



Universidad de las Ciencias Informáticas

Facultad 4



# Integración de los instrumentos de evaluación para la gestión de las evidencias en la plataforma educativa Zera

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

## **Autores:**

Dunet Calderín Gil

Leonardo Pérez Cabrera

## **Tutores:**

Ing. Mairelis Gari Maribona

MSc. Abel Ernesto Lorente Rodríguez

## **Co-Tutores:**

Ing. Irina Ivis Santiesteban Pérez

Ing. Yoandy Pérez Caceres

Ing. Jose Antonio Soto Pérez

**La Habana, junio 2012**

**“Año 54 de la Revolución”**

## Declaración de autoría

Declaramos que somos los autores de este trabajo y autorizamos a la Facultad 4 de la Universidad de las Ciencias Informáticas, así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del año 2012.

---

**Autora**

Dunet Calderín Gil

---

**Autor**

Leonardo Pérez Cabrera

---

**Tutora**

Ing. Mairelis Gari Maribona

---

**Tutor**

MsC. Abel Ernesto Lorente  
Rodríguez

---

**Co-tutora**

Ing. Irina Ivis Santiesteban  
Pérez

---

**Co-tutor**

Ing. Yoandy Pérez Caceres

---

**Co-tutor**

Ing. Jose Antonio Soto Pérez

## Dedicatoria

### *De Dunet:*

Le quisiera dedicar el presente trabajo a mi familia y a mi novio por su apoyo incondicional, pero en especial a mis padres por haberse sacrificado tanto por mi y por mis hermanos, por ser mis amigos, por llorar cuando me sentía triste y estar felices cuando yo lo estaba, por ser mis tutores incansables y mi apoyo en todos los momentos de mi vida; por ser los padres que cualquier hijo soñaría tener.

### *De Leonardo:*

A mis padres que siempre quisieron verme aquí y que este también era su sueño.

## Resumen

Desde su creación, los instrumentos de evaluación han servido de apoyo a los procesos de enseñanza-aprendizaje. Con el auge de Internet ha surgido la educación a distancia y con ella la creación de diferentes plataformas para la gestión de la educación y el aprendizaje: las llamadas plataformas educativas, en las cuales se han implementado algunos de los instrumentos de evaluación existentes. El presente trabajo expone cómo se integrarán los instrumentos de evaluación que han sido identificados en la plataforma educativa Zera, para brindarle al estudiante una potente herramienta de aval donde se almacenen las evidencias de aprendizaje, logros y habilidades que el estudiante alcanza durante su vida estudiantil: el portafolio electrónico. Además se realizó un estudio de la situación de estos en el ámbito nacional e internacional para determinar qué estándar implementar para lograr que exista la interoperabilidad de los datos de los estudiantes con otras plataformas educativas existentes. Se argumentaron las tecnologías, herramientas, lenguajes y metodología que serán empleadas para el desarrollo de la solución propuesta.

**Palabras clave:** portafolio electrónico, evidencias de aprendizaje, instrumentos de evaluación, interoperabilidad.

## Índice

<b>Introducción .....</b>	<b>1</b>
<b>Capítulo I: Fundamentación teórica .....</b>	<b>7</b>
Introducción .....	7
1.1. Valoración de los conceptos asociados al dominio del problema .....	7
1.2. IE identificados en Zera .....	8
1.2.2. IE que se emplean actualmente .....	9
1.2.2. IE que no se emplean .....	10
1.2. Panorama internacional sobre los IE identificados en Zera .....	12
1.3. Relación de los instrumentos de evaluación con el portafolio .....	15
1.4. Portafolio electrónico .....	16
1.4.1. Tipos de portafolio electrónico .....	17
1.4.2. Estructura de un portafolio electrónico .....	17
1.4.3. Panorama nacional de los portafolios electrónicos .....	18
1.4.4. Panorama internacional de los portafolios electrónicos .....	18
1.4.5. IMS Leap2A .....	21
1.5.1. Metodología de desarrollo de software .....	22
1.5.2. Lenguaje de modelado .....	23
1.5.3. Lenguajes de desarrollo .....	24
1.5.4. Framework .....	26
1.5.5. Sistema gestor de Base de Datos .....	29
1.5.6. Servidor web .....	29
1.5.7. Entorno de desarrollo .....	30
1.5.8. Herramientas CASE para el modelado UML .....	31
1.6. Conclusiones parciales del capítulo .....	32

<b>Capítulo II: Características del sistema.....</b>	<b>33</b>
Introducción.....	33
2.1. Modelo de dominio.....	33
2.1.1. Conceptos del modelo de dominio.....	33
2.1.2. Modelo de dominio.....	35
2.2. Descripción de la solución.....	36
2.2.1. Para los instrumentos de evaluación.....	36
2.2.2. Para el portafolio del estudiante.....	38
2.3. Requerimientos del software.....	39
2.3.1. Definición de los requisitos funcionales.....	39
2.3.2. Definición de los requisitos no funcionales.....	43
2.4. Presentación del diagrama de Caso de Uso.....	44
2.4.1. Descripción de los actores del sistema.....	44
2.4.2. Consideraciones previas al diagrama de casos de uso.....	45
2.4.3. Patrones de casos de uso.....	46
2.4.4. Diagrama de actores del sistema.....	46
2.4.5. Diagrama de casos de uso del sistema.....	47
2.4.6. Descripción de los casos de uso del sistema.....	48
2.5. Conclusiones parciales del capítulo.....	49
<b>Capítulo III: Análisis y diseño de la propuesta del sistema.....</b>	<b>50</b>
Introducción.....	50
3.1. Modelo del análisis.....	50
3.1.1. Diagrama de clases del análisis.....	50
3.2. Patrones en Symfony.....	52
3.2.1. Patrón arquitectónico Modelo- Vista- Controlador (MVC).....	52
3.2.2. Patrones de diseño.....	53

3.3. Modelo de diseño .....	55
3.3.1. Diagrama de clases del diseño .....	55
3.3.2. Diagrama de paquete del diseño .....	58
3.3.3. Modelo de despliegue .....	59
3.4. Diseño de la Base de datos .....	59
3.5. Conclusiones parciales del capítulo .....	60
<b>Capítulo IV: Implementación y prueba .....</b>	<b>61</b>
Introducción .....	61
4.1. Modelo de implementación .....	61
4.1.1. Diagrama de componentes .....	61
4.2. Pruebas de calidad del software .....	63
4.2.1. Niveles de prueba .....	63
4.2.2. Métodos de prueba .....	63
4.2.3. Diseño de Casos de prueba .....	64
4.3. Resultados obtenidos .....	64
4.3. Pruebas para comprobar las especificaciones implementadas del Leap2A .....	66
4.4. Conclusiones parciales del capítulo .....	67
<b>Conclusiones .....</b>	<b>68</b>
<b>Recomendaciones .....</b>	<b>70</b>
<b>Referencia bibliográfica .....</b>	<b>71</b>

## Introducción

En el marco educativo, los instrumentos de evaluación son las herramientas que usa el docente para obtener evidencias de los desempeños de los estudiantes en el proceso de enseñanza - aprendizaje (1). Estas evidencias se refieren a la comprobación de los saberes de un individuo que pueden ser de conocimiento, desempeño o comportamiento (conocer, hacer, ser). Del análisis de estas se pueden deducir los logros de aprendizaje, las destrezas y habilidades de cada individuo y establecer o no el desarrollo de determinadas competencias. Ante la necesidad de englobar estos logros y conocimientos, procesar la información de los mismos e intercambiarlos entre diferentes sistemas, surgieron los portafolios electrónicos.

Un portafolio electrónico es una colección de materiales orientada a objetivos y organizada, que demuestra el conocimiento y las habilidades de una persona a lo largo del tiempo. El contenido, la organización y la presentación de materiales en portafolios varían enormemente dependiendo de su audiencia y objetivo. Sin embargo, todos los portafolios muestran pruebas tangibles del crecimiento de un individuo y su desarrollo. (2)

Con el auge de los portafolios electrónicos surgieron cuestiones relacionadas con la interoperabilidad y usabilidad de los mismos. Para ello se han desarrollado un conjunto de herramientas, especificaciones y estándares. Entre las especificaciones que se han desarrollado para estos fines se encuentran; el paquete de información del estudiante (Learner Information Package o IMS – LIP), el IMS ePortafolio y el Leap2A.

Con la era de Internet el mundo educativo ha evolucionado y los profesionales de la educación tienen múltiples razones para aprovechar al máximo las nuevas Tecnologías de la Información y las Comunicaciones (TIC). Para impulsar la incorporación de las TIC al proceso de enseñanza - aprendizaje se han creado herramientas, entre las que se destacan los Sistemas de Gestión de Aprendizaje o LMS (en inglés Learning Management System). Los LMS son aplicaciones Web que facilitan la interacción entre los docentes y los estudiantes, aportan herramientas para la gestión de contenidos académicos y permiten el seguimiento y la evaluación (3).

En el Centro de Tecnologías para la Formación (FORTES) de la Universidad de las Ciencias Informáticas actualmente se está desarrollando una Plataforma LMS para la gestión del aprendizaje llamada Zera, la cual está basada en la concepción pedagógica de hiperentornos de aprendizaje<sup>1</sup> y ha incorporado funcionalidades incluidas en plataformas mundialmente conocidas como Sakai, Moodle y Dokeos.

Actualmente en la plataforma educativa Zera existen una serie de instrumentos de evaluación que presentan deficiencias en cuanto a la adquisición de la evaluación, ya que en el caso del Foro no emite una evaluación, la visualización del resultado en el caso del Reactivo debido a que no muestra, una vez concluida la actividad, los ejercicios realizados con las respuestas emitidas y nomenclatura en el caso del instrumento Evidencia. Además existen otros, que a la hora de analizar el desempeño del estudiante, no se tienen en cuenta, lo que dificulta al docente tener un criterio real del aprendizaje alcanzado por los mismos.

Existe además el módulo Portafolio en el que se almacenan todos los resultados del estudiante, sus calificaciones y las notificaciones asociadas a estas y brinda la posibilidad de compartir las evidencias evaluadas con otros compañeros de grupo. Sin embargo, atendiendo al significado de las evidencias en el proceso de enseñanza – aprendizaje y la gestión de estas en el desarrollo de las competencias adquiridas por los estudiantes, lo implementado actualmente en el portafolio de la plataforma no satisface las necesidades actuales, ya que en él no se almacenan todas las evidencias de aprendizaje alcanzadas por el estudiante, no permite consultar sus datos personales y no está implementado bajo las especificaciones de ninguno de los estándares anteriormente mencionados, lo que impide que no exista la interoperabilidad<sup>2</sup> de los datos de los estudiantes con otras plataformas educativas ya existentes. Además, con el portafolio de la plataforma educativa Zera se dificulta dar un seguimiento completo al aprendizaje de los estudiantes ya que al terminar un programa de estudio o realizar un cambio de grupo o de institución no se tiene acceso a las calificaciones y evidencias obtenidas en el proceso de enseñanza –

---

<sup>1</sup> **Hiperentornos de aprendizaje:** Sistema informático basado en tecnología hipermedia, en el que se mezclan diversas tipologías de software educativo con el mejor propósito de atender las necesidades didácticas.

<sup>2</sup> Es la capacidad que tiene un producto o un sistema, cuyas interfaces son totalmente conocidas, para funcionar con otros productos o sistemas existentes o futuros y eso sin restricción de acceso o de implementación.

aprendizaje. Esto se debe a que el negocio concebido para el portafolio se centraba en la relación que existía entre un estudiante y su docente.

En el contexto de estos procesos, los diseños de las diferentes disciplinas propician un marco de trabajo común en el cual se establecen los objetivos generales, los valores, habilidades, desempeños y competencias que se pretende que los estudiantes adquieran. El diseño de estas y la planificación de diversas actividades de formación podrían verse relacionadas con la gestión de Evidencias<sup>3</sup>. A pesar de que en la plataforma se realiza la asignación de tareas para que el estudiante la responda y adjunte una evidencia asociada a la misma, este proceso de asignación está limitado al trabajo del docente en sus grupos, impidiendo así la autogeneración de evidencias por parte de los estudiantes fuera del marco evaluativo; además una vez creada y asignada la tarea, el docente no puede utilizarla en otros programas de estudios, ni siquiera en otro de sus grupos. Por lo que no existe un lugar donde se gestionen de forma centralizada para su posterior uso, así como su gestión desde la creación y desarrollo de la disciplina o materia.

Una vez analizadas las necesidades existentes se define como **problema de investigación**: ¿Cómo facilitar el análisis de las evidencias de aprendizaje en la plataforma educativa Zera?

El **objeto de estudio** fijado se enmarca en el proceso de recopilación de evidencias del aprendizaje de los estudiantes en plataformas educativas y el **campo de acción** recae en la integración de los instrumentos de evaluación en la plataforma educativa Zera.

**El objetivo general del presente trabajo es:**

- Integrar los instrumentos de evaluación existentes en la plataforma educativa Zera para facilitar el análisis de las evidencias.

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

1. Realizar un análisis del estudio del estado del arte.
2. Determinar la información que brindan los instrumentos de evaluación que pueda ser utilizada como evidencias del aprendizaje.
3. Describir la propuesta del sistema.

---

<sup>3</sup> Evidencia: Nomenclatura de uno de los instrumentos de evaluación existentes en la plataforma educativa Zera

4. Facilitar la interoperabilidad de los datos de los estudiantes.
5. Extender el uso de las funcionalidades para la gestión de evidencias en la creación y edición de los contenidos.
6. Validar la solución propuesta.

A partir de los objetivos trazados se plantea la siguiente **idea a defender**: La integración de los instrumentos de evaluación existentes en la plataforma educativa Zera y la estandarización del portafolio actual, facilitaría el análisis del desempeño de los estudiantes en el marco educativo a través de las evidencias de aprendizaje.

**Para dar cumplimiento a los objetivos del trabajo se planificaron las siguientes tareas:**

- Investigación de las tendencias tecnológicas actuales y estándares educativos utilizados para un portafolio a nivel mundial.
- Investigación sobre la información que brindan los instrumentos de evaluación que pueda ser utilizada como evidencias de aprendizaje.
- Identificación y especificación de las funcionalidades.
- Realización de los casos de usos identificados.
- Realización del diagrama de clases del análisis.
- Realización del diagrama de clases del diseño aplicando los patrones de diseño.
- Modelación del diagrama de componentes del sistema.
- Modelación del diagrama de despliegue del sistema.
- Implementación de las funcionalidades identificadas.
- Realización de pruebas de calidad a lo largo de todo el ciclo de desarrollo.

## **Métodos investigativos**

Para llevar a cabo la investigación, se emplean los siguientes métodos:

### **Métodos teóricos**

- Análisis histórico – lógico: el uso de este método permitirá realizar el estudio del arte, o sea, para investigar las tendencias tecnológicas actuales y estándares educativos utilizados para un portafolio a nivel mundial e investigar sobre la información que brindan los instrumentos de evaluación que pueda ser utilizada como evidencias de aprendizaje.

- Modelación: este método se empleará para realizar los casos de usos identificados que dan cumplimiento a los requisitos funcionales y no funcionales de la solución, así como la realización de diferentes diagramas tales como: diagrama de clases del análisis, diagrama de clases del diseño para cada caso de uso aplicando en ellos los patrones de diseño, el diagrama de componentes y de despliegue del sistema.

Para el desarrollo del marco teórico y la definición de los temas vinculados con los portafolios electrónicos se utilizó el proceso de análisis y síntesis y el inductivo- deductivo para la descripción e implementación de las funcionalidades propuestas.

## **Métodos empíricos**

A través de la observación se pudieron recoger datos importantes durante el desarrollo de la solución. Además se utilizó la técnica de la entrevista con el líder de proyecto para conocer lo que deseaba el cliente y para obtener su aprobación de la solución propuesta.

## **Resultados esperados:**

Con la realización del presente trabajo de diploma se pretende crear las bases para mejorar las herramientas de gestión de las evidencias en la plataforma educativa Zera. Con la integración de los instrumentos de evaluación, se procura mejorar el análisis de los resultados de los estudiantes y permitir la interoperabilidad de los datos almacenados con otras plataformas. De esta forma se pretende influir positivamente en la portabilidad y usabilidad de la plataforma educativa Zera.

**El presente documento consta de cuatro capítulos**, desarrollados a partir del estudio realizado. La descripción de los mismos se muestra a continuación:

## **Capítulo I: Fundamentación teórica:**

En este capítulo se hace referencia a los elementos teóricos que soportan la investigación, se presentan los lenguajes de programación y modelado, así como tecnologías y metodologías que se ajustan al desarrollo del trabajo, fundamentando su selección en base al estudio realizado. Se describen las soluciones similares existentes y se realizan comparaciones para llegar a la propuesta de solución que sustenta la investigación.

## **Capítulo II:** Características del sistema:

En este capítulo se realiza una descripción de las características del sistema a implementar. Además se especifican los requisitos funcionales y no funcionales y se realizan los diagramas: Modelo de dominio, Diagrama de Actores y Casos de usos del sistema.

## **Capítulo III:** Análisis y Diseño del sistema:

Se modelan los diagramas de clases del análisis y del diseño para cada caso de uso de la propuesta del sistema a desarrollar. Además se elaboran los diagramas de paquetes del análisis y diseño, los diagramas de interacción del análisis, el diagrama de despliegue y el diseño de la base de datos.

## **Capítulo IV:** Implementación y Prueba:

Se desarrolla la implementación, utilizando el lenguaje de programación y la metodología seleccionada, documentándose todo el proceso. Se definen los tipos de pruebas, se diseñan los casos de prueba que se le aplicarán al sistema y se muestran los resultados de los diferentes tipos de pruebas realizadas.

## Capítulo I: Fundamentación teórica

### Introducción

El primer paso para comenzar una investigación es realizar el estudio del estado del arte. En el caso de la presente investigación es muy importante conocer cómo se comportan los instrumentos de evaluación en otras plataformas educativas para determinar una propuesta de solución para la plataforma educativa Zera, centrándose principalmente en el portafolio electrónico ya que este será el encargado de almacenar toda la información que generan los restantes instrumentos de evaluación que se analicen. Para determinar cómo desarrollar la investigación es necesario además hacer un estudio de las tecnologías, metodologías de desarrollo, herramientas y lenguajes tanto de desarrollo como de modelado a emplear para implementar la solución de dicho estudio.

#### 1.1. Valoración de los conceptos asociados al dominio del problema

##### Evidencias de aprendizaje

Se entiende por “**evidencia**” al conjunto de pruebas que demuestran que se ha cubierto satisfactoriamente un requerimiento, una norma o parámetro de desempeño, una competencia o un resultado de aprendizaje.

(4)

La evidencia es un conjunto de elementos tangibles que nos permiten demostrar que se ha logrado cubrir de manera satisfactoria un requerimiento o un criterio específico de desempeño, una competencia o bien el resultado de un aprendizaje. (5)

Es la manifestación del aprendizaje, que se refiere a la comprobación de lo que sabe, sabe hacer y es el aprendiz. Pueden ser de conocimiento y de desempeño, de las cuales se pueden inferir los logros de aprendizaje y establecer el desarrollo o no de las competencias. (6)

En sentido general la evidencia es el elemento que demuestra lo que sabe hacer y es el individuo, pueden ser tanto de desempeño, como de comportamiento, deduciendo de ella los logros de aprendizaje que el estudiante adquiere. Cada evidencia es respaldada por un conjunto de pruebas que garantizan la validez de la misma, un ejemplo de ello es la calificación.

