

**Universidad de las Ciencias Informáticas**

**Facultad 4**



# **Desarrollo de componentes para la interoperabilidad de la plataforma educativa Zera**

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

**Autores:**

Gisela Sú Rodríguez  
Argel Jesús Carrasco Oliva

**Tutor:**

Ing. Yaismel Miranda Pons

**Co-Tutora:**

MSc. Yaillet Martínez Pérez

**La Habana, junio 2012**



### Declaración de autoría

Declaramos ser autores del presente trabajo de diploma y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

---

Gisela Sú Rodríguez

Firma del Autor

---

Argel J. Carrasco Oliva

Firma del Autor

---

Ing. Yaismel Miranda Pons

Firma del Tutor

---

MSc. Yaillet Martínez Pérez

Firma de la Co-Tutora



## Dedicatoria

**Gisela:**

*A mi mamita, que es la razón por la que me esfuerzo en ser cada día mejor, por guiarme y estar siempre a mi lado, para darle otro motivo para que se sienta orgullosa de su hija.*

*A mi hermanita adorada, que quiero más que nada en el mundo, para que siga mi ejemplo y comprenda que todos los sueños se hacen realidad.*

**Argel:**

*A mi mamá por su apoyo incondicional, dedicación en todo momento y por educarme como un hombre de bien.*

*A mi novia Yaris por estar siempre a mi lado brindándome su amor, dedicación, preocupación, confianza e incondicionalidad, es lo mejor que me ha pasado en la vida.*

*A mi papá que aunque está lejos siempre ha confiado en mí.*

*A mis abuelos que son como unos padres para mí, por su enseñanza.*

*A mi querido hermano.*

*A mis tíos, por su preocupación y ayuda.*

*A mis primas y primos que son como hermanos para mí.*



## Agradecimientos

*De Gisela:*

*A mi mamita querida, por darme la oportunidad de demostrarle lo importante que es para mí, todos mis logros se los debo a ella.*

*A mi hermanita, por estar a mi lado en los momentos difíciles de mi vida, comprenderme y ayudarme siempre.*

*A mi abuelo Gil, que aunque no está entre nosotros se sentiría orgulloso de vivir este momento, por ser un padre para mí y protegerme siempre.*

*A mi Puchito, por demostrarme que no hay nada imposible para alcanzar mis sueños, gracias por apoyarme incondicionalmente y guiarme por el camino correcto, por confiar en mí y ser mi alma gemela.*

*A toda mi familia, por ayudarme y consentirme, sin su apoyo mi vida hubiera sido diferente.*

*A mis amigos y compañeros durante estos cinco años de carrera, en especial a Yise por aprender a entenderme.*

*A mi compañero de tesis, por enseñarme que con deseos se pueden lograr muchas cosas.*

*A los profes del proyecto y a todos los que contribuyeron a lo largo de la carrera para llegar este momento.*



*De Argel:*

*A mi mamá, por todo su esfuerzo y dedicación.*

*A mi novia Yaris por su preocupación, confianza, amor y por estar siempre a mi lado apoyándome en todo momento, es el mejor regalo que me ha dado la vida.*

*A mi papá, que aunque está lejos sé que tiene fe y confianza en mí.*

*A mis abuelos, por su preocupación y educación.*

*A mis tíos Daniel y Elizabeth y a mis primas, que siempre han estado presentes cuando los he necesitado.*

*A Maikel y Liannys, por ser amigos incondicionales, prestos a dar su mano en todo momento.*

*A todos mis amigos de la universidad, con los que compartí día a día.*

*A mi tutor y compañera de tesis por su preocupación, dedicación y esfuerzo.*

*A todas las personas que de una forma u otra hicieron posible que hoy alcanzara este sueño.*

*A la UCI por haberme formado como un profesional.*



### Resumen

El avance de las Tecnologías de la Información y las Comunicaciones, ha tenido un impacto significativo en el proceso de enseñanza-aprendizaje, incorporando nuevas herramientas que permiten una mejor comunicación e interacción entre alumnos y docentes, entre las que se destacan las plataformas de gestión de aprendizaje. La Universidad de la Ciencias Informáticas no está exenta a estos nuevos cambios, trazándose como meta implementar una de estas plataformas bautizada bajo el nombre de Zera, constituyendo el objetivo de este trabajo desarrollar componentes para la interoperabilidad de la plataforma educativa Zera con otras herramientas e-learning. Durante el desarrollo de la investigación se estudiaron los principales estándares, tecnologías y herramientas utilizadas por plataformas educativas existentes para interoperar con aplicaciones e-learning, seleccionando las adecuadas para la solución. Haciendo uso de la metodología de desarrollo Proceso Unificado de Rational, se realizaron los artefactos propuestos por los flujos de trabajo modelamiento de negocio, requisitos, análisis y diseño, implementación y prueba. Finalmente se obtuvieron componentes funcionales agrupados en forma de plugins, aplicando tecnologías y estándares de interoperabilidad entre sistemas que ofrecen dentro de sus principales funcionalidades: la gestión de los servicios web externos y los que provee la plataforma, el lanzamiento de aplicaciones externas dentro de la misma, así como la búsqueda y el almacenamiento de objetos de aprendizaje en repositorios.



## Índice

Introducción.....	2
Capítulo 1: Fundamentación Teórica .....	7
1. Introducción.....	7
2. Aprendizaje electrónico.....	7
2.1. Herramientas e-learning .....	8
3. Interoperabilidad entre PGA y otras herramientas e-learning .....	11
3.1. ¿Qué es un estándar? .....	12
3.2. Estándares para la interoperabilidad .....	12
3.3. ¿Qué son los servicios web? .....	18
3.4. Soluciones similares.....	23
4. Entornos de desarrollo .....	25
4.1. Metodología de desarrollo de software.....	25
4.2. Lenguaje de modelado .....	26
4.3. Herramienta CASE.....	26
4.4. Lenguajes de desarrollo .....	27
4.5. Frameworks para el desarrollo.....	29
4.6. Entorno de Desarrollo Integrado .....	30
4.7. Sistema Gestor de Base de Datos .....	31
4.8. Servidor Web .....	31
5. Conclusiones del capítulo.....	32
Capítulo 2: Características del Sistema .....	33
1. Introducción.....	33



2.	Modelo de dominio.....	33
3.	Descripción del sistema propuesto .....	34
4.	Requisitos del software .....	34
4.1.	Requisitos funcionales.....	34
4.2.	Requisitos no funcionales .....	38
5.	Modelo de casos de uso del sistema .....	41
5.1.	Patrones de caso de uso .....	41
5.2.	Descripción de los actores del sistema.....	42
5.3.	Diagrama de casos de uso del sistema .....	42
5.4.	Descripción de los casos de uso del sistema.....	43
6.	Conclusiones parciales .....	45
Capítulo 3: Análisis y diseño del Sistema.....		46
1.	Introducción.....	46
2.	Modelo de análisis .....	46
2.1.	Diagrama de clases del análisis .....	47
2.2.	Diagrama de colaboración .....	47
3.	Patrón arquitectónico Modelo – Vista – Controlador .....	47
4.	Patrones de diseño .....	49
5.	Patrones de servicio.....	50
6.	Modelo de diseño.....	50
6.1.	Diagrama de clases del diseño .....	50
7.	Modelo de datos .....	52
8.	Diagrama de despliegue.....	53





9. Conclusiones Parciales .....	54
Capítulo 4: Implementación y Prueba .....	55
1. Introducción.....	55
2. Modelo de implementación.....	55
2.1. Diagrama de componentes .....	55
3. Pruebas de software .....	56
3.1. Estrategia de pruebas .....	57
3.2. Métodos de prueba .....	57
4. Conclusiones Parciales .....	61
Conclusiones Generales .....	62
Recomendaciones .....	63
Referencias bibliográficas .....	64
Bibliografía consultada.....	68
Glosario de términos.....	70



### Introducción

El crecimiento acelerado de las Tecnologías de la Información y las Comunicaciones (TIC) en la actualidad y su uso como medio de apoyo al proceso de enseñanza-aprendizaje, favorece el desarrollo de instrumentos, aplicaciones y métodos de enseñanza que permiten una mejor comunicación e interacción entre alumnos y docentes, además de ofrecer una nueva opción dentro de la modalidad de educación a distancia: aprendizaje electrónico o e-learning.

El e-learning, utiliza herramientas o medios electrónicos para llevar a cabo los procesos de enseñanza-aprendizaje a través de la red, donde se destacan las plataformas de gestión de aprendizaje (en lo adelante PGA), también conocidas como Sistemas de Gestión de Aprendizaje (en inglés Learning Management System, LMS), que son plataformas encargadas de administrar, distribuir y controlar los procesos de enseñanza-aprendizaje en línea. Además de proporcionar funciones administrativas y de seguimiento que posibilitan y controlan el acceso a los contenidos, implementan recursos de comunicación y llevan a cabo el seguimiento de los usuarios que utilizan la herramienta. [1]

En la Universidad de las Ciencias Informáticas (UCI), específicamente en el Centro de Tecnologías para la Formación (FORTES) de la facultad 4, se desarrolla una PGA nombrada Zera, que tiene su origen en la concepción pedagógica denominada hiperentornos de aprendizaje, con la característica de integrar en un mismo producto elementos representativos de diversas tipologías de software educativo como: multimedias, simuladores, laboratorios virtuales, asistentes matemáticos, libros electrónicos, juegos instructivos, entre otros.

La plataforma educativa Zera como medio de apoyo al proceso de enseñanza-aprendizaje gestiona disímiles recursos que son utilizados para enriquecer los contenidos mostrados a los estudiantes, además de usar herramientas como los simuladores para constatar en la práctica los conocimientos obtenidos. Sin embargo, actualmente no garantiza de manera eficiente la interacción y el intercambio de recursos que puedan ser reutilizados con otras aplicaciones, así como la integración de las actividades realizadas en herramientas de apoyo a los procesos desarrollados en la plataforma; problema conocido como interoperabilidad.

Como estrategia de integración y para proveer funcionalidades a la plataforma que permitan interoperar con otras aplicaciones de apoyo a los procesos de enseñanza-aprendizaje, se estandarizó un modelo conocido como JURL, basado en los principios de REST (acrónimo de Representational State Transfer),



que utiliza los métodos del protocolo HTTP (en inglés Hypertext Transfer Protocol). Esta solución es utilizada para integrar los procesos del simulador físico matemático llamado FISIM y el asistente matemático MATGRAF con la plataforma educativa Zera. No obstante, la misma no resuelve todas las necesidades identificadas hasta el momento, debido a las siguientes deficiencias:

- No implementa estándares eficientes de seguridad.
- Se basa en un único protocolo de intercambio de datos.
- No facilita la búsqueda y el almacenamiento de recursos educativos en repositorios de objetos de aprendizaje (en lo adelante ROA).
- No estandariza una estructura de mensajes para el intercambio de información.
- No permite el intercambio de información referente al proceso de enseñanza-aprendizaje con sistemas de gestión académica (en lo adelante SGA).

Partiendo del análisis de la situación planteada, se define el siguiente **problema de investigación**:  
¿Cómo facilitar la interacción y el intercambio de recursos e información de forma segura entre la plataforma educativa Zera y otras herramientas e-learning?

El **objeto de estudio** de la presente investigación es la interoperabilidad entre plataformas de gestión de aprendizaje y otras herramientas e-learning.

El **campo de acción** se centra en la interoperabilidad de la plataforma educativa Zera con otras herramientas e-learning.

El **objetivo general** es desarrollar componentes para la interoperabilidad de la plataforma educativa Zera con otras herramientas e-learning.

A partir del objetivo general definido, se derivan los siguientes **objetivos específicos**:

- Realizar el **análisis** de componentes que permitan la interoperabilidad de la plataforma educativa Zera con otras herramientas e-learning.
- **Diseñar** componentes que permitan la interoperabilidad de la plataforma educativa Zera con otras herramientas e-learning.
- **Implementar** y **probar** componentes que permitan la interoperabilidad de la plataforma Educativa



Zera con otras herramientas e-learning.

### **Idea a defender:**

El desarrollo de componentes para la interoperabilidad de la plataforma educativa Zera, facilitará la interacción y el intercambio de recursos e información de forma segura con otras herramientas e-learning.

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

- Estudio de los principales conceptos y tecnologías utilizadas en el desarrollo de servicios web.
- Análisis del estado actual de los estándares de interoperabilidad entre PGA y otros sistemas e-learning.
- Análisis y definición de la metodología de desarrollo a utilizar, así como las herramientas de modelado.
- Análisis y definición de los lenguajes de programación a utilizar, así como las herramientas adecuadas que cumplan los requisitos para ser utilizadas en el desarrollo.
- Identificación y especificación de los casos de uso del sistema.
- Elaboración del modelo de casos de uso que da cumplimiento a los requisitos funcionales y no funcionales asociados a las funcionalidades a desarrollar.
- Elaboración de diagramas de clases de análisis y diseño para cada caso de uso del sistema.
- Diseño del modelo de datos.
- Elaboración del modelo de despliegue.
- Confección de los diagramas de componentes del sistema.
- Implementación de funcionalidades que permitan la interoperabilidad de la plataforma educativa Zera con otras herramientas e-learning.
- Validación de la propuesta de solución.

El **aporte de la investigación** radica en permitir que otras aplicaciones e-learning interoperen con la plataforma educativa Zera, añadiendo a la misma funcionalidades basadas en estándares de interoperabilidad, servicios web y mecanismos de seguridad.



Lo **novedoso de la investigación** está reflejado en la incorporación de estándares de interoperabilidad entre sistemas a la plataforma educativa Zera, que permiten la interacción y el intercambio de recursos e información entre diferentes aplicaciones e-learning.

Con el fin de resolver y dar cumplimiento a los objetivos y las tareas propuestas se emplearon los siguientes **métodos de investigación**:

### **Métodos teóricos:**

- **Analítico - sintético:** En la investigación se utilizó este método para realizar un análisis teórico e identificar los principales conceptos a incluir en la fundamentación teórica, además del estudio y análisis de la información, permitiendo extraer elementos importantes relacionados con el objeto de estudio.
- **Histórico - lógico:** Durante la investigación este método permitió estudiar cómo ha evolucionado el uso de estándares de interoperabilidad entre sistemas e-learning, así como identificar las tecnologías, los lenguajes y la metodología de desarrollo a utilizar en el desarrollo de la solución propuesta.
- **Modelación:** Este método sirvió de apoyo en la realización de los artefactos correspondientes al análisis, diseño e implementación de las funcionalidades propuestas.

### **Métodos empíricos:**

- **Observación:** Este método permitió estudiar cómo funciona el proceso de interoperabilidad entre aplicaciones diferentes y los principales problemas asociados a este, posibilitando estudiar de cerca el objeto de la investigación.



### **Estructura capitular:**

#### **Capítulo I:** Fundamentación Teórica.

En este capítulo se exponen los elementos teóricos que sustentan la investigación. Se realiza un estudio de los estándares de interoperabilidad entre sistemas e-learning y las soluciones existentes en plataformas similares. Además de hacer un análisis y seleccionar en base al estudio realizado la metodología, las herramientas y las tecnologías adecuadas para el desarrollo de la investigación.

#### **Capítulo II:** Características del Sistema.

En este capítulo se describen las características del sistema propuesto. Se determinan los requerimientos funcionales y no funcionales, con los que se realiza el diagrama de casos de uso del sistema. Además, se obtiene el modelo de dominio y las descripciones correspondientes a cada caso de uso.

#### **Capítulo III:** Análisis y diseño del Sistema.

En este capítulo se realiza el análisis y diseño del sistema. Se obtienen los diagramas de clases del análisis, de colaboración, así como los diagramas de clases con estereotipos web que brindan una visión clara del producto. Además, se diseña el modelo de datos y se realizan las descripciones correspondientes.

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

Los aspectos del desarrollo son tratados en este capítulo. Se describe cómo está implementado el sistema en términos de componentes. Además, se definen los tipos de pruebas y los casos de prueba realizados al software.



### Capítulo 1: Fundamentación Teórica

#### 1. Introducción

El auge que ha adquirido el aprendizaje electrónico en los últimos años ha impulsado el desarrollo de numerosas herramientas que fortalecen los procesos de enseñanza-aprendizaje, propiciando que las instituciones educativas cuenten con recursos, información y aplicaciones heterogéneas distribuidas en la red. Con el objetivo de compartir y lograr comunicar todos estos recursos entre aplicaciones diferentes, existen estándares y especificaciones que dan respuesta a este problema conocido como interoperabilidad, haciéndose necesario realizar un estudio de los mismos y de las tecnologías idóneas para su implementación, fundamentando de este modo las bases de la investigación.

#### 2. Aprendizaje electrónico

El desarrollo de las TIC ha impulsado importantes cambios en el proceso de enseñanza-aprendizaje tradicional, dando paso a una nueva modalidad educativa conocida como educación a distancia, donde el alumno tiene posibilidades de capacitación y aprendizaje de forma autónoma.

En este ámbito surge el aprendizaje electrónico, conocido como e-learning, que permite reducir los obstáculos de espacio y tiempo utilizando las tecnologías como intermediarias para elevar la productividad y la flexibilidad del proceso de enseñanza-aprendizaje. Muchos son los conceptos y definiciones dados sobre qué es el e-learning, como es el caso de Mariano Bernárdez en su libro *Diseño, producción e implementación de e-learning*, donde lo define como “todas aquellas metodologías, estrategias o sistemas de aprendizaje que emplean tecnología digital y/o comunicación mediada por ordenadores para producir, transmitir, distribuir y organizar conocimiento entre individuos, comunidades y organizaciones”. [2]

También es definido como la “capacitación no presencial que, a través de plataformas tecnológicas, posibilita y flexibiliza el acceso y el tiempo en el proceso de enseñanza-aprendizaje, adecuándolos a las habilidades, necesidades y disponibilidades de cada discente, además de garantizar ambientes de aprendizaje colaborativos mediante el uso de herramientas de comunicación síncrona y asíncrona, potenciando en suma el proceso de gestión basado en competencias”. [3]

Para la investigación se toma como referencia esta última definición, centrándose en la necesidad de intercambiar entre las diferentes herramientas existentes dentro de un entorno e-learning la información y los recursos distribuidos que almacenan.



