



Instituto Superior Politécnico “José Antonio Echeverría”
Facultad de Ingeniería Industrial
Centro de Estudios de Ingeniería y Sistemas

Herramienta de autor para la creación y gestión de objetos de aprendizaje reutilizables

**Trabajo de diploma para optar por el título
de Ingeniería en Informática**

Autores

Adrian Cid Almaguer
Juenlis Enrique Coss Piña

Tutores

MSc. David Leyva Leyva
Ing. Daymy Tamayo Ávila

Ciudad de La Habana, Cuba
Junio, 2006

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas y al Centro de Estudios de Ingeniería y Sistemas para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del _____.

Juenlis Enrique Coss Piña

Adrian Cid Almaguer

Autores

MSc. David Leyva Leyva

Ing. Daymy Tamayo Ávila

Tutores

*“Es justamente la posibilidad de realizar un sueño
lo que hace que la vida sea interesante”*

Paulo Coelho

Dedicatoria

*A quien con su sufrimiento me enseñó las lecciones de la vida,
me enseñó a caminar, a levantarme si me caigo, a aprender de mis errores,
fuente de mi vida y de mis ganas de vivir.*

A mi madre.

Juenlis E. Coss Piña

A mi familia y mis amigos.

Adrian Cid Almaguer

A todas las personas que creen que un mundo sin Windows es posible.

Agradecimientos

Agradecemos a todos los que nos han acompañado en este largo camino de la vida y la educación dándonos fuerzas para salir adelante y realizar nuestros sueños.

Gracias a todos.

Resumen

En los últimos años la enseñanza virtual ha ocupado un lugar significativo en el entorno educativo a escala mundial, lo que ha propiciado la aparición de novedosas tecnologías y numerosas propuestas de estandarización. Con este trabajo “Herramienta de autor para la creación y gestión de objetos de aprendizaje” se pretende desarrollar una aplicación Web que elabore *objetos de contenidos educativos* basados en las especificaciones SCORM (Modelo de Referencia para Objetos de Contenidos Intercambiables), para que puedan ser utilizados por las diferentes plataformas de educación a distancia que tengan implementados estos estándares.

Para su elaboración se realizó un estudio de las herramientas que implementan paquetes SCORM, lo cual derivó la decisión de utilizar software “open source” (código abierto) para el desarrollo de esta propuesta: una herramienta Web flexible, de fácil administración, interacción agradable y cómodo uso, que posibilite la creación de paquetes SCORM que puedan ser utilizados para la elaboración de cursos por los diferentes Sistemas de Gestión de Contenidos de Aprendizaje (LCMS).

Abstract

During the last years, virtual learning has occupied an important place in the educative campus all over the world, which have afforded the growing of new technologies and several proposals of standarizations. With this work " Herramienta de autor para la creación y gestión de objetos de aprendizaje" it is intended to develop a WEB application, based on specifications SCORM, (Shareable Content Object Reference Model) which work on educative contents objects to be used for the different stages of distance learning having those available standars.

For that a deep study on the tools was developed. That work guided us to determine the use of the software "open source" to achieve the proposal: a flexible web tool, easy-manageable and nice interaction which give the possibility to create SCORM packets which can be used for the creation of courses to the different Learning Content Management Systems (LCMS).

Índice

INTRODUCCIÓN.....	1
CAPÍTULO 1	
FUNDAMENTACIÓN TEÓRICA.....	6
1.1 INTRODUCCIÓN.....	6
1.2 CONCEPTOS FUNDAMENTALES Y SUS CARACTERÍSTICAS.....	6
1.2.1 <i>El e-learning como modalidad educativa.....</i>	<i>6</i>
1.2.1.1 ¿Qué es el e-learning?.....	6
1.2.1.2 Herramientas para el acceso al e-learning.....	7
1.2.2 <i>Objetos de aprendizaje.....</i>	<i>11</i>
1.2.2.1 Los Objetos de Aprendizajes Reutilizables (RLOs).....	11
1.2.3 <i>Estándares e-learning.....</i>	<i>13</i>
1.2.3.1 ¿Qué son los estándares e-learning?.....	13
1.2.4 <i>El estándar SCORM.....</i>	<i>18</i>
1.2.4.1 Metadatos de SCORM.....	19
1.2.5 <i>Internet e Intranet.....</i>	<i>27</i>
1.2.6 <i>Aplicaciones Web frente a sitios Web.....</i>	<i>30</i>
1.2.7 <i>Modelo Cliente Servidor.....</i>	<i>31</i>
1.2.8 <i>El software libre.....</i>	<i>32</i>
1.3 HERRAMIENTAS PARA LA CREACIÓN Y GESTIÓN DE OBJETOS DE APRENDIZAJE.....	36
1.3.1 <i>Introducción.....</i>	<i>36</i>
1.3.2 <i>Experiencias Generales.....</i>	<i>36</i>
1.3.3 <i>Intentos de llevar a la web.....</i>	<i>38</i>
1.4 LENGUAJES DE PROGRAMACIÓN PARA LA WEB.....	39
1.5 SERVIDORES WEB.....	43
1.6 SISTEMAS DE GESTIÓN DE BASES DE DATOS.....	45
1.7 METODOLOGÍA DE DESARROLLO DEL SOFTWARE.....	47
1.7.1 <i>Rational Unified Process (RUP).....</i>	<i>48</i>
1.8 CONCLUSIONES DEL CAPÍTULO.....	52
CAPÍTULO 2	
DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....	53
2.1 INTRODUCCIÓN.....	53
2.2 MODELO DE DOMINIO.....	53
2.2.1 <i>Descripción del Problema de Dominio.....</i>	<i>54</i>
2.3 DEFINICIÓN DE LAS ENTIDADES Y LOS CONCEPTOS PRINCIPALES.....	56
2.4 REPRESENTACIÓN DEL MODELO DEL DOMINIO.....	57
2.5 ACTORES DEL SISTEMA A AUTOMATIZAR.....	57
2.6 REQUISITOS DEL SISTEMA.....	58
2.6.1 <i>Requisitos funcionales.....</i>	<i>58</i>
2.6.2 <i>Requisitos no funcionales.....</i>	<i>58</i>
2.7 DIAGRAMA DE CASO DE USOS DEL SISTEMA.....	60
2.8 DESCRIPCIÓN DE LOS CASOS DE USO.....	60
2.9 DIAGRAMAS DEL DISEÑO.....	67
2.10 DISEÑO DE LA INTERFAZ.....	74
2.11 TRATAMIENTO DE ERRORES.....	75
2.12 AYUDA.....	76
2.13 DISEÑO DE LA BASE DE DATOS.....	76
2.13.1 <i>Diagrama de clases persistentes.....</i>	<i>76</i>
2.13.2 <i>Modelo de datos.....</i>	<i>77</i>
2.14 DIAGRAMA DE DESPLIEGUE.....	78
2.15 CONCLUSIONES DEL CAPÍTULO.....	79
CAPÍTULO 3	
ESTUDIO DE FACTIBILIDAD.....	80
3.1 INTRODUCCIÓN.....	80

<u>3.2 ESTIMACIÓN DEL ESFUERZO.....</u>	<u>80</u>
<u>3.3 BENEFICIOS TANGIBLES E INTANGIBLES.....</u>	<u>85</u>
<u>3.4 ANÁLISIS DE COSTOS Y BENEFICIOS.....</u>	<u>86</u>
<u>3.5 VALORACIÓN DE SOSTENIBILIDAD SEGÚN LA DIMENSIÓN SOCIO-CULTURAL, ECONÓMICA, AMBIENTAL Y TECNOLÓGICA.</u>	<u>86</u>
<u>3.6 CONCLUSIONES DEL CAPÍTULO.....</u>	<u>88</u>
<u>CONCLUSIONES GENERALES.....</u>	<u>89</u>
<u>RECOMENDACIONES.....</u>	<u>90</u>
<u>REFERENCIAS BIBLIOGRÁFICAS.....</u>	<u>91</u>
<u>ANEXOS.....</u>	<u>100</u>
<u>ANEXO 1. DIAGRAMA DE GANTT.....</u>	<u>100</u>

Introducción

Actualmente las universidades de todo el mundo están muy vinculadas al uso de las Tecnologías de la Informática y las Comunicaciones (TICs) en el proceso de enseñanza-aprendizaje. La Enseñanza Virtual o e-learning ha cobrado fuerzas en los últimos años; su aplicación se ha convertido en un requisito casi indispensable en los centros de educación superior a nivel mundial.

Uno de los factores que se tienen en cuenta para la asimilación de las TICs en estos centros es la selección de un LCMS¹, mediante el cual se realiza la distribución y gestión de cursos a través de la red, mejorando la calidad del proceso de enseñanza aprendizaje.

Entre los LCMS más difundidos encontramos la plataforma Moodle² que “es un paquete de software para la creación de cursos y sitios Web basados en Internet, diseñado para dar soporte a un marco de educación social constructivista”. [21]

Moodle para la creación de los cursos manipula objetos de aprendizaje que son piezas individuales de contenido que sirven con fines instruccionales, ejemplos de ellos son páginas Web, documentos, imágenes o gráficas digitales, archivos de video y audio, etc.

Para lograr que otras plataformas de educación a distancia como Blackboard y WebCT manipulen de la misma forma estos recursos educativos, organizaciones internacionales como el Institute for Electrical and Electronic Engineers Learning Technology Standards Committee (IEEE LTSC); Advanced Distributed Learning (ADL); IMS Global Learning Consortium y Aviation Industry CBT Committee (AICC) han propuesto varias especificaciones y estándares que

¹ Learning Content Management Systems. Este término es abundado en el capítulo I

² Sistema que será objeto de estudio para la realización de las pruebas de nuestra aplicación.

garantizan la interoperabilidad, reusabilidad, durabilidad y accesibilidad de los objetos de aprendizaje.

Uno de los estándares más utilizados por los LCMS y soportado por Moodle es SCORM, que es un modelo de referencia que establece un modo de desarrollar, empaquetar y gestionar la distribución de unidades formativas digitales.

Para que los paquetes SCORM sean reutilizables, es necesario que estos y los objetos de aprendizaje por los que están compuestos sean descritos a través de metadatos, que son datos que describen características importantes como: título, descripción, palabras claves, entre otros.

Para la creación de paquetes SCORM existen varias herramientas de autor que son las encargadas de confeccionar los cursos de acuerdo a los estándares establecidos internacionalmente, dando la posibilidad de añadir recursos con sus metadatos y organizar la estructura del curso, siendo de gran utilidad para la creación de paquetes SCORM reutilizables que cumplan con las especificaciones y estándares propuestos.

Entre las herramientas de autor podemos encontrar el Exelearning, Reload y Lompad que son de libre distribución, basadas en licencias de código abierto, siendo esto una de sus ventajas, sin embargo a pesar que unas permiten la creación de objetos de aprendizaje o paquetes SCORM y otras posibilitan la adición de metadatos descriptivos a los paquetes ya creados; en su mayoría carecen de funcionalidades para la descripción de los metadatos necesarios en la creación de objetos de aprendizaje reutilizables, elemento indispensable para lograr que el contenido pueda ser acoplado con mayor facilidad en los repositorios de objetos de aprendizaje que implementen los estándares.

Otras deficiencias presentes en estas herramientas es que muchas son aplicaciones Desktop, obligando al usuario a su previa instalación. El diseño de los entornos de trabajo no son accesibles y poco usables, generando problemas de ubicación dentro de la herramienta, que conjuntamente con las implementaciones parciales de las funcionalidades que deben estar presente

en las herramientas de autor conllevan al usuario a la utilización de más de una herramienta para la creación de paquetes SCORM.

Tomando en cuenta la situación explicada anteriormente, se ha planteado el **problema fundamental** ¿Cómo crear y gestionar objetos de aprendizaje, basándose en las especificaciones SCORM, para propiciar la reutilización de los mismos en plataformas de educación a distancia que soportan este estándar?

Se define como **objeto de estudio** de esta investigación las herramientas de autor para la creación de paquetes de objetos de aprendizaje y el **campo de acción** la creación de paquetes de objetos de aprendizaje reutilizables que incluyan metadatos.

Para solucionar el problema planteado el presente trabajo tiene como **objetivo** desarrollar una aplicación Web dinámica basada en código abierto que permita generar paquetes de objetos de aprendizaje, posibilitando la reutilización de los contenidos didácticos en los cursos elaborados en plataformas de educación a distancia que soporten los estándares correspondientes.

Para dar cumplimiento a los objetivos se han planteado las siguientes **tareas**:

- Realizar un estudio de las especificaciones y estándares desarrollados para e-learning y para los Objetos de Aprendizaje.
- Evaluar las Herramientas de Autor existentes.
- Hacer el análisis y diseño de la Herramienta de Autor para la creación y gestión de Objetos de Aprendizaje reutilizables.
- Implementar una Herramienta de Autor para la creación y gestión de Objetos de Aprendizaje reutilizables.
- Utilizar la filosofía cliente servidor para la realización de la Herramienta de Autor para la creación y gestión de Objetos de Aprendizaje reutilizables.

- Implantar la Herramienta de Autor para la creación y gestión de Objetos de Aprendizaje reutilizables.

Métodos de investigación

Para llevar a cabo este proyecto han de combinarse métodos de investigación, teóricos y empíricos.

Teóricos:

- Modelación: permite realizar un estudio de las diferentes herramientas de autor existentes, sus cualidades y peculiaridades, para el correcto desarrollo del sistema
- Análisis: desglosando el problema para que, en pequeños problemas conduzca a una mejor solución y encontrar las relaciones entre ellos.

Empíricos:

- Observación: posibilita la obtención del conocimiento acerca del comportamiento de los sistemas existentes hasta el momento, además nos permitió conocer la realidad mediante la percepción directa de los objetos y fenómenos.
- Revisión de Documentos: se emplea para el reconocimiento del funcionamiento de las aplicaciones y las formas en que operan.

El presente trabajo consta de introducción, tres capítulos, conclusiones, recomendaciones y bibliografía:

En el capítulo 1, *Fundamentación teórica*, se da una breve descripción de diferentes conceptos imprescindibles que fundamentan la investigación, las ventajas del uso del software libre, las herramientas de programación actuales necesarias para dar cumplimiento a los objetivos trazados; y por último se aborda la metodología de ingeniería del software para el desarrollo del sistema.

En el capítulo 2, *Diseño e implementación*, se abordan los distintos pasos de la metodología expuesta en el capítulo inicial para el desarrollo de la aplicación, describiendo cada una de sus fases.

El capítulo 3 y final, *Estudio de factibilidad*, presenta un estudio de factibilidad apoyado en las normas ofrecidas por la estimación por casos de uso. Se analizan además, los aspectos necesarios para una correcta valoración del impacto socioeconómico y medioambiental que tendrá la implantación del sistema.

Fundamentación Teórica.

1.1 Introducción.

En el capítulo se hace una exposición de algunos conceptos necesarios para el desarrollo de aplicaciones para e-learning. Se abordan los distintos tipos de e-learning que existen en la actualidad y se explican aspectos relacionados con los estándares SCORM (Sharable Content Object Reference Model), los objetos de aprendizaje, los LMS, LCMS, qué son y para qué se utilizan.

Además, se describen algunas de las herramientas de autor más usadas en la creación y gestión de objetos de aprendizaje. Por último se da una explicación acerca del uso de algunas metodologías para el desarrollo de sistemas informáticos, haciendo énfasis en la que se propone para el desarrollo de este trabajo.

1.2 Conceptos fundamentales y sus características.

1.2.1 El e-learning como modalidad educativa

1.2.1.1 ¿Qué es el e-learning?

Desde la irrupción de Internet en nuestras vidas, las posibilidades de acceso a la formación se han ido incrementando en la medida en que la Red nos va permitiendo acceder a más personas y ofrecer ambientes de aprendizaje más complejos y elaborados.

El término e-learning se refiere a la utilización de las nuevas tecnologías de la información y la comunicación con un propósito de aprendizaje [7].

Una de estas tecnologías es Internet, pero también puede incluir la multimedia o los simuladores.

Una de las principales ventajas del e-learning es la facilidad de acceso. La formación puede llegar a más personas, puesto que desaparecen las barreras espacio-temporales.

De esta forma, personas que antes tenían dificultades para estar en contacto continuo con los procesos de formación, por problemas de desplazamiento al centro donde se imparten los cursos, por escasez de tiempo, por incapacidad física para asistir a clase, por vivir en pequeñas aldeas poco comunicadas con el exterior, tienen ahora todo un abanico de posibilidades puestas a su disposición para una continua formación.

E-Learning es una modalidad formativa que permite una educación a distancia donde los alumnos acceden a los contenidos, actividades y tareas del curso a través de las Plataformas Tecnológicas [7], no compartiendo necesariamente con los profesores un mismo espacio físico; permitiendo además una formación semi-presencial, siendo un recurso utilizado para reducir el grado de presencialidad de los cursos de formación; asistiendo los alumnos solo a algunas sesiones presenciales, no para recibir información sino para resolver dudas, observar y practicar.

1.2.1.2 Herramientas para el acceso al e-learning.

El repaso de los progresos de las soluciones e-learning en los últimos años revela la aparición y desarrollo de plataformas como los CMS (Sistemas de Gestión de Contenido), LMS (Sistemas de Gestión de Aprendizaje) y los LCMS. También existe otro término CMS, utilizado en este trabajo, que es empleado para denominar Sistemas para la Gestión de Cursos. A continuación se explican cuáles son las principales características de estos sistemas:

CMS o Content Management Systems:

Se conoce como CMS a las aplicaciones que en la industria de las publicaciones online permiten la creación y administración de contenido [29]. Actualmente se usa para denominar a los sistemas que sirven para administrar el contenido de páginas Web. El objetivo que estos programas informáticos persiguen y cumplen con enorme eficacia es la creación y la gestión de información online, estando esa información compuesta por textos (artículos, informes, entre otros), gráficos, vídeos, sonido, etc.

Para la creación de la información a servir se utiliza la estrategia de separar el contenido de la presentación. De hecho, las personas que se encargan de cada aspecto suelen ser distintas, además la información se construye ensamblando porciones de contenidos que se denominan “componentes”, esos pedazos de información pueden considerarse independientes entre sí y su característica más importante es que son reusables. Además, la utilización de componentes reusables permite personalizar la información a la medida del usuario que la consume.

Estas ideas justifican el éxito de los sistemas CMS, y son perfectamente trasladables al mundo del e-learning. Pero desafortunadamente estos sistemas CMS no tienen la capacidad de gestionar el proceso de aprendizaje, lo cual es esencial en todo proyecto de e-learning.

LMS o Learning Management Systems (Sistemas de Gestión de Aprendizaje):

Surgieron como respuesta a las necesidades expresadas anteriormente. [29] Permiten planificar el aprendizaje de acuerdo a las necesidades de los usuarios, sean estos estudiantes, trabajadores, empresas, etc. Permiten también mejorar las competencias de los usuarios de los cursos y su intercomunicación. Es posible adaptar la formación a los requisitos de la empresa y al propio desarrollo profesional. Permiten la distribución de cursos, recursos, noticias y contenidos relacionados con la formación en general;

además, pueden servir como soporte para el registro de los cursistas, acceso a recursos tales como material audiovisual, demos, etc.

La implementación de una plataforma LMS no garantiza, sin embargo, los medios para la creación y generación de los cursos necesarios para la organización. Desde la perspectiva de los materiales docentes simplemente actúa como plataforma de distribución donde se remarca la idea de que en un sistema LMS la mínima unidad de instrucción es el curso en sí mismo. Los LMS permiten una eficiente gestión de los usuarios, pero demuestran carencias en la gestión de los contenidos.

LCMS o Learning Content Management Systems (Sistemas de Gestión de Contenido de Aprendizaje):

Los LCMS representan la integración de dos vías tradicionalmente separadas: los CMS y los LMS. [29] Estos dos mundos se han desarrollado de espaldas entre sí, ajenos a una realidad incuestionable. El aprendizaje a través de Internet necesariamente requiere de recursos que permitan tanto la creación como la distribución de contenidos integrados en una misma plataforma; esto permite a expertos en cualquier área del saber, pero no necesariamente a expertos en el manejo del software específico de generación de materiales, diseñar, crear, distribuir y controlar la eficacia del proceso de aprendizaje de una forma sencilla, rápida y eficiente.

Aparentemente los LCMS podrían contribuir a resolver muchos de los problemas anteriormente mencionados. Por lo tanto, una plataforma LCMS además de garantizar el control del proceso de aprendizaje, debe facilitar la creación, almacenamiento y reparto de los contenidos, con las siguientes características:

1. Herramientas sencillas que facilitan la creación de contenidos, en forma de herramientas de autor en el sistema, incluyendo editores con el fin de eliminar la necesidad de manejar editores HTML.

