

2011

**Universidad de las Ciencias Informáticas**  
***Facultad 4***



**Título:** “Implementación del nivel A de IMS-LD para la herramienta de autor web CRODA”

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

**Autor**

Mario Manuel Falcón García

**Tutor**

Ing. Osvaldo E. Stable Vilches

**Cotutor**

MCs. María Caridad Valdés Rodríguez

La Habana, junio de 2011  
“Año 53 de la Revolución”

# Declaración de autoría

Por este medio declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con este trabajo.

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

Mario Manuel Falcón García

Ing. Osvaldo E. Stable Vilches

---

---

Firma del Autor

Firma del Tutor

## Resumen

El empleo de diferentes estándares y especificaciones del ámbito del e-learning ha permitido lograr la interoperabilidad en las plataformas destinadas a gestionar los Objetos de Aprendizaje (OA). El diseño instruccional (DI) es una herramienta esencial para incrementar la calidad de los cursos y programas de formación que no ha sido suficientemente atendida en la teleformación (educación a distancia). La Universidad de las Ciencias Informáticas (UCI) no está ajena a estos procesos. En el Departamento de Herramientas Educativas del Centro FORTES perteneciente a la UCI se desarrolló una herramienta de autor web: CRODA. La misma, a pesar de contar con todas las características y funcionalidades que le permitían gestionar OA, era incapaz de incrementar la calidad de los recursos educativos que generaba más allá de las bondades que le ofrecía el estándar SCORM 1.2. El presente trabajo describe el proceso de implementación de un módulo de Diseño Instruccional que, a través de la especificación IMS-LD, hasta su nivel A, posibilitó la realización del diseño instruccional de los OA generados en la herramienta de autor CRODA. Este módulo contribuyó al incremento de la calidad de los recursos didácticos confeccionados para la docencia en la UCI.

**Palabras claves:** CRODA, diseño instruccional, e-learning, estándares, IMS-LD, objeto de aprendizaje.

<b>Capítulo 1. Fundamentación Teórica.....</b>	<b>11</b>
Introducción .....	11
1.1 Diseño instruccional.....	11
1.2 Especificación IMS – Learning Design (IMS – LD).....	12
1.2.1 Funcionamiento general de IMS-LD.....	13
1.2.2 Niveles de IMS-LD.....	15
1.2.2.1 Nivel A .....	15
1.2.2.2 Nivel B .....	17
1.2.2.3 Nivel C .....	18
1.3 Herramientas de autor. ....	18
1.3.1 EasyProf .....	19
1.3.2 Authorware .....	20
1.3.3 Reload .....	21
1.3.4 CompendiumLD.....	23
1.3.5 MOT y MOT plus .....	24
1.4 Herramientas y tecnologías a utilizar. ....	26
1.4.1 Metodología de desarrollo. ....	27
1.4.1.1 RUP.....	27
1.4.2 Lenguaje Unificado de Modelado (UML) .....	29
1.4.3 Visual Paradigm como herramienta CASE.....	29
1.4.4 Servidor web Apache.....	30
1.4.5 Sistemas de gestión de bases de datos.....	32
1.4.5.1 PostgreSQL .....	33
1.4.5.2 Exist-DB.....	34
1.4.6 Framework de desarrollo Symfony.....	34

## 5 | Índice

1.4.7 Librería de javascript Ext JS .....	37
1.4.8 IDE de desarrollo NetBeans .....	38
1.4.9 Lenguajes de programación y etiquetado .....	40
1.4.9.1 PHP .....	40
1.4.9.2 Javascript .....	41
1.4.9.3 XML .....	42
1.5 Conclusiones del capítulo. ....	43
<b>Capítulo 2. Implementación del módulo de Diseño Instruccional. ....</b>	<b>44</b>
Introducción .....	44
2.1 Modelo de implementación .....	44
2.2 Diagrama de paquetes.....	44
2.3 Diagramas de Componentes .....	46
2.4 Modelo de Datos.....	52
2.4.1 Modelo de base de datos relacional.....	52
2.4.1.1 Descripción de las tablas de la base de datos postgresQL.....	53
2.4.2 Modelo de datos XML.....	56
2.5 Diagrama de despliegue .....	57
2.6 Conclusiones del capítulo. ....	58
<b>Capítulo 3. Pruebas del módulo de Diseño Instruccional. ....</b>	<b>59</b>
Introducción .....	59
3.1 Prueba de Caja Negra .....	59
3.1.1 Caso de Prueba.....	60
3.2 Conclusiones del capítulo. ....	72
<b>Conclusiones generales. ....</b>	<b>74</b>
<b>Recomendaciones .....</b>	<b>75</b>
<b>Bibliografía .....</b>	<b>76</b>
<b>Glosario de términos .....</b>	<b>79</b>

## Introducción

El vertiginoso desarrollo de las tecnologías de la información y las comunicaciones ha impactado considerablemente el campo de la educación. En este ámbito la formación presencial tradicional ha dado paso a procesos de enseñanza-aprendizaje no presenciales donde se emplea el e-learning. “El e-learning es un conjunto de tecnologías, aplicaciones y servicios orientados a facilitar la enseñanza y el aprendizaje a través de Internet/Intranet, que facilita el acceso a la información y la comunicación con otros participantes”. (Red TTnet, 2005)

Para darle solución a la problemática del interés de los participantes de los cursos en las instituciones que utilizan el e-learning se emplean métodos de diseño didácticos. Cuando estos métodos se vuelven sistémicos se habla entonces de diseño instruccional (DI). El DI es una herramienta esencial para incrementar la calidad de los cursos y programas de formación que no ha sido suficientemente atendida en la teleformación (educación a distancia).(Orellana, Suárez, y Belloch 2001)

El DI es un proceso sistémico y sistemático por medio del cual a partir del análisis de una necesidad de aprendizaje, se seleccionan y desarrollan las actividades y recursos para satisfacerla, así como los procedimientos para evaluar si dicho aprendizaje fue alcanzado. (Gil 2009)

El diseño instruccional (DI) sería entonces la manifestación de un sistema de aprendizaje, en el que se tienen en cuenta una serie de necesidades y particularidades del estudiante. El mismo tiene determinadas características en las que se apoya el e-learning para lograr mayor eficiencia y calidad en el aprendizaje. Existen diferentes modelos de DI que tienen determinada eficacia en dependencia del ambiente donde se implementarán o la consideración del educador que selecciona el más adecuado a sus necesidades y las de sus estudiantes.

## 7 | Introducción

Los modelos de DI permiten hacer más efectivo el proceso de aprendizaje mediante la realización de una planificación adecuada propiciando así la adquisición de las destrezas necesarias para la utilización de las tecnologías en las actividades llevadas a cabo por los diferentes actores propuestos. Conceden especial importancia a la formación de profesores conscientes de las limitaciones y posibilidades que las tecnologías les ofrecen, lo que les permite elaborar propuestas didácticas encaminadas a que, los procesos de formación y sus estudiantes, respondan a las exigencias del mundo actual.

La mayoría de las plataformas que trabajan el e-learning utilizan estándares con el consiguiente beneficio de interoperabilidad entre distintas de ellas y una mayor coherencia en el empaquetado de recursos. Su empleo garantiza la durabilidad, interoperabilidad, accesibilidad y reutilización de los mismos.

Entre los estándares empleados, existe uno, que abarca satisfactoriamente los conceptos fundamentales del diseño instruccional: el IMS – Learning Design (IMS – LD). IMS –LD es un lenguaje de modelado educativo que tiene como objetivo definir formalmente una estructura semántica para anotar los procesos de enseñanza y aprendizaje, y así convertirlos en entidades reutilizables entre diferentes cursos y aplicaciones.(Berlanga 2005)

La Universidad de las Ciencias Informáticas (UCI) no es ajena a estos procesos y en el desarrollo que lleva a cabo de diferentes herramientas y plataformas que emplean e-learning tiene en cuenta la utilización de diferentes estándares. En el Departamento de Producción de Herramientas Educativas perteneciente al Centro FORTES de la UCI se desarrolla la herramienta de autor web CRODA. CRODA es una herramienta diseñada con el objetivo de apoyar la creación de OA que utiliza la tecnología de los mismos siguiendo las normas del estándar SCORM 1.2. Entre algunas de sus funciones se encuentran la creación de diferentes tipos de preguntas, desarrollo de plantillas con la estructura inicial que puede tener algún OA, además la posibilidad de poder modificarlas y hacerlas públicas a los demás que interactúen con la herramienta. Posteriormente se incluyeron funcionalidades para apoyar el trabajo colaborativo.

## 8 | Introducción

CRODA es una herramienta de autor que a pesar de contar con todas las características y funcionalidades para gestionar OA es incapaz de incrementar la calidad de los recursos educativos que genera más allá de las bondades que le ofrece el estándar SCORM 1.2. Partiendo de ello surge la siguiente interrogante cómo **problema de investigación**: ¿Cómo propiciar el diseño instruccional de los objetos de aprendizaje, generados por la herramienta de autor web CRODA, que posibilite el incremento de la calidad de los recursos didácticos para la docencia en la UCI?

Es posible definir entonces como **objeto de investigación**, el proceso para el diseño instruccional de recursos didácticos mediado por las tecnologías.

Como **objetivo general** se plantea, implementar un módulo que a través de la especificación IMS-LD, hasta su nivel A, posibilite la realización del diseño instruccional de OA generados en la herramienta de autor CRODA.

Como **objetivos específicos** se relacionan los siguientes:

- Evaluar la propuesta de análisis y el diseño del módulo IMS-LD, hasta el nivel A, para CRODA.
- Implementar el nivel A del estándar IMS-LD en CRODA.
- Probar el nivel A del estándar IMS-LD en CRODA.

Para el cumplimiento de los objetivos expuestos se plantean las siguientes **tareas**:

1. Estudio de referentes teóricos relacionados con la implementación del estándar IMS-LD hasta el nivel A.
2. Evaluación del análisis y el diseño del módulo de IMS-LD, nivel A, para CRODA.



## 9 | Introducción

3. Elaboración de la propuesta de implementación del módulo de Diseño Instruccional basado en el estándar IMS – LD, hasta el nivel A, en CRODA.
4. Implementación del módulo de Diseño Instruccional basado en el estándar IMS- LD, hasta el nivel A, en CRODA.
5. Ejecución de las pruebas del módulo de Diseño Instruccional en CRODA.

Derivado de los objetivos propuestos a alcanzar y las tareas relacionadas con los mismos se define como **campo de acción**, las especificaciones que posibilitan el diseño instruccional de objetos de aprendizaje en la herramienta de autor CRODA.

En vistas a guiar el desarrollo de este trabajo se toma la siguiente **idea a defender**: si se desarrolla un módulo en la herramienta de autor CRODA que permita la realización del diseño instruccional a partir de la especificación IMS-LD, hasta su nivel A, entonces se propicia la elaboración de OA con mayor calidad como recurso didáctico para la docencia en la UCI.

Para el cumplimiento de estas tareas se hace necesaria la utilización de algunos **métodos de investigación** como el **Analítico-Sintético**, mediante el cual se extrae la información más relevante acerca de las metodologías, tecnologías y herramientas posibles a ser utilizadas en el desarrollo de este módulo, lo que permite seleccionar y sintetizar las características más acordes a nuestros objetivos, tomando en cuenta las ventajas de cada una ; el **Histórico-Lógico**, que permite llevar a cabo un estudio de la evolución de las diferentes herramientas de diseño instruccional existentes actualmente para evaluar posibles mejoras a incluir en la nuestra. La **Observación** con la cual se toman las mejores prácticas de sistemas con la misma finalidad del que pretendemos desarrollar así como sus vulnerabilidades para convertirlas en ventajas de este nuevo producto.

El presente trabajo se estructura en tres capítulos:

## 10 | Introducción

- **Capítulo 1:** Se hace una revisión de los principales conceptos asociados al diseño instruccional así como a la especificación IMS – LD, su estructura y funcionamiento. Además se detallan diferentes herramientas que implementan dicho estándar y las tecnologías necesarias para su implementación.
- **Capítulo 2:** Se detalla el proceso de implementación del módulo de Diseño Instruccional. En el mismo se explican el modelo de implementación, los diagramas de paquetes así como los de componentes del sistema. Además se describen los modelos de datos, tanto el relacional como el xml. Incluye una explicación de las funcionalidades más críticas y complejas.
- **Capítulo 3:** Se reproduce el proceso de pruebas. Se desglosan las pruebas de caja negra a través de casos de prueba estructurados a partir de casos de uso. Se relacionan los errores detectados así como el proceso de depuración de los mismos en tres iteraciones.

## Capítulo 1. Fundamentación Teórica

### Introducción

En este capítulo se encuentra un bosquejo de los principales conceptos necesarios para entender el funcionamiento del módulo de Diseño Instruccional basado en la especificación IMS - LD. También se hace una revisión de distintas herramientas que gestionan DI y se selecciona una de ellas para la implementación del nivel A del estándar IMS – LD así como una serie de tecnologías y herramientas necesarias para este fin.

### 1.1 Diseño instruccional.

El rápido crecimiento de Internet y otros modelos de enseñanza que contemplan la utilización de medios electrónicos para su aprovechamiento cabal, ha planteado a su vez un gran problema el cual consiste en cómo aprovechar de mejor manera las potencialidades y facilidades que los alumnos tienen a la hora de asimilar el conocimiento. El diseño instruccional es la solución hallada para satisfacer estas necesidades.

Diseño instruccional, en su definición más sencilla, es un proceso sistemático, planificado y estructurado donde se produce una variedad de materiales educativos atemperados a las necesidades de los educandos, asegurándose así la calidad del aprendizaje.(Yukavetsky 2008)

El DI se nutre de diferentes ciencias como son el caso de las ciencias sociales, de la ingeniería, de la información y otras ciencias en general. En el mismo se hace un completo análisis de las necesidades y metas educativas a cumplir y, posteriormente, se diseña e implementa un mecanismo que permita alcanzar esos objetivos.

## 1.2 Especificación IMS – Learning Design (IMS – LD)

La especificación IMS-LD, desarrollada por la Open University of Netherlands en el año 2003, está centrada en la flexibilidad para codificar escenarios de aprendizaje para estudiantes. No tiene modelo pedagógico alguno asociado, lo que en otras palabras significa, que puede ser utilizado como meta-modelo pedagógico con la capacidad de asimilar cualquier escenario de aprendizaje.

Esta especificación, pedagógicamente neutra, tiene su basamento en un lenguaje de modelado educativo que sigue el objetivo de definir formalmente una estructura semántica para conformar el proceso de enseñanza-aprendizaje. Describe una agrupación de actividades que posteriormente realizarán alumnos y profesores, indicando en qué momento, con qué recursos didácticos y servicios y además bajo qué condiciones lo harán. Pedagógicamente neutral se refiere a que IMS – LD no se implementa siguiendo un modelo pedagógico específico ya que posee la capacidad de representar los existentes.

IMS – LD refuerza la asociación entre objeto didáctico reutilizable, actividades y roles de personas que intervienen en el proceso educativo, asemejándose más al modelo apreciable en las clases presenciales.(Vázquez 2007)

Por lo anteriormente expuesto, a pesar de que IMS - LD no es un estándar sino una especificación, se considera el idóneo para lograr la incorporación del diseño instruccional en la herramienta web CRODA sin tomar partido pedagógico alguno lo que le confiere un incremento en la flexibilidad del manejo de los OA.