### 2.1. Herramientas e-learning

Dentro del gran número de herramientas y recursos tecnológicos que sirven de apoyo a los procesos de enseñanza-aprendizaje se encuentran las PGA y los ROA, que permiten diseñar y elaborar los recursos necesarios para la gestión, administración y evaluación de las actividades educativas. Aplicaciones que facilitan la gestión de las actividades académicas de estudiantes como los SGA. Además de laboratorios virtuales, asistentes matemáticos y simuladores utilizados para llevar a la práctica los conocimientos adquiridos.

Después de realizar un estudio sobre las diferentes definiciones y posturas adoptadas por varios autores sobre las herramientas antes mencionadas, para la presente investigación se asumen los conceptos expuestos a continuación.

#### 2.1.1. Plataformas de gestión de aprendizaje

Las PGA, también conocidas como LMS, constituyen una herramienta informática y telemática organizada en función de objetivos formativos de forma integral (es decir que se pueden conseguir exclusivamente dentro de ella) y de principios de intervención psicopedagógicos y organizativos [4], integrando un conjunto de herramientas que permiten administrar el sistema, la comunicación y colaboración entre usuarios, la evaluación del aprendizaje, el almacenamiento de la información y la gestión de contenidos para los procesos de enseñanza-aprendizaje en línea, ya sea completamente a distancia, presencial, o que combine ambas modalidades.

De manera general, permiten la capacitación no presencial a través de tecnologías que posibilitan y flexibilizan el acceso y el tiempo en el proceso de enseñanza-aprendizaje, adecuándolo a las habilidades, necesidades y disponibilidades de cada alumno o docente.

#### Plataforma educativa Zera

La plataforma educativa Zera, es una PGA que surge con el objetivo de adaptar los productos de la Colección Futuro del Ministerio de Educación en Cuba (MINED) a las escuelas mexicanas. Su desarrollo se basa en la concepción pedagógica denominada hiperentorno<sup>1</sup> de aprendizaje, propuesta y desarrollada por pedagogos y especialistas del MINED en el año 2001.

---

<sup>1</sup> **Hiperentorno:** Mezcla de diferentes tipos de software educativos sustentados en tecnología hipertexto que constituyen un medio de apoyo al proceso de enseñanza-aprendizaje.





La primera versión de esta plataforma integra estándares y especificaciones utilizados en soluciones similares como SCORM<sup>2</sup> 2004, en su tercera edición y QTI<sup>3</sup> 2.0, implementada sobre tecnologías y herramientas libres. Se encuentra estructurada en seis subsistemas estrechamente relacionados: Gestión de materias, Administración, Gestión de recursos interactivos, Administración del aprendizaje, Reportes y Bachiller. Este último constituye el núcleo fundamental de la plataforma, desde el cual es posible crear, estructurar y visualizar cada hiperentorno, además de gestionar cada uno de sus módulos. También se integran a este subsistema las funcionalidades de un asistente matemático y un simulador de apoyo a las asignaturas de Física y Matemática.

### 2.1.2. Repositorios de objetos de aprendizaje

Un objeto de aprendizaje (en lo adelante OA) es cualquier recurso con intención formativa, compuesto por uno o varios elementos digitales descritos con metadatos<sup>4</sup> que puede ser utilizado y reutilizado dentro de un entorno e-learning. Los ROA facilitan la colección o almacenamiento, gestión y reutilización de OA a través de la red, además de almacenar ubicaciones de objetos alojados en otros sitios, tanto en línea como en ubicaciones locales. [5]

Para acceder a los recursos de un ROA, varias instituciones se han encargado de estandarizar mecanismos de acceso que hacen posible a sistemas, herramientas y usuarios utilizar un recurso sin necesidad de emplear tiempo en crearlo.

### 2.1.3. Sistemas de gestión académica

Los SGA constituyen una potente herramienta para almacenar y procesar la información referente al proceso docente de una institución educativa, así como el intercambio de los datos almacenados con otros sistemas como las PGA.

Por lo general engloban todo lo relacionado a la matrícula y prematrícula de los estudiantes, el control de las evaluaciones y la asistencia de los mismos, abarcan la gestión de los planes de estudios, las carreras, los horarios de clases, la información de los profesores y otros datos académicos importantes. [6]

---

<sup>2</sup> **SCORM**: Shareable Content Object Reference Model. Conjunto de especificaciones para el desarrollo, empaquetamiento y distribución de material educativo.

<sup>3</sup> **IMS QTI**: Question and Test Interoperability. Estándar que permite la interoperabilidad entre sistemas que trabajan con objetos de aprendizaje, centrándose en la definición de un modelo para preguntas y pruebas, incluyendo también respuestas.

<sup>4</sup> **Metadato**: Información estandarizada que describe las características principales de un recurso.



### **2.1.4. Laboratorio virtual**

Un laboratorio virtual es un sistema informático que pretende simular el ambiente de un laboratorio real mediante simulaciones interactivas que permiten desarrollar prácticas de laboratorio a distancia, y así ayudar en el tratamiento de conceptos básicos, observar, investigar, realizar actividades, además de apoyar al estudiante en la elaboración e intercambio de resultados desde cualquier lugar sin necesidad de cumplir con un horario preestablecido. [7]

Estos surgen con la intención de crear medios de apoyo al estudiante que le permitan complementar en la práctica los conocimientos teóricos adquiridos, además de que muchos integran sus resultados con aplicaciones de apoyo a los procesos de enseñanza-aprendizaje que tienen en cuenta las evaluaciones obtenidas en estas prácticas como parte del plan de estudio.

### **2.1.5. Asistente matemático**

Un asistente matemático, es un programa informático utilizado como instrumento pedagógico para apoyar los procesos de enseñanza-aprendizaje que facilita a los estudiantes el estudio de funciones representadas de forma gráfica.

El uso de esta herramienta brinda la posibilidad de experimentar y comprender el verdadero alcance de un teorema o la efectividad de un algoritmo, analizando los resultados que se obtienen al variar las hipótesis, condiciones iniciales y otros parámetros. Además de que las representaciones gráficas permiten la comprensión de muchos conceptos, así como el aprendizaje autónomo del estudiante. [8]

### **2.1.6. Simulador**

Los simuladores utilizados como herramienta de apoyo al proceso de enseñanza-aprendizaje, son “objetos de aprendizaje que mediante un programa de software intentan modelar parte de una réplica de los fenómenos de la realidad y su propósito es que el usuario construya conocimiento a partir del trabajo exploratorio, la inferencia y el aprendizaje por descubrimiento. Estos se desarrollan en un entorno interactivo que permite al usuario modificar parámetros y ver cómo reacciona el sistema ante un cambio producido.” [9]

En general, se caracterizan por tratar fenómenos físicos basados en modelos matemáticos presentes en la bibliografía docente del nivel de educación correspondiente.



En la práctica, se tiene la tendencia de crear entornos interoperables que permitan compartir e intercambiar los recursos e información que ofrecen estas herramientas e-learning. La presente investigación se centra en la interoperabilidad de forma transparente de la plataforma educativa Zera con herramientas como ROA, SGA, laboratorios virtuales, asistentes matemáticos y simuladores. De este modo se podrán intercambiar los recursos educativos presentes en la plataforma con ROA, utilizar la información de estudiantes almacenadas en SGA y registrar el resultado de las actividades realizadas en simuladores, asistentes matemáticos y laboratorios virtuales para cada estudiante.

### **3. Interoperabilidad entre PGA y otras herramientas e-learning**

La interoperabilidad ha sido definida y caracterizada por diferentes autores e instituciones. Según el IEEE (acrónimo de Institute of Electrical and Eletronic Engineers) es “la habilidad de dos o más sistemas o componentes para intercambiar información y para usar la información que ha sido intercambiada” [10].

En el boletín trimestral del Instituto Tecnológico de Informática se define como “la habilidad de compartir e intercambiar información usando sintaxis y semánticas comunes para cumplir una relación funcional de una aplicación específica a través del uso de una interfaz común”. [11]

También ha sido definida como la capacidad que tiene un producto o un sistema, cuyas interfaces son totalmente conocidas para funcionar con otros productos o sistemas existentes, sin restricción de acceso o de implementación. [12]

En tanto el comité técnico de la Organización Internacional de Estandarización (ISO) se refiere a la interoperabilidad como la capacidad de un sistema o componente de un sistema para proporcionar intercambio de información y el procesamiento cooperativo entre aplicaciones. [13]

Teniendo en cuenta las definiciones anteriores, para el marco de la presente investigación se define interoperabilidad como la habilidad para intercambiar y compartir información entre sistemas informáticos con características, funciones y tecnologías distintas a través de interfaces conocidas.

En las PGA, la interoperabilidad se logra a través del uso de estándares, especificaciones y servicios que facilitan la interacción y el intercambio de recursos educativos desarrollados en diferentes herramientas e-learning ubicadas en distintas localizaciones.



### 3.1. ¿Qué es un estándar?

Los estándares son acuerdos internacionales documentados o normas establecidas por consenso mundial sobre cómo realizar algo. Estos solo pueden ser definidos una vez que las especificaciones han sido determinadas, discutidas y validadas, llegándose a un acuerdo para su estandarización por instituciones ampliamente reconocidas como por ejemplo el IEEE/LTSC (en inglés Learning Technologies Standardization Committee) perteneciente al IEEE o la ISO [14].

Se puede decir, que los estándares e-learning son patrones, modelos o reglas relacionados con el campo del aprendizaje electrónico, proporcionando contenidos reutilizables, comunicación e intercambio de información con otros sistemas, así como la extensión de los servicios y las capacidades de estos [15]. Existen dos tipos de estándares; estándares de jure, cuando son certificados por una organización acreditada, y estándares de facto, cuando la especificación se adopta por un grupo mayoritario de individuos sin haber pasado por un proceso formal de acreditación.

De esta forma, los estándares en e-learning están encaminados a servir como facilitadores de la durabilidad y de la reutilización en el tiempo de los contenidos y de la interoperabilidad, es decir, facilitar el intercambio de los contenidos entre diversas plataformas y sistemas.

### 3.2. Estándares para la interoperabilidad

En la actualidad existen disímiles estándares orientados hacia una forma común de identificar, definir y comunicar los recursos involucrados en un entorno e-learning. A continuación se hace un análisis de los principales estándares de interoperabilidad entre herramientas e-learning con el objetivo de utilizar los apropiados para el desarrollo de la investigación.

#### 3.2.1. Interfaz de Consulta Simple

El estándar Interfaz de Consulta Simple (en inglés Simple Query Interface, SQI), facilita la interoperabilidad entre ROA y aplicaciones e-learning con el objetivo de realizar búsquedas y recuperar OA distribuidos en distintos repositorios de la red. En su especificación, propone una API<sup>5</sup> para acceder a los servicios disponibles en los ROA, definidos mediante métodos y funciones que permiten establecer sesiones y realizar consultas síncronas y asíncronas desde otras aplicaciones e-learning.

---

<sup>5</sup> **API:** Interfaz de Programación de Aplicaciones. Conjunto de funciones y procedimientos, que expone cierta biblioteca para ser utilizados por otro software como una capa de abstracción.



Para poder realizar una consulta con SQI a los ROA, en primer lugar la fuente de consulta debe establecer una conexión con el destino a través de una sesión que posibilita la comunicación entre ambos sistemas. Las sesiones pueden ser persistentes, es decir, se mantienen indefinidamente, o no persistentes, mantenidas un período de tiempo limitado. Una vez que una sesión ha sido establecida, la interfaz de consulta en el destino espera por peticiones de búsqueda desde la fuente. [16]

En general, SQI provee una interfaz simple y fácil de implementar a través de servicios web básicos como la gestión de sesión, la autenticación y servicios de aplicación para la gestión de consultas que garantizan la interoperabilidad con ROA, mediante un conjunto de métodos que se muestran agrupados por categorías a continuación [17]:

Categoría	Descripción	Métodos
<b>Query Parameter Configuration</b>	Métodos para la configuración de los parámetros de las consultas.	setQueryLanguage() setMaxQueryResults() setMaxDuration() setResultsFormat()
<b>Synchronous Query Methods</b>	Métodos que permiten retornar los resultados de las consultas, seleccionar el número de resultados retornados por una consulta y conocer el número total de resultados de una consulta.	setResultSetSize() synchronousQuery() getTotalResultsCount()
<b>Asynchronous Query Methods</b>	En estos métodos el destino envía los resultados de consultas hacia la fuente y la fuente indica la localización antes de enviar una consulta asíncrona.	setSourceLocation() asynchronousQuery() queryResultsListener()

**Tabla 1.1 Descripción de los métodos del estándar SQI**

### 3.2.2. Iniciativa de Conocimiento Abierto

La Iniciativa de Conocimiento Abierto (en inglés Open Knowledge Initiative, OKI), define una arquitectura abierta y extensible para la interoperabilidad entre diferentes sistemas e-learning. Es una iniciativa que surge de la colaboración del Instituto Tecnológico de Massachusetts (MIT) con otras instituciones de estandarización y varias universidades, con el propósito de promover especificaciones que describen cómo los componentes de un entorno e-learning pueden comunicarse.



La especificación brinda un conjunto de interfaces agrupadas en servicios. Cada uno de estos servicios se denominan OSID (acrónimo de Open Service Interface Definition), que son simplemente varias API conceptuales expresadas en distintos lenguajes de programación, compatibles con otras tecnologías y especificaciones como SOAP (acrónimo de Simple Object Access Protocol) o WSDL (acrónimo de Web Services Description Language). [18]

### Arquitectura OSID de OKI

Las diferentes OSID de OKI describen las funcionalidades que ofrece cada servicio, cómo deben ser utilizadas y las restricciones para su implementación, permitiendo que cada servicio sea desarrollado y actualizado de forma independiente sin afectar el resto de la aplicación.

La especificación está formada por dieciocho OSID, aunque no es necesario implementar todas las interfaces o los métodos que propone cada interfaz. A continuación se describen algunas de estas guías OSID [19]:

- **osid\_authentication:** Soporta invocar procesos de autenticación. La implementación de estos servicios son responsables de garantizar cualquier información apropiada para llevar a cabo la autenticación. Los servicios que maneja esta OSID permiten probar si un usuario es autenticado, para ello devuelve el identificador que corresponde a cada usuario, los tipos de autenticación posible y además destruye todas las autenticaciones o solo aquellas de un tipo dado.
- **osid\_authorization:** Provee una forma de definir quién está haciendo qué y cuándo.
- **osid\_repository:** Abarca el almacenamiento y recuperación de contenido digital en repositorios, ya sean documentos, material de cursos, imágenes, entre otros.
- **osid\_logging:** Se utiliza cuando las aplicaciones necesitan leer y grabar ciertos tipos de trazas para poder hacer una serie de controles al sistema y así obtener datos de seguimiento. Un mecanismo de este tipo tiene que permitir diferentes formatos de trazas y prioridad entre las mismas.
- **osid\_messaging:** Servicio que permite el envío de mensajes para comunicarse entre diferentes componentes o aplicaciones.
- **osid\_configuration:** Brinda soporte a la hora de configurar, administrar y visualizar los valores de configuración que tendrán las aplicaciones y los servicios.



### 3.2.3. Interoperabilidad de Repositorios Digitales

La Interoperabilidad de Repositorios Digitales de IMS<sup>6</sup> (en inglés IMS Digital Repositories Interoperability, IMS-DRI), provee recomendaciones para la interoperabilidad de los ROA. Define una arquitectura funcional para facilitar el acceso a los contenidos almacenados en los repositorios desde herramientas e-learning como LMS o LCMS (en inglés Learning Content Management Systems, Sistema de Gestión de Contenidos de Aprendizaje).

Esta especificación se propone para la interoperabilidad entre servicios o aplicaciones que tienen las funciones comunes de un repositorio: buscar, exponer, coleccionar, enviar, almacenar, pedir, entregar y alertar [5]. Entre las funciones básicas, se reconocen cinco combinaciones principales que se muestran a continuación [20]:

<b>Función</b>	<b>Descripción</b>
<b>Buscar/Exponer (Search/Expose)</b>	Ejecuta la búsqueda de los metadatos asociados a los contenidos que el repositorio expone.
<b>Colectar/Exponer (Gather/Expose)</b>	Define la solicitud de los metadatos que el repositorio expone, la agregación de los metadatos para utilizarse en búsquedas subsecuentes y la agregación de metadatos para crear nuevos repositorios.
<b>Enviar/Almacenar (Submit/Store)</b>	Se enfoca en la manera que un objeto se mueve a un repositorio desde un sitio accesible por red y cómo el objeto será representado en el repositorio para su acceso.
<b>Pedir/Entregar (Request/Deliver)</b>	Posibilita, una vez que el usuario ha localizado los metadatos en la función Buscar, solicitar al repositorio acceso para el recurso.
<b>Alertar/Exponer (Alert/Expose)</b>	Función clave que permite a través del correo electrónico notificar a los usuarios sobre los eventos ocurridos en el repositorio.

**Tabla 1.2 Descripción de las funciones básicas del estándar IMS-DRI**

<sup>6</sup> **IMS:** Instructional Management Systems. Consorcio internacional formado por varias instituciones encargado de definir y distribuir especificaciones para actividades de educación en línea.



### 3.2.4. Interoperabilidad de Herramientas de Aprendizaje

La Interoperabilidad de Herramientas de Aprendizaje de IMS (en inglés IMS Learning Tool Interoperability (IMS-LTI), provee un marco único o la forma estándar para integrar aplicaciones de apoyo a los procesos de enseñanza-aprendizaje en un LMS. De este modo se podrán consumir aplicaciones externas, como si formaran parte del sistema.

La especificación IMS- LTI define dos tipos de integración [21]:

**Full LTI:** Implica un proceso formal, donde la herramienta consumidora (por lo general un LMS) y el proveedor de la herramienta llegan a un acuerdo acerca de los servicios que se pueden ejecutar en tiempo de ejecución, las políticas de seguridad a aplicar y los destinos desde donde la herramienta será ejecutada en el sistema consumidor.

**Basic LTI:** Expone un único destino en la herramienta proveedora. El procedimiento para establecer un vínculo al destino se basa en un enlace simple utilizando los servicios en tiempo de ejecución, mediante una clave y un secreto compartido por la herramienta que consume los servicios y la que los provee. Además de utilizar el protocolo OAuth<sup>7</sup> para la seguridad en el flujo de mensajes.

