rango de heterogeneidad del grupo puede verse aumentado, lo que conlleva a que el profesor se vea ante la necesidad de realizar modificaciones o adaptaciones más significativas que permitan a los estudiantes acceder al conocimiento (10) .

1.2 Adaptación del contenido

La organización y puesta en práctica del proceso de enseñanza aprendizaje implica un complejo proceso de toma de decisiones, en el cual el papel del profesor es determinante. Tomar decisiones adecuadas a la realidad de los alumnos, depende en buena medida, de la habilidad que tenga el profesor para reconocer las características y necesidades de sus alumnos, así como para ajustar la respuesta educativa en función de sus necesidades de aprendizaje.

La mayor parte de los esfuerzos de los últimos años se han centrado en adaptar el contenido al estudiante, a sus capacidades y sus metas. Se considera aquí que la adaptación de contenido educativo incluye procesos como: la creación de anotaciones y la ocultación de información (7).

Las modificaciones o adaptaciones en el contenido es una de las estrategias que permiten adecuar el currículo para permitir el aprendizaje de todos los estudiantes, seleccionando la extensión de la información o el grado de dificultad, que vendrá determinado por el grado de abstracción o la complejidad del concepto o proceso a aprender. Adaptar los contenidos hace referencia a su selección, priorización, secuenciación, organización y planificación. Pero, con ello, no se trata de empobrecer el aprendizaje (6).

1.2.1 Adaptación del secuenciamiento

Se denomina secuenciamiento a un determinado conjunto de elementos que se ordenan en una determinada sucesión, esto es, uno detrás de otros o uno delante de otros (11).

Una de las maneras de adaptar el proceso de enseñanza aprendizaje es ajustándolo a los rasgos individuales o preferencias de cada estudiante, de tal forma se podrá enseñar a los escolares utilizando métodos pedagógicos que se relacionen con sus fortalezas y preferencias.

El diseño y desarrollo de los contenidos que se vayan a incluir en un curso de formación en línea, requieren una adecuada estructuración y una minuciosa planificación que facilite su seguimiento por parte de los participantes en dicho curso, ya que este aspecto contribuirá a facilitar, no solo el proceso de aprendizaje, sino también las posibilidades de control y seguimiento por parte del formador (12).

Las actividades a plantear en un modelo de formación, pueden ser de dos tipos, dependiendo de que el proceso formativo cuente o no con el soporte y seguimiento de un tutor. Por tanto, se podrán plantear, dos tipos de actividades:

- 1- Actividades autoformativas: es decir, aquellas que no requieren el seguimiento de un tutor, ya que el alumno puede realizar la misma de forma individual. El sistema se encarga de realizar la revisión de dicha actividad y de presentar los resultados al participante.
- 2- Actividades abiertas de tipo individual o colaborativo: este tipo de actividades no cuentan con una secuenciación tal y están pensadas para ser utilizadas en procesos de formación tutorizados, en los cuales, el proceso de corrección y evaluación, se prevé llevar a cabo por parte de un tutor (13).

Las actividades autoformativas cuentan con tres tipos de actividades basadas en la secuenciación, las cuales son:

- Secuencias dirigidas (aprendizaje autoguiado): en este no hay ninguna interacción con el usuario, sino que los contenidos se presentan al alumno de una manera predefinida, ya sea aleatoria o por una forma definida por el tutor o creador de la actividad.
- Secuencias guiadas por el propio alumno (aprendizaje autodidacta): este tipo permite al alumno decidir los contenidos que desea visualizar. Puede ser una secuencia guiada totalmente o parcial, donde se le imponen ciertas restricciones a modo de prerrequisitos.
- Secuencias adaptativas: en este tipo, el sistema es capaz de decidir la manera de secuenciar los contenidos basándose en las características y preferencias del alumno (13).

Si se adopta la modalidad de adaptación a las preferencias individuales, como los estilos de aprendizaje, primero hay que seleccionar las estrategias instruccionales apropiadas y consistentes con el objetivo de aprendizaje y, sobre la base de estas estrategias, optar por las más adecuadas a cada uno de los estilos de aprendizaje. Esto permitirá definir las distintas alternativas de presentación de un mismo contenido para cada uno de los estilos de aprendizajes del modelo utilizado para basar la adaptación (14).

Para cada una de estas opciones de presentación de contenidos es necesario establecer los formatos de información que serán utilizados (texto, audio, imágenes,

animaciones, vídeo), las distintas secuencias de los mismos, las herramientas disponibles para las actividades a desarrollar y finalmente, a partir de estos parámetros, las técnicas de adaptación a utilizar para conseguir que un usuario específico, con un estilo de aprendizaje preponderante, acceda a los contenidos adecuados a sus preferencias (14).

1.3 Secuenciación de actividades y contenidos de aprendizaje

Se piensa en los contenidos a enseñar como todo lo que se debe enseñar, pero no siempre ese todo puede ser enseñado. En ocasiones el tiempo u otras variables no son suficientes para enseñar todo, entonces es necesario seleccionar y es la selección de contenidos un punto central en el proceso enseñanza aprendizaje. La organización, selección y secuenciación de los contenidos de aprendizaje ha sido un tema de interés central para el diseño curricular de los programas instruccionales a los distintos niveles, desde la unidad didáctica hasta los estudios de ciclo formativo o carrera; o desde el punto de vista de las instituciones y centros educativos. Así mismo, más allá del interés que haya podido tener en los últimos tiempos desde el punto de vista del diseño instruccional tecnológico: la organización y la secuenciación de los contenidos de enseñanza constituyen el punto neurálgico del itinerario que conducirá a diseñar los procesos de aprendizaje (4).

La secuenciación de alguna manera, es una actividad de la que se tiene poca experiencia previa, sin embargo, en mayor o menor medida, más o menos conscientemente, forma parte de la tarea de todos los docentes. La finalidad de la secuenciación es establecer una ordenación de los contenidos de enseñanza que asegure el enlace entre los objetivos educativos y las actividades de aprendizaje de los alumnos, de tal manera que la organización del trabajo formativo dé garantías suficientes para la consecución de las intenciones formativas propias del programa de formación, la comunidad educativa o de la institución (15).

Establecer una secuencia de contenidos implica analizar los contenidos que pretenden secuenciarse, implantar una selección de los aspectos más relevantes, definir sus relaciones en un momento dado y a lo largo del tiempo.

La toma de decisiones relacionadas con las secuencias de contenidos no son solamente un proceso técnico, sino que se constituyen en el encuentro de algunos de los aspectos básicos de la educación: el sentido de los contenidos enseñados y las relaciones entre los conocimientos escolares y los cotidianos, las relaciones entre los contenidos a enseñar.

Para poder realizar un proceso de secuenciación es necesario determinar los elementos que deben ser objeto del mismo y los principios que deben orientarlo.

Para secuenciar, entre otras cosas, es necesario:

- Reconocer el volumen de conocimiento sobre el que se va a trabajar para secuenciar. A partir de él seleccionarlos y jerarquizarlos a partir del diagnóstico institucional.
- Analizar los contenidos a enseñar: comprenderlos, visualizar sus ideas fundamentales, desentrañarlos, relacionarlos entre ellos y con los saberes previos exigidos para su apropiación.

Se conoce de la importancia de la secuenciación para los procesos de aprendizaje, el orden del tratamiento de los contenidos y la forma en que se presentan, por ello es preciso dedicar mucha atención a las secuencias educativas, a su justificación, a su análisis y a su revisión (16).

1.4 El *e-learning* como modalidad educativa.

El *e-learning* proporciona la oportunidad de crear ambientes de aprendizaje centrados en el estudiante. Estos escenarios se caracterizan además por ser interactivos, eficientes, fácilmente accesibles y distribuidos (17).

De ahí la importancia de optimizar los sistemas de *e-learning*, de racionalizar y justificar este sistema de aprendizaje para crear entornos de interacción entre los alumnos y el material académico.

Un sistema *e-learning* debe ser considerado como una herramienta que permita desarrollar un amplio campo de soluciones para enriquecer el conocimiento y el perfeccionamiento de las personas promoviendo su participación para crear y compartir el conocimiento. Es así como un sistema *e-learning* propicia un nuevo entorno que promueva el aprendizaje, gracias al aprovechamiento de los recursos de la red y la facilidad de interacción entre usuarios y tutores (18).

Una característica del *e-learning* es la creación, adopción y distribución de contenidos, así como la adaptación del ritmo de aprendizaje y la disponibilidad de las herramientas de aprendizaje independientemente de los límites de horarios o ubicación geográfica. Estas características permiten al alumno intercambiar opiniones y realizar diversas actividades, siempre mediante el uso de las TIC. El intercambio mediante las TIC, se lleva a cabo en los Entornos Virtuales de Aprendizaje (EVA), desarrollados con diversas herramientas o

sistemas, conocidos como plataformas de *e-learning*. Teniendo en cuenta esto se puede decir que una solución *e-learning* está formada por dos elementos fundamentales: plataforma y contenido.

De acuerdo con un estudio realizado por Facundo Ortega, catedrático de la Universidad de Córdoba, Argentina, una plataforma educativa no es más que: "(...) una herramienta ya sea física o virtual que brinda la capacidad de interactuar con uno o varios usuarios con fines pedagógicos. Además, se considera un proceso que contribuye a la evolución de los procesos de aprendizaje y enseñanza, que complementa o presenta alternativas en los procesos de la educación tradicional. En la actualidad, la mayor parte de las plataformas educativas son programas computacionales (software) o equipos electrónicos (hardware)" (19).

Según Peñalvo, en su artículo, titulado "Estado actual de los sistemas *e-learning*", define esta modalidad de enseñanza como: "*La capacitación no presencial que, a través de plataformas tecnológicas, posibilita y flexibiliza el acceso y el tiempo en el proceso de enseñanza-aprendizaje, adecuándolos a las habilidades, necesidades y disponibilidades de cada discente, además de garantizar ambientes de aprendizaje colaborativos mediante el uso de herramientas de comunicación síncrona y asíncrona, potenciando en suma el proceso de gestión basado en competencias*" (20).

El *e-learning* asincrónico se refiere a una participación diferida en el tiempo por parte de los estudiantes. En este caso el contenido y las actividades están disponibles a través de un servidor web. Es así como los estudiantes pueden acceder a la plataforma *e-learning*, acceder a los recursos y colaborar con sus pares y profesores en cualquier lugar (18).

1.5 Herramientas para el acceso al *e-learning*

El *e-learning* se basa en el empleo de las TIC para crear, distribuir y facilitar el conocimiento en cualquier momento y en cualquier lugar. Se utiliza internet como medio de distribución de la información y el ordenador, el teléfono móvil o las tabletas electrónicas como dispositivos de acceso a esa información a través de un servidor de páginas web en la que se desarrollan las acciones formativas al que acceden los alumnos, profesores, coordinadores y un entorno de administración, desde dónde se configuran los cursos, se dan de alta los alumnos, se importan contenidos, se habilitan servicios, entre otras acciones (21).

Según (Pedruelo, 2004) el núcleo central de los sistemas de *e-learning* es la plataforma que soporta al EVA. Básicamente, se trata de una aplicación o un conjunto de aplicaciones basadas en tecnología web cuyas funciones principales son:

- ✓ Gestión de los usuarios: inscripción, control de acceso, control de los aprendizajes e itinerarios formativos, generación de informes, etc.
- ✓ Gestión de los cursos: incluye su distribución, el registro de la actividad de los usuarios (alumnos principalmente), interacciones con el material educativo, evaluaciones, tiempos de acceso, etc.
- ✓ Gestión de los servicios de comunicación como apoyo del material didáctico y soporte para los aprendizajes. Algunas de las principales herramientas son el correo electrónico, los foros de discusión y el chat.

1.6 Sistemas de gestión del aprendizaje

Un LMS, o sistema de gestión de aprendizaje, es una herramienta informática y telemática organizada en función de unos objetivos formativos de forma integral, es decir, que se puedan conseguir exclusivamente dentro de ella y de unos principios de intervención psicopedagógica y organizativos, de manera que se cumplen las siguientes características básicas (22).

- ✓ Herramientas de Comunicación: la comunicación es un aspecto fundamental para el aprendizaje y para las relaciones sociales. De ahí que la plataforma deba contar con variados sistemas de comunicación tanto síncronos, como asíncronos (foro, correo). Pero no se trata de colocarlos en la web, cada uno de estos recursos debe ubicarse en un área con un fin didáctico predeterminado. Por ejemplo, puede haber un chat al que sólo acceden los miembros de este grupo y cuyo fin es el trabajo colaborativo, pero también puede haber un chat como medio para realizar tutorías a trabajos específicos y por tanto involucra sólo a los realizadores de ese trabajo.
- ✓ Servicios y áreas configurables: cada curso, cada tipo de alumno, cada materia, puede requerir diferentes elementos o recursos. Es importante que los servicios y áreas que compongan la plataforma sean configurables dependiendo de cada curso concreto. Por ejemplo, en un curso a los profesores les puede interesar que los alumnos dispongan de un área privada para trabajar en grupo y, sin embargo, en otros, que trabajen de forma individual.

Sin embargo, la implementación de una plataforma LMS, no garantiza los medios para la creación y generación de los cursos necesarios para la organización. Desde la perspectiva

de los materiales docentes simplemente actúa como plataforma de distribución donde se remarca la idea de que en un sistema LMS la mínima unidad de instrucción es el curso en sí mismo. Los LMS permiten una eficiente gestión de los usuarios, pero demuestran carencias en la gestión de los contenidos.

1.7 Plataforma educativa ZERA

En la UCI se desarrolló un LMS denominado Plataforma Educativa ZERA, en la cual se ha logrado fomentar un entorno que sirve en gran medida de apoyo en el proceso de enseñanza-aprendizaje. Entre sus principales características se pueden destacar:

- ✓ Basada en hiper-entornos de aprendizajes.
- ✓ Creación de cursos con una estructura capitular donde el contenido se muestra como un libro: capítulo, temas y subtemas, avance del contenido (marcador de libro), resaltado, apuntes al contenido.
- ✓ Creación de 30 tipos de recursos educativos.
- ✓ Creación de 11 tipologías de ejercicios.
- ✓ Soporte para las especificaciones IMS-QTI y SCORM.
- ✓ Sugerencias de uso y registro de avance.
- ✓ Evaluación por rúbricas y por competencias.
- ✓ Ideal para el trabajo semi-presencial (*b-learning*) y a distancia (*e-learning*).
- ✓ Atención diferenciada: recorridos dirigidos, softareas, orientaciones de trabajos.
- ✓ Sistema distribuido, ideal para instituciones con problemas de conectividad.
- ✓ Gestión académica.
- ✓ Fórum, chat, herramientas de la web 2.0, sistema con varios roles, etc.

Este conjunto de características convierte a ZERA en una plataforma capaz de competir con sus similares tanto a nivel nacional como internacional. ZERA a su vez, está compuesta por diferentes módulos dentro de los cuales se puede destacar el módulo docente por ser un espacio que permite al profesor desarrollar actividades que le son muy útiles para su trabajo tanto con la plataforma, como con sus estudiantes y demás profesores.

1.7.1 Estructura del módulo docente

Dentro de las funciones llevadas a cabo por un profesor en este módulo está el seguimiento a sus estudiantes a través de la revisión de trabajos orientados por él, siendo esta tarea vital tanto para estudiantes como para los propios profesores. Cuenta con un lugar donde el profesor puede registrar sus trabajos, materiales, informes, recomendaciones sobre

algún contenido, esto no solo para su consulta sino también para la de otros profesores y estudiantes. Además, puede registrar su avance programático en la asignatura, así como mostrar a sus estudiantes artículos que le hayan parecido interesantes, relacionados con la asignatura que este imparte.

