Universidad de las Ciencias Informáticas

Facultad No. 4

"Gestión de tareas y rutas de aprendizaje para la Plataforma educativa ZERA".

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS

Autor: José Ramón Hernández Vega

Tutor: Ing. Yoennis Garrido Vargas

Cotutor: Ing. Nilber Barban Góngora

La Habana, 2011

"Año 53 de la Revolución"

Declaración de autoría

Declaro que soy el único autor del trabajo ***"Gestión de tareas y rutas de aprendizaje para la Plataforma educativa ZERA"*** y autorizo a la Facultad No. 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Autor: _____
José Ramón Hernández Vega

Tutor: _____
Ing. Yoennis Garrido Vargas

Cotutor: _____
Ing. Nilber Barban Góngora

*"Haz primero lo que es **necesario**, luego lo que es **posible***
*y de pronto te encontrarás haciendo lo **imposible**".*

Paulo Coelho

Dedicatoria:

A mi madre; *artífice de mi formación.*

Agradecimientos:

A los que me leerán con el corazón.

Un agradecimiento especial a mis ***tutores****, por enseñarme con sus críticas constructivas e intercambio de experiencias; por la innegable habilidad de profundizar en cada detalle de este proyecto.*

Mi más grande agradecimiento a todos ustedes, ***mis compañeros de la UCI****, por haber estado allí cuando más les necesité, por la unidad, por darme la oportunidad de haberles conocido y de poder navegar juntos por ese mar que ahora nos cubre con su espuma de recuerdos.*

A mi mamá, a papo, mi hermanita, mis tías y mis tíos, y a mi chaparrita y su familia*, por creer en mí; por amarme y enseñarme los más puros valores del ser humano.*

A mi abuelita*, por darme confianza y cariño.*

A mi papá y mis hermanos *que me dieron su apoyo.*

A todos aquellos que de una forma u otra ayudaron en la realización de este trabajo.

Gracias y mi afecto especial a todos, ***José Ramón***

Resumen

Hoy en día, existe una amplia gama de sistemas y plataformas que soportan la gestión de aprendizaje virtual pero en la práctica se integran poco a modelos pedagógicos existentes. El presente trabajo propone la creación de nuevas funcionalidades para la gestión y asignación de tareas a estudiantes para incluir como parte de la Plataforma para la gestión del aprendizaje (ZERA), que está siendo desarrollada por estudiantes y profesores que pertenecen a la Facultad No. 4 de la Universidad de las Ciencias Informáticas, la cual constituye la primera de este tipo, basado en el concepto de hiperentornos de aprendizaje para el modelo pedagógico cubano, pues resume nuevas ideas y conceptos que son el resultado de las investigaciones necesarias relacionadas con el proceso de desarrollo de software. En este contexto se desea obtener un producto capaz de hacer más sencillo, flexible y didáctico el proceso de gestión y asignación de tareas a estudiantes, para el cual se han realizado cuantiosas investigaciones con la intención de subyugar de un modo novedoso todas las limitaciones que impedían a los sistemas y plataformas que soportan la gestión de aprendizaje virtual, integrarse con el modelo pedagógico cubano. Para ello se propone desarrollar componentes, que serán manejados dentro del módulo Docente de la plataforma educativa ZERA, dirigido a facilitar el desempeño profesional del Docente.

Índice

Introducción

El mundo de hoy está caracterizado por la evolución, en sentido general, de todos los medios de comunicación, marcado por un avance sistemático y sostenido en la tecnología, de la cual la informática y el desarrollo de software implícito a ésta, representan un punto fundamental. De ahí que en los últimos años se han integrado el uso de las tecnologías de la información con elementos pedagógicos y didácticos que influyen en la formación, capacitación y enseñanza de los estudiantes, es decir, nace una modalidad de aprendizaje conocida como *e-learning*.

En la práctica, para llevar a cabo un programa de formación basado en *e-learning*, se hace uso de plataformas o sistemas de software que permiten la comunicación e interacción entre profesores, alumnos y contenidos. Para ello, existen herramientas que facilitan impartir y dar seguimiento administrativo a los cursos en línea, así como a la gestión de los contenidos digitales, además de hacerles llegar formas, métodos y prácticas usuales que permitan mejorar el entorno de aprendizaje, y por tanto, contribuir a la adquisición de habilidades necesarias en su formación.

Cuba, reconocida a nivel mundial en el campo de la educación, ha hecho énfasis en el proceso de Informatización de la Sociedad, evidenciando un avance en el desarrollo de software educativo, dirigido a apoyar con materiales didácticos el proceso de enseñanza – aprendizaje, y convertirse en los medios de enseñanza por excelencia.

Con el transcurso de los años, en el país se han desarrollado varias colecciones de software educativos para los diferentes sistemas de enseñanza cubanos (primario, secundario, preuniversitario). Estas se basan en un concepto denominado “hiperentornos de aprendizaje”, o “Plataformas Educativas”, propuesto por especialistas y pedagogos del Ministerio de Educación. Dichos sistemas no son más que una mezcla de varios software educativos, basados en tecnologías multimedia. Fue así que surgió la Colección Futuro, la cual cuenta con diferentes módulos, como por ejemplo: “Temas”, “Ejercicios”, “Biblioteca”, “Juegos”, “Resultados” y “Profesor”. La mayoría de estos, a excepción del módulo “Profesor” están diseñados para los estudiantes.

A pesar de ello, las funcionalidades implementadas hasta el momento en dichas colecciones presentan limitaciones, no cumpliendo así con las necesidades de los docentes. Los profesores no pueden crear tareas docentes (Softareas) que permitan integrar varios aspectos (ejercicios, evidencias y recorridos dirigidos), ni gestionar en tiempo real la creación de las secuencias didácticas o recorridos dirigidos, en las que el docente pueda seleccionar la información del

contenido que realmente le interesa, o simplemente personalizar el aprendizaje a sus estudiantes para atender diferencias individuales. Tampoco es posible asociar recursos interactivos (cuestionarios, imágenes, videos, audios, ejercicios o colecciones de estos) a los Recorridos Dirigidos (RD). Por tal motivo el desarrollo de funcionalidades específicas para el docente se fundamenta en la necesidad de hacer al mismo partícipe del entorno.

Por todo lo anteriormente expuesto **el problema de la investigación queda resumido en la siguiente interrogante.** ¿Cómo facilitar la creación, asignación y visualización de tareas y actividades a los estudiantes mediante la Plataforma ZERA?

El **objetivo general** es desarrollar componentes que permitan gestionar y supervisar las tareas de los estudiantes, de manera que el docente pueda darle seguimiento como apoyo al proceso de enseñanza – aprendizaje.

El **objeto de estudio** de la investigación se centra en Plataformas Digitales de Contenidos Educativos.

El **campo de acción** en el que está enmarcado el trabajo, es el proceso de desarrollo de componentes que faciliten la gestión de tareas.

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

1. Realización de un estudio general del Estado del Arte donde se investigue todas las herramientas posibles a utilizar para el diseño e implementación de los componentes propuestos para la plataforma educativa ZERA.

2. Realización de la fase de requerimientos de la gestión de tareas y las rutas de aprendizaje.

3. Realización de un análisis y diseño donde se gestione la creación, asignación y visualización de tareas y recorridos dirigidos para la plataforma educativa ZERA.

4. Implementación de componentes que permitan la creación, asignación y visualización de tareas y recorridos dirigidos.

5. Elaboración de los casos de pruebas para estos componentes, e integrar la solución obtenida a la Plataforma ZERA.

Con el desarrollo de estos componentes se pretende lograr:

- Permitirle al docente, la posibilidad de gestionar, asignar, así como supervisar las Softareas para los estudiantes de la plataforma ZERA.
- Posibilitarle al docente realizar la gestión de recorridos dirigidos en tiempo real.
- Permitirle al docente pre visualizar tanto las Softareas como los recorridos dirigidos una vez creados estos.
- Que los estudiantes puedan visualizar las Softareas que les han sido asignadas por los profesores como método para adquirir conocimiento.
- Que los estudiantes puedan visualizar los recorridos dirigidos o caminos de aprendizaje realizados por el docente como modo de preparación autónoma.

Por lo anteriormente puntualizado, surge la siguiente **idea a defender**: S*i se desarrollan estos componentes para la plataforma ZERA, los docentes podrán gestionar Recorridos Dirigidos y* Softareas*, así como asignárselas y visualizárselas a los estudiantes, sin importar la poca experiencia acumulada en el tema.*

Los **métodos de investigación** utilizados fueron**:**

De los Métodos Teóricos se escogió:

Histórico - Lógico, y Analítico - Sintético. Estos métodos fueron escogidos para la investigación, por su importancia y necesidad a la hora de desarrollar la misma, por ejemplo el histórico - lógico se utilizó para conocer la evolución y desarrollo de los sistemas de gestión de tareas, en Plataformas Digitales de Contenidos Educativos. Por otra parte el analítico - sintético se usó para el análisis y descomposición de toda la información referente a este tema, para luego filtrar lo más importante y estudiar las diferencias, para darle cumplimiento a las tareas de investigación trazadas.

De los Métodos Empíricos:

Observación. Se utilizó el método observación, para agrupar deficiencias encontradas durante el desarrollo de la investigación de procesos de gestión de contenidos similares, para que estas, no sean un impedimento a la hora del desarrollo de estos componentes.

Para la realización de este trabajo se planearon las siguientes **tareas a cumplir:**

1. Realizar el diseño teórico para la solución propuesta.
2. Realizar un estudio y selección de las metodologías existentes para el desarrollo de aplicaciones que contribuyen al enriquecimiento de proceso de enseñanza - aprendizaje.
3. Analizar los procesos de gestión de tareas, en sistemas informáticos con características similares.
4. Definir la metodología de desarrollo de software y el lenguaje de modelado a utilizar.
5. Realizar el análisis de la solución.
6. Especificar los Casos de Uso.
7. Realizar diagramas de clases del análisis.
8. Realizar el modelo de Casos de Uso que da cumplimiento a los requisitos funcionales y no funcionales de los componentes a desarrollar en la plataforma ZERA.
9. Realizar diagramas de clases de diseño para cada Caso de Uso.
10. Definir las herramientas y tecnologías a utilizar para la construcción de la solución.
11. Investigar sobre el framework de desarrollo Symfony.
12. Investigar sobre el lenguaje de programación a utilizar.
13. Realizar el Modelo de Datos correspondiente a los componentes propuestos.
14. Implementar las clases definidas en el modelo de clases del diseño.
15. Implementar los componentes necesarios para dar solución al diagrama de Clases de Diseño propuestos.
16. Integrar la solución obtenida a la Plataforma ZERA.
17. Implementar la gestión y visualización de las SoftTareas.
18. Implementar la gestión de las rutas de aprendizaje o recorridos dirigidos.
19. Realizar pruebas unitarias al código.

Estructura Capitular

Capítulo 1: Fundamentación teórica.

En este capítulo se especifica la fundamentación teórica que soporta la investigación, así como las posibles herramientas a utilizar tanto para el diseño, como para la implementación, y los argumentos de por qué estas son las que hacen falta para el desarrollo de los componentes sugeridos.

Capítulo 2: Características del sistema.

Se identifica las principales características que exige el sistema, y se hace un levantamiento de requisitos de los componentes para la gestión de tareas en la plataforma ZERA, seguido de un paso esencial, que es la realización de diagrama de Casos de Uso del sistema y la descripción de los mismos.

Capítulo 3: Análisis y diseño de los componentes para la gestión de tareas en la plataforma ZERA.

Se realiza los diagramas de clases del Análisis y de las clases Web, de manera que reflejen una vía factible para el diseño del sistema y su funcionamiento.

Capítulo 4: Implementación y prueba de los componentes para la gestión de tareas en la plataforma ZERA.

Se describe como está implementado el sistema, a través de los diagramas de componentes y el diagrama de despliegue. Se exponen y detallan los diferentes aspectos que se tuvieron en cuenta durante su desarrollo.

Capítulo I. Fundamentación teórica.

Introducción

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 de adoptar ciertas habilidades de posibles herramientas a utilizar tanto para el diseño, como para la implementación, y los argumentos de por qué la selección de las mismas para darle cumplimiento a las tareas especificadas que llevaran a cabo todo el desarrollo de los componentes.

1.1. Conceptos asociados al dominio de problemas.

1.1.1. E-learning.

Según Francisco José García Peñalvo, en su artículo realizado en el año 2006, titulado ***"Estado actual de los sistemas e-learning"***, define a 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".* [1]

1.1.2. Plataforma Educativa.

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)".* [2]

1.1.3. Software Educativo.

Se denomina **software educativo** al destinado a la enseñanza y el aprendizaje autónomo, que, además, permite el desarrollo de ciertas habilidades cognitivas. El enfoque de la instrucción asistida por computadora pretende facilitar la tarea del educador, sustituyéndole parcialmente en

su labor. El software educacional resultante, generalmente presenta una secuencia de lecciones, o módulos de aprendizaje. También generalmente incluye métodos de evaluación automática, utilizando preguntas cerradas.

Según Jaime Sánchez Ilabaca, en su libro publicado en el año 1999, titulado: ***"Construyendo y aprendiendo con el computador",*** define el software educativo como: *"Cualquier programa computacional cuyas características estructurales y funcionales, sirvan de apoyo al proceso de enseñar, aprender y administrar…"* y luego continúa, *"… software educativo es aquel material de aprendizaje especialmente diseñado para ser utilizado con una computadora en los procesos de enseñar y aprender"*. [3]

1.1.4. Discente.

Discente. *Del latín discens.* Persona que cursa estudios y recibe enseñanzas. (La clase académica se divide en docentes y discentes). Persona que recibe un aprendizaje y unos conocimientos de otra persona. [4]

1.1.5. Componentes.

Un componente de software es un elemento de un sistema que ofrece un servicio predefinido, y es capaz de comunicarse con otros componentes. La reusabilidad es una característica importante de los componentes de software, siendo diseñado e implementado de acuerdo a especificaciones de tal forma que pueda ser reutilizado en módulos diferentes de un mismo entorno de desarrollo.

1.1.6. Ejercicio.

Ejercicio: Actividad asignada, con el fin de desarrollar una determinada habilidad. Los ejercicios permiten comprobar el nivel de asimilación de un contenido ya impartido a un discente.

1.2. Software con características similares.

1.2.1. Plataformas Internacionales.

En el mundo de hoy existe una amplia gama de plataformas destinadas a la enseñanza virtual. Estas plataformas incluyen herramientas adaptadas a las necesidades de la institución para la que se desarrollan o adaptan. Algunas de ellas están estandarizadas (aunque permiten la adaptación a situaciones concretas), mientras que otras son completamente personalizadas. Es por ello que esta investigación centra su esfuerzo en dos plataformas que por sus características resulta necesario mencionar.

Conforme se encuentra en los portales oficiales de Claroline y Moodle se define a ambos de la siguiente manera:

1.2.1.1. Claroline.

Claroline es un Sistema de Administración de cursos basados en Web. Este sistema permite a maestros (profesores, conferencistas), crear y administrar sitios web que pueden ser desplegados en un navegador o browser. Claroline ha sido desarrollado por una red internacional de profesores y desarrolladores alrededor del mundo, con programas enteramente reciclados o piezas de código encontrado en la vasta librería de GPL Open Source. Este Software fue comenzado inicialmente en la Universidad de Louvain (Bélgica) y lanzado bajo la licencia Open Source (GPL o fuente abierta). Una comunidad de desarrolladores alrededor del mundo ha contribuido para el desarrollo de esta plataforma. [5]

Claroline ofrece un grupo de ventajas como las mencionadas a continuación:

- Publicar documentos en cualquier formato (Word, PDF, HTML, Video, entre otros).
- Administrar foros de discusión tanto públicos como privados.
- Administrar una lista de enlaces o ligas (links).
- Crear grupos de estudiantes.
- Confeccionar ejercicios.
- Puede estructurar una agenda con tareas y plazos (fechas límites).
- Hacer anuncios (también vía correo electrónico).
- Sus estudiantes pueden enviar documentos (tareas).

Claroline propone en uno de sus acápites la creación de *"camino de aprendizaje"*, los cuales en concepto se asemejan a los *"recorridos dirigidos"* propuestos por la plataforma ZERA, la diferencia radica en que para Claroline un *"camino de aprendizaje",* no es más que un resumen de un tema ya creado de antemano y que solo es editable durante la inserción de contenidos.