El funcionamiento de una aplicación lanzada dentro de un LMS haciendo uso del estándar IMS-Basic LTI se resume en la figura siguiente:

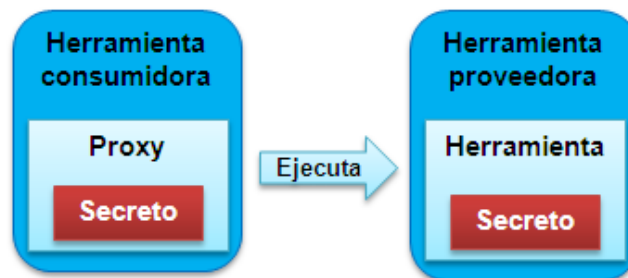


Figura 1.1 Funcionamiento del estándar IMS-Basic LTI

La herramienta consumidora (en lo adelante HC) identifica al LMS que utiliza la herramienta proveedora (en lo adelante HP). Ambas herramientas comparten una clave y un secreto que permite a la HC firmar los

<sup>7</sup> **OAuth** (<http://www.oauth.net>): Mecanismo de seguridad diseñado para proteger las peticiones por POST y GET.





datos que se envían a la HP basándose en el protocolo OAuth. Una vez verificados estos datos por la HP, se ejecutará la aplicación.

### 3.2.5. Interfaz de Publicación Simple

La Interfaz de Publicación Simple (en inglés Simple Publishing Interface, SPI) facilita la publicación de OA y metadatos en un ROA. Para ello, define una API formada por un conjunto de métodos que requieren la creación de una sesión en el repositorio donde se va a publicar el OA. Los métodos propuestos en su especificación pueden ser de dos tipos: orientados a la publicación de OA y orientados a la publicación de metadatos.

Estos se resumen en la siguiente tabla [16]:

Tipo	Descripción	Métodos
<b>Orientados a la publicación de OA.</b>	Métodos que permiten definir el tipo y la localización de un OA que va ser almacenado en un ROA, así como la publicación o eliminación del mismo.	setDataFormat() setSourceLocation() submitResourceByValue() submitResourceByReference() notifyRetrievalStatus() deleteResource()
<b>Orientados a la publicación de metadatos.</b>	Métodos que permiten controlar el esquema de los metadatos que serán publicados, así como el envío o eliminación de estos.	setMetadataSchema() submitMetadataRecord() deleteMetadataRecord() associate() dissociate()

**Tabla 1.3 Descripción de los métodos del estándar SPI**

Teniendo en cuenta las características antes expuestas sobre cada estándar, en la solución será utilizado SQI para realizar búsquedas de OA en ROA, SPI para la publicación en ROA de los recursos educativos almacenados en la plataforma, así como IMS-Basic LTI con el objetivo de integrar herramientas e-learning de forma transparente a la solución.



### 3.3. ¿Qué son los servicios web?

En la actualidad existen varias definiciones de lo que son los servicios web, por lo que se dificulta dar una definición exacta que abarque todo lo que ellos implican.

La comunidad World Wide Web Consortium (en lo adelante W3C) lo define como un “conjunto de aplicaciones o tecnologías con capacidad para interoperar en la web que intercambian datos entre sí con el objetivo de ofrecer servicios, basados en estándares de comunicación para el transporte, la codificación y el intercambio de datos entre sistemas heterogéneos”. [22]

Otros autores, afirman que es cualquier servicio que está disponible a través de internet, utilizando un estándar XML (en inglés Extensible Markup Language) como sistema de mensajería, además de no estar ligado a ningún sistema operativo o lenguaje de programación. [23]

Por lo que se puede llegar a la consideración de que un servicio web es un conjunto de tecnologías o aplicaciones que permiten interoperar de manera estandarizada en la red, independientemente del lenguaje de programación o sistema operativo.

A continuación, se expone una breve descripción de los protocolos y tecnologías que forman parte de estos estándares ampliamente aceptados y utilizados por los servicios web.

#### 3.3.1. Protocolos de comunicación entre aplicaciones

##### SOAP

SOAP (acrónimo de Simple Object Access Protocol), es un protocolo ligero diseñado para el intercambio de información estructurada en un entorno descentralizado y distribuido. Utiliza XML para la codificación de los mensajes que pueden ser intercambiados a través de varios protocolos de transporte como HTTP, SMTP<sup>8</sup>, entre otros. Es independiente del lenguaje de programación y consta de tres partes: una envoltura que define un marco para describir lo que está en un mensaje y cómo procesarlo, un conjunto de reglas de codificación para expresar instancias a solicitudes de tipos de datos definidos y una convención para representar llamadas a procedimientos remotos y respuestas. [24]

---

<sup>8</sup> **SMTP (Simple Mail Transfer Protocol):** Protocolo de red basado en texto utilizado para el intercambio de mensajes de correo electrónico.



De manera general, es un protocolo que proporciona un mecanismo estándar para estructurar mensajes utilizando XML, con el fin del intercambiar información con entornos heterogéneos utilizando el protocolo de transporte HTTP y otros, independiente de cualquier modelo de programación.

### **XML-RPC**

XML-RPC (acrónimo de XML-Remote Procedure Call), es un protocolo de llamada a procedimientos remotos que usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes. Un mensaje XML-RPC utiliza peticiones POST de HTTP para el envío de la información, indicando el procedimiento que se va a ejecutar en el servidor y los parámetros de la solicitud en XML. [25]

Importante agregar que es un protocolo ligero y fácil de usar, independiente del ambiente donde se ejecute y del sistema operativo utilizado, constituyendo una especificación simple que permite a estructuras de datos complejas ser transmitidas y procesadas.

### **REST**

Varios autores coinciden que REST (acrónimo de Representational State Transfer) es un estilo de arquitectura para aplicaciones que solicitan y manipulan recursos en la web utilizando los métodos estándar de HTTP: GET, POST, PUT y DELETE. [26]

Además, su creador Roy Thomas Fielding define REST como un conjunto de principios de arquitectura que se usan para describir cualquier interfaz web simple que utiliza XML y HTTP, sin las abstracciones adicionales de los protocolos basados en patrones de intercambio de mensajes como el protocolo de servicios web SOAP. [27]

REST es muy usado en la actualidad, ejemplo de esto son las diferentes API de servicios de internet que proveen sistemas populares como Facebook<sup>9</sup>, Twitter<sup>10</sup>, etc.

### **JSON-RPC**

JSON-RPC (acrónimo de JSON-Remote Procedure Call), es un protocolo de llamadas a procedimientos remotos codificadas en JSON (acrónimo de JavaScript Object Notation) similar a XML-RPC. Permite el envío de notificaciones y llamadas múltiples al servidor que pueden ser respondidas en orden o no, donde

---

<sup>9</sup> **Facebook:** Sitio web que ofrece acceso gratuito a la red social más grande del mundo.

<sup>10</sup> **Twitter:** Sitio social con servicios de microblogging y mensajería.



cada solicitud invoca un servicio remoto a través de HTTP. [28]

Todas las peticiones y respuestas son objetos simples, serializados usando JSON. Una petición es una llamada a un método específico brindado por el sistema remoto y debe contener tres propiedades: el nombre del método que se desea invocar, los parámetros del método y un identificador de la petición. El proveedor envía una respuesta con los resultados obtenidos de la invocación del método y el identificador de la petición respondida.

### **AMF 3**

AMF 3 (acrónimo de Action Message Format 3), es un formato binario compacto que se utiliza para serializar gráficos de objetos ActionScript<sup>11</sup>. Una vez serializado un gráfico de objeto codificado en AMF se puede utilizar para conservar y recuperar el estado de la aplicación a través de sesiones o permitir la comunicación de dos sistemas a través del intercambio de datos. Además soporta el envío de objetos complejos por referencia, ayudando a evitar el envío de instancias redundantes. [29]

Permite la comunicación entre Adobe Flash Player y una aplicación servidora remota que procesa las solicitudes enviadas, mapeando los requerimientos del objeto y sus métodos. Las especificaciones de AMF 3 son de libre disponibilidad y sirven como referencia para establecer qué tipos de mensajes pueden ser enviados entre Flash Player y el servidor. [30]

Cada uno de los protocolos estudiados forma parte de la solución, aprovechando las potencialidades individuales de los mismos para enriquecer los resultados de la investigación.

### **3.3.2. Tecnologías de servicios web**

#### **Lenguaje de Mercado Extensible**

El Lenguaje de Mercado Extensible (en inglés Extensible Markup Language, XML) fue desarrollado por el W3C para estructurar, almacenar e intercambiar información. Describe los datos de tal manera que es posible estructurarlos utilizando etiquetas, similar a HTML [31], facilitando la validación e interpretación de los mismos entre diferentes aplicaciones.

Es importante resaltar que todo documento XML tiene una entidad raíz, además posee una estructura

---

<sup>11</sup> **ActionScript**: Lenguaje de programación orientado a objetos, utilizado para el desarrollo de aplicaciones web animadas.



lógica y una física. La física está compuesta de unidades llamadas entidades que representan una parte del documento. La lógica del documento XML está formada por elementos que representan la estructura de la información que se está manejando. Esta estructura se organiza a través de normas establecidas que validan cuando un documento XML está bien formado y cuando no.

### Lenguaje de Descripción de Servicios Web

El Lenguaje de Descripción de Servicios Web (en inglés Web Services Description Language, WSDL) permite describir los servicios web utilizando XML. Se usa para localizar y ubicar los servicios que se proveen en la red, mediante la descripción de algunas de sus características, su localización, los métodos y parámetros que soporta.

Para la definición de un servicio web, WSDL se basa en los siguientes elementos XML [25]:

Elemento	Significado
<b>&lt;portType&gt;</b>	Operaciones proporcionadas por el servicio web.
<b>&lt;message&gt;</b>	Mensajes usados por el servicio web.
<b>&lt;types&gt;</b>	Los tipos de datos utilizados por el servicio web.
<b>&lt;binding&gt;</b>	Los protocolos de comunicaciones usados por el servicio web.

**Tabla 1.4 Elementos XML**

En términos generales, un documento WSDL describe la forma mediante la cual clientes externos pueden interactuar con los servicios web, contiene información acerca de la interfaz, la semántica y los aspectos de gestión involucrados en una solicitud realizada a un servicio web.

### 3.3.3. Seguridad en los servicios web

#### Capa de Conexión Segura

La Capa de Conexión Segura (en inglés Secure Socket Layer, SSL), es un sistema de protocolos de carácter general para conseguir un canal o medio seguro de comunicación a través de la red. Está basado en la aplicación conjunta de criptografía simétrica, criptografía asimétrica, certificados digitales y firmas digitales, con el objetivo de proteger las transacciones en línea y ayudar a mejorar la seguridad de los



sitios web. Además, permite la confidencialidad y la autenticación durante el intercambio de datos sensibles, como por ejemplo contraseñas de acceso a sistemas privados. [32]

De este modo, cuando un usuario accede a un servicio el servidor envía una clave pública y la certificación, el navegador cliente recibe estos datos y es el encargado de codificar toda la información que luego es transmitida de forma cifrada, haciendo que solo el servidor sea capaz de interpretarla. Así, el usuario envía los datos sin intervenir en los intercambios producidos.

### **Seguridad de Servicios Web**

La especificación Seguridad de Servicios Web (en inglés Web Services Security, WS-Security o WSS) ofrece los medios necesarios para aplicarle seguridad a los servicios web. Describe las mejoras relacionadas a la mensajería SOAP para brindar protección a través de la integridad, confidencialidad y seguridad en los mensajes. Estos mecanismos pueden ser utilizados para dar cabida a una amplia variedad de modelos de seguridad y tecnologías de encriptación. Está diseñado para ser extensible y describe cómo codificar señales binarias de seguridad. [33]

La especificación SOAP, por sí sola no provee los mecanismos necesarios para dar seguridad a los mensajes. Aquí, es donde entra a jugar su papel WS-Security, definiendo cómo utilizar los tokens de seguridad, firmas XML y encriptación XML en los mensajes SOAP para proporcionar autenticación, confidencialidad e integridad a los servicios web.

### **Firma XML**

La Firma XML (en inglés XML Signature), implementa mecanismos y especificaciones para crear firmas digitales basadas en XML. Facilita la autenticación de mensajes y servicios de autenticación de firmas para datos de cualquier tipo, tanto si se encuentra en el XML que incluye la firma o en cualquier otra parte. Puede aplicarse a cualquier contenido digital, incluyendo XML, y se basa principalmente en asociar claves con los datos de consulta para garantizar a través de una firma digital la autenticidad de los mismos. [34]

Es una especificación desarrollada bajo el amparo de la W3C que permite la firma completa o parcial de documentos utilizando una notación XML estándar. La especificación define un conjunto de algoritmos, sus URI<sup>12</sup> y los requisitos para su implementación.

---

<sup>12</sup> **URI:** Uniform Resource Identifier. Cadena corta de caracteres que identifica inequívocamente un recurso accesible en una red o sistema.



### **OAuth**

OAuth, es un protocolo abierto que ofrece autorización segura de interfaces estándares para aplicaciones de escritorio, móviles y web. Propone un método para que los clientes accedan a los recursos de un servidor en representación del propietario de un recurso. Además provee un proceso a los usuarios finales para autorizar el acceso de terceros a los recursos de sus servidores sin tener que compartir sus credenciales. [35]

Constituye una manera simple, segura y rápida de publicar y acceder a datos protegidos. Se trata de un modelo de autorización de código abierto, basado principalmente en normas existentes que aseguran que las credenciales de seguridad puedan ser suministradas y verificadas por diferentes plataformas de software. Es usado en la actualidad por varias API de servicios populares como Google, Twitter y Facebook.

Para la seguridad de los servicios que proveen los componentes se utiliza el protocolo OAuth y WS-Security.

### **3.4. Soluciones similares**

El desarrollo de las TIC, ha impulsado el surgimiento de diferentes iniciativas encaminadas a lograr la interoperabilidad entre las herramientas de apoyo a los procesos de enseñanza-aprendizaje. En este ámbito, existen disímiles plataformas educativas que implementan las principales tecnologías, estándares de interoperabilidad entre sistemas y protocolos de comunicación utilizados a nivel mundial. Algunas de las soluciones relevantes son analizadas a continuación.

#### **3.4.1. Sakai v2.8**

Sakai, es un sistema de código abierto desarrollado en Java para la gestión del aprendizaje que permite crear herramientas de colaboración, investigación y servicios. Tiene en cuenta el diseño de una arquitectura orientada a servicios, además de garantizar interoperar en tiempo real mediante la inclusión de las OSID del estándar OKI y el desarrollo de numerosas API propias orientadas a la interoperabilidad de aplicaciones y datos.

Facilita la interoperabilidad con repositorios a través de un consumidor del estándar SQL y la `osid_repository` de OKI, para ello, implementa un conjunto de métodos de interfaz que permiten a las



aplicaciones buscar, recuperar y cargar contenido, en dependencia de los permisos asociados. Además, cuenta con la herramienta Basic LTI Portlet que implementa la especificación IMS- Basic LTI para el intercambio de información con una aplicación alojada externamente. Entre las principales integraciones existentes se encuentran los ROA DSpace, Harvest Road Hive y Fedora, y los SGA People Soft y Ocu.

### 3.4.2. Moodle v2.0

Moodle es un paquete de software libre para la creación de cursos y sitios web de aprendizaje en línea. Recientemente, en su versión 2.0 brinda soporte para servicios web, mediante la integración de una API que facilita la interoperabilidad con aplicaciones e-learning. Hasta el momento permite interoperar a través de los protocolos REST, SOAP y XML-RPC. Además, la API provee servicios de acuerdo al sistema de privilegios que implementa la plataforma para garantizar mecanismos eficientes de seguridad.

Tiene una interfaz de servicios web para el intercambio de datos e información con otros sistemas, como por ejemplo: gestionar los datos de usuarios, gestionar las inscripciones de cursos, gestionar cursos y extraer información de calificaciones. También, posee el plugin<sup>13</sup> BasicLTI4Moodle que es un consumidor de IMS-Basic LTI para el consumo de herramientas externas dentro de Moodle.

### 3.4.3. Blackboard Learn v9.1

Es una plataforma propietaria perteneciente a la línea de productos de la compañía Blackboard Inc que provee funcionalidades para administrar cursos en línea. Este sistema integra aplicaciones como: procesadores de texto, hojas de cálculo, mensajería y herramientas de comunicación que ofertan cursos en línea de forma total o parcial. Cuenta con un conjunto de API y servicios web para la interacción con productos de terceros, además de proporcionar apoyo al estándar IMS-Basic LTI para simplificar la integración con proveedores de herramientas. También implementa la `osid_repository` de OKI y un consumidor de SQL para reutilizar recursos e información de ROA.

La observación de los procesos realizados en estas plataformas con el objetivo de interoperar con aplicaciones e-learning, permitió constatar que los componentes a desarrollar durante investigación estarán basados en los estándares de interoperabilidad antes estudiados, además de integrar a estos, funcionalidades seguras basadas en servicios web, contando con las funcionalidades que brindan estas plataformas como apoyo para el desarrollo de los mismos.

---

<sup>13</sup> **Plugin:** Es una extensión que encapsula código fuente reutilizable.





### 4. Entornos de desarrollo

Los entornos de desarrollo son herramientas para desarrollar software sobre ambientes amigables. En este apartado, se hace énfasis en las principales herramientas y tecnologías utilizadas en el desarrollo de la investigación, priorizándose las definidas como parte de la Arquitectura de Software de la plataforma educativa Zera.

#### 4.1. Metodología de desarrollo de software

Una metodología es un conjunto de procedimientos, técnicas, herramientas y documentos que facilitan el desarrollo de un software. En la presente investigación se describen las características principales que permitieron utilizar al Proceso Unificado de Rational como metodología para guiar la investigación.

#### Proceso Unificado de Rational

El Proceso Unificado de Rational (en inglés Rational Unified Process, RUP), es una metodología de desarrollo de software para satisfacer los requerimientos de los usuarios finales en tiempo y con el presupuesto establecido. Fue desarrollado por Rational Software y puede ser adaptado y extendido a las necesidades de la organización que lo adopte. [36]

Adaptándolo a las características de la investigación, servirá para regir el desarrollo de la aplicación a través de los flujos de trabajo modelamiento de negocio, requisitos, análisis y diseño, implementación y prueba. De este modo, el punto inicial del desarrollo, estará determinado por la descripción de los conceptos asociados al dominio del problema que constituyen la entrada fundamental para la captura de requisitos. Una vez agrupados los requerimientos en casos de uso, se podrá realizar en primer lugar el modelo de análisis, el de diseño y por último el de implementación. Posibilitando una vez obtenidos todos estos artefactos, preparar las pruebas necesarias para verificar que el sistema proporciona las características descritas en los casos de uso.