Frecuentemente los términos estándar y especificación se utilizan indistintamente, no obstante, es importante puntualizar su diferencia. Si una tecnología, formato o método ha sido ratificado por algún organismo oficial de estandarización, se trata de un estándar. Pero si una tecnología, formato o método propuesto no ha sido aprobado por algún organismo oficial de estandarización, se trata de una

## 13 | Capítulo I. Fundamentación Teórica

especificación. Aunque, en algunos casos, una especificación puede considerarse un estándar de facto si su uso es extendido y entretanto se ratifica como estándar. (Corley 2003)

Según Singh & Reed (2002), no hay estrictamente estándar e-learning, sino grupos desarrollando especificaciones. No es común hacer en la práctica la diferencia entre unos y otros, se puede perfectamente encontrar a algunas especificaciones e-learning llamadas como estándares. Lo que no está mal del todo, debido a que muchas de ellas han sido adoptadas por importantes organizaciones.

Es válido aclarar además que, IMS – LD por sí sólo, no garantiza la adquisición de conocimientos. Esto depende de otros factores como son el uso al que sea sometido el recurso educativo en dependencia de las actividades realizadas por profesores y estudiantes. Un buen punto de partida sería el escoger adecuadamente el modelo de diseño instruccional a emplear para reforzar el proceso de enseñanza-aprendizaje.

### 1.2.1 Funcionamiento general de IMS-LD

En IMS - LD no se incluyen los materiales o recursos que utiliza. En este caso al definirse un Diseño de Aprendizaje (Learning Design [LD]) debe añadirse posteriormente una Unidad de Aprendizaje (Unit of Learning [UoL]). Una UoL representa la unidad completa de formación y auto-contenida como, por ejemplo, un curso o una lección.

En sí, una UoL contiene todos los recursos asociados como ejercicios, objetos u recursos de aprendizaje e información para configurar distintos servicios (ej. conferencia, correo electrónico, búsqueda).

Para ejemplificar los conceptos descritos anteriormente proponemos los siguientes esquemas que muestran un paquete de contenidos *IMS Content Packaging* (Ver Figura 1).



Figura 1: Estructura de un paquete IMS Content Packaging.

Disponible en: [http://bibliodoc.uci.cu/TD/TD\\_02852\\_10.pdf](http://bibliodoc.uci.cu/TD/TD_02852_10.pdf)



Figura 2: Estructura de una Unidad de Aprendizaje (UoL) en IMS Learning Design.

Disponible en: [http://bibliodoc.uci.cu/TD/TD\\_02852\\_10.pdf](http://bibliodoc.uci.cu/TD/TD_02852_10.pdf)

Para fusionar la definición del Diseño de Aprendizaje (LD) con los recursos asociados mediante una Unidad de Aprendizaje (UoL) es necesario un mecanismo de empaquetamiento ( *Ver Figura 2*). En sentido general un usuario edita un OA en una herramienta destinada a este fin, para el caso que nos ocupa CRODA. Posteriormente es esta herramienta la encargada de salvar los procedimientos definidos sobre el OA en materia de Diseño Instruccional como un archivo XML (*eXtensible Markup Language*) y garantizando que cumpla con los parámetros descritos por IMS – LD (ambientes, roles, obras, métodos, actividades, propiedades y notificaciones).

### 1.2.2 Niveles de IMS-LD

En aras de facilitar la implementación de IMS-LD y hacerlo de forma progresiva, la especificación se divide en tres niveles que pueden denominarse 1, 2 y 3 o A, B y C como se toma de ahora en adelante. El primer nivel (A) es la base del segundo (B) y así sucesivamente. Estos últimos dos niveles le agregan funcionalidad y potencia.

La existencia de estos niveles no indica que haya que implementarlos todos para lograr la implementación de la especificación. Los mismos se utilizan en la profundidad que requiera la entidad en la que se utilizarán en dependencia de la complejidad de sus necesidades.

#### 1.2.2.1 Nivel A

El Nivel A de la especificación es donde se incluyen los conceptos básicos los que consisten en las obras divididas en actos en la que distintos actores desempeñan diferentes roles. Las estructuras de actividades presentes en este nivel son la esencia de los caminos del aprendizaje con ramificaciones y dan paso a procesos colaborativos. En el seguimiento de las actividades es posible respetar su orden de aparición o no.

## 16 | Capítulo I. Fundamentación Teórica

Es este nivel no será posible consultar o modificar los valores de las propiedades de los atributos de los elementos básicos definidos en él. Ello provoca que el flujo de una actividad sea fijo y que los resultados de una determinada actividad condicionen la ejecución o no del resto. Aún en estas condiciones es posible crear modelos en los que intervengan diferentes participantes lo que es un adelanto frente a modelos que se restringen a un solo tipo de usuario.

El nivel A, constituye el núcleo y define las actividades de aprendizaje, las de soporte, las estructuras, los entornos, el método, la instancia, los actos, los roles, los recursos y la comunicación entre ellos. Además los usuarios podrán usar diversos recursos externos como enlaces web y algunos servicios como foros y chats para facilitar la colaboración entre estos.(Vázquez 2007)

### **Elementos del Nivel**

**Persona (Person):** Es la persona que adopta un determinado rol.

**Rol (Role):** Por defecto son, learner y staff (aprendiz y grupo de trabajo). Aunque es posible crear unos nuevos a partir de estos dos originales que pueden heredar características de sus predecesores.

**Actividad (Activity):** Existen dos tipos de actividades, learning activity y support activity (actividades de apoyo y actividades de aprendizaje), así como estructuras de actividades que agrupan dos o más de éstas e incluso otras estructuras.

**Ambiente (Environment):** El ambiente es una colección estructurada de servicios (services), objetos de aprendizajes (learning object) y sub-ambientes.

**Método (Method):** El método no es más que el mecanismo que permite relacionar los roles, las actividades y los ambientes. Se compone de tres elementos:

- **Obra (Play):** Un método se compone de una o más obras, y éstas a su vez de uno o varios actos.



## 17 | Capítulo I. Fundamentación Teórica

- **Acto (Act):** Es aquí donde se establecen las distintas actividades en Partes de Rol adecuadas. Un acto involucra a un participante o a varios, pero cada uno de ellos lleva a cabo una única actividad.
- **Parte del Rol (Role-Part):** Asocia únicamente un solo rol a una única actividad, teniendo como objetivo, especificar lo que ese rol debe cumplir en el Acto.

### 1.2.2.2 Nivel B

El nivel B aporta las propiedades y condiciones a la especificación. Las propiedades en sí son pares de atributos-valor que parten de una configuración (valor) inicial y pueden ser alteradas durante la ejecución de la Unidad de Aprendizaje.

Para el caso de las condiciones, son consultas que se ejecutan sobre las propiedades en un momento determinado. De esta forma es posible alterar el orden de ejecución de las actividades al verificar una actividad si otra se ha cumplido satisfactoriamente o ha arrojado los resultados esperados.

En síntesis, este nivel permite que el resultado de una actividad influya en el resto a través de la comprobación de este resultado por una condición determinada a cumplir. De esta forma se modifica el proceso de aprendizaje haciendo que el flujo de las actividades no sea fijo como en el nivel anterior.

EL nivel B, añade condiciones y propiedades, locales y globales, que permiten almacenar información.(Vázquez 2007)

#### **Elementos del Nivel**

**Condiciones (Condition):** Son condiciones que permiten la personalización así como secuencias e interacciones más elaboradas.

**Propiedades (*Property*):** Son propiedades y pueden ser utilizadas para dirigir las actividades de aprendizaje así como también para registrar los resultados.

### 1.2.2.3 Nivel C

Aún con la incorporación de los otros dos niveles (A y B) las actividades se ejecutan en determinada secuenciación. Una actividad determinada, incluso si es bloqueada o activada por una previa, debe esperar su turno de ejecución. No es posible realizar actividades en paralelo para un mismo rol en esta especificación.

Es por ello que la especificación introduce el mecanismo de notificación o de envío de mensajes a través de eventos. Con esta opción las actividades pueden ser interrumpidas en plena ejecución de ser necesario incluso por el propio *Learning Management System* (LMS). Con ello es posible realizar flujos de aprendizaje modificables en tiempo real y así alterar la secuenciación de las actividades.

Concluyendo así, este nivel añade notificaciones, es decir, acciones asociadas a eventos que se disparan automáticamente. (Vázquez 2007)

#### **Elementos del Nivel**

**Notificación (*Notification*):** Son notificaciones, es decir, acciones asociadas a eventos que se disparan automáticamente. Se originan por el resultado de una actividad y pueden producir un comportamiento distinto al esperado en una actividad en tiempo real.

## 1.3 Herramientas de autor.

Existen distintos tipos de herramientas de apoyo al e-learning, entre ellas los repositorios de objetos de aprendizaje y las herramientas de autor. Las herramientas de autor son aplicaciones que permiten la

creación de diferentes contenidos educativos. Estas posibilitan al usuario realizar un producto con fines educativos y generan un entorno de aprendizaje dinámico. Los objetivos se pueden adecuar según los conocimientos y las habilidades que el autor desee desarrollar.

Las herramientas de autor tienen la posibilidad de cambiar el flujo de la información según las necesidades del alumno, relacionar palabras, incluir cuestionarios y además, marcadores que evalúan el conocimiento que el estudiante vaya adquiriendo.

### 1.3.1 EasyProf

Es una herramienta de autor privada, basada en java y al estarlo es necesidad de ella tener algunos plugins del Sun Java, como lo son el Java Runtime y Java media frameworks. Dicha herramienta también soporta diferentes formatos como, por ejemplo: de imágenes (PNG, JPG, GIF), de audio (WAV, MP3, AU, MID), de video (WMV, QT, RL) y además algunos de texto, tales como, RTF, TXT, HTML y PDF, asimismo posee contenidos publicados en diferentes formatos como los antes mencionados. Esta herramienta utiliza el estándar SCORM 1.2 de la ADL, cumpliendo así con los previos requisitos que deben tener los contenidos para ello.

Como otras características se tienen además, que brinda la posibilidad de ver el contenido en otros idiomas, gestión de metadatos, compatibilidad con funcionalidades de terceros, gestión de herramientas complementarias al contenido compatible con SCORM, creación de diferentes exámenes.(Villar 2007)

*Como algunas ventajas de esta herramienta se tienen que:*

- Presenta facilidad de gestión de acciones y eventos de contenidos.
- Permite la publicación de contenidos PDF sin herramientas adicionales de ADOBE.

## 20 | Capítulo I. Fundamentación Teórica

- Presenta una ayuda y manual bastante completas.
- Desarrolla contenidos compatibles y certificables con SCORM 1.2.
- Presenta la posibilidad de incorporar herramientas complementarias para la accesibilidad en contenidos.

*Algunas desventajas:*

- Pocos formatos de publicación.
- Escaso soporte de formato de elementos multimedia.
- Necesita instalar los plugins (java) para su posterior utilización.
- No soporta aún el estándar SCORM 2004.(Denis y Varona 2010)

### 1.3.2 Authorware

Esta herramienta, al igual que la anterior, es privativa. Presenta algunos requisitos para su instalación como la cantidad de memoria RAM que emplea, el espacio que utiliza en disco, etc. Soporta diferentes tipos de formatos como por ejemplo, de imagen (GIF, JPG, PNG, BMP, TGA, entre otros), de audio (AIFF, MIDI, WAV, MP3), formatos de videos (WAN, AVI, MPG, MOV) y además los de textos, tales como RTF, XML, HTML, TXT y PDF.

La misma es compatible con Internet Explorer y Firefox. Utiliza el estándar IMS, así como el SCORM 1.2 de la ADL. Posibilita la incorporación de flash o código javascript, editor de metadatos SCORM, varios OA ya predefinidos, presenta la posibilidad mediante lenguajes script de implementar otras funciones. Presenta varias herramientas disponibles y además da la posibilidad de desarrollar diferentes exámenes.(Villar 2007)

## 21 | Capítulo I. Fundamentación Teórica

*Authorware, presenta además algunas ventajas, tales como:*

- Elevado número de objetos, propiedades y funciones.
- Una herramienta valiosa y potente por su contenido.
- Control avanzado sobre la gestión de contenidos SCORM.
- Disponible tanto para Windows como para Macintosh.

*Algunas desventajas son:*

- La ayuda no se encuentra disponible en castellano.
- Interfaz compleja, es necesario precisar información antes.
- Necesita plugins propietarios para mostrar contenidos.(Denis y Varona 2010)

### **1.3.3 Reload**

Reload es una herramienta que permite la creación de contenidos educativos y que además posibilita el DI. Más específicamente, Reload crea y edita paquetes e inserta metadatos conforme a las especificaciones de ADL e IMS. Con esta herramienta es posible confeccionar y ver los paquetes en un navegador web. Utiliza el estándar SCORM y permite comunicación con un LMS.

Este editor tiene un valor significativo para la educación, pues permite a los autores de contenidos transmitir sus OA en un formato compatible con determinadas especificaciones.(Liber 2006)

*Reload Editor permite las siguientes funciones:*

## 22 | Capítulo I. Fundamentación Teórica

- Crear, importar, editar y exportar paquetes de contenidos.
- Empaquetar contenidos creados con otras herramientas.
- Darle un propósito nuevo a sus contenidos a través de la reorganización y recatalogación de los mismos.
- Preparar contenidos para almacenar en sitios destinados a tales efectos.
- Entregar contenido a usuarios finales usando la herramienta de guardado previo de contenidos.

Como se observa la única herramienta de las antes expuesta que implementa el DI en su contenido es Reload, para hacer más amigable el ambiente para la audiencia de hacia quién esté dirigido un recurso educativo determinado.(Denis y Varona 2010)

*Algunas desventajas detectadas en Reload:*

- Servicio Índice-Búsqueda

Según el apartado Índice-búsqueda de Servicio en IMS-LD, el “Tipo de Elemento” debe coincidir con el elemento seleccionado. Ejemplo de ello es que si se selecciona un elemento de tipo “Actividad de Aprendizaje” el campo “Tipo de Elemento” debe reflejar esta elección. En Reload se da la libertad de que el usuario defina el tipo de elemento de forma independiente y permite crear el ambiente aunque los mismos no coincidan.

- Importar XML

Reload permite importar archivos independientemente de que sean XML como especifica el estándar IMS – LD. Esta deficiencia trae consigo que el sistema incorpore archivos de formato incompatible con su funcionamiento y que lo almacene a su vez en su base de datos.

### 1.3.4 CompendiumLD

Antes de entrar en detalles sobre las características de CompendiumLD debemos hacer referencia a la herramienta a partir de la cual fue desarrollada: Compendium. Compendium es una herramienta de software que proporciona una interfaz visual flexible para gestionar las conexiones entre la información y las ideas.(Compendium Institute 2010)

En el sitio oficial *Compendium Institute* explican que esta herramienta es el resultado de 15 años de investigación en el campo de la inserción de hipertexto, modelado colaborativo, memoria organizacional, argumentación asistida por ordenador y facilitación de reuniones. Además aclaran que no están actualmente brindando soporte al software o a la documentación.

CompendiumLD es una herramienta de software para el diseño de actividades de aprendizaje utilizando una interfaz visual flexible. Proporciona un conjunto de íconos para representar los componentes de las actividades de aprendizaje. Estos íconos se pueden arrastrar y soltar, y se conectan para formar un mapa que representa una actividad de aprendizaje. CompendiumLD es una versión especializada del Compendium. El mapeo mental o software de la argumentación proporciona un conjunto predeterminado de íconos para la creación de mapas para visualizar las conexiones entre las ideas. Todas las facilidades proporcionadas por Compendium se incluyen dentro de CompendiumLD.(Compendium Institute 2010)

#### *Ventajas:*

- Puede ejecutarse sobre los sistemas operativos Windows, Mac OS y Linux.