## Evaluación

La evaluación del aprendizaje es un proceso consustancial al desarrollo del proceso docente educativo. Tiene como propósito comprobar el grado de cumplimiento de los objetivos formulados en los planes y programas de estudio de la educación superior, mediante la valoración de los conocimientos y habilidades que los estudiantes van adquiriendo y desarrollando; así como, por la conducta que manifiestan en el proceso docente educativo. Constituye, a su vez, una vía para la retroalimentación y la regulación de dicho proceso. (7)

### Instrumentos de evaluación (IE)

Los instrumentos, como herramientas utilizadas para recolectar información nos ayudan a la medición, la cual constituye una actividad presente en la práctica docente, ésta a su vez nos aproxima al monitoreo y evaluación del proceso educativo. (8)

Los instrumentos se constituyen en el soporte físico que se emplea para recoger la información sobre los aprendizajes esperados de los estudiantes. Todo instrumento provoca o estimula la presencia o manifestación de los que se pretende evaluar. Contiene un conjunto estructurado de ítems los cuales posibilitan la obtención de la información deseada. (9)

Se pueden definir a los instrumentos de evaluación como los recursos que se emplean para recolectar y registrar información sobre un proceso, en el caso del presente trabajo se enfoca al proceso de enseñanza - aprendizaje.

### 1.2. IE identificados en Zera

En la literatura educativa, la recogida de evidencias del aprendizaje se asocia al uso de los instrumentos de evaluación y en este contexto se entremezclan instrumentos e indicadores<sup>4</sup> de evaluación que pretenden diagnosticar cuánto ha aprendido el estudiante. Existe una amplia diversidad de instrumentos de evaluación a utilizar, de manera que, además, de servir como diagnóstico de lo aprendido, son utilizados como una retroalimentación para el proceso de aprendizaje.

En el caso específico de la plataforma educativa Zera se identificaron los siguientes instrumentos de evaluación: Reactivos, Evidencias, Portafolio, Foro, Softarea, Ejercicios y Enfrenta retos. Los tres primeros

---

<sup>4</sup> El concepto que brinda la Real academia española plantea que la palabra **indicador** sirve para indicar o señalar algo.

a pesar de poseer deficiencias, sí son utilizados actualmente como instrumentos de evaluación, pero los restantes, a pesar de serlo, no se utilizan.

Es importante señalar que la nomenclatura de los instrumentos de evaluación: Reactivos, Softarea y Enfrenta retos fue definida por pedagogos del Ministerio de la Educación de Cuba que trabajan conjuntamente con el proyecto. En el caso de Evidencias, fue definido por los clientes.

## 1.2.2. IE que se emplean actualmente

### **Reactivos:**

Los reactivos son cuestionarios basados en los ejercicios brindados por la plataforma. La evaluación le es otorgada al estudiante de forma automática por el sistema al concluir su resolución.

Un dato relevante es que el resultado que genera cada reactivo, no se almacena en el portafolio ni le permite al docente emitir una observación sobre su resultado si lo estima necesario.

### **Evidencias**

Las “Evidencias” son orientaciones que se le asignan al estudiante para realizar una o varias actividades, que generan una respuesta (evidencia) a la tarea. En este caso se utiliza la definición de evidencia de aprendizaje mencionada en el epígrafe 1.1 de este capítulo, pero la nomenclatura de la actividad tiende a confundir, pues la evidencia es lo que adjunta el estudiante y la actividad en sí sería una asignación de una tarea investigativa.

Este instrumento actualmente, además de lo anteriormente mencionado, no brinda la posibilidad de relacionar dicha asignación, con los capítulos, temas y subtemas que existen en el programa de estudio, que son los que pueden o no, estar asociados a conocimientos específicos<sup>5</sup>. Además, al docente no le posibilita seleccionar alguna “evidencia” creada, lo cual provoca que cada vez que desee realizar una asignación deba crearla nuevamente. En general, de este instrumento se puede obtener una evidencia del aprendizaje del estudiante, su calificación correspondiente y una observación que realiza el docente cuando emite dicha calificación.

### **Portafolio**

Actualmente el portafolio de la plataforma educativa Zera cuenta con cuatro espacios:

---

<sup>5</sup> **Conocimientos específicos:** unidad básica o mínima que conforma el sistema de conocimientos para una clase determinada.

- Notificaciones, donde se muestra un listado de todas las notificaciones asociadas a las evaluaciones y/o asignaciones realizadas al estudiante. Las mismas se muestran en forma de oración y además se le brinda la posibilidad al estudiante de eliminarlas.
- Calificaciones, donde se muestran solamente las calificaciones asociadas a la evidencia que adjunta el estudiante, que están asociadas a las asignaciones de “Evidencias”.
- Evidencias compartidas, donde se almacenan todos los trabajos ya evaluados que le son compartidos al estudiante. Cuenta con un sistema de consulta y además le permite al estudiante gestionar comentarios para los mismos.
- Archivo de evidencias, donde se muestra un listado de aquellas evidencias evaluadas (solamente las evidencias que se generan con el instrumento de evaluación “Evidencias”).

## 1.2.2. IE que no se emplean

### **Foro:**

Actualmente el foro en la plataforma educativa Zera es un espacio colaborativo para la comunicación y el debate, al docente no se le brinda la posibilidad de evaluar el desempeño de sus estudiantes, lo que provoca que en el caso de alcanzar algún logro de aprendizaje no se le tome en cuenta.

Dentro de los indicadores que el docente pudiera utilizar para emitir una evaluación y proporcionar una evidencia se encuentran: el tiempo que emplea un estudiante en el foro, la calidad de respuestas que emite y la cantidad de intervenciones que realiza el estudiante (en este indicador se valora la cantidad de preguntas y respuestas expuestas). (10)

### **Softarea:**

La Softarea es una tarea que el estudiante debe hacer, apoyada con materiales que están en la plataforma, es la unión de un recorrido dirigido<sup>6</sup>, uno o varios reactivos y la asignación de una o varias “evidencias”. Estas dos últimas generan una evaluación independiente, pero la Softarea en sí, no emite una evaluación general. Además, este espacio no se encuentra relacionado con el cumplimiento de

---

<sup>6</sup> Recorrido dirigido: constituyen rutas de aprendizaje creados por el profesor a partir del contenido oficial de la materia, ejercicios y recursos mostrados en la biblioteca. Permite estructurar el contenido seleccionado mediante el uso de plantillas previamente establecidas.

conocimientos específicos, lo que provoca que si el estudiante vence alguno realizando una Softarea, no se le tenga en cuenta. Además de que no le brinda la posibilidad al docente de emitir una observación del desempeño del estudiante en dicha actividad.

## Enfrenta retos:

Este espacio cuenta con tres secciones: Investiga y aprende<sup>7</sup>, Aprende a resolver<sup>8</sup> y Valora tus posibilidades<sup>9</sup>. Todas son actividades de práctica, lo que indica que no generan una evidencia dentro de la plataforma. A pesar de que actualmente se tienen en cuenta de esta forma, es importante señalar que las evidencias relacionadas con ellos, que no se gestionan, pueden ser el resultado del avance del desempeño y de los conocimientos que va adquiriendo el estudiante. Además como son orientaciones de actividades, podrían verse relacionado con la gestión de las “Evidencias”, o sea, además de ser actividades de práctica pueden formar parte de un banco de tareas para asignar y evaluar si el docente, el administrador central y editor lo estiman conveniente.

## Ejercicios:

Es un espacio en el módulo Prácticas que al igual que los Enfrenta retos se utiliza para practicar el conocimiento adquirido. Son una colección de ejercicios que demuestra no solo el avance en el conocimiento, también permite establecer un control del interés y desempeño que el estudiante posee. Actualmente este instrumento, al estar relacionados con los capítulos, temas y subtemas del programa de estudio, presenta la posibilidad de estar ligado a conocimientos específicos, generan una calificación por cada ejercicio realizado de la colección (no general) y al docente le muestra la evidencia del mismo.

## Tabla resumen sobre los IE identificados en Zera

- 
- 7 **Investiga y aprende:** Es un tipo de ejercicio abierto que debe comportarse de la misma forma que una evidencia, la diferencia con esta es que podrían no ser evaluativos y si formativos.
- 8 **Aprende a resolver:** Problemas propios de una disciplina que pueden o no, ser evaluados mediante Reactivos y preparan al estudiante para el desarrollo de habilidades específicas y así preparase para exámenes finales y otro tipo de evaluaciones.
- 9 **Valora tus posibilidades:** Recurso que permite al estudiante realizar pruebas o cuestionarios aplicados para: optar por becas, ingreso a universidades, comprobación de conocimientos, pruebas finales de la disciplina, y otros.

	Información que brindan actualmente			
	Calificación	Ligado a conocimientos específicos	Evidencia	Observaciones
Reactivos	x	x		
“Evidencias”	x		x	x
Foro				
Softarea	De forma independiente	De forma indirecta y parcialmente	x	
Enfrenta retos				
Portafolio	x		x	
Ejercicios	De forma independiente	x	x (Para el docente)	

*Tabla 1. Información que brindan los IE identificados en Zera*

Para comprender la tabla expuesta, es necesario resaltar que el término “De forma independiente” se refiere a que dentro de dicho instrumento existen evaluaciones pero en sí no emiten una calificación general. Y en el caso de la Softarea, el hecho de que los conocimientos específicos se encuentran ligado indirectamente, es debido a que los reactivos sí se encuentran y parcialmente porque con las “evidencias” no suceden así.

De los datos expuestos anteriormente se concluye que los instrumentos identificados, en su totalidad, no aportan la información necesaria (la expuesta en la Tabla 1) para ser utilizada como evidencias del aprendizaje.

## 1.2. Panorama internacional sobre los IE identificados en Zera.

Hasta el momento el estudio se ha centrado en los conceptos fundamentales que se relacionan con los instrumentos de evaluación y cómo estos se ven reflejados en la plataforma educativa Zera. Dichos instrumentos son herramientas que encuentran su aplicación en varias plataformas conocidas.

Para realizar un estudio sobre el comportamiento de algunos de los instrumentos de evaluación identificados, se seleccionaron las plataformas a analizar durante toda la investigación. Para ello se tuvieron en cuenta una serie de documentos, desde el 2008 hasta la actualidad ((11) (12) (13) (14)) y del análisis realizado se identificaron las siguientes plataformas comunes: Moodle, Dokeos, Sakai, Claroline, Ilias, A Tutor, Blackboard, WebCT, Docebo.

Por la necesidad que existía de estudiar algunas plataformas educativas, el proyecto Alfaomega (proyecto que desarrolla la plataforma educativa Zera) decidió poner a disposición del equipo de trabajo las plataformas Sakai y Dokeos. Aprovechando la experiencia con la plataforma Moodle (actualmente utilizada en la Universidad de las Ciencias Informáticas) y la interacción directa con Sakai y Dokeos, se decidió realizar un estudio para determinar cómo utilizan la evaluación en los siguientes instrumentos de evaluación: los foros, los cuestionarios (Reactivos) y la asignación de tareas (“Evidencias”). El objetivo de dicho análisis es arribar a una propuesta que permita ser aplicada a estos, de manera que aporten evidencias de aprendizaje. Las restantes plataformas no fueron analizadas pues se consideró que la interacción directa es mejor que la búsqueda excesiva por internet tratando de encontrar información confiable sobre cómo se desenvuelven estas plataformas cuando de instrumentos de evaluación se refiere.

Moodle	Sakai	Dokeos
<b>Instrumentos de evaluación</b>		
<p><u>Foros:</u> Se organizan por hilos de discusión. El docente cuando crea un tema en el foro puede activar la posibilidad de calificar los mensajes, así como permitir que otros usuarios lo califiquen. Al ser calificables, el docente puede usar un foro como una actividad evaluable del curso. Una vez emitida dicha</p>	<p><u>Foros:</u> El docente puede crear un foro general y otros para temas diferentes pudiendo intervenir en ellos si lo considera oportuno. Los estudiantes pueden proponer cuestiones a debatir en el foro y responder a cuestiones planteadas por otros participantes. Estos mensajes sólo pueden leerlos los estudiantes miembros del foro</p>	<p><u>Foros:</u> Permite la calificación, pero no de los comentarios. El docente cuando crea un tema, se le brinda la posibilidad de categorizarlo como él desee, según las categorías que existan almacenadas y las que él cree. En sentido general, esta plataforma le permite al docente calificar el desempeño general</p>

<p>calificación, automáticamente, junto con la opinión del estudiante, se almacena en un módulo llamado Calificaciones.</p>	<p>(puede ser toda la clase o un grupo de estudiantes). No son calificables.</p>	<p>del estudiante en ese tema y modificar dicha calificación tantas veces lo considere necesario.</p>
<p><u>Tarea:</u> Es el espacio en el que se orienta una tarea para que el estudiante la realice y adjunte una evidencia (en cualquier formato de archivo). Al realizar esta acción por parte del estudiante el docente emite una calificación y una observación que se adjuntan a la página de la tarea de cada estudiante. Se le envía un mensaje de notificación al estudiante y además el docente tiene la posibilidad de permitir el reenvío de la tarea tras su calificación.</p>	<p><u>Tarea:</u> Permite poner a disposición de los estudiantes, los trabajos y prácticas de la asignatura, así como corregirlos y calificarlos. El docente crea una tarea, establece las normas de realización y, si lo desea, adjunta un archivo con el enunciado correspondiente. Una vez disponible, los estudiantes realizan la tarea (directamente en Sakai o en su ordenador, según lo defina el docente) y se lo envían para que lo corrija.</p>	<p><u>Tarea:</u> Permite al docente realizar una descripción de la misma y configurarla de acuerdo a como estime conveniente. Automáticamente el sistema la asigna a los estudiantes de los grupos a los que le imparte clases, sin opciones para seleccionar a qué estudiantes asignarla. El estudiante adjunta la respuesta de la misma y posteriormente el docente la califica.</p>
<p><u>Cuestionarios:</u> En este espacio, al estudiante una vez terminado el cuestionario, se le muestra automáticamente la calificación obtenida (no en todos los casos es así, pues el docente puede modificar los datos que se muestran) y en caso de que él lo determine se mostrará un mensaje acorde a dicha</p>	<p><u>Cuestionarios:</u> Los cuestionarios no proporcionan ninguna calificación, pero sí datos estadísticos sobre las respuestas. El estudiante los contesta de forma anónima.</p>	<p><u>Cuestionarios:</u> Existe un módulo llamado Ejercicios donde se gestionan ejercicios (en forma de cuestionarios) de diferentes tipos. Al igual que los foros, el sistema le brinda la posibilidad de seleccionar el tipo de categoría del ejercicio, decidir su máximo puntaje y enviarlo a todos sus estudiantes sin</p>

<p>calificación. Además le permite al estudiante, si el docente lo considera necesario, poder observar las respuestas que emitió y saber cuáles son correctas y cuáles incorrectas, sin posibilidad de modificarlas. Además le brinda la posibilidad de establecer la cantidad de intentos a realizar por el estudiante.</p>		<p>brindarle la posibilidad de seleccionar los que el docente determine que deban realizar dicho cuestionario. El sistema lo califica automáticamente. Se le permite al estudiante un único intento.</p>
--	--	--

*Tabla 2. Comportamiento de los IE identificados (en la plataforma educativa Zera) en distintas plataformas*

De la tabla resumen se concluye que los foros pueden convertirse en instrumentos de evaluación que aportan evidencias del aprendizaje del estudiante. Se observa que el comportamiento de la asignación de tareas (“Evidencias”) y de los cuestionarios (Reactivos), a pesar de ser similar en las tres plataformas, pueden ser gestionadas de formas diferentes, aportando de esta manera, otros elementos a considerar en su utilización en la plataforma educativa Zera.

Es importante señalar que no se analizaron los Enfrenta retos, Ejercicios y las Softareas pues son elementos conceptuales que solamente se manejan en la plataforma educativa Zera. En el caso del Portafolio, a continuación se abordarán aspectos relacionados con él, pues es el instrumento de evaluación más importante sobre el que se estará abordando durante la investigación, ya que en él se almacenan todas las evidencias de aprendizaje que generan los restantes identificados.

### **1.3. Relación de los instrumentos de evaluación con el portafolio**

Para almacenar el cúmulo de evidencias que generan los instrumentos de evaluación se han creado diferentes espacios, algunos de ellos son nombrados como: Resultados, Calificaciones o Portafolio. Este último es el que actualmente utiliza la plataforma educativa Zera.

A pesar de que en ella existe este portafolio y a partir de las deficiencias detectadas anteriormente, conjuntamente con las características que debe poseer un portafolio electrónico, se puede afirmar que lo desarrollado actualmente no cumple con las características para ser categorizado como se define. A continuación se justifica esta afirmación.

## 1.4. Portafolio electrónico

El portafolio electrónico, también conocido como Portafolio digital, e-Portafolio o Webfolio, es la versión digital de lo que es un portafolio tradicional con características y ventajas adicionales. Muchos autores han definido al portafolio electrónico, a continuación se citan tres definiciones, que para los autores del presente trabajo son las más completas:

Los portafolios electrónicos han surgido como una valiosa herramienta en línea que los estudiantes, la facultad y las instituciones pueden utilizar para recolectar, almacenar, actualizar y compartir información. Permiten que los estudiantes reflejen su aprendizaje, se comuniquen con sus instructores, tengan información escolar y muestren ejemplos de su trabajo a empleadores potenciales. (15)

Un portafolio electrónico es una colección de materiales digitalizados que incluyen demostraciones, recursos y logros que representan a un individuo, un grupo o una institución. Puede comprender texto, gráficos o elementos multimedia que se pueden consultar en Internet o en otro medio electrónico como un CD-ROM o un DVD. Es más que una simple colección, también sirve como herramienta para administrar, organizar y controlar los trabajos creados con diferentes aplicaciones. Incluyen reflexión personal y por lo regular promueven el intercambio de ideas y retroalimentación. (16)

Un portafolio electrónico es una colección de evidencias que, no solamente resumen el trabajo académico de un estudiante, sino que incluye procesos de aprendizaje individual, describen procesos metacognitivos individuales y procesos socioafectivos grupales, presentan juicios de evaluación acerca del desempeño integral, valoran el logro de objetivos y el desarrollo de competencias y establecen metas futuras de desarrollo personal y profesional. (17)

A partir de las tres definiciones expuestas anteriormente, se puede afirmar que un portafolio electrónico se puede definir como una galería de experiencias que tiene muchas potencialidades, una de las que más resalta es que en su totalidad constituye un aval que posee el individuo de las competencias que adquiere. De esta forma, cada portafolio es único y muestra al individuo que lo asume tal y como es.

## 1.4.1. Tipos de portafolio electrónico

Existen cuatro tipos: (18)

- Portafolio de Evaluación: Permite valorar la consecución de criterios específicos para obtener una titulación o trabajo.
- Portafolio de Aprendizaje: Permite ofrecer información sobre los objetivos de aprendizaje incorporando tanto reflexiones y autoevaluación del estudiante como del docente.
- Portafolio de “Demostración de las mejores prácticas”: Permite presentar información o logros a audiencias concretas.
- Portafolio de Transición: Permite aportar evidencias y registros de utilidad en momentos de transición o paso de un nivel académico a otro.

En dependencia del tipo de audiencia puede ser clasificado en: portafolio del estudiante y del docente.