En dicho módulo se le permite al profesor realizar un conjunto de funcionalidades, pero a su vez, este no permite que el profesor pueda crear actividades docentes que se adapten a la diversidad de conocimientos y habilidades de sus estudiantes; elemento que trae como consecuencia que los estudiantes participen en las mismas actividades y vean el mismo contenido independientemente de las diferencias que existan entre ellos. También, la incorrecta incursión de los estudiantes por las diferentes actividades, asociado a la independencia y libertad para navegar por los contenidos, lo que en principio y para determinado tipo de estudiante podría ser favorable, sin embargo, para otros podría ser un distractor. Por otra parte, la forma de presentar los contenidos a los estudiantes en ocasiones se ve limitada, pues a pesar de trabajar en la estructuración de los mismos y presentarlos atendiendo a los currículos, en muchas ocasiones, se hace necesario mostrarlos de acuerdo a las características de cada estudiante, sus resultados y desempeño.

1.8 Recorridos dirigidos

Las actividades de aprendizaje se pueden secuenciar y organizar para lograr un aprendizaje más efectivo. Este secuenciamiento es lo que se ha denominado flujo de aprendizaje, rutas de aprendizaje o RD (23). 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.

De esta forma, se pueden considerar, por ejemplo, distintos “recorridos” dirigidos o de aprendizaje, como tareas que puedan ser realizadas en paralelo o trabajos que deben completarse en subgrupos antes de continuar con el desarrollo del curso (23).

1.9 Estudio del estado del arte de las herramientas

En este acápite se realiza un estudio de los sistemas similares y se muestra como evidencia algunas características de los mismos en aras de analizar las ventajas y desventajas para así alcanzar una solución capaz de corregir en gran medida los principales problemas

identificados. A su vez, se pretende resolver los problemas actuales de la plataforma educativa ZERA con respecto a la adaptación del contenido y actividades de aprendizaje.

Dokeos

Dokeos es un sistema de aprendizaje virtual basado en la web, técnicamente conocido como un LMS o Sistema de Gestión de Contenido (CMS, del inglés Content Management System). Es intuitivo y fácil de usar para todo tipo de usuario (profesores, formadores, estudiantes), ofrece una amplia gama de herramientas y facilita la creación y organización de contenidos interactivos y ejercicios (24).

Presenta un itinerario de aprendizaje dividido en dos pasos. El progreso de los estudiantes puede controlarse por los prerrequisitos, es decir tiene que completar ciertos pasos antes de continuar. El itinerario ofrece el potencial de crear un viaje a través de una base de conocimientos que incluye recursos, test, tareas, discusiones y evaluación.

Moodle

Moodle define una tarea como cualquier trabajo, labor o actividad que asigna el docente al estudiante. Los alumnos pueden subir sus tareas (en cualquier formato de archivo) al servidor, se registra la fecha en que se han subido. Para la creación de una tarea se debe elegir la opción que permite crear una actividad, la tarea tiene asociado título, descripción detallada de la actividad que se va a asignar, además incluye puntos que deben cubrir y objetivos (25).

La desventaja que presenta Moodle es que muestra los mismos contenidos a todos los alumnos. Es decir, no tiene forma de guiar el proceso de aprendizaje de cada alumno, asignando a cada cual las tareas enfocadas en resolver los problemas individuales.

Claroline

Claroline presenta la herramienta Usuarios, y dentro de ella dispone de una opción que permite dar seguimiento a las estadísticas sobre las actividades que ha realizado un usuario en particular en el curso. Dentro de los datos que brindan se encuentran los resultados de los ejercicios realizados, las rutas de aprendizajes, los trabajos publicados y documentos (26).

Chamilo

Chamilo es un nuevo proyecto que opta por el código abierto y gratuito, fácilmente adaptable para contextos diferentes. Les permite a los docentes construir cursos en línea

como soporte a la modalidad presencial o netamente virtuales, pero una vez creados estos cursos no permiten adaptarse a las necesidades de cada estudiante (27).

Edmodo

Edmodo es una aplicación cuyo objetivo principal es permitir la comunicación entre profesores y alumnos. Crea grupos privados con acceso limitado a docentes, alumnos y padres, dispone de un espacio de comunicación entre los diferentes roles mediante mensajes y alertas, gestiona las calificaciones de los alumnos, asigna tareas a los alumnos y gestiona las calificaciones de las mismas, da acceso a los padres a los grupos en los que estén asignados sus hijos, permitiendo estar informados de la actividad de sus hijos y tener la posibilidad de comunicación con los profesores (28).

Luego de haber realizado un análisis de las plataformas mencionadas anteriormente se puede concluir, que las mismas se basan más en la interacción del estudiante-profesor que en el uso de recorridos dirigidos para la correcta atención diferenciada dadas las necesidades de cada estudiante. Además, que ninguno de los LMS antes nombrado hace uso de los RD ni de la secuenciación en función de algún tipo de estudiante en especial como tampoco de un alumno en particular.

1.10 Metodología de desarrollo de software

Una metodología de desarrollo de software es un enfoque estructurado para el desarrollo de software que incluye modelos de sistemas, notaciones, reglas, sugerencias de diseño y guías de procesos (29). Para el desarrollo del sistema propuesto se realiza un estudio de las metodologías ágiles, puesto que las mismas proponen una estrecha relación entre el cliente y el equipo de desarrollo, no existe un contrato tradicional, se aplican a proyectos con requerimientos cambiantes o imprecisos, y están dirigidas fundamentalmente para equipos pequeños; se debe hacer entregas funcionales continuamente y el cliente es parte del equipo de desarrollo (30).

Metodologías Ágiles:

- XP (eXtremeProgramming)
- Scrum
- Extreme ModelingAUP (Agile Unified Process)

En esta sección se describen las metodologías SCRUM, AUP y XP:

- ✓ **Scrum** es un proceso en el que se aplican de manera regular un conjunto de mejores prácticas para trabajar en equipo y obtener el mejor resultado posible de un proyecto. En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales (31).
- ✓ **XP** es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes y simplicidad en las soluciones implementadas. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (30).
- ✓ **AUP** es una versión simplificada de Proceso Unificado Racional(RUP) que describe de forma simple y fácil de comprender el uso de técnicas ágiles para el desarrollo de aplicaciones que permanezcan dentro de los conceptos de RUP. AUP promueve principios ágiles, de los cuales se acentúan particularmente los siguientes: integración continua, refactorización, propiedad colectiva del código, bienestar del programador. AUP establece que debe determinarse una persona que se encargue de velar por el cumplimiento de los requisitos y las prioridades del sistema, esta persona puede ser o bien el cliente en cuestión o un equipo de analistas. AUP es flexible y propone los mismos roles y artefactos que RUP, solo que no hay necesidad de generar toda la documentación que se requiere en cada flujo de trabajo (32).

1.10.1 Fundamentos de la selección

De acuerdo con las características que presentan los servicios a desarrollar y tras el estudio realizado de las diferentes metodologías de desarrollo, se determina como metodología a utilizar AUP, específicamente la variante AUP-UCI, de forma tal que el desarrollo de la propuesta de solución se adapte al ciclo de vida definido para la actividad productiva de la UCI. Esta variante surge debido a que no existe una metodología de software universal, por

lo que las metodologías deben ser adaptadas a las características de cada proyecto (equipo de desarrollo, recursos, etc.).

AUP-UCI define cuatro escenarios de trabajo siendo elegido para llevar a cabo la presente investigación el escenario número cuatro, debido a que el cliente estará siempre acompañando al equipo de desarrollo para convertir los detalles en requisitos y así implementarlos generando como artefacto fundamental historias de usuarios.

1.11 Herramientas para el desarrollo de software

A continuación, se evidencia el estudio realizado y la selección de las herramientas para el desarrollo de software a utilizar durante la implementación de la solución, con el objetivo de una mejor comprensión. Las mismas permiten la confección y el desarrollo del componente.

1.11.1 Lenguajes del lado del servidor

Algunos ejemplos de lenguajes de programación son:

- ✓ JavaScript
- ✓ PHP

PHP

Es un lenguaje de código abierto muy popular, adecuado para el desarrollo web y que puede ser incrustado en Hyper Text Markup Language(HTML) por sus siglas en ingles. Es popular porque un gran número de páginas y portales web están creadas con PHP. Incrustado en HTML significa que, en un mismo archivo se puede combinar código PHP con código HTML, siguiendo unas reglas. PHP se utiliza para generar páginas web dinámicas (33). La versión a utilizar será la 5.0.

1.11.2 Lenguajes del lado del cliente

JavaScript

Es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (34).

Es un lenguaje sencillo y muy útil. Además, contiene una serie de funciones y sentencias utilizadas para facilitar la interacción con los documentos HTML, poder manipular de manera sencilla el Document Object Model (DOM) por sus siglas en inglés, desarrollar aplicaciones usando AJAX3 y manipular eventos.

HTML

HTML es un lenguaje de programación que se utiliza para el desarrollo de páginas de Internet. Se trata de la sigla que corresponde a Hyper Text Markup Language, es decir, Lenguaje de Formato de Documentos para Hipertexto. Consiste en un formato abierto que surgió a partir de las etiquetas SGML (34).

Este lenguaje es el que se utiliza para especificar los nombres de las etiquetas que se utilizarán al ordenar, no existen reglas para dicha organización, por eso se dice que es un sistema de formato abierto. Es un lenguaje muy simple y general que sirve para definir otros lenguajes que tienen que ver con el formato de los documentos. El texto en él se crea a partir de etiquetas, también llamadas tags, que permiten interconectar diversos conceptos y formatos.

El HTML permite ciertos códigos que se conocen como scripts, los cuales brindan instrucciones específicas a los navegadores que se encargan de procesar el lenguaje. Entre los scripts que pueden agregarse, los más conocidos y utilizados son JavaScript y PHP (34).

Para el desarrollo de la aplicación se decide utilizar HTML en su versión 5.

CSS

Las hojas de estilo en cascada (Cascading Style Sheets, CSS), permiten desarrollar la creatividad en el diseño. CSS constituye un mecanismo para asociar estilos de composición a documentos estructurados, del tipo HTML o XML. Aplicables a cualquier navegador, admiten un mayor control sobre los distintos elementos de una página, permitiendo definir el estilo de las fuentes, el color, el espaciado del texto, la posición del contenido, e incluso variaciones en el sonido en los elementos auditivos. Estos estilos pueden definirse para luego ser aplicados al código de cualquier documento (34).

Para el desarrollo de la aplicación se decide utilizar CSS en su versión 3.0.

JQuery

Es una biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web (35).

jQuery es software libre y de código abierto. Al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código.

1.11.1 Marcos de trabajo

Un entorno de desarrollo es una estructura de soporte definida, en la cual otro proyecto de software puede ser organizado y desarrollado. Suelen incluir: soporte de programas, bibliotecas para desarrollar y unir diferentes componentes de un proyecto de desarrollo de programas (36).

Symfony es un completo entorno de trabajo 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 (37).

Para el desarrollo de la solución se escogió Symfony por ser el marco utilizado en la implementación y desarrollo de la plataforma educativa ZERA, el cual se integrará a esta plataforma, la versión a utilizar será la 1.4.20.

Symfony fue diseñado para que se ajustara a los siguientes requisitos:

- ✓ Independiente del sistema gestor de bases de datos.
- ✓ 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 mejores prácticas y patrones de diseño para la web.
- ✓ Preparado para aplicaciones empresariales y adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- ✓ Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros (37).

Bootstrap es un framework que utilizando como base CSS3, permite desarrollar interfaces adaptables a distintos dispositivos ajustándose a su tamaño, además de elementos con estilos predefinidos que se irán describiendo en su apartado correspondiente.

Ofrece un diseño adaptativo, también conocido como Responsive Design. A pesar de la diversidad de elementos que aporta Bootstrap, se puede considerar que sus diseños son simples, limpios y rápidos; por lo que se ofrece una gran agilidad a la hora de cargarse en los navegadores y adaptarlos al tamaño de cada dispositivo.

Bootstrap combina el uso de JavaScript con plantillas CSS3 para ofrecer diversos elementos tales como botones, menús desplegables, formularios incluyendo todos sus elementos e integración jQuery para ofrecer ventanas y tooltips dinámicos (38).

1.11.4 Tecnologías del lado del cliente

Las tecnologías del lado del cliente no son más que los lenguajes en los que funcionan las aplicaciones web en sus navegadores.

AJAX

Son las siglas de Asynchronous JavaScript And XML, (Javascript asíncrono y XML). No es en sí un lenguaje de programación, sino una nueva técnica que combina varios lenguajes de programación. Permite crear páginas interactivas. La ventaja de AJAX respecto a otros lenguajes de programación web es la asincronía.

Ajax es una combinación de los siguientes lenguajes de programación y elementos:

HTML (o XHTML) y CSS: Base para el diseño de la página.

DOM y Javascript: Forma de acceder dinámicamente a las partes de la página.

Objeto XMLHttpRequest: Es el que permite la comunicación asíncrona con el servidor.

XML: Formato en el que están los datos que se solicitan al servidor; aunque otros formatos también pueden funcionar, como son HTML, texto plano (txt), json, etc.

1.11.5 Lenguaje y herramienta de modelado

El lenguaje modelado unificado Unified Modeling Language (UML) por sus siglas en inglés es un lenguaje gráfico que sirve para visualizar, especificar, construir y documentar un sistema de software. Ofrece un estándar para describir un “plano” del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del

sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables (32).

Herramienta para el Modelado UML

Las herramientas para el modelado son aplicaciones informáticas que permiten representar unidades y procesos que estarán presentes en toda la realización del software. Estas proporcionan un soporte para el desarrollo del software, lo cual facilita el proceso de construcción del mismo (39).

Visual Paradigm for UML es una herramienta de modelado desarrollada para asistir el proceso de Ingeniería de Software, este se encuentra basado en UML y soporta el ciclo de vida completo del desarrollo de software, además cuenta con funcionalidades más avanzadas que las presentes en el Rational Rose, lo que permite agilizar considerablemente el trabajo. Sus principales características son las siguientes: presenta licencia gratuita y comercial, soporta aplicaciones web, disponible en varios idiomas, fácil de instalar y actualizar, compatible entre versiones, entorno gráfico amigable para el usuario y disponible en múltiples plataformas (Windows/Linux/Mac OS X) (40). Para el desarrollo del sistema se utilizará la versión 8.0.

1.11.6 Servidor de Base de Datos

Un servidor de base de datos es un programa que brinda servicios de base de datos a otros programas u otras computadoras (41).

PostgreSQL es un motor de bases de datos orientada a objetos y libre, que tiene prestaciones y funcionalidades equivalentes a muchos gestores de bases de datos comerciales. Es más completo que MySQL ya que permite métodos almacenados, restricciones de integridad, vistas, etc (42).

Se basa en el uso de un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (43).

La versión a utilizar será la 9.2.4. Para su selección como el gestor de bases de datos para el desarrollo de la propuesta de solución se tuvo en cuenta su carácter libre, el conocimiento previo sobre el mismo y la facilidad de manejo de las bases de datos PostgreSQL. Además de ser la usada por el centro para el desarrollo de la plataforma.

1.11.7 Entorno de Desarrollo Integrado

Un IDE, es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, PHP, Python, Java, C#, Delphi, Visual Basic, etc (44).

NetBeans IDE: Es un proyecto de código abierto dedicado a proveer un sólido desarrollo de software, dirigido fundamentalmente a las necesidades de los desarrolladores y los usuarios; dotándolos de una herramienta para el desarrollo rápido y fácil de productos de software. NetBeans es una herramienta modular de desarrollo que incluye un avanzado editor multilenguaje y un detector de errores, además permite la integración con PHP, la librería JQuery de JavaScript y Symfony. Además, cuenta con una amplia documentación y una gran comunidad de usuarios (45). La versión a utilizar es la 7.2.

1.11.8 Servidor web

Un servidor web es un “programa que atiende y responde a las diversas peticiones de los navegadores, proporcionándoles los recursos que solicitan mediante el protocolo HTTP o el protocolo HTTPS.