## 24 | Capítulo I. Fundamentación Teórica

- Es muy sencillo de utilizar ya que sólo requiere de arrastrar los íconos que trae predefinidos y luego relacionarlos entre sí.
- Puede importar diferentes recursos electrónicos como: PDF, PPT, entre otros, así como XML generados por la propia herramienta.
- El código fuente de CompendiumLD está bajo la licencia GNU Lesser General Public (LGPL), versión 3.

### *Desventajas:*

- Apertura de recursos automáticamente en Mac OS. Para deshabilitar esta función es necesario editar un archivo de configuración manualmente.
- Es una aplicación java por lo que requiere de la instalación previa de Java Runtime Environment (JRE).
- La dependencia de java en los diferentes sistemas operativos para instalar el CompendiumLD puede causar conflictos que requieran la actualización de la versión de java e incluso los driver de las tarjetas de video en Windows.

### **1.3.5 MOT y MOT plus**

MOT es un software que permite construir una representación gráfica del conocimiento o "mapas mentales" en diferentes campos del conocimiento además de subrayar las relaciones entre las unidades del conocimiento. Posee además sofisticadas funcionalidades gráficas. MOT plus añade al editor MOT básicas poderosas capacidades multi-dominio y de edición por capas, así como funciones de etiquetado para facilitar el modelado del conocimiento por equipos. La última versión permite crear el diseño de



## 25 | Capítulo I. Fundamentación Teórica

aprendizaje de acuerdo a la especificación IMS – LD y exportarlo correctamente a dicho formato. También permite la construcción gráfica de ontologías de acuerdo a la norma de OWL-DL.

El editor de conocimiento MOT permite al usuario crear una representación visual de los conocimientos en diversos campos y resaltar gráficamente las relaciones del conocimiento. Una serie de íconos se utilizan para representar distintos tipos de conocimiento, es decir, conceptos, procedimientos, principios y hechos, así como los diferentes tipos de relaciones: la composición, la regulación, la especialización, la precedencia, la entrada/salida, e instanciación.(Technologies Cogigraph Inc 2008)

### *Ventajas:*

- Proporcionan la representación subyacente de los objetos. Mejor adaptado que las herramientas CASE a la ingeniería pedagógica.
- Fácil de aprender y usar.
- Capaz de producir modelos altamente integrados.
- Vincula la pedagogía y los materiales didácticos.
- Técnica de representación de amplio rango aplicada a todos los campos cognitivos.

### *Desventajas:*

- El software sólo está disponible en dos idiomas: inglés y francés, así como su documentación.
- Es una aplicación de escritorio.
- Es necesario pagar una licencia para su utilización.

### **IMS-LD™ Scenario Editors**

El editor MOT+LD incrustado en MOT plus fue el primer intento de ayudar a los diseñadores a construir gráficamente escenarios de modelos de aprendizaje compatibles con IMS – LD, la construcción de los mismos se centra en el método MISA. El modelo gráfico es validado y exportado como una instancia del esquema XML de IMS – LD. Este archivo XML puede ser leído en los editores de IMS – LD basados en formularios, como RELOAD, si es necesario añadir las condiciones del nivel B o las notificaciones del nivel C. El XML resultante puede ser ejecutado por los reproductores (players) de IMS – LD o plataformas para ofrecer sesiones de aprendizaje en línea a sus usuarios.

Este editor simplifica la edición gráfica de escenarios de DI, especialmente, la estructura de control del flujo de actividades lo que está cubierto en los niveles B y C de la especificación IMS – LD. Las propiedades y condiciones descritas en dichos niveles pueden ser incluidas en el modelo gráfico para alterar el flujo de actividades, notificar a un actor o presentar un recurso dependiendo de acciones previas almacenadas en un usuario, grupo de archivos o modelo.(Technologies Cogigraph Inc 2008)

### **1.4 Herramientas y tecnologías a utilizar.**

La realización del módulo que empleará la especificación IMS-LD hasta su nivel A requiere del empleo de una serie de herramientas, tecnologías y métodos de desarrollo. Del estudio realizado en el trabajo de diploma precedente a éste titulado: “Análisis y diseño del módulo IMS-Learning Design para CRODA” de los autores Yordania Márquez Denis y Eduardo Arguelles Varona se derivan una serie de herramientas, tecnologías y metodologías mencionadas anteriormente.

A continuación se relacionan los aspectos tratados en esta tesis precedente que son de interés para el proceso de implementación. Los mismos son empleados como base para el desarrollo del módulo de Diseño Instruccional. En los casos necesarios se realizan acotaciones así como la inclusión de referentes más actuales de la tecnología y las herramientas tratadas.

### 1.4.1 Metodología de desarrollo.

La metodología de desarrollo es un proceso complejo. La misma provee un marco de trabajo usado para reestructurar, planificar y controlar el proceso de desarrollo en sistemas de información.

Se entiende por metodología de desarrollo una colección de documentación formal referente a los procesos, las políticas y los procedimientos que intervienen en el desarrollo del software. En Ingles, software development methodology (SDM) o system development life cycle (SDLC).

La finalidad de una metodología de desarrollo es garantizar la eficacia (ej. cumplir los requisitos iniciales) y la eficiencia (ej. minimizar las pérdidas de tiempo) en el proceso de generación de software.(Cuaresma 2008)

#### 1.4.1.1 RUP.

La metodología que se utiliza para la investigación de este módulo es *Rational Unified Process* (RUP), habiendo concluido que es la más adecuada para el desarrollo y modelado de la herramienta, debido a que además utiliza el Unified Modeling Language (UML), como lenguaje de modelación.

RUP cuenta con numerosas características, pero dentro de ellas se pueden encontrar con que es dirigido por casos de uso, donde se representa a través de requerimientos lo que los usuarios futuros desean que se les dé como resultado. Como otra característica importante tenemos que es centrado en la arquitectura lo esto significa que los usuarios y trabajadores del proyecto han de estar de acuerdo en cómo se va a ver el sistema y además de cuales elementos se van a utilizar para lograr el objetivo principal del cliente. Y como una última características, iterativo e incremental.

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo. También está

## 28 | Capítulo I. Fundamentación Teórica

compuesto por fases cada una con sus objetivos principales, éstas son la fase de inicio, elaboración, construcción y transición.

Aun habiéndose mencionado todas estas características que hacen de RUP una metodología sobradamente probada no es suficiente para justificar su inclusión. Es por ello que tomo los argumentos de Daniel Perovich en su presentación “*Rational Unified Process*” (2008) para expresar por qué utilizar RUP:

- Provee un entorno de proceso de desarrollo configurable, basado en estándares.
- Permite tener claro y accesible el proceso de desarrollo que se sigue.
- Permite ser configurado a las necesidades de la organización y del proyecto.
- Provee a cada participante con la parte del proceso que le compete directamente, filtrando el resto.

Además de los beneficios inherentes a la utilización de esta metodología es necesario tener en cuenta una

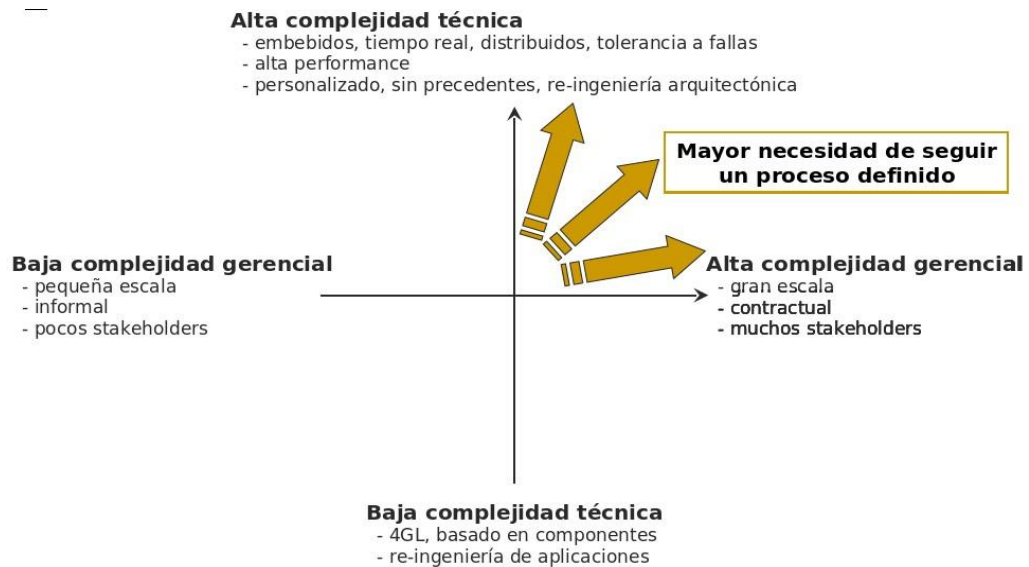


Figura 3: ¿Cuándo utilizar RUP? (Perovich 2008)

serie de necesidades que hagan factible su empleo ( *Ver Figura 3*).

Como se puede apreciar en los proyectos de alta complejidad en los que intervienen gran cantidad de personal son los más factibles para aplicar esta metodología ( *Ver Figura 3*). Mucho más si tenemos en cuenta la necesidad de seguir una guía bien definida y cumplir plazos estrictos de ejecución.

### 1.4.2 Lenguaje Unificado de Modelado (UML)

El lenguaje de modelado es un conjunto de símbolos y de maneras de disponerlos de forma estandarizada empleado para modelar parte de un diseño de software orientado a objetos. Como mejor representante de esta vertiente se encuentra el Lenguaje Unificado de Modelado (siglas en inglés, UML).

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software.

Debido a que UML es un lenguaje, cuenta con reglas para relacionar elementos gráficos que se combinan para formar el diagrama en sí. Este lenguaje de modelado no es una guía para realizar análisis y diseño orientado a objetos. En otras palabras es un lenguaje de modelado no un proceso. Éste por el contrario permite hacer modelación de un sistema con tecnología orientada a objetos.

El objetivo de los diagramas que se desarrollan con este lenguaje es presentar diversas perspectivas del sistema, que además se les conoce como modelos. Es importante destacar que un modelo UML describe lo que supuestamente hará un sistema, pero no dice, cómo implementar dicho sistema.

### 1.4.3 Visual Paradigm como herramienta CASE

Como herramienta UML se utilizó el Visual Paradigm. Ésta soporta todo el ciclo de vida del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Este software de

modelado ayuda a que sea más rápida la construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los diagramas de clases, código inverso, genera código y documentación desde los diagramas. Proporciona además diferentes tutoriales que sirven de ayuda y mejoran el entendimiento de dicha herramienta. Puede integrarse además con el IDE NetBeans.

Precisamente su capacidad de integración con el IDE de desarrollo Netbeans es una de las razones principales para seleccionar esta herramienta. Netbeans como se explicará más adelante es el IDE seleccionado para desarrollar la codificación del módulo de Diseño Instruccional. Otra de las características que hace del Visual Paradigm una atractiva propuesta es que es posible instalarlo en Linux con lo que la gama de herramientas de desarrollo y de documentación se encuentran agrupadas en el mismo sistema operativo.

### 1.4.4 Servidor web Apache

Los servidores web son grandes proveedores de información para todo tipo de usuarios. Estos surgieron con motivo de la necesidad que tenían o requerían algunas empresas de compartir información con un grupo de clientes. A raíz de eso un sistema gestor de bases de datos (SGBD), deberá ofrecer soluciones de una forma fiable, rentable y de alto nivel de rendimiento. Debe proporcionar además servicio de forma global, y en la medida de lo posible, independientemente de la plataforma en la que sea instalado.

El servidor HTTP Apache es el resultado del esfuerzo de varios desarrolladores que desde el año 1994 decidieron fusionar sus propias extensiones y correcciones de errores del “demonio” desarrollado por Rob McCool para el *National Center for Supercomputing Applications* en la Universidad de Illinois, EE.UU.

Entre sus características destacan:

## 31 | Capítulo I. Fundamentación Teórica

- Multiplataforma.
- Es un servidor web conforme al protocolo HTTP/1.1
- Apache es una tecnología gratuita de código fuente abierta. Esto le da una transparencia a este software de manera que si queremos ver que es lo que estamos instalando como servidor, lo podemos saber, sin ningún secreto, sin ninguna puerta trasera.
- Apache es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor web Apache. Actualmente existen muchos módulos para Apache que son adaptables a éste, y están ahí para que se instalen cuando sea necesario. Trabaja con gran cantidad de Perl, PHP y otros lenguajes de script. También trabaja con Java y páginas JSP.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto.
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.

A todas estas características además de ser el servidor web más utilizado a nivel internacional en estos momentos se suman otra serie de características que llaman la atención de los que apuestan por él como una solución segura y estable. Entre estas características adicionales se encuentran:(The Apache Software Foundation 2011)

- Alojamiento de archivos estáticos.
- Proporcionar equilibrio de carga.

- Conmutación por error (mod\_proxy).
- Seguridad y filtrado (mod\_security).
- Reescritura (mod\_rewrite).

### 1.4.5 Sistemas de gestión de bases de datos

Los sistemas de gestión de bases de datos (en inglés, *database management system*) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que lo utilizan. Su propósito general es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante para una organización. Entre los objetivos que deben cumplir se encuentran:

- **Abstracción de la información.** Los usuarios no deben conocer detalles acerca del almacenamiento físico de los datos.
- **Independencia.** Posibilidad de modificar el esquema tanto lógico como físico de la base de datos sin modificar las aplicaciones que hacen uso de ella.
- **Consistencia.** Actualización simultánea de información redundante y uso de condiciones.
- **Seguridad.** Diferentes categorías de permisos para usuarios o grupos de usuarios
- **Manejo de transacciones.** Simplificación de la modificación de datos.
- **Tiempo de respuesta.** Tiempo que toma que la información sea accedida o almacenada.



### 1.4.5.1 PostgreSQL

PostgreSQL es un sistema de gestión de base de datos objeto-relacional. Creado en la Universidad de California está bajo la licencia de BSD (Berkeley Software Distribution), donde por el contrario a la GPL, permite el uso del código fuente en software no libre. Presenta las 4 libertades del software libre, además se puede utilizar para fines comerciales de cualquier tipo, puede revenderse, etc.

Como otras características de este gestor se encuentran:(Quiñones A. 2008)

- La atomicidad, esta propiedad asegura que la operación se haya realizado o no, así se asegura que ante un fallo del sistema lo que se realizaba no queda a medias.
- Consistencia en la propiedad, esto asegura que las tareas una vez empezadas se finalicen correctamente y completas.
- Aislamiento en la propiedad, asegura que una operación no puede afectar a las otras.
- Durabilidad, asegura que una vez realizadas las operaciones, aunque exista fallas en el sistema éstas se mantienen persistentes.
- Se ejecuta en casi todos los sistemas operativos, tales como Linux, Unix, OS, Beos, Windows, etc.
- Diversidad de documentación, amplia, organizada, libre y además pública.
- Adaptable a las necesidades del cliente.

Todas estas cualidades hacen de PostgreSQL el gestor de base de datos relacional más efectivo para el desarrollo de aplicaciones en entornos libres. Mucho más si se tiene en cuenta su capacidad multiplataforma a la hora de alojarlo en un servidor determinado.