Según la JISC-CETIS (Organización encargada de los estándares para la educación en el Reino Unido) existe otro tipo de portafolio definido con el nombre de portafolio de desarrollo personal. Un portafolio para la planificación del desarrollo personal contiene registros de aprendizaje, el desempeño, los logros y los resultados de esa reflexión, incluidos los planes para el desarrollo futuro. Esto podría incluir un portafolio de aprendizaje, pero va más allá de eso, ya que suele estar relacionada con el desarrollo profesional y el empleo, por lo que también probablemente es utilizada como un portafolio de presentación.

Un portafolio de presentación (...) “muestra los mejores trabajos del estudiante, escogidos bajo su criterio, que debe justificar su elección. De este modo el estudiante aprende a ser crítico con su trabajo y a autoevaluarse. Al mostrar sus mejores trabajos, en una presentación o en la web, el estudiante mejora su autoestima y aumenta su motivación”. (19)

## 1.4.2. Estructura de un portafolio electrónico

Un portafolio electrónico se divide en tres espacios o secciones (17):

- ⤴ **Espacio personal:** Es donde el estudiante plasma libremente todo lo que quiera acerca de su persona, sus intereses, gustos, entre otros; la idea es que en este espacio se pueda reflejar parte de su personalidad.
- ⤴ **Espacio profesional:** Es donde el estudiante plasma sus actividades extracurriculares, tiene un espacio para hablar de las prácticas profesionales que realizó, y de igual forma, si está desarrollando una tesis, puede incluir la sinopsis de esta en el espacio.

- ▲ **Espacio académico:** Este es el lugar que va más relacionado con las materias que se haya cursando; la información que se encuentra plasmada es la que tiene que ver con las asignaturas.

### 1.4.3. Panorama nacional de los portafolios electrónicos

En Cuba, a pesar de que se han realizado investigaciones sobre los portafolios electrónicos, no existe un uso masivo del mismo. La solución informática más cercana se encontró en la Universidad de las Ciencias Informáticas: el Enterprise Content Management (ECM) Alfresco en su versión “Community”, que a pesar de no ser categorizado como un portafolio electrónico posee características similares.

Alfresco es una herramienta utilizada por los docentes y administradores de la universidad para controlar y gestionar grandes volúmenes de documentación, además de brindar varias alternativas para buscar cualquier información contenida en ellos. Posee además, un perfil del usuario, opciones para cambiar la apariencia, la opción para crear nuevos espacios y contenidos, las opciones de Copiar y Pegar todo y un Espacio personal donde se encuentra toda la documentación, estructurado por tipos de expedientes (creados por el administrador), que cada uno de ellos gestiona de sus estudiantes. Y además permite la opción de importar como .acp o .zip.

Una vez analizada la situación nacional se continuará el estudio basado en el contexto de los portafolios electrónicos a nivel internacional.

### 1.4.4. Panorama internacional de los portafolios electrónicos

Con el auge de los portafolios electrónicos surgieron deficiencias relacionadas con la interoperabilidad, usabilidad y portabilidad de los mismos. Para lograr que el que desee implementarlo siga una misma línea de desarrollo, surgieron los estándares para la creación de portafolios que permiten el almacenamiento de la información para su procesamiento, mantenimiento e interoperabilidad en distintos sistemas de gestión del aprendizaje.

Cuando se hace referencia a portafolios electrónicos, los tres estándares más conocidos son: IMS Learner Information Package (LIP), IMS e-Portafolio y Leap2A.

A continuación se seleccionarán las plataformas más utilizadas que tienen incorporado un portafolio electrónico y algunas de las herramientas que se utilizan para su creación. Todo esto persigue como objetivo: realizar un análisis de cómo exportan los datos de los estudiantes, para arribar a una conclusión

y determinar cuál de los estándares utilizar en la nueva versión del portafolio de la plataforma educativa Zera.

## Selección de las plataformas educativas a analizar

Teniendo como precedente la selección de las plataformas a analizar durante toda la investigación, realizada en el epígrafe 1.4, se elaboró el siguiente cuadro resumen.

Plataformas seleccionadas	¿Posee un portafolio?	Tipos de Software
Moodle	Posee una herramienta para su creación.	Libre
Sakai	Si	Libre
Dokeos	No	Libre
Claroline	No	Libre
Ilias	Si	Libre
A Tutor	Si	Libre
Docebo	Si	Libre
Blackboard	Si	Propietario
WebCT	Si	Propietario

*Tabla 3. Plataformas seleccionadas para el análisis del uso de los portafolios electrónicos*

De la tabla comparativa se seleccionaron las siguientes plataformas a analizar: Moodle, Sakai, Ilias, A Tutor, Blackboard, WebCT y Docebo. Se descartó la posibilidad de estudiar a Dokeos y Claroline por la inexistencia de un portafolio en ellas.

## Selección de las herramientas para la creación de portafolios electrónicos

Existen numerosas herramientas para la creación de un portafolio electrónico, entre los que se encuentran: Exabis, MyStuff, SPDC Portafolio, Mahara, Pebble Pad, My progressfile, ePet y MyKnowledgeMap. Cada una de estas herramientas comparten características comunes que se resumen en: recolección, reflexión, organización y compartición de evidencias.

## Selección del estándar a implementar para el portafolio de Zera

Una vez seleccionadas las plataformas educativas y las herramientas para la creación de portafolios electrónicos, se elaboró el siguiente cuadro comparativo.

	Exporta a:				
	IMS LIP	IMS e-Portafolio	Leap2A	HTML (.zip)	Aspectos de interés
<b>Plataformas</b>					
Moodle 2.1.2					Utiliza la herramienta Exabis.
Sakai				x	
A Tutor					Utiliza la herramienta My Stuff.
Ilias				x	Planifican exportar a Leap2A o PDF.
Blackboard				x	Además exporta a un estándar que definieron: D2L.
WebCT				x	
Docebo					No exporta
<b>Herramientas</b>					
Mahara			x	x	
Exabis					Exporta a SCORM
My Stuff				x	Además exporta a RTF.
SPDC					No exporta
Pebble Pad			x		
ePet			x		
My progressFile			x		
MyKnowledgeMap			x		

Tabla 4. Comparación para inferir qué estándar implementar.

Del análisis realizado se concluye que el estándar más utilizado por todas las plataformas y herramientas seleccionadas es el Leap2A.

Una vez determinado qué estándar desarrollar se detallarán las características del mismo.

## 1.4.5. IMS Leap2A

Esta familia de especificaciones se destina a cubrir la representación de varios tipos de información, centrada en los individuos, que recogen, crean, reflexionan y utilizan su propia información para el aprendizaje, el desarrollo, la auto-presentación, o con fines relacionados. (20)

Esta información suele ser descrita o recogida por los propios individuos, pudiendo cubrir: lo que han hecho, logrado, escrito, o se sienten orgullosos, qué o a quién ayuda o les ha ayudado, lo que aspiran, en lo que son buenos, evidencias y reflexiones sobre cualquiera de estos y tal vez comentarios de otras personas. (20)

Los materiales de aprendizaje a menudo tienen restricciones de derechos de autor, pero en general la información de la cartera puede ser utilizada libremente por la persona que crea la cartera o tenido acceso a ella. (20)

No es el propósito de estas especificaciones cubrir la información recogida por los demás sobre un individuo, sobre la cual dicha persona no tiene fácil acceso o control.

Esta especificación utiliza el formato de sindicación Atom<sup>10</sup>. Atom fue elegido como la base para Leap2A con el argumento de que hubo un ajuste razonable entre el átomo y la información de la cartera y debido a que Atom se aplica ampliamente en muchos sistemas. (20)

Leap2A ofrece un modelo para representar la información y los recursos creados, controlados o recogidos por las personas, descripciones de lo que han logrado, creado, hecho o experimentado; información sobre sí mismos, sus habilidades y cualidades; cualquier tipo de información de apoyo o pruebas de cualquier fuente, sus pensamientos o reflexiones sobre lo pasado, presente o futuro, sus planes y las presentaciones de las selecciones de esta información a otras personas. (20)

Leap2A especifica una estructura de pequeñas unidades independientes (los elementos) de información y permite describir y referenciar recursos que no necesariamente tienen que ser archivos, a través de un conjunto de relaciones entre elementos de información y entre los elementos y los recursos que

---

<sup>10</sup> Atom es un formato de documento basado en XML que describe la lista relacionada a la información conocida como "feeds". Los feeds se compone de una serie de elementos, conocidos como "entradas", cada uno con un conjunto ampliable de adjuntos metadatos. El caso de uso principal que se ocupa de Atom es la sindicación de Web contenidos como weblogs y titulares de noticias a los sitios Web, así como directamente a los agentes de usuario. (21)

representan y se refieren. No se especifica ni se requiere una estructura para los recursos que no son archivos. (20)

Leap2A también detalla un conjunto de categorías que se pueden usar en elementos de Leap2A y permite la aplicación de otras categorías definidas por individuos u organizaciones. (20)

## **Modelo de datos del Leap2A**

Este estándar posee un modelo de datos en el cual se evidencia su flexibilidad, pues un atributo puede relacionarse con los restantes. Para profundizar los datos que maneja el estándar Ver [Anexo 1](#): Modelo de datos del Leap2A.

**Licencia:** Ver [Anexo 2](#) Dictamen emitido por la universidad de la Licencia del Leap2A.

## **1.5. Ambiente de desarrollo**

Obtener un software con la calidad requerida, implica la utilización de metodologías, tecnologías y herramientas que permitan conformar el área de trabajo. Para comenzar el desarrollo de la plataforma educativa Zera se realizó un estudio de las tecnologías y tendencias de metodologías, lenguajes y herramientas de modelado, frameworks, servidores web, lenguajes de desarrollo y gestores de bases de datos con el objetivo de definir el ambiente de desarrollo del mismo. Debido a que la solución que se desarrollará es parte de dicha plataforma, se decidió trabajar sobre el ambiente de desarrollo ya definido. A continuación se detallarán las características del mismo.

### **1.5.1. Metodología de desarrollo de software**

Las metodologías de desarrollo de software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto de software. (22)

Existen dos tipos de metodologías: las ágiles y las tradicionales. Esta última es la que utiliza la plataforma educativa Zera y (...) “se caracteriza por imponer una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente”. (23)

### **Proceso unificado de desarrollo (RUP)**

RUP (en inglés Rational Unified Process) provee un acercamiento disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de

software de alta calidad que satisfaga los requerimientos de los usuarios finales (respetando cronograma y presupuesto). (24)

## Características (24)

- Entrega una forma disciplinada de asignar tareas y responsabilidades: esto posibilita una mayor organización dentro del equipo de trabajo definiendo quién hace qué, cuándo y cómo.
- Su meta es asegurar la producción de software de alta calidad, que cumpla las necesidades de los usuarios.
- Utiliza el desarrollo iterativo para enfrentar el riesgo: con esto el énfasis de cada flujo de trabajo variará a través del ciclo de vida ayudando a mitigar los riesgos en forma temprana y continua y con un progreso demostrable.
- El desarrollo se guía por casos de uso: los casos de usos se especifican, se diseñan y son los que guían el desarrollo de lo que se implementará.
- Sigue los pasos intuitivos necesarios a la hora de desarrollar el software.

Con el empleo de RUP se garantiza principalmente la documentación de la solución y se expone un conjunto de actividades orientadas a especificar, construir y documentar los artefactos necesarios para el mismo, sirviendo esto para futuras versiones de lo que se desarrollará.

## **1.5.2. Lenguaje de modelado**

Un lenguaje proporciona un vocabulario y unas reglas para permitir una comunicación (25). El lenguaje de modelado se centra en la representación gráfica de un sistema informático conformando un conjunto de vistas, símbolos y reglas que ayudan a la confección de los diagramas que indicarán qué y cómo implementar el sistema.

## **UML**

El lenguaje unificado de desarrollo o UML (en inglés Unified Modeling Language) es el lenguaje en el que se basa la metodología seleccionada y sirve para el modelado completo de sistemas complejos, tanto en el diseño de los sistemas de software como para la arquitectura hardware donde se ejecuten. (25)

UML aporta las siguientes ventajas: (25)

- Mayor rigor en la especificación.
- Permite realizar una verificación y validación del modelo realizado.

- Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y a la inversa (a partir del código fuente generar los modelos).

En sentido general mediante este lenguaje se podrá especificar todas las decisiones tomadas en el análisis, diseño e implementación de la solución, construyéndose de esta forma modelos completos y permitirá documentar todos los artefactos del proceso de desarrollo. Se seleccionó la versión UML 2.0 como lenguaje de modelado.

### 1.5.3. Lenguajes de desarrollo

#### Lenguajes del lado del cliente

- **Lenguaje Extensible de Marcado de Hipertexto (XHTML):**

XHTML es una versión más estricta y limpia de HTML, que nace precisamente con el objetivo de reemplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML. XHTML extiende HTML 4.0 combinando la sintaxis de HTML, diseñado para mostrar datos, con la de XML, diseñado para describir los datos. Ante la llegada al mercado de un gran número de dispositivos, XHTML surge como el lenguaje cuyo etiquetado, más estricto que HTML, va a permitir una correcta interpretación de la información independientemente del dispositivo desde el que se accede a ella. XHTML, al estar orientado al uso de un etiquetado correcto, exige una serie de requisitos básicos a cumplir en lo que a código se refiere. Entre estos requisitos básicos se puede mencionar una estructuración coherente dentro del documento donde se incluirían elementos correctamente anidados, etiquetas en minúsculas, elementos cerrados correctamente, atributos de valores entrecomillados, etc. (26)

Se utilizará XHTML en su versión 1.0.

- **Hojas de Estilo en Cascada (CSS)**

CSS es el lenguaje para describir la presentación de páginas Web, incluidos los colores, el diseño, y las fuentes. Se permite una adaptación de la presentación a diferentes tipos de dispositivos, tales como pantallas grandes, pantallas pequeñas, o impresoras. CSS es independiente de HTML y se puede utilizar con cualquier lenguaje de marcado basado en XML. La separación de código HTML de CSS hace más fácil para mantener los sitios, las hojas de estilo a través de compartir páginas y páginas de adaptar a diferentes ambientes. Esto se conoce como la separación de la estructura o contenido de la presentación. (26) Se utilizará CSS en su versión 2.1.

## ○ **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, 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. (27)

## **Tecnología del lado del cliente**

### ○ **Ajax**

El término AJAX es un acrónimo de Asynchronous JavaScript + XML, que se puede traducir como “JavaScript asíncrono + XML”. No es una tecnología en sí mismo, en realidad se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes. Las tecnologías que la forman son (28):

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

En la presente investigación esta tecnología se utilizará principalmente para la comunicación asíncrona<sup>11</sup> de datos.

## **Lenguajes del lado del servidor**

### ○ **Hypertext Preprocessor (PHP)**

PHP o preprocesador de hipertexto (en inglés Hypertext Preprocessor) es un lenguaje interpretado de alto nivel, especialmente pensado para desarrollos web y el cual puede ser incrustado en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender. La meta de este lenguaje es

---

<sup>11</sup> Asíncrona: los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página.

permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil, aunque se pueda hacer mucho más con PHP. (29)

Este lenguaje se utilizará para interactuar con los datos de la aplicación y la comunicación con la base de datos. Se decidió emplear la versión 5.3.3.

- **XML (Lenguaje de Etiquetado Extensible):**

Es un lenguaje muy simple pero estricto, que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML permite la lectura de datos a través de diferentes aplicaciones que lo usen. (30)

Este lenguaje se utilizará para almacenar toda la información referente al estándar a implementar. Se empleará su versión 1.0.

## **1.5.4. Framework**

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 incompleta y configurable a la que se le puede añadir las últimas piezas para construir una aplicación concreta. Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar el código ya existente y promover buenas prácticas de desarrollo con el uso de patrones. Un framework Web, por tanto, se puede definir como un conjunto de componentes que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web. (31)

## **Capa de Presentación**

Dentro del framework de la capa de presentación relacionado con javascript se utilizará jQuery en su versión 1.5. jQuery es una biblioteca de javascript rápida y concisa que simplifica el recorrido de un documento HTML, de manejo de eventos, animaciones e interacciones Ajax para el desarrollo web rápido. (32) Estas características posibilitan que los contenidos se puedan mostrar y se interactúe con estos de forma dinámica, pudiéndose usar este framework para apoyar el cumplimiento de las funcionalidades necesarias.

Otro framework que se tuvo en cuenta en la capa de presentación fue el framework para el desarrollo de CSS. Este es una herramienta muy útil que mejora el tiempo de desarrollo en la Capa Presentación de los estilos. Se trata de una serie de declaraciones de estilos definidos de forma estándar para que sirvan como base para el diseño de todo tipo de webs (31). En sentido general un framework CSS se utiliza para reiniciar los estilos predefinidos por el navegador facilitando que la aplicación pueda verse igual en la gran mayoría de los navegadores e incluye otras características como pueden ser: estilos predefinidos para tablas, enlaces, input, entre otros.

- **BlueprintCSS (Versión 1.0):** es un framework CSS, que tiene como objetivo reducir su tiempo de desarrollo. Da una base sólida para construir un proyecto, con una red de fácil uso, la tipografía sensible, plugins útiles, e incluso una hoja de estilo para la impresión.(33)

## Capa de Lógica de Negocio

Symfony es el framework utilizado para optimizar el desarrollo de aplicaciones web, desarrollado en el lenguaje del lado del servidor PHP y que separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación. Este framework presenta las siguientes características: (34)

- Implementa el patrón Modelo Vista Controlador o MVC (en inglés, Model View Controller).
- Proporciona una estructura al código fuente, forzando al desarrollador a crear código más legible para un futuro mantenimiento.
- Encapsula operaciones complejas en instrucciones sencillas.
- Reutiliza código.
- Se puede ejecutar tanto en plataformas \*nix (Unix, Linux, etc.) como en plataformas Windows
- Presenta un CRUD (Crear, Obtener, Actualizar y Borrar o Create, Retrieve, Update y Delete en inglés) integrado para la interacción con la base de datos.
- Usa Programación Orientada a Objetos.
- Son independientes de base de datos que posibilita soportar varios gestores como MySQL, Oracle y Postgres, siendo importante para si en un futuro se cambia el gestor.
- Soporta integración con el ORM (Mapeador de Objetos Relacional) Doctrine y con los frameworks de la capa de presentación jQuery y Blueprint.

El empleo de Symfony facilita la retroalimentación, superación y reparación de errores debido a que la comunidad de usuarios y documentación es muy amplia, los usuarios proveen diversos plugins que pueden ser reutilizables y esa es una de sus grandes fortalezas. Se decide utilizar su versión 1.4.15.

## Capa de Acceso a Datos

Un Mapeador de Objeto Relacional o ORM (en inglés Object Relation Mapper) es una tecnología de programación que permite convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, es decir, las tablas de la base de datos pasan a ser clases y registros, objetos que se manejan con facilidad (35). Utilizar un ORM tiene una serie de ventajas que facilitan enormemente tareas comunes y de mantenimiento (36):

- **Reutilización:** la principal ventaja que aporta un ORM es la reutilización permitiendo llamar a los métodos de un objeto de datos desde distintas partes de la aplicación e incluso desde diferentes aplicaciones.
- **Encapsulación:** la capa ORM encapsula la lógica de los datos pudiendo hacer cambios que afectan a toda la aplicación únicamente modificando una función.
- **Portabilidad:** utilizar una capa de abstracción que permite cambiar en mitad de un proyecto de una base de datos MySQL a una Oracle sin complicación.
- **Seguridad:** los ORM suelen implementar mecanismos de seguridad que protegen la aplicación de los ataques más comunes como inyecciones SQL.
- **Mantenimiento del código:** gracias a la correcta organización de la capa de datos, modificar y mantener el código es una tarea sencilla.