Entre los servidores web más usados se puede mencionar:

Apache

Apache 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 (Apache Group) que, usando Internet y la web para comunicarse, planifican y desarrollan el servidor y la documentación relacionada. Además, se caracteriza por ser un servidor web que posee gran rapidez, goza de mucha eficiencia y flexibilidad y se adapta a los nuevos protocolos HTTP. Todo ello se une al hecho de ser multiplataforma, adaptable a diferentes entornos y necesidades y extensible a partir de su condición de modularidad (46).

La selección de Apache como servidor web estuvo determinada porque el mismo está respaldado por una comunidad de usuarios numerosa, por lo que existe una documentación

extensa del mismo disponible en Internet, además del conocimiento previo del autor sobre el trabajo con dicho servidor web. La versión a utilizar es la 2.4.10.

1.12 Conclusiones

Se logró realizar un análisis de los conceptos referentes al proceso de gestión de contenidos, las herramientas para el acceso al e-learning en los LMS, así como las herramientas tecnologías y la metodología guía para el desarrollo de la investigación.

Para el desarrollo de dichas funcionalidades y tomando como referencia las características mencionadas y explicadas en la selección de las herramientas y metodologías se utilizará: como lenguaje de programación PHP v5.0, integrando el marco de trabajo Symfony 1.4.20, como IDE NetBeans 7.2 como servidor Web Apache 2.4.10, gestor de base de datos PostgreSQL 9.2.4, para apoyar el proceso de desarrollo de software se decidió utilizar la metodología AUP en su variante UCI y el lenguaje de modelado UML 2.0 y como herramienta para el modelado de los diagramas necesarios Visual Paradigm 8.0.

Capítulo 99 Propuesta de solución

Para lograr una mejor comprensión del sistema que se desea implementar, es necesario una correcta definición de los conceptos asociados al dominio del problema que intervienen en este durante el desarrollo del software.

En este capítulo se realiza una descripción de la solución propuesta con el objetivo de proporcionar un mejor entendimiento del flujo de trabajo asociado al sistema. Además, se representa el diagrama del Modelo del dominio con la correspondiente descripción de cada uno de sus elementos. También se lleva a cabo la captura de requisitos pertenecientes al Flujo de Requerimientos, a través del cual se determinan los requerimientos funcionales y no funcionales con el objetivo de conocer con claridad las capacidades y cualidades que el producto debe tener.

2.1 Descripción de la solución propuesta

Se integrarán a la Plataforma Educativa ZERA nuevas funcionalidades que le permitan al docente, secuenciar los capítulos, temas y subtemas definidos dentro de un curso, con el objetivo de adaptarlos a las diferencias individuales de cada estudiante. Además, permitirá al profesor definir como serán visualizados los cursos a los estudiantes. El sistema contará con la actual gestión de recorridos dirigidos, los cuales son una tarea que el profesor le diseña a cada estudiante en particular o a un grupo enfocándose en las características particulares de su aprendizaje.

Se proporcionarán funcionalidades que permitirán la gestión de secuencias de contenidos y actividades. Para la creación de estas, el docente seleccionará los capítulos, temas y subtemas a secuenciar. Con las diferentes actividades que contiene el subtema podrá ir estructurando la secuencia e ir conformando los recorridos dirigidos personalizados. Se podrá modificar una actividad secuenciada cambiando los elementos que se consideren necesarios, siguiendo un proceso similar al que se describe en la creación. Una vez que existan subtemas secuenciados, el profesor podrá decidir en dependencia de las características de sus estudiantes si verán secuenciado o no el contenido de un curso. En caso de que el docente decida mostrar el contenido de un programa de estudio secuenciado, se le mostrará al estudiante las actividades que puede realizar en los diferentes capítulos, temas y subtemas de un curso. El camino recorrido por el estudiante

al transitar por una secuencia podrá ser diferente al que recorran otros estudiantes en los mismos contenidos.

2.2 Modelo de dominio

EL modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las “cosas” que existen o los eventos que suceden en el entorno en que trabaja el sistema.

2.2.1 Conceptos del dominio

Docente: es la persona encargada de configurar el modo en el cual se mostrarán los contenidos a los estudiantes, ya sea secuenciado o no.

Estudiante: es la persona que resuelve las actividades del contenido del curso, visualiza los recursos y las actividades con las que cuenta, esté secuenciado o no, con el objetivo de incrementar sus conocimientos, y al mismo tiempo que el profesor lleve un seguimiento de su aprendizaje, para atender sus diferencias individuales.

Recorrido dirigido: es la secuencia que se crea sobre las actividades que contiene un subtema de un determinado capítulo.

Recorrido dirigido personalizado: es la secuencia que se crea sobre las actividades que contiene un curso en general.

Elemento: representa los diferentes tipos de actividades que se pueden incluir en una actividad secuenciada, estos son: páginas y ejercicios.

Página: son los diferentes elementos por los que está compuesto un contenido. Contiene textos y puede incluir varios recursos y ejercicios.

Ejercicio: es una rutina que el estudiante debe realizar con el objetivo de fijar un contenido.

2.2.2 Diagrama de modelado del dominio

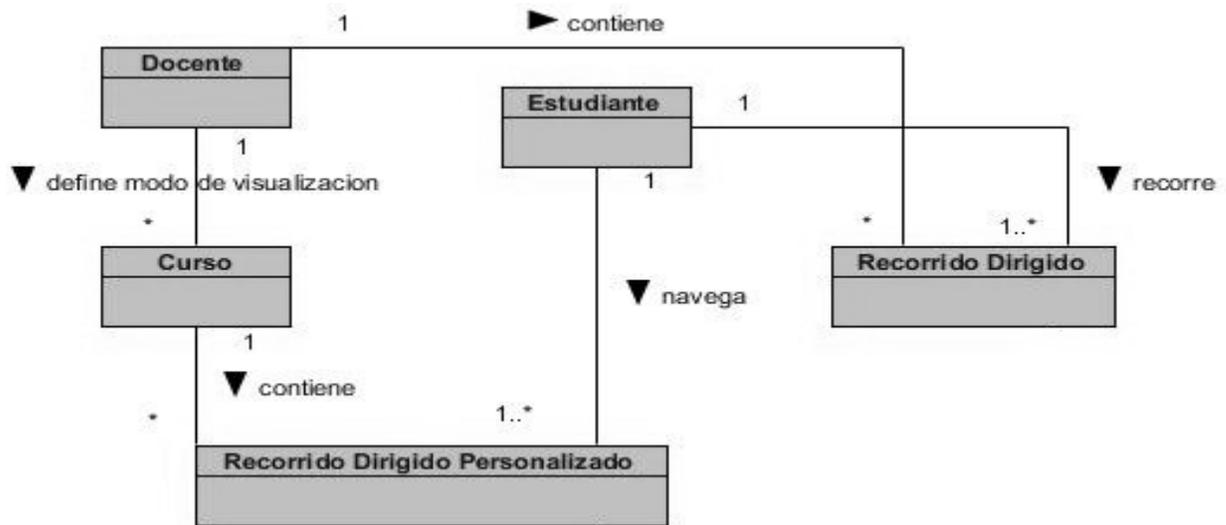


Figura 1 Diagrama de modelado del dominio

2.3 Requerimientos del software

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

2.3.1 Requerimientos Funcionales

RF1: Crear recorrido dirigido personalizado: el docente podrá crear recorridos dirigidos personalizados de acuerdo a las diferencias individuales de cada estudiante.

RF2: Editar recorrido dirigido personalizado: el docente podrá editar un recorrido dirigido personalizado ya existente.

RF3: Listar recorridos dirigidos personalizados: el docente podrá listar los recorridos dirigidos personalizados existentes.

RF4: Eliminar recorrido dirigido personalizado: el docente podrá eliminar un recorrido dirigido personalizado existente.

RF5: Mostrar recorrido dirigido personalizado: el estudiante podrá ver el recorrido dirigido personalizado por el profesor para él en caso de haber sido creado.

RF6: Seleccionar del índice: el docente podrá seleccionar del índice de la asignatura, los capítulos, temas y subtemas que se desea secuenciar.

RF7: Ver actividades de la secuencia: el docente podrá ver cada una de las actividades que componen la secuencia.

RF8: Modificar elementos de la secuencia: el docente podrá modificar los diferentes elementos que componen la secuencia.

RF9: Editar nombre y descripción de la actividad secuenciada: el docente podrá modificar el nombre y la descripción de la secuencia de forma rápida y sencilla.

RF10: Realizar recorrido dirigido personalizado: el estudiante transitará por el curso de la forma en la que se le definió sobre los subtemas de los programas de estudio.

RF11: Mostrar datos de la actividad secuenciada: el docente podrá ver el nombre, la descripción, el subtema y la materia a la que pertenece cada una de las actividades secuenciadas.

RF12: Definir modo de visualización del contenido: el docente podrá configurar la forma en que desea que sus estudiantes vean el programa de estudio ya sea secuenciado o no.

RF13: Notificar al estudiante: el docente podrá notificarle al estudiante mediante el correo interno de la plataforma cada vez que su curso sea personalizado.

2.3.2 Requerimientos no funcionales

Los requerimientos no funcionales (RNF) son propiedades o cualidades con las que el producto debe cumplir, como restricciones al diseño o funcionamiento del sistema software (tal como requisitos de funcionamiento, estándares de calidad, o requisitos del diseño) (47).

A continuación, se muestran los requisitos.

RNF1. Usabilidad

El módulo estará diseñado de manera que los usuarios adquieran las habilidades necesarias para explotarlo en un tiempo reducido.

Serán reutilizados diseños similares a los que cuenta la Plataforma Educativa ZERA con vistas a aprovechar la experiencia de usuario.

RNF2. Apariencia o Interfaz Externa

Consistente: las acciones similares deben ejecutarse de forma similar.

RNF3. Seguridad

El acceso a la información debe estar restringido por usuario, contraseña y rol.

RNF4. Disponibilidad

Las horas de operaciones del sistema deberán ser de 24 horas al día, 7 días a la semana, los 365 días del año.

2.4 Actores del sistema

El sistema contará con los siguientes actores: docente y estudiante. A continuación, se describen las acciones que podrán realizar dichos actores en el sistema.

2.4.1 Descripción de los actores del sistema

Tabla 1 Descripción de los actores del sistema.

Actor	Descripción
Docente	Actor que podrá crear, editar, eliminar y ver los datos de los recorridos dirigidos personalizados.
Estudiante	Actor que transitará el curso secuenciado o no.

2.5 Historias de usuario

Una historia de usuario es una representación de un requisito de software escrito en una o dos frases utilizando el lenguaje común del usuario. Las historias de usuario son utilizadas en las metodologías de desarrollo ágiles para la especificación de requisitos.

Tabla 2 Historia de usuario crear recorrido dirigido personalizado.

Historia de usuario	
Número: 1	Nombre del requisito: Crear RDP
Programador: Armando Ávila Riveron	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 3 días
	Tiempo Real: 2 días
Descripción: 1- Objetivo: Permitir incluir nuevos recorridos dirigidos en el sistema. 2- Acciones para lograr el objetivo (precondiciones y datos): Para crear un nuevo recorrido dirigido hay que: <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol docente. - Tener asignado un grupo al cual impartir clases. 	

3- Comportamientos válidos y no válidos (flujo central y alternos):

Los campos nombre y descripción son obligatorios.

Nombre: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 100 caracteres

Descripción: campo de texto que permite cualquier carácter.

4- Flujo de la acción a realizar:

- El sistema debe permitir incluir y/o seleccionar los datos para incluir un nuevo RDP.
- Cuando el usuario incluye y/o selecciona correctamente los datos necesarios para incluir un RDP, el sistema mostrará un mensaje de elemento creado correctamente.

Observaciones: Los RDP son creados para cada estudiante en particular, no puede haber dependencia entre ellos.

Prototipo de interfaz:

The screenshot shows a web application interface. At the top, there is a header bar with the name 'male' on the left and navigation icons (list, mail, user profile) on the right. Below the header is a menu bar with options: 'Contenidos', 'Principal', 'Tareas', 'Actividades', 'Docente', and 'Personalizar RD'. The 'Personalizar RD' option is currently selected. Below the menu bar is a search bar with a magnifying glass icon and a dropdown menu. The main content area displays the title 'Añadir un Recorrido Dirigido Personalizado.' followed by a form with the following fields: 'Nombre del Recorrido' (text input), 'Estudiante' (dropdown menu with the text 'Seleccione al menos uno:'), and two buttons: 'Cancelar' and 'Guardar'.

2.6 Modelo de análisis

Un modelo de análisis se describe utilizando el lenguaje de los desarrolladores y puede por tanto introducir un mayor formalismo y ser utilizado para razonar sobre los aspectos internos del sistema. Es utilizado fundamentalmente por los desarrolladores para comprender cómo debería ser diseñado e implementado el sistema (Ivar Jacobson, 2000).

2.7 Diagrama de clases del análisis

Una clase de análisis representa una abstracción de una o varias clases, ajustando las mismas a uno de los tres estereotipos existentes sobre las clases utilizados por el modelo de dominio: de interfaz, de control o de entidad.

La clase interfaz se utiliza generalmente para modelar la interacción entre el sistema y los actores, la clase de control es utilizada habitualmente para representar coordinación, secuenciación, transacciones y son las encargadas de manejar y coordinar las acciones y los flujos de control principal, por su parte la clase entidad es usada para modelar la información que tiene una vida larga y a veces es persistente, asimismo muestran una estructura de datos lógica y contribuyen a comprender de qué información depende el sistema (32). A continuación se presenta el diagrama de clases el resto de los diagramas se muestra en el anexo dos y el anexo tres.

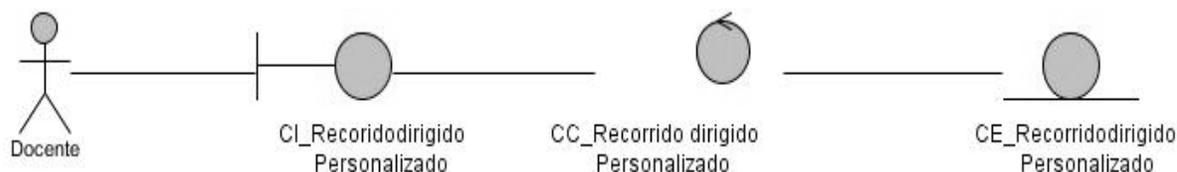


Figura 2 Diagrama de clases del análisis Crear RD.

2.8 Diagrama de colaboración del análisis

Un diagrama de colaboración es esencialmente un diagrama que muestra interacciones organizadas alrededor de los roles. A diferencia de los diagramas de secuencia, los diagramas de colaboración, también llamados diagramas de comunicación, muestran explícitamente las relaciones de los roles. Por otra parte, un diagrama de comunicación no muestra el tiempo como una dimensión aparte, por lo que resulta necesario etiquetar con números de secuencia tanto la secuencia de mensajes como los hilos concurrentes (32).

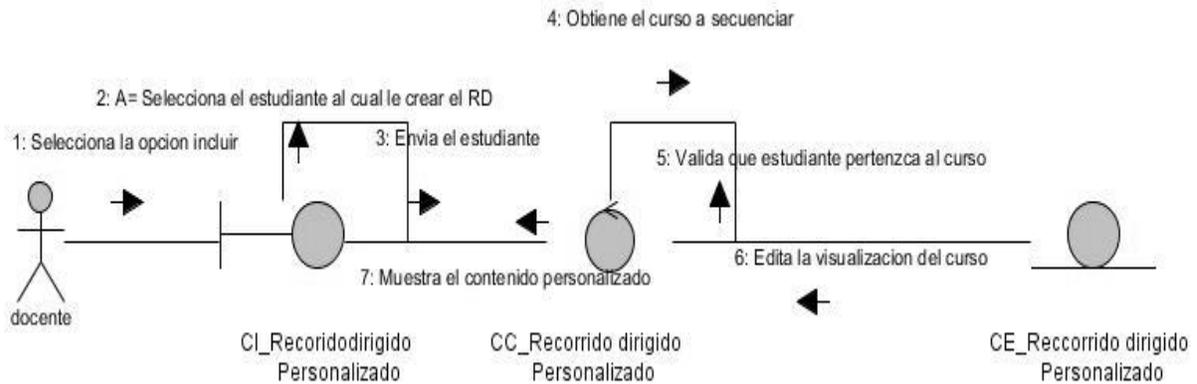


Figura 3 Diagrama de colaboración del análisis crear RD.