1.2.1.2. Moodle.

Moodle es un paquete de software para la creación de cursos y sitios Web basados en Internet. Es un proyecto en desarrollo diseñado para dar soporte a un marco de educación social constructivista. Moodle se distribuye gratuitamente como Software libre (Open Source) (bajo la

Licencia Pública GNU). Puede funcionar en cualquier ordenador que soporte PHP, además de soportar varios tipos de bases de datos (en especial MySQL). El diseño y el desarrollo de Moodle se basan en una determinada filosofía del aprendizaje, una forma de pensar que a menudo se denomina *"pedagogía construccionista social"*. [6]

Moodle ofrece un grupo de ventajas como las mencionadas a continuación:

- Promueve una pedagogía constructivista social (colaboración, actividades, reflexión crítica, etc.).
- Apropiada para el 100% de las clases en línea, así como también para complementar el aprendizaje presencial.
- Tiene una interfaz de navegador de tecnología sencilla, ligera, eficiente y compatible.
- Puede especificarse la fecha final de entrega de una tarea y la calificación máxima que se le podrá asignar.
- Los estudiantes pueden subir sus tareas (en cualquier formato de archivo) al servidor. Se registra la fecha en que se han subido.
- Se permite enviar tareas fuera de tiempo, pero el profesor puede ver claramente el tiempo de retraso.
- Para cada tarea en particular, puede evaluarse a la clase entera (calificaciones y comentarios) en una única página con un único formulario.
- Las observaciones del profesor se adjuntan a la página de la tarea de cada estudiante y se le envía un mensaje de notificación.
- El profesor tiene la posibilidad de permitir el reenvío de una tarea tras su calificación (para volver a calificarla).

Moodle cuenta con un módulo denominado *"Módulo de Tareas"* en el cual el sistema de evaluación de las clases se hace solo con subir al sitio una evidencia, es decir una tarea anteriormente orientada por el profesor. En cambio esta solución se basa en la creación de *"recorridos dirigidos",* los cuales son creados por los profesores a tiempo real y en cualquier momento, integrando en ellos cualquier tipo de recurso a disposición de la plataforma ZERA, y en los cuales es posible atender a cada estudiante según sus características en el aprendizaje.

1.2.2. Plataformas Nacionales.

En Cuba se han desarrollado varias colecciones de software educativos para los diferentes sistemas educacionales cubanos, por lo que resulta importante mencionar los avances logrados por el país en este campo.

1.2.2.1. Colección Futuro.

Como expresara Iraida Calzadilla Rodríguez en un artículo publicado el año 2004 en el sitio web (http://www.cubahora.cu): *"la **Colección Futuro** fue producida por los Centros de Desarrollo de Software de los 16 institutos superiores pedagógicos, y rectorados por el Departamento de Software Educativos (DSE), del Ministerio de Educación de Cuba, el conjunto de programas no solo abarca las asignaturas que se imparten a los estudiantes en las diferentes áreas de conocimiento, sino también materiales dirigidos a la preparación de profesores y directivos. Fue creada con el propósito de apoyar el proceso de transformación educacional que se lleva a cabo en la enseñanza preuniversitaria, técnico-profesional y de adultos. Esta colección cuenta con un grupo de software con características similares, en cuanto a arquitectura se refiere y entre los que se encuentran **Eureka** (Matemática), **El arte de las letras** (Español-Literatura), **Campo y sustancia** (Física), **Un mundo mejor es posible** (Historia Contemporánea, de América, y de Cuba), **Nuestro planeta** (Geografía)".* [7]

Es importante señalar que la idea inicial de los *"Recorridos Dirigidos"* es de esta colección, pero que fueron creados con la herramienta propietaria ToolBox y cuentan con una serie de limitantes que serán resueltas en la solución propuesta, por ejemplo como son que los profesores no pueden crear tareas docentes que permitan integrar varios aspectos (ejercicios, evidencias y recorridos dirigidos), ni gestionar en tiempo real la creación de las secuencias didácticas o recorridos dirigidos, en las que el docente pueda seleccionar la información del contenido que realmente le interesa, o simplemente personalizar el aprendizaje a sus estudiantes para atender diferencias individuales. Tampoco es posible asociar recursos interactivos (cuestionarios, imágenes, videos, audios, ejercicios o colecciones de éstos) a los Recorridos Dirigidos (RD), así como agregar resúmenes o comentarios.

1.3. Metodologías y estándares para el desarrollo del software.

Las metodologías de desarrollo de software engloban procedimientos, técnicas, documentación y herramientas que tienen como base fundamental el fortalecimiento de las actividades de los proyectos y procesos de desarrollo de un software. Actualmente no existe una metodología

universal que le haga frente con éxito a cualquier proyecto de desarrollo de software, ni que se adapte a todo tipo de sistema, debido a que las características y recursos de cada equipo de desarrollo no siempre son las mismas, pero existen propuestas de metodologías más tradiciones que se adaptan rigurosamente a las actividades involucradas en diferentes proyectos. Entre las metodologías más utilizadas se encuentran:

1.3.1. Rational Unified Process (RUP).

Philippe Kruchten en su libro titulado ***"The Rational Unified Process an Introduction"*** publicado en el año 2001 expresa: *"RUP es una metodología robusta que provee un enfoque disciplinado en la asignación de tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción de software de muy alta calidad que satisfaga las necesidades de los usuarios finales, dentro de un calendario y presupuesto predecible. La arquitectura provee la estructura sobre la cual guiar el trabajo en iteraciones, mientras que los casos de uso definen las metas y dirigen el trabajo en cada iteración. Remover cualquiera de estos conceptos reducirá severamente el valor del Proceso Unificado. Es como una mesa de tres patas, sin alguna de ellas, la mesa se caerá".* [8]

Dirigido por casos de uso: Un caso de uso es una pieza en la funcionalidad del sistema que le da al usuario un resultado de valor. Los casos de uso capturan los requerimientos funcionales. Todos los casos de uso juntos constituyen el modelo de casos de uso el cual describe la funcionalidad completa del sistema. Este modelo reemplaza la tradicional especificación funcional del sistema. Una especificación funcional tradicional se concentra en responder la pregunta: ¿Qué se supone que el sistema debe hacer? La estrategia de casos de uso puede ser definida agregando tres palabras al final de la pregunta: ¿por cada usuario? Estas tres palabras tienen una implicación importante, nos fuerzan a pensar en términos del valor a los usuarios y no solamente en términos de las funciones que sería bueno que tuviera. Sin embargo, los casos de uso no son solamente una herramienta para especificar los requerimientos del sistema, también dirigen su diseño, implementación y pruebas, esto es, dirigen el proceso de desarrollo. [8]

Centrado en la arquitectura: El concepto de arquitectura de software involucra los aspectos estáticos y dinámicos más significativos del sistema. La arquitectura surge de las necesidades de la empresa, tal y como las interpretan los usuarios, y tal y como están reflejadas en los casos de uso. Sin embargo, también está influenciada por muchos otros factores, tales como la plataforma de software en la que se ejecutará, la disponibilidad de componentes reutilizables, consideraciones de instalación, sistemas legados, requerimientos no funcionales (ej. desempeño, confiabilidad). La

arquitectura es la vista del diseño completo con las características más importantes, más visibles y dejando los detalles a un lado. Ya que lo importante depende en parte del criterio, el cual a su vez viene con la experiencia, el valor de la arquitectura depende del personal asignado a esta tarea. Sin embargo, el proceso ayuda al arquitecto a enfocarse en las metas correctas, tales como claridad (understandability), flexibilidad en los cambios futuros (resilience) y reuso. [8]

Iterativo e incremental: Los desarrolladores basan su selección de qué van a implementar en una iteración en dos factores. Primero, la iteración trata con un grupo de casos de uso que en conjunto extienden la usabilidad del producto. Segundo, la iteración trata con los riesgos más importantes. Las iteraciones sucesivas construyen los artefactos del desarrollo a partir del estado en el que fueron dejados en la iteración anterior. En cada iteración, los desarrolladores identifican y especifican los casos de uso relevantes, crean el diseño usando la arquitectura como guía, implementan el diseño en componentes y verifican que los componentes satisfacen los casos de uso. Si una iteración cumple sus metas – y usualmente lo hace – el desarrollo continúa con la siguiente iteración. Cuando la iteración no cumple con sus metas, los desarrolladores deben revisar sus decisiones previas y probar un nuevo enfoque. [8]

1.3.2. Extreme Programming (XP).

XP es una metodología ágil que se proyecta a ser un modelo de desarrollo común, sencillo y adaptable a las características cambiantes y exigentes de empresas y clientes. Está basada en cuatro principios: *simplicidad, comunicación, retroalimentación y valor*. Además, orientada por pruebas y refactorización, se diseña e implementan las pruebas antes de programar la funcionalidad, el programador crea sus propios test de unidad. [9]

XP desarrolla cuatro actividades que guiarán el desarrollo:

Codificar: Es necesario codificar y plasmar nuestras ideas a través del código. En programación, el código expresa la interpretación del problema, así se puede utilizar el código para comunicar, para hacer comunes las ideas, y por tanto para aprender y mejorar. [9]

Hacer pruebas: Las características del software que no pueden ser demostradas mediante pruebas simplemente no existen. Las pruebas dan la oportunidad de saber si lo implementado es lo que en realidad se tenía en mente. Las pruebas nos indican que nuestro trabajo funciona, cuando no podemos pensar en ninguna prueba que pudiese originar un fallo en nuestro sistema, entonces habremos acabado por completo. [9]

Escuchar: Se tiene que preguntar si lo obtenido es lo deseado, y se tiene que preguntar quién necesita la información. Por tanto, es necesario escuchar a nuestros clientes, cuáles son los problemas de su negocio, se debe de tener una escucha activa explicando lo que es fácil y difícil de obtener, y la realimentación entre ambos nos ayudan a todos a entender los problemas. [9]

Diseño: El diseño crea una estructura que organiza la lógica del sistema, un buen diseño permite que el sistema crezca con cambios en un solo lugar. Los diseños deben de ser sencillos, si alguna parte del sistema es de desarrollo complejo, lo apropiado es dividirla en varias. Si hay fallos en el diseño o malos diseños, estos deben de ser corregidos cuanto antes. [9]

En resumen, la metodología RUP es más apropiada para proyectos grandes como es el caso de la plataforma ZERA, dado que requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas, partiendo de tres pilares fundamentales, dirigidos por casos de uso, iterativos e incrementales y centrados en la arquitectura. Basados en las ventajas que representa esos tres pilares a la hora de desarrollar un software con robustez y eficiencia se recomienda utilizar RUP como metodología para el desarrollo del software, para la solución propuesta. Además no requiere la presencia permanente del cliente, característica presente en las funcionalidades a desarrollar. También RUP genera una gran cantidad de documentación detalla durante todo el proceso de desarrollo, la cual es necesaria tanto para todo el personal del proyecto como para el cliente.

1.4. Soporte de la modelación.

Según Oscar Casasola Romero en su artículo titulado: “Introducción al Unified Modeling Language (UML)”. Define al Unified Modeling Language (UML) como: “un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Uno de los objetivos principales de la creación de UML es posibilitar el intercambio de modelos entre las distintas herramientas CASE orientadas a objetos del mercado. Para ello era necesario definir una notación y semántica común”. [10]

La principal característica de UML por lo cual se recomienda para la solución propuesta, es un estándar, no existe otra especificación de diseño orientado a objetos, ya que es el resultado de las tres opciones existentes en el mercado. Su utilización es independiente del lenguaje de programación y de las características de los proyectos, pues ha sido diseñado para modelar cualquier tipo de proyectos, tanto informáticos como de arquitectura, o de cualquier otra rama,

proporciona a los desarrolladores un vocabulario que incluye tres categorías: elementos, relaciones y diagramas.

1.5. Selección de herramientas y lenguajes de programación.

1.5.1. HyperText Markup Language (HTML).

Guillermo Prado Ajona explica que: HTML es el lenguaje utilizado para la creación de páginas Web. HTML es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado World Wide Web Consortium, más conocido como W3C. Como se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de forma muy similar en cualquier navegador de cualquier sistema operativo. [11]

W3C define el lenguaje HTML como "un lenguaje reconocido universalmente y que permite publicar información de forma global". Desde su creación, el lenguaje HTML ha pasado de ser un lenguaje utilizado exclusivamente para crear documentos electrónicos a ser un lenguaje que se utiliza en muchas aplicaciones electrónicas como buscadores, tiendas online y banca electrónica. [12]

Por su gran flexibilidad y organización para la creación de páginas Web que sean visibles por diversos navegadores multiplataforma, se decide utilizar este estándar para la creación y visualización de los RD de la Plataforma Educativa ZERA.

1.5.2. Hypertext Preprocessor (PHP).

PHP es un lenguaje interpretado de propósito general ampliamente usado, diseñado especialmente para desarrollo web y que puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. [13]

Entre sus principales características se encuentran:

- Es un lenguaje multiplataforma.
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.

- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Permite aplicar técnicas de programación orientada a objetos.

Las razones de utilizar PHP es, que este lenguaje es del lado del servidor, es robusto y a la vez sencillo. Una de sus grandes cualidades es su versatilidad al momento de escribir código, su sencillez en la sintaxis, e inclusive su seguridad. Además dicho lenguaje satisface una de las principales características de la Plataforma Educativa ZERA (sistema Web en línea con arquitectura cliente – servidor) y permite creación y generación de páginas dinámicas de forma rápida.

PHP es gratuito, y tiene como ventaja, que existe una documentación muy amplia en internet, por lo que se encuentra fácilmente tutoriales y guías gratuitas sobre cualquier duda de cómo utilizarlo, es fácil de aprender, además posee una gran variedad de funciones que pueden ser utilizadas para mejorar el rendimiento de programas. Además, es cada vez más seguro y estable a medida que pasa el tiempo y aumenta su versión. Por lo anteriormente explicado, se escogió PHP para el desarrollo de esta solución. Brinda soporte a diversos gestores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras y se integra con varias librerías externas, lo que extiende aún más su capacidad y flexibilidad de uso.

1.5.3. Framework para el desarrollo.

El término framework, se refiere a una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica compuesta de componentes reutilizables, que pueden personalizase, modificarse o cambiarse, según las necesidades del software a desarrollar.

1.5.3.1. Symfony.

Symfony es un framework muy completo, diseñado para optimizar el desarrollo de las aplicaciones web mediante algunas de sus principales características. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web

compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador, dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que no se debe reinventar la rueda cada vez que se crea una nueva aplicación web. [14]

Características de Symfony que lo hacen factible:

- Fácil de instalar y configurar en sistemas Windows, Mac y Linux.
- Funciona con todas las bases de datos comunes (MySQL, PostgreSQL, SQLite, Oracle, MS SQL Server).
- Compatible solamente con PHP 5 desde hace años, para asegurar el mayor rendimiento y acceso a las características más avanzadas de PHP.
- Basado en la premisa de "convenir en vez de configurar", en la que el desarrollador solo debe configurar aquello que no es convencional.
- Preparado para aplicaciones empresariales, ya que se puede adaptar con facilidad a las políticas y arquitecturas propias de cada empresa u organización.
- Flexible hasta cualquier límite y extensible mediante un completo mecanismo de plugins.
- Publicado bajo licencia MIT de software libre y apoyado por una empresa comprometida con su desarrollo.

Symfony se considera un framework estable, productivo y muy bien documentado y se integra perfectamente con PHP 5, siendo este el lenguaje de programación que se utiliza del lado del servidor para la creación de la solución propuesta. Además los integrantes del equipo de desarrollo poseen conocimientos del mismo y no será necesario un tiempo de capacitación para la implementación pudiendo así aprovechar al máximo las potencialidades del mismo. Recomendándose para el desarrollo del presente trabajo el uso de este framework.

1.5.4. JavaScript.

Como nos describe Javier Eguíluz Pérez en su libro ***"Introducción a JavaScript"****: "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". [15]

Algunas ventajas del JavaScript:

- El lenguaje de scripting es seguro y fiable.
- Los script tienen capacidades limitadas, por razones de seguridad.
- El código JavaScript se ejecuta en el cliente.
- El código del script debe descargarse completamente.