Los ORM a utilizar es:

- **Doctrine en su versión 1.2.2:** es un ORM para PHP 5.2.3 y posterior. Además de todas las ventajas que conlleva un ORM, uno de sus puntos fuertes es su lenguaje inspirado en el HQL de Hibernate, uno de los mejores ORM. Soporta migraciones, eventos, paginación e interfaces de líneas de comando. (35)

El empleo de Doctrine posibilitará un trabajo más rápido y eficaz, tanto es así que en la versión de Propel 2.0 se usarán muchas características de Doctrine 2.0. Posee una amplia documentación lo que facilita la retroalimentación continua y presenta soporte para migraciones. Se decide utilizar la versión 1.2.2 de dicha ORM.

## 1.5.5. Sistema gestor de Base de Datos

Un Sistema Gestor de Base de Datos o SGBD (en inglés, DBMS: DataBase Management System) es un sistema de software de propósito general que facilita los procesos de definición, construcción y manipulación de datos para distintas aplicaciones. (37)

El SGBD utilizado es PostgreSQL en su versión 9.0. El mismo presenta las siguientes características (38):

- Funciona en los principales sistemas operativos, como Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows
- Tiene soporte completo para claves foráneas, uniones, vistas, disparadores y procedimientos almacenados.
- Es compatible con el almacenamiento de objetos binarios, incluyendo imágenes, sonidos y videos.
- Posee una extensa documentación de carácter excepcional.
- Es altamente escalable tanto en la enorme cantidad de datos que pueden manejar como en el número de usuarios concurrentes que pueden permitir, siendo necesario para un mejor funcionamiento de la plataforma.
- Brindan la posibilidad de replicar los datos de una base de datos a otra.

PostgreSQL cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación de confiabilidad, integridad de datos y corrección. Presenta una amplia documentación muy bien organizada, pública y libre, con comentarios de los propios usuarios siendo estos aspectos importantes a la hora de dar solución tanto a errores del propio gestor como a errores del trabajo con las bases de datos, así como para una mejor retroalimentación y superación en el uso de esta herramienta.

## 1.5.6. Servidor web

Un servidor web es un programa que gestiona cualquier aplicación en el lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando una respuesta en cualquier lenguaje o aplicación en el lado del cliente. (39)

Apache es el servidor web que se utiliza para el desarrollo de la plataforma Zera y la presente investigación utilizará su versión 2.0. Se caracteriza por ser de código abierto, multiplataforma, soportar PHP y contar con una alta comunidad de usuarios lo que posibilita que se dé solución a muchos errores que se puedan presentar durante el proceso de desarrollo.

## 1.5.7. Entorno de desarrollo

Un Entorno de Desarrollo Integrado o IDE (en inglés Integrated Development Environment) es un conjunto de programas que se ejecutan desde una única interfaz de usuario. Por ejemplo, los lenguajes de programación incluyen a menudo un editor de texto, compilador y depurador, que se activan todos y la función de un menú común de IDE. (40)

### NetBeans

NetBeans es de código abierto, soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Existe un número importante de módulos para extenderlo. Es un producto libre y gratuito sin restricciones de uso. Entre sus características se encuentra un sistema de proyectos basado en control de versiones y refactorización. (41)

Entre las características que proporciona se encuentran: (41)

1. Nos propone un esqueleto para organizar nuestro código fuente, el editor conjuntamente integra los lenguajes como HTML, JavaScript y CSS.
2. El IDE de NetBeans para PHP también ofrece la línea de comandos de depuración: La salida del programa PHP aparece en una pantalla de línea de comandos en el IDE de sí mismo y se puede inspeccionar el código HTML generado sin tener que cambiar a un navegador.
3. Posee un sistema para examinar todo los directorios de cada proyecto, haciendo reconocimiento y carga de clases, métodos y objetos, para acelerar la programación.
4. Gracias a NetBeans ya es posible dejar de lado la consola de comandos de Symfony y centrarse en desarrollar en el IDE, además se encuentra cargadas todas las clases, ayuda en línea, etc.
5. Su editor, sobre todo en el editor de PHP, es mucho más ágil y a la vez robusto, contiene más ayuda en línea, reconocimiento de sintaxis y todo lo que provee las versiones de PHP.
6. Integra muy bien la utilización Xdebug, gracias a esto podemos inspeccionar y examinar cada variable local, establecer puntos de interrupción y evaluar el código en nuestra lógica.
7. Desde el editor es posible realizar la administración de estos sistemas versionados, sus commit, branch, importar, exportar, revert, clonar, etc.

Se ha seleccionado NetBeans en su versión 7.1 para el desarrollo.

## 1.5.8. Herramientas CASE para el modelado UML

Con el auge que ha adquirido UML (Lenguaje Unificado de Modelado- Unified Modeling Language) en los últimos años, se han creado numerosas herramientas de modelado de diagramas UML, donde seleccionar la herramienta CASE (Ingeniería de Software asistido por Computadora –Computer Aided Software Engineering) para UML se ha convertido en un tarea muy complicada.

Se puede definir a las herramientas CASE como un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software. (42)

### Visual Paradigm

Visual Paradigm for UML (VP-UML) es una herramienta de diseño UML y herramienta CASE UML diseñado para ayudar al desarrollo de software. VP-UML soporta los principales estándares de la industria tales como Lenguaje de Modelado Unificado (UML), SysML, BPMN, XMI, etc ofrece un completo conjunto de herramientas de los equipos de desarrollo de software necesario para la captura de requisitos, software de planificación, la planificación de controles, el modelado de clases, el modelado de datos, etc. (43)

Está concebida para soportar el ciclo de vida completo del proceso de desarrollo del software. Dentro de sus características se destacan las siguientes (43):

- Posee un diseño centrado en casos de uso y enfocado al negocio generando así un software con mayor calidad.
- Disponible para múltiples plataformas.
- Posee capacidades de ingeniería directa e inversa.
- Soporta aplicaciones Web.
- Las imágenes y reportes que genera son de muy buena calidad.
- Su modelo y código permanece sincronizado en todo el ciclo de desarrollo.
- Está disponible en múltiples versiones, para cada necesidad.
- Es fácil de instalar y actualizar.
- Transforma los diagramas de Entidad- Relación en tablas de base de datos.
- Genera informes.
- Importación y exportación de ficheros XML.
- Permite la reorganización de las figuras y conectores de los diagramas UML.

Se ha seleccionado Visual Paradigm en su versión 8.0 como herramienta CASE de desarrollo.

## 1.6. Conclusiones parciales del capítulo

Luego del recorrido teórico presentado sobre las diferentes perspectivas encontradas al analizar el uso de los instrumentos de evaluación que existen en la plataforma educativa Zera y de los portafolios electrónicos en el sistema educativo, se concluye:

- Atendiendo al concepto de los instrumentos de evaluación, se identificaron los siguientes en la plataforma educativa Zera: los Foros, Reactivos, Evidencias, Softarea, Enfrenta retos, Ejercicios y Portafolio. Este último es el más importante, pues es el que almacena todas las evidencias de aprendizaje que el estudiante adquiere. **De forma general estos instrumentos no brindan la información necesaria para dar seguimiento al desempeño de los logros y habilidades alcanzados por los estudiantes.**
- Del análisis de los portafolios y las especificaciones existentes, para hacer del portafolio de la plataforma educativa Zera una potente herramienta de aval del estudiante, la mejor opción es convertirlo a un Portafolio de Desarrollo Personal implementado bajo las especificaciones del Leap2A.

## Capítulo II: Características del sistema

### Introducción

Una vez concluido el estudio del estado del arte es necesario realizar un modelo de negocio o dominio para lograr una mayor comprensión de lo que se desarrollará y en base a ello elaborar los requisitos funcionales y no funcionales que sostendrá la solución propuesta y los diferentes artefactos que dan paso al análisis y diseño del sistema.

### 2.1. Modelo de dominio

En la presente investigación se decide realizar el modelo de dominio debido a que los procesos de negocio no estaban bien definidos. En función de esto se fueron manejando varios conceptos y asociaciones presentes en la plataforma educativa Zera y en las plataformas y herramientas seleccionadas en el epígrafe 1.2 del Capítulo I, quedando definido de esta forma el modelo de dominio.

En sentido general el modelo de dominio es utilizado por el analista como un medio para comprender los conceptos que utilizan los usuarios y su relación con los procesos que se desean desarrollar.

#### 2.1.1. Conceptos del modelo de dominio

- **Portafolio:** Es donde se almacenan todas las evidencias de aprendizaje adquiridas, así como los datos personales, sus calificaciones o cualquier documentación relacionada con el estudiante.
- **Usuarios:** Son las personas que generalmente acceden al portafolio, pueden comportarse como Estudiante, Docente, Tutor o Director.
- **Estudiante:** Persona que cursa estudios, tiene el propósito de aprender una materia en busca de conocimientos.
- **Docente:** Persona responsable de educar, guiar y supervisar a los estudiantes en el proceso de enseñanza – aprendizaje.
- **Tutor:** Persona responsable de guiar y supervisar las actividades de los estudiantes.

- **Director:** Persona encargada de dirigir una institución. Es también aquella persona responsable de guiar y supervisar las actividades de los estudiantes.
- **Calificaciones:** Es la nota que se emite para dar a conocer el valor que adquiere una evidencia.
- **Perfil:** Es toda la información personal del estudiante.
- **Notificaciones:** Es la acción y efecto de notificar.
- **Evidencia:** Es el elemento tangible que demuestra lo que sabe y es el individuo, pueden ser tanto de desempeño, como de comportamiento, deduciendo de ella los logros de aprendizaje que el estudiante adquiere.
- **Curso:** Período escolar en el que se evalúan diferentes materias.
- **Conocimientos específicos:** Unidad básica o mínima que conforma el sistema de conocimientos para una clase determinada.
- **Debate:** Es la acción de emitir criterios entre un grupo de personas.
- **Investigación:** Es una tarea evaluativa que orienta el docente.
- **Cuestionario:** Son un conjunto de ejercicios y/o preguntas.
- **Trabajo extra clase:** Son aquellas investigaciones o actividades que el estudiante realiza independiente de todas las tareas que el docente le orienta en el curso.
- **Reconocimiento:** Es la acción de notificar que nos destacamos en una labor, actividad o evento.
- **Asignación:** Es una acción que realiza el docente. Está dividido en 3 fases: la selección de la actividad a realizar, la selección de los estudiantes a los que le desea asignar dicha actividad y la presentación de los datos de la misma.
- **Actividades:** Conjunto de tareas que le son asignadas al estudiante.
- **Recorrido dirigido:** Es la pieza fundamental de la softarea, pues es el que guía, mediante una secuencia de pasos, cómo resolverla.

- **Softarea:** Es una tarea docente, que en el proceso de su solución, es necesario interactuar con determinado software educativo, para procesar determinada información. Incluye dos tipos de actividades: Cuestionario e Investigación.
- **Ejercicio:** Es una interrogante o una afirmación que arroja como resultado una solución.
- **Competencias:** Es el conjunto de los logros, habilidades y destrezas que el estudiante va adquiriendo durante toda su vida.

## 2.1.2. Modelo de dominio

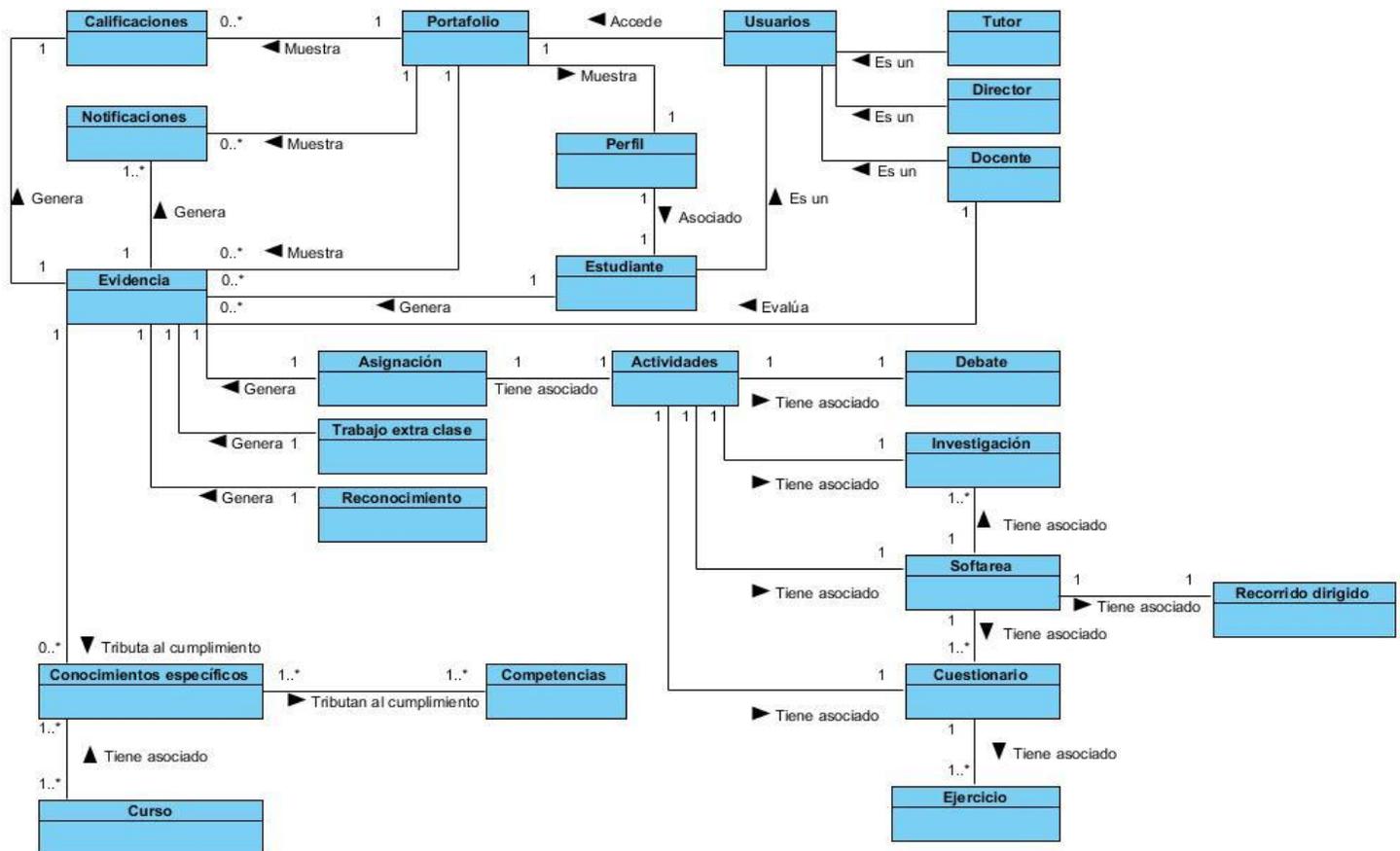


Ilustración 1. Representación del Modelo de Dominio

## 2.2. Descripción de la solución

A partir del análisis expuesto en el Capítulo 1 se definieron características a tener en cuenta para los instrumentos de evaluación identificados y para el portafolio, con el objetivo de enriquecer estos espacios y aportarle mayor calidad a la plataforma educativa Zera.

### 2.2.1. Para los instrumentos de evaluación

Se utilizarán como instrumentos de evaluación el Foro, Softareas, Reactivos, “Evidencias”, Ejercicios, Enfrenta retos y Portafolio. Los seis primeros se encargarán de arrojar evidencias del desempeño y los logros alcanzados por los estudiantes para archivarlos en el portafolio de la plataforma educativa Zera. A continuación se menciona de forma general la información que brindarán cada uno de ellos:

#### **Foro:**

Este instrumento deberá ser capaz de brindarle al docente la posibilidad de crear temas evaluativos en el foro, que puedan estar relacionados o no, con contenidos y conocimientos específicos (como lo determine el docente). Para cada tema creado por el docente le permitirá emitir un criterio (calificación) y una observación sobre el desempeño de sus estudiantes en el mismo. Esta propuesta quedará para futura implementación.

#### **Reactivos:**

Este instrumento deberá ser capaz de permitirle al docente emitir una observación sobre el desempeño del estudiante en esta actividad. Para el caso del estudiante le permitirá ver la solución con las respuestas que emitió sin posibilidades de modificación. Es importante destacar que, una vez almacenada esta evidencia en el portafolio del estudiante, sólo se exportarán los datos asociados a la actividad (nombre del reactivo, calificación, observaciones emitidas por el docente y los conocimientos específicos con los que se encuentra relacionado), no así con los ejercicios asociados a la misma.

#### **“Evidencias”:**

Este instrumento primeramente cambiará su nomenclatura por Investigación. Las funcionalidades propuestas se localizarán en dos subsistemas: Resources y Bachelor, en lo adelante Recursos y Bachiller. En el primero se sentarán las bases para crear un banco de **Recursos investigativos** que serán del tipo Enfrenta retos (Investiga y aprende, Aprende a resolver y Valora tus posibilidades) e Investigación y

estarán o no, asociados al cumplimiento de conocimientos específicos, aunque serían recomendable que siempre estuviesen asociados a conocimientos específicos, para así facilitar a docentes y estudiantes el análisis del desempeño académico.

En el caso del subsistema Bachiller, cuando el docente desee realizar una nueva asignación, se le mostrará el banco de Recursos investigativos anteriormente mencionado y se le brindará la posibilidad de gestionar otros nuevos. Además, podrá listar aquellos Enfrenta retos del módulo **Prácticas** con el objetivo de no limitar al docente a la hora de seleccionar que recurso desea asignar.

Es importante destacar que cuando se crean recursos investigativos del tipo Enfrenta retos, estos se visualizarán directamente en el módulo Prácticas del subsistema Bachiller, a menos que el que lo gestione decida seleccionar la opción *“Recomendado para evaluar”*, en tal caso se visualizaría en la sección Recursos Investigativos de ese mismo subsistema. Los recursos investigativos del tipo Investigación se visualizarán en el módulo Recursos investigativos de Bachiller.

### **Enfrenta retos:**

Este instrumento al convertirse en un recurso investigativo, dejará de ser parte de los Recursos interactivos. Cada Enfrenta retos del módulo Prácticas brindará la posibilidad de incluir la evidencia asociada junto con su descripción.

### **Softareas:**

Entre las funcionalidades propuestas para este instrumento se encuentran la posibilidad de brindar una calificación final, permitirle al docente realizar una observación sobre la misma y ligarla al vencimiento de conocimientos específicos.

### **Ejercicios:**

Este instrumento arrojará, por cada ejercicio que realice el estudiante, la siguiente información para ser almacenada en el portafolio de la plataforma educativa Zera: título del ejercicio, calificación general, cantidad de intentos y la respuesta por cada intento.