### 1.4.5.2 Exist-DB

En la actualidad existen un grupo de gestores de bases de datos para el almacenamiento de datos nativos y su gestión. No todos los gestores soportan y pueden trabajar con los mismos datos, es decir, existen gestores para datos que requieran un tratamiento especial.

Puesto que existe un incremento del uso de tales datos específicos, como son los XML, y de la gran utilidad de éstos, se ha hecho necesario el trabajo con respecto a ellos. Así, en los últimos tiempos, han aparecido un gran número de bases de datos que nos permiten almacenar documentos XML, entre otros.(Meier 2002)

Existen dos tipos de bases de datos XML que permiten el almacenamiento de estos datos:

- XML-Enabled, que mapean documentos XML en una base de dato relacional u objeto-relacional.
- XNDB, bases de datos nativas donde se utilizan como unidad funcional de almacenamiento estructuras específicas para documentos XML.

EXist-db es un sistema de gestión de base de datos que almacena XML de acuerdo con el modelo de datos XML, el mismo es de código abierto. Dentro de sus principales características están:

- Soporte para lenguajes de consultas XML como XQuery, XPath y XSLT.
- Es posible indexar documentos.
- Soporte para actualización de datos y para multitud de protocolos como SOAP, XML-RPC, WebDav y REST.

### 1.4.6 Framework de desarrollo Symfony

## 35 | Capítulo I. Fundamentación Teórica

Los frameworks definen en términos generales un grupo de conceptos estandarizados, prácticas y criterios para enfocar un determinado problema, por lo que sirven para enfrentar, basado en soluciones anteriores, nuevos problemas de índole similar a los enfrentados anteriormente.

Debido a esto en un framework pueden incluirse soportes de programa, bibliotecas y lenguaje interpretado. Además de poder proporcionar estructura al código fuente, simplificándolo y obligando al desarrollador a crear un código más legible. Con ello facilita la programación de aplicaciones y se encapsulan operaciones complejas en instrucciones sumamente sencillas, ayudándonos a simplificar mucho el trabajo.

Symfony está desarrollado completamente con PHP 5. Ha sido utilizado en numerosos proyectos reales y se utiliza en la actualidad en muchos sitios de gran éxito, tanto en la empresa del software como en el ámbito de los negocios, entre otros.

Existen una gran cantidad de características, las cuales proporcionan muchas facilidades. Dentro de éstas están:(Potencier y Zaninotto 2008)

- Facilidades a la hora de instalarlo en la mayoría de las plataformas.
- No importa el gestor de base de datos que se use. Es independiente de éste y es compatible con casi todos los gestores de bases de datos.
- Gran sencillez de uso y flexibilidad para su adaptación a grandes y complejos cambios.
- Los desarrolladores deben configurar sólo lo que no es convencional.
- Mantiene y permite hacer uso de las mejores prácticas y patrones de diseño web.

## 36 | Capítulo I. Fundamentación Teórica

- Es adaptable para aplicaciones de empresas, acoplándose a las propias características de éstas. Es además, lo suficientemente estable para desarrollar aplicaciones a largo plazo.
- Falibilidad en cuanto a lectura de código, incluyendo comentarios de phpDocumentor permitiendo su fácil mantenimiento.
- Puede ser fácil de extender, por lo que permite su integración con librerías y clases desarrolladas por otros.

Dentro de sus grandes ventajas se tiene que:

- Separa la lógica del negocio, la lógica del servidor y la presentación de la aplicación web.
- Automatiza tareas elementales, posibilitando así al desarrollador concentrarse en los problemas específicos y de mayor rigor.
- Contiene un gran número de clases y herramientas encaminadas a reducir el tiempo de elaboración de un producto.
- Su principal resultado es la reutilización de piezas de código, entre otros aspectos, por lo que cuando se crea un nuevo producto se le ha embebido gran parte de las mismas que propone el framework.

Todo ello conlleva a esta selección como base para desarrollar. Sus sobradas ventajas y probadas capacidades así lo ameritan. Ello sumado a las potencialidades visuales que confiere el empleo de Ext JS crea la amalgama perfecta entre este framework y la librería de javascript mencionada y explicada a continuación.

### 1.4.7 Librería de javascript Ext JS

En ciencias informáticas, una librería es un conjunto de subprogramas que se utilizan para desarrollar un software. Éstas contienen datos y fragmentos de códigos que desempeñan funcionalidades para programas. De ellas se puede hacer uso, para ahorrar tiempo y no reinventar lo desarrollado debido a que satisfacen determinadas necesidades de forma óptima, aunque también se puede modificar sus elementos o adaptar de acuerdo a las necesidades del desarrollador.

Ext JS es una librería de javascript para la creación de aplicaciones que favorecen en gran medida la comodidad para los clientes. En principio esta librería fue construida como una extensión de *Yahoo User Interface* (YUI), la cual es una de las mejores librerías de javascript para crear páginas web ricas en efectos visuales avanzados.

Con el tiempo, debido a su gran desempeño, componentes de usuario personalizables, buen diseño y documentación, se convirtió en un framework independiente. De esta forma, a principios de 2007 se creó una compañía para comercializar y dar soporte al framework Ext JS.

*Características:*

- Código reutilizable.
- Interoperabilidad con otros frameworks como JQuery, Prototype, YUI.
- Está orientada a la programación de interfaces de usuario de tipo desktop.
- Posee controles para campos de texto, áreas de texto; y controladores-selectores para campos de números, botón de selección, cuadro de selección, campos de números, fecha y otros. También

posibilita crear barras de herramientas y menús con estilo de aplicaciones de escritorio, diálogos modales y eventos.

- Posee un gran número de usuarios así como soporte comercial.

En un corto período de tiempo, Ext JS ha evolucionado de un práctico conjunto de extensiones a una biblioteca popular en lo que mucha gente cree es la más madura librería javascript para el desarrollo de interfaz de usuario disponible en la actualidad.

Ext JS se centra en la creación de interfaces de usuario en una aplicación web. De hecho, es mejor conocida por su colección de primera clase de *widgets* de interfaz de usuario. Permite crear aplicaciones web que imiten la apariencia de aplicaciones de escritorio nativas, así como también imita la mayor parte de las funcionalidades que estas aplicaciones proporcionan. En resumen, Ext JS le permite crear verdaderas aplicaciones *Rich Internet Applications (RIA)*.(Zammetti 2009)

La ventaja de las aplicaciones RIA radica en la combinación de las mejores características que ofrecen las aplicaciones web y las aplicaciones tradicionales. Normalmente en las aplicaciones web, hay una recarga continua de páginas cada vez que el usuario pulsa sobre un enlace lo que produce un tráfico muy alto entre el cliente y el servidor. En los entornos RIA, en cambio, desde el principio se carga toda la aplicación, y sólo se produce comunicación con el servidor cuando se necesitan datos externos.

### 1.4.8 IDE de desarrollo NetBeans

Un entorno de desarrollo integrado (en inglés, *integrated development environment [IDE]*) es un programa informático compuesto por un conjunto de herramientas de programación. Puede especializarse en un lenguaje de programación o en múltiples. En otras palabras es un entorno de desarrollo que ha sido empaquetado en un programa de aplicación.

## 39 | Capítulo I. Fundamentación Teórica

Netbeans es un entorno de desarrollo integrado ganador de múltiples premios disponible para los sistemas operativos Windows, Mac, Linux y Solaris. Es un IDE de código abierto y una plataforma de aplicación que permite a los desarrolladores crear de forma rápida tanto aplicaciones web como de escritorio.

A continuación se explican las características que hacen de NetBeans 6.9.1 integrado a PHP, el IDE seleccionado para desarrollar el módulo de Diseño Instruccional:

### 1. Creación de proyectos PHP.

NetBeans suministra una estructura para los proyectos que pueden ser creados junto a este IDE, propone un esqueleto para organizar el código fuente, el editor conjuntamente integra lenguajes como HTML, javascript y CSS. Además posee un sistema para examinar todos los directorios de cada proyecto, haciendo reconocimiento y carga de clases, métodos y objetos, para acelerar la programación.

### 2. Integración con Symfony.

La integración de NetBeans 6.9.1 con el popular framework de PHP, Symfony, agiliza mucho más el proceso de desarrollo de aplicaciones web. Ello posibilita prescindir de la consola de comandos ya que desde el IDE se pueden realizar todas las tareas rutinarias de trabajo con el Symfony.

### 3. Editor de código fuente.

Mejora en el editor de PHP. Es mucho más ágil y a la vez robusto, contiene más ayuda en línea, reconocimiento de sintaxis y todo lo referente la última versión de PHP: la 5.3.

### 4. Depuración de PHP.

NetBeans integra muy bien la utilización de Xdebug, gracias a esto, es posible inspeccionar y examinar cada variable local, establecer puntos de interrupción y evaluar la lógica del código. Ofrece además una línea de comandos de depuración lo que permite inspeccionar el código HTML generado sin tener que emplear un navegador web.

### 5. Integración con sistemas de control de versiones.

Puede integrarse con sistemas de control de versiones, tales como: SVN, CVS, Mercurial y Git. Desde el editor es posible realizar la administración de estos sistemas de versionado, como por ejemplo sus funciones: commit, branch, importar, exportar, revert, clonar, entre otras.

## 1.4.9 Lenguajes de programación y etiquetado

Actualmente existen diferentes lenguajes de programación para desarrollar en la web. Éstos han ido surgiendo debido a las tendencias y necesidades de las plataformas. Desde los inicios de Internet, los usuarios tenían diferentes demandas y se les dio solución mediante lenguajes estáticos. A medida que transcurría el tiempo las tecnologías fueron desarrollándose y surgieron nuevos problemas a solucionar. Esta situación originó lenguajes de programación dinámicos para la web que permitieran interactuar con los usuarios y utilizaran sistemas de bases de datos.

El concepto de lenguaje de etiquetado o de marcas fue extrapolado de la antigua práctica de marcar los manuscritos con instrucciones de impresión en los bordes y llevado al campo de la informática. La novedad de este lenguaje consiste en la separación de la presentación de la estructura del texto.

### 1.4.9.1 PHP



## 41 | Capítulo I. Fundamentación Teórica

Acrónimo de PHP: Hypertext Preprocessor. Es un lenguaje interpretado de alto nivel embebido en páginas HTML. Se instala en el servidor y funciona con versiones de Apache, Microsoft IIS, Netscape Enterprise Server y otros.

La forma de usar PHP es insertando código PHP dentro del código HTML de un sitio web. Cuando un cliente visita la página web que contiene éste código, el servidor lo ejecuta y el cliente sólo recibe el resultado. Es por ello que este lenguaje se clasifica como “del lado del servidor” pues su ejecución es enteramente sobre esta plataforma.

Su ejecución, es por tanto en el servidor, a diferencia de otros lenguajes de programación que se ejecutan en el navegador. PHP permite la conexión a numerosas bases de datos, incluyendo MySQL, Oracle, ODBC, etc. Siendo compatible con bases de datos PostgreSQL que fue la selección anterior. Puede ser ejecutado en la mayoría de los sistemas operativos (Windows, Mac OS, Linux, Unix).

El framework de desarrollo seleccionado, Symfony, está desarrollado en este lenguaje del lado del servidor. Es por ello que su selección se sustenta en la necesidad de desarrollar sobre Symfony, un framework de PHP. A esto se suma su probada efectividad en soluciones web y su continuo desarrollo, soporte y comunidad internacional. Mencionar además la abundante bibliografía que existe en la web sobre el mismo.

### 1.4.9.2 Javascript

Javascript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como aparición y desaparición de texto, animaciones, acciones que se activan al pulsar botones u otros elementos 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 el mismo se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.(Eguíluz Pérez 2007)

La importancia del empleo de javascript radica en la posibilidad de interacción con la aplicación por los usuarios de la misma. Es un lenguaje sencillo presente en todos los navegadores y que además tienen en cuenta muchos organismos de normalización internacionales. Su función principal es la de mejorar la gestión de la interfaz cliente/servidor.

Este lenguaje interpretado es también la base de la librería Ext JS descrita anteriormente y por tanto parte de la columna vertebral de la aplicación. Es por ello que su selección es lo más aconsejable para la adaptación y mejoras en la librería seleccionada.

### 1.4.9.3 XML

XML, *Extensible Markup Language* (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Tiene sus inicios en 1996 y vio su primera versión el 10 de febrero de 1998.

Puesto a que es una adaptación más simplificada del SGML, (en inglés, *Standard Generalized Markup Language* [lenguaje de marcado generalizado]) permite definir una gramática de lenguajes específicos, como mismo HTML es un lenguaje definido por SGML. Por lo que no es un lenguaje en específico, sino una manera de definir lenguajes que cubran distintas necesidades.

Es un conjunto de reglas que se usan para definir etiquetas semánticas las cuales organizan un documento en diferentes partes. Siendo así un meta lenguaje que define sintaxis para definir otros lenguajes etiquetados estructurados.(Bray 2008)

## 43 | Capítulo I. Fundamentación Teórica

Su principal característica es que no posee etiquetas definidas previamente, por lo que el propio autor las define como desee. Además de poseer datos compuestos de múltiples aplicaciones.

La extensibilidad y flexibilidad de este lenguaje permite agrupar una variedad amplia de aplicaciones, desde páginas web hasta bases de datos. Posibilita la gestión y manipulación de los datos desde el propio cliente web. Por otro lado es la forma en que se expresa el manifiesto de la especificación IMS – LD por lo que su empleo es imprescindible junto el gestor de base de datos Exist-DB.

### **1.5 Conclusiones del capítulo.**

Tras el minucioso análisis de las herramientas existentes en el campo del e-learning con las facilidades de gestionar el DI se concluye que incluso las de mayor calidad presentan deficiencias. Algunas son difíciles de encontrar. Sin embargo, otras son notablemente evidentes y afectan la propia efectividad de realizar un tratamiento correcto de la especificación IMS – LD.

Se decide implementar el módulo de Diseño Instruccional en la herramienta de autor web CRODA por ser un producto desarrollado en el propio centro FORTES de la UCI. Esta herramienta cuenta además con los roles y sistemas de tratamiento de OA necesarios para la gestión del DI que se pretende desarrollar.

Teniendo en cuenta la selección precedente de las tecnologías y herramientas necesarias para producir un producto de calidad probada y que cuente con las características de IMS – LD se hace evidente que un aspecto importante es la garantía que de casi la totalidad de ellas cuentan con las libertades del software libre. Tienen además el aval de su utilización en proyectos de gran envergadura y solidez. Cuentan además con una amplia documentación así como una robusta comunidad de desarrollo.

## Capítulo 2. Implementación del módulo de Diseño Instruccional.

### Introducción

En este capítulo se describe inicialmente la estructura del sistema y cómo se agrupan sus elementos fundamentales. Para esta tarea se emplearán los diagramas de paquetes puesto que su representación es más general. Además se hará énfasis en las funcionalidades desarrolladas de comportamiento más complejo. Para la descripción de las mismas se utilizan diagramas de componentes que evidencien las relaciones entre sus script y archivos php.