2.9 Diagrama de clases del diseño

Los diagramas de clases del diseño son llamados diagramas estáticos porque muestran las diferentes clases que componen un sistema, cómo se relacionan las clases con otras (32).

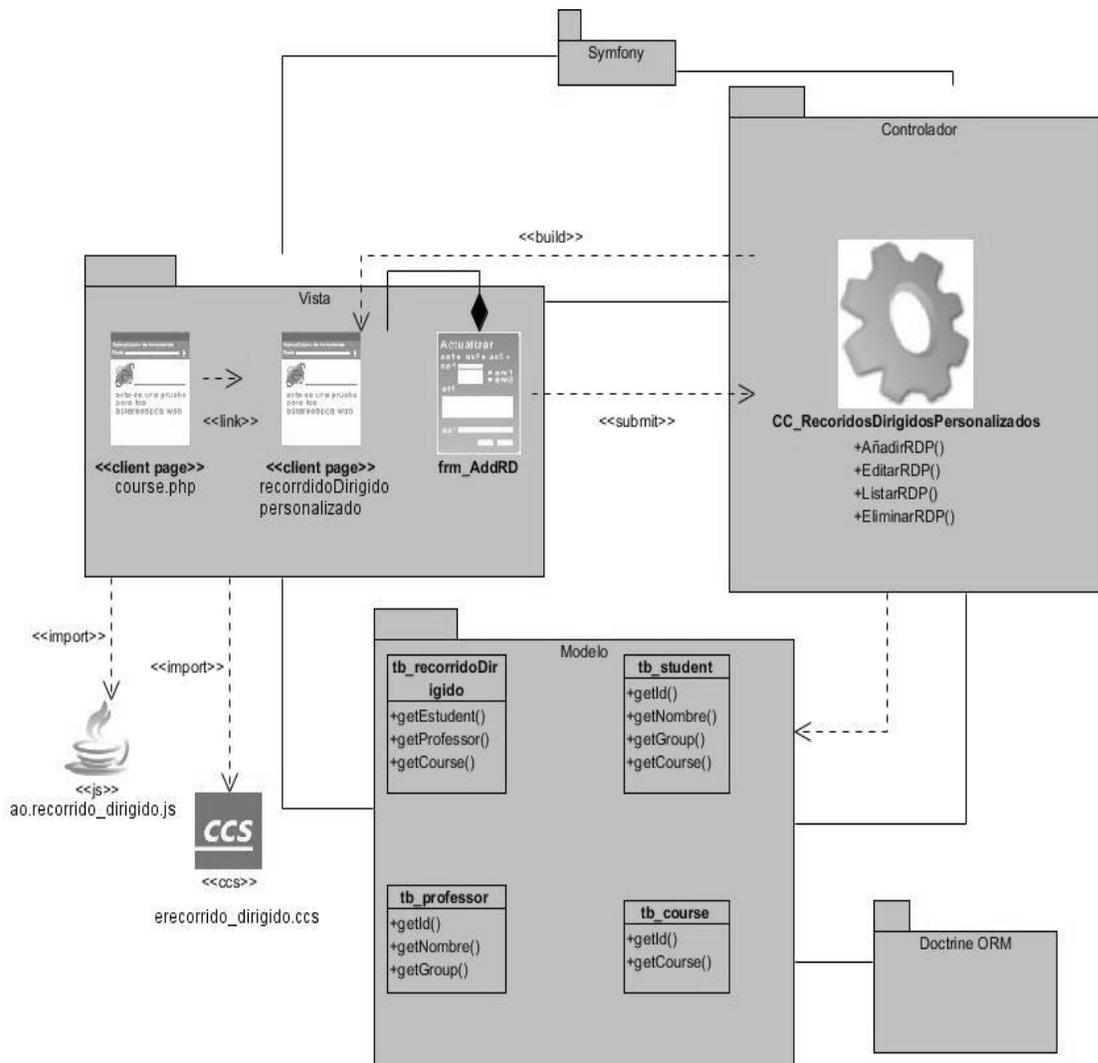


Figura 4 Diagrama de Clases del dominio Crear RD.

2.10 Diagrama de secuencia del diseño

El propósito del diseño es especificar una solución que trabaje y pueda ser fácilmente convertida en código fuente y construir una arquitectura simple y fácilmente extensible. Las clases definidas en el análisis fueron detalladas, y se añadieron nuevas clases para manejar áreas técnicas como base de datos, interfaz del usuario, comunicación, dispositivos, etcétera.

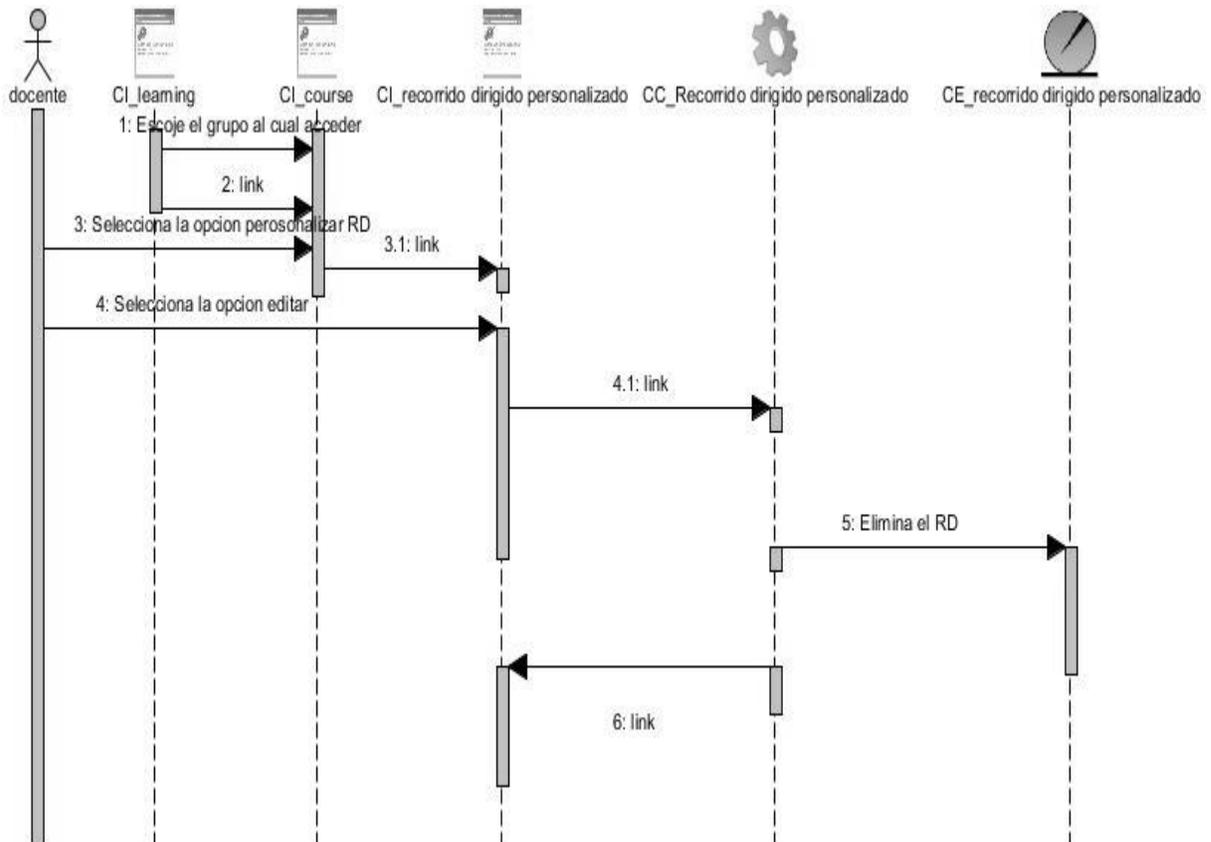


Figura 5 Diagrama de secuencia del diseño crear RD.

2.11 Diagrama de despliegue

El diagrama de despliegue es utilizado para mostrar los nodos y conexiones del modelo de despliegue, así como la asignación de los objetos a los nodos (32).



Figura 6 Diagrama de despliegue.

2.12 Patrones de diseños aplicados

Symfony, como framework, hace uso en su implementación de un conjunto de patrones de diseño, los cuales proveen un esquema para refinar los subsistemas y componentes de un sistema de software, o las relaciones entre ellos. “Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software”. En otras palabras, brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares (48).

Symfony es un framework que sigue la mayoría de los patrones de diseño para la web, entre ellos los patrones GRASP (Patrones Generales de Software para Asignación de Responsabilidades, traducido al inglés General Responsibility Assignment Software Patterns) y GoF (Grupo de 4, traducido al inglés Gang of Four) (37).

2.12.1 Patrones GRASP

Describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones; son una serie de buenas prácticas enfocadas a la calidad del software (49).

Experto: Consiste en que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo o ejecutarlo. De este modo se obtendrá un diseño con mayor cohesión y así la información se mantiene encapsulada, es decir, disminuye el acoplamiento (49).

```
public function executeAddContentRD(sfWebRequest $request) {
    try {
        $study_program = sfUserTools::getSessionVar('study_program_id');
        $study_program_name = sfUserTools::getSessionVar('study_program_name');
        $profesor = sfUserTools::getSessionVar('user_id');
        $estudiante = $request->getParameter('estudiante');

        $estudiante_obj = Doctrine::getTable('TbStudent')->findOneById($estudiante);

        $estudiante_name = $estudiante_obj->getSfGuardUser()->getFullName();

        $matter = sfUserTools::getSessionVar('matter_id');

        $matter_obj = Doctrine::getTable('TbMatter')->findOneById($matter);
    }
}
```

Figura 7 Patrón Experto.

En la imagen se muestra la creación de un recorrido dirigido, en la cual se puede observar que la clase actions.class.php con la definición de variables accede a toda la información necesaria para su creación.

Controlador: Es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. Guía la asignación de responsabilidades relacionadas con la creación de objetos, permitiendo instanciar y crear las clases que le son necesarias para cumplir sus funcionalidades. La arquitectura MVC brinda una capa específicamente para los controladores, que son el núcleo de este, y especifica la presencia de este patrón (49).

```

<select id="estudiante" name="estudiante" required="required" placeholder="Estudiante">
  <option value="" selected="selected">Seleccione</option>
  <?php
  $dataStudents = "";
  foreach ($students as $key => $student) {
    $dataStudents .= '<option value="' . $student['student_id'] . '">' . $student['student_first_name'] . ' ' . $student['student_first_last_name'] . ' ' . $student['student_second_last_name'] . '</option>';
  }
  echo($dataStudents);
  ?>
  <-
  <option value="<?php echo $students[0]['student_id'] ?>"><?php echo $students[0]['student_first_name'] ?></option>
  ->
</select>
</div>
<div class="col-lg-offset-2 col-lg-10">

```

Figura 8 Patrón Controlador.

En la imagen se muestra cómo se escogen los datos del estudiante los cuales serán usados en la creación del recorrido dirigido personalizado.

Creador: Asigna responsabilidades relacionadas con la creación de objetos o instanciación de nuevos objetos o clases. El propósito fundamental de este patrón es encontrar un creador que se debe conectar con el objeto producido en cualquier evento (49).

```

public function executeListContent(sfWebRequest $request) {
  $body = $body = $this->getPartial('sfGuardForgotPassword/new_password', array('user' => $this->sf_guard_zfToolsEvents::newSendMail($subject, $body, array($user)));
  // Creando variables de la acción para pasar información a la plantilla
  $this->setVar('parametro', 'valor');
  $this->parametro = 'valor'; // Versión corta.

  // $conn = Doctrine_Manager::connection();
  // $conn->beginTransaction();
  $group_id = $request->getParameter('group_id', sfUserTools::getSessionVar('group_id'));

  $user_object = zfCourse::getUserCourse();
  $professor_id = $user_object->getProfessorId();
  $students = TbGroupTable::getInstance()->showStudentsbyTeacher2($professor_id, $group_id);
  var_dump($user_object);
}

```

Figura 9 Patrón Creador.

En la imagen se muestra como con el método de creación de la lista de contenidos en la clase actions.php se asegura que cada vez que se vaya a crear una lista de contenidos dentro de la herramienta son instanciados los elementos necesarios para así proceder a la creación de la lista de contenido.

Alta Cohesión: Brinda una solución al problema de “asignar una responsabilidad de manera que la cohesión permanezca alta” (50). En el caso del sistema en las clases controladoras se evidencia el uso de este patrón pues las mismas están formadas por varias funcionalidades que están estrechamente relacionadas, siendo estas las encargadas de definir las acciones y colaborar con otras para realizar tareas que las implican a todas. Se evidencia en la clase actions.class.php la cual está formada por varias funcionalidades que están estrechamente relacionadas, siendo 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: Posibilita la idea de tener las clases lo menos relacionadas y en caso de cualquier modificación la repercusión de la misma sea menor potenciando la reutilización (49). Las clases que implementan la lógica de negocio y de acceso a datos se encuentran en el modelo, estas clases no tienen asociaciones con las de la vista o el controlador por lo que la dependencia en este caso es baja, demostrándose así el uso de este patrón. Este se evidencia en la clase actions.class.php que hereda únicamente de sfActions. Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia sea baja.

2.13 Patrón arquitectónico Modelo Vista Controlador

Modelo Vista Controlador (MVC): Es una propuesta de diseño de *software* utilizada para implementar sistemas donde se requiere el uso de interfaces de usuario. Surge de la necesidad de crear *software* más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos (51).

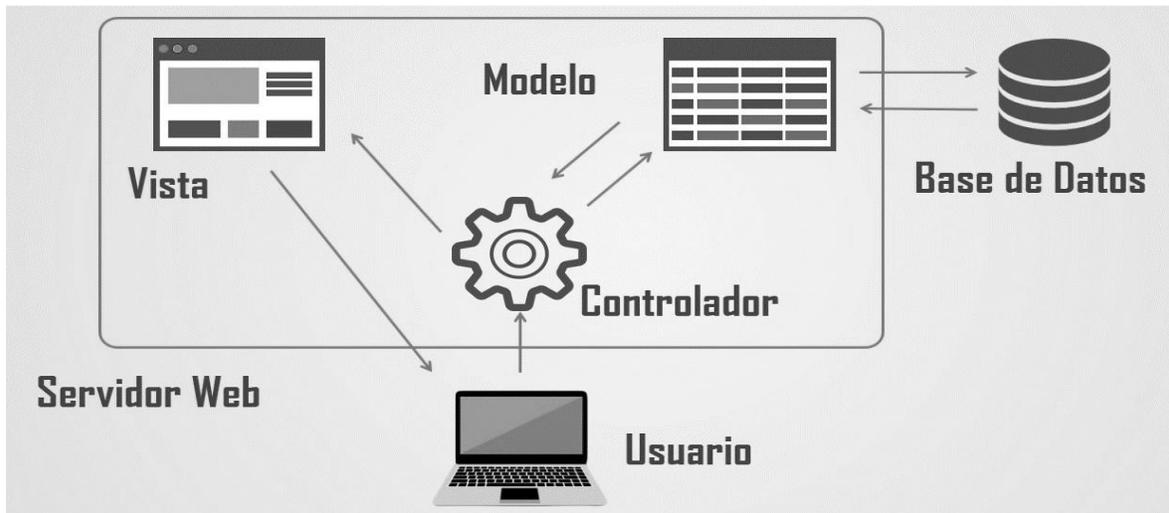


Figura 10 Patrón modelo vista controlador.