### 2.2.2. Para el portafolio del estudiante

Las funcionalidades propuestas se localizarán en el subsistema Learning, en lo adelante Administración del aprendizaje y tendrán acceso los siguientes roles: Director, Tutor, Docente y Estudiante. Los tres primeros serán solamente consultores de la información que se gestiona en el portafolio (es importante destacar que en la presente investigación se describirá la propuesta de la interacción de los roles Director, Tutor y Docente con el portafolio, pero su implementación se realizará en futuras versiones). Este se divide en siete secciones, donde se encontrarán las siguientes funcionalidades:

- **Pantalla principal:** En esta sección se muestran todos los datos del perfil del estudiante (primer nombre, segundo nombre, primer apellido, segundo apellido, credencial (foto), fecha de nacimiento, género, estado civil, áreas de interés, sitios de interés, intereses, acerca de mi persona, país origen, dirección actual, dirección postal, correo electrónico, teléfono fijo, teléfono móvil y dirección de su sitio personal). Además brindará un espacio con notificaciones recientes y otro con accesos directos a las opciones que se encuentran en el portafolio.
- **Mi perfil:** Esta sección estará dividida en dos subsecciones: *Sobre mí e Información de contacto*. En sentido general le ofrecerá al estudiante la posibilidad de introducir los siguientes datos de su perfil: credencial (foto), estado civil, sitios de interés, intereses, acerca de mi persona, dirección de su sitio personal y teléfono móvil. Los restantes se tomarán del perfil de usuario que gestiona la plataforma.

Por la necesidad de comprobar la validez de algunos datos, se agregará al perfil de usuario de la plataforma educativa Zera los siguientes datos: país origen, dirección actual, dirección postal.

A esta sección tendrá acceso solamente el estudiante.

- **Notificaciones:** En esta sección se agruparán las notificaciones del estudiante. Las funcionalidades previstas serán: la visualización de un sistema de búsqueda por fecha de envío y además la posibilidad, solo al estudiante, de eliminar aquellas notificaciones que desee.
- **Calificaciones:** Esta sección se agrupará en tres sub-secciones: *Promedio, Evaluaciones frecuentes, Competencias*.

Entre las funcionalidades previstas se encuentran: la visualización de la nota final de cada programa de estudio cursado por el estudiante y las evidencias asociadas a los diferentes tipos de actividades que existen actualmente en la plataforma (Reactivos en el aula, Reactivos en la plataforma, Actividad en el aula, Actividad en la plataforma y Evidencias), la visualización de las competencias vencidas, no vencidas y las que le faltan por cumplir al estudiante y la visualización de las evidencias asociadas a los diferentes tipos de actividades, mencionadas anteriormente, de aquellos programas de estudio que el estudiante esté cursando.

Es importante señalar que en el caso del estado de la competencia (Vencido, no vencido y por cumplir) lo determinará el docente desde el módulo Docente.

- **Mis evidencias:** En esta sección se almacenarán todas las evidencias del aprendizaje alcanzado por el estudiante. Las funcionalidades previstas serán: la gestión de evidencias que no están relacionadas con ninguna orientación realizada por parte de los docentes, la gestión de secciones, la visualización de conocimientos específicos asociados a cada una de dichas evidencias y la visualización y gestión de los comentarios. Además brindará la posibilidad de organizar dicho espacio y ver los datos asociados a cada evidencia almacenada.
- **Evidencias compartidas:** Esta sección mostrará las evidencias que le han sido compartida al estudiante dueño del portafolio y le permitirá, para cada evidencia, la posibilidad de insertar y editar un único comentario.
- **Exportar:** Esta sección permitirá exportar todos los datos almacenados en el portafolio bajo las especificaciones del estándar seleccionado: Leap2A.

## 2.3. Requerimientos del software

Los requerimientos de sistemas deben mostrar todo lo que el sistema debe hacer más todas las restricciones sobre la funcionalidad (44). Deben ser suficientemente entendibles como para que pueda llegarse a un acuerdo entre el cliente y los desarrolladores sobre qué debe y qué no debe hacer el sistema.

### 2.3.1. Definición de los requisitos funcionales

RF 1 Generar evidencia

RF 1.1 Generar evidencia: Esta funcionalidad describe cuáles son los datos que se almacenarán en el portafolio del estudiante que sirven como evidencia del aprendizaje y que se tomarán cuando se evalúe un instrumento de evaluación (Softareas, Investigación, Ejercicios y Reactivos) o se adjunta una evidencia como respuesta a una Investigación tanto evaluativo como del módulo Práctica (Enfrenta retos: Investiga y Aprende, Aprende a resolver y Valora tus posibilidades).

**RF 2** Filtrar campos para realizar la consulta: Muestra un conjunto de filtros que permiten realizar una búsqueda más precisa en el listado.

**Para la gestión de los Recursos investigativos:** Los requisitos funcionales que se describirán a continuación, se localizarán en dos subsistemas: Recursos y Bachiller. En el primero tendrán acceso el Administrador central y el Editor, y en el último solamente el Docente.

**RF 3** Gestionar Recursos investigativos:

- RF 3.1 Incluir recurso investigativo.
- RF 3.2 Eliminar recurso investigativo: Para el caso del Docente podrá eliminar aquellos recursos que cree.
- RF 3.3 Editar datos del recurso investigativo: Para el caso del Docente podrá editar aquellos recursos que cree.
- RF 3.4 Ver datos del recurso investigativo.

**Para el portafolio del estudiante:**

**RF4** Consultar datos del perfil

- RF 4.1 Mostrar los datos del perfil del estudiante.

**RF 5** Mostrar accesos directos de algunas opciones del portafolio: Esto permite la ejecución de algunos procesos directamente.

**RF 6** Editar perfil

- RF 6.1 Editar datos del perfil: El sistema permitirá, únicamente al estudiante, editar algunos datos de su perfil como son: credencial (foto del usuario), estado civil, actividades, sitios de interés, intereses, acerca de mi persona, teléfono móvil y dirección de su sitio personal.

## **RF 7** Consultar Notificaciones:

- RF 7.1 Mostrar un listado de notificaciones.
- RF 7.2 Eliminar notificaciones: Esto se le permitirá solamente al estudiante.
- RF 7.3 Mostrar un listado de notificaciones recientes. Estas notificaciones serán desde el último día que accedió el estudiante al portafolio hasta el día que accede.

## **RF 8** Consultar promedio basándose en los programas de estudio cursados:

- RF 8.1 Mostrar un listado de programas de estudio ya cursados por el estudiante.
- RF 8.2 Mostrar la nota final de los programas de estudio ya cursados.
- RF 8.3 Mostrar porciento de cumplimiento por cada tipo de actividad.
- RF 8.4 Mostrar promedio general del estudiante.

## **RF 9** Consultar evaluaciones frecuentes:

- RF 9.1 Mostrar un listado de los programas de estudio que está cursando el estudiante.

## **RF 10** Consultar competencias:

- RF 10.1 Mostrar un listado de las competencias gestionadas en la plataforma.
- RF 10.2 Mostrar el estado de cada competencia: Si se encuentra vencido, no vencido o por cumplir.
- RF 10.3 Mostrar los conocimientos específicos por cada competencia.
- RF 10.4 Mostrar por cada conocimiento específico las evidencias asociadas junto con su calificación.
- RF 10.5 Mostrar porciento de desempeño actual por conocimiento específico.

**RF 11** Mostrar el listado de los tipos de actividades (Actividad en el aula, Actividad en la plataforma, Reactivo en el aula, Reactivo en la plataforma y Evidencias) que existen en la plataforma y por cada uno, mostrar el listado de las evidencias asociadas junto con sus calificaciones. Por cada tipo de actividad se visualizará además su porciento de cumplimiento.

## **RF 12** Consultar evidencias compartidas

- RF12.1 Mostrar un listado de las evidencias compartidas.
- RF 12.2 Eliminar evidencia compartida. Esta acción sólo podrá ser realizada por el estudiante.

## **RF 13** Consultar mis evidencias

- RF13.1 Mostrar un listado de evidencias.

- RF 13.2 Mostrar un listado de los conocimientos específicos asociados a una evidencia.
- RF 13.3 Ordenar secciones. Sólo tendrá acceso a esta opción el estudiante.
- RF 13.4 Mostrar datos de la evidencia.
- RF 13.5 Ver la evidencia: Se refiere a abrir el documento, imagen etc.

### **RF 14** Compartir evidencias

- RF 14.1 Mostrar un listado de los estudiantes del mismo grupo del dueño del portafolio, que posean la misma evidencia ya evaluada.
- RF 14.2 Mostrar un listado de los docentes al estudiante (aquellos que se encuentran impartiendo un programa de estudio).
- RF 14.3 Dejar de compartir las evidencias que han sido compartidas.

### **RF 15** Gestionar evidencias personales.

- RF 15.1 Incluir nueva evidencia.
- RF 15.2 Editar datos de la nueva evidencia.
- RF 15.3 Eliminar evidencia.
- RF 15.4 Ver la evidencia.

### **RF 16** Gestionar secciones.

- RF 16.1 Incluir nueva sección.
- RF 16.2 Editar datos de la sección.
- RF 16.3 Eliminar sección.
- RF 16.4 Ver datos de la sección: Muestra todas las evidencias asociadas a la sección.

### **RF 17** Exportar datos del portafolio

- RF 17.1 Exportar los datos del portafolio utilizando el estándar Leap2A.

**RF 18** Gestionar comentarios: Los requisitos de incluir, editar y eliminar comentarios se encontrarán en la sección Evidencias compartidas del portafolio. En cuanto a la sección Mis evidencias se encontrarán los requisitos eliminar y ver comentarios.

- RF 18.1 Incluir comentario.
- RF 18.2 Editar comentario.
- RF18.3 Eliminar comentario.

- RF 18.4 Ver comentarios.

## 2.3.2. Definición de los requisitos no funcionales

- **Soporte**

**RNF 1:** Ser generado en tecnología Web para ser accesible a través de Internet.

**RNF 2:** Presente soporte para futuras migraciones.

**RNF 3:** Ejecutarse sobre cualquier navegador, siendo como mínimo compatible con:

- Explorer 7.0 y superior.
- Mozilla Firefox 3.6 y superior.
- Opera 10.00 y superior.
- Chrome 7.0 y superior.

- **Hardware**

**RNF 4** Los usuarios finales deberán contar como mínimo con:

- Procesador Pentium II o superior.
- 512MB de RAM.
- 20GB de HDD.
- Si no cuentan con servidor local: conexión de banda ancha 256 Kbps como mínimo.

- **Restricciones de diseño e implementación**

**RNF 5:** Lenguaje y tecnología de desarrollo: XHTML 1.0, XML 1.0, CSS 2.1, Javascript, Ajax y PHP 5.3.3.

**RNF 6:** El marco de trabajo base de desarrollo que se utilizará es: Symfony 1.4.15

**RNF 7:** Como IDE se empleará NetBeans 7.1.

**RNF 8:** Como servidor Web se explotará Apache 2.0.

**RNF 9:** El Sistema Gestor de Base de Datos deberá ser PostgreSQL 9.0.

- **Usabilidad**

**RNF 10:** Los elementos gráficos como los íconos deberán contar con un tooltip o mensaje flotante que señalen el tipo de recurso al que se refiere.

**RNF 11:** Las notificaciones recientes deberán contar, en caso de ser muy extensas, con un tooltip o mensaje flotante cuando se posicione el puntero del mouse encima de las mismas.

**RNF 12:** Permitir el almacenamiento de objetos binarios, incluyendo imágenes, sonidos y videos.

**RNF 13:** Los usuarios deberán tener conocimientos básicos del manejo de las computadoras y los sistemas operativos. Además la aplicación deberá contar con una interfaz y navegación funcional, amena, asequible, para todo tipo de usuario. Se debe tener acceso al menú general desde cualquiera de las páginas.

- **Eficiencia**

**RNF 14:** Cuando un usuario, sin importar su rol, permanece inactivo durante 20 minutos, se cierra la sesión automáticamente.

- **RNF 15 Apariencia o Interfaz Externa:**

La interfaz debe ser amigable y sugerente para el usuario, la información será mostrada con claridad y organización. Diseño sencillo, con pocas imágenes y gráficos, para que la ejecución de las acciones se realice de manera rápida.

## 2.4. Presentación del diagrama de Caso de Uso

A continuación se describirán los casos de usos (en lo adelante CU) del sistema propuesto y los actores que interactúan con ellos.

### 2.4.1. Descripción de los actores del sistema

Los actores identificados son aquellas personas que van a inicializar e interactuar con los CU del sistema.

Actor	Descripción
Estudiante	Actor que podrá realizar dentro de su portafolio cambios en los datos de su perfil, gestión de secciones y evidencias, podrá consultar sus notificaciones,

	calificaciones, las evidencias que le son compartidas y para cada una de ellas podrá insertar o editar un comentario y eliminar aquellas evidencias que desee. Además podrá compartir sus evidencias y exportar su portafolio.
<b>Docente</b>	Actor que podrá gestionar, asignar y evaluar los recursos investigativos. Además podrá consultar todos los espacios del portafolio de cada uno de sus estudiantes, solamente no tendrá acceso al espacio Exportar y Mi perfil.
<b>Tutor (padres)</b>	Actor que podrá consultar todos los espacios del portafolio de sus hijos, solamente no tendrá acceso al espacio de Exportar y Mi perfil.
<b>Editor</b>	Actor que podrá consultar y gestionar los recursos investigativos desde el subsistema Recursos.
<b>Director</b>	Actor que podrá consultar todos los espacios del portafolio de cada estudiante de la plataforma, solamente no tendrá acceso al espacio “Mi perfil”.
<b>Administrador central</b>	Actor que podrá consultar y gestionar los recursos investigativos desde el subsistema Recursos.

*Tabla 5. Actores del sistema*

## 2.4.2. Consideraciones previas al diagrama de casos de uso

Para poder comprender el diagrama de CU, se realizó un diagrama de actores del sistema (Ver Ilustración 2). Para este último, se utilizó la siguiente relación para la elaboración del mismo.

- **Generalización/Especialización entre actores:** Una relación de generalización de una clase hija de actor a otra clase padre de actor indica que el hijo hereda el rol que la clase padre puede jugar respecto a un caso de uso. Se ve reflejado en Usuario de Bachelor que representa la generalización de los actores: Estudiante, Docente, Tutor y Director, así como, el Gestor de recursos que representa a los actores Docente, Administrador central y Editor.

### 2.4.3. Patrones de casos de uso

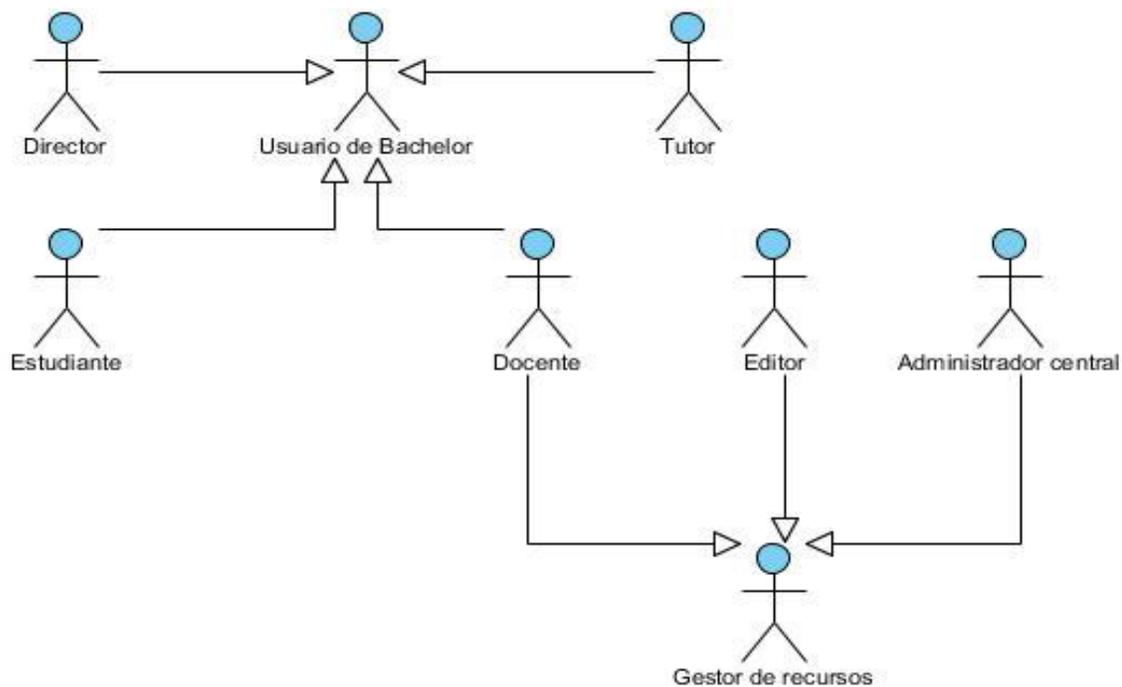
Una vez analizado el diagrama de actores elaborado se pasará a detallar los patrones de casos de uso utilizados que ayudaron a conformar el diagrama de CU del sistema.

**CRUD (Creating, Reading, Updating, Deleting):** este patrón se basa en la fusión de CU simples para formar una unidad conceptual. Se puede apreciar el uso de este en los CU Gestionar Recursos investigativos, Gestionar evidencias personales, Gestionar comentarios y Gestionar sección.

**Múltiples actores:** el uso de este patrón se evidencia específicamente en la variante roles comunes, la cual establece que varios actores jueguen el mismo rol sobre el CU. Este rol es representado por otro actor, heredado por los actores que comparten este rol. Siendo reflejado a continuación en la Ilustración 2, la cual muestra una parte del diagrama de CU del sistema.

**Concordancia:** Extrae una subsecuencia de acciones que aparecen en diferentes lugares del flujo de casos de uso y es expresado por separado. La variante empleada fue: Reusabilidad. Dicha variante consta de tres casos de uso. El primero es denominado subsecuencia común ya que modela una secuencia de acciones que aparecerán en múltiples casos de uso en el modelo y los otros casos de uso modelan el uso del sistema que comparte la subsecuencia común de acciones. Es por ello que deben existir al menos dos de ellos.

### 2.4.4. Diagrama de actores del sistema



*Ilustración 2. Diagrama de Actores del Sistema*

El diagrama de actores del sistema se realizó para comprender con mayor facilidad el diagrama de caso de usos y tomar una vista general de los actores que interactúan con la solución.

## **2.4.5. Diagrama de casos de uso del sistema**

El diagrama de casos de uso del sistema muestra la relación entre los usuarios y los CU con los que interactúan y nos ayuda a comprender como funciona el sistema.