### 2.1 Modelo de implementación

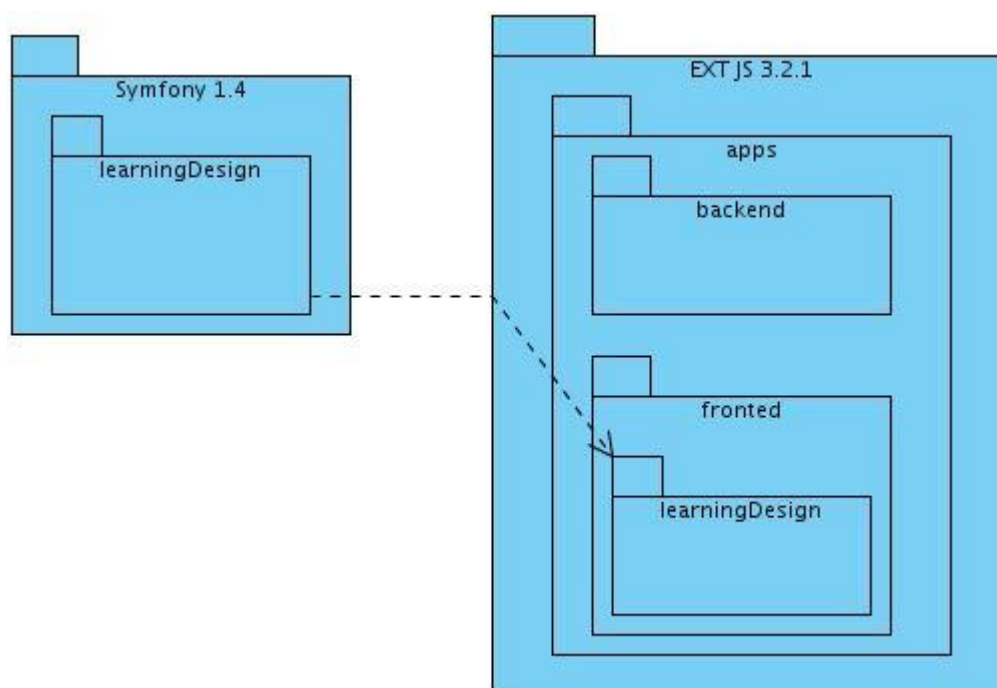
El modelo de implementación describe cómo los elementos del modelo de diseño se implementan en términos de componentes. Describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados y cómo dependen los componentes unos de otros.(Acuña 2009)

### 2.2 Diagrama de paquetes

En la figura se observa un diagrama de paquetes general que esboza los dos grandes conjuntos lógicos en los que se divide el sistema: la parte referente el framework de desarrollo Symfony en su versión 1.4 y la librería de javascript EXT JS 3.2.1 (*Ver Figura 4*)

## 45 | Capítulo 2. Implementación del módulo de Diseño Instruccional.

Como parte de la estructura estandarizada que se desea implementar en CRODA 2.0 se decidió extrapolar la forma de estructurar los archivos en las versiones más recientes de Symfony y aplicarlo a la



*Figura 4: Diagrama General del Sistema.*

estructura de la librería javascript Ext JS.

Para lograr dicho propósito fue necesario crear nuevos scripts que aglutinarán el resto de las dependencias de la librería siguiendo el patrón del framework. En las carpetas apps, extjs y core se agruparon los archivos siendo esta última la contenedora de los scripts: core.js y module.js.

Los scripts mencionados anteriormente son los encargados de hacer funcionar las librerías del Ext JS como si se tratase del propio Symfony sobre sus diferentes módulos. Con esta forma de organización fue

posible tratar los conjuntos de scripts referentes a un módulo en el Symfony como módulos independientes dentro del Ext JS. Los mismos se encuentran dentro de la carpeta apps.

Es por ello que el conjunto de funciones scripts relacionadas con el módulo de Diseño Instruccional se encuentran dentro de esta localización en la carpeta learningDesign de forma ordenada y siguiendo patrones de prefijos para denotar la funcionalidad de los mismos.

### 2.3 Diagramas de Componentes

El flujo de información en la aplicación web comienza en el *index.php* del framework Symfony. El mismo redirecciona sobre el *actions.class.php* del **módulo main** (Ver Figura 5). Este *actions* tiene asociado una plantilla (*template*) que es la que a través de un esquema (*layout*) define los comportamientos comunes a todas las páginas del sistema. Este procedimiento se denomina Decorar. (Potencier y Zaninotto 2008)

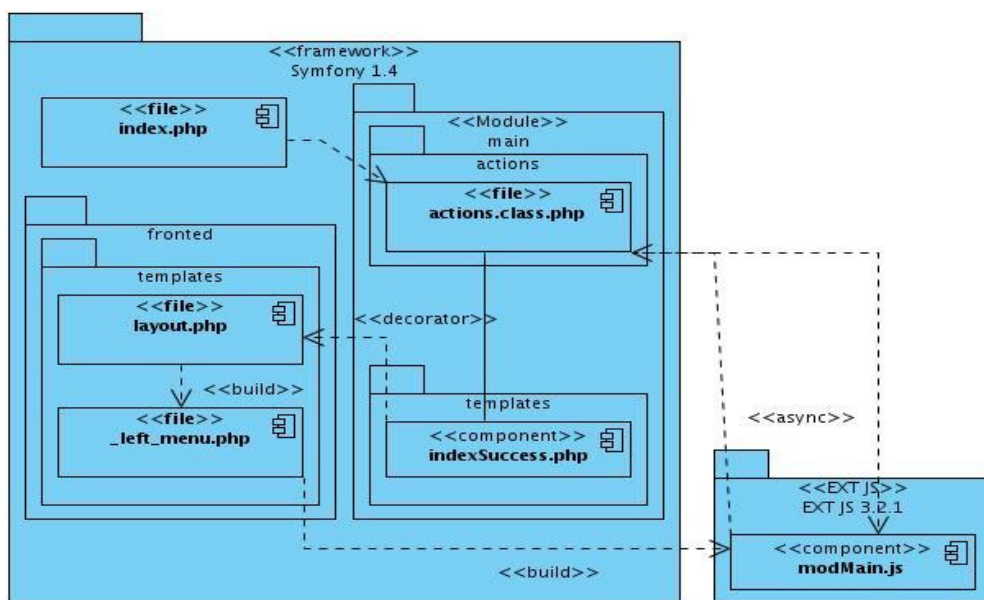


Figura 5: Diagrama de Componentes General Symfony 1.4 integrados al EXT JS 3.2.1.

## 47 | Capítulo 2. Implementación del módulo de Diseño Instruccional.

El *layout.php* por su parte es el encargado de dar paso a un parcial (*partial*) denominado: *\_left\_menu.php*. Este parcial se ocupa del lateral izquierdo de la página de inicio y de hacer la petición al script *modMain.js* de la librería Ext JS, el cual maneja el área de los componentes visuales en dicha librería (*fronted*).

En el *modMain.js* se encuentra la función *buildTreeCreation* que al ser invocada envía una petición asincrónica (*async*) al *action* del **módulo main** y recibe lo que debe graficar en el árbol de opciones que despliega.

A partir del *script modMain* se interrelacionan una serie de ficheros que son el núcleo del funcionamiento del **módulo learningDesign**. Los mismos son: *modLearningDesign.js*, *cmpWPGeneral.js*, *Model.js*, *cmpWPMainOA.js* y *cmpWPMainEditar.js* (Ver Figura 6).

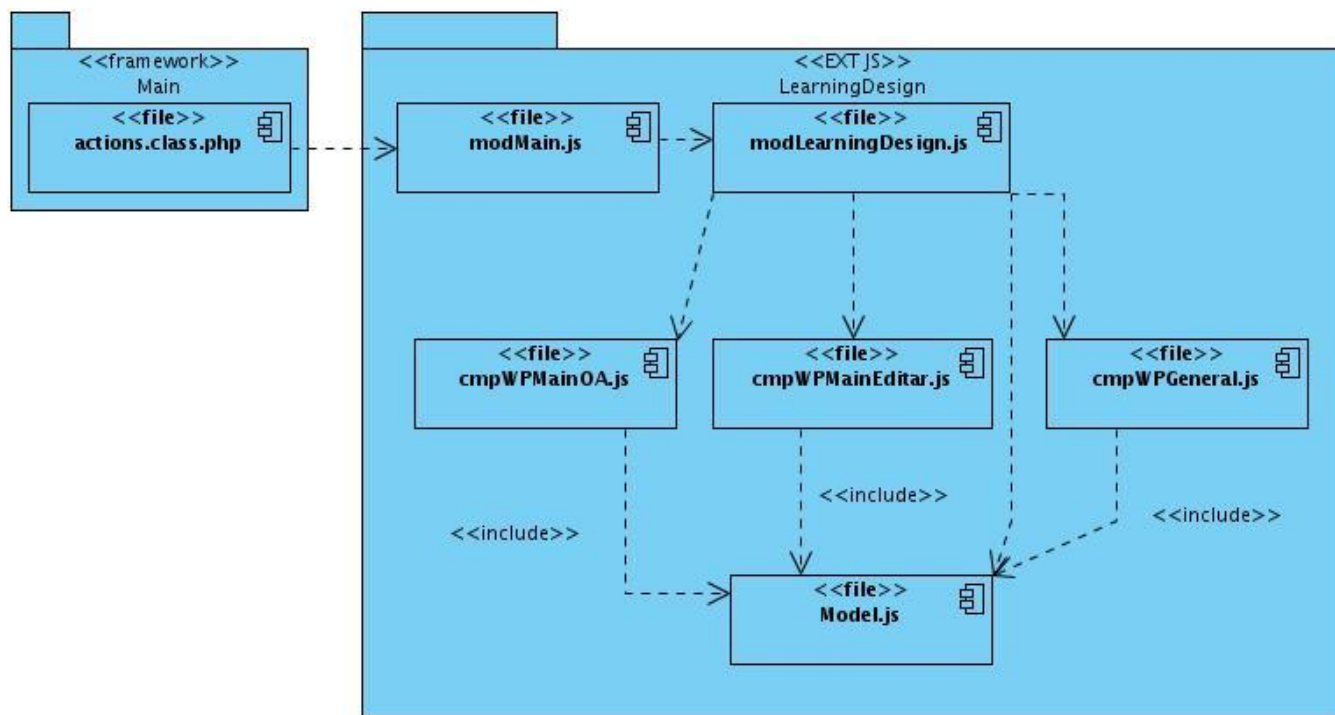


Figura 6: Diagrama de Componentes General del Módulo LearningDesign.

Una instancia de *modLearningDesign* es invocada mediante el *modMain.js*. El primero toma el control del área de trabajo en su totalidad suprimiendo los bloques laterales. Crea además, una instancia del objeto *Main()* que será el encargado de mantener los datos asequibles durante toda la ejecución del sistema.

El script *cmpWPGeneral.js* ejecuta casi la totalidad de las funciones incluidas en *Model.js* que son las encargadas de crear la jerarquía de elementos que conforman la especificación IMS-LD. Los archivos *cmpWPMainOA.js* y *cmpWPMainEditor.js* reciben en forma de parámetro una instancia de *Main()*. De esta



## 49 | Capítulo 2. Implementación del módulo de Diseño Instruccional.

forma se garantiza la permanencia de los datos que serán manejados posteriormente antes de ser almacenados en la base de datos XML.

Este comportamiento es el que se describía en el Capítulo 1 cuando se explicaba la selección de la librería Ext JS para lograr la implementación de una aplicación de tipo RIA. Estos *scripts* son los que garantizan que el sistema funcione con autonomía en la mayoría de sus funciones. Al no depender de conexiones a las bases de datos ni PostgreSQL ni XML puede gestionar el DI en tiempo real.

La **vista principal** de la aplicación se crea a través de los siguientes *scripts*: *cmpLDTree.js*, *cmpTPanel\_LD.js*, *Model.js* y *cmpWPGeneral.js*. Este último es el que invoca al resto de los *scripts* e inicializa las diferentes estructuras para ir completando el DI mediante las funciones contenidas en *Model.js* (Ver Figura 7).

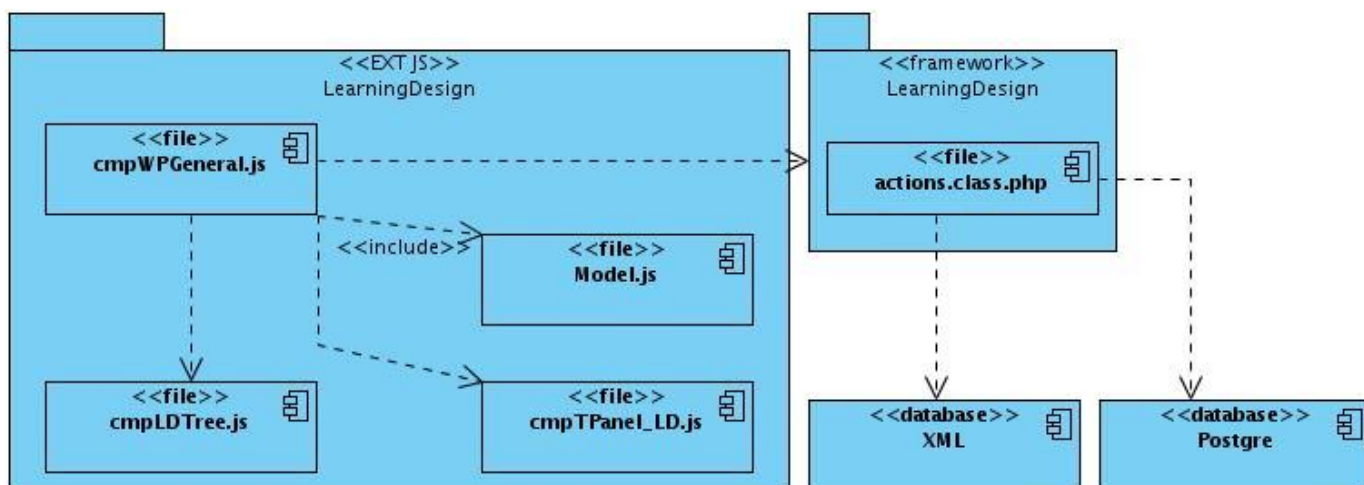


Figura 7: Diagrama de Componentes. Vista Principal.

A través de `actions.class.php`, `cmpWPGeneral.js` se comunica con las bases de datos, tanto la postgresQL como la XML. El mecanismo empleado se explicará detalladamente a continuación.

El componente `cmpLDTree.js` es el encargado de renderizar el “árbol” de opciones que aparece a la izquierda de la aplicación desde el cual se mejora la navegabilidad entre los diferentes componentes del diseño. Esta es una desventaja de algunas de las herramientas de autor descritas en el Capítulo 1.

Por su parte `cmpTPanel_LD.js` agrupa los visuales del resto de los componentes de la especificación IMS – LD. Su función consiste en proveer un sistema de pestañas donde se alojan los demás componentes.

### **Funcionalidades generales**

- Exportar

## 51 | Capítulo 2. Implementación del módulo de Diseño Instruccional.

Partiendo de *cmpWPGeneral.js* se hace una petición de tipo POST al *actions.class.php* del **módulo LearningDesign** para que almacene toda la información proporcionada por el diseñador instruccional en la base de datos XML (Ver Figura 8).