…pero posee un problema importante, y es que el código es visible y puede ser leído por cualquiera, incluso si está protegido con las leyes del copyright y por tanto puede ser reproducido o alterado en cualquier momento". [15]

1.5.5. Cascading Style Sheets (CSS).

En su libro titulado "Cascading Style Sheets" Owen Briggs define CSS como: "un lenguaje de estilo en cascada, usado para definir la presentación de un documento estructurado, escrito en HTML o XML (y por extensión en XHTML). La introducción de CSS ha permitido en muchos casos reemplazar el uso de tablas. Sin embargo CSS todavía no permite la versatilidad que ofrecían las tablas, lograr un diagramado de una página compleja suele ser una tarea difícil en CSS y las diferencias entre navegadores dificultan aún más la tarea". [16]

Se expone, que entre sus principales características se encuentran:

- Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario. Por ejemplo, para ser impresa, mostrada en un dispositivo móvil, o ser "leída" por un sintetizador de voz.

El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño (siempre y cuando no se utilice estilo en línea).

1.5.6. Asynchronous JavaScript and XML (Ajax).

En su artículo: “Asynchronous JavaScript Technology and XML (Ajax) With the Java Platform”, el ingeniero Greg Murray explica lo siguiente:

“*AJAX es una técnica de desarrollo web que genera aplicaciones web interactivas combinando algunas técnicas, entre las que se encuentran*”: [17]

Document Object Model (DOM) para visualizar dinámicamente e interactuar con la información presentada.

- XML, XSLT para intercambiar y manipular datos.
- CSS para definir el aspecto (look and feel) del documento.
- JSON y JSON-RPC pueden ser alternativas a XML/XSLT.
- XMLHttpRequest para recuperar datos de forma asincrónica.

Jesse James Garrett en su artículo publicado en el año 2005**, “Ajax: A New Approach to Web Applications”** expone que*: “AJAX es un patrón de diseño que propone un nuevo modelo de interacción Web combinando las tecnologías anteriores. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones. Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript y DOM”.* [18]

Según el sitio oficial en español la W3C enumera las siguientes características de Ajax. [12]

- Se reduce el tamaño de la información intercambiada.
- Se libera de procesamiento a la parte servidora (se realiza en la parte cliente).
- AJAX actualiza porciones de la página en vez de la página completa.
- Las aplicaciones son más interactivas, responden a las interacciones del usuario más rápidamente, al estilo aplicaciones de escritorio.

- Estas aplicaciones tienen un aspecto (look and feel) muy similar a las aplicaciones de escritorio tradicionales sin depender de plugins o características específicas de los navegadores.

Se selecciona Ajax como lenguaje Javascript del lado del cliente, debido a las ventajas antes mencionadas agregando, se puede integrar con muchos framework que posee una GPL versión 2 y otra MIT, ejemplo jQuery, lo cual facilita su uso en proyectos libres o privativos, según las necesidades del cliente.

1.5.7. PostgreSQL 8.4.

De acuerdo con el portal en español de **PostgreSQL** (http://www.postgresql-es.org), este no es más que es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales. [19]

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. [19]

Sus características técnicas la hacen una de las bases de datos más potentes y robustos del mercado. Su desarrollo comenzó hace más de 15 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. Estas razones antes mencionadas hacen recomendar a PostgreSQL para la solución propuesta.

1.5.8. NetBeans 6.9.

NetBeans 6.9 IDE es un entorno de desarrollo - una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso. [20]

NetBeans ofrece una versión del IDE a medida para el desarrollo de sitios web PHP que comprenden una variedad de secuencias de comandos y lenguajes de marcado. El editor de PHP es dinámicamente integrada con HTML, JavaScript y CSS funciones de edición. [20]

NetBeans se enfoca en el código y acelerar el código de escaneo mediante la exclusión de directorios individuales en las propiedades del proyecto. NetBeans IDE es totalmente compatible con el desarrollo iterativo, por lo que las pruebas proyectos PHP sigue los patrones clásicos familiar para los desarrolladores web. [20]

Por las ventajas que representa el uso del NetBeans IDE para el desarrollo de aplicaciones web, se recomienda el uso de este para la solución propuesta.

1.5.9. Visual Paradigm 6.4.

Es una herramienta diseñada para desarrollar software con programación orientada a objetos. Busca reducir la duración del ciclo de desarrollo, brindando ayuda a arquitectos, analistas, diseñadores y desarrolladores; permite el uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación y tiene capacidades de ingeniería directa e inversa. Usa UML como lenguaje de modelado. Presenta una interfaz de uso intuitiva y con muchas facilidades a la hora de modelar los diagramas que soportan la Ingeniería de Requerimientos. [21]

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

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

Visual Paradigm es una herramienta muy valiosa por todas las características que posee y por esto se ha escogido como herramienta CASE de desarrollo.

1.6. Conclusiones parciales.

Con el análisis de este capítulo se ha adquirido una base elemental para poder desarrollar los componentes enunciados que serán incluidos en la Plataforma Educativa ZERA. Se alcanzó una objetiva visión, sobre cuales herramientas son las más idóneas para el diseño e implementación de la presente investigación. Se ofrece una explicación detallada de los lenguajes de programación a utilizar, así como el gestor de base de datos seleccionado y la metodología de desarrollo para guiar y documentar la realización de la solución propuesta.

Capítulo II. Características del sistema.

Introducción.

En el presente capítulo se abordarán puntos fundamentales para el desarrollo de un producto que sea capaz de satisfacer las necesidades de un cliente, teniendo en cuenta las especificidades y exigencias por parte de estos. Se identificará las principales características que requiere el sistema, además del levantamiento de requisitos de los componentes necesarios para la gestión, asignación y visualización de tareas en la plataforma ZERA.

2.1. Modelo de Dominio.

Cualquier sistema puede ser confuso, pero si la pretensión es comprenderlo se necesita dividirlo en pequeñas partes que pueden ser representadas a través de modelos. El Modelo de Dominio de un software se realiza con el objetivo de representar como está estructurado todo el negocio, cuando apenas solo se tienen pequeños argumentos, es decir, que solo se identificaron pocos conceptos y objetos relacionados con el tema, no logrando definir procesos específicos del sistema.

Con el desarrollo del modelo de dominio, se manifiestan los principales conceptos con los que se trabaja en la plataforma para dar un mejor entendimiento del mismo.

Por situaciones similares en la plataforma educativa ZERA, se propone realizar un modelo de dominio, en vez de un modelo de negocio.

2.1.1. Análisis de los conceptos del dominio.

Es necesario antes de mostrar el diagrama de modelo de dominio, dejar evidenciado algunos conceptos relacionados con el mismo, para expresar de forma clara la idea que se procura especificar con el diagrama en cuestión.

1. **ZERA:** Software educativo destinada a apoyar el proceso de enseñanza – aprendizaje.

2. **Hiperentornos de aprendizaje:** Sistema informático basado en tecnología hipermedia, en la que se mesclan disimiles dispositivos que representan las diversas tipologías de software educativo con el designio de atender las necesidades didácticas.

3. **Módulo.** *Del latín modulus,* es una pieza o un conjunto unitario de piezas que, en una construcción, se repiten para hacerla más sencilla, regular y económica. El módulo, por lo

tanto, forma parte de un sistema y mantiene algún tipo de relación o vínculo con el resto de los componentes. [4]

4. **Usuario del hiperentorno:** Este usuario generalizará un grupo de actividades comunes en diferentes actores, dentro del hiperentorno.

5. **Tarea:** Tarea es un módulo diseñado para los estudiantes, en el cual pueden encontrar visualizadas las Softareas, los recursos y las evidencias asignadas por los docentes.

6. **Docente:** Módulo que tiene como principal objetivo agrupar las acciones específicas que se le asignan al docente dentro del hiperentornos.

7. **Docente (Profesor):** Es el encargado de realizar la asignación de Softareas, Ejercicios, Evidencias y Recorridos Dirigidos a estudiantes y/o grupos de ellos, así como a la gestión de Softareas y Recorridos Dirigidos, además de poder darle solución a cualquiera de estas tareas docentes, a pesar de no estar pensadas para él.

8. **Estudiante o Discente:** Es la persona que recibe y debe resolver cualquier tarea asignada por un profesor o docente, con el objetivo de incrementar sus conocimientos, y al mismo tiempo que el profesor lleve un seguimiento de su aprendizaje, para atender diferencias individuales.

9. **Recurso:** Son elementos utilizados en componentes, lo que facilitan un mejor entendimiento de un contenido asignado por docentes.

10. **Softarea:** Se define como una tarea docente, en la cual, al estudiante durante el proceso de resolución, le resulta necesario interactuar con determinado software educativo, para procesar determinada información.

11. **Evidencia:** Es una tarea docente, que se le asigna a un estudiante, o grupos de ellos, dentro de una Softarea, o sola, en la cual el estudiante ha tenido que investigar, o simplemente resolverla en el instante, pero ha tenido que subir el documento a un lugar en específico dentro de la plataforma, como evidencia de su resolución.

12. **Administración del aprendizaje:** De forma general es un entorno más amplio que el Docente tiene de encargarse de la gestión de todo el contenido pensado para estudiantes.

13. **Gestión de tareas:** La gestión de tareas no es más, que el espacio que tiene el Docente para poder crear una tarea, así como modificar cualquier parte de esta, donde le agregue o le quite lo que desee, además de eliminar de una lista de tareas, una de ellas.

14. **Asignación reactivo:** Es donde el Docente, tiene la posibilidad de asignar una Softarea a un estudiante, o grupos de ellos, conjuntamente con Evidencias y\o Ejercicios.

15. **Recorrido dirigido:** No es más, que un resumen de todo el contenido gestionado en clases apoyado por recursos multimedia, textos y referencias bibliográficas, el cual es asignado a un estudiante o grupos de éstos mostrado de una forma secuencial previamente definido por el docente.

2.1.2. Diagrama del modelo de dominio.

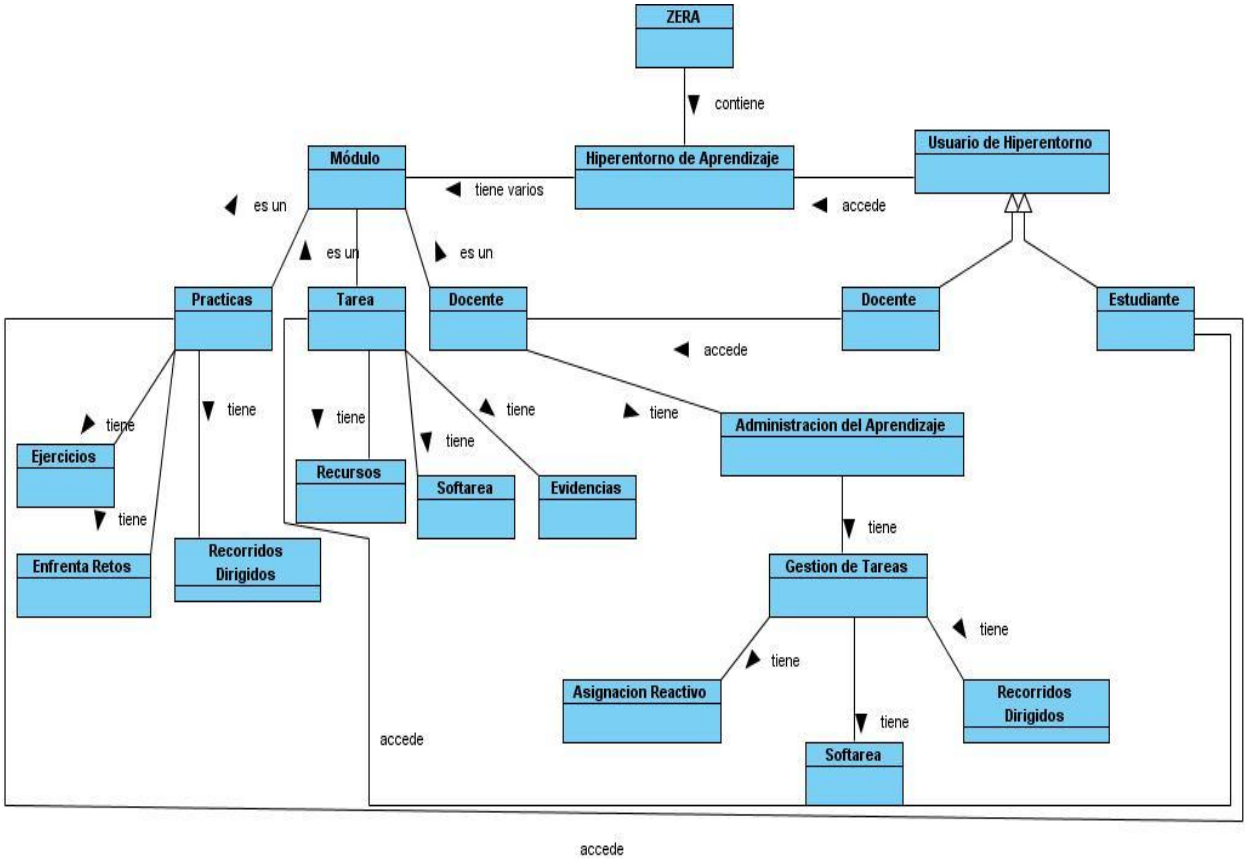

Imagen 1. Diagrama del Modelo de Dominio.

2.2. Descripción del sistema propuesto.

Los componentes inmersos dentro de la Plataforma Educativa ZERA, tienen como principal objetivo permitirles a los profesores de una forma fácil y flexible, la gestión, asignación y visualización de tareas docentes a estudiantes, así como la satisfacción de que los estudiantes resuelvan la misma sin ningún problema al interactuar con dichos componentes.

Entre estos componentes se encuentran los Recorridos Dirigidos (RD) que durante el proceso de su creación, el profesor selecciona del índice, especificando Capítulo, Tema, y/o Subtema de un contenido ya impartido en clases, para conformar un nuevo RD, además de los recursos que desee agregarle o eliminarle a determinadas páginas ya creadas. También se le permite organizar estos según su conveniencia, a partir de modelo de plantilla que escoja.

Por otra parte puede confeccionar una tarea que integre todos estos componentes, obteniendo un componente mucho más completo y con más funcionalidades, a este componente se le denomina Softarea; aquí el docente selecciona un recorrido dirigido, y una evidencia, y/o uno o varios ejercicios, selecciona también a cuales estudiantes y/o grupos de ellos les desea asignar la tarea, determinando una fecha y hora límite de entrega.

En general esta explicación constituye una breve descripción de las funcionalidades pensadas para ser desarrolladas en los componentes propuestos con un único sentido; el de auxiliar a docentes en el proceso de enseñanza-aprendizaje a partir del uso de la plataforma educativa ZERA.

2.3. Requerimientos del software.

Los requerimientos de software son las condiciones o capacidades que debe cumplir el sistema para que se ajuste a las necesidades del cliente. Los requisitos de un software son clasificados en dos grupos, Requisitos funcionales (RF) y Requisitos no funcionales (RNF).

2.3.1. Requerimientos funcionales (RF).

Los requerimientos funcionales describen las capacidades o condiciones que el sistema debe ser capaz de cumplir. En el presente trabajo quedan resumidos los siguientes:

RF1: Gestionar Softareas.

RF1.1: Incluir Softarea. El docente tiene la responsabilidad de incluir una nueva tarea en la lista de tareas ya creadas para un grupo de estudiantes, con el objetivo de ir chequeando

cómo va el aprendizaje de sus estudiantes mediante esta evaluación integral, compuesta por un recorrido dirigido, una evidencia, y/o ejercicios, permitiendo evaluar a los estudiantes en diferentes aspectos.

RF1.2: Ver Softarea. El profesor puede visualizar la Softarea creada y corroborar la creación de la misma mediante su plena visualización.

RF1.3: Modificar Softarea. Una vez que el docente haya consultado la Softarea, y necesite hacer una modificación al contenido añadido, o a la descripción de la misma, tendrá la oportunidad de modificar la Softarea, y guardar los cambios realizados para ser publicada nuevamente.

RF1.4: Eliminar Softarea. El docente puede eliminar una Softarea de la lista creada, con previa selección de la misma.

RF2: Consultar Softareas.

RF2.1: Filtrar Softareas según un criterio especificado. En el módulo Tareas, se encontrará una lista de Softareas a ser consultadas por varias personas con acceso a este módulo. Las mismas se podrán filtrar por Estado y Rango de fecha para agilizar el proceso de búsqueda.