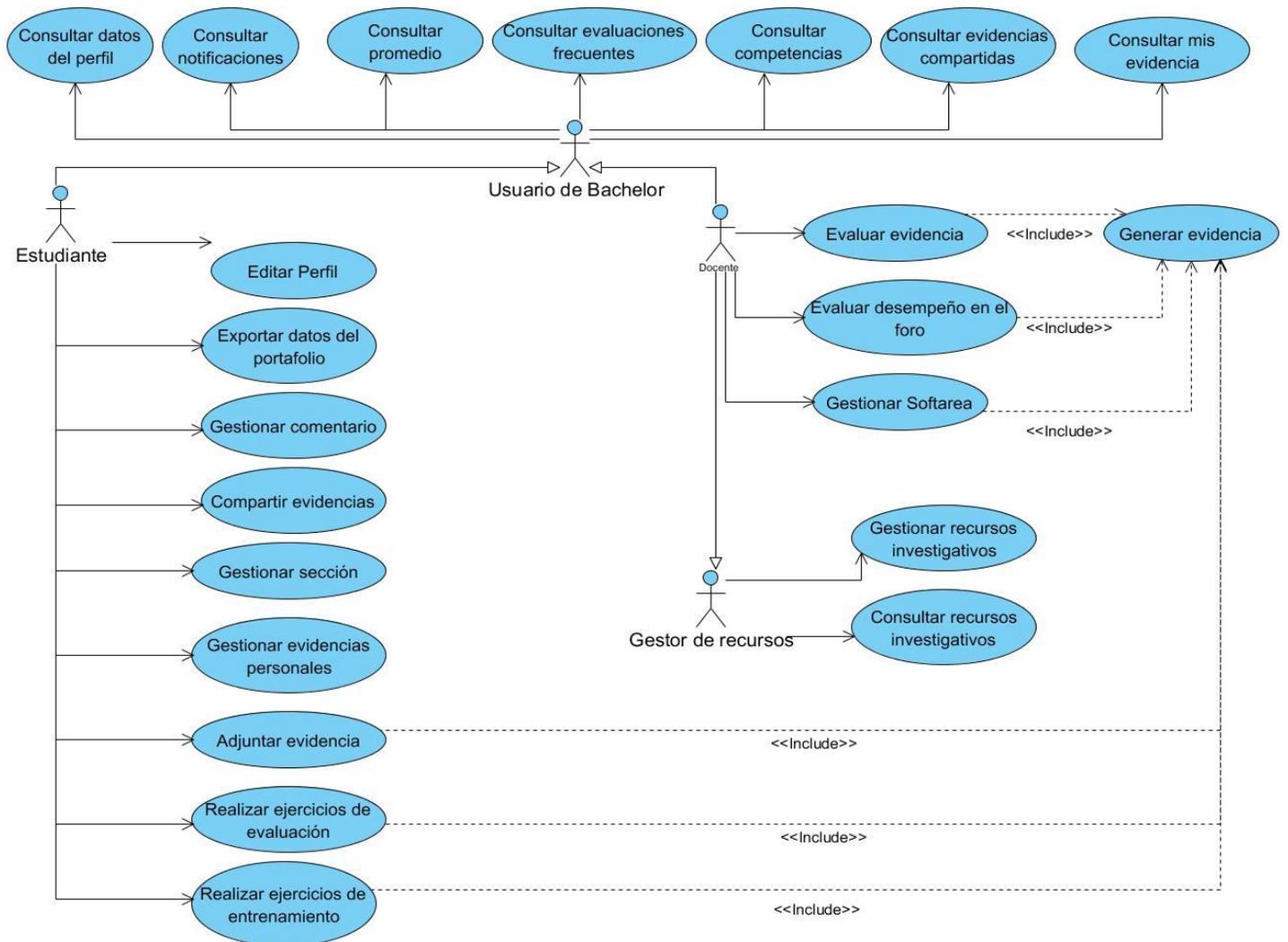


Ilustración 3. Diagrama de Caso de uso del sistema

## 2.4.6. Descripción de los casos de uso del sistema

Del análisis de los requisitos funcionales se elaboraron los CU que describirán cómo funciona el sistema y su relación con los actores. Para ver todas las descripciones de los CU dirigirse al [Anexo 3: Descripciones de los Casos de uso.](#)

## **2.5. Conclusiones parciales del capítulo**

Con la realización del presente capítulo se logró realizar un análisis profundo del dominio obteniendo como resultado la propuesta de solución. Para ello fue necesario reflejar los principales conceptos del modelo de dominio y dejar bien definidos los requisitos funcionales y no funcionales que sustentarán la propuesta realizada. Esto permitió que se realizara una descripción de cada actor y CU ayudando de esta forma a comprender cuáles son los actores que están involucrados en la solución y qué acciones realizará la aplicación con ellos. Partiendo de este punto, se dará comienzo a la construcción de la propuesta elaborada, teniendo en cuenta el cumplimiento de los requerimientos propuestos.

## Capítulo III: Análisis y diseño de la propuesta del sistema

### Introducción

Luego de ser identificados y descritos todos los requisitos del sistema a desarrollar es importante realizar un análisis de ellos con el objetivo de obtener una mejor comprensión antes de entrar al diseño, garantizando de esta forma una arquitectura robusta y capaz de sobrevivir a cambios. El resultado de este proceso es conocido como el modelo de análisis que es la entrada esencial en el diseño. En el diseño se modela el sistema y se encuentra su forma para que soporte todos los requisitos, incluyendo los no funcionales.

### 3.1. Modelo del análisis

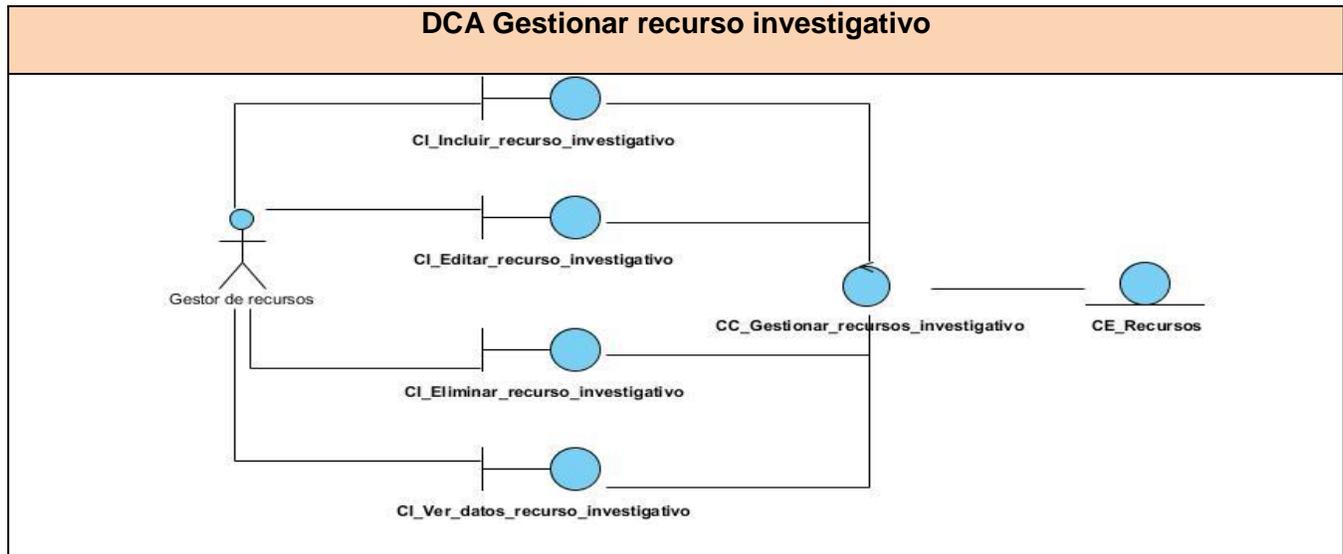
El modelo de análisis especifica el comportamiento funcional del sistema, independientemente de los aspectos relativos al ambiente en el que va a ser finalmente implementado. Captura completamente y con exactitud los requerimientos del sistema. (45)

Es el resultado del flujo de trabajo del análisis y en sentido general, es un modelo de objetos conceptual que analiza los requisitos mediante su refinamiento y estructura. Incluye entre otros, los siguientes elementos: (45)

#### 3.1.1. Diagrama de clases del análisis

Los diagramas de clases del análisis son utilizados para entender de forma general la estructura del sistema y sirve como entrada en la etapa del diseño. (45)

A continuación se muestra el diagrama de clase del análisis (DCA) para uno de los dos CU significativos de la investigación, pues el **CU Generar evidencia** se incluye dentro de los CU asociados a los instrumentos de evaluación mencionados durante la investigación. Los restantes se encuentran recogidos en el [Anexo 4](#): Diagramas de clases del análisis. Si además se desean consultar los diagramas de interacción del análisis ver [Anexo 5](#): Diagramas de interacción del análisis.



### 3.1.2. Diagrama de paquetes del análisis

Los paquetes del análisis proporcionan un medio para organizar los artefactos del modelo del análisis en piezas manejables. (45)

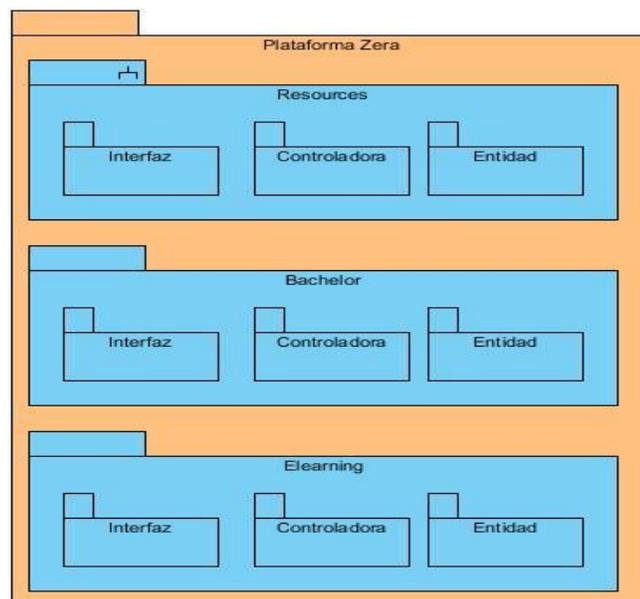


Ilustración 4. Diagrama de paquete del análisis

## 3.2. Patrones en Symfony

Una vez concluido el análisis se comenzará a realizar el modelo de diseño del sistema, pero antes, es necesario entender algunos conceptos que influyen en la comprensión del mismo. Para ello, es necesario comenzar enunciando las características de los patrones que utiliza el framework seleccionado: Symfony. Symfony utiliza patrones de diseño y un patrón arquitectónico, los cuáles se detallarán a continuación.

### 3.2.1. Patrón arquitectónico Modelo- Vista- Controlador (MVC)

Los patrones de arquitectura expresan un esquema organizativo estructural fundamental para sistemas de software. (46)

El MVC es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. Se ve frecuentemente en aplicaciones web (46). Esto permite que los cambios en la vista no afecten el resto de la aplicación, proporcionando como ventajas la reutilización de los componentes y la simplicidad en el mantenimiento del sistema. Está conformado por los siguientes niveles: (47)

**El Modelo:** Representa la información con la que trabaja la aplicación, es decir, su lógica de negocio, las reglas, restricciones y condiciones definidas para la operación de la aplicación. Se puede dividir en:

- Acceso a los datos: La ORM Doctrine se encarga de generar automáticamente las clases que utilizará la capa modelo, ya que crea el esqueleto o estructura básica de las clases y genera automáticamente el código necesario.
- La abstracción de la base de datos: es completamente invisible al programador. Así, si se cambia el sistema gestor de bases de datos en cualquier momento, no se debe reescribir ni una línea de código, ya que tan sólo es necesario modificar un parámetro en un archivo de configuración.

**La Vista:** Transforma el modelo en una página web que permite al usuario interactuar con ella. En ambos plugins (Recursos investigativos y Portafolio), estas páginas están ubicadas en el directorio templates/. La vista se puede dividir en:

- Vista: es la interfaz con la que el usuario interactúa con la aplicación Web.
- Plantilla: sólo se encarga de visualizar las variables definidas en el controlador. Para que estos componentes interactúen entre sí correctamente, es necesario añadir cierto código (son aquellas

páginas que se visualizan en el directorio templates/ de la siguiente forma: nombreSuccess.php y \_nombreSuccess.php).

- Layout: es global en toda la aplicación o al menos en un grupo de páginas.

**El Controlador:** Se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el Modelo o en la Vista. Se puede dividir en:

- Controlador frontal: redirecciona los pedidos que el usuario hace sobre la vista, al par modulo/acción correspondiente.
- Acción: es el componente de código mínimo que se puede crear en el framework, donde se ilustra cómo una función recibe unos parámetros de entrada.
- Los objetos de pedido (request), respuesta (response) y sesión (session), que dan acceso a los parámetros de la petición, las cabeceras de las respuestas y a los datos persistentes del usuario.
- Los filtros, los cuales son trozos de código ejecutados para cada petición, antes o después de una acción.

### 3.2.2. Patrones de diseño

Los patrones de diseño ofrecen una buena solución a problemas de diseño comunes en el desarrollo de aplicaciones software y otros temas relacionados como diseño de interacción o interfaces. Symfony, trae incluido por defecto en su arquitectura varios patrones de diseño, que están concebidos para que el programador los aplique obligatoriamente. A continuación se explicarán algunos de los patrones utilizados directamente en la solución.

#### **Patrones GRASP (General Responsibility Assignment Software Patterns) (48)**

- **Experto:** Es uno de los patrones que más se utiliza cuando se trabaja con Symfony, con la inclusión de la librería Doctrine para mapear la base de datos. En el modelo existen dos clases fundamentales: las clases encargadas de hacer las consultas a la base de datos utilizando Doctrine, ya que tienen los atributos necesarios para efectuar dicha función, por tanto tienen la responsabilidad de realizar directamente las acciones con la base de datos y las clases de acceso a datos, son las responsables de interactuar con las clases de abstracción de datos y devolver los objetos que necesitan los controladores en su forma original.

- **Creador:** Este patrón se usa para establecer la responsabilidad de quién es el encargado de crear objetos de una clase determinada. Es usado para crear objetos entidades, de formularios y de clases de abstracción a la base de datos.
- **Alta cohesión:** Symfony permite la organización del trabajo en cuanto a la estructura del proyecto y la asignación de responsabilidades con una alta cohesión. Un ejemplo de ello es la clase Actions, la cual está formada por varias funcionalidades que están estrechamente relacionadas, siendo la misma la responsable de definir las acciones para las plantillas y colaborar con otras para realizar diferentes operaciones, instanciar objetos y acceder a las propiedades.
- **Bajo acoplamiento:** La clase Actions hereda únicamente de sfActions para alcanzar un bajo acoplamiento de clases. Las clases que implementan la lógica del negocio y de acceso a datos se encuentran en el modelo, las cuales no tienen asociaciones con las de la vista o el controlador, lo que proporciona que la dependencia en este caso sea baja.
- **Controlador:** En Symfony todas las peticiones Web son manipuladas por un solo controlador frontal (sfActions), que es el punto de entrada único de toda la aplicación en un entorno determinado. Este patrón se evidencia en las clases sfFrontController, sfWebFrontController, sfContext, los “actions” y el index.php del ambiente.

## Patrones GoF (Gang of Four, en español Pandilla de los Cuatro) (48)

- **Singleton:** Clase sfRouting – método getInstance Esta clase la utiliza el controlador frontal (sfWebFrontController) y se encarga de enrutar todas las peticiones que se hagan a la aplicación. El singleton sfRouting precisa otros métodos muy útiles para la gestión manual de las rutas: ClearRoutes(), hasRoutes(), getRoutesByName().
- **Decorator (Decorador):** Este método pertenece a la clase abstracta sfView, padre de todas las vistas, que contienen un decorador para permitir agregar funcionalidades dinámicamente. El archivo nombrado layout.php es el que contiene el Layout de la página. Este archivo, conocido también como plantilla global, guarda el código HTML que es usual en todas las páginas del sistema, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el

layout, o si se mira desde el otro punto de vista, el layout decora la plantilla. Este procedimiento es una implementación del patrón Decorator.

Además se utilizó el patrón:

- **Strategy (Estrategia):** El objetivo de este patrón es definir un grupo de clases que representan un conjunto de posibles comportamientos. Estos comportamientos pueden ser fácilmente intercambiados en una aplicación, modificando la funcionalidad en cualquier instante. Es utilizado cuando se necesita mostrar los datos de una evidencia.

### 3.3. Modelo de diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Además, sirve de abstracción de la implementación del sistema y es utilizada como entrada fundamental de las actividades de implementación. (45)

#### 3.3.1. Diagrama de clases del diseño

Los diagramas de clases del diseño son una representación más concreta y detallada que los diagramas de clases del análisis, incluyendo las clases activas, sus operaciones, atributos y requisitos de implementación. (45)

A continuación se presenta el diagrama de clases del diseño correspondiente a los **CU Gestionar y Consultar recursos investigativos**. Los diagramas de clases del diseño correspondiente a todos los CU que describen el portafolio de la plataforma educativa Zera se encuentra en el [Anexo 6: Diagramas de clases del diseño de los CU relacionados con el portafolio](#).

Para comprender los diagramas elaborados es importante puntualizar que tanto los diagramas asociados a los recursos investigativos (representado por el nombre *sfInvestigationPlugin*) como del portafolio (representados por el nombre *sfPortafolioPlugin*) se visualizan como paquetes porque ambos serán plugins a desarrollar.

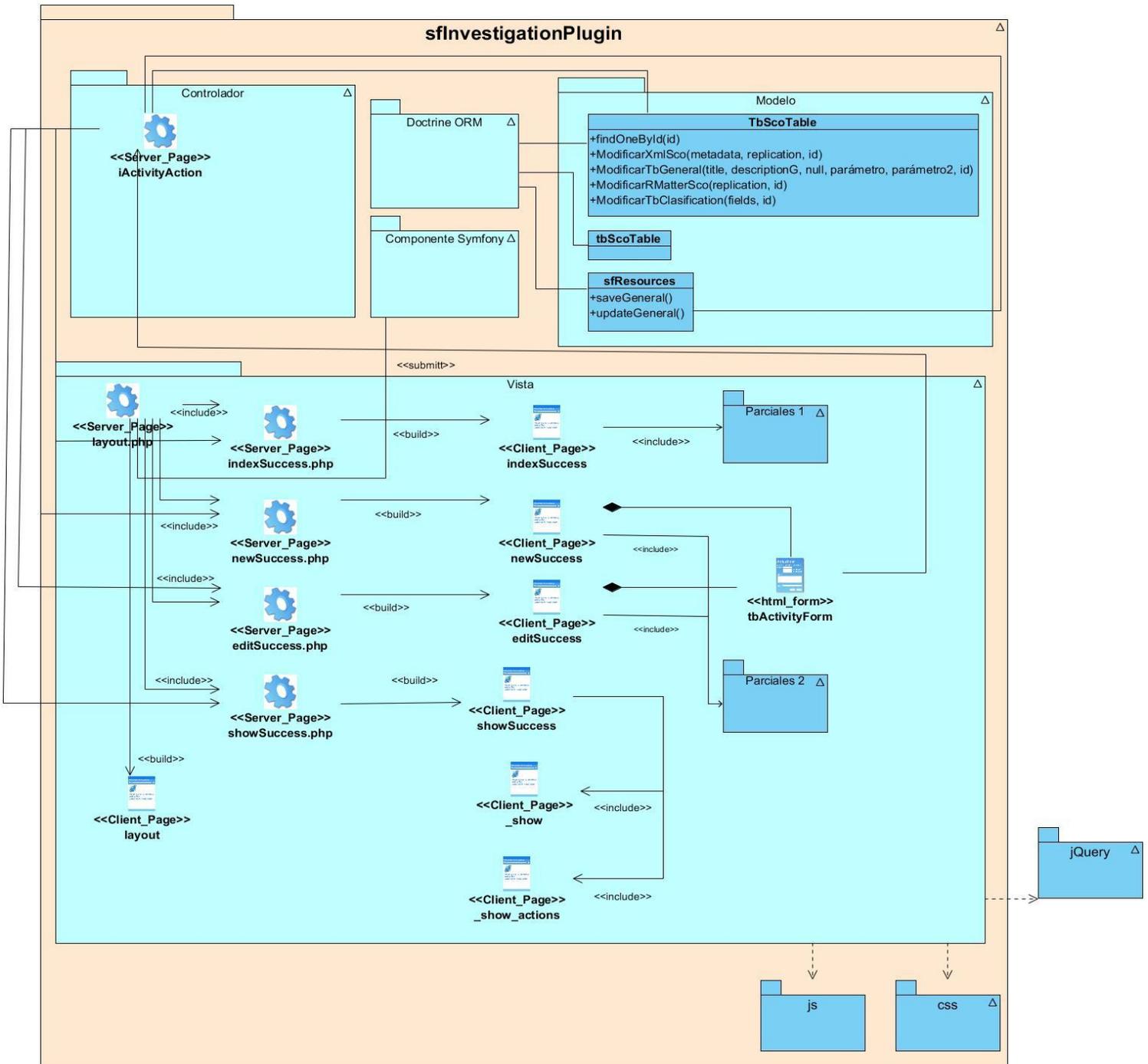
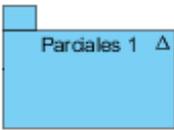
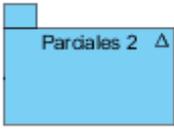
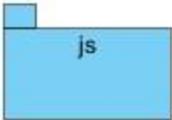


Ilustración 5. Diagrama de clases del diseño de los CU Consultar y Gestionar recursos investigativos

Para comprender la ilustración expuesta anteriormente, es necesario puntualizar los métodos que se encuentran en la clase controladora y las clases que se encuentran en los paquetes **Parciales 1**, **Parciales 2**, **js** y **css**. Tener en cuenta que dichos paquetes no son reales, se utilizaron para lograr una mayor comprensión del diagrama.