A continuación, se describe el funcionamiento del patrón arquitectónico MVC (52).

El modelo representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El modelo no tiene conocimiento específico de los controladores o de las vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el modelo y sus vistas, y notificar a las vistas cuando cambia el modelo.

La vista maneja la presentación visual de los datos representados por el modelo. Genera una representación visual del modelo y muestra los datos al usuario. Interactúa preferentemente con el controlador, pero es posible que trate directamente con el modelo a través de una referencia al propio modelo.

El controlador proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el modelo, centra toda la interacción entre la vista y el modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del modelo o por alteraciones de la vista. Interactúa con el modelo a través de una referencia al propio modelo.

2.14 Conclusiones

La identificación de los requerimientos funcionales y no funcionales permitió determinar las exigencias y restricciones que el sistema debe cumplir.

Se elaboró una propuesta de solución del sistema a partir de determinar los artefactos generados por la metodología AUP-UCI necesarios para iniciar el análisis y diseño de dicha propuesta.

Se especificó el patrón arquitectónico y de diseño utilizados en el desarrollo de las funcionalidades.

Capítulo 999 Implementación y prueba

Las entradas fundamentales del flujo de trabajo implementación son los artefactos que se generaron durante el flujo de trabajo análisis y el diseño. Este flujo de trabajo tiene como objetivo implementar clases y objetos en forma de componentes, definir la organización del código y probar los componentes desarrollados. Para poner a prueba los componentes desarrollados, se comienza a realizar la evaluación de la calidad del producto, para ello es necesario que dicho componente funcione como está previsto y que la validación de los requisitos se aplique correctamente.

3.1 Modelo de implementación

El modelo de implementación es la entrada principal de las etapas de prueba que siguen a la implementación. Más concretamente, durante la etapa de prueba cada construcción generada durante la implementación es sometida a pruebas de integración, y posiblemente también a pruebas de sistema (53).

3.1.1 Diagrama de componentes

Los diagramas de componentes se utilizan para modelar la vista estática de un sistema. Comprende entre sus principales objetivos mostrar la dependencia lógica entre los distintos componentes del software y representar las relaciones entre los elementos que forman el código del sistema implementado (32).

Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los diagramas de componentes prevalecen en el campo de la arquitectura de software, pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema.

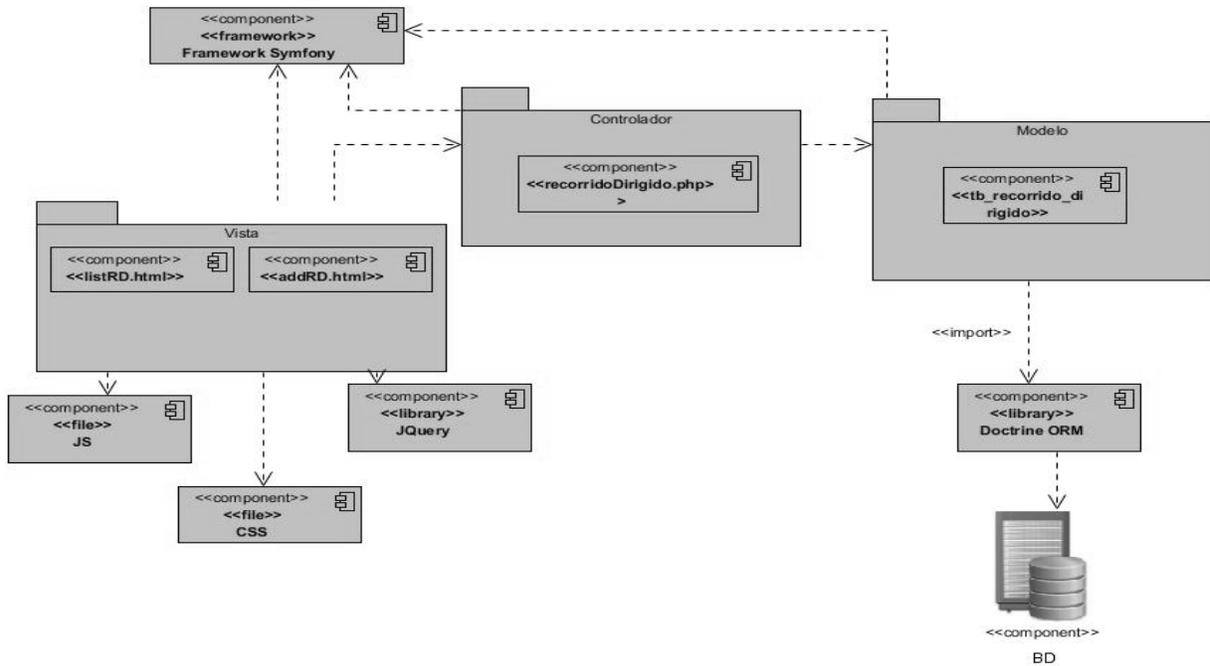


Figura 11 Diagrama de componentes crear RD.

3.2 Modelado de datos

En esta sección se presenta el modelo de datos que estará compuesto por las entidades que pasarán a ser las tablas de la base de datos, para ser utilizadas por las funcionalidades a desarrollar. Este se encuentra simplificado para una mejor visualización, y para un mejor entendimiento las tablas son descritas en el Anexo # 3.

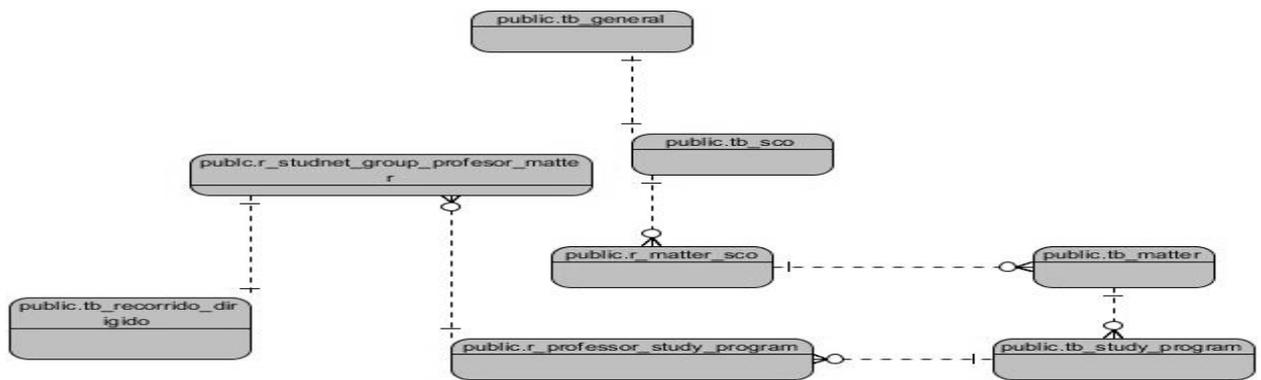


Figura 12 Modelado de datos.

3.3 Pruebas de software

El único instrumento adecuado para determinar el status de la calidad de un producto software es el proceso de pruebas. En este proceso se ejecutan pruebas dirigidas a componentes del software o al sistema de software en su totalidad, con el objetivo de medir el grado en que el software cumple con los requerimientos. En las pruebas se usan casos de prueba, especificados de forma estructurada mediante técnicas de prueba (54).

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

3.3.1 Niveles de Prueba

El proceso de pruebas se realiza en varios niveles, estos se enfocan a determinados objetivos y están estrechamente relacionados con los tipos de pruebas. En cada uno de estos niveles de prueba, se podrán ejecutar diferentes tipos de prueba tales como: pruebas funcionales, no funcionales, de arquitectura y asociadas al cambio de los productos (55).

A continuación, una breve descripción de cada nivel de prueba:

- ✓ **Pruebas unitarias:** este tipo de pruebas son ejecutadas normalmente por el equipo de desarrollo, consisten en la ejecución de actividades que le permitan verificar al desarrollador que los componentes unitarios están codificados bajo condiciones de robustez, soportando el ingreso de datos erróneos o inesperados y demostrando así la capacidad de tratar errores de manera controlada.
- ✓ **Pruebas de integración:** este tipo de pruebas son ejecutadas por el equipo de desarrollo y consisten en la comprobación de que elementos del software que interactúan entre sí, funcionan de manera correcta.
- ✓ **Pruebas de sistema:** este tipo de pruebas deben ser ejecutadas idealmente por un equipo de pruebas ajeno al equipo de desarrollo, una buena práctica en este punto corresponde a la tercerización de esta responsabilidad. La obligación de este equipo, consiste en la ejecución de actividades de prueba en donde se debe verificar que la funcionalidad total de un sistema fue implementada de acuerdo a los documentos de especificación definidos en el proyecto.

3.3.2 Diseño de casos de prueba

Se trata de diseñar pruebas que tengan la mayor probabilidad de encontrar el mayor número de errores con la mínima cantidad de esfuerzo y de tiempo. Cualquier producto de ingeniería se puede probar de dos formas (56).

- ✓ Pruebas de caja negra: realizar pruebas de forma que se compruebe que cada función es operativa.
- ✓ Pruebas de caja blanca: desarrollar pruebas de forma que se asegure que la operación interna se ajusta a las especificaciones, y que todos los componentes internos se han probado de forma adecuada.

En la prueba de la caja negra, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta (56).

Tabla 3 Diseño de caso de prueba Incluir RDP.

Escenario	Descripción	Grupo	Estudiante	Respuesta del sistema	Flujo central
EC 1.1 Opción Crear RDP.	Selecciona la opción de nuevo elemento			Brinda la posibilidad de seleccionar de manera obligatoria los siguientes datos. Grupo, Estudiante, podrá cancelar la operación cuando desee.	Curso/Recorrido Dirigido Personalizado
EC 1.2 Opción de Guardar los datos.	Introduce y/o selecciona los datos y selecciona la opción guardar los datos.	V	V	Valida los datos. Crea un recorrido dirigido personalizado. Muestra el listado de recorridos dirigidos personalizados y muestra el siguiente mensaje "Elemento añadido correctamente".	Curso/Recorrido Dirigido Personalizado/Guardar
EC 1.3 Opción de cancelar.	Selecciona la opción de Cancelar.			Elimina los datos creados. Regresa al listado de recorridos dirigidos personalizados.	Curso/Recorrido Dirigido Personalizado/Cancelar
EC 1.4 Datos incompletos	Existen datos incompletos.	I	V	Muestra un mensaje de información. Muestra un indicador sobre los campos vacíos. Regresa al EC 1.1.	Curso/Recorrido Dirigido Personalizado/Guardar
		V	I		
		V	V		
EC 1.5 Datos incorrectos	Existen datos incorrectos.	I	V	Muestra un mensaje de información. Muestra un indicador sobre los campos incorrectos.	Curso/Recorrido Dirigido Personalizado/Guardar
		V	I		

3.3.4 Resultados Obtenidos

Durante el desarrollo de las funcionalidades para la visualización y gestión de la secuenciación de contenidos y actividades se realizaron pruebas unitarias al código con el objetivo de comprobar su correcta implementación. Estas pruebas fueron realizadas por el propio desarrollador, apoyándose en la compilación paso a paso que brinda el IDE. Los resultados de dichas pruebas no fueron documentados pues se realizaron en conjunto con el desarrollo de la solución. Durante el desarrollo de las funcionalidades para la visualización y gestión de la secuenciación de contenidos y actividades se realizaron pruebas unitarias al código con el objetivo de comprobar su correcta implementación.

Estas pruebas fueron realizadas por el propio desarrollador, apoyándose en la compilación paso a paso que brinda el IDE. Los resultados de dichas pruebas no fueron documentados pues se realizaron en conjunto con el desarrollo de la solución.

Además, se realizaron pruebas de caja negra haciendo uso de los diseños de casos de prueba con el objetivo de comprobar el correcto funcionamiento de las funcionalidades desarrolladas. A continuación, se muestra una tabla resumen de las no conformidades identificadas durante las tres iteraciones de pruebas realizadas. En la tabla se muestran los requisitos funcionales y asociado a cada uno, por cada iteración, la cantidad de no conformidades encontradas (NC). Además, se muestra la cantidad de no conformidades identificadas clasificadas en satisfechas (S) y no satisfechas (NS).

Tabla 4 Pruebas de Caja Negra.

RF	Iteración I			Iteración II			Iteración III		
	NC	S	NS	NC	S	NS	NC	S	NS
RF 1	2	2	-	-	-	-	-	-	-
RF 2	1	1	-	-	-	-	-	-	-
RF 3	1	1	-	1	-	1	-	-	-
RF 4	3	2	1	2	2	-	-	-	-

RF 5	2	1	1	1	1	-	-	-	-
RF 6	-	-	-	-	-	-	-	-	-
RF 7	1	1	-	-	-	-	-	-	-
RF 8	3	2	1	1	1	-	-	-	-
RF 9	2	1	1	1	1	-	1	1	-
RF 10	-	-	-	-	-	-	-	-	-
RF 11	1	-	1	2	1	1	1	1	-
RF 12	3	2	1	1	-	1	1	1	-
RF 13	-	-	-	-	-	-	-	-	-
Total	19	13	6	9	6	3	3	3	-

A continuación, se muestra un gráfico donde se puntualiza por iteraciones el total de no conformidades identificadas, el total de no conformidades resueltas y la cantidad de no conformidades pendientes.

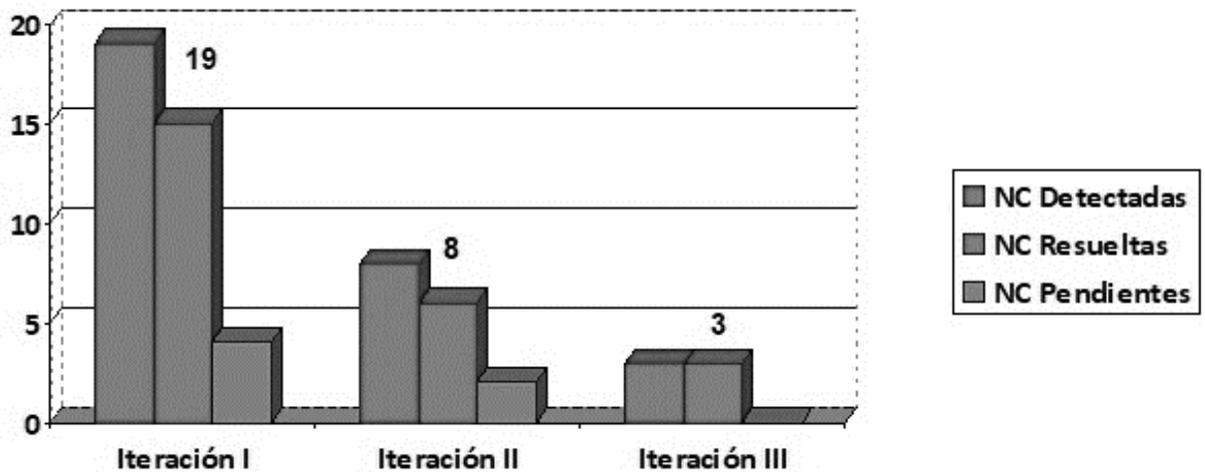


Figura 13 Total de no conformidades identificadas.

3.4 Conclusiones

Durante el flujo de trabajo implementación se generaron los artefactos que conformaron el modelo de datos, el cual describe cómo los elementos del modelo de diseño se implementan y organizan. Con las pruebas realizadas a la herramienta desarrollada, en la primera iteración se identificaron 19 no conformidades. En la segunda iteración se hallaron ocho no conformidades. En la tercera iteración, tres no conformidades, las cuales fueron resueltas por lo que se puede concluir que la aplicación desarrollada cumple con la calidad requerida.