RF2.2: Mostrar listado de Softareas. Una vez escogido los filtros para buscar una o varias tareas en específico, el sistema permitirá mostrar la lista con las coincidencias encontradas según lo especificado.

RF2.3: Ordenar Softareas según un criterio especificado. Automáticamente todas las Softareas se muestran ordenadas por la fecha de actualización.

RF3: Realizar Softareas. El estudiante tendrá un espacio para poder resolver la tarea asignada por su profesor, y que posteriormente será evaluada. Este espacio es en el módulo Tarea, donde podrá encontrar la asignación y se le permitirá seleccionar la opción comenzar y enviarla posteriormente como está establecido.

RF4: Gestionar Recorrido Dirigido.

RF4.1: Incluir Recorrido Dirigido. En el módulo Docente, el profesor contará con varias responsabilidades entre las que se encuentran, incluir un nuevo Recorrido Dirigido: en esta

sección podrá confeccionar un resumen a los estudiantes, donde le priorice el contenido según sus necesidades y solamente válido para su autoestudio. Esta actividad no incluye evaluación.

RF4.2: Ver Recorrido Dirigido. Con la consulta del Recorrido Dirigido creado, el profesor podrá cerciorarse si se equivocó o no a la hora de guardar y publicar el mismo.

RF4.3: Modificar Recorrido Dirigido. Después de una consulta precedente de un Recorrido Dirigido, se percata el profesor que debía de modificar el mismo, entonces el sistema brindará la posibilidad de hacerlo y luego guardar los cambios realizados para publicarlo si ese es su deseo.

RF4.4 Eliminar Recorrido Dirigido. De la lista de Recorridos Dirigidos creada en el módulo Docente, el profesor tiene el derecho de eliminar alguno, si por cualquier motivo así lo requiere.

RF5: Consultar Recorrido Dirigido.

RF5.1: Filtrar Recorridos Dirigidos según criterios especificados. Mediante el módulo Prácticas se podrán visualizar los recorridos dirigidos publicados, donde las personas que tengan acceso podrán consultarlos y buscar uno en concreto, por mediación de filtros, como por ejemplo, filtrar por el Nombre, Rango de fecha y/o Estado.

RF5.2: Mostrar listado de Recorridos Dirigidos. Cuando se tengan escogidos los tipos de filtros por los que se quiere buscar un recorrido dirigido, se mostrarán listados todos, según lo encontrado.

RF5.3: Ordenar Recorridos Dirigidos según un criterio especificado. Seguidamente los recorridos dirigidos se mostrarán ordenados por la fecha de actualización, así se hayan encontrado según el criterio de búsqueda.

2.3.2. Requerimientos no funcionales (RNF).

Los requerimientos no funcionales son propiedades o condiciones que el producto debe tener. Estas características son las que hacen al producto más atrayente, usable, rápido y confiable, dichas propiedades son muy importantes ya que hacen un producto aceptable con respecto a usabilidad, seguridad, y cumplimiento de la funcionalidad solicitada.

Usabilidad:

- El sistema podrá ser usado por cualquier usuario que tenga conocimientos básicos en el campo de la informática.
- La aplicación deberá tener una interfaz y navegación asequible, amena y funcional, tanto para usuarios expertos, como para principiantes.

Seguridad:

- La seguridad de un software incluye la información contenida, que en este caso se le llama privilegiada o confidencial, y para ello existen una serie de estándares, protocolos, métodos, reglas, herramientas y leyes concebidas para minimizar los posibles riesgos a la infraestructura o a la información, que para este software, algunas de estas medidas son:
    - Base de datos de usuarios y calificaciones ofuscada.
    - Base de datos de elementos multimedia y contenidos encriptada.
    - Comunicación web de transferencia de datos encriptados.
    - Acceso mono-usuario a la plataforma, estableciendo sesiones de trabajo.
    - Para sincronización: protocolo de autenticación por IP así como usuario y contraseña de servidor local con central.

Eficiencia:

- El sistema deberá dar respuesta de forma ágil, en un tiempo máximo de 5 segundos a las acciones indicadas por el usuario.
- Cuando un usuario, sin importar su rol, permanece inactivo durante 20 minutos, se cierra la sesión automáticamente.

Soporte:

- La plataforma y sus contenidos deben cumplir con los siguientes puntos:
    - Ser generado en tecnología Web para ser accesible a través de Internet.

- Debe ser capaz de trabajar sobre cualquier navegador, siendo como mínimo que sea compatible con los navegadores, Explorer 7.0 y superior, Mozilla Firefox, Opera, Chrome y Safari.

- La plataforma deberá ser accesible en dispositivos móviles como celulares, BlackBerry, iPod, iPhone, y otros PDA.

- Siempre que se requiere de un plugin o instalación adicional en escritorio, la plataforma avisará del requerimiento, dando la información necesaria para obtener dicho elemento requerido. Los elementos autorizados son: Java, Aplicaciones de escritorio para laboratorios, Flash, Adobe Acrobat Reader.

Restricciones de diseño:

- Lenguaje de programación: PHP 5.2.x
- El marco de trabajo base de desarrollo que se utilizará es: Symfony 1.4.
- Como IDE se empleará NetBeans 6.9.
- El SGDB deberá ser PostgreSQL 8.4.

Interfaz externa:

- La aplicación contará con una interfaz sencilla y amigable.
- El diseño permitirá la distinción visual de los componentes del sistema.
- El sistema proporcionará la información de forma clara y legible, permitiendo a los usuarios una interpretación correcta de la misma.

Interfaces de hardware:

- Las computadoras que brindarán el servicio cliente del sistema no deberán de presentar potencias menores a las brindadas por una Pentium 4, con al menos 512 MB de RAM y 200 MB de espacio en el disco.

Legales, derecho de autor y otros.

- Una vez terminado el sistema debe ser sometido a una evaluación y certificación por parte del cliente del producto.

2.4. Modelo de casos de uso del sistema.

El modelo de casos de uso describe las funcionalidades propuestas para el sistema. Un caso de uso representa una unidad discreta de interacción entre un usuario (humano o máquina) y el sistema. Cada caso de uso tiene una descripción que representa la funcionalidad que se construirá en los componentes propuestos. Un caso de uso puede "incluir" la funcionalidad de otro caso de uso, "extender" a otro caso de uso con su propio comportamiento, y además los actores pueden relacionarse entre ellos mediante la relación de Generalización/Especialización.

2.4.1. Diagrama de actores.

Imagen 2. Diagrama de actores.

2.4.2. Descripción de los actores del sistema.

Tabla 1. Descripción de los actores del sistema

Actor	Descripción
Estudiante	El actor con rol de estudiante, es el encargado de realizar las Softareas y Evidencias asignadas por los docentes y visualizadas en el módulo Tareas, para luego ser evaluada según su desempeño, además estudiarse los recorridos dirigidos publicados de diferentes asignaturas en el módulo de Prácticas.
Docente	El actor con rol de docente, es quien gestiona las Softareas que serán asignadas a estudiantes, además de la creación de evidencias y selección de ejercicios inmersos dentro de esta, también gestiona los recorridos dirigidos,

	para facilitarle un resumen de todo el contenido ya impartido en clases a los estudiantes, mediante el módulo Docente.
(Usuario del Hiperentorno)	Este usuario generalizará un grupo de actividades comunes dentro del hiperentorno, como por ejemplo podrá consultar todos los recorridos dirigidos, además de las Softareas.

2.4.3. Diagrama de casos de uso del sistema.

Para el diagrama de caso de uso de los componentes estudiados, se utilizará la relación de Generalización/Especialización entre actores, ya que los estudiantes y los profesores pueden consultar las Softareas y los Recorridos Dirigidos, y como ambos no pueden inicializar un mismo caso de uso, se creó un nuevo usuario, llamado Usuario del Hiperentorno, del cual heredarán estos usuarios con características similares a la hora de realizar estas acciones.

También se manejará la relación **Extender** para los casos de uso Consultar Softarea, y Realizar Softarea, ya que para que el estudiante pueda realizar alguna Softarea, debe primeramente consultar la lista de Softareas asignadas de tener más de una, y seleccionar la opción *Comenzar* para realizar la misma, si así lo desea, porque puede simplemente consultarlas. Además del CU Gestionar Softarea, extienden los CU Gestionar Recorrido Dirigido y Gestionar Evidencias, porque estos se realizarían en caso de que el docente no le interese seleccionar de la lista ya creada de recorridos y evidencias uno o más de estos, y tenga que acudir a crear nuevos.

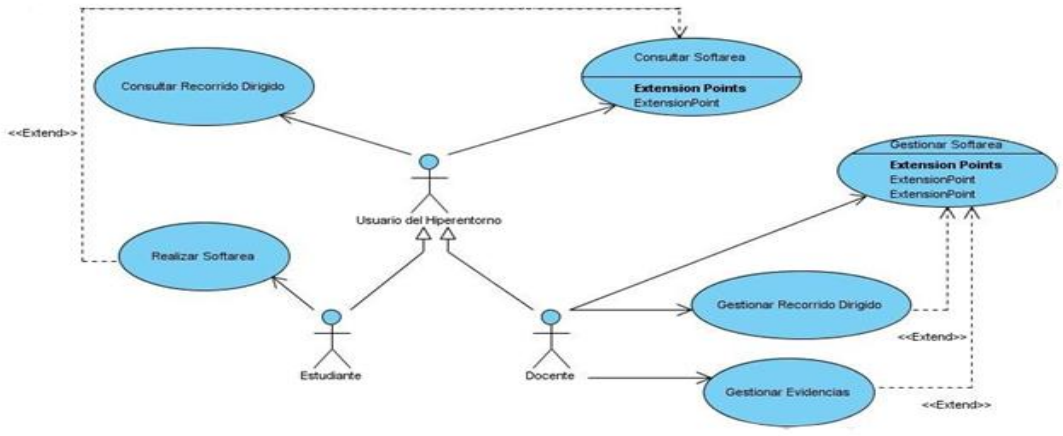

Imagen 3. Diagrama de caso de uso del sistema.

2.4.4. Descripción de los casos de uso del sistema.

Tabla 2. DCU Gestionar Softarea

Caso de Uso	Gestionar Softarea
Actores	Docente (Inicia)
Resumen	El caso de uso se inicia cuando el Docente selecciona la opción que le permite realizar una acción sobre las tareas. El actor puede incluir, ver y modificar las tareas. En caso de que seleccione la opción de incluir tarea, el sistema dará la posibilidad de insertar los datos que se necesitan para llenar esta plantilla. Si el actor elige la opción de ver tarea, el sistema mostrará el contenido de la tarea en cuestión. Si el actor elige la opción de modificar tarea, el sistema mostrará los datos que pueden ser editables dentro de la misma, y una vez realizados los cambios, guardará las modificaciones.
Precondiciones	Debe haberse generado el escritorio de trabajo del usuario autenticado. Para incluir, modificar o eliminar debe haberse ejecutado el caso de uso Consultar Tarea. Para incluir una tarea, el contenido asociado debe estar seleccionado previamente. Para ver una tarea, debe estar seleccionado previamente, y el actor debe tener permiso de acceder al contenido, ya sea porque es el autor, un rol superior en jerarquía, o porque se le ha asignado temporalmente. Para modificar una tarea, debe estar seleccionado previamente y el actor debe ser el responsable temporal del mismo. Para eliminar una tarea, debe estar seleccionado previamente y el actor debe tener el permiso de eliminar el elemento.
Referencia	RF1.

Tabla 3. DCU Consultar Softarea.

Caso de Uso	Consultar Softarea
Actores	Usuario del Hiperentorno (Inicia)

Resumen	El caso de uso se inicia cuando el actor solicita ver un listado de tareas. El sistema permite crear el listado a partir de un grupo de datos predefinidos para la búsqueda de posibles coincidencias. El sistema muestra el listado de tareas coincidentes con los datos de búsqueda y termina el caso de uso.
Precondiciones	Debe haberse generado el escritorio de trabajo del usuario autenticado. Debe estar seleccionado el menú principal la opción Softarea y dentro de esta la opción de Consultar Tarea.
Referencia	RF2.

Tabla 4. DCU Realizar Softarea.

Caso de Uso	**Realizar Softarea**
Actores	Estudiante (Inicia)
Resumen	El caso de uso inicia cuando el estudiante selecciona la opción de consultar tareas, el sistema muestra las tareas asignadas y brinda la opción de seleccionar y comenzar una de las tareas asignadas. El sistema muestra la tarea seleccionada y brinda las posibilidades de ver las instrucciones, terminar la tarea, enviarla y cerrar la pantalla. El caso de uso termina.
Precondiciones	Ha sido generado el escritorio de trabajo del usuario. Las tareas han sido creadas y asignadas al usuario. Para comenzar la tarea debe haberse ejecutado el CU Consultar tarea y el actor debe haber seleccionado la opción de comenzar tarea. La fecha de culminación debe ser superior o igual a la fecha actual.
Referencia	RF3.

Tabla 5. DCU Gestionar Recorrido Dirigido.

Caso de Uso	**Gestionar Recorrido Dirigido**

Actores	Docente (Inicia)
Resumen	El caso de uso se inicia cuando el Docente selecciona la opción que le permite realizar una acción sobre el Recorrido Dirigido. El actor puede incluir, ver, modificar y eliminar un Recorrido Dirigido. En caso de que seleccione la opción de incluir Recorrido, el sistema dará la posibilidad de insertar los datos que se necesitan para llenar esta plantilla. Para que el actor pueda elegir una de las opciones Ver, Modificar o Eliminar, debe anteriormente haber seleccionado un recorrido dirigido de la lista de estos. Si el actor elige la opción de ver Recorrido el sistema mostrará el contenido del recorrido en cuestión. Si el actor elige la opción de modificar el recorrido, el sistema mostrará los datos del recorrido para ser cambiados, y una vez realizados los cambios, guardará las modificaciones. El sistema permite ver una lista de recorridos dirigidos, terminando así el caso de uso.
Precondiciones	Debe haberse generado el escritorio de trabajo del usuario autenticado. Para ver un recorrido, debe estar seleccionado previamente, y el actor debe tener permiso de acceder al contenido, ya sea porque es el autor, un rol superior en jerarquía, o porque se le ha asignado temporalmente. Para modificar un recorrido, debe estar seleccionado previamente y el actor debe ser el responsable temporal del mismo. Para eliminar un recorrido, debe estar seleccionado previamente y el actor debe tener el permiso de eliminar el elemento.
Referencia	RF4.

Tabla 6. DCU Consultar Recorrido Dirigido.

Caso de Uso	**Consultar Recorrido Dirigido**
Actores	Usuario del Hiperentorno (Inicia)
Resumen	El caso de uso se inicia cuando el actor solicita ver un listado de Recorridos Dirigidos. El sistema permite crear el listado a partir de un grupo de datos

	predefinidos para la búsqueda de posibles coincidencias. El sistema muestra el listado de Recorridos Dirigidos coincidentes con los datos de búsqueda y termina el caso de uso.
Precondiciones	Debe haberse generado el escritorio de trabajo del usuario autenticado. Se seleccionó la opción recorrido dirigido. El actor debe tener los permisos necesarios para realizar estas acciones.
Referencia	RF5.

2.4.5. Especificaciones de los CU de sistema (Gestionar Recorrido Dirigido).

Tabla 7. CU Gestionar Recorrido Dirigido (Flujo básico de Incluir Recorrido Dirigido).