Además, el uso de esta metodología permitirá realizar iteraciones completas a cada caso de uso, garantizando que se prioricen las funcionalidades más complejas primero. Para ello, el equipo de desarrollo es responsable de especializarse en cada uno de los roles necesarios en cada flujo de trabajo, entre los que se destacan: especificador de requisitos, analista de sistemas, diseñador de interfaz de usuario, diseñador de base de datos, implementador, entre otros.



### 4.2. Lenguaje de modelado

Un lenguaje de modelado constituye una guía estándar que provee un conjunto de diagramas, símbolos y mecanismos para diseñar sistemas informáticos. En la investigación se resaltan las potencialidades que brinda el Lenguaje Unificado de Modelado, propuesto por la metodología RUP.

#### Lenguaje Unificado de Modelado v2.0

El Lenguaje Unificado de Modelado (en inglés Unified Modeling Language, UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software [37]. Establece un conjunto de elementos gráficos que se combinan para modelar sistemas orientados a objetos, mediante diagramas estándar, dentro de los que se encuentran los de clases, de objetos, de casos de uso, de componentes, de despliegue, de estados, de actividades y de interacción.

#### Características [38]:

- UML es un lenguaje de modelado de propósito general que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática.
- UML no pretende ser un método de desarrollo completo. No incluye un proceso de desarrollo paso a paso. UML incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso.
- Es orientado a objetos, permitiendo al programador que organice su programa de acuerdo con abstracciones de alto nivel, siendo estas cercanas a la forma de pensar de las personas.

### 4.3. Herramienta CASE

Las herramientas CASE (en inglés Computer Aided Software Engineering) son aplicaciones informáticas que apoyan el proceso de desarrollo de software a través del modelado de sistemas. Para modelar los artefactos de la investigación se utilizan las potencialidades que brinda Visual Paradigm.

#### Visual Paradigm v8.0

Visual Paradigm, es una herramienta CASE multiplataforma que brinda soporte para todos los diagramas que propone el lenguaje de modelado UML. Es fácil de instalar y actualizar, además de ser compatible entre ediciones. Permite realizar ingeniería inversa, así como generar base de datos y diagramas de flujo de datos. Admite exportar e importar ficheros de proyectos realizados, así como imágenes en los formatos



JPG, PNG y SVG. También incorpora funcionalidades para el trabajo en equipo sobre un mismo diagrama en tiempo real.

### **4.4. Lenguajes de desarrollo**

Los lenguajes de desarrollo posibilitan crear potentes herramientas informáticas. En el caso del desarrollo de aplicaciones web, se dividen en dos grupos: lenguajes del lado del cliente y lenguajes del lado del servidor.

#### **4.4.1. Lenguajes del lado del cliente**

Los lenguajes del lado del cliente son aquellos que basan su procesamiento en un cliente web, es decir, se procesan directamente por el navegador.

##### **Lenguaje de Marcado de Hipertexto Extensible v1.0**

El Lenguaje Extensible de Marcado de Hipertexto (en inglés eXtensible Hypertext Markup Language, XHTML) es muy similar al lenguaje HTML. De hecho, XHTML no es más que una adaptación de HTML al lenguaje XML, que se utiliza para crear la mayoría las páginas web. El propio W3C lo define como “un lenguaje reconocido universalmente y que permite publicar información de forma global”. [39]

Debido a que XHTML es un documento XML, puede utilizar otras herramientas de XML como XSLT (en inglés Extensible Stylesheet Language Transformations) que es un lenguaje para transformar contenido XML.

##### **Hojas de Estilo en Cascada v2.1**

Las Hojas de Estilo en Cascada (en inglés Cascading Style Sheets, CCS), son un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla. Esta forma de descripción proporciona a los desarrolladores el control total sobre los estilos y el formato de sus documentos. CSS es utilizado para dar estilo a documentos HTML y XML, separando el contenido de la presentación, permitiendo a los desarrolladores web controlar el formato de múltiples páginas web al mismo tiempo. [40]

En general, es un lenguaje para describir la presentación de documentos web, incluidos los colores, el diseño, y las fuentes. Es independiente de HTML y se puede utilizar con cualquier lenguaje de marcado basado en XML.



### JavaScript v1.8.1

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, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. [41]

Permite la verificación y procesamiento de los datos introducidos por el usuario antes de ser enviados al servidor y es soportado por los navegadores disponibles actualmente.

### Ajax

En realidad, el término AJAX es un acrónimo de Asynchronous JavaScript + XML, que se puede traducir como "JavaScript asíncrono + XML". Ajax 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 forman AJAX son: XHTML y CSS para crear una presentación basada en estándares, DOM<sup>14</sup> para la interacción y manipulación dinámica de la presentación, XML, XSLT y JSON para el intercambio y la manipulación de la información, XHR (acrónimo de XMLHttpRequest) para el intercambio asíncrono de información y JavaScript para unir todas las demás tecnologías. [42]

De este modo se aumenta la interactividad, velocidad y usabilidad en las aplicaciones, gracias a la posibilidad de realizar cambios sobre las páginas sin necesidad de recargarlas.

#### 4.4.2. Lenguaje del lado del servidor

Los lenguajes del lado servidor son reconocidos, ejecutados e interpretados por el propio servidor y luego son enviados al cliente en un formato comprensible para él.

### PHP v5.3.6

PHP, es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que

---

<sup>14</sup> **DOM:** Document Object Model. Estructura de objetos que genera el navegador cuando se carga un documento, el cual puede ser alterado mediante JavaScript para cambiar dinámicamente los contenidos y el aspecto de una página.



puede ser incrustado en HTML. Tiene soporte para cualquiera de los principales sistemas operativos del mercado y para una gran variedad de base de datos dentro de las que se incluye PostgreSQL. Además de soportar la mayoría de los servidores web de hoy en día, incluyendo Apache. [43]

Entre sus características principales se incluyen la autogeneración, almacenamiento y edición de archivos, procesamiento de información en formularios, manipulación de archivos en varios formatos como XML y YAML, entre otros. Posee librerías que facilitan el trabajo con servicios web, además los integrantes del equipo de desarrollo poseen conocimientos del mismo, lo que permite un ágil desarrollo de la solución.

### **4.5. Frameworks para el desarrollo**

Un framework o marco de trabajo es una estructura de archivos y utilidades que aceleran la programación de una aplicación informática, proporcionando una metodología de trabajo que sistematiza y facilita la generación de formularios, funciones y módulos de uso común, permitiendo al desarrollador dedicar su atención hacia los aspectos específicos de cada aplicación.

#### **4.5.1. Framework para el cliente**

##### **Jquery v1.5**

Jquery es un marco de trabajo rápido, poderoso y fácil de utilizar que permite a los desarrolladores y diseñadores web agregar elementos dinámicos e interactivos a sus sitios. Simplifica la manera de interactuar con los documentos HTML, así como la manipulación del árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX. Además cuenta con un gran número de plugins que permiten extender sus funcionalidades.

#### **4.5.2. Frameworks para el servidor**

##### **Zend Framework v1.11.11**

Zend Framework es un marco de trabajo de código abierto desarrollado para PHP que implementa el patrón Modelo-Vista-Controlador (MVC), es orientado a objetos y sus componentes tienen un bajo acoplamiento, permitiendo que sean usados en forma independiente. Entre estos componentes se encuentran los relacionados con los servicios web que ofrecen soporte para consumir y proveer servicios mediante los protocolos utilizados a nivel mundial. Además posee una amplia documentación.



### Symfony v1.4.15

Symfony es el marco de trabajo definido para el desarrollo de la plataforma educativa Zera, el cual posee una versión estable y hasta el momento satisface las necesidades de esta aplicación.

Está diseñado para optimizar el desarrollo de las aplicaciones web, separando la lógica de negocio, la lógica del servidor y la presentación. Propone varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja automatizando tareas comunes. Además, posee una extensa documentación, cuenta con una amplia variedad de plugins que facilitan el desarrollo y es compatible con el gestor de base de datos PostgreSQL.

Este marco de trabajo brinda una capa de abstracción a la base de datos a través de un ORM<sup>15</sup>. Para facilitar el desarrollo de la plataforma educativa Zera se utiliza Doctrine, brindando la capacidad de escribir opcionalmente consultas de base de datos en un lenguaje orientado a objetos.

#### Características particulares de Symfony:

- Suficientemente flexible como para adaptarse a cualquier proyecto.
- Extensible para utilizar librerías de terceros.
- Es independiente del gestor de base de datos.
- La utilización de librerías, ayuda a tener un código minimizado en la presentación.
- Sigue las mejores prácticas y patrones de diseño para la web.
- Maneja un completo sistema de prueba para depurar en modo desarrollo. [44]

#### 4.6. Entorno de Desarrollo Integrado

Un Entorno de Desarrollo Integrado (en lo adelante IDE), es un programa informático compuesto por un conjunto de herramientas que facilita el trabajo de los desarrolladores, puede estar orientado a uno o varios lenguajes de programación y brinda facilidades como: resaltado de sintaxis y completamiento de código.

---

<sup>15</sup> **ORM:** Object Relational Mapper. Permite acceder a la base de datos desde un contexto orientado a objetos, brindando varias ventajas entre las que se encuentra la reutilización, la portabilidad y la seguridad.



### **Net Beans v7.0.1**

NetBeans es un reconocido entorno de desarrollo integrado de código abierto y de distribución gratuita apoyado por una amplia comunidades de desarrolladores. Entre sus funciones incluye resaltado de sintaxis para los lenguajes de programación PHP, HTML, JavaScript, CSS, entre otros. Brinda depuración y un potente completamiento de código. Integra la gestión de configuración y las herramientas de control de versiones, lo que contribuye a mejorar el trabajo en equipo.

Ofrece soporte para los frameworks Symfony y JQuery agilizando el desarrollo de la solución. La integración con Symfony posibilita la ejecución de tareas tales como crear módulos, limpiar caché, entre otras. Además, este IDE utiliza asistentes para crear servicios REST y generar código para invocar servicios web (tanto en REST como en SOAP).

### **4.7. Sistema Gestor de Base de Datos**

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

### **PostgreSQL v9.1**

PostgreSQL, es SGBD multiplataforma y altamente escalable. Permite acomodar enorme cantidad de datos y usuarios concurrentes. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Presenta sofisticadas funciones de replicación, backup en línea y consultas de gran complejidad. [45]

Se adapta para trabajar a alta velocidad en escenarios con gran carga de trabajo, además de constituir el SGBD familiarizado con el equipo de desarrollo.

### **4.8. Servidor Web**

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

### **Apache v2.2.20**

Apache es un Servidor Web multiplataforma de código abierto totalmente gratuito. Cuenta con un elaborado índice de directorios y un directorio de alias. Permite generar informes de errores HTTP y



gestionar recursos para procesos hijos. Se caracteriza por ser configurable y robusto, además es fácil de instalar y goza de gran popularidad y aceptación.

### **5. Conclusiones del capítulo**

El estudio realizado sobre las principales tendencias de interoperabilidad entre PGA y otras herramientas e-learning, permitió identificar los principales estándares de interoperabilidad entre sistemas que implementan las plataformas existentes, y de este modo seleccionar los adecuados para integrar a la plataforma educativa Zera. Además la definición de las herramientas y tecnologías de desarrollo, sentó las bases para el análisis, diseño e implementación de la solución al problema de investigación.





## Capítulo 2: Características del Sistema

### 1. Introducción

En el desarrollo de un software, es de vital importancia definir los procesos que intervienen en este para lograr así un mejor entendimiento del sistema a desarrollar entre clientes y desarrolladores. De este modo, se podrán identificar los requisitos básicos en los que estará enfocado el equipo de trabajo, agrupados en casos de uso que guiarán el desarrollo.

### 2. Modelo de dominio

El modelo de dominio, facilita la comprensión de los principales conceptos asociados al negocio del dominio de un problema, representados en un modelo conceptual. Para la presente investigación, se decide realizar este artefacto, debido a que no es posible identificar con claridad los procesos, actores y trabajadores que realizan las actividades asociadas al negocio. Además, su correcta definición contribuirá a lograr una mejor comunicación entre usuarios, desarrolladores y clientes, mediante el establecimiento de un vocabulario común que permita entender el funcionamiento del sistema.

#### Diagrama del modelo de dominio

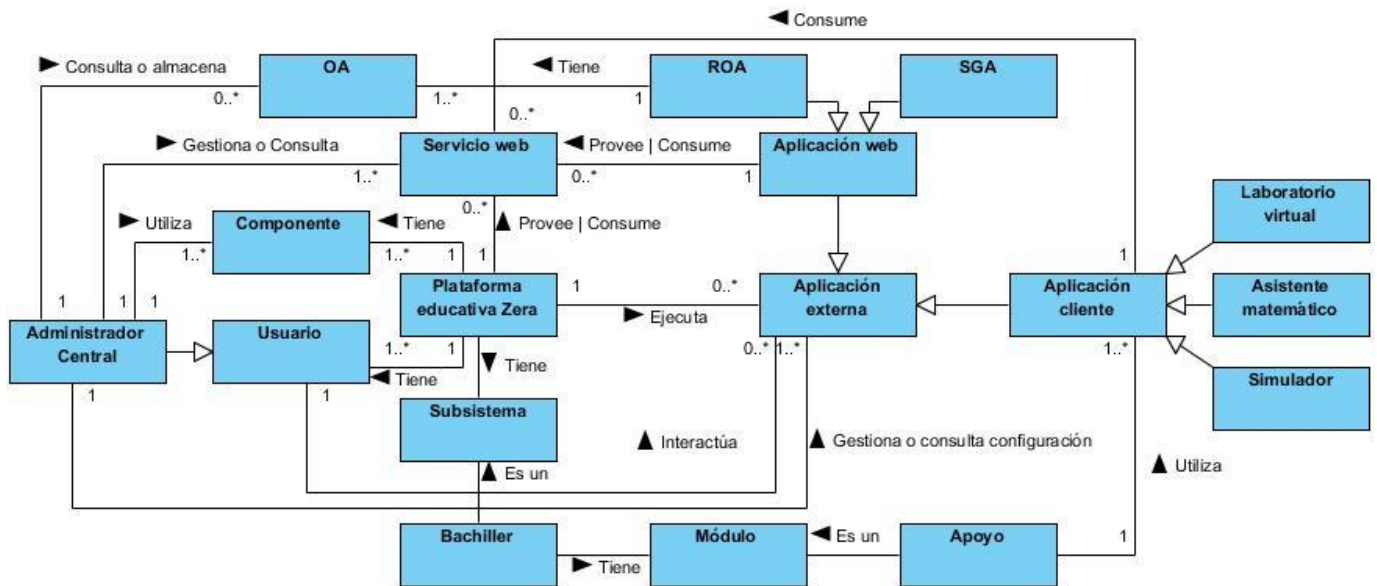


Figura 2.1 Modelo de dominio<sup>16</sup>

<sup>16</sup> Para comprender los conceptos asociados al modelo de dominio, ver el glosario de términos.



En el diagrama anterior, se muestra el modelo de dominio que relaciona los principales conceptos identificados para llevar a cabo el intercambio e información de recursos con aplicaciones de apoyo a los procesos de enseñanza aprendizaje en la plataforma educativa Zera.

### 3. Descripción del sistema propuesto

El sistema propuesto tiene como objetivo el desarrollo de varios componentes que permitan la interacción e intercambio de recursos e información entre la plataforma educativa Zera y otras aplicaciones externas.

Entre las principales funcionalidades que brindarán estos componentes se destacan:

- La gestión de los servicios web externos que puede consumir el sistema, así como los proveedores de los mismos, ya sean ROA o SGA.
- La gestión de los servicios web que provee el sistema mediante los protocolos SOAP, REST, AMF, XML-RPC y JSON-RPC. Permitiendo definir además, los permisos de cada servicio haciendo uso del sistema de privilegios de la plataforma
- El lanzamiento de aplicaciones externas dentro del sistema, mediante el estándar IMS-Basic LTI, utilizando el protocolo OAuth para firmar los datos que se envían.
- La búsqueda de OA almacenados en diferentes repositorios haciendo uso de un consumidor del estándar SQL. Además del almacenamiento de los recursos educativos (imágenes, sonido, videos, animaciones, simulaciones, contenidos) desarrollados en la plataforma hacia ROA a través del estándar SPI empaquetados con SCORM 2004.

### 4. Requisitos del software

La identificación de los requisitos de software permite definir las condiciones o capacidades que debe cumplir el sistema para satisfacer las necesidades del cliente. Estos se dividen en dos grupos: requisitos funcionales y requisitos no funcionales.

#### 4.1. Requisitos funcionales

Los requisitos funcionales (en lo adelante RF) posibilitan identificar las condiciones que el sistema debe cumplir. Para el correcto funcionamiento de la solución propuesta se espera que el sistema permita:

**RF 1: Gestionar configuración de aplicación externa.**



**RF 1.1 Incluir nueva configuración de aplicación externa:** Se brinda al usuario la posibilidad de introducir o seleccionar los datos necesarios para crear una nueva configuración que permita el acceso a una aplicación externa, mediante la opción incluir.

**RF 1.2 Ver datos de configuración de la aplicación externa:** Brinda la posibilidad de ver los datos relevantes de configuración de la aplicación externa a través del vínculo mostrar. Debe existir al menos una configuración previamente creada.

**RF 1.3 Modificar configuración de aplicación externa:** Se brinda al usuario la posibilidad de modificar la configuración para el acceso a una aplicación externa determinada a través del vínculo editar. Debe existir al menos una configuración previamente creada.

**RF 1.4 Eliminar configuración de aplicación externa:** Se le brinda al usuario la posibilidad de eliminar la configuración de una aplicación externa determinada, permitiéndose confirmar la eliminación. Debe existir al menos una configuración previamente creada.

**RF 2: Ejecutar configuración de aplicación externa:** Permite lanzar una aplicación externa, como si formara parte del sistema. La configuración correspondiente a la aplicación debe ser creada previamente.

**RF 3: Consultar configuración de aplicación externa.**

**RF 3.1 Mostrar un listado de configuraciones de aplicaciones externas:** Muestra al usuario un listado con las configuraciones de aplicaciones externas existentes. Debe haber al menos una configuración previamente creada.

**RF 3.2 Buscar una configuración de aplicación externa dado criterios de búsqueda:** Para ello debe existir al menos una configuración de aplicación externa creada, permitiéndose realizar la búsqueda mediante la opción buscar una vez seleccionados los criterios de búsqueda por los cuales se desea filtrar.