	<p>Los métodos que se encuentran son: <i>preExecute()</i>, <i>processInvalidForm()</i>, <i>executeIndex()</i>, <i>executeFilter()</i>, <i>executeCreate()</i>, <i>executeNew()</i>, <i>executeLoadOtherActivitiesForm()</i>, <i>processForm()</i>, <i>executeEdit()</i>, <i>executeUpdate()</i>, <i>workInReplication()</i>, <i>executeBatch()</i>, <i>executeBatchDelete()</i>, <i>getFilters()</i>, <i>setFilters()</i>, <i>isEmptyFilter()</i>, <i>getPager()</i>, <i>setPage()</i>, <i>getPage()</i>, <i>buildQuery()</i>, <i>addSortQuery()</i> y <i>getSort()</i>.</p>
	<p>Se encuentran las siguientes páginas clients (client page): <i>_filters</i>, <i>_list_td_tabular</i>, <i>_filters_field</i>, <i>_list_td_batch_actions</i>, <i>_list</i>, <i>_list_actions</i>, <i>_list_td_actions</i>, <i>_pagination</i>, <i>_list_th_tabular</i>, <i>_list_batch_actions</i>, <i>_list_header</i>. La página cliente (client page) <i>indexSuccess</i> se encuentra relacionado con cada una de las client page enunciadas.</p>
	<p>Se encuentran las siguientes páginas clients (client page): <i>loadOtherActivitiesFormSuccess()</i>, <i>_partialInvestigation()</i>, <i>_partialResearchAndLearn()</i>, <i>_partialLearnHowtoSolve()</i>, <i>_partialRateYourChances()</i>, <i>_assets()</i>, <i>_form()</i> y <i>_form_fieldset()</i>. Las páginas clientes (client page) <i>newSuccess</i> y <i>editSuccess</i> se encuentran relacionado con cada una de las client page enunciadas.</p>
	<p>Se encuentran las clases: <i>recursos.js</i>, <i>plug_media_JS.js</i>, <i>activities.js</i> y <i>plug_fileUpload_investigations.js</i>.</p>

	Se encuentran las clases: <i>principal.css</i> , <i>plug_ires_resource.css</i> , <i>aofiltersmatter.css</i> , <i>default.css</i> , <i>iresources.css</i> , <i>plug_media_css.css</i> , <i>AOPagedplugin.css</i> y <i>investigations_bachelor.css</i> .
--	--

### 3.3.2. Diagrama de paquete del diseño

El diagrama de paquete del diseño es un medio que nos muestra cómo estará organizado nuestro diseño mediante carpetas (45). En este caso, su distribución está apoyada en la estructura de Symfony.

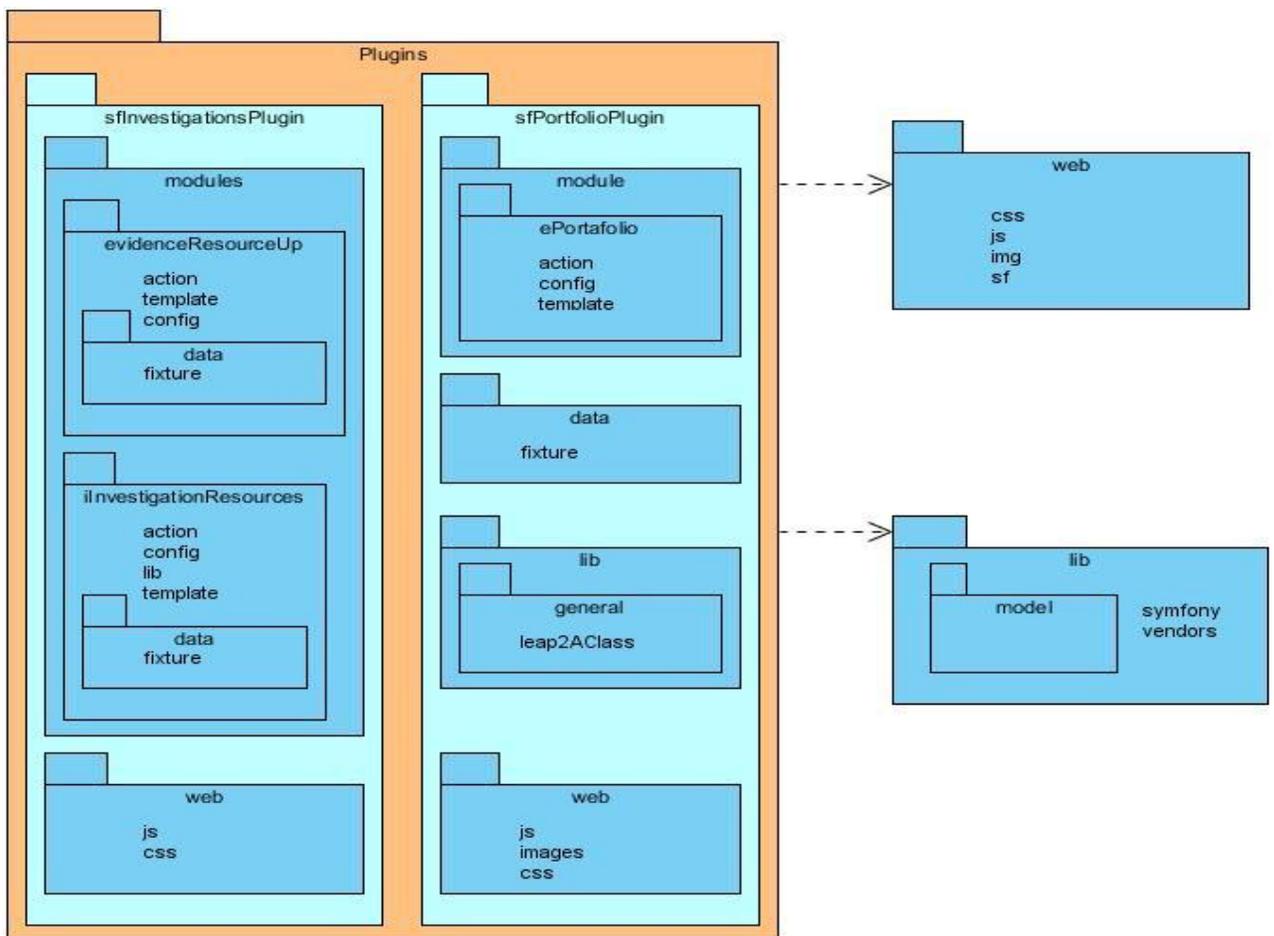


Ilustración 6. Diagrama de paquete del diseño

### 3.3.3. Modelo de despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuyen las funcionalidades entre los nodos de cómputo. Se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en el diseño (45). Con su elaboración queda especificado el hardware, software de interfaces, sistemas operativos y soporte que conformarán nuestro sistema.

A continuación se presenta el modelo de despliegue propuesto:

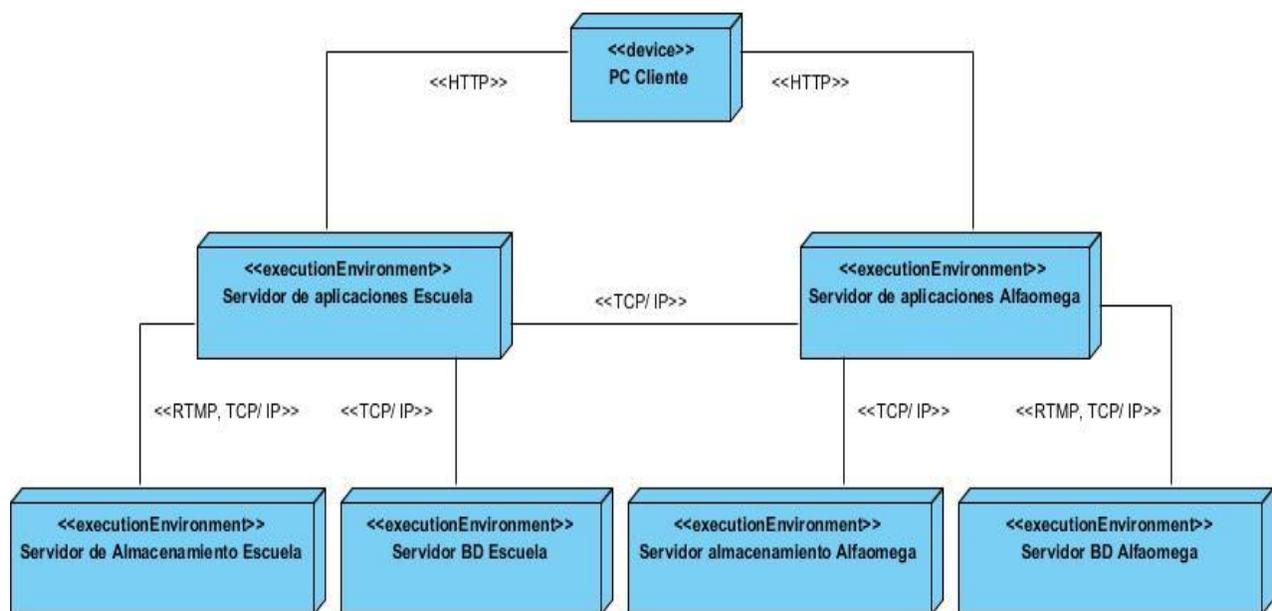


Ilustración 7. Modelo de despliegue

### 3.4. Diseño de la Base de datos

En esta sección se presenta el modelo de datos que estará compuesto por las entidades que pasarán a ser las tablas de la base de datos, para ser utilizadas por las funcionalidades a desarrollar. Las descripciones de cada una de ellas se pueden encontrar en el [Anexo 7](#): Descripciones de las tablas de la base de datos.

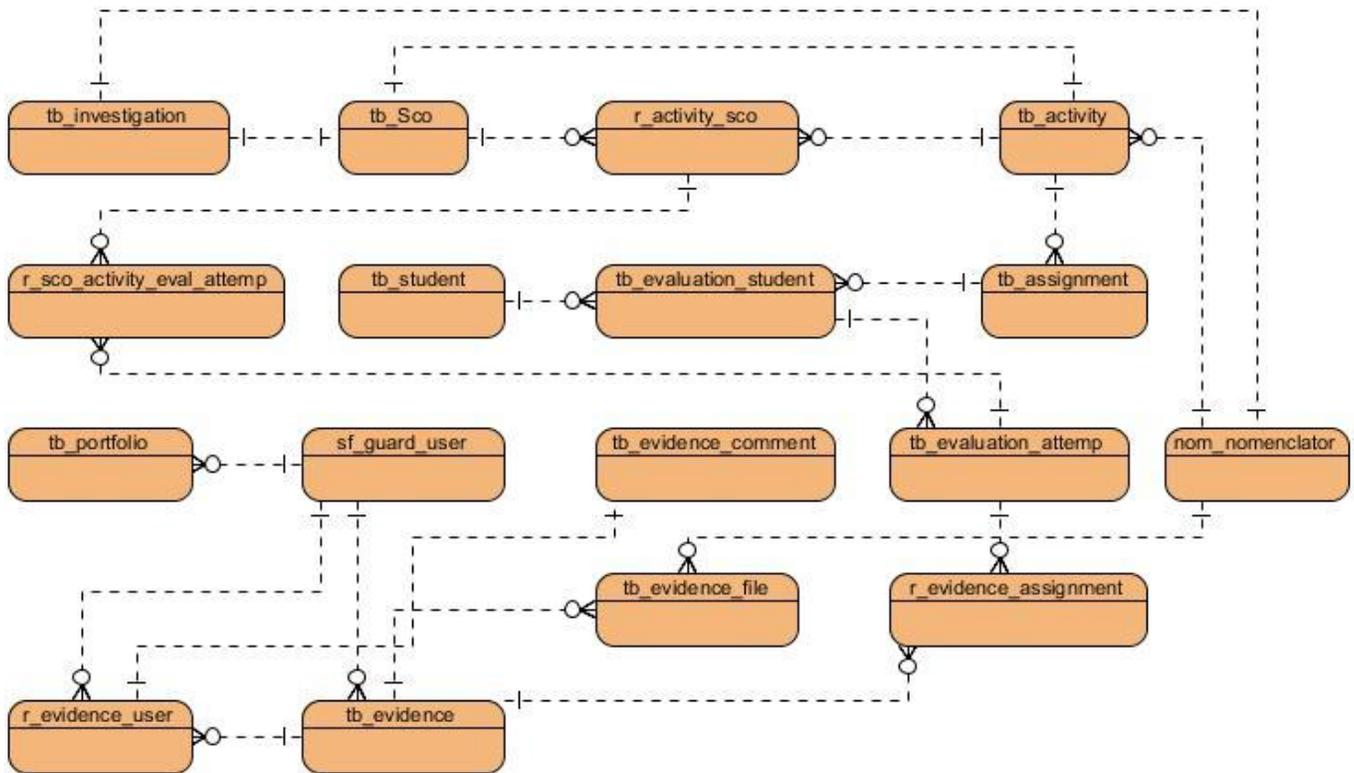


Ilustración 8. Diseño de la base de datos

### 3.5. Conclusiones parciales del capítulo

Durante el flujo de trabajo análisis y diseño llevado a cabo se pudieron obtener los siguientes artefactos: el modelo del análisis, donde se elaboraron los diagramas de clases, de paquetes y de interacción del análisis; el modelo de diseño, donde se diseñaron los diagramas de clases del diseño, el diagrama de paquetes, el modelo de despliegue y el diagrama entidad relación que define el diseño de la base de datos. Esto facilitó la comprensión de la solución y ayudó a determinar cómo se desarrollará el sistema a partir de los requerimientos elaborados y las funcionalidades previstas, garantizando con gran precisión lo que se implementará.

## Capítulo IV: Implementación y prueba

### Introducción

La implementación es la última fase del desarrollo de sistemas. Los artefactos generados durante el análisis y el diseño son la entrada fundamental a dicho flujo, el cual tiene como propósito: implementar las clases y objetos en forma de componentes, probar los componentes desarrollados e integrarlos con el sistema ejecutable. Al implementar un sistema lo primero que se debe tener en cuenta son los requerimientos y permitir que los usuarios puedan interactuar con él. Para comprobar y demostrar la calidad de lo implementado se realizan una serie de pruebas que validarán que se cumple con todos los requerimientos previstos, convirtiéndolo de esta forma en el producto deseado.

### 4.1. Modelo de implementación

El modelo de implementación es la entrada principal de las etapas de prueba que siguen a la implementación. Más concretamente, durante la etapa de prueba cada construcción generada durante la implementación es sometida a pruebas de integración, y posiblemente también a pruebas de sistema. (49)

#### 4.1.1. Diagrama de componentes

Se utilizan para modelar la vista estática de un sistema. Muestran la organización y las dependencias lógicas entre un conjunto de componentes de software, sean estos componentes de código fuente, librerías, binarios o ejecutables. (49)

A continuación se presenta el diagrama de componentes propuesto para el plugin *sfiInvestigationPlugin* correspondiente a la solución de los recursos investigativos. Los restantes diagramas de componentes relacionados con el portafolio de la plataforma educativa Zera se pueden encontrar en el [Anexo 8: Diagramas de componentes relacionados con los CU del portafolio](#).

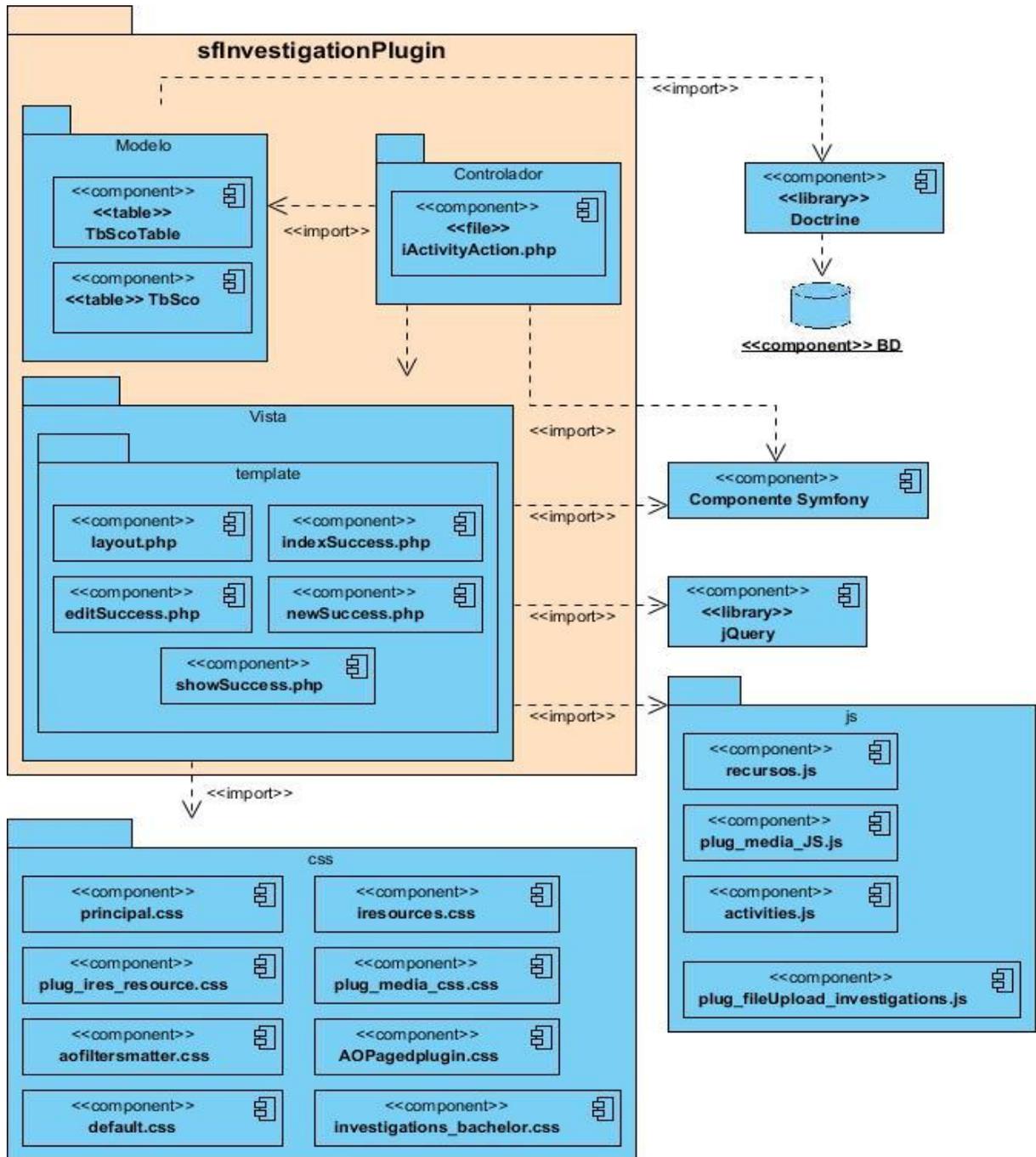


Ilustración 9. Diagrama de componentes del CU Consultar y Gestionar recursos investigativos

## 4.2. Pruebas de calidad del software

Las pruebas del software son un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación. 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 programa de ordenador o videojuego. Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas. (50)

### 4.2.1. Niveles de prueba

Existen diferentes niveles de pruebas que se emplean para la identificación de errores. Algunos de ellos son (51):

- **Pruebas de Integración:** a partir del esquema del diseño, los módulos probados se vuelven a probar combinados para probar sus interfaces. Comprueban la correcta unión de los componentes entre sí a través de sus interfaces, y si cumplen con las funcionalidades establecidas.
- **Pruebas del Sistema:** prueban a fondo el sistema, comprobando su funcionalidad e integridad globalmente, en un entorno lo más parecido posible al entorno final de producción.
- **Pruebas de regresión:** comprueban que los cambios sobre un componente de un sistema de información, no introduzcan un comportamiento no deseado en otros componentes no modificados.