Acciones del actor	**Respuesta del sistema**
1. El caso de uso se inicia cuando el actor selecciona la opción de realizar una acción sobre el Recorrido Dirigido.	
	2. Brinda la posibilidad de realizar las acciones: • Incluir una nueva entidad. • Ver los datos de la entidad. Ver Sección 1: “Ver datos del Recorrido Dirigido”. • Modificar los datos de la entidad. Ver Sección 2:”Modificar datos del Recorrido Dirigido”. • Eliminar la entidad. Ver Sección 3: “Eliminar Recorrido Dirigido”.
3. Selecciona la opción de incluir una nueva entidad.	
	4. Permite consultar los datos predeterminados del Índice: • Capítulo. • Tema. • Subtema.
5. Selecciona del índice, el Capítulo, Tema o Subtema.	
	6. Muestra siempre el Tema de lo seleccionado en el índice. Permite ver de lo seleccionado además de obtener una vista previa mediante un ícono

	de: • Ejercicios. • Recursos. Permite añadir mediante un ícono o arrastrando de esa página a la nueva creada: • Recursos. • Ejercicios. • Añadir un ejercicio en una sola página. • Una página. • Todas las páginas asociadas al Capítulo, Tema o Subtema según lo seleccionado. Y permite además mediante un ícono: • Eliminar Recurso de la selección. • Eliminar Ejercicio de la selección. • Eliminar un Texto de la selección. Permite: • Escoger Plantilla. • Crear una página. • Guardar página. • Eliminar página. • Guardar RD. • Eliminar RD. • Cancelar.
7. Selecciona la opción de añadir una página.	
	8. Se añade la página.
	9. Regresa al paso 5 del Flujo Básico.
*. Selecciona la opción Guardar Página.	
	*. Brinda la posibilidad de introducir: • Nombre de la página. • Intención didáctica. Y permite: • Aceptar. • Cancelar.
*. Introduce el Nombre de la Página, la Intención didáctica y selecciona la opción Aceptar.	
	**. Valida los datos.*
	**. Guarda la página.*
	*. Regresa al paso 5 del Flujo Básico.
10. Selecciona la opción de Guardar RD.	
	11. Brinda la posibilidad de introducir: • Nombre del RD. • Descripción del RD. Permite seleccionar si desea:

	• Publicar RD. Permite además: • Aceptar. • Cancelar.
12. Introdujo el Nombre del RD, la descripción del mismo, seleccionó la opción de activar el RD, y la opción Aceptar.	
	13. Valida los datos.
	14. Guarda el RD.
	15. Se activa el RD a los estudiantes.
	16. El caso de uso termina.

Tabla 8. CU Gestionar Recorrido Dirigido (Flujo alterno de Incluir Recorrido Dirigido).

5.1 Selecciona la opción que le permite ver los Ejercicios.	
Acciones del actor	**Respuesta del sistema**
	5.1.1. Muestra una lista de ejercicios.
	5.1.2. Regresa al paso 5 del Flujo Básico.
5.2 Selecciona la opción que le permite ver los Recursos.	
Acciones del actor	**Respuesta del sistema**
	5.2.1. Muestra una lista de recursos.
	5.2.2. Regresa al paso 5 del Flujo Básico.

Tabla 9. CU Gestionar Recorrido Dirigido (Flujo básico de Ver Recorrido Dirigido).

Acciones del actor	**Respuesta del sistema**
17. El actor selecciona la opción de Ver.	
	18. Muestra: • Siempre el contenido de la 1ra página correspondiente al RD existente. • Una lista de todas las páginas del RD. • Intención Didáctica, de cada página. Y permite: • Ir a la página anterior. • Ir a la página siguiente. • Salir de la vista actual.
19. Selecciona la opción de salir de la vista actual.	
	20. Muestra la vista anterior.
	21. El caso de uso termina.

Tabla 10. CU Gestionar Recorrido Dirigido (Flujo alterno de Ver Recorrido Dirigido).

1. a Selecciona ir a la página anterior.	
Acciones del actor	**Respuesta del sistema**
	1. a.1 Si se encuentra después de la 1ra página, regresa a la página anterior.
	1. a. 2 Si se encuentra en la 1ra página, no hace nada.
1. b Selecciona ir a la página siguiente.	
Acciones del actor	**Acciones del actor**
	1. b.1 Si se encuentra antes de la última página, puede ir a la página siguiente.
	1. b. 2 Si se encuentra en la última página, no hace nada.

Tabla 11. CU Gestionar Recorrido Dirigido (Flujo básico de Modificar Recorrido Dirigido).

Acciones del actor	**Respuesta del sistema**
1. El actor selecciona la opción de Modificar.	
	2. Muestra un índice con las páginas que componen el RD seleccionado.
3. Selecciona del índice la página a ser editada.	
	4. Muestra los datos de esa página, y brinda la posibilidad de cambiar sus valores ya sea introduciendo nuevos datos o eliminando los existentes. Permite además: • Crear una nueva página. • Eliminar una página. • Guardar los datos. • Salir de la vista actual.
5. Modifica los datos de la página seleccionada y selecciona la opción de Guardar los datos.	
	6. Valida los datos.
	7. Guarda los datos modificados.
	8. Actualiza los datos del RD.

Tabla 12. CU Gestionar Recorrido Dirigido (Flujo alterno de Modificar Recorrido Dirigido).

3. c Selecciona la opción de Crear una nueva página.	
Acciones del actor	**Respuesta del sistema**
	3. c.1 *Ver Sección: "Incluir datos del Recorrido Dirigido."*
	3. c.2 Regresa al paso 3 del flujo básico.
3. d Selecciona la opción de Eliminar una página.	
Acciones del actor	**Respuesta del sistema**
	3. d.1 *Ver Sección 3: "Eliminar datos del*

	Recorrido Dirigido."
	3. d.2 Regresa al paso 3 del flujo básico.
3. e Selecciona la opción de Salir de la vista actual.	
Acciones del actor	**Respuesta del sistema**
	3. e. 1 Sale de la vista actual.
	3. e. 2 El caso de uso termina.

Tabla 13. CU Gestionar Recorrido Dirigido (Flujo básico de Eliminar Recorrido Dirigido).

Acciones del actor	**Respuesta del sistema**
1. El actor selecciona la opción de Eliminar.	
	2. Muestra un mensaje de confirmación. Y permite: • Aceptar • Cancelar la operación.
3. Selecciona la opción Aceptar.	
	4. Oculta el Recorrido Dirigido.
	5. Muestra un mensaje de información.
	6. Regresa a la vista anterior actualizando los datos.
	7. El caso de uso termina.

Tabla 14. CU Gestionar Recorrido Dirigido (Flujo alterno de Eliminar Recorrido Dirigido).

3.a El actor selecciona la opción de Cancelar.	
Acciones del actor	**Respuesta del sistema**
	3. a.1 Regresa a la vista anterior.
	3. a.2 El caso de uso termina.

Para ver el resto de las especificaciones de los CU, *Consultar Anexo I*.

2.5 Conclusiones parciales.

Este capítulo fue primordial en el desarrollo de los componentes planteados, puesto que se identificaron los requisitos funcionales y no funcionales que debe cumplir el sistema en desarrollo para una buena satisfacción de ambas partes, de quien lo hizo, y para quienes está pensado, también se definieron los conceptos fundamentales asociados, los que fueron utilizados para realizar el Modelo de Dominio. Además se definieron los actores del sistema y la descripción de los mismos. Se delimitaron los diagramas de caso de uso del sistema como buena guía para un buen progreso del producto.

Capítulo III. Análisis y diseño de los componentes para la gestión de tareas en la plataforma ZERA.

Introducción.

Este capítulo se considera bastante significativo, pues se modelan los diagramas de clase del Análisis y del Diseño de cada uno de los componentes a desarrollar. Al mismo tiempo se diseña la estructura de la Base de Datos a manejar, tomando como base las clases persistentes, lo que garantiza una adecuada visión del diagrama de clases del diseño y por tanto de manera general, refleja una vía factible para el diseño del sistema y su funcionamiento.

3.1. Modelo de análisis.

El Modelo de Análisis es manejado principalmente para obtener una primera visión del sistema y comprender en detalle los requisitos funcionales. En esta sección se modelan las clases del análisis, proporcionando una vista interna del sistema, estructurado por clases estereotipadas: [8]

- Entidad: Modela la información que posee larga vida y que es a menudo persistente, en otras palabras representa la información reflejada en el caso de uso. [8]
- Interfaz: Es la encargada de modelar la interacción entre el sistema y sus actores. [8]
- Control: Coordina la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso. [8]

3.1.1. Diagrama de clases del análisis.

Este diagrama de análisis está compuesto por las clases CI: Clase Interfaz, CC: Clase Controladora, CE: Clase Entidad, las que harán de cierta forma perceptible los Casos de Uso de los componentes sugeridos.

Imagen 4. DCA Realizar Softarea.

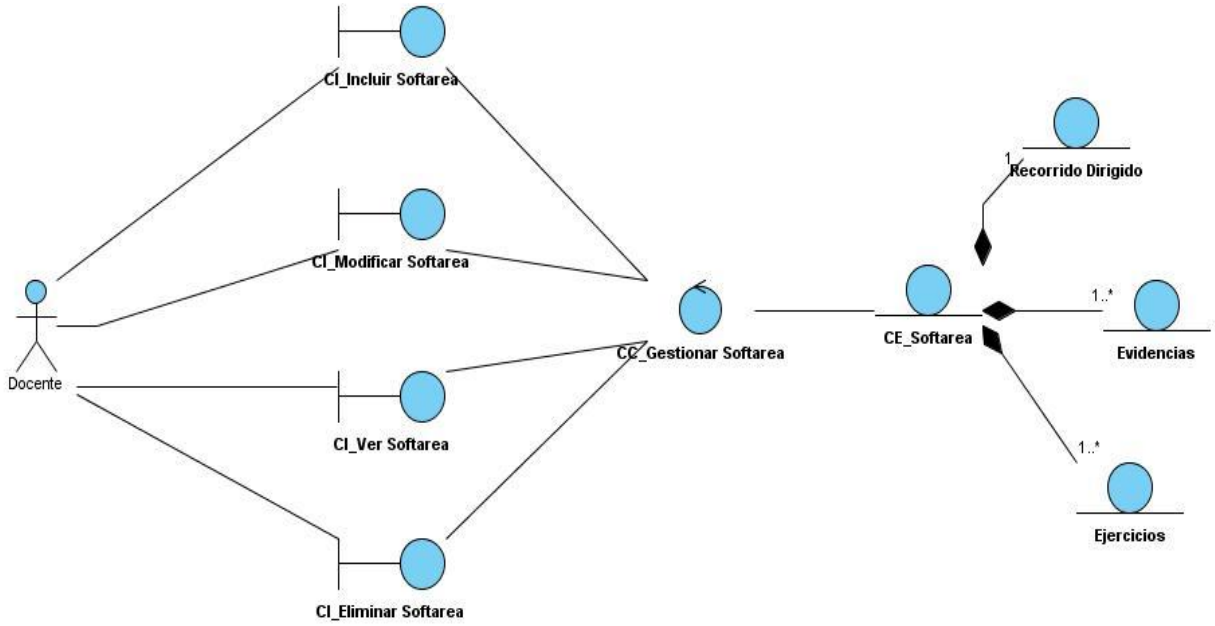

Imagen 5. DCA Gestionar Softarea.

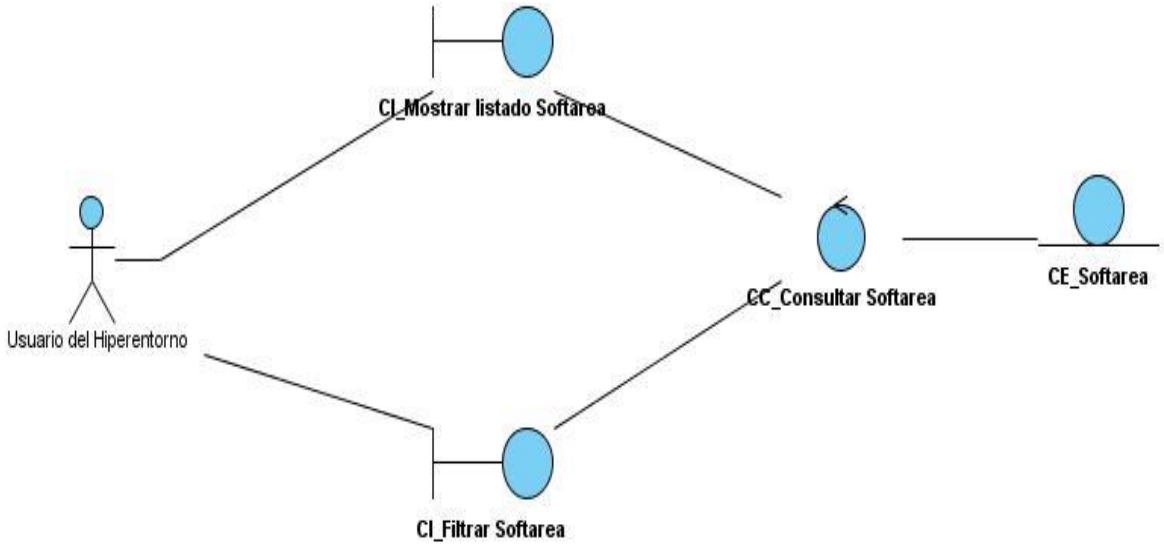

Imagen 6. DCA Consultar Softarea.

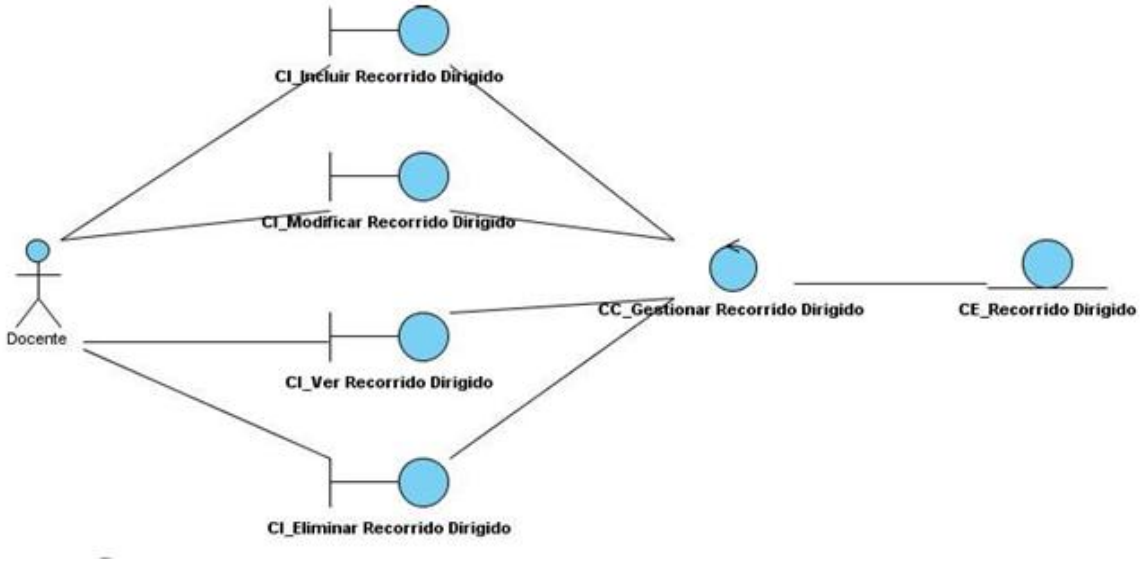

Imagen 7. DCA Gestionar Recorrido Dirigido.

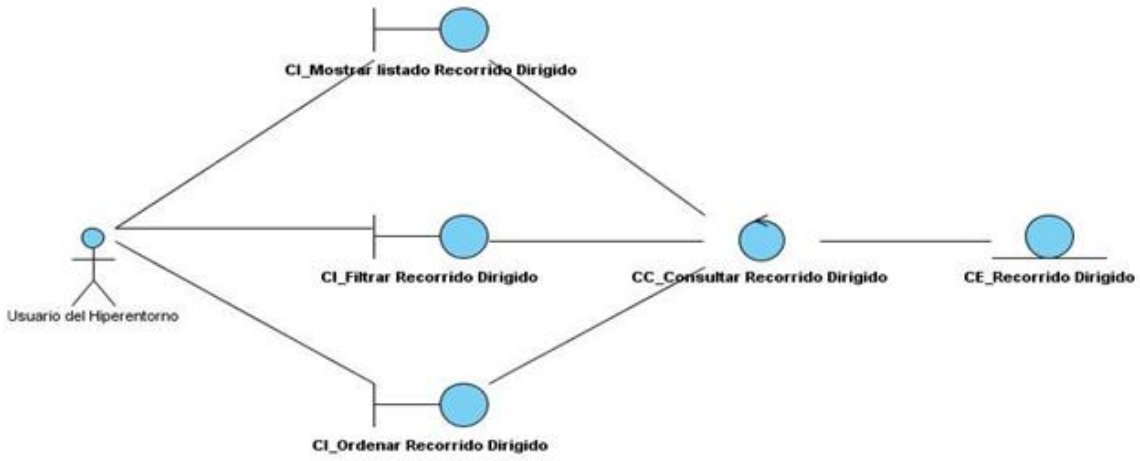

Imagen 8. DCA Consultar Recorrido Dirigido.