2. Sistemas flexibles de diseño y distribución de los cursos que permitan adaptarse a las necesidades de la organización y a los diferentes sistemas y ritmos de aprendizaje de los usuarios.

3. Posibilidad de reusar los objetos de aprendizaje; de hecho cada pieza de conocimiento debiera ser tratado como un objeto de aprendizaje reusable (RLO) y mantenido a disposición de los integradores que requieran su empleo en los diseños de los diversos cursos.

4. Herramientas para la administración del sistema que permita las matriculaciones, el seguimiento del aprendizaje, el uso de los tiempos, la trazabilidad de los usuarios, la adecuación de los contenidos, etc.

5. Herramientas para la evaluación tanto inicial como de los aprendizajes que se producen a lo largo del curso; y ello tanto en lo que se refiere a los cursos en general como a los objetos de aprendizaje en particular; el sistema debe proveer de recursos suficientes para valorar los aprendizajes bajo distintos niveles de dificultad y diferentes modalidades de medición.

6. Conectividad con otros LCMS y en general adecuación a los estándares actuales.

7. Herramientas para la comunicación y el aprendizaje colaborativo. Incluirá recursos tanto síncronos como asíncronos que faciliten la comunicación sencilla entre iguales y con el profesorado, y además recursos para el aprendizaje colaborativo que permitan compartir recursos de conocimiento y realizar trabajos en grupo.

8. Mecanismos de seguridad y protección del conocimiento almacenado; dicha seguridad dependerá del uso de los privilegios de los diferentes usuarios y de las diferentes funciones que los mismos desarrollan dentro de la organización,

y afectará a las cargas y descargas de documentación así como al acceso a la misma.

9. Sencillez en la migración de contenidos para facilitar la adaptación a las diferentes necesidades y escenarios de formación que se puedan presentar.

10. Facilidad de instalación que hagan innecesarias las adaptaciones, personalizaciones y demás procesos que encarecen el producto y retrasan el proceso de instalación.

CMS (Sistema para la Gestión de Cursos):

Los Sistemas para la Gestión de Cursos son aplicaciones Web que ofrecen a los profesores herramientas para crear un curso dentro de un sitio Web y facilitar el control del acceso a dicho curso.

Los Sistemas para la Gestión de Cursos ofrecen también una gran variedad de herramientas que posibilitan crear un curso de manera efectiva. Proporcionan la forma de “subir” y “bajar” materiales al curso, realizar discusiones y Chat, efectuar tareas, evaluaciones, resúmenes, entre otras actividades.

1.2.2 Objetos de aprendizaje.

1.2.2.1 Los Objetos de Aprendizajes Reutilizables (RLOs).

La Web puede ser considerada [18] como una “gran colección de objetos digitales”. Se pueden encontrar recursos de diferentes formatos y para aplicaciones de todo tipo y sobre cualquier tema. Sin embargo, el problema es que esta colección no tiene un orden, la búsqueda es compleja entre tanta información dispar y la permanencia de la información es impredecible. Para que los recursos de información disponibles en la Web sean verdaderamente explotables se han dado iniciativas y tecnologías para la organización de los recursos.

Los objetos de aprendizaje, también llamados objetos didácticos [26], surgen con el interés de compartir recursos y para su reutilización en el ámbito

educativo, originados bajo el paradigma de la orientación a objetos. Es aplicado a “materiales digitales creados como pequeñas piezas de contenido o de información, con la finalidad de maximizar el número de situaciones educativas en que las que el recurso pueda ser utilizado.”[18]

Otra definición que muestra más claramente su utilización es la que plantea [29] que un objeto de aprendizaje es: “La unidad mínima de experiencia de instrucción que contiene un objetivo, una actividad de aprendizaje, y una evaluación.”

Un componente importante para la reutilización de un objeto es que esté adecuadamente descrito, en este caso a través de sus correspondientes metadatos.

Un registro de metadatos consiste en “un conjunto de atributos o elementos necesarios para describir el recurso en cuestión”. Para la implementación de los metadatos el estándar más utilizado actualmente es el LOM (Learning Objects Metadata).

Por otro lado, si los metadatos utilizados se apegan a los estándares propuestos y se basan en tecnologías abiertas, también la reutilización de éstos toma un papel de gran valor para compartir información entre repositorios. La consistencia semántica en la descripción de los recursos y la normalización de los datos que cada elemento contenga, también son factores clave para la posible reutilización a través de sistemas.

El uso de objetos de aprendizaje en el desarrollo de cursos, favorece la implementación de más y mejores sistemas de categorización y búsqueda, mecanismos robustos para actualización y envío de datos, y la estructuración y definición de recursos educativos. [1]

La estrategia de granular los contenidos educacionales en pequeños Objetos de Aprendizaje es un avance significativo en la producción de materiales educativos, pero esto por sí solo no es suficiente. Para su real utilidad se requiere que estos objetos sean compatibles con diversos ambientes y

sistemas de administración de aprendizajes, fáciles de migrar de una plataforma a otra, fáciles de localizar, acceder, archivar y reutilizar. [31]

La satisfacción de estos requisitos dará una vida útil más larga a los materiales didácticos electrónicos y su valor será mayor. De ahí la importancia de la utilización de estándares para el diseño y descripción de los objetos de aprendizaje.

1.2.3 Estándares e-learning.

1.2.3.1 ¿Qué son los estándares e-learning?

Dentro de los entornos e-learning participan individuos con distintos intereses y objetivos, sistemas informáticos con funciones diversas y tecnologías heterogéneas, así como contenidos con características, objetivos y formatos de todo tipo. En este complejo ámbito, un reto es la interoperabilidad, esto es, entornos o sistemas de diferentes desarrolladores para distintas aplicaciones y contenidos diversos, que trabajen juntos en sistemas distribuidos de aprendizaje.

Para lograr dicha interoperabilidad es necesario apegarse a los estándares y a las especificaciones que importantes grupos del sector e-learning desarrollan.

En la educación en línea, los estándares se ven como necesarios ahora más que antes, dado el alcance global que tienen las aplicaciones e-learning en los sistemas de telecomunicación, y el creciente interés de los individuos en la autoformación y en el aprendizaje a lo largo de toda la vida; todo ello marca un aumento en el uso de estos modelos que crecen de forma dispersa y demandan de esfuerzos conjuntos para expandirse.

La estandarización se requiere a distintos niveles, primero, cuando los recursos son creados deben considerarse tecnologías, políticas y formatos compatibles con lo común en el sector; segundo, cuando esos recursos son incluidos en un repositorio, y deben ser descritos, se utilizarán esquemas que aseguren su fácil

localización y compatibilidad con otros sistemas de metadatos; tercero, cuando esos recursos sean utilizados y tengan que incorporarse a diferentes servicios, repositorios, plataformas y aplicaciones en un contexto dado; y cuarto, cuando los sistemas involucrados en un entorno tengan que interoperar con otros para cumplir sus funciones o ampliar sus capacidades.

La creación de estándares globales es una tarea compleja, sin embargo, diferentes grupos trabajan en el desarrollo tanto de especificaciones como de estándares en los distintos niveles requeridos [8], para establecer entornos e-learning integrados.

1.2.3.2 Diferencia entre estándares y especificaciones

Un estándar es un patrón, una tipificación o una norma sobre cómo realizar algo; los hay de dos tipos: estándares de jure, cuando provienen de una organización acreditada que certifica una especificación, y estándares de facto, cuando la especificación se adopta por un grupo mayoritario de individuos. Entonces, un estándar regularmente proviene de una especificación, esto es, de un conjunto de declaraciones detalladas y exactas de los requisitos funcionales y particularidades de algo que quiere construirse, instalarse o manufacturarse.

Los estándares sólo pueden producirlos cuerpos internacionales reconocidos por uno o varios gobiernos nacionales; por ejemplo, cuando se habla de estándares web producidos por el Consorcio de la World Wide Web, éstos realmente, producen especificaciones, no estándares.

No hay un proceso específico para la conformación de un estándar de jure, pero de forma típica se siguen los siguientes pasos:

- Investigación y desarrollo.
- Desarrollo de una especificación.
- Pruebas.

- Acreditación e internacionalización del estado del estándar.

En el área educativa, las especificaciones son formalmente remitidas al LTSC (Learning Technology Standards Committe), Comité especializado en e-learning del IEEE, única organización acreditada de estandarización. Hasta el momento, el único estándar emitido es LOM (Learning Object Metadata), un esquema de metadatos para la descripción de recursos educativos.

1.2.3.3 Cuerpos de especificaciones y de estándares.

Los esfuerzos de los cuerpos que desarrollan especificaciones y estándares están orientados hacia una forma común de identificar, definir y comunicar a todos los recursos involucrados en un entorno e-learning (contenidos, docentes, estudiantes, aplicaciones, proveedores, etcétera). A continuación, se describen los cuerpos que trabajan en el desarrollo de propuestas para la estandarización del e-learning:

AICC: Es una asociación de entrenamiento profesional basado en tecnología. Se reconoce como una de las precursoras de la estandarización de materiales para entrenamiento profesional. <http://www.aicc.org/>

IMS Global Consortium Inc. Cuenta con miembros de organizaciones comerciales, educativas y gubernamentales dedicadas a definir y distribuir arquitecturas abiertas para actividades de educación en línea. Uno de sus resultados es lo que se conoce como el estándar IMS. <http://www.imsproject.org/>

ADL: La misión de ADL es la de proveer acceso de la más alta calidad en educación y entrenamiento, en cualquier lugar y momento, para lo cual crean el modelo SCORM. <http://www.adlnet.org/>

ARIADNE (Alliance of Remote Instructional Authoring and Distribution Networks for Europe): Patrocinado por la Unión Europea, este proyecto está enfocado al desarrollo de herramientas y metodologías para producir, administrar y

reutilizar elementos pedagógicos basados en computadora, así como la currícula de entrenamiento a distancia. <http://www.ariadne-eu.org/>

IEEE/LTSC: Es una asociación internacional que dentro de su organización cuenta con el Comité de Estándares para Tecnología del Aprendizaje o LTSC, encargado de desarrollar estándares técnicos, recomendaciones y guías para la tecnología educativa. <http://www.ieee.org/>

W3C. (World Wide Web Consortium). Se encarga del desarrollo de tecnologías interoperables (especificaciones, normas, software y herramientas) para aprovechar todo el potencial del web. Aunque no está directamente vinculado con el desarrollo del e-learning es importante mencionarlo, ya que de la interoperabilidad de la web dependen muchas de las funciones de la educación en línea. <http://www.w3.org/>

De estos grupos de desarrollo, las propuestas más ampliamente adoptadas, en gran número de herramientas, han sido las de IMS Learning Consortium y la de ADL, que han tenido como resultado las especificaciones IMS y el modelo SCORM, respectivamente.

1.2.3.4 Ventajas del uso de estándares.

Los estándares han sido un elemento indispensable para la masificación de cualquier tecnología, puesto que favorecen el crecimiento, la expansión y la generalización. En el campo del e-learning están en proceso de comprensión y de adopción, y conforme se vayan introduciendo cada vez más, habrá beneficios en interoperabilidad, reutilización, manejabilidad, accesibilidad, durabilidad, escalabilidad y confiabilidad, tanto en contenidos como en infraestructura y funcionalidad.

Muchas de las empresas y organizaciones que hacen desarrollos o imparten programas de educación a distancia ven estos beneficios reflejados como:

- Contenidos reutilizables.
- Consistencia en la descripción de los contenidos.
- Normalización en la organización de sus recursos.
- Acceso a más contenidos, de mayores fuentes y fácilmente localizables.
- Persistencia de sus acervos.
- Migración sencilla de sus sistemas a nuevas versiones, e incluso a una nueva plataforma.
- Comunicación e intercambio de información con otros sistemas.
- Administración de la información apropiada tanto del recurso como del estudiante.
- Extensión de los servicios y de las capacidades de las plataformas.
- Inversión asegurada en la infraestructura por mayor tiempo.

La utilización de estándares amplía las opciones de los usuarios finales y reduce las restricciones de los sistemas propietarios y de soluciones aisladas. Las instituciones, los docentes, los estudiantes y los proveedores se verán beneficiados al contar con contenidos flexibles, plataformas homogéneas y bases de datos compartidas y distribuidas.

1.2.4 El estándar SCORM.

El Modelo Referenciado de Objetos de Contenido Compartible (SCORM, por sus siglas en inglés) es un conjunto de especificaciones técnicas que permiten desarrollar, empaquetar y transportar contenidos educativos de alta calidad entre distintos entornos sin perder funcionalidades. [7] Estas especificaciones detallan cómo deben publicarse los contenidos y usarse los metadatos, además de representar la estructura de los cursos por medio de XML y el uso de API (Application Programming Interface).

Los materiales educativos se elaboran asegurándose del cumplimiento de cuatro principios: **accesibilidad**, **interoperabilidad**, **durabilidad** y **reutilización**, los cuales tienen como consecuencia la reducción de los costes y de los tiempos.

El SCORM resuelve el requisito de **accesibilidad** proporcionando una manera estándar de empaquetamiento de contenidos como objetos reutilizables y auto contenido. [8]

La **interoperabilidad** se presenta como la habilidad de poder utilizar en distintas plataformas componentes educativos creados con diferentes herramientas y desde cualquier ubicación.[8] SCORM responde a este requerimiento estandarizando la manera de comunicar entre los LMS y los objetos de contenido, especificando de cómo se debe agregar y empaquetar el contenido.

La **durabilidad** está dada por la capacidad de un componente educativo de hacer frente a los cambios tecnológicos sin un rediseño, reconfiguración o sobrescribir el código. [8]

La **reutilización** de los objetos de aprendizaje se logra mediante la flexibilidad que brinda el SCORM de incorporar componentes educativos en múltiples aplicaciones y contextos, debido al modo uniforme de empaquetamiento de contenidos a través de estándares y especificaciones que este proporciona. [8]

Se puede decir que SCORM consta de tres componentes:

Empaquetamiento de contenidos. Se refiere a la manera en que se guardan los contenidos de un curso, el modo en que están ligados entre sí y la forma en la que se entregará la información al usuario. Todos estos datos se concentran en un archivo llamado **imsmanifest.xml**

Ejecución de comunicaciones. Detalla el ambiente para ejecutar la información y consta de dos partes; los *comandos de ejecución* y los *metadatos del estudiante*.

Metadatos del curso. Son de dos tipos; los que incluyen la *información del curso* en sí, y los que contienen el *material del estudiante*.

1.2.4.1 Metadatos de SCORM

Según IEEE, cada elemento de los metadatos de LOM es opcional, lo que implica que el desarrollador opcionalmente puede escoger cuales elementos usar. SCORM establece requisitos adicionales, que indican cuales elementos son de carácter obligatorio.

El Modelo de Información de LOM está compuesto por nueve categorías:

La categoría General agrupa información general del Componente del Modelo de Contenido como un todo.

La categoría Ciclo de Vida contiene características relacionadas con la historia y el estado actual del Componente del Modelo de Contenido y qué lo ha afectado en su evolución.

La categoría Meta-metadato se usa para describir información sobre los metadatos.

La categoría Técnica se usa para describir requerimientos y características técnicas de los Componentes del Modelo de Contenido.

La categoría Educativa describe características educacionales y pedagógicas de los Componentes del Modelo de Contenido.

La categoría Derechos contiene los derechos de propiedad intelectual y condiciones de uso del Componente del Modelo de Contenido.

La categoría Relación agrupa características que definen relaciones entre el Componente del Modelo de Contenido y otros componentes.

La categoría Comentario se usa para proveer comentarios del uso educacional del Componente del Modelo de Contenido e información sobre cuándo y por quiénes fueron creados los comentarios.

La categoría Clasificación se usa para describir dónde se encuentra el Componente del Modelo de Contenido, de acuerdo a un sistema de clasificación particular.

Algunos elementos usan el término SPM (el máximo permitido más pequeño) para describir la multiplicidad de los elementos. El SPM indica a las aplicaciones que procesan metadatos que tienen que procesar al menos ese número de elementos o caracteres, pero son libres de excederlo.

Elemento <lom>

Todas las instancias de metadatos tienen a <lom> como el elemento raíz. En un paquete de contenido, todos los metadatos se encuentran dentro del elemento <metadata> encontrado en el fichero imsmanifest.xml. El nodo raíz encapsula todas las categorías descritas anteriormente. No hay un orden para las nueve categorías, los elementos hijos pueden aparecer en cualquier orden.

Todas las declaraciones de espacio de nombres deben estar dentro del elemento <lom>. Esto incluye cualquier espacio de nombre que sea considerado ampliación de los metadatos. Sin embargo, esto no es considerado un requerimiento teniendo en cuenta las especificaciones de XML,

ADL considera que esto es una “buena práctica” y estimula que se provea esa información. Este elemento al igual que sus nodos hijos contiene como espacio de nombre: <http://ltsc.ieee.org/xsd/LOM> y como prefijo de espacio de nombre: lom.

El elemento <lom> contiene importantes elementos que SCORM requiere para describir todos los Componentes del Modelo de Contenido. Este elemento debe aparecer sólo una vez dentro de su nodo padre.

Hijos:

<general>

<lifeCycle>

<metametadata>

<technical>

<educational>

<rights>

<relation>

<annotation>

<clasification>

A continuación se hace una descripción sólo de los metadatos que son de interés para el desarrollo de este trabajo.

Elemento <general>

La categoría General agrupa la información general que describe al recurso como un todo. El recurso a describir en este caso son los componentes del Modelo de Contenido de SCORM (Recurso, SCO, Actividad, una Organización de Contenidos o el paquete como un todo). Esta información general importante para describir al componente.

Es un elemento padre y debe aparecer solo una vez dentro de su nodo padre.

Hijos:

<title>

<description>

<keyword>

Elemento <title>

Representa el nombre del componente.

Multiplicidad	SPM	Tipo de Dato	SPM
1 y solo 1	-	LangString	1000

Elemento <description>

Representa una descripción textual del Componente del Modelo de Contenido.

Multiplicidad	SPM	Tipo de Dato	SPM
1 o más	10	LangString	2000

Elemento <keyword>

Se usa para definir palabras o frases que describen el objeto de aprendizaje. Si existen más de una palabra clave se deben usar varias instancias del elemento.

Multiplicidad	SPM	Tipo de Dato	SPM
1 o más	10	LangString	1000

Elemento <lifeCycle>

Agrupar características relacionadas a la historia y al estado actual del Componente del Modelo de Contenido y qué lo afectó en su evolución. Incluye

el estado del componente, versión y una lista de organizaciones que han afectado su estado de una forma u otra.

Es un elemento padre y debe aparecer 1 sola vez dentro de su nodo padre.

Hijo:

<version>

Elemento <version>

Describe la edición del Componente del Modelo de Contenido. Un componente puede tener varias versiones o ediciones. Este elemento permite la descripción de la versión del componente en su ciclo de vida.

Multiplicidad	SPM	Tipo de Dato	SPM
1 y solo 1	-	LangString	50

Elemento <metametadata>

En la versión 1.3 de SCORM este elemento cambia por <metaMetadata>.

Contiene elementos que describen los metadatos del componente y no el componente en cuestión. Esta categoría describe cómo los metadatos pueden ser identificados, quien creó la instancia, cómo, cuándo y con cuáles referencias.

Es un elemento padre y debe aparecer solo 1 vez dentro de su nodo padre.

Hijos:

<metadatascheme>

Elemento <metadatascheme>

Representa el nombre y la versión de la especificación utilizada para crear las instancias de los metadatos. Si se encuentran varios valores, entonces la instancia de los metadatos se conformará de múltiples esquemas de metadatos.

SCORM requiere el siguiente esquema de metadatos:

LOMv1.0 Indica que se usó la versión 1.0 del Esquema Base de LOM.

Si otros esquemas de metadatos son relevantes, pueden aparecer. Sin embargo, esto no es relevante para SCORM 2004.

Multiplicidad	SPM	Tipo de Dato	SPM
0 o 1	-	CharacterString	100

Para la versión SCORM 2004 este elemento cambia por <metadataSchema>. Su multiplicidad cambia para 2 o más, siendo requerido por SCORM en esta versión también este esquema de metadatos:

ADLv1.0 Indica que se usó el Perfil de Aplicación de Metadatos de SCORM.

Elemento <technical>

Describe las características y requerimientos técnicos del Componente del Modelo de Contenido.

Es un elemento padre y debe aparecer solo 1 vez dentro de su nodo padre.

Hijos:

<format>

Elemento <format>

Es usado para identificar el formato del Componente del Modelo de Contenido.

Multiplicidad	SPM	Tipo de Dato	SPM
1 o más	40	CharacterString	500

Debe ser un tipo de datos MIME representado en RFC 2048:1996.

Elemento <rights>

Describe los derechos de propiedad intelectual y condiciones de uso para el componente del modelo de contenido SCORM. Este elemento debe ser usado para describir cualquier o todos los derechos digitales del componente SCORM(costo de uso, copyright, etc)

El elemento solo debe aparecer una sola y solo una vez.

Hijos:

<copyrightandotherrestriccions>

Elemento < copyrightsandotherrestriccions >

Si es usado con un valor positivo (yes) el elemento <description> puede ser usado para describir detalles adicionales acordes al copyright y otras restricciones.

Es usado para identificar el formato del Componente del Modelo de Contenido.

Multiplicidad	SPM	Tipo de Dato	SPM
1 y solo 1	-	Vocabulary	-

Vocabulario que debe usarse:

- yes
- no

1.2.5 Internet e Intranet

El fenómeno de Internet tiene su origen en la década del 60 del siglo XX y se relaciona con un proyecto de defensa financiado por el gobierno de Estados Unidos. Gracias a esta iniciativa, hoy es posible buscar, crear y transferir información en tiempo real para millones de personas.

Para desarrollar el proyecto se constituyó un grupo financiado por la Defense Advanced Research Project Agency (DARPA). Luego la conexión se extendió a computadoras pertenecientes a entidades universitarias, que participaron activamente en éste y en otros proyectos del Departamento de Defensa para intercambiar mensajes e información a través de esta red.

Desde ese minuto el crecimiento de Internet ha sido fenomenal, el proyecto comenzó a expandirse en los inicios de la década del 90, época en que la red se convirtió en una herramienta esencial de comunicación, información e integración, que permite a los usuarios ahorrar tiempo y dinero, además de tener a su alcance una serie de productos y servicios sin fronteras de espacio o tiempo.

Internet se ha convertido en uno de los medios de comunicación más extendido en toda la historia de la humanidad. Esta bien llamada "Red de redes", permite establecer la cooperación y colaboración entre gran número de comunidades y grupos de interés por temas específicos, distribuidos por todo el planeta. A través de ella es posible encontrar toda clase de software para una gran variedad de computadoras y sistemas operativos, pueden consultarse catálogos de bibliotecas del mundo, acceder a bases de datos con los temas más diversos y transferir copias de los documentos encontrados, es posible visualizar y copiar archivos de imágenes con fotografías o reproducciones de cuadros, pueden conversar a tiempo real dos o más personas, separadas por miles de kilómetros de distancia. [13]

Sin duda alguna la sociedad en general se ha visto influenciada de alguna u otra forma por el uso de Internet. Las diferentes instituciones y organizaciones de la sociedad han sido testigo de ello y han incorporado a sus esquemas tradicionales mecanismos para adaptarse a la actual era de la información. Se ha vuelto cotidiano el uso de los servicios que brinda Internet. No se puede dejar de nombrar el uso del correo electrónico y de las salas de conversación que han convertido al correo tradicional en algo casi obsoleto en lo que a comunicación se refiere sin dejar de tener en cuenta ya el salto en servicios

como publicidad y ventas, así como entretenimientos y noticias que por esta vía ganan en inmediatez y pueden llegar a cualquier lugar.

De esta forma millones de personas utilizan servicios y herramientas de Internet pero muchas de ellas o bien conocen muy poco o ignoran la esencia de cómo es que verdaderamente funciona.

La esencia de Internet

Para que dos computadoras se comuniquen vía Internet, debe existir un camino físico que las una (líneas telefónicas, conmutadas, redes digitales, enlaces satelitales, microondas, fibra óptica, cable coaxial) y un protocolo de comunicación entre ellos (TCP/IP). El desarrollo de Internet no sólo se ha traducido en beneficios para los usuarios, sino también para las empresas, organismos, instituciones, que usan a Internet como medio idóneo para el comercio electrónico. Este abarca aquellos negocios efectivos logrados a través de Internet en cuanto a transacciones o pagos realizados; en ellos al menos participan el comerciante, el proveedor de servicios y el cliente.

Los principales productos que se transan a través de la red son: información, música, viajes, libros, hardware y software. En este contexto, la Tarjeta de Crédito Bancaria es el medio de pago predilecto: un 90% de las transacciones se cancelan a través de este medio.

Cuando se habla de Internet, no se debe de enmarcar solamente a grandes distancias y a herramientas y servicios de uso global, también se puede encontrar funcionalidades específicas que, basándose en la idea de Internet pero ya en un marco limitado o más reducido, han logrado utilizar pequeñas redes para el manejo específico de sus objetivos y necesidades de las diversas instituciones, dando paso a algo así como una "Internet local", llamándose comúnmente Intranet.

1.2.6 Aplicaciones Web frente a sitios Web.

Una aplicación Web es aquella que los usuarios usan desde un servidor Web a través de Internet o de una intranet. Estas son populares debido a que todos los sistemas operativos poseen un navegador, se actualizan y mantienen sin necesidad de distribuir e instalar el software en los potenciales clientes. [12]

Las aplicaciones Web se desarrollan como una extensión de los Sistemas Web para agregar funcionalidad de negocio al proceso. En términos más simples, una Aplicación Web es un Sistema Web que permite a los usuarios ejecutar lógica de negocio a través del Navegador (Browser), o lo que es lo mismo: modificar el estado del negocio. [12] Esto se realiza cuando se actualizan las informaciones o datos que maneja la aplicación a través de las reglas del negocio.

Las Aplicaciones Web utilizan las tecnologías existentes para generar contenidos dinámicos y permitir a los usuarios del sistema modificar la lógica del negocio en el servidor. Si no existe lógica de negocios en el servidor, el sistema no puede ser considerado una aplicación Web, en ese caso se considera como un sitio Web.

La arquitectura de un Sitio Web es simple. Contiene como componentes principales: el Servidor Web, una Red y un Navegador o cliente. La arquitectura de una aplicación Web además incluye la aplicación en el Servidor, que es la que permite al sistema manejar lógica de negocio y tener un estado. [25]

Una aplicación Web llega a resolver una gama de problemas dentro del entorno empresarial muy diversa, e incluso llega a hacerlo de manera más eficiente a cómo estaba concebido inicialmente y para esto se construyen aplicaciones que engloban una serie de procesos que son independientes pero a la vez interactúan entre sí y cumplen una serie de requisitos específicos, a este conjunto de aplicaciones informáticas se les nombra sistemas.

1.2.7 Modelo Cliente Servidor.

Se dice que la arquitectura Cliente/Servidor es la integración distribuida de un sistema en red, con los recursos, medios y aplicaciones que, definidos modularmente en los servidores, administran, ejecutan y atienden las solicitudes de los clientes; todos interrelacionados física y lógicamente, compartiendo datos, procesos e información. Se establece así un enlace de comunicación transparente entre los elementos que conforman la estructura. Entre las principales características de la arquitectura Cliente/Servidor, se pueden destacar las siguientes: [28]

- El servidor presenta a todos sus clientes una interfaz única y bien definida.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Los cambios en el servidor implican pocos o ningún cambio en el cliente.

Ventajas de la arquitectura cliente-servidor: [30]

- El servidor no necesita potencia de procesamiento, parte del proceso se reparte con los clientes.
- Se reduce el tráfico de red considerablemente. Idealmente, el cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre.

Las arquitecturas de dos capas contienen tres componentes distribuidos en dos capas: cliente (solicitante de servicios) y servidor (proveedor de servicios).

Los tres componentes son: [30]

1. Interfaz de usuario al sistema. Tales como una sesión, entradas de texto, desplegado de menús, etc.
2. Administración de procesamiento. Tales como la ejecución de procesos, el monitoreo de los mismos y servicios de procesamiento de recursos.
3. Administración de bases de datos. Tales como los servicios de acceso a datos y archivos.

La arquitectura de software de tres capas emergió en la década de los noventas para solventar las limitaciones de la arquitectura de dos capas. La tercera capa (capa de servicios) se localiza entre la interfaz de usuarios (cliente) y el administrador de datos (servidor). Esta capa intermedia provee de servicios para la administración de procesos (tal como desarrollo, monitoreo y alimentación de procesos) que son compartidos por múltiples aplicaciones. [30].

El servidor de la capa intermedia (también conocido como servidor de aplicaciones) centraliza la lógica de las aplicaciones, haciendo que la administración de cambios sea más sencilla. En arquitecturas más simples, cualquier cambio en la lógica, implica reescribir todas las aplicaciones que dependan de esta. [30]

1.2.8 El software libre.

En 1983 un grupo de hackers, liderados por Richard Stallman consideran que el software no debe ser objeto comercial pues, para ellos, se trata de conocimiento científico y como tal, debe transmitirse libre y ser útil para el progreso de la humanidad. Con esta intención crean la Fundación para el Software Libre (FSF, *Free Software Foundation*) y el proyecto GNU, con el objetivo de producir aplicaciones de libre distribución amparadas en una licencia que las proteja de las patentes comerciales, esta licencia se conoce con las siglas GPL (*General Public License*).

A principios de los años 90, Linus Benedict Torvald, un estudiante finlandés desarrolló, a partir de una versión educativa del sistema operativo Unix, llamada Minix, un sistema operativo totalmente nuevo llamado Linux. Este sistema era totalmente compatible con los estándares de Unix y el día 5 de octubre de 1991 se publicó la primera versión oficial. [9]

En 1992, Linux se integra en el proyecto GNU y a partir de ese momento, un grupo de entusiastas comienzan a mejorar el código y a compartirlo a través de la red Internet hasta conseguir los niveles de desarrollo actuales. [9]

Hoy en día, el software libre es una realidad. Las soluciones diseñadas pueden ser utilizadas por cualquier persona, ya que ahora no hay que ser un experto en informática para poder trabajar con las aplicaciones creadas para el entorno de usuario final.

Los consultores opinan que el software libre es una amenaza competitiva para los sistemas basados en software propietario. Una encuesta realizada por la consultora Forrester indica que más del 70% de las 50 grandes compañías con una facturación superior a 1.000 millones de dólares contemplan implantar sistemas de software libre en un futuro próximo, así como que más del 50% de las 2.500 mayores empresas utilizan algún producto informático basado en código abierto (Open Source). [6]

Ventajas para las empresas.

Costo del software: Cuando una organización contrata a un desarrollador, el código fuente es propiedad de la empresa, por lo que se ata a dicho desarrollador y es posible realizar posteriores modificaciones con otros proveedores. Puede parecer a primeras que es más costoso, pero a la larga la decisión puede ser un gran ahorro de costos en licencias.

Estabilidad: ¿Qué pasa si el fabricante deja de fabricar el producto o se cambian las funcionalidades porque no son rentables? Entonces la inversión se

habrá esfumado con él. En un momento en que la tecnología controla muchas partes del negocio, las necesidades de la empresa no cambian de forma inmediata, por lo que un desarrollo bien orientado podrá ser ampliado o modificado manteniendo una estabilidad tecnológica.

Personalización: Nunca un software propietario va a tener las características que necesitamos a medida. Con el software libre se pueden incorporar funciones acorde a las necesidades propias. **Seguridad:** ¿Es realmente más seguro un software propietario? Un hacker puede disponer de información abundante sobre un software que esté implantado en muchas empresas y sea reconocido, mientras que desconocerá los puntos débiles de una aplicación muy concreta y específica realizada en una entidad. Esto no descarta que el desarrollador tenga que tomar las medidas oportunas en el proyecto.

Hardware: En la mayoría de los casos, las nuevas actualizaciones de software propietario requieren de ampliaciones de hardware. Con el software libre en la mayoría de los casos se puede mantener la misma infraestructura. [23]

Ventajas para la enseñanza.

El software libre puede adaptarse a las necesidades docentes de un curso dado. Puede, por ejemplo, modificarse para ofrecer a los alumnos una versión simplificada, o darle una apariencia adecuada a los conocimientos del alumno (por ejemplo, similar a la de las herramientas con las que el alumno está familiarizado).

Si se usan programas libres, el alumno puede reproducir todo el entorno de prácticas, con total exactitud, en cualquier otra computadora sin ningún problema de licencias y sin costos extra para el alumno. Así, para cada curso se podría estampar un CD que incluya todas las herramientas utilizadas, que se repartiría a los alumnos para que saquen sus propias copias.

En el caso de que la enseñanza sea para informáticos, donde conocen las interioridades de las herramientas, la disposición del código fuente es fundamental. Esto permite, con gran facilidad y sin problemas de licencias ni acuerdos especiales con los fabricantes, ver cómo están hechas algunas herramientas reales, de calidad comercial y de esta forma, enseñar con el ejemplo, que es una de las mejores formas de enseñar informática. En general, parece razonable que las entidades educativas, no favorezcan unas empresas sobre otras. De hecho, el favorecer a una empresa sobre otra en la educación es especialmente grave, pues da a la empresa favorecida una ventaja enorme sobre la competencia: los alumnos están formados para utilizar sus productos, y por tanto preferirán usarlos frente a los de la competencia, incluso si son peores o más caros. Con el software libre esto no ocurre, cualquier empresa puede comercializar y vender servicios para cualquier producto libre. [19]

1.3 Herramientas para la creación y gestión de objetos de aprendizaje

1.3.1 Introducción

Para el desarrollo de herramientas de autor se requiere hacer un profundo estudio de los estándares y sus especificaciones. En el mundo existen escasas herramientas que implementan las especificaciones de SCORM, y desgraciadamente ninguna lo hace completamente, siendo el punto más débil de estas la nula o escasa presencia de metadatos que describan los objetos de aprendizaje.

Estas herramientas elaboradas hasta el momento, provienen de países donde no predomina el idioma español, existiendo una escasa participación en los proyectos, de desarrolladores multilingües. Esto trae consigo que las herramientas no estén disponibles en varios idiomas, lo que dificulta su utilización para usuarios finales que no dominen el idioma de la aplicación. Han existido algunos intentos de traducción de las herramientas, pero aún son deficientes.

Es importante, por lo tanto, hacer un repaso a lo que está hecho hasta hoy que mayor impacto ha logrado generar, para poder analizar sus características y el alcance logrado por ellas.

1.3.2 Experiencias Generales

Hasta el momento se conocen algunos sistemas consolidados como son:

- Reload
- Exelearning
- Lompad

El Reload, es un software de libre distribución, disponible en su Web (<http://www.reload.ac.uk>). Es una aplicación Java, que puede ejecutarse en cualquier plataforma capaz de ejecutar aplicaciones Java. Reload está en continuo desarrollo, pero desde la versión v1.1 (11 de Agosto de 2003) es estable y funcional. Las versiones posteriores incorporan mejoras en cuanto a usabilidad y personalización. Esta herramienta nos permite generar objetos de aprendizaje (SCO) de manera que cumplan con las especificaciones de SCORM, pudiendo también ejecutar y ver nuestros paquetes en un navegador web. El contenido SCORM puede ser más complejo, y permite la comunicación con un LMS mediante un entorno de ejecución basado en una serie de APIs. Reload no permite añadir a los paquetes de contenido las características que le permitan comunicarse con la plataforma, para ello es necesario herramientas de autor y de edición web como por ejemplo Dreamweaver. Si disponemos de tal contenido, para probarlo tendremos que cargarlo en un LMS que soporte tales características, pero esto puede ser problemático si no somos administradores de la plataforma. Reload Player permite solventar esa situación. Reload Player es, por tanto, un tipo de LMS pero desprovisto de las herramientas de gestión de usuario y de discusión que habitualmente facilitan las plataformas de eLearning, presentando además como principal inconveniente que no permite crear el conocimiento, es decir, hacer texto, subir fotos, etc., sino que ya dado este lo empaqueta según el estándar seleccionado.

El Exelearning o El proyecto Aprendizaje Electrónico del redactor de XHTML (eXe) como también es conocido en el mundo, es un ambiente “fuera de línea” que asiste a profesores y académicos en diseñar y publicar cursos en Internet sin la necesidad de aprender HTML o XML. eXe no es LMS, sino es un ambiente para trazar el contenido aprendizaje basado en web. Con el eXe los usuarios pueden desarrollar las estructuras de capacitación que satisfagan sus necesidades de la entrega del contenido, y crear los recursos que son flexibles y fáciles de mantener, aunque no pueden adicionarle metadatos a los objetos de aprendizaje. El proyecto del eXe apunta a fomentar el desarrollo de colaboración. La dirección estratégica y las peticiones de nuevas adiciones al programa son dadas con prioridad por un grupo de usuarios representativos

del Sector de la Educación Universitario en Nueva Zelanda donde fue creado. El eXe es un proyecto de Software Libre financiado por la Comisión de Educación Terciaria de Nueva Zelanda, está programado en Python y se instala como una extensión del navegador Firefox siendo un impedimento para muchos usuarios pues depende de este programa para ejecutarse. El eXe está actualmente disponible en su sitio oficial (<http://exelearning.org/>), pero solamente para los sistemas operativos Windows XP y Linux, siendo esta otra de sus desventajas.

El Lompad es una herramienta elaborada en Java, que no permite crear objetos de aprendizaje, solo permite crear los metadatos para ficheros SCORM previamente creados. Actualmente es bilingüe y tiene incorporado el francés y el inglés, soporta, las especificaciones del IEEE-LOM y SCORM, así como CANCORE (Canadian Core Learning Object Metadata Application Profile).

1.3.3 Intentos de llevar a la web

El principal trabajo que se está desarrollando sobre plataformas WEB es el Weload. El Weload está basado en la “Portable Web Server Utility” de Phillip Perkmann (p.perkmann@bildungsservice.at) y en el Reload, aunque solo incorpora una pequeña parte de sus funcionalidades, está programado en PHP por Hartmut Karrasch (hartmut.karrasch@iqsh.de) bajo la licencia “Creative Commons License” y está disponible desde su página web <http://weload.lemnetz.de>.

Weload necesita para su funcionamiento del servidor web Apache y del Sistema Gestor de Bases de Datos (SGDB) MySql, los cuales vienen incluidos en el paquete de distribución de la aplicación. Weload puede ser instalado tanto en el disco duro como en una memoria USB, al igual que en una gran variedad de Sistemas Operativos.

La Web ofrece y aprovecha múltiples beneficios como la conectividad de las redes locales o intranets, el acceso a sistemas basados en ella, así como el acceso a la información necesaria para un gran número de personas con escasos conocimientos de informática y con reales necesidades de acceso y manipulación en su actividad diaria. Teniendo en cuenta lo anterior se puede apreciar la utilidad de desarrollar herramientas que brinden en mayor medida una adecuada respuesta a los requerimientos y necesidades de las comunidades de Educación a Distancia, brindando mayores posibilidades de adaptaciones futuras y acceso a las informaciones, todo sobre bases del uso de un lenguaje de programación sobre plataforma Web.

1.4 Lenguajes de programación para la Web.

Las primeras Web representaban la información de forma estática, haciendo que muchos sitios debían actualizarse manualmente, lo que traía molestias, tardanza y la incorporación de un ejército de personas para realizar el proceso en dependencia del tamaño, haciendo que fuera muy costosa la actualización e hizo que muchos permanecieran abandonados, desactualizados e incluso la gran mayoría de estos sitios Web dejaron de existir. Esta situación cambió definitivamente al aparecer las aplicaciones dinámicas, las que como anillo al dedo comenzaron a prevalecer sobre las estáticas y son hoy en día las que dominan todo el espacio virtual de Internet. Ejemplos de este tipo de aplicaciones; sitios de noticia, investigación, publicaciones, entretenimiento y de búsquedas se hicieron realidad gracias a la implementación en lenguajes de programación.

Estos lenguajes se pueden dividir a fin de entender mejor su clasificación en dos partes fundamentales: los lenguajes del lado del Servidor y los lenguajes del lado del Cliente, apuntando que ambos reconocen la filosofía de la arquitectura Cliente/Servidor para las plataformas Web.

Entre los lenguajes que se encuentran en el lado del servidor podemos encontrar como los más importantes por el auge que tienen, PERL, ASP, PHP,

Java, JSP. Ellos están caracterizados por implementar la lógica del negocio dentro del Servidor, además de encargarse de la tarea del acceso a las Bases de Datos, el tratamiento de la Información, etc.

En el lado del cliente se encuentran principalmente el JavaScript (JScript) y el Visual Basic Script (VBScript), que son los encargados de aportar eficacia y dinamismo a la actividad de la aplicación en los navegadores al ejecutarse ciertas funcionalidades que no recargan al servidor, brindando eficiencia extra en cuanto a rapidez y optimización de los canales de comunicación.

En el caso del VBScript este es el que se usa como lenguaje básico a la hora de programar en ASP del lado del servidor, ya que su mayor desventaja radica en que solo es soportado por el Microsoft Internet Explorer, Navegador Web de su fabricante. Por otro lado JScript es soportado por la mayoría de los navegadores existentes actualmente.

Esta distinción en los lenguajes de programación para la Web ha sido necesaria debido a que la Web funciona en modo "Desconectado", o sea, un usuario a través de un navegador hace una petición de una página Web a un Servidor Web (Request), el Servidor recepciona la petición, la procesa y le envía la Respuesta al Cliente (Response), este la recepciona e inmediatamente se desconecta. [16]

Practical Extracting and Reporting Language (Perl).

Lenguaje de programación utilizado comúnmente para crear las aplicaciones CGI (Common Gateway Interface) para la Web. Perl es un lenguaje de programación muy práctico para extraer información proveniente de archivos de texto y generar informes a partir del contenido de esos ficheros.

Perl es un lenguaje de uso libre, eso quiere decir que es totalmente gratis. Antes estaba muy asociado a la plataforma Unix, pero en la actualidad está disponible en otros sistemas operativos. Perl está creado a partir de lenguajes

como el C, pero está diseñado para ser más práctico y fácil que este último. De esta manera un programador que haya trabajado con el lenguaje C tendrá pocas dificultades en entenderlo y utilizarlo rápidamente. Una diferencia notable de Perl con respecto a otros lenguajes es que no pone límites al tamaño de los datos con los que trabaja, el límite está dado por la memoria que en ese momento se encuentre disponible. [4]

Active Server Pages (ASP).

Páginas Active Server (ASP, Active Server Pages) es un entorno para crear y ejecutar aplicaciones dinámicas e interactivas en plataforma Web.

Se pueden combinar páginas HTML, secuencias de comandos y componentes ActiveX para crear páginas y aplicaciones Web interactivas. Esta es una tecnología que ha desarrollado Microsoft para ser ejecutado por su servidor Web Internet Information Server (IIS), utilizando el lenguaje Visual Basic Script o JScript (JavaScript de Microsoft). [2]

Actualmente se ha presentado una versión de ASP, el ASP.NET, que comprende algunas mejoras en cuanto a posibilidades del lenguaje y rapidez con la que funciona. ASP.NET tiene algunas diferencias en cuanto a sintaxis con el ASP, de modo que se ha de tratar de distinta manera uno de otro. Para implementarlo es necesario montar en el Servidor la Plataforma .NET. ASP.NET está basado en el Entorno Común de Ejecución de Lenguajes (Common Language Runtime, o "CLR"). Además permite el uso de una gran variedad de lenguajes de programación y, por tanto, usted puede escoger el mejor lenguaje según su opinión para su aplicación o particionar su solución en varios lenguajes. [11]

Java Server Pages (JSP).

JSP en español se traduce como Páginas de Servidor Java. Esta es una tecnología orientada a crear páginas Web con programación en Java. Con JSP podemos crear aplicaciones Web que mezclen HTML estático con HTML

dinámicamente. [3] Tiene varias ventajas con respecto a ASP, primero, la parte dinámica está escrita en Java, no en Visual Basic, por eso es mucho más poderosa y fácil de usar. Segundo, es portable a otros sistemas operativos y servidores Web. Las páginas JSP podremos escribirlas con nuestro editor HTML/XML habitual. El motor de las páginas JSP está basado en los servlets de Java – programas en Java destinados a ejecutarse en el servidor –, aunque el número de desarrolladores que pueden afrontar la programación de JSP es mucho mayor, dado que resulta mucho más sencillo de aprender que los servlets. [22]

Extensible Markup Language (XML).

XML no es un lenguaje de marcado como su nombre pudiera sugerir, es fácil y su uso es libre totalmente. XML, con todas las tecnologías relacionadas, representa una manera distinta de hacer las cosas, más avanzada, cuya principal novedad consiste en permitir compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes. [24] Es fácilmente procesable tanto por humanos como por software al separar radicalmente la información o el contenido de su presentación o formato. Así pues, el XML juega un papel importantísimo en este mundo actual, que tiende a la globalización y la compatibilidad entre los sistemas, ya que es la tecnología la que permitirá compartir la información de una manera segura, fiable y fácil. Además, permite al programador y los soportes dedicar sus esfuerzos a las tareas importantes cuando trabaja con los datos, ya que algunas tareas tediosas como la validación de estos o el recorrido de las estructuras corre a cargo del lenguaje y está especificado por el estándar, de modo que el programador no tiene que preocuparse por ello. XML es importante en el mundo de Internet y fundamentalmente el negocio y la publicidad porque existen muchos sistemas diferentes que tienen que comunicarse entre sí, pero como se ha podido imaginar, interesa por igual a todas las ramas de la informática y el tratamiento de datos, permitiendo muchos avances a la hora de trabajar con ellos. [5]

PHP: Hypertext Preprocessor (PHP).

El PHP fue creado originalmente en 1994 por Rasmus Lerdorf, inicialmente llamado PHP/FI. Es un lenguaje de alto nivel que se ejecuta en el lado del servidor, es gratuito e independiente de plataforma, o sea que es soportado por casi todos los sistemas operativos, rápido, con una gran librería de funciones y mucha documentación disponible. Es también un lenguaje interpretado y embebido en el HTML. Como PHP es un lenguaje de programación dentro de la corriente de “Open Source” (Código abierto), a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. [10]

Este lenguaje se distingue en que un script escrito en otro lenguaje de programación como Perl o C, en vez de escribir un programa con muchos comandos para crear una salida en HTML, se escribe el código HTML con cierto código PHP incluido en el mismo, que producirá cierta salida. El código PHP se incluye entre etiquetas especiales de comienzo y final que permite entrar y salir del modo PHP.

Para el desarrollo de la aplicación objeto de esta investigación se ha seleccionado al PHP por sus potencialidades y las ventajas que presenta como software libre.

1.5 Servidores Web

Los lenguajes de programación sobre la Web necesitan de un programa que atienda las peticiones de los sitios por los que navegamos, por lo que con esta finalidad fueron creados los servidores web.

Un servidor web es un programa que implementa el *protocolo HTTP* (hypertext transfer protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML [16].

Cabe destacar el hecho de que la palabra *servidor* identifica tanto al programa como a la máquina en la que dicho programa se ejecuta. Existe, por tanto, cierta ambigüedad en el término, aunque no será difícil diferenciar a cuál de los dos nos referimos en cada caso. Un servidor web se encarga de mantenerse a la espera de *peticiones HTTP* llevada a cabo por un *cliente HTTP* que solemos conocer como *navegador*. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita.

Internet Information Server (ISS):

Internet Information Services (o Server), IIS, es una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del *Option Pack* para Windows NT. Luego fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS [14].

El servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

Este servidor tiene como deficiencia que es propiedad de Microsoft Corporation, por lo tanto solo funciona en sistemas operativos Windows, además de sus grandes problemas de seguridad.

Apache

Como servidor de páginas Web utilizaremos Apache por su versatilidad, flexibilidad y cuestiones de compatibilidad con el lenguaje de desarrollo Web que utilizaremos.

Apache es el programa servidor HTTP. Gracias a él se puede practicar la creación y publicación de documentos PHP. Apache es multiplataforma y es uno de los mejores servidores Web utilizados en la red Internet y uno de los más populares, utilizado en más del 64% de los servidores que sirven Sitios Web en la red de redes, por lo que este servidor es uno de los mayores triunfos del software libre. [15]

Es un servidor de Web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos.

Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo y con la API de programación de módulos.

Incentiva la retroalimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para solución de los mismos.

1.6 Sistemas de Gestión de Bases de Datos.

Los lenguajes de programación sobre la Web se caracterizan por un gran dinamismo y este dinamismo se logra a través de una gestión de información, mediante los cuales intercambiamos con espacios de datos e informaciones que están almacenadas de manera organizada.

Entonces se desprende que un Sistema de Gestión de Bases de Datos (SGBD) es una herramienta que permite, mediante procedimientos o lenguajes, utilizar o actualizar datos almacenados más o menos permanentemente en una computadora, los que, organizados y relacionados entre sí, constituyen una base de datos. A esta organización de datos se le incorporan una serie de funciones y operaciones como definir los registros, sus campos, atributos, relaciones y también insertar, modificar, eliminar y consultar los datos. [17]

Para realizar la interconexión entre un SGBD y una aplicación Web es necesario utilizar un lenguaje de programación en el lado del servidor y para definir los

datos y las estructuras, así como para hacer las consultas sobre los datos, el SQL (*Structured Query Language*), algo así como un lenguaje estructurado de consultas que su uso es estándar para la comunicación entre las aplicaciones y los SGBD.

Las consultas son una combinación de instrucciones que son transferidas desde el código de la aplicación Web hasta el SGBD utilizado para actualizar y manipular las bases de datos.

Entre los SGBD comúnmente utilizados en el mundo tenemos Oracle, MySQL, Microsoft SQL Server, PostgreSQL, InterBase, entre otros. Todos estos presentan un enfoque relacional con un buen basamento matemático centrado en el Álgebra Relacional.

Entre los SGBD más populares para las aplicaciones Web, está el MySQL, asociado al PHP por su desempeño efectivo, rapidez y posibilidades casi ilimitadas, insertadas dentro del movimiento del software libre en el que los desarrolladores les han dado gran impulso, por lo que aquí tenemos el complemento que almacene las informaciones que serán manejadas en nuestra aplicación.

El Microsoft SQL Server con el ASP, este gestor de bases de datos es un sistema propietario, o sea, hay que pagar por usarlo pero es un producto muy bien logrado que en sus fuertes están los procedimientos almacenados que lo hace ganar en rendimiento y la replicación que no es más que mantener dos o más servidores de bases de datos al mismo tiempo actualizándose simultáneamente, sin embargo tiene entre sus problemas el más relevante es el de la seguridad.

El PostgreSQL es también un SGBD de uso libre cuyo mayor fuerte es el trabajo con objetos relacionales y está llamado a ser uno de los más potentes y usados.

Ahora bien, todo este conjunto de conocimientos en el que se conjuga un lenguaje de programación para aplicaciones Web, y el dominio de un SGBD no

garantiza completamente el éxito para desarrollar aplicaciones de calidad. Llevar a término una aplicación ya sea Web o de escritorio, trae aparejado conjugar los conocimientos aprendidos con la experiencia adquirida, ésta es una tarea que siempre ha sido sumamente difícil, máxime si son aplicaciones de una alta complejidad o con una gran cantidad de requerimientos. Dirigir una actividad como esta traería enormes riesgos si se hace de manera empírica, sin un mecanismo que guiara al desarrollador o a un equipo desde el inicio hasta el final garantizando una óptima calidad del producto terminado, esto es, contar con un tipo de organización para controlar el tiempo de desarrollo, el personal involucrado, el equipamiento, riesgos, costos y un correcto proceso para analizar y darle solución a cualquier problema.

1.7 Metodología de desarrollo del software.

Las metodologías imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo hacen desarrollando un proceso detallado con un fuerte énfasis en planificar inspirado por otras disciplinas de la ingeniería.

El uso de una metodología para la elaboración de un producto informático, garantiza determinadas características en el mismo, dentro de ellas la calidad, factor clave tanto para el cliente como para el productor.

El tiempo es uno de los factores críticos que afecta todo producto, más si no se ha hecho un buen estudio de la aplicación que se va a realizar. La cantidad de personal; algunas veces en exceso, otras en déficit, los sistemas de organización, los métodos de control, el dominio sobre el tema y sobre las herramientas de desarrollo por parte de los analistas y programadores, la falta de conocimientos sobre asuntos informáticos por el lado de los clientes, son otros de los factores que afectan todo el ciclo de desarrollo de una aplicación.

Sin duda alguna las metodologías que existen actualmente se adecuan al desarrollo de la mayoría de las aplicaciones, puesto que han surgido de la experiencia en la producción acumulada por varios años, incluso por varias décadas; muchas de ellas son el resultado de la evolución y desarrollo de otras unido al cambio producido en el mundo informático, influenciado por el cambio de paradigmas en la programación. [27]

Luego de un estudio de algunas de las metodologías existentes y analizado el estado de cada uno de los aspectos que se tienen en cuenta en el desarrollo de una aplicación; personal, herramientas de desarrollo, tiempo, equipamiento y otros tantos, nos podemos percatar de la coincidencia que existe entre ellas en algunas fases, pero igual encontramos sus diferencias, marcadas por diversos elementos como por ejemplo el paradigma de programación, e incluso en las fases que resultan coincidentes es notable la diferencia entre una y otra metodología.

Dada las características de la aplicación que se desarrollará; tiempo de desarrollo, condiciones de trabajo, desarrolladores, cliente, producto a realizar, se decidió la utilización de RUP, metodología que se inserta dentro del grupo de las metodologías ágiles y más usadas actualmente en el mundo.

1.7.1 Rational Unified Process (RUP).

Proceso de desarrollo propuesto por “Rational Software Corporation” resultado del esfuerzo de las tres últimas décadas en desarrollo de software y de la experiencia de sus creadores Ivar Jacobson, Grady Booch y James Rumbaugh.

Su antecedente más importante lo ubicamos en 1967 con la Metodología Ericsson (Ericsson Approach), ésta es una aproximación de desarrollo basada en componentes, que introdujo el concepto de caso de uso; entre los años de 1987 a 1995 Jacobson funda la compañía “Objectory AB” y lanza el proceso de desarrollo Objectory (abreviación de Object Factory), posteriormente en 1995

“Rational Software Corporation” adquiere “Objectory AB” y es entre 1995 y 1997 que se desarrolla “Rational Objectory Process (ROP)” fruto del encuentro y evolución de Objectory 3.8 y la Metodología Rational (Rational Approach) que adopta por primera vez UML (Universal Modeling Language) como Lenguaje de Modelación Universal.

A principios de los noventa, la guerra de los métodos hizo evidente la necesidad de unificar criterios, es así como Grady Booch autor del método Booch y James Rumbaugh (desarrollador para General Electric) se unieron en Rational en 1994, después en 1995 se une Jacobson y gracias al esfuerzo de varias compañías y metodologistas evolucionó UML hasta ser un estándar en 1997, el cual es adoptado en todos los modelos del ROP. Desde ese entonces y a la cabeza de Booch, Jacobson y Rumbaugh. [20]

Rational ha desarrollado e incorporado diversos elementos para expandir el RUP, destacándose especialmente el flujo de trabajo conocido como modelación del negocio, es así como en junio del 1998 se lanza Rational Unified Process 5.0 evolucionado hasta la actualidad bajo el nombre de RUP.

Características fundamentales

Iterativo e Incremental: Para hacer más manejable un proyecto se recomienda dividirlo en ciclos. Para cada ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un mini-proyecto cuyo núcleo fundamental está constituido por una o más iteraciones de las actividades principales básicas de cualquier proceso de desarrollo.

Desarrollo basado en componentes: La creación de sistemas intensivos en software requiere dividir el sistema en componentes con interfaces bien definidas, que posteriormente serán ensamblados para generar el sistema. Esta característica en un proceso de desarrollo permite que el sistema se vaya creando a medida que se obtienen o que se desarrollen y maduren sus componentes.

Utilización de un único lenguaje de modelado: UML es adoptado como único lenguaje de modelado para el desarrollo de todos los modelos.

Fases del RUP.

Este proceso de desarrollo considera que cualquier desarrollo de un sistema software debe pasar por cuatro fases de desarrollo y los diversos flujos de trabajo involucrados dentro de cada fase con una representación gráfica en cual de los flujos se hace mayor énfasis según la fase, cabe destacar el flujo de trabajo concerniente al negocio.

Fase 1: Preparación Inicial (“Inception”)

Su objetivo principal es establecer los objetivos para el ciclo de vida del producto. En esta fase se establece el caso del negocio con el fin de delimitar el alcance del sistema, saber qué se cubrirá y delimitar el alcance del proyecto.

Fase 2: Preparación Detallada (“Elaboration”)

Su objetivo principal es plantear la arquitectura para el ciclo de vida del producto. En esta fase se realiza la captura de la mayor parte de los requerimientos funcionales, manejando los riesgos que interfieran con los objetivos del sistema, acumulando la información necesaria para el plan de construcción y obteniendo suficiente información para hacer realizable el caso del negocio.

Fase 3: Construcción (“Construction”)

Su objetivo principal es alcanzar la capacidad operacional del producto. En esta fase a través de sucesivas iteraciones e incrementos se desarrolla un

producto software, listo para operar, éste es frecuentemente llamado versión beta.

Fase 4: Transición (“Transition”)

Su objetivo principal es realizar la entrega del producto operando, una vez realizadas las pruebas de aceptación por un grupo especial de usuarios y habiendo efectuado los ajustes y correcciones que sean requeridos. [20]

1.8 Conclusiones del capítulo.

Al concluir este capítulo se ha podido apreciar todo lo referente a la base teórica que fundamenta la presente investigación, acercando al lector a los conceptos fundamentales, el estado del arte, los lenguajes de programación para la web, los servidores web, los sistemas de gestión de bases de datos y por último la metodología de ingeniería del software a utilizar.

Descripción de la solución propuesta.

2.1 Introducción.

En este capítulo se describirá brevemente el flujo actual de los procesos involucrados en el problema manifestado, haciendo una valoración crítica de cómo se ejecutan actualmente estos procesos.

Se detallarán los distintos pasos de la metodología propuesta en el capítulo anterior (RUP) para el desarrollo de la aplicación. Esto incluirá la elaboración del modelo del dominio, la exposición de los requerimientos de la plataforma; definición de Actores y relaciones entre ellos, los Diagramas de Caso de Uso y las descripciones textuales de los mismos; los Diagramas de Clases del Diseño, Diagrama de Clases Persistentes, Modelo de datos y Diagrama de despliegue. Además se expondrán algunos elementos relacionados el diseño de la interfaz y el tratamiento de errores.

2.2 Modelo de Dominio

Existen por lo menos dos aproximaciones para documentar un sistema en una forma utilizable para los desarrolladores de software: modelado del dominio y modelado del negocio. El modelo del dominio describe los conceptos importantes del contexto como objetos del dominio, y enlaza estos objetos unos con otros. La identificación y asignación de un nombre para estos objetos ayuda a desarrollar un glosario de términos que permitirán comunicarse mejor a los que trabajen en un sistema.

Las entidades del negocio se derivan a partir de los clientes del negocio. Las clases del dominio se obtienen de la base del conocimiento de unos pocos expertos del dominio, o posiblemente del conocimiento (otras clases del dominio, especificaciones de requisitos, etc.) asociado con sistemas similares al que se está desarrollando.

Tal es el caso del presente trabajo donde se desarrolla este modelo a partir del estudio de las especificaciones propuestas por el estándar SCORM y del funcionamiento de algunas de las herramientas de autor existentes.

2.2.1 Descripción del Problema de Dominio.

El sistema Herramienta de autor para la creación y gestión de objetos de aprendizaje reutilizables permite generar paquetes SCORM a partir de estos últimos, posibilitando la reutilización de los contenidos didácticos en los cursos elaborados en plataformas de educación a distancia que soporten estos paquetes. Para lo cual se desea que el sistema sea accesible a través de un entorno Intranet/Internet.

El sistema presenta en su pantalla principal un mensaje de bienvenida describiendo los servicios ofrecidos al usuario junto con la opción para registrarse, o si ya se está registrado, poder entrar al sistema mediante la inserción del correo y una contraseña previamente escogidos, dichos datos deben validarse.

Una vez registrado el usuario, y después de haberse validado su acceso, puede seleccionar las siguientes actividades:

Subir Recurso:

Permite a los usuarios subir objetos de aprendizaje al sistema, para lo cual debe llenar algunos campos que se corresponden con las especificaciones del estándar SCORM, como son el título, la descripción, las palabras claves y el copyright.

Actualizar Recurso:

Los usuarios pueden actualizar los recursos que han subido anteriormente, brindándoles la posibilidad de cambiarles los metadatos que lo describen, cambiarles la posición que estos ocuparán dentro del curso, así como borrarlos definitivamente.

Vista previa:

Le permite a los usuarios ver una vista previa de cómo se le visualizará el paquete SCORM una vez subido a una plataforma de educación a distancia.

Generar SCORM:

Esta opción le permite al usuario generar el paquete SCORM que contendrá los objetos de aprendizaje subidos anteriormente, así como describir dicho paquete, dándole además la posibilidad de descargarlo para su maquina.

El sistema ofrecerá también a los usuarios la posibilidad de editar su perfil en caso de que el tipo de autenticación sea por email, así como eliminar su usuario del sistema.

2.3 Definición de las entidades y los conceptos principales.

Objetos de Aprendizaje: Cualquier entidad digital que pueda ser usada en el aprendizaje. Representación digital de texto, imágenes, audio, video, etc.

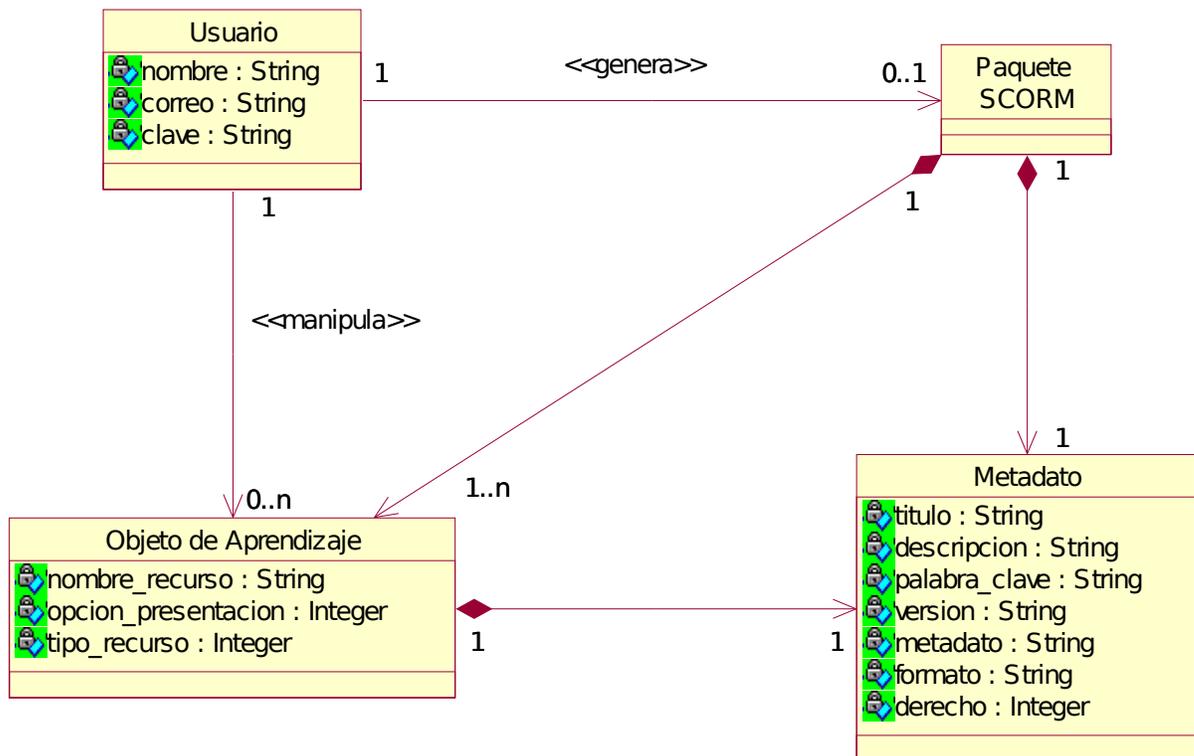
Metadatos: Descripciones de los recursos.

Paquete SCORM: Conjunto de objetos de aprendizajes reutilizables empaquetados según el estándar SCORM.

SCORM: Shareable Content Object Reference Model (Modelo de Referencia de Objetos de Contenido Intercambiable).

2.4 Representación del modelo del dominio.

Mediante el diagrama que se muestra en la figura, se visualizan y relacionan las principales clases conceptuales del dominio.



2.5 Actores del sistema a automatizar.

Nombre del actor	Justificación
Usuario Anónimo	El usuario es anónimo mientras no se autentique.
Usuario Registrado	El usuario después de autenticado, puede subir recursos, actualizarlos y descargarlos, crear paquetes y actualizar sus datos.

2.6 Requisitos del sistema.

Los requisitos son capacidades y condiciones con las cuales debe cumplir el sistema ya sean funcionales o no funcionales.

2.6.1 Requisitos funcionales.

Los requisitos funcionales indican el comportamiento del sistema. Posteriormente estos requisitos son modelados a través del diagrama de casos de uso del sistema.

- R1 Registrar usuario.
- R2 Autenticar usuario.
 - R2.1 Recuperar la clave del usuario.
- R3 Modificar usuario
 - R3.1 Actualizar los datos del usuario.
 - R3.2 Eliminar usuario.
- R4 Gestionar recurso.
 - R4.1 Subir recurso.
 - R4.2 Visualizar recursos.
 - R4.3 Modificar recurso.
 - R4.4 Eliminar recurso.
- R5 Descargar recurso.
- R6 Mostrar Vista previa del paquete
- R7 Generar paquete.

2.6.2 Requisitos no funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

Apariencia o interfaz externa

- Se debe tener en cuenta algunos elementos de diseño como gráficos de encabezamiento, estilos y formatos de texto, estilo y paletas de color de los gráficos y colores o patrones del fondo.
- No debe usarse tecnología de frames.

Rendimiento

- Para un funcionamiento óptimo de la aplicación se seguirán las diferentes técnicas de elaboración en la Web, que faciliten el rápido acceso a sus páginas. La herramienta propuesta debe ser rápida y el tiempo de respuesta debe ser el mínimo posible, adecuado a la rapidez con que el usuario requiere la respuesta a su acción.

Mantenibilidad

- Utilización de estándares para el desarrollo de aplicaciones de e-learning, lo que facilita la migración y actualización de recursos.
- Utilización de estándares para el desarrollo de aplicaciones Web.

Portabilidad

- La aplicación debe funcionar en plataformas Unix, Windows y Macintosh, siendo posible el acceso a través de cualquier navegador web.

Seguridad y privacidad

- Necesidad de autenticación doble ante acciones muy delicadas.

Legales

La plataforma escogida para el desarrollo de la aplicación, debe estar basada en la licencia GNU/GPL.

Ayuda y documentación en línea

- Concebir un sistema de ayuda de forma tal que en todo momento le permita al usuario orientarse respecto a las opciones que le brinda el sistema, utilizando textos explicativos que describan la acción que se está realizando.

Software

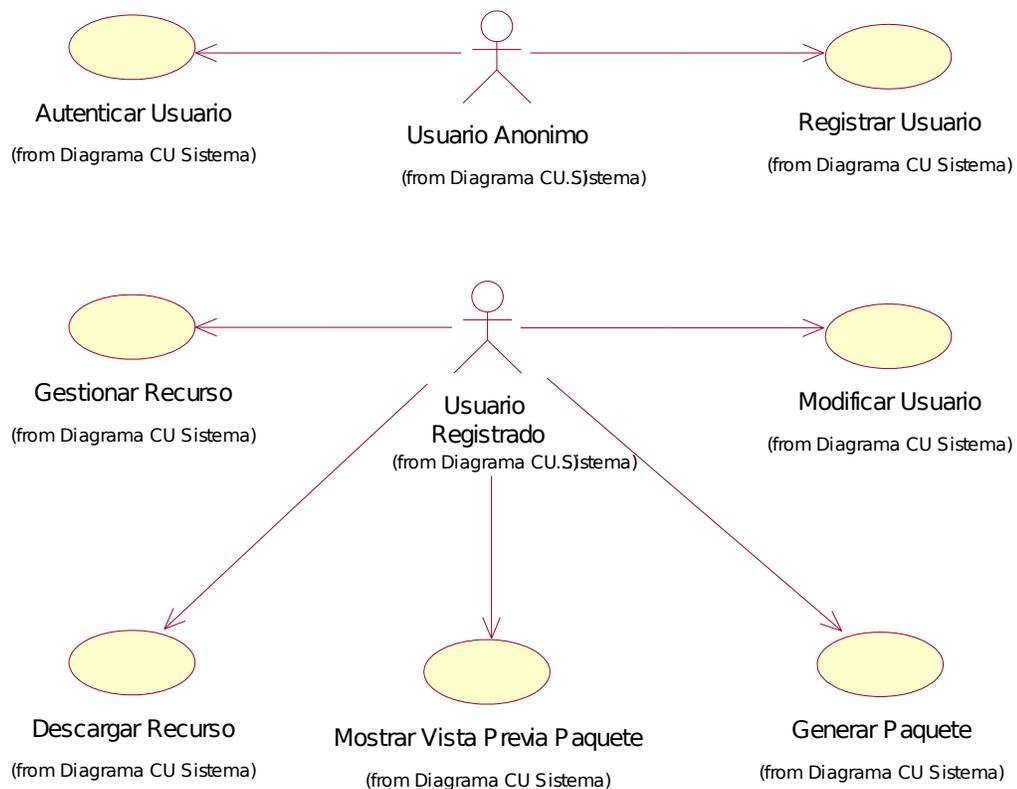
- El sistema se implementará con tecnología PHP.
- El PHP debe tener activa la extensión domxml.

- El sistema podrá utilizar cualquier tipo de Base de Datos que sea soportada por la librería de abstracción de acceso a datos ADODB.

Hardware

- Microprocesador 200 MHz.
- 32 MB de memoria RAM.
- 4 GB de disco duro.

2.7 Diagrama de caso de usos del sistema.



2.8 Descripción de los casos de uso.

Nombre del caso de uso	Registrar Usuario
Actores	Usuario Anónimo (Inicia).
Resumen	El caso de uso inicia cuando un usuario anónimo decide crearse una cuenta de usuario del sistema. Este proporciona los datos que el sistema le solicita,

	los cuales son chequeados y validados. Una vez hecha la comprobación se crea la nueva cuenta y se le envía un correo de comprobación con la contraseña. El usuario debe verificar este correo para poder activar su cuenta y empezar a usar el sistema. El caso de uso finaliza cuando el sistema ha creado la cuenta de usuario, se cierra la aplicación o se accede a otra opción.
Referencia	R1
Precondiciones	
Poscondiciones	Se debe haber creado el usuario y haber enviado un correo de confirmación con la contraseña.
Prototipo	
 <p>The screenshot shows a web interface for 'ROKS' with a registration form. On the left, there is a login section with fields for 'Usuario:' and 'Clave:', a 'Recuperar Clave' link, and an 'entrar' button. Below this is a 'Registrarse' button. On the right, under the heading 'Registrar Usuario', there are fields for 'Nombre y Apellidos:' and 'Correo electrónico:', and a 'registrar' button.</p>	

Nombre del caso de uso	Autenticar Usuario
Actores	Usuario Anónimo (Inicia).
Resumen	El caso de uso inicia cuando el usuario anónimo decide entrar al sistema y proporciona su usuario y contraseña. El sistema verifica que los datos entrados sean correctos y le da acceso, en caso contrario le muestra un mensaje de error. Si el usuario anónimo accede a la opción Recuperar Contraseña y proporciona su dirección de correo, en caso de existir en el sistema se le enviará una nueva clave sino se le mostrará un mensaje de

	error. El caso de uso finaliza cuando el usuario se ha autenticado o recuperado la clave; se cierra la aplicación o accede a otra opción.
Referencia	R2
Precondiciones	
Poscondiciones	El usuario debe haberse autenticado con éxito o recuperado la clave.
Prototipo	
	

Nombre del caso de uso	Modificar Usuario.
Actores	Usuario Registrado (Inicia).
Resumen	El caso de uso se inicia cuando el usuario registrado escoge la opción actualizar perfil. El sistema edita los datos del usuario, el cual hace los cambios que desee. En caso de escoger la opción eliminar usuario y entrado su contraseña el sistema verifica la misma y procede a eliminarlo, si la información no es válida se le mostrará un mensaje de error. El caso de uso finaliza una vez actualizados los datos, cuando se cierra la aplicación o se accede a otra opción.
Referencia	R3
Precondiciones	El usuario debe haberse autenticado. La cuenta del usuario debe ser de tipo manual.
Poscondiciones	Se deben haber actualizado los datos del usuario y en caso de haber actualizado el correo se le debe

haber enviado un correo de confirmación.

Prototipo

Nombre del caso de uso	Gestionar Recurso.
Actores	Usuario Registrado (Inicia).
Resumen	El caso de uso se inicia cuando el usuario registrado escoge la opción de gestionar recurso. Dentro del mismo puede subir, posicionar, actualizar o eliminar un recurso. Este finaliza cuando se han efectuado las acciones sobre el recurso, se cierra la aplicación o accede a otra opción.
Referencia	R3
Precondiciones	El usuario debe haberse autenticado. El recurso debe existir.
Poscondiciones	Debe haberse modificado el recurso.
Prototipo	



Hola Juenlis Coss

- [Área de trabajo](#)
- [Subir Recurso](#)
- [Subir Url](#)
- [Generar SCORM](#)

- [Editar Perfil](#)
- [Eliminar Usuario](#)
- [Salir](#)

Subir Recurso

Fichero:

General

Título:

Descripción:

Palabras Claves:

Derechos

copyright:

Nombre del caso de uso	Descargar Recurso.
Actores	Usuario Registrado (Inicia).
Resumen	El caso de uso se inicia cuando el usuario registrado escoge la opción de descargar un recurso, dándole la posibilidad de llevar el mismo a su máquina y finaliza luego de haber descargado el recurso, cerrado la aplicación o accedido a otra opción.
Referencia	R5
Precondiciones	Debe haberse autenticado. El recurso debe existir.
Poscondiciones	Debe haberse descargado el recurso.
Prototipo	



Hola Juenlis Coss

- [Área de trabajo](#)
- [Subir Recurso](#)
- [Subir Url](#)
- [Generar SCORM](#)

- [Editar Perfil](#)
- [Eliminar Usuario](#)
- [Salir](#)

Mostrar Recursos

	Título	Opción	Posición	
<input type="checkbox"/>	ejemplo.txt	<input type="radio"/> <input checked="" type="radio"/>	<input type="button" value="v"/>	Editar
<input type="checkbox"/>	ejemplo.txt	<input type="radio"/> <input checked="" type="radio"/>	<input type="button" value="v"/> <input type="button" value="^"/>	Editar
<input type="checkbox"/>	ejemplo.txt	<input checked="" type="radio"/> <input type="radio"/>	<input type="button" value="^"/>	Editar

Nombre del caso de uso	Mostrar Vista Previa Paquete.
Actores	Usuario Registrado (Inicia).
Resumen	El caso de uso se inicia cuando el usuario registrado escoge la opción Mostrar Vista Previa Paquete, escogiendo el diseño con el cual desea que se le visualice el contenido del paquete SCORM una vez subido a una plataforma de educación a distancia. Una vez mostrada la vista previa se finaliza con esto el caso de uso o una vez cerrada la aplicación o accedido a otra opción.
Referencia	R6
Precondiciones	Debe haberse autenticado. Deben existir recursos en el sistema.
Poscondiciones	Debe haberse mostrado la vista previa del paquete SCORM.

Nombre del caso de uso	Generar Paquete.
Actores	Usuario Registrado (Inicia).
Resumen	El caso de uso se inicia cuando el usuario registrado escoge la opción Generar Paquete, donde seleccionará el diseño con el que se mostrara el contenido del paquete SCORM. Una vez generado el paquete es descargado a la máquina local del usuario, concluyendo con esto el caso de uso.
Referencia	R7
Precondiciones	Debe haberse autenticado. Deben existir recursos en el sistema.
Poscondiciones	Debe haberse descargado el paquete SCORM.
Prototipo	



Hola Juenlis Coss

- [Área de trabajo](#)
- [Subir Recurso](#)
- [Subir Url](#)
- [Generar SCORM](#)

- [Editar Perfil](#)
- [Eliminar Usuario](#)
- [Salir](#)

Generar SCORM

General

Título:

Descripción:

Palabras Claves:

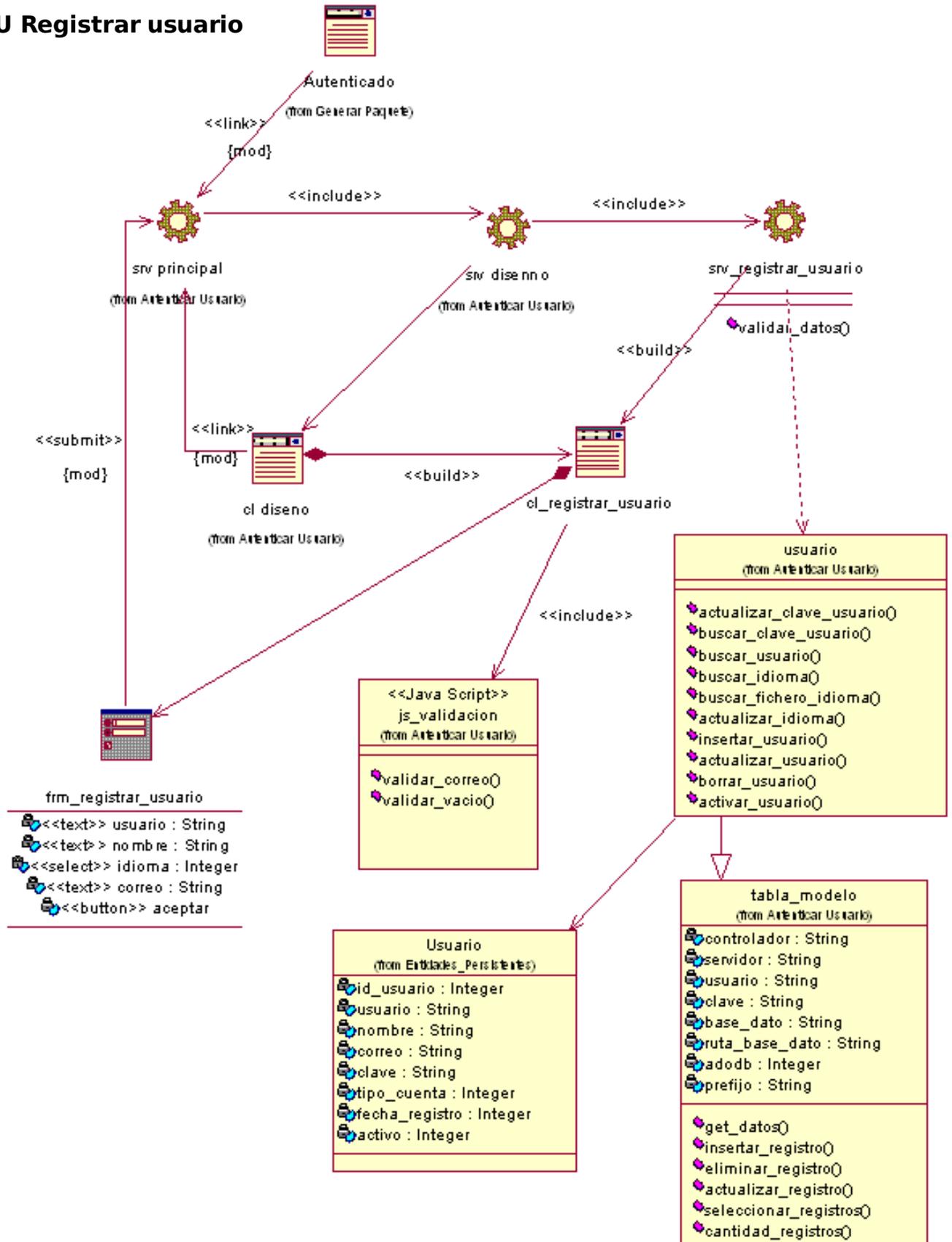
Derechos

copyright:

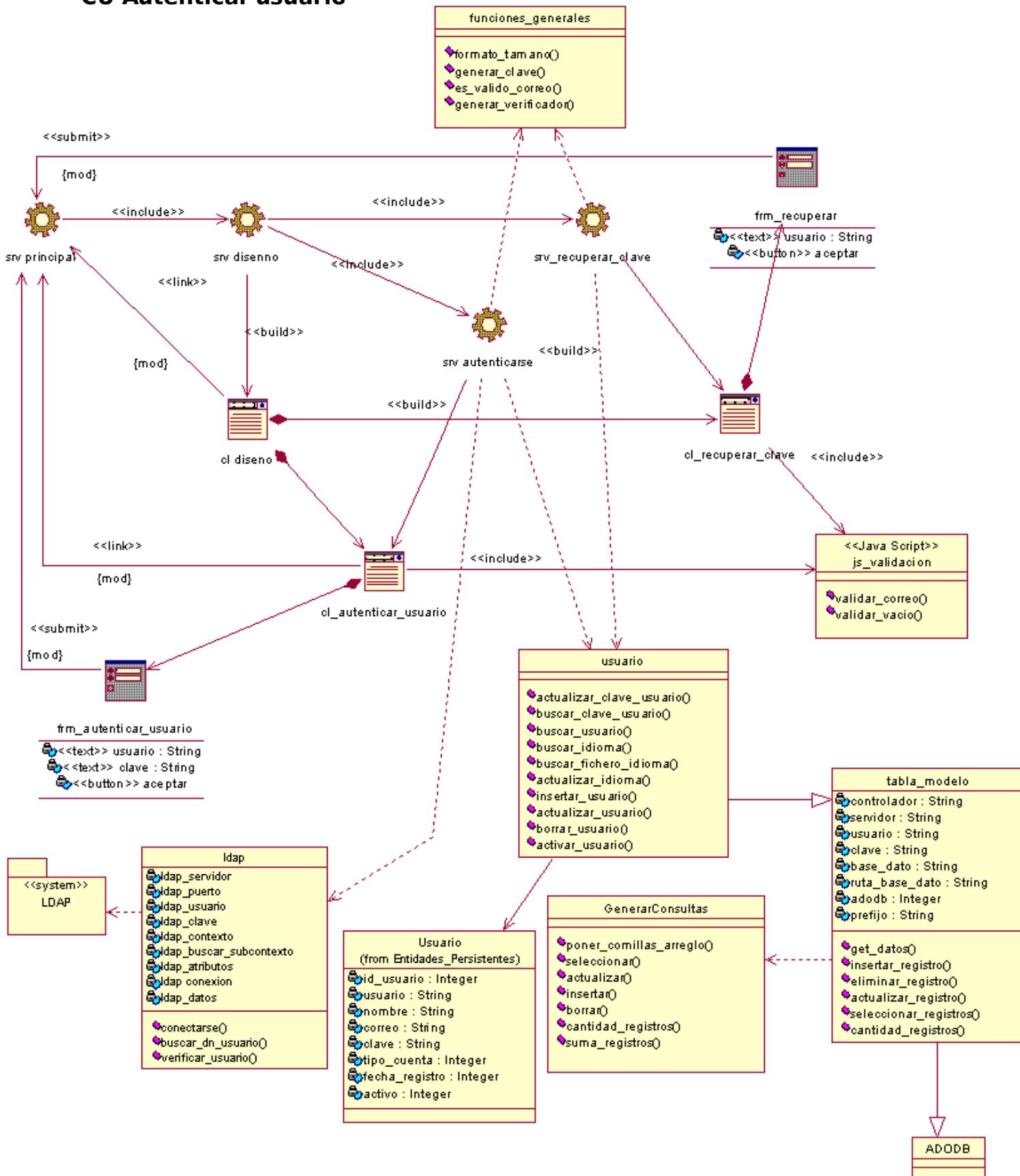
Generar SCORM

2.9 Diagramas del diseño.

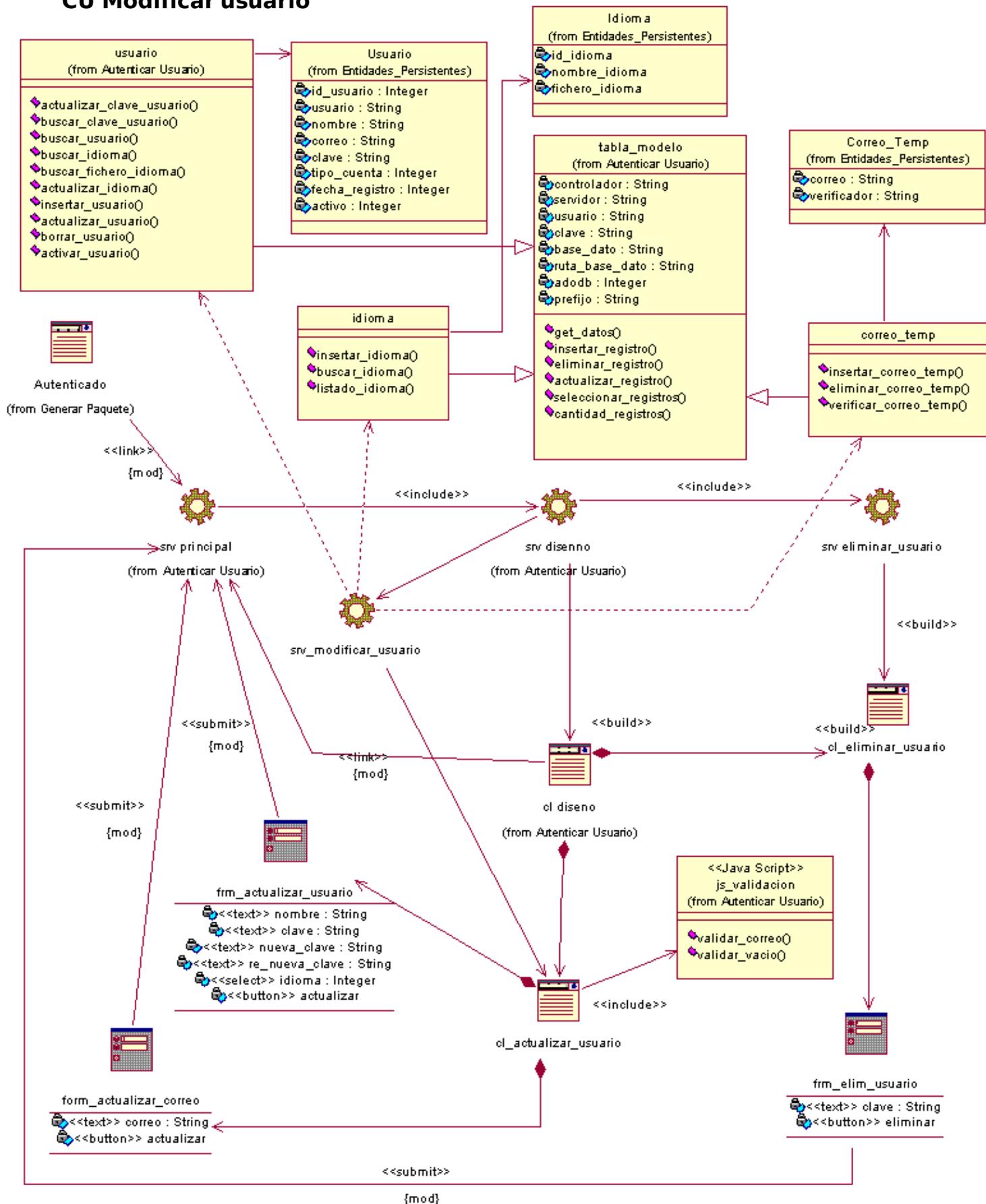
CU Registrar usuario



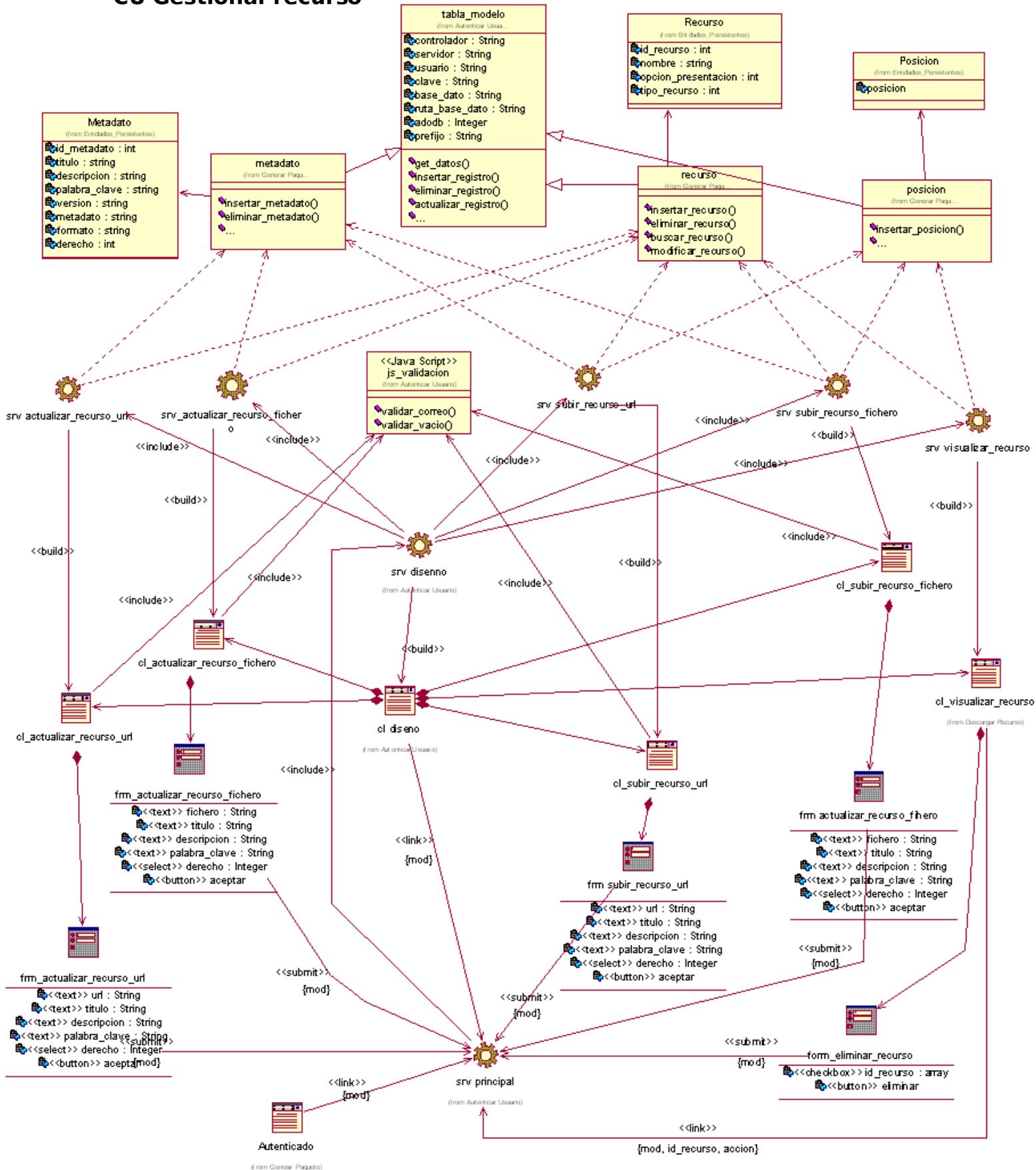
CU Autenticar usuario



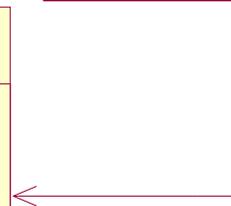
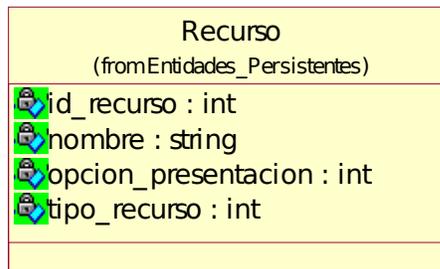
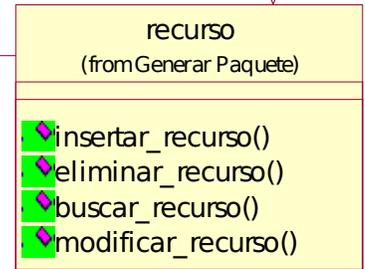
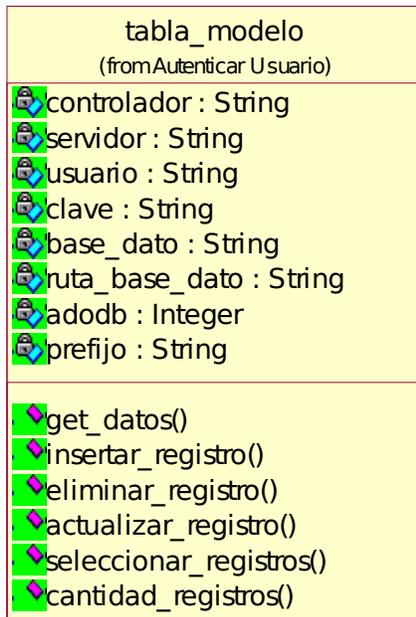
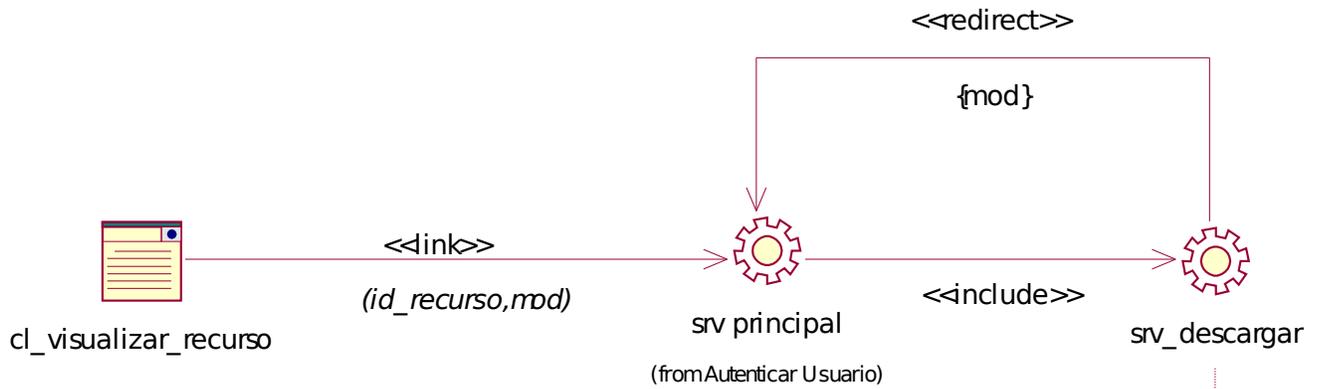
CU Modificar usuario



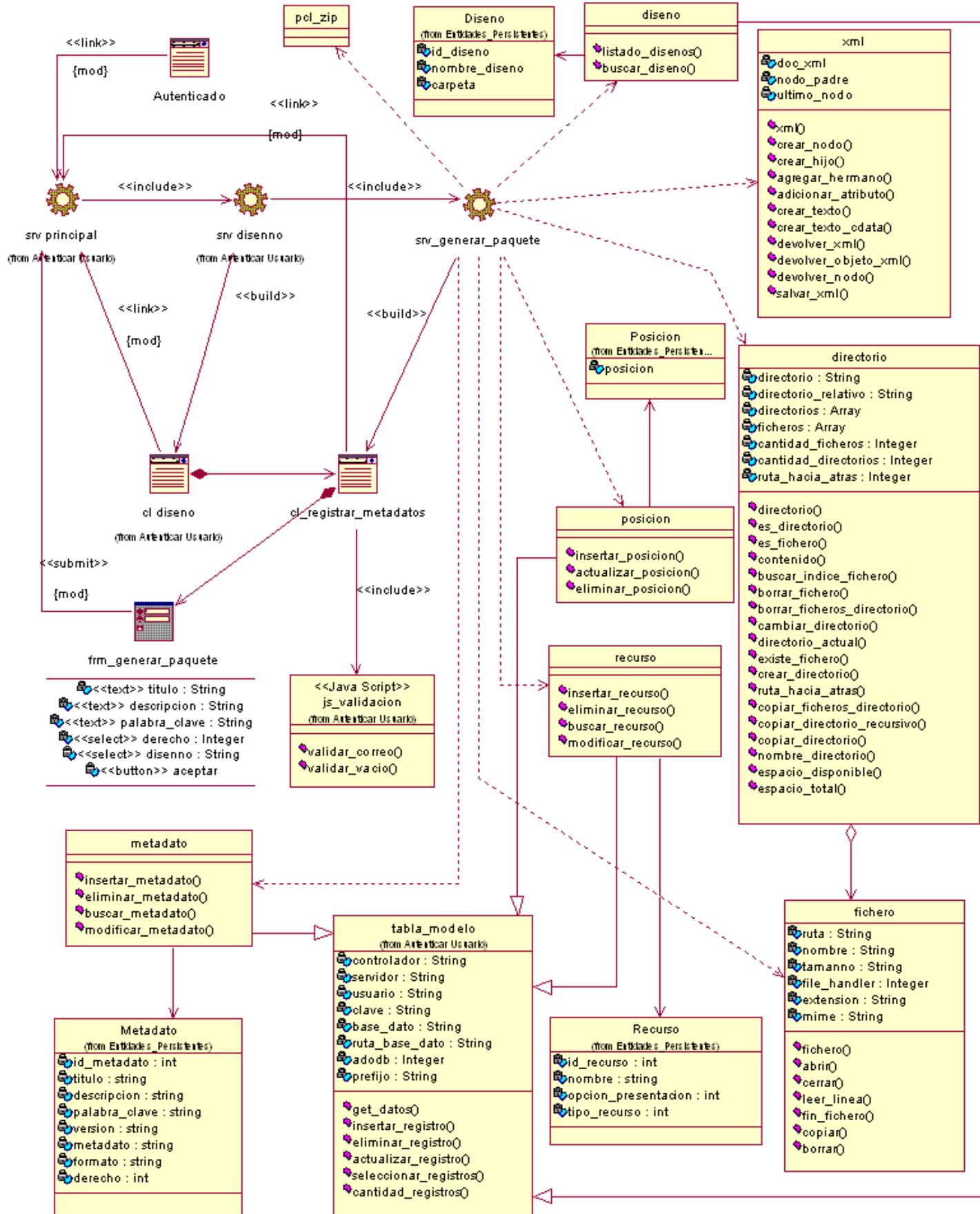
CU Gestionar recurso



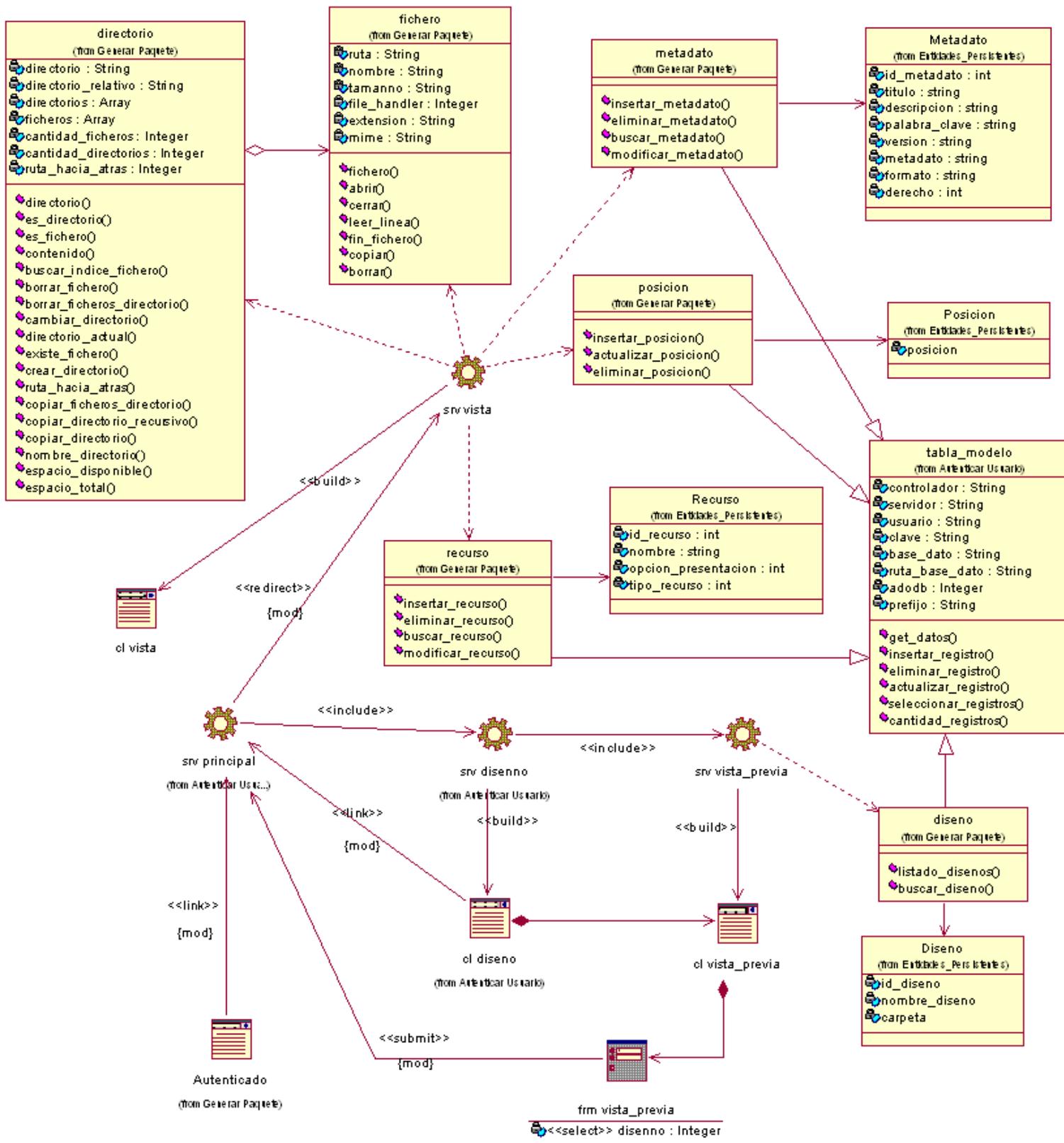
CU Descargar recurso



CU Generar paquete



CU Mostrar Vista Previa Paquete



2.10 Diseño de la interfaz.

El diseño de la interfaz es otro de los puntos fundamentales a tratar a la hora de la presentación de la aplicación teniendo en cuenta que esa es la capa de presentación al usuario y por lo tanto debe ser lo más amigable y comprensible posible.

En el diseño de las pantallas se tuvo en cuenta varios aspectos como: Organización de los elementos en la pantalla, dónde se coloca la información y cómo se estructura.

Las páginas de la interfaz serán diseñadas de forma uniforme garantizando:

- El equilibrio en la organización de la información, por ejemplo, que todas las páginas que muestran información siempre la mostrarán en el mismo orden.
- Diseño flexible que le permita a los usuarios ajustar el tamaño de la página combinando tamaños en píxeles y por porcentos.
- La optimización de la cantidad de elementos en la pantalla, ayudando al fácil manejo y mejor comprensión de la información mostrada en pantalla.
- La unidad, donde cada elemento de la pantalla se diseñará siguiendo un patrón de tamaño, colores y formas, utilizando para ello hojas de estilos.

No se utilizarán colores fuertes ni brillantes para la comodidad de los usuarios de las aplicaciones que interactuarán con el sistema. Los elementos que se repitan en las distintas páginas, se situarán en un mismo lugar para mejor manejo de la información. También se trabajará sobre la base de que las páginas no se encuentren muy cargadas, minimizando el uso de imágenes, y poniendo solo la información necesaria para mayor claridad.

2.11 Tratamiento de errores.

El tratamiento de los errores es un paso fundamental para el buen funcionamiento de un sistema, garantizando la seguridad, armonía y facilidad de uso de la aplicación.

En el sistema la validación para el tratamiento de errores serán referidos principalmente a errores de la interacción del usuario con la aplicación, que son aquellos en los que se puede incurrir a falta de conocimiento o experiencia en la explotación del sistema. Por lo que, antes de iniciar cualquier operación, el sistema verifica si el modulo al cual el usuario está accediendo existe.

Por otra parte, fue elegida una forma de tratar los errores de manera que las operaciones de inserción, eliminación y modificación de registros en la Base de Datos se realice de forma correcta, para lo cual, aprovechando las posibilidades de la interfaz gráfica, fueron establecidos mecanismos que visualicen la información dándole la opción de elegir o seleccionar la misma, evitando con esto que en la medida de lo posible el usuario juegue un papel más activo en la captación de información tratando de minimizar los errores que pueda tener la aplicación.

Evidentemente los errores ocurrirán incluso con los usuarios de más habilidad y experiencia, por lo que para el caso de los datos que sean adicionados por un usuario del sistema, se implementarán funciones que validen la entrada de los mismos y en caso de haber errores, se mostrarán mensajes que aclararen la inserción, eliminación o modificación errónea de algún dato.

Debido a la importancia y sensibilidad de los datos que se manejarán en la aplicación se hace necesario que se garantice la protección y seguridad de la información para el óptimo funcionamiento del sistema.

Para darle solución a este problema la seguridad en el sitio está implementada a través del servidor de base de datos y el uso de variables de sesión para restringir el acceso de los usuarios a determinadas páginas, garantizando con esto que determinada información presente en el sitio sólo se muestre a los usuarios registrados, así como se válida que la información que se le muestra al usuario sea de su propiedad.

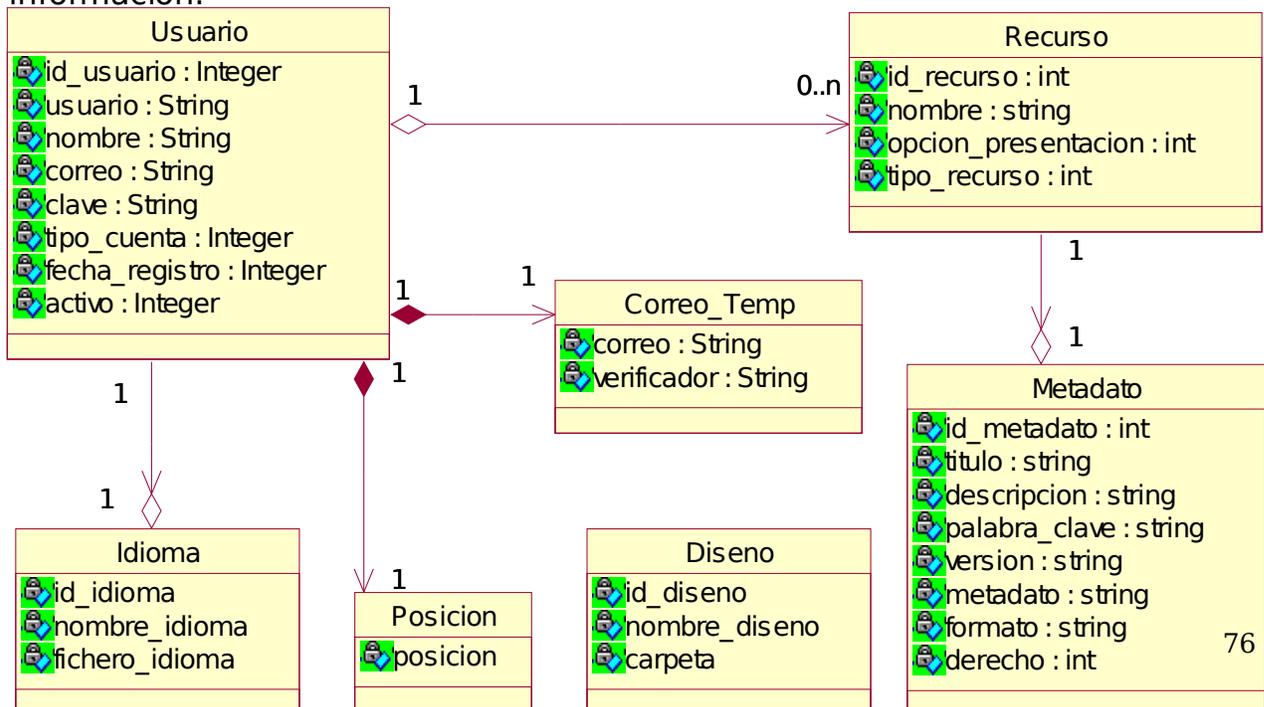
2.12 Ayuda.

El sistema es lo suficientemente claro y amigable para los usuarios del mismo, por lo que la utilización de una ayuda podría ser un aspecto secundario, aunque en el sistema existen textos aclarativos que describen la acción que se está realizando.

2.13 Diseño de la base de datos.

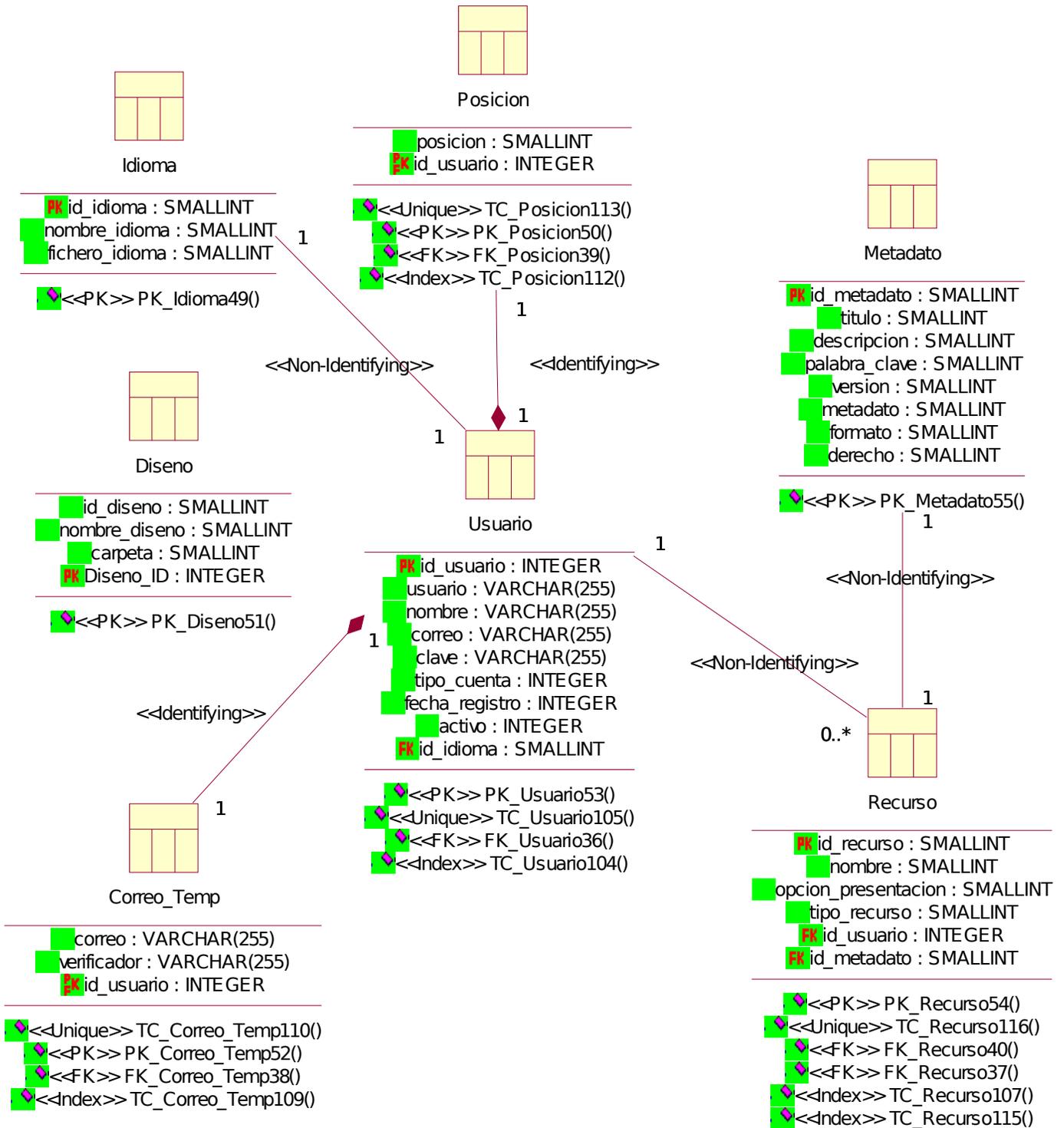
2.13.1 Diagrama de clases persistentes

Las clases persistentes son las clases que necesitan ser capaz de guardar su estado en un medio permanente, la necesidad de guardar su estado esta dado por al almacenamiento físico permanente de la información de la clase, para la copia de seguridad en caso del fracaso del sistema, o para el intercambio de información.



2.13.2 Modelo de datos.

El modelo de los datos describe la representación lógica y física de datos persistentes en el sistema.

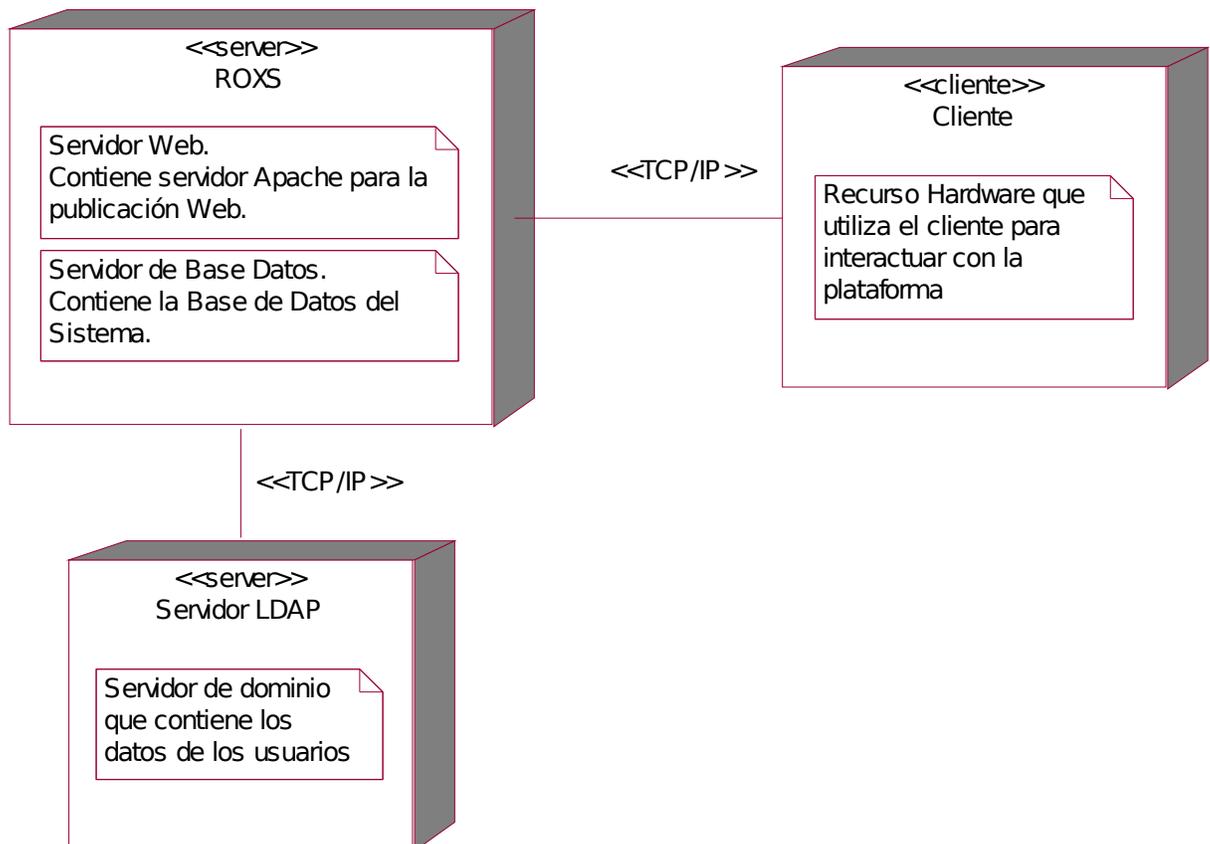


2.14 Diagrama de despliegue.

Los diagramas de despliegue se utilizan para razonar sobre la topología de procesadores y dispositivos sobre los que se ejecuta el software. En [Booch, 1999] se define un diagrama de despliegue como un diagrama que muestra la configuración de los nodos que participan en la ejecución y de los componentes que residen en ellos. Gráficamente, un diagrama de despliegue es una colección de nodos y arcos.

Mediante el diagrama de despliegue podemos ver cómo se encuentran relacionados físicamente los componentes de la aplicación.

En este sistema se está modelando que la base de datos se encuentre en el mismo servidor de la aplicación Web.



2.15 Conclusiones del capítulo.

En este capítulo se mostraron las principales clases del dominio del problema para un mayor entendimiento del negocio, se describieron propuestas de solución para el desarrollo del software a través de la muestra de varias vistas para llevar a cabo el proceso de análisis, diseño e implementación del sistema.

Estudio de factibilidad.

3.1 Introducción.

El estudio de factibilidad es un paso importante en la realización de un proyecto, pues brinda al equipo de trabajo información inicial relacionada con el costo del producto, tiempo estimado de desarrollo, cantidad de personas que intervienen, entre otros.

En este capítulo se abordarán aspectos relacionados con la estimación de esfuerzos (costes) de desarrollo del sistema. Por último se realizará una valoración de sostenibilidad del producto atendiendo a la dimensión socio-cultural, económica, ambiental y tecnológica.

3.2 Estimación del esfuerzo.

El método de estimación mediante el Análisis de Puntos de Casos de Uso es un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de pesos a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de estos factores.

Cálculo de Puntos de Caso de Uso sin Ajustar

El primer paso para la estimación consiste en el cálculo de los Puntos de Casos de Uso sin ajustar. Este valor se calcula a partir de la siguiente fórmula:

$$UUCP = UAW + UUCW$$

Donde:

UUCP: Puntos de Casos de Uso sin ajustar

UAW: Factor de Peso de los Actores sin ajustar

UUCW: Factor de Peso de los Casos de Uso sin ajustar

Factor de Peso de los Actores sin ajustar (UAW).

Para conocer este valor se hacen cálculos mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos.

Tipo Actor	Descripción	Peso	Actores	Total
Simple	Sistema con sistema a través de interfaz de programación.	1	0	0
Medio	Sistema con sistema mediante protocolo de interfaz basada en texto.	2	0	0
Complejo	Persona que interactúa con el sistema mediante interfaz gráfica.	3	2	6

Factor de Peso de los Casos de Uso sin ajustar (UUCW).

El valor se obtiene mediante un análisis de la cantidad de Casos de Uso existentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo.

Tipo de CU	Descripción	Peso	Cantida d	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	6	30
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	1	10
Complejo	El caso de uso tiene más de 8 transacciones.	15	0	0

Entonces: $UUCP = UAW + UUCW = 6 + 40 = 46$

Cálculo de Puntos de Casos de Uso ajustados.

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar éste valor mediante la siguiente ecuación:

$$UCP = UUCP \times TCF \times EF$$

Donde:

UCP: Puntos de Casos de Uso ajustados

UUCP: Puntos de Casos de Uso sin ajustar

TCF: Factor de complejidad técnica

EF: Factor de ambiente

Factor de complejidad técnica (TCF).

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante.

El Factor de complejidad técnica se calcula mediante la siguiente ecuación:

$$TCF = 0.6 + 0.01 \times \sum (\text{Peso}_i \times \text{Valor asignado}_i)$$

Factor	Descripción	Peso	Valor	Total
--------	-------------	------	-------	-------

T1	Sistema distribuido	2	0	0
T2	Tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	1	1
T4	Funcionamiento Interno complejo	1	1	1
T5	El código debe ser reutilizable	1	5	5
T6	Facilidad de instalación	0,5	3	1,5
T7	Facilidad de uso	0,5	4	2
T8	Portabilidad	2	5	10
T9	Facilidad de cambio	1	1	1
T10	Concurrencia	1	1	1
T11	Incluye objetivos especiales de seguridad	1	3	3
T12	Provee acceso directo a terceras partes	1	5	5
T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	1	1

$$TCF = 0,955$$

Factor de ambiente (EF)

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo.

$$EF = 1.4 - 0.03 \times \sum (\text{Peso}_i \times \text{Valor asignado}_i)$$

Factor	Descripción	Peso	Valor	Total
E1	Familiaridad con el modelo de proyecto utilizado	1,5	3	4,5
E2	Experiencia en la aplicación	0,5	2	1
E3	Experiencia en la orientación a objetivos.	1	5	5
E4	Capacidad del analista líder.	0,5	3	1,5
E5	Motivación.	1	4	4
E6	Estabilidad de requerimientos	2	2	4
E7	Personal Part-Time	-1	0	0
E8	Dificultad del lenguaje de programación	-1	1	-1

$$EF = 0,83$$

Entonces: $UCP = UUCP \times TCF \times EF = 76 \times 0,925 \times 0,68 = 36,4619$

De los Puntos de Casos de Uso a la estimación del esfuerzo.

El esfuerzo en horas-hombre viene dado por:

$$E = UCP \times CF$$

Donde:

E: esfuerzo estimado en horas-hombre

UCP: Puntos de Casos de Uso ajustados

CF: factor de conversión

Para calcular CF

CF = 20 horas-hombre (si $Total_{EF} \leq 2$)

CF = 28 horas-hombre (si $Total_{EF} = 3$ ó $Total_{EF} = 4$)

CF = abandonar o cambiar proyecto (si $Total_{EF} \geq 5$)

$Total_{EF} = Cant\ EF < 3$ (entre E1 -E6) + $Cant\ EF > 3$ (entre E7, E8)

$$Total_{EF} = 2 + 0 = 2$$

CF = 20 horas-hombre

$$E = UCP \times CF = 36,4619 \times 20 = 729,238 \text{ horas-hombre}$$

Este método proporciona una estimación del esfuerzo en horas-hombre contemplando sólo el desarrollo de la funcionalidad especificada en los Casos de Uso. Para estimar la duración total del proyecto, se distribuye el esfuerzo entre las diferentes actividades del proyecto.

Actividad	Porcentaje %	Horas-Hombres
Análisis	10	182,3095
Diseño	20	364,619
Implementación	40	729,238
Pruebas	15	273,46425
Sobrecarga (otras)	15	273,46425

actividades)		
Total	100	1823,095

Teniendo en cuenta que el tiempo de trabajo planificado para un mes es de 240 horas, el tiempo de desarrollo se estima en aproximadamente 8 meses, como participan 2 personas, en aproximadamente 4 meses se estima que finalice el desarrollo del proyecto.

Para la realización de este proyecto se requirió de un extenso período de investigación, durante el cual se realizó una amplia revisión bibliográfica, debido a que fue necesario realizar un estudio de los estándares conforme a los cuales se implementaría el sistema. Además se hizo un estudio de algunas de las herramientas de autor para crear paquetes SCORM para su explotación, como Reload Editor, Weload, Exelearning, Lompad.

3.3 Beneficios tangibles e intangibles

La formación continuada hoy en día es una necesidad no sólo en el ámbito personal sino también en el ámbito empresarial. Es por esto que a pesar de que este trabajo no se desarrolla con fines comerciales, no se debe obviar dicha posibilidad. En este sentido es importante destacar el servicio que puede ofertarse a las empresas, impartiendo cursos de interés específico a través de esta plataforma.

Por otro lado, el desarrollo del software que se propone puede proporcionar una serie de beneficios en el proceso de enseñanza aprendizaje, al facilitar el impulso de nuevos métodos de enseñanza centrados en el estudiante que posibiliten el desarrollo de las potencialidades individuales, así como mecanismos que facilitan la creación de los contenidos educativos.

Además, el uso efectivo de esta herramienta tendría gran repercusión como apoyo al proceso docente-educativo. Esto debido a que proporciona una serie

de mecanismos que flexibiliza, en términos de tiempo y espacio, la interacción entre los implicados en el proceso.

3.4 Análisis de costos y beneficios

El desarrollo de un producto informático siempre tiene un costo. Este puede estar justificado por los beneficios tanto tangibles como intangibles que origina el mismo.

El costo que significa la implementación en este caso no es elevado si se tiene en cuenta las importantes contribuciones que trae consigo al desarrollo de e-learning en la Universidad.

Además, la tecnología utilizada para el desarrollo del sistema es totalmente libre, por tanto no es necesario incurrir en gastos en el pago de licencias de uso.

También contribuye al ahorro de un sinnúmero de recursos, como bibliografía plana, transporte, entre otros.

Por ello se plantea que es factible el desarrollo del mismo.

3.5 Valoración de sostenibilidad según la dimensión socio-cultural, económica, ambiental y tecnológica.

La explotación de una plataforma para el e-learning en la universidad, tiene una influencia positiva desde el punto de vista socio-cultural, debido a que potencia el acceso a la superación, ofreciendo una nueva alternativa y un sistema de amplia disponibilidad para la capacitación.

Esta modalidad de superación brinda a los estudiantes y profesores mucho más tiempo y flexibilidad en términos de plazos y desplazamientos, al no

desvincular a los implicados de sus actividades, aumentando el número de personas que pueden acceder a los cursos.

Además, en el momento en que la Educación Superior cubana asume el reto de la Universalización, esta plataforma puede contribuir positivamente en este proceso. En ello pueden influir los Joven Club de Computación y los recursos informáticos existentes en los centros educacionales de todos los municipios.

La plataforma contribuye además a desarrollar valores en los estudiantes, ya que se necesita un alto nivel de disciplina y autorregulación para lograr atravesar cada curso con éxito.

Por otra parte, en la realización de toda investigación se debe tener presente el impacto medioambiental del trabajo y de las soluciones propuestas, visto éste como el conjunto de alternativas positivas y/o negativas que sufren los elementos del entorno producto del funcionamiento y desarrollo.

Tomando como punto de partida este concepto, se puede plantear que la realización de este trabajo no impacta negativamente el entorno, exceptuando la pequeña e indirecta influencia que pueda ejercer el consumo de electricidad y recursos empleados en la investigación. La solución propuesta como resultado del análisis influye positivamente en el entorno, ya que evita la transportación y minimiza el empleo de materiales de oficina.

La interfaz está diseñada de forma que facilita la navegabilidad del sitio, con uso de colores claros y contrastados. Además de no se usarse recursos que contribuyan a desviar la atención de la actividad principal.

En cuanto a los beneficios económicos que este trabajo traerá consigo debemos plantear que resulta impreciso y en extremo engorroso arribar a un estimado fidedigno. Cabe destacar que contribuye al ahorro de una enorme cantidad de recursos materiales y financieros que se gastarían si los mismos cursos se impartieran de forma presencial. Es recomendable realizar previamente una identificación de las verdaderas necesidades que presentan

los profesionales que recibirán los mismos, y de la demanda existente de un conocimiento específico.

Por otro lado, la reutilización de los contenidos de aprendizaje, contribuye a la construcción de materiales educativos de calidad con menos esfuerzo. Como los objetos de aprendizaje son fáciles de acceder y compartir, aumenta el valor del material educativo al ser reutilizado en diversos contextos.

Sin embargo, la incorporación del e-learning tanto como apoyo a la docencia presencial, como en la educación a distancia, si se quiere que sea de buena calidad y efectiva, requiere grandes inversiones de tiempo y dedicación; capacitar a los equipos docentes; repensar programas, contenidos y la didáctica correspondiente; integrar una variedad de distintas capacidades que no son necesarias en la educación tradicional.

En cuanto al mantenimiento del producto, debido al uso de estándares se facilita la interoperabilidad y la reusabilidad de los recursos, lo cual beneficia directamente la migración de datos y la implementación de nuevas funcionalidades.

Finalmente, se considera que la solución propuesta contribuye al desarrollo, con un impacto positivo en las distintas esferas involucradas.

3.6 Conclusiones del capítulo

En este capítulo se analizó la factibilidad de realización del sistema, se calculó el costo de producción del mismo, el tiempo que se estimó en que debía estar listo, el esfuerzo que debía realizar el equipo de desarrollo y la cantidad de personas necesarias para la realización del sistema, permitiendo confirmar la factibilidad de la construcción del sistema propuesto; y por último se concluyó que el sistema influye positivamente en el desarrollo sostenible.

Conclusiones Generales

En este trabajo se analizaron aspectos relacionados con las tendencias actuales del e-learning y la necesidad de crear y gestionar objetos de aprendizaje reutilizables, basados en las especificaciones SCORM, para que puedan ser manipulados por plataformas que soporten el estándar. Con la realización de esta investigación se arriba a las siguientes conclusiones:

- El sistema emplea los estándares internacionales establecidos para la implementación de herramientas de autor.
- Para su desarrollo se utilizan las técnicas más recientes y se trabaja con la filosofía cliente servidor.
- La interfaz es intuitiva, el ambiente de trabajo es cómodo y cumple los estándares de usabilidad y accesibilidad establecidos por la W3C

Como resultado del presente trabajo se logró desarrollar una aplicación web que facilita la creación de objetos de aprendizaje reutilizables para ser usados en plataformas de educación a distancia y repositorios de objetos de aprendizaje. Se considera que han sido cumplidos los objetivos planteados.

Recomendaciones

Con vistas al desarrollo futuro de este proyecto se recomienda:

- Incentivar la creación de nuevas herramientas para continuar desarrollando el e-learning en la Universidad de las Ciencias Informáticas.
- Utilizar los estándares internacionales para el desarrollo de este campo de investigación.
- Utilizar el sistema propuesto como apoyo al proceso docente en la Universidad de las Ciencias Informáticas.
- Ampliar las funcionalidades del sistema desarrollado y del repositorio de objetos de aprendizaje ROA para lograr la comunicación entre ambos.
- Desarrollar una versión Desktop portable.

Referencias bibliográficas.

[1] **Aedo Cuevas, Ignacio; Santacruz Valencia, Liliana Patricia; Delgado Kloos, Carlos.** *ELO: entorno para la generación, integración y reutilización de objetos de aprendizaje.*

<http://www.elearningworkshops.com/docs/scom/elo-doc30.pdf>.

(1/03/05)

[2] **Álvarez, Miguel.** *¿Qué es ASP?*

<http://www.desarrolloweb.com/articulos/393.php>. (05/03/06)

[3] **Álvarez, Miguel.** *¿Qué es JSP?*

<http://www.desarrolloweb.com/articulos/831.php?manual=27>.

(05/03/06)

[4] **Álvarez, Miguel.** *¿Qué es Perl?*

<http://www.desarrolloweb.com/articulos/541.php>. (05/03/06)

[5] **Álvarez, Miguel.** *¿Qué es XML?*

<http://www.desarrolloweb.com/articulos/449.php>. (05/03/06)

[6] **Andrade, Carmen; García, Ludi.** *Ventajas del Software Libre.*

<http://www.lambdau.com>. (15/01/06)

[7] **AulaGlobal.** *Bases del elearning - Que es el elearning?.*

<http://www.aulaglobal.net.ve/observatorio/articles.php?lng=es&pg=7>

[7](http://www.aulaglobal.net.ve/observatorio/articles.php?lng=es&pg=7). (27/02/06)

[8] **AulaGlobal.** *Estandares – ADL SCORM a profundidad.*

<http://www.aulaglobal.net.ve/observatorio/articles.php?lng=es&pg=1>

[03](http://www.aulaglobal.net.ve/observatorio/articles.php?lng=es&pg=1). (12/05/06)

[9] **Cáceres, Oriol; Madrazo, Alejandro.** *Breve historia de Linux.* **Open Source La Revista del Software Libre en Cuba, No 3, Febrero 2005.**

[10] **Colectivo de autores.** *Historia de PHP y proyectos relacionados.*
<http://www.php.net/manual/es/history.php>. (21/03/06).

[11] **Fahnle, Pablo.** *¿Qué es ASP.net?*
http://www.programacion.com/asp/articulo/aspnet_quees/. (07/04/06)

[12] **Foundation, Wikimedia** *Aplicacion Web.*
http://es.wikipedia.org/wiki/Aplicacion_web. (12/12/05)

[13] **Foundation, Wikimedia.** *Internet.*
<http://es.wikipedia.org/wiki/Internet> (12/12/05)

[14] **Foundation, Wikimedia** *Internet Information Services.*
<http://es.wikipedia.org/wiki/IIS>. (06/04/06)

[15] **Foundation, Wikimedia.** *Servidor HTTP Apache.*
http://es.wikipedia.org/wiki/Servidor_HTTP_Apache. (21/01/06)

[16] **Foundation, Wikimedia.** *Servidor web.*
http://es.wikipedia.org/wiki/Servidor_web (06/04/06)

[17] **Foundation, Wikimedia.** *Sistema de gestión de base de datos.*
<http://es.wikipedia.org/wiki/SGBD>. (21/01/06)

[18] **García Peñalvo, Francisco; López Guzmán Clara, Pernías Peco, Pedro.** *Desarrollo de repositorios de objetos de aprendizaje a través de la reutilización de los metadatos de una colección digital: de Dublin Core a IMS.*
RED. Revista de Educación a Distancia, número monográfico II, febrero 2005.

- [19] **González-Barahonda, Jesús.** *Software libre en la enseñanza informática.* **Todo Linux. No 8. Mayo 2001.**
- [20] **Jacobson, I.; Booch, G.; Rumbaugh J.** *El proceso unificado de desarrollo de software.* **Felix Varela, La Habana 2004.**
- [21] **Lafuente, Eloy.** *Acerca de Moodle.*
http://docs.moodle.org/es/Acerca_de_Moodle. (05/06/06)
- [22] **Palos, Juan Antonio.** *Servlets y JSP.* <http://java.sun.com> (27/03/06)
- [23] **Pocalles, Joseph.** *¿Cuáles son las ventajas del software libre para las PYMES?* <http://www.winred.com>. (06/04/06)
- [24] **Reino, Alfredo.** *Introducción a XML.*
<http://sunsite.unam.mx/archivos/xml/IntroXMLc.pdf> (30/03/06)
- [25] **Saborit Ramírez, Yúnier; Soto López, Nilet María.** *Propuesta para un sistema de catalogación y recuperación de recursos de información.* **Trabajo de diploma para optar por el título de Ingeniero Informático. Instituto Superior Politécnico “José Antonio Echeverría”, Ciudad de la Habana, junio 2004.**
- [26] **Salvador Sánchez, Alonso; Sanjuán Martínez, Oscar.** *Reusabilidad de objetos didácticos mediante el uso de genericidad.*
<http://www.pucp.edu.pe/eventos/sisoft/trabajosacep.htm>. (1/03/05)
- [27] **Toledano, Israel; Labrada Israel.** *Sistema informativo de recursos humanos.* **Trabajo para optar por el título de ingeniero informático, Universidad de Holguín, Holguín, julio 2005**
- [28] **Torres, A; Méndez, L.** *Sistema de Promoción y Gestión Comercial para la oficina de Transferencia Tecnológica de la Universidad de Cienfuegos.* **Trabajo de Diploma para optar por el título de Ingeniero Informático,**

Instituto Superior Politécnico “José Antonio Echeverría”, Ciudad de la Habana, junio 2005.

[29] **Uriarte Romo, Jesús; Gómez Benítez, Manuel.** *e-Learning perspectivas de las plataformas que lo soportan.*

<http://www.elearningworkshops.com/docs/estrategia/lcmsUPVcastellano.pdf> (01/03/06)

[30] **Valido, Y. y Moreira, Y.** *SAIMM: Sistema de Apoyo Integral a la Misión Milagro.* **Trabajo de Diploma para optar por el título de Ingeniero Informático, Instituto Superior Politécnico “José Antonio Echeverría”, Ciudad de la Habana, junio 2005.**

[31] **Varas, María Leonor.** *Repositorios de Objetos de Aprendizaje.*

http://www.alejandria.cl/recursos/documentos/documento_varas.doc (1/03/05)

Bibliografía

ADLNet. *The SCORM Content Aggregation Model, Advanced Distributed Learning Initiative 2001.* <http://www.adlnet.org/>. (06/03/06)

ADLNet. *The SCORM Content Aggregation Model, Advanced Distributed Learning Initiative 2006.* <http://www.adlnet.org/>. (06/03/06)

Auckland, Universidad de. *About Exelearning.*
<http://exelearning.org/?q=about> (28/03/06)

Auckland, Universidad de. *El redactor del contenido aprendizaje electrónico eLearning XHTML editor.*

http://exelearning.org/files/Spanish_exe_DLE_A4.pdf. (28/03/06)

AulaGlobal. *Objetos de Aprendizaje - SCO y Modelo de Contenidos .*

<http://www.aulaglobal.net.ve/observatorio/articles.php?lng=es&pg=89>. (27/02/06)

Barales, Juan Pablo. *Gateway para el reciclaje de Sistemas E-learning no conformes a SCORM. Trabajo Final de la carrera de grado (Analista en Computación), Universidad Nacional de Rio Cuarto, Argentina, abril 2005.*

Cádiz, Universidad. *Manual para la creación de Paquetes SCORM y su importación a Moodle.*

http://virtual.uca.es/portalFormacion/docs/portaFormacion/carpetaLocal2/manual_scorm.pdf?action=download. (28/03/06)

Dieguez, Jorge. *Nuevas versiones de las herramientas Reload.*

<http://www.elearningworkshops.com/modules.php?name=News&file=article&sid=368>. (05/03/06)

Draft Standard for Learning Object Metadata. Learning Technology Standards Committee of IEEE, USA, 2002.

García, Juan Egea. *Reload.*

<http://www.um.es/atica/gat/tdm/reload/sesion1.pdf> (28/03/06)

García, Pedro J. *Implantación de dotLRN en la Universidad de Valencia.*
Proyecto fin de carrera, Universidad de Valencia, julio 2004

Guerrero, L. A. *Proceso Unificado de Desarrollo de Software.*

<http://www.dcc.uchile.cl/~luguerre/cc51h/clase23.html>. (04/04/06).

Jaume I, Universitat. *Directrices CanCore 2.0.*

<http://cent.uji.es/octeto/node/1092>. (29/03/06)

Lim, John. *ADODB. Database Abstraction Library for PHP (and Python).*

<http://adodb.sourceforge.net/>. (05/04/06)

Mariño, Olga. *Objetos de aprendizaje, escenarios de aprendizaje y gestión de conocimiento.*

http://www.colombiaaprende.edu.co/html/docentes/1596/articles-81108_archivo.pdf. (28/03/06)

Ortiz, Fernando. *Librería ADODB para PHP.*

<http://www.lacorona.com.mx/fortiz/adodb/>. (05/04/06)

Romero, Daniel Omar. *Gateway para el Reciclaje de Sistemas E-learning que no cumplen con SCORM.*

http://www.ateneonline.net/datos/65_03_Romero_Daniel.pdf.
(28/03/06)

Tamayo, Daymy; Querejeta, Andrey. *Plataforma para el desarrollo e-learning en la Universidad de Holguín "Oscar Lucero Moya". Trabajo para optar por el título de ingeniero informático, Universidad de Holguín, Holguín, junio 2005*

WORKSHOPS , eLearning. *Recursos Comunidad eLearning WORKSHOPS.*

http://www.elearningworkshops.com/modules.php?name=Web_Links&I_op=viewlink&cid=32&orderby=dateA. (27/02/06)

Glosario de siglas y términos.

A

Apache: Servidor HTTP de páginas Web desarrollado por la Apache Software Foundation.

Aplicación: Es el programa que el usuario activa para trabajar en el ordenador. Existen muchos programas de ordenador que pueden clasificarse como aplicación. Generalmente se les conoce como Software.

B

Browser: Navegador. Herramienta para “Visitar” o visualizar el contenido de las páginas Web y sitios FTP en Internet.

C

Content Management Systems (CMS) o Sistemas de Gestión de Contenidos: Aplicaciones software que en la industria de las publicaciones online permiten la generación de los sitios web dinámicos.

D

DARPA: *Defense Advanced Research Project Agency*. Agencia para proyectos de investigaciones avanzadas de Defensa. Entidad donde surgió Internet.

E

Especificación: Es un documento técnico que describe los componentes (parte estática) y el comportamiento (parte dinámica) de un determinado sistema.

Estructura cliente / servidor: Consiste en que los clientes piden que una tarea sea realizada y el servidor realiza dicha tarea y regresa la información al cliente a través de la red.

e-Learning: Abarca al conjunto de las metodologías y estrategias de aprendizaje que emplean tecnología digital o informática para producir,

transmitir, distribuir, y organizar conocimiento entre individuos, comunidades y organizaciones

F

FSF: *Free Software Foundation*. Fundación del Software Libre.

FTP: *File Transfer Protocol*. Protocolo de Transferencia de Ficheros.

G

GNU: Gnu is Not Unix. Proyecto de la Fundación del Software Libre para crear un sistema UNIX libre.

GPL: *General Public License*. Licencia Pública General. Licencia utilizada por el movimiento del software libre en el mundo para protegerse de las patentes comerciales.

H

Herramienta Case: Ingeniería de sistemas asistida por ordenador (Computer-Aided Systems Engineering - CASE) es la aplicación de tecnología informática a las actividades, las técnicas y las metodologías propias de desarrollo de sistemas. Su objetivo es automatizar o apoyar una o más fases del ciclo de vida del desarrollo de sistemas.

HTTP: (HyperText Transfer Protocol; Protocolo de transferencia de hipertexto) Es un protocolo del nivel de aplicación usado para la transferencia de información entre sistemas, de forma clara y rápida. Ha sido usado por el World-Wide Web desde 1990.

I

Intranet: Red privada, desarrollada dentro de una compañía que utiliza el mismo software y provee de información similar que Internet, solo que es únicamente para el uso interno.

M

Metadatos: Conjunto de atributos o elementos necesarios para describir un recurso en cuestión.

O

Objetos de aprendizaje: materiales digitales creados como pequeñas piezas de contenido o de información, con la finalidad de maximizar el número de situaciones educativas en que las que el recurso pueda ser utilizado.

Open Source: *Código abierto*. Manera de nombrar también a las aplicaciones desarrolladas bajo el amparo de un producto con licencia GPL.

R

Red local: Se trata de una red de comunicación de datos geográficamente limitada, por ejemplo, una empresa.

T

TCP/IP: Sistema de protocolos, definidos en RFC 793, en los que se basa buena parte de la comunicación de Internet. TCP/IP es el estándar de protocolo de comunicaciones requerido por las computadoras que acceden a Internet.

TICs: Tecnologías de la Informática y las Comunicaciones.

Anexos

Anexo 1. Diagrama de Gantt