**RF 4: Gestionar servicio web.**

**RF 4.1 Incluir servicio web:** Permite crear un nuevo servicio web que podrá ser consumido por otros sistemas. Para realizar esta acción el usuario podrá seleccionar el servicio que desea agregar.



**RF 4.2 Ver datos del servicio web:** Brinda la posibilidad de ver los datos del servicio web que podrá ser consumido por otros sistemas mediante el vínculo mostrar. Debe existir al menos un servicio previamente creado.

**RF 4.3 Modificar servicio web:** Se brinda al usuario la posibilidad de modificar un servicio que será consumido por otro sistema determinado a través del vínculo editar. Debe existir al menos un servicio previamente creado.

**RF 4.4 Eliminar servicio web:** Se brinda al usuario la posibilidad de eliminar un servicio determinado, permitiéndose confirmar la eliminación. Debe existir al menos un servicio previamente creado.

### **RF 5: Consultar servicio web.**

**RF 5.1 Mostrar un listado de servicios web:** Muestra al usuario un listado con los servicios web existentes que pueden consumir otros sistemas. Debe haber al menos un servicio web previamente creado.

**RF 5.2 Buscar un servicio web dado criterios de búsqueda:** Para ello debe existir al menos un servicio web creado, permitiéndose realizar la búsqueda mediante la opción buscar una vez seleccionados los criterios de búsqueda por los cuales se desea filtrar.

### **RF 6: Gestionar proveedor de servicios web.**

**RF 6.1 Incluir proveedor de servicios web:** Permite añadir un nuevo proveedor de servicios web con el objetivo de consumir servicios.

**RF 6.2 Ver datos del proveedor de servicios web:** Brinda la posibilidad de ver los datos del proveedor de servicios web mediante el vínculo mostrar. Debe existir al menos un proveedor de servicio previamente creado.

**RF 6.3 Modificar proveedor de servicios web:** Se brinda al usuario la posibilidad de modificar un proveedor de servicio determinado a través del vínculo editar. Debe existir al menos un proveedor de servicio previamente creado.

**RF 6.4 Eliminar proveedor de servicios web:** Se brinda al usuario la posibilidad de eliminar un proveedor de servicio determinado, permitiéndose confirmar la eliminación. Debe existir al menos un proveedor de servicio previamente creado.



#### **RF 7: Consultar proveedor de servicios web.**

**RF 7.1 Mostrar un listado de proveedores de servicios web:** Muestra al usuario un listado con los proveedores de servicios web existentes. Debe haber al menos un proveedor de servicios web previamente creado.

**RF 7.2 Buscar un proveedor de servicios web dado criterios de búsqueda:** Para ello debe existir al menos un proveedor de servicio web creado, permitiéndose realizar la búsqueda mediante la opción buscar una vez seleccionados los criterios de búsqueda por los cuales se desea filtrar.

#### **RF 8: Gestionar servicio web externo.**

**RF 8.1 Incluir servicio web externo:** Permite añadir un nuevo servicio web externo.

**RF 8.2 Ver datos del servicio web externo:** Brinda la posibilidad de ver los datos del servicio web externo mediante el vínculo mostrar. Debe existir al menos un servicio previamente creado.

**RF 8.3 Modificar servicio web externo:** Se da al usuario la posibilidad de modificar un servicio web externo determinado a través del vínculo editar. Debe existir al menos un servicio previamente creado.

**RF 8.4 Eliminar servicio web externo:** Garantiza al usuario la posibilidad de eliminar un servicio web externo determinado, permitiéndose confirmar la eliminación. Debe existir al menos un servicio previamente creado.

#### **RF 9: Consultar servicio web externo.**

**RF 9.1 Mostrar un listado de servicios web externos:** Muestra al usuario un listado con los servicios web externos existentes. Debe haber al menos un servicio web externo previamente creado.

**RF 9.2 Buscar un servicio web externo dado criterios de búsqueda:** Para ello debe existir al menos un servicio web externo creado, permitiéndose realizar la búsqueda mediante la opción buscar una vez seleccionados los criterios de búsqueda por los cuales se desea filtrar.

#### **RF 10: Configurar protocolo.**

**RF 10.1 Activar protocolo:** Permite activar los protocolos de acceso a los servicios web disponibles en el sistema. Para ello debe seleccionar el protocolo previamente.



**RF 10.2 Desactivar protocolo:** Permite desactivar los protocolos de acceso a los servicios web disponibles en el sistema. Para ello debe seleccionar el protocolo previamente.

**RF 11: Consultar protocolo.**

**RF 11.1 Mostrar un listado de protocolos:** Muestra al usuario un listado con los protocolos existentes.

**RF 11.2 Buscar un protocolo dado criterios de búsqueda:** Permite realizar la búsqueda mediante la opción buscar una vez seleccionados los criterios de búsqueda por los cuales se desea filtrar.

**RF 12: Ver datos de un protocolo:** Garantiza la posibilidad de ver los datos del protocolo mediante el vínculo mostrar.

**RF 13: Consultar objeto de aprendizaje.**

**RF 13.1: Buscar objeto de aprendizaje dado criterios de búsqueda:** Permite buscar objetos de aprendizaje almacenados en un ROA mediante la opción buscar. Debe haberse seleccionado previamente un criterio de búsqueda.

**RF 13.2: Mostrar listado de objetos de aprendizaje:** Muestra un listado de objetos de aprendizaje como resultado de la búsqueda.

**RF 14: Importar objeto de aprendizaje:** Permite almacenar un objeto de aprendizaje en el sistema.

**RF 15: Almacenar recurso:** Permite almacenar recursos en un ROA.

**RF 16: Configurar permiso de un servicio web:** Se brinda al usuario la posibilidad de asociarle a los grupos<sup>17</sup> definidos en la plataforma los métodos de los servicios web a los que tendrá acceso. Debe existir al menos un servicio web previamente creado.

#### **4.2. Requisitos no funcionales**

Los requisitos no funcionales (en los adelante RNF) permiten definir las cualidades o propiedades que el software debe tener, con el objetivo de crear un producto final que sea fácil de utilizar, rápido y confiable, entre otras características.

---

<sup>17</sup> **Grupo:** Representa los grupos de seguridad definidos en la plataforma para el acceso a las funcionalidades.



### Software:

**RNF 1:** Tener cualquier navegador instalado, siendo como mínimo compatible con: Internet Explorer 7.0, Mozilla Firefox 3.6, Opera 10.06, Chrome 10.0 o Safari 5.0.

**RNF 2:** Contar con uno de los siguientes sistemas operativos:

- Windows XP o superior.
- GNU Linux.

### Hardware:

**RNF 3:** Los servidores centrales deberán cubrir las siguientes características o contar con una variante equivalente a:

- Servidores web con procesador: Intel Xeon de 4 núcleos. La cantidad de núcleos depende de la cantidad de usuarios conectados. (Ejemplo: 80,000 usuarios – 7 núcleos). RAM: 8 GB o más, con posibilidades de expansión en caso de ser necesario.
- Servidor de base de datos con procesador: Intel Xeon de 4 núcleos. La cantidad de núcleos depende de la cantidad de usuarios conectados. (Ejemplo: 80,000 usuarios – 7 núcleos). RAM: 8 GB o más, con posibilidades de expansión en caso de ser necesario.

Los usuarios finales deberán contar como mínimo con:

- Procesador Pentium II o superior.
- 1GB de RAM.
- 20GB de HDD.

### Diseño e implementación:

**RNF 4:** Lenguaje de programación: PHP v5.3.6.

**RNF 5:** Framework del lado del cliente: jQuery v1.5

**RNF 6:** El marco de trabajo base de desarrollo que se utilizará es: Symfony v1.4.15 y Zend Framework v1.11.11.

**RNF 7:** Como IDE se empleará: NetBeans v7.0.1.



**RNF 8:** Como servidor web se explotará Apache v2.2.20.

**RNF 9:** El SGBD deberá ser PostgreSQL v9.1.

### **Apariencia o interfaz externa:**

**RNF 10:** El diseño de interfaz debe ser sencillo y fácil de usar, con reconocimiento visual a través de elementos visibles que identifiquen cada acción a realizar. Será formal, serio y con una navegación sugerente.

**RNF 11:** El sistema proporcionará claridad y correcta organización de la información, permitiendo la interpretación correcta e inequívoca de ésta.

### **Seguridad:**

**RNF 12:** Transferencia de mensajes: firmados.

**RNF 13:** Comunicación web de transferencia de datos: encriptados.

**RNF 14:** Acceso mono-usuario al sistema, estableciendo sesiones de trabajo.

**RNF 15:** El sistema debe realizar un seguimiento de traza de las acciones realizadas.

**RNF 16:** Los usuarios del sistema deben tener acceso (según sus permisos) en todo momento a la información solicitada.

### **Usabilidad:**

**RNF 17:** El sistema podrá ser usado por cualquier persona que posea conocimientos básicos de informática.

**RNF 18:** El sistema debe tener acceso al menú general desde cualquiera de sus páginas.

**RNF 19:** Los elementos gráficos como los iconos deberán contar con un mensaje flotante que señalen el tipo de recurso al que se refiere.

**RNF 20:** Se debe mantener informado al usuario del resultado de las acciones realizadas.

**RNF 21:** Se debe brindar soporte para varios idiomas.

**RNF 22:** Cada campo debe tener un indicador de ayuda.





### Rendimiento:

**RNF 23:** Se garantizará que la respuesta a peticiones de los usuarios del sistema sea en un período de tiempo breve (de segundos), al igual que la velocidad de procesamiento de la información.

### Eficiencia:

**RNF 24:** Cuando un usuario, sin importar su rol, permanece inactivo durante 20 minutos, se cierra la sesión automáticamente.

## 5. Modelo de casos de uso del sistema

### 5.1. Patrones de caso de uso

Los patrones son soluciones simples compuestas por una pareja problema/solución que capturan las mejores prácticas para modelar casos de uso (CU), fundamentadas en la experiencia para problemas específicos y comunes que pueden emplearse en diferentes contextos. A continuación se enuncian los patrones presentes en el diagrama de casos de uso del sistema que brindan la posibilidad de estructurar y organizar de manera entendible el modelado del sistema:

- **CRUD completo:** Es un patrón que unifica varios CU simples en una unidad. Permite modelar en un CU diferentes operaciones para administrar cierta entidad, tales como crear, leer, modificar y eliminar. En la figura siguiente se muestra un ejemplo de su uso:

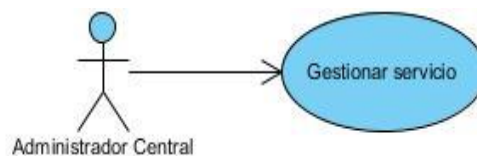


Figura 2.2 Patrón CRUD completo

- **Extensión concreta:** Es un patrón que involucra dos CU y una relación de extensión, donde el CU extendido puede ser instanciado por su cuenta o por el CU base. En la figura siguiente se muestra un ejemplo de su uso:

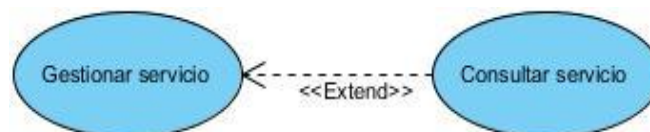


Figura 2.3 Patrón extensión concreta



### 5.2. Descripción de los actores del sistema

Actor	Descripción
<b>Administrador Central</b>	Actor que gestiona y consulta los servicios web que provee y consume la plataforma, los proveedores de servicios, los protocolos, además de las configuraciones de aplicaciones externas. También puede realizar búsquedas de OA en ROA y almacenar recursos en los mismos.

Tabla 2.1 Descripción de los actores del sistema

### 5.3. Diagrama de casos de uso del sistema

En la figura siguiente se muestra el diagrama de casos de uso definido a partir de los requerimientos del sistema, estructurado utilizando los patrones de CU CRUD completo y extensión concreta.

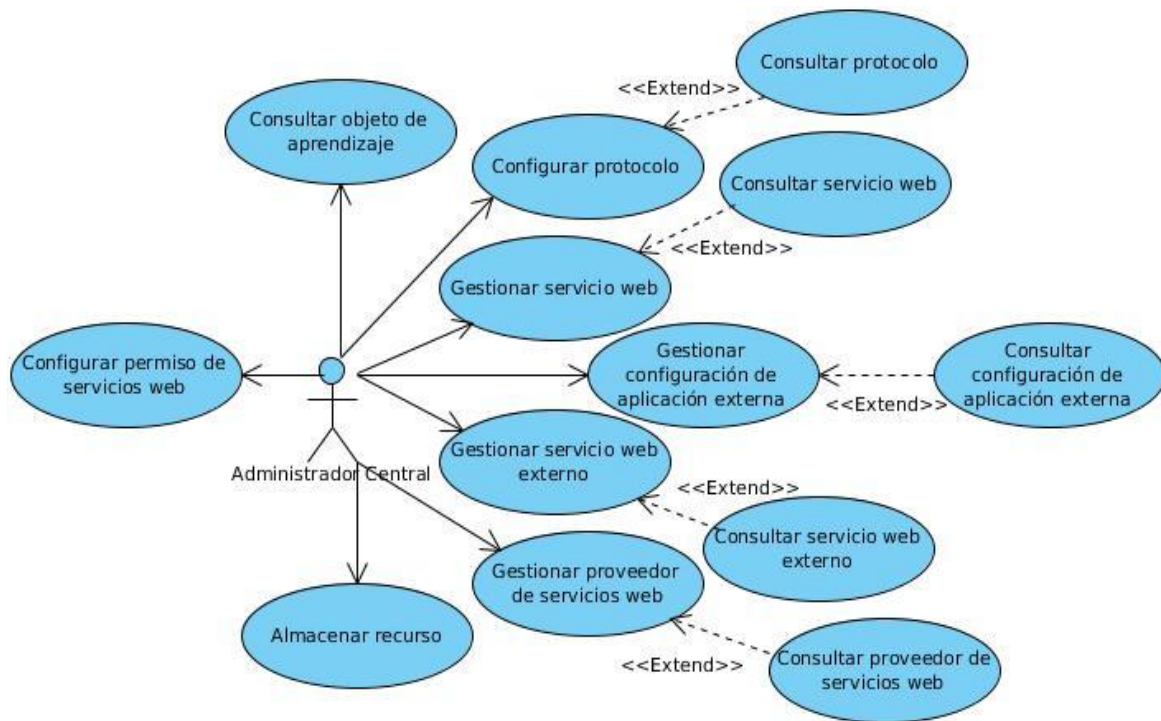


Figura 2.4 Diagrama de casos de uso del sistema



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

A continuación se muestra la descripción del CU Configurar permiso de servicios web. Para consultar las descripciones generales de los restantes CU, ver el Anexo # 1.

<b>Descripción general del CU Configurar permiso de servicios web</b>		
<b>Objetivo</b>	Configurar los permisos de los grupos establecidos en la plataforma a los servicios web.	
<b>Actores</b>	Administrador Central (Inicia): Configura los permisos de los grupos establecidos en la plataforma a los servicios web.	
<b>Resumen</b>	El caso de uso inicia cuando el Administrador Central selecciona la opción que le permite configurar los permisos de los grupos establecidos en el sistema a los servicios web. El sistema brinda la posibilidad de asociar al grupo seleccionado los métodos de un servicio web a los que tendrá acceso. El actor asocia al grupo seleccionado los métodos del servicio y el caso de uso termina.	
<b>Complejidad</b>	Baja	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	Debe haberse generado el escritorio de trabajo del usuario autenticado. Para asociar un servicio web debe haberse incluido previamente y el actor debe ser el responsable temporal de la acción.	
<b>Postcondiciones</b>	Se configuró el acceso de un grupo a los servicios web disponibles en la plataforma por el actor.	
<b>Flujo de eventos</b>		
<b>Flujo básico Configurar permiso de servicios web</b>		
	<b>Actor</b>	<b>Sistema</b>
1.	Selecciona la opción de configurar los permisos de los grupos establecidos en la plataforma a un servicio web.	
2.		Muestra un listado con los métodos de los servicios web disponibles en el sistema (en caso de no existir ninguno, muestra un



		mensaje de información). Brinda la posibilidad de seleccionar el siguiente criterio: ➤ Grupo.
3.	Selecciona el grupo.	
4.		Permite asociarle los métodos de los servicios web disponibles.
5.	Asocia los métodos de los servicios web a los que tendrá acceso el grupo.	
6.		<i>Valida los datos.</i>
7.		<i>Establece los permisos del servicio web.</i>
8.		Muestra un mensaje de confirmación.
9.		Termina el caso de uso.
<b>Flujos alternos</b>		
<b>Nº 3. a El actor selecciona otro grupo para asociarle los servicios web a los que tendrá acceso.</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Muestra un listado con los métodos de los servicios web disponibles.
2.		Regresa al paso 4 del flujo básico.
<b>Nº 6. a Existen datos incompletos.</b>		
	<b>Actor</b>	<b>Sistema</b>
1.		Muestra un mensaje de información.
2.		Muestra un indicador sobre el campo.
3.		Regresa al paso 3 del flujo básico.

Tabla 2.2 Descripción del CU\_ Configurar permiso de servicios web



## 6. Conclusiones parciales

La identificación de los principales conceptos asociados a la solución, reflejados en el modelo de dominio permitió definir los requisitos funcionales y no funcionales que sustentan la propuesta de solución representados a través de los CU del sistema. Así, se dejan creadas las bases para comenzar con la construcción de la propuesta, velando por el cumplimiento los requerimientos identificados.



## Capítulo 3: Análisis y diseño del Sistema

### 1. Introducción

La entrada fundamental para el análisis y diseño de la solución son los casos de uso del sistema definidos y especificados en el flujo de trabajo anterior. Utilizando los mismos, se describe cómo el sistema será realizado teniendo en cuenta las restricciones impuestas, indicando con precisión lo que se debe programar.

### 2. Modelo de análisis

El modelo de análisis proporciona una visión general del sistema descrito en el lenguaje de los desarrolladores, representando las clases del análisis y sus objetos organizados en paquetes que colaboran.

A continuación se describe brevemente la clasificación de las clases del análisis que pueden ser identificadas en el dominio de un problema, representadas a través de tres estereotipos básicos: interfaz, control y entidad.