3.2. Patrones arquitectónicos utilizados en Symfony.

En términos generales, un patrón es un conjunto de información que proporciona respuesta a un conjunto de problemas similares, es decir, un patrón es una solución a un problema en un contexto, donde:

- Contexto son las situaciones recurrentes a las que es posible aplicar el patrón.

- Problema es el conjunto de metas y restricciones que se dan en ese contexto.
- Solución es el diseño a aplicar para conseguir las metas dentro de las restricciones.

El framework Symfony está basado en varios patrones imprescindibles para la realización de cualquier proyecto, sugiriendo una producción de Software más resistente al cambio, establece problemas Pareja-Solución, ayuda a especificar interfaces, reutiliza el Código, usa documentación estándar, entre otras ventajas.

Existen patrones tanto para la arquitectura, como para el diseño. Aquí se detallarán los más importantes y utilizados en el desarrollo de los componentes que serán usados en la Plataforma educativa ZERA.

3.2.1. Patrón arquitectónico.

El patrón Modelo – Vista – Controlador es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos para alcanzar un mantenimiento más sencillo de las aplicaciones, además obliga a organizar el código de acuerdo a sus convenciones.

- El **modelo** representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La **vista** transforma el modelo en una página web que permite al usuario interactuar con ella.
- El **controlador** se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original.

Este patrón es muy aplicable en Symfony, y por consiguiente se estará utilizando en el desarrollo de este trabajo. [14]

3.3. Modelo de diseño.

El modelo de diseño expande y detalla el modelo de análisis tomando en cuenta todas las implicaciones y restricciones técnicas. 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.

3.3.1. Diagrama de clases del diseño.

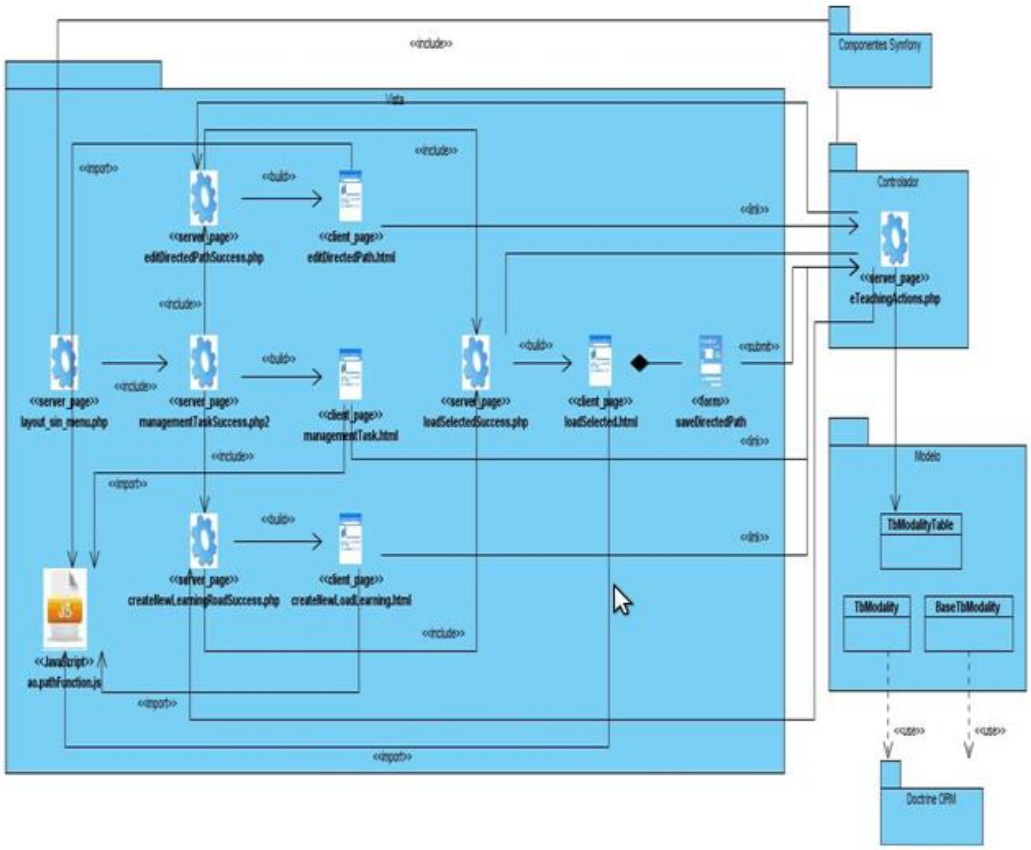

Imagen 9. Diagrama de clases del diseño (Gestionar RD).

Para ver más diagramas de clases del diseño, *consultar Anexo II*.

3.4. Diseño de la base de datos.

En esta sección se presenta el modelo entidad relación referente a la base de datos de las funcionalidades a desarrollar y la descripción de las tablas más importantes que la conforman, para que los desarrolladores tengan una buena concepción de la estructura que posee la BD.

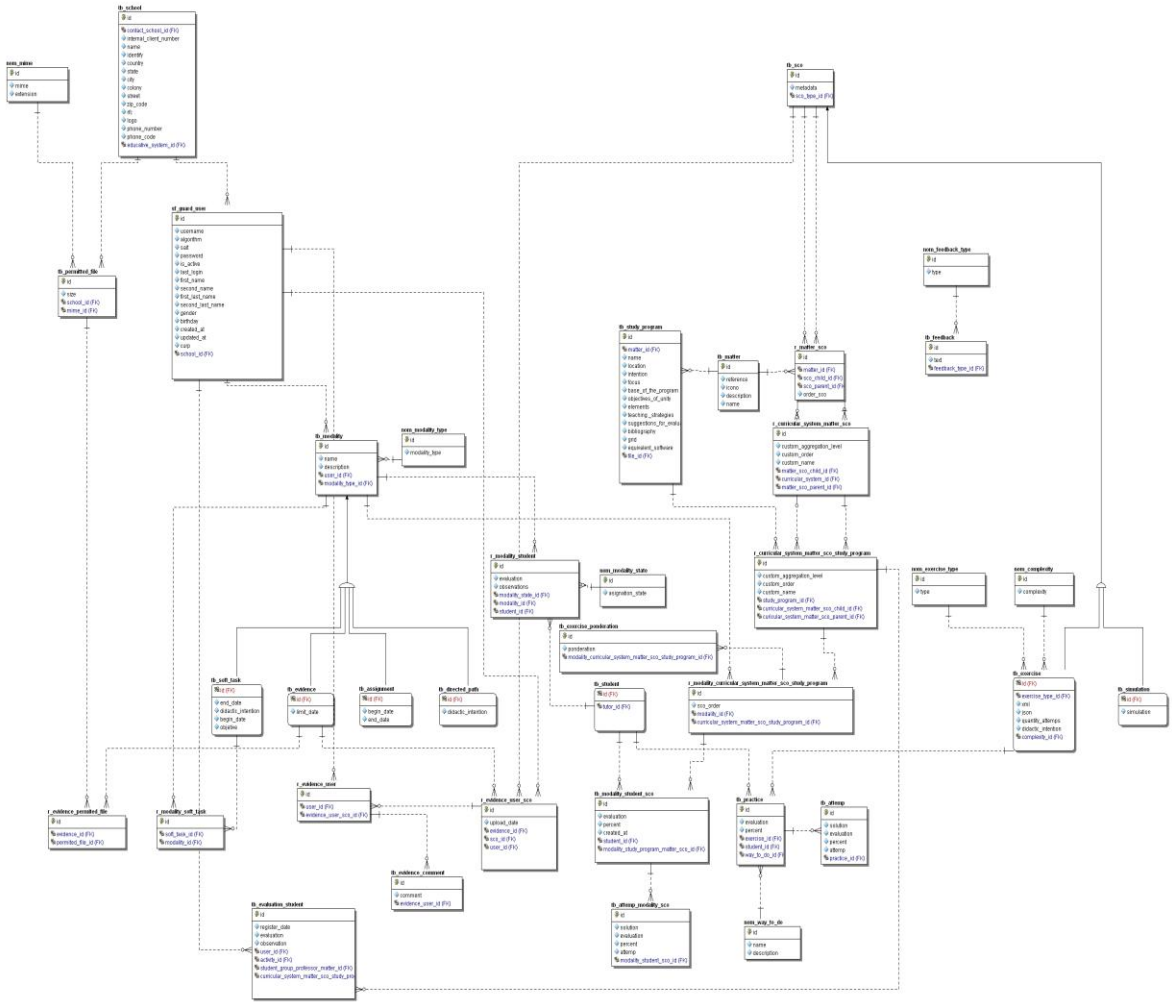

Imagen 10. Diseño de la Base de Datos.

3.4.1. Descripción de las tablas.

Tabla 14. Descripción de la tabla tb_modality

tb_modality.		
Descripción: Tabla donde se almacenan todas las modalidades presentes en la plataforma ZERA.		
Atributo.	**Tipo.**	**Descripción.**
id	Varchar(128)	Etiqueta única que identifica al objeto (**tb_modality**) en la tabla.

name	Varchar(128)	Representa el nombre de la modalidad.
description	Varchar(1000)	Representa el campo ámbito y cualquier otro campo que contenga una cadena con longitud de caracteres de 1000.
user_id	Varchar(128)	Identificador único del usuario que crea la modalidad.
modality_type_id	Varchar(128)	Tipo de modalidad.

3.5. Conclusiones parciales.

Durante la realización de este acápite se obtuvo los diagramas del Análisis, que son esenciales a la hora de comprender los requisitos funcionales que guían el desenvolvimiento del trabajo, y algo fundamental para entender la base de datos, de la cual estos componentes forman parte, y la descripción de cada una de sus tablas.

Capítulo IV. Implementación y prueba de los componentes para la gestión de Tareas en la plataforma ZERA.

Introducción.

En el presente capitulo se describe como está implementado el sistema, a través de los diagramas de componentes y el diagrama de despliegue. Se exponen y detallan los diferentes aspectos que se tuvieron en cuenta durante su desarrollo, además de las pruebas realizadas a las funcionalidades propuestas, en las cuales se verifican que estas estén correctamente desarrolladas.

4.1. Modelo de implementación.

El modelo de implementación describe cómo son implementadas las clases del modelo de diseño en términos de componentes. Además, organizar componentes según los mecanismos de estructuración disponible en el entorno de implementación y el lenguaje de programación utilizado así como su dependencia de otros componentes. A partir de este modelo se obtienen los diagramas de despliegue y de componentes que serán mostrados a continuación.

4.1.1. Diagrama de despliegue.

El diagrama de despliegue permite modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes, así como la representación de la distribución física del sistema a través de nodos.

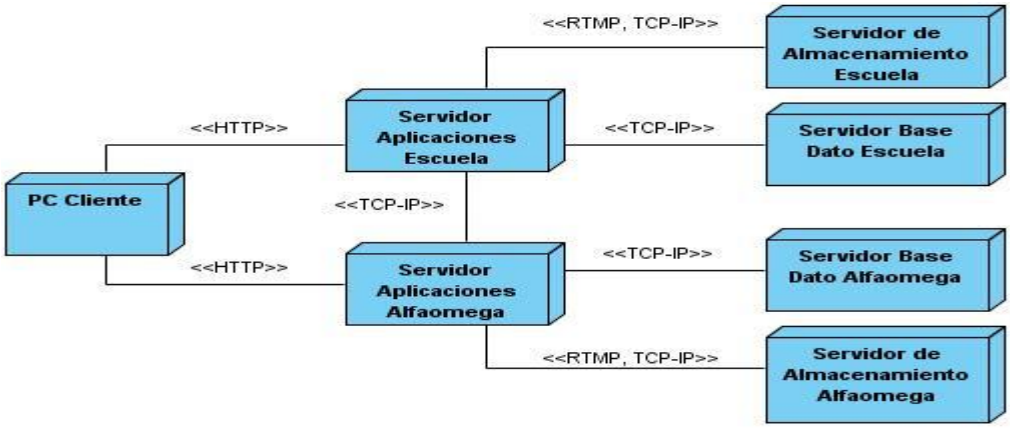

Imagen 11. Diagrama de despliegue.

4.1.2.Diagrama de componentes.

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre los mismos. Estos componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los diagramas de componentes siguientes fueron realizados siguiendo el patrón Modelo-Vista-Controlador utilizado por el framework Symfony.

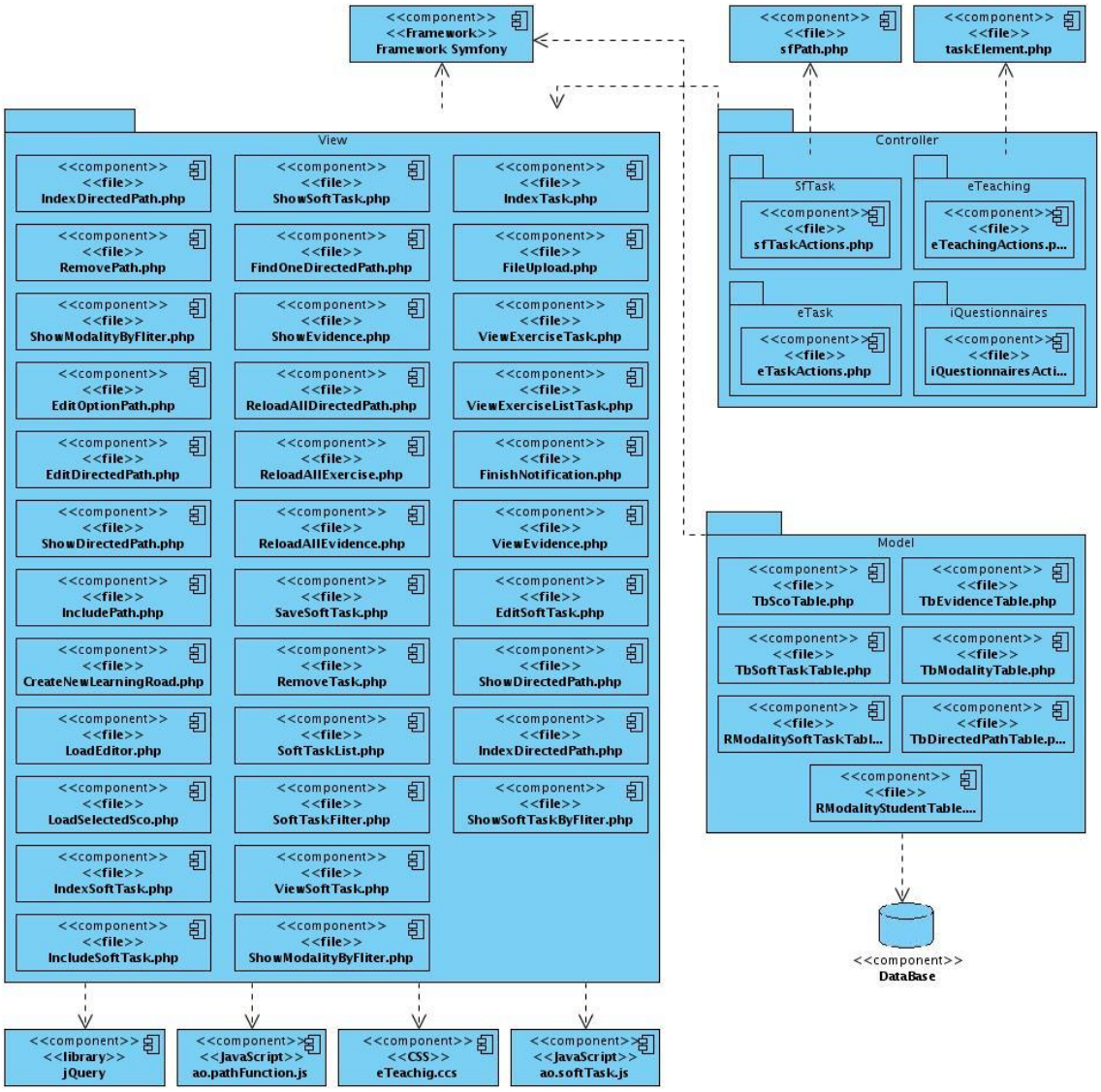

Imagen 12. Diagrama de Componentes.