### 4.2.2. Métodos de prueba

La metodología seleccionada (RUP) propone dos métodos de pruebas: caja blanca y caja negra. A continuación se describen ambos métodos:

- **Pruebas de caja blanca:** La prueba de caja blanca, denominada a veces prueba de caja de cristal es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante los métodos de prueba de caja blanca, el ingeniero del software puede obtener casos de prueba que garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo, ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa; ejecuten todos los ciclos en sus límites y con sus límites operacionales, y ejerciten las estructuras internas de datos para asegurar su validez. (52)

- **Pruebas de caja negra:** las pruebas de caja negra, también denominada prueba de comportamiento, se centran en los requisitos funcionales del software. O sea, la prueba de caja negra permite al ingeniero del software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Las pruebas de caja negra se llevan a cabo sobre la interfaz del software, obviando el comportamiento interno y la estructura del programa.(50)

### 4.2.3. Diseño de Casos de prueba

La información referente a los casos de pruebas elaborados se pueden encontrar en el [Anexo 9](#): Diseño de Casos de prueba.

## 4.3 Resultados obtenidos

Durante el desarrollo del módulo Recursos investigativos y Portafolio se realizaron diferentes tipos de pruebas para ir comprobando el funcionamiento del producto final.

Para evaluar la solución desarrollada se realizaron varias iteraciones para demostrar la calidad de la misma. Para ello se aplicó el método de caja negra y los niveles de pruebas expuestos anteriormente en cada una de las iteraciones, en las cuales estuvieron implicados el personal del proyecto y el equipo de calidad arrojando resultados visibles.

A continuación se especifican las pruebas realizadas a la solución:

- **Pruebas internas:** Son las pruebas de calidad realizadas por las analistas de otros años pertenecientes al proyecto, con el fin de entregar un producto con la menor cantidad de errores posibles. Se centraron en el cumplimiento de las funcionalidades descritas en el listado de requerimientos y de los casos de prueba del sistema.
- **Pruebas cruzadas:** Fueron realizadas por el equipo de analistas del proyecto (analistas de diferentes módulos), con el objetivo de encontrar la mayor cantidad de errores posibles en términos de formato de campos, validaciones, entre otras.
- **Pruebas de liberación:** Fueron realizadas por la calidad interna del proyecto con el objetivo de revisar exhaustivamente las funcionalidades y declararlas en estado óptimo para su entrega al cliente.

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

## Pruebas internas: A la documentación

### 1era Iteración

Módulo	Casos de uso	No conformidades	No proceden	Total
Recursos investigativos	2	2	0	2
Portafolio	13	14	0	14

Tabla 6. Pruebas internas: 1era Iteración

### 2da Iteración

Módulo	Casos de uso	No conformidades	No proceden	Total
Recursos investigativos	2	3	1	2
Portafolio	13	4	2	2

Tabla 7. Pruebas internas: 2da Iteración

## Pruebas internas: Al sistema

### 1era Iteración

Módulo	Casos de uso	No conformidades	No proceden	Total
Recursos investigativos	2	4	1	3
Portafolio	13	2	0	2

Tabla 8. Pruebas internas: 1era Iteración

### 2da Iteración

Módulo	Casos de uso	No conformidades	No proceden	Total
--------	--------------	------------------	-------------	-------

Recursos investigativos	2	7	0	7
Portafolio	13	13	0	13

Tabla 9. Pruebas internas: 2da Iteración

Pruebas cruzadas: Al sistema				
Módulo	Casos de uso	No conformidades	No proceden	Total
Recursos investigativos	2	6	0	7
Portafolio	14	9	0	10

Tabla 10. Pruebas cruzadas: 1era Iteración

Pruebas cruzadas: Al sistema				
Módulo	Casos de uso	No conformidades	No proceden	Total
Recursos investigativos	2	8	0	8
Portafolio	14	14	0	14

Tabla 11. Pruebas liberación: 1era Iteración

Todas las NC fueron resueltas.

### 4.3. Pruebas para comprobar las especificaciones implementadas del Leap2A

Para comprobar el correcto funcionamiento del estándar implementado se exportaron los datos de un estudiante de la plataforma educativa Zera y se importó en la herramienta Mahara<sup>12</sup>. Como resultado se importaron todos los datos del portafolio del estudiante, dígame su perfil de usuario, sus evidencias tanto

<sup>12</sup> Mahara: Herramienta para la creación de portafolios electrónicos.

de los programas de estudio que cursa en la plataforma como las extracurriculares y las certificaciones que posee. Además se exportaron las competencias y conocimientos específicos que ha vencido el estudiante. De esta forma se logró cumplir con las especificaciones del estándar, logrando la interoperabilidad entre dos sistemas.

#### **4.4. Conclusiones parciales del capítulo**

Los artefactos generados en esta etapa de desarrollo conformaron el modelo de implementación, el cual describió cómo los elementos del modelo de diseño se implementan y organizan. Una vez desarrolladas todas las funcionalidades previstas, se realizaron diferentes tipos de pruebas que se utilizaron para mostrar que la solución se encontraba libre de errores y poseía la calidad prevista.

## Conclusiones

Con la culminación del presente trabajo de diploma se obtuvieron dos módulos funcionales integrados a la plataforma educativa Zera (Recursos investigativos y Portafolio), entregando toda la documentación generada durante el proceso de desarrollo del software. En sentido general se logró:

1. Realizar una propuesta para cada instrumento de evaluación obteniendo la información necesaria para ser utilizada como evidencias de aprendizaje y desempeño del estudiante
2. Integrar los instrumentos de evaluación identificados logrando de esta forma el intercambio y la recopilación de la información de las evidencias del aprendizaje, el desempeño y las habilidades que va adquiriendo el estudiante a lo largo del tiempo.
3. Conformar un portafolio único donde se almacenan todas las evidencias de aprendizaje y desempeño del estudiante de todos los programas de estudio
4. Convertir al portafolio a un tipo de portafolio de desarrollo personal implementado bajo las especificaciones del estándar Leap2A.
5. Gestionar recursos investigativos desde la creación de los programas de estudio proporcionando un marco de trabajo común para todos los docentes.
6. Conformar mejores expedientes de los estudiantes ya que en él se encuentra toda la información sobre el mismo, no solo su perfil, también su desempeño y comportamiento ante el estudio. Además existe una mayor organización de la información.
7. Brindar la posibilidad de realizar un análisis general del avance del aprendizaje apoyándose en las competencias que debe vencer el estudiante y los resultados arrojados en cada programa de estudio que cursa.

8. Permitir que los procesos de caracterización y selección sean transparentes ya que el portafolio muestra tal y como es el estudiante y los logros que ha alcanzado, sirviendo como base para diferenciarlos unos de los otros.

Todos los logros alcanzados con la realización del portafolio de la plataforma educativa Zera aportaron las siguientes ventajas para el estudiante:

- No pierde las evidencias de aprendizaje, los logros, destrezas y habilidades cuando culmina un programa de estudio.
- Sirve como una herramienta de aval ya que muestra al estudiante tal y como es y los logros que ha alcanzado.

Con respecto a otras plataformas que poseen un portafolio electrónico y a las herramientas que se utilizan para su creación, el portafolio de la plataforma educativa Zera posee dos grandes fortalezas:

1. Muestra y gestiona el aprendizaje basado en competencias.
2. Almacena todas las evidencias de los conocimientos adquiridos por el estudiante en la plataforma de manera automática. Esto significa que el estudiante no se tiene que preocupar por almacenar las evidencias que genera en la plataforma, el portafolio realiza esta labor.

Además:

La solución propuesta se ajusta con la metodología, lenguajes y herramientas seleccionadas para el desarrollo del proyecto.

## Recomendaciones

A continuación se enuncian una serie de recomendaciones a tener en cuenta para futuros trabajos tomando como referencia el actual:

1. Continuar el estudio de los portafolios electrónicos y de los instrumentos de evaluación para fortalecer la plataforma educativa Zera.
2. Incorporar a la plataforma un portafolio para los restantes roles y así lograr que estos también preserven evidencias de su desempeño, habilidades y logros alcanzados.
3. Incluir las opciones para exportar que provee la herramienta para la creación de portafolios Mahara, dígase: exportar como HTML y brindarle la posibilidad al usuario de exportar las secciones que él desee.
4. Convertir al foro en un instrumento de evaluación con las características expuestas y propuesta en el Capítulo III. Además, integrar los nuevos instrumentos que se incorporen a la plataforma con el portafolio.
5. Incorporarle funcionalidades al portafolio de la plataforma Zera que permitan una mayor personalización del mismo.
6. Se recomienda aprovechar los recursos de los que dispone la universidad e implementar un sistema de gestión de portafolios para los estudiantes y profesores o habilitar las herramientas que incluye el Moodle para su creación.

## Referencia bibliográfica

1. Despresbiteris, Léa. *Instrumentos y técnicas de evaluación en la educación media técnico-profesional: La necesidad de una visión más diversificada*. 2008.
2. Espinosa, M<sup>a</sup> Paz Prendes y Sánchez Vera, M<sup>a</sup> del Mar. *Portafolio electrónico: Posibilidades para el docente*. Marzo 2008.
3. B., Mónica María Agudelo. *Plataformas educativas. Programa de integración de tecnologías de la información y la comunicación a la docencia*. [En línea] Programa Integración de Tecnologías a la Docencia, 2008. [Citado el: 15 de Enero de 2012.] <http://aprendeenlinea.udea.edu.co/banco/html/plataformaseducativas/>.
4. administración, Universidad autónoma de baja California. Facultad de contabilidad y. *Formación y actualización docente para un nuevo modelo educativo*. California : s.n., 2008.
5. Aguirre, Francisco Ayala. *El modelo de formación por competencias*.
6. aprendizaje, Portal web del SENA: Servicio nacional de. [www.sena.edu.co](http://www.sena.edu.co). [En línea] 2010. [Citado el: 15 de Enero de 2012.] <http://www.sena.edu.co/Portal/Servicio+al+Ciudadano/Glosario/>.
7. **RESOLUCIÓN No. 210 / 2007**
8. *Instrumentos de evaluación*. García-Garro, Dr. Alberto Javier, y otros. No. 3, Artemisa : Revista Mexicana de Anestesiología, Julio-Septiembre 2007, Vol. Vol. 30. pp 158-164.
9. Sapientiae, Universidad Católica Sedes. *Técnicas e instrumentos de evaluación*. Perú : Dirección nacional de educación secundaria y superior tecnológica DINESST, 2006.
10. Mitjans, Yaima Llaugert y Merño Almaguer, Yusdelkys. *Propuesta de los instrumentos y herramientas para la evaluación del aprendizaje y su aplicación e integración en la plataforma Zera para la gestión del aprendizaje*. La Habana : s.n., 2010.
11. FORTES, Centro de Tecnología para la formación. *Estudio de competencia en el mercado de herramientas de e-learning*. La Habana : s.n., 2011.

12. S.L, eLearning Solutions. *Informe sobre Sakai 2.8, Moodle 2.0 y. 07 de septiembre de 2011.*
13. Espinosa, María Paz Prendes. *Plataformas de campus virtual. Análisis comparativo de la situación actual en las universidades españolas.* Murcia : s.n., 2009. EA-2008-0257.
14. Sánchez, Santiago Gonzáles. *Revisión de plataformas de entorno de aprendizaje.* Pueblo Libre, Lima - Perú : Facultad de Ingeniería de Sistemas, Cómputo y Telecomunicaciones.
15. EDUCASE Learning Initiative. *E- Portafolio s.* [En línea] [Citado el: 21 de Enero de 2012.] <http://www.educause.edu/ELI/Archives/EPortafolio s/5524>.
16. UDLAP. *Conoce a los Portafolios.* [En línea] 2006. [Citado el: 21 de Enero de 2012.] <http://portafolios.udlap.mx/conoce.aspx>.
17. Ricardo, Carmen Tulia, Rivera, Jorge y Santacruz1, Juan Manuel. *Implementación de un Portafolio Educativo Virtual.* Junio 2008. ISSN 2011-575X #6.
18. *El e-Portafolio como estrategia de enseñanza y Aprendizaje.* Gallego, Domingo, Cacheiro, María Luz y Martín, Ana M<sup>a</sup>. 30, s.l. : EDUTEC Revista electrónica de <tecnología educativa, Noviembre del 2009. ISSN 1135-9250..
19. Fernández, Cristina Alconada. [www.dulac.es](http://www.dulac.es). *El portafolio digital.* [En línea] Noviembre de 2005. [Citado el: 14 de Febrero de 2012.] <http://www.dulac.es/Proyectos/El%20Portafolio /EL%20Portafolio %20Digital.htm>
20. Leap2A ePortafolio portability. *Sitio oficial.* [En línea] [Citado el: 21 de Enero de 2012.] <http://www.leapspecs.org/2A/about>
21. J. Gregorio, Ed. *PROPOSED STANDARD. The Atom Publishing Protocol.* October 2007.
22. Carrillo Pérez, Isaías, Pérez González, Rodrigo y Rodríguez Martín, Aureliano David. *Metodología de desarrollo de software.* Octubre 2008.
23. Figueroa, Roberth G. y Cabrera, Armando A. *Metodologías tradicionales vs Metodologías ágiles.*
24. *Guía de RUP.*
25. Orallo, Enrique Hernández. *El lenguaje unificado de modelado (UML).*

26. w3c. Sitio oficial. *Guía Breve de XHTML*. [En línea] [Citado el: 21 de Febrero de 2012.] <http://w3c.es/Divulgacion/GuiasBreves/XHTML>.
27. Pérez, Javier Eguíluz. *Introducción a JavaScript*.
28. Pérez Eguíluz, Javier. *Introducción a AJAX*.
29. *Sitio oficial de PHP*. [En línea] [Citado el: 15 de Febrero de 2012.] php.net.
30. W3C España. *Guía Breve de Tecnologías XML*. [En línea] [Citado el: 6 de Diciembre de 2011.] <http://www.w3c.es/Divulgacion/GuiasBreves/TecnologiasXML>.
31. RichFaces, Jsf, Seam, Jboss y JPA. *Comparativa de Frameworks. Ejemplo práctico de uso*.
32. jQuery. . *Sitio oficial*. [En línea] [Citado el: 6 de Diciembre de 2012.] <http://jquery.com/>.
33. Blueprintcss. *Sitio oficial*. [En línea] <http://blueprintcss.org/>.
34. Potencier, F. *Symfony la guía definitiva*. 2009.
35. doctrine. *Sitio oficial*. [En línea] <http://www.doctrine-project.org/>.
36. Mata, Manel Pérez. tecnoretas. *Qué es Doctrine ORM?* . [En línea] 3 de Julio de 2009. [Citado el: 11 de Febrero de 2012.]
37. *Sistemas gestores de base de datos*. Pérez, María Teresa Garzón. 30, Córdoba : Innovación y experiencias educativas, Mayo 2010. ISSN 1988-6047.
38. Postgres SQL. *Sitio oficial*. [En línea] <http://www.postgresql.org/>.
39. Fredy, Carranza Athó. *Sistema manejador de Base de datos*. Trujillo, Perú : s.n., 2005.
40. PCMAG.com. [En línea] [Citado el: 15 de Enero de 2012.] [http://www.pcmag.com/encyclopedia\\_term/0,1237,t=IDE&i=44707,00.asp](http://www.pcmag.com/encyclopedia_term/0,1237,t=IDE&i=44707,00.asp).
41. Netbeans. *Sitio oficial*. [En línea] <http://netbeans.org/>.
42. María, Jesus. *Herramientas Case*. Colección Cultura Informática. [En línea] [Citado el: 16 de Diciembre de 2012.] <http://www.inei.gob.pe/biblioineipub/bancopub/Inf/Lib5103/Libro.pdf>.
43. Visual Paradigm. *Sitio oficial*. [En línea] <http://www.visual-paradigm.com/product/vpuml/>

44. Rotta, Ing. Luis Zuloaga. *Análisis de requerimientos*.
45. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. *El proceso Unificado de Desarrollo de Software: La guía completa del Proceso Unificado escrita por sus creadores*. 2000.
46. Sebastián, Juan. **Modelo Vista Controlador. Definición y Características**. [En línea] 13 de Noviembre de 2010. [Citado el: 5 de Marzo de 2012.] <http://www.comusoft.com/modelo-vista-controlador-definicion-y-caracteristicas>.
47. Abraham, Haward, Josmary y Sarays. **Bienvenidos al Mundo Symfony**. [En línea] <http://symfony.webs.com/mvc.htm>.
48. Fowler, Martin. *Patterns of Enterprise Application Architecture*. s.l. : Addison-Wesley. ISBN: 0-32112-742-0.
49. James Rumbaugh, Ivar Jacobson, Grady Booch. *El Proceso Unificado de Desarrollo de Software. Manual de Referencia*. s.l. : Addison Wesley.
50. Arrieche, Elizabeth. *Pruebas del sistema*. Enero 2011.
51. Gutiérrez, Javier. Isi. *Introducción al proceso de prueba*. [En línea] [Citado el: 3 de Mayo de 2012.] [http://www.isi.us.es/~javier/cursos\\_ficheros/02.SR.pdf](http://www.isi.us.es/~javier/cursos_ficheros/02.SR.pdf)
52. Wiley, John. *Software Engineering Standards*. 2001. Standard 829-1983.
53. **IMS Global Learning Consortium. Sitio oficial**. [En línea] <http://www.imsglobal.org/background.html>.