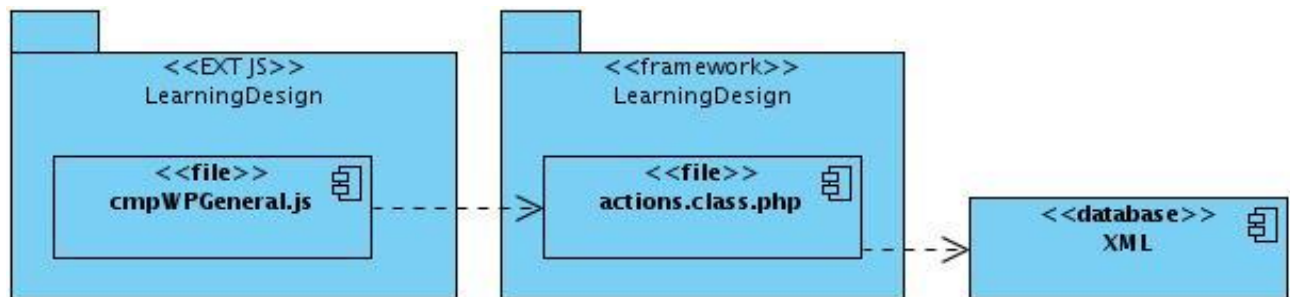


Figura 8: Diagrama de Componentes. Funcionalidad Exportar.

Esta petición está dirigida a la función privada *ExistDb()* que es la encargada de establecer los parámetros necesarios a almacenar en el XML. La misma se apoya en la función auxiliar *QueryExistDb()* para ejecutar la consulta a la base de datos. De esta forma los datos manejados en la aplicación quedan registrados en el manifiesto correspondiente.

- **Guardar**

Con el *script cmpWpGeneral.js* como base se ejecuta una petición POST con destino a la función *SaveDesign()* del *actions.class.php* del **módulo LearningDesign** (Ver Figura 9).

Primeramente se decodifica el envío y posteriormente se procede a extraer de forma individual los valores

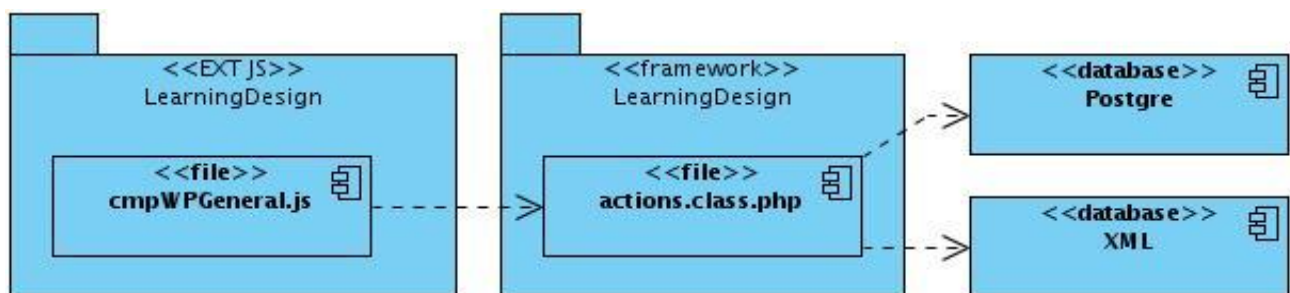


Figura 9: Diagrama de Componentes. Funcionalidad Guardar.

correspondientes al diseño que se desea almacenar.

Con una serie de consultas ejecutadas en la base de datos postgresQL se establece una correlación entre el manifiesto que se almacenará a continuación en la base de datos XML y la constancia de este procedimiento en las tablas del modelo relacional. En otras palabras, cada vez que sea necesario almacenar un nuevo manifiesto resultante del diseño de un diseñador instruccional debe quedar una constancia en la base de datos tradicional (postgresQL) de este proceso para próximas consultas.

## 2.4 Modelo de Datos

En CRODA los datos se almacenan en dos bases de datos diferentes. Una de ellas es la base de datos postgresQL muy similar a la SQL tradicional pero con la diferencia de que esta no es propietaria. La otra es una base de datos nativos (XML).

### 2.4.1 Modelo de base de datos relacional

La base de datos postgresQL de CRODA cuenta con 22 tablas de las cuales el módulo de Diseño Instruccional emplea tan sólo tres. Éstas son: *tb\_pif*, *tb\_learning\_design* y *sf\_guard\_user* (Ver Figura 10).

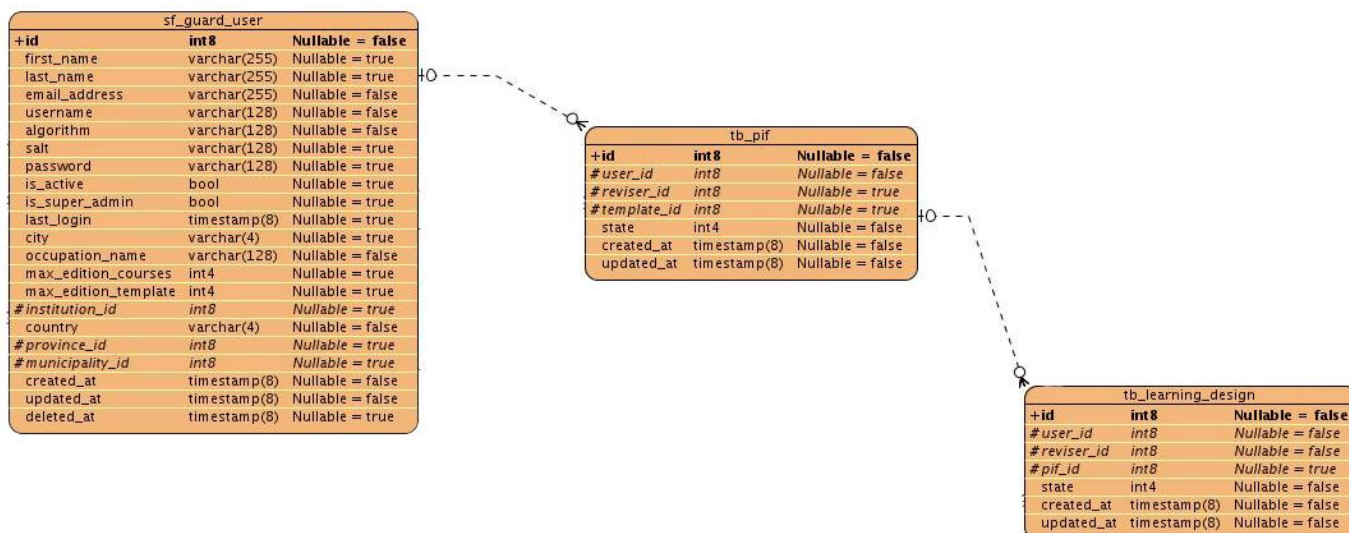


Figura 10: Base de datos postgresQL. Tablas.

Las tablas *tb\_pif* y *tb\_learning\_design* cuentan con una estructura similar. Una llave primaria que es el *id*. La diferencia radica en algunos atributos como son el id de la plantilla (*id\_template*) y el id del pif (*id\_pif*).

### 2.4.1.1 Descripción de las tablas de la base de datos postgresQL

<b>Nombre:</b>	sf_Guard_User.
<b>Descripción:</b>	Tabla referente a los usuarios de la aplicación, la cual contiene los datos generales y específicos de los mismos, nombre de usuario, contraseña, nombre y apellidos, rol que ocupará en la aplicación ( creador, generador ), el

54 | Capítulo 2. Implementación del módulo de Diseño Instruccional.

	tipo de acceso a la herramienta, a través de LDAP o manual así como el identificador del propio usuario.	
<b>Atributo:</b>	<b>Tipo:</b>	<b>Descripción:</b>
Id.	Integer.	Identificador del usuario.
City.	Varchar.	Cuidad de origen.
occupation_name.	Varchar.	Identificador del rol.
max_edition_courses.	Integer.	Máximo de cursos en edición.
max_edition_template.	Integer.	Máximo de plantillas en edición.
institution_id.	Integer.	Identificador de la institución.
Country.	Integer.	País.
province_id.	Integer.	Identificador de la provincia.
municipality_id.	Integer.	Identificador del municipio.
Image.	Varchar.	Imagen.

<b>Nombre:</b>	tb_pif.	
<b>Descripción:</b>	Tabla referente a los OA generados en la aplicación, la cual contiene los datos generales y específicos de los mismos, nombre del OA, autor, fecha de creado, fecha de culminación, estado del OA, así como el identificador del mismo.	
<b>Atributo:</b>	<b>Tipo:</b>	<b>Descripción:</b>

55 | Capítulo 2. Implementación del módulo de Diseño Instruccional.

Id.	Integer.	Identificador.
user_id.	Integer.	Identificador del usuario.
reviser_id.	Integer.	Identificador del revisor.
template_id.	Integer.	Identificador de la plantilla.
State.	Integer.	Estado.
Type_standard_id.	Integer.	Identificador del tipo de estándar.
Created_at.	Timestamp.	Fecha de creación.
Updated_at.	Timestamp.	Fecha de actualización.

<b>Nombre:</b>	tb_learning_design.	
<b>Descripción:</b>	Tabla referente a los DI generados en el módulo de Diseño Instruccional, la cual contiene los datos generales de los mismos, id del revisor, del usuario, fecha de creación y de actualización, estado del DI, id del pif contenedor, así como el identificador del mismo.	
<b>Atributo:</b>	<b>Tipo:</b>	<b>Descripción:</b>
Id.	Integer.	Identificador.
user_id.	Integer.	Identificador del usuario.
reviser_id.	Integer.	Identificador del revisor.
pif_id.	Integer.	Identificador del pif.
State.	Integer.	Estado.

Created_at.	Timestamp.	Fecha de creación.
Updated_at.	Timestamp.	Fecha de actualización.

## 2.4.2 Modelo de datos XML.

La diferencia fundamental de una base de datos nativa XML en comparación con una relacional es que la primera define un modelo lógico en comparación con el relacional de la última.

Los XML que se encuentran estructurados pero los datos específicos aún no han sido almacenados en ellos son denominados “manifiesto” en estos sistemas. Constituyen la declaración de la estructura lógica que guiará el almacenamiento de información etiquetada en ellos.

Los datos XML son almacenados en colecciones para su correcto indexado y ganar en organicidad. La colección más general que engloba al resto se denomina *croda*. En ella se almacenan los *pifs* que coinciden con la tabla *tb\_pifs* de la base de datos postgresSQL para alcanzar la correlación que se explicaba en el capítulo anterior (Ver Figura 11). Dentro de *pifs* se almacena el manifiesto de la especificación IMS – LD. El mismo se construye según la siguiente estructura: *imsmanifest <id del usuario>.xml*.





Figura 11: Esquema de las colecciones de la base de datos nativa Exist DB.

## 2.5 Diagrama de despliegue

El diagrama de despliegue del módulo de Diseño Instruccional que se muestra a continuación (Ver Figura 12) tiene la característica de especificar dos tipos de bases de datos independientes. Una relacional y una nativa para albergar el manifiesto una vez concluida su elaboración en el módulo de Diseño Instruccional.

Esta especificación es importante para comprender las necesidades del sistema implementado. La base de datos relacional (postgresql) es la encargada de mantener el control de los recursos creados en el sistema mientras que la de datos XML es la responsable del almacenamiento de los manifiestos que son la constancia del DI creado a partir del módulo mencionado (Ver Figura 12).

El usuario accede al sistema a través de una PC Cliente con la cual establece una conexión mediante el protocolo HTTP con el servidor web Apache. El servidor web además, se comunica con un servidor de base de datos PostgreSQL haciendo uso del protocolo ADO para registrar las referencias principales de los paquetes educativos creados. Por otro lado el servidor web también establece una conexión con el servidor nativo de XML (Exist DB) aprovechando el protocolo SOAP para almacenar los manifiestos de los DI generados por la herramienta.

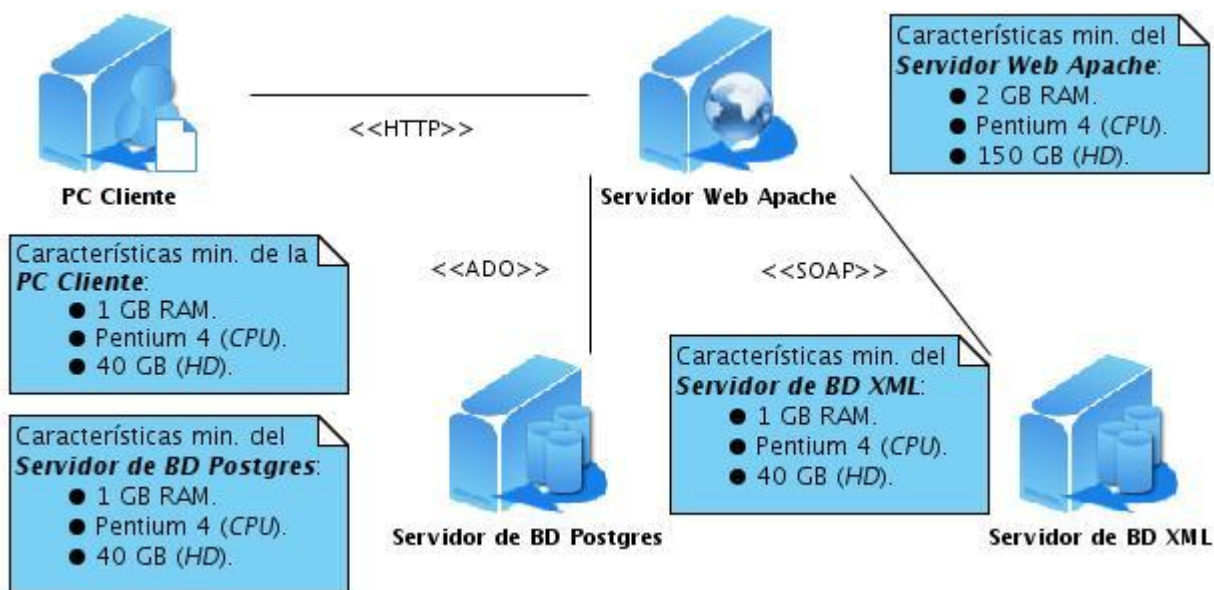


Figura 12: Diagrama de despliegue del módulo de Diseño Instruccional.

## 2.6 Conclusiones del capítulo.

Tras el completamiento del proceso de implementación se pudo comprobar la dificultad real de desarrollar una aplicación web capaz de gestionar el DI. Se logró integrar de forma satisfactoria además el módulo de Diseño Instruccional desarrollado utilizando la librería javascript Ext JS al sistema de funcionamiento de CRODA alojado en el framework Symfony.

En el desarrollo del módulo de Diseño Instruccional se puso en práctica de forma satisfactoria la adaptación de la estructura de funcionamiento de Symfony adaptada al sistema de funcionamiento de la librería Ext JS. Se generaron además los diferentes artefactos y modelos necesarios para describir el comportamiento de la herramienta y su gestión de las bases de datos, tanto postgresQL como XML.

## Capítulo 3. Pruebas del módulo de Diseño Instruccional.

### Introducción

Las pruebas de software son una técnica clásica para la verificación y validación del software, junto a otras alternativas de evaluación como revisiones, auditorías, métricas, etc.(Fernández, Lara, y Gutiérrez 2003) A los efectos de este trabajo se asume las pruebas de software como “la ejecución de un programa con el fin de detectar defectos”.(Myers 1979)

En la ejecución de las pruebas lo que se persigue es la detección de discrepancias entre la salida obtenida y la esperada. Dicha situación se interpreta entonces como un síntoma de problemas con el software. Los criterios de selección (denominados criterios de cobertura) suelen pretender la cobertura de un porcentaje o una representación del universo posible de elementos estructurales o de situaciones funcionales.(Fernández, Lara, y Gutiérrez 2003) De hecho, algunos autores hablan más de selección de casos de pruebas que de diseño de los mismos.(Choi y Mayrhauser 2000)

En el presente capítulo se describe el proceso de pruebas de Caja Negra para validar el funcionamiento de las interfaces del sistema implementado. Con las mismas se pretende detectar fallas y errores de manejo de datos en las interfaces de la aplicación web.