4.2. Pruebas de software.

Las pruebas de software son los procesos que permiten verificar y revelar la calidad de un producto software. Son utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un producto informático. Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas. [22]

4.2.1. Niveles de las pruebas.

Pruebas de Integración: Se comprueba la compatibilidad y funcionalidad de los interfaces entre las distintas partes que componen un sistema, estas partes pueden ser módulos, aplicaciones individuales, aplicaciones cliente/servidor, etc. Este tipo de pruebas es especialmente relevante en aplicaciones distribuidas. [22]

Pruebas de Validación: Son las pruebas realizadas sobre un software completamente integrado para evaluar el cumplimiento con los requisitos especificados. [22]

Pruebas de Sistema: El software ya validado se integra con el resto del sistema donde algunos tipos de pruebas a considerar son: [22]

- Rendimiento: determinan los tiempos de respuesta, el espacio que ocupa el módulo en disco o en memoria, el flujo de datos que genera a través de un canal de comunicaciones, etc.
- Resistencia: determinan hasta donde puede soportar el programa determinadas condiciones extremas.
- Robustez: determinan la capacidad del programa para soportar entradas incorrectas.
- Seguridad: se determinan los niveles de permiso de usuarios, las operaciones de acceso al sistema y acceso a datos.
- Usabilidad: se determina la calidad de la experiencia de un usuario en la forma en la que éste interactúa con el sistema, se considera la facilidad de uso y el grado de satisfacción del usuario.
- Instalación: se determinan las operaciones de arranque y actualización del software.

Pruebas de Aceptación: Son las que realizará el cliente, para determinar si el sistema cumple con lo deseado y se obtiene la conformidad del mismo. [22]

4.2.2. Métodos de prueba.

En este acápite se describen las pruebas de caja blanca y las pruebas de caja negra poniendo más énfasis en esta última pues mediante el uso de casos de prueba se pudo comprobar la validez de las respuestas ofrecidas por el sistema dada una serie de entradas específicas.

- Prueba de caja negra: Esta prueba conociendo una función específica para la que fue diseñado el producto, se pueden diseñar pruebas que demuestren que dicha función está bien realizada. Dichas pruebas son llevadas a cabo sobre la interfaz del software, es decir, de la función, actuando sobre ella como una caja negra, proporcionando unas entradas y estudiando las salidas para ver si son o no las esperadas. [22]

- Prueba de caja blanca: Estas pruebas están dirigidas a las funciones internas. Entre las técnicas usadas se encuentran; la cobertura de caminos (pruebas que hagan que se recorran todos los posibles caminos de ejecución), pruebas sobre las expresiones lógico-aritmeticas, pruebas de camino de datos (definición-uso de variables), comprobación de bucles (se verifican los bucles para 0,1 y n iteraciones, y luego para las iteraciones máximas, máximas menos uno y más uno). [22]

4.2.3. Diseño de los casos de prueba.

Los casos de prueba que a continuación se describen fueron creados con el objetivo de detectar el mayor número de defectos posibles a las funcionalidades implementadas descritas anteriormente.

Tabla 15. Caso de prueba Realizar Softarea.

ID del escenario	Escenario	Variable	Respuesta del sistema	Flujo central
EC 1	Selecciona la opción de comenzar una tarea.		Muestra el contenido de la tarea. Y brinda las opciones de: • Guardar sin terminar • Ir a la página Siguiente. • Ir a la página Anterior. • Enviar. • Cancelar. • Salir de la vista actual.	Escritorio de Trabajo/Tareas/Softarea/Clic sobre el nombre de la Softarea / Botón Ver

EC 2	Termina la tarea.			Escritorio de Trabajo/Tareas/Softarea/Clic sobre el nombre de la Softarea / Botón Ver /Botón Finalizar
EC 3	Selecciona la opción cancelar.		Cancela la tarea. Regresa a la pantalla anterior.	Escritorio de Trabajo/Tareas/Softarea/Clic sobre el nombre de la Softarea / Botón Ver /Botón Salir
EC 4	Selecciona la opción página siguiente.		Muestra el siguiente objeto de aprendizaje. Si el actor se encuentra en la última página, se pone inactivo.	Escritorio de Trabajo/Tareas/Softarea/Clic sobre el nombre de la Softarea / Botón Ver /Botón Siguiente
EC 5	Selecciona la opción página anterior.		Muestra el objeto de aprendizaje anterior. Si el actor se encuentra en la primera página, se pone inactivo.	Escritorio de Trabajo/Tareas/Softarea/Clic sobre el nombre de la Softarea / Botón Ver /Botón Anterior
EC 6	Selecciona la opción salir de la vista actual.		Sale de la vista actual.	Escritorio de Trabajo/Tareas/Softarea/Clic sobre el nombre de la Softarea / Botón Ver /Botón Salir

Para ver el resto de los casos de prueba, *consultar Anexo III.*

4.2.4. Resultados obtenidos en las pruebas.

Las pruebas unitarias se realizaron sistemáticamente en dependencia del avance del desarrollo de las funcionalidades, comprobando en todo momento que lo desarrollado estuviera acorde a lo descrito. Estas pruebas fueron realizadas por el desarrollador durante el proceso de creación de las funcionalidades gracias a una series ventajas que brinda el IDE de desarrollo **NetBeans** 6.9, pero al no estar planificadas no se registraron los resultados alcanzados con las mismas.

Las pruebas que se realizaron, descritas a continuación, pertenecen al nivel de sistema y utilizan el método de caja negra:

- Pruebas cruzadas: Cada equipo de desarrollo de determinado módulo (desarrollador y analista), estuvo encargado de hacer estas pruebas, con el objetivo de detectar la mayor cantidad de problemas posibles en cuanto a pautas de diseño y errores de validación se trata.

- Pruebas Internas: Fueron realizadas por un grupo de analistas encargadas de la calidad interna de la plataforma ZERA, con el objetivo de que fueran detectados la menor cantidad de

problemas durante las pruebas de liberación. A continuación se muestran los resultados obtenidos durante estas pruebas:

Tabla 16. Resultados obtenidos durante las pruebas internas.

Nombre de los casos de prueba	No Conformidades			
	Alta	Media	Baja	Total
CP Realizar Softarea.				0
CP Consultar Recorrido Dirigido.			X	1
CP Consultar Softarea.	X			2
CP Gestionar Softarea.		X		4
CP Realizar Recorrido Dirigido	X		X	2

- Pruebas de liberación: Estas pruebas son realizadas por Calisoft, con el objetivo de declarar la plataforma ZERA en óptimas condiciones para ser entregada al cliente. A continuación se muestran los resultados obtenidos durante la primera iteración estas pruebas:

Tabla 17. Resultados obtenidos durante la primera iteración de las pruebas de liberación.

Nombre de los casos de prueba	No Conformidades			
	Alta	Media	Baja	Total
CP Realizar Softarea.			X	1
CP Consultar Recorrido Dirigido.			X	3
CP Consultar Softarea.			X	4
CP Gestionar Softarea.			X	3
CP Realizar Recorrido Dirigido				0

4.3. Conclusiones parciales.

En este capítulo se definieron los casos de prueba basados en CU, los cuales fueron de gran ayuda a la hora guiar las pruebas en cuanto a las funcionalidades del sistema para ser aceptado. Estas pruebas se le realizaron a la aplicación y tuvieron el objetivo de detectar y corregir errores, mejorando así en gran medida la calidad del software y favoreciendo de esta manera la aceptación final del cliente.

Conclusiones generales.

El desarrollo de la investigación aportó conocimientos teóricos básicos sobre la gestión de tareas y rutas de aprendizaje, así como el manejo de procesos como éste en plataformas con características similares en Cuba y el mundo, demostrando lo importante que resultan estos sistemas a la hora de utilizar contenidos y recursos, durante el proceso de creación de tareas entre varios LMS.

La selección de la Metodología de desarrollo RUP permitió la guía del proceso y posibilitó la creación de artefactos fundamentales que permitieron la construcción de las funcionalidades en los componentes implementados, con una alta calidad y que cumple con las expectativas esperadas por el usuario.

Una vez implementados estos componentes que permiten la gestión de tareas y rutas de aprendizaje para la plataforma ZERA, y luego de las pruebas de calidad aplicadas para la detección y corrección de errores existentes, se demostró de manera fehaciente su validez, así como el cumplimiento de las características requeridas y de los objetivos propuestos.

La documentación generada con este trabajo posibilitó la implementación de los componentes para la gestión de tareas y rutas de aprendizaje, además de servir de material de apoyo para futuras investigaciones.

Recomendaciones.

La creación de un plug-in que se encargue de realizar la gestión de los recorridos dirigidos o rutas de aprendizaje.

La posibilidad de separar las plantillas creadas en el visor de plantillas durante la gestión de los recorridos dirigidos o rutas de aprendizaje.

Referencias bibliográficas.

[1] García Peñalvo, Francisco José. [2006]: *"Estado actual de los sistemas e-learning"*. Universidad de Salamanca, España.

[2] Ortega, Facundo. [2003] *"La Universidad. Entre la gestión y el conocimiento".* Córdoba. Argentina.

[3] Sánchez Ilabaca, Jaime. [1999]: *"Construyendo y aprendiendo con el computador"* Universidad de Chile, Chile.

[4] Real Academia Española. Real Academia Española. [En línea] [2001]: http://buscon.rae.es/

[5] Claroline. [En línea], [2010]: http://doc.claroline.net/es/index.php/Portada.

[6] Moodle. [En Línea], [2011]: http://docs.moodle.org/es/Acerca_de_Moodle

[7] Calzadilla Rodríguez, Iraida (Colección Futuro). [En Línea], [2004]: http://www.cubahora.cu/index.php?tpl=buscar/ver-not_buscar.tpl.html&newsid_obj_id=1002645

[8] Kruchten, Philippe. [2001]: "The Rational Unified Process an Introduction".

[9] Extreme Programming (XP). [En Línea], [2009]: http://www.extremeprogramming.org/.

[10] Casasola Romero, Oscar. "*Introducción al Unified Modeling Language (UML)".* [En Línea], [2010]: http://www.programacion.com/articulo/introduccion_a_uml_181.

[11] Prado Ajona, Guillermo (HTML). [En Línea], Universidad de la Rioja, España [2011]: https://belenus.unirioja.es/~guprado/pagweb/carachtml.html

[12] W3C (Definición de HTML) (Características de Ajax). [En Línea], [2011]: http://www.w3c.es/.

[13] PHP. [En línea], [2010]: http://php.net/manual/es.

[14] Symfony. [En línea], [2010]: http://www.symfony.es/.

[15] Eguíluz Pérez, Javier. [En Línea]: [2009]. *"Introducción a Java Script"*: http://www.librosweb.es/javascript

[16] Briggs, Owen. [2003]: “Cascading Style Sheets”. Universidad de Boston, Estados Unidos de América.

[17] Greg Murray. [En Línea], [2008]. *“Asynchronous JavaScript Technology and XML (Ajax) With the Java Platform”:* http://www.oracle.com/technetwork/articles/javaee/ajax-135201.html

[18] James Garrett, Jesse. [En Línea], [2008]. *“Ajax: A New Approach to Web Applications”:* http://www.adaptivepath.com/ideas/essays/archives/000385.php

[19] PostgreSQL. [En Línea], [2011]: http://www.postgresql-es.org/sobre_postgresql

[20] NetBeans 6.9. [En Línea], [2011]: http://netbeans.org/index_es.html

[21] Visual-Paradigm. (*Sitio Web oficial Visual-Paradigm).* [En línea], [2008]: http://www.visual-paradigm.com/product/vpuml/.

[22] Pressman. [2005]: *“Ingeniería de Software. Un enfoque práctico”*.

Bibliografía.

Briggs, Owen. [2003]: “Cascading Style Sheets”. Universidad de Boston, Estados Unidos de América.

Claroline. [En línea], [2010]: http://doc.claroline.net/es/index.php/Portada.

Calzadilla Rodríguez, Iraida (Colección Futuro). [En Línea], [2004]: http://www.cubahora.cu/index.php?tpl=buscar/ver-not_buscar.tpl.html&newsid_obj_id=1002645

Casasola Romero, Oscar. “*Introducción al Unified Modeling Language (UML)”.* [En Línea], [2010]: http://www.programacion.com/articulo/introduccion_a_uml_181.

Extreme Programming (XP). [En Línea], [2009]: http://www.extremeprogramming.org/.

Eguíluz Pérez, Javier. [En Línea]: [2009]. *“Introducción a Java Script”*: http://www.librosweb.es/javascript

Fabien Potencier, François Zaninotto. [En Línea][2008]: “Symfony la guía definitiva”. www.librosweb.es

García Peñalvo, Francisco José. [2006]: *“Estado actual de los sistemas e-learning”*. Universidad de Salamanca, España.

Greg Murray. [En Línea], [2008]. *“Asynchronous JavaScript Technology and XML (Ajax) With the Java Platform”:* http://www.oracle.com/technetwork/articles/javaee/ajax-135201.html

James Garrett, Jesse. [En Línea], [2008]. *“Ajax: A New Approach to Web Applications”:* http://www.adaptivepath.com/ideas/essays/archives/000385.php

Kruchten, Philippe. [2001]: “The Rational Unified Process an Introduction”.

Moodle. [En Línea], [2011]: http://docs.moodle.org/es/Acerca_de_Moodle

NetBeans 6.9. [En Línea], [2011]: http://netbeans.org/index_es.html

Ortega, Facundo. [2003] *“La Universidad. Entre la gestión y el conocimiento”.* Córdoba. Argentina.

Prado Ajona, Guillermo (HTML). [En Línea], Universidad de la Rioja, España [2011]: https://belenus.unirioja.es/~guprado/pagweb/carachtml.html

PHP. [En línea], [2010]: http://php.net/manual/es.

PostgreSQL. [En Línea], [2011]: http://www.postgresql-es.org/sobre_postgresql

Pressman. [2005]: *"Ingeniería de Software. Un enfoque práctico"*.

Real Academia Española. Real Academia Española. [En línea] [2001]: http://buscon.rae.es/

Symfony. [En línea], [2010]: http://www.symfony.es/.

Sánchez Ilabaca, Jaime. [1999]: *"Construyendo y aprendiendo con el computador"* Universidad de Chile, Chile.

Visual-Paradigm. (*Sitio Web oficial Visual-Paradigm).* [En línea], [2008]: http://www.visual-paradigm.com/product/vpuml/.

W3C (Definición de HTML) (Características de Ajax). [En Línea], [2011]: http://www.w3c.es/.

Glosario de Términos.

Autónoma: Condición de realizar alguna acción por iniciativa o interés propio.

Tiempo Real: cuando se realiza una transacción que le ha sido ordenada desde un terminal en ese mismo momento, sin espera alguna.

Anexos

Anexo I. Especificaciones de los CU de sistema.

1. CU Gestionar Softarea.

1.1 Flujo básico de CU Incluir Softarea.

Tabla 20. Flujo básico de CU Incluir Softarea

Acciones del actor	Respuesta del sistema
1. El caso de uso se inicia cuando el actor selecciona la opción de realizar una acción sobre una Tarea.	
	2. Brinda la posibilidad de realizar las siguientes acciones: • Incluir una nueva tarea. • Ver los datos de la tarea. Ver Sección 1: *"Ver datos de la tarea".* • Modificar los datos de la entidad. Ver Sección 2:"Modificar datos de la tarea". • Eliminar la entidad. Ver Sección 3: *"Eliminar tarea".*
3. Selecciona la opción de incluir una nueva tarea.	
	4. Muestra los tipos de elementos asociados a la tarea y permite seleccionar: • Recorrido Dirigido. • Estudiantes y\o grupos de ellos. Y permite además: • Crear Recorrido Dirigido. • Actualizar Listado_RD. • Seleccionar todos los estudiantes. • Desmarcar todos los estudiantes. • Ir al botón Siguiente. • Salir de la Gestión de la Softarea.
5. Selecciona un RD y el, o los estudiantes involucrados.	
6. Selecciona el botón Siguiente.	
	7. Muestra una lista de: • Evidencias. • Ejercicios. Y permite seleccionar al menos uno de estos elementos. Permite: • Crear Evidencia. • Actualizar Listado_E.