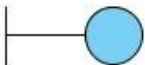

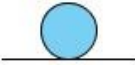
Nombre	Características	Estereotipo
Interfaz	Modelan la interacción entre el sistema y sus actores.	 CI_Nombre
Control	Modelan los aspectos dinámicos del sistema, coordinando las acciones y los flujos de control principales de los objetos que implementan la funcionalidad del CU.	 CC_Nombre
Entidad	Modelan información que posee larga vida y que es a menudo persistente.	 CE_Nombre

Tabla 3.1 Clasificación de las clases del análisis



### 2.1. Diagrama de clases del análisis

La representación del diagrama de clases del análisis (DCA) para el CU Configurar permiso de servicios web se muestra a continuación. Los restantes diagramas se encuentran en el Anexo # 2.

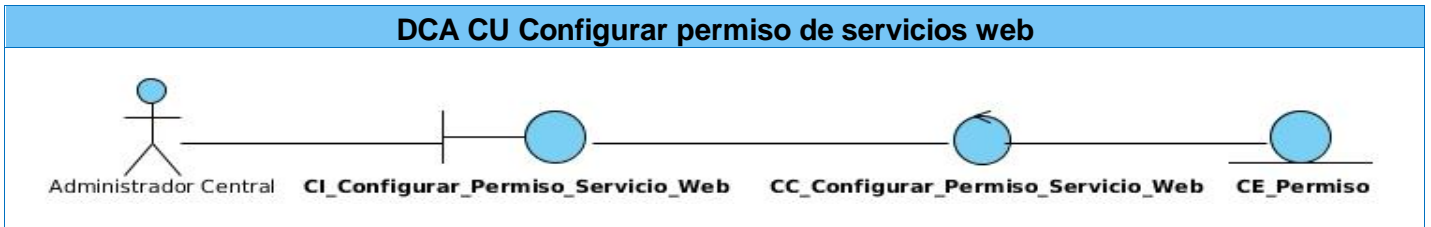


Tabla 3.2 DCA\_CU Configurar permiso de servicios web

### 2.2. Diagrama de colaboración

La representación de los diagramas de colaboración (DC) para el CU Configurar permiso de servicios web se muestra a continuación. Los restantes diagramas se encuentran en el Anexo # 3.

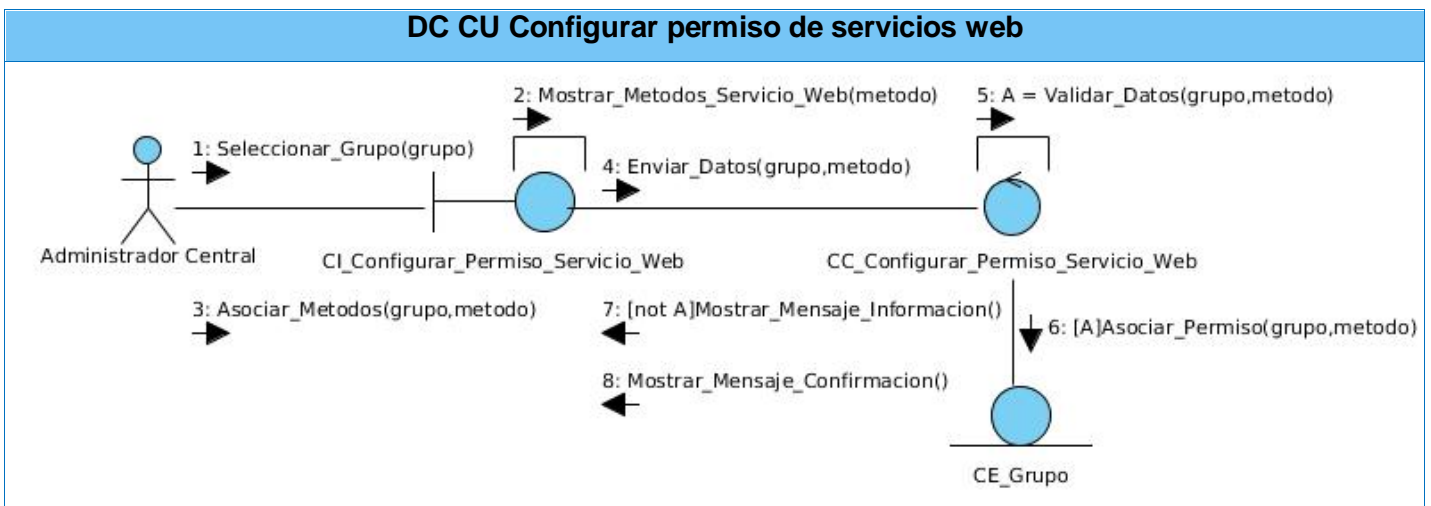


Tabla 3.3 DC\_CU Configurar permiso de servicios web

### 3. Patrón arquitectónico Modelo – Vista – Controlador

La plataforma educativa Zera se desarrolla haciendo uso del framework Symfony, utilizando la lógica que implementa el estilo arquitectónico de llamada y retorno conocido como Modelo Vista Controlador (MVC), representado en la figura 3.1.

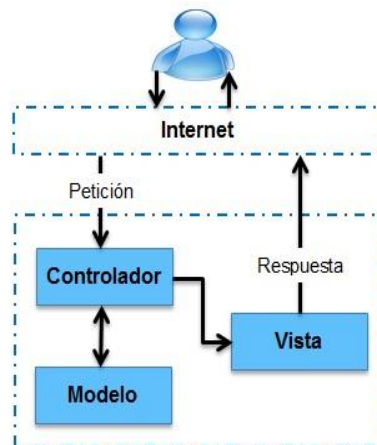


El desarrollo de la solución propuesta se basa en el uso de este patrón, agrupando las funcionalidades en los siguientes plugins Symfony:

- **sfAOBasicLTIPlugin:** Encapsula las funcionalidades para la configuración de aplicaciones externas a la plataforma, basado en el estándar IMS-Basic LTI.
- **sfAOServicesPlugin:** Agrupa las funcionalidades que posibilitan la configuración de los elementos necesarios para el acceso a los servicios web, así como la búsqueda de OA mediante el estándar SQL en repositorios y la publicación de recursos haciendo uso de la especificación SPI.

El patrón MVC separa las funcionalidades en tres capas, otorgando a cada capa las siguientes responsabilidades [40]:

- **Modelo:** Representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- **Vista:** Es con quien interactúa el usuario, siendo responsable de presentar la información que se maneja.
- **Controlador:** Se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.



**Figura 3.1 Patrón MVC**

Para estructurar mejor el diseño de estos plugins, se aplican un conjunto de patrones que serán analizados en el apartado siguiente.





#### 4. Patrones de diseño

Los patrones de diseño brindan solución a problemas comunes que pueden ser encontrados durante el diseño, perfeccionando los componentes de un sistema de software y sus relaciones. Dentro de los patrones utilizados para estructurar el diseño del sistema y que incluye por defecto el framework de desarrollo se destacan los siguientes:

- **Creador:** Permite crear objetos de una clase determinada. Es utilizado en la mayoría de las clases controladoras para crear instancias de formularios y entidades.
- **Controlador:** Se basa en asignar la responsabilidad de todos los eventos realizados a una clase específica que constituye el único punto de entrada para cada evento. En la solución propuesta se define un controlador frontal que recibe las peticiones realizadas (ubicado en el archivo administración.php), asociándolas a acciones predefinidas que permiten ejecutar la petición enviada.
- **Observador:** Este patrón permite definir una dependencia entre objetos de forma que cuando un objeto cambia de estado, todos sus objetos dependientes son notificados y actualizados. Es utilizado para actualizar los usuarios conectados y las sesiones establecidas al autenticarse determinado usuario en el sistema.
- **Decorador:** El objetivo de este patrón es añadir dinámicamente funcionalidad a un objeto. Es utilizado en la vista, donde cada plantilla mostrada por la ejecución de cualquier acción es decorada con un layout.

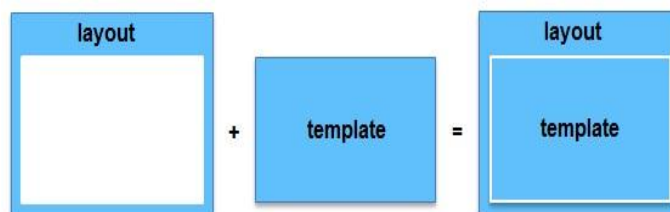


Figura 3.2 Patrón de diseño Decorador

- **Experto:** Este es uno de los más utilizados, mediante el cual se asignan responsabilidades a la clase que cuenta con la información necesaria. Se evidencia haciendo uso de la librería externa Doctrine, utilizada por Symfony para realizar la capa de abstracción en el modelo y encapsular la



lógica de los datos.

- **Singleton:** Define que una clase solo tiene una instancia, proporcionando un punto de acceso global a ella. Es aplicado en el uso de las variables globales de Symfony (sfConfig y sfContext) para la asignación de mensajes y atributos al usuario y la obtención de variables globales de configuración.

### 5. Patrones de servicio

En el desarrollo de la solución fueron aplicados algunos de los patrones de servicio utilizados en soluciones similares para estandarizar la comunicación entre distintos sistemas de software. Entre estos se encuentran los siguientes:

- **Service Interface:** La implementación de este patrón permite que la capa de servicios expuesta en un sistema se encuentre desacoplada de la capa de negocio. Este es aplicado en la interfaz de servicios web de la solución, donde toda la implementación necesaria para comunicarse con otros sistemas se encuentra separada de la lógica de negocio asociada a la aplicación.
- **Services Gateway:** Este patrón se aplica cuando al diseñar un sistema se desacopla la lógica de acceso a los servicios del resto de la aplicación. En la solución, se utiliza para encapsular las funcionalidades que permiten consumir los servicios web que exponen los proveedores en librerías separadas de la lógica de negocio de la aplicación.

### 6. Modelo de diseño

El modelo de diseño, es una abstracción de la implementación del sistema que describe la realización física de los casos de uso, centrado en cómo los requisitos funcionales y no funcionales impactan en el desarrollo de la aplicación. Representa todas las clases del diseño, subsistemas, paquetes, colaboraciones y las relaciones entre ellos, constituyendo la entrada principal a las actividades de implementación.

#### 6.1. Diagrama de clases del diseño

Los diagramas de clases del diseño de la solución propuesta se basan en la implementación del patrón MVC que realiza Symfony (Ver Anexo # 4), integrando algunos de los patrones de diseño mencionados anteriormente con los elementos principales que propone esta arquitectura.



A continuación se describen algunos de los elementos que conforman los diagramas de clases del diseño (DCD), agrupados en paquetes.

**Paquete Vista:** Contiene las clases que muestran la información al usuario como resultado de ejecutar una acción determinada, utilizando un layout que tiene el código HTML común en todas las páginas para decorar las plantillas de la acción que se esté ejecutando.

**Paquete Controlador:** Contiene las clases encargadas de relacionar la lógica del negocio con la presentación. Cuenta con un controlador frontal que constituye el único punto de entrada a la aplicación, siendo el responsable de cargar la configuración y determinar la acción a ejecutarse en dependencia de la petición realizada. Una vez ejecutada una acción, esta se encarga de verificar la integridad de la petición y preparar los datos requeridos para la capa de presentación.

**Paquete Modelo:** Contiene las entidades generadas por Doctrine ORM en correspondencia con las tablas de la base de datos que almacenan toda la información que maneja la aplicación.

**Paquete Lib:** Contiene librerías personalizadas que responden a las funcionalidades propias de cada componente.

**Paquete Doctrine ORM:** Contiene las funcionalidades que permiten el acceso y la modificación de los datos almacenados en la base de datos mediante objetos, utilizando una capa de tipo ORM.

**Paquete JavaScript:** Contiene las clases JavaScript necesarias para las plantillas.

**Paquete CSS:** Contiene las clases CCS necesarias en las plantillas.

**Paquete Componentes Symfony:** Contiene los componentes y configuraciones propias del framework.

La representación del DCD para el CU Configurar permiso de servicios web se muestra a continuación. Los restantes diagramas se encuentran en el Anexo # 5.

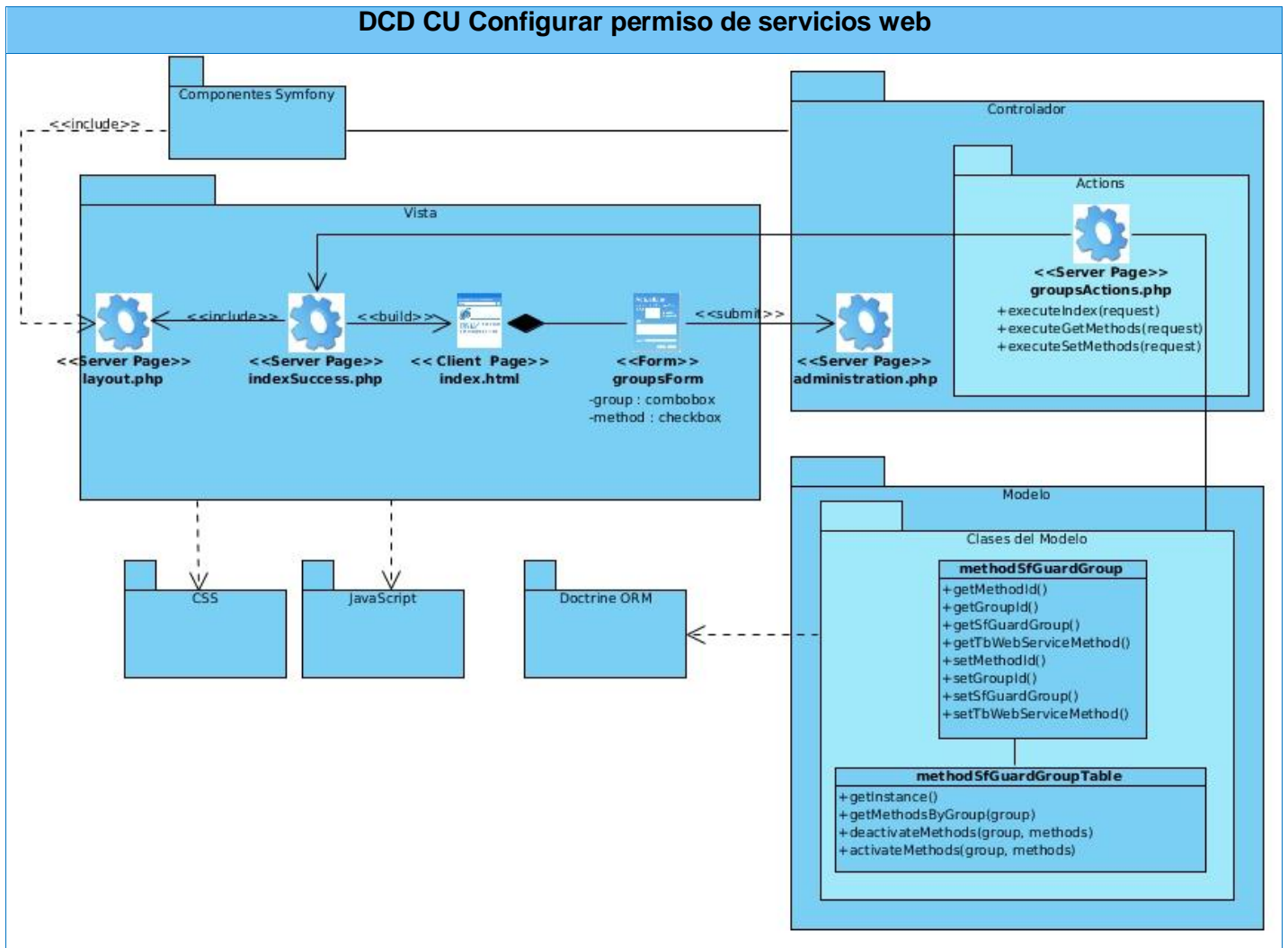


Tabla 3.4 DCD\_CU Configurar permiso de servicios web

## 7. Modelo de datos

El modelo de datos se utiliza para describir la estructura lógica y física de la información persistente que puede ser gestionada por el sistema. A continuación se muestra el diseño de las tablas necesarias para el desarrollo de las funcionalidades que formarán parte de la base de datos después de identificar las clases persistentes y realizar las transformaciones correspondientes.

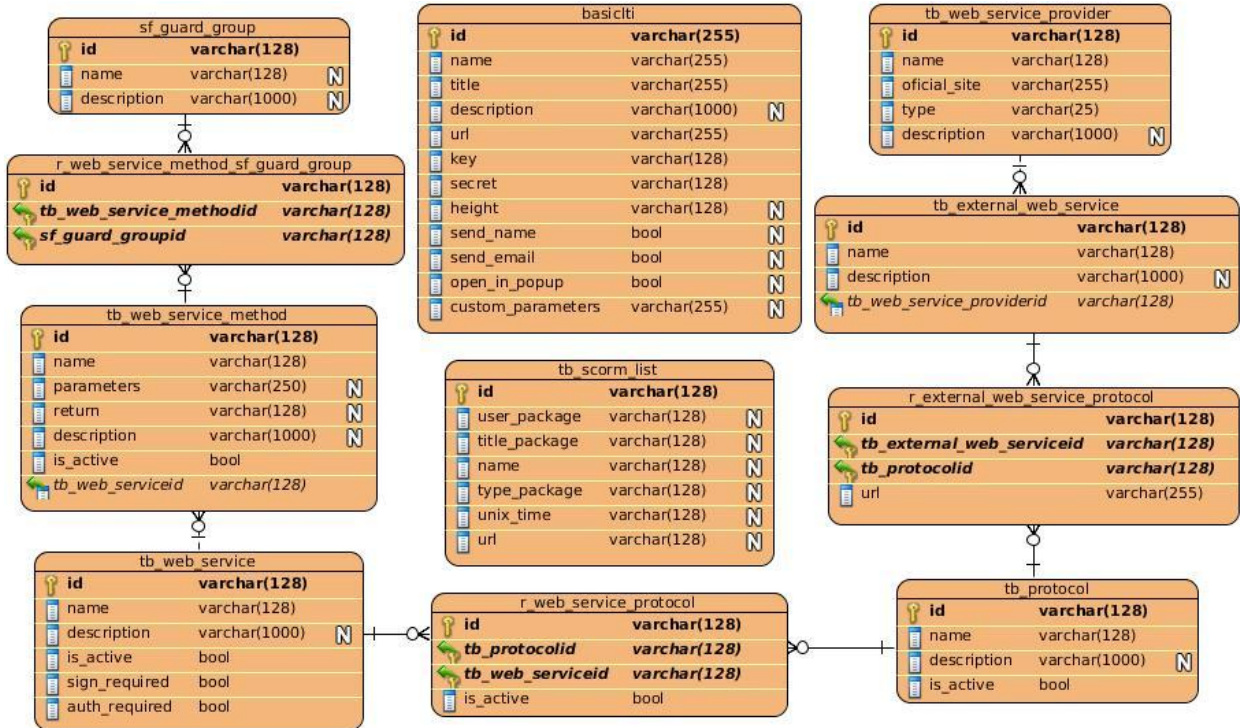


Figura 3.3 Diseño de la base de datos<sup>18</sup>

## 8. Diagrama de despliegue

El diagrama de despliegue define la arquitectura física del sistema. Es utilizado para representar las relaciones de los elementos desarrollados en tiempo de ejecución, visualizando la distribución de los componentes de software en nodos físicos. En la figura 3.4 se muestra el diagrama de despliegue definido como parte de la solución, donde el cliente se conecta al servidor central, siendo este último el encargado de comunicarse con las restantes aplicaciones mediante el protocolo de transporte HTTP. En el caso de los módulos de apoyo, una vez ejecutados, los mismos podrán consumir los servicios almacenados en el servidor central.

<sup>18</sup> Para ver la descripción asociada a cada tabla de la base de datos, ver Anexo # 6.

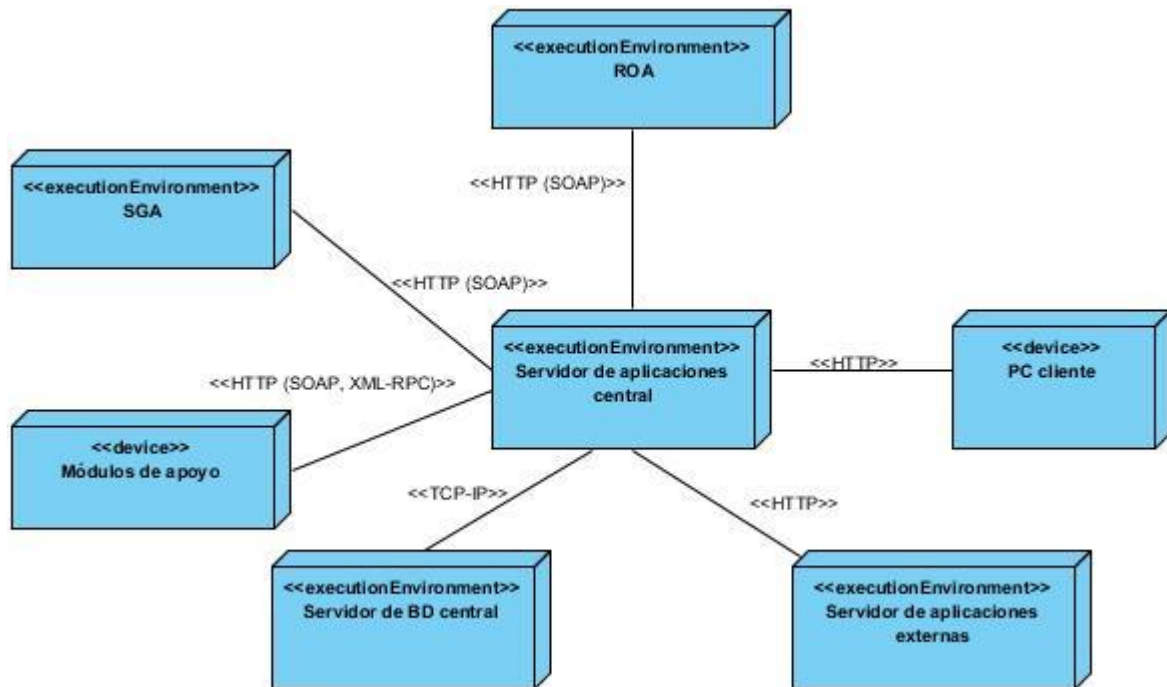


Figura 3.4 Diagrama de despliegue

## 9. Conclusiones Parciales

El desarrollo del análisis y diseño del sistema, permitió definir las bases necesarias para la implementación. Así, con la descripción realizada sobre los patrones de diseño se pudo estructurar los DCD como entrada principal para la implementación. Además el desarrollo del diagrama de despliegue, permitirá describir el sistema en términos de componentes en el próximo flujo de trabajo.



### Capítulo 4: Implementación y Prueba

#### 1. Introducción

La descripción de los elementos necesarios relacionados con la implementación y validación de un producto de software, cuentan como entrada principal con el resultado de los artefactos obtenidos durante el análisis y diseño de la solución. Para ello, se muestra la organización del sistema en términos de componentes, así como las dependencias con los nodos físicos donde funcionará la aplicación, permitiendo que los componentes desarrollados cumplan con los requisitos establecidos y puedan ser integrados a un sistema ejecutable.

#### 2. Modelo de implementación

El modelo de implementación describe los elementos del diseño en términos de componentes<sup>19</sup> y su organización de acuerdo a los nodos específicos del diagrama de despliegue, definiendo las principales unidades que se pueden integrar, versionar, desplegar y cambiar por separado.

##### 2.1. Diagrama de componentes

Los diagramas de componentes son usados para estructurar el modelo de implementación en término de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación. De este modo, se modela la vista estática de un sistema, estableciendo la organización y las dependencias entre un conjunto de componentes de software, sean estos código fuente, paquetes, librerías, binarios o ejecutables.

La representación del diagrama de componentes (DCO) para el CU Configurar permiso de servicios web se muestra a continuación. Los restantes diagramas se encuentran en el Anexo # 7.

---

<sup>19</sup> **Componente:** Parte física de un sistema que encapsula código fuente expuesto mediante interfaces.

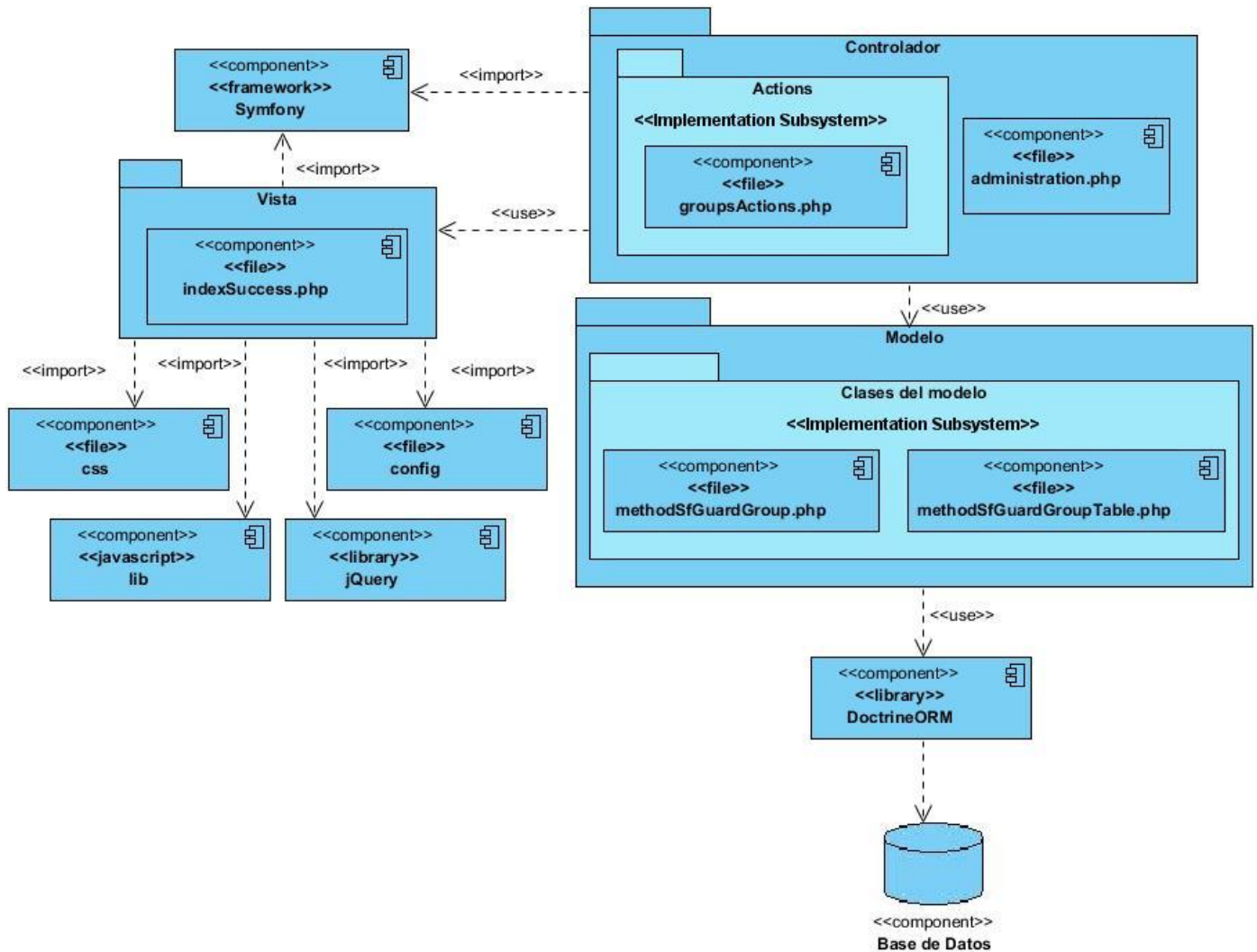


Figura 4.1 DCO CU Configurar permiso de servicios web

### 3. Pruebas de software

Las pruebas de software, se encargan de verificar la calidad de un sistema o componente que es ejecutado bajo condiciones o requisitos especificados previamente. Tienen el objetivo de encontrar y documentar errores, garantizando que el producto final funcione como fue diseñado e implemente de manera correcta los requerimientos identificados.





### 3.1. Estrategia de pruebas

La estrategia de prueba utilizada para evaluar los componentes desarrollados inició con la evaluación de los elementos simples y pequeños, hasta lograr probar todo el software en su conjunto. Como ayuda a la detección de los errores existentes, el sistema fue sometido a los siguientes niveles de pruebas [44]:

- Pruebas unitarias: Permiten verificar el correcto funcionamiento de cada módulo por separado, centrándose en la lógica del procesamiento interno y en las estructuras de datos dentro de los límites de un componente.
- Pruebas de integración: Se centran en comprobar que los módulos probados por separado funcionen en conjunto, con el objetivo de verificar que interactúan correctamente a través de sus interfaces y cubren las funcionalidades establecidas en los requisitos.
- Pruebas del sistema: Tienen como propósito probar que el sistema funciona correctamente después de haber integrado los componentes desarrollados al mismo.

### 3.2. Métodos de prueba

Para comprobar la calidad de los componentes desarrollados, se realizó una revisión de las especificaciones del diseño y la codificación en correspondencia con los requisitos establecidos, aplicando los métodos de prueba: Caja Blanca y Caja Negra.

#### 3.2.1. Pruebas de Caja Blanca

Las pruebas de Caja Blanca se basan en un minucioso examen de las funciones internas del software, garantizando que se ejerciten los caminos independientes, las decisiones lógicas, los bucles y las estructuras de datos internas.

Debido a la complejidad de la solución, este método de prueba fue ejecutado como parte de las pruebas unitarias realizadas por el desarrollador a medida que la solución fue desarrollada, revisando el funcionamiento interno de cada componente haciendo uso del framework de pruebas Lime<sup>20</sup> que propone Symfony.

En la figura siguiente se presenta un ejemplo de las pruebas realizadas a la clase **wsClassManager**:

---

<sup>20</sup> **Lime**: Framework que permite automatizar las pruebas unitarias realizadas al código.



```
1..10
# wsClassManager
ok 1 - new crea un objeto de la clase correctamente.
ok 2 - getDir() - Retorna un string con el directorio de clases de los servicios web.
ok 3 - getWebServicesInFile() - Devuelve un arreglo con las clases de los servicios web existentes.
ok 4 - getAvailableWebServices() - Devuelve un arreglo con los servicios web disponibles para su inclusión.
ok 5 - getAvailableWebServiceMethods() - Devuelve un Doctrine_Collection con los métodos del servicio web pasado como parámetro.
ok 6 - getMethodsByUserIdAndWebService() - Devuelve un Doctrine_Collection con los métodos disponibles para un usuario pasando como parámetro su id y el nombre del servicio web.
ok 7 - isAvailableWebService() - Si está disponible el servicio web pasado como parámetro devuelve el objeto TbWebService
ok 8 - isAvailableProtocol() - Si está disponible el protocolo pasado como parámetro devuelve el objeto TbProtocol.
not ok 9 - isAvailableProtocolForWebService() - Si está disponible el protocolo para un servicio web dado su nombre devuelve true.
# Failed test (./test/unit/sfAOServicesPlugin/managerTest.php at line 37)
# variable isn't a 'boolean' it's a 'TbProtocol'
not ok 10 - isAvailableMethodForWebService() - Si está disponible un método para un servicio web devuelve true.
# Failed test (./test/unit/sfAOServicesPlugin/managerTest.php at line 38)
# variable isn't a 'boolean' it's a 'TbWebServiceMethod'
# Looks like you failed 2 tests of 10.
```

Figura 4.2 Resultados de las pruebas realizadas a la clase wsClassManager

Como resultado de la ejecución de las pruebas a esta clase se identificaron como no conformidades (NC) que los métodos `isAvailableProtocolFormWebService()` y `isAvailableMethodForWebService()` no devuelven el tipo de dato correcto, lo que permitió refactorizar el código y eliminar posibles errores en la solución.

A continuación, se muestra el resultado de las pruebas realizadas a esta clase después de corregir las NC.

```
1..10
# wsClassManager
ok 1 - new crea un objeto de la clase correctamente.
ok 2 - getDir() - Retorna un string con el directorio de clases de los servicios web.
ok 3 - getWebServicesInFile() - Devuelve un arreglo con las clases de los servicios web existentes.
ok 4 - getAvailableWebServices() - Devuelve un arreglo con los servicios web disponibles para su inclusión.
ok 5 - getAvailableWebServiceMethods() - Devuelve un Doctrine_Collection con los métodos del servicio web pasado como parámetro.
ok 6 - getMethodsByUserIdAndWebService() - Devuelve un Doctrine_Collection con los métodos disponibles para un usuario pasando como parámetro su id y el nombre del servicio web.
ok 7 - isAvailableWebService() - Si está disponible el servicio web pasado como parámetro devuelve el objeto TbWebService
ok 8 - isAvailableProtocol() - Si está disponible el protocolo pasado como parámetro devuelve el objeto TbProtocol.
ok 9 - isAvailableProtocolForWebService() - Si está disponible el protocolo para un servicio web dado su nombre devuelve true.
ok 10 - isAvailableMethodForWebService() - Si está disponible un método para un servicio web devuelve true.
# Looks like everything went fine.
```

Figura 4.3 Corrección de las NC identificadas en las pruebas a la clase wsClassManager



### 3.2.2. Pruebas de Caja Negra

Las pruebas de Caja Negra se llevan a cabo sobre la interfaz del programa, centrándose en los requisitos funcionales del software para comprobar que todas las funcionalidades son operativas. De las técnicas propuestas por este método se utiliza la partición de equivalencia, dividiendo los campos de entrada en clases de datos de los que se derivan casos de prueba<sup>21</sup> (en lo adelante, CP).

A continuación se presenta el diseño de CP para el CU Configurar permiso de servicios web, el resto puede encontrarse en el Anexo # 8.

#### Diseño de caso de prueba CU Configurar permiso de servicios web

##### Sección 1: Configurar permiso de servicios web.

Escenario	Descripción	Grupo	Métodos	Respuesta del sistema	Flujo central
EC 1.1	Selecciona la opción de configurar los permisos de los grupos establecidos en la plataforma a un servicio web.			<p>Muestra un listado con los métodos de los servicios web disponibles en el sistema (en caso de no existir ninguno, muestra un mensaje de información).</p> <p>Brinda la posibilidad de seleccionar el siguiente criterio:</p> <ul style="list-style-type: none"> <li>➤ Grupo.</li> </ul> <p>Y permite:</p> <ul style="list-style-type: none"> <li>➤ Asociarle los métodos de los servicios web disponibles.</li> </ul>	Administración/Menú Principal/Servicios/ Permisos de Servicios Web

<sup>21</sup> **Caso de Prueba:** Conjunto de entradas con datos de prueba, condiciones de ejecución y resultados esperados, cuyo propósito es identificar y comunicar las condiciones que se llevarán a cabo en la prueba.

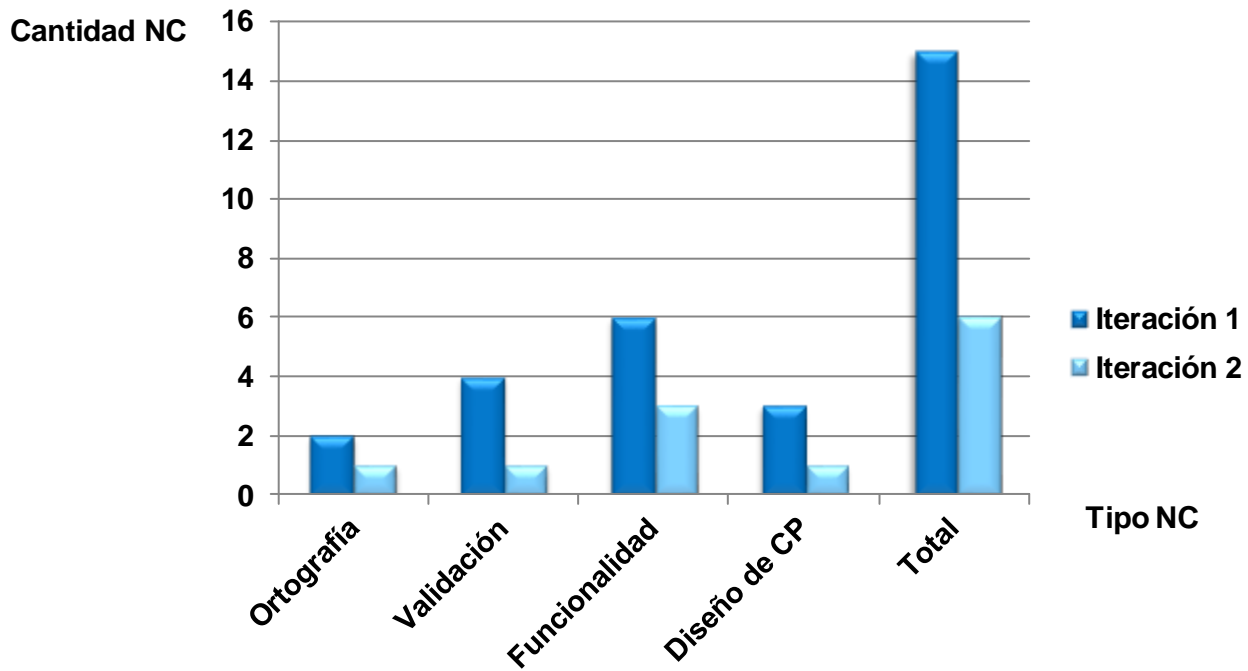


EC 1.2	Selecciona el grupo.	V	N/A	Permite asociarle los métodos de los servicios web disponibles.	Administración/Menú Principal/Servicios/Permisos de Servicios Web
EC 1.3	Asocia los métodos de los servicios web a los que tendrá acceso el grupo.	V	V	Valida los datos. Establece los permisos del servicio web. Muestra un mensaje de confirmación.	Administración/Menú Principal/Servicios/Permisos de Servicios Web/Botón Asociar
EC 1.4	Selecciona otro grupo para asociarle los servicios web a los que tendrá acceso.	V	N/A	Muestra un listado con los métodos de los servicios web disponibles. <u>Regresa al EC 1.3</u>	Administración/Menú Principal/Servicios/Permisos de Servicios Web
EC 1.5	Existen datos incompletos.	I	N/A	Muestra un mensaje de información. Muestra un indicador sobre el campo. <u>Regresa al EC 1.2</u>	Administración/Menú Principal/Servicios/Permisos de Servicios Web/Botón Asociar

**Tabla 4.1 Diseño de caso de prueba CU Configurar permiso de servicios web**

Para evaluar la solución desarrollada utilizando el método de Caja Negra, se realizaron varias iteraciones de pruebas por parte del equipo de desarrollo del proyecto, permitiendo validar el correcto funcionamiento de los componentes, así como su integración al resto de los módulos de la plataforma educativa Zera.

A continuación se muestra un gráfico con el resultado de las NC detectadas durante la ejecución de las dos iteraciones realizadas en la revisión del producto.



Gráfica 4.1 No conformidades

Cada una de las NC identificadas fue resuelta, influyendo en el mejoramiento de la propuesta para obtener componentes libres de errores y satisfacer las necesidades de los usuarios finales.

#### 4. Conclusiones Parciales

En este capítulo se validó la implementación de los componentes desarrollados. Se definieron los casos de prueba basados en los métodos de Caja Blanca y Caja Negra, que sirvieron para verificar el cumplimiento de las necesidades identificadas. Las pruebas efectuadas al producto permitieron corregir errores, que mejoran la calidad de la solución y posibilitan la aceptación final por parte del cliente.



### **Conclusiones Generales**

El desarrollo de la investigación aportó conocimientos sobre los mecanismos de interoperabilidad utilizados por las plataformas de gestión de aprendizaje Moodle, Sakai y Blackboard, lo que permitió seleccionar los principales estándares de interoperabilidad de la solución.

El uso de la metodología RUP, posibilitó guiar el proceso de desarrollo y generar toda la documentación necesaria para la construcción de las funcionalidades en el tiempo establecido, sirviendo como material de apoyo a futuras investigaciones.

La aplicación de estándares de interoperabilidad y seguridad en el desarrollo de la solución, demostró su validez, así como el cumplimiento de las características requeridas y de los objetivos propuestos, avalada por las pruebas aplicadas para la detección y corrección de errores.

Los componentes `sfAOBasicLTIPlugin` y `sfAOServicesPlugin` obtenidos como resultado de la investigación, permiten interoperar de forma estandarizada con aplicaciones de apoyo al proceso de enseñanza-aprendizaje como ROA, SGA, laboratorios virtuales, simuladores y asistentes matemáticos.



### Recomendaciones

Después de concluir la investigación, se recomienda:

- Evaluar la arquitectura actual de la plataforma educativa Zera, para determinar si es factible implantar una Arquitectura Orientada a Servicios.
- Realizar el estudio de otros estándares de interoperabilidad y seguridad que puedan ser incorporados a la plataforma.
- Estudiar y analizar otras tecnologías para el intercambio de datos como Websockets.



### Referencias bibliográficas

1. **María Agudelo, Mónica.** Plataformas Educativas. [En línea] [Citado el: 27 de septiembre de 2011.] <http://aprendeonline.udea.edu.co/banco/html/plataformaseducativas/>.
2. **Bernárdez, Mariano L.** Diseño, producción e implementación de E-learning. Bloomington, Indiana : AuthorHouse, 2007. ISBN: 978-1-4343-2108-4.
3. **Peñalvo, Francisco José García.** Estado actual de los sistemas e-learning. [En línea] Ediciones Universidad de Salamanca, 2008. [Citado el: 5 de noviembre de 2011.] [http://campus.usal.es/~teoriaeducacion/rev\\_numero\\_06\\_2/n6\\_02\\_art\\_garcia\\_penalvo.htm](http://campus.usal.es/~teoriaeducacion/rev_numero_06_2/n6_02_art_garcia_penalvo.htm).
4. **Ros, Miguel Zapata.** Sistemas de gestión del aprendizaje – Plataformas de teleformación. No 9, Universidad de Murcia : Revista de educación a distancia, 2003, Vol. 2003. ISSN: 1578-7680.
5. **Guzmán, Clara López.** Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning. Salamanca, España : Universidad de Salamanca, 2005.
6. **Tamayo Palma, Dariel Enrique y Avila Portales, Yisel .** Análisis y Diseño de los procesos de gestión de carreras para el sistema Akademos v2.0. Ciudad de La Habana, 2009.
7. **Hernández Ferrer, Kenia y Vázquez Mora, Mirio.** Laboratorio Virtual de Física: Propuestas de nuevas Funcionalidades en la UCI. Ciudad de La Habana, 2007.
8. **Ricardo, Anelys Vargas.** Programa realizado en el ambiente del asistente matemático DERIVE para la determinación de raíces de ecuaciones. Revista Ciencias.com, 2007. ISPN: EEyAFkAEEuAOkSggLo.
9. Portal Colombia Aprende. [En línea] Ministerio de Educación Nacional de Colombia. [Citado el: 5 de noviembre de 2011.] <http://www.colombiaprende.edu.co/html/directivos/1598/article-75224.html>.
10. IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries. New York : IEEE, 1990.
11. Boletín Trimestral del instituto Tecnológico de Informática. [En línea] 2006. [Citado el: 31 de mayo de 2012.] <http://www.iti.es>.





12. **Echeverri Villota, Ángela María y Monsalve Galeano, Elizabeth.** Análisis, diseño e implementación de un prototipo de repositorio de objetos virtuales de aprendizaje para la UCP. Pereira: Repositorio Institucional Biblioteca Universidad Católica, 2011.
13. **ISO.** Draft International Standard ISO/DIS 19101 Geographic information - Reference Model. 2001.
14. **Vázquez, José Manuel Márquez.** Estado del arte del eLearning. Ideas para la definición de una plataforma universal. Sevilla : Universidad de Sevilla, 2007.
15. **Ruiz, Elio San Cristóbal.** Metodología, estructura y desarrollo de interfaces intermedias para la conexión de laboratorios remotos y virtuales a plataformas educativas. Universidad Nacional de Educación a Distancia, 2010.
16. **Cabezuelo, Antonio Sarasa.** Diseño de repositorios digitales interoperables. Cataluña: Universidad Abierta de Cataluña, 2011.
17. **Simon, Bernd, Massart, David y Duval, Erik.** Simple Query Interface Specification. 2005.
18. **Fuente, Gema Bueno de la.** Análisis de la interoperabilidad entre los sistemas de apoyo a la formación de Tecminho. Madrid : Archivo Abierto Institucional de la Universidad Carlos III de Madrid, 2008.
19. Open Knowledge Initiative. Open Service Interface Definitions v 2.0.0. [En línea] 2004. [Citado el: 15 de noviembre de 2011.] [http://heanet.dl.sourceforge.net/project/okiproject/Doc%20%28previous%20OSID%20versions%29/Full%20Documentation%20Set/OSID\\_Documentation\\_rc6.1.pdf](http://heanet.dl.sourceforge.net/project/okiproject/Doc%20%28previous%20OSID%20versions%29/Full%20Documentation%20Set/OSID_Documentation_rc6.1.pdf).
20. **Sarasa, Antonio, Canabal, Manuel y Sacristán, Juan Carlos .** Agrega: Repositorios de objetos de aprendizaje interoperables. Salamanca : Universidad de Salamanca, 2008.
21. IMS GLC Learning Tools Interoperability Basic LTI Implementation Guide. [En línea] IMS Global Learning Consortium, 2010. [Citado el: 20 de noviembre de 2011.] <http://www.imsglobal.org/lti/blti/bltiv1p0/ltiBLTIimgv1p0.html>.
22. Guía Breve de Servicios Web. [En línea] W3C. [Citado el: 3 de octubre de 2011.] <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.
23. **Cerami, Ethan.** Web Services Essentials. O'Reilly, 2002. ISBN: 0-596-00224-6.



24. SOAP Version 1.2 W3C Recommendation. [En línea] W3C. [Citado el: 20 de noviembre de 2011.] <http://www.w3.org/TR/soap12-part1/>.
25. **Mateu, Carles**. Desarrollo de aplicaciones web. Cataluña : Universidad Abierta de Cataluña, 2004. ISBN: 84-9788-118-4.
26. **Ralf Wirdemann, Hamburgo**. Desarrollo REST con Rails. España, 2007.
27. **Fielding, Roy Thomas**. Architectural Styles and the Design of Network-based Software Architectures. Irvine : University of California, 2000.
28. JSON-RPC Specifications. [En línea] JSON-RPC. [Citado el: 15 de enero de 2012.] <http://json-rpc.org/wiki/specification>.
29. **Inc, Adobe Systems**. Action Message Format -- AMF 3. 2006.
30. Guía de Referencia del Programador: Zend Framework. [En línea] Zend Technologies Inc., 2009. [Citado el: 31 de mayo de 2012.] <http://code.google.com/p/zendframework-hispano>.
31. Guía Breve de Tecnologías XML. [En línea] W3C. [Citado el: 6 de octubre de 2011.] <http://www.w3c.es/Divulgacion/GuiasBreves/TecnologiasXML>.
32. **Moreno, Luciano**. Transacciones seguras. [En línea] [Citado el: 15 de octubre de 2011.] [http://www.educastur.princast.es/fp/hola/hola\\_bus/cursos/curso17/documentos/seguras.pdf](http://www.educastur.princast.es/fp/hola/hola_bus/cursos/curso17/documentos/seguras.pdf).
33. **Atkinson, Bob y Della Libera, Giovanni**. Web Services Security (WS-Security). International Business Machines Corporation, Microsoft Corporation, 2002.
34. Guía Breve de Seguridad. [En línea] W3C. [Citado el: 20 de octubre de 2011.] <http://www.w3c.es/Divulgacion/GuiasBreves/Seguridad>.
35. OAuth. [En línea] OAuth.net. [Citado el: 20 de enero de 2012.] <http://oauth.net/>.
36. **Figuroa, Roberth G., J. Solís, Camilo y Cabrera, Armando A**. "Metodologías Tradicionales VS. Metodologías Ágiles". Loja, Ecuador : Universidad Técnica Particular de Loja.
37. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady** . El Lenguaje Unificado de Modelado. Manual de Referencia. Madrid : Addison-Wesley, 1998. ISBN: 84-7829-037-0.



38. **Lisette Feria Sotto, Yerandy Manso Guerra.** Análisis, diseño, implementación y prueba de una extensión para la Plataforma Zera, que permita la incorporación del estándar SCORM. Ciudad de La Habana, 2011.
39. **Pérez, Javier Eguíluz.** Introducción al XHTML. LibrosWeb. 2008.
40. Guía breve de CSS. [En línea] W3C. [Citado el: 5 de diciembre de 2011.] <http://www.w3c.es/divulgacion/guiasbreves/hojasestilo>.
41. **Pérez, Javier Eguíluz.** Introducción a JavaScript. Librosweb. 2009.
42. —. Introducción a AJAX. Librosweb. 2008.
43. ¿Qué se puede hacer con PHP? [En línea] Php.net. [Citado el: 15 de diciembre de 2011.] <http://www.php.net/manual/es/intro-whatcando.php>.
44. **Potencier, Fabien.** Symfony, la guía definitiva. Librosweb. 2008.
45. **Group, The PostgreSQL Global Development.** PostgreSQL 8.1.0 Documentation. 1996.
46. **S.Pressman, Roger.** Ingeniería del Software. Un enfoque práctico. Sexta Edición. Mc Graw-Hill, 2005. ISBN: 970-10-5473-3.



### **Bibliografía consultada**

**Peñalvo, Francisco José García.** Estado actual de los sistemas e-learning. [En línea] Ediciones Universidad de Salamanca, 2008. [Citado el: 5 de noviembre de 2011.] [http://campus.usal.es/~teoriaeducacion/rev\\_numero\\_06\\_2/n6\\_02\\_art\\_garcia\\_penalvo.htm](http://campus.usal.es/~teoriaeducacion/rev_numero_06_2/n6_02_art_garcia_penalvo.htm).

**Cabezuelo, Antonio Sarasa.** Diseño de repositorios digitales interoperables. Cataluña : Universidad Abierta de Cataluña, 2011.

**Simon, Bernd, Massart, David y Duval, Erik.** Simple Query Interface Specification. 2005.

IMS GLC Learning Tools Interoperability Basic LTI Implementation Guide. [En línea] IMS Global Learning Consortium, 2010. [Citado el: 20 de noviembre de 2011.] <http://www.imsglobal.org/lti/blti/bltiv1p0/ltiBLTIimgv1p0.html>.

SOAP Version 1.2 W3C Recommendation. [En línea] W3C. [Citado el: 20 de noviembre de 2011.] <http://www.w3.org/TR/soap12-part1/>.

**Mateu, Carles.** Desarrollo de aplicaciones web. Cataluña : Universidad Abierta de Cataluña, 2004. ISBN: 84-9788-118-4.

JSON-RPC Specifications. [En línea] JSON-RPC. [Citado el: 15 de enero de 2012.] <http://json-rpc.org/wiki/specification>.

**Atkinson, Bob y Della Libera, Giovanni.** Web Services Security (WS-Security). International Business Machines Corporation, Microsoft Corporation, 2002.

OAuth. [En línea] OAuth.net. [Citado el: 20 de enero de 2012.] <http://oauth.net/>.

**Jacobson, Ivar, Booch, Grady y Rumbaugh, James .** El Proceso Unificado de Desarrollo de Software. Madrid : Addison-Wesley, 2000. ISBN: 84-7829-036-2.

**Rumbaugh, James, Jacobson, Ivar y Booch, Grady .** El Lenguaje Unificado de Modelado. Manual de Referencia. Madrid : Addison-Wesley, 1998. ISBN: 84-7829-037-0.

**Potencier, Fabien.** Symfony, la guía definitiva. Librosweb. 2008.

**S.Pressman, Roger.** Ingeniería del Software. Un enfoque práctico. Sexta Edición. Mc Graw-Hill, 2005. ISBN: 970-10-5473-3.



**Murray, Peter.** Basic LTI Portlet. [En línea] 2006. [Citado el: 15 de febrero de 2012.] <https://confluence.sakaiproject.org/display/RES/Intersection+between+Sakai+and+Fedora>.

Informe sobre Sakai 2.8, Moodle 2.0 y Bb Learn 9.1. Andalucía : eLearning Solutions S.L, 2011 .

**Marquina, Ernesto, Parra, José David y Rodríguez Mier, Mario .** Guía de Patrones, Prácticas y Arquitectura .NET Versión 2.5 : Microsoft Services, 2008.

**Hamilton, Kim y Miles, Russell .** Learning UML 2.0 : O'Reilly, 2006. ISBN: 0-596-00982-8.

**Cerami, Ethan.** Web Services Essentials: O'Reilly, 2002 . ISBN: 0-596-00224-6.

**Ravioli, Pablo.** Lenguaje de programación para paginas web. [En línea] 2011. [Citado el: 3 de 12 de 2011.] <http://www.monografias.com/trabajos7/html/html.shtml> .

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

NetBeans. [En línea] Oracle Corporation. [Citado el: 31 de mayo de 2012.] <http://www.netbeans.org>.



### Glosario de términos

**Administrador Central:** Usuario que se encarga de la administración global del sitio y tiene permisos sobre los principales componentes de la plataforma.

**Aplicación cliente:** Herramienta que funciona sobre el propio sistema operativo.

**Aplicación externa:** Herramienta externa a la plataforma creada con el fin de apoyar los procesos de enseñanza-aprendizaje.

**Aplicación web:** Herramienta que permite a los usuarios acceder a un servidor web mediante un navegador.

**Apoyo:** Módulo de la plataforma educativa Zera que utiliza aplicaciones clientes como simuladores, asistentes matemáticos y laboratorios virtuales.

**Asistente matemático:** Aplicación de escritorio que permite el estudio de funciones representadas de forma gráfica.

**Bachiller:** Es el subsistema encargado de la gestión académica en la plataforma.

**Componente:** Es una extensión que encapsula funcionalidades que permiten la reutilización de código.

**Laboratorio virtual:** Sistema informático que pretende simular el ambiente de un laboratorio real mediante simulaciones interactivas que permiten desarrollar prácticas de laboratorio a distancia.

**Módulo:** Unidad de un sistema que engloba un conjunto de funcionalidades o componentes que tienen un mismo fin u objetivo.

**OA:** Objeto de aprendizaje. Es cualquier recurso con intención formativa, compuesto de uno o varios elementos digitales descritos con metadatos, que puede ser utilizado y reutilizado dentro de un entorno e-learning.

**Plataforma educativa Zera:** Plataforma educativa destinada a apoyar el proceso de enseñanza-aprendizaje.

**ROA:** Repositorio de objetos de aprendizaje. Es una herramienta que facilita la colección o almacenamiento, gestión y reutilización de objetos de aprendizaje a través de la red.



**Servicio web:** Conjunto de funcionalidades que puede proveer determinado sistema para ser utilizadas por otra aplicación.

**SGA:** Sistema de gestión académica. Herramienta que permite almacenar y procesar toda la información referente al proceso docente de un centro de estudios.

**Simulador:** Aplicación de escritorio que permite realizar simulaciones de fenómenos de la realidad.

**Subsistema:** Componente principal de un sistema que agrupa funcionalidades o elementos relacionados.

**Usuario:** Individuo que tiene acceso a las funcionalidades de la plataforma, representado mediante una dirección de correo electrónico.