### 3.1 Prueba de Caja Negra

Las pruebas de Caja Negra que también puede se conocen como: pruebas de caja opaca, pruebas funcionales, pruebas de entrada/salida o pruebas inducidas por los datos.

## 60 | Capítulo 3. Pruebas del módulo de Diseño Instruccional.

Las pruebas de caja negra se centran en lo que se espera de un módulo, es decir, intentan encontrar casos en que el módulo no se atiene a su especificación. Por ello se denominan pruebas funcionales, y el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro. Las pruebas de caja negra están especialmente indicadas en aquellos módulos que van a ser interfaz con el usuario.(Mañas 1994)

Esta recomendación de Mañas no es definitiva puesto que más adelante expresa que estas pruebas pueden ser aplicadas además a otros módulos del sistema sin que los mismos hayan sido creados con la finalidad de desempeñarse como interfaces.

### 3.1.1 Caso de Prueba

#### **Gestionar Aspectos Generales del Diseño de Aprendizaje.**

##### *Descripción general.*

El CU inicia cuando el diseñador instruccional decide crear, modificar o eliminar aspectos generales. Finaliza cuando la acción sobre el diseño instruccional es realizada.

##### *Condiciones de ejecución.*

El usuario debe estar autenticado en la aplicación.

##### *Secciones.*

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad
SC 1: Mostrar Aspectos Generales del	EC 1.1: Mostrar datos.	Se muestra una serie de campos con los datos generales del diseño.

61 | Capítulo 3. Pruebas del módulo de Diseño Instruccional.

DI.		
SC 2: Crear Aspectos Generales del DI.	EC 2.1: Adicionar objetivos.	Se selecciona la opción “Adicionar” objetivos y se muestra el formulario correspondiente, con los campos que son necesarios.
	EC 2.2: Ingresar datos de un objetivo.	Se ingresan los datos necesarios y se presiona el botón “Aceptar”. Se adiciona el objetivo y se regresa a EC 1.1.
	EC 2.3: Cancelar datos de un objetivo.	Se presiona el botón “Cancelar”, no se guardan los cambios y se regresa a EC 2.2.
	EC 2.4: Cerrar ventana “Objetivos”.	Se presiona el botón “Cerrar”, no se guardan los cambios y se regresa a EC 1.1.
	EC 2.5: Adicionar objetivos dependientes.	Se selecciona la opción “Adicionar Dependencia” objetivo y se muestra el formulario correspondiente, con los campos que son necesarios.
	EC 2.6: Ingresar datos de un objetivo dependiente.	Se ingresan los datos necesarios y se presiona el botón “Aceptar”. Se adiciona el objetivo y se regresa a EC 1.1.
	EC 2.7: Cancelar datos de un objetivo dependiente.	Se presiona el botón “Cancelar”, no se guardan los cambios y se regresa a EC 2.2.
	EC 2.8: Cerrar ventana “Objetivos” dependientes.	Se presiona el botón “Cerrar”, no se guardan los cambios y se regresa a EC 1.1.
	EC 2.9: Modificar datos de un objetivo dependiente no	Se selecciona la opción “Adicionar Dependencia” sin haber seleccionado previamente un objetivo dependiente. Se

seleccionado.	muestra un mensaje indicando que debe seleccionar al menos un objetivo dependiente.
EC 2.10: Adicionar prerrequisitos.	Se selecciona la opción “Adicionar” prerrequisitos. Se muestra el formulario correspondiente, con los campos que son necesarios.
EC 2.11: Ingresar datos de un prerrequisito.	Se ingresan los datos necesarios y se presiona el botón “Aceptar”. Se adiciona el objetivo y se regresa a EC 1.1.
EC 2.12: Cancelar datos de un prerrequisito.	Se presiona el botón “Cancelar”, no se guardan los cambios y se regresa a EC 2.6.
EC 2.13: Cerrar ventana “Prerrequisitos”.	Se presiona el botón “Cerrar”, no se guardan los cambios y se regresa a EC 1.1.
EC 2.14: Adicionar prerrequisito dependientes.	Se selecciona la opción “Adicionar Dependencia” prerrequisito y se muestra el formulario correspondiente, con los campos que son necesarios.
EC 2.15: Ingresar datos de un prerrequisito dependiente.	Se ingresan los datos necesarios y se presiona el botón “Aceptar”. Se adiciona el prerrequisito y se regresa a EC 1.1.
EC 2.16: Cancelar datos de un prerrequisito dependiente.	Se presiona el botón “Cancelar”, no se guardan los cambios y se regresa a EC 2.2.
EC 2.17: Cerrar ventana “Prerrequisitos” dependientes.	Se presiona el botón “Cerrar”, no se guardan los cambios y se regresa a EC 1.1.

63 | Capítulo 3. Pruebas del módulo de Diseño Instruccional.

	EC 2.18: Modificar datos de un prerrequisito dependiente no seleccionado.	Se selecciona la opción “Adicionar Dependencia” sin haber seleccionado previamente un prerrequisito dependiente. Se muestra un mensaje indicando que debe seleccionar al menos un prerrequisito dependiente.
SC 3: Modificar Aspectos Generales del DI.	EC 3.1: Modificar datos de un objetivo.	Se selecciona la pestaña “Objetivos” y esta muestra todos los objetivos planteados para este diseño. Se selecciona la opción “Editar” y se muestra el formulario para efectuar la modificación del objetivo.
	EC 3.2: Confirmar modificación.	Se realizan los cambios necesarios en el objetivo. Se envía un mensaje de confirmación indicando si realmente se desea realizar los cambios.
	EC 3.3: Confirmar mensaje de modificación.	Se acepta el mensaje de confirmación, se guardan las modificaciones hechas al objetivo y se regresa a EC 1.1.
	EC 3.4: Denegar mensaje de modificación.	No se acepta el mensaje de confirmación, por lo que no se modifica el objetivo.
	EC 3.5: Cerrar ventana “Objetivo”.	Se presiona el botón “Cerrar”, no se guardan los cambios y se regresa a EC 1.1.
	EC 3.6: Modificar datos de un objetivo no seleccionado.	Se selecciona la opción “Editar” sin haber seleccionado previamente un objetivo. Se muestra un mensaje indicando que debe seleccionar al menos un objetivo para que pueda ser editado.
	EC 3.7: Modificar datos de un prerrequisito.	Se selecciona la pestaña “Prerrequisitos” y esta muestra todos los prerrequisitos planteados para este diseño. Se

64 | Capítulo 3. Pruebas del módulo de Diseño Instruccional.

		selecciona la opción “Editar” y se muestra el formulario para efectuar la modificación del prerrequisito.
	EC 3.8: Confirmar modificación.	Se realizan los cambios necesarios en el prerrequisito. Se envía un mensaje de confirmación si realmente se desea realizar los cambios.
	EC 3.9: Confirmar mensaje de modificación.	Se acepta el mensaje de confirmación, se guardan las modificaciones hechas al prerrequisito y se regresa a EC 1.1.
	EC 3.10: Denegar mensaje de modificación.	No se acepta el mensaje de confirmación, por lo que no se modifica el prerrequisito y se regresa a EC 1.1.
	EC 3.11: Cerrar ventana “Prerrequisitos”.	Se presiona el botón “Cerrar”, no se guardan los cambios y se regresa a EC 1.1.
	EC 3.12: Modificar datos de un prerrequisito no seleccionado.	Se selecciona la opción “Editar” sin haber seleccionado previamente un prerrequisito. Se muestra un mensaje indicando que debe seleccionar al menos un prerrequisito para que pueda ser editado.
SC 4: Eliminar Aspectos Generales del DI	EC 4.1: Eliminar objetivos.	Se selecciona un objetivo de la lista y seguidamente la opción “Eliminar”. Se muestra un mensaje indicando si realmente se desea eliminar el objetivo seleccionado.
	EC 4.2: ‘Confirmar mensaje de eliminación.	Se acepta el mensaje de confirmación, se elimina el objetivo seleccionado y se regresa a EC 1.1.
	EC 4.3: Denegar mensaje	No se acepta el mensaje de confirmación, por lo que no se



	de modificación.	elimina el objetivo y se regresa a EC 1.1.
	EC 4.4: Eliminar datos de un objetivo no seleccionado.	Se selecciona la opción “Eliminar” sin haber seleccionado previamente un objetivo. Se muestra un mensaje indicando que debe seleccionar al menos un objetivo para que pueda ser eliminado.
	EC 4.5: Eliminar prerrequisito.	Se selecciona un objetivo de la lista y seguidamente la opción “Eliminar”. Se muestra un mensaje indicando si realmente se desea eliminar el prerrequisito seleccionado.
	EC 4.6: Confirmar mensaje de eliminación.	Se acepta el mensaje de confirmación, se elimina el prerrequisito seleccionado y se regresa a EC 1.1.
	EC 4.7: Denegar mensaje de modificación.	No se acepta el mensaje de confirmación, por lo que no se elimina el prerrequisito y se regresa a EC 1.1.
	EC 4.8: Eliminar datos de un prerrequisito no seleccionado.	Se selecciona la opción “Eliminar” sin haber seleccionado previamente un prerrequisito. Se muestra un mensaje indicando que debe seleccionar al menos un prerrequisito para que pueda ser eliminado.

*Descripción de variables*

No	Nombre del campo	Clasificación	Valor nulo	Descripción
1	Nombre	Campo de texto	Sí.	Texto, números y símbolos.
2	Referencia	Lista desplegable.	No.	Texto, números y símbolos.

3	Visible	Lista desplegable.	No.	Solo texto.
---	---------	--------------------	-----	-------------

*Ejecución de casos de prueba***SC 1 Mostrar Aspectos Generales del DI**

ID del escenario	Escenario	Nombre	Resultado de la prueba
EC 1.1	Mostrar datos.	V	Satisfactorio.

**SC 2 Crear Aspectos Generales del DI**

ID del escenario	Escenario	Nombre	Referencia	Visible	Resultado de la prueba
1	EC 2.1	V	V	V	Satisfactorio.
2	EC 2.2	V	V	V	Satisfactorio.
3	EC 2.3	N/A	N/A	N/A	Satisfactorio.
4	EC 2.4	N/A	N/A	N/A	Satisfactorio.
5	EC 2.5	V	V	V	Satisfactorio.
6	EC 2.6	N/A	N/A	N/A	Satisfactorio.
7	EC 2.7	N/A	N/A	N/A	Satisfactorio.
8	EC 2.8	N/A	N/A	N/A	Satisfactorio.
9	EC 2.9	N/A	N/A	N/A	Satisfactorio.
10	EC 2.10	V	V	V	Satisfactorio.
11	EC 2.11	V	V	V	Satisfactorio.

12	EC 2.12	N/A	N/A	N/A	Satisfactorio.
13	EC 2.13	N/A	N/A	N/A	Satisfactorio.
14	EC 2.14	V	V	V	Satisfactorio.
15	EC 2.15	V	V	V	Satisfactorio.
16	EC 2.16	N/A	N/A	N/A	Satisfactorio.
17	EC 2.17	N/A	N/A	N/A	Satisfactorio.
18	EC 2.18	N/A	N/A	N/A	Satisfactorio.

**SC 3 Modificar Aspectos Generales del DI**

ID del escenario	Escenario	Nombre	Referencia	Visible	Resultado de la prueba
1	EC 3.1	V	V	V	Satisfactorio.
2	EC 3.2	V	V	V	No satisfactorio.
3	EC 3.3	V	V	V	No satisfactorio.
4	EC 3.4	V	V	V	No satisfactorio.
5	EC 3.5	V	V	V	Satisfactorio.
6	EC 3.6	V	V	V	Satisfactorio.
7	EC 3.7	V	V	V	Satisfactorio.
8	EC 3.8	V	V	V	No satisfactorio.
9	EC 3.9	V	V	V	No satisfactorio.
10	EC 3.10	V	V	V	No satisfactorio.

11	EC 3.11	V	V	V	Satisfactorio.
12	EC 3.12	V	V	V	Satisfactorio.

**SC 4 Eliminar Aspectos Generales del DI**

ID del escenario	Escenario	Nombre	Referencia	Visible	Resultado de la prueba
1	EC 4.1	V	V	V	No satisfactorio.
2	EC 4.2	V	V	V	No satisfactorio.
3	EC 4.3	V	V	V	No satisfactorio.
4	EC 4.4	V	V	V	Satisfactorio.
5	EC 4.5	V	V	V	No satisfactorio.
6	EC 4.6	V	V	V	No satisfactorio.
7	EC 4.7	V	V	V	No satisfactorio.
8	EC 4.8	V	V	V	Satisfactorio.

*Registro de defectos y dificultades detectadas*

Elementos	No.	No conformidad	Etapas de detección	Estado NC
<b>Aplicación.</b>	1	En la sección Modificar Aspectos Generales del DI, se debe enviar un mensaje de confirmación indicando si realmente se desean realizar los cambios al objetivo luego de este ser editado. Pero dicho mensaje no existe.	1ra iteración.	Pendiente 05/05/11

Elementos	No.	No conformidad	Etapas de detección	Estado NC
<b>Aplicación.</b>	2	En la sección Modificar Aspectos Generales del DI, se debe aceptar el mensaje de confirmación enviado sobre el objetivo, pero no se acepta ningún mensaje pues este no existe.	1ra iteración.	Pendiente 05/05/11
<b>Aplicación.</b>	3	En la sección Modificar Aspectos Generales del DI, no se debe aceptar el mensaje de confirmación sobre el objetivo, pero no se realiza esta acción pues dicho mensaje no existe.	1ra iteración.	Pendiente 05/05/11
<b>Aplicación.</b>	4	En la sección Modificar Aspectos Generales del DI, se debe enviar un mensaje de confirmación indicando si realmente se desean realizar los cambios al prerrequisito luego de este ser editado. Pero dicho mensaje no existe.	1ra iteración.	Pendiente 05/05/11
<b>Aplicación.</b>	5	En la sección Modificar Aspectos Generales del DI, se debe aceptar el mensaje de confirmación enviado sobre el prerrequisito, pero no se acepta ningún mensaje pues este no existe.	1ra iteración.	Pendiente 05/05/11
<b>Aplicación.</b>	6	En la sección Modificar Aspectos Generales del DI, no se debe aceptar el mensaje de confirmación sobre el prerrequisito, pero no se	1ra iteración.	Pendiente 05/05/11

70 | Capítulo 3. Pruebas del módulo de Diseño Instruccional.

Elementos	No.	No conformidad	Etapas de detección	Estado NC
		realiza esta acción pues dicho mensaje no existe.		
<b>Aplicación.</b>	7	En la sección Eliminar Aspectos Generales del DI, se debe enviar un mensaje de confirmación indicando si realmente se desea eliminar el objetivo. Pero dicho mensaje no existe.	1ra iteración.	Pendiente 05/05/11
<b>Aplicación.</b>	8	En la sección Eliminar Aspectos Generales del DI, se debe aceptar el mensaje de confirmación enviado sobre el objetivo, pero no se acepta ningún mensaje pues este no existe.	1ra iteración.	Pendiente 05/05/11
<b>Aplicación.</b>	9	En la sección Eliminar Aspectos Generales del DI, no se debe aceptar el mensaje de confirmación sobre el objetivo, pero no se realiza esta acción pues dicho mensaje no existe.	1ra iteración.	Pendiente 05/05/11
<b>Aplicación.</b>	10	En la sección Eliminar Aspectos Generales del DI, se debe enviar un mensaje de confirmación indicando si realmente se desea eliminar el prerrequisito. Pero dicho mensaje no existe.	1ra iteración.	Pendiente 05/05/11
<b>Aplicación.</b>	11	En la sección Eliminar Aspectos Generales del DI, se debe aceptar el mensaje de		Pendiente

71 | Capítulo 3. Pruebas del módulo de Diseño Instruccional.

Elementos	No.	No conformidad	Etapas de detección	Estado NC
		confirmación enviado sobre el prerrequisito, pero no se acepta ningún mensaje pues este no existe.	1ra iteración.	05/05/11
<b>Aplicación.</b>	12	En la sección Eliminar Aspectos Generales del DI, no se debe aceptar el mensaje de confirmación sobre el prerrequisito, pero no se realiza esta acción pues dicho mensaje no existe.	1ra iteración.	Pendiente 05/05/11
<b>Aplicación.</b>	13	En la sección Modificar Aspectos Generales del DI, se debe enviar un mensaje de confirmación indicando si realmente se desean realizar los cambios al objetivo luego de este ser editado.	2da iteración.	Resuelta 10/05/11
<b>Aplicación.</b>	14	En la sección Modificar Aspectos Generales del DI, se debe aceptar el mensaje de confirmación enviado sobre el objetivo.	2da iteración.	Resuelta 10/05/11
<b>Aplicación.</b>	15	En la sección Modificar Aspectos Generales del DI, se debe enviar un mensaje de confirmación indicando si realmente se desean realizar los cambios al prerrequisito luego de este ser editado.	2da iteración.	Resuelta 10/05/11

Elementos	No.	No conformidad	Etapas de detección	Estado NC
<b>Aplicación.</b>	16	En la sección Modificar Aspectos Generales del DI, se debe aceptar el mensaje de confirmación enviado sobre el prerrequisito.	2da iteración.	Resuelta 10/05/11
<b>Aplicación.</b>	17	En la sección Eliminar Aspectos Generales del DI, se debe enviar un mensaje de confirmación indicando si realmente se desea eliminar el objetivo.	2da iteración.	Resuelta 10/05/11
<b>Aplicación.</b>	18	En la sección Eliminar Aspectos Generales del DI, se debe aceptar el mensaje de confirmación enviado sobre el objetivo.	2da iteración.	Resuelta 10/05/11
<b>Aplicación.</b>	19	En la sección Eliminar Aspectos Generales del DI, se debe enviar un mensaje de confirmación indicando si realmente se desea eliminar el prerrequisito.	2da iteración.	Resuelta 10/05/11
<b>Aplicación.</b>	20	En la sección Eliminar Aspectos Generales del DI, se debe aceptar el mensaje de confirmación enviado sobre el prerrequisito.	2da iteración.	Resuelta 10/05/11

### 3.2 Conclusiones del capítulo.

Al concluir el presente capítulo se logró probar de forma satisfactoria el módulo de Diseño Instruccional. Durante la ejecución de las pruebas de Caja Negra mediante los casos de prueba se identificaron y



## 73 | Capítulo 3. Pruebas del módulo de Diseño Instruccional.

corrigieron una serie de errores que fueron pasados por alto durante la etapa de implementación de este sistema. Este procedimiento mejoró la calidad del producto final y garantizó que las funcionalidades definidas inicialmente estuviesen implementadas correctamente.

## Conclusiones generales.

En la realización del presente trabajo se dio cumplimiento de forma satisfactoria a los objetivos propuestos inicialmente. La implementación del módulo de Diseño Instruccional arrojó las siguientes conclusiones:

- La evaluación de la propuesta de análisis y diseño del módulo de IMS – LD, hasta el nivel A, para CRODA hizo evidente incompatibilidades con la renovación de las herramientas y tecnologías propuestas. Las mismas fueron solventadas y pudo reutilizarse la mayor parte de la propuesta inicial en la presente solución.
- La implementación del nivel A del estándar IMS –LD en el módulo de Diseño Instruccional en CRODA posibilitó la realización del DI en la misma. Se alcanzó además la meta de incrementar la calidad de los OA generados en esta herramienta de autor con el consiguiente beneficio para la docencia en la UCI.
- Por último se probaron de forma satisfactoria las funcionalidades desarrolladas en este sistema a través de pruebas de Caja Negra. Durante este proceso se detectaron algunas deficiencias en el tratamiento de errores de la aplicación que fueron rápidamente corregidas para lograr un producto final de mayor calidad.

## Recomendaciones

Se recomienda la implementación de los restantes niveles de la especificación IMS – LD, o sea, los niveles B y C para alcanzar mayor funcionalidad en la gestión del DI en la herramienta de autor CRODA.

Además sería aconsejable el desarrollo de un reproductor (*player*) que permita emular el manifiesto generado por el módulo de Diseño Instruccional, y así, los profesores y estudiantes podrían apreciar la efectividad del DI durante su ejecución.

## Bibliografía

- Acuña, Karenny Brito.** Por qué utilizar RUP para desarrollar aplicaciones web. 2009a. [citado: 30 Abril 2011]. Disponible en la web: <<http://www.eumed.net/libros/2009c/584/Por%20que%20utilizar%20RUP%20para%20desarrollar%20aplicaciones%20web.htm>>.
- Acuña, Karenny Brito.** RUP Diseño e implementación del sistema. *eumed.net* 2009b. [citado: 21 Marzo 2011]. Disponible en la web: <<http://www.eumed.net/libros/2009c/584/RUP%20Diseno%20e%20implementacion%20del%20sistema.htm>>.
- ADL.** Comunidad de eLearning. eLearnign WORKSHOPS. 2004. Disponible en la web: <<http://www.elearningworkshops.com/modules.php?name=News&file=article&sid=335>>.
- Berlanga, Adriana J.** *IMS Learning Design: Hacia la Descripción Estandarizada de los Procesos de Enseñanza*. 2005 [citado: 3 Junio 2011]. Disponible en la web: <<http://avellano.fis.usal.es/~aberlanga/files/Pubs/BerlangaetaISINTICE05-Pub.pdf>>.
- Bray, Tim.** *Extensible Markup Language (XML)*. 2008.
- Choi, E.M., y A. von Mayrhauser.** Testing object-oriented systems using extended use-cases. Vol. 2 of, [PDPTA Conference], 2000 [citado: 22 Mayo 2011]. Disponible en la web: <<http://www.informatik.uni-trier.de/~ley/db/conf/pdpta/pdpta2000.html#ChoiM00>>.
- Colectivo de autores.** Campus virtual de salud pública. 2009. Disponible en la web: <<http://devserver.paho.org/virtualcampus/moodle/mod/glossary/view.php?id=1138&mode=date>>.
- Compendium Institute.** CompendiumLD version 1.0: Readme. 2010. [citado: 3 Junio 2011]. Disponible en la web: <<http://compendiumld.open.ac.uk/documentation/readme/ReadMe.html>>.
- Corley, L.** *Consejería de educación Ciencia y tecnología*. 2003 [citado: 3 Junio 2011].
- Cuaresma, Sergi Blanco.** Metodologías de desarrollo. Febrero 2008. [citado: 22 Mayo 2011]. Disponible en la web: <<http://www.marblestation.com/?p=644>>.

## 77 | Bibliografía

- Denis, Yordania M., y Eduardo A. Varona.** Análisis y diseño del módulo IMS-Learning Design para CRODA. 2010. [citado: 21 Marzo 2011]. Disponible en la web: <[http://bibliodoc.uci.cu/TD/TD\\_02852\\_10.pdf](http://bibliodoc.uci.cu/TD/TD_02852_10.pdf)>.
- Fernández, Luis, Pedro J. Lara, y Celia Gutiérrez.** Generación de casos de prueba a partir de especificaciones UML. 2003. [citado: 22 Mayo 2011]. Disponible en la web: <<http://www.esi.uem.es/~plaraber/Investigacion/JICS2003.pdf>>.
- Flores, Adriana J. Berlanga, y Francisco J. García Peñalvo.** *Introducción a los Estándares y Especificaciones para Ambientes e-learning*. 2004.
- Gil, Mary Cárdenas.** Ateneo Empresarial. 2009. [citado: 3 Junio 2011]. Disponible en la web: <<http://ateneo-empresarial.com/archives/category/disenoinstruccional>>.
- IMS Learning Design Information Model.** IMS Learning Design Information Model 2003a. [citado: 28 Marzo 2011]. Disponible en la web: <[http://www.imslobal.org/learningdesign/ldv1p0/imsld\\_infov1p0.html#1495294](http://www.imslobal.org/learningdesign/ldv1p0/imsld_infov1p0.html#1495294)>.
- IMS Learning Design XML Binding.** *IMS Learning Design XML Binding* 2003b. [citado: 28 Marzo 2011]. Disponible en la web: <[http://www.imslobal.org/learningdesign/ldv1p0/imsld\\_bindv1p0.html#1502862](http://www.imslobal.org/learningdesign/ldv1p0/imsld_bindv1p0.html#1502862)>.
- Larraz, Rosana.** Cuaderno Intercultural. 2010. Disponible en la web: <<http://www.cuadernointercultural.com/>>.
- Liber, Oleg.** RELOAD Reusable eLearning Object Authoring. 2006. [citado: 3 Junio 2011]. Disponible en la web: <<http://www.reload.ac.uk/>>.
- Mañas, José A.** Prueba de Programas. Marzo 1994. [citado: 22 Mayo 2011]. Disponible en la web: <<http://www.lab.dit.upm.es/~lprg/material/apuntes/pruebas/testing.htm#s22>>.
- Meier, Wolfgang.** *eXist: An Open Source Native XML Database*. 2002.
- Myers, G. J.** The art of software testing. 1979. [citado: 22 Mayo 2011]. Disponible en la web: <<http://www.carlosfau.com.ar/nqi/nqifiles/The%20Art%20of%20Software%20Testing%20-%20Second%20Edition.pdf>>.

## 78 | Bibliografía

- Orellana, N., J.M. Suárez, y C. Belloch.** EL DISEÑO INSTRUCCIONAL, UNA DIMENSIÓN CLAVE INSUFICIENTEMENTE ATENDIDA EN LA TELEFORMACIÓN. 2001. [citado: 3 Junio 2011]. Disponible en la web: <[www.uv.es/~bellochc/doc%20UTE/VE2001\\_042.pdf](http://www.uv.es/~bellochc/doc%20UTE/VE2001_042.pdf)>.
- Eguíluz Pérez, Javier.** Introducción a JavaScript. 2007.
- Perovich, Daniel.** Rational Unified Process. 2008. Disponible en la web: <[https://www.u-cursos.cl/ingenieria/2008/1/CC61H/1/material\\_docente/objeto/161148](https://www.u-cursos.cl/ingenieria/2008/1/CC61H/1/material_docente/objeto/161148)>.
- Potencier, Fabien, y François Zaninotto.** *Symfony, la guía definitiva*. Diciembre 2008.
- Quiñones A., Ernesto.** *Introducción a postgresql*. 2008.
- Red TTnet.** La formación sin distancia. Estudio realizado por el Grupo de Trabajo de e-learning de la red TTnet. 2005.
- Technologies Cogigraph Inc.** TCI - Technologies Cogigraph Inc. > PRODUCTS > MOT and MOTplus. 2008. [citado: 3 Junio 2011]. Disponible en la web: <<http://www.cogigraph.com/Produits/MOTetMOTplus/tabid/995/language/en-US/Default.aspx>>.
- The Apache Software Foundation.** The Apache Software Foundation. 2011. [citado: 22 Mayo 2011]. Disponible en la web: <<http://www.apache.org/>>.
- Torres, Javier Gómez.** Aprender a Pensar. Repensar la educación. 2010. Disponible en la web: <<http://tecnofilos.aprenderapensar.net/2010/01/29/scorm-1-2-y-scorm-2004/>>.
- Vázquez, José Manuel Márquez.** *Estado del arte del eLearning. Ideas para la definición de una plataforma universal*. 2007. Disponible en la web: <<http://www.lsi.us.es/docs/doctorado/memorias/Marquez,%20Jose%20M.pdf>>
- Villar, José.** EasyProf. Powerful Multimedia Course Authoring. 2007. [citado: 3 Junio 2011]. Disponible en la web: <<http://www.easyprof.com/index.php?>>.
- Yukavetsky, Gloria J.** La elaboración de un módulo instruccional. 2008. [citado: 3 Junio 2011].
- Zammetti, Frank W.** *Practical Ext JS Projects with Gears*. Steve Anglin [United States of America]: Apress, Inc, 2009.

## Glosario de términos

- ✓ **HTTP:** *HyperText Transport Protocol* (Protocolo de Transporte de Hipertexto). Protocolo para mover archivos de hipertexto a través de Internet.
- ✓ **LMS:** - *Learning Management System* (Sistema de gestión de aprendizaje). Es un programa que permite organizar materiales y actividades de formación en cursos, gestionar la matrícula de los estudiantes, hacer seguimiento de su proceso de aprendizaje, evaluarlos, comunicarse con ellos mediante foros de discusión, chat o correo electrónico, etc.
- ✓ **MISA:** MISA es una metodología que ayuda al diseño de un sistema de aprendizaje. No es solamente el resultado de numerosos años de investigación en el campo de la ingeniería pedagógica sino también el resultado de experimentos realizados a través de diversos proyectos de formación.
- ✓ **OWL-DL:** *Web Ontology Language Description Logic* es el lenguaje basado en la DL SHOIN e indicado para los usuarios que requieren el máximo de expresividad mientras exigen completitud computacional (se garantiza que todas las conclusiones son computables) y decibilidad (todos los cálculos acaban en un tiempo finito); sin embargo, no soporta tipos de datos definidos por los usuarios (DataType).
- ✓ **REST:** La Transferencia de Estado Representacional (*Representational State Transfer*) o REST es una técnica de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web.

- ✓ **RIA:** *Rich Internet Applications* (Aplicaciones de Internet Enriquecidas). Son aplicaciones web que tienen la mayoría de las características de las aplicaciones tradicionales, estas aplicaciones utilizan un “navegador web” estandarizado para ejecutarse y por medio de un “plugin” o una “virtual machine”, se agregan las características adicionales.
- ✓ **SGML:** Sistema para definir lenguajes para dar formato a documentos.
- ✓ **SOAP:** *Simple Object Access Protocol*, protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.
- ✓ **XML-RPC:** Es un protocolo de llamada a procedimiento remoto que usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes.
- ✓ **Xpath:** (XML Path Language) es un lenguaje que permite construir expresiones que recorren y procesan un documento XML.
- ✓ **XQuery:** Es un lenguaje de consulta diseñado para consultar colecciones de datos XML. Es semánticamente similar a SQL, pero incluye algunas capacidades de programación.
- ✓ **XSLT:** XSLT o Transformaciones XSL es un estándar de la organización W3C que presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML.
- ✓ **WebDav:** Es un grupo de trabajo del Internet Engineering Task Force. El término significa "Edición y versionado distribuidos sobre la web".
- ✓ **Widget:** Es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de *widgets* o *Widget Engine*. Entre sus objetivos están dar fácil acceso a funciones frecuentemente usadas y proveer de información visual.