	• Ir al botón Anterior. • Ir al botón Siguiente. • Salir de la Gestión de la Softarea.
8. Selecciona uno o más ejercicios y/o evidencia.	
9. Selecciona el botón Siguiente.	
	10. Permite seleccionar: • Fecha de entrega. • Hora en que cierra. E introducir: • Nombre (Identificador de la tarea). • Propósito. • Acciones a llevar a cabo. Y permite: • Ir al botón Anterior. • Ir al botón Siguiente. • Salir de la Gestión de la Softarea.
11. Selecciona la fecha de entrega, la hora en que cierra, e introduce el nombre de la Softarea, propósito y acciones a llevar a cabo.	
12. Selecciona el botón Siguiente.	
	13. Muestra un resumen de la tarea y permite: • Ir al botón Anterior. • Finalizar y asignar. • Salir de la Gestión de la Softarea.
14. El actor selecciona la opción de Finalizar y asignar.	
	15. Valida los datos.
	16. Crea la tarea.
	17. Finaliza la tarea y la envía a los estudiantes seleccionados.
	18. La asignación de la tarea se notifica a la mensajería del estudiante.
	19. Se crea un evento en la agenda, con la fecha definida en la tarea.
	20. El caso de uso termina.

1.2 Flujo alterno de CU Incluir Softarea.

Tabla 21. Flujo alterno de CU Incluir Softarea

3. a El actor selecciona la opción de Crear Recorrido Dirigido.	
Acciones del actor	**Respuesta del sistema**
	3. a Muestra el *CU Gestionar Recorrido Dirigido.*
3. b El actor selecciona la opción de Actualizar Listado_RD.	

Acciones del actor	**Respuesta del sistema**
	3. b. 1 Se actualiza el listado de recorridos dirigidos.
	3. b. 2 Regresa al paso 3 del flujo básico.
3. c El actor selecciona la opción de Seleccionar todos los estudiantes.	
Acciones del actor	**Respuesta del sistema**
	3. c Se seleccionan todos los estudiantes.
3. d El actor selecciona la opción de Desmarcar todos los estudiantes.	
Acciones del actor	**Respuesta del sistema**
	3. d Se desmarcaran todos los estudiantes.
3.e El actor selecciona la opción de Crear Evidencia.	
Acciones del actor	**Respuesta del sistema**
	3.e Muestra el *CU Gestionar Evidencia.*
3. f El actor selecciona la opción de Actualizar Listado_E.	
Acciones del actor	**Respuesta del sistema**
	3. f. 1 Se actualiza el listado de las evidencias.
	3. f. 2 Regresa al paso 6 del flujo básico.
*. El actor selecciona el botón Anterior.	
Acciones del actor	**Respuesta del sistema**
	*. Regresa a la página anterior.
3. g Existen campo incompletos.	
Acciones del actor	**Respuesta del sistema**
	3. g.1 Sobre cada campo incompleto muestra un mensaje de información.
	3. g.2 Regresa al paso 4 del Flujo Básico.
3. h Existen datos incorrectos.	
Acciones del actor	**Respuesta del sistema**
	3. h.1 Sobre cada campo incorrecto muestra un mensaje de información.
	3. h.2 Regresa al paso 9 del Flujo Básico.
*. Selecciona la opción Cancelar.	
Acciones del actor	**Respuesta del sistema**
	*. 1 Muestra un mensaje de información.
	*. 2 Regresa a la pantalla anterior.

Anexo II. Diagrama de clases del diseño.

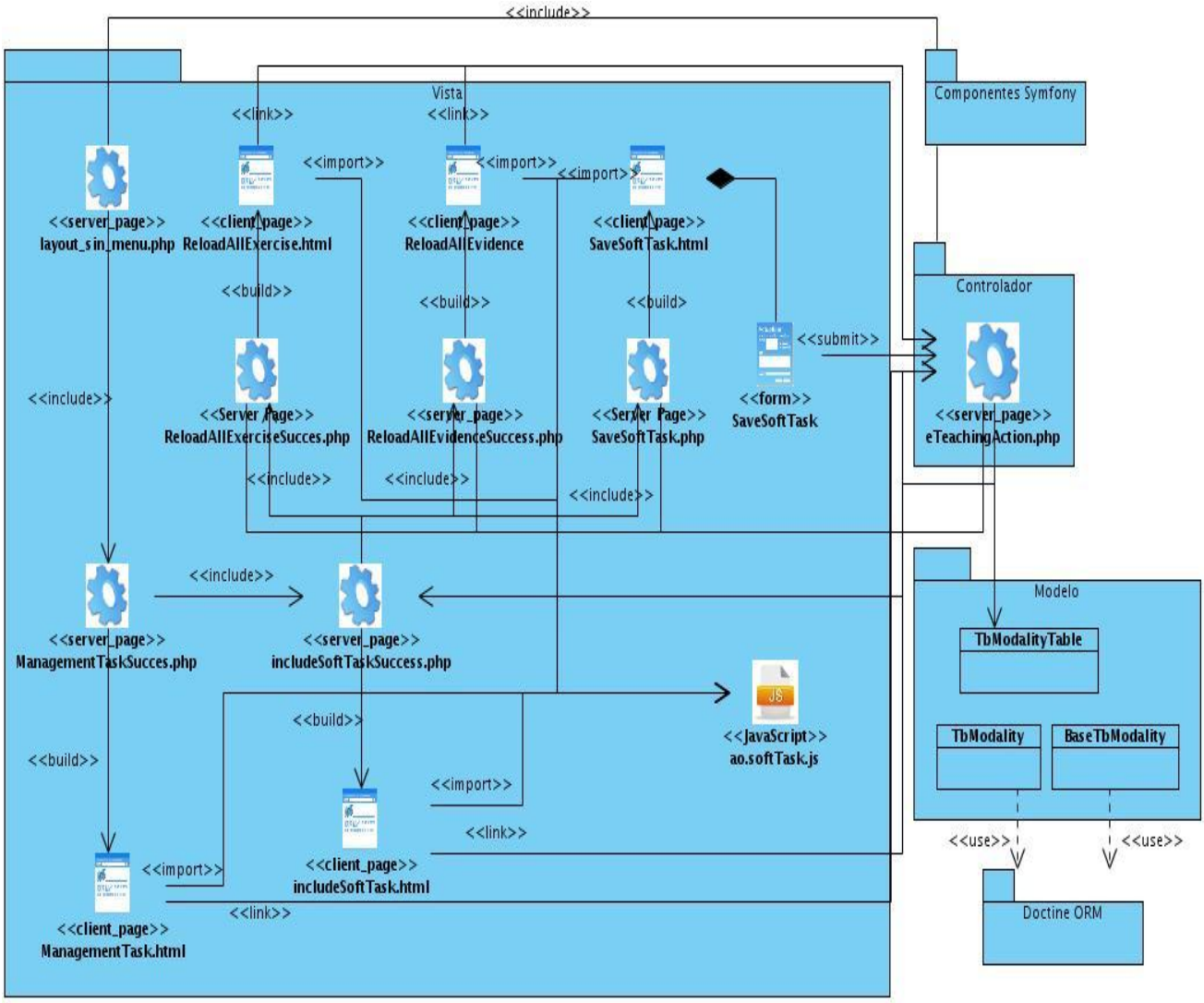

Imagen 13. Diagrama de cases del diseño (Gestionar ST).

Anexo III. Casos de prueba.

1. Caso de prueba gestionar Softarea.

Tabla 32. Caso de prueba Gestionar Softarea (Incluir Softarea)

ID del escenario	Escenario	RD	Est	Ev	Ej	FE	HC	N	P	Respuesta del sistema	Flujo central
EC 1	Selecciona la opción de incluir una nueva tarea.	*V*	*V*							Muestra los tipos de elementos asociados a la tarea y permite seleccionar: • Recorrido Dirigido. • Estudiantes y\o grupos de ellos. Y permite además: • Crear Recorrido Dirigido. • Actualizar Listado RD. • Seleccionar todos los estudiantes. • Desmarcar todos los estudiantes. • Ir al botón Siguiente. • Salir de la Gestión de la Softarea.	Escritorio de trabajo/Docente/ Administración del Aprendizaje / Gestión de Tareas /Softarea/ link Ver sección /Botón nueva
EC 2	Selecciona un RD.			*V*	*N\A*					Muestra una lista de: • Evidencias. • Ejercicios. Y permite seleccionar al menos uno de estos elementos. Permite: • Crear Evidencia.	Escritorio de trabajo/Docente/ Administración del Aprendizaje / Gestión de Tareas /Softarea/ link Ver sección /Botón nueva/Lista de recorrido dirigido/ Clic sobre el nombre
				N\A	*V*						

	Selecciona el o los estudiantes involucrados. Selecciona el botón Siguiente.									• Actualizar Listado E. • Ir al botón Anterior. • Ir al botón Siguiente. • Salir de la Gestión de la Softarea.	del RD Escritorio de trabajo/Docente/ Administración del Aprendizaje / Gestión de Tareas /Softarea/ link Ver sección /Botón nueva/ Clic sobre el checkbox del grupo o de los estudiantes en específico/ Botón de asociar (flecha que apunta a la derecha)/Botón siguiente
EC 3	Selecciona una evidencia y uno o más ejercicios y/o Selecciona el botón Siguiente.					*V*	*V*	*V*	*V*	Permite seleccionar: • Fecha de entrega. • Hora en que cierra. • E introducir: • Nombre (Identificador de la tarea). • Propósito. • Acciones a llevar a cabo. • Y permite: • Ir al botón Anterior. • Ir al botón Siguiente. • Salir de la Gestión de la Softarea.	Escritorio de trabajo/Docente/ Administración del Aprendizaje / Gestión de Tareas /Softarea/ link Ver sección /Botón nueva/ Clic sobre el checkbox del grupo o de los estudiantes en específico/ Botón de asociar (flecha que apunta a la derecha)/Botón siguiente/ Listado de evidencias/Clic en el checkbox de la evidencia/Botón Siguiente/ Árbol de contenidos/ Clic en el checkbox del nivel del árbol deseado/ Clic en el checkbox de

											los ejercicios deseados/Botón Siguiente
EC 4	Selecciona la fecha de inicio, fecha de entrega, la hora en que cierra, e introduce el nombre de la Softarea, propósito y acciones a llevar a cabo. Selecciona el botón Siguiente.									Muestra un resumen de la tarea y permite: • Ir al botón Anterior. • Finalizar y asignar. • Salir de la Gestión de la Softarea.	Escritorio de trabajo/Docente/ Administración del Aprendizaje / Gestión de Tareas /Softarea/ link Ver sección /Botón nueva/ Clic sobre el checkbox del grupo o de los estudiantes en específico/ Botón de asociar (flecha que apunta a la derecha)/Botón siguiente/ Listado de evidencias/Clic en el checkbox de la evidencia/Botón Siguiente/ Árbol de contenidos/ Clic en el checkbox del nivel del árbol deseado/ Clic en el checkbox de los ejercicios deseados/Botón Siguiente/Datos generales de la Softarea/Botón siguiente
EC 5	El actor selecciona la opción de Finalizar y asignar.									*Valida los datos.* *Crea la tarea.* Finaliza la tarea y la envía a los estudiantes seleccionados.	Escritorio de trabajo/Docente/ Administración del Aprendizaje / Gestión de Tareas /Softarea/ link Ver sección /Botón nueva/ Clic sobre el checkbox del grupo o de los estudiantes en específico/ Botón de asociar (flecha que
										La asignación de la tarea se notifica a la mensajería del estudiante. *Se crea un evento en la agenda, con la*	

										fecha definida en la tarea. El caso de uso termina.	apunta a la derecha)/Botón siguiente/ Listado de evidencias/Clic en el checkbox de la evidencia/Botón Siguiente/ Árbol de contenidos/ Clic en el checkbox del nivel del árbol deseado/ Clic en el checkbox de los ejercicios deseados/Botón Siguiente/Datos generales de la Softarea/Botón siguiente/Botón Salvar
EC 6	Selecciona la opción de Crear Recorrido Dirigido.									Muestra el *CU Gestionar Recorrido Dirigido.*	Escritorio de trabajo/Docente/ Administración del Aprendizaje / Gestión de Tareas /Softarea/ link Ver sección /Botón nueva/Lista de recorrido dirigido/ Botón Crear RD
EC 7	Selecciona la opción de Actualizar Listado RD.									Se actualiza el listado de recorridos dirigidos. Regresa al paso 3 del flujo básico.	Escritorio de trabajo/Docente/ Administración del Aprendizaje / Gestión de Tareas /Softarea/ link Ver sección /Botón nueva/Lista de recorrido dirigido/ Botón Crear RD/ Botón Actualizar
EC 8	Selecciona la opción de Desmarcar todos los estudiantes.									Se desmarcaran todos los estudiantes.	Escritorio de trabajo/Docente/ Administración del Aprendizaje / Gestión de

											Tareas /Softarea/ link Ver sección /Botón nueva/ Clic sobre el checkbox del grupo o de los estudiantes en específico/ Botón de asociar (flecha que apunta a la izquierda)
EC 9	Selecciona la opción de Crear Evidencia.									Muestra el *CU Gestionar Evidencia.*	Escritorio de trabajo/Docente/ Administración del Aprendizaje / Gestión de Tareas /Softarea/ link Ver sección /Botón nueva/ Clic sobre el checkbox del grupo o de los estudiantes en específico/ Botón de asociar (flecha que apunta a la derecha)/Botón siguiente/ Gestión de evidencias/Botón salvar
EC 10	Selecciona la opción de Actualizar Listado E.									Se actualiza el listado de las evidencias. Regresa al paso 6 del flujo básico.	Escritorio de trabajo/Docente/ Administración del Aprendizaje / Gestión de Tareas /Softarea/ link Ver sección /Botón nueva/ Clic sobre el checkbox del grupo o de los estudiantes en específico/ Botón de asociar (flecha que apunta a la derecha)/Botón siguiente/ Gestión de evidencias/

											Botón salvar /Botón Actualizar
EC 11	Selecciona el botón Anterior.									Regresa a la pantalla anterior.	Escritorio de trabajo/Docente/ Administración del Aprendizaje / Gestión de Tareas /Softarea/Botón Siguiente/ Botón Anterior
EC 12	Existen campo incompletos.	V	V	V	V	V	V	V	V	Se crea la tarea exitosamente. Sobre cada campo incompleto muestra un mensaje de información.	Escritorio de trabajo/Docente/ Administración del Aprendizaje / Gestión de Tareas /Softarea/ link Ver sección /Botón nueva/ Clic sobre el checkbox del grupo o de los estudiantes en específico/ Botón de asociar (flecha que apunta a la derecha)/Botón siguiente/ Gestión de evidencias/Botón salvar
		V	V	N\A	V	V	V	V	V		
		V	V	V	N\A	V	V	V	V		
		I	V	I	I	V	V	V	V		
		V	I	I	I	V	V	V	V		
		V	V	V	N\A	I	V	V	V		
		V	V	N\A	V	V	I	V	V		
		V	V	I	I	V	V	I	V		
		V	V	I	I	V	V	V	I		
		V	V	I	I	V	V	V	V		
EC 13	Existen Datos incorrectos.	N\A	N\A	N\A	N\A	N\A	N\A	I	V	Sobre cada campo incorrecto muestra un mensaje de información.	Escritorio de trabajo/Docente/ Administración del Aprendizaje / Gestión de Tareas /Softarea/ link Ver sección /Botón nueva/ Clic sobre el checkbox del grupo o de los estudiantes en específico/ Botón de asociar (flecha que apunta a la derecha)/Botón siguiente/
		N\A	N\A	N\A	N\A	N\A	N\A	V	I		
		N\A	N\A	N\A	N\A	N\A	N\A	V	V		

											Gestión de evidencias/Botón salvar
EC 14	Selecciona la opción de Cancelar de la Gestión de la Softarea.									Muestra un mensaje de información. Permite: Aceptar. Cancelar.	Escritorio de trabajo/Docente/ Administración del Aprendizaje / Gestión de Tareas /Softarea/ link Ver sección /Botón nueva/Botón Cancelar
EC 15	Selecciona la opción Guardar sin terminar.									Sale de la gestión de la Softarea.	Escritorio de trabajo/Docente/ Administración del Aprendizaje / Gestión de Tareas /Softarea/ link Ver sección /Botón nueva/ Clic sobre el checkbox del grupo o de los estudiantes en específico/ Botón de asociar (flecha que apunta a la derecha)/Botón siguiente/Botón Guardar sin terminar

Leyenda para el caso de prueba (Incluir Softarea):

- RD – Recorrido Dirigido.
- Ev – Evidencia.
- Ej – Ejercicios.
- FE – Fecha de entrega.
- HC – Hora de cierre.
- N – Nombre.
- P – Propósito.
- Est – Estudiantes.