Conclusiones

El presente trabajo de diploma da cumplimiento a las tareas de investigación propuestas inicialmente, logrando adaptar el contenido de un curso a las diferencias individuales de uno o varios estudiantes en la plataforma educativa ZERA permitiendo arrojar los siguientes resultados:

- El marco teórico elaborado permitió fundamentar y elaborar los conceptos relacionados con el proceso de gestión de contenidos en sistemas para el aprendizaje y sus principales tendencias.
- La metodología utilizada, permitió guiar todo el proceso de desarrollo del software, generando los principales artefactos que permitieron una mejor comprensión y desarrollo de la herramienta.
- Se validó la solución a través del tipo de prueba funcional caja negra, arrojando como resultado el buen funcionamiento de la solución, quedando de esta forma resuelto el problema investigativo.

Recomendaciones

Los resultados obtenidos, luego del desarrollo de la presente investigación satisfacen los requisitos definidos, no obstante, se recomienda:

- Continuar el desarrollo de la investigación con el objetivo de perfeccionar y aumentar las funcionalidades de la herramienta.
- Capacitar a todo el personal involucrado en el proceso de gestión de contenidos dentro de la plataforma para su óptimo funcionamiento.

Bibliografía

1. **Roberto Carneiro, Juan Carlos Toscano, Tamara Díaz.** *Los desafíos de las TIC para el cambio educativo.* Madrid, España : Organización de Estados Iberoamericanos para la Educación, la Ciencia y la Cultura (OEI), 2018. ISBN: 978-84-7666-197-0. Madrid, España..
2. **S, Isabel.** *Beneficios de la tecnología en educación.* Santiago de Chile, Chile : s.n., 2016.
3. **Manuel Area Moreira, Jordi Adell Segura.** *e-Learning: Enseñar y Aprender en Espacios Virtuales.* 2009.
4. **Ros, Miguel Zapata.** *Secuenciación de contenidos. Especificaciones para la secuenciación instruccional de objetos de aprendizaje.* Alcalá de Henares : s.n., 2009.
5. **Miguel Angel Conde, Carlos Muñoz, Francisco Jose Garcia.** *Sistemas de adaptación de contenidos para dispositivos móviles.* Albacete : s.n., 2008.
6. **Inclusiva.** *www.ite.educacion.es.* [En línea] 2011. [Citado el: 1 de diciembre de 2015.] <http://www.ite.educacion.es/formacion/materiales/72/cd/curso/unidad3/u3.II.2.htm>. 978-84-369-4471-6..
7. **Santos, Sergio Gutiérrez.** *Secuenciamiento de actividades educativas orientado a la reutilización y la auto-organización en tutoría inteligente.* Madrid, España : Universidad Carlos III de Madrid. Departamento de Ingeniería Telemática., 2009.
8. **León, Rolando Alfredo Hernández.** *El paradigma cuantitativo de la investigación científica.* Habana : s.n., 2002.
9. **Jameson, Anthony.** *Systems that adapt to their users: an integrative overview.* *Systems that adapt to their users: an integrative overview.* s.l. : International University, 2003.
10. **Cynthia Duk, Ana M. Hernández, Pí a Sius.** *Las Adaptaciones Curriculares una estrategia de individualización de la enseñanza.* 2018.
11. **Definición.** [En línea] 2011. [Citado el: 3 de diciembre de 2015.] www.definicion.mx/secuencia.
12. **Arechabaleta, Marta González.** *Cómo desarrollar un curso de formación on line. Aspectos importantes a tener en cuenta.* 2016.
13. —. *Cómo desarrollar contenidos para la formación basados en objetos de aprendizaje.* Madrid, España : ComuNET, S.A, 2005.
14. **Marcela Prieto Ferraro, Francisco José García Peñalvo.** *Metodología para diseñar la adaptación de la presentación de contenidos en sistemas hipermedia adaptativos basados en estilos de aprendizaje.* Salamanca : Ediciones Universidad de Salamanca, 2005.

15. *Revista de Educación a Distancia*. Ros, Miguel Zapata. 2005, Vol. Secuenciación de contenidos y objetos de aprendizaje.
16. *Revista Mexicana de Investigación*. Educación, Ministerio de. Mexico : s.n., 2003., Vols. Aportes para la secuenciación de contenidos- instancia de elaboración institucional.
17. *Plataformas abiertas de e-learning para el soporte de contenidos educativos abiertos*. Boneu, Josep M. Cataluña : Uniersitat Oberta de Cataluña, 2007, Vol. 4.
18. Morgado, Erla Maria Morales. *Gestiion del conocimiento en sistemas e-learning, basado en objetos de aprendizaje, cualitativa y pedagogicamente definidos*. Unversidad de Salamanca, España : Universidad de Salamanca, 2010. ISBN 978-84-7800-174-3..
19. Ortega, Facundo. *La Universidad. Entre la gestión y el conocimiento*. Cordoba : s.n., 2003.
20. Peñalvo., Francisco José García. *Estado actual de los sistemas e-learning*. Salamanca : Universidad de Salamanca., 2006.
21. Santos, Sergio Gutiérrez. *Secuenciamiento de actividades educativas orientado a la reutilizacion y la auto-organizacion en tutoria inteligente*. Madrid, España : Universidad Carlos III de Madrid. Departamento de Ingeniería Telemática., 2009.
22. *Revista de Educación a Distancia* . Zapata, Miguel. 2003, Vol. Sistemas de gestión del aprendizaje plataformas de teleformación.
23. Britain, Sandy. *A review of learning design: concept, specifications and tools*. 2004.
24. Pecquet., Emmanuel. *Creando y publicando cursos virtuales con Dokeos 1.8*. 2007.
25. Cuello, Rafael Oliver. *La evaluación del aprendizaje y la plataforma docente moodle*. 2016.
26. Quan, Roberto Argueta. *Claroline manual del profesor*. 2009.
27. Maes, Jean-Marie. *Chamilo 2.0: A Second Generation Open Source E-learning and Collaboration Platform*. Belgium : University College Ghent, 2010.
28. Garrido, Antonio. Edmodo. [En línea] 17 de Octubre de 2011. [Citado el: 12 de diciembre de 2015.] <http://edmodo.antoniogarrido.es/introduccion.html>.
29. Valdez, José Luis Cendejas. *Modelo de desarrollo de software integral y colaborativo*. 2014. ISBN-13: 978-84-16036-63-9..
30. Letelier, Patricio. *Metodologías ágiles para el desarrollo de software: extreme programming (XP)*. 2006.
31. IBM. *IBM*. [En línea] 22 de noviembre de 2010. [Citado el: 25 de febrero de 2016.] <https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Ration>

al+Team+Concert+for+Scrum+Projects/page/SCRUM+como+metodolog%C3%ADa.

32. James Rumbaugh, Ivar Jacobson, Grady Booch. *The unified modeling language reference manual. Second edition.* Canada : s.n., 2004. ISBN 0-321-24562-8.
33. Olivia Rodríguez López, Indalecio Mendoza Uribe, José Guadalupe Rosario de la Cruz. *Operación y actualización del sistema de verificación de pronóstico de lluvia máxima.* Mexico : s.n., 2015.
34. Studios, AulaMax. HTML, JAVASCRIPT Y CSS. [En línea] [Citado el: 18 de marzo de 2016.] <http://aulamaxstudio.com/component/content/article/35-latest-news/128-html-javascript-y-css.html>.
35. Mendez, Max. Innovacion y desarrollo web. [En línea] [Citado el: 21 de marzo de 2016.] <https://maxmendez.net/habilidades>.
36. Aranguren, Gilbert. Topicos generales de ingenieria de software. *Topicos generales de ingenieria de software.* [En línea] 4 de julio de 2014. [Citado el: 15 de Noviembre de 2016.] <https://ingsoftwarei2014.wordpress.com/category/framework-de-desarrollo-de-software/>.
37. Fabien Potencier, François Zaninotto. Symfony 2 guia definitiva. *Symfony 2 guia definitiva.* [En línea] 2 de abril de 2016. [Citado el: 21 de noviembre de 2016.] http://librosweb.es/libro/symfony_1_2/capitulo_2/el_patron_mvc.html.
38. Beatriz Gavilán Ruiz, Raúl Alonso Calvo. *Desarrollo de un sistema seguro de administración e integración de servicios y librerías software.* madrid : s.n., 2016.
39. Zambrano, Antonio Nicolas Camacho. *Herramienta para el analisis de requerimientos dentro de la pequeña empresa desarrolladora de software en Bogotá.* 2005.
40. Pasaje, Julio Luis Medina. *Metodologia y herarmientas UML para el modelado y analisis de sistemas de tiempo real orientado a objetos.* 2005.
41. Alegsa, Leandro. Diccionariode Informatica y Tecnologia. *Diccionariode Informatica y Tecnologia.* [En línea] 21 de agosto de 2015. [Citado el: 11 de mayo de 2016.] www.alegsa.com.ar/Dic/servidor_de_base_de_datos.php.
42. Rafael Camps Paré, Luis Alberto Casillas Santillán, Dolors Costal Costa, Marc Gibert Ginestà, Carme Martín Escofet, Oscar Pérez Mora. *Software libre. Bases de datos.* 2005. ISBN: 84-9788-269-5.
43. Cognos. Capacitación tecnológica . [En línea] [Citado el: 28 de marzo de 2016.] <http://www.cognos-capacitacion.com/cursos/Cont/PG-100>.

44. WordPress. fergarciac. *fergarcia*. [En línea] 4 de febrero de 2013. [Citado el: 15 de noviembre de 2016.] <https://fergarcia.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide/>.
45. Oracle. www.netbeans.org. *www.netbeans.org*. [En línea] 10 de abril de 2015. https://netbeans.org/index_es.html.
46. Batista, Deysi Teresa Suris. *Aplicación web para el control del tiempo de máquina en el instituto superior minero metalúrgico*. Cuba : s.n., 2016. ISSN 2306.
47. Pytel, P., y otros. *INGENIERÍA DE REQUISITOS BASADA EN TÉCNICAS DE INGENIERÍA DEL CONOCIMIENTO*. [Documento] 2011. ISBN: 978-950-673-892-1.
48. Alberto Cortez, Ana Garis. *Aplicación de Perfiles UML en la Especificación de Patrones de Comportamiento*. Argentina : Argentine Symposium on Software Engineering, 2012.
49. Carmona, Juan García. *Solid y GRASP. Buenas prácticas hacia el éxito en el desarrollo de software*. 2012.
50. Larman, Craig. *UML y patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado*. s.l. : Pearson Educación, 2003.
51. Mestras, Juan Pavón. Estructura de las Aplicaciones Orientadas a Objetos. *Programación Orientada a Objetos*. [En línea] 2009. [Citado el: 18 de marzo de 2016.] <https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf>.
52. Tema 6 Patrones de diseño. *Ingeniería del Software II*. 2011.
53. James Rumbaugh, Grady Booch, Ivar Jacobson. *El lenguaje Unificado de Modelado. Manual de Referencia*. Madrid : Addison Wesley, 2000.
54. Pruebasdesoftware. *Pruebasdesoftware*. [En línea] 28 de abril de 2016. <http://www.pruebasdesoftware.com/laspruebasdesoftware>.
55. *Modelo formal de pruebas funcionales de software para alcanzar el Nivel de Madurez Integrado 2*. Milton Eduardo Escobar Sánchez, Walter Marcelo Fuertes Díaz. 39, 2015, Vol. 24. ISSN 0121-1129, ISSN 2357-5328.
56. Lemus, Guillermo. *tipos-de-pruebas-de-software*. *tipos-de-pruebas-de-software*. [En línea] 3 de mayo de 2016. <http://es.slideshare.net/GuillermoLemus/tipos-de-pruebas-de-software>.
57. Juan Pavón Mestras. Estructura de las Aplicaciones Orientadas a Objetos. *Programación Orientada a Objetos*. [En línea] 2009. [Citado el: 18 de Marzo de 2017.] <https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf>.
58. Fabien Potencier, François Zaninotto. *Symfony 2 guía definitiva*. *Symfony 2 guía definitiva*. [En línea] 2 de abril de 2016. [Citado el: 21 de noviembre de 2016.] http://librosweb.es/libro/symfony_1_2/capitulo_2/el_patron_mvc.html.

Anexo 1 Descripción de las HU.

Tabla 5 HU Editar recorrido dirigido personalizado.

Número: 2	Nombre del requisito: Editar recorrido dirigido personalizado
Programador: Armando Ávila Riveron	Iteración Asignada: NA
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: NA	Tiempo Real: 2 días
<p>Descripción:</p> <p>1- Objetivo: Permitir editar un RDP</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para editar un RDP hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol docente. - Debe existir en el sistema al menos un RDP. <p>3- Comportamientos válidos y no válidos (flujo central y alternos): Los campos nombre y descripción son obligatorios. Nombre: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 100 caracteres Descripción: campo de texto que permite cualquier carácter</p> <p>4- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> - El sistema debe permitir Editar un RDP, esta acción puede realizarse seleccionando la opción editar en el listado de recorridos dirigidos personalizados. - Cuando el usuario modifica de forma correcta los datos necesarios y selecciona la opción Actualizar, se muestra un mensaje de información de que el RDP fue modificado de forma correcta. - Si los datos están incompletos o incorrectos se señalarán los campos en cuestión dando la posibilidad al usuario de realizar nuevamente la acción en cuestión. - Si selecciona la opción Cancelar regresará a la vista previa. 	

Observaciones: Los RDP son creados para cada estudiante en particular no puede haber dependencia entre ellos.

Prototipo de interfaz:

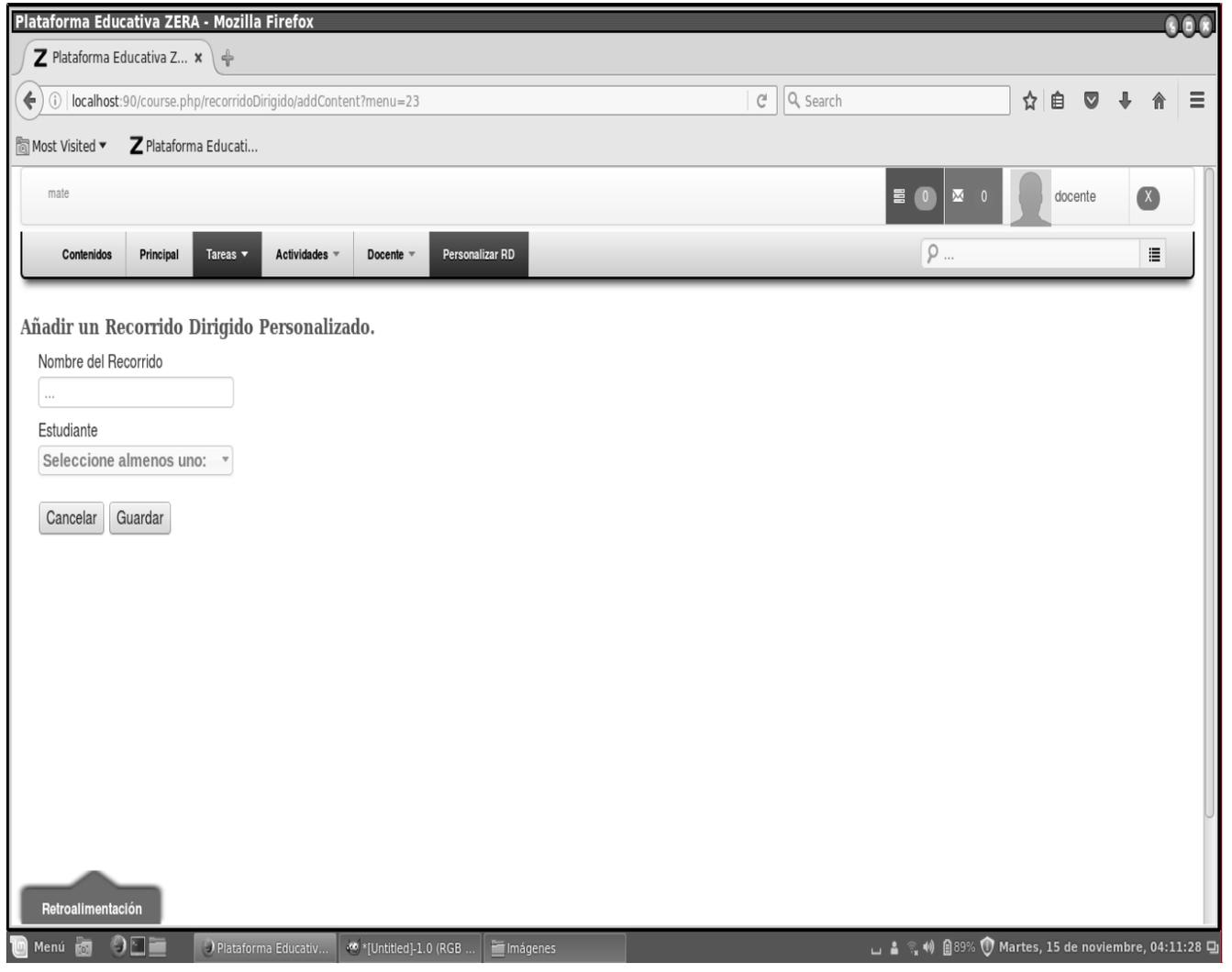


Tabla 6 HU Eliminar RDP.

Número: 3	Nombre del requisito: Eliminar RDP
Pogramador: Armando Ávila Riveron	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: NA	Tiempo Real: 2 días

Descripción:

1- Objetivo:

Permitir eliminar un RDP en el sistema.

2- Acciones para lograr el objetivo (precondiciones y datos):

Para eliminar un RDP hay que:

- Estar autenticado en el sistema con el rol docente.
- Debe existir en el sistema al menos una categoría.

3- Flujo de la acción a realizar:

- El sistema permite eliminar un recorrido dirigido personalizado, para eliminarlo el sistema muestra un listado de los mismos permitiendo marcar un elemento a eliminar, selecciona la opción eliminar y presiona la opción Aplicar.
- En caso de querer eliminar un elemento se puede realizar seleccionando la opción eliminar de las opciones que muestra el propio elemento o desde la vista previa del mismo. Para ambos casos el sistema muestra un mensaje de confirmación para Aceptar o no la acción que se está realizando.
- Si selecciona la opción Aceptar se actualiza el listado de categorías y el sistema muestra un mensaje de información.

Observaciones: Los RDP son creados para cada estudiante en particular no puede haber dependencia entre ellos.

Prototipo de interfaz:

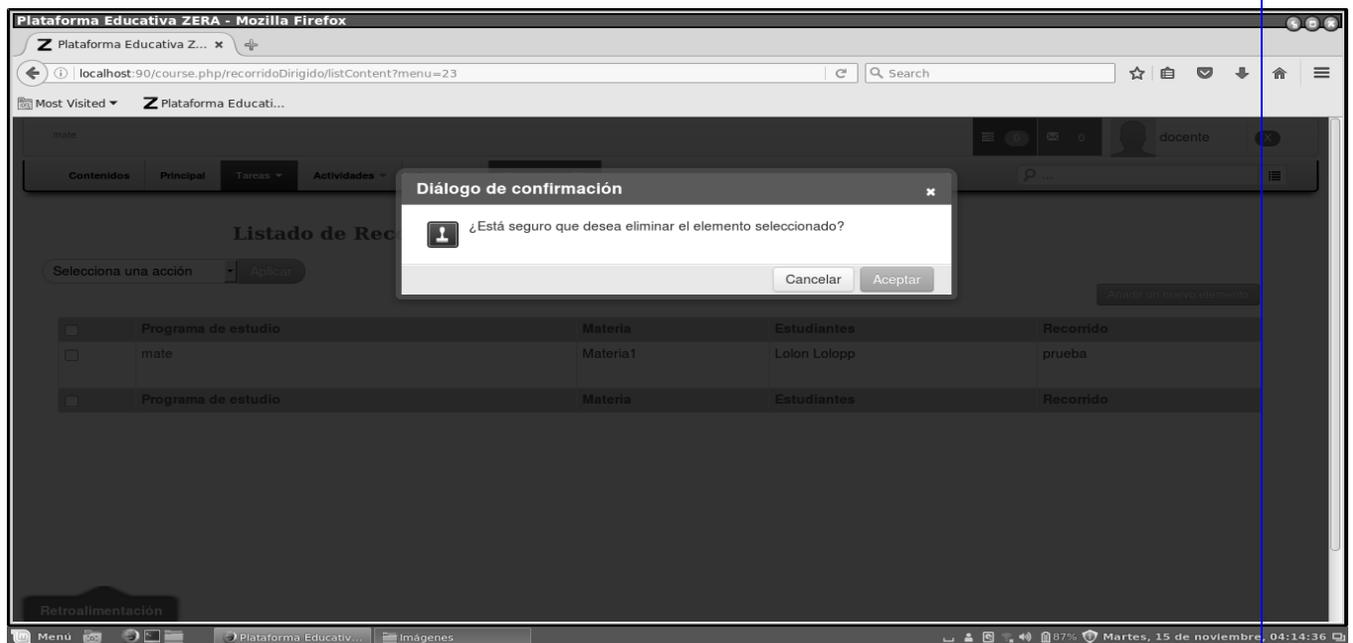


Tabla 7 HU Listar RDP.

Número: 4	Nombre del requisito: Listar RDP
Programador: Armando Ávila Riveron	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 1 día
Riesgo en Desarrollo: NA	Tiempo Real: 1 día
<p>Descripción:</p> <p>Objetivo:</p> <p>Permitir listar los datos de los recorridos dirigidos personalizados en el sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos):</p> <p>Para listar categorías hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol docente. - Debe existir en el sistema al menos un recorrido dirigido personalizado. <p>3- Flujo de la acción a realizar:</p> <p>Cuando el usuario selecciona la opción Recorridos Dirigidos Personalizados aparecen todas los RDP registrados en el sistema. Además, el usuario tiene la posibilidad de ver detalles de un RDP, modificarla y eliminarla.</p> <p>Observaciones: Los RDP son creados para cada estudiante en particular no puede haber dependencia entre ellos.</p>	
<p>Prototipo de interfaz:</p> 	

Tabla 8 HU Notificar al estudiante.

Número: 1	Nombre del requisito: Notificar al estudiante
Programador: Armando Ávila Riveron	Iteración Asignada: 1era
Prioridad: Alta	Tiempo Estimado: 3 días
	Tiempo Real: 2 días
<p>Descripción:</p> <p>1- Objetivo: Permitir notificar al estudiante una vez el docente haya personalizado su curso.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para notificar al estudiante hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol docente. - Tener asignado un grupo al cual impartir clases. <p>3- Comportamientos válidos y no válidos (flujo central y alternos): Los campos nombre y descripción son obligatorios. Nombre: campo de texto que admite caracteres alfabéticos y tiene un máximo de hasta 100 caracteres Descripción: campo de texto que permite cualquier carácter</p> <p>4- Flujo de la acción a realizar:</p> <ul style="list-style-type: none"> - El sistema debe permitir incluir y/o seleccionar los datos para incluir un nuevo RDP. - Cuando el usuario incluye y/o selecciona correctamente los datos necesarios para incluir o editar un RDP, el docente notificará al estudiante de que su curso fue personalizado mediante el botón 'Guardar'. 	
Observaciones: Los RDP son creados para cada estudiante en particular no puede haber dependencia entre ellos.	
Prototipo de interfaz:	



Tabla 8 HU Realizar RDP.

Tabla 9 HU Seleccionar del índice.

Número: 7	Nombre del requisito: Seleccionar del índice
:	
Programador: Armando Ávila Riveron	Iteración Asignada: 1era

Prioridad: Alta	Tiempo Estimado: 3 días
Riesgo en Desarrollo: NA	Tiempo Real: 3 días
<p>Descripción:</p> <p>1- Objetivo: Permitir seleccionar del índice de una materia para crear recorrido dirigido personalizado en el sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para ver detalles de una categoría hay que: - Estar autenticado en el sistema con el rol docente. - Debe existir en el sistema al menos una materia en el sistema para seleccionar del índice el orden y los elementos que conformaran el RDP.</p> <p>3- Comportamientos válidos y no válidos (flujo central y alternos): Debe haberse creado la materia con sus contenidos asociados con anterioridad.</p> <p>4- Flujo de la acción a realizar: El docente accede al contenido de la materia, este escoge los capítulos, temas y subtemas los cuales formaran parte del RDP.</p> <p>Observaciones Los RDP son creados para cada estudiante en particular no puede haber dependencia entre ellos.</p> <p>Prototipo de interfaz:</p>	

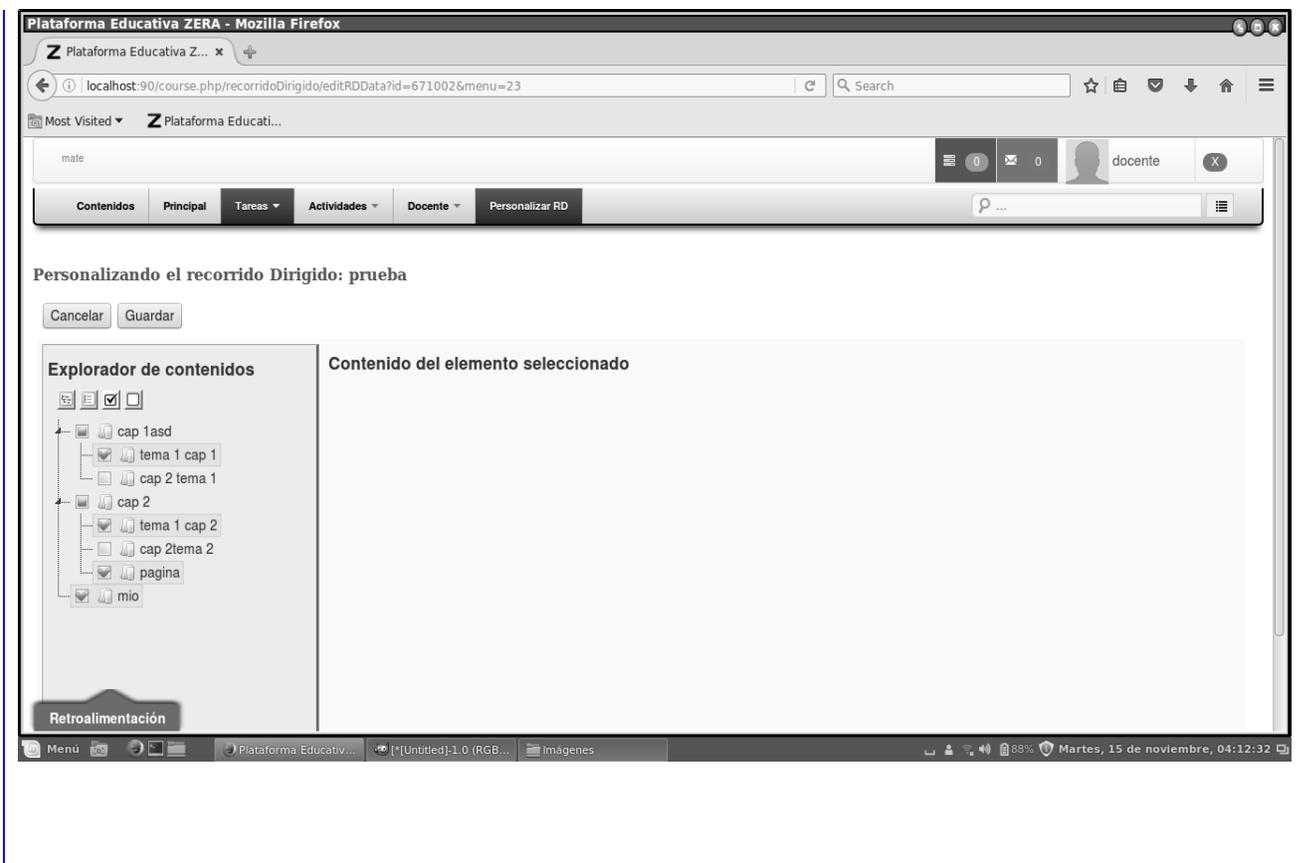


Tabla 10 HU Ver datos del RDP.

Número: 5	Nombre del requisito: Ver datos del RDP.
:	
Programador: Armando Ávila Riveron	Iteración Asignada: 1era
Prioridad: Media	Tiempo Estimado: 3 días
Riesgo en Desarrollo: NA	Tiempo Real: 3 días
<p>Descripción:</p> <p>1- Objetivo: Permitir visualizar los datos de una actividad dentro de un recorrido dirigido personalizado en el sistema.</p> <p>2- Acciones para lograr el objetivo (precondiciones y datos): Para ver detalles de una categoría hay que:</p> <ul style="list-style-type: none"> - Estar autenticado en el sistema con el rol docente. - Debe existir en el sistema al menos un recorrido dirigido personalizado en el sistema. 	

3- Comportamientos válidos y no válidos (flujo central y alternos):

Debe haberse creado el RDP con anterioridad, del mismo se pueden ver las actividades que lo conforman.

4- Flujo de la acción a realizar:

Inicialmente se muestra al usuario un listado con RDP que han sido incluidos en el sistema. Una vez seleccionado un RDP, podrá ver sus actividades seleccionando la opción "Ver".

Observaciones Los RDP son creados para cada estudiante en particular no puede haber dependencia entre ellos.

Prototipo de interfaz:

The screenshot displays the 'Plataforma Educativa ZERA' interface in Mozilla Firefox. The browser address bar shows 'localhost:90/course.php/recorridoDirigido/listContent?menu=23'. The user is logged in as 'mate'. The navigation menu includes 'Contenidos', 'Principal', 'Tareas', 'Actividades', 'Docente', and 'Personalizar RD'. The main content area is titled 'Listado de Recorridos Dirigidos Personalizados.' and features a dropdown menu for 'Selecciona una acción' with an 'Aplicar' button. A table lists the personalized guided tours, and a button 'Añadir un nuevo elemento' is visible. The interface also includes a 'Retroalimentación' button and a system tray at the bottom showing the date and time as 'Martes, 15 de noviembre, 04:09:30'.

<input type="checkbox"/>	Programa de estudio	Materia	Estudiantes	Recorrido
<input type="checkbox"/>	mate	Materia1	Lolon Lolopp	prueba
<input type="checkbox"/>	Programa de estudio	Materia	Estudiantes	Recorrido

Anexo 2 Diagramas de clases.

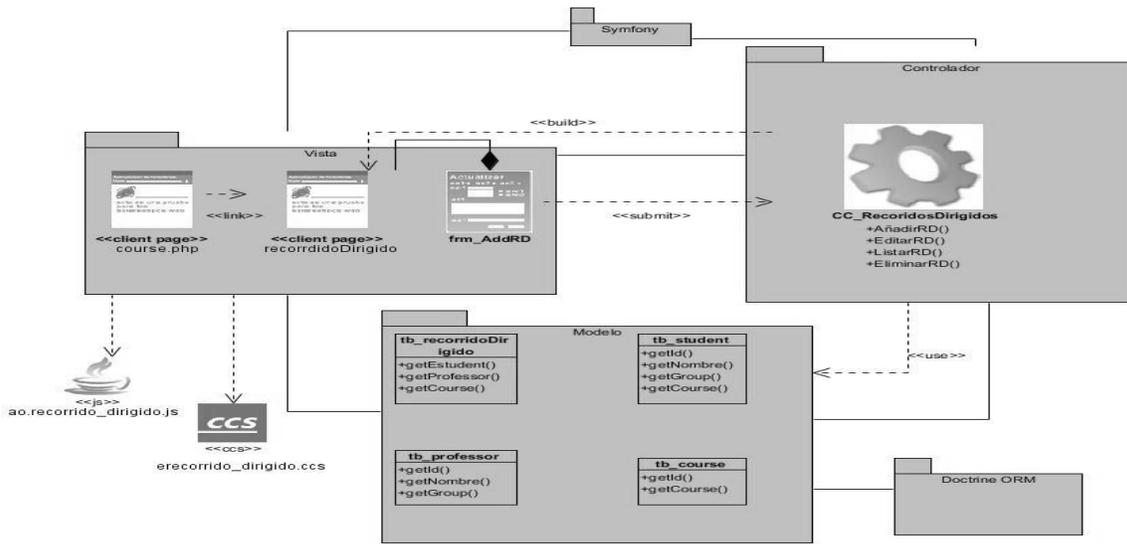


Figura 14 Diagramas de clases del diseño Editar RD.

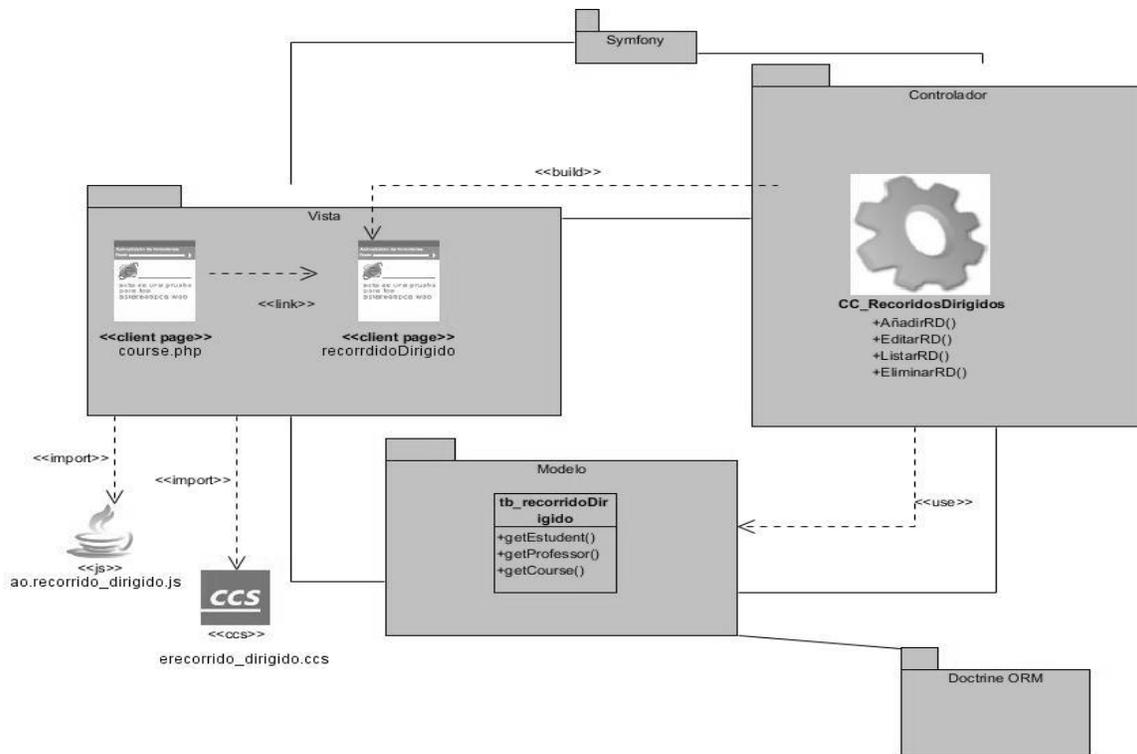


Figura 15 Diagramas de clases del diseño Eliminar RD.

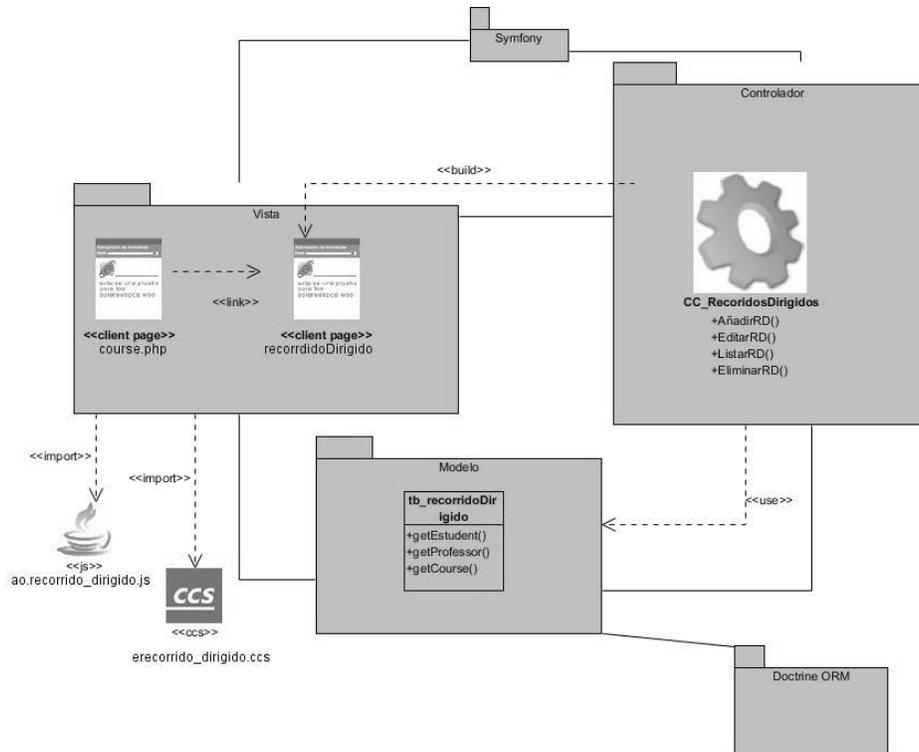


Figura 16 Diagramas de clases del diseño Listar RD.

Anexo 3 Diagramas de Secuencia.

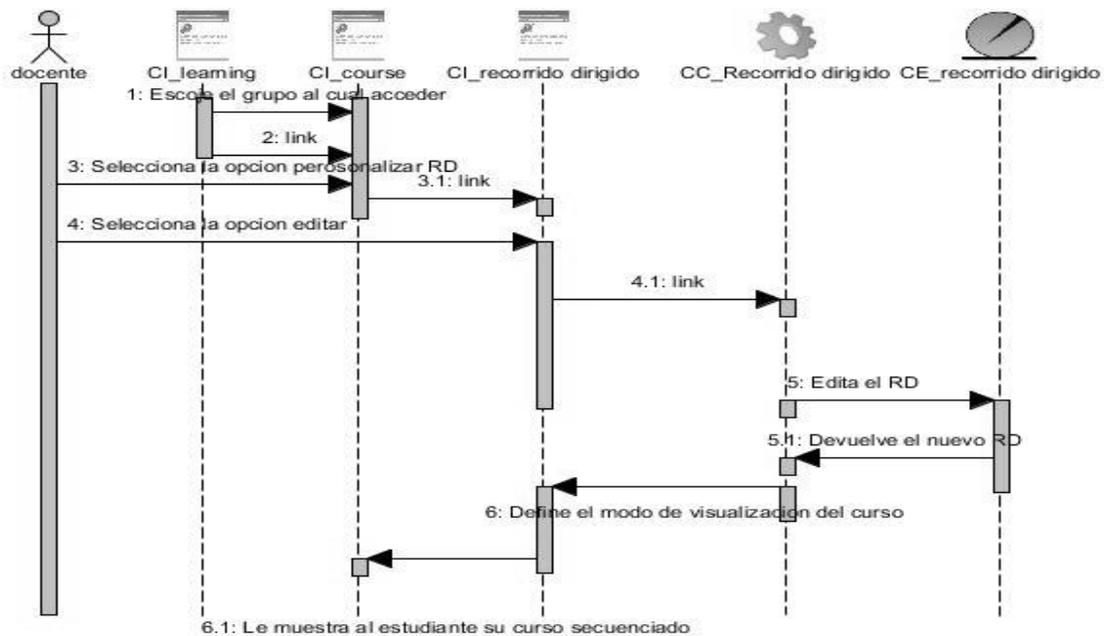


Figura 17 Diagramas de Secuencia Editar RD.

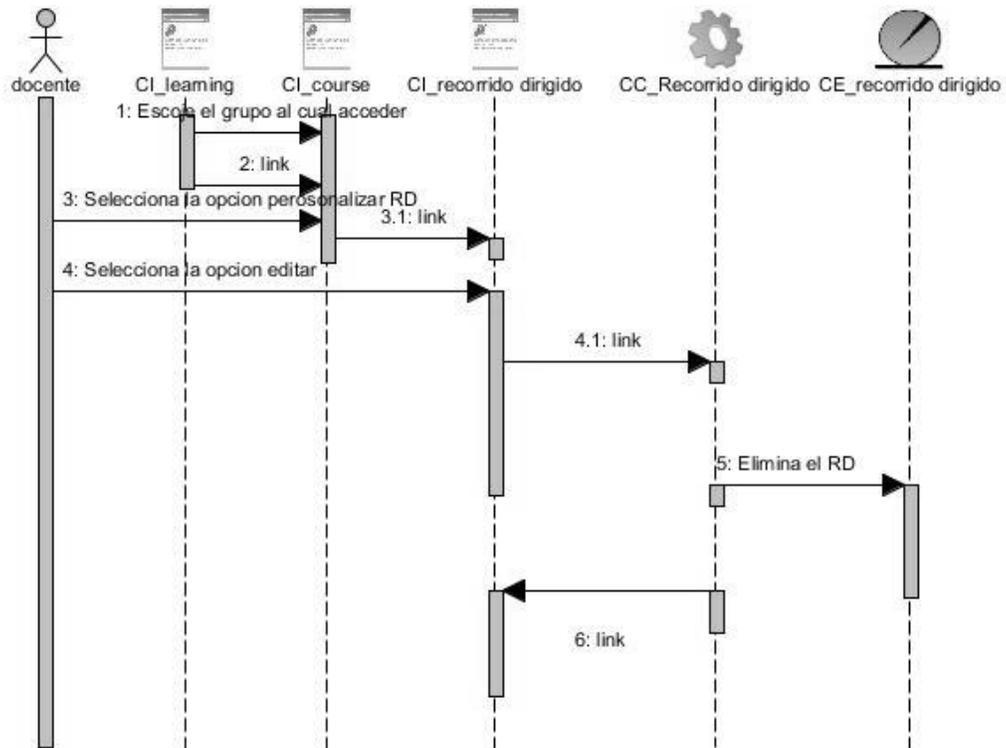


Figura 18 Diagramas de Secuencia Eliminar RD.

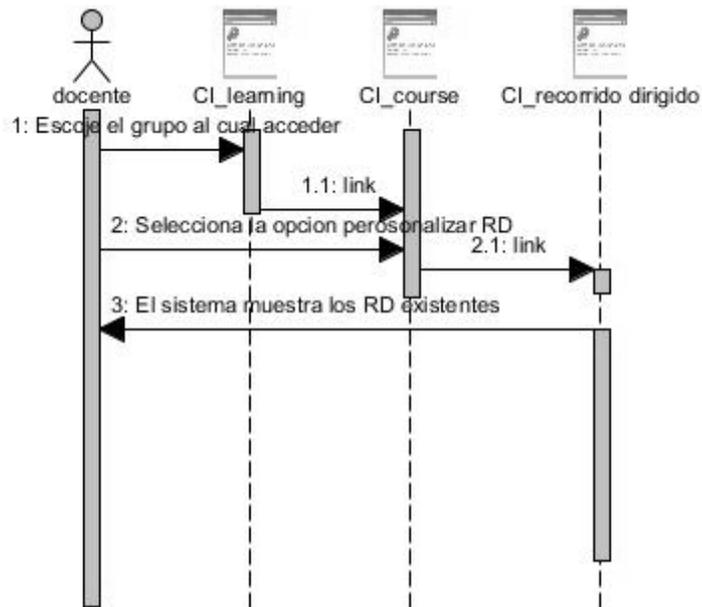


Figura 19 Diagramas de Secuencia Listar RD.

Anexo 4 Descripciones de las tablas de base de datos.

Tabla 11 Guarda los recursos.

tb_sco		
Descripción: Guarda los recursos.		
Atributo	Tipo	Descripción
id	varying (128)	Identificador del recurso
sco_type_id	varying (128)	Almacena el identificador del tipo de recurso que se almacena
replication	varying (5)	Está compuesto por 5 bit, de los cuales se tienen en cuenta el primero (1 -eliminado,0-no eliminado), el segundo (1 -está visible en la biblioteca,0-no lo está) y el tercero (1 -está visible y se puede consultar,0-no está visible)

Tabla 12 Guarda las materias creadas.

tb_matter		
Descripción: Guarda las materias creadas.		
Atributo	Tipo	Descripción
id	varying (128)	Identificador de la materia.
reference	varying (128)	Almacena la cadena que se usará como referencia para la materia.
name	varying (128)	Almacena el nombre de la materia.

Tabla 13 Guarda los programas de estudio creados.

tb_study_program		
Descripción: Guarda los programas de estudio creados.		
Atributo	Tipo	Descripción
id	varying (128)	Identificador del programa de estudio.

matter_id	varying (128)	Identificador de la materia a la que pertenece el programa de estudio.
name	Varying (250)	Almacena el nombre del programa de estudio.

Tabla 14 Guarda los datos generales de los recursos guardados en tb_sco.

tb_general		
Descripción: Guarda los datos generales de los recursos guardados en tb_sco.		
Atributo	Tipo	Descripción
id	varying (128)	Identificador del recurso.
title	varying (500)	Almacena el título del recurso.
description	text	Almacena la descripción del recurso

Tabla 15 Guarda los datos generales de los recorridos dirigidos guardados.

tb_recorrido_dirigido		
Descripción: Guarda los datos generales de los recorridos dirigidos guardados.		
Atributo	Tipo	Descripción
id	varying (128)	Identificador del recorrido dirigido.
title	varying (500)	Almacena el título del recurso.
description	text	Almacena la descripción del recurso

Tabla 16 Guarda la relación entre estudiante, profesor grupo y programa de estudio.

r_student_group_professor_matter		
Descripción: Guarda la relación entre estudiante, profesor grupo y programa de estudio.		
Atributo	Tipo	Descripción
id	varying (128)	Identificador de la relación de la relación.
student_group_id	varying (128)	Almacena el identificador del grupo del estudiante.

professor_study_p rogram_id	varying (128)	Almacena el identificador del programa de estudio del profesor.
--------------------------------	---------------	---

Tabla 17 Guarda la relación entre la tabla tb_mater y los pares padre-hijo de tb_sco.

r_matter_sco		
Descripción: Guarda la relación entre la tabla tb_mater y los pares padre-hijo de tb_sco.		
Atributo	Tipo	Descripción
id	varying (128)	Identificador de la relación de la relación.
matter_id	varying (128)	Almacena el identificador de la materia a la que pertenece la relación.
sco_child_id	varying (128)	Almacena el identificador del elemento hijo en la relación.
sco_parent_id	varying (128)	Almacena el identificador del elemento padre en la relación.
order_sco	bigint	Almacena el orden que tendrán los elementos hijos.

Tabla 18 Guarda la relación entre la tabla tb_study_program y la tabla tb_professor.

r_professor_study_program		
Descripción: Guarda la relación entre la tabla tb_study_program y la tabla tb_professor.		
Atributo	Tipo	Descripción
id	varying (128)	Identificador de la relación de la relación.
study_program_id	varying (128)	Almacena el identificador del programa de estudio al que pertenece la relación.
professor_id	varying (128)	Alacena el identificador del profesor al que pertenece a la relación

