

Ingeniería Informática



UNIVERSITAT DE VALÈNCIA

VNIVERSITAT & DE VALÈNCIA

PROYECTO FIN DE CARRERA

JULIO 2004

**IMPLANTACIÓN DE dotLRN EN LA
UNIVERSIDAD DE VALENCIA.
PROYECTO: AULA VIRTUAL**

Autor: Pedro J. García Pozo

Director: Wladimiro Díaz

1. INTRODUCCIÓN.....	6
1.1. COMUNIDADES ON-LINE, E-LEARNING.....	6
1.2 PROMOTORES DEL PROYECTO. LA ENTREVISTA	7
1.3 OBJETIVOS.....	8
1.4 SITUACIÓN ACTUAL	13
1.5 SELECCIÓN DE ALTERNATIVAS	21
1.6 VIABILIDAD	21
2. ESTADO DEL ARTE	22
2.1 INTRODUCCIÓN	22
2.2 DEFINICIÓN DE E-LEARNING	25
2.2.1 Características de e-learning	26
2.2.2 Componentes de un sistema e-learning (LMS).....	28
2.2.3. Courseware o Contenidos.....	31
2.2.4 Comunicación sincrónica y asíncrona	32
2.2.4 Metadatos.....	33
2.3 ESTANDARES.....	36
2.3.1 Que se debe estandarizar	37
2.3.2 Organizaciones para la estandarización.....	41
2.4 VENTAJAS Y DESVENTAJAS DE ELEARNING.....	46
2.5 E-LEARNIG EN LAS UNIVERSIDADES ESPAÑOLAS.....	50
3. ANÁLISIS DE REQUISITOS	52
3.1 PROBLEMAS ACTUALES E IDENTIFICACIÓN DE REQUISITOS.	53
4. PLANIFICACIÓN Y COSTES	56
4 .1 Definir los propósitos	56
4.2 División del trabajo	57
4.3 Estimación de tiempos	58
4.4 Identificación de hitos	59
4.5 Encadenamiento de actividades.....	59
4.6 Planificación temporal	60
4.7 Coste de la implantación de dotLRN	62

5. DESCRIPCIÓN DEL SISTEMA	64
5.1 CÓMO SURGE OPENACS	64
5.2 QUE ES UNA APLICACIÓN FRAMEWORKS	65
5.3 LA ARQUITECTURA DE OPENACS.....	66
5.3.2 Proceso de las peticiones	69
5.3.3 El sistema de permisos	70
5.3.4 El sistema de plantillas	78
5.4 DOTLRN	81
5.4.1 Diseño y arquitectura del Sistema.....	84
5.4.2. Internacionalización en OpenACS y dotLRN	91
5.4.3. DotLRN y SCORM.....	92
6. CASO PRÁCTICO:	94
“DOTLRN EN LA UNIVERSIDAD DE VALENCIA”	94
6.1 ANÁLISIS DE REQUISITOS	100
6.1.1 Requisitos por usuarios.....	100
6.1.2 Requerimientos hardware y software.....	101
6.2 INSTALACIÓN DE DOTLRN	108
6.3 IMPLEMENTACIÓN	110
6.3.1 Carga de usuarios	111
6.3.2 Carga de asignaturas, centros y períodos.....	119
6.3.3 Matricular los alumnos y generar los cursos a los que pertenece.....	123
6.3.4 Portlet’s para visualizar información de las asignaturas..	124
6.4 PERSONALIZACIÓN DE DOTLRN	132
6.4.1 Características generales	132
6.4.2 El administrador de dotLRN.....	136
6.4.2.1 Cursos en dotLRN.....	139
6.4.2.2 Comunidades en dotLRN	142
6.4.3 El profesor	144
6.4.4 El alumno	146

7. CONCLUSIONES Y TRABAJOS FUTUROS	148
7.1 CONCLUSIONES.....	148
7.2 TRABAJOS FUTUROS.....	149
8. BIBLIOGRAFÍA.....	152
8.1 ENLACES	152
8.2 TESIS Y TRABAJOS	153
8.3 LIBROS	154
ANEXO I - ESTÁNDARES E-LEARNING	156
INICIATIVAS DE ESTÁNDARES SOBRE E-LEARNING.....	159
AICC, Aviation Industry CBT Comitee.....	159
IEEE Learning Technologies Standards Committee (LTSC).....	160
IMS Global Learning Consortium, Inc.	162
ADL SCORM.....	164
SCORM Content Aggregation Model	166
SCORM Run-Time Environment.....	167

CAPITULO 1

1. Introducción

1.1. Comunidades on-line, e-learning.

A medida que la sociedad va cambiando también aparecen nuevas inquietudes en la gente. Las sociedades con más recursos tienen más tiempo para formarse y además dedican parte de su tiempo a esa tarea. Además si observamos la prensa y la dedicación en la Internet a la formación nos damos cuenta que el número de URL's que se dedican a esto son casi infinitas.

Existen varias formas de adquirir conocimientos, desde el acto cotidiano de relacionarse con las personas, hasta la manera más tradicional de asistir a una clase en cualquier centro educativo. Si eres nuevo en una empresa siempre encontrarás compañeros que se vuelcan contigo y te ayudan a conocer su funcionamiento, a conocer sus herramientas, etc. Todo esto entre muchas otras cosas.

Podemos definir una comunidad como un grupo de gente con diversos grados de experiencia o pericia en el cual los expertos tratan de ayudar a los principiantes y estos mejoran sus habilidades.

Una comunidad en línea es una aplicación Web que permite a la gente trabajar juntos, educar a distancia, o vender productos o servicios entre si. Cualquier actividad que requiera la colaboración entre un grupo de gente es candidata a desarrollarse bajo una comunidad en línea.

¿Por qué son las comunidades en línea tan importantes? Hay gente que no se sienta delante del ordenador para que este le ayude, sino porque es la manera más fácil y más rápida de tener acceso a Internet y esto representa entrar en contacto con otras personas que te pueden ayudar. Lo que les atrae de Internet son la gran cantidad de datos, los grandes depósitos de conocimiento, los servicios de comercio electrónico y, sobre todo, el ponerse en contacto con otras personas que comparten sus conocimientos.

Todo esto requiere la colaboración entre la gente. Una gran base de conocimiento no se puede crear por una sola persona. Un servicio de

comercio electrónico necesita un equipo de negocios, otro de comercio y ventas y otro de logística. El estar conectado con otra persona que esté dispuesta a ayudarte con un problema puede provocar un importante cruce de correo o los servicios de un foro.

¿Por qué cualquier persona desearía crear una comunidad en línea? Para los individuos, crear una comunidad o una subcomunidad proporciona la satisfacción de la enseñanza sin el dolor de cabeza de la enseñanza a tiempo completo o de preparar toda la burocracia de la enseñanza profesional. Para las organizaciones no lucrativas, la comunidad en línea presenta una oportunidad de unir los esfuerzos de mucha gente hacia una meta social. Para el negocio, la cuota de mercado en línea de las estructuras de la comunidad, permite un mejor servicio al cliente, y promueve lealtad.

Como el principal objetivo que tienen las Universidades es la educación, y la socialización de los individuos, puede aprovechar el avance notorio de las evoluciones tecnológicas de la información en cuanto a calidad de las redes, multimedia, productos informáticos etc. y cambiar la enseñanza tradicional hacia el e-learning.

1.2 Promotores del Proyecto. La entrevista

El proyecto es promovido desde el Vicerrectorado de Nuevas Tecnologías y Ciencias de la Información en colaboración con el Centro de Cálculo de esta Universidad.

Tras realizar una reunión con el Director del Centro de Cálculo y el vicerrector de Nuevas Tecnologías y las personas implicadas se plantea la necesidad que tiene la Universidad de Valencia en implantar un herramienta potente para llevar a cabo la realización del proyecto de "Aula Virtual".

Este proyecto engloba la necesidad de desarrollar:

1. Cursos completos, para dar apoyo a la docencia on-line,
2. Microweb's espacio público y privado para todas las asignaturas que actualmente dispone la Universidad de Valencia.

3. Soporte de comunidades para la gestión de proyectos, aunque esta parte en principio no se pondrá en marcha salvo solicitud expresa.

La idea de orientar mi proyecto hacia el e-learning surge después de conocer a través de Darío Roig que tras evaluar una serie de plataformas sobre e-learning, tanto de libre distribución como de pago y su vinculación en la Universidad de Valencia como programador y experto en teleformación, La Universidad de Valencia se plantea instalar y adaptar una de estas plataformas en su organización. Se pretende que en el segundo cuatrimestre del curso 2003-2004 se utilice la plataforma dotLRN en la Universidad bajo demanda, es decir, se ofrecerá como un nuevo servicio para que los profesores interesados puedan impartir algunas asignaturas y de esta forma embarcarse en este mundo de la teleformación, de esa manera se podrán sacar conclusiones y valorar la experiencia al terminar este cuatrimestre.

En cuanto a los promotores del proyecto debo citar los siguientes:

- * D. Salvador Roca. Subdirector del Centro Calculo de la Universidad de Valencia
- * D. Agustín López. Programador del Centro Calculo de la Universidad de Valencia.
- * D. Darío Roig García. Técnico Medio de la Universidad de Valencia.
- * D. José A. Vázquez Alvadalejo. Director del Servicio de Informática de la Universidad de Valencia
- * D. Vicente Cerveron. Vicerrector de tecnologías y ciencias de la información de la Universidad de Valencia.

Todo el trabajo ha sido dirigido y supervisado por el director de este proyecto:

D. Wladimiro Diaz Villanueva (Profesor del Departamento de Informática de la U.V.)

1.3 Objetivos

Aunque actualmente la Universidad de Valencia ya dispone de una plataforma comercial conocida y muy potente, WebCT, se plantea la

necesidad de implantar una plataforma de código abierto (GPL), para realizar la teleformación.

La elección de una plataforma de código abierto es porque se dispone de presupuesto cero para el desarrollo de software, no se quiere invertir dinero en esta parte, se prefiere invertir en recursos humanos y material para aprovechar el momento del GPL que parece ser la tendencia actual de las grandes empresas aparte del coste de licencias que conlleva una plataforma comercial. Otra causa de optar por el software libre es porque se está intentando huir de plataformas rígidas de pago que no ofrecen modularidad clara y no son muy genéricas. Las características que se evaluaron y se tomaron en cuenta para decidir fueron:

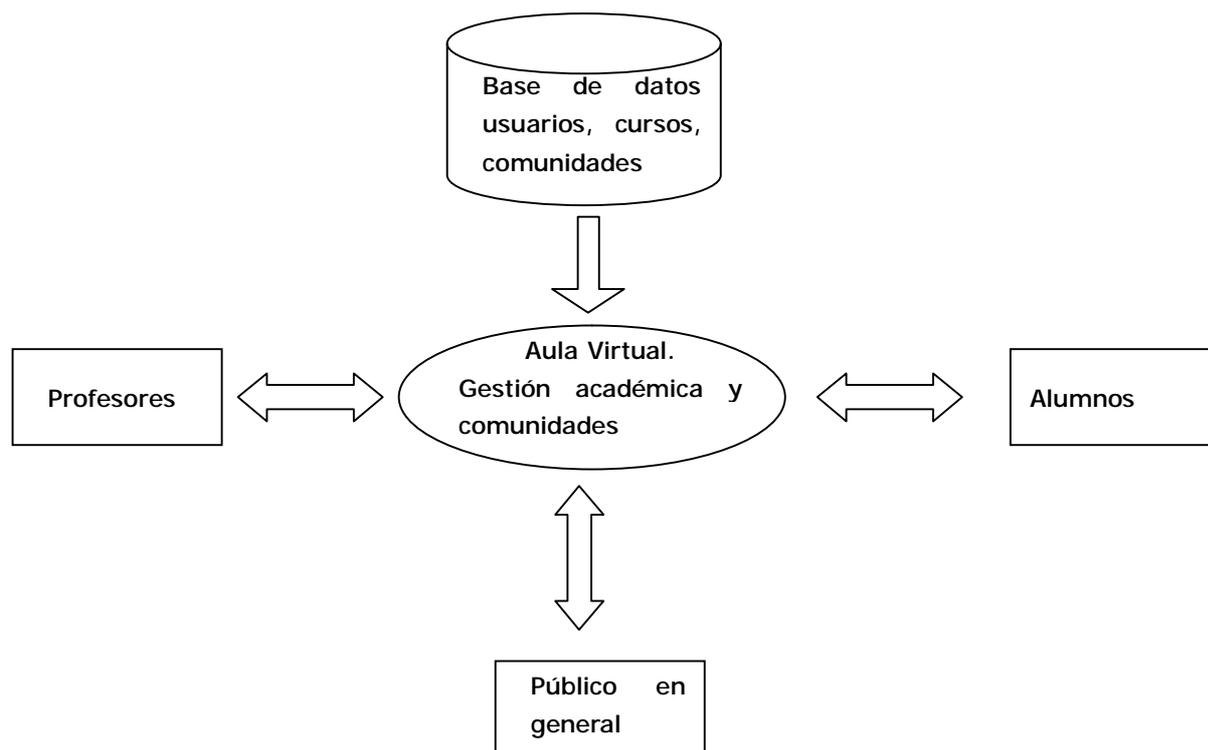
- Usabilidad
- Mecanismos de comunicación entre usuarios dentro de la plataforma
- Seguridad del sistema
- Manejo de Cursos
- Administración del sistema
- Accesibilidad
- Escalabilidad

Estas características fueron expuestas y evaluadas en el estudio de Dario Roig¹ sobre plataformas de e-learning, presentado como proyecto final de carrera, en él se concluyó tras un minucioso trabajo comparativo y de evaluación de plataformas que la plataforma GPL que más se adaptaba a estas necesidades es dotLRN y en cuanto a las de pago se seleccionó a WebCT, más adelante veremos una pequeña comparación entre las características de ambas. Por este motivo no se va a profundizar en este tema sino que partimos ya de la base que la plataforma es dotLRN.

Otro peso muy fuerte a la hora de elegir la plataforma lo ha tenido el que existan otras empresas e Universidades de gran prestigio que ya están utilizandola, podemos mencionar a la UNED, el MIT, Greenpeace, El Banco Mundial, etc... Para reforzar la idea y tomar contacto con gente que ya hubiese tenido contacto con esta plataforma también se hizo un viaje a Madrid para reunirse con los responsables del proyecto en la UNED y pedirles consejo a cerca de su experiencia, tras la reunión aun salieron más reforzadas nuestras teorías de que debía ser dotLRN la plataforma a elegir.

¹ Para ver sus informes sobre la elección de plataformas visitar [Http://www.uv.es/ticape](http://www.uv.es/ticape)

Para poder planificar los objetivos reales y las funciones a realizar, debemos saber cual va a ser la situación de la plataforma a instalar y el entorno que constantemente va a estar interactuando con ella.



Una vez vista cual será la función global que va a desempeñar nuestra plataforma, deberemos indicar que funciones mínimas ofrecerá y para que se asemeje lo máximo posible a las necesidades reales que tiene la Universidad.

Los objetivos y funciones que busca este proyecto son los siguientes:

- Conseguir una plataforma para la gestión de cursos on-line y sirva de apoyo a la docencia via Internet.
- Que gestione foros
- Que gestione noticias
- Que gestione y comparta ficheros
- Que tenga calendario
- Que gestione comunidades
- Que se integre con las bases de datos académicas de la Universidad.

Todas estas funciones con unos niveles de seguridad fiables para que en todo momento las personas que interactúan con la plataforma se hayan tenido que validar para acceder a estos recursos.

Estas funciones serán las que tienen que ver con las características mínimas que debe cumplir cualquier plataforma para la realización de teleformación.

dotLRN se está utilizando actualmente en el MIT (U.S), Universidad de Heidelberg (Alemania), Universidad de Cambridge (U.K), Universidad de Bergen (Noruega), Universidad de Sydney (Australia), Universidad Galileo (Guatemala) y Universidad de Copenhagen (Dinamarca), característica muy valorada también a la hora de seleccionar esta plataforma, también se esta utilizando en organizaciones que no son únicamente Universidades, como son Greenpeace, etc..

Tabla resumen de los principales Casos de Uso de dotLRN:

Universidad	Estudiantes	Usuarios	Plataforma
Sloan - MIT, USA	11.000	11.000	Sun /Oracle
Heidelberg University, Alemania	30.000	3.000	Sun / Oracle
Vienna University of Economics, Austria (Wirtschaftsuniversität Wien)	25.000	16.000	Linux / PostgreSQL
Universidad Nacional de Educación a Distancia, España	180.000	4.500	Linux / Oracle
University of Cambridge, Reino Unido	16.500	3.500	N/A
Universidad Galileo, Guatemala	3.000	3.000	Linux / Oracle
Berzeit University, Palestina	5.000	5.000	Linux / Oracle
University of Sydney - WEG, Australia	35.000	3.000	Linux / PostgreSQL
Universidad Nacional de Ingeniería, Nicaragua	10.000	N/A	Linux / PostgreSQL
University of Copenhagen, Dinamarca	35.000	N/A	N/A
The University of Bergen, Noruega	17.000	N/A	N/A

Los módulos que nos garantizan poder realizar learning y los que instalaremos al principio son:

File Storage: A través de un interfaz simple, los usuarios pueden transferir y pueden compartir una variedad de archivos, documentos, imágenes, PDF, y URLs.

Los foros: son un tablón de anuncios o discusión el sistema de grupo para la comunicación asincrónica y colaboración.

Calendario: proporciona dos niveles de funcionalidad: 1) administradores pueden anunciar las entradas del calendario para una clase o comunidad; 2) los usuarios pueden ver todos sus eventos en su calendario personal en una vista única. Pueden importarse los datos desde Outlook, y puede sincronizarse con el calendario de su PDA.

Noticias: Sirven para mostrar la comunicación en un solo sentido, el administrador del grupo o los propios usuarios del grupo.

Homework: Este portlet nos permite gestionar las tareas para que cada alumno pueda enviar sus tareas y a los instructores poder evaluarlas.

Survey: Herramienta para realizar exámenes, los profesores pueden crear exámenes on-line en formato html, o en texto.

Bulk mail: Permite a los administradores del grupo enviar mensajes de correo individuales o colectivos a través de su cliente de correo.

Weblogs: dotLRN se distribuye con un paquete integrado que permite a los estudiantes tener weblogs personales así como weblogs para las clases, los subgrupos, y las comunidades. Éstos se pueden utilizar para tomar notas individuales o de grupos.

1.4 Situación actual

No es que actualmente la Universidad no disponga de ningún sistema para la teleformación, en este sentido es todo lo contrario ya que dispone de una de las herramientas de pago más interesantes en este sentido y que ofrece más características, nos referimos a WebCT, pero como he comentado el ser de pago y la tendencia actual hacia apostar por el GPL y no vincularse a una empresa para realizar estas funciones llevan a la Universidad de Valencia a plantearse el porque no apostar por una de estas herramientas.

La plataforma WebCT esta ubicada en la Unidad de Apoyo en Blasco Ibáñez y actualmente esta dando docencia pero a muy bajo nivel, el principal obstáculo para lanzarla a toda la Comunidad Universitaria es el elevado coste de las licencias.

Veamos algunas características de esta plataforma, aunque si alguien quiere conocer mucho más a cerca de WebCT pueden consultar el trabajo realizado por Darío Roig en su proyecto <http://www.uv.es/ticape>.

WebCT. Es un sistema telemático desarrollado por la Universidad British Columbia. Es uno de los sistemas telemáticos de formación en Internet más usado a nivel mundial, y es usado actualmente en numerosas universidades de todo el mundo y no sólo eso, es una de las mejores plataformas recogidas (ya sean estas GPL o de pago) tanto a nivel de herramientas como a nivel de estándares, es la referencia a partir de la cuál muchos desarrolladores han iniciado sus proyectos: unos para corregir sus defectos y otros para acercarse a su modelo.

Lo primero que se debe destacar de esta plataforma es que cumple con los estándares: IMS Learning Resources Meta-data Specification, IMS Enterprise Specification for Student Data, IMS Content & Packaging Specification y IMS Question & Test Specifications.

El lado servidor requiere: una base de datos Oracle, sistema operativo Red Hat ó Windows 2000 y está programada con tecnología J2EE. Hace uso de LDAP para la autenticación y WEBDAV. Sirve para todo tipo de aprendizaje ya que permite acceso offline a los contenidos que están almacenados en CD-ROM, dispone de herramientas síncronas como son la pizarra compartida, chat en tiempo real, etc. Dispone de calendario de progreso, plantillas de cursos, mail interno, comunidad de estudiantes, grupos de trabajo, etc.

- Principales servicios, podemos destacar los siguientes:

Servicio de acceso My WebCT: El control de seguridad y acceso que se establece en WebCT para los diferentes cursos se realiza a través de este servicio, mediante la introducción de un nombre de usuario y una contraseña. Mi WebCT es un servicio central desde donde los estudiantes pueden entrar en todos los cursos desde una única página, sin necesidad de repetir los datos de usuario (ID) ni la contraseña (password).

Uso sencillo e intuitivo: la navegación a través de los diferentes entornos de este sistema se realiza a través de una serie de menús y herramientas jerárquicas en forma de árbol, por lo que no requiere prácticamente para su utilización ningún conocimiento técnico. El profesor o tutor puede personalizar cómo el alumno accederá a los contenidos o herramientas proporcionados en el curso para un acceso directo a determinados elementos o servicios. Por otra parte, el contenido se puede estructurar en Módulos, éstos en Unidades Didácticas, y cada una de éstas podría tener una presentación, objetivos y enlaces a los contenidos. Barra de navegación.

Integración de contenidos en HTML: la forma de preparar los contenidos en este sistema se efectúa a través de su edición en formato HTML, pudiéndose utilizar cualquier herramienta comercial de edición de HTML, o la ha incorporado en el mismo sistema.

Herramientas para la gestión de los cursos: a través del servicio My WebCT, los profesores, tutores o administradores podrán acceder a la lista de cursos para su administración (organizar los cursos, actualizarlos, publicar sus contenidos, definir la forma de presentación de éstos, administrar los ficheros del curso...), gestionar el alta y baja de alumnos,

personalización de herramientas y servicios disponibles para los distintos usuarios, así como las opciones de visualización de los mismos, ayuda disponible en pantalla, etc.

Servicios de comunicación: Este sistema, al igual que otros sistemas telemáticos estudiados, incorpora una serie de herramientas de comunicación síncronas y asíncronas. Son los foros electrónicos (de tipo moderado y libres), correo electrónico, tablón de noticias (en el que se pueden publicar eventos públicos o privados), chat y pizarra electrónica. Gestión de foros.

Sistema de ayuda incorporado: en este sistema se ha integrado ayuda sensible al contexto de muchas de sus características y herramientas, denominada Minihelp, que el tutor puede activar o desactivar. Además, el tutor puede añadir una breve descripción de cada componente que el alumno utilizará a lo largo del curso.

Creación automatizada de un índice de contenidos y glosario de términos: este sistema utiliza un editor de índice de contenidos para construir un índice basado en el contenido de las páginas del curso. Esta herramienta permite diseñar las principales entradas y subentradas del índice. Asimismo, permite crear un glosario de términos para el curso, pudiendo contener el mismo texto e imágenes.

Gran variedad de herramientas de evaluación: Las herramientas de evaluación que dispone WebCT son de tres tipos: grupos de estudio, para la realización de proyectos y/o trabajos en común por un determinado grupo de estudiantes; trabajos, que son prácticas asignadas a los alumnos por el tutor; autoevaluación, que son pruebas tipo test, y por último, pruebas (exámenes tipo test para ser realizados online). Edición de un test de autoevaluación. Seguimiento de los alumnos: A través de las herramientas Resultados, Mi evolución y Mi página, los profesores o tutores podrán realizar el seguimiento de la evolución del alumno en el curso.

Inserción de todo tipo de elementos multimedia: la de integración de material multimedia en WebCT es posible, debido a que la edición de contenido en WebCT se realiza a través de la edición y carga de páginas HTML. Esto permite utilizar un editor HTML para realizar los cambios necesarios en las páginas de contenido, integrando por tanto las típicas características multimedia que se integran en las páginas en formato HTML

(imágenes, hiperenlaces a través de hipertexto, enlaces a páginas externas y URLs, así como inserción de audio, vídeo e integración con Shockwave de Macromedia).

En la siguiente tabla se han comparado las dos plataformas que esta gastando la Universidad. Comparación facilitada por edutools. [12]

	dotLRN	WebCT 3.8 edición campus
Herramientas de comunicación evaluadas.		
Foros de discusión	El software incluye ayuda para foros de discusión. Los hilos del foro se pueden ver por la fecha o por el autor. Los instructores pueden crear ambientes separados de discusión para grupos pequeños. Los grupos pueden estar abiertos a todos o solamente a un sistema selecto estudiantes. Las respuestas pueden incluir, una imagen o URL. Los envíos pueden ser texto plano, texto con formato HTML. Los foros pueden estar moderados.	Los foros de discusión se pueden ver por la fecha, por el hilo de discusión, y por título. Los instructores pueden determinar el nivel de la implicación (leída, escribir, o fijar anonymously) para los estudiantes. Los instructores pueden crear ambientes separados de discusión para grupos pequeños.
Intercambiar ficheros	Los estudiantes pueden compartir el contenido de sus carpetas personales con otros estudiantes y sus instructores.	Los estudiantes pueden subir archivos a una carpeta compartida del grupo.
E-mail interno	Los estudiantes deben tener una dirección de correo externa en Internet. Los estudiantes pueden enviar mails individuales o a todo el grupo. Todos los email enviados del sistema están archivados.	Los estudiantes pueden enviar mails individuales o a todo el grupo. Los estudiantes pueden unir y los ficheros de archivo y pueden remitir mensajes a las cuentas externas del email.
Online Journal/notes	El sistema monta una herramienta del weblog, que permite que los estudiantes tengan weblogs personales así como los weblogs para las clases, los subgrupos, y las comunidades. Esto blogging funcionalmente incluye la ayuda completa de RSS, la ayuda limitada del trackback, la ayuda limitada de Blogger API, y la entrada de texto ajustada a formato.	Los estudiantes pueden unir notas a cualquier página. Los estudiantes pueden combinar sus notas con el contenido del curso para crear una guía imprimible del estudio.
Chat en tiempo real		Es una herramienta Java- para chat, puede tener espacios privados de charla

		y mensajes privados. El sistema crea los registros archivo para todos los chats. La herramienta de chat apoya hasta cuatro discusiones simultáneas por grupo.
Servicio de video		
Whiteboard		Soporta un Whiteboard
Herramientas de productividad.		
Bookmarks	Los estudiantes pueden no perder de vista todas sus asignaciones, plazos, y fechas debidas en sus calendarios en línea personales. Los instructores pueden fijar acontecimientos y avisos en el calendario en línea del curso. Las entradas en el calendario en línea del curso se pueden fijar para la clase entera, un equipo específico, o un usuario específico. Todos los estudiantes tienen un Home Page personal, a todos los cursos, y a grupos en quienes alistan al estudiante y todo el curso y system-wide acontecimientos así como éstos de su calendario personal.	Los instructores y los estudiantes pueden fijar acontecimientos y avisos en el calendario en línea del curso. Los estudiantes pueden ver sus lecturas y actividades terminadas y pendientes del curso.
Calendario/progresión	Los estudiantes pueden tener a la vista todas sus anotaciones, plazos, y fechas debidas en sus calendarios en línea personales. Los instructores pueden fijar acontecimientos y avisos en el calendario en línea del curso. Las entradas en el calendario en línea del curso se pueden fijar para la clase entera, un equipo específico, o un usuario específico.	Los instructores y los estudiantes pueden fijar acontecimientos y avisos en el calendario del curso. Los estudiantes pueden visión sus lecturas y actividades terminadas y pendientes del curso. Los estudiantes pueden visión sus grados.
Ayuda	Los estudiantes tienen ayuda sensible del contexto. El sistema incluye un módulo del FAQ para generar los FAQ nuevos.	El sistema incluye una guía del estudiante. Los estudiantes tienen ayuda sensible del contexto en los foros de discusión y el email interno.
Buscador de cursos	Los estudiantes pueden buscar todos los hilos de discusión dentro de sus carpetas.	Los estudiantes pueden buscar todas las notas del curso, hilos de discusión de discusión, y líneas sujetas del email en su curso.
Trabajo offline/shyncronize	Los estudiantes pueden descargar el contenido para un curso entero en un formato que se pueda almacenar localmente, y sincronizan acontecimientos del calendario con los calendarios de la perspectiva de Microsoft.	Los estudiantes pueden compilar y descargar el contenido para un curso entero en un formato que se pueda imprimir o almacenar localmente. Sobre volver a entrar un curso, los estudiantes tienen la opción de reasumir en la página pasada vista.

Herramientas para implicar a los estudiantes.		
Groupwork	Los instructores pueden asignar estudiantes a los grupos. Cada grupo puede tener su propio foro de discusión, agrupa la lista de emails, exámenes, citas de calendario, intercambio del archivos, asignado la dirección del grupo y subgrupos.	Los instructores pueden asignar a estudiantes a los grupos o el sistema puede crear aleatoriamente grupos. Cada grupo puede tener su propio foro, carpetas compartidas.
Autovaloración (Exámenes)	Los instructores pueden crear autovaloraciones.	Los instructores pueden crear autovaloraciones. El sistema corrige automáticamente las opciones múltiples, true/false. Los instructores pueden utilizar el editor de ecuaciones de MathML para permitir a estudiantes introducir y corregir notaciones matemáticas.
Comunidad del Estudiante	Los estudiantes pueden crear clubs y grupos de estudio en línea en el sistema. Los estudiantes pueden enviar email a sus grupos, utilizar un calendario compartido de aviso, y la parte privada de ficheros dentro del grupo	Los estudiantes de diversos cursos pueden obrar recíprocamente en system-wide a través del chat. Los estudiantes pueden crear los clubs en línea, y grupos de estudio en el de sistema
Listas del estudiante	Los estudiantes pueden utilizar su carpeta privada (en un curso para exhibir su trabajo. Las páginas personales pueden incluir su foto, información personal y enlaces a los websites importantes.	Los estudiantes pueden crear un Home Page personal.
Herramientas de administración.		
Autenticación	Los instructores pueden proteger el acceso a los cursos individuales con un username y contraseña. El sistema puede también authenticar contra un servidor externo de LDAP, mail server de IMAP/POP o usar el Kerberos o protocolo de RADIO. Los administradores pueden instalar una autenticación contra una fuente secundaria o en caso que la fuente primaria (e.g.. servidor de LDAP) falle. Las conexiones del usuario se pueden cifrar con el SSL. El sistema tiene una opción del recordatorio de la contraseña y los estudiantes pueden mantener sus propias contraseñas. Las contraseñas almacenadas en la base de datos del sistema se cifran.	Los administradores pueden proteger el acceso a los cursos individuales con un username y una contraseña. El acceso puede también ser restricto basado en IP ADDRESS. Las conexiones del usuario se pueden cifrar con el SSL. El sistema tiene una opción del recordatorio de la contraseña. El sistema puede también authenticar contra un servidor externo de LDAP o usar el protocolo del Kerberos. Los administradores pueden instalar una autenticación contra una fuente secundaria (e.g. los system?s poseer la base de datos) en caso que la fuente primaria (e.g.)fails del servidor de LDAP.

<p>Autorización del curso</p>	<p>Los instructores pueden asignar diversos niveles del acceso al sistema y a los cursos basados en los roles predefinidos siguientes: estudiante, instructor, ayudante de enseñanza. Los instructores o los estudiantes pueden ser asignados diversos papeles en diversos cursos y contextos del grupo.</p>	<p>Los administradores pueden asignar diversos niveles del acceso al sistema o a los cursos basados en los roles predefinidos siguientes: instructores, estudiantes, diseñadores, ayudantes de enseñanza, y administradores de sistema. Los instructores o los estudiantes pueden ser asignados diversos papeles en diversos cursos.</p>
<p>Servicios recibidos</p>	<p>Los sistemas recibidos son ofrecidos por un número de compañías situadas alrededor del mundo.</p>	<p>El abastecedor del producto ofrece un sistema recibido para las licencias estándares y superiores.</p>
<p>Herramientas para la entrega de tareas.</p>		
<p>Exámenes</p>	<p>Los instructores pueden crear preguntas del examen</p>	<p>Los instructores pueden crear preguntas true/false automáticamente, opción múltiple, emparejando, respuestas calculada, y preguntas cortas de respuesta. Los instructores pueden también crear preguntas de ensayo. Los instructores pueden importar preguntas de los bancos existentes de lprueba. Los instructores pueden fijar un límite de tiempo en una prueba. Los instructores pueden utilizar el redactor de la ecuación de la lengua del margen de beneficio de las matemáticas para permitir a estudiantes entrar en y corregir notaciones matemáticas. Los instructores pueden también utilizar direcciones del IP para restringir el acceso a las pruebas. Las ayudas del sistema proctored exámenes.</p>
<p>Gestión del curso</p>	<p>Los instructores pueden lanzar selectivamente los materiales basados en actividad anterior del curso. Los instructores pueden lanzar selectivamente los avisos basados en fechas específicas del comienzo y del extremo. Los instructores pueden ligar el contenido del curso a los acontecimientos específicos del calendario.</p>	<p>Los instructores pueden personalizar el acceso a los materiales específicos del curso basados en calidad de miembro de grupo, actividad anterior del curso, o funcionamiento del estudiante. El sistema puede sincronizar con las fechas del curso definidas por el calendario institucional.</p>
<p>Ayuda del instructor</p>	<p>Los instructores pueden formar grupos en línea para compartir documentos, y experiencias, con los instructores en su organización, usando foros de la discusión otras herramientas de colaboración. Los instructores pueden también someter un informe de apuro del helpdesk a</p>	<p>Los instructores pueden tener acceso al manual en línea de la ayuda, a la ayuda sensible del contexto, y a las comunidades numerosas de la ayuda del instructor para compartir la información en un número de foros disciplinar-específicos o generales del interés. Los instructores pueden tomar un curso en línea libre, y suscriben a una lista el</p>

	través de una forma en línea y seguir su resolución en la tela.	enviar del instructor.
Herramientas para calificar en línea	No	Los instructores pueden marcar todos los gravámenes anotados no automáticamente en línea. Los instructores pueden asignar el crédito parcial para ciertas respuestas. Los instructores pueden agregar los grados para las asignaciones fuera de línea al gradebook en línea. Los instructores pueden importar y exportar una versión coma-delimitada del gradebook de un programa de hoja de balance externo. Los instructores pueden corregir manualmente todos los grados. Los instructores pueden crear una escala que califica del curso. Los instructores pueden delegar la responsabilidad de calificar asignaciones. El gradebook apoya la creación de las columnas de encargo que pueden contener la información del grado u otros detalles instructor-determinados.

Viendo la tabla anterior y a modo comparativo se observa que WebCT dispone de más y mejores servicios, pero no es una diferencia notable para pensar que se apuesta por una plataforma pobre ya que el respaldo de las demás Universidades y la colaboración de la comunidad hace que sea una plataforma con nuevas releases mejorando la calidad de la misma.

Una vez vistos los objetivos y descrita la situación actual del sistema dejaremos la parte técnica en cuanto a requisitos de sistema para el capítulo en el que describiremos el Caso de Estudio en concreto.

1.5 Selección de alternativas

Basándonos en el estudio presentado por Dario Roig donde se han evaluado las principales plataformas que existen en el mercado tanto de pago como de libre distribución, a la lectura de otros proyectos sobre e-learning donde se evalúan otras plataformas de libre distribución y a la experiencia que demuestra que hayan universidades de tanto volumen y prestigio utilizando esta plataforma, se decide instalar dotLRN ya que en este momento es la favorita y cumple todas las funciones requeridas.

No incidiremos más en este apartado ya que la decisión de la plataforma a instalar ya esta tomada y por lo tanto no existe la necesidad de presentar otras alternativas para valorar, si que se tendrán en cuenta todas sus características para poder resaltar los puntos más fuertes o más débiles de dotLRN.

1.6 Viabilidad

A priori podemos decir que la viabilidad operativa del proyecto está garantizada debido a que existe un gran apoyo en la realización del proyecto por parte de la Universidad de Valencia.

Están muy interesados en que se pueda utilizar la plataforma para la docencia. Estos ven problemática la situación actual de su sistema de e-learning WebCT, por lo que son muy receptivos a cambios en este, es decir, no plantearán objeciones al abandono en un futuro de la plataforma actual.

Además son los propios interesados los que han decidido adaptar la plataforma a la Universidad de Valencia y dedicar recursos humanos para este fin.

CAPITULO 2

2. Estado del Arte

2.1 Introducción

La formación a distancia ha experimentado una gran expansión durante los últimos años, de tal manera que el número de instituciones, públicas y privadas, que imparten este tipo de enseñanza se ha duplicado en el último quinquenio, ya que mundialmente ha habido un notable aumento de alumnos que reciben tal tipo de enseñanzas.

La gran expansión de la enseñanza a distancia coincide con el “boom” de los medios tecnológicos de los últimos decenios, que ha supuesto el ofrecer numerosas herramientas facilitadoras de este tipo de enseñanza: radio, televisión, magnetófono, magnetoscopio, microordenador, software multimedia interactivo, la web, el correo electrónico, etc.

Si por un momento imagináramos que pudiéramos proponer a un alumno presencial y a otro alumno a distancia de hace cien años que continuaran sus estudios en el entorno actual, observaríamos que, mientras el alumno presencial apenas notaría la diferencia del tipo de clase que se le ofrece y podría seguir cualquier curso sin más (profesor impartiendo una clase magistral que, a lo sumo, utiliza algún medio de apoyo audiovisual – diapositiva, video o PC con proyección en pantalla-), el alumno a distancia de hace cien años sería incapaz posiblemente de continuar sus clases en un centro a distancia actual, ya que, con toda seguridad, necesitaría un auténtico curso previo de introducción a los medios tecnológicos que serían indispensables para su trabajo cotidiano: el teléfono, el fax, la radio, la televisión, el microordenador personal (procesador de textos y software interactivo multimedia, correo electrónico, chats, etc.)

Podemos concluir que, mientras la enseñanza presencial apenas ha cambiado, al menos en los últimos cien años, en la enseñanza a distancia el cambio ha sido espectacular, y muy especialmente gracias a la continua incorporación de nuevos medios tecnológicos.

La principal característica de los nuevos entornos de aprendizaje virtual es que utilizan la Web como única plataforma de distribución. Esto implica que disponen de una serie de capacidades hasta ahora inexistentes para que las aplicaciones de enseñanza asistidas por ordenador permitan superar algunas de las deficiencias que existían en los sistemas de e-learning del ayer:

- Desaparición real de los problemas de espacio y de horarios. Los alumnos pueden realizar su aprendizaje desde cualquier sitio y a cualquier hora. Los sistemas de enseñanza asistida por computadora abren 24 horas al día.
- Proporcionan un canal de comunicación entre alumnos, y entre éstos y profesores. De este modo desaparece por completo la relación que algunos establecen entre estudiar a distancia y aprender en solitario. Es posible ofrecer entornos de telepresencia en donde todos los participantes en el proceso formativo compartan un "aula virtual". La participación de los estudiantes puede llegar a ser superior en un entorno como éste que en un aula convencional.
- El canal de comunicación que se establece puede utilizarse con finalidades de seguimiento y tutorización de los alumnos por parte de los profesores. La información extraída de este seguimiento puede ser empleada posteriormente para labores de evaluación.
- Dado que, habitualmente, el ámbito de actuación de estos sistemas es universal, los alumnos pueden elegir entre una gran diversidad de materias, cursos y especialidades. Éstos pueden ser preparados por los mejores especialistas en cada materia, para ser distribuidos a un conjunto amplio de estudiantes dispersos geográficamente. No sólo aquellos alumnos cercanos físicamente a ellos se benefician de sus conocimientos.

Al mismo tiempo aparecen una serie de problemas específicos asociados a la utilización masiva de Internet entre los que cabe mencionar:

- El aumento en la heterogeneidad de los productos y en la interacción entre personas y sistemas, únicamente entre personas y únicamente entre sistemas.
- El espectacular aumento de información disponible y su dispersión en distintos sistemas y aplicaciones, lo que implica la necesidad de poner en comunicación distintos productos software y plataformas.

- La organización de procesos de aprendizaje distribuidos, motivada por la dispersión geográfica de los usuarios de los cursos.

El profundo cambio que ha supuesto Internet en el campo de las aplicaciones educativas no sólo afecta a aspectos puramente tecnológicos: también los paradigmas educativos se están alterando como consecuencia de las nuevas demandas sociales, en las que las redes globales de comunicación juegan un papel esencial. Actualmente se imponen los modelos educativos centrados en el alumno: ha crecido la demanda hacia una formación continua y “de por vida” en contraposición a los métodos educativos tradicionales en los que la formación se recibe en periodos determinados y centros específicos. Se trata de formar profesionales con poco tiempo disponible y que exigen obtener un elevado rendimiento de su tiempo.

Debemos destacar también el componente principal de cualquier aplicación educativa: los contenidos educativos. Parece evidente que en última instancia, el éxito de una aplicación educativa radica en la calidad de sus contenidos. Por lo general, en la mayoría de las aplicaciones disponibles, los cursos se crean con el fin de cubrir una necesidad de aprendizaje concreta. Sin embargo, producir desde cero materiales educativos de alta calidad es una labor ardua que lleva mucho tiempo y requiere conocimiento de diversos expertos en distintos campos.

Además, en los últimos años, hemos sido testigos de cómo los esfuerzos para adaptar contenidos educativos a un formato determinado (previas inversiones de grandes cantidades de tiempo y dinero) se perdían al realizar cambios de sistema o actualizaciones de las aplicaciones: los contenidos estaban íntimamente ligados a su formato y herramienta de creación/visualización.

La gestión del conocimiento se ha convertido en un tema recurrente en informática. En tal discurso, el conocimiento no sólo se identifica cómo un nuevo factor de producción de las sociedades preindustriales, sino cómo un producto en sí mismo. Para poder determinar su valor de cambio, el conocimiento debe ser creado, almacenado y gestionado, razón por la cual emergen las tecnologías de la información y comunicación cómo soporte a la gestión del conocimiento.

Por tanto, aparece una necesidad de encontrar procedimientos que permitan la reutilización efectiva de material ya desarrollado. Idealmente, que faciliten la reutilización no sólo dentro de la misma aplicación para distintos cursos, sino entre aplicaciones diferentes con herramientas de creación de contenidos y plataformas distintas.

2.2 Definición de e-learning

Podemos definir el e-Learning como el suministro de programas educativos y sistemas de aprendizaje a través de medios electrónicos. El e-Learning se basa en el uso de una computadora u otro dispositivo electrónico (por ejemplo, un teléfono móvil) para proveer a las personas de material educativo.

La educación a distancia creó las bases para el desarrollo del e-Learning, el cual viene a resolver algunas dificultades en cuanto a tiempos, sincronización de agendas, asistencia y viajes, problemas típicos de la educación tradicional.

Así mismo, el e-Learning puede involucrar una mayor variedad de equipo que la educación en línea. El término de e-Learning o educación electrónica abarca un amplio paquete de aplicaciones y procesos, como el aprendizaje basado en Web, capacitación basada en computadoras, salones de clases virtuales y colaboración digital (trabajo en grupo).

Diversas definiciones de e-Learning

El concepto de e-Learning es comprendido fácilmente por la mayoría de la gente. Aun así, esta industria tiene pendiente una definición precisa de este término. Para darnos una idea de las variantes que existen actualmente en la concepción del aprendizaje electrónico, examinemos algunas de las definiciones más comunes:

Técnicamente, el e-Learning es la entrega de material educativo vía cualquier medio electrónico, incluyendo el Internet, Intranets, Extranets, audio, vídeo, red satelital, televisión interactiva, CD y DVD, entre otros medios.

Para los educadores, e-Learning es el uso de tecnologías de redes y comunicaciones para diseñar, seleccionar, administrar, entregar y extender la educación.

Siendo breves y prácticos, el e-Learning es el aprendizaje basado en tecnología.

Para los elocuentes, el e-Learning es el empleo del poder de la red mundial para proporcionar educación, en cualquier momento, en cualquier lugar.

Siendo descriptivos, la educación electrónica es la capacitación y adiestramiento de estudiantes y empleados usando materiales disponibles para Web a través del Internet, llegando a ofrecer sofisticadas facilidades como flujo de audio y vídeo, presentaciones en PowerPoint, vínculos a información relativa al tema publicada en el Web, animación, libros electrónicos y aplicaciones para la generación y edición de imágenes.

Para los epigrafistas, el e-Learning representa la convergencia del aprendizaje y el Internet.

Y finalmente, para los visionarios y futuristas, el e-Learning es a la educación convencional lo que el e-Business a los negocios ordinarios.

Aunque es importante destacar que no porque se envíe un curso electrónicamente, tendrá necesariamente que tratarse de e-Learning.

2.2.1 Características de e-learning

Conforme se van perfilando los requerimientos exigidos a los entornos de enseñanza a través de la Web, se hace más evidente que éstos son demasiado complejos y numerosos para ser cubiertos por un único sistema monolítico. Por el contrario, la tendencia actual es la de integrar un conjunto de sistemas especializados. A nuestro modo de ver, la arquitectura de un sistema e-learning desarrollado mediante la integración de un conjunto de componentes independientes, debería tener las siguientes características:

- *Abierta*. El objetivo es crear aplicaciones e-learning interoperables y conectables entre sí de forma sencilla ("plug-and-play"), es decir, que

herramientas comerciales de fabricantes distintos puedan ensamblarse en un único sistema global. Para ello es necesario que el marco de definición de la arquitectura del sistema sea conforme a un modelo estándar.

- *Escalable*. Independientemente del tamaño inicial con que se conciba el sistema, la arquitectura debe estar definida de tal forma que permita su crecimiento. Por ejemplo, al ir aumentando el repositorio de objetos educativos las aplicaciones encargadas de gestionarlos deben tener capacidad suficiente para no sobrecargarse.
- *Global*. Permitir la diversidad lingüística y cultural. Este es uno de los objetivos con mayor dificultad, puesto que la gran mayoría de las aplicaciones están destinadas para una audiencia anglosajona: actualmente existen varios esfuerzos de estandarización en marcha cuyo objetivo es presentar un mismo contenido (incluso un mismo entorno de aprendizaje) en diferentes lenguas en función del usuario a quien esté destinado.
- *Integrada*. No sólo entre los componentes del propio sistema, sino entre otras aplicaciones no directamente relacionadas con el aprendizaje (e.g. recursos humanos, financieras, de gestión del conocimiento). El objetivo es conseguir la interoperabilidad entre todas ellas.
- *Flexible*. Es importante la capacidad de implementar nuevas soluciones sin tener que efectuar grandes cambios en la arquitectura del sistema.

Por otro lado, la arquitectura está fuertemente condicionada por los roles definidos para los actores que interactúan con el sistema. Identificamos tres categorías básicas entre los usuarios de un entorno e-learning:

- Consumidores de e-learning. A esta categoría pertenecen alumnos y profesores: ambos interactúan con las mismas herramientas, aunque la forma de interacción varía.
- Proveedores de e-learning. En este grupo están englobados los proveedores o creadores de contenidos educativos (responsables de la tarea de diseñar el proceso de instrucción de tal forma que se consiga un cierto objetivo educativo).
- Coordinadores. En este último grupo englobamos a los administradores del sistema (cuya tarea es la de gestionar los elementos de los catálogos, los horarios, los recursos -sesiones de

aprendizaje, tutores, equipos disponibles- y los aspectos económicos).

2.2.2 Componentes de un sistema e-learning (LMS)

Aunque las funcionalidades de un sistema de enseñanza pueden ser muchas y muy variadas, se pueden agrupar en tres sistemas independientes y bien diferenciados:

Sistema de Gestión de Contenidos

Dentro de este primer grupo, se engloban todas aquellas tareas relacionadas con la creación, recuperación, ensamblado, localización y almacenamiento de contenidos y recursos educativos. Está asociado al rol de los proveedores de e-learning y sus componentes básicos son:

- Una herramienta autoría e integración de los recursos educativos, cuyo objetivo sea la creación de cursos completos (a través de la integración de objetos educativos mediante la definición de jerarquías) o de recursos educativos atómicos (dentro de estos recursos también se deben considerar las evaluaciones). Debe permitir la creación de contenidos en cualquier formato (texto, vídeo, audio, imágenes, etc.) y asociarlos al nivel deseado en la jerarquía del curso. El API (Application Programming Interface) de esta herramienta también debe permitir la importación de recursos ya creados y ensamblarlos al curso que se está creando. La mayoría de las herramientas de este tipo que encontramos en el mercado se ajustan a las especificaciones IMS y SCORM.
- Un sistema de creación y almacenamiento para los metadatos asociados a los objetos educativos.
- Un almacén de objetos educativos. Debe disponer de todos los servicios tradicionales asociados a un repositorio tradicional, como control sobre la versión, actualizaciones, notificaciones de modificación, bloqueo de entrada/salida de los objetos etc.
- Una herramienta para la gestión, localización y recuperación de los recursos educativos. Esta herramienta utilizará el repositorio de metadatos y el de los recursos educativos para localizar y recuperar un objeto concreto atendiendo a los criterios de búsqueda solicitados

- por el usuario. El autor visualiza previamente el objeto y determina su aplicabilidad al nuevo contexto.
- Una herramienta para ensamblar los contenidos educativos, que permitirá al autor recuperar un objeto a través de la herramienta de gestión, localización y recuperación de recursos e integrarlo en la estructura del curso por medio de una plantilla o jerarquía almacenada en el almacén de contenidos. La herramienta de ensamblado debe permitir también visualizar previamente los contenidos para, de esa manera, facilitar al autor la labor de ensamblado. Una de las soluciones posibles para realizar la visualización de los objetos marcados mediante XML son las hojas de estilo.
 - Una herramienta de publicación. Cuando se libera una oferta de aprendizaje debe publicarse a través del entorno de distribución. La herramienta de publicación tiene como función asignar un conjunto de metadatos que permiten al entorno de distribución conocer la forma adecuada para explotar la oferta. Una vez publicada, los sistemas de gestión del aprendizaje pueden ofrecerla a sus alumnos.

Sistema de Distribución de Contenidos

Su función es la determinar la forma en que se van a presentar los contenidos al usuario, para ello es recomendable disponer de las herramientas adecuadas para llevar a cabo las siguientes funciones:

- Presentación de contenidos. Su función es la de generar de manera dinámica contenidos personalizados contrastando el perfil o modelo del alumno y sus preferencias, con los contenidos: recogerá la información correspondiente al perfil del alumno e identificará los contenidos que se adecuan a sus características. También identifica la plantilla adecuada para presentar un curso o lección determinados y el medio de distribución elegido (Web, impresión, etc.).
- Gestión de la distribución. Si existen varias localizaciones posibles del material requerido para publicación, esta herramienta se encargará de determinar la más próxima al usuario para optimizar la velocidad de presentación.

Sistema de Gestión del Aprendizaje (Learning Management System)

Esta área está asociada al rol de los consumidores de e-learning (alumnos y tutores). El término LMS, por lo general, hace referencia a un tipo de sistema diseñado para realizar el seguimiento de la interacción con los contenidos educativos y realizar la gestión de los mecanismos de interacción con el sistema, incluyendo la navegación, la selección de las ofertas educativas y la conexión con el sistema de distribución de contenidos.

El objetivo primario de un LMS es proporcionar y administrar los diferentes recursos de un sistema de enseñanza a los estudiantes, siguiendo su progreso y su rendimiento. En particular, las características principales de un LMS, que lo diferencian de las herramientas tradicionales de enseñanza (CMI, Computer Managed Instruction), son:

- Personalización. La combinación entre los perfiles de usuario y sus preferencias personales es la base para conseguir un entorno personalizado dinámicamente. Los perfiles contienen información obtenida por el sistema sobre el alumno como su nivel de conocimiento, las puntuaciones obtenidas mediante los sistemas de evaluación, el tiempo de conexión a los cursos, etc. Las preferencias personales son introducidas por el alumno y almacenan datos relativos al formato de presentación, al idioma natal, etc.
- Búsqueda y navegación en los catálogos educativos. El alumno puede acceder a las ofertas educativas disponibles y seleccionar la que desea.
- Registro. Ofrece facilidades para registrarse en un determinado curso que se conoce a través del catálogo. Este mecanismo gestiona también el registro de datos históricos del alumno, la planificación del aprendizaje, las notificaciones sobre novedades y modificaciones de los cursos, los horarios, los tutores encargados, las listas de espera y la descatalogación o recomendación de ciertas ofertas educativas para un perfil determinado de alumno.
- Seguimiento de la evolución del alumno. Su función es la de registrar el progreso del alumno a través de su interacción con una determinada oferta educativa. Almacena datos como tiempo invertido en cada curso, cursos visitados, puntuación obtenida en las pruebas parciales, etc.
- Mecanismos de valoración y evaluación. Inclusión de mecanismos de evaluación anteriores y posteriores al seguimiento de un curso. Las evaluaciones previas permiten definir mecanismos para pre-

seleccionar únicamente aquellos contenidos que se ajustan a los conocimientos previos del alumno. Las evaluaciones posteriores aportan información para llevar a cabo el seguimiento del progreso del alumno.

- Definición de procesos de colaboración on-line. Mecanismos síncronos y asíncronos de comunicación entre todos los implicados en la experiencia educativa. Administración de las conexiones con otros sistemas de información. Necesaria para la compartición de objetos educativos, y para la importación y exportación de perfiles de alumnos.

El siguiente esquema (Figura 1) muestra el conjunto de funcionalidades que residen en el LMS.

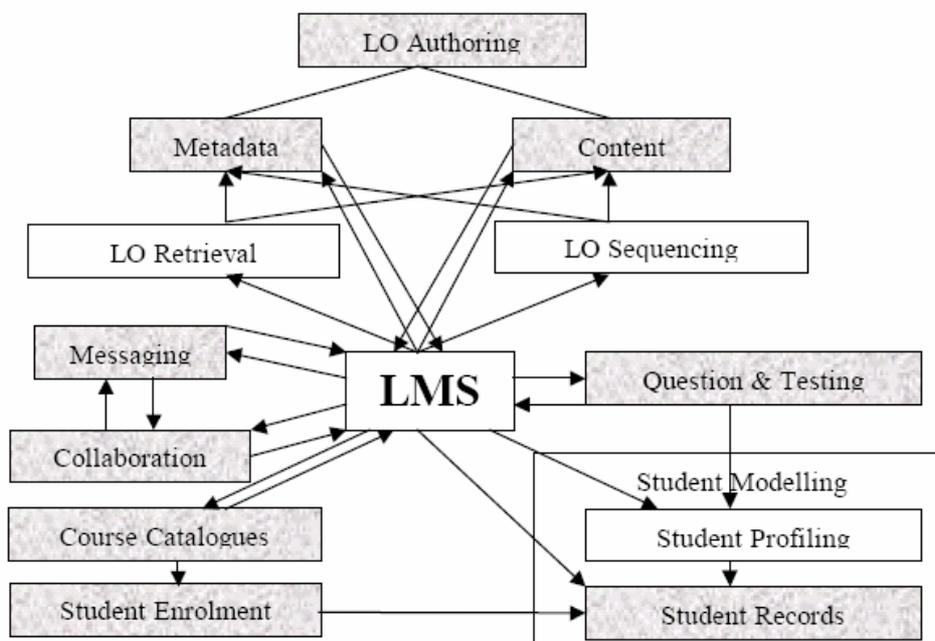


Ilustración 1. Modelo referencia de definición de un LMS. Interacción entre sus componentes (Fernández-Manjón 2002).

2.2.3. Courseware o Contenidos

Es el material de aprendizaje que se pone a disposición del alumno. Los contenidos para e-learning pueden estar en diversos formatos, en función de su adecuación a la materia tratada. El más habitual es el WBT (Web Based Training), cursos online con elementos multimedia e

interactivos que permiten que el usuario avance por el contenido evaluando lo que aprende.

Sin embargo, en otros casos puede tratarse de una sesión de "aula virtual", basada en videoconferencia y apoyada con una presentación en forma de diapositivas tipo Powerpoint, o bien en explicaciones en una "pizarra virtual". En este tipo de sesiones los usuarios interactúan con el docente, dado que son actividades sincrónicas en tiempo real. Lo habitual es que se complementen con materiales online tipo WBT o documentación accesoria que puede ser descargada e impresa.

Otras veces el contenido no se presta a su presentación multimedia, por lo que se opta por materiales en forma de documentos que pueden ser descargados, complementados con actividades online tales como foros de discusión o charlas con los tutores. En definitiva, cualquier tipo de representación de los contenidos puede venir conjuntada con las demás y todas formar parte de un mismo sistema elearning.

2.2.4 Comunicación sincronía y asíncrona

Una vez vistas las principales características de los LMS debemos hacer una distinción entre los dos tipos de comunicación existente:

a) Sincrónica: Los alumnos se conectan en tiempo real con el tutor y/o profesor, por tanto el proceso de aprendizaje sólo queda diferido en el espacio. Todos los participantes están conectados a la vez y se comunican directamente entre ellos. Los ejemplos incluyen clases virtuales, conferencias de audio y vídeo, telefonía de Internet, y reproducciones vía satélite en directo de ida y vuelta de clases presenciales a estudiantes.

b) Asíncrona: Los alumnos se conectan, de acuerdo con sus tiempos, dentro de los parámetros que imponga el curso. En este caso el material con que se trabaja asume mayor importancia al igual que las tutorías, por tanto el proceso de aprendizaje queda diferido en el tiempo y en el espacio. Unos ejemplos pueden ser cursos tomados vía Internet o CD-ROM a tu propio ritmo, clases en vídeo, presentaciones Web con recursos audio/vídeo, chats online y grupos de discusiones y e-mail.

Un sistema sincrónico es aquel que ofrece comunicación en tiempo real entre los estudiantes o con los tutores. Por ejemplo, las charlas o la videoconferencia. Los sistemas asincrónicos no ofrecen comunicación en tiempo real, pero por el contrario ofrecen como ventaja que las discusiones y aportes de los participantes quedan registrados y el usuario puede estudiarlos con detenimiento antes de ofrecer su aporte o respuesta. La diferencia fundamental entre el e-learning y la enseñanza tradicional a distancia está en esa combinación de los tres factores, en proporción variable en función de la materia a tratar: seguimiento + contenido + comunicación.

Los distintos componentes de una solución e-learning se pueden ver del siguiente modo:

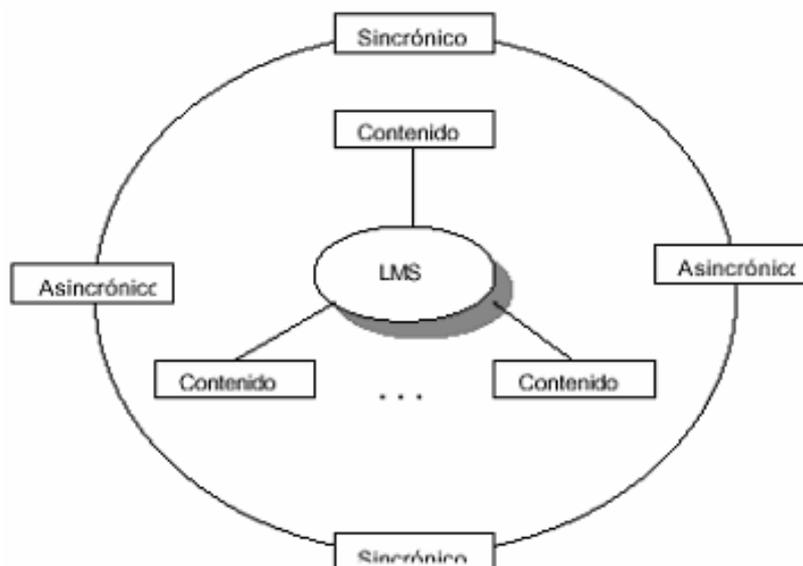


Ilustración 2. Relación entre los elementos de un LMS

2.2.4 Metadatos

Los metadatos son *información acerca de información*. Los metadatos consisten en un conjunto de propiedades de un documento, son la estructuración de la información. Por definición, los metadatos son datos, además de *datos sobre datos*. (BERNERS-LEE).

La mayoría de los estándares actuales se centran principalmente en el intercambio de datos y muchos de los esfuerzos futuros, durante algún tiempo, girarán en torno al intercambio de datos.

Los metadatos juegan un papel importante en la consecución de un nivel de *información compartida* [interoperatividad] entre diferentes comunidades con diferentes tipos de información y tecnología para crear nuevos y más poderosos tipos de información.

Precisamente, la interoperatividad es uno de los dos objetivos generales propuestos por el Consorcio W3C para alcanzar con la implementación del Web Semántico -el otro objetivo es la *evolutividad* -. Los metadatos son la base de los Learning Objects sin los cuales no existiría el aprendizaje a distancia, por esta razón son tan importantes y deben ser analizados.

Se aprecia un desarrollo parejo entre las propuestas sobre metadatos y la implementación del Web Semántico, que centra gran parte de sus trabajos actuales en torno a las *ontologías*.

RDF es una tecnología para implementar metadatos. RDF (Resource Description Framework) es una especificación propuesta por el W3C que ofrece *interoperatividad* entre aplicaciones que intercambian *información comprensible por la máquina* en la Web [Web Semántico] para proporcionar una infraestructura que soporte actividades de metadatos.

Los Objetos de Aprendizaje son elementos de un nuevo tipo de instrucción basada en computador originados en el paradigma de orientación a objetos. Se entienden como *entidades digitales que se pueden distribuir a través de Internet*, por lo cual mucha gente puede acceder a ellas simultáneamente y los diseñadores pueden utilizarlos para construir pequeñas piezas de componentes instruccionales reutilizables en diferentes contextos. Estas piezas pueden ser autocontenidas y pueden incluir en su estructura otros objetos. Además, la noción de pequeñas piezas de material instruccional, sugiere que esas partes se pueden reensamblar para soportar objetivos instruccionales individuales.

Para hacer referencia a estos Objetos de Aprendizaje se utilizan diferentes nombres, lo que ha creado confusión y ha dificultado, en cierta forma, la comunicación entre los grupos interesados en su utilización. Por tal motivo, ha sido necesario llegar a un consenso para elegir el término de mayor aceptación entre la comunidad. Con el objetivo de promover el uso de los Objetos de Aprendizaje, el LTSC (*Learning Technology Standards*

Commitee) del IEEE, seleccionó el término *Learning Objects* (Objetos de Aprendizaje) para describir esos pequeños componentes instruccionales, y estableció un grupo de trabajo, el cual se encargó de proporcionar una definición: "cualquier entidad digital, o no digital, que puede ser usada, reutilizada o referenciada durante el aprendizaje soportado en ordenador. David A. Wiley II propone otra definición basada en la propuesta por el LTSC (IEEE), y se refiere a los Objetos de Aprendizaje como "cualquier recurso digital que puede ser reutilizado para soportar aprendizaje". Esta es una definición en apariencia más "estrecha", pero que sintetiza mucho mejor la idea de lo que debe ser un Objeto de Aprendizaje. Se diferencia de la definición del LTSC en dos aspectos fundamentales. En primer lugar, descarta explícitamente los recursos no digitales y no reutilizables y elimina la frase "soportados en tecnología", puesto que implícitamente todos los Objetos de Aprendizaje son digitales (es decir, se representan digitalmente). En segundo lugar, la palabra "durante" en la definición del LTSC se sustituye por la frase "para soportar", puesto que al decir *el uso de un objeto "durante" el aprendizaje* no conecta al objeto con su uso en el aprendizaje.

Al hacer una revisión de los recursos didácticos ofrecidos a través de Internet, es frecuente encontrar que muchos de ellos tienen tópicos parecidos. Lo deseable sería poder encontrar cursos bien estructurados con tópicos genéricos que abarcaran las características más relevantes del curso, y que además las instituciones y grupos interesados en el desarrollo de cursos pudieran compartir sus recursos para generar nuevos materiales instruccionales. Es precisamente ésta una de las ventajas que ofrece la tecnología de Objetos de Aprendizaje. Desde la perspectiva de la reusabilidad la definición de componentes hace decrecer el tiempo y el costo de desarrollo de contenidos, y desde la perspectiva de distribución, un alto nivel de individualización permite la personalización curricular de acuerdo con los intereses y necesidades individuales. Además, el uso de Objetos de Aprendizaje en el desarrollo de cursos, favorece la implementación de más y mejores sistemas de categorización y búsqueda, mecanismos robustos para actualización y envío de datos, y la estructuración y definición de recursos educativos.

Dos de los estándares de metadatos en el ámbito educativo son: Dublin Core y LOM (Learning Object Metadata), que actualmente trabajan en colaboración. Existen otras propuestas de estándar como EML (Educational Modelling Language), centrada en los *procesos* de aprendizaje.

Metadatos	Estructuración de contenidos (partes de un recurso)
Ontología	Estructuración de la semántica del recurso. Descripción formal de contenidos Modelar conceptos y relaciones
RDF	Tecnología para describir recursos Posibilita metadatos en la Web

Ilustración 3. Relación entre metadatos, ontologías y RDF.²

2.3 Estándares

El proceso de elaboración de un estándar es similar al de creación y aprobación de las leyes: una vez se ha realizado el grueso del trabajo, este debe ser ratificado por un organismo oficial. Puede parecer un proceso lento y poco efectivo, pero hay que tener en cuenta que el éxito de un estándar radica en su nivel de aceptación, por lo que un grupo de estandarización debe ser un organismo que se encargue de recopilar requisitos de múltiples fuentes y elabore con ellos una especificación consensuada.

La obtención de un estándar formal se consigue como resultado de los esfuerzos combinados de numerosos organismos y consorcios que se agrupan de acuerdo a tres niveles de trabajo (Lindner 2001):

- Nivel de especificación. En este primer paso del proceso, se trabaja en la elaboración de recomendaciones basadas en el análisis de las necesidades de los propios participantes. El objetivo es proponer la especificación elaborada a la comunidad e-learning. Es en este nivel dónde se encuentra IMS.

- Nivel de validación. En esta fase del proceso, se desarrollan nuevos productos que incorporan las especificaciones elaboradas en el paso anterior y se inician programas piloto con el fin de valorar la efectividad y aplicabilidad de la especificación. Así mismo, se crean modelos de referencia que muestran cómo las distintas especificaciones y estándares pueden ensamblarse para integrar un sistema e-learning global. En este nivel se encuentra la iniciativa ADL.

- Nivel de estandarización. Es el paso final de la elaboración, las especificaciones que ya han sido validadas, son retomadas por los

² [5] Véase bibliografía.

organismos oficiales de estandarización que se encargan de darles un último refinamiento, consolidación, clarificación de los requisitos que satisfacen y acreditación. Es importante distinguir entre la especificación (que es un proceso de trabajo en evolución) y el estándar acreditado (basado en implementaciones reales y las experiencias obtenidas). Algunos de los grupos y consorcios del sector están implicados en dos o incluso en todas las fases anteriores. La razón es que actualmente no existe prácticamente solapamiento entre las actividades realizadas por los grupos: cada nivel juega un papel distinto con responsabilidades diferentes. De hecho, uno de los principales intereses de la comunidad e-learning en estos momentos es el de clarificar los roles y responsabilidades de las distintas agrupaciones y coordinar sus actividades. En la siguiente figura (ilustración 4) se muestran los tres niveles que acabamos de presentar, y dónde se encuadran las diferentes organizaciones dentro de esos niveles.

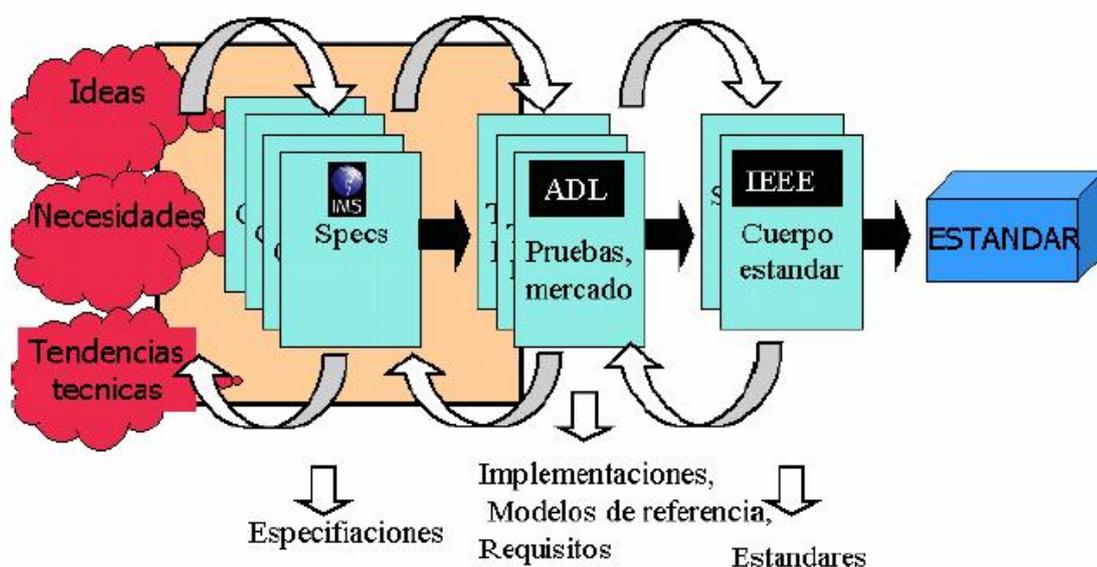


Ilustración 4. Niveles de trabajo.

2.3.1 Que se debe estandarizar

Hay que hacer un repaso por los diferentes aspectos de los sistemas de enseñanza que son candidatos a ser estandarizados, y en los cuales existen grupos de trabajo definiendo recomendaciones y propuestas.

- *Requisitos técnicos.* No todas las computadoras tienen las mismas características. Existen recomendaciones orientadas a la definición de las características que deben poseer los ordenadores sobre los que se ejecuten

los LMS. Habitualmente, las instituciones que publican recomendaciones a este respecto realizan también propuestas de los formatos multimedia que deben ser utilizados dentro de los propios recursos educativos así como los protocolos para su transmisión a través de la red.

- *Organización de los contenidos educativos.* Cada sistema de aprendizaje tiene sus propios formatos para definir la organización de un curso. Para poder transferir cursos entre sistemas dispares es necesario alcanzar definiciones comunes, tanto para el formato de definición de su estructura estática, como para el que determina su comportamiento dinámico.

- *Información personal y académica del alumno.* De la misma manera que formatos comunes para la definición de cursos permiten la transferencia de éstos entre sistemas heterogéneos, la existencia de formatos únicos para la definición de expedientes de los alumnos hace posible su exportación a otros sistemas educativos. Es necesario decidir la información que debe estar presente en un expediente y el formato que se utilizará para definirla. Dentro de los estándares para perfiles y expedientes de alumnos se ha incluido información estática (e.g. datos personales) que no depende de la interacción del alumno con el sistema de aprendizaje, y también información dinámica (e.g. calificaciones) que se modifica a medida que el alumno avanza en su proceso de aprendizaje.

- *Material de evaluación.* Se definen diferentes formatos de presentación de un mismo cuestionario y se establecen mecanismos para determinar los diferentes criterios que deben ser utilizados a la hora de ejecutar las evaluaciones automáticas. Al igual que en los casos anteriores, un estándar en los materiales de evaluación permite el intercambio y reutilización de las evaluaciones.

- *Definición de formatos para descripciones de recursos.* La descripción de los recursos educativos cobra una enorme importancia en el momento que se pretende que los sistemas que los utilizan se los intercambian. Por eso, es necesario establecer formatos comunes para la definición de aspectos como el tipo de contenido que se ofrece, sus relaciones con otros, a quién está dirigido, requisitos técnicos, etc. Todo ello con el propósito de facilitar la búsqueda y localización de los recursos más adecuados para unas necesidades de aprendizaje concretas. La tendencia

actual se enfoca hacia la utilización de metadatos. Este es el único área de todas las que exponemos que actualmente cuenta con un estándar.

- *Mecanismos de transferencia de cursos.* Aparte de la definición de formatos comunes para la definición y estructuración de los cursos y los procedimientos adecuados para su localización, es necesario establecer mecanismos que faciliten la transferencia de cursos encapsulados desde las instituciones dedicadas a su desarrollo a aquellas especializadas en la provisión de servicios educativos utilizando recursos electrónicos. Actualmente existen grupos de trabajo dedicados a la definición de formatos de empaquetado para todos los elementos necesarios en la transferencia de un curso. De este modo es posible la transferencia de todos ellos a través de una entidad individual y de una manera muy sencilla. En el siguiente capítulo estudiaremos en detalle una de las organizaciones encargadas de la definición de este tipo de formato.

- *Entornos de ejecución.* Para permitir la reutilización de contenidos es necesario establecer una clara separación entre éstos y los sistemas software encargados de su gestión (LMS). Los entornos de ejecución son responsables de la entrega de contenidos a los alumnos, de la supervisión de la interacción de éstos con los contenidos, y de decidir cuál es el siguiente recurso educativo que debe ser entregado sobre la base de la estructuración del curso y a las interacciones previas del alumno. Los sistemas de gestión de aprendizaje incluyen los entornos de ejecución dentro de ellos, pero aparte poseen otras muchas herramientas que facilitan el proceso de aprendizaje. La interoperabilidad entre contenidos y entornos de ejecución de diferentes fabricantes vendrá dada por una interfaz creada por los grupos que se encargan de la interoperabilidad de estos entornos.

- *Arquitecturas software.* Este parte se refiere a la identificación de los diferentes componentes software que deben constituir un sistema de aprendizaje electrónico distribuido. Aunque se realizaron algunas propuestas iniciales en este sentido, no se cubrieron los objetivos deseados. De hecho, no han sido publicadas recomendaciones para las interfaces software que deben ofrecer componentes como los mencionados.

- *Servicios de intermediación y búsqueda.* Otro aspecto que es objeto de estudio es la definición de arquitecturas para la construcción de buscadores y sistemas de intermediación. Su objetivo final es permitir la localización de los recursos educativos más adecuados a unas necesidades

concretas. Los modelos de información utilizados por estos sistemas están basados en propuestas para metadatos educativos como las introducidas más arriba.

2.3.2 Organizaciones para la estandarización

Hay una diferencia importante en la manera en que se conduce el proceso de estandarización entre Europa y América: el americano es un proceso en el que el mercado y las iniciativas comerciales tienen un mayor peso, mientras que en Europa el motor principal son las iniciativas gubernamentales y las instituciones oficiales. Además, hasta la adopción de XML como vía de implementación, no existían mecanismos de expresión común: de hecho, con la adopción de XML como estándar de facto para la representación de metadatos por parte de la comunidad educativa, se ha llegado a una serie de consensos que han unificado la labor entre los principales grupos de estandarización.

Actualmente se está produciendo un proceso de confluencia y colaboración entre todas las iniciativas implicadas en la estandarización. A continuación se hace un repaso breve a las principales organizaciones involucradas en este proceso de estandarización.

LTSC

El comité de estandarización de tecnologías aplicadas al aprendizaje, Learning Technology Standardization Committee (LTSC) , perteneciente al Institute of Electrical and Electronics Engineers (IEEE), abarca prácticamente todos los aspectos del aprendizaje basado en ordenador.

Su misión principal es "desarrollar estándares técnicos, prácticas recomendadas y guías para componentes software, herramientas, tecnologías y métodos de diseño que faciliten el desarrollo, implantación, mantenimiento e interoperabilidad de implementación en ordenadores de sistemas educativos".

Los 15 subcomités que componen el LTSC están organizados en 5 áreas de trabajo: (1) aspectos generales, (2) aspectos relacionados con el contenido, (3) aspectos relacionados con el aprendiz, (4) datos y metadatos y (5) sistemas y aplicaciones de gestión.

Las primeras propuestas del LTSC están relacionadas con una arquitectura de sistema y un modelo de referencia para un sistema LMS, un modelo de datos para describir al aprendiz y metadatos educativos. Asimismo están analizando propuestas de organizaciones externas.

IMS

El proyecto IMS (IMS 2002) fue creado en 1997 por EDUCASE, un consorcio de instituciones educativas y sus socios empresariales como una tentativa de desarrollar estándares abiertos para sistemas de enseñanza asistida por computadora.

Los primeros trabajos del IMS se centraron en la definición de un modelo y una arquitectura para sistemas de aprendizaje distribuido. Sin embargo, sus esfuerzos se reorientaron rápidamente al percatarse de que primero necesitaban un modelo de datos adecuado para describir los recursos, estructuras y demás elementos manejados por los componentes de la arquitectura. Hoy en día los trabajos de IMS se centran en: metadatos, empaquetado de contenidos, definiciones de tests y cuestionarios, especificaciones de perfiles de alumnos y gestión de grupos, recomendaciones para objetos educativos distribuidos, organización de cursos bajo enfoques pedagógicos y secuenciamiento.

ADL/SCORM

En noviembre de 1997 el Departamento de Defensa de los EE.UU. y la oficina de Ciencia y Tecnología de la Casa Blanca lanzaron la iniciativa Advanced Distributed Learning (ADL) (ADL 2002). ADL surge como respuesta a las necesidades de uno de los mayores consumidores de software del mundo y forma parte del esfuerzo que el gobierno norteamericano viene realizando con el objetivo de conseguir una enseñanza de calidad, en el que también están implicados los departamentos de Educación y Trabajo.

ADL se ha centrado desde un principio en el aprendizaje sobre la Web. Su trabajo ha acompañado al de otras instituciones, principalmente IEEE, IMS y AICC, para buscar aquellos puntos críticos del aprendizaje sobre la web en los que sería recomendable especificar interfaces consensuadas.

ADL ha sido una de las organizaciones más activas en las reuniones y encuentros de estos organismos y fruto de ello es un conjunto de especificaciones que, bajo la denominación SCORM (Sharable Content Object Reference Model) (SCORM 2001), modelo de referencia para objetos educativos software que puedan compartirse, propone un entorno de ejecución y un modelo de metadatos y estructuras de cursos. La versión 1.1 de SCORM fue publicada el 31 de enero de 2001.

ADL espera que el ámbito de aplicación de la especificación se vea ampliado en el futuro y que sirva como punto de convergencia para el resto de recomendaciones de otras instituciones implicadas en el proceso de estandarización. En 1999 el ADL estableció el Co-Lab como grupo de trabajo responsable de comprobar y validar las nuevas recomendaciones de la iniciativa ADL, verificar el grado de cumplimiento del estándar por parte de productos comerciales externos y realizar las certificaciones correspondientes. Asimismo, este grupo es el responsable de desarrollar prototipos conformes a SCORM y divulgar los resultados obtenidos.

Finalmente, el grupo TFADLAT, Total Force Advanced Distributed Learning Action Team, es la interfaz entre la iniciativa ADL y el secretario de defensa norteamericano. Se encarga de realizar recomendaciones al Departamento de Defensa en aspectos relacionados con el aprendizaje. Es el grupo responsable de asegurar que la plantilla interna de este departamento tenga acceso a los métodos y tecnologías de aprendizaje más avanzados, así como promover los estándares especificados y desarrollados por la organización.

A día de hoy, ADL es el modelo de referencia para las especificaciones de IMS, que a su vez es el productor de especificaciones para ADL.

AICC

El comité para CBTs de la industria de la aviación, Aviation Industry CBT Committe (AICC) (AICC 2002) aparece como respuesta natural a las necesidades de una industria que consume una gran cantidad de software educativo para la formación de sus aprendices de piloto.

Las recomendaciones del AICC son publicadas en tres tipos de documentos: recomendaciones y guías AICC, AICC Guidelines and Recommendations (AGRs), informes técnicos y documentos de trabajo. Los AGRs son documentos cortos que representan la postura oficial del AICC en las diferentes áreas que son objeto de estandarización.

Los trabajos del AICC contemplan, entre otros, la definición de requisitos hardware y software para los ordenadores de los alumnos, los periféricos necesarios, los formatos aceptados para los elementos multimedia que componen los cursos, así como recomendaciones para las interfaces de usuario. Otra de sus principales aportaciones es su propuesta para entornos de ejecución. La recomendación del AICC en este sentido

contempla sistemas autónomos en donde la comunicación es realizada a través de ficheros; sistemas de aprendizaje para la Web, con una interfaz definida sobre el protocolo HTTP; y finalmente, un esquema basado en una interfaz de programación que hace transparente el protocolo subyacente.

En este trabajo el AICC ha colaborado estrechamente con la iniciativa ADL del Departamento de Defensa de los EE.UU., y es por eso por lo que realizamos una mención de esta iniciativa en este apartado.

OKI

El proyecto OKI está patrocinado por la Fundación Mellon durante un periodo inicial de dos años. Liderado por el Instituto Tecnológico de Massachussets (Massachussets Institute of Technology (MIT 2002), en colaboración con la Universidad de Stanford, cuenta con la participación de varias universidades americanas, entre ellas la Universidad de Dartmouth, la Universidad de Harvard, la Universidad de Carolina del Norte, la Universidad de Michigan, la Universidad de Pensilvania y la Universidad de Wisconsin.

El objetivo principal del proyecto cubre lo que en estos momentos es una necesidad básica para la comunidad educativa en la Web: diseñar y desarrollar una arquitectura abierta y extensible para los Sistemas de Gestión del Aprendizaje (Learning Management Systems, LMS).

Como resultado del proyecto se pretende elaborar una especificación que sirva como marco para la arquitectura de un LMS y desarrollar un conjunto de herramientas que implementen dicha arquitectura y prueben la viabilidad de la especificación.

La arquitectura OKI se caracteriza por dos capas independientes, compuestas de un conjunto de servicios básicos que se implementan mediante APIs (Application Programming Interface). La primera de estas capas se denomina Servicios Comunes y da soporte al conjunto de servicios básicos o de infraestructura de un LMS (administrativos, de autenticación, de definición de roles de usuario, etc.). En la segunda capa, Servicios Educativos, se define el conjunto de servicios que tienen que ver con la función educativa del LMS. El objetivo de conseguir un conjunto de APIs que no esté ligado a un servicio concreto, es crear una capa de enlace que independice el software educativo de la infraestructura, de tal manera que los cambios en un módulo determinado no afecten al resto. Cada uno de los

servicios está también implementado como un módulo independiente que no comparte con el resto objetos ni interfaces.

Al definir un LMS como un conjunto de servicios básicos implementados mediante APIs independientes, el programador de la aplicación no necesita conocer los detalles particulares de cómo está implementado un determinado servicio. Además es posible realizar varias implementaciones de un mismo servicio sin modificar la aplicación: mientras la implementación de un determinado servicio mantenga su API, las implementaciones de un servicio pueden modificarse sin requerirse ningún cambio en la aplicación que utiliza el API.

Las organizaciones que lideran el desarrollo de especificaciones para la tecnología de enseñanza han firmado un acuerdo de colaboración y coordinación de sus actividades. Esta coalición informal incluye a ADL, OKI, IMS y SIF (Schools Interoperability Framework). Este grupo intenta formalizar sus actividades para conseguir enfocar sus esfuerzos hacia la consecución de un estándar común.

2.4 Ventajas y desventajas de elearning

El beneficio más evidente de la educación virtual reside en que brinda a estudiantes y profesores mucho más tiempo y flexibilidad en términos de plazos y desplazamientos.

Si bien hay plazos estrictos para entregar los trabajos, los estudiantes disponen de mucho más tiempo mientras tengan un teléfono cerca para poder llevar adelante su tarea (esto agrega días o semanas que antes se perdían en traslados) y permite que la educación no sea interrumpida tan fácilmente por viajes o traslados.

Algunas de las ventajas de la formación a distancia desde el aspecto de educación virtual son:

1) Apertura.

- Eliminación o reducción de las barreras de acceso a los cursos o nivel de estudio.
- Diversificación y ampliación de la oferta de cursos.
- Oportunidad de formación adaptada a las exigencias actuales y a las personas que, por una multitud muy diversa de motivos, no pudieron recibir un tipo de enseñanza que ahora sí pueden realizar.

2) Flexibilidad.

- Ausencia de rigidez en cuanto a requisitos de espacio (¿dónde estudiar?).
- Combinación eficaz de estudio y trabajo para muchos alumnos (en especial nuestros alumnos universitarios).
- Ausencia de ruptura del alumno con su ambiente profesional, cultural y familiar.
- Formación fuera del contexto del aula.

3) Eficacia.

- El alumno, centro del proceso de aprendizaje y sujeto activo de su formación ve respetado su propio ritmo de aprendizaje.
- Formación teórico-práctica, relacionada con la propia experiencia del alumno que se encuentra en contacto directo con su actividad profesional que, a su vez, será mejorada.

- Contenidos elaborados por especialistas (que serán, a su vez, tutores) que lo serán también en la utilización de recursos tecnológicos y multimedia.
 - Frecuente comunicación bidireccional que garantiza un aprendizaje dinámico e innovador.
 - No "enseña" únicamente conocimientos, sino que enseña a "aprender a aprender".
- 4) Formación y atención permanente y personal.
- Actividad de alumno. Se favorece el desarrollo de la iniciativa, de actitudes, de intereses, de hábitos educativos en el alumno.
 - Capacitación para el trabajo y superación del nivel cultural de cada alumno.
 - Atención a las demandas, intereses y aspiraciones de diversos grupos por medio de actividades de formación y seguimiento de la misma adaptadas.
- 5) Economía.
- Reducción de costos en relación con el sistema de enseñanza presencial al eliminar pequeños grupos, al evitar gastos de residencia, locomoción, etc. en los alumnos, al evitar el abandono del lugar de trabajo para un tiempo de formación, etc.
 - Los altos costos iniciales son reducidos tras un corto rodaje.

El aprendizaje virtual incrementa la capacidad de pensamiento crítico y las habilidades para resolver problemas prácticos de los estudiantes. Tener que leer y responder a las opiniones de sus compañeros de curso exige a los estudiantes evaluar diferentes puntos de vista sobre un tema. Incluso lidiar con los problemas técnicos de redes y PC's los prepara para un mundo en el que la informática aún está naciendo con dificultad de muchas incertidumbres propias de su estado de disciplina en desarrollo.

El incremento de la interacción personalizada entre docente y alumnos constituye por sí sólo el más grande de sus logros desde el punto de vista del diseño instruccional; es difícil imaginarse cómo podría lograrse semejante interacción entre todos los estudiantes en un aula tradicional.

Permite disminuir el número de los grupos formándolos por afinidades y además es posible incluir fácilmente expertos externos.

Por supuesto, hay muchas desventajas en el hecho de no tener presencia física en una institución, como estar fuera de las reuniones o eventos que requieren interacción personal.

Lo que es más, la interacción con individuos sólo por vía virtual reduce la "banda" de comunicación a un solo "canal" lo que da por resultado relaciones interpersonales menos profundas y completas. Sin embargo, ésta es otra área en la que hace falta investigar más, ya que los impactos psicológicos y las consecuencias sociales de las "relaciones electrónicas" son por ahora ampliamente desconocidas.

También existen inconvenientes que hay que procurar subsanar antes de iniciar las actividades de formación a distancia, tales como:

- Limitaciones para alcanzar el objetivo de socialización, elemento esencial en el proceso de enseñanza/ aprendizaje, debido a las escasas ocasiones que propicia para la interacción (física, cara a cara) de los alumnos con el profesor y entre sí.
- En etapas pre-universitarias, limitación para alcanzar los objetivos del área afectivo-actitudinal, así como los objetivos del área psicomotora a no ser que se desarrollen por medio de actividades presenciales previamente establecidas para el desarrollo de los mismos.
- Cierta empobrecimiento del intercambio directo de experiencias que proporciona la relación educativa profesor-alumno. (Esto se refiere, exclusivamente, al papel del profesor, no a su papel como tutor).
- Posibilidad de cierto retraso o lentitud en el feedback (retroalimentación) y en la rectificación de posibles errores (conceptuales, metodológicos, etc.), aunque puede ser perfectamente suplido por medio de la utilización de los medios tecnológicos.
- Necesidad de una rigurosa planificación tanto a largo como a corto plazo, con las desventajas que esto puede ocasionar.
- El peligro inicial que puede suponer la homogeneización de los materiales.
- La necesidad que implica en el alumno el tener acceso a ciertos medios tecnológicos y a ser competente en su utilización.

En definitiva, la capacitación a distancia permite, entre otras cuestiones, generar en el destinatario la planificación de su propio tiempo, hacer foco en aquellas dificultades idiosincrásicas de su institución, posibilitar el desarrollo de la autonomía profesional, facilitar procesos de

auto evaluación. Pero por otro lado puede suponer falta de socialización del alumno además de que si la plataforma no realiza las actividades necesarias para el aprendizaje entonces se incurre en una posible falta de contexto que provoca en el estudiante una falta de interés ya que sólo recibiría en este caso contenidos y más contenidos sin ninguna ayuda.

Para una buena educación se hace necesario que la plataforma disponga tecnológicamente hablando de una buena base que permita por un lado generar herramientas robustas para la generación y muestra de contenidos (herramientas para el aprendizaje asíncrono) y por otro lado herramientas robustas para la comunicación tanto con el profesor como con los estudiantes (herramientas para el aprendizaje síncrono). Si sólo se pueden ofrecer herramientas para el aprendizaje asíncrono será necesario que vaya acompañado dicho aprendizaje de docencia presencial.

2.5 E-learning en las Universidades Españolas

Veamos actualmente como esta el e-learning en España, aunque sabemos que esto está en continuo cambio.

UNIVERSIDADES ON LINE

- **UOC (Universidad Oberta de Cataluña).**

Su labor en el campus virtual empezó en el curso 1995-1996.

Hoy cuenta con más de 20.000 alumnos virtuales.

- Estudios en castellano. Diplomatura en Ciencias Empresariales, Licenciatura en Psicología, Psicopedagogía, Documentación, Derecho, Ciencias del Trabajo, Administración y Dirección de Empresas, Ingenierías Técnicas en Informática de Gestión y de Sistemas.

- En lengua catalana. Licenciaturas en Psicopedagogía, Humanidades, Filología Catalana, Documentación, Administración y Gestión de Empresas, Derecho, Ciencias Políticas, Ciencias del Trabajo y Psicología; Diplomaturas en Informática y las Ingenierías Técnicas en Informática de Gestión y Sistemas.

Además, oferta títulos propios como el de Graduado Multimedia y cursos de postgrado en castellano en un gran abanico de áreas.

Oferta también cursos de formación abierta, en colaboración con la Universidad Complutense de Madrid.

- **UNED (Universidad Nacional a Distancia)**

Desde el curso 2001-2002 se pueden cursar estudios a través de Internet.

Ellos también están utilizando una versión de dotLRN más antigua para gestionar todos los cursos de e-learning, aunque de momento y tras su personalización no se plantean el cambio a nuevas versiones.

Los alumnos pueden acceder a través de la red a los primeros años de Filologías Hispánica e Inglesa, Historia, Economía, Informática de Sistemas y de Gestión, Psicología, Psicopedagogía, Educación Social, Administración y Gestión de Empresas. Al igual que a cursos de Doctorado.

Otras Universidades

Si bien la oferta de enseñanza on line no se ha generalizado, van incorporándose algunas universidades que ofrecen algunos títulos y cursos de postgrado. Podrían destacarse, entre otras:

- Universidad Carlos III de Madrid. Realiza master de Tributación y un programa de Aula Virtual basado en la enseñanza de asignaturas de libre elección.
- Universidad de la Rioja. Ofrece el segundo ciclo de Historia y Ciencias de la Música y la licenciatura en Ciencias del Trabajo.
- Universidad Politécnica de Madrid, a través del CEPADE (Centro de Estudios de Postgrado de Administración de Empresas) oferta master y cursos de postgrado, así como algunas asignaturas de libre elección.
- Universidad de Barcelona. Cursos de postgrado en Farmacia.
- Universidad de Deusto ha creado una Escuela Virtual que oferta master y cursos relacionados con la empresa.
- Universidad de Valencia oferta cursos de idiomas y formación empresarial. Y en este cuatrimestre ya están dando la docencia aquellos profesores que buenamente y con mucha fe han confiado en nosotros para empezar a utilizar la plataforma dotLRN y ayudarnos a evaluar como funciona.

CAPITULO 3

3. Análisis de requisitos

En este capítulo trataremos de ver las necesidades específicas que tiene la Universidad de Valencia para lanzar el proyecto "Aula Virtual", proyecto dedicado a la creación de un nuevo servicio para la comunidad universitaria ubicado en el ámbito de la teleformación.

Hoy en día son más las Universidades que están invirtiendo en la enseñanza a distancia o la realización de cursos on-line.

Lo que se plantea la Universidad de Valencia es mejorar la calidad de la enseñanza y apoyar la docencia utilizando una herramienta robusta y fiable para no desbancarse en este sector.

No vamos a evaluar plataformas sino que tras la elección de la plataforma dotLRN veremos que cumple con las características necesarias que requiere un proyecto de este tipo y que sea una plataforma con un nivel pedagógico óptimo.

Como avanzamos en el primer capítulo donde indicamos cual serían nuestros objetivos, vimos que hay unos módulos que instalar en la plataforma para que se cumplan esos objetivos, pero también hay que desarrollar algunos módulos para que se puedan llevar a cabo todos los objetivos. Veamos que carencias posee la plataforma por las que se han tenido que desarrollar paquetes específicos.

Una vez tenemos claro cual va a ser la función global que va a desempeñar nuestra aplicación, tenemos que esbozar las características mínimas que se requiere que cumpla nuestro sistema.

Las funciones base que deberá cumplir nuestro sistema son las siguientes:

- Conseguir una plataforma para la gestión de cursos on-line y poder impartir la docencia vía Internet.
- Que gestione foros

- Que gestione noticias
- Que gestione y comparta ficheros
- Que tenga calendario
- Que gestione comunidades
- Que se integre con las bases de datos académicas de la Universidad.

En cuanto a los módulos base seleccionados para implantarse en el “aula virtual” que nos garantizaran que las funciones antes descritas se cumplirán y desempeñará las actividades requeridas para el e-learning, citaremos los siguientes:

File Storage.
Foros.
Calendario.
Faq's.
Noticias.
Homework.
Autenticación
Survey.
Bulk mail.

Estos módulos están integrados al instalar dotLRN, probados y no presentan problemas, por lo que si la instalación es correcta solo hay que activarlos.

3.1 Problemas actuales e identificación de requisitos.

Para poder comprender cuales son los principales requisitos funcionales es imprescindible conocer las carencias de la plataforma.

Al hacer referencia al tema de integrar la plataforma con las bases de datos académicas de la Universidad ninguna plataforma GPL dispone de ningún procedimiento genérico para la carga de estos datos y es por ello que se han tenido que desarrollar:

- Modulo para la carga de usuarios, tanto profesores como alumnos, en la plataforma.

- Módulo para la carga de asignaturas, centros, cursos y matricular a los alumnos en sus asignaturas, generando así los cursos.
- Módulo para visualizar información pública de los centros y asignaturas.

A la vista de los problemas actuales y en conjunto con los objetivos marcados ya podemos afirmar que el objetivo global será implantar la plataforma en la Universidad de Valencia e integrarla con toda la información académica que esta dispone, teniendo en cuenta algunos puntos:

1. Mantenimiento informático actualizado de la base de datos de usuarios, asignaturas y centros.
2. Diferenciación del sistema entre profesores, alumnos y personal administrativo.
3. Disponibilidad de los módulos antes descritos que doten al sistema para realizar el e-learning.

CAPITULO 4

4. Planificación y costes

Es este capítulo trataremos de centrar la estimación del esfuerzo que tendremos que realizar para desarrollar la implantación. Por esfuerzo nos referimos a la cantidad de recursos humanos, usualmente medido en meses/hombre. Nos basamos en la especificación de requisitos antes descrita.

Una vez tengamos el esfuerzo, meses/hombre, hay que asignar estos esfuerzos a tareas y persona, dado que no suele cobrar lo mismo el analista que el programador, el que tiene experiencia y el que no, etc. Lo razonable es identificar a grandes rasgos las diferentes componentes del proceso de implantación de la plataforma a modo que a cada una se puede asignar determinado personal.

Una vez conocidas las tareas a realizar se deberá programar (planificar), el proceso de desarrollo.

4.1 Definir los propósitos

Estudiar e Implantar en la Universidad una plataforma que cubra todas las necesidades para la realización de e-learning en el ámbito académico.

Objetivos

- Recoger información a cerca de todo lo que conlleva el elearning.
- Documentar el estado actual de la Universidad en este campo.
- Conocer la plataforma dotLRN y familiarizarse con ella para conocer su sistema.
- Evaluar los requisitos mínimos de la Universidad para llevar a cabo el trabajo.
- Implantar esta plataforma con éxito.
- Realizar las pruebas oportunas.

Para cumplir estos objetivos debemos de hacer una planificación, por lo que haremos una división del trabajo para realizar estos objetivos.

4.2 División del trabajo

Los objetivos antes descritos los hemos dividido de la siguiente manera:

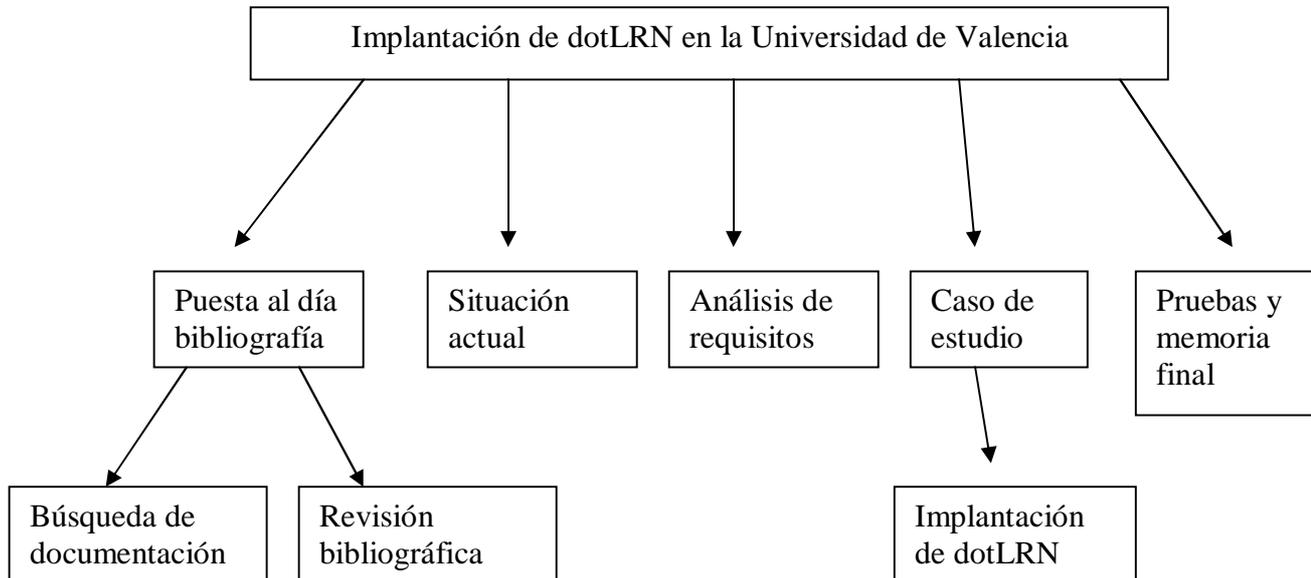


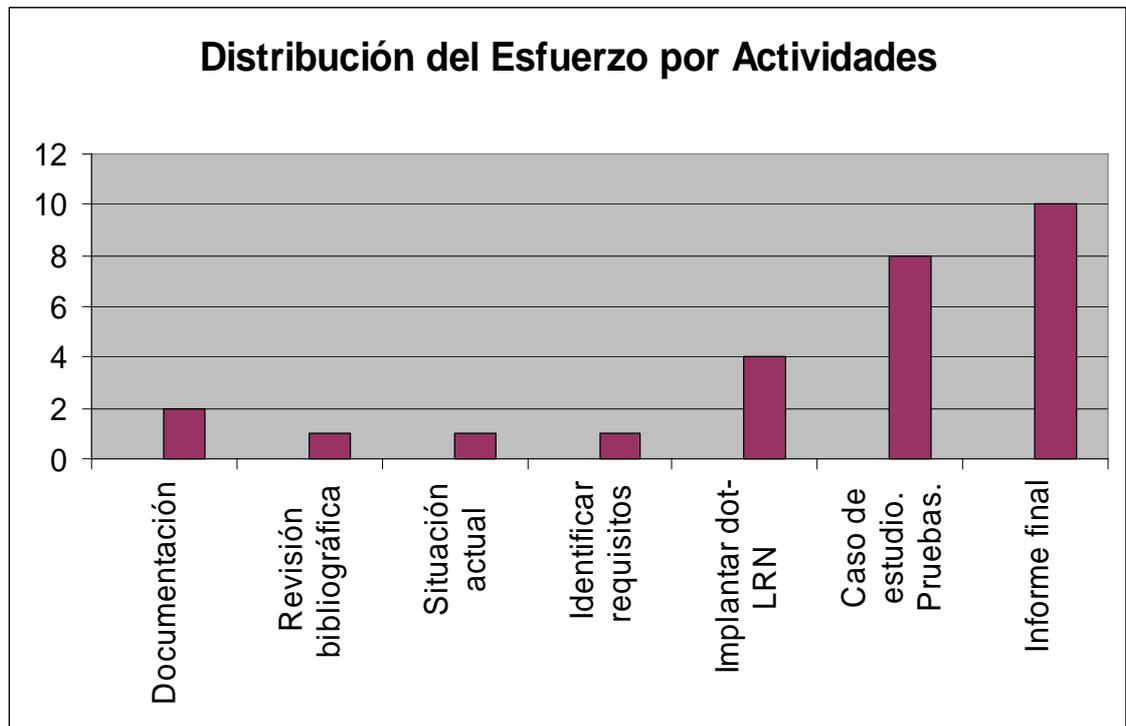
Ilustración 5. División del trabajo.

La descomposición del trabajo tiene en cuenta la búsqueda y revisión bibliográfica habituales en cualquier proyecto. A continuación describiremos la situación actual de la Universidad en este ámbito, una vez conocida la situación actual deberemos analizar esa situación y ver cuales son los requerimientos que tendrá la Universidad para desarrollar la implantación. El caso de estudio será una tarea un tanto peculiar porque se puede empezar antes de donde esta situado ya que para adquirir experiencia con la herramienta lo mejor es cuanto antes se pueda probar cosas con ella mejor. Por ultimo como es habitual en los proyectos finales de carrera procederemos a la realización de la memoria.

4.3 Estimación de tiempos

En este momento debemos intentar acoplar los tiempos a las tareas antes vistas. Para la realización del proyecto se han dispuesto de 5 meses y la cantidad de personal es la siguiente: (1 analista, 2 programadores).

Actividad	Duración Estimada
Documentación	2 semanas
Revisión bibliográfica	1 semanas
Situación actual	1 semanas
Identificar requisitos	1 semanas
Implantar dot-LRN	4 semanas
Caso de estudio. Pruebas	8 semanas
Informe final	10 semanas
Esfuerzo total	semanas (aprox. 5'5 meses)



4.4 Identificación de hitos

Los objetivos intermedios que hemos detectado para la consecución de la meta final, son tres: la finalización de la documentación necesaria para poder comenzar nuestro trabajo (hito 1; H1), analizar y conocer la situación actual de la universidad (hito 2; H2) y consecución del objetivo final (hito 3; H3). Hemos seleccionado como hitos las tareas más críticas para la consecución de nuestro proyecto.

4.5 Encadenamiento de actividades

Para conocer el orden adecuado en el que hay que realizar las tareas utilizaremos un diagrama de actividades del tipo PERT. Este diagrama estará compuesto por los hitos detectados posicionados en el orden correcto en forma de óvalos, llamados H1, H2 y H3. Además se han añadido fechas y cifras a cada nodo, que indican la fecha de comienzo y las semanas de duración. Asumimos que un mes tiene cuatro semanas, solo contamos los días laborables, y que existe la posibilidad del solapamiento de las tareas.

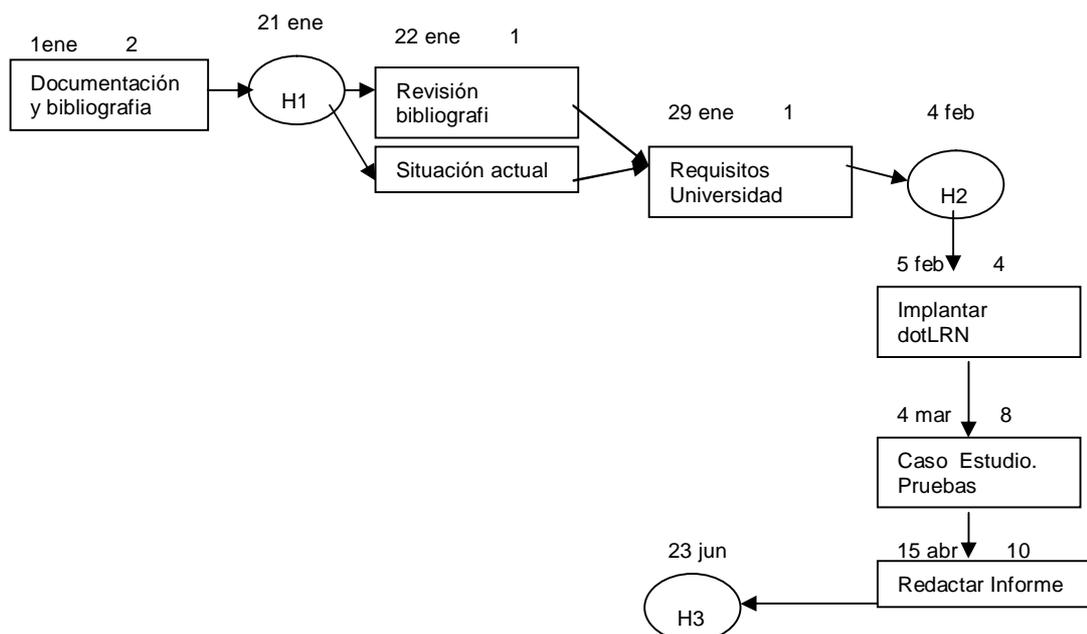


Ilustración 6. Diagrama de Pert.

Exceptuando de la tarea Revisión bibliográfica, las restantes siguen un orden secuencial, por tanto el camino crítico está compuesto por la consecución de estas tareas.

Una de las tareas más complejas es la implantación de la plataforma, esta tarea requiere un gran esfuerzo en cuanto a que todo este a punto para poder adquirir experiencia con la plataforma.

4.6 Planificación temporal

El diagrama de Gantt nos indica la planificación del proyecto representando la duración y la secuencialidad según el diagrama de actividades, además de las que realicemos a la vez o en paralelo.

En la ilustración 7 representa el calendario del proyecto, cada actividad queda representada con una barra horizontal y la longitud de cada nodo se usa para representar la duración de cada actividad, los hitos quedan identificados mediante rombos, estos hitos nos permiten evaluar la situación del proyecto y nos hacen tener preparadas las tareas anteriores porque sino no podemos avanzar en la elaboración del proyecto.

La única actividad que no cae dentro del camino crítico es la "Revisión bibliográfica" que se puede retrasar una semana, aquí tenemos mayor libertad de finalización, sin alterar la planificación prevista, situación contraria de la que ocurre con las actividades situadas en el camino crítico, que no podemos retrasarlas. Esta holgura es debida a que hasta que no empecemos con la actividad de "Implantar dotLRN" no hace falta que esté acabada.

El proyecto tiene una duración aproximada de 5'5 meses. El horario de trabajo que se ha planeado es de 8 horas diarias, de lunes a viernes, unas 40 horas semanales. Aunque la disponibilidad real no ha sido esta ya que al estar trabajando hay semanas que no he dedicado tanto esfuerzo al proyecto.

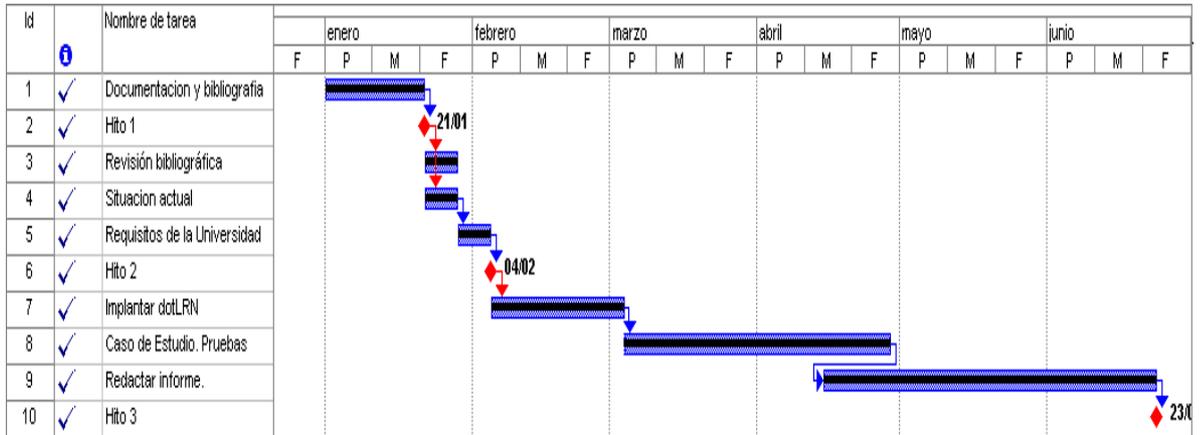


Ilustración 7. Diagrama de Gantt

4.7 Coste de la implantación de dotLRN

Para calcular el coste de implantación nos vamos a basar en el coste de recursos humanos y en el hardware necesario para soportar dicha instalación.

Hay que tener en cuenta que hay que generar módulos para adaptar la aplicación a la Universidad de Valencia. Resumiendo las tareas más críticas y básicas a realizar podemos destacar.

1. Personalización de la plataforma según la apariencia utilizada en otros servicios de la Universidad de Valencia.
2. Generación de scripts para la inserción automática de usuarios.
3. Generación de scripts para la inserción automática de asignaturas.
4. Generación de scripts para la inserción automática de los centros.
5. Generación de scripts para asociar los alumnos con las asignaturas matriculadas, así como los profesores responsables de las asignaturas.
6. Adaptar la plataforma a las necesidades descritas en el análisis de requisitos.
7. Instalar y configurar los servidores tanto de bases de datos, postgres, como los servidores web, AOLServer, y la maquina donde instalar la plataforma dotLRN.
8. Mantenimiento de la aplicación y corrección de errores, que no han sido pocos.
9. Gestión de los problemas cotidianos de los usuarios.
10. Atender los e-mails de los usuarios y contestar sus dudas.

Para todo esto sería necesario el trabajo de un analista, dos programadores, y un operador a tiempo parcial sólo dos días a la semana, el analista y los programadores se encargarían del mantenimiento de los servidores y aplicación. El operador se encargara de ejecutar los scripts rutinarios y atender las preguntas de los usuarios.

Para los cálculos de coste se han tenido en cuenta que un Analista cobra unos 24.600 €, un programador cobra unos 21.600 € y un Operador cobra unos 18.430 €.

El coste de recursos humanos como solo trabajan durante 5'5 meses serían los siguientes:

Recursos humanos	Coste anual	Coste 5 '5 meses
Analista . Full	24600	12300 €
2 Programador Full	21600	21600 €
Operador. 2 días	18430	1800 €
Total		35710,00 €

En principio se han utilizado 3 servidores más un balanceador de carga, las maquinas que se utilizaran como servidor, incluyendo sus componentes nos llevará a los siguientes costes:

1. Un servidor será un IBM Eserver 325 con procesador Opteron dual 2 Ghz y 5 Gb de Ram, sistema de discos Fiber Chanel. Aunque no seria necesario que fuese IBM por el tema de pagar la marca, aquí se opta por montar este servidor. Coste aprox. 6.000 €
2. 2 servidores clónicos PC Pentium IV con 2 módulos de 512 MB RAM DDR 400, discos de 80 GB DD 7200 rpm, Coste aprox. 3.000 €

COSTE TOTAL DEL PROYECTO:

Coste Personal	35710 €
Coste Hardware	9000 €
TOTAL	44710,00 €

CAPITULO 5

5. DESCRIPCION DEL SISTEMA

En este capítulo describiremos el sistema que vamos a implantar de manera que podamos entender como funciona, intentaremos describir las dos arquitecturas que comprenden esta herramienta de e-learning. Primero OpenACS y después dotLRN.

5.1 Cómo surge OpenACS

Las herramientas para crear comunidades virtuales y compartir ideas en Internet obligatoriamente deben ser capaces de gestionar contenidos dinámicos. Esto se hace sin grandes dificultades usando una base de datos relacional y un lenguaje de programación. Sin embargo, repetir este trabajo para cada nuevo proyecto no es razonable, ni para el desarrollador por la carga de trabajo que implica, ni para el cliente por su alto coste.

A mediados de los 90, Philip Greenspun, autor del libro "*Philip and Alex's Guide to Web Publishing*", creó un conjunto de herramientas modulares que intentaban dar una respuesta genérica a la mayoría de las necesidades de una comunidad virtual en Internet. El software que Philip escribió constituyó la base de su tesis doctoral en el MIT, y también le permitió fundar la empresa Arsdigita. El conjunto de estas herramientas se denominó Arsdigita Community System (ACS) y se liberaron bajo la licencia GNU General Public Licence (GPL). La base de datos relacional que Philip eligió por sus prestaciones fue Oracle y el lenguaje de programación TCL.

ACS ha demostrado ser una herramienta muy potente y sencilla, capaz de dar solución a proyectos de gran envergadura (son ejemplos el sistema Global Development Gateway Web del Banco Mundial, MIT escuela de negocios en Sloan y la intranet del departamento de ventas de Siemens que engloba 10.000 empleados en 35 países). Sin embargo, tras varios años de éxitos sucesivos, Arsdigita desapareció principalmente por problemas de gestión ajenos a los fundadores, que fueron obligados a abandonar la empresa. Finalmente, Red Hat contrató a algunos de los programadores de Arsdigita y adquirió los activos que quedaron tras el cierre de Arsdigita.

Este hubiese sido el final de la historia si Arsdigita hubiese sido una empresa desarrolladora de software cerrado y propietario. Pero puesto que ACS se liberó bajo la licencia GNU General Public Licence (GPL), la comunidad de desarrolladores retomó el proyecto donde Arsdigita lo dejó.

A partir de este punto es donde toma protagonismo una versión de ACS basada en el trabajo de Arsdigita conocida como OpenACS, basada en la base de datos de código abierto PostgreSQL.

Hoy en día es un proyecto maduro, con más de 7000 miembros y 10 compañías que proporcionan apoyo comercial y desarrollo. La versión más reciente, OACS 5.0.2 se ha lanzado para primeros del 2004. La arquitectura no es dependiente en cualquier RDBMS particular, y ya opera con Oracle o Postgres, con la habilidad de extender el funcionamiento a otras bases de datos, si es necesario.

5.2 Que es una aplicación Frameworks

Una aplicación frameworks [2] puede definirse como una aplicación reusable, "semi-completa" que puede utilizarse para producir aplicaciones personalizadas. Una aplicación framework proporciona diferentes beneficios a los diseñadores de la aplicación como modularidad, reusabilidad, extensibilidad. Un frameworks puede tener un enfoque general, o utilizarse para resolver pequeños cambios específicos en un dominio particular. Utilizaremos el termino "web application framework" para describir éstas arquitecturas específicamente diseñadas para satisfacer las necesidades particulares de un dominio para una aplicación web.

OACS entra en esta última categoría de aplicaciones frameworks. OACS permite a los diseñadores del website hacer aplicaciones web modulares y rápidamente extensibles, y a la vez logra niveles muy altos de software reusable, utilizando las características de los paquetes del core de OACS. Fayad et al. propone un conjunto de criterios [4] para evaluar empresas orientadas a objetos que están utilizando frameworks y evaluando el éxito de OACS como un "web application framework".

Los criterios más relevantes son:

- Madurez de la funcionalidad.
- Aporta extensibilidad.

- Catalogo de objetos de negocio (BOs) y enduring business themes (EBTs)
- Estabilidad del software
- Soporte a objetos distribuidos y escalabilidad.
- La plataforma busca independencia o portabilidad.
- Capacidad de integración con otras plataformas.
- Documentación del core.
- Rapidez del web.
- Una base económica acorde con la elección de la plataforma.

5.3 La arquitectura de OpenACS

El framework de OACS esta principalmente programado en Tcl como un conjunto de aplicaciones que actúan entrelazadamente con AOLserver y un RDBMS (Oracle o Postgres), puede estar dividido en varios paquetes del core, servicios y aplicaciones. Los paquetes del core son aquéllos que se requieren para que OACS funcione. Estos incluyen un Kernel, gestión de paquetes, utilidades para la administración de los usuarios, el sitemap, y otros objetos comunes, un sistema de plantillas que separa claramente la lógica de negocio de la lógica de presentación, y un sistema de mensajería y correo backend. Los servicios como el paquete de eventos son aplicaciones de lógica reusable y no son ninguna aplicación dependiente y no requieren un interfaz del usuario. Los paquetes que se denominan aplicación tienen un interfaz de usuario y su código esta claramente dividido en la lógica de aplicación y la lógica de presentación.

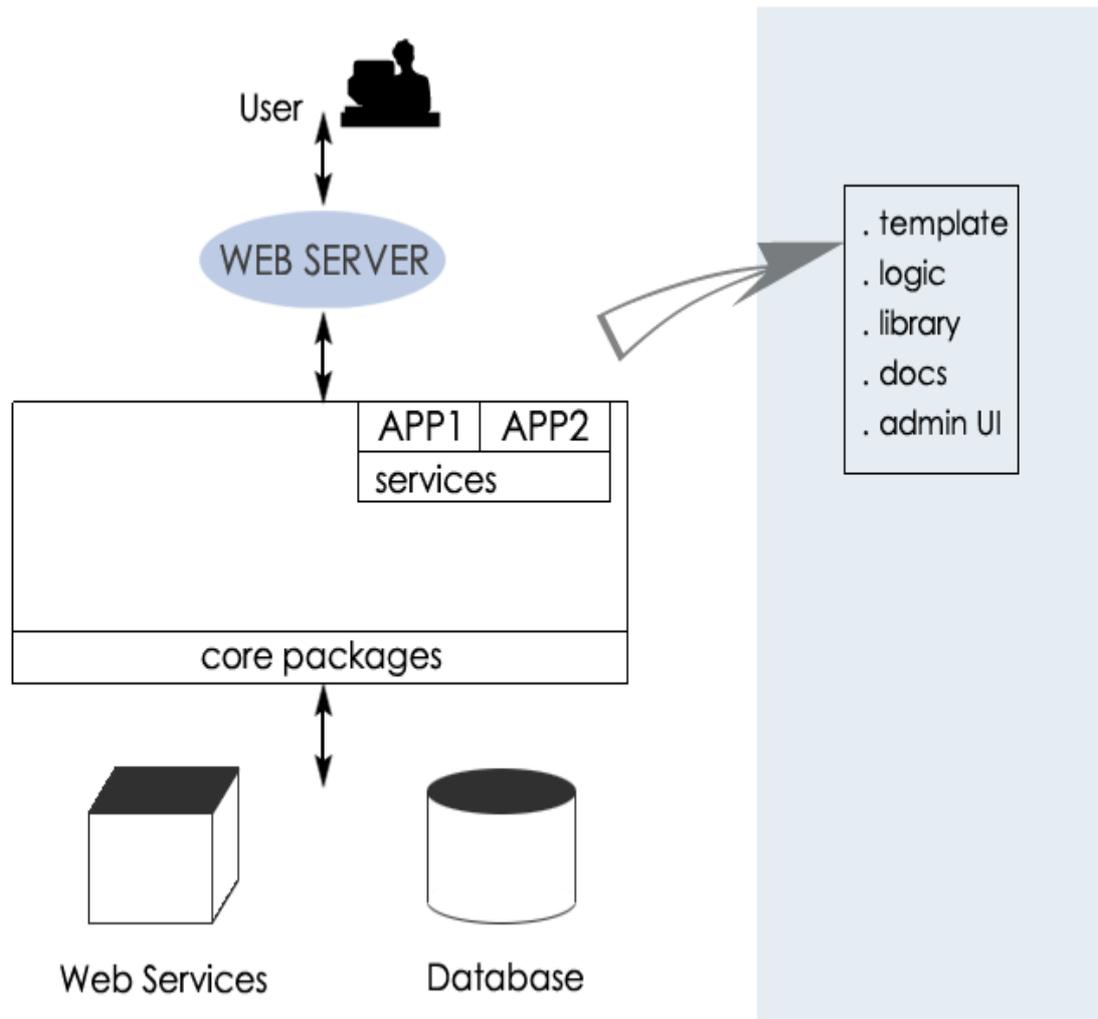


Ilustración 8. Arquitectura OpenACS .

Cada servicio y aplicación requieren un modelo de datos que se integrará en el framework. La integración de estos modelos de datos permiten la reusabilidad y diferentes funcionalidades. Para aumentar la reusabilidad el framework tiene una arquitectura orientada a objetos pero, ya que el RDBMS no implementa las funcionalidades de programación OO, estos deben ser añadidos ad-hoc por el armazón.

Los objetos de OACS tienen sus atributos guardados en tablas y un juego de métodos definidos como los paquetes de PL/SQL. Estos paquetes contienen los procedimientos que proporcionan la interface para el modelo de datos.

La idea que hay detrás de la arquitectura de OO de OACS es que cada pedazo de información que puede ser reusable debe ser un objeto. Como los RDBMS no son inherentemente OO, las estructuras de datos se integran

en el framework creando una tabla llamada "acs_objects" que guarda un registro de cada objeto de OACS. Otra tabla llamada "acs_object", define los atributos generales guardados en cada objeto, incluso un único ID, un tipo del objeto. El concepto de los tipos de objetos en OACS es equivalente a las clases en OO de los lenguajes de programación, desafortunadamente ya que se utiliza un RDBMS el trabajo adicional hay que realizarlo.

Aunque la reusabilidad es la meta principal, gracias a este diseño, el sistema de permisos puede ayudar al diseñador a definir que se permite hacer a los usuarios, esto esta definido en las filas de la tabla para cada aplicación. Cada objeto tiene un atributo llamado "context_id" que especifica los permisos predefinidos para un objeto y su alcance.

Se definen los tipos del objeto con atributos definidos en el acs_object_types y acs_attributes. Los tipos del objeto en OACS pueden heredar los atributos de un tipo del padre, por lo tanto los tipos del sistema son jerárquicos. En conclusión el modelo de datos en OACS se lleva a cabo en un RDBMS y esta herencia necesita ser realizada explícitamente.

Para utilizar el sistema de objetos del OACS se requieren cuatro pasos, todos de los cuales se llevan a cabo con el modelo de los datos de aplicación:

- 1 crear el tipo del objeto, la mayoría será subtipo de acs_object
- 2 crear los atributos de objeto
- 3 crear los métodos de objeto
- 4 crear la tabla para almacenar los datos de objeto.

Una aplicación de OACS también requerirá la lógica de negocio y la lógica de presentación. OACS 4.5b utiliza un sistema de plantilla para separar estos dos claramente en diferentes conjuntos de archivos. Utiliza un subdirectorio estándar (el www), dónde cada usuario que visualiza una pagina se almacena en por lo menos dos archivos distintos. Los archivos .tcl tienen los tcl y queries en SQL que actúan como las fuentes de datos, así como los emparejamientos de name/value. Los ficheros .adp devuelven estos valores en HTML usando un juego de etiquetas especiales y substituciones de esas fuente de datos.

Muchas aplicaciones requerirán lógica de administración e UI, estos se separarán del resto en otro directorio (el admin). Una aplicación de OACS

también requiere documentación almacenada en un tercer directorio distinto (el doc). El HTML archivado allí se manejará por un sistema de la documentación centralizado.

Finalmente, un paquete completo puede tratarse como otro objeto en el OACS, esto significa que pueden instalarse distintas instancias del mismo paquete en el mismo website. Por ejemplo, el paquete calendario puede ser instanciado por comunidades diferentes en un website:

/ el community1/calendar, / el community2/calendar,... / el communityN/calendar. Los website podrían estar usando N calendarios diferentes (por ejemplo uno para cada comunidad), esto tiene datos separados pero una sola base de código.

5.3.2 Proceso de las peticiones

El propósito de una aplicación web es procesar las peticiones de los usuarios dinámicamente en un servidor de web. Ya que estas funcionalidades son un punto caliente para el sistema, estas se integran en un Procesador de Peticiones (RP) el cual maneja cada una de las peticiones. La ilustración 9, muestra un diagrama simplificado de esta sucesión y de cómo el Procesador de Peticiones de OACS maneja las peticiones de una página.

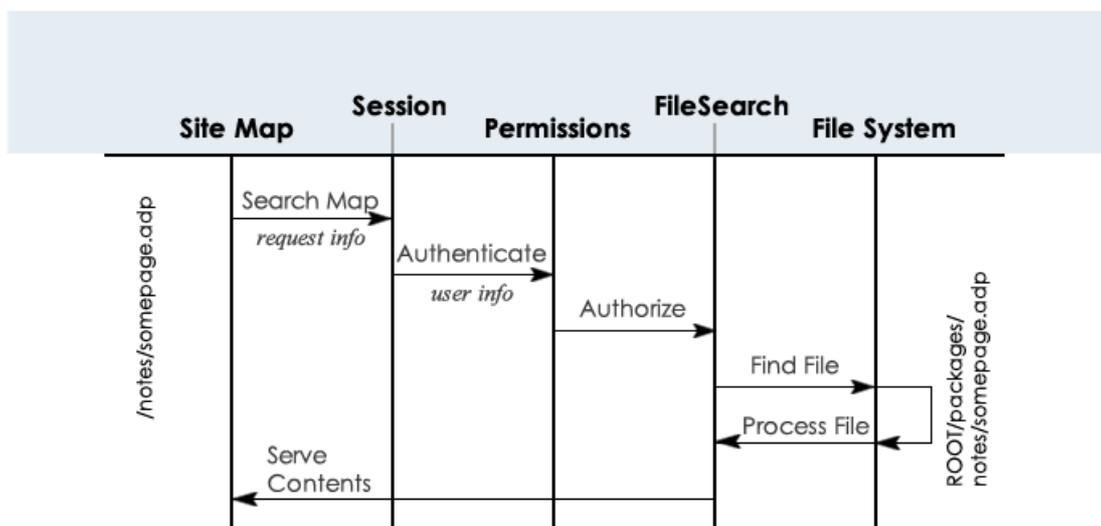


Ilustración 9. Proceso de una petición.

Mapa del sitio. El RP mapea la URL a un archivo físico. Esto se hace buscando sobre los datos del sitio y se mapean las URLs en instancias de la aplicación.

La autenticación. El RP examina información de la sesión enviada por el navegador del cliente utilizando cookies. El RP prepara nuevos tokens de sesión si es necesario.

La autorización. Una vez el usuario se ha autenticado, el RP verifica si el usuario tiene el permiso de acceso para el file/object pedido. El control de acceso es gestionado por el sistema de permisos.

Procesar la URL. El RP busca el archivo a ser servido. Dependiendo de la extensión, el archivo podría servirse directamente (.html) o enviarlo al sistema de plantilla (.adp) o al intérprete del tcl (tcl).

5.3.3 El sistema de permisos

El sistema de permisos de OpenACS 5.0x permite a diseñadores y a administradores poner políticas de control de accesos a nivel del objeto, es decir, cualquier aplicación u objeto del sistema representados por una fila en el la tabla del acs_objects puede controlarse vía un simple PL/SQL o interfaz de Tcl. El sistema de permisos maneja un modelo de datos que permite mediante scripts verificar permisos usando una simple llamada al API.

Aunque todo esto puede sonar fácil y maravilloso, ningún diseñador o administrador querrían especificar explícitamente los derechos de control de acceso para cada usuario y cada objeto en un sitio. Por consiguiente, OpenACS 5.0.0b1 tiene dos mecanismos auxiliares para hacer esto más fácil: Primero, el sistema de Grupos les permite a los usuarios agruparse de maneras flexibles. Segundo, el modelo del objeto define una noción de "*contexto del objeto*" que permite a las aplicaciones agrupar objetos en dominios de seguridad más grandes. Vamos a intentar aclarar esto.

Grupos

En OpenACS 3.x, el sistema de grupos permitió a diseñadores y a administradores definir agrupaciones simples de usuarios. Cada grupo tenía un nombre "human readable" y único ID, y había una sola tabla para mapear los usuarios a los grupos.

El sistema de grupos 3.x, aunque muy útil, estaba limitado. La limitación principal: los grupos no podían contener o incluir otros grupos. No podías expresar el hecho que todos los miembros del grupo A también estaban incluidos en el grupo B - por ejemplo, para modelar una compañía con múltiples oficinas. La compañía debería tener un grupo principal que representa a todos los empleados, y cada oficina debe tener un grupo representando sus empleados. Obviamente, se querría que cada miembro de un grupo de empleado de oficina fuera automáticamente miembro del grupo de empleado de compañía.

En OpenACS 3.x, no podía expresar tampoco el hecho que un miembro del grupo A sea del grupo B, sin implicar que todos sus miembros también sean miembros de B. Este tipo de relación se da en el mundo real, aunque no muy a menudo. Un ejemplo bueno es una organización como Greenpeace que puede tener como miembros individuos y organizaciones: aunque el Club de la Sierra puede tener miembros de una organización de Greenpeace, todos sus miembros necesariamente no son miembros de Greenpeace.

OpenACS 5.0.0b1 resuelve estos problemas planteados introduciendo una nueva abstracción llamada, "party". Las parties tienen una definición recursiva, y podemos ilustrar cómo funciona con la siguiente simplificación del modelo de datos. Primero, define la tabla "parties" dónde cada party tiene una dirección del correo electrónico y un URL para la información de contacto.

```
create table parties (  
    party_id integer not null references acs_objects(object_id),  
    email varchar(100),  
    url varchar(100)  
)
```

Ahora se definen dos subtipos de party, uno para las personas, y uno para los grupos:

```
create table groups (  
    group_id not null references parties(party_id),  
    group_name varchar(100) not null  
)  
  
create table persons (  
    person_id not null references parties(party_id),  
    first_names varchar(100) not null,
```

```
last_name varchar(100) not null  
l)
```

La tabla de usuarios también se define en este modelo de datos como un subtipo de la tabla persona. Contiene muchas de las columnas básicas que la tabla de usuarios del viejo OpenACS 3.x contuvo.

Finalmente, se definen dos relaciones, una para el número de miembros del grupo y otra para la composición de grupo. La relación de la composición nos permite expresar el hecho de que cada miembro de grupo A también debe ser un miembro del grupo B. Esta relación nos permite definir una jerarquía de grupos en lugar de los grupos planos simples que en el 3.x existían.

La relación entre usuarios es parecida a la que había en 3.x, sólo que mapea los grupos en las parties en lugar de los grupos a los usuarios. Lo que esto significa es que cada miembro de un grupo es una party en lugar de simplemente un usuario. Es decir, los grupos consisten en miembros que son una persona o un grupo entero. Esto nos permite decir que un grupo A debería ser un miembro del grupo B.

El modelo de datos de los grupos es recursivo. El hecho que se mapeen las parties como una persona o un grupo tiene mucho poder, permitiéndonos modelar agrupaciones jerárquicas complejas de personas y grupos que eran difícil o imposible planear en 3.x.

Los detalles completos del modelo de datos de grupos están más allá del alcance de esta introducción - yo le he comentado simplemente para entender como trabajan los permisos. Para un completo detalle, se puede mirar las parties en OpenACS 4 u OpenACS 4 Diseño de los Grupos.

Permisos

El modelo de datos de permisos es bastante simple. El modelo de datos es un mapeo entre privilegios, parties y objetos. Antes vimos como se implementaban la parties y ahora vamos a explicar como se implementan los privilegios.

En OpenACS 5.0.0b1, el modelo da derecho para realizar alguna operación en algún objeto. Son unidades básicas para implementar las

políticas de control de acceso. Por ejemplo, en el filesystem de Unix llevamos a cabo el control de acceso concediéndoles alguna combinación de privilegios a los usuarios para que lea, escriba o ejecute los archivos y directorios. En OpenACS 5.0.0b1, la tabla de privilegios esta organizada jerárquicamente para que los diseñadores puedan definir conjuntos de privilegios y agregarlos juntos. Por ejemplo, si tenemos privilegios de lectura, escritura, crear y borrar, sería conveniente combinarlos en un nuevo privilegio llamado "admin". Entonces si nosotros le concedemos este privilegio a un usuario se concede automáticamente todos los privilegios por herencia que contiene el usuario. El OpenACS 5.0.0b1 el kernel realmente modela y define estos privilegios como sigue:

```
begin
    acs_privilege.create_privilege('read');
    acs_privilege.create_privilege('write');
    acs_privilege.create_privilege('create');
    acs_privilege.create_privilege('delete');
    acs_privilege.create_privilege('admin');
    acs_privilege.add_child('admin', 'read');
    acs_privilege.add_child('admin', 'write');
    acs_privilege.add_child('admin', 'create');
    acs_privilege.add_child('admin', 'delete');
    commit;
end;
```

Para dar permiso a un usuario para realizar una operación sobre un objeto particular se debe llamar a `acs_permission.grant_permission` así:

```
acs_permission.grant_permission (
    object_id => some_object_id,
    grantee_id => some_party_id,
    privilege => 'some_privilege_name'
);
```

Usando simplemente estos mecanismos es suficiente para diseñadores y administradores definir el control de acceso eficazmente para cada objeto en el sistema. Pero sería tedioso hacer esto individualmente a cada objeto: en muchos casos, o algunas veces preferiremos controlar los permisos sobre grupos grandes de objetos en el sitio. Se utilizarán los contextos para lograr esta fin.

Contexto de Objetos (Object Context)

En OpenACS 5.0.0b1, un contexto de objeto es una generalización del mecanismo del scoping introducida en OpenACS 3.x. "Scoping" y "scope" (alcance) estos términos están mejor explicados en el ejemplo: veamos algunas filas hipotéticas en la table del address_book:

...	Scope	User_id	Group_id	...
...	User	123		...
...	group		456	
...	public			

Ilustración 10. Contexto del objeto libro de direcciones.

La primera fila representa una entrada en la libreta de direcciones personal del Usuario 123, la segunda fila representa una entrada en la libreta de direcciones compartida del Grupo del Usuario 456, y la tercera fila representa una entrada en la libreta de direcciones pública del sitio.

De esta manera, las columnas del scoping identifican el contexto de seguridad al que un objeto dado pertenece, dónde cada contexto o es una persona o un grupo de las personas o es público general.

En OpenACS 5.0.0b1, en lugar de irrumpir el mundo en un conjunto limitado de scopes, cada objeto vive en un solo contexto. Un contexto simplemente es otro objeto que representa el dominio de seguridad al que el objeto pertenece. Por consiguiente, si un objeto A no tiene ningún permiso explícitamente, entonces el sistema mirará la columna del "context_id" en el "acs_objects" y verificará los permisos del objeto en el contexto de allí. Dos cosas controlan el alcance de esta búsqueda: la estructura de la propia jerarquía del contexto, y el valor del flag del "security_inherit_p" en cada objeto. Si este flag se pone a 't', entonces la búsqueda automática a través del contexto se pasa, no se hace. Si ponemos este campo a 'f' podemos sobreponer los permisos predefinidos en un subtree de algún contexto.

Un ejemplo bueno de cómo usar esta jerarquía se ve en la aplicación de foros. Con sólo los permisos heredados por la jerarquía no es obvio cómo

inicializar razonablemente la lista de control de acceso al crear un mensaje. Es posible, que tengamos que conceder explícitamente privilegios de lectura y escritura siempre que creemos un mensaje, esto es impensable. En OpenACS 5.0.0b1, una cosa razonable es crear un objeto que representa un foro, y apunta el campo del `context_id` del nuevo mensaje del foro. Entonces, concedemos a cada usuario el permiso de lectura en este foro. Por defecto, tendrán el acceso de lectura automáticamente en el nuevo mensaje que insertemos, el sistema automáticamente inspecciona los permisos en el contexto del mensaje. Para permitirle al creador del mensaje cambiar los permisos del mensaje después de que se ha enviado, le concedemos acceso de escritura al usuario sobre el mensaje, y lo hacemos.

Este mecanismo permite a diseñadores y a administradores definir una jerarquía que consigue la estructura que necesitan para el control de acceso en su aplicación. El cuadro siguiente muestra una jerarquía de contexto típico en un sitio cualquiera:

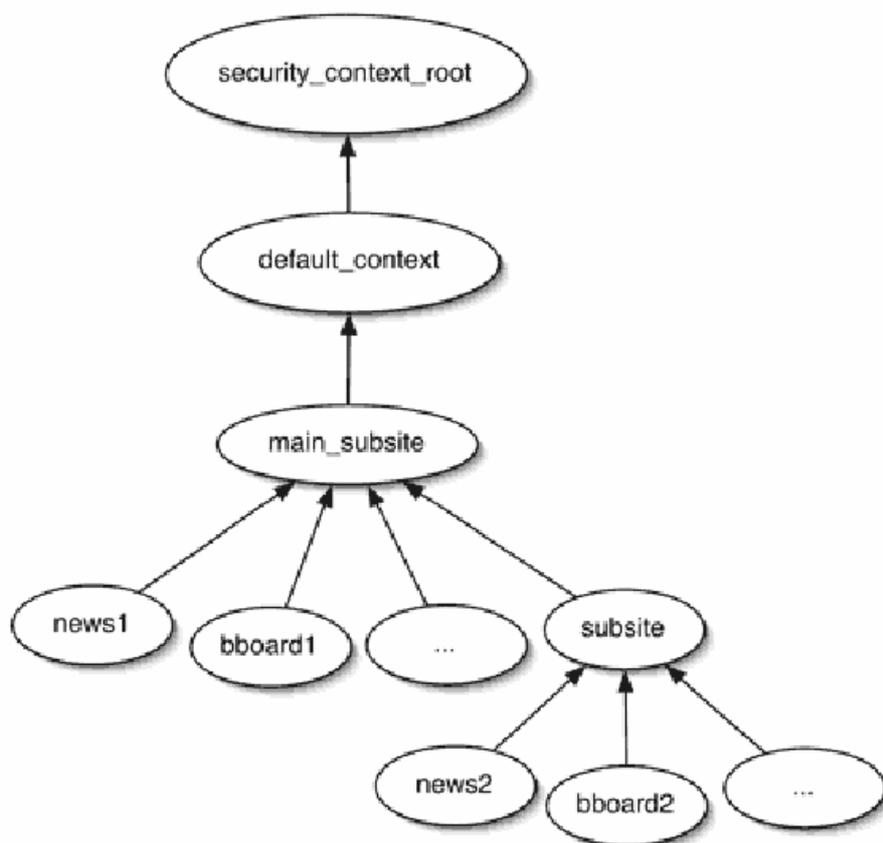


Ilustración 11. Estructura jerárquica de permisos.

Resaltemos algunas cosas. Primero, los dos contextos superiores de la imagen son "mágicos" en algún sentido ya que son creados por defecto

por OpenACS para un propósito específico. El "default_context" representa la raíz de la jerarquía del contexto para el sitio entero. Todos los permisos buscan el camino en el árbol hasta ese punto y entonces se paran.

Si concedemos los permisos en este objeto, entonces por defecto esos permisos se sobrepone para cada objeto en el sistema sin tener en cuenta el subsite del que ellos cuelgan. El "security_context_root" tiene un papel ligeramente diferente. Si algún objeto no tiene ningún permiso atado a él, y su valor para el "security_inherit_p" es 'f', o el "context_id" es nulo, entonces se utiliza este contexto por defecto.

Ejemplo

El paquete se llama "notas". Se puede chequear desde el almacén del repositorio CVS, para hacer esto leer estas instrucciones [<http://acs40.arsdigita.com/acs40-proyecto-central/client-build.html>] cómo usar las cajas anónimas y entonces la caja el módulo acs-packages/notes.

```
% export CVSROOT=:pserver:anonymous@cvs.arsdigita.com:/usr/local/cvsroot
% cvs login # the password is acsrules
% cvs checkout acs-packages/notes
```

Después de haber descargado el paquete, mirar la página de index.tcl en el directorio del www. Mirar también el código en su navegador [<http://cvs.arsdigita.com/cgi-bin/cvsweb.pl/acs-packages/notes/www/index.tcl?rev=1.3&content-type=text/x-cvsweb-markup>]. El código debe parecerse a esto:

```
# main index page for notes.
ad_page_contract {
    @author you
    @cvs-id $Id: permissions.xml,v 1.7.2.1 2003/11/17 16:38:09 jeffd
Exp $} -properties {
    notes:multirow
    context_bar:onevalue
    create_p:onevalue
}
set package_id [ad_conn package_id]
set user_id [ad_conn user_id]

set context_bar [ad_context_bar]
set create_p [ad_permission_p $package_id create]

db_multirow notes notes {
    select note_id, owner_id, title, body,
```

```

        decode(acs_permission.permission_p(note_id,
            :user_id,
            'write'),
            't', 1,
            'f', 0) as write_p,
        decode(acs_permission.permission_p(note_id,
            :user_id,
            'admin'),
            't', 1,
            'f', 0) as admin_p,
        decode(acs_permission.permission_p(note_id,
            :user_id,
            'delete'),
            't', 1,
            'f', 0) as delete_p

    from notes n, acs_objects o
    where n.note_id = o.object_id
    and o.context_id = :package_id
    and acs_permission.permission_p(note_id, :user_id, 'read') = 't'
    order by creation_date
}
ad_return_template

```

Este ejemplo muestra los Tcl y el API PL/SQL para verificar los permisos. El Tcl proc `ad_permission_p` y los PL/SQL ambos funcionan con el `acs_permission.permission_p` los dos devuelven un flag que indica si el usuario actual tiene permiso para realizar la acción dada. Por defecto, el procedimiento Tcl extrae el `user_id` fuera del entorno de petición, mientras el procedimiento de SQL lo toma como un argumento.

También muestra cómo visualizamos usando items más complicados que usan el sistema de plantilla. El código aquí crea una fuente de datos *multirow*, es decir una fuente de datos que representa una consulta que devuelve múltiples filas de la base de datos. Por cada fila, nos devuelve el ID de la nota, el ID del dueño de la nota, el título, el cuerpo y tres flags que indican si el usuario tiene privilegios de escritura, administración, y borrado. También, la cláusula `WHERE` de la consulta asegura que sólo vemos notas que se nos permiten ver.

Luego, si comprobamos el fichero `index.adp` [el <http://cvs.arsdigita.com/cgi-bin/cvsweb.pl/acs-packages/notes/www/index.adp?rev=1.1&content-type=text/x-cvsweb-encarecimiento>]. Este es más complicado. La parte principal de esta página usa una etiqueta de plantilla múltiple. Si queremos probar, podemos reemplazar el cuerpo principal de la etiqueta múltiple con esto:

```
<multiple name=notes>
```

```
<td>@notes.title@</td><td>@notes.body</td>
</multiple>
```

Esto creará una tabla con una nota por la fila.

Ahora si ponemos código más complicado. Introducimos:

```
<if @notes.write_p@ eq 1>
  <a href=add-edit?note_id=@notes.note_id@>@notes.title@</a>
</if>
<else>
  @notes.title@
</else>
```

Esto visualiza el título de la nota como un enlace o texto plano dependiendo si antes le dimos privilegios de escritura sobre el objeto. La etiqueta tag es algo que en el sistema de plantillas de OpenACS 5.0.0b1 se define para soportar la representación condicional. En la guía del diseñador de plantillas [/[doc/acs-templating/developer-guide.html](http://doc.acs-templating/developer-guide.html)] podemos encontrar más información sobre esto.

Si estudiamos el resto del sistema veremos que la aplicación “notas” no ata los privilegios explícitamente a los objetos que crea. Todos los privilegios se heredan del contexto del objeto que en este caso es del paquete. La asunción es que el administrador del sitio creó una instancia de notas para cada usuario que el quiso, y puso un conjunto de permisos apropiadamente.

5.3.4 El sistema de plantillas

En OpenACS 5.0.0b1 el Sistema de Plantillas (ATS) se diseña para permitir a diseñadores tener separada claramente la lógica de la aplicación de la lógica de diseño. El intento es tener toda la lógica relacionada con la manipulación la base de datos y otros datos del estado del uso en un lugar, y toda la lógica relacionada con exhibir el estado del uso en otro lugar. Esto proporciona una personalización más rápida al diseñador y actualizaciones más fáciles, y también permite a diseñadores gráficos trabajar más independientemente.

En ATS, se escriben dos archivos para cada página (usuario-visible) en el sistema. Uno es un archivo plano .tcl y el otro es un archivo especial de .adp Los archivos .tcl ejecutan un script que prepara un conjunto de

parámetros name/value que se conocen como las fuentes de datos. Estas fuentes de datos [/ doc/acs-templating/guide/data.html] son generalmente el resultado de los ficheros Tcl y/o consultas a la base de datos o alguna combinación de estas. El sistema de plantillas pone disponibles automáticamente los archivos .adp, o visualiza parte de la plantilla, que se escribe con una combinación de HTML, y de pares valor que proporcionan substituciones obtenidas de las fuente de datos.

En el contexto global del ejemplo visto antes, la aplicación Notas, OpenACS mostrará cómo preparar una simple plantilla que visualiza un formulario al usuario para que pueda introducir las nuevas notas en el sistema.

Ejemplo Las Notas

Para que el uso de las notas sea útil, tenemos que permitir que los usuarios incorporen datos a la base de datos. Normalmente, esto se hace con dos páginas: una que exhibe una forma para la entrada de datos, y otra página que ejecuta el código para poner al día la base de datos y dice al usuario si la operación se realizó con éxito. A continuación, veremos como se utiliza el sistema de plantillas para construir la primera de estas páginas.

El archivo .tcl para la entrada del formulario en la plantilla es bastante simple. Aquí, la única cosa que necesitamos de la base de datos es una nueva identificación para que el objeto nota que sea insertado. Abrimos un archivo llamado note-add.tcl en el directorio del ROOT/packages/notes/www, y ponemos el código siguiente:

```
ad_page_contract {
    Form to add a note in OpenACS Notes.
    @author Jane Coder
    @creation-date 11 Oct 2000
} -properties {
    note_id:onevalue
    submit_label:onevalue
    target:onevalue
    page_title:onevalue
} -query {
}
set user_id [ad_verify_and_get_user_id]

db_lrow user_name {
    select first_names || ' ' || last_name as user_name
    from users
    where forum_id = :user_id
}
set page_title "Add a note for $user_name"
set submit_label "Add"
set target "note-add-2"
```

Algunas aclaraciones a cerca del texto:

- El procedimiento `ad_page_contract` es siempre la primera cosa que un archivo `.tcl` llama, si está bajo el directorio de WWW (es decir no un archivo de la biblioteca del Tcl). Hace la validación de los valores de la entrada de la petición HTTP (es decir variables del formulario) y en este caso, - la cláusula de las características se utiliza para instalar las fuentes de datos que enviaremos encima a la pagina `.adp`. En este caso, utilizamos solamente la clase posible más simple de fuente de datos, llamada `onevalue`, que apenas llevan a cabo un solo valor de la secuencia.
- Después de ser declarado en el `ad_page_contract`, cada característica es una variable simple en el Tcl. El sistema de plantilla pasa el valor final de la variable a la plantilla del `adp` cuando se procesa el archivo `tcl`.
- El `ad_return_template` de la llamada dice al sistema qué página de la plantilla del `adp` debe traer para visualizar las características que se han procesado. Por defecto, el sistema de plantillas buscará un archivo con el mismo nombre que el archivo `.tcl` que acaba de ejecutar, pero con una extensión `.adp`.

Escribimos después la página correspondiente del `adp`. Esta página hace salir el HTML para el formulario, y también contiene los placeholders donde los valores se substituyen adentro de las características configuradas por el archivo del `tcl`. Creamos un archivo llamado `note-add.adp`, y insertamos este texto:

```
<master src="master">
<property name="title">@page_title@</property>
<property name="context_bar">@context_bar@</property>
<form action=@target@>
<p>Title:
<input type="text" name="title" value="">
</p>
<p>Body:
<input type="text" name="title" value="">
</p>
<p>
<center>
<input type=submit value="@submit_label@">
</center>
</p>
```

El punto principal de las notas esta aquí: cuando queremos sustituir un valor en una página, ponemos el nombre de la fuente de datos entre los caracteres "@ ". Otro cosa a destacar es el uso de una plantilla maestra: Las plantillas maestras permiten centralizar código para visualizar que se usa a lo largo de una aplicación en un solo archivo. En este caso, nosotros pensamos tener una plantilla maestra que genera las cabeceras de las páginas normales y los pies de página para nosotros - crearemos los archivos master.adp de esta manera:

```
<%= [ad_header $title] %>
<h2>@title@</h2>
<%= [eval ad_context_bar $context_bar] %>
<hr>
<slave>
<br clear="all">
<%= [ad_footer] %>
```

5.4 DotLRN

Ahora veremos que es dotlrn y que relación mantiene con OpenACS.

Sloan School of Management, es la escuela de negocios del Massachusetts Institute of Technology (MIT). En 1999 Sloan junto con la empresa Arsdigita (ahora parte de Redhat) iniciaron el desarrollo de un sistema de gestión que pudiese ser adaptado a sus sistemas de información internos. La primer versión llamada ACES estaba basada en el Arsdigita

Community System, ahora llamado OpenACS visto en el punto anterior de este capítulo. Esta plataforma permitía que ACES tuviese un gran número de funcionalidades para la colaboración entre alumnos.

La arquitectura fue diseñada por ingenieros de MIT para facilitar la implementación de comunidades virtuales con un mínimo de esfuerzo, y al mismo tiempo con la suficiente capacidad de extenderse a nuevas áreas de desarrollo, incluso a aquellas que no fueron contempladas en el diseño original. Este tipo de diseño de software es comúnmente llamado Object Oriented Application Frameworks.

OpenACS es el framework web y dotlrn es su sistema de e-learning y de gestión de comunidades. Es un proyecto open source que proporciona gestión de cursos y comunidades. Es escalable, robusto, extensible y cumple el estándar SCORM³. OpenACS implementa un modelo de datos orientado a objetos que los programadores pueden modificar. Los usuarios, o incluso administradores del sistema tienen una interface web que les permite crear departamentos y escuelas dentro de los cuales se distribuyen los cursos. Cada uno de estos cursos tiene una página creada automáticamente ("portal") dentro de la cual aparecen todas las funcionalidades que el profesor considere importante para un curso particular. La apariencia de este portal puede ser adaptada por el profesor sin necesidad de programar. Cada usuario del sistema (Profesor, Alumno, Ayudante, Administrador, etc) tiene una página personal donde la información de todos los cursos y comunidades a las que pertenece es compilada y presentada en forma uniforme.

Cada portal normalmente incluye:

- Foros de discusión. Desde allí los profesores pueden organizar actividades de colaboración virtual, donde alumnos, auxiliares y profesores discuten algún tema o trabajan en algún proyecto común.
- Agendas y eventos. Los alumnos y profesores tienen otras actividades que van más allá de un curso particular, estas funcionalidades permiten "sincronizar" agendas personales con las agendas de cursos y comunidades. La utilización de estándares

3

SCORM (Shareable Course Object Reference Model)

permite crear mecanismos de sincronización con herramientas populares como iCal, MS Outlook.

- Noticias. Donde los profesores pueden anunciar o recordar a los alumnos eventos tales como exámenes, tareas, etc.

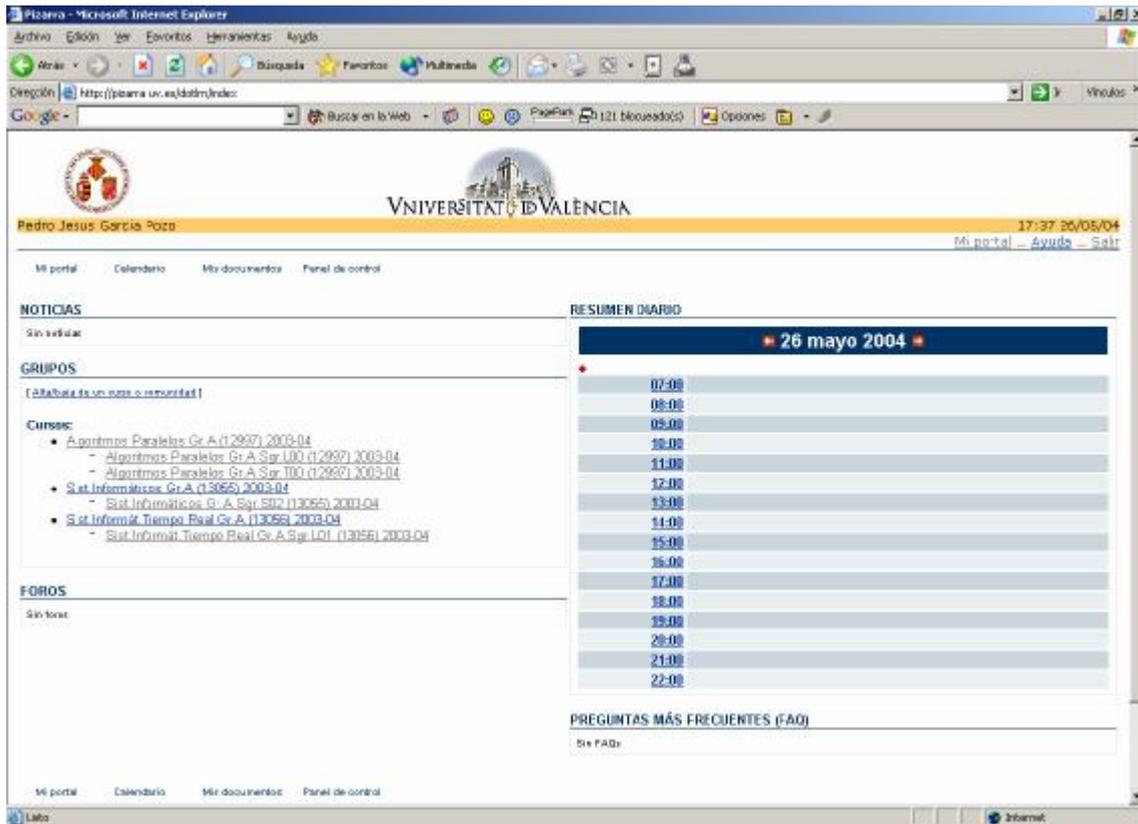


Ilustración 12. Ejemplo de portal de un alumno.

El software que se usa para entregar los cursos on-line actualmente y la colaboración se basa en la plataforma OpenACS que implementa un modelo orientado a objetos y una base de datos relacional, la base de datos utilizada fue Oracle, pero la nueva plataforma también permite utilizar Postgres, una excelente base de datos distribuida como software libre. Dotlrn es el encargado de incluir funcionalidad al portal en la parte de e-learning, gestión de estudiantes, clases, foros, compartir ficheros, etc.

Con la evolución de las tecnologías de la información y con el aprovechamiento de la red, los estudiantes cada vez cambian más su rol de estudiante presencial, las universidades presionadas por esta tendencia infrenable se ven forzadas a tomar medidas para no retrasarse en esta área ya que el e-learning resulta ser la respuesta que dan las nuevas tecnologías ante el cambio que esta experimentando la sociedad.

Las instituciones están viendo el desarrollo y la rapidez que abarca este nuevo producto y no quieren encerrarse en un producto comercial y por eso están basándose en software "Open Source" y estándares abiertos.

Hay muchas organizaciones que están cambiando y a menudo les gusta más, desarrollar sus sistemas de información a sus propios procesos comerciales y así se benefician del software de código abierto, ya que estos permiten la personalización completamente. Normalmente estas organizaciones colaboran distribuyendo sus fuentes a otras organizaciones que pueden utilizarlos según la licencia o autorización otorgada.

El código abierto permite a las instituciones rehusar y personalizar la aplicación hecha por otras instituciones, e incluso construir en él nuevos rasgos. El desarrollo de código abierto ha sido adoptado por muchos como una manera de "el customisability" permitiendo reusabilidad, y al mismo tiempo reduce los costos de la licencia. Abrir el código tiene las ventajas grandes de ver crecer los sistemas rápidamente. Construir tu propia plataforma es a menudo muy caro y arriesgado, ya que no se aprovechan las economías que hacen el software más barato, comparando con la creación del sistema desde el principio.

5.4.1 Diseño y arquitectura del Sistema

dotLRN se construyó originalmente por la corporación de ArsDigita (ahora la parte de RedHat) para Sloan School con el fin de gestionar el MIT.

Diseño del Sistema

La plataforma inicial ACS fue implementada inicialmente en un servidor web AOLServer con el lenguaje de programación TCL y como gestor de base de datos Oracle RDBMS. En diciembre de 1999, el proyecto de OpenACS fue iniciado por varios diseñadores en busca de conseguir una versión de ACS que correría sobre un gestor RDBMS open source, Postgres. Hoy, OpenACS es un proyecto maduro en su propio derecho, con encima de 7000 miembros y por lo menos 10 compañías que proporcionan apoyo comercial y desarrollo.

DotLRN usa el framework orientado a objetos de OpenACS para la creación de aplicaciones web. OpenACS proporciona un modelo de datos standar y un API. La arquitectura orientada a objetos del framework's incluye un repositorio (content repository) utilizado por la mayoría de las aplicaciones que necesitan almacenar el "content" un módulo de gestión de user/group usado por el sistema de permisos, una petición procesada que habilita persistencia, autenticación y plantillas.

En OpenACS, cuando una página es solicitada por un usuario, la petición procesada autentifica al usuario utilizando el módulo user/group (la parte interna "core" del sistema), para autorizar o no la petición. Si se autoriza, la página es personalizada usando al sistema de plantillas y el modelo del usuario. El sistema de plantillas ensambla la página con los diferentes trozos devueltos, y es enviada por el servidor al usuario. El sistema de plantilla ensambla la página utilizando contenidos que pueden venir de aplicaciones diferentes (por ejemplo una aplicación como calendario o un evento particular en un horario de la clase) y cada trozo de contenido se trata en una manera similar, con un API standar. El sistema de permisos permite a administradores dar los permisos de (read/write/administer) a cualquier content "objeto" del sistema. La petición procesada proporciona persistencia así como el mapeo de una URL particular a un file/program individual que el servidor usa para producir esa página.

El modelo de datos estándar de OpenACS es muy práctico por varias razones. Para el usuario, permite de forma extensa una personalización para todas las aplicaciones de un sitio particular. Para administradores, OpenACS mantiene con un solo sistema de plantillas, gestión de paquetes y sistema de permisos en todas las aplicaciones. Si un administrador fuera a usar un sistema de gestión comercial, un sistema de gestión de learning de otro y herramientas de colaboración desde otros, la integración de todas estas herramientas sería embarazosa, y cara. Los sistemas adicionales se utilizarían en un sólo sentido y muchas plantillas necesitarían ser mantenidas.

Un único modelo de datos también puede explotarse para proporcionar características inteligentes como la clasificación automática del documento con herramientas o la nueva recuperación de información.

DotLRN Arquitectura

dotLRN ofrece varias aplicaciones colaborativas como son los calendarios, los foros, los ficheros compartidos, herramientas de valoración, etc. Eso sumado a los que el almacén de OpenACS proporciona (es decir el e-comercio, gestión de contenidos, el workflow, el webmail, el etc). Muchas de estas funcionalidades son similares a las encontradas en otros LMS como WebCT y Pizarra, por citar algunos, a veces con rasgos adicionales hechos posible por el modelo de datos común.

La arquitectura del almacén OpenACS se describe en la Figura 6. El diagrama tiene cinco componentes principales: el usuario, webserver, almacén, base de datos y servicios. La arquitectura del dotLRN es similar de muchas maneras a la Vista de WebCT, pero usa OpenACS en lugar del almacén de Weblogic de BEA, y puede usar otras bases de datos además de Oracle, particularmente PostgreSQL.

DotLRN Portal de usuario

La arquitectura de OpenACS mostrada en Figura 13 está compuesta de los paquetes del acs-core que mantienen a través del API la funcionalidad común entre otros servicios y paquetes. Los servicios son paquetes que no interactúan con el usuario y establecen un interfaz entre los paquetes, dónde reside la aplicación real. Como dotLRN se construye con el framework de OpenACS, hereda todas las funcionalidades de OpenACS, modularidad y características.

Como ejemplo, dotLRN usa el calendario de OpenACS. Distintas instancias del calendario pueden visualizarse en el portal de un estudiante, visualizando en una sola página todos los eventos por cada curso en los que el estudiante está matriculado. Cada vista de un paquete se llama un "portlet". El sistema adapta el portal a cada usuario recogiendo la información sobre los cursos y comunidades en los que el usuario está matriculado o pertenece, y qué aplicaciones de estos cursos y comunidades están usando. Cada una de estas aplicaciones generará contenidos apropiados mostrados en un portlet. También puede personalizarse por el usuario que puede agregar o puede quitar éstos y otros portlets.

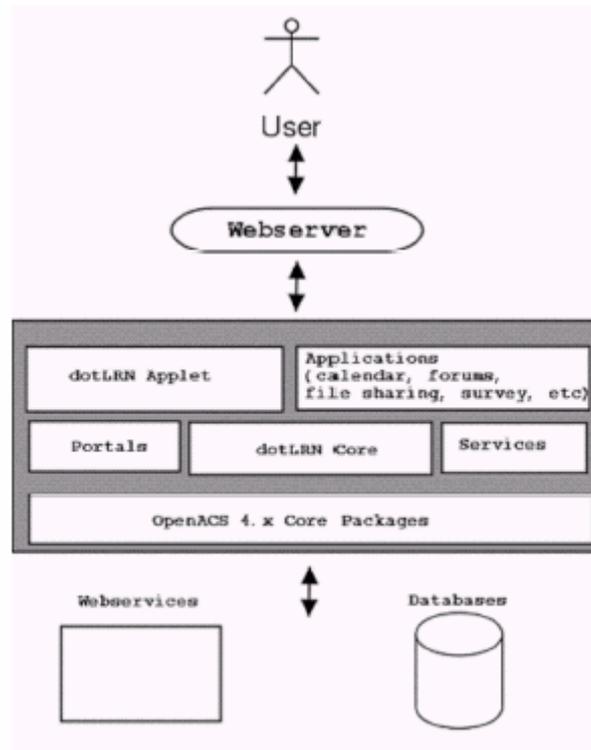


Ilustración 13. Arquitectura Openacs y dotLRN

DotLRN Comunidades

Nos sirven para compartir un espacio de trabajo, varios integrantes de la comunidad pueden aprender a través de los cursos que cogen de la comunidad y aportar a esta sus estudios. En una compañía, las comunidades podrían representar los grupos de interés especiales, o proyectos o simplemente recolecciones sociales. En la universidad, clubes y sociedades es muy común. Se apoyan ambos tipos de actividades en dotLRN. Esto requiere scoping muy claro de cada aplicación para que cada course/community pueden tener sus propias aplicaciones fijas, plantillas y permisos. dotLRN usa un "subsites" concepto que separa los contextos entre los diferentes tipos de comunidades ("las clases" y "los clubes", para nosotros asignaturas y comunidades). Cada caso del subsite ("dotLRN_community") puede tener su propia estructura de permisos (es decir diferentes personas con es diferentes).

La Figura 14 muestran en un diagrama de clases cómo las comunidades de dotLRN pueden ser clases o clubes, ambos tipos proporcionan juegos diferentes de herramientas de colaboración, los roles para personas y plantillas con valor predeterminado. En ambos casos, estas comunidades tienen su propia instancia de calendarios, foros, las

noticias, los grupos, el archivo compartido, etc. La figura 14 también muestra esta estructura y cómo las clases relacionan el departamento con la clase subject. “los departamentos” describen la estructura orgánica normal dentro de la institución, “el Subject” contiene información que repite en los diferentes casos de un “Class.” “Class” especifica personas y sus papeles, salida y el acabado fecha, y otros atributos que son específico a ese caso. Por ejemplo, “los Web Aplicación frameworks” (un subjectc) podría enseñarse al “el Departamento de Informática” (un departamento) en el segundo semestre de 2002, por Juan Teacher, bajo el nombre “los Web Aplicación Armazones” (una asignatura, clase).

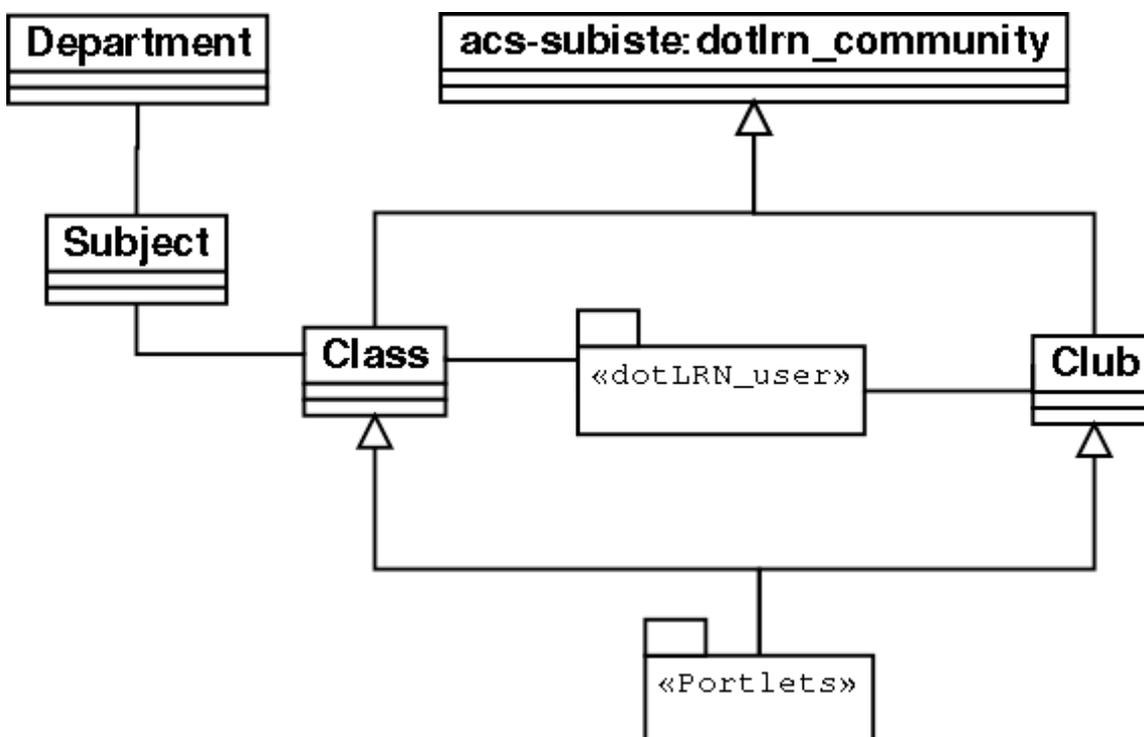


Ilustración 14. Diagrama de clases en dotLRN

Portlets

dotLRN usa la arquitectura del portal OpenACS para generar el interfaz del usuario. Los portlets de dotLRN generan el interfaz apropiado entre las aplicaciones de OpenACS y los portales de dotLRN. Haciendo que, estos portlets entreguen funcionalidad personalizada basada en las aplicaciones de OpenACS. Figure 7 muestra un diagrama de la clase del diseño de un portlet.

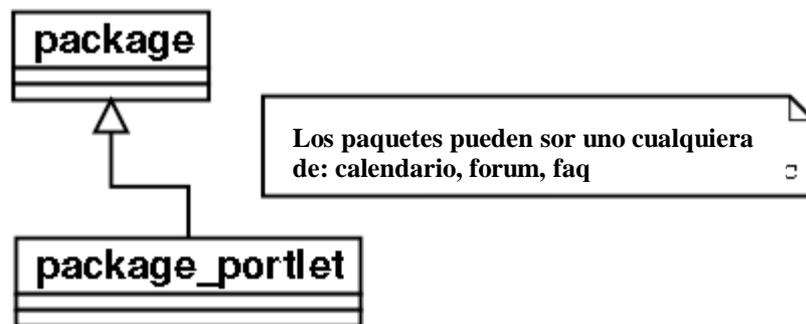


Ilustración 7. Portlets en dotLRN

DotLRN usuarios

OpenACS puede soportar complejas jerarquías del usuario definiendo "parties" esto puede ser: "los usuarios" identificados por un distinto correo electrónico, o una "persona" con el distintivo primero y el último nombre, o un "grupo" que quizá uno o más del anterior, o incluso otros grupos. Estas relaciones se muestran en Figura 15.

Con esta arquitectura de dotLRN los usuarios pueden tener más de un rol en particular. Por ejemplo, un usuario podría ser un ayudante instrucción en una clase y un estudiante en otro. Por lo tanto, el papel de un usuario en una clase determinaría lo que el usuario se le permite hacer. Todos los nombres de roles y entidades en dotLRN, como la clase, las comunidades, instructor, etc, pueden cambiarse fácilmente a las necesidades de la institución o país dónde el sistema se este usando. La clase del dotLRN es una subclass de las ""parties" de OpenACS con el "rol" como un atributo, este puede tomar los valores como el Estudiante, Curso Assistant, Instructor, etc.,

Los cursos presenciales o semi-presenciales varían a lo largo del tiempo, en la estructura y los contenidos se apoyan en los resultados obtenidos. Los cursos on-line pueden ir más allá y pueden cambiar para cada estudiante. Utilizando al estudiante y aprendiendo de los modelos de cada estudiante puede ser la experiencia para adaptar el curso original. Esta característica se llama aprendizaje adaptable y son uno de las metas de la investigación con dotLRN.

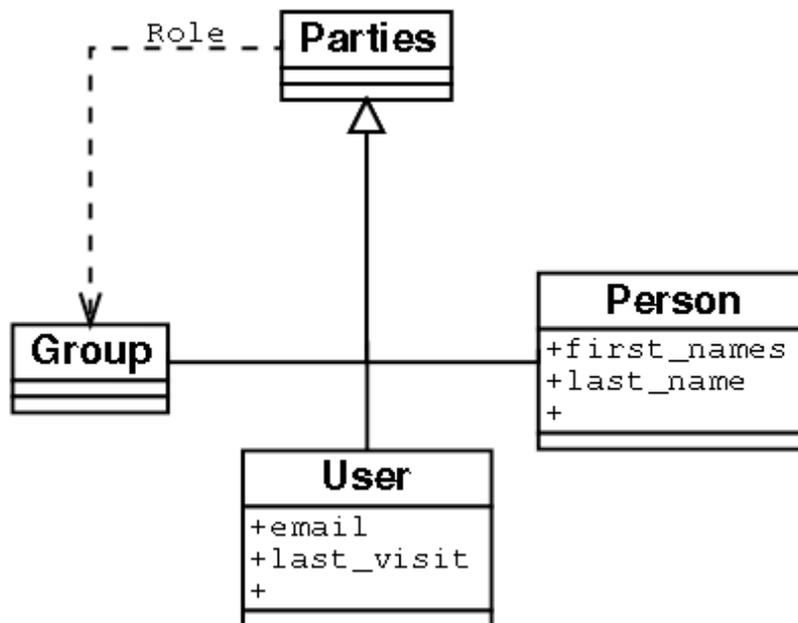


Ilustración 15. Paquete "parties" de OpenACS.

Content Repository y workflow en OpenACS

Las aplicaciones de Web manejan grandes cantidades de pequeñas piezas de contenidos. Estas piezas pueden venir en el posting de los foros de discusión, en los eventos de un calendario, en las noticias, en los adjuntos e incluso los elementos multimedia. En OpenACS cada pedazo es registrado y almacenado en un "Content Repository" que estandariza la

manera en que las aplicaciones acceden y maneja esta información. Esta capa adicional de complejidad hace posible la reusabilidad de todas las funcionalidades de gestión de contenidos como los permisos, formato u otras funcionalidades diferentes que no pueden haber estado incluidas en el momento de diseñar una aplicación. Un servicio creado para el Content Repository, como una recuperación de información o el sistema de clasificación de documentos automático, puede usarse automáticamente por todas las aplicaciones en el sistema. Con una arquitectura diferente, el servicio no necesitaría ser implementado de nuevo o adaptado, solamente para cada aplicación, lo hace a través de una representación interior.

Un ejemplo sería; para desarrollar código en OpenACS, necesitamos construir un paquete nuevo. Un paquete nuevo es una colección de páginas web, código Tcl, y tablas de bases de datos y procedimientos. Para hacer esta tarea se utiliza en Content Repository, esto nos hace la faena más fácil aunque nos limita un poco la funcionalidad.

El servicio workflow de OpenACS también agrega capacidades útiles al dotLRN desde que puede usarse para hacer aplicaciones "workflow aware" de una forma estandarizada. Los workflow habilitan a las aplicaciones para usar todas las alertas y las capacidades de gestión de eventos construidas en el paquete workflow. El sistema workflow podría colocar los correos electrónicos o SMS para ser enviados cuando una tarea ha sido terminada o se necesita empezar, podría poner en orden las tareas para mostrar en los usuarios, centralizar el calendario o podría proporcionar rasgos que no fueran considerados en el momento de diseñarlos en las aplicaciones.

5.4.2. Internacionalización en OpenACS y dotLRN

A través de una colaboración entre Sloan School of Management, están traduciendo en la Universidad de Heidelberg y Greenpeace International, OpenACS y dotLRN e internacionalizaron a varios idiomas (locales). El proyecto de internacionalización permitirá a dotLRN estar disponible en varios idiomas como son inglés, español, y alemán. Habrá mejorado el apoyo para los cursos corrientes entre zonas diferentes.

5.4.3. DotLRN y SCORM

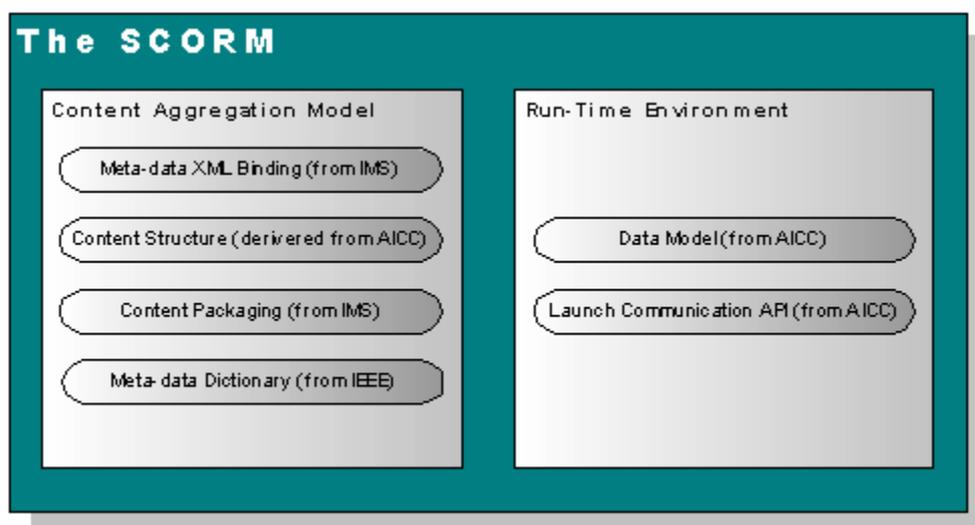
La estructura de las normas es un requerimiento muy importante en la infraestructura del e-learning. La comunidad de OpenACS se comprometió a seguir las normas, el más reciente esfuerzo fue hacer compatible dotLRN con las normas SCORM 1.3. Aunque en la próxima versión de dotLRN ya se integrará el paquete LORS que nos permite la exportación e importación de objetos SCORM haciéndolo compatible con las normas SCORM 1.2.

El Sharable Content Reference Model (SCORM) es uno del más importante de estas iniciativas y proporciona:

- **Accesibilidad:** habilidad para acceder desde cualquier sitio y trabajar desde cualquier sitio
- **Interoperatividad:** Poder utilizar cualquier curso desarrollado en otra plataforma distinta.
- **Durabilidad:** A pesar de que existan cambios tecnológicos durante el curso, los trabajos puedan entregarse conforme fueron diseñados.
- **Reusabilidad:** los cursos pueden reutilizarse para otros cursos o formar parte en otros contenidos.

El SCORM es una colección de especificaciones generado desde los grupos de industria como el IMS el Consorcio de Aprendizaje Global, Inc., la Industria de la Aviación CBT (Computer Based Training) el Comité (AICC), Alliance of Remote Instructional Authoring & Distribution Networks de Europe (ARIADNE) y el Instituto de Eléctrico y Electrónica Ingenieros (IEEE) Learning Technologies Standards Committee (LTSC).

SCORM define un modelo de referencia para generar aprendizaje via web basado en "Content Aggregation Model" y "Runtime Environment."



Las principales características de la implementación de SCORM 1.3 en dotLRN, fueron desarrolladas por Sussdorff-Roy GbR (Alemania) y el Web Engineering Grupo (la Universidad de Sydney). El proyecto incluye las funcionalidades siguientes:

- Importar/exportar objetos SCORM.
- Una herramienta para gestionar los cursos de SCORM dentro del armazón de OpenACS/dotLRN, Una herramienta administrativa para manejar cursos SCORM, sucesiones (las actividades), SCO/As (aprendiendo los objetos), y recursos (los archivos).
- Un entorno de ejecución(lanzando y entregando SCOs y rastreando)
- Secuencias de Objetos e-learning basado en IMS.

CAPITULO 6

6. Caso práctico:

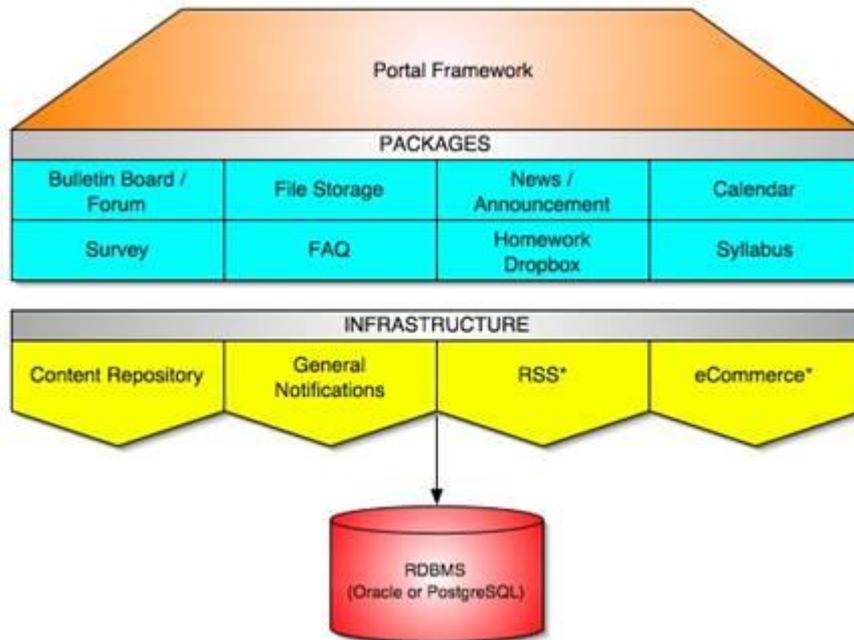
“dotLRN en la Universida de Valencia”

Como hemos comentado dotLRN es un software bajo licencia GNU General Public License (GPL), en constante modificación para integrar los estándares actuales. Las versiones más reciente nos ofrecen sustanciales mejoras y auguran un futuro prometedor.

.LRN v3, dio énfasis al apoyo de las normas (e.g. IEEE, IMS, SCORM, OKI).

.LRN v4, prevista para otoño del 2004, proveerá más funcionalidad a las herramientas de e-learning en cuanto a gestión de contenidos, webdav,Loors Managent.

DotLRN se distribuye con un mínimo de aplicaciones y servicios ya instalados por defecto que son necesarias para la gestión de comunidades y para el proceso de elearning. En el siguiente gráfico veremos como los principales módulos están instalados y en que nivel se encuentran. Esta plataforma nos permite la validación de usuarios vía LDAP, administración de usuarios, comunidades y cursos a través del web, con lo que desde una máquina cualquiera que disponga de un navegador podemos administrar fácilmente el sitio. Soporta gestión de grupos, portal de usuarios personalizable, compartir ficheros, calendario, foros de discusión, FAQ, noticias, página de bienvenida personal, etc.



Los principales servicios y módulos que nos proporciona la plataforma son los siguientes.

- Validación LDAP ante los servidores que dispone la Universidad de Valencia. Política muy seguida en todos los servicios que proporciona el servicio de Informática a toda la comunidad.
- Portal personal para cualquier miembro con acceso a la plataforma, es decir con cuenta de correo activa. El portal personal es configurable, y en esta se pueden ver los principales paquetes que el administrador haya habilitado como son; los cursos a los que pertenece, foros, preguntas frecuentes, noticias, calendario personal en el que visualiza un resumen diario de las citas que tiene el usuario ya sean de su propio calendario o citas de cualquier curso en el que este inscrito.
- Gestión de comunidades para trabajo en grupo.
- Permitir compartir ficheros con los demás miembros del curso, o tenerlos él privados.
- Herramienta para la creación de exámenes, pruebas tipo test.
- Traducción de la plataforma a diversos lenguajes: árabe, australiano, inglés, alemán, francés, castellano, noruego, sueco, danés, polaco, italiano, griego, portugués, ruso, sueco, indonesio, ucraniano y chino.

Una vez vistas las características principales de la plataforma GPL a implantar en la Universidad de Valencia pasamos a realizar un estudio de la instalación de esta plataforma en una organización tan exigente y con tantos usuarios.



Debemos destacar que actualmente las organizaciones con más usuarios en este tipo de plataforma son las que se muestran abajo y que la universidad de Valencia ya esta incluida como caso de estudio.:

Name of Institution	Contact	Students (Users) Platform	Screenshots	Case Study
Asia				
Birzeit University Birzeit, West Bank, Palestine	N/A	5,000 (5,000) Linux/Oracle	N/A	Case Study
University of Delhi Delhi, India	N/A	200,000 (N/A) N/A	N/A	Case Study
Australia				
University of Sydney - WEG Sydney, Australia	N/A	35,000 (200) OSX/PostgreSQL	N/A	Case Study
Europe				
American University in Bulgaria Sofia, Bulgaria	N/A	N/A (N/A) N/A	N/A	
Fachhochschule Heilbronn Heilbronn, Germany	N/A	3,800 (120) Linux/PostgreSQL	N/A	Case Study
Freie Universität Berlin Berlin, Germany	N/A	N/A (N/A) N/A	N/A	Case Study
Københavns Universitets Copenhagen, Denmark	N/A	35,000 (N/A) N/A	N/A	Case Study
Universidad Nacional de Educación a Distancia Spain	N/A	180,000 (4,500) Linux/Oracle	N/A	Case Study
Universität Heidelberg Heidelberg, Germany	Carl Blesius crb <at> uni- hd <dot> de	30,000 (3,000) Sun/Oracle	Screenshot 1	Case Study
Universität Mannheim Mannheim, Germany	N/A	36,000 (21,000)	N/A	Case Study
Universidad de Valencia Valencia, Spain	Dario Roig	80,000 (40,000 students, 5,500 subjects) Linux/PostgreSQL	N/A	
Universitetet i Bergen Bergen, Norway	N/A	25,000 (13,000) Linux/Oracle	N/A	Case Study
University of Cambridge Cambridge, UK	N/A	16,500 (3,500) N/A	N/A	Case Study
Wirtschaftsuniversität Wien Vienna, Austria	N/A	25,000 (16,000) Linux/PostgreSQL	N/A	Case Study
North America				

MIT, Sloan School of Management Cambridge, USA	Alfred Essa aessa <at> mit <dot> edu	11.000 (11,000) Sun/Oracle	Screenshot 1 Screenshot 2	Case Study
St. Mary's College of Maryland St. Mary's City, USA	N/A	N/A (N/A) N/A	N/A	Case Study
University of Massachusetts Donahue Institute Boston, USA	N/A	N/A (N/A) N/A	N/A	Case Study
South America				
FLACSO (Faculdade Latino-americana de Ciências Sociais) Brasília, Brazil	N/A	N/A (N/A) N/A	N/A	
ITESM (Tecnológico de Monterrey) Monterrey, Mexico	N/A	N/A (N/A) N/A	N/A	
Universidad Agraria de la Habana Habana, Cuba	N/A	2,500 (500) Linux/PostgreSQL	N/A	Case Study
Universidad del Cauca Popayán, Colombia	N/A	N/A (N/A) N/A	N/A	
Universidad Galileo Guatemala City, Guatemala	N/A	3,000 (3,000) Linux/Oracle	N/A	Case Study
Universidad Nacional de Ingeniería Managua, Nicaragua	N/A	10,000 (N/A) Linux/PostgreSQL	N/A	Case Study

Organizaciones

Name of Organisation	Contact	Users Platform
Empresa Brasileira de Produtos Agrícolas Brasília, Brazil	N/A	N/A N/A
Centro de Gestão do Conhecimento at MRE Brasília, Brazil	N/A	N/A N/A
Ministério do Desenvolvimento Indústria e Comércio Exterior Brasília, Brazil	N/A	N/A N/A
Greenpeace International Amsterdam, The Netherlands	N/A	70,000 Linux/Oracle
Elearning Network of Australasia Australia	Rafael Calvo	100 Linux/Postgres
AIESEC New York, USA	N/A	35,000 Linux/Oracle

Grupos de investigación

Name of Organisation	Description
Web Engineering Group at University of Sydney Sydney, Australia	N/A
E-LANE (European-Latin American New Education) Various Universities in Europe and Latin America	Promover un ambiente integrado de aprendizaje en el marco de la sociedad de información, en niveles de entrenamiento tanto académico como no académico. El proyecto propone el desarrollo de una plataforma abierta de tele-enseñanza mediante la integración de aplicaciones consolidadas, el diseño de una metodología innovadora de aprendizaje orientada y soportada en dicha plataforma, y la integración de contenidos de cursos de las más prestigiosas instituciones educativas en Europa y Latinoamérica, con el objetivo de proporcionar a la sociedad material educacional de excelente calidad a bajo costo.

	Partners Europe Universidad Carlos III de Madrid Coordination Partner Madrid, Spain Telefónica I+D Madrid, Spain Institut National des Télécommunications Évry, France Trinity College Dublin Dublin, Ireland University of Reading Reading, UK
	Partners Latin America Tecnológico de Monterrey Monterrey, Mexico Universidad del Cauca Popayán, Colombia Universidad de Chile Santiago, Chile Universidad Galileo Guatemala City, Guatemala Universidade Estadual de Campinas São Paulo, Brazil

Viendo esta grafica podemos pensar a priori que es una arquitectura muy seguida por muchas y distintas organizaciones, lo que nos da crédito en pensar que no tendremos problemas de escalabilidad ya que hay otras organizaciones con más usuarios que la universidad, aunque si es posible que en temas de hardware encontremos limitaciones por parte de los servidores ya que en principio no se sabe de antemano como van a responder con un carga real de tantos usuarios, hablamos de unos 60.000 usuarios en la Universidad de Valencia. Es por este motivo que se ha optado en instalar la plataforma de manera piloto para este cuatrimestre y evaluar los resultados al final del mismo para no incurrir en ningún error cuando se lance en plena producción. Los profesores que deseen dar su docencia a través de esta plataforma sólo deben solicitar que se les de de alta su docencia y automáticamente puede comprobar que asignaturas tiene en docencia y que alumnos. Los alumnos no deben hacer nada ya que son

miembros de esos cursos y a partir del momento de solicitar el alta el profesor ellos tendrán disponibles esas asignaturas, mientras tanto pueden utilizar su portal.

No nos planteamos hacer un estudio exacto de lo que costaría implantar la plataforma, o que herramientas nuevas se deben desarrollar en la plataforma ante una organización tan exigente, sino facilitar esta labor y ser un modelo de referencia ya que para hacer este fin sería hacer un proyecto entero que se escapa de este fin. Así que a continuación tratare de plantear que requerimientos tiene actualmente la universidad basándome en cifras de usuarios y material disponible.

6.1 Análisis de requisitos

He basado el análisis en dos variables. Por una parte averiguaremos el número potencial de usuarios que tendrá que soportar la plataforma y los requisitos hardware que deberán instalarse para que funcione notablemente.

6.1.1 Requisitos por usuarios

Los datos que muestro a continuación se han recogido del recull de datos estadísticos del año 2003 realizado por el Servicio de Análisis y Planificación de la Universidad y que se encuentra disponible en la web <http://www.uv.es/sap/recull/recull>.

La Universidad dispone de 3 grandes Campus, Burjasot-Paterna, Blasco Ibáñez y Taronges; dentro de cada campus están las diferentes facultades y centros de Investigación que forman su conjunto global, además de los Campus también existen algunos Institutos de Investigación y facultades que no están físicamente en ningún campus, como puede ser Magisterio.

	USUARIOS
PERSONAL UNIVERSIDAD	1628
ALUMNOS	49858
PROFESORES	3159
TOTAL	54645

Vista la población activa en cuanto a usuarios podemos ver que hay dos características muy críticas que nuestra plataforma debe afrontar, **integridad y escalabilidad**.

En cuanto al tema de **integridad** como he comentado anteriormente la plataforma dispone de un modulo de autenticación vía LDAP⁴, protocolo de acceso muy utilizado actualmente.

La personalización de la plataforma no plantea problemas porque haciendo uso de las plantillas se ha conseguido que siga el diseño de otros servicios que proporciona el centro de calculo.

⁴ RFC 1777 estándar de Internet.

La plataforma dispone de soporte multilingüe y además una persona se ha dedicado a la traducción de todos los mensajes al catalán por lo que la herramienta ya esta disponible en 3 idiomas: Ingles, Español y Catalán.

6.1.2 Requerimientos hardware y software

Este punto es el que más dudas plantea ya que no se dispone de ninguna experiencia propia con esta plataforma, de forma que a priori no se ha podido prever las necesidades de hardware para la citada carga de usuarios prevista anteriormente.

Por ejemplo:

¿ Tendremos limitaciones del tipo ?

- Número aproximado de usuarios.
- Número y tamaño de cursos ofrecidos.
- Expectativas de utilización del sistema por parte de los usuarios.
- Los tiempos de respuesta de la BD son óptimos.

Pues en principio se siguen los consejos dados por usuarios y desarrolladores de la plataforma OpenACS en los que se recomienda:

Malte Sussdorff on Jan 14 2004 05:11:12

>>if you have the money and the resources to maintain, use two servers, one for the database with lot's of RAM and another one for the AOLserver.

>>For the database server (also true if you only go with one server):

>> Dual Processor (I'm highly satisfied with Opterons, but that may be too expensive)
- 4GB of RAM, or more (that's what we use on aiesec.net, so far it is enough, but less is not recommended).
- Multi Channel SCSI RAID with fast harddisks. Take a lot of harddisks, that are fast and can be accessed simultaneously. We use a two channel setup with two mirrored harddisks each and strip across both channels. This reduces your harddisks from becoming the bottleneck. Maybe SerialATA is a better alternative though.

>>As for the server for AOLserver, you can use any computer, though you'll need quite some storage space for your users files. OTOH, "quite some" still does not fill up our 250GB RAID, therefore it is all a matter of perspective.

>>Last but not least, think about backups. You will need to backup your database along for all the files in the content-repository (I'd not recommend storing content in the database, but use the filesystem based version). Unless you have a central backup storage at your university, get a (cheap) different computer with lots of harddisk.

Denis Roy on Jan 16 2004 11:57:05

When you are planning your hardware setup, it very much depends on the number of concurrent users. Having a lot of data in your database doesn't really matter if there are only say 20 users online at the same time.

Since Malte is referring to AIESEC, let me give you some more concrete info:

We have 120K registered users. Only about 20K are active since it is also an alumni database. Our webserver is a Dual-P3 with around 933 Mhz each and 2 GB RAM. Our database server is a Dual-Opteron system as Malte already mentioned.

With this setup we support about 70 concurrent users on OACS 4.6.3 and .LRN 1.0 without any noticeable negative impact on performance. When more users try to access the website, performance will decrease considerably. I found out that the webserver is the bottleneck because even when the website is almost not responding anymore at all (around 90 concurrent users) the database is still doing fine. Testing a second webserver at the same time was showing great performance on the second webserver.

We are currently installing a load-balanced solution which will make us very flexible on the frontend (webserver) side. It's just a matter of time until the database will become the bottleneck.

If anyone is interested, I will let you know about the results of our load-balancing and how many users we can handle with it.

También se evalúan los casos de estudios detallados en la propia web de dotLRN, ahí se contempla que hay organizaciones con más usuarios que la universidad lo que da confianza para lanzar la plataforma en plan piloto y esperar resultados ya que las versiones de OpenACS utilizadas en algunas organizaciones son anteriores (v.1) y muy a medida de cada organización.

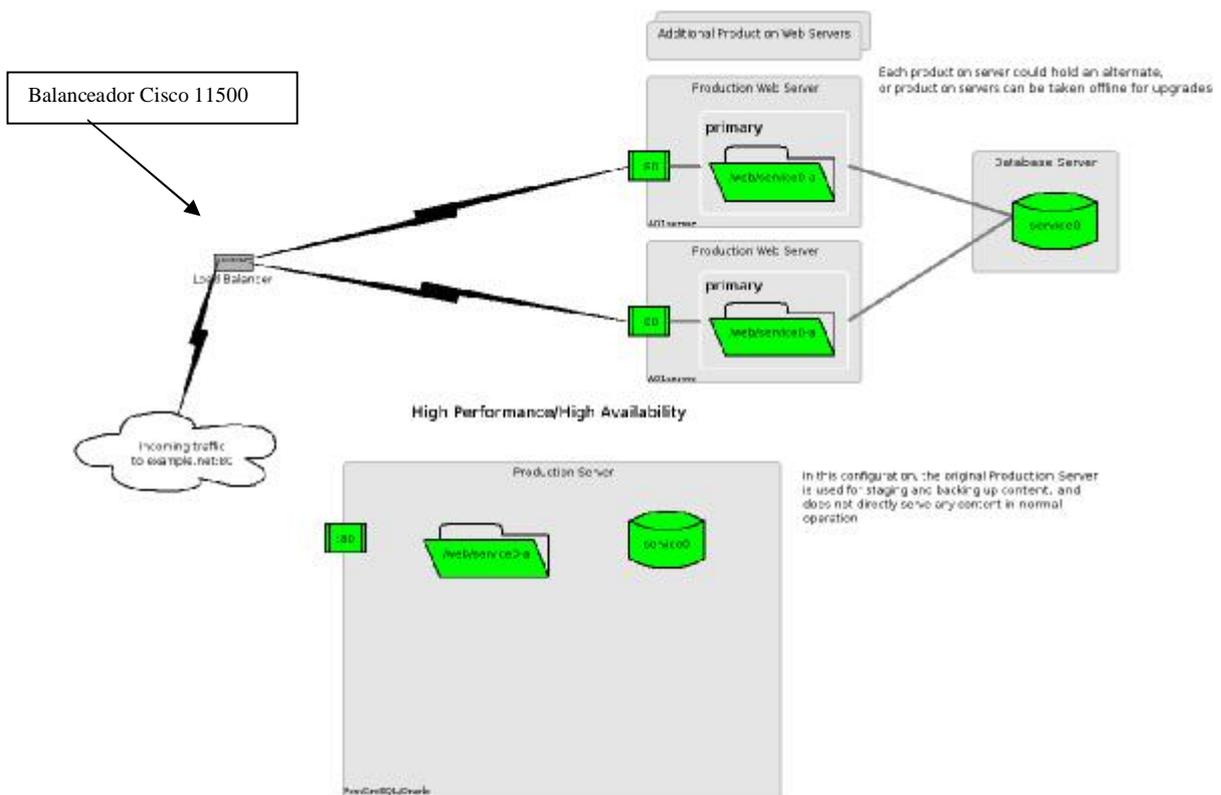
Se hacen reuniones con los técnicos de la UNED para ver su experiencia y tras evaluar los informes y se decide implantarla.

La arquitectura seleccionada que se ha instalado es compuesta por los siguientes elementos:

Se ha optado por colocar un balanceador de carga, Cisco Content Switch CCS 11500, actúa como un router siendo capaz de analizar hasta la 7 capas del modelo de comunicación, 3 servidores dedicados, uno para cada servicio, BD, servidor web, plataforma OpenACS.

El balanceador de se sitúa antes de los servidores web's y este se encarga de redirigir las peticiones hechas por los usuarios a <http://pizarra.uv.es>, y encaminarlas al servidor web que en ese momento este menos cargado (existen 2 servidores web's), veamos la siguiente figura.

También se encarga de balancear el servidor de LDAP.



Al ser una arquitectura modular y poder distribuir la carga se opta por dedicar para cada tarea una determinada maquina, de esta forma se pretende que cumpla su función lo más eficientemente posible.

Servidor para la Base de Datos utilizada PostgreSQL	
Maquina	IBM Eserver 325
S.O.	Linux Debian 2.4
Procesador:	Opteron dual 2 Gz
Ram:	5 Gb
Sistema de discos:	Fiber Chanel
Tarjeta de red	1000

2 Servidores para la aplicación dotLRN y servidor web AOLServer	
Maquina:	Clonico PC
S.O.	Linux Debian 2.4
Procesador:	Pentium IV 2'4 Ghz
Ram:	1 Gb
Sistema de discos:	IDE
Tarjeta de red	1000

En cuanto a la tecnología software de la plataforma debemos separar los diferentes servicios que hacen el funcionamiento de esta:

En primer lugar el servidor web.

El servidor web recomendado es AOLserver y es este el que se ha instalado porque AOLserver proviene de America Online, el mayor proveedor de Internet de Estados Unidos y esponsor del proyecto Mozilla (navegador web de código abierto). Inicialmente AOLserver era un servidor cerrado (propietario), pero gracias a las gestiones de Philip Greenspun se ha liberado bajo la licencia AOLserver Public License.

AOLserver ofrece muchas de las características de Apache: Es de código abierto, fácil de configurar, flexible, y tiene un API en C y Tcl para escribir módulos. Pero AOLserver tiene una arquitectura fundamentalmente diferente a la de Apache, que a menudo hace que sea una elección más eficiente. En particular, AOLserver incluye un intérprete de Tcl, arquitectura multihilos, un API database y un pool para conexión a la base de datos, y

un sistema para crear páginas dinámicas denominado ADP (AOLserver Dynamic Pages). Si el sitio web que se va a construir requiere numerosas conexiones a la base de datos, entonces merece la pena considerar AOLserver como alternativa a Apache.

Quizá sea por esta razón que no se ha utilizado Apache ya que actualmente se sabe y si se consultan las estadísticas de netcraft que es Apache con diferencia el servidor web más utilizado y el más seguro aunque en esto no podemos afirmarlo del todo porque al final si el administrador no esta al tanto de los posibles errores y modificaciones de seguridad, es decir, no se hace un mantenimiento del mismo ningún servidor es seguro.

En segundo lugar la Base de datos

Se puede montar sobre Oracle o Postgres, en principio por temas de licencias se opta por Postgres.

PostgreSQL se distribuye bajo una licencia flexible de BSD y es sin duda la base de datos más avanzada en la actualidad. Años de pruebas y miles de usuarios en todo en mundo lo demuestran.

Ventajas que ofrece PostgreSQL

Características: PostgreSQL tiene muchas de las funcionalidades de las bases de datos comerciales, y algunas que ni siquiera éstas tienen.

Funcionamiento: PostgreSQL se ejecuta en dos modos. El modo normal fsync, que guarda en el disco toda transacción completada, es más lento que la mayoría de las bases de datos comerciales, en parte porque son pocas las que hacen esto. El modo no-fsync, que es menos fiable que el anterior, es más rápido que las bases de datos comerciales, aunque en este modo un crash del sistema puede significar la pérdida de alguna transacción.

Fiabilidad: Los desarrolladores de PostgreSQL se preocupan de producir versiones que han estado al menos un mes en fase de beta testing, estables y con el menor número de bugs posible. Además, una comunidad enorme de usuarios y desarrolladores se preocupan de revisar constantemente de código y de publicar las mejoras.

Precio: PostgreSQL es gratuito, tanto para fines no comerciales como para fines comerciales. Lo mejor es que su futuro no está sujeto a las decisiones de ninguna empresa.

Soporte: Existe abundante documentación, listas de correo, acceso directo a los desarrolladores, empresas que ofrecen consultoría, y, lo más importante, el código fuente.

En tercer lugar el framework **ArsDigita Community System (ACS)**, el servidor de aplicaciones

Ya ha quedado claro que dotLRN va unido a OpenACS e instalando a partir de la versión 4.0 de OpenACS ya están integrados los paquetes de dotLRN.

Teniendo como origen el desarrollo de photo.net, se creó un conjunto de herramientas completamente modulares orientadas a dar servicios de colaboración a través de Internet. Estas herramientas se conocen con el nombre de **ArsDigita Community System (ACS)** cuando la base de datos utilizada es Oracle, y **OpenACS** cuando la base de datos es PostgreSQL. Es por tanto una base extensible sobre la que poder crear sitios web totalmente personalizados.

Una buena documentación es fundamental y los diferentes módulos de ACS no sólo están perfectamente documentados, sino que detrás de ellos existe una comunidad vibrante de desarrolladores y usuarios que continuamente revisan y documentan el código. Varios servicios de gran importancia en Internet están basados en ACS, entre ellos el sistema **Global Development Gateway Web** del Banco Mundial y la intranet del departamento de ventas de Siemens que engloba 10.000 empleados en 35 países

En cuarto lugar el S.O.

La elección ha sido clara, ya que se esta basando en software GPL se opta por instalar una versión estable de Debian, Kernel 2.4.

Todos los servidores y software arriba mencionados se instalan sobre el sistema operativo LINUX a través de una distribución de Debian. Este S.O. goza de mayor aprecio entre los expertos del campo que un servidor NT aunque aquí ni se llevo a plantear. Linux es un sistema operativo muy estable, escalable, seguro y que consume pocos recursos. Además, tiene la ventaja de que su coste es prácticamente nulo y de que está mantenido por una comunidad de desarrolladores y colaboradores muy activa. Para una empresa, instalar Linux en sus servidores, sin necesidad de mantenimiento,

para determinadas tareas, como servidores de ficheros, servidores web, proxys, firewalls, servicios de correo, etc, puede suponer un importante ahorro de costes y una mejora de la velocidad de la red en general. Linux cada vez está más preparado para actuar en un entorno gráfico atractivo y dispone de programas como WordPerfect o el nuevo StarOffice gratuitos y de calidad equivalente a Ms Office. Como ventaja adicional, tiene la gestión remota de todas las máquinas, facilidad de personalización (limitar sólo la ejecución de ciertas aplicaciones, realizar backups nocturnos), mantener la misma configuración se acceda desde el ordenador que se acceda, etc. Todo esto, unido a un menor consumo de recursos y mayor estabilidad que clientes Windows.

Los pasos seguidos en la instalación de la plataforma resulto de gran ayuda los descritos en la web oficial de OpenACS <http://openacs.org/doc/openacs-HEAD/install-steps.html> . Como referencia están muy bien explicados.

A modo resumen enumeramos el orden seguido:

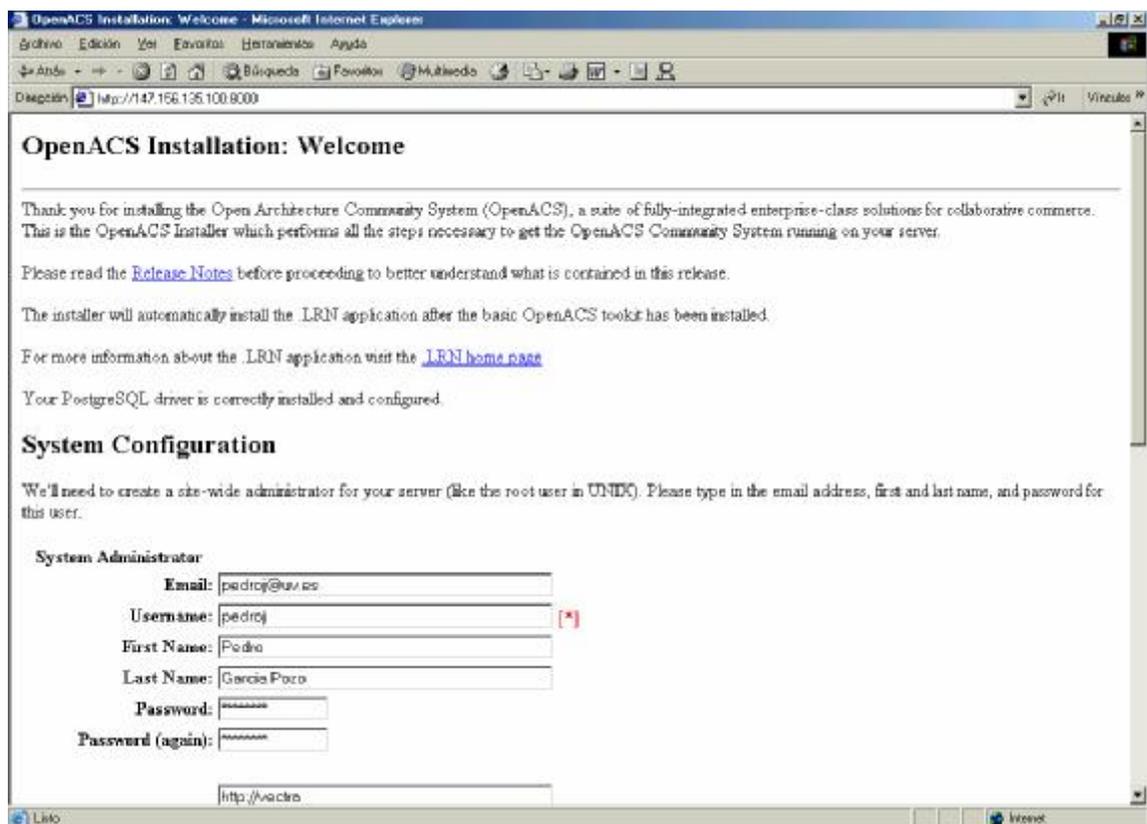
-
1. Instalar el S.O. en nuestro caso Debian versión de Linux.
 2. Instalar la BD en nuestro caso PostgreSQL v. 7.4.2
 3. Instalar el servidor web AOLServer 4.
 4. Instalar el framework OpenACS 5.0.4.
-

Una vez generados los ficheros de configuración se puede lanzar la plataforma y empezar a interactuar con ella.

6.2 Instalación de dotLRN

Para hacer la instalación de la plataforma, como comentamos ya anteriormente, de una manera ordenada y documentada hemos seguido las recomendaciones sobre la instalación de OpenACS en la web <http://openacs.org/doc/current/install-steps.html> . Una vez configurado los elementos tecnológicos (AoIserver, PostgreSQL, S.O elegido), y preparado el hardware necesario, se procede a su instalación.

Si hemos instalado el sistema con éxito la primera pantalla que nos devuelve el servidor, Figura 16, es un formulario para introducir algunos parámetros como son el nombre del superusuario de la plataforma, el mail que aparecerá en cualquier página de error, breve descripción del sistema. Una vez rellenamos estos campos se envía el formulario y el servidor empieza a crear toda la estructura de tablas en la BD, pasados unos minutos ya podemos lanzar de nuevo nuestra plataforma y ahora tendremos una página de bienvenida con dos casillas de autenticación, usuario y password.



OpenACS Installation: Welcome

Thank you for installing the Open Architecture Community System (OpenACS), a state of fully-integrated enterprise-class solutions for collaborative commerce. This is the OpenACS Installer which performs all the steps necessary to get the OpenACS Community System running on your server.

Please read the [Release Notes](#) before proceeding to better understand what is contained in this release.

The installer will automatically install the LRN application after the basic OpenACS toolkit has been installed.

For more information about the LRN application visit the [LRN home page](#)

Your PostgreSQL driver is correctly installed and configured.

System Configuration

We'll need to create a site-wide administrator for your server (like the root user in UNIX). Please type in the email address, first and last name, and password for this user.

System Administrator

Email:

Username:

First Name:

Last Name:

Password:

Password (again):

Ilustración 16. Primera pantalla tras la instalación.

A partir de este momento debemos empezar con la personalización del sitio;

1. Decidir el sistema de almacenamiento de los ficheros, si se hace en la BD o fuera de esta.
2. Crear las autoridades necesarias como serán las locales y configurar para que permita la validación LDAP.
3. Apariencia para que siga una relación con las web que ofrece la universidad.
4. Carga de usuarios, asignando su rol, (profesor, alumno, staff).
5. Carga de asignaturas y centros de la universidad.

Para los procesos 4 y 5 se han desarrollado procedimientos para que se realice de forma automatizada estas actividades, en el siguiente punto veremos como hemos conseguido esto.

Los usuarios cuando entren en el aula virtual (<http://pizarra.uv.es>) se encontrarán la pantalla de validación que les permitirá autenticarse en la comunidad universitaria ya que utiliza un servidor LDAP⁵ para hacer esto. Existen 3 autoridades creadas actualmente, UV, LOCAL, EXTERNO, por defecto aparece UV que es la que utilizara cualquier miembro de la comunidad.

Figura 17.



Ilustración 17. Pantalla de validación

⁵ LDAP ("Lightweight Directory Access Protocol")

Una vez se ha conseguido la validación te aparece tu portal personal donde puedes comenzar a trabajar.

6.3 Implementación

A continuación veremos los paquetes que se han desarrollado para hacer posible el funcionamiento de la plataforma y cumplir los requisitos descritos en este proyecto.

La principal preocupación es la integración de la BD de dotLRN con las bases de datos académicas que tiene la Universidad en DB2.

Para desarrollar código en OpenACS, tenemos que construir un paquete. Un paquete es una colección de paginas web, código en tcl, tablas y procedimientos. Un paquete con interface para el usuario se le llama aplicación, un paquete que provee de funciones a otros paquetes y no tiene una interface de usuario, se conoce como servicio. En dotLRN existe otra estructura que son los portlets, el sistema recoge la información que tiene cada usuario a cerca de cursos o comunidades y la muestra en unos portlets, para simplificar podemos decir que cualquier vista de un paquete es un portlet.

Un paquete puede ser instalado, actualizado, borrado. Se comunica con otros paquetes a través del API.

Como dijimos en nuestro análisis de requisitos los paquetes que necesitamos son los siguientes:

- Carga de usuarios, asignándoles el rol que les corresponda en la plataforma, profesor, alumno, personal de servicios.
- Carga de asignaturas, centros, cursos, períodos.
- Matricular los alumnos en los cursos a los que pertenece.
- Portlet para ver información de la asignatura.
- Portlet para visualizar información pública de los centros y asignaturas.

6.3.1 Carga de usuarios

El actualizar la BD mediante código es uno de los retos más importantes que abarca este proyecto, esto es así porque como hemos comentado más de una vez la plataforma no dispone todavía de ninguna herramienta para la carga masiva de usuarios. De modo que se ha tenido que analizar la BD del sistema y ver que tablas son las afectadas y que tipo de datos son los obligatorios para que la carga de usuarios se haga correctamente.

La carga de de usuarios son los primeros datos que se insertan en la BD, los usuarios se dan de alta como usuarios de openacs y a la vez como usuarios de dotlrn.

Las tablas afectadas son las siguientes: users y dotlrn_users.

```

-----
          Table "public.users"
-----
Column          |          Type          | Modifiers
-----
user_id         | integer                | not null
authority_id    | integer                |
username        | character varying(100) | not null
screen_name     | character varying(100) |
priv_name       | integer                | not null default 0
priv_email      | integer                | not null default 5
email_verified_p | boolean                | default true
email_bouncing_p | boolean                | not null default false
no_alerts_until | timestamp with time zone |
last_visit      | timestamp with time zone |
second_to_last_visit | timestamp with time zone |
n_sessions      | integer                | not null default 1
password        | character(40)           |
salt            | character(40)           |
password_question | character varying(1000) |
password_answer | character varying(1000) |
password_changed_date | timestamp with time zone |
auth_token      | character varying(100) |
Indexes:
  "users_pk" primary key, btree (user_id)
  "users_authority_username_un" unique, btree (authority_id, username)
  "users_screen_name_un" unique, btree (screen_name)
  "users_screename_lower_idx" btree (lower((screen_name)::text))
  "users_username_lower_idx" btree (lower((username)::text))
Foreign-key constraints:
  "users_user_id_fk" FOREIGN KEY (user_id) REFERENCES persons(person_id)
  "users_auth_authorities_fk" FOREIGN KEY (authority_id) REFERENCES auth_au
thorities(authority_id)
    
```

```

View "dotlrn_users"
-----
Column | Type | Modifiers
-----|-----|-----
rel_id | integer |
portal_id | integer |
theme_id | integer |
id | character varying(100) |
user_id | integer |
first_names | character varying(100) |
last_name | character varying(100) |
email | character varying(100) |
type | character varying(100) |
pretty_type | character varying(200) |
rel_type | character varying(100) |
group_id | integer |
segment_id | integer |
View definition: SELECT acs_rels.rel_id, dotlrn_user_profile_rels.portal_id, dotlrn_user_profile_rels.theme_id, dotlrn_user_profile_rels.id, users.user_id, persons.first_names, persons.last_name, parties.email, dotlrn_user_types."type", dotlrn_user_types.pretty_name AS pretty_type, dotlrn_user_types.rel_type, dotlrn_user_types.group_id, dotlrn_user_types.segment_id FROM dotlrn_user_profile_rels, dotlrn_user_types, acs_rels, parties, users, persons WHERE (((dotlrn_user_profile_rels.rel_id = acs_rels.rel_id) AND (acs_rels.object_id_one = dotlrn_user_types.group_id)) AND (acs_rels.object_id_two = parties.party_id)) AND (parties.party_id = users.user_id) AND (users.user_id = persons.person_id));
    
```

Una vez detectadas y analizadas las tablas se necesita saber que campos de estas son necesarios para que la inserción de los usuarios sea correcta.

De la tabla "Users" necesitamos rellenar al menos:

Username	Usuario
email	Dirección de correo
firstname	Nombre
lastname	Apellidos
tipouser	Tipo de usuario E,P,S
autoridad	Por defecto se les pone LDAP
password	Un valor aleatorio ya que valida LDAP
limitado	Si su acceso esta limitado, por defecto NO
email_verified_p	Se pone a true, correo verificado

De la tabla "dotlrn_users" necesitamos rellenar al menos:

User_id	Id del usuario en la tabla users
Type	Rol, estudiante, profesor, secretaria, admin

Para la que la inserción de los datos sea correcta, y las relaciones entre las tablas se mantenga se han tenido que seguir un orden en el proceso de carga de los datos y actualización de las tablas.

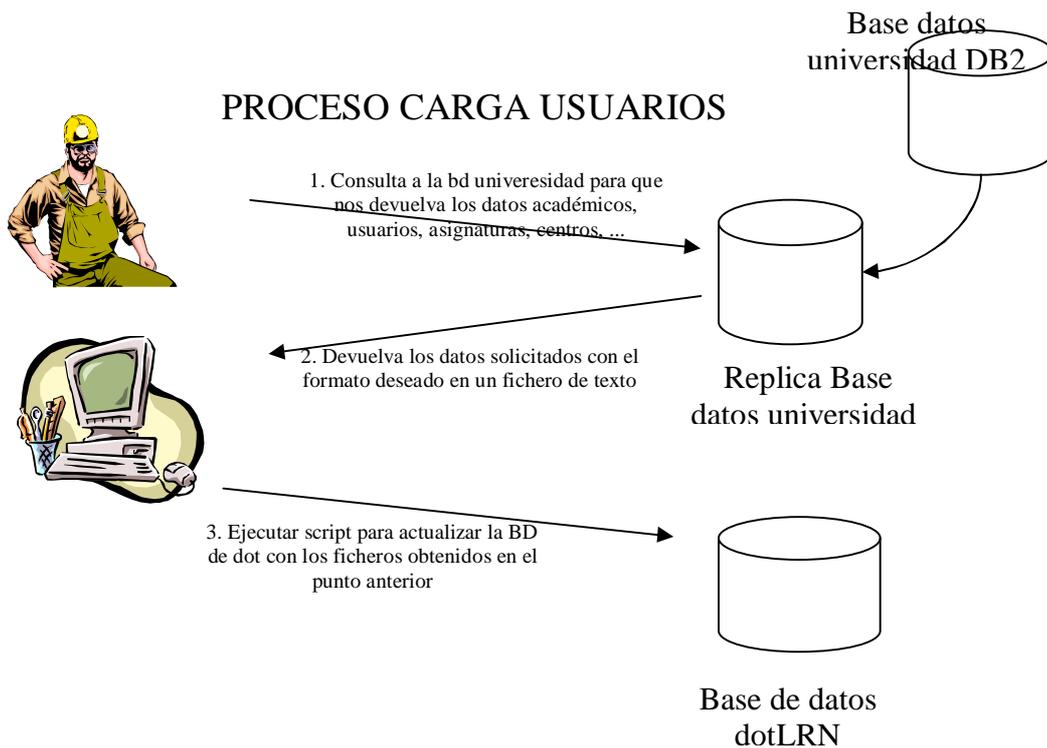
La ejecución de los diferentes scripts a la hora de insertar en las tablas de la BD la actualización debe seguir este orden:

0	Usuarios
1	Períodos (Terms)
2	Departamentos (Centros)
3	Asignaturas
4	Cursos

Al inicio del cuatrimestre se han cargado en la plataforma todos los centros, los usuarios, se han creado los períodos, las asignaturas y a petición de los profesores se crean los cursos, proceso que denominaremos "matricular en curso".

El lenguaje de programación empleado para la generación de los scripts ha sido el perl (Practical Extraction and Report Language), lenguaje de scripts que corre en la mayoría de los sistemas hoy disponibles, aporta una gran flexibilidad en el tratamiento de cadenas, a parte de ser un lenguaje muy difundido en la comunidad de programadores de software libre.

Veamos como se hace la carga de usuarios.



1. Partimos de que tenemos un fichero de texto con el "username" de todos los usuarios de la Universidad que queremos introducir en la plataforma.

```

carrasc
arrillag
cperez
pascuall
doris
oliver
capafons
dasic
estebanc
.....
    
```

2. Después se hace la consulta a bancuv y se obtienen todos los datos referentes a cada usuario para insertar en la tabla "users".
3. Se ejecuta el script "USER_ALL.PL" que se encarga de ir pasando una serie de ficheros para cargar todos los usuarios en la BD de la plataforma.

Los ficheros contienen la siguiente información:

```

USUARIO
sergongo
sergongo@alumni.uv.es
Sergio
Gonzalez Gomez
E
ldap
pwpwpwpw
0
mapecos
mapecos@alumni.uv.es
Maria Isabel
Pellicer Costa
E
ldap
pwpwpwpw
0
paulo
paulo@alumni.uv.es
Paula
Lorente Rius
E
ldap
pwpwpwpw
0
    
```

4. La llamada que se encarga de ejecutar el código en tcl que hay en el fichero results.tcl y que esta contenido en un paquete llamado siuadmin que se ha creado para este propósito, es esta, \$cmd = `"/usr/bin/wget -a $mensajes -O - http://pizarra1.uv.es/siuadmin/results.tcl?scriptname=\$pathdata/user_\$usuarios.cmd".`

El aspecto de "USER_ALL.PL" es el siguiente:

```
#!/usr/bin/perl
#cd /u/soft/GROUPWARE/SCRIPTS/PAQUETES
#rm mensajes.txt output.txt
#./user_all.pl >> output.txt

use strict;

my ($path, $contador, $pathdata, $mensajes, $fileoutput, $user, $fn, $linea, $cmd)
$path = "/u/soft/GROUPWARE/SCRIPTS/PAQUETES/";
$pathdata = $path . "DATA/";
$mensajes = $path . "mensajes.txt";
$fileoutput = $path . "output.txt";
$fn = $path . "USUARIOS.txt";

$contador = 0;
open (FH, "<$fn") or die "Can't open '$fn': $!";
while (<FH>)
{
  ++$contador;
  $linea = $_;
  chomp $linea;
  $user = $linea;

  if (index ($user, "#") != 0) {
    print "$contador Cogiendo la informacion del usuario '$user'\n";
    $cmd = $path . "coge_bancuv.sh " . $user;
    #print $cmd . "\n";
    system ($cmd);

    print "...troceando la informacion\n";
    $cmd = $path . "corta_file_user.pl " . $pathdata . "user_" . $user . ".dat";
    #print $cmd . "\n";
    system ($cmd);

    print "...ejecutando el script en pizarra con wget\n";
    $cmd = "/usr/bin/wget -a $mensajes -O - http://pizarra1.uv.es/siuadmin/results.tcl?scriptname=\$pathdata/user\_\$user.cmd";
    #print $cmd . "\n";
    system ($cmd);
  }
  else
  {
    #print "$contador Skipping el usuario '$user'\n";
  }
}

close (FH);
```

5. Parte del fichero results.tcl donde se hace la inserción de los usuarios es el siguiente:

```

ad_page_contract {

    @cvs-id $Id: index.tcl,v 1.16 2003/10/16 20:45:43 lars Exp $
} -query {
    scriptname:notnull
} -properties {
    context:onevalue
    subsite_name:onevalue
    subsite_url:onevalue
    nodes:multirow
    admin_p:onevalue
    user_id:onevalue
    show_members_page_link_p:onevalue
}

set main_site_p [string equal [ad_conn package_url] "/"]
set page_title [ad_conn instance_name]
set context [list]
set package_id [ad_conn package_id]
#dot2 set admin_p [permission::permission_p -object_id $package_id -party_id [ad_conn untrusted_user_id] -privilege admin]
set admin_p [permission::permission_p -object_id $package_id -party_id [ad_conn authuser] -privilege admin]
set redirect_url "error.tcl"

set theIP [ns_conn peeraddr]

#DEJO HACER SIN VALIDAR
if {[string equal $theIP "147.156.2.56"] == 1 || [string equal $theIP "147.156.9.2"] == 1 || [string equal $theIP "147.156.9.3"] == 1 || [string equal $theIP "147.156.9.4"] == 1 || [string equal $theIP "147.156.9.5"] == 1} {
    ns_log Notice [format "Entrando sin validar en SIUVADMIN desde IP=%s" $theIP]
} else {
    if {!$admin_p} {
        ns_log Notice [format "Error entrando validando en SIUVADMIN desde IP=%s" $theIP]
        set error "No eres administrador"
        ad_returnredirect [append tmp $redirect_url "?error=" $error]
        ad_script_abort
    }
}

set user_id [ad_conn user_id]
set subsite_name [ad_conn instance_name]
set subsite_url [subsite::get_element -element url]

if {[file exists $scriptname] == 0} {
    set error "Fichero no existe o no puede abrirlo el usuario OPENACS"
    ad_returnredirect [append tmp $redirect_url "?error=" $error]
    ad_script_abort
}

randomInit [ns_time]

ad_proc lee { fd } {
    gets $fd loclinea
    while {[string length $loclinea] > 0 && [string index $loclinea 0] == "#"} {
        gets $fd loclinea
    }
    return $loclinea
}

ad_proc lee2 { fd linea } {
    upvar linea loclinea
    set res [gets $fd loclinea]
    while {[string length $loclinea] > 0 && [string index $loclinea 0] == "#"} {
        set res [gets $fd loclinea]
    }
    return [string length $loclinea]
}

```

```

}

set contmatresult 1
set linea ""
set IsCommand 0

#ABRO EL FICHERO DE SCRIPT
set fscript [open $scriptname r]
while {[!ee2 $fscript $linea] > 0} {

if {[string first "_" $linea] == 0} {
#ES UN COMANDO
set IsCommand 1
set modo $linea
} else {
#ABRO EL FICHERO DE DATOS
ns_log Notice [format "ABRIENDO FICHERO: %s" $linea]
set IsCommand 0
set fd [open $linea r]
set conttotal 1
set contbien 0
set contmal 0
set modo [lee $fd]
}

ns_log Notice [format "ENTRANDO EN: %s" $modo]
ns_log Notice "-----"

#####
#USUARIO
#####
if {[string equal -nocase $modo "USUARIO"] == 1} {
while {[!ee2 $fd $linea] > 0} {
set username $linea ;#lopezj
set email [lee $fd] ;#lopezj@uv.es
set firstname [lee $fd] ;#Agustin
set lastname [lee $fd] ;#Lopez Bueno
set tipouser [lee $fd] ;#E
set autoridad [lee $fd] ;#ldap
set password [lee $fd] ;#pwpwpw
set limitado [lee $fd] ;#0 NO limitado, 1 SI limitado

set email_verified_p 0
set read_private_data_p read_private_data

#BUSCO LA AUTORIDAD LDAP
if {[catch {db_1row lautoridad "select authority_id from auth_authorities where short_name = :autoridad"} result]} {
} else {
}

#USER DE OPENACS
set user_info(username) $username
set user_info(email) $email
set user_info(last_name) $lastname
set user_info(first_names) $firstname
set user_info(authority_id) $authority_id
array set xresult [auth::create_local_account -authority_id $authority_id -username $username -array user_info]
switch $xresult(creation_status) {
ok {

#USER DE DOT: necesito el id numerico del usuario. Roles en DOTLRN: student, professor, external, admin

db_1row user_id "select user_id from users where username = :username"

#AJUSTO LA PW
if {[string equal password "" ] != 1} {
ad_change_password $user_id $password
}

acs_privacy::set_user_read_private_data -user_id $user_id -object_id [dotlrn::get_package_id] -value
Sread_private_data_p

if {[string equal -nocase $tipouser "E"] == 1} {
if {$limitado == 0} {
set result [dotlrn::user_add -type student -can_browse -user_id $user_id]
} else {

```

```

        set result [dotlrn::user_add -type student -user_id $user_id]
    }
} elseif {[string equal -nocase $tipouser "P"] == 1} {
    if {$limitado == 0} {
        set result [dotlrn::user_add -type professor -can_browse -user_id $user_id]
    } else {
        set result [dotlrn::user_add -type professor -user_id $user_id]
    }
} elseif {[string equal -nocase $tipouser "A"] == 1} {
    if {$limitado == 0} {
        set result [dotlrn::user_add -type admin -can_browse -user_id $user_id]
    } else {
        set result [dotlrn::user_add -type admin -user_id $user_id]
    }
} elseif {[string equal -nocase $tipouser "X"] == 1} {
    if {$limitado == 0} {
        set result [dotlrn::user_add -type external -can_browse -user_id $user_id]
    } else {
        set result [dotlrn::user_add -type external -user_id $user_id]
    }
}
}

set guest_p "f"
dotlrn_privacy::set_user_guest_p -user_id $user_id -value $guest_p

ns_log Notice [format "OK EN dotlrn::user_add: %d, '%s' '%s' '%s' '%s' '%s'" $conttotal $username $email $lastname
$firstname $limitado]
set contbien [expr $contbien + 1]
}
default {
    ns_log Notice [format "ERROR EN dotlrn::user_add: %d, %s, %s, %s, %s, %s" $conttotal $username $email $limitado
$result(creation_status) $result(creation_message)]
    set contmal [expr $contmal + 1]
    foreach { elm_name elm_error } $result(element_messages) {
        ns_log Notice $elm_name $elm_error
    }
}
}
}
set conttotal [expr $conttotal + 1]

```

6.3.2 Carga de asignaturas, centros y períodos

Para la carga de asignaturas, centros y períodos las tablas que hay que actualizar son las siguientes; "dotlrn_classes", "dotlrn_departaments", "dotlrn_terms".

La estructura de las tablas es la siguiente:

Asignaturas, dotlrn-classes;

Column	Type	Modifiers
class_key	character varying(100)	
pretty_name	character varying(100)	
description	character varying(4000)	
package_id	integer	
supertype	character varying(100)	
url	character varying	
department_key	character varying(100)	

View definition: SELECT dotlrn_classes.class_key, dotlrn_community_types.pretty_name, dotlrn_community_types.description, dotlrn_community_types.package_id, dotlrn_community_types.supertype, (SELECT site_node_url(site_nodes.node_id) AS site_node_url FROM site_nodes WHERE (site_nodes.object_id = dotlrn_community_types.package_id)) AS url, dotlrn_classes.department_key FROM dotlrn_classes, dotlrn_community_types WHERE (dotlrn_classes.class_key = dotlrn_community_types.community_type);

Centros, dotlrn_departments;

Column	Type	Modifiers
department_key	character varying(100)	
pretty_name	character varying(100)	
description	character varying(4000)	
package_id	integer	
supertype	character varying(100)	
url	character varying	
external_url	character varying(4000)	

View definition: SELECT dotlrn_departments.department_key, dotlrn_community_types.pretty_name, dotlrn_community_types.description, dotlrn_community_types.package_id, dotlrn_community_types.supertype, (SELECT site_node_url(site_nodes.node_id) AS site_node_url FROM site_nodes WHERE (site_nodes.object_id = dotlrn_community_types.package_id)) AS url, dotlrn_departments.external_url FROM dotlrn_departments, dotlrn_community_types WHERE (dotlrn_departments.department_key = dotlrn_community_types.community_type);

Períodos, dotlrn_terms;

Column	Type	Modifiers
term_id	integer	not null
term_name	character varying(20)	not null
term_year	character varying(9)	not null
start_date	date	not null default now()
end_date	date	not null default (now() + '180 days')

Primary key: dotlrn_terms_pk
Triggers: RI_ConstraintTrigger_128302, RI_ConstraintTrigger_128304

Una vez detectadas y analizadas las tablas se necesita saber que campos de estas son necesarios para que la inserción de los usuarios sea correcta.

De la tabla "dotlrn_classes" asignaturas, debemos actualizar al menos estos datos:

asign_name	Código de la asignatura
dep_key	Clave del centro al que pertenece la asignatura
asign_description	Nombre de la asignatura

De la tabla "dotlrn_departments" centros, debemos actualizar al menos estos datos:

dep_name	Código del centro
dep_description	Nombre de la facultad o centro
dep_external_url	Url del centro

De la tabla "dotlrn_terms" periodos, debemos actualizar al menos estos datos:

term_name	Nombre del período
term_year	año
start_date	Fecha inicio del período
end_date	Fecha fin del período

Para la carga de todos estos datos se sigue el mismo orden y se utilizan los mismos script's descritos en la carga de usuarios solo que cambiamos el nombre del fichero que le pasamos al fichero "results.tcl"

Es decir, primero consulta a BD de bancuv para obtener todos los datos, una vez tenemos los datos con el formato requerido, solo nos falta invocar el fichero "results.tcl" pasando los ficheros obtenidos.

Veamos el fichero de asignaturas que nos devuelve bancuv;

```
ASIGNATURA
09567
c001
Fund.Tecnologías Cuánticas (6 Cr.)
09568
c001
Organ.Administ.Empresas (6 Cr.)
09594
c001
Hª.Ciencias y Técnicas (4,5 Cr.)
09644
c001
Teledet.Recurs.Natur. (3 Cr.)
09851
c001
```

Veamos el fichero de centros "departaments", que nos devuelve bancuv;

```
DEPARTAMENTO
c001
FACULTAT DE FÍSICA
http://www.uv.es
c002
FACULTAT DE QUÍMICA
http://www.uv.es
c003
FAC. DE DRET
http://www.uv.es
```

Veamos el fichero de períodos "terms", que nos devuelve bancuv;

```
PERIODO
2003-04
2004
2003-09-01
2004-08-31
```

Vistas del fichero "results.tcl", donde se insertan las asignaturas, centros y períodos.

```
#####
#TERM o PERIODO
#####
} elseif {[string equal -nocase $modo "PERIODO"] == 1} {
  while {[lindex $fd $linea] > 0} {
    set term_name $linea ;#2002-03
    set term_year [lindex $fd] ;#2003
    set start_date [lindex $fd] ;#2002-09-01
    set end_date [lindex $fd] ;#2003-08-31

    #TERM DE DOT
    if {[catch {dotlrn_term::new -term_name $term_name -term_year $term_year -start_date $start_date -end_date $end_date}
result]} {
      #ERROR
      ns_log Notice [format "ERROR EN dotlrn_term::new--> %d, '%s' '%s' '%s' '%s'" $conttotal $term_name $term_year
$start_date $end_date]
      set contmal [expr $contmal + 1]
    } else {
      #NO ERROR
      ns_log Notice [format "OK EN dotlrn_term::new: %d, '%s' '%s' '%s' '%s'" $conttotal $term_name $term_year $start_date
$end_date]
      set contbien [expr $contbien + 1]
    }
    set conttotal [expr $conttotal + 1]
  }
#####
#DEPARTAMENTO
#####
} elseif {[string equal -nocase $modo "DEPARTAMENTO"] == 1} {
  while {[lindex $fd $linea] > 0} {
    set dep_name $linea ;#c001
    set dep_description [lindex $fd] ;#FACULTAT DE FISICA
    set dep_external_url [lindex $fd] ;#http://www.uv.es

    #DEPARTAMENTO DE DOT
    if {[catch {dotlrn_department::new -pretty_name $dep_name -description $dep_description -external_url $dep_external_url}
result]} {
      #ERROR
      ns_log Notice [format "ERROR EN dotlrn_department::new--> %d, '%s' '%s' '%s' '%s'" $conttotal $dep_name $dep_description
$dep_external_url]
      set contmal [expr $contmal + 1]
    } else {
      #NO ERROR
      ns_log Notice [format "OK EN dotlrn_department::new: %d, '%s' '%s' '%s' '%s'" $conttotal $dep_name $dep_description
$dep_external_url]
      set contbien [expr $contbien + 1]
    }
    set conttotal [expr $conttotal + 1]
  }
#####
#CLUB
#####
} elseif {[string equal -nocase $modo "CLUB"] == 1} {
  while {[lindex $fd $linea] > 0} {
    set club_name $linea
    set club_description [lindex $fd]
    set club_join_policy [lindex $fd]

    #CLUB DE DOT (es lo que usan para COMUNIDAD)
    if {[catch {dotlrn_club::new -pretty_name $club_name -description $club_description -join_policy $club_join_policy} result]}
{
      #ERROR
      ns_log Notice [format "ERROR EN dotlrn_club::new --> %d, '%s' '%s' '%s' '%s'" $conttotal $club_name $club_description
$club_join_policy]
      set contmal [expr $contmal + 1]
    } else {
      #NO ERROR
      ns_log Notice [format "OK EN dotlrn_club::new: %d, '%s' '%s' '%s' '%s'" $conttotal $club_name $club_description
$club join policv]
    }
  }
#####
```

6.3.3 Matricular los alumnos y generar los cursos a los que pertenece

Para crear los cursos, necesitamos que la información anterior este cargada en la BD.

Como hemos comentado anteriormente la generación de cursos para este cuatrimestre se ha hecho bajo demanda, esto quiere decir que los profesores que han querido dar sus asignaturas utilizando la plataforma nos han enviado una solicitud en esta dirección <http://solicitudes.uv.es> diciendo quien son y que desean tener la docencia activada.

Una vez recibida la solicitud, seguimos el mismo mecanismo antes descrito.

Con el "username" del profesor, hacemos una petición a bancuv y nos devuelve toda la información académica de este; por ejemplo del profesor cerveron nos devuelve un fichero de texto con la siguiente estructura.

* Curso – Etiqueta para que nuestro script haga lo que hemos programado.
* Asignaturas que da ese profesor
* Subgrupos, que son los laboratorios que da.
* Usuarios que están matriculados en ese curso, nos devuelve los alumnos que han pagado la matricula y los profesores responsables.
* Periodos
* Calendarios de la asignatura, fechas de exámenes

Una vez tenemos el fichero generado y con toda la información bien formateada solo debemos invocar el fichero "results.tcl" y se generará el curso correspondiente con todos los usuarios y profesores que lo forman.

A partir de ese momento el profesor ya puede subir apuntes, utilizar los foros, etc. Y los alumnos ya pueden ver que al entrar en su portal están matriculados en esa asignatura.

6.3.4 Portlet's para visualizar información de las asignaturas

Siempre que se crea un paquete como se explico en el capitulo de descripción del sistema se crean unos directorios por defecto en los que debemos añadir nuestro código fuente. La creación de un portlet simplemente es la vista de un paquete, más formalmente es aquella información que le devuelve el sistema al usuario por medio de estos portlets.

Se han creado 3 portlets para realizar cada una de las diferentes vistas de la información de las asignaturas y centros de toda la comunidad Universitaria. Son el aisginfo, dotlrn-asiginfobig, dotlrn-asiginfosmall.

En cada paquete se crean 3 directorios por defecto: sql, tcl, www, y un fichero de información donde se puede ver la versión del paquete, autor, sitio, etc. En el directorio sql colocaremos los scripts para la creación de las tablas en la BD, según el soporte elegido tenemos un directorio para postgres u Oracle. Dentro del directorio www si desarrollamos paginas web colocamos los ficheros que se ejecutaran al invocar el paquete como son el "index.tcl, index.adp" y los ficheros para tratamientos de errores.

Para que un paquete se convierta en un portlet los pasos que debemos seguir son los siguientes:

- Instalar el paquete y coger la clave generada en el fichero "nombredelpaquete".info
- Editar los parámetros del sitio y añadir esta clave en el portal de usuarios para que lo añada y lo puedan ver los usuarios en su portal
- Modificar las plantillas del sitio para añadirlo.

Vista paso 1:

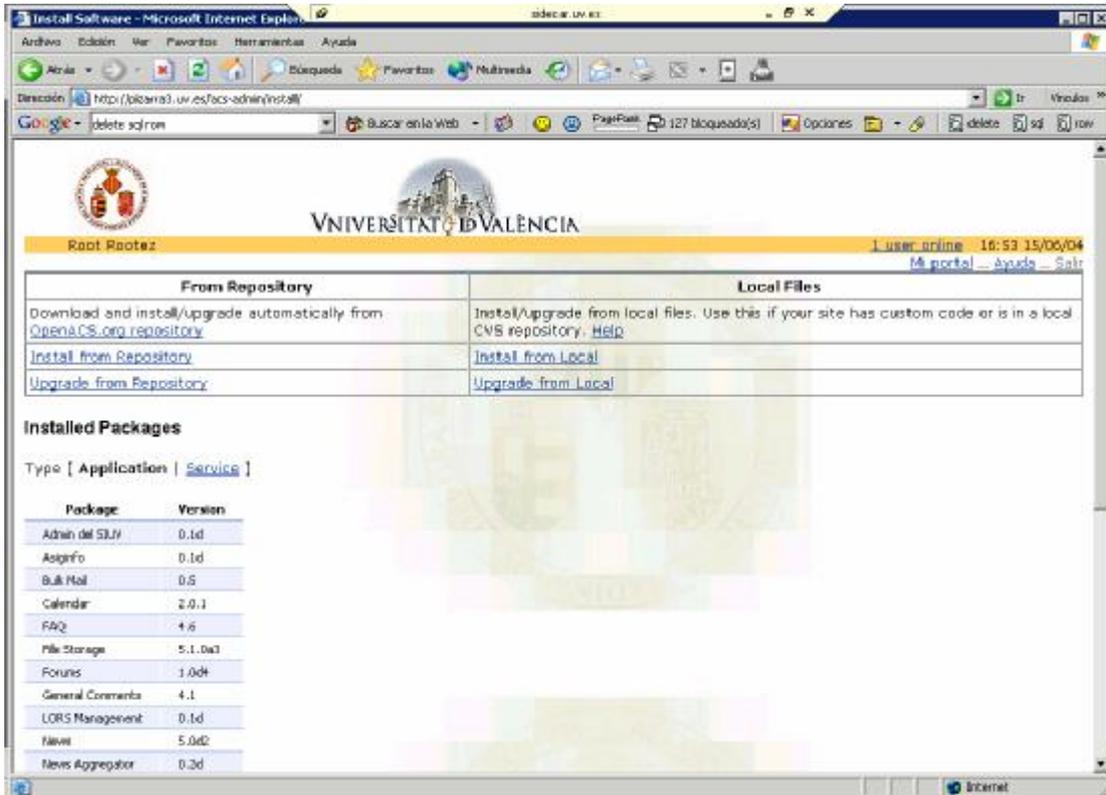


Ilustración 18. Instalar paquete desde modo local

Vista paso 2:

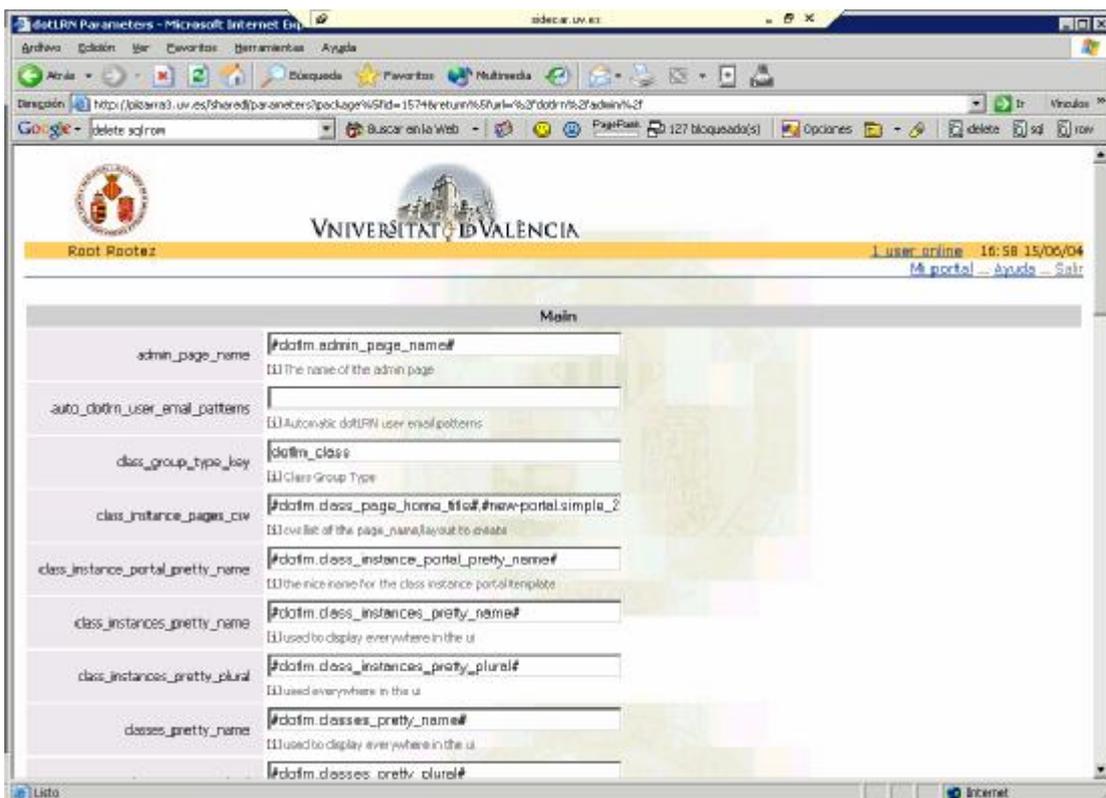


Ilustración 19. Editar parámetros del sitio

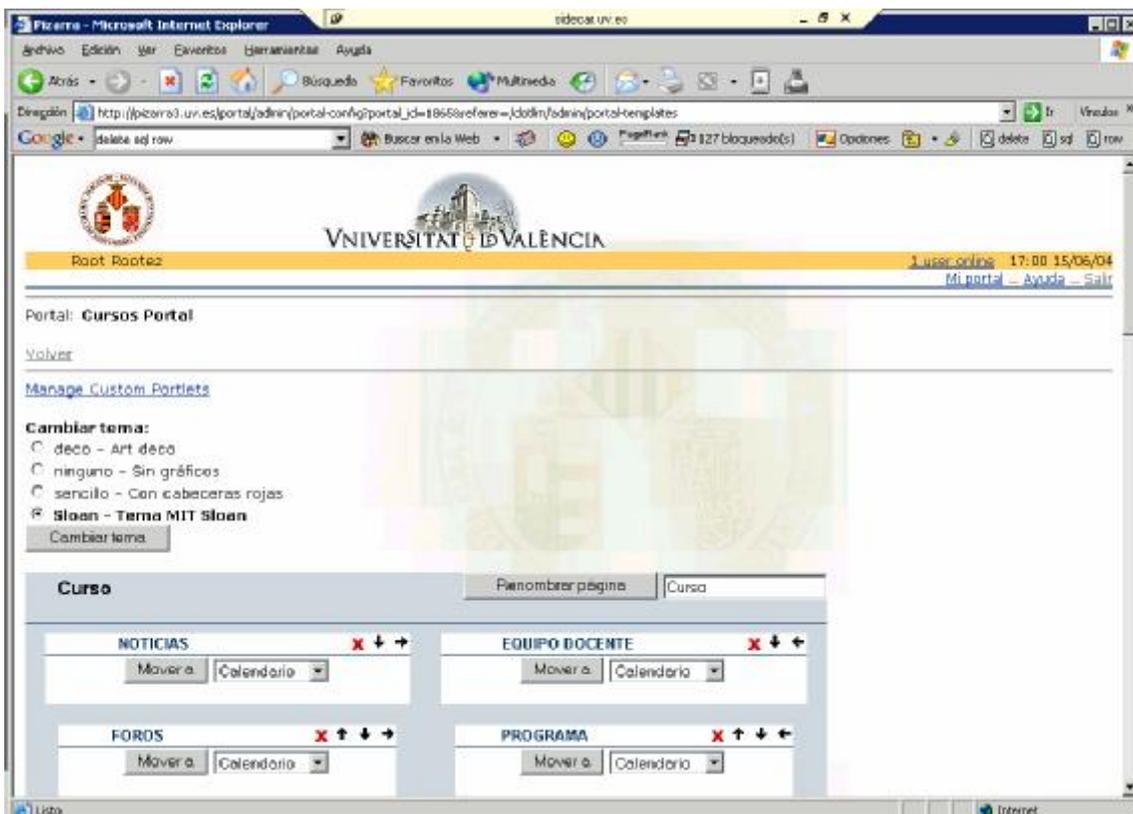
Vista paso 3:

Ilustración 20. Añadir portlets en la plantilla.

Pasaremos a explicar que hace cada portlet. Funcionalmente solo se encargan de interpretar el html que le devuelve el servidor bancuv, ya que se ha decidido de momento que sea el propio servidor de datos el que devuelve toda la información ya con formato. De esta manera solo nos preocupamos de la validación y comprobación de usuario.

Veamos que hace el paquete "dotlrn-asigninfosmall". Este se encarga de la visualización de información a cerca de la asignatura como es, el nombre, número de créditos, grupos, alumnos, etc, toda esta información se carga de forma dinámica. Veamos una imagen de esta información. El portlet es el llamado información breve.

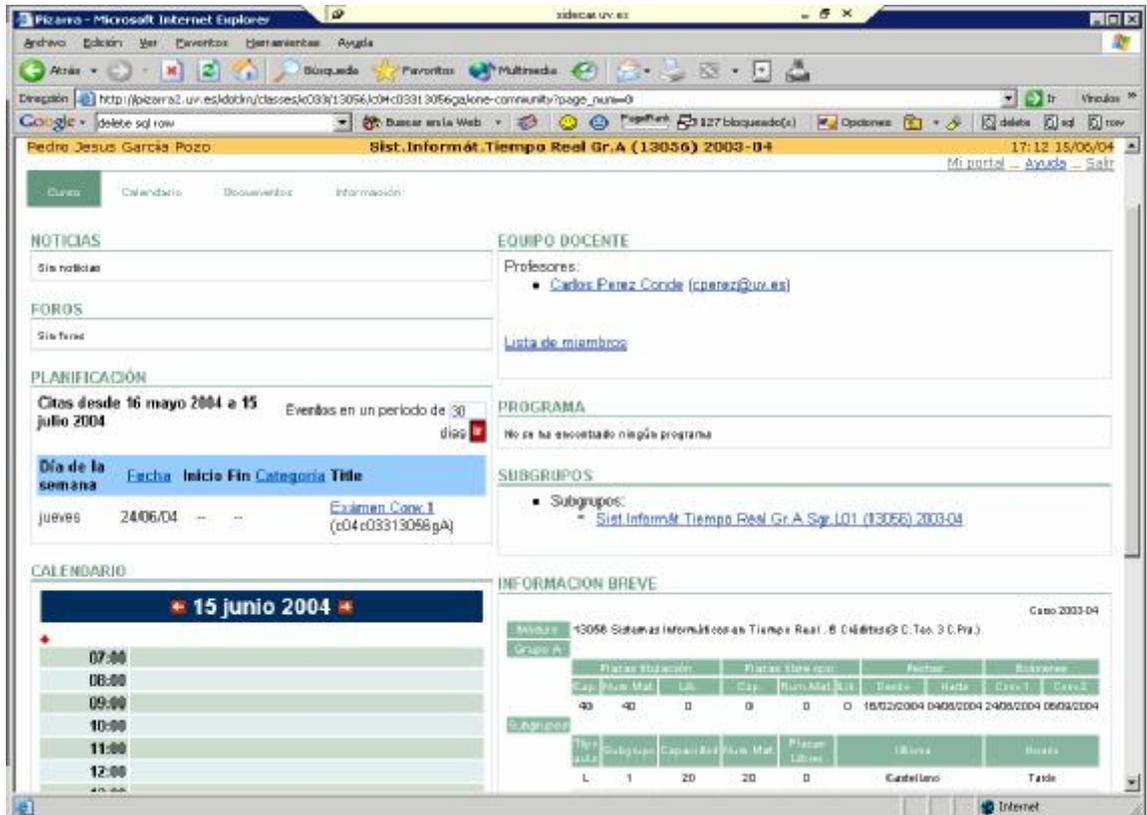


Ilustración 21. Vista portlet información breve.

Como hemos dicho que el servidor de datos nos devuelve la información con formato, el código que mostramos abajo es muy sencillo, siempre que se invoca un paquete se ejecuta por defecto el index.tcl, que recoge unos parámetros y se los pasa al index.adp que es el encargado de procesar dicha información y devolverla al navegador para que la interprete, en nuestro caso el fichero index.adp solamente se encarga de invocar una plantilla para que la utilice con la información recibida.

Veamos los 2 ficheros, el "Index.tcl":

```
#dotlrn-asiginfobig/www/asiginfobig-portlet.tcl
ad_page_contract {
  The display logic for the asiginfobig portlet

  @version $Id: asiginfobig-portlet.tcl,v 0.1 2004/02/02 18:05:05 peterm Exp $
} -properties {
  asiginfobig_id:onevalue
}
array set config $cf
set community_id [dotlrn_community::get_community_id]

set user_id [ad_conn user_id]
set lang [ad_get_cookie ad_locale "es_ES"]
if {[catch {db_1row lang "select coalesce(locale, 'es_ES') as lang from user_preferences up, users us where up.user_id=us.user_id and
us.user_id=:user_id"} result]} {
  #ERROR
  set lang [ad_get_cookie ad_locale "es_ES"]
} else {
}

set timeout 10
set depth 3
set redirect_url "error.tcl"

set cod [dotlrn_community::get_community_name $community_id]
set url [format "http://bancuv.uv.es/pls/uv0/oca.muestra_asignatura_bigapplet?lang=%s&cod=%s&tipo=G" $lang $cod]

set contenturl [ns_httpget $url $timeout $depth]

if {[string first "PL/SQL:" $contenturl] > 0} {
  set contenturl [dotlrn_community::get_community_description -community_id [dotlrn_community::get_community_id]]
}
```

El fichero "Index.adp":

```
<%
#dotlrn-asiginfosmall/www/asiginfosmall-portlet.adp

%>

<if @config.shaded_p@ ne "t">
  @contenturl;noquote@
</else>
```

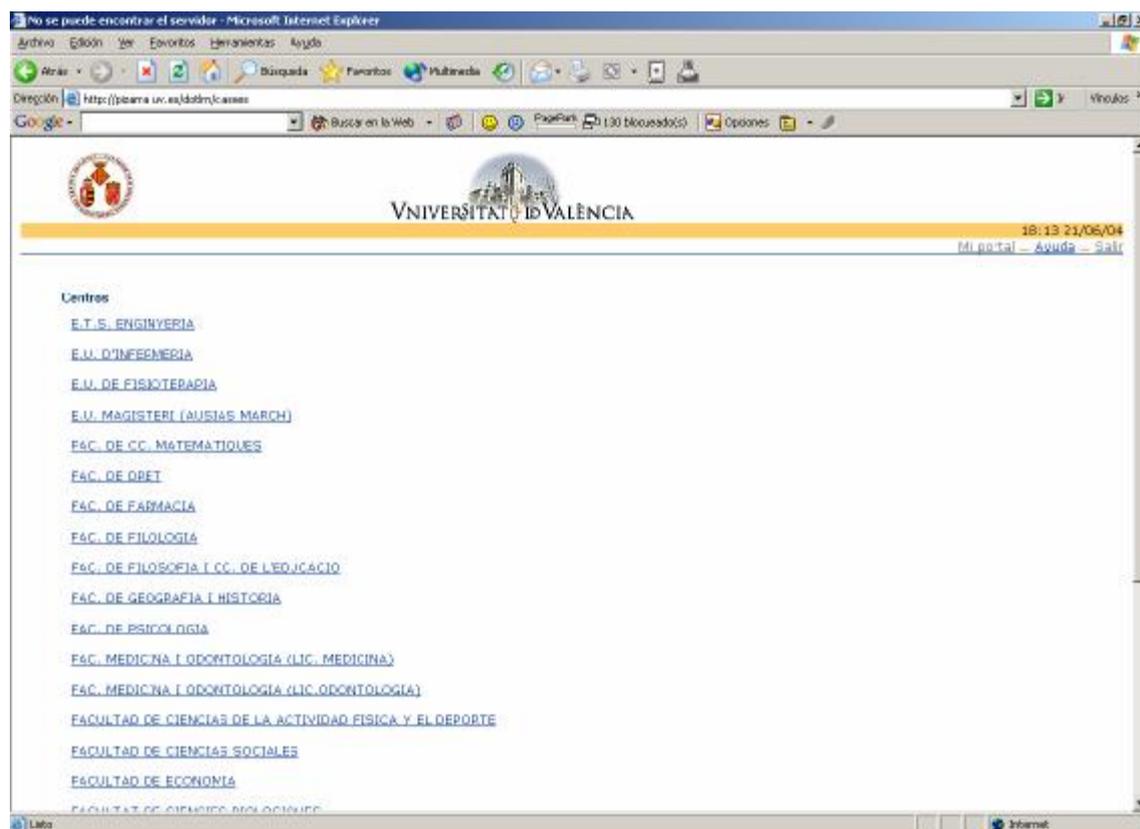
El portlet "dotlrn-asiginfo-big" hace lo mismo que el anterior solo que devuelve un formato más grande y lo que hacemos es mostrar esa información en una página más dentro de la asignatura.

El último portlet lo que intenta es mostrar la información pública sin tener que validarse en la plataforma, esta tarea la hace la aplicación "asiginfo", este paquete está montado bajo la capa de OpenACS ya que de esta manera se puede acceder a él sin necesidad de validación alguna, esto es así por se pretende que esta información esté disponible para la gente de fuera de la comunidad universitaria y además pronto que enlazara con la web oficial de estudiantes.

La dirección para ver esta información es la siguiente:

<http://pizarra.uv.es/dotlrn/classes>

Y la vista de esta información es la siguiente:



Nuestro prototipo solo se encarga de enviar al navegador la información que le devuelve el servidor de bases de datos. El código es muy similar al anterior. El fichero "Index.tcl".

```
ad_page_contract {
  @cvs-id $Id: index.tcl,v 1.16 2003/10/16 20:45:43 lars Exp $
} {
  format:naturalnum,notnull
  cmd:notnull
  cod:notnull
  tipo:notnull
  asigurl:
} -properties {
}

set lang [ad_get_cookie ad_locale "es_ES"]
set timeout 10
set depth 3
set redirect_url "error.tcl"

set user_id [ad_conn user_id]
if {$user_id <= 0} { #NO AUTENTICADO
  set enviarcabeceras "SI"
} else {
  set enviarcabeceras "NO"
}

if {$format == 1} {
  if {[string equal -nocase $cmd "infoasig"] == 1} {
    #A=asignatura G=grupo
    if {[string equal -nocase $tipo "A"] == 1} {
      set url [format
"http://bancuv.uv.es/pls/uv0/oca.muestra_asignatura?lang=%s&cod=%s&tipo=%s&enviarcab=
%s" $lang $cod "A" $enviarcabeceras]
    } elseif {[string equal -nocase $tipo "G"] == 1} {
      set url [format
"http://bancuv.uv.es/pls/uv0/oca.muestra_asignatura?lang=%s&cod=%s&tipo=%s&enviarcab=
%s" $lang $cod "G" $enviarcabeceras]
      ns_log Notice [format "url=%s" $url]
    } else {
      set error "Invalid parameter tipo!"
      ad_returnredirect [append tmp $redirect_url "?error=" $error]
      ad_script_abort
    }
  } else {
    set error "Invalid parameter cmd!"
    ad_returnredirect [append tmp $redirect_url "?error=" $error]
    ad_script_abort
  }
} else {
  set error "Invalid paramater format!"
  ad_returnredirect [append tmp $redirect_url "?error=" $error]
  ad_script_abort
}

set contenturl [ns_httpget $url $timeout $depth]

if {[string first "PL/SQL: error" $contenturl] > 0} {
  ad_returnredirect $asigurl
  ad_script_abort
}
```

Y el fichero "Index.adp":

```
<%
#asiginfo/www/index.adp
%>

<master>
<%
if { @enviarcabeceras@ eq "SI" } {
%>
<link rel="stylesheet" type="text/css" href="/resources/dotlrn/dotlrn-master.css" media="all">
<%
}
%>

@contenturl;noquote@

<%
for {set key 1} {$key <= [array size matresult]} {incr key} {
  adp_puts "[array get matresult $key]<BR>"
}
%>
```

6.4 Personalización de dotLRN

A partir de ahora intentaremos ver que apariencia tiene la plataforma tras su personalización a la universidad, que módulos son los que se han lanzado primeramente para la fase piloto, interfaces de usuario, funcionalidades, administraciones de cursos, uso de la plataforma en general.

6.4.1 Características generales

Los módulos y paquetes más significativos que se han instalado en la versión de dotLRN para la Universidad son los siguientes:

1. Una nueva versión del paquete de autenticación que permite autenticar usuarios de diversos modos:
 - a. Tiene una base de datos que permite mecanismos de autenticación internos y externos.
 - b. Es posible configurar múltiples autoridades de autenticación, y se pueden clasificar. Actualmente permite módulos pluggable Unix (PAM). (que permite que el administrador del sistema elija cualquier de los métodos de autenticación - e.j. RADIO, Kerberos, IMAP, ESTALLIDO, AFS, etc.)
 - c. El módulo en LRN LDAP gestiona la sincronización en tiempo real del usuario. La infraestructura externa de la autenticación es modular para facilitar posibles gestores futuros.
 - d. Las conexiones del usuario se pueden cifrar con el SSL.
 - e. El sistema tiene una opción del recordatorio y/o del reajuste de la contraseña de correo, que se puede fijar para cada autoridad. Las contraseñas almacenadas en la base de datos del sistema se cifran.

2. La internacionalización hace que la aplicación soporte cerca de 40 idiomas, desde el administrador podemos añadir los diferentes idiomas que queremos que nuestra plataforma ofrezca a los usuarios, serán estos los que una vez dentro de su portal podrán cambiar entre los idiomas que el administrador a habilitado.

- a. También permite la corrección de mensajes on-line por lo que es muy cómodo para el traductor ir enriqueciendo la bd de idiomas,
 - b. Si su idioma no se encuentra en la BD a través de este sistema puedes ir traduciendo los mensajes.
 - c. Entre estos idiomas está el Catalán que se ha traducido todo integro en nuestro servicio de informática.
 - d. Permite seleccionar la zona horaria.
 - e. Podemos importar y exportar mensajes en el idioma que queramos.
3. Permite la creación de diferentes web-site dentro de la misma plataforma.
 4. La plataforma esta extendida por todo el mundo. Vease los partners y casos de uso actuales. <http://dotlrn.org/partners/>
 5. Nueva versión para la exportación e importación de estandares. Versión beta de .LRN Learning Object Repository System (LORS) y sistema de gestión de LORS (LORSm). LORS es un sistema de bibliotecas desarrolladas en la universidad de Sydney que amplía dotLRN para incorporar IMS Metadata y especificaciones de empaquetado de contenidos así como extensiones del ADL SCORM. Se diseña de modo que cualquier aplicación dotLRN ó de OpenACS pueda hacer uso sus funciones fácilmente. La herramienta de gestión de LORS permite importar, exportar, gestion y entrega de contenidos basados en estándares dentro del dotLRN. Nos permitirá utilizar una herramienta de gran alcance para crear contenidos que luego podremos subir más adelante en nuestra dotLRN (o cualquier otro sistema que soporte los estándares e-learning).
 6. E-Mail. La versión actual asume que todos los usuarios para tener una cuenta de correo, antes deben tener una cuenta en su institución actual y que integran esta cuenta de correo con la de dotLRN. Otras opciones de la integración del webmail: Un paquete disponible y programado en php se puede configurar para funcionar en el mismo webserver si se quiere (estas configuraciones no son parte del paquete estándar del LRN). El software tiene funcionalidades para todos los miembros de un grupo o de administradores solamente. Un instructor/group admin puede determinar si éste está disponible solamente para el administrador o para los miembros de students/regular. El correo se puede enviar usando HTML o el texto plano. El correo se puede

enviar inmediatamente o programarse para que se envíe. El correo también puede ser enviado a los usuarios que han respondido al examen. Esto se utiliza con frecuencia en la planificación de citas (avisos) o recordar a estudiantes fechas de un examen. Todos los correos enviados del sistema están guardados. Las notificaciones automáticas del correo están disponibles para los módulos de foros y el de exámenes.

7. Noticias. Las noticias o los "avisos" proporcionan un mecanismo fácil para incitar la comunicación unidireccional entre los administradores del grupo y los miembros del grupo. Las noticias se fijan al portlet de las noticias en la página de class/community pero también pueden ser montadas un nivel superior sobre el portal personal. Las noticias pueden incluir HTML y el administrador fija las fechas del lanzamiento y de vencimiento para cada noticia.
8. Calendario. Los profesores pueden fijar acontecimientos y avisos en el calendario on-line del curso. Los administradores de la comunidad pueden fijar acontecimientos y actividades del club en el calendario de la comunidad. Los estudiantes tendrán a la vista todos los avisos, plazos, y fechas incluidas las de sus calendarios personales. El "MySpace" del estudiante proporciona una vista unificada de todos estos acontecimientos del calendario en una vista del día, o una vista de la semana y o una vista del mes. Las páginas personales de las asignaturas y de la comunidad proporcionan las mismas vistas. Los subgrupos pueden tener sus propios calendarios para sus reuniones etc. El calendario permite la repetición de citas. El paquete calendario también permite la sincronización básica con programas como outlook.
9. En cuanto a la actualización de la plataforma podemos destacar herramientas como el repositorio CVS de OpenAcs. CVS es un sistema concurrente de Versiones. Normalmente se usa como una forma de almacenar el código fuente, ya que mantiene las versiones de todos los archivos de manera que no se pierda nada, y se registra el uso que hacen los diferentes miembros de la comunidad OpenAcs. También sirve para combinar código desarrollado por diferentes usuarios. Todo el código se almacena en un ordenador central que se puede visitar en <http://cvs.openacs.org/cvs/>.
10. Chat. La única herramienta síncrona que dispone actualmente la plataforma es el Chat. No incluye soporte interno para elementos

multimedia de tipo audio o video. Y este paquete de chat no esta muy probado y presenta problemas.

6.4.2 El administrador de dotLRN

Desde el administrador de dotLRN se puede manejar toda la plataforma. Tiene una herramienta para gestionar el sitio via-web muy potente y fácil de utilizar.

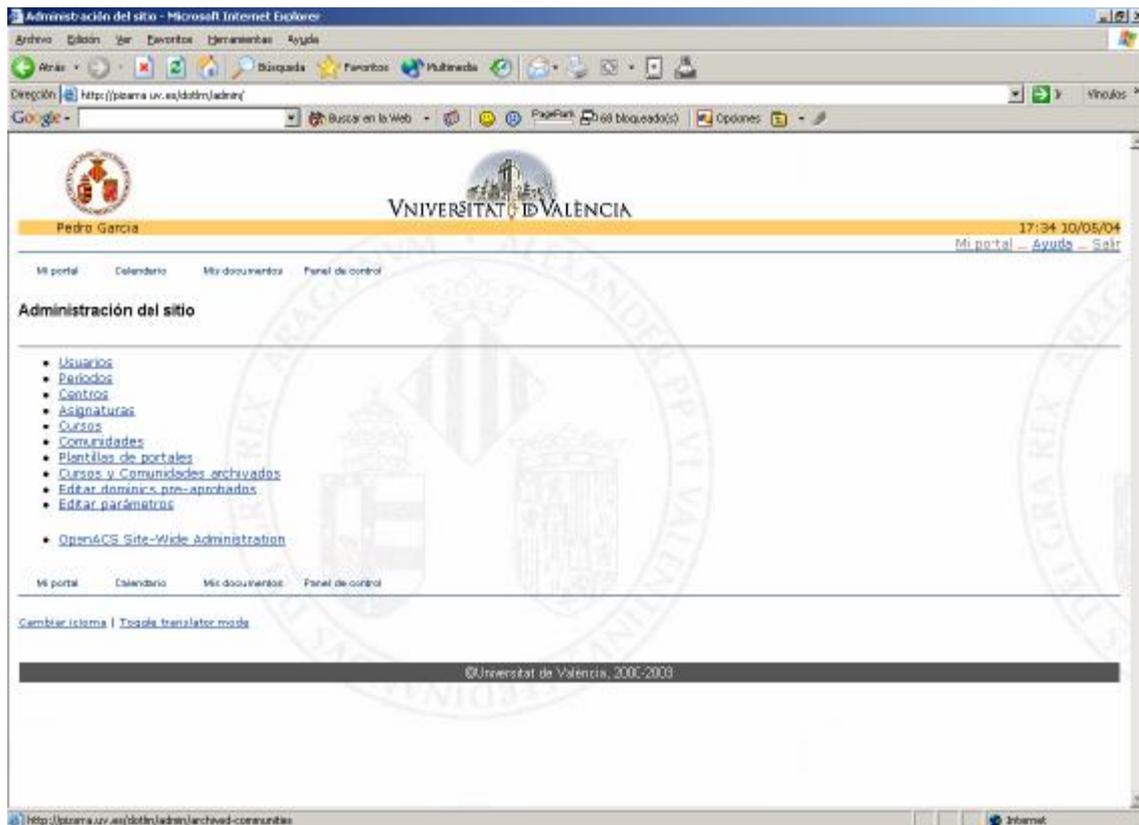


Ilustración 22. Vista del Administrador de dotLRN.

Como administrador del sitio se puede apreciar que desde esta pantalla podemos administrar.

- **Usuarios:** desde aquí podemos crear usuarios, buscar usuarios, una vez creado el usuario le debemos asignar el rol que desempeñara en la plataforma: alumno, profesor, invitado, o si será administrador del sitio.
- **Asignaturas:** Desde aquí podemos gestionar las asignaturas, dar de alta nuevas asignaturas, al dar de alta una asignatura hay que asignarla a un centro, darle su código numérico y un nombre. Los códigos son los que tienen académicamente.

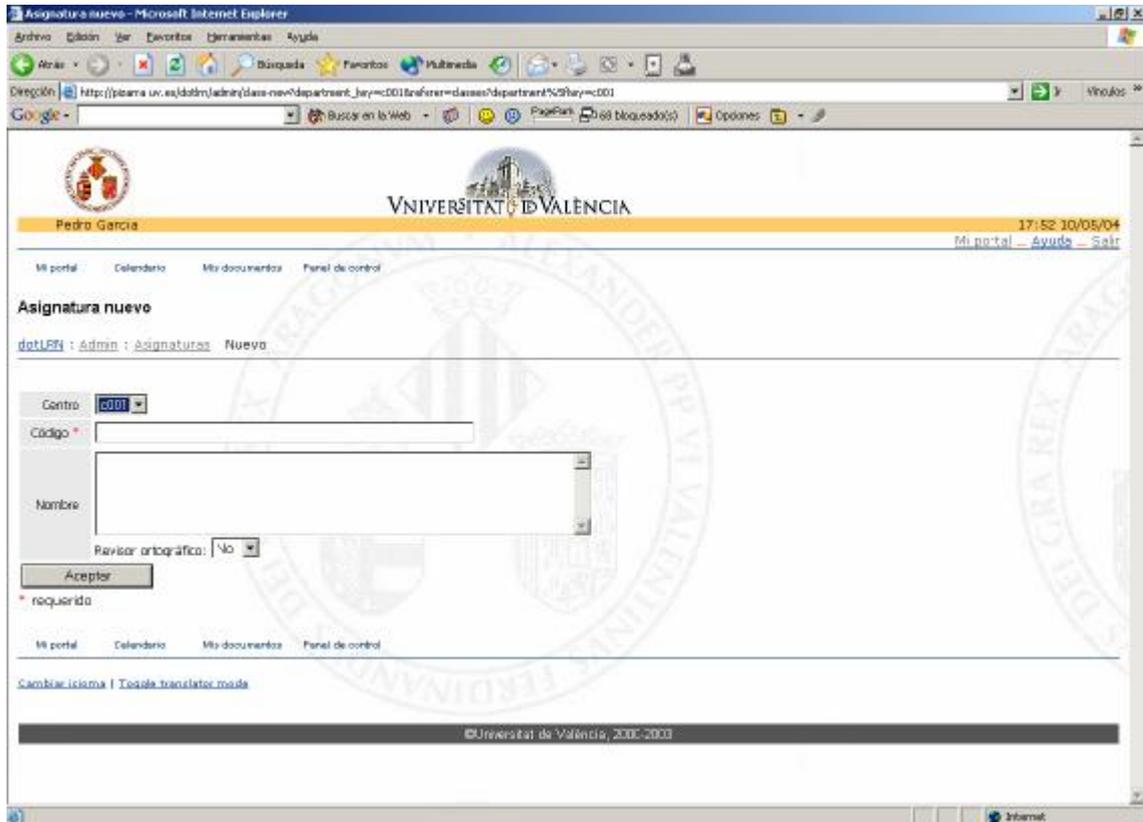


Ilustración 23. Alta nueva asignatura.

- **Cursos:** Desde esta pantalla podemos ver los cursos dados de alta en el sistema, curso se entiende como la tupla formada por los siguientes elementos:

Centro – Asignatura – Período

Es decir entendemos curso como la relación existente entre estos tres elementos.

Centro: centro donde se imparte.

Asignatura: asignatura docente.

Período: Período en que se imparte ese curso.

- **Comunidades:** desde esta opción vamos a la administración de las comunidades que hay en la plataforma. El administrador es el encargado de crear las comunidades. Una vez creada la comunidad y

añadidos los miembros deberá decidir quienes serán administradores de esa comunidad para que la puedan gestionar.

- Plantillas portales. Diferentes vistas de la plataforma predeterminadas, puede cambiar el aspecto de la plataforma desde aquí.
- Cursos y comunidades archivados. Históricos almacenados.
- Dominios preaprobados. Los dominios que aquí se añadan, cuando un usuario se de de alta automáticamente se le añade en dotlrn.
- Editar parámetros. Parámetros de configuración de la plataforma.
- Openacs site-wide administrador. Desde este menú accedemos a la configuración y administración de la parte de OpenACS, es decir gestión de paquetes, instalación de software, internacionalización, autenticación, usuarios de OpenACS, administrar los diferentes subsites. Desde la opción developers-admin ya podemos entrar en contacto con la administración de la plataforma, aquí podemos crear, instalas o gestionar paquetes.

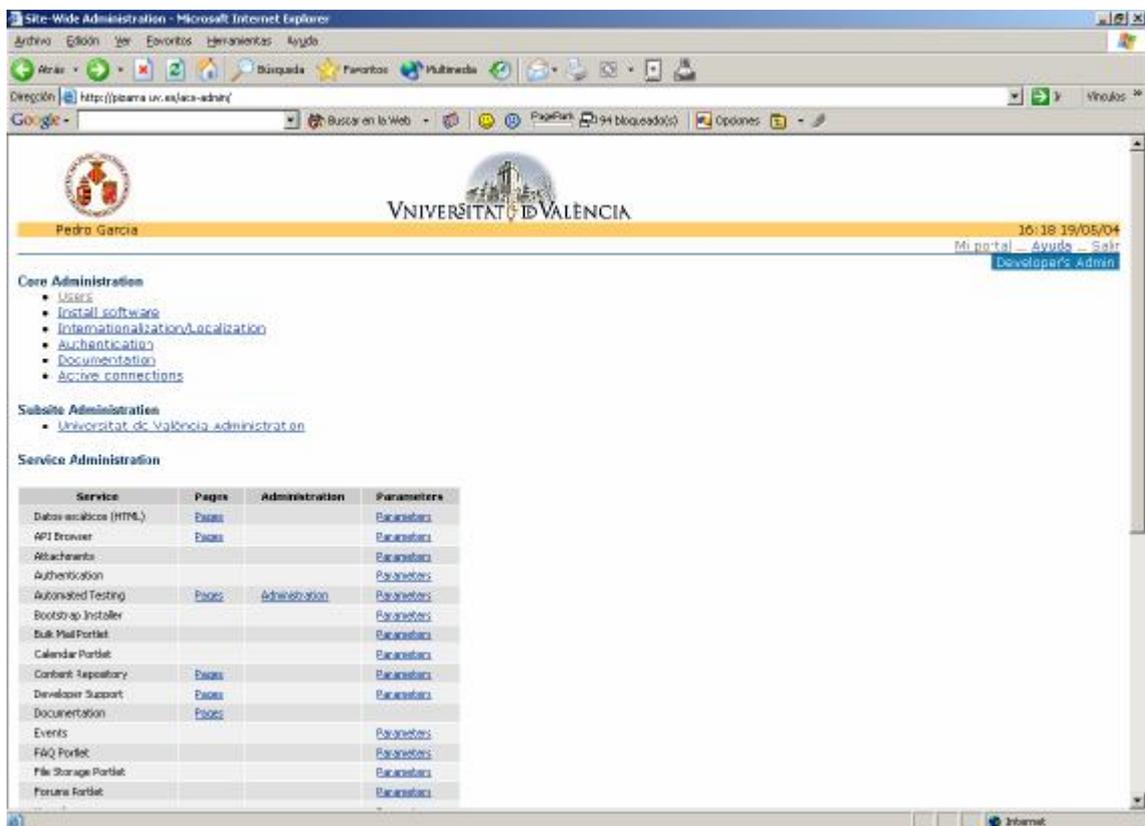


Ilustración 24. Opciones del site-wide administrator.

6.4.2.1 Cursos en dotLRN

Los cursos como hemos dicho antes son una combinación de (asignatura-periodo-centro), la plataforma no soporta ningún mecanismo para la creación de más de un curso a la vez por que se ha tenido que desarrollar un paquete expresamente para hacer esta función, este script solicita a la DB de docencia los cursos (DB en Oracle, es una replica de DB2) y luego los inserta en la plataforma.

Los cursos aparecen en color azul para diferenciarlos de las comunidades y del espacio personal.

Si se desea añadir algún curso desde la plataforma se debe añadir desde el menú de asignaturas, y una vez seleccionado el centro y la asignatura a la que queremos crear el curso ya podemos crearlo, solo falta añadirle el período docente y ya esta creado, sólo falta que el administrador decida que miembros son administradores y a partir de ese momento ya pueden los propios usuarios gestionar el curso. Las pantallas capturadas están mostradas en el punto anterior porque son funciones del administrador.

La inscripción en los cursos puede dejarse abierta pero en nuestro caso no tiene sentido ya que a priori los alumnos ya están matriculados en todas las asignaturas que han solicitado en su matrícula. El proceso de inscripción en las diferentes asignaturas en las que esta matriculado un alumno también se ha desarrollado expresamente un paquete para hacer esta tarea.

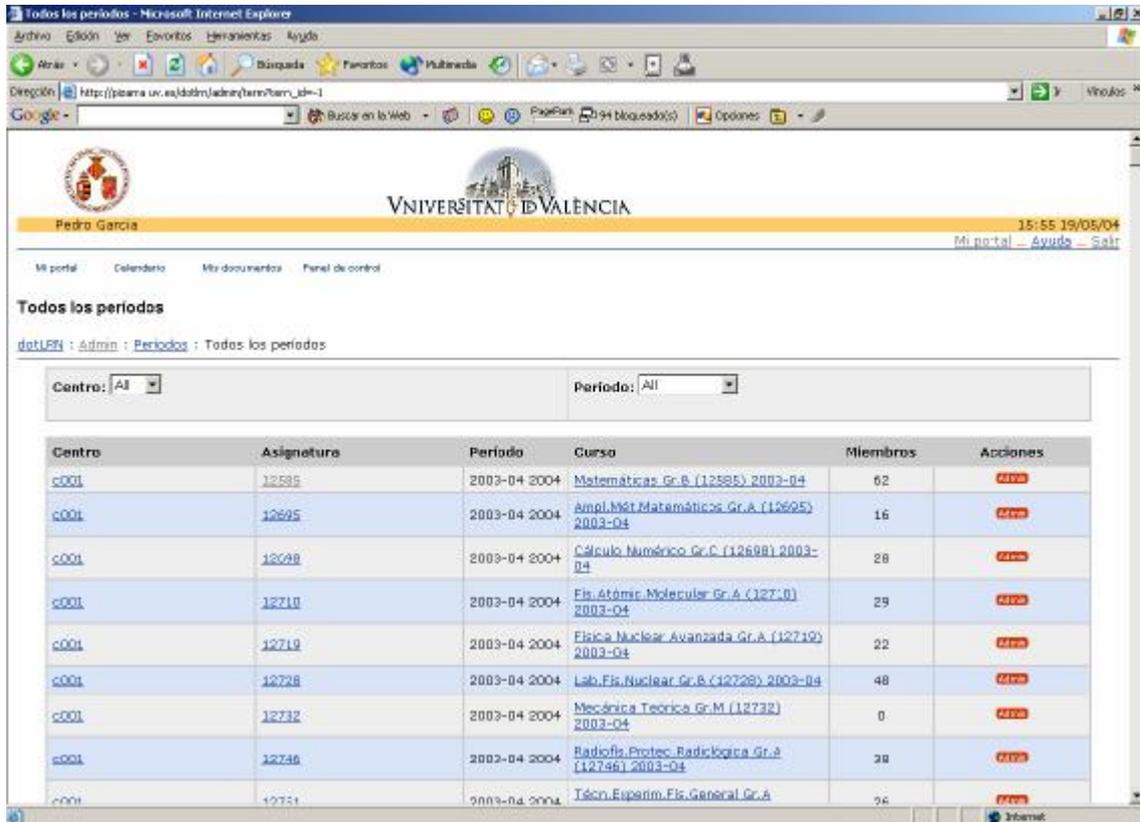


Ilustración 25. Cursos disponibles.

Desde la vista anterior nos permite administrar cualquier curso, pinchando en el botón admin, es aquí donde le decimos que miembros son administradores, ya que esta vista sólo la tiene el administrador de la plataforma.

Otra vista diferente es cuando un usuario esta visitando un curso, subgrupos, dependiendo del rol que desempeñe en la plataforma podrá o no administrar el curso. Pero las pestañas para moverse por el curso e interactuar con él son las siguientes:

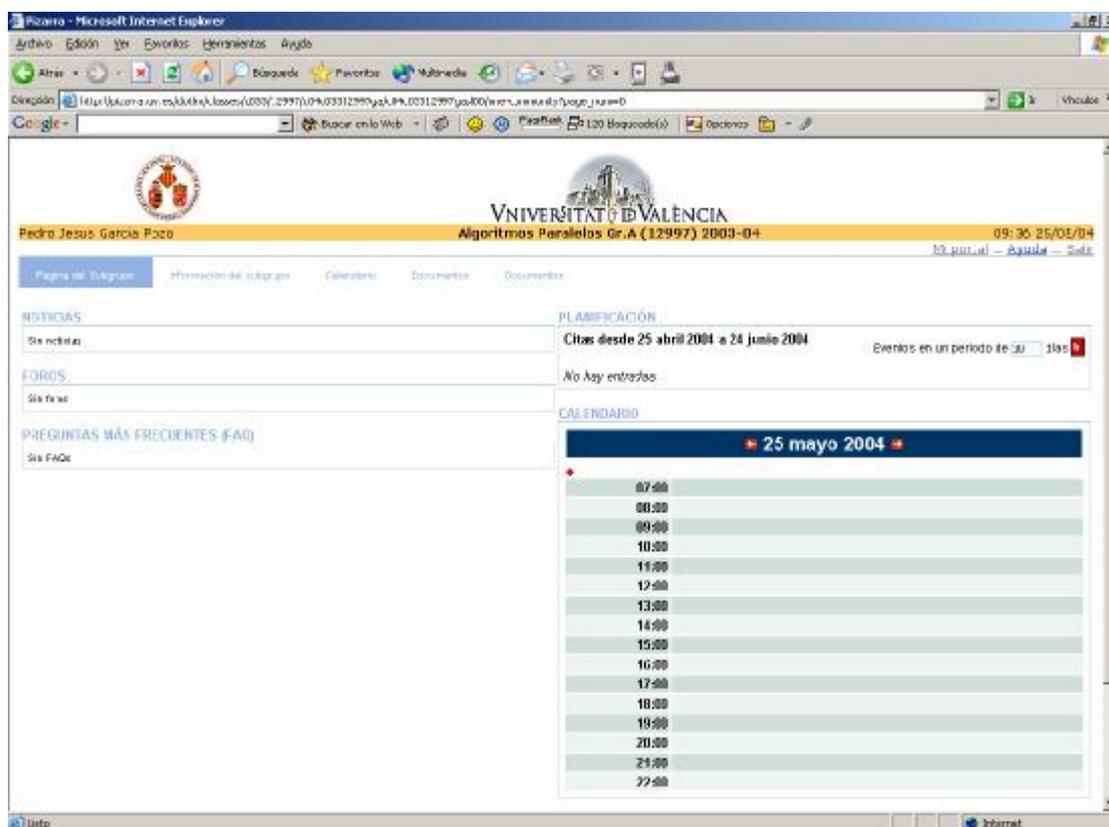


Ilustración 26. Vista de un curso.

- **Página del subgrupo:** Es el sitio donde tenemos disponibles los portlets siguientes; (noticias, foros, preguntas frecuentes, planificación, calendario).
- **Información del subgrupo:** Aquí nos aparece una lista con todos los miembros del curso y el rol tienen. También podemos enviar un correo a todos los miembros.
- **Calendario:** Desde aquí podemos gestionar el calendario del curso.
- **Documentos:** Aquí es donde encontraremos algunos directorios ya creados como son las tareas, los exámenes, apuntes, notas, proyectos y documentos públicos.
- **Admin.:** Solo lo tienen disponible los administradores del sistema o profesores del curso, son los encargados de gestionarlos. Esta es la pestaña dotada de más funcionalidad.

Las principales funciones a la hora de administrar un curso las veremos en el punto siguiente ya que es el profesor el encargado de gestionar todo esto.

6.4.2.2 Comunidades en dotLRN

Una vez se han creado las comunidades por el administrador la función de administrador de la misma ya recae sobre el miembro que sea administrador de esa comunidad.

Las comunidades aparecen en otro color para diferenciarlas de los cursos y del portal personal, son de color naranja y no ofrece gran diferencia de los cursos, sino todo lo contrario son muy similares, característica a su favor en cuanto continua con una misma filosofía de trabajo.

La principal función de las comunidades será la de compartir información de tal manera que todos los miembros puedan ser participes de los avances en los temas que trate su comunidad.

Dependiendo del usuario que este visitando una comunidad, ya sea miembro o administrador, o no administrar el curso. Pero las pestañas para moverse por la comunidad e interactuar con ella son las siguientes:

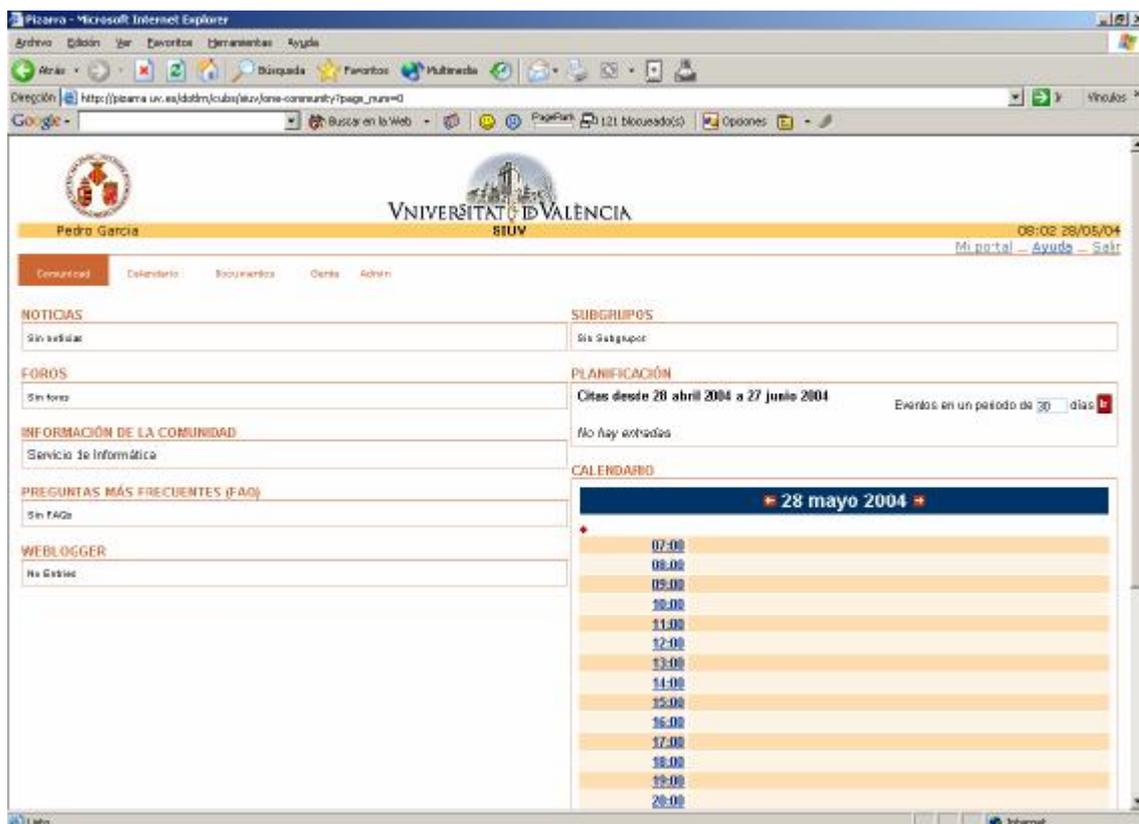


Ilustración 27. Vista de las comunidades.

En esta pantalla tenemos disponible:

- **Comunidad:** Aquí tenemos disponibles todos los portlets principales, el calendario, foro, noticias, weblogger, faq, planificación.
- **Calendario:** Desde aquí podemos incluir citas en el calendario, tener diferentes vistas de este calendario.
- **Documentos:** Aquí podemos crear nuestros directorios y subir los archivos que queramos a ellos, y por defecto tenemos una carpeta publica que será donde dejaremos los documentos para que los demás miembros de la comunidad puedan utilizarlos.
- **Gente:** Visualizamos la lista de miembros, según la política aplicada para enviar correo a los demás miembros nos aparecerá una opción para enviar a todos los miembros o individualmente.
- **Admin.:** La vista para un administrador de una comunidad sería la siguiente.

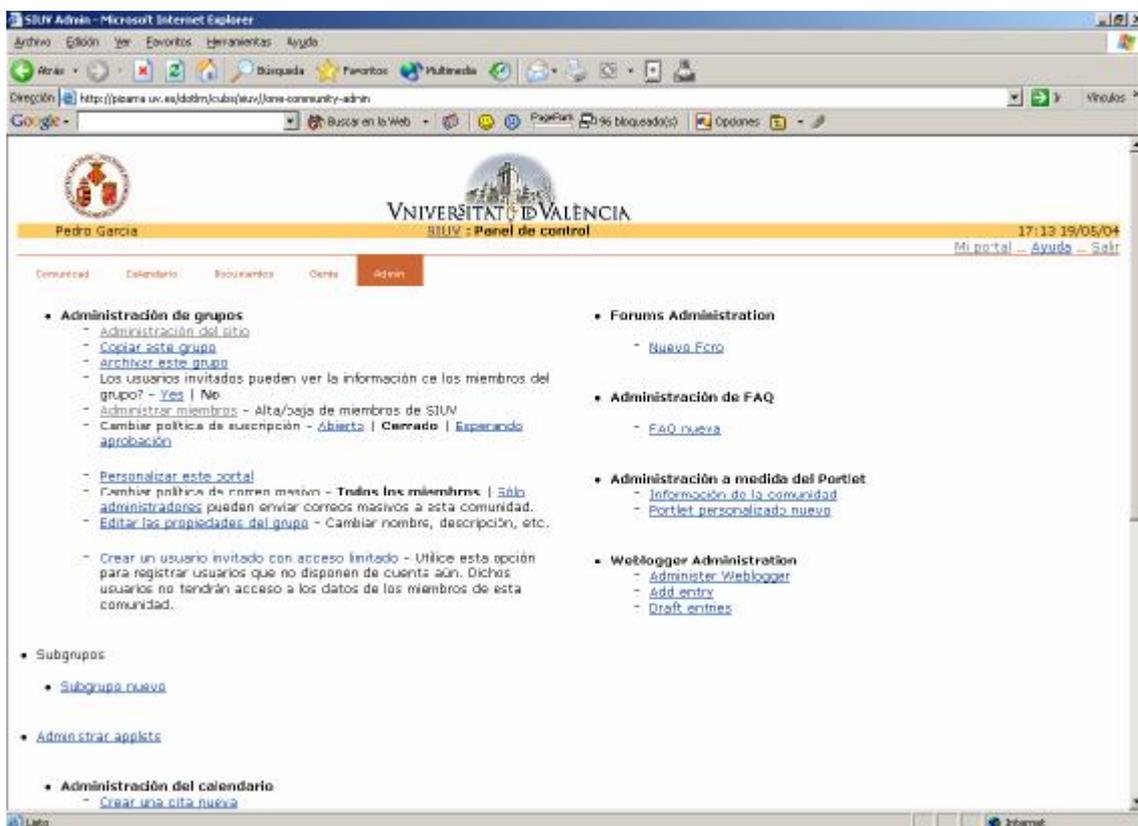


Ilustración 28. Administración de una comunidad.

Desde esta pantalla podemos gestionar los miembros de la comunidad, el foro (crear los foros y habilitarlos o deshabilitarlos), las faq's (Crear faq's y habilitarlas). Cambiar la política de suscripción en la comunidad. Administrar las noticias, el calendario, el modulo

survey (modulo para generar exámenes), permitir el envío de correo masivo a la comunidad.

Administrar el weblogger, un weblog (también llamado blog, diario personal o cuaderno de bitácora) es una herramienta de trabajo colaborativa asincrónica - y hasta cierto punto sincrónica- que permite a un autor o varios publicar mediante un simple navegador, un sitio web con una información muy dinámica (con cambios frecuentes) fundamentalmente compuesto por información textual (comentarios, artículos, enlaces) y añadir además otros elementos (imágenes, archivos de audio, adjuntar ficheros, etc.), automáticamente ordenados de forma cronológica.

6.4.3 El profesor

El rol de profesor mantiene toda su similitud con el profesor de cualquier sistema de educación.

A continuación describiremos las principales actividades que puede planificar un profesor en los cursos:

1. **Porlet Documentos:** Es el encargado de subir los apuntes y crear las tareas, publicar los exámenes, subir las notas, dejar documentos públicos para todo el mundo. Los documentos pueden ser cualquier tipo de formato, incluso puede subir alguna estructura de directorio subiendolo en formato comprimido zip.
2. **Porlet Calendario:** Es el encargado de poner las citas referentes al curso, días de entrega de los trabajos pendientes, publicar días de las prácticas.
3. **Porlet admin.:** Como encargado de administrar los cursos y hacer que toda la información referente al curso este disponible deberá gestionar la Política de suscripción: como dijimos anteriormente aparecerá "cerrada", pero podría estar "abierto" a todo el público o en "espera de aprobación". Deberá gestionar la Política para el envío de correo masivo. Por defecto sólo el profesor podrá hacer esto, pero también puede cambiarse desde aquí. Podrá decidir su el sistema envía una alerta a los miembros en forma de mail cada vez que genere una tarea nueva. Debe gestionar las noticias y los foros

creando uno como mínimo si desea que se utilizen, faq's también como administrador del curso deberá crear uno para iniciar su uso. Podrá crear los exámenes con el módulo survey.

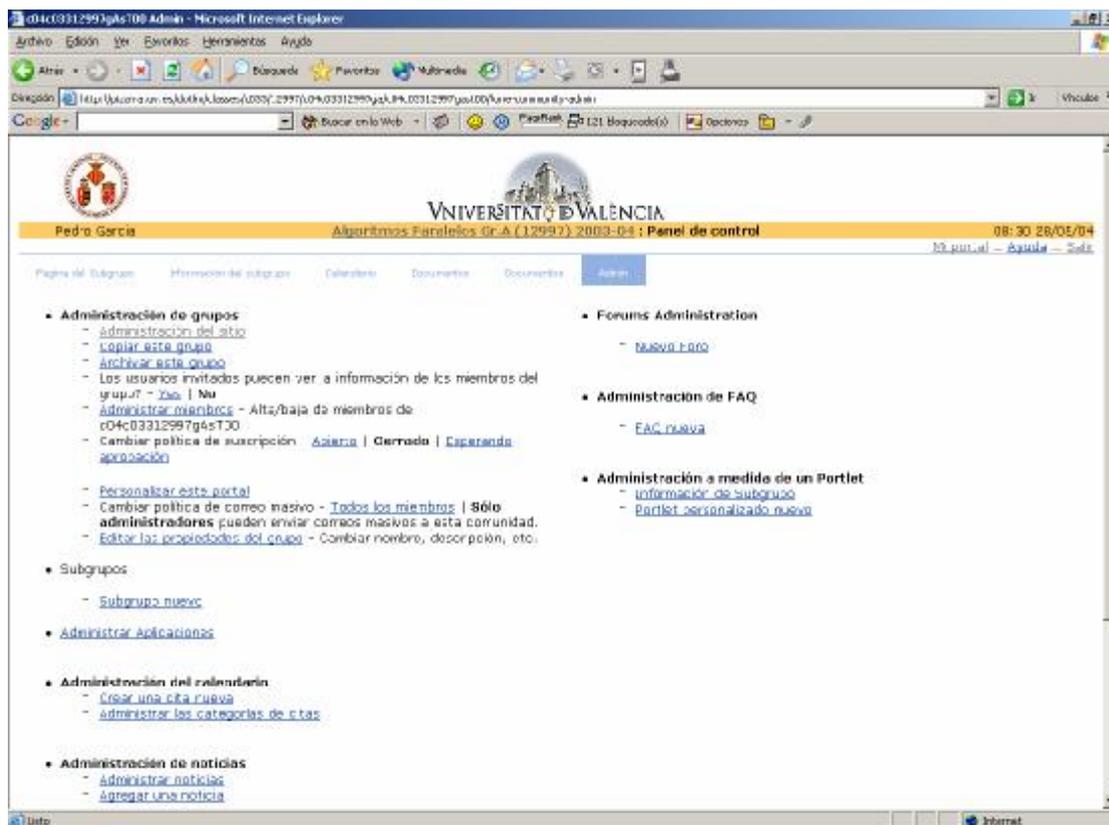


Ilustración 29. Administración de un curso.

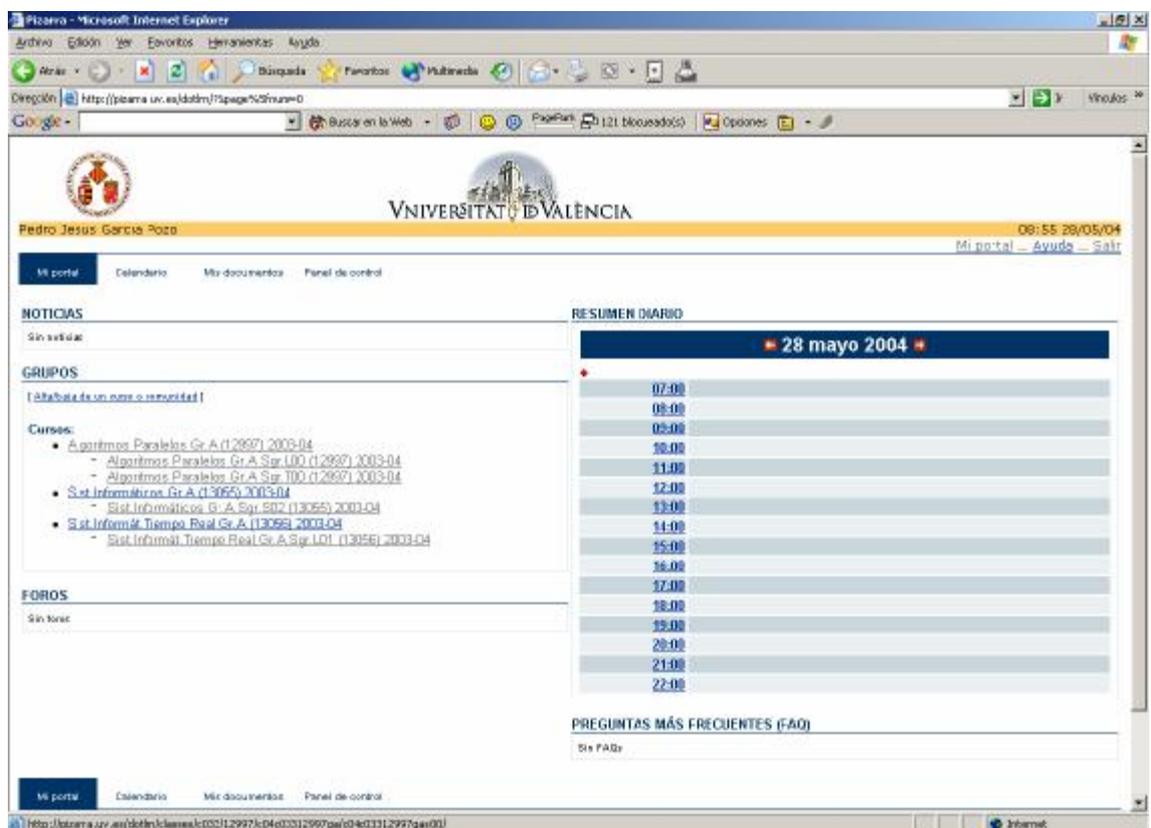
Algunas notas a tener en cuenta:

- El tamaño máximo de los ficheros esta limitado a 2 Mb aunque esto es configurable por el superusuario, de momento se ha dejado como viene por defecto.
- Los ficheros pueden ser de cualquier tipo como páginas web, archivos de audio, documentos de word, animaciones, videos, etc.. aunque la plataforma no entiende de estos distintos ficheros, gracias al navegador y de los plug-gins o controles active-X soportados podremos ejecutar en nuestra maquina todos estos distintos archivos.
- Si se sube un archivo con el mismo nombre de otro que ya existe se sobrescribe perdiendo el antiguo.
- Esta interfaz de administrador solo esta disponible para el superusuario y administradores del curso. Aunque el puede cambiar el rol a cualquier miembro del curso.

- Puede cambiar la apariencia del curso, aunque esto en nuestra organización no tiene sentido y no se permitirá hacer.

6.4.4 El alumno

El estudiante es un usuario consumidor frente a la plataforma, podrá hacer uso de todas las herramientas que tiene habilitadas. Como ya hemos comentado los cursos aparecen cerrados de tal forma que nadie puede inscribirse en un curso utilizando la plataforma.



El estudiante como puede verse en la figura anterior nada más validarse accede a su portal desde donde puede ver las asignaturas que esta matriculado ese curso. Luego con solo hacer un clic en la asignatura deseada puede acceder a ese curso.

La pega más evidente que tiene la plataforma por el momento es que casi todas las herramientas son asíncronas, pero ya se esta trabajando en módulos con el chat y en breve estarán disponibles.

El estudiante puede configurar a su gusto la plataforma para visualizar la información como él desee.

Cada usuario pueda especificar su propia zona horaria, y todas las fechas marcadas en dotLRN se traducirán a esa zona horaria (las fechas de escritura de mensajes, de entrega de tareas, etc.). También pretende que cada usuario pueda configurar su propio perfil, el idioma.

No se puede pasar de un curso a otro de manera directa, debe realizarse siempre a través de la página principal, pero esta muy accesible siempre el volver a la página principal.

En el foro no se le permite la generación de nuevos hilos de conversación.

Se requiere que el computador del estudiante disponga de todo el software necesario para poder ver los contenidos. Es decir si un profesor cuelga sus apuntes con un determinado software el alumno deberá tenerlo instalado para verlo.

El calendario lo debe utilizar como agenda personal aunque en la pantalla principal le saldrán las citas que los profesores pongan en las asignaturas así como las fechas de los exámenes que son cargadas automáticamente a la vez que se cargan las asignaturas en la plataforma.

El modulo survey permite la generación de exámenes, estos en la versión que hay ahora son meros formularios que al ser contestados por los alumnos se envían a la plataforma de vuelta, cuando el profesor quiera ver las contestaciones de estos accede y ve por alumno las respuestas hechas, un inconveniente comparado con otras plataformas es que no corrige.

CAPITULO 7

7. Conclusiones y trabajos futuros

7.1 Conclusiones

Una vez concluido el proyecto se verá si los objetivos planteados al inicio del mismo se han cumplido o no.

Viendo el gran interés que están mostrando todas las universidades y grandes empresas en esta tecnología veo de gran utilidad haber orientado mi proyecto en esta línea. La posibilidad de seguir en este camino depende de la suerte pero siempre ayuda el esfuerzo realizado y la formación adquirida en esta materia a lo largo de estos meses de duración del proyecto.

El éxito de esta plataforma se debe a la gran confianza mostrada y al desarrollo continuo de instituciones tan importantes como el MIT, UNED, Universidad de Sydney y a partir de este momento la propia Universidad de Valencia. A esto debemos unir el auge que esta teniendo el software GPL y la comunidad activa de usuarios aportando soluciones y desarrollando nuevos módulos, "openacs.org", comunidad con más de 7500 usuarios.

En cuanto a los objetivos que nos marcamos podemos dar por concluido el trabajo ya que la implantación de la plataforma se ha llevado a cabo y durante este cuatrimestre se ha utilizado por diferentes profesores a quienes se les agradece el esfuerzo hecho para ayudar en este sentido.

Como nuestro principal objetivo ha sido implantar este software, podemos destacar que donde más esfuerzo hemos requerido ha sido en la integración de las Bases de datos académicas ya que el sistema posee cerca de 400 tablas y había que estudiar muy bien toda relación posible entre ellas para hacer la carga de datos correctamente.

Seguramente existan errores y cosas por depurar pero esto se logra como en todo gran proyecto de desarrollo informático, que no es de otra forma que lanzarlo en fase piloto para solventar los problemas más críticos y una vez solucionados y corregidos poder hacer un lanzamiento en producción.

Podemos concluir que la instalación ha sido satisfactoria, que la integración con las bases de datos académicas se ha conseguido y que el proyecto promete.

7.2 Trabajos futuros

El resultado de este proyecto no debe de ser otro de servir como guía a la hora de la implantación de la plataforma en la Universidad.

El principal método de trabajo a seguir una vez implantada la plataforma es el mantenimiento de esta y sobre todo seguir de cerca el desarrollo de nuevos paquetes que permitan el desarrollo de e-learning con esta plataforma.

Los módulos más esperados y los que se deberían ir probando son:

- LORS es un conjunto de librerías que permiten la importación y exportación de metadatos (IMS) y de ADL SCORM, utilidad muy solicitada por toda la comunidad para poder introducir en la plataforma materiales procedentes de otras plataformas o para llevarlos.
- Otro paquete a tener en cuenta es el File Storage ya que la nueva versión permite WEBDAV. (extensión del protocolo http 1.1 y que nos permite realizar operaciones con ficheros como, escritura, borrado, a través del protocolo http y utilizando un navegador).
- SMS seguir el hilo a las evoluciones que surjan en este sentido, la idea es poder hacer que los avisos que se quieran puedan ser enviados al móvil de los alumnos vía SMS.
- Sobre todo mejorar los procesos de carga de usuarios y alumnos, ya que esta solución se ha impuesto por las urgencias con que se ha lanzado el proyecto.
- Atención de usuarios, sugerencias, mantenimiento, etc..
- Formación de los profesores en esta herramienta, con repeticiones de los cursos a lo largo de los primeros años de esta implantación, ayudara a que los profesores se involucren en esto y no lo vean como una montaña que sólo les va a dar más trabajo.

En cuanto a las necesidades físicas se debería seguir investigando en los temas de escalabilidad ya que se han detectado algunas deficiencias en

cuanto al volumen de datos que deben manejar los SGBD y buscar algún tipo de solución para evitar posibles cuellos de botella.

Seguir de cerca proyectos internacionales o europeos como puede ser E-Lane, en cuanto al desarrollo que mantenga esta tecnología y quedarse desfasado o anclado con algún problema.

Por todas las razones expuestas se ve como un nuevo mundo donde queda mucho que investigar y donde se pueden aportar muchas cosas si el proyecto es apoyado desde la cima.

CAPITULO 8

8. Bibliografía

8.1 Enlaces

La mayoría de referencias y documentos leídos y consultados para realizar este trabajo se han obtenido de Internet por lo que paso ha realizar una breve cita de las URL's más utilizadas.

- [1] Página oficial de la plataforma dotLRN. (<http://www.dotlrn.org>)
- [2] Página oficial del servidor web AOLserver. (<http://www.aolserver.com>)
- [3] Página oficial de OpenACS. (<http://www.openacs.org>)
- [4]. Cline M, Girou M. and Young H. Enduring Business Themes (EBTs), sideline in Building Application Frameworks: Object-oriented foundations of framework design, M. Fayad, D. Schmidt R. Johnson Eds. John Wiley and Sons 1999.
- [5]. "Estandares de metadatos sobre el aprendizaje (learning Metadata Standard)".
http://www.iaa.upf.es/~jblat/material/doctorat/students/jcc/Learning_Metadata_Standards.htm
- [6] E-learning <http://www.elearningworkshops.com/>
- [7] Definición de e-learning
<http://www.informaticamilenium.com.mx/paginas/mn/articulo78.htm>
- [9] Centro de supercomputación de Galicia <http://www.cesga.es>
- [10] Una herramienta estratégica de cambio
<http://www.alfa.une.edu.ve/biblio/BiblioGeneral/S/S/sistemas1.asp>
- [11] Diagnóstico del e-learning corporativo en España
<http://www.elearningamericalatina.com>
- [12] Comparación de plataformas <http://www.edutools.info>
- Definición de Blended learning
<http://www.educaweb.com/esp/servicios/monografico/formacionvirtual/1181108.asp>

- Portal de teleformación de la Universidad de Valencia.
<http://www.uv.es/ticape>
- Grupo de investigadores de la Universidad de Australia, principalmente los artículos de Rafael Calvo que hacen referencia a dotLRN,
<http://www.weg.ee.usyd.edu.au/>
- MIT Sloan School of Management/ Massachusetts Institute of Technology
<http://dotlrn.mit.edu/>
- OS Media, e-Learning, URL: <http://www.qsmedia.es/elearning/default.cfm>
- SCORM Concepts, URL:
<http://www.eduworks.com/LOTT/tutorial/scormconcepts.html>
- IMS (Instructional Management System) Global Learning Consortium,
URL:
<http://www.imsproject.org/>
- Advanced Distributed Learning (ADL) Initiative, URL:
<http://www.adlnet.org>
- AICC, The Aviation Industry CBT Committee, URL: <http://www.aicc.org/>
-

8.2 Tesis y trabajos

- Formación y Nuevas Tecnologías: Posibilidades y condiciones de la Teleformación como espacio de aprendizaje.
<http://prometeo.us.es/idea/mie/pub/marcelo/Formacion%20y%20NNTT.pdf>
- ¿Qué aporta internet al cambio pedagógico en la educación superior?
<http://webpages.ull.es/users/manarea/Documentos/documento7.htm>
- Desarrollo y experimentación de una plataforma de teleeducación síncrona sobre una red de cable de banda ancha.
<http://www.virtualeduca.org/virtualeduca/virtual/actas2002/actas02/722.pdf>

- "Proyecto de evaluación de plataformas de teleformación para la implantación en el ámbito universitario". Autor: Darío Roig. Año: 2003.
- "E-learning. Análisis de plataformas de libre distribución". Autor: Sebastián Delgado. Año: 2003.

8.3 Libros

- Alcantud, F. :<<Teleformación Diseño para todos>> Universidad de Valencia (Servicio de publicaciones) (1999).
- Pressman, R. :<<Ingeniería del Software: Un enfoque práctico>> MC Graw Hill: 5ª. Edición (2002).
- Ruipérez, G. :<< Educación virtual y eLearning>> auna Fundación. (2003).
- Zarandieta Morán, F. La educación por Internet. Madrid: Anaya Multimedia, 2002
- Marcelo García, C. E-Learning. Teleformación. Barcelona: Gestión 2000, 2002.
- Pearson, Prentice Hall <<e-Learning, las mejores prácticas en España>> 2003.
- Christia W. R, Gregorio Martin, << El proyecto fin de carrera en Ingeniería Informática>>. 2002.

ANEXO I - Estándares e-learning

En las actuales soluciones e-learning, generalmente los contenidos preparados para un sistema no pueden ser fácilmente transferidos a otro. Los estándares e-learning son el vehículo a través del cual será posible dotar de flexibilidad a las soluciones e-learning, tanto en contenido como en infraestructura. Ellos han abierto una puerta hacia una manera más coherente de empaquetar los recursos y contenidos, tanto para los estudiantes como para los desarrolladores.

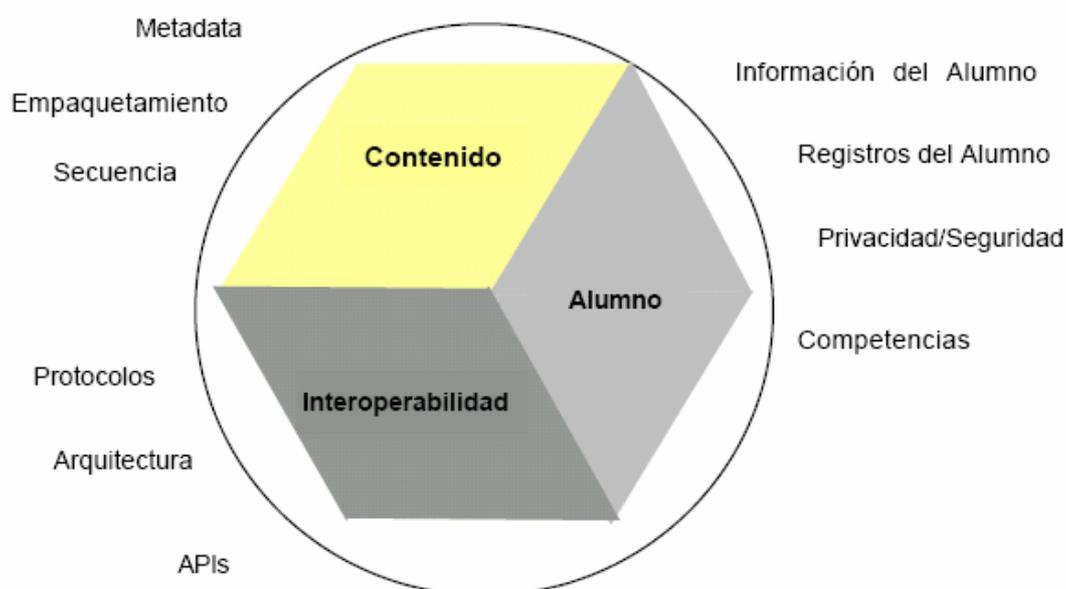
Estándares comunes para asuntos tales como metadata de contenidos, empaquetamiento de contenidos, secuencia de contenidos, interoperabilidad de preguntas y tests, perfil de alumnos, interacción en tiempo de ejecución, etc., son requisitos indispensables para el éxito de la economía del conocimiento y para el futuro del e-learning.

En el mercado existen tanto LMS como Courseware de muchos fabricantes distintos. Por ello se hace necesaria una normativa que compatibilice los distintos sistemas y cursos a fin de lograr dos objetivos:

1. Que un curso de cualquier fabricante pueda ser cargado en cualquier LMS de otro fabricante.
2. Que los resultados de la actividad de los usuarios en el curso puedan ser registrados por el LMS.

Como se puede ver en la siguiente figura, los distintos estándares que se desarrollan hoy en día para la industria del e-learning se pueden clasificar en los siguientes tipos:

- **Sobre el Contenido o Curso:** Estructuras de los contenidos, empaquetamiento de contenidos, seguimiento de los resultados.
- **Sobre el Alumno:** Almacenamiento e intercambio de información del alumno, competencias (habilidades) del alumno, privacidad y seguridad.
- **Sobre la interoperabilidad:** Integración de componentes del LMS, interoperabilidad entre múltiples LMS.



Al hablar sobre un estándar e-learning, nos estamos refiriendo a un conjunto de reglas en común para las compañías dedicadas a la tecnología e-learning. Estas reglas especifican cómo los fabricantes pueden construir cursos on-line y las plataformas sobre las cuales son impartidos estos cursos de tal manera de que puedan interactuar unas con otras. Estas reglas proveen modelos comunes de información para cursos e-learning y plataformas LMS, que básicamente permiten a los sistemas y a los cursos compartir datos o "hablar" con otros. Esto también nos da la posibilidad de incorporar contenidos de distintos proveedores en un solo programa de estudios.

Estas reglas además, definen un modelo de empaquetamiento estándar para los contenidos. Los contenidos pueden ser empaquetados como "objetos de aprendizaje" (learning objects o LO), de tal forma de permitir a los desarrolladores crear contenidos que puedan ser fácilmente reutilizados e integrados en distintos cursos.

Finalmente, los estándares permiten crear tecnologías de aprendizaje más poderosas, y "personalizar" el aprendizaje basándose en las necesidades individuales de los alumnos.

Básicamente, lo que se persigue con la aplicación de un estándar para el e-learning es lo siguiente:

- **Durabilidad:** Que la tecnología desarrollada con el estándar evite la obsolescencia de los cursos.
- **Interoperabilidad:** Que se pueda intercambiar información a través de una amplia variedad de LMS.
- **Accesibilidad:** Que se permita un seguimiento del comportamiento de los alumnos.
- **Reusabilidad:** Que los distintos cursos y objetos de aprendizaje puedan ser reutilizados con diferentes herramientas y en distintas plataformas.

Esta compatibilidad ofrece muchas ventajas a los consumidores de e-learning.

- Garantizan la viabilidad futura de su inversión, impidiendo que sea dependiente de una única tecnología, de modo que en caso de cambiar de LMS la inversión realizada en cursos no se pierde.
- Aumenta la oferta de cursos disponibles en el mercado, reduciendo de este modo los costos de adquisición y evitando costosos desarrollos a medida en muchos casos.
- Posibilita el intercambio y compraventa de cursos, permitiendo incluso que las organizaciones obtengan rendimientos extraordinarios sobre sus inversiones.
- Facilita la aparición de herramientas estándar para la creación de contenidos, de modo que las propias organizaciones puedan desarrollar sus contenidos sin recurrir a especialistas en e-learning.

Estrictamente hablando, no existe un estándar e-learning disponible hoy en día. Lo que existe es una serie de grupos y organizaciones que desarrollan especificaciones (protocolos). Hasta la fecha, ninguna de estas especificaciones ha sido formalmente adoptada como estándar en la industria del e-learning. Estas especificaciones no dejan de ser recomendaciones, que por el momento la industria trata de seguir.

Iniciativas de estándares sobre e-learning

Dentro de las principales iniciativas de estándar para e-learning podemos mencionar:

AICC, Aviation Industry CBT Comitee

La industria de la aviación ha sido tradicionalmente un gran consumidor de formación, por lo que en 1992 decidieron crear un comité que desarrollase una normativa para sus proveedores de formación basada en computador. De este modo garantizaban la armonización de los requerimientos de los cursos, así como la homogeneización de los resultados obtenidos de los mismos.

Fue el primer organismo creado para crear un conjunto de normas que permitiese el intercambio de cursos CBT (Computer Based-Training) entre diferentes sistemas.

Las especificaciones del AICC cubren nueve áreas principales, que van desde los learning objects (LO) hasta los learning management systems (LMS). Normalmente, cuando una compañía dice que cumple con las especificaciones AICC, significa que cumple con al menos una de estas guidelines y recomendaciones (AICC Guidelines and Recommendations, AGRs).

La lista completa de AGRs es la siguiente:

AGR 001: AICC Publications
AGR 002: Courseware Delivery Stations
AGR 003: Digital Audio
AGR 004: Operating/Windowing System
AGR 005: CBT Peripheral Devices
AGR 006: Computer-Managed Instruction
AGR 007: Courseware Interchange
AGR 008: Digital Video
AGR 009: Icon Standards: User Interface
AGR 010: Web-Based Computer-Managed Instruction

Aunque la AICC ha publicado varias guías, la más seguida es la AGR 010 que habla de la interoperabilidad de las plataformas de formación y los cursos.

En esta guía se resuelven dos de los problemas fundamentales:

1. La carga sin problemas en un LMS de cursos creados por terceros. Este objetivo se consigue definiendo el curso como una entidad totalmente independiente de la plataforma, y creando un sistema (ficheros) de descripción del curso que pueda ser entendido por cualquier plataforma.
2. La comunicación entre el LMS y el curso, de tal modo que el curso pueda obtener información necesaria sobre el usuario, y después transmitir los resultados de las interacciones y evaluaciones realizadas por el mismo a la plataforma a fin de su almacenamiento y tratamiento estadístico.

Este segundo objetivo es logrado mediante la definición de un mecanismo de comunicación entre el curso y la plataforma, y un conjunto de datos mínimos que deben ser transmitidos del curso a la plataforma y viceversa. La AICC describe dos mecanismos, uno más sencillo y extendido basado en el protocolo http, y otro mediante una API.

La AICC cuenta con un programa de certificación (a diferencia de las otras iniciativas) y dispone de un test suite que le permite a las compañías verificar que sus productos son compatibles con otros sistemas que cumplen con las especificaciones AICC.

Actualmente la AGR 010 de la AICC es el “estándar de facto” en la industria del e-Learning.

IEEE Learning Technologies Standards Committee (LTSC)

Se trata de un organismo que promueve la creación de una norma ISO, una normativa estándar real de amplia aceptación. El LTSC se encarga de preparar normas técnicas, prácticas y guías recomendadas para el uso informático de componentes y sistemas de educación y de formación, en concreto, los componentes de software, las herramientas, las tecnologías y

los métodos de diseño que facilitan su desarrollo, despliegue, mantenimiento e interoperación.

Lo que hizo fue recoger el trabajo del comité de la AICC y mejorarlo, creando la noción de metadata (información sobre los datos, una descripción más detallada que la ofrecida por la AGR 010 de la AICC de los contenidos del curso).

LTSC tiene más de una docena de grupos de trabajo (working groups o WGs) y grupos de estudio (study groups o SGs) que desarrollan especificaciones para la industria del e-learning.

Los siguientes grupos de trabajo son parte de las actividades generales de la IEEE LTSC:

IEEE 1484.1 Architecture and Reference Model

IEEE 1484.3 Glossary

Los siguientes grupos de trabajo son parte de las actividades relacionadas con los datos y el metadata:

IEEE 1484.12 Learning Object Metadata

IEEE 1484.14 Semantics and Exchange Bindings

IEEE 1484.15 Data Interchange Protocols

Los siguientes grupos de trabajo son parte de las actividades relacionadas con los LMS y las aplicaciones:

IEEE 1484.11 Computer Managed Instruction

IEEE 1484.18 Platforms and Media Profiles

IEEE 1484.20 Competency Definitions

LTSC también trabaja en forma coordinada con otra iniciativa denominada ISO JTC1 SC36, que es un subcomité formado en forma conjunta por la ISO (International Standard Organization) y por la IEC (International Electrotechnical Commission), dedicado a la normalización en el ámbito de las Tecnologías de la Información para la formación, educación y aprendizaje.

IMS Global Learning Consortium, Inc.

Este Consorcio está formado por miembros provenientes de organizaciones educacionales, empresas públicas y privadas. Su misión es desarrollar y promover especificaciones abiertas para facilitar las actividades del aprendizaje on line.

El trabajo de la IEEE fue recogido por esta corporación privada creada por algunas de las empresas más importantes del sector. Su objetivo fue la creación de un formato que pusiese en práctica las recomendaciones de la IEEE y la AICC.

Lo que se hizo fue definir un tipo de fichero XML para la descripción de los contenidos de los cursos. De tal modo que cualquier LMS pueda, leyendo su fichero de configuración IMSMANIFEST.XML, cargar el curso.

A continuación se describen las principales iniciativas de este comité:

Learning Object Metadata (LOM)

Esta especificación entrega una guía sobre cómo los contenidos deben ser identificados o "etiquetados" y sobre cómo se debe organizar la información de los alumnos de manera de que se puedan intercambiar entre los distintos servicios involucrados en un sistema de gestión de aprendizaje (LMS). La especificación para metadata del IMS consta de tres documentos: IMS Learning Resource Meta-data Information Model, IMS Learning Resource XML Binding Specifications, IMS Learning Resource Meta-data Best Practices and Implementation Guide.

Empaquetamiento de Contenidos (Content Packaging)

Esta especificación provee la funcionalidad para describir y empaquetar material de aprendizaje, ya sea un curso individual o una colección de cursos, en paquetes portables e interoperables. El empaquetamiento de contenidos está vinculado a la descripción, estructura, y ubicación de los materiales de aprendizaje on-line, y a la definición de algunos tipos particulares de contenidos.

La idea es que el contenido desarrollado bajo este estándar sea utilizado en una variedad de sistemas de gestión de aprendizaje (LMS). Esta especificación ha sido comercializada por Microsoft bajo el nombre de LRN (Learning Resource Interchange).

Interoperabilidad de Preguntas y Tests (Question and Test Interoperability, QTI)

El IMS QTI propone una estructura de datos XML para codificar preguntas y test online. El objetivo de esta especificación es permitir el intercambio de estos tests y datos de evaluación entre distintos LMS.

Empaquetamiento de Información del Alumno (Learner Information Packaging, LIP)

Esta especificación define estructuras XML para el intercambio de información de los alumnos entre sistemas de gestión de aprendizaje, sistemas de recursos humanos, sistemas de gestión del conocimiento, y cualquier otro sistema utilizado en el proceso de aprendizaje. Actualmente, existen varios desarrolladores de productos que tienen en vista adoptar esta especificación.

Secuencia Simple (Simple Sequencing)

Esta especificación define reglas que describen el flujo de instrucciones a través del contenido según el resultado de las interacciones de un alumno con el contenido. Esta representación de flujo condicionado puede ser creada manualmente o a través de herramientas compatibles con esta especificación. Una vez creado, la representación de la secuencia puede ser intercambiada entre sistemas diseñados para entregar componentes instruccionales a los alumnos.

Diseño del Aprendizaje (Learning Design)

Este grupo de trabajo del IMS investiga sobre las maneras de describir y codificar las metodologías de aprendizaje incorporadas en una solución e-learning.

Repositorios Digitales (Digital Repositories)

El IMS está en el proceso de creación de especificaciones y recomendaciones para la interoperación entre repositorios digitales.

Definición de competencias (Competency Definitions)

El IMS (al igual que la IEEE) están en el proceso de crear una manera estandarizada de describir, referenciar e intercambiar definiciones de competencias. En esta especificación, el término competencia es usado en un sentido muy general, que incluye habilidades, conocimiento, tareas, y resultados de aprendizaje. Esta especificación entrega una manera de

representar formalmente las características principales de una competencia, independiente de su uso en un contexto en particular, permitiendo así su interoperabilidad entre distintos LMS.

Accesibilidad (Accessibility)

Este grupo de trabajo promueve el contenido de aprendizaje accesibles a través de recomendaciones, guidelines, y modificaciones a otras especificaciones. Tecnología accesible se refiere a la tecnología que puede ser usada sin tener acceso pleno a una o más canales de entrada y salida, usualmente visuales y auditivas.

ADL SCORM

Formada en 1997, la iniciativa ADL (Advanced Distributed Learning), es un programa del Departamento de Defensa de los Estados Unidos y de la Oficina de Ciencia y Tecnología de la Casa Blanca para desarrollar principios y guías de trabajo necesarias para el desarrollo y la implementación eficiente, efectiva y en gran escala, de formación educativa sobre nuevas tecnologías Web.

Este organismo recogió “lo mejor” de las anteriores iniciativas (el sistema de descripción de cursos en XML de la IMS, y el mecanismo de intercambio de información mediante una API de la AICC) y las refundió y mejoró en su propio estándar: SCORM, Shareable Content Object Reference Model (Modelo de Referencia para Objetos de Contenidos Intercambiables).

SCORM proporciona un marco de trabajo y una referencia de implementación detallada que permite a los contenidos y a los sistemas usar SCORM para “hablar” con otros sistemas, logrando así interoperabilidad, reusabilidad y adaptabilidad.

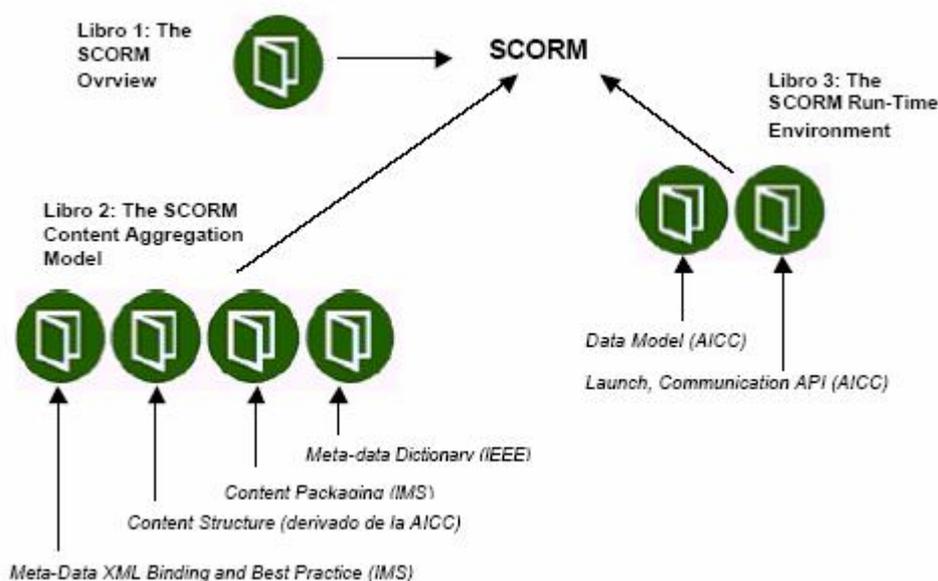
Todo esto se reafirma mediante las siguientes posibilidades:

- la disponibilidad de un Sistema de Gestión de Aprendizaje o LMS basado en Web para lanzar diferentes contenidos que se han desarrollado por varios autores usando herramientas de diversos vendedores,
- la disponibilidad de diversos LMS producidos por diferentes vendedores para lanzar un mismo contenido, y

- la disponibilidad de múltiples productos o entornos LMS basados en Web para acceder a un repositorio común de contenidos.

Las especificaciones de SCORM están organizadas como “libros” separados. La mayoría de estas especificaciones son tomadas desde otras organizaciones. Estos “libros” técnicos se agrupan bajo dos tópicos principales: Content Aggregation Model y Run-Time Environment.

Como lo muestra la siguiente figura, la actual versión 1.2 de SCORM ha sido dividido en tres libros que se detallan a continuación:



Libro 1: Scorm Overview. Contiene una descripción general de la iniciativa de ADL, un análisis de SCORM, y un resumen de las especificaciones técnicas contenidas en las siguientes secciones.

Libro 2: Scorm Content Aggregation Model. Contiene una guía para identificar y agregar recursos dentro de un contenido de aprendizaje estructurado. Este libro describe una nomenclatura para el contenido de aprendizaje, describe el SCORM Content Packaging (empaquetamiento de contenidos) y hace referencia al IMS Learning Resource Meta-data Information Model, el cual está basado en el IEEE LTSC Learning Object Metadata (LOM) Specification, que fue el resultado de un esfuerzo en conjunto entre el IMS Global Learning Consortium y la Alliance of Remote Instructional Authoring and Distribution Networks for Europe (ARIADNE).

Libro 3: Scorm Run-Time Environment: Incluye una guía para lanzar contenidos y hacerle un seguimiento en un ambiente basado en Web. Este libro es derivado del CMI001 Guidelines for Interoperability de la AICC.

SCORM también divide el mundo de la tecnología e-learning en componentes funcionales. Los principales componentes son: Learning Management System (LMS) y Sharable Content Objects (SCOs). SCO se refiere a objetos de aprendizaje reusables y estandarizados. Otros componentes en el modelo SCORM son herramientas que crean los SCOs y los ensamblan en unidades de aprendizaje más grandes (un curso por ejemplo).

SCORM Content Aggregation Model

El objetivo del modelo de agregación de contenidos de SCORM es proveer un medio común de componer contenidos educativos desde diversas fuentes compartibles y reusables. Define cómo un contenido educativo puede ser identificado, descrito y agregado dentro de un curso o una parte de un curso, y cómo puede ser compartido por diversos LMS o por diversos repositorios.

El modelo, incluye especificaciones para los metadata y el CSF (Content Structure Format):

- Los metadata (datos sobre los datos) constituyen la clave para la reusabilidad. Describen e identifican los contenidos educativos, de manera que pueden formar la base de los repositorios. Se han especificados basándose en las recomendaciones IEEE LSTC Learning Object Metadata (LOM). Los metadata se aplican a tres niveles: a los "assets" (elementos de contenidos de más bajo nivel), a los SCO (Sharable Content Objects) y bloques de SCO's, y al CSF.
- Content Structure Format. El proceso de diseño y creación de un curso comprende la construcción de un conjunto de objetos de contenidos educativos, relacionados entre sí mediante cierta estructura. Este es el objetivo del Content Structure Format (Formato para la Estructura de los Contenidos), proporcionar un medio de agregación de bloques de contenidos, aplicando una estructura y asociándola a una taxonomía para que tengan una representación y un comportamiento común en cualquier LMS.

El modelo CSF ha sido desarrollado a partir de las especificaciones CMI (Computer Manager Instruction) de la AICC. Posteriormente, realizando una reorganización entre las especificaciones de ADL, el AICC, el IEEE e IMS Global Learning Consortium, se ha llegado a un nuevo modelo representado en la IMS Content Packaging Specification, tal y como se expone a continuación.

Un CSF es un componente necesario para mover un contenido educativo de un lugar a otro, pero no es suficiente por sí mismo. Es necesario agregar y guardar los contenidos en un paquete. Para ello está diseñado el Content Packaging. Packaging o empaquetar, es el proceso de identificar todos los recursos necesarios para representar los contenidos educativos y después reunir todos los recursos junto a un manifiesto.

ADL señala en su nueva versión que el CSF de SCORM V1.1 no es adecuado para el empaquetamiento, y por lo tanto sus elementos han sido "remapeados" en dos nuevas estructuras, Content Aggregation Package Application Profile y Content Aggregation Manifest, que incluyen la mayoría de la información del anterior CSF pero que añade un nuevo método de inventariar todos los ficheros requeridos para distribuir los contenidos e identificar sus relaciones.

SCORM Run-Time Environment.

El objetivo del entorno operativo o de ejecución de SCORM es proporcionar un medio para la interoperabilidad entre los objetos compartibles de contenidos, SCO, y los sistemas de gestión de aprendizaje, LMS.

Un requerimiento de SCORM es que el contenido educativo sea interoperativo a través de múltiples LMS, sin tener en cuenta las herramientas que se usen para crear o usar los contenidos. Para que esto sea posible, debe existir un método común para lanzar un contenido, un método común para que los contenidos se comuniquen con el LMS y elementos de datos predefinidos que sean intercambiables entre el LMS y el contenido durante su ejecución.

Los tres componentes del entorno de ejecución de SCORM son:

1. El launch. Es el mecanismo que define el método común para que los LMS lancen un SCO basado en Web. Este mecanismo define los procedimientos y las responsabilidades para el establecimiento de la comunicación entre el contenido a mostrar y el LMS. El protocolo de comunicación está estandarizado a través del uso común del API.

El LMS puede implementar la presentación de los SCO del modo que se desee, por ejemplo, desarrollando un mecanismo de adaptación al usuario (mediante técnicas de aprendizaje simbólico), o bien, puede delegar esa responsabilidad al cliente permitiéndole que navegue por el curso libremente a través de menús.

2. La API (Application Program Interface). Es el mecanismo para informar al LMS del estado del contenido (por ejemplo si está inicializado, finalizado o en error) y es usado para intercambiar datos entre el LMS y los SCO (por ejemplo datos de tiempo, de puntuación, etc.). La API es simplemente un conjunto de funciones predefinidas que se ponen a disposición de los SCO, como por ejemplo LMSInitialize o LMSSetValue.
3. El Modelo de Datos. Es una lista estandarizada de elementos usados para definir la información a intercambiar, por ejemplo, el estado del contenido educativo. Son elementos de datos que tanto el LMS como el SCO van a conocer. Es responsabilidad del LMS mantener el estado de los datos requeridos a lo largo de las sesiones, y el SCO los utilizará en el caso de que se necesite su reutilización entre una sesión y otra.

Existen diversos modelos de datos en desarrollo en varias organizaciones. Incluyen datos sobre el perfil del estudiante, información de estado, iteraciones sobre test y preguntas, evaluaciones, etc. El Modelo SCORM está basado directamente del CMI Data Model del AICC.

Actualmente en versión 1.2, se espera que con la publicación de su versión 2.0 se convierta en el estándar más seguido.

IMS

El proyecto Instructional Management System (IMS), es un intento por conseguir una especificación para el desarrollo del potencial de Internet como entorno de formación.

IMS reúne un conjunto de organizaciones académicas, comerciales y gubernamentales que trabajan en construir la arquitectura de Internet para el aprendizaje. El proyecto fue fundado en 1997 y existe bajo los auspicios de EDUCAUSE's National Learning Infrastructure Initiative (EDUCASE 2002). El mismo grupo define así su misión: "El objetivo del proyecto IMS es la amplia adopción de especificaciones que permitirán que contenidos y entornos de aprendizaje distribuido de múltiples autores puedan trabajar juntos. A tal fin, El proyecto producirá una especificación técnica y un prototipo como prueba de conceptos.(IMS 1997)

No cabe duda de la importancia del trabajo que este grupo viene desarrollