

UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE INFORMÁTICA



PROYECTO DE SISTEMAS INFORMÁTICOS

**E-LEARNING PERSONALIZADO EN FUNCIÓN DE
ESTILOS DE APRENDIZAJE Y UTILIZANDO
ESTÁNDARES EDUCATIVOS EMERGENTES**

PROFESOR DIRECTOR: PILAR SANCHO THOMAS

**AUTORES: RAÚL IGLESIAS GONZÁLEZ
ANTONIO ORTIZ-REPISO VILLEGAS
RAMÓN PICAZO SOTOS**

Curso 2005-2006

Autorizamos a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria, como el código, la documentación y/o el prototipo desarrollado.

Firmado:

Raúl Iglesias González

Antonio Ortiz-Repiso Villegas

Ramón Picazo Sotos

Índice

1	Resumen	5
1.1	Resumen en español	5
1.2	Resumen en inglés	5
2	Introducción.....	6
2.1	E-Learning: Pasado y futuro.....	6
2.2	Estándares E-Learning.....	8
2.2.1	Tipos de estándares.....	8
2.2.2	¿Por qué hemos escogido IMS Learning Design para nuestro sistema?	10
2.3	IMS Learning Design	11
2.4	Tecnologías.....	18
2.4.1	Java	18
2.4.2	JavaServer Pages (JSP).....	20
2.4.3	JavaScript	21
2.4.4	Html	21
2.4.5	XML	22
2.4.6	Reload Editor.....	23
2.4.7	Macromedia DreamWeaver.....	24
2.4.8	Photoshop	24
2.4.9	MySQL.....	25
2.4.10	Apache / Tomcat.....	26
2.4.11	Rational Rose.....	27
2.5	Estilos de Aprendizaje	28
2.6	Nuestro sistema, motivación y objetivos.....	30
3	Sistema desarrollado.....	33
3.1	Arquitectura del sistema	33
3.1.1	JAVA	33
3.1.2	HTML.....	42
3.1.3	JSP	44
3.1.4	MySQL	45
3.1.5	Apache / Tomcat.....	45
3.1.6	Diagramas UML.....	46
3.2	Funcionamiento interno del sistema	53

4	Sistemas de características similares.....	72
4.1	WebCT	72
4.2	Blackboard.....	74
4.3	Moodle.....	75
4.4	Otras herramientas	77
4.4.1	TelEduc.....	77
4.4.2	DotLRN	77
4.4.3	Claroline.net	78
4.4.4	Dokeos	78
4.5	Estándares.....	79
4.6	Herramientas con personalización.....	80
4.7	Conclusiones.....	81
5	Diferencias entre la especificación y nuestro sistema.....	82
6	Conclusiones	89
6.1	Conclusiones.....	89
6.2	¿Qué queda por hacer?	90
7	Glosario	93
8	Bibliografía.....	95

1 Resumen

1.1 Resumen en español

A pesar de los avances, sobre todo desde el punto de vista de interoperabilidad, que ha supuesto el desarrollo de estándares educativos, a día de hoy, las especificaciones existentes no prestan excesiva atención a la personalización de los recursos. Crear cursos y aplicaciones que proporcionen enseñanza personalizada es especialmente complejo en la web donde existe mucha información distribuida, creada con muy distintos propósitos y por tanto muy heterogénea.

El objetivo de este proyecto es diseñar un curso personalizado distribuido vía web, que utilizando el modelo de objetos educativos, tecnologías relacionadas con la web semántica y haciendo uso de la especificación “IMS Learning Design”, permita adaptar los contenidos de manera dinámica en función del perfil de adquisición del conocimiento que más se adapte a cada alumno. Se utilizará para la catalogación de los estilos de aprendizaje el modelo postulado por Felder-Silverman, que cataloga a los diferentes alumnos por su manera de adquirir el conocimiento en 4 dicotomías (Activos/Reflexivos, Sensitivos/Intuitivos, Visuales/Verbales y Globales/Secuenciales).

El curso que se propone desarrollar será una introducción a la programación en C++, de tal manera que, una vez catalogado un alumno en fase inicial, se le proponga un método concreto de formación. Pongamos por caso, un alumno activo aprende mejor si trabaja en base a ejemplos y simulaciones, y obtiene mejores rendimientos en trabajo en grupo, mientras que un alumno reflexivo trabaja mejor solo y pensando primero sobre textos y teoría antes de abordar trabajo práctico.

1.2 Resumen en inglés

In spite of the advances, mainly from the point of view of interoperability, that have supposed the development of educative standards, nowadays, the existing specifications don't pay the attention that deserves to the personalization of the resources. Creating courses and applications that provide customized education is specially complex in the Web, where there is a lot of distributed information which is created with very different intentions and very heterogenous.

The objective of this project is to design a customized course distributed via Web, using the model of educative objects, technologies related to the semantic Web and the specification "IMS Learning Design", which allows to adapt the contents by a dynamic way, depending on the learning style which fit better with each student. The model postulated by Felder-Silverman will be used to classify the students into four groups (Actives/Reflectives, Sensings/Intuitives, Visuals/Verbals and Globals/Sequentials).

The course to deploy will be an introduction into C++ programming, so once a student has been catalogued, a personalize course will be shown to him. For instance, an active student learns better if he works with examples and simulations, and gets better results working in a group, whereas a reflective student works better alone and thinking first on texts and theory before approaching practical work.

2 Introducción

2.1 E-Learning: Pasado y futuro

Hacia finales de los años noventa proliferaron en España (y en todo el mundo) portales especializados en la oferta de cursos por Internet, sobre todo en aquellas empresas y entidades que ya estaban dando formación. Por tanto, ofrecían temarios propios, muchas veces con materias desarrolladas anteriormente y para una formación presencial. Trataban de competir en el explosivo mercado de la formación que se avecinaba, como consecuencia del abaratamiento que significa impartir cursos por Internet y el acceso a todo tipo de alumnos y lugares (incluso países).

Las primeras experiencias fueron a veces traumáticas para formadores y alumnos. Los **principales problemas que se plantearon** fueron los siguientes:

- Los cursos no estaban actualizados, no aportando los ejercicios y casos prácticos que requerían los alumnos.
- El servicio de tutoría se delegaba en personal deficientemente cualificado, ya que no era necesaria su presencia en clase.
- No se ofrecían servicios tutoriales plenos, como puede ser la confección del Currículum del alumno, la resolución de casos presentados por este (muchos trabajan en empresas en donde se encuentran con problemas reales sobre los temas estudiados). Tampoco existían exigencias sobre la realización de ejercicios o el examen final.
- Las plataformas informáticas para la edición y actualización de los temarios así como para la gestión de los alumnos (matrícula, expediente, correcciones, puntuación, cobros, etc.) eran muy costosas y deficientes, lo que creaba problemas de seguimiento y aumento de los costes.
- Por parte de los usuarios no estaba extendido el uso de líneas ADSL o tarifas planas, lo que hacía tediosa y cara la formación vía Internet.

Sin embargo, como todo sector económico, este ha ido madurando poco a poco en los últimos años. Hoy en día, **lo que busca un futuro alumno en e-Learning y debe ofrecer una empresa de formación**, son las siguientes condiciones:

- Informarse con el máximo detalle del contenido de la acción formativa (a veces basta con un índice pormenorizado, que contenga todas y cada una de las materias que se verán durante el estudio) así como las características del curso (objetivos del mismo, metodología de estudio, condiciones de examen, precio total, etc.).
- Conocer de antemano el historial profesional del profesor (con nombre y apellidos) para evaluar su capacidad como tutor durante el estudio.
- Comprobar la posibilidad de proponer casos por parte del alumno o que podrá ampliar y consultar con el tutor materias de su interés a través de información complementaria, Internet, etc.
- Apoyo en la búsqueda de trabajo o preparación del currículum del alumno.
- Acreditación del certificado de estudios por parte de entidades oficiales o de reconocido prestigio.

- Cursos posteriores que puedan complementar su formación y servicio de bibliografía disponible.

Según reconocidos estudios lo que más valoran los alumnos es la posibilidad de comunicación continua y abierta con el tutor, lo que corrobora que e-Learning tiene más de auto estudio que de formación dirigida.

Actualmente todavía falla en esta modalidad el e-fulfillment, esto es, la capacidad de atención personalizada al alumno.

El futuro del e-Learning está asegurado como la fórmula más competitiva para ambas partes y será sin duda la vía de formación más extendida en el futuro, por los siguientes motivos:

- Los costes para el Centro por acción formativa son significativamente más bajos que una formación no virtual (actualmente se regalan cursos por Internet).
- La posibilidad de los alumnos para acceder desde cualquier punto geográfico y la inmediatez de la comunicación (ya que no se necesitan desplazamientos ni incurrir en los gastos derivados de los mismos).
- Cada alumno marca el ritmo de estudio, sin horarios ni calendarios.
- Curiosamente la atención es personal aunque sea virtual y por tanto suele ser más intensa.

Desde luego, en las grandes organizaciones descentralizadas con cientos y miles de centros de trabajo (Bancos, Corporaciones, Ministerios, etc.), este será el único medio de distribuir formación entre sus empleados de manera efectiva, rápida y barata.

2.2 Estándares E-Learning

2.2.1 Tipos de estándares

Institute for Electrical and Electronic Engineers Learning Technology Standards Committee (IEEE LTSC)

ltsc.ieee.org



El IEEE (que se lee como "i triple e") es un conocido cuerpo multinacional que desarrolla estándares internacionales para sistemas eléctricos, electrónicos, computacionales y comunicacionales. IEEE está organizado en distintos comités que se juntan y analizan las distintas tecnologías, entregando como resultado una especificación o recomendación en forma de estándar. Uno de éstos es el Learning Technology Standards Committee (LTSC) o Comité para los Estándares de la Tecnología del Aprendizaje.

La especificación más reconocida del trabajo de IEEE LTSC es la especificación de los Metadatos de los Objetos de Aprendizaje o Learning Object Metadata (LOM) que define elementos para describir los recursos de aprendizaje. IMS y ADL utilizan los elementos y la estructura de LOM en sus respectivas especificaciones.

AICC

www.aicc.org



Creado en 1988, AICC es un grupo internacional de profesionales del entrenamiento y capacitación basada en tecnología. A finales de la década de los 80's, Internet y la Web aún no existían, por lo tanto AICC se crea cuando sólo se hablaba del CBT (Computer-Based Training) o Capacitación Basada en Computadoras.

AICC publica varias guías, incluyendo algunas relacionadas con el hardware y software. Sin embargo, las que han tenido una mayor repercusión son las dedicadas a la Instrucción Administrada por Computadoras o CMI (computer-managed instruction).

Por ejemplo, la guía de AICC CMI001 entrega recomendaciones y lineamientos a seguir para lograr la interoperabilidad entre LMS. Siguiendo estas especificaciones AICC pueden desarrollarse contenidos que serán intercambiables entre distintos LMS que soporten y apliquen la Guía AICC CMI001.

Advanced Distributed Learning (ADL)

www.adlnet.org



ADL nace como una organización auspiciada por el gobierno de Estados Unidos (el consumidor más grande de e-Learning en el planeta), con el objetivo de investigar y desarrollar especificaciones que fomenten la adopción y el avance del e-Learning

La misión de ADL es clara y simple: Buscar mecanismos para asegurar educación y materiales de capacitación de alta calidad que puedan ajustarse a las necesidades de cada institución. Como consecuencia de estas investigaciones y recomendaciones, ADL se está transformando en un gran impulsor para convertir las especificaciones en estándares.

La publicación que más ha repercutido de ADL es el Modelo de Referencia de Objetos de Contenido Compartibles, o Shareable Content Object Reference Mode, conocido ya en el ambiente como SCORM.

La especificación SCORM logra combinar de excelente forma los elementos de IEEE, AIIC e IMS en un único documento consolidado de fácil implementación.

IMS

www.imsglobal.org



IMS (Instruction Management Systems) es un consorcio que agrupa a vendedores, productores, implementadores y consumidores de e-learning, y que se enfoca completamente a desarrollar especificaciones en formato XML. Entre sus miembros se encuentra Microsoft, Apple, ORACLE, WebCT, Blackboard y Boeing.

Las especificaciones IMS cubren un amplio rango de características que se persiguen hacer interoperables entre plataformas, que van desde los metadatos, la interoperabilidad de intercambiar el diseño instruccional entre plataformas, hasta la creación de cursos online para alumnos que tengan alguna discapacidad visual, auditiva u otra.

2.2.2 ¿Por qué hemos escogido IMS Learning Design para nuestro sistema?

La flexibilidad de la especificación IMS y la facilidad con la que se puede adecuar para llevar a cabo nuestro sistema de personalización del aprendizaje sin salirnos de los estándares (este proceso lo detallaremos más adelante) han sido algunas de las principales razones por las que hemos elegido dicha especificación.

Sin duda, otros motivos de suma importancia son el éxito de la especificación IMS en el mercado actual del e-learning, el que sea una especificación orientada al lenguaje XML, lo que proporciona un nivel de accesibilidad altísimo, ya que es el lenguaje estándar para intercambio de información entre aplicaciones más utilizado en Internet, o el increíble potencial que se puede desarrollar utilizando la especificación IMS de cara a la personalización del aprendizaje para cada alumno, lo que nos interesaba enormemente.

2.3 IMS Learning Design

Definición

Un Diseño de Aprendizaje (Learning Design) es definido en la especificación IMS como "*una descripción de un método que permite a los alumnos alcanzar ciertos objetivos de aprendizaje por medio del desarrollo de ciertas actividades de aprendizaje en un cierto orden en el contexto de un cierto ambiente de aprendizaje*".

IMS Learning Design

IMS (Instruction Management Systems) es un consorcio que agrupa a vendedores, productores, implementadores y consumidores de e-learning, y que se enfoca completamente a desarrollar especificaciones en formato XML. Entre sus miembros se encuentra Microsoft, Apple, ORACLE, WebCT, Blackboard y Boeing.

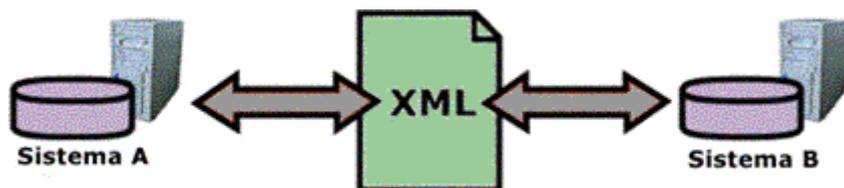
Las especificaciones IMS cubren un amplio rango de características que se persiguen hacer ínter operables entre plataformas, que van desde los meta datos, la interoperabilidad de intercambiar el diseño instruccional entre plataformas, hasta la creación de cursos online para alumnos que tengan alguna discapacidad visual, auditiva u otra cualquiera.

Dentro de las especificaciones que ya han sido liberadas y trabajadas por IMS se encuentran las siguientes:

- IMS Learning Resources Meta-data Specifications (Octubre 01, 2001) genera una forma uniforme para describir los recursos de aprendizaje de manera que éstos puedan ser fácilmente encontrados, por medio del uso de herramientas de búsqueda que sean capaces de interpretar estos meta datos.
- IMS Enterprise Specification (Julio 16, 2002) está dirigida a aplicaciones y servicios administrativos que necesitan compartir datos sobre los alumnos, cursos, rendimiento a través de sistemas operativos, plataformas, interfaces de usuario.
- IMS Content Packaging Specification (Agosto 10, 2002) hará más fácil crear objetos de contenido reutilizables que serán útiles en una variedad de sistemas de aprendizaje.
- IMS Question & Test Specification (Febrero 13, 2002) soluciona la necesidad de poder compartir ítems de exámenes y otras herramientas de evaluación a través de sistemas distintos.
- IMS Learner Profiles Specification (Marzo 18, 2003) buscará formas de organizar la información del alumno de manera que los sistemas de aprendizaje puedan ser más responsivos a las necesidades específicas de cada usuario.

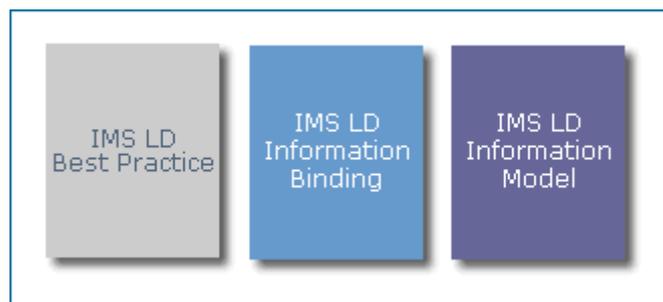
- IMS Reusable Competency Definition Specification (Octubre 25, 2002) define un modelo de información para describir, referenciar e intercambiar definiciones de competencias, principalmente en el contexto del aprendizaje online y distribuido. En esta especificación, la palabra competencia es utilizada en un sentido muy general que incluye habilidades, conocimientos, tareas y resultados del aprendizaje. Esta especificación entrega una manera de representar formalmente las características claves de una competencia independiente de su uso en un contexto en particular. Permite la interoperabilidad entre los sistemas de aprendizaje que manejan información de competencia entregándoles medios para referirse a definiciones comunes con significados comunes.
- IMS Learning Design Specification (Febrero 13, 2003) que orienta a describir el Diseño Instruccional y Diseño de Aprendizaje que acompañará a un curso online.
- IMS Guidelines for Developing Accessible Learning Applications (Abril 28, 2003) Guías desarrolladas por el Grupo de Trabajo sobre Accesibilidad de IMS entregará un marco de trabajo para las comunidades de aprendizaje distribuido. Este marco establecerá el escenario para las soluciones existentes, las oportunidades y posibilidades para implementarlas, y las áreas donde más desarrollo e innovación se necesitan en las tecnologías educacionales para asegurar que la educación es para todos, en cualquier lugar y en cualquier momento.
- IMS Digital Repositories (Enero 30, 2003) Busca integrar el aprendizaje online con los recursos de información a través de bodegas o depósitos digitales para almacenar colecciones digitales de documentos.
- IMS Simple Sequencing (Marzo 20, 2003) Especifica cómo los objetos de aprendizaje son ordenados y presentados a un alumno.

Todas las especificaciones de IMS, establecen a **XML** como el lenguaje seleccionado para realizar los intercambios de información entre plataformas, dado que XML nació justamente como el lenguaje para intercambiar datos a través de aplicaciones en Internet. IMS LD es un lenguaje para modelar unidades de estudio.



Las especificaciones IMS ya han sido incorporadas por bastantes LMS (Learning Management System), entre ellos WebCT, BlackBoard, EdCenter y LUVIT.

La documentación IMS LD (Learning Design) consiste de tres partes o subdocumentos:



- **IMS LD Best Practice:** Este documento es quizás el más narrativo de todos y su misión es brindar recomendaciones de cómo implementar o poner en práctica esta especificación.
- **IMS LD Information Binding:** Describe en detalle los elementos y atributos en XML que permitirán implementar IMS LD.
- **IMS LD Information Model:** Describe las estructuras de datos de esta especificación.

Un Learning Design modelado utilizando el lenguaje descrito en IMS LD, captura quién hace qué, cuándo y utilizando qué materiales y servicios para alcanzar objetivos de aprendizaje.

La especificación describe cómo realizar este modelamiento y brinda un mecanismo para hacerlo a través del XML. El proceso de describir cómo se expresa algo a través de XML, se conoce como "XML Binding" (ligadura XML). Utilizando esta ligadura se puede describir el proceso de aprendizaje.

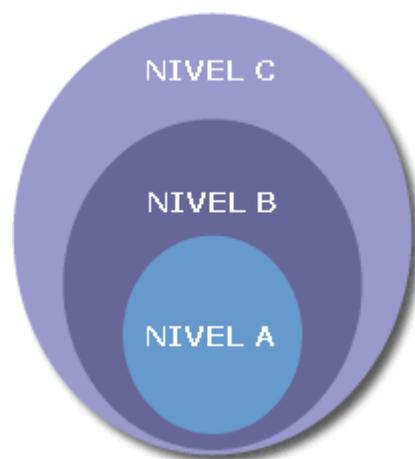
La idea es que un documento XML que describe un proceso de aprendizaje sea "cargado en" una aplicación que soporte IMS LD y se "reproduzca". Reproducir una instancia significa que una vez que a las personas les han sido asignados diferentes roles (por ejemplo, alumnos, tutores y administradores), las distintas actividades descritas en el documento se pueden desarrollar.

La relación entre el lenguaje IMS LD y un reproductor IMS LD es análogo a la relación HTML y navegador- ambos lenguajes necesitan ser interpretados por una aplicación.

Un reproductor de IMS LD puede ser una aplicación independiente o podría ser parte de un Virtual Learning Environment (VLE), Course Management System (CMS).

IMS LD está dividido en tres partes distintas para facilitar tanto la producción de la especificación como su subsiguiente implementación. Estas partes se conocen como Nivel A (Level A), Nivel B (Level B) y Nivel C (Level C).

Para cada nivel se ofrecen esquemas XML distintos, y los Niveles B y C se integran y extienden a nivel previo.



El **Nivel A** contiene la mayor parte de los elementos de IMS LD, incluyendo las actividades, ambientes, servicios, roles, actos, etc.

Las especificaciones actuales del IMS reflejan un modelo de un solo usuario, como único alumno, interactuando con el contenido y siendo examinado.

El Learning Design proporciona la capacidad de diseñar unidades de aprendizaje que simultáneamente incluyan varios roles que se pueden desempeñar por varios actores. Permite a las actividades ser especificadas en “flujos de aprendizaje” coordinados, que son análogos a los flujos de trabajo en grupo.

Apoya así el aprendizaje en grupo y colaborativo de diversas clases, la importancia de las cuales se reconoce cada vez más en el entrenamiento comercial y las esferas educativas. Puede todavía ser utilizado para apoyar el modelo del único alumno a través de la creación de una unidad de aprendizaje con un solo rol y sin ninguna interacción entre los otros alumnos. Si varios alumnos se asignan al mismo rol, cada uno de ellos trabaja con los recursos que le han sido asignados en solitario.

Como Learning Design separa **Actividades** de las **Estructuras de Actividades** y éstas de los **Roles** y **Recursos**, todos se convierten en componentes reutilizables. Se reúnen bajo el concepto de un **Método** (Method) que utiliza la estructura familiar de un **Play** con **Actos** (Acts) y **Role-Parts** en cada Acto.

Ciertos servicios también se especifican en la especificación del Learning Design. Un servicio proporciona una función general tal como un email, una comunicación, o un servicio de aviso que no se pueda especificar usando un URL en el tiempo del diseño, sino que por el contrario se puede especificar solamente por el sistema en tiempo de ejecución cuando el Learning Design es instanciado y se han asignado personas a los distintos roles.

El concepto de un servicio no se apoya actualmente en Content Packaging donde un URL fijo tiene que ser especificado en tiempo de diseño. Esta parte de la especificación tiene un XML Binding separado para poderla también utilizar independientemente del resto del Learning Design, por ejemplo como extensión al Secuenciamiento Simple.

El **Nivel B** añade **Propiedades** y **Condiciones** al Nivel A, lo que permite la personalización así como secuencias e interacciones más elaboradas basadas en los portafolios de los alumnos. Las propiedades pueden ser utilizadas para guiar las actividades de aprendizaje así como para registrar los resultados.

El nivel B del Learning Design incluye características y condiciones genéricas. Hay dos tipos de características propuestas: “Interna” y “Externa”. Las características internas tienen los nombres y rango de valores que se definen en tiempo de diseño y gobiernan el flujo de acontecimientos de una manera predeterminada. Las características externas y sus vocabularios tienen que ser convenidos más extensamente (“global”). Los ejemplos de éstos son los campos y los términos establecidos por las extensiones de la especificación de accesibilidad. Otros se pueden desarrollar en una versión futura de la especificación LIP de IMS.

Al estudiante, el nivel B le agrega personalización, el pre-conocimiento de soporte, preferencias, y accesibilidad, permitiendo a éstos ser tomados en cuenta en un Learning Design.

El **Nivel C** añade paso de mensajes al Nivel B. Un mensaje se gatilla o se "dispara" por un resultado y puede producir que una nueva actividad quede disponible para el rol que se está ejecutando.

Las actividades se pueden entonces fijar como consecuencia de cambios dinámicos a los perfiles del alumno y/o de los acontecimientos generados en el curso de las actividades que aprenden. Puede también ser utilizada para accionar los mensajes que son enviados dinámicamente a los participantes.

Más generalmente, permite la automatización de las actividades del flujo de aprendizaje, que son accionadas por la conclusión de las tareas. Los acontecimientos de colaboración pueden ser apoyados cuando las actividades de los roles sean dependientes del estado de otras actividades. Éstos pueden por lo tanto ser diseñados como una red de eventos, más bien que como una ordenación de eventos previamente planificada. Una consecuencia de esta dependencia de eventos en tiempo de ejecución es que las actividades fijadas a los alumnos ya no son predecibles, dependen de cómo transcurra el curso.

En los niveles A y B, el orden de las actividades de los alumnos es predecible, aunque por supuesto en el nivel B con el uso de propiedades y condiciones el flujo de aprendizaje puede llegar a ser condicional.

Adaptar unidades de aprendizaje al perfil del alumno

Un modelo de aprendizaje hace que un tutor inicie el proceso de aprendizaje identificando los resultados deseados para un alumno en particular. Determinando el conocimiento anterior del individuo y entendiendo las fuerzas y las necesidades especiales del estudiante, el tutor puede identificar actividades relevantes y tirar de ellas juntas en una unidad individualizada de aprendizaje, que entonces se entrega al alumno.

En nuestro caso, el papel de tutor lo realiza el sistema automáticamente. Es el propio sistema el que decide, gracias a la inteligencia artificial que le hemos proporcionado, que actividades coge para integrarlas en una unidad de aprendizaje y mandársela al alumno para que éste la estudie.

Nosotros nos hemos centrado en el Nivel A y en el Nivel B. La parte del Nivel A la hemos realizado estrictamente basándonos en la especificación Learning Design, usando la herramienta de Reload Editor Learning Design, mientras que por el contrario, la parte correspondiente al Nivel B la hemos realizado siguiendo la especificación pero no de manera estricta, es decir, que no hemos usado las Propiedades y Condiciones facilitadas por la especificación IMS Learning Design, sino que lo hemos implementado a nuestra manera pero siguiendo la especificación.

IMS Learning Object Metadata

Hemos tenido en cuenta también la especificación para los metadatos, ya que en la implementación de los distintos archivos XML que usamos en nuestro proyecto, tenemos la necesidad de incluir meta-datos con la información necesaria para cada uno de los archivos.

Estos metadatos son recogidos de **Taxonomías** que nosotros hemos definido. Para crearlas nos hemos basado en la especificación de IMS Learning Object Metadata.

El IEEE LOM estándar define un juego de elementos de meta-datos que pueden ser usados para describir el aprendizaje de recursos. Este incluye los nombres de elementos, definiciones, tipos de datos y las longitudes de los campos. El estándar incluye declaraciones de conformidad de como los documentos de meta-datos deben ser organizados y como las aplicaciones deben comportarse a fin de ser considerado conforme al IEEE.

2.4 Tecnologías

2.4.1 Java

Java es un lenguaje de programación orientado a objetos desarrollado por James Gosling y sus compañeros de Sun Microsystems al inicio de la década de 1990. A diferencia de los lenguajes de programación convencionales, que generalmente están diseñados para ser compilados a código nativo, Java es compilado en un bytecode que es ejecutado (usando normalmente un compilador JIT), por una máquina virtual Java.

El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos mucho más simple y elimina herramientas de bajo nivel como punteros.

Nuestra aplicación utiliza este lenguaje de programación para implementar el motor de búsqueda, alojado en el servidor, que utilizará nuestra herramienta para seleccionar los Learning Objects (objetos de aprendizaje) más adecuados para el alumno, según sus características y sus preferencias a la hora del aprendizaje.

Nos hemos decidido por el lenguaje Java principalmente por ser uno de los lenguajes de programación orientada a objetos más utilizados del mundo, por su simplicidad y fácil manejo y depuración y ante todo por la posibilidad de ejecutar en cualquier parte (run everywhere).

Además, en la parte del servidor, Java es más popular que nunca, con muchos sitios empleando páginas JavaServer, conectores como Tomcat para Apache y otras tecnologías Java.

Nosotros utilizaremos para nuestra aplicación el conector Tomcat de Apache, el cual ya hemos usado en alguna otra ocasión y sabemos que soporta un número bastante grande de conexiones simultáneas sin ningún tipo de problema.

En algunos casos, se habla de Java como un lenguaje menos eficiente que otros lenguajes de programación orientada a objetos, como por ejemplo C++, debido a su manejo interno de los punteros, lo que obliga a tener un recolector de basura que gestione la memoria dinámica y esto puede derivar en una bajada del rendimiento de la aplicación. Sin embargo, el rendimiento de una aplicación está determinado por multitud de factores, por lo que no es fácil hacer una comparación que resulte totalmente objetiva.

En tiempo de ejecución, el rendimiento de una aplicación Java depende más de la eficiencia del compilador, o la JVM, que de las propiedades intrínsecas del lenguaje. El bytecode de Java puede ser interpretado en tiempo de ejecución por la máquina virtual, o bien compilado al cargarse el programa, o durante la propia ejecución, para generar código nativo que se ejecuta directamente sobre el hardware. Si es interpretado, será más lento que usando el código máquina intrínseco de la plataforma destino. Si es compilado, durante la carga inicial o la ejecución, la penalización está en el tiempo necesario para llevar a cabo la compilación.

Algunas características del propio lenguaje conllevan una penalización en tiempo, aunque no son únicas de Java. Algunas de ellas son el chequeo de los límites de arrays, chequeo en tiempo de ejecución de tipos, y la indirección de funciones virtuales.

El uso de un recolector de basura para eliminar de forma automática aquellos objetos no requeridos, añade una sobrecarga que puede afectar al rendimiento, o ser apenas apreciable, dependiendo de la tecnología del recolector y de la aplicación en concreto. Las JVM modernas usan recolectores de basura que gracias a rápidos algoritmos de manejo de memoria, consiguen que algunas aplicaciones puedan ejecutarse más eficientemente.

2.4.2 JavaServer Pages (JSP)

JavaServer Pages (JSP) es la tecnología para generar páginas web de forma dinámica en el servidor, desarrollado por Sun Microsystems, basada en scripts que utilizan una variante del lenguaje java.

La tecnología JSP, o de JavaServer Pages, es una tecnología Java que permite a los programadores generar dinámicamente HTML, XML o algún otro tipo de página web. Esta tecnología permite al código Java y a algunas acciones predefinidas ser embebidas en el contenido estático. En las **JSP**, se escribe el texto que va a ser devuelto en la salida (normalmente código HTML) incluyendo código java dentro de él para poder modificar o generar contenido dinámicamente. El código java se incluye dentro de las marcas de etiqueta `<% y %>`.

La principal ventaja de **JSP** frente a otros lenguajes es que permite integrarse con clases Java (.class) lo que permite separar en niveles las aplicaciones web, almacenando en clases java las partes que consumen más recursos así como las que requieren más seguridad, y dejando la parte encargada de formatear el documento html en el archivo jsp.

Además Java se caracteriza por ser un lenguaje que puede ejecutarse en cualquier sistema, lo que sumado a jsp le da muchísima versatilidad.

Sin embargo **JSP** no se puede considerar un script al 100% ya que antes de ejecutarse el servidor web compila el script y genera un servlet, por lo tanto se puede decir que aunque este proceso sea transparente para el programador no deja de ser una aplicación compilada. La ventaja de esto es algo más de rapidez y disponer del API de Java en su totalidad.

Debido a esto la tecnología JSP, así como Java está teniendo mucho peso en el desarrollo web profesional.

El uso que hemos realizado de **JSP** en nuestro sistema es, como bien decimos anteriormente, la de usar clases Java ya implementadas en el html. Esto nos da mayor rapidez, ya que al cargarse la página html, el código JSP es compilado y ya se tienen los resultados necesarios cuando sean requeridos sin tener que perder tiempo en calcularlos en ese momento.

2.4.3 JavaScript

JavaScript es un lenguaje interpretado orientado a las páginas web, con una sintaxis semejante a la del lenguaje Java.

El lenguaje fue inventado por Brendan Eich en la empresa Netscape Communications, que es la que fabricó los primeros navegadores de Internet comerciales. Apareció por primera vez en el producto de Netscape llamado Netscape Navigator 2.0.

Tradicionalmente, se venía utilizando en páginas web HTML, para realizar tareas y operaciones en el marco de la aplicación cliente servidor. Con la irrupción de Web 2.0, **JavaScript** se ha convertido en un verdadero lenguaje de programación que aporta la potencia de cálculo del navegador para aumentar la usabilidad de aplicaciones Web con técnicas avanzadas como AJAX o JCC.

En nuestro sistema, el uso que hemos realizado de JavaScript ha sido principalmente para realizar tareas y operaciones en el marco de la aplicación cliente servidor. Hemos usado JavaScript por ejemplo a la hora de pinchar en los botones para realizar ciertas comprobaciones, como que los campos no estén vacíos, o que no sea la última posición de un vector...

2.4.4 Html

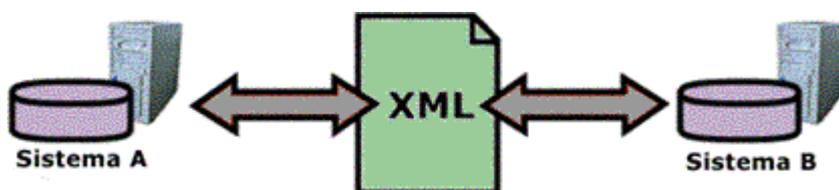
El **HTML**, acrónimo inglés de **Hypertext Markup Language** (lenguaje de marcado de documentos hipertextual), es un lenguaje de marcado diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet y a los navegadores del tipo Internet Explorer, Opera, Firefox o Netscape, el HTML se ha convertido en uno de los formatos más populares que existen para la elaboración de páginas web.

En nuestro caso éste ha sido el lenguaje utilizado para el diseño de las páginas webs que utilizaremos como interfaz con el usuario. El usuario se registrará, se identificará, accederá a la información y a los cursos a través de las páginas web diseñadas con HTML. A fin de cuentas estas páginas serán la única conexión que tendrá el usuario con el resto de la aplicación.

2.4.5 XML

XML (siglas en inglés de **eXtensible Markup Language**, ‘lenguaje de marcado extensible’) es un lenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (cómo HTML es un lenguaje definido por SGML).

XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.



Si la información se transfiere en XML cualquier aplicación podría escribir un documento de texto plano con los datos que estaba manejando en formato XML y otra aplicación recibir esta información y trabajar con ella.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

Sin duda el lenguaje XML juega un papel fundamental en nuestra aplicación. Toda la información referente a los cursos, Learning Objects, metadatos, desarrollo del curso para cada alumno, etc., se gestionará, almacenará y se intercambiará con el usuario a través de documentos XML. De hecho, la especificación IMS está especialmente diseñada para trabajar con este tipo de lenguaje de intercambio de información.

2.4.6 Reload Editor

Reload Editor es un empaquetador de contenido y un editor de Metadatos. Es la herramienta adecuada para aquellas personas que trabajan o crean materiales educativos y desean que estos puedan ser compartidos.

Crear un paquete de contenido es la mejor forma de compartir materiales educativos. Todos los archivos requeridos se almacenan convenientemente dentro de un único archivo .zip y la estructura estándar creada puede ser entendida por un programa de repositorio y por Entornos de Aprendizaje.

Los Metadatos describen los contenidos del Paquete de Contenido, y también pueden proveer información sobre requerimientos técnicos, restricciones de uso y anotaciones de un tercero. Al añadir Metadatos a un recurso hacemos que éste pueda ser clasificado, indexado y recuperado desde los repositorios.

Decimos que Los Paquetes de Contenido son usados para almacenar contenido educativo pero, por supuesto, contenido educativo puede significar cualquier cosa: desde una única imagen que puede usarse como ejemplo hasta un curso Web completo con extractos de video y sonido, archivos de Acrobat, simulaciones de Java, animaciones Flash, documentos de Word y cuestionarios. Los Metadatos añadidos al Paquete de Contenido permiten a otros usuarios evaluar su idoneidad para sus necesidades.

El empaquetador de contenido funciona de forma sencilla: Todo el contenido que se necesita para preparar un material educativo es recopilado conjuntamente. Un archivo especial, el Manifiesto -en inglés manifest- que debe llamarse imsmanifest.xml se crea entonces. El manifiesto almacena información sobre el recurso en un muy específico y convenido formato establecido por un grupo llamado IMS. El manifiesto es una especie de etiqueta adjunta al lado de un paquete.

El manifiesto contiene la siguiente información:

- **Metadatos:** sobre el contenido, almacenado por el curso como un conjunto o archivos individuales.
- **Recursos:** qué archivos se necesitan y dónde residen en el Paquete de Contenido. Esta sección también se encarga de registrar las posibles dependencias (por ejemplo, que una página Web requiere una hoja de estilo y algunas imágenes para visualizarse correctamente).
- **Organizaciones:** que describen cómo interactúan los recursos individuales. Esta información puede ser necesitada por cualquier persona o paquete de contenido que vayan a trabajar posteriormente con el contenido empaquetado. Se pueden proporcionar organizaciones alternativas.

Por conveniencia, los paquetes de contenido son almacenados en ficheros .zip. En este caso, el archivo imsmanifest.xml debe estar ubicado en el directorio base del fichero .zip de forma que pueda ser fácilmente encontrado por cualquier programa que necesite utilizarlo.

Crear el fichero del manifiesto implica crear un archivo en formato XML. Aunque similar al HTML, XML es mucho más complejo y, por tanto, menos sencillo de trabajar con él. A través de una interfaz de usuario de forma visual para la tarea de creación del manifiesto, Reload Editor proporciona una forma muy simple de crear paquetes de contenido y Metadatos, simplemente arrastrando archivos de un panel a otro y escribiendo la información dentro de campos en formularios. Reload Editor también permite previsualizar el Paquete de Contenido con el fin de comprobar que se obtiene lo que se esperaba.

En nuestro caso hemos empleado la herramienta Reload Editor como una guía o ejemplo en el tratamiento de los Metadatos y en el modo en el que se crea el archivo `imsmanifest.xml`, tratando de ver cómo una herramienta como Reload Editor se ajusta a los estándares de IMS y tratar así de tomar una dirección similar en la resolución de problemas como la **creación dinámica del archivo manifest, la gestión de Metadatos o la recolección en el repositorio de Learning Objects adecuados para el alumno**, para así asegurarnos de que herramientas bastante extendidas y reconocidas en el entorno del E-Learning, como es el caso de Reload Editor, no tendrían problemas en trabajar conjuntamente con nuestra aplicación.

2.4.7 Macromedia DreamWeaver

Macromedia Dreamweaver es un editor de páginas web, creado por Macromedia. Es el programa de este tipo más utilizado en el sector del diseño y la programación web, por sus funcionalidades, su integración con otras herramientas como Macromedia Flash y, recientemente, por su soporte de los estándares del World Wide Web Consortium. Su principal competidor es Microsoft Frontpage. Tiene soporte tanto para edición de imágenes como para animación a través de su integración con otras herramientas.

La gran baza de este editor sobre otros, y una de las razones por la que lo hemos escogido para el diseño de nuestras páginas, es su gran poder de ampliación y personalización del mismo, puesto que este programa, sus rutinas (como la de insertar un hipervínculo, una imagen o añadir un comportamiento) están hechas en Javascript-C lo que le ofrece una gran flexibilidad en estas materias. Esto hace que los archivos del programa no sean instrucciones de C++, sino rutinas de Javascript que hace que sea un programa mucho más fluido.

2.4.8 Photoshop

Photoshop es una aplicación informática de edición y retoque de imágenes bitmap elaborada por la compañía de software Adobe inicialmente para computadores Apple pero posteriormente también para plataformas PC.

El uso de Photoshop ha sido imprescindible para el diseño y tratamiento de imágenes, principalmente las utilizadas en el diseño de las páginas web.

2.4.9 MySQL

MySQL es uno de los Sistemas Gestores de bases de Datos (SQL) más populares desarrollados bajo la filosofía de código abierto. Según las cifras del fabricante, existirían más de seis millones de copias de MySQL funcionando en la actualidad, lo que supera la base instalada de cualquier otra herramienta de bases de datos.

La desarrolla y mantiene la empresa MySQL AB pero puede utilizarse gratuitamente y su código fuente está disponible.

MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas web con contenido dinámico, justamente por su simplicidad; aquellos elementos faltantes fueron llenados por la vía de las aplicaciones que la utilizan.

Poco a poco los vacíos en MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- Transacciones y claves foráneas
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

En nuestra aplicación MySQL será el sistema gestor de la base de datos de los alumnos con sus campos asociados (p.e. identificador, tema por el que van, contraseñas, etc.) alojado en el servidor. Hemos optado por utilizar MySQL como sistema gestor de nuestra base de datos por varios motivos:

- Es un sistema de código abierto, lo cual reduce costes no existen problemas de licencia.
- Es uno de los sistemas gestores de bases de datos más importantes en el mundo y uno de los más extendidos.
- Emplea el lenguaje SQL, lenguaje universal para el desarrollo y mantenimiento de bases de datos.
- Tiene capacidad más que suficiente para trabajar y gestionar niveles altos de concurrencia y tamaños de datos importantes.
- Experiencias anteriores con éste gestor con óptimos resultados.

2.4.10 Apache / Tomcat

Tomcat (también llamado **Jakarta Tomcat** o **Apache Tomcat**) funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems. Se le considera un servidor de aplicaciones.

No funciona con cualquier servidor web con soporte para servlets y JSPs. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets del Tomcat a menudo se presenta en combinación con el servidor web Apache.

Puede, a su vez, funcionar como servidor web por sí mismo. Opera de tal manera en entornos de desarrollo poco exigentes en términos de velocidad y de manejo de transacciones.

Cabe destacar que Tomcat fue escrito en Java, por lo que funciona en cualquier sistema operativo que disponga de la máquina virtual. Lo desarrollan y lo mantienen miembros de la Apache Software Foundation y voluntarios independientes. Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la *Apache Software Licence*.

ARCHIVOS	CONTENIDO
bin	arranque, cierre, y otros scripts y ejecutables
common	clases comunes que pueden utilizar Catalina y las aplicaciones web
conf	ficheros XML y los correspondientes DTDs para la configuración de Tomcat
logs	logs de Catalina y de las aplicaciones
server	clases utilizadas solamente por Catalina
shared	clases compartidas por todas las aplicaciones web
webapps	directorio que contiene las aplicaciones web
work	almacenamiento temporal de ficheros y directorios

Tabla 1. Jerarquía de directorios de instalación de Tomcat

Breve reseña histórica

Tomcat empezó siendo una implementación de la especificación de los servlets comenzada por James Duncan Davidson, que trabajaba como arquitecto de software en Sun y que posteriormente ayudó a hacer el proyecto *open source* y colaboró en su donación a la Apache Software Foundation.

Duncan Davidson inicialmente esperaba que el proyecto se convirtiese en *open source* y dado que la mayoría de los proyectos open source tienen libros de O'Reilly asociados con un animal en la portada, quiso ponerle al proyecto nombre de animal. Eligió *Tomcat* (gato), pretendiendo representar la capacidad de cuidarse por sí mismo, de ser independiente.

Las primeras distribuciones de Tomcat fueron las versiones 3.0.x. Las versiones más recientes son las 5.x, que implementan las especificaciones de Servlet 2.4 y de JSP 2.0. En las versiones 4.x, Jakarta Tomcat utiliza el contenedor de servlets.

VERSIONES	CARACTERÍSTICAS
Tomcat 3.x	Implementado a partir de las especificaciones Servlet 2.2 y JSP 1.1
	Recarga de servlets
Tomcat 4.x	Implementado a partir de las especificaciones Servlet 2.3 y JSP 1.2
	Contenedor de servlets rediseñado como Catalina
	Motor JSP rediseñado con Jasper
	Conector Coyote
Tomcat 5.x	Implementado a partir de las especificaciones Servlet 2.4 y JSP 2.0
	Recolección de basura reducida
	Capa envolvente nativa para Windows y Unix para la integración de las plataformas

Tabla 2. Características de las diferentes versiones de Tomcat.

2.4.11 Rational Rose

UML (Unified Modeling Language) se ha convertido en el estándar de facto para definir, organizar y visualizar los elementos que configuran la arquitectura de una aplicación orientada a objetos.

Rational Rose es la herramienta CASE desarrollada por los creadores de UML (Booch, Rumbaugh y Jacobson), que cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables.

El navegador UML de Rational Rose nos permite establecer una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable. Facilita el desarrollo de un proceso cooperativo en el que todos los agentes tienen sus propias vistas de información (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue), pero utilizan un lenguaje común para comprender y comunicar la estructura y la funcionalidad del sistema en construcción.

Hemos utilizado la herramienta Rational Rose para generar los diagramas UML que describen las clases Java más importantes del proyecto. También la hemos empleado para generar los diagramas de secuencia que muestran la comunicación entre las clases en los procesos más importantes de nuestro sistema.

2.5 Estilos de Aprendizaje

El Dr. Felder hizo una clasificación inicial en cinco diferentes dimensiones o estilos de aprendizaje.

Estos estilos de aprendizaje eran el **inductivo / deductivo, visual / auditivo, activo / reflexivo, detección / intuición, secuencial / global**. Uno de ellos, el **inductivo / deductivo** actualmente ha sido desechado quedando en tan sólo cuatro estilos. El motivo de prescindir de este estilo de aprendizaje, nos explica Felder, es para evitar que otros profesionales de la enseñanza que utilicen los resultados de este test para dejar de lado la parte inductiva (enseñanza basada en problemas) pues la mayoría de los estudiantes se decantan por una enseñanza más deductiva.

Otro de ellos era el conocido por **visual / auditivo** pero lo modificó por el estilo **visual / verbal**, debido a que era claro que la parte visual la conformaban los dibujos, diagramas, cuadros, animaciones y la parte auditiva era constituida por palabras habladas y sonidos. Folder encontró dificultades a la hora de ajustar las explicaciones escritas a una de estas dos categorías. Puesto que el lenguaje escrito es percibido de manera visual sería un error catalogarla como auditiva, pero también sería un error catalogarla como visual pues no es equivalente a un dibujo o un diagrama. Por ello definió el nuevo estilo de aprendizaje como **visual / verbal** en el cual quedaba estructurado todo más claramente, incluyendo en la misma categoría (verbal) tanto las palabras escritas como las palabras habladas.

De manera que diferenciaríamos entre los alumnos que preferirían una enseñanza visual, porque recuerdan mejor lo que ven (diagramas, dibujos, tablas, animaciones) y los que preferirían una enseñanza verbal, porque sacan más provecho de explicaciones ya sean escritas o habladas. De todas maneras todo el mundo aprende mucho más y mejor si el aprendizaje se basa en ambas categorías.

El **activo / reflexivo** distingue entre aquellos alumnos que prefieren realizar un aprendizaje más activo, discutiendo sobre lo que están aprendiendo o explicándoselo a otros para entre todos entenderlo mejor. Por tanto cabe deducir que a un alumno con esta preferencia le guste trabajar en grupo. En cambio al alumno que prefiere un método más reflexivo, le gusta pensar sobre el concepto que está intentando asimilar de una manera más solitaria. Por lo tanto un alumno activo intentaría comprobar como funciona o que utilidad tiene lo que ha entendido, o está intentando entender, mientras un alumno reflexivo meditaría sobre lo que está aprendiendo.

Otro estilo sería el de **detección / intuición** que diferencia entre los alumnos que prefieren aprender los hechos sin pararse a pensar en lo que pudieran tener esos hechos a su alrededor y los alumnos que prefieren descubrir las relaciones, las conexiones que pudiera haber entre todo aquello que están aprendiendo, sacando a la luz nuevas posibilidades. Aquello que prefieren la detección son partidarios de resolver los problemas siguiendo un método, de una manera sistemática, son prácticos y cuidadosos, son muy buenos memorizando hechos y les gusta que lo que están aprendiendo puedan aplicarlo a la vida real o tengan en la vida real un ejemplo en el que fijarse, por el contrario los que se dejan llevar por la intuición les gusta innovar y ser originales a la

hora de la resolución de los problemas y no resolverlos de una manera repetitiva, son buenos a la hora de aprender conceptos abstractos y fórmulas matemáticas, les gusta trabajar más rápido y ser innovadores y no les gustan los cursos que se basan en la memorización de hechos y cálculos repetitivos, los cuales les supone una rutina insoportable. Lo ideal en un buen aprendizaje es poder tener un equilibrio entre ambas categorías pues si bien hay que seguir un método, una rutina a la hora de llevar a cabo dicho aprendizaje, esa misma rutina puede llegar a desanimarnos si no lo mezclamos con un poco de innovación y originalidad.

Y por último nos queda el **secuencial / global**. Hay alumnos que prefieren aprender todo paso a paso y no pasar al siguiente hasta que no se ha dominado al anterior. También tienen que seguir un orden lógico a la hora de resolver problemas. A estos alumnos les gusta un aprendizaje secuencial. Por lo contrario hay otros que prefieren ver lo que están aprendiendo como un todo, de manera global y sin ningún tipo de orden, para después entender cada parte. En cuanto a este estilo de aprendizaje también es bueno tener un equilibrio razonable entre los dos puntos de vista.

Dimensions of Learning and Teaching Styles

<i>Preferred Learning Style</i>		<i>Corresponding Teaching Style</i>	
sensory } intuitive }	perception	concrete } abstract }	content
visual } auditory }	input	visual } verbal }	presentation
inductive } deductive }	organization	inductive } deductive }	organization
active } reflective }	processing	active } passive }	student participation
sequential } global }	understanding	sequential } global }	perspective

2.6 Nuestro sistema, motivación y objetivos

Cuando se nos presentó la oportunidad de realizar este proyecto nos pareció una gran idea. En principio reunía características de inteligencia artificial (como son las decisiones de la aplicación sobre cuales son los Learning Objects más adecuados para cada usuario) junto con E-Learning, lo que despertaba una gran curiosidad en todos los miembros del grupo, y era algo en lo que teníamos mucho interés en profundizar.

El proyecto a desarrollar trataba de personalizar la enseñanza de un alumno de acuerdo a su estilo de aprendizaje, es decir, adaptar un mismo curso a distintos perfiles de alumnos, pudiendo además cambiar el estilo de enseñanza en un momento dado, si los resultados obtenidos por el alumno no eran los esperados.

Sin embargo había varios problemas. No teníamos ningún conocimiento sobre Learning Design ni sobre los estándares que se utilizan en este campo. No sabíamos nada del uso de lenguajes como XML, ni de herramientas básicas en el campo del E-Learning como era Reload Editor. Además, **no existía ningún precedente, ninguna aplicación ya diseñada y probada que hiciera lo que nosotros íbamos a tratar de implementar**. No teníamos ninguna guía, libro o herramienta ya diseñada en la que basarnos para realizar nuestro proyecto.

Debido a todos estos problemas **el trabajo de documentación era completamente fundamental para que el proyecto saliera adelante**. El E-Learning abarca una cantidad de posibilidades y de complejidad enorme todavía sin desarrollar, ya que es un campo de la enseñanza y de la informática que ha aparecido hace relativamente poco tiempo, y está aún por explorar.

De este modo **la comunicación con nuestra directora de proyecto, Pilar Sancho Thomas, se convirtió en algo imprescindible**. Fijamos reuniones cada 15 días, además de correos electrónicos y llamadas, explicándonos una y otra vez estándares, objetivos, distintas metodologías y todo tipo de cuestiones que nos asaltaban sobre el proyecto. De esta manera tratábamos de evitar desviarnos de los objetivos fijados en un principio. A pesar de todo esto, más de una vez, nos vimos obligados a dar marcha atrás sobre temas en los que habíamos profundizado demasiado, dejando otros más importantes sin explorar, cometiendo errores que esperamos que, gracias a nuestro esfuerzo dedicado y al de Pilar, las personas que continúen nuestro trabajo puedan evitar, realizando así más cómodamente su trabajo y con los objetivos algo más clarificados que con los que nosotros empezamos.

Uno de los principales temas en los que debíamos profundizar mucho, antes de comenzar ningún tipo de desarrollo, era decidir qué estándar de Learning Design íbamos a seguir en el desarrollo de la aplicación. Era necesario analizar los distintos estándares para comprobar cual era el que mejor se adaptaba a nuestras necesidades, sobre cual de ellos podríamos basar nuestra aplicación. **Después de barajar varias posibilidades y con la ayuda de Pilar escogimos IMS Learning Design**. IMS se adecuaba perfectamente a las características que pensábamos desarrollar.

La flexibilidad de la especificación IMS y la facilidad con la que se puede adecuar para llevar a cabo nuestro sistema de personalización del aprendizaje sin salirnos de los estándares fueron las principales razones que nos condujeron a escoger IMS Learning Design, ante el resto de especificaciones.

En cuanto a las metas propuestas al inicio del proyecto, nuestro principal objetivo siempre fue **tratar de innovar en el campo del E-Learning**. Sabíamos antes de comenzar el proyecto que era algo complicado, ya que no existía, o al menos no teníamos conocimiento, ningún trabajo similar al que se nos planteaba.

Tratamos de conseguir **una aplicación** capaz de decidir cual es el método de aprendizaje más adecuado para una persona determinada y, en función de ese método, sea **capaz de seleccionar adecuadamente el contenido y formato de la materia que quiera aprender** y montar así un curso completamente personalizado a sus necesidades, tratando de abstraernos en la mayor medida de cuál es la materia a impartir.

Tratamos de conseguir que **cada alumno se sienta diferenciado** en un entorno de aprendizaje que generalmente se conoce como un aprendizaje más frío y desnaturalizado.

Tratamos de conseguir que **cada alumno consiga un entorno de aprendizaje completamente óptimo para sus características**, tal y como trataría de hacerlo cualquier profesor de colegio, instituto, universidad, etc, para cada uno de sus alumnos.

Incluso **tratamos de conseguir una enseñanza mucho más orientada al alumno que la que se puede impartir con la enseñanza “tradicional”**, ya que ningún profesor (a no ser que se traten de clases particulares) puede conseguir dar una clase óptima para cada uno de los alumnos, debido a que cada alumno de la clase tendrá un método de aprendizaje asociado, el cual le dará mejores resultados que los demás métodos, diferente al de sus compañeros.

De ésta forma, **con una mixtura entre enseñanza presencial e E-Learning personalizado creemos que se pueden conseguir algunas ventajas** muy importantes:

- Enseñanza con el método óptimo para cada alumno, lo que deriva en:
 - Aprendizaje más rápido.
 - Menos fracaso escolar.
 - Más interés por parte de los alumnos.
 - Mayor motivación para los profesores.
- Costes más bajos para los formadores.
- Mayor flexibilidad para los alumnos, aprender donde quieran y cuando quieran con el E-Learning.
- Posibilidad de auto - fijación de objetivos.
- Posibilidad de cambio de método de aprendizaje en cualquier momento si no se obtienen los resultados esperados.

En definitiva, **conseguimos una enseñanza completamente adaptada a las necesidades del alumno con unos costes significativamente más bajos** que los de la enseñanza “tradicional”.

Sin embargo, este proyecto tan solo abre las puertas hacia todo lo propuesto anteriormente. Este es el principio de un gran trabajo que aún queda por desarrollar si verdaderamente queremos que todo lo anterior se lleve a cabo. Ha sido un comienzo duro, pero como ya hemos dicho anteriormente, **esperamos que resulte clarificador para las personas que continúen y mejoren nuestro trabajo**. Ahora que hemos terminado nuestro trabajo y hemos profundizado en el tema, tenemos una gran cantidad de mejoras e ideas en la cabeza que, aunque ya no tenemos tiempo suficiente para poder implementarlas, trataremos de exponerlas con la mayor claridad posible más adelante, para que quizá otros puedan llevarlas a cabo con éxito y sin demasiadas dificultades.

3 Sistema desarrollado

3.1 Arquitectura del sistema

Nuestro sistema está implementado principalmente en **Java**. Utilizamos páginas **HTML** que constituyen la interfaz de nuestro proyecto, las cuales hacen referencia a páginas **JSP** y además usan distintas hojas de estilo. Además, tenemos una base de datos en **MySQL** y el sistema está montado en un servidor **Apache**.

Después de explicar el contenido de nuestro proyecto referente a cada una de las tecnologías anteriores, mostraremos unos diagramas de secuencia para que se entienda la interacción entre cada una de ellas.

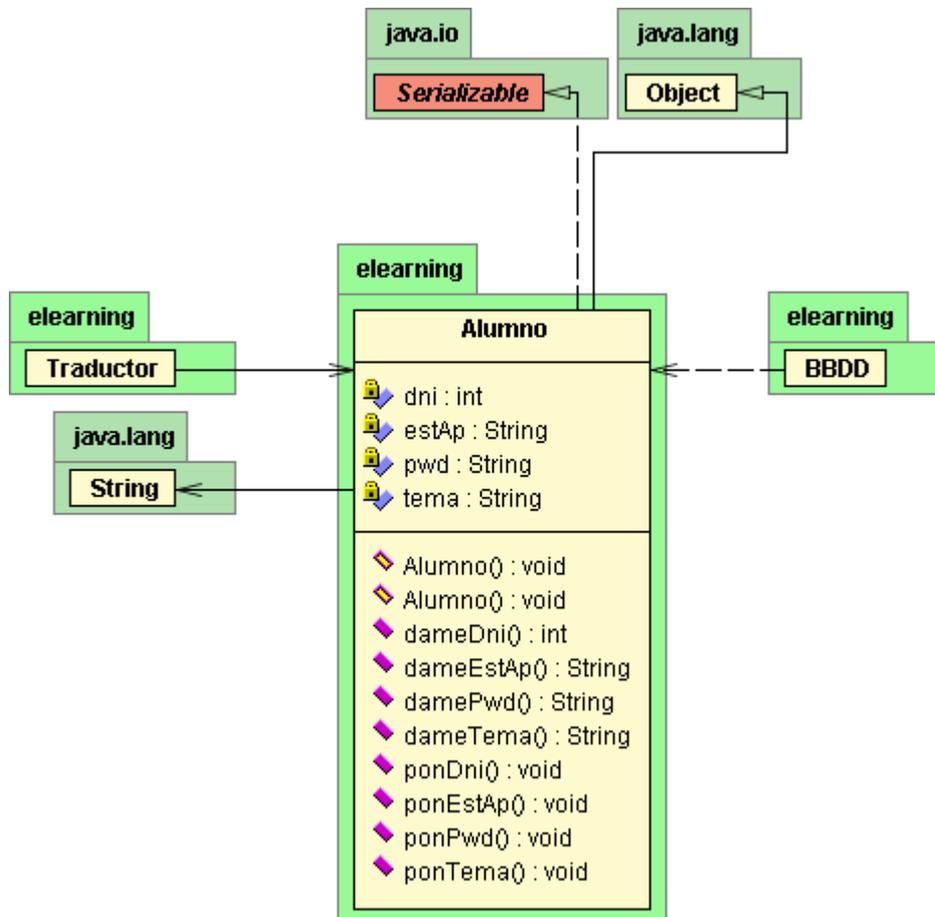
3.1.1 JAVA

Hemos creado un proyecto Java, que hemos llamado **ProyectoSSII**. En él, hemos creado un paquete **elearning** dónde hemos implementado todas las clases necesarias para el buen funcionamiento del sistema. Además, cabe destacar que todas estas clases han sido comentadas mediante el estilo JavaDoc.

A continuación vamos a enumerar las distintas clases que tenemos implementadas en Java.

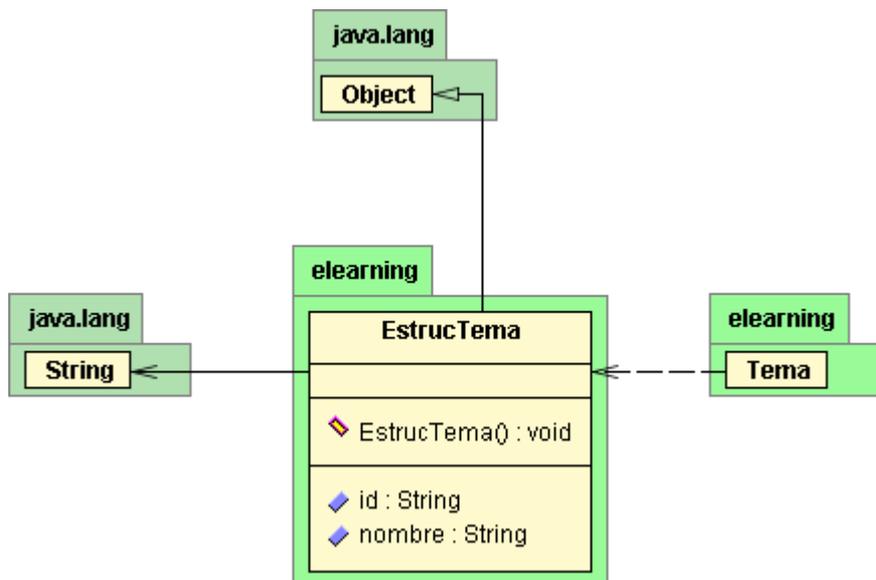
Alumno:

La clase Alumno contiene los distintos atributos que caracterizan a un objeto Alumno, el **DNI**, la **contraseña**, el **estilo de aprendizaje** al cual pertenece y el **tema** por el que va el alumno. Esta clase tiene implementada los distintos accesores y mutadores para poder saber o poner el valor de un atributo.



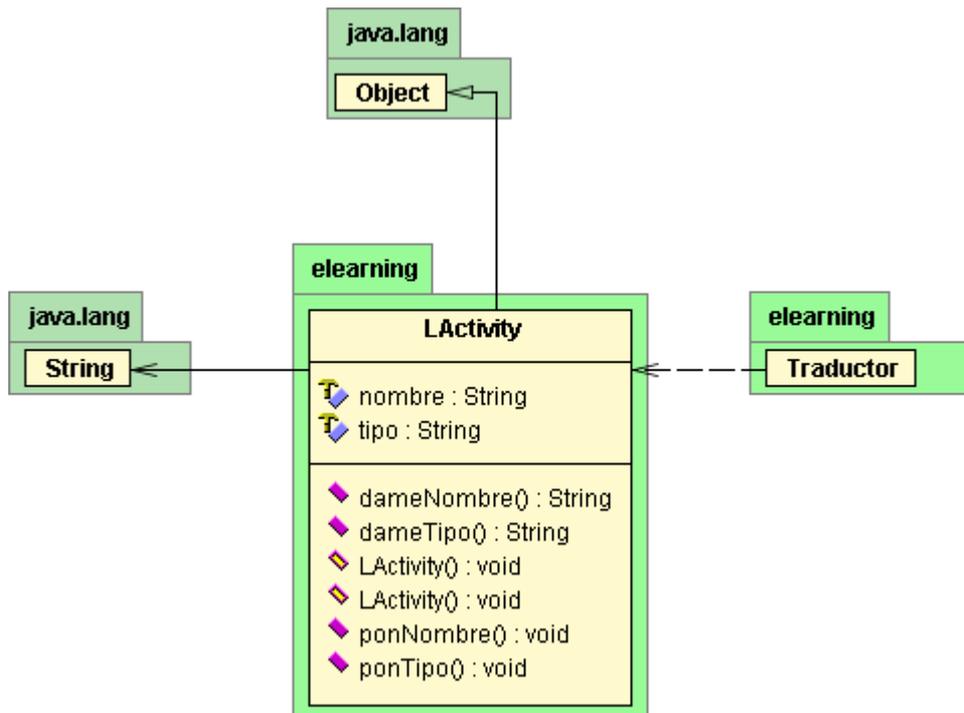
EstrucTema:

Esta clase es la definición de la estructura de los objetos que serán almacenados en un vector para poder saber por cual de todos los temas va un alumno. Para ello, tenemos esta estructura que lo que tiene son 2 atributos: el **identificador** del tema, y el **nombre** del tema. Al igual que la clase anterior, esta también tiene la implementación de los accesotes y mutadores para cada uno de los atributos.



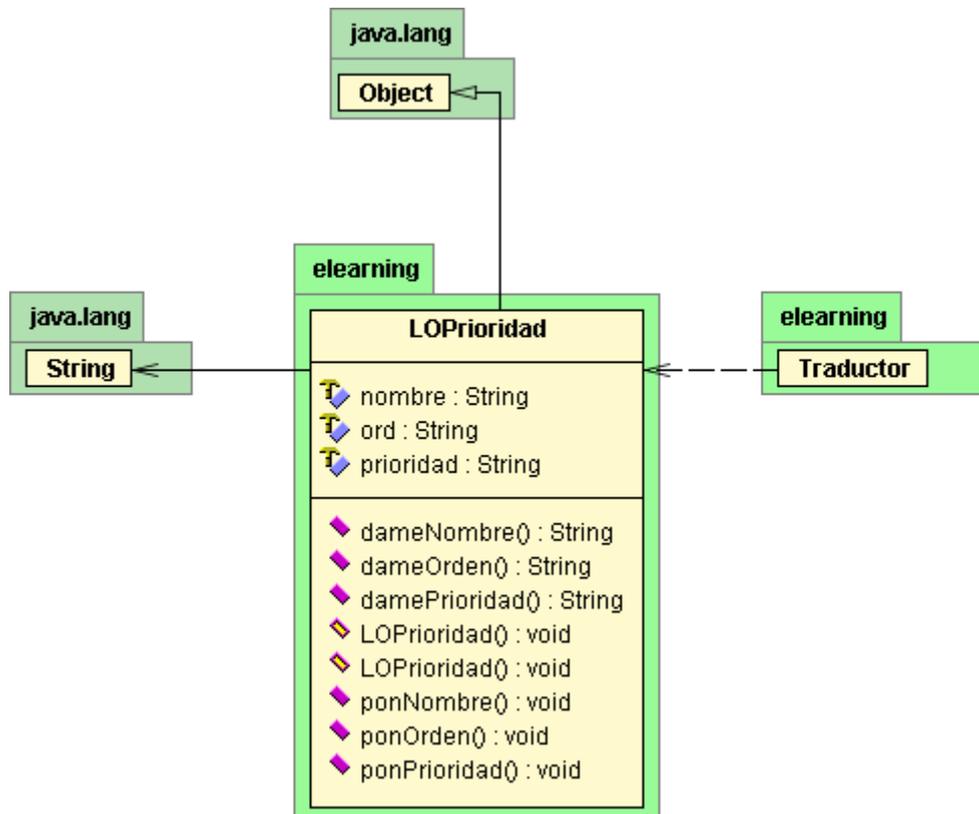
LActivity:

Como en los casos anteriores, esta clase es la definición de la estructura de una Learning Activity con el **nombre** del fichero y el **tipo** de learning activity que es, es decir Teoría, Ejemplo o Ejercicio.



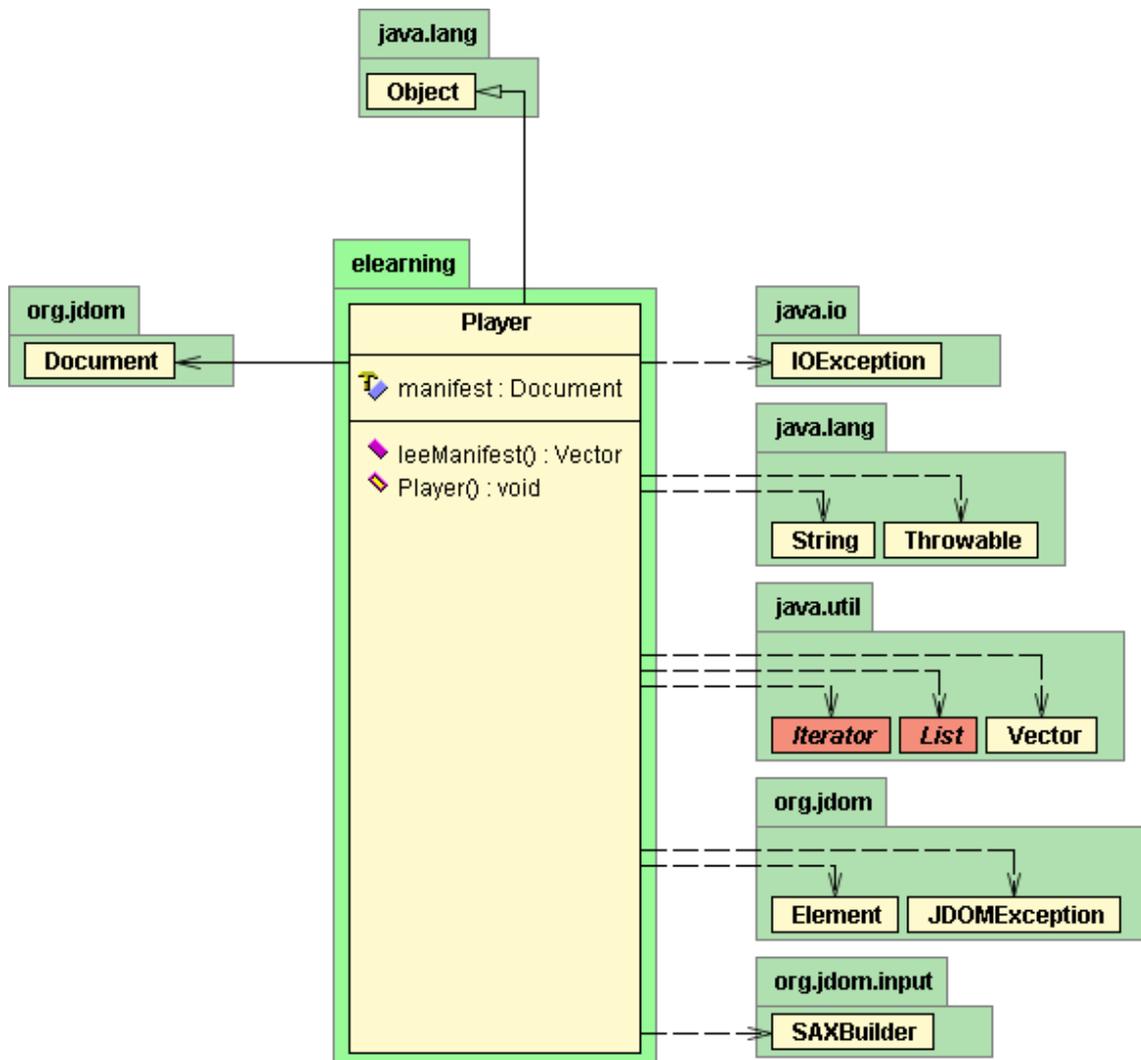
LOPrioridad:

Esta clase nos proporciona la estructura para un Learning Object, almacenando en sus atributos el **nombre**, la **prioridad** y el **orden** de un learning object concreto.



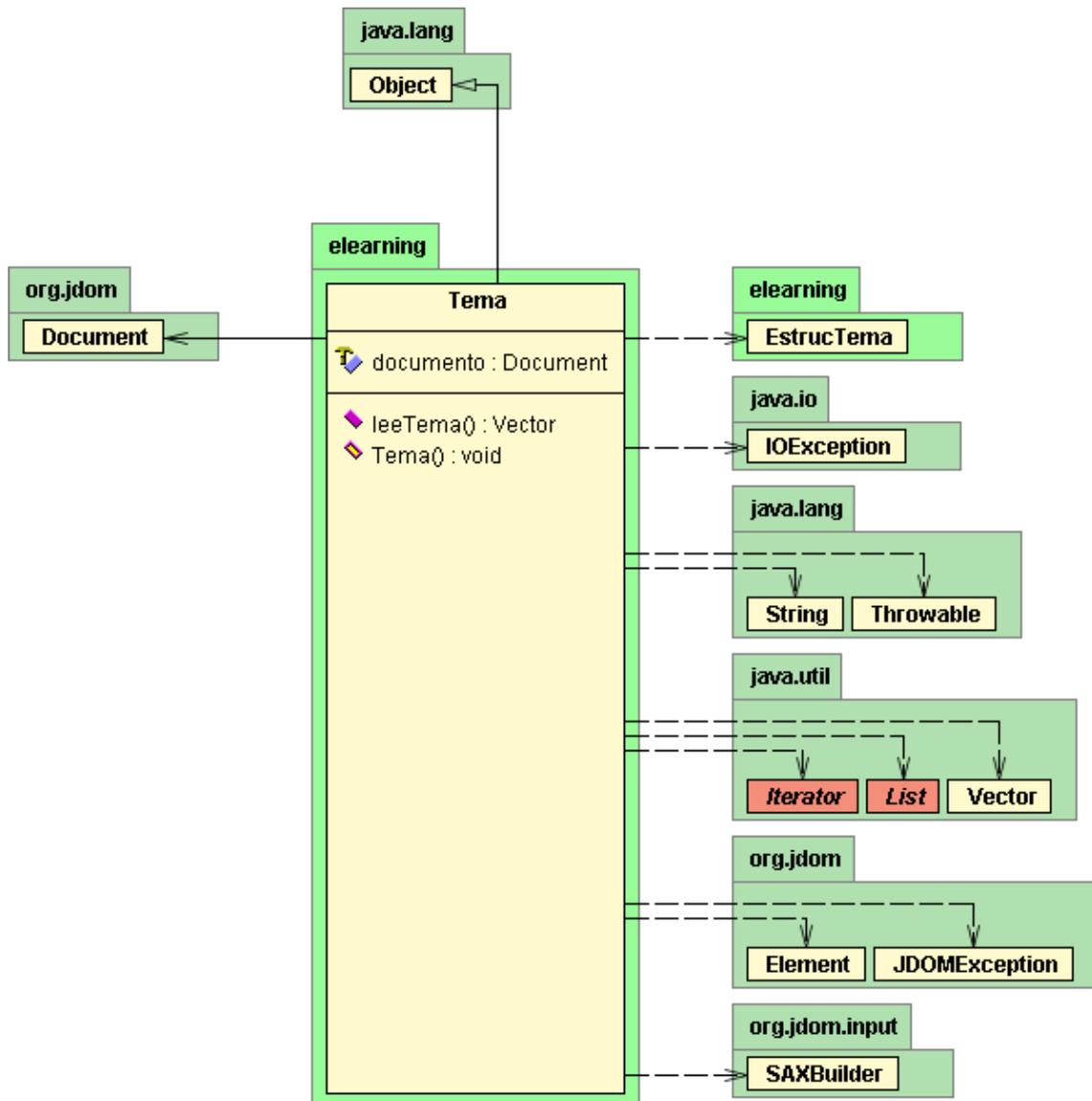
Player:

Esta clase es la encargada de recibir el manifest creado dinámicamente en la clase Traductor y crear un Vector que almacena en orden las referencias a las páginas html de un tema concreto, para que un alumno pueda visualizarlas siguiendo además el criterio de su estilo de aprendizaje.



Tema:

Esta clase se encarga de leer el archivo Tema.xml y devolver un Vector que almacena en orden objetos de tipo EstructTema, es decir, devuelve en orden los temas con su identificador y su nombre.



Traductor:

Esta clase es la principal de todo el proyecto. Es la que realiza todo el proceso de creación del manifest de forma dinámica. El método principal de esta clase es **construirManifest(String uol)** que recibe como parámetro la ruta del archivo xml correspondiente a la Unidad de Aprendizaje del tema por el cual va el alumno.

De este método se suceden las llamadas a los distintos y numerosos métodos de los que consta la clase, hasta que el manifest queda totalmente construido.

3.1.2 HTML

A continuación vamos a enumerar las distintas páginas que tenemos implementadas en HTML:

[acceso.htm](#):

Página que contiene el formulario para poder acceder al curso. Este formulario realiza un action sobre la página **login.jsp**.

[accesoErrorNoExiste.htm](#):

Si al realizar el login, resulta que el usuario no existe, se redirige a la página **paginaError.htm** que contiene en su frame izquierdo esta página que es igual que la anterior pero con el mensaje de que el usuario no existe.

[bienvenida.htm](#):

Frame central de la página **paginaInicio.htm**. Contiene un mensaje de bienvenida al curso de Introducción a la programación en C++.

[finCurso.htm](#):

Página que muestra un mensaje de enhorabuena al haber concluido el curso por completo.

[indice.htm](#):

Página que contiene el botón “Cerrar Sesión” y además según vaya el alumno avanzando en el curso irá conteniendo el índice de los temas anteriores para que pueda moverse libremente por el mismo.

[inicio.htm](#):

Página que contiene un mensaje de los creadores del proyecto en la página inicial del curso, es decir en paginaInicio.htm.

paginaCurso.htm:

Página dividida en 3 frames distintos:

- frame izquierdo: índice.htm
- frame central: este frame tiene el nombre de “curso”, pero no tiene ninguna página asociada inicialmente. El contenido que se irá mostrando serán las distintas páginas que vayan leyéndose del vector devuelto por la clase Player.
- frame inferior: flechas.jsp

paginaError.htm:

Página dividida en 3 frames distintos:

- frame izquierdo: accesoErrorNoExiste.htm
- frame central: bienvenida.htm
- frame inferior: inicio.htm

paginaInicio.htm:

Página dividida en 3 frames distintos:

- frame izquierdo: acceso.htm
- frame central: bienvenida.htm
- frame inferior: inicio.htm

registro.htm:

Página que contiene el formulario para poder registrarse en el curso. Este formulario realiza un action sobre la página **registroAl.jsp**.

registroCompletado.htm:

Página que muestra una serie de mensajes explicando al usuario lo que tiene que realizar tras haberse registrado con éxito en nuestro curso.

registroErrorExiste.htm:

Si al realizar el registro, resulta que el usuario intenta registrarse con un DNI ya almacenado en la base de datos, el sistema se redirige a esta página que es igual que la página de **registro.htm** pero con el mensaje de que el usuario ya existe.

3.1.3 JSP

A continuación vamos a enumerar las distintas páginas que tenemos implementadas en JSP:

cambioTema.jsp:

Este archivo jsp es llamado cuando se pulsa cualquiera de los botones “Ir” de la página `indice.htm`. Lo que realiza es comprobar cual ha sido el botón pulsado, y dependiendo de cual haya sido crea el manifest para el tema elegido y según el estilo de aprendizaje necesario.

finalizarTema.jsp:

La acción a realizar por esta página consiste en comprobar la situación actual del alumno, es decir, el tema actual por el que va el alumno. A continuación busca en el vector devuelto por la clase **Tema** la posición del tema actual y una vez encontrada realiza la construcción del manifest para la siguiente posición, ya que será el próximo tema que deberá estudiar el alumno, y se lo muestra por pantalla.

flechas.jsp:

Esta página contiene las flechas para ir avanzando hacia delante o hacia atrás en el curso. Además contiene el botón finalizar tema que al ser pulsado realiza un action sobre la página **finalizarTema.jsp**. Esta página recibe el vector creado por la clase **Player** que contiene las referencias a las páginas en orden del tema actual. Al ir pulsando las diferentes flechas, vamos recorriendo ese vector en un sentido o en otro dependiendo de cual de las dos flechas pulsemos.

login.jsp:

Esta página recoge los datos introducidos en el formulario de la página **acceso.htm**, comprueba que ese alumno existe, es decir si está almacenado en la base de datos. Si existe, se crea el manifest llamando a las diferentes clases con el tema por el que vaya el alumno y se le manda a la página **paginaCurso.htm** con la primera página a mostrar en el frame central “curso”. Si el alumno no existe, se le redirige a la página **paginaError.htm** mostrándole el error de que el usuario especificado no existe.

registroAl.jsp:

Esta página recoge los datos introducidos en el formulario de la página **registro.htm**, y comprueba que ese alumno no está ya registrado en nuestra base de datos. Si no está ya registrado, se almacenan todos sus datos en una nueva entrada de la base de datos. Si por el contrario ya existiera ese DNI, se redirige la página a **registroErrorExiste.htm**, mostrándole el error de que el usuario que ha intentado registrar ya existe.

3.1.4 MySQL

A continuación mostramos las instrucciones necesarias para crear la base de datos que usamos para almacenar los datos de los alumnos.

Primero creamos la base de datos que contendrá nuestra tabla:

```
create database proyecto;
```

A continuación accedemos a la base de datos creada:

```
use proyecto;
```

Y finalmente creamos la tabla **Alumnos** que contendrá la información necesaria:

```
create table Alumnos (  
    DNI INT(8) UNSIGNED primary key,  
    pwd varChar(20),  
    estilo varChar(20),  
    tema varChar(20)  
);
```

3.1.5 Apache / Tomcat

En nuestro sistema, al tratarse de una aplicación cliente-servidor, necesitamos de un servidor en el que tener almacenado el curso y que se encargue de montar el curso que finalmente disfrutará el alumno. Para ello nosotros usamos el servidor proporcionado por Apache, Tomcat 5.5.

Hemos tenido que incluir en el servidor, dentro de la carpeta *webapps*, y dentro a su vez de la carpeta correspondiente al proyecto, todos los archivos necesarios para que la aplicación funcione correctamente: todas las imágenes, todos los archivos: xml, html, y jsp. Además se ha de incluir en la carpeta **WEB-INF** los archivos .class de las clases que hemos implementado.

3.1.6 Diagramas UML

A continuación detallaremos mediante diagramas de secuencia, la comunicación entre las clases para los procesos más importantes de nuestro sistema.

Página de Inicio:

Cuando el alumno llega a nuestra página de inicio hay dos posibilidades, que esté ya registrado y pueda realizar el Login directamente o que no lo esté y deba registrarse en nuestro sistema para poder acceder al curso.

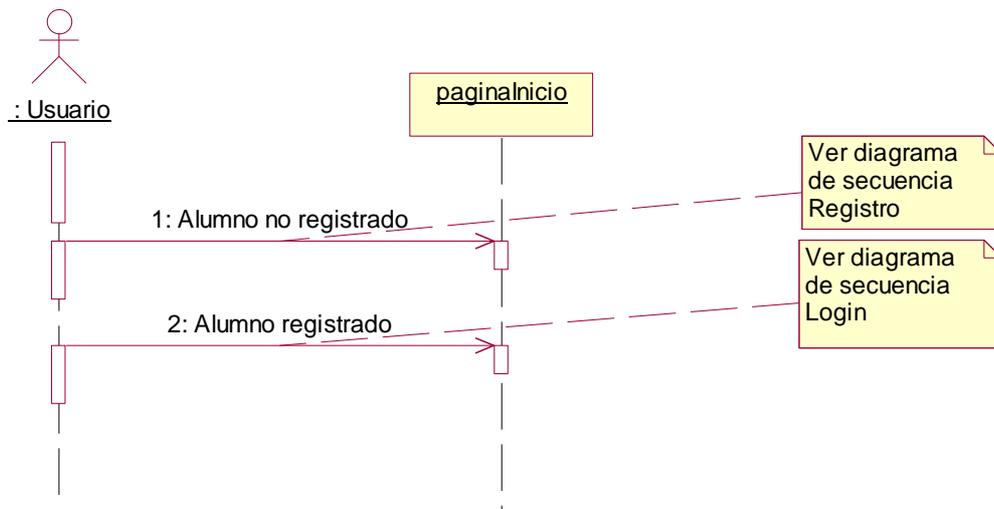


Fig. 2.1.6.1: Diagrama de la página inicial del sistema.

Login:

Diagrama de secuencia que muestra la interacción de la página de Inicio con la base de datos. De esta forma podremos comprobar si el alumno está ya registrado para poder acceder al sistema y mostrarle el tema correspondiente.

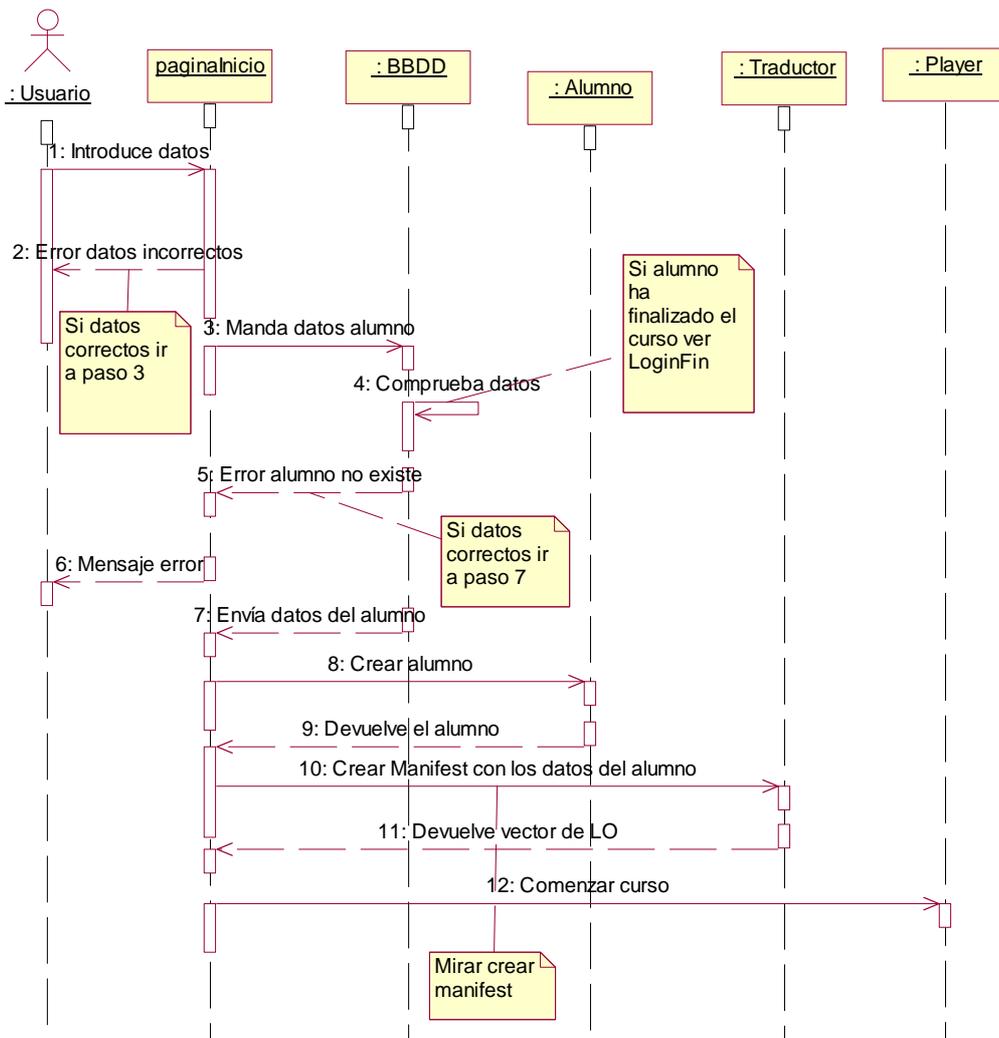


Fig. 2.1.6.2: Diagrama de Login.

LoginFin:

Este diagrama muestra la interacción de la página de Inicio con la base de datos. En este caso se trata del diagrama para un alumno que ya ha finalizado el curso anteriormente. Por lo tanto este diagrama es similar al anterior, sin embargo existen algunas diferencias importantes. El sistema recibirá siempre el primer tema y tendrá la posibilidad de cambiar al tema que quiera empleando el navegador izquierdo.

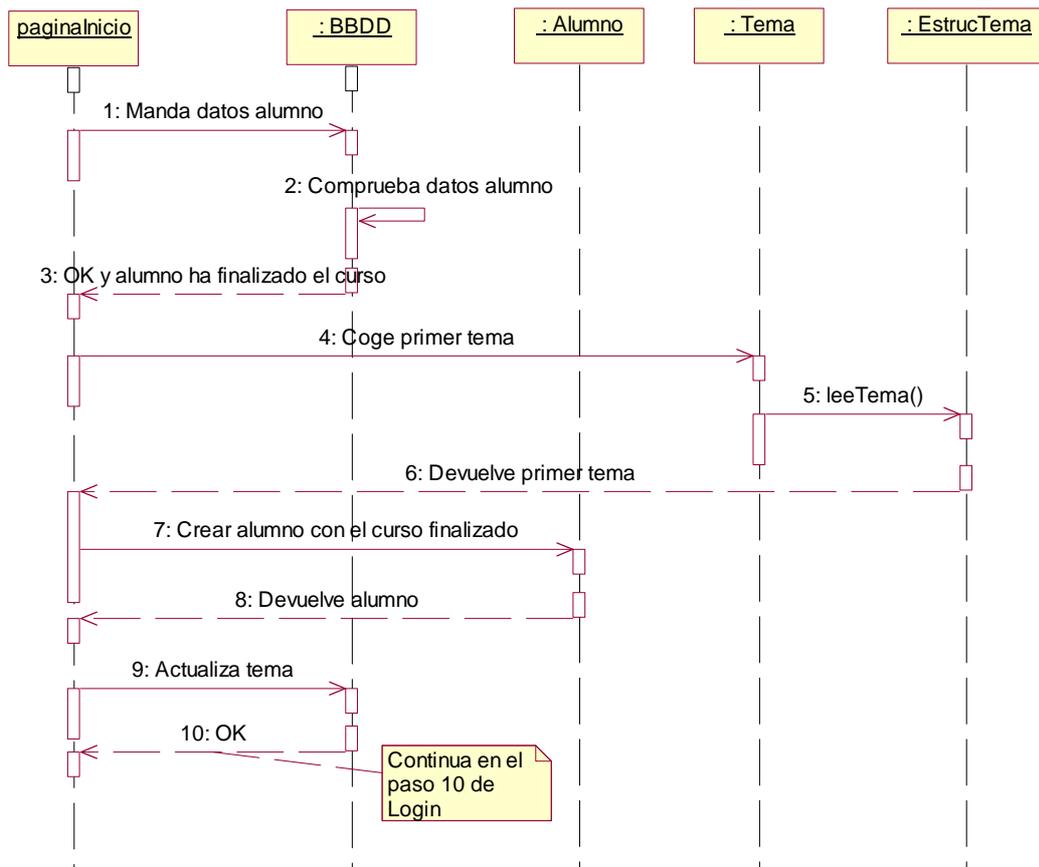


Fig. 2.1.6.3: Diagrama de LoginFin.

Registro:

Este diagrama muestra la interacción del usuario con la página de registro y ésta a su vez con la base de datos. El usuario debe introducir sus datos que serán comprobados (falta algún dato por introducir, el dni del alumno ya está registrado en la base de datos...) y si no hay ningún error será almacenado en nuestra base de datos habiéndole asignado el sistema automáticamente el primer tema.

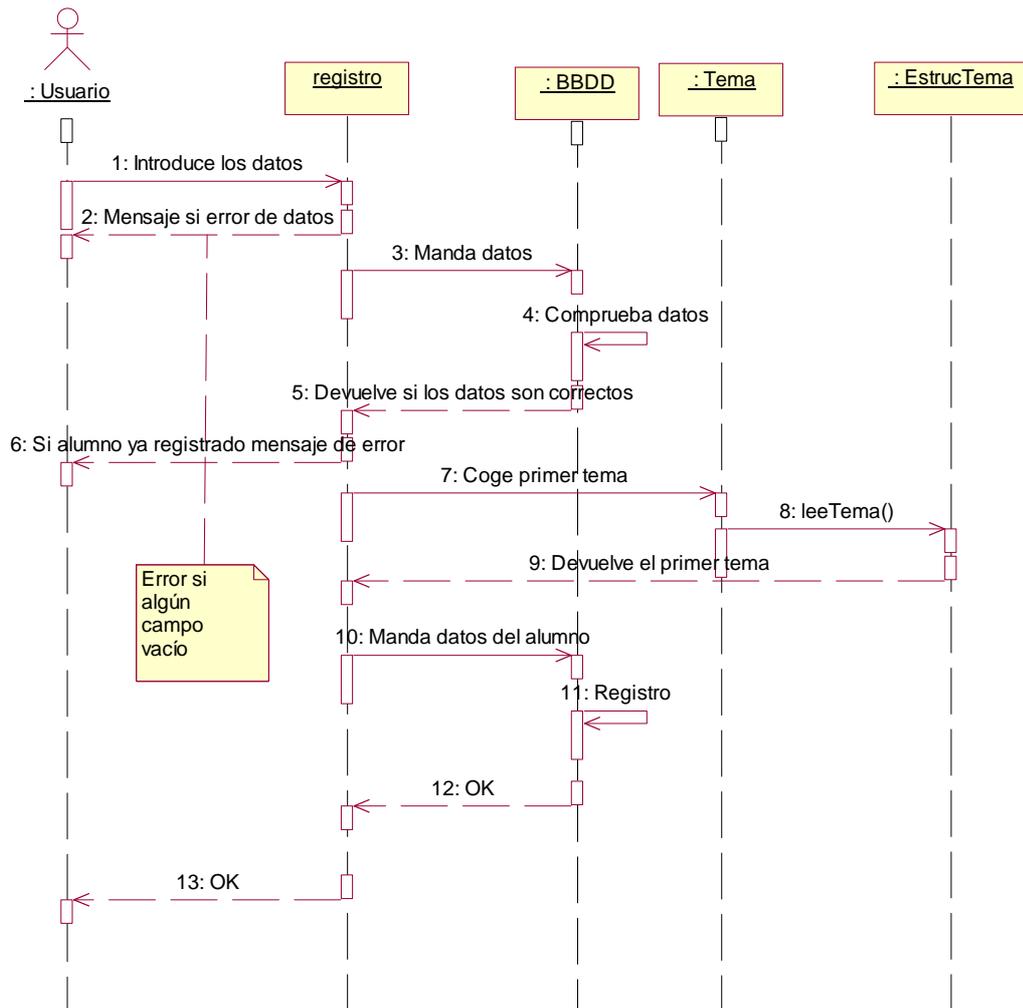


Fig. 2.1.6.4: Diagrama de Registro.

Visualizar Curso:

Este diagrama nos muestra cómo el usuario es capaz de interactuar con el sistema mediante la interfaz que será distinta dependiendo del número del tema por el que vaya el alumno en ese mismo instante. Si el alumno ha terminado el curso, se le mostrará un índice con todos los temas posibles para que pueda acceder libremente a cualquiera de ellos. Si por el contrario, el alumno no ha terminado el curso se le mostrarán en el índice todos los temas anteriores al actual.

Además, el alumno podrá avanzar o retroceder por los distintos Learning Objects gracias a las flechas contenidas en la interfaz.

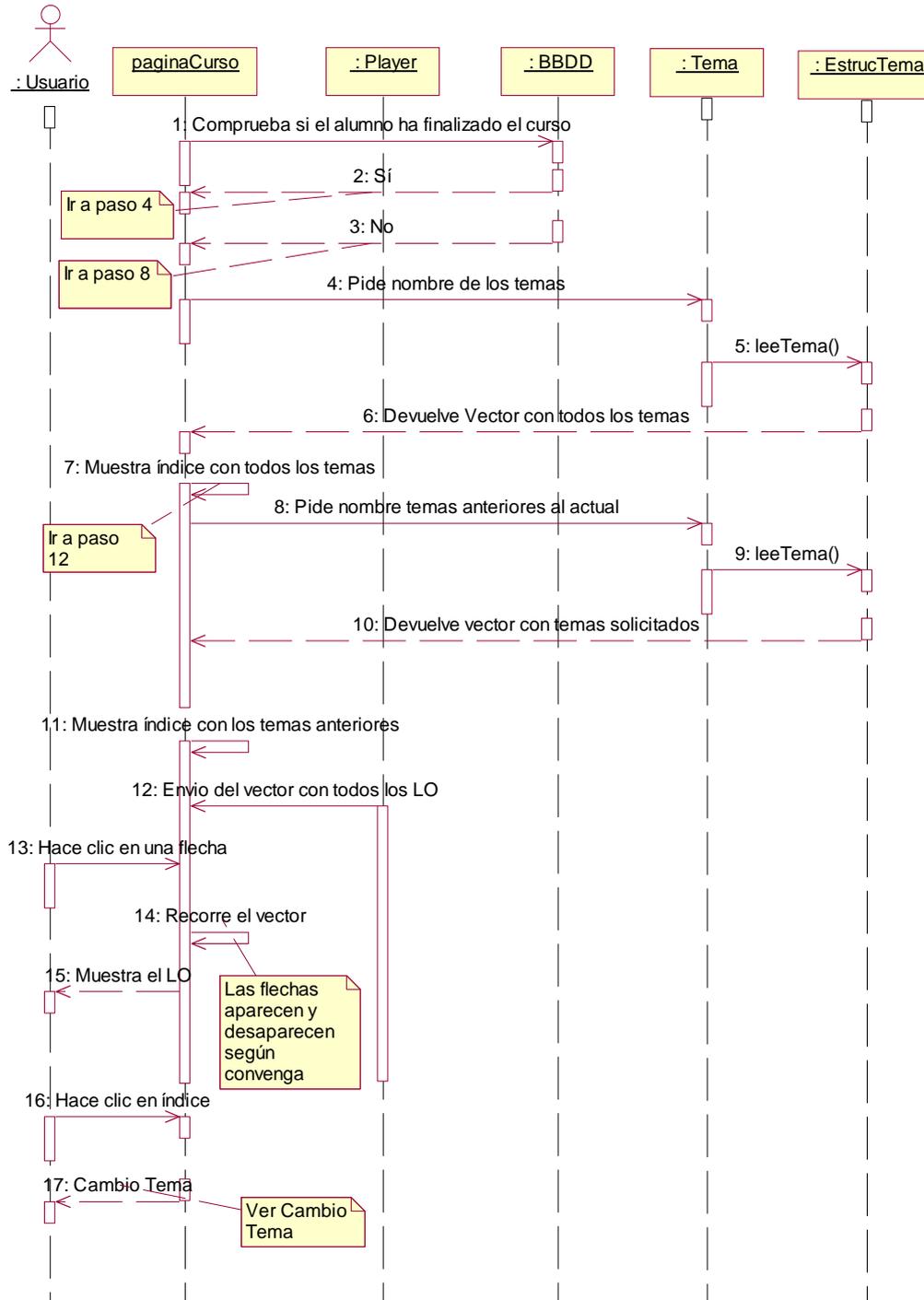


Fig. 2.1.6.5: Diagrama de Visualizar Curso.

Finalizar Tema:

Este diagrama muestra como el sistema comprueba en que tema se encuentra actualmente el alumno, genera el manifest para el siguiente tema y se lo muestra. La nueva información del alumno será actualizada en la base de datos.

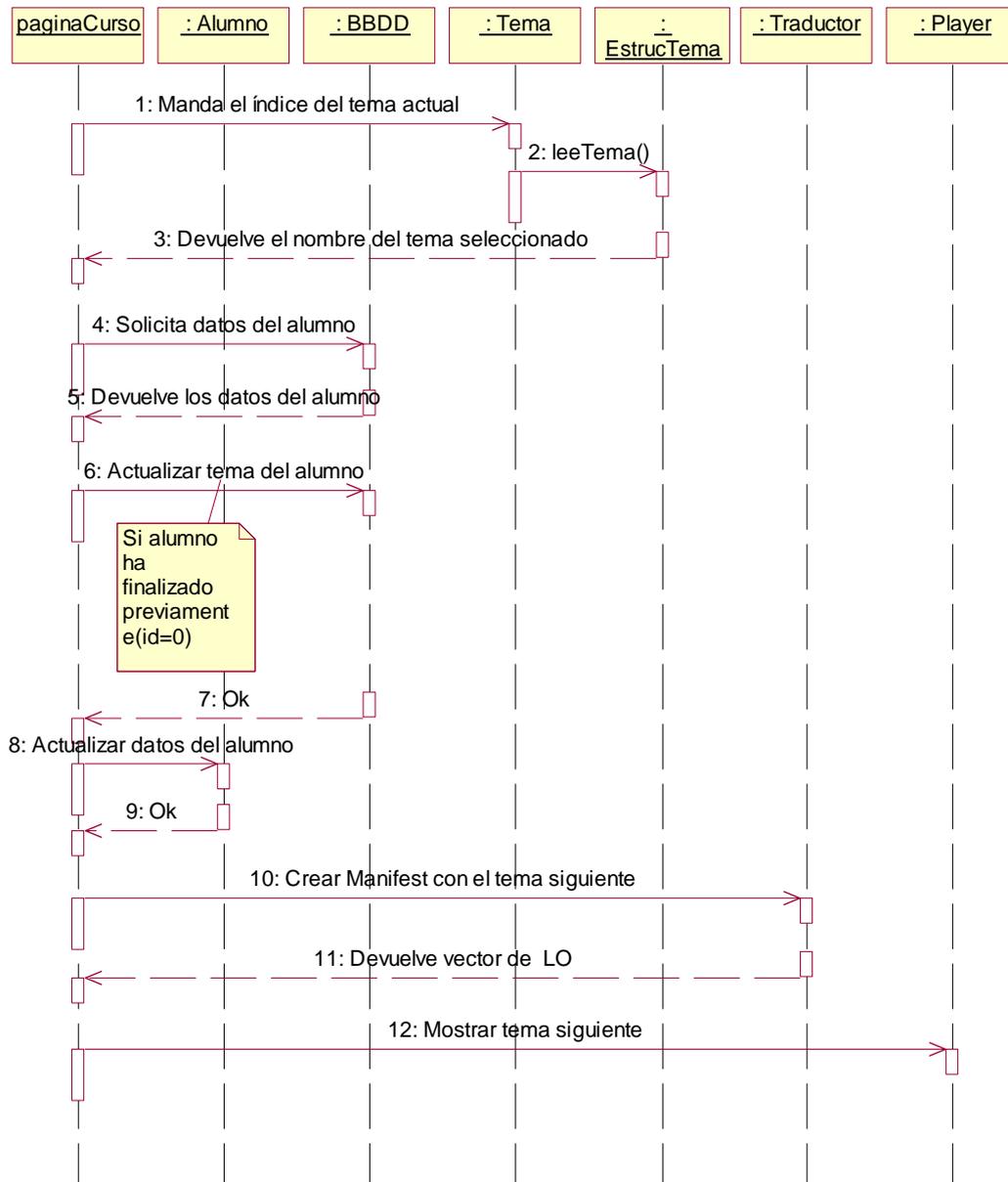


Fig. 2.1.6.6: Diagrama de Finalizar Tema.

Cambio Tema:

Este diagrama muestra como el sistema comprueba en que tema se encuentra actualmente el alumno, genera el manifest para tema que haya seleccionado y se lo muestra. La nueva información del alumno será actualizada en la base de datos.

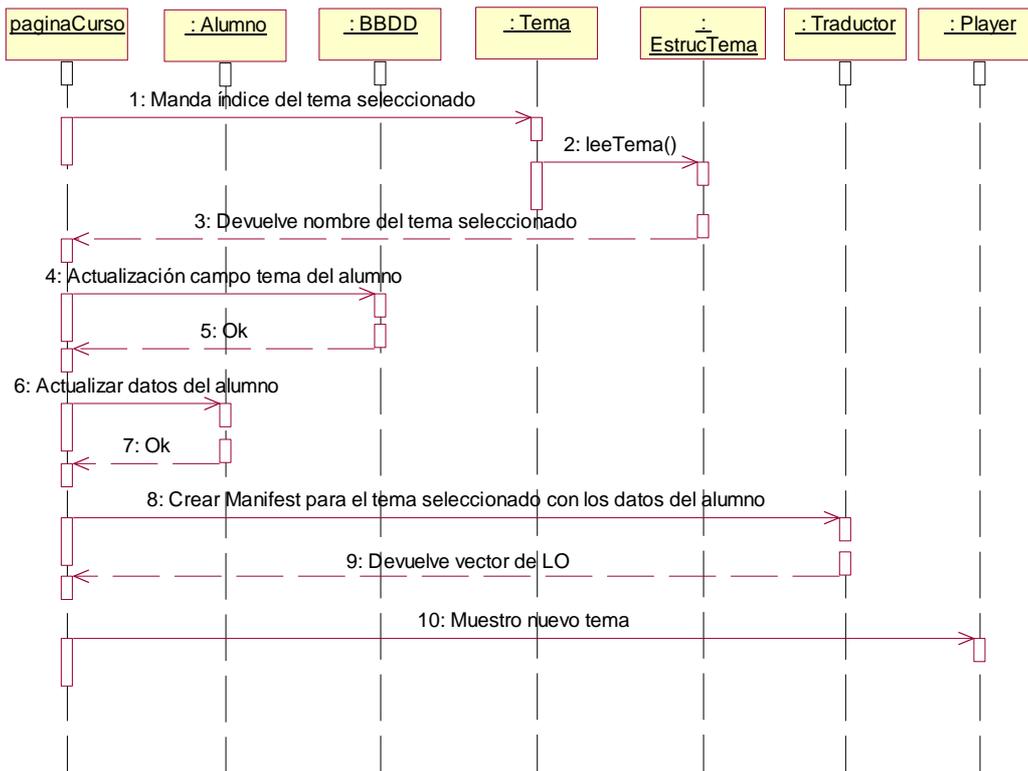


Fig. 2.1.6.7: Diagrama de Cambio Tema.

3.2 Funcionamiento interno del sistema

Para poder acceder a nuestro sistema, hemos creado una mini plataforma LMS (Learning Management System) dónde los usuarios pueden acceder vía Web y registrarse en el curso de Introducción a la programación en C++ (Fig. 2.2.1).

Usuarios registrados

Dni:

Contraseña:

Autoregistro: Si usted no ha registrado todavía su Dni y Contraseña, pulse sobre el botón Autoregistro para registrarse como usuario. Después de cumplimentar el formulario de Autoregistro, puede empezar el curso.
[Autoregistro](#)

CURSO DE INTRODUCCIÓN A LA PROGRAMACIÓN EN C++



Creado por: Raúl Iglesias González, Antonio Ortiz-Repiso Villegas y Ramón Picazo Sotos



Fig. 2.2.1: Página principal de la Web del curso Introducción a la programación en C++.

En primer lugar, un usuario debe antes de nada registrarse en el sistema mediante su número de **DNI** y una **contraseña**, eligiendo **el estilo de aprendizaje** que quiere para realizar el curso (Fig. 2.2.2).



Fig. 2.2.2: Página de Registro del curso Introducción a la programación en C++.

Automáticamente este alumno quedará registrado en la base de datos, si no hay ningún problema, es decir que el DNI introducido por el usuario ya exista o que no haya introducido bien los datos, y se le asignará el primer tema. Esto se realiza mediante la asignación al campo Id y al campo Tema de la Base de Datos, el primer elemento del vector resultante de leer el fichero Tema.xml contenido en la carpeta taxon (se explicará más adelante).

En este archivo Tema.xml, se tiene la estructura del curso, el nombre de los temas, de los subtemas y el identificador de cada uno de los temas y subtemas. Lo que se realiza con este archivo es recorrerlo y almacenar en un vector el identificador y el nombre de cada uno de los temas.

Posteriormente, cuando el alumno se ha registrado lo que se hace es asignar al alumno recién registrado la primera posición de ese vector, que corresponde obviamente al primer tema.

Si el registro se ha realizado correctamente, se le mostrará la siguiente pantalla (Fig. 2.2.3):

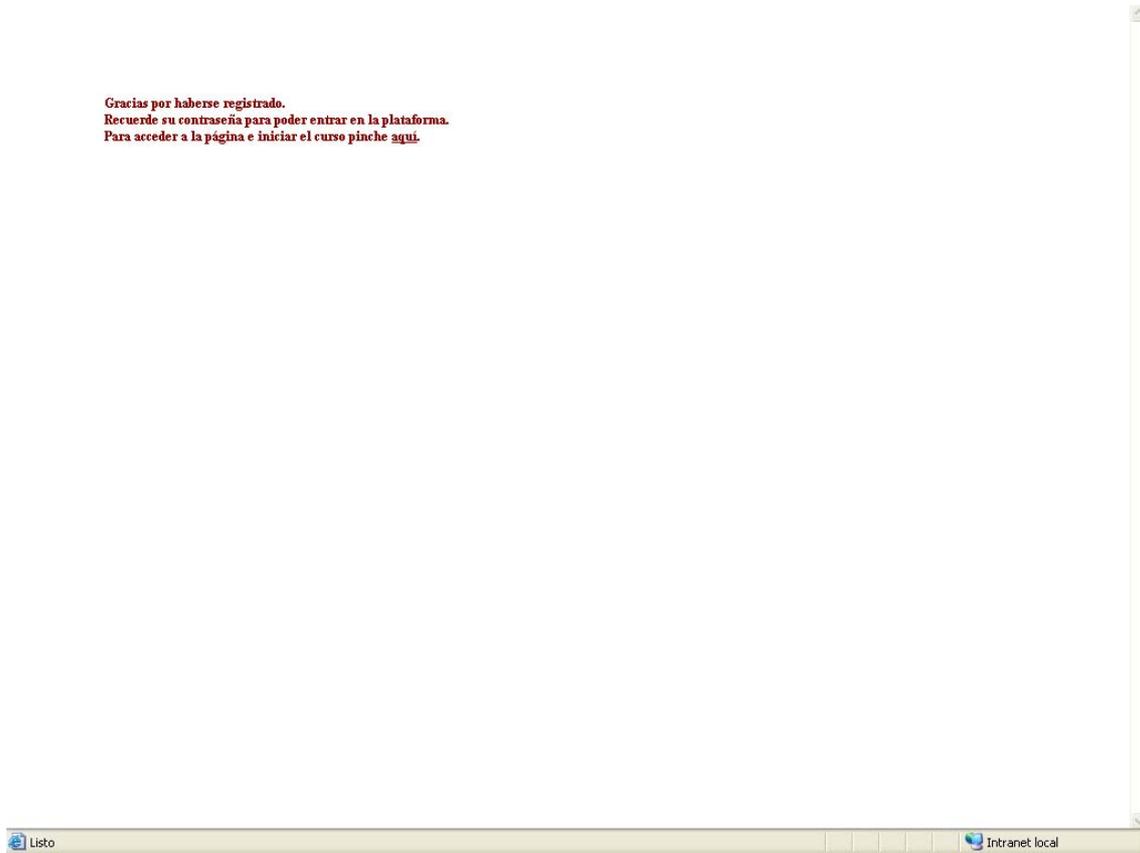


Fig. 2.2.3: Registro realizado correctamente. El usuario puede pinchar en el enlace para realizar el login.

Si por el contrario, ha ocurrido algún error al realizar el registro, se mostrarán una de las siguientes pantallas al usuario según el tipo de error que haya ocurrido:

The screenshot shows a web browser window with a registration form titled "REGISTRO". The form contains the following elements:

- Dni:** A text input field.
- Contraseña:** A text input field.
- Estilo de Aprendizaje:** A group of four radio buttons:
 - Verbal Activo
 - Verbal Reflexivo
 - Visual Activo
 - Visual Reflexivo
- Enviar:** A button.

Below the form, a red error message reads: "Debe rellenar los campos para autoregistrarse." The browser's taskbar at the bottom shows the system tray with "Listo" on the left and "Intranet.local" on the right.

Fig. 2.2.4: Error al realizar el Registro. Faltan campos por rellenar.



Fig. 2.2.5: Error al realizar el Registro. DNI ya existente en la BBDD.

Una vez se haya autorregistrado el alumno, este puede hacer el login usando los datos anteriores. Si el procedimiento es correcto, el sistema realiza los pasos siguientes:

1. Se busca en la base de datos al alumno que realiza el login, se cogen sus características y se crea un alumno con esas características.
2. Se crea un objeto de la clase Traductor de java, con el alumno anteriormente creado.
3. Se llama al método **construirmanifest** (“ruta del xml de la UoL”).
4. Una vez el manifest ha sido creado de forma **dinámica**, se recorre para ir mostrando las páginas en orden.

El método **construirmanifest** (“ruta del xml de la UoL”), es el que realiza todo el proceso.

A continuación mostramos el diagrama de secuencia que muestra el proceso realizado por la clase Traductor, cuya función es crear el manifest de manera dinámica. Mostramos las funciones que realiza y las carpetas a las que accede para consultar archivos XML necesarios para crear el manifest adecuadamente (Fig. 2.2.6).

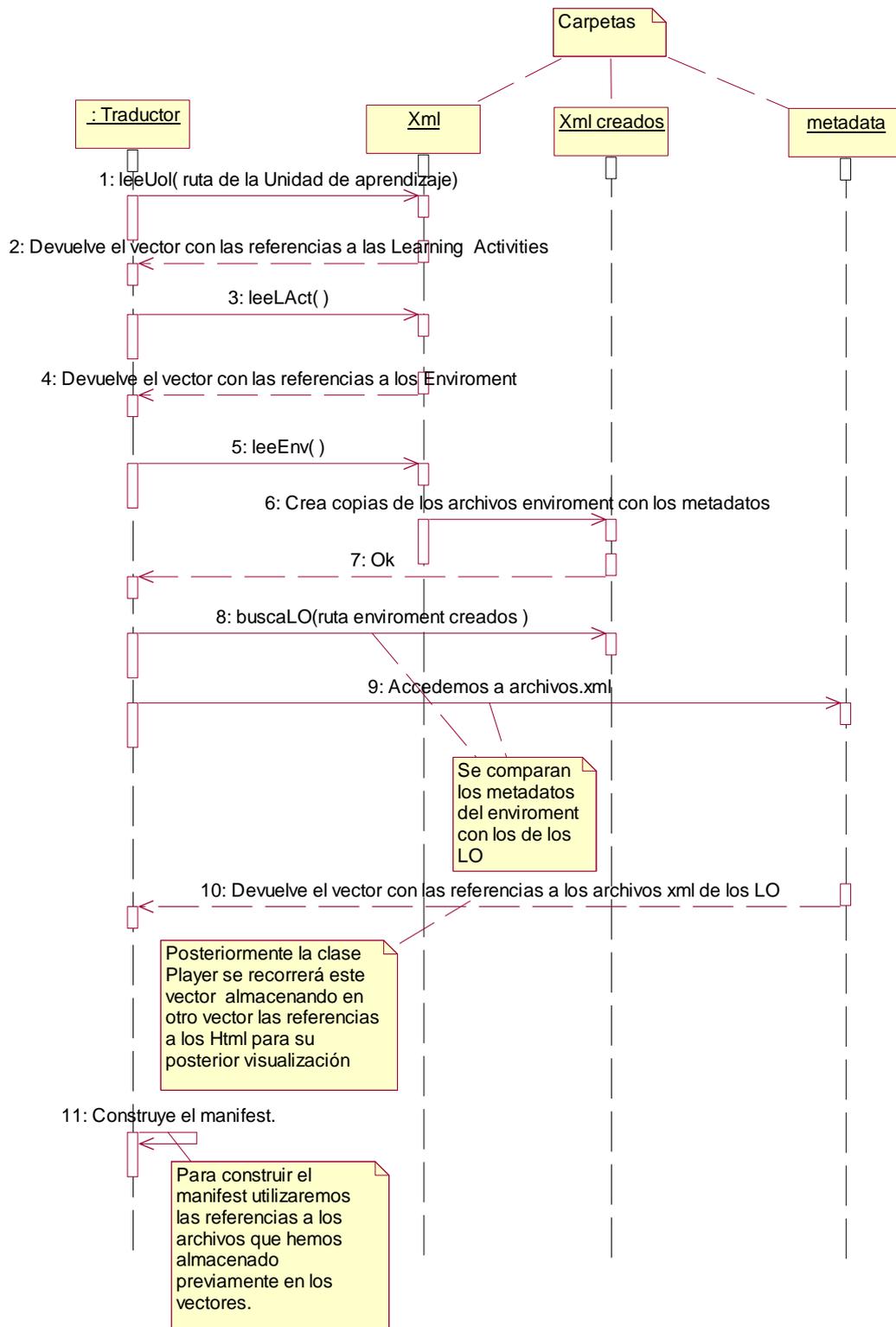


Fig. 2.2.6: Diagrama de secuencia que muestra las distintas etapas para construir el manifest.

En nuestro sistema, tenemos distintas carpetas que usamos para el proceso de construir el manifest:

- xml: dividida por temas, cada carpeta de un tema en concreto contiene 7 archivos .xml. El primer archivo es el de la Unit of Learning de ese tema. Este archivo, tiene referencias a otros 3 archivos, que son las Learning Activities. Son 3 archivos porque son las Learning Activities de Teoría, Ejemplos y Ejercicios. Cada archivo .xml de Learning Activity hace referencia a su Environment asociado, por lo tanto hay otros 3 archivos .xml de Environment que están inicialmente vacíos, sólo contienen el tag <tipoDoc> relleno según sea el Environment de Teoría, Ejemplos o Ejercicios.
- xmlcreados: en esta carpeta se crearán en la carpeta del tema correspondiente, 3 archivos .xml. Estos archivos son copias de los archivos .xml de Environment anteriormente mencionados, pero habiéndoles añadido todos los tags necesarios para posteriormente realizar la búsqueda de los Learning Objects adecuados. Los tags que se le añaden al archivo son <estilo> y <tema>, y se recogen de las características del alumno para el cual se está creando el manifest.
- metadata: dividida también por temas, cada carpeta de un tema contiene 3 carpetas más, Teoría, Ejemplo y Ejercicio, conteniendo cada una de ellas tantos archivos .xml como archivos .htm haya para ese tema. Estos archivos .xml, contienen toda la información relacionada con su archivo .htm asociado:
 - ruta de su .htm asociado
 - estilo de aprendizaje al que pertenece
 - tema
 - tipo de documento
 - prioridad

Además, cada carpeta de un tema concreto contiene un archivo .xml más. Este archivo se llama **archivos.xml**, y contiene referencias a todos y cada uno de los archivos .xml de esa carpeta.

- taxon: esta carpeta contiene los archivos .xml correspondientes a los taxon creados por nosotros. Estos archivos son:
 - estilosapr.xml: contiene los distintos estilos de aprendizaje que hemos definido para nuestro sistema.
 - ordenacion.xml: contiene la estructura requerida en los metadatos para poder ordenar el curso.
 - Prioridad.xml: archivo que contiene las distintas prioridades definidas para ordenar los .htm de manera adecuada.
 - TipDoc.xml: contiene la enumeración de los distintos tipos de documentos que hemos definido.
 - Tema.xml: contiene los identificadores y los nombres de los distintos temas que hemos incluido en el sistema.

- **Curso:** esta carpeta contiene todos los archivos htm del curso. Está dividida por temas y dentro de la carpeta de cada uno de los temas se divide en otras tres carpetas: Teoría, Ejemplo y Ejercicio, conteniendo cada una de ellas los distintos Learning Objects necesarios para construir el curso según el estilo de aprendizaje de cada alumno.

Una vez explicado todas las carpetas con los archivos correspondientes que contiene nuestro sistema, pasamos a explicar el procedimiento que sigue para construir el manifest de manera **dinámica**.

Supongamos que el alumno va por el tema de **Introducción al lenguaje C++** y es **Visual Activo**.

Primeramente, accede a la carpeta **xml/Introduccion al lenguaje C++**, en esta carpeta se recorre el archivo **Introduccion al lenguaje C++.xml**, y almacena en un vector las referencias a los archivos .xml de las Learning Activities. Cuando ya ha terminado de recorrerse el documento, realizamos el ordenamiento de las Learning Activities según el **estilo de aprendizaje** al que pertenece el alumno. Si es un alumno **activo**, modificamos el vector de tal manera que en la primera posición aparezca la referencia al archivo de **Ejemplo**, luego **Ejercicios** y finalmente **Teoría**. En cambio, si el alumno es **reflexivo**, se ordena de tal manera que sea primero la **Teoría**, **Ejemplo** y para terminar **Ejercicios**.

Una vez realizada la ordenación, nos recorremos el vector. Cada posición del vector es la referencia al archivo .xml asociado con la Learning Activity. Por lo tanto, para cada archivo, nos lo recorremos almacenando en otro vector las referencias a los archivos Environment. Una vez almacenadas todas las referencias a los archivos .xml de Environment, lo que hacemos es ir creando copias de cada uno de los 3 archivos .xml añadiendo los tags necesarios y guardándolos en la carpeta **xmlcreados/Introduccion al lenguaje C++**.

Lo que se realiza ahora es comparar con cada archivo .xml de la carpeta **metadata/Introduccion al lenguaje C++**, si cumple las características añadidas a los archivos de Environment anteriormente creados. Para ello, nos recorremos en un primer lugar el archivo **archivos.xml**, que nos devuelve una lista con todas las referencias de los archivos existentes en la carpeta del tema correspondiente. El archivo .xml que cumpla las condiciones, se guarda su referencia en un vector. Una vez creado dicho vector, se procede a hacer la ordenación de dichas referencias a los Learning Objects por orden decreciente de prioridad. Los Learning Objects que más prioridad tengan han de mostrarse con antelación a los menos prioritarios. En nuestro sistema la prioridad máxima se representa por: **prioridad1**, la siguiente sería **prioridad2** y así sucesivamente.

Después de hacer dicha ordenación por prioridades, hacemos una nueva distinción entre los diferentes estilos de aprendizaje. En este caso se trata de mostrar diferente número de archivos de teoría, que de ejemplos y de ejercicios. Así a un alumno que sea activo se le mostrarán mayor número de archivos de ejemplos que de ejercicios. Por el contrario si el estilo del alumno es reflexivo se le muestran menos archivos de ejercicios que de ejemplos. En cuanto a los archivos de teoría se le muestran todos, puesto que consideramos que en un curso de programación la parte teórica es indispensable para el aprendizaje.

Finalmente se procede a construir el manifest. Se escribe en un fichero la cabecera del archivo <manifest>, y se procede a recorrer todos los vectores anteriormente creados ya que cada uno de ellos contiene las referencias a archivos .xml. Por lo tanto se empieza copiando el contenido del archivo **Introduccion al lenguaje C++.xml**, posteriormente se copia el contenido en orden de los ficheros cuyas referencias están en el vector que contiene las referencias a las Learning Activities. Se realiza el mismo procedimiento con el vector que contiene las referencias a los ficheros .xml de Environment creados y finalmente se recorre el último vector que contiene las referencias de los .xml de los metadata de los Learning Objects y se copia el contenido de estos ficheros en el archivo manifest que se está creando. Una vez copiados todos los ficheros, se cierra la cabecera </manifest> y ya se tiene creado el archivo manifest.xml de manera dinámica quedando de esta manera:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <manifest>

- <activity-structure identifier=" Introduccion al lenguaje C++.xml" >
  <title>Introduccion</title>
  <learning-activity-ref ref="IntroTLAct.xml" />
  <learning-activity-ref ref="IntroEjemLAct.xml" />
  <learning-activity-ref ref="IntroEjerLAct.xml" />
</activity-structure>

- <learning-activity identifier="IntroEjemLAct.xml" isvisible="true">
  <title>Introduccion Ejemplos</title>
  <environment-ref ref="IntroEjemEnv.xml" />
- <metadata>
  <tipDoc>Ejemplo</tipDoc>
</metadata>
</learning-activity>

- <learning-activity identifier="IntroEjerLAct.xml" isvisible="true">
  <title>Introduccion Ejercicios</title>
  <environment-ref ref="IntroEjerEnv.xml" />
- <metadata>
  <tipDoc>Ejercicio</tipDoc>
</metadata>
</learning-activity>

- <learning-activity identifier="IntroTLAct.xml" isvisible="true">
  <title>Introduccion Teoria</title>
  <environment-ref ref="IntroTEnv.xml" />
- <metadata>
  <tipDoc>Teoria</tipDoc>
</metadata>
</learning-activity>
```

```

- <environment identifier="IntroEjemEnv.xml">
  <title>Introduccion Ejemplos</title>
-  <metadata>
    <tipDoc>Ejemplo</tipDoc>
    <estilo>verbal activo</estilo>
    <tema>Introduccion al lenguaje C++ </tema>
  </metadata>
</environment>

- <environment identifier="IntroEjerEnv.xml">
  <title>Introduccion Ejercicios</title>
-  <metadata>
    <tipDoc>Ejercicio</tipDoc>
    <estilo>verbal activo</estilo>
    <tema>Introduccion al lenguaje C++ </tema>
  </metadata>
</environment>

- <environment identifier="IntroTEnv.xml">
  <title>Introduccion Teoria</title>
-  <metadata>
    <tipDoc>Teoria</tipDoc>
    <estilo>verbal activo</estilo>
    <tema>Introduccion al lenguaje C++ </tema>
  </metadata>
</environment>

- <learning-object identifier="Ejemplo/ejemplo1.xml" href="Curso/Introduccion al
  lenguaje C++/Ejemplo/ejemplo1.htm" isvisible="true">
  <title>EstrucProgVerbRef</title>
-  <metadata>
    <estilo>Todos</estilo>
    <tema>Introduccion al lenguaje C++ </tema>
    <tipDoc>Ejemplo</tipDoc>
    <prioridad>prioridad1 </prioridad>
  </metadata>
</learning-object>

- <learning-object identifier="Teoria/concepbasicosVerAct.xml"
  href="Curso/Introduccion al lenguaje
  C++/Teoria/concepbasicosVerAct.htm" isvisible="true">
  <title>ConcepBasicosVerAct</title>
-  <metadata>
    <estilo>Verbal Activo</estilo>
    <tema>Introduccion al lenguaje C++ </tema>
    <tipDoc>Teoria</tipDoc>
    <prioridad>prioridad2 </prioridad>
  </metadata>
</learning-object>

```

```
- <learning-object identifier="Teoria/estrucprogVerAct.xml"
  href="Curso/Introduccion al lenguaje C++/Teoria/estrucprogVerAct.htm"
  isvisible="true">
  <title>EstrucProgVerbAct</title>
- <metadata>
  <estilo>Verbal Activo</estilo>
  <tema>Introduccion al lenguaje C++</tema>
  <tipDoc>Teoria</tipDoc>
  <prioridad>prioridad3</prioridad>
</metadata>
</learning-object>

</manifest>
```

Lo único que queda por hacer es recorrerse este archivo y coger las referencias de los archivos .htm y guardarlas en un vector que se recorrerá para ir mostrando las páginas en orden. Para ello cogemos cada apartado <learning-object> y de cada uno de ellos almacenamos el contenido del atributo **href**.

```
<learning-object identifier="Ejemplo/ejemplo1.xml"
  href="Curso/Introduccion al lenguaje C++/Ejemplo/ejemplo1.htm"
  isvisible="true">
  <title>EstrucProgVerbRef</title>
- <metadata>
  <estilo>Todos</estilo>
  <tema>Introduccion al lenguaje C++</tema>
  <tipDoc>Ejemplo</tipDoc>
  <prioridad>prioridad1</prioridad>
</metadata>
</learning-object>
```

Por lo tanto, si el procedimiento de login es correcto se realiza todo este proceso mostrando al alumno la primera página del tema por el que vaya según su estilo de aprendizaje (en nuestro ejemplo el primer tema de forma visual activa), y se le deja total libertad para que estudie de forma tranquila y a su ritmo (Fig. 2.2.7).



Fig. 2.2.7: Primera página del tema **Introducción al lenguaje C++** del curso para un alumno **Visual Activo**.

Si por el contrario el procedimiento de login no es correcto, se muestran, como anteriormente en el caso de registrarse, las siguientes pantallas según el tipo de error:

Debe rellenar los campos para hacer el login

Usuarios registrados

Dni:

Contraseña:

Autorregistro: Si usted no ha registrado todavía su Dni y Contraseña, pulse sobre el botón Autorregistro para registrarse como usuario. Después de cumplimentar el formulario de Autorregistro, puede empezar el curso.

[Autorregistro](#)

CURSO DE INTRODUCCIÓN A LA PROGRAMACIÓN EN C++



Creado por: Raúl Iglesias González, Antonio Ortiz-Repiso Villegas y Ramón Picazo Sotos



Fig. 2.2.8: Error al realizar el login. Campos del formulario sin rellenar.

El Dni/Contraseña no corresponde a ningún usuario

Usuarios registrados

Dni:

Contraseña:

Autorregistro: Si usted no ha registrado todavía su Dni y Contraseña, pulse sobre el botón Autorregistro para registrarse como usuario. Después de cumplimentar el formulario de Autorregistro, puede empezar el curso.

[Autorregistro](#)

CURSO DE INTRODUCCIÓN A LA PROGRAMACIÓN EN C++



Creado por: Raúl Iglesias González, Antonio Ortiz-Repiso Villegas y Ramón Picazo Sotos



Fig. 2.2.9: Error al realizar el login. El DNI y la contraseña no corresponden a ningún usuario.

Cuando un alumno ha llegado al final de un tema, le aparecerá en la pantalla en la esquina inferior derecha un botón de **Finalizar Tema** (Fig. 2.2.10). Al pulsar este botón, el sistema automáticamente actualizará en la base de datos los campos del alumno asignándole el siguiente tema y se le mostrará por pantalla la primera página del tema siguiente acorde a su estilo de aprendizaje (Fig. 2.2.11).

La actualización en la base de datos se realiza consultando primero en que tema se encuentra actualmente el alumno. Después se obtiene el vector anteriormente mencionado, que contiene la estructura en orden de los temas con su identificador y el nombre del tema, y se compara el identificador del tema actual y se busca en ese vector. Una vez haya sido encontrado se verifica que no es el último tema, si no lo es, se pasa a la siguiente posición del vector, se cogen el identificador y el nombre del tema y se actualiza la base de datos. Si es el último tema, lo que se realiza es la actualización de la base de datos asignando un "0" al identificador y poniendo "Fin Curso" en el campo del tema y se muestra al alumno una página donde se le indica que ha concluido el curso (Fig. 2.2.12).

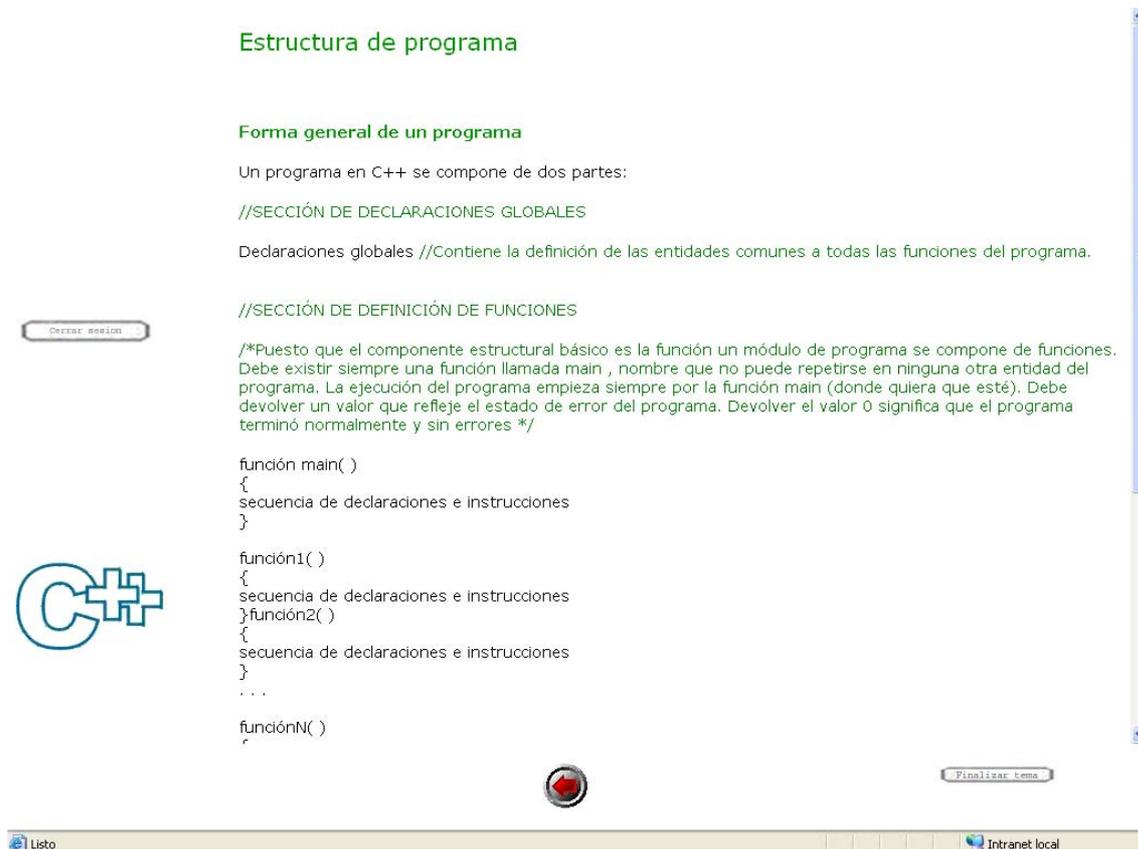


Fig. 2.2.10: Fin del tema. Botón Finalizar Tema.

Ejercicios Resueltos: Tipos estructurados

Abstract

Ejercicios resueltos sobre los tipos estructurados: Arrays (unidimensional y multidimensional), Registros.

1. Diseña una estructura de registro adecuada para almacenar la siguiente información de los libros de una biblioteca: Título, Páginas, Autor, Fecha de impresión

Solución:

```
struct TipoFecha{
    int dia, mes, ano;
};

TipoLibro{
    char Titulo [50];
    unsigned int paginas;
    char Autor [100];
    TipoFecha Fecha;
};

TipoLibro Libro;
```

2. Escriba un programa que cree un array con los caracteres de la 'a' a la 'f'. Después sustituya todas las vocales por el caracter '#'

Solución:

```
int main(){
    char arrayCaracteres [] = {'a', 'b', 'c', 'd', 'e', 'f'};
    int tamanoArray = sizeof(arrayCaracteres) / sizeof(char);
    char caracter;
    for (int i = 0; i < tamanoArray; i++) {
```

1. Introducción al lenguaje C++

Cerrar sesión



Fig. 2.2.11: Primera página del tema siguiente acorde al estilo de aprendizaje del alumno.

HA COMPLETADO CON ÉXITO
EL CURSO DE
INTRODUCCIÓN AL LENGUAJE C++
¡¡ENHORABUENA!!

Cerrar sesión

Fig. 2.2.12: Última página del curso. Muestra al alumno la finalización del mismo.

Además, cuando un alumno ha terminado ya uno o varios temas, se le da la posibilidad de poder acceder a los temas ya terminados para consultar o recordar alguno de los conocimientos previamente estudiados.

Para ello, le mostramos un índice en el navegador izquierdo de la pantalla con los distintos temas a los cuales puede acceder y un botón para que pueda hacer clic y dirigirse al tema elegido (Fig. 2.2.13).

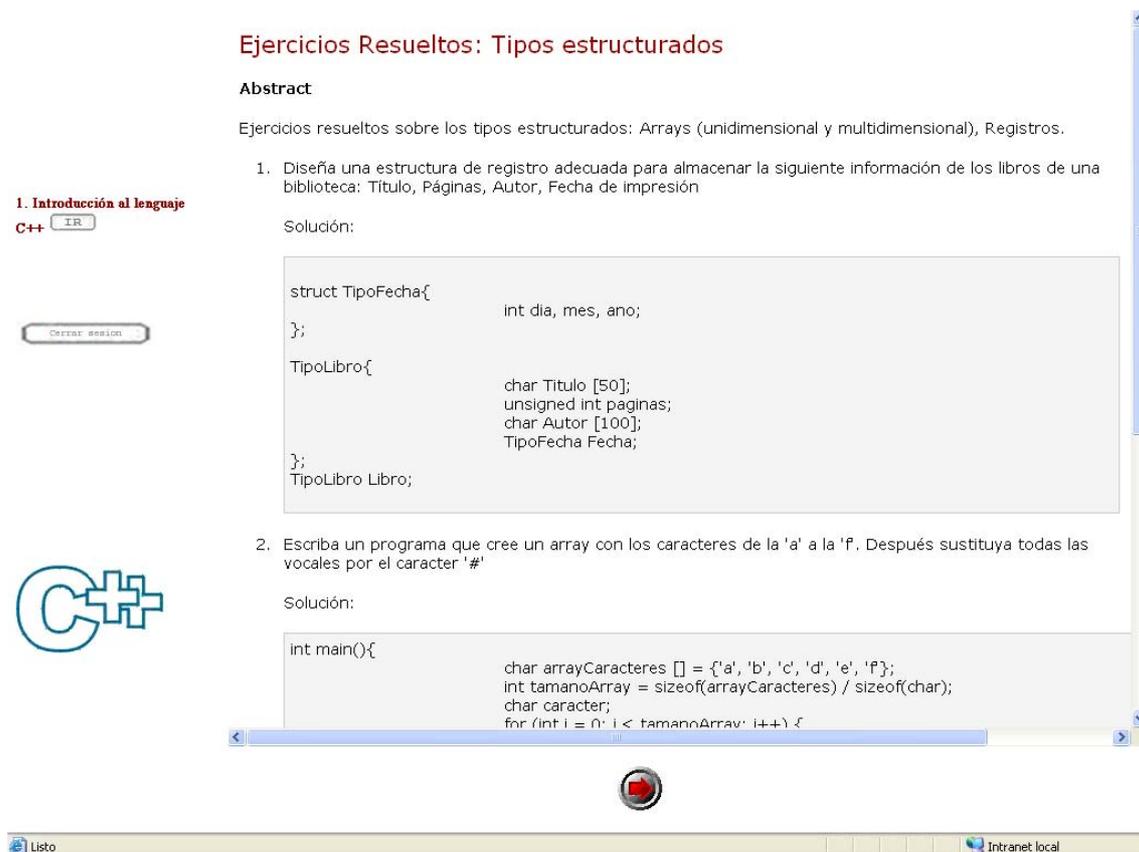


Fig. 2.2.13: Índice en el navegador izquierdo con los distintos temas a los que puede ir.

Al pulsar ese botón, el sistema crea de nuevo el manifest de forma dinámica como previamente hemos explicado y muestra al alumno la primera página del tema que quiere visualizar acorde a su estilo de aprendizaje. En la figura 2.2.13 el alumno está estudiando el 2º tema, por lo tanto en el índice sólo le aparece la opción para ir al primer tema.

Cuando un alumno ha terminado el curso y decide volver a entrar para revisar los contenidos, en el índice se le muestran todos y cada uno de los temas para que pueda ir directamente a cualquier tema sin tener que recorrerse innecesariamente el curso entero de nuevo (Fig. 2.2.14). El mecanismo para mostrarle la primera página del tema elegido es exactamente el mismo que el anteriormente explicado, el manifest siempre se crea de forma dinámica.

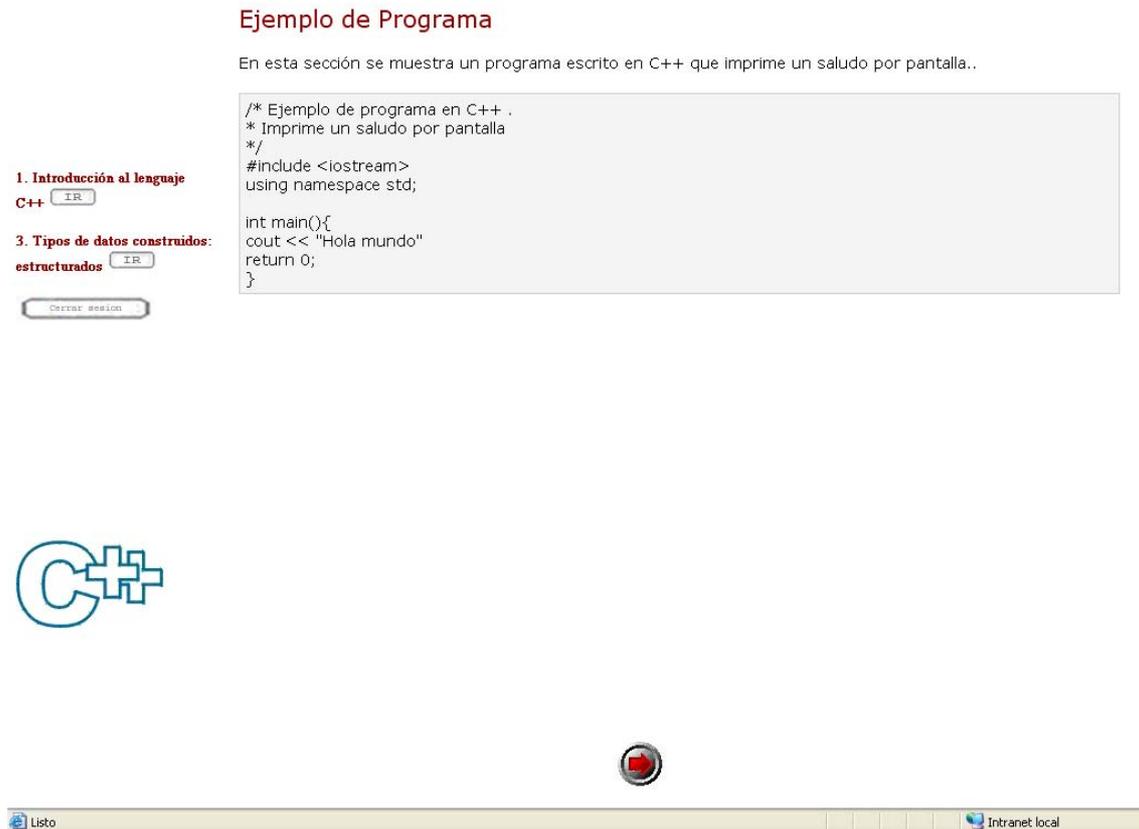


Fig. 2.2.14: Aparecen todos los temas en el índice cuando el alumno ha finalizado el curso.

En este ejemplo el alumno ha terminado ya el curso, por lo tanto la siguiente vez que visite nuestro curso para revisar conceptos, le aparecerán en el índice todos los temas a los cuales puede dirigirse. En cuanto acceda, se le muestra siempre la primera página del primer tema según el estilo de aprendizaje del alumno.

Cabe destacar además, que las páginas de teoría son distintas para los alumnos pertenecientes al estilo de aprendizaje Visual o Verbal. El contenido de las mismas está explicado de forma distinta para cada uno de ellos. En el caso de los alumnos verbales, el contenido es más denso y teórico mientras que por el contrario, para los alumnos visuales el contenido teórico es más esquemático.

Para terminar, se puede observar en las figuras anteriores que hay un botón "Cerrar Sesión" en el navegador izquierdo del curso. Este botón es para cerrar el curso si el alumno ya no quiere seguir estudiando y prefiere dejarlo para otro momento. Cuando el alumno vuelva a entrar en nuestro curso, se le mostrará la primera página del tema en el cual se quedó.

Con respecto a la estructura de la base de datos, se trata de una única tabla llamada **Alumnos**. Los atributos que tiene son:

- DNI
- Pwd (contraseña)
- Estilo (estilo de aprendizaje del alumno)
- Id (identificador del tema actual del alumno)
- Tema (nombre del tema actual del alumno)

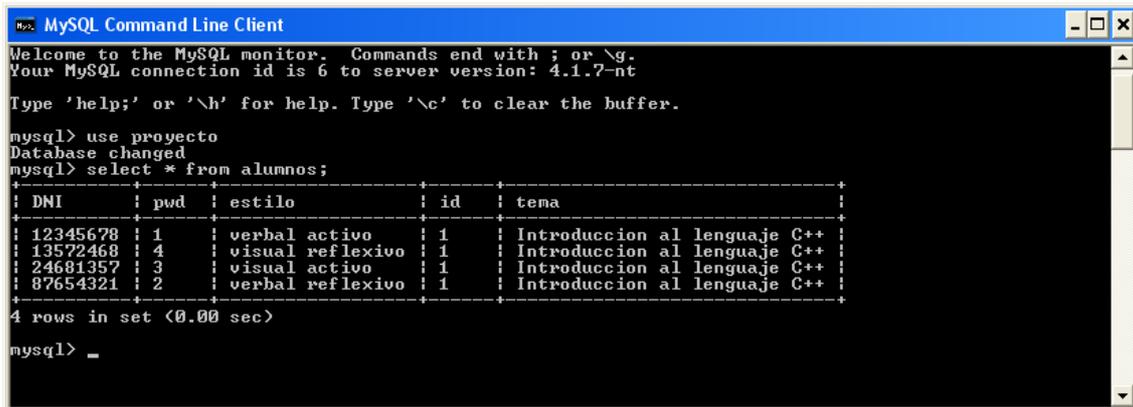


Fig. 2.2.15: Tabla Alumnos de la base de datos. Muestra los alumnos registrados en el curso.

Como se puede observar, todos los alumnos están registrados con el primer tema de Introducción al lenguaje C++. Si entramos con el alumno **visual activo** como anteriormente y llegamos a pasar al siguiente tema, la actualización de la base de datos se haría como se ha explicado anteriormente y quedaría de la siguiente manera:

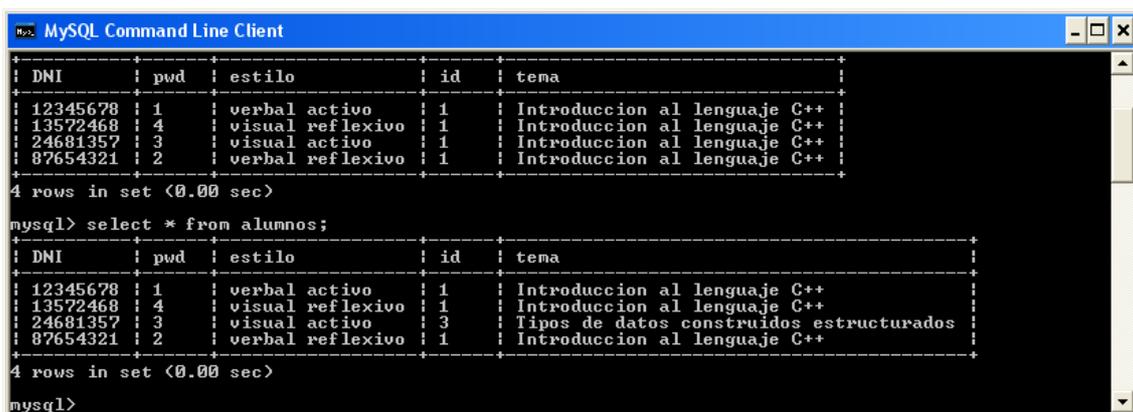


Fig. 2.2.16: Tabla Alumnos actualizada. Se observa que el tema y el id han cambiado.

Si además, terminamos el curso con este mismo alumno, se actualizará la base de datos como hemos explicado anteriormente, es decir, en el campo id se asignará un “0” y en el campo tema “Fin Curso”.

```

mysql> use proyecto
Database changed
mysql> select * from alumnos;
+-----+-----+-----+-----+-----+
| DNI      | pwd | estilo          | id | tema                               |
+-----+-----+-----+-----+-----+
| 12345678 | 1   | verbal activo   | 1  | Introduccion al lenguaje C++      |
| 13572468 | 4   | visual reflexivo | 3  | Introduccion al lenguaje C++      |
| 24681357 | 3   | visual activo   | 0  | Fin Curso                          |
| 87654321 | 2   | verbal reflexivo | 3  | Introduccion al lenguaje C++      |
+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql>
    
```

Fig. 2.2.17: Alumno ha terminado el curso. Se observa que el tema y el id han cambiado.

Si volvemos a entrar al curso con este mismo alumno, observaremos que el id ya no cambiará nunca, se mantendrá siempre a “0” indicando que ese alumno ha finalizado el curso y lo que irá actualizándose es el campo tema, que irá cambiando para poder saber en que tema se encuentra el alumno cuando este decida cambiar de tema bien pulsando el botón finalizar tema o bien pulsando en algún tema del índice. Esto es necesario ya que necesitamos tener una referencia del tema actual para poder pasarle a otro tema.

```

mysql> select * from alumnos;
+-----+-----+-----+-----+-----+
| DNI      | pwd | estilo          | id | tema                               |
+-----+-----+-----+-----+-----+
| 12345678 | 1   | verbal activo   | 1  | Introduccion al lenguaje C++      |
| 13572468 | 4   | visual reflexivo | 3  | Introduccion al lenguaje C++      |
| 24681357 | 3   | visual activo   | 0  | Fin Curso                          |
| 87654321 | 2   | verbal reflexivo | 3  | Introduccion al lenguaje C++      |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from alumnos;
+-----+-----+-----+-----+-----+
| DNI      | pwd | estilo          | id | tema                               |
+-----+-----+-----+-----+-----+
| 12345678 | 1   | verbal activo   | 1  | Introduccion al lenguaje C++      |
| 13572468 | 4   | visual reflexivo | 3  | Introduccion al lenguaje C++      |
| 24681357 | 3   | visual activo   | 0  | Introduccion al lenguaje C++      |
| 87654321 | 2   | verbal reflexivo | 3  | Introduccion al lenguaje C++      |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
    
```

Fig. 2.2.18: Alumno ha terminado el curso. Se observa que el tema ha cambiado pero el id=0.

4 Sistemas de características similares

4.1 WebCT



WebCT es una plataforma informática de teleformación (e-learning) que permite construir cursos interactivos e impartir formación a través de Internet, llevando a cabo la tutorización y el seguimiento de los alumnos. Para ello dispone de datos referentes al tiempo, lugar y fecha en la que los alumnos han visitado cada zona del curso. Facilita la creación de un sofisticado entorno educacional basado en la *World Wide Web*. Puede usarse para crear un curso completo *on-line*, o simplemente para publicar material suplementario a la docencia presencial. Esta aplicación permite también, hacer un análisis estadístico exhaustivo, individualmente o para un grupo de alumnos determinado, de los resultados de los ejercicios o exámenes.

WebCT cuenta con infinidad de herramientas de comunicación, contenidos de evaluación y estudio. Así mismo, permite una inigualable flexibilidad en la personalización de la presentación de un curso *on-line*, así como en el tipo de archivos que permite incorporar a dicho curso. Por este motivo pueden incorporarse, por ejemplo, archivos de audio y vídeo en los que el alumno puede leer un texto y mediante la activación de un botón escuchar la pronunciación de una persona nativa, y lo que es esencial para la enseñanza de idiomas, la posibilidad de poder oírlo cuantas veces quiera. En el caso de los archivos de vídeo, la imagen apoya al sonido y la comprensión se hace más fácil y amena.

Es una aplicación que proporciona un entorno educativo flexible donde los alumnos pueden, además de aprender, compartir experiencias y conocimientos con comunidades virtuales compuestas por usuarios del sistema, ofreciendo un conjunto de herramientas de comunicación (foro, chat, correo), evaluación (autoevaluación, corrección automática, etc. cuyos resultados se salvaguardan en áreas protegidas mediante *password* para su posterior procesamiento estadístico). A través de WebCT estudiantes y profesores pueden interactuar aún cuando no se encuentren en el mismo espacio físico.

La plataforma WebCT fue desarrollada en el Departamento de Ciencias de la Computación de la Universidad de la Columbia Británica (Canadá). Sus creadores, Murray W. Goldberg y Sasan Salari, comenzaron el desarrollo de WebCT gracias a una beca en 1995. La primera versión del producto, que apareció en el mercado en 1997 sólo pretendía aplicar las nuevas tecnologías a la preparación de cursos para enriquecer las experiencias docentes con sus alumnos, pero pronto aparece el dólar. En 1999, compra el proyecto la Universal Learning Technology Co. (ULT), una empresa de Massachussets (EE UU), fundada y dirigida por Carol Vallone, una de las mujeres más influyentes en el mundo de la ciencia y la tecnología. Actualmente se comercializa con criterios empresariales: un buen producto a un precio razonable (*WebCT Pricing & Licensing*).

Hoy en día la WebCT es una herramienta utilizada en más de 2.211 universidades y más 148.460 facultades. Cuenta con más de 379.407 cursos y más de 10.663.141 estudiantes la utilizan en más de 80 países de todo el mundo.



Fig. 4.1. Ejemplo curso con WebCT

Versiones

WebCT tiene actualmente dos versiones: WebCT *Vista* y WebCT *Campus Edition*. *Vista* es la versión profesional completa dirigida a empresas, y *Campus Edition* (Versión Universitaria) es ofrecida a instituciones que ya tienen servicios tales como sistemas de almacenamiento de archivos y herramientas para registro de cursos.

El programa no tiene un coste establecido, pero este depende del tamaño de la institución y de como va ser usada.

Libros y publicaciones

El programa también es usado para realizar publicaciones electrónicas. Para poder usar un libro de texto u otra herramienta de aprendizaje publicada en formato WebCT, algunos editores requieren que los estudiantes compren un password o código en una librería o por Internet. El programa también permite la integración del material preparado localmente con el material comprado de las editoriales.

Crítica

WebCT, especialmente la versión *Vista*, rompe muchas de las pautas de uso de Internet. Dentro de los problemas presentados se tienen: Deshabilita la opción del botón retroceder en los navegadores de Internet, utiliza Javascripts con URL y el uso de pestañas (lengüetas o tabs) en las páginas desplegadas se vuelve deficiente. Muchos usuarios manifiestan dificultades al entrar o acceder al contenido de las páginas o subir trabajos o asignaciones de las clases, independientemente de si usan conexiones de Internet por teléfono (dial-up) o banda ancha.

WebCT *Vista* es actualmente el tema de un pleito jurídico (La Marca vs. Capella University) en alegaciones de que no es accesible a los estudiantes con discapacidades. Este pleito creará, probablemente, una jurisprudencia que tendrá ramificaciones en la necesidad de diseñar programas online que sean completamente accesibles a todos los estudiantes.

4.2 Blackboard



Blackboard

Blackboard Inc. (NASDAQ: BBBB) es una compañía de software con sede en Washington, DC, EEUU. Fundada en 1997, Blackboard comenzó como firma consultora con un contrato con la organización sin fines de lucro IMS Global Learning Consortium (<http://www.imsglobal.org>). En 1998, Blackboard LLC se fusionó con CourseInfo LLC, una pequeña compañía proveedora de programas de administración de cursos originaria de la Universidad de Cornell. La nueva compañía se conoció como Blackboard Inc. La primera línea de productos de aprendizaje en línea (e-learning) fue llamada Blackboard Courseinfo, pero luego el nombre Courseinfo fue desechado en el 2000. Blackboard se convirtió en una compañía con acciones al público en Junio de 2004. A fecha de 2005, Blackboard desarrolló y licenció aplicaciones de programas empresariales y servicios relacionados a más de 2200 instituciones educativas en más de 60 países. Estas instituciones usan el programa de BlackBoard para administrar aprendizaje en línea (e-learning), procesamiento de transacciones, comercio electrónico (e-commerce), y manejo de comunidades en línea (online).

Fusión con WebCT

El 12 de octubre de 2005, BlackBoard Inc. anunció planes de fusión con WebCT, una compañía rival de programas de aprendizaje en línea, pero el anuncio suscitó rápidamente inquietudes en los expertos antimonopolio del Departamento de Justicia de EEUU. Se acordó que la compañía fusionada llevaría el nombre de **BlackBoard**. Blackboard Inc. completó la fusión con WebCT el 28 de febrero de 2006, dirigida por el Presidente y CEO de BlackBoard, Michael Chasen.

Productos BlackBoard

- Blackboard Academic Suite está compuesto por:
 - Blackboard Learning System, un entorno de manejo de cursos
 - Blackboard Community System, para comunidades en línea y sistemas de portales.
 - Blackboard Content System, un sistema para el manejo de contenido
- Blackboard Commerce Suite, está compuesto por:
 - Blackboard Transaction System, un sistema de procesamiento de transacciones (tarjeta débito) para identificaciones de universidades.
 - Blackboard Community System, Un sistema para transacciones de comercio electrónico
 - Bb One, una red comercial para procesar transacciones de tarjetas débito patrocinadas por BlackBoard

Blackboard también tiene una arquitectura abierta, llamada Building Blocks (<http://buildingblocks.blackboard.com>), que puede usarse para extender la funcionalidad de los productos Blackboard o integrarlos con otros sistemas de programas.

4.3 Moodle



Moodle es una plataforma de aprendizaje a distancia que es de software libre. Tiene una relativamente grande y creciente base de usuarios. Es un sistema de gestión de la enseñanza (course management system o learning management system en inglés), es decir, una aplicación diseñada para ayudar a los educadores a crear cursos de calidad en línea. Estos tipos de sistema de aprendizaje a distancia a veces son también llamados ambientes de aprendizaje virtual o educación en línea.

Moodle es una aplicación web que puede funcionar en cualquier ordenador (computador) en el que pueda correr PHP, y soporta varios tipos de bases de datos (en especial MySQL y PostgreSQL).

Moodle fue creado por Martin Dougiamas, quien trabajó como administrador de WebCT en la Universidad Curtin, y se basó en trabajos sobre el **constructivismo en pedagogía, que afirman que el conocimiento se construye en la mente del estudiante en lugar de ser transmitido sin cambios a partir de libros o enseñanzas. Un profesor que opera desde este punto de vista crea un ambiente centrado en el estudiante que lo ayuda a construir ese conocimiento en base a sus habilidades y conocimientos propios en lugar de simplemente publicar y transmitir la información que consideran que los estudiantes deben conocer.**

Moodle ha venido evolucionando desde 1999 y nuevas versiones siguen siendo producidas. En enero de 2005, la base de usuarios registrados incluye 2.600 sitios en más de 100 países y está traducido a más de 50 idiomas. El sitio más grande reporta tener actualmente 6.000 cursos y 30.000 estudiantes.

La palabra Moodle era al principio un acrónimo de Modular Object-Oriented Dynamic Learning Environment (Entorno de Aprendizaje Dinámico Orientado a Objetos y Modular), lo que resulta fundamentalmente útil para programadores y teóricos de la educación. También es un verbo que describe el proceso de deambular perezosamente a través de algo, y hacer las cosas cuando se te ocurre hacerlas, una placentera chapuza que a menudo te lleva a la visión y la creatividad. Las dos acepciones se aplican a la manera en que se desarrolló Moodle y a la manera en que un estudiante o profesor podría aproximarse al estudio o enseñanza de un curso en línea.

Ventajas

Una de las características más atractivas de Moodle, que también aparece en otros gestores de contenido educativo, es la posibilidad de que los alumnos participen en la creación de glosarios, y en todas las lecciones se generan automáticamente enlaces a las palabras incluidas en estos.

Otra es que se pueden crear cuestionarios muy rápidamente importando ficheros de texto con la forma

```
¿Quién descubrió América? {  
  =Cristóbal Colón  
  ~Américo Vespucio  
  ~Hernán Cortés  
  ~Abraham Lincoln  
}
```

Desventajas

Algunas actividades pueden ser un poco mecánicas. La creación de actividades desde el entorno web es un poco lenta; se puede instalar localmente la aplicación para evitarlo, pero entonces tenemos que instalar un paquete específico que incluye Apache y PHP.

4.4 Otras herramientas

4.4.1 TelEduc



TelEduc es un ambiente de enseñanza a distancia por el cual se pueden realizar cursos a través de Internet. Está siendo desarrollado conjuntamente por el Núcleo de Informática Aplicada a la Educación (Nied) y por el Instituto de Computación (IC) de la Universidad Estatal de Campinas (Unicamp).

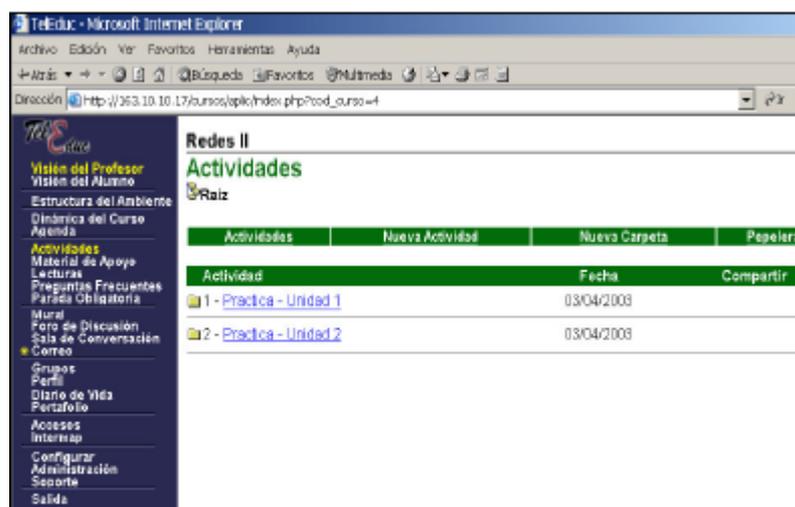


Fig.4.4.1.1 Ejemplo de curso usando la herramienta TelEduc.

4.4.2 DotLRN



.LRN es una comunidad global de educadores, diseñadores, y programadores que comparten todos juntos la dirección de la innovación educativa. Se trata de software de código abierto.

4.4.3 Claroline.net

claroline.net

Claroline es software libre basado en PHP/MySQL. Es un Sistema de Administración de cursos basados en Web LCMS. Este sistema permite a maestros (profesores, conferenciantes) e instituciones crear y administrar sitios web que pueden ser visualizados en un navegador web Internet Explorer, Netscape, Mozilla, etc.).

4.4.4 Dokeos



Dokeos es un entorno de e-learning y una aplicación de administración de contenidos de cursos y también una herramienta de colaboración. Es software libre y esta bajo la licencia GNU GPL, el desarrollo es internacional y colaborativo. Es también certificado por la OSI y puede ser usado como un sistema de administración de contenido para educación y educadores

4.5 Estándares

Cada uno de los grupos que han desarrollado estas herramientas ha creado sus propios formatos de estructuración de contenidos y almacenamiento de información, lo que trae como consecuencia la imposibilidad de interoperar entre las distintas plataformas, o crear productos únicos que funcionen con distintas plataformas. Todo debido a esta incompatibilidad de formatos. Conscientes de la necesidad de establecer y mejorar este intercambio, y cooperación, distintas organizaciones y empresas relacionadas con el mundo del software, la capacitación y la educación están trabajando en la creación de estándares y especificaciones que logren crear plataformas, materiales y recursos interoperables entre distintas herramientas LMS(Learning Management System), además de la creación de nuevos productos que se dediquen a generar estos recursos interoperables, o que ayuden a transformar los existentes.

Los estándares más importantes son:

- IEEE
- IMS
- AIC
- ADL/SCORM

En cuanto a los estándares que han utilizado las tres grandes herramientas explicadas anteriormente, Moodle ha creado varios estándares: un estándar para empaquetar contenidos, se han creado metadatos para escribir los recursos educativos, se ha estandarizado la información sobre los perfiles de los alumnos, se ha trabajado en la interoperabilidad entre tests y cuestionarios, en las metodologías, la secuenciación de los contenidos, la definición de las competencias, los repositorios digitales, la matriculación de los alumnos y la accesibilidad de los sistemas. Pero a la hora de intercambiar cursos con otra plataforma, todos estos estándares no sirven de nada si por un lado no se ha seguido un estándar común como base, o si no se poseen funcionalidades que permita ese intercambio. Este es el caso de Moodle, ya que puede importar sin problemas paquetes creados según el estándar SCORM 1.3 (SCORM 2004). Además hay también utilidades varias para ayudar a la migración y demos de cómo pasar por ejemplo de WebCT a Moodle. Y cabe destacar también que es posible la integración con LAMS(es un editor visual de secuencias de actividades didácticas "inspirado" en el estándar IMS Learning Design) mediante una actividad o un formato de curso.

En el caso de WebCT, lo que ha desarrollado son paquetes de contenidos, llamados e-packs (paquetes electrónicos) consistentes en materiales de aprendizaje prediseñados y que pueden ser adquiridos e incorporados automáticamente a los contenidos de un curso.

Por último destacar que es el estándar IMS el que más empresas adheridas tiene, y en cuyas filas se encuentran tanto empresas del mundo del software y hardware, como instituciones educativas. Entre sus miembros podemos encontrar a Microsoft, University Of Cambridge, Boeing, Apple, California State University, Sun Microsystems, Oracle, Cisco, **WebCT** y **BlackBoard**.

4.6 Herramientas con personalización

Existen herramientas que permiten la personalización de los cursos atendiendo a las preferencias de los alumnos. Una de ellas es ATutor que permite al usuario actuar sobre diferentes modos de aprendizaje:

- ✓ Modo Visual.
- ✓ Modo Verbal.
- ✓ Modo Cinético.



ATutor es un Sistema de Gestión de Contenidos de Aprendizaje - Learning Content Management System (LCMS) de código abierto, que se utiliza para presentar contenido educativo o instruccional en la web, o impartir cursos independientes online, y está diseñado teniendo en cuenta la accesibilidad y adaptabilidad.

ATutor se instala y se actualiza en unos minutos, y tiene la posibilidad de desarrollar plantillas personalizadas para cambiar todo el aspecto de la aplicación. El sistema permite a los tutores o profesores construir, empaquetar y redistribuir contenido educativo basado en tecnología web, importar y exportar contenido, e impartir cursos online. Los alumnos pueden aprender en un entorno que se adapta a sus necesidades.

ATutor es el primer LCMS inclusivo, que cumple las especificaciones de accesibilidad del W3C (WCAG 1.0) nivel AA, proporcionando acceso a todos los potenciales alumnos, profesores y administradores, incluyendo aquellos con discapacidades que pueden estar accediendo a través de tecnologías asistidas. La conformidad con las especificaciones XHTML 1.0 del W3C garantiza que ATutor se presente de manera consistente en cualquier tecnología que soporte los estándares web.

También adopta las especificaciones de empaquetado de contenido IMS/SCORM, posibilitando la reutilización de contenido que puede ser intercambiado entre los diferentes sistemas de e-learning. El contenido creado en otro sistema compatible con IMS o SCORM puede ser importado en ATutor, y viceversa.

ATutor funciona con Apache, MySQL y PHP, se distribuye gratuitamente bajo licencia GPL.

4.7 Conclusiones

Las tres grandes herramientas explicadas anteriormente (WebCT, BlackBoard y Moodle) comprenden la mayoría de cursos que están desarrollados, basándose en e-learning, en la red. Todos ellos poseen la facultad de que, a la hora de la implementación de un curso, cada profesor puede personalizarlo a su gusto y a las preferencias de los alumnos a los que vaya dirigido. Pero dicha personalización la llevan a cabo los profesores antes de que los alumnos participen y no se va adaptando a las necesidades cambiantes de los alumnos. Además no da la opción a los alumnos de personalizar el curso por ellos mismos.

Como ya hemos explicado, si existen herramientas que permitan a los usuarios definirse de una manera o de otra, y a partir de ahí, mostrarles el curso de una u otra manera. Pero esta funcionalidad también es incompleta, pues una vez el alumno se ha definido de una manera no se modifica posteriormente.

Es por tanto, algo novedoso implementar un curso e-learning, que no solo sea personalizable tanto por el profesor, como por el alumno, si no que cree de manera dinámica el curso, consiguiendo que se adapte no sólo al alumno si no al estado en el que se encuentre el estilo de aprendizaje de dicho alumno en cada momento.

5 Diferencias entre la especificación y nuestro sistema

Anteriormente se explicó en el apartado 1.3 de este mismo documento la especificación IMS Learning Design. Con la explicación aportada en el apartado 2.2 sobre el funcionamiento del sistema y con la información que vamos a añadir a continuación, vamos a ver cuales han sido las diferencias entre el sistema implementado y el teórico resultado que habríamos obtenido siguiendo en todos los niveles la especificación IMS Learning Design.

Como se explica en la especificación, en el nivel A de dicho estándar se diferencia entre **Activity Structures**, **Learning Activities**, **Environments** y **Learning Objects**. Todos ellos contenidos dentro de una **Unidad de Aprendizaje** (Unit of Learning, UoL).

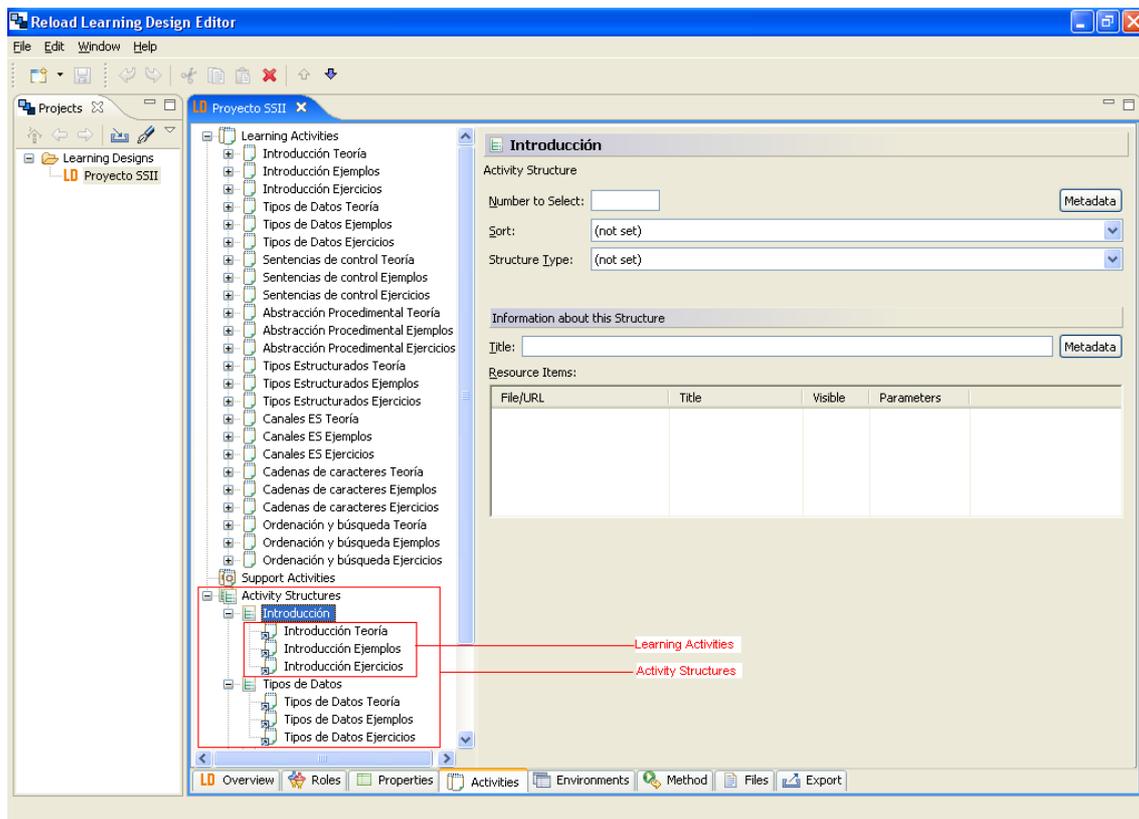


Fig. 4.1: Activity Structures.

Cada Activity Structure contiene diferentes Learning Activities que están asociadas a él (Fig. 4.1).

Asimismo, cada uno de los Learning Activities tiene asociado un Environment (Fig. 4.2).

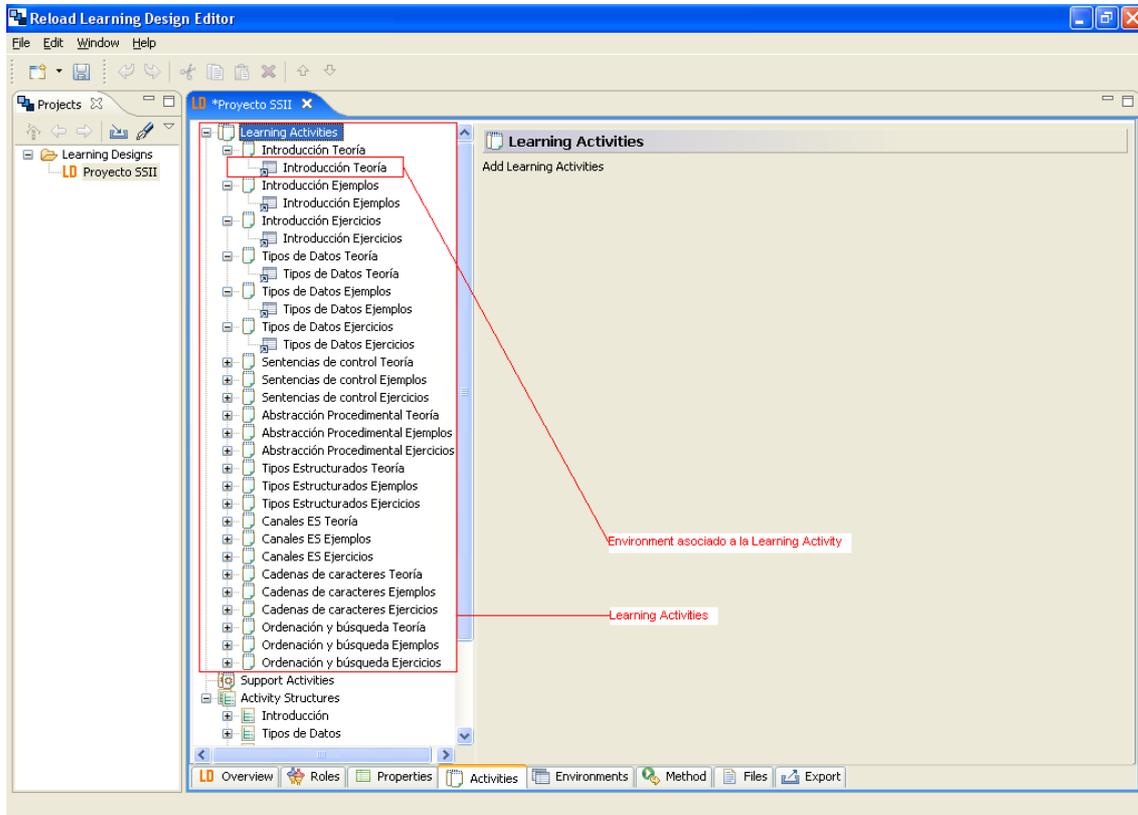


Fig. 4.2: Learning Activities.

Y cada uno de los Environment tiene asociado una serie de Learning Objects diferentes (Fig. 4.3).

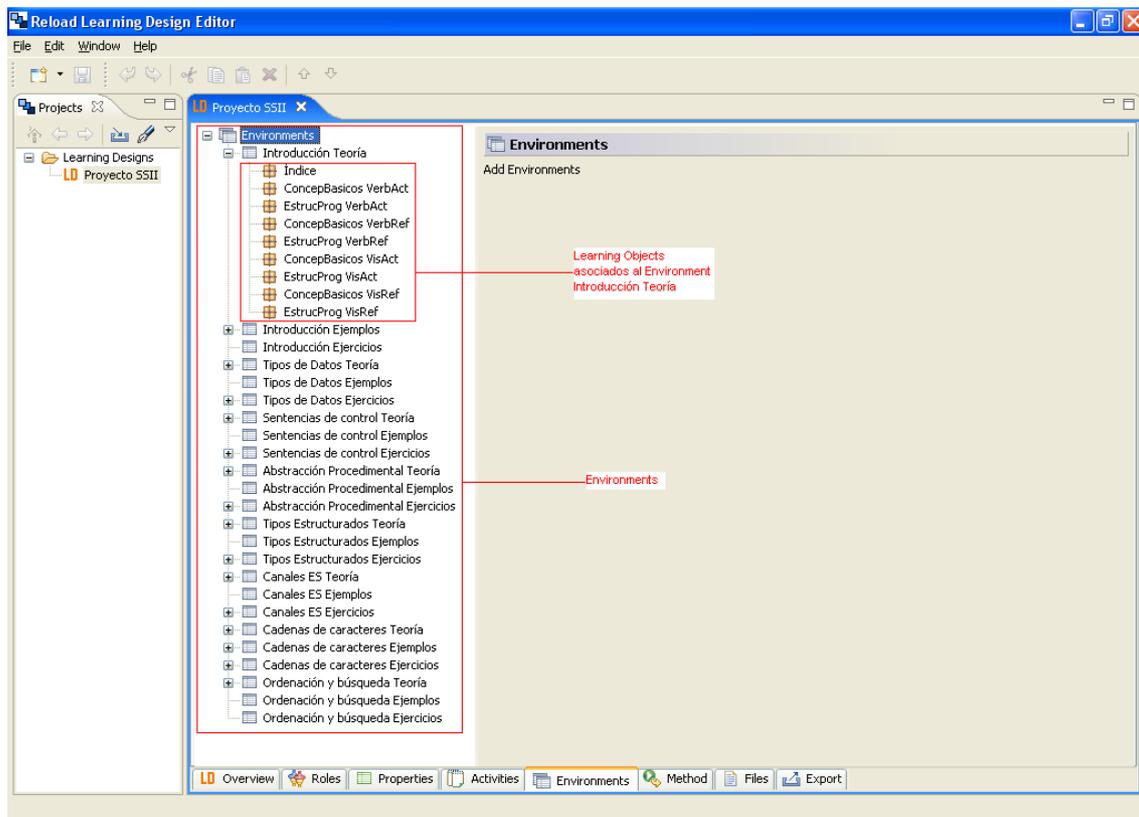


Fig. 4.3: Environments.

Es por todas estas asociaciones que para realizar nuestro sistema, nos hemos ayudado de la herramienta Reload Editor Learning Design, ya que nos facilitaba el trabajo de añadir estructuras y asociar las learning activities.

Esta herramienta cumple a raja tabla la especificación, es por ello que no hay ninguna diferencia entre la implementación de nuestro sistema y lo que a la especificación se refiere en el nivel A.

Para cumplir fielmente lo que en la especificación se cita, añadimos todo lo necesario a través de la herramienta Reload Editor (Activity Structures, Learning Activities, Environments, Learning Objects), ya que esta herramienta crea automáticamente un documento XML que contiene todas las asociaciones y todas las referencias de lo añadido.

Una vez añadido todo lo que pensamos era indispensable, decidimos modularizar nuestro sistema al máximo. Es decir, decidimos crear un archivo XML por cada apartado añadido en el archivo XML que crea el Reload Editor. Por lo tanto creamos tantos archivos XML como Activity Structures e hicimos lo mismo con los Learning Activities y con los Environments. Esta división en estructuras tan pequeñas, nos iba a permitir tener una gran flexibilidad a la hora de programar, así como a la hora de ampliar el proyecto y buscarle nuevas funcionalidades a las especificadas inicialmente.

Los archivos XML referentes a una Activity Structure, contienen referencias a los tres Learning Activities que tiene asociados:

- Introducción Teoría
- Introducción Ejemplo
- Introducción Ejercicio

Podemos ver estos XML en la carpeta “xml” del proyecto. Dentro de esa carpeta están las distintas carpetas de cada uno de los temas. Elegimos por ejemplo la carpeta del primer tema: “Introducción al lenguaje C++” y abrimos el archivo XML correspondiente a la Unidad de Aprendizaje de este tema, que en nuestro caso corresponde a la Activity Structure (Fig. 4.4).

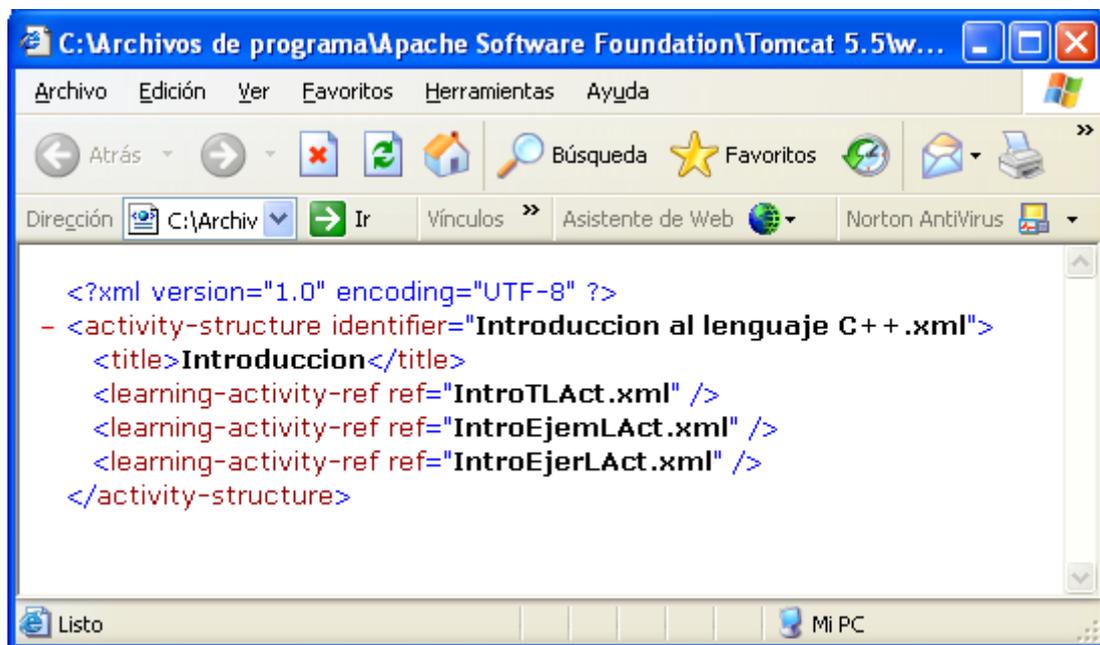


Fig. 4.4: Archivo xml de la Unit of Learning del primer tema.

Análogamente, los archivos XML para las distintas Learning Activities contienen las referencias a los Environments asociados.

En cambio, con respecto a los Environments hemos tenido que hacer ciertas modificaciones. El archivo XML que nos proporciona la herramienta Reload Editor al añadir todos los Learning Objects asociados al Environment contiene todas las referencias de dichos Learning Objects. Esto no es para nada útil en nuestro sistema, ya que de lo que se trata es de usar unos Learning Objects u otros según el estilo de aprendizaje de cada alumno. Por lo tanto, quitamos todas las referencias en los archivos XML referentes a los Environments y lo que añadimos fue, siguiendo el estándar de Learning Object Metadata (LOM), información acerca de los metadatos.

Como ya se explico en el apartado anterior (2.2 Funcionamiento interno de nuestro sistema), se crean dinámicamente tres archivos XML referentes a los tres archivos de Environments asociados a las Learning Activities que contienen todos los metadatos necesarios y a partir de aquí se pasa a elegir los Learning Objects que encajan con esos metadatos.

Por lo tanto, para terminar las diferencias de nuestro sistema con la especificación en el nivel A, sólo queda decir que la única diferencia entre dicha especificación y la implementación realizada por nosotros es que los Environments no contienen referencias a Learning Objects sino que contienen exclusivamente los metadatos que se necesitan para realizar posteriormente la elección de los Learning Objects que cumplan los requisitos impuestos por esos metadatos. Esto debe ser así, ya que queremos un sistema que cree el manifest dinámicamente y que elija los Learning Objects que va a mostrar a los alumnos según el estilo de aprendizaje en tiempo de ejecución.

Con respecto al nivel B, como hemos explicado en el apartado 1.3, añade propiedades y condiciones al nivel A. Esto permite crear un curso personalizado a las características de cada alumno.

La herramienta Reload Editor da la opción de añadir dichas propiedades (Fig. 4.5) y condiciones (Fig. 4.6 y 4.7) para cumplir con el estándar IMS Learning Design.

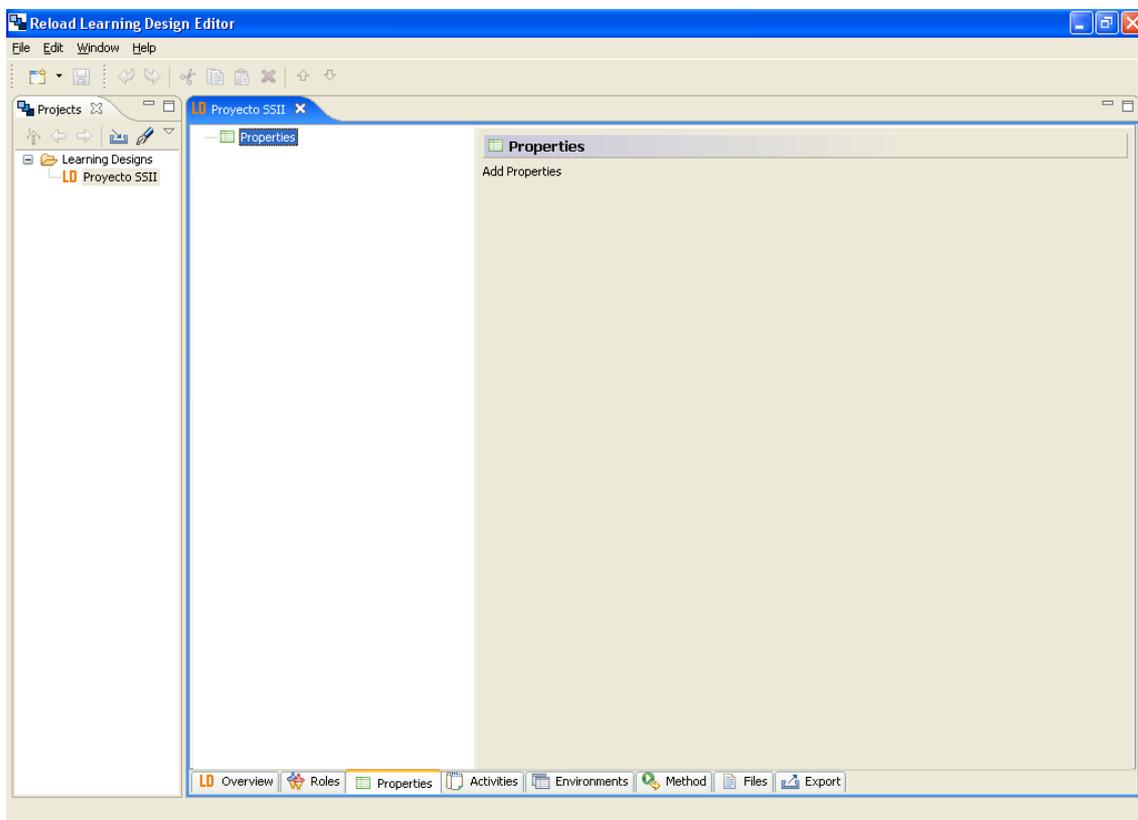


Fig. 4.5: Properties en Reload Editor.

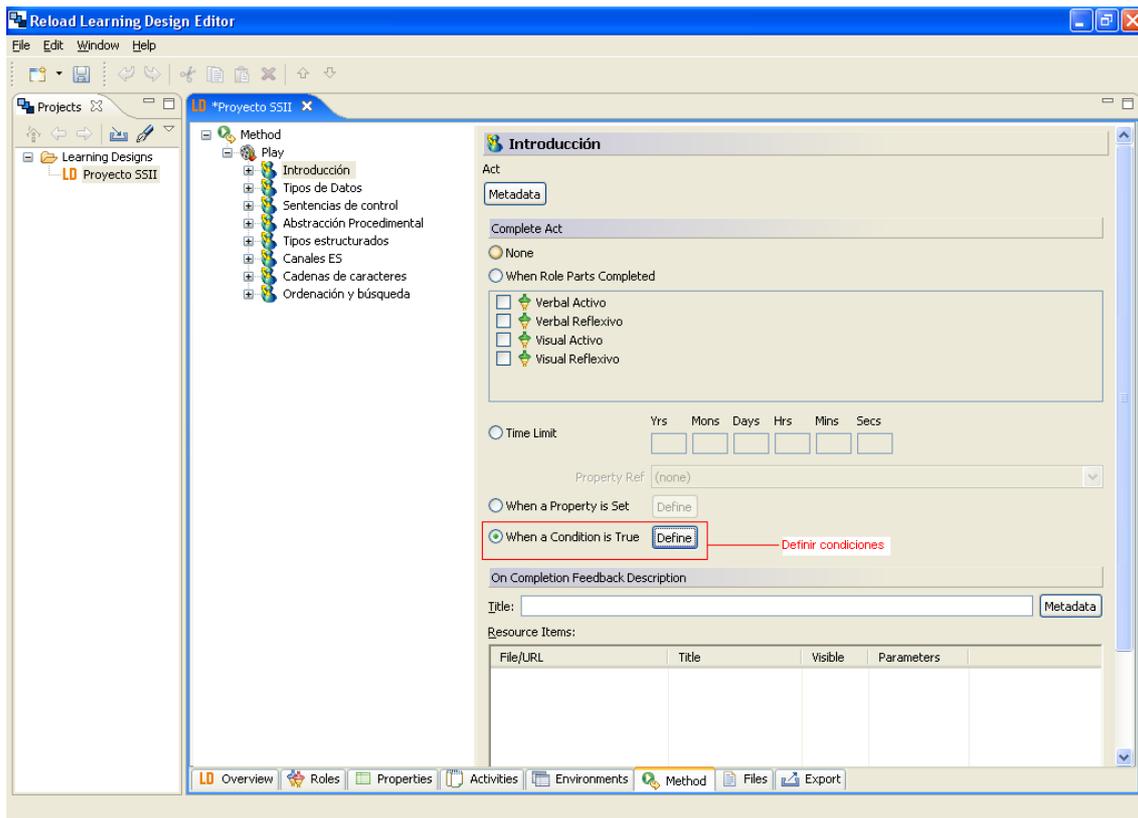


Fig. 4.6: Añadir condición a un Acto.

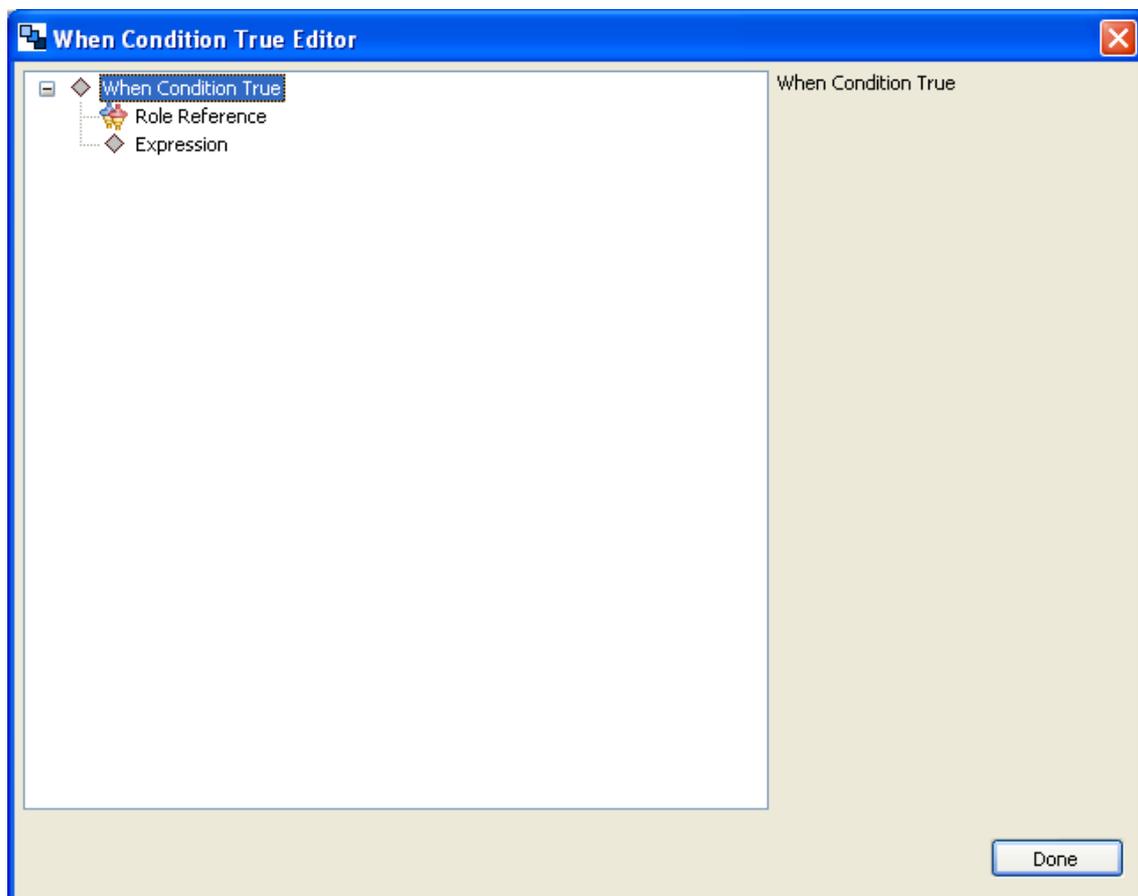


Fig. 4.7: Definir dicha condición para ese Acto.

Sin embargo, nosotros hemos optado por implementar estas propiedades y condiciones directamente en código Java, pero siguiendo el estándar especificado por IMS Learning Design.

Necesitábamos realizarlo de esta manera, ya que los archivos xml referentes a los Environments sólo tenían algunos metadatos rellenos. Teníamos por tanto que crear copias de estos archivos con todos los metadatos necesarios, para posteriormente comparar esos metadatos con aquellos de los Learning Objects. Esto sólo podíamos hacerlo a través de código Java, ya que intentarlo con Reload Editor habría sido una tarea más que complicada.

Este mecanismo es lo que permite crear esos archivos Environments dinámicamente en tiempo de ejecución y crear así gracias a esto el manifest dinámico que será distinto para cada alumno según su estilo de aprendizaje y el tema que esté estudiando.

6 Conclusiones

6.1 Conclusiones

Quizá la parte más importante del proyecto ha sido el trabajo de investigación y documentación. Gracias a este proyecto se nos ha dado la oportunidad de **conocer en profundidad varias tecnologías emergentes en el mercado del E-Learning**.

Con la ayuda de nuestra directora de proyecto Pilar Sancho Thomas hemos podido aprender tecnologías básicas como es **el uso del lenguaje XML** para el intercambio de información entre aplicaciones o **la utilización de Metadatos**. Además nos hemos visto inmersos en el enorme mundo del E-Learning y del Learning Design, un campo muy poco explorado todavía y que **nos ha brindado la oportunidad de ser innovadores y tratar de mejorar los métodos de aprendizaje empleados habitualmente**.

Además también **hemos aprendido a manejarnos con mucha soltura con la herramienta Reload Editor**, completamente básica para el Learning Design y que nos ha servido de guía en el proyecto. Antes de comenzar con el Learning Design dinámico tuvimos que familiarizarnos con los content packages y las utilidades que ofrecían.

Hemos estudiado en profundidad **el estándar IMS Learning Design** para tratar de sacarle todo el jugo posible de cara a la personalización del aprendizaje, y hemos tratado de descubrir hasta donde podíamos llegar sin salirnos de los estándares conocidos.

Además creemos que hemos abierto una puerta hacia un terreno enorme sin explorar. En principio **podríamos dividir nuestro trabajo de este año en cuatro fases**, cada una tan importante como la anterior:

- **Estudio de viabilidad del proyecto.** Posibilidades y alcance.
- **Labores de investigación continuas**, al no existir referencias anteriores.
- **Implementación de una aplicación que llevara a cabo todo lo propuesto a través de la investigación**, o al menos la mayor parte.
- **Estudio de las posibles mejoras y modificaciones del sistema** y cómo llevarlas a cabo.

El **trabajo de viabilidad** realizado en los inicios del proyecto nos demostró que el proyecto se podía llevar a cabo. **Las labores de investigación** durante todo el año nos han ido mostrando cómo podíamos llevarlo a cabo y hasta que punto podíamos llegar. La **posterior implementación de una aplicación** sencilla pero con la abstracción suficiente para soportar fuertes mejoras de cara a un futuro cercano lleva a cabo una prueba de que lo que proponemos es posible y fácilmente ampliable.

Además, tras ir finalizando nuestro tiempo de trabajo en el proyecto, hemos tratado de realizar una tarea que nos parecía fundamental. No queríamos dejar éste proyecto sin compartir lo que habíamos aprendido durante este año de trabajo y las ideas que habían ido surgiendo. Así que para ello **hemos elaborado un estudio de las ampliaciones que consideramos más importantes para el mejor funcionamiento del sistema** y así facilitar las labores de revisión y mejoras. Esto lo detallaremos en el siguiente apartado.

6.2 ¿Qué queda por hacer?

Como ya hemos comentado anteriormente, este proyecto pretende ser el principio de un trabajo mucho más amplio. Por ello **hemos tratado de abstraer el sistema lo máximo posible para que cualquier cambio o mejora, implique un trabajo mínimo para el diseñador o el programador que realice las modificaciones.**

Tras un año de trabajo sobre el tema tenemos bastantes ideas sobre como podríamos mejorar el sistema, o incluso incluir nuevas funcionalidades que no hemos podido desarrollar por falta de tiempo. Sin embargo, **a continuación detallaremos las que creemos más importantes y significativas para el sistema**, para que en un futuro próximo, las personas que continúen nuestro trabajo tengan en esta memoria un guía para poder llevarlas a cabo sin demasiados problemas.

Nuestro sistema no decide cual es el tipo de aprendizaje óptimo para un alumno, la aplicación facilita ciertas opciones al usuario para que éste decida cual es el estilo de aprendizaje que mejor se adapta a sus características. Esto tiene sus inconvenientes, ya que el usuario puede equivocarse en su elección, no tener conocimientos pedagógicos suficientes para poder discernir que tipo de aprendizaje es el más adecuado para él, etc. **Sería mejor que la propia aplicación sea capaz de decidir con certeza que tipo de aprendizaje es mejor para cada alumno.** ¿Cómo podemos resolver este problema? En principio no será necesario tener unos conocimientos de psicopedagogía elevados para poder resolverlo. Existe un test creado por Richard M. Felder y Bárbara A. Soloman llamado “*Index of Learning Styles Questionnaire*”¹ o “**Cuestionario índice sobre los estilos de aprendizaje**”² en español, gracias al cual, respondiendo a una serie de preguntas cortas determinará cual es nuestro estilo de aprendizaje más adecuado. Incluyendo este cuestionario en la aplicación o enlazando desde ella y recogiendo los resultados podremos clasificar exactamente al usuario en el estilo de aprendizaje óptimo para él sin necesidad de que este decida por nosotros.

¹: Más información y test completo en <http://www.engr.ncsu.edu/learningstyles/ilsweb.html>

²: Más información y test completo en http://213.96.34.21:8000/~tbaus/formulari_test_felder.htm

Siguiendo la misma línea de mejoras sería interesante aunque quizá, algo más complejo estudiar la posibilidad de no categorizar al alumno en un estilo de aprendizaje determinado sin más, sino recoger los datos del test anterior y realizar una ponderación de cada uno de los estilos. **Es decir, un alumno no será simplemente categorizado como visual y activo, sino que será, por ejemplo, un 70% visual y un 30% verbal y un 67% activo y un 33% reflexivo.** De esta forma podemos tener una personalización mucho mayor de las características del alumno y poder optimizar así los recursos que necesita el alumno para obtener el mayor rendimiento del curso que le proporciona el sistema. **Además de esta forma sí que podremos afirmar con rotundidad que cada alumno tiene exactamente un curso hecho a la medida de sus necesidades.**

Teniendo una mayor exactitud en la categorización del estilo de aprendizaje de un alumno podremos también organizar el curso, los Learning Objects, de una forma más adecuada para él. Así podremos escoger unos Learning Object de un tipo u otro en la medida en la que el alumno esté categorizado en ese tipo de aprendizaje. Por ejemplo si el alumno A es un 60% Activo y el alumno B un 95%, en nuestra aplicación los dos son almacenados como alumnos activos y recibirán el mismo número de ejercicios. Sin embargo, utilizando porcentajes, el alumno B debería recibir un mayor número de ejercicios y ejemplos que el alumno A, que recibirá más teoría.

Una nueva funcionalidad a incluir en la aplicación, es **la inclusión de métodos de evaluación al finalizar cada unidad de aprendizaje.** Nuestro sistema está completamente orientado a ello y no necesitaría ningún cambio para incluir esta mejora. La inclusión de test, exámenes o **algún tipo de feedback por parte del usuario nos permitiría conocer como está funcionando el curso para el alumno,** si está aprendiendo lo suficiente, le parece entretenido, o por el contrario le parece una pérdida de tiempo.

Utilizando los métodos de evaluación comentados en el párrafo anterior podríamos decidir si el alumno está cumpliendo los objetivos que se programan inicialmente en las unidades de aprendizaje. **Si el alumno no está satisfecho con el desarrollo del curso, o no está cumpliendo los objetivos marcados, siempre se puede modificar el estilo de aprendizaje asignado para tratar de motivarle.** Así si vemos que el alumno ha sido catalogado como Verbal y no está obteniendo los resultados esperados podremos modificar su perfil a Visual y comprobar si los resultados son mejores que los obtenidos anteriormente. **En éste sentido el sistema es completamente flexible y puede cambiarse el estilo de aprendizaje de un alumno en cualquier momento sin ningún tipo de problema.**

La aplicación dispone de momento de tres tipos de Learning Objects distintos, clasificados de la siguiente manera teoría, ejemplos y ejercicios. Esto es fácilmente modificable y ampliable, simplemente será necesario marcar los nuevos Learning Objects con los tags adecuados en los metadatos e incluir nuevas categorías de objetos en el código, lo cual es bastante sencillo debido a la fuerte abstracción del código en este sentido. Así **podríamos incluir nuevos Learning Objects más complejos, como pueden ser videos, audios, búsquedas, etc...**

Otras **funcionalidades** a incluir que nos parecen interesantes serían las **que ayudan a la comunicación entre el profesorado y los alumnos o la interacción entre los propios usuarios**. Así realizar **foros entre los estudiantes del mismo curso, o la inclusión de chats en la aplicación, ayudarían al aprendizaje de los alumnos**, dándoles nuevas vías de aprendizaje, como pueden ser la realización de ejercicios en grupo o la posibilidad de comunicarse en tiempo real con su tutor para la resolución de dudas.

Otra funcionalidad útil sería un buscador de términos para el alumno. Si un alumno está estudiando y no recuerda algún término o concepto puede buscarlo en el curso a través de un buscador interno de la aplicación. En principio no creemos que la inclusión de un buscador sea muy complicada, ya que compañías como Google ya han puesto a disposición pública buscadores similares al que estamos proponiendo para uso público, con lo cual se podría estudiar el coste de incluirlo en el sistema³.

Por último, pero no por ello menos importante, podría realizarse una **mejora del diseño de la interfaz gráfica**, es decir de las páginas web, para que el alumno se sienta cómodo mientras realiza el curso y **la interfaz sea lo suficientemente clara y descriptiva para cualquier usuario**. La mejor forma de comprobar si la interfaz es adecuada es **mediante la realización de pruebas con un número elevado de personas, estudiando las quejas y sugerencias de los testers** para así tener una fiabilidad máxima del éxito de la interfaz.

³: Más información en <http://www.tuarroba.com/index.php?id=tuagoogle>

7 Glosario

Estándar e-learning

- **Definición:** Agrupación de especificaciones o recomendaciones sobre el desarrollo de aplicaciones E-Learning.
- **Referencias:**
 - Página 6 – Distintos estándares.

Estilos de aprendizaje

- **Definición:** Diferentes métodos de enseñanza, dependiendo de las características del alumno.
- **Referencias:**
 - Página 26 – Explicación de los diferentes estilos de aprendizaje según el Dr. Folder.
 - Página 51 – Diferentes opciones sobre los estilos de aprendizaje a la hora de registrarse en el sistema.
 - Página 58 – Cómo influyen los distintos estilos de aprendizaje en la creación del manifest y por tanto en la visualización del curso.

IMS Learning Design

- **Definición:** Estándar de E-Learning enfocado completamente a desarrollar especificaciones en formato XML.
- **Referencias:**
 - Página 8 – Motivos que nos han llevado a elegir esta especificación.
 - Página 9 – Estudio en profundidad del estándar IMS.

Manifest

- **Definición:** Archivo XML que contiene la estructura principal de un curso de E-Learning.
- **Referencias:**
 - Página 21 – Uso del manifest en el Reload Editor.
 - Página 56 – Diagrama de secuencia de la construcción dinámica del manifest para nuestro sistema.
 - Página 58 – Explicación detallada de la creación dinámica del archivo manifest.xml.
 - Página 59 – Ejemplo del archivo manifest.xml.

Plataforma LMS

- **Definición:** Learning Management System. Plataforma que tiene la función de mostrar los cursos E-Learning descritos en el archivo manifest, así como otras funcionalidades para los alumnos (chats, foros...).
- **Referencias:**
 - Página 51 – Descripción de nuestra aplicación, que simula una pequeña plataforma LMS.

Reload Editor

- **Definición:** Empaquetador de contenido y un editor de Metadatos, siguiendo el estándar IMS Learning Design.
- **Referencias:**
 - Página 21 – Descripción de la herramienta Reload Editor.
 - Página 70 – Diferencias y similitudes del curso generado empleando solamente la herramienta Reload Editor o nuestra aplicación.

UML

- **Definición:** Lenguaje Unificado de Modelado. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.
- **Referencias:**
 - Página 32 – Descripción de las clases java más importantes del sistema mediante diagramas UML.
 - Página 44 – Diagramas de secuencia de algunas de las funcionalidades más importantes de la aplicación.

XML

- **Definición:** eXtensible Markup Language. Se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas.
- **Referencias:**
 - Página 11 – Uso de los archivos XML en el estándar IMS Learning Design.
 - Página 20 – Definición y uso del XML e introducción de como usamos este lenguaje en nuestra aplicación.
 - Página 57 – Utilización de los archivos XML en el desarrollo del sistema.

8 Bibliografía

Pasado y futuro del e-learning

<http://www.ofertaformativa.com/pasado-y-futuro-del-e-learning.htm>

Estándares educativos

<http://www.learnativity.com/standards.html>

<http://www.statusquo.cl/standards/ims/ld/faqs.shtml>

http://ecampus.uniacc.cl/textos/tecnologia/eduardo_hernandez/eduardo.htm

Respecto a IMS Learning Design

<http://www.imsglobal.org>

<http://www.statusquo.cl/standards/ims/ld/faqs.shtml>

Plataformas e-learning

<http://demo.e-ducativa.com/>

<http://aulas.ciberaula.com/demo/login/index.php>

XML

http://www.programacion.com/tutorial/javaxml/1/#1_5

<http://www.programacion.com/java/foros/26/msg/194499/>

<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=domxml>

<http://www.osmosislatina.com/xml/domsax.htm>

<http://www.jdom.org/downloads/index.html>

Taxon path

<http://www.mindspring.com/~wason/TDW/drtomclassificationcas.html>

Herramientas similares

<http://www.atutor.ca/>

<http://teleduc.nied.unicamp.br/~teleduc/>

<http://www.uv.es/ticape/docs/nabuen/PFC.pdf>

<http://www.linti.unlp.edu.ar/publicaciones/recientes/diaz-osorio-amadeo-unlp.pdf>

<http://www.linux-magazine.es/issue/15/educacion.pdf>

<http://www.dotlrn.org/>

http://reddigital.cnice.mec.es/6/Articulos/articulo_resumen.php?articulo=2

<http://www.blackboard.com/extend/b2/>

<http://moodle.org/>

<http://www.webct.com/>

Otros

<http://www.wikipedia.org/>

<http://www.programacion.net/html/foros/8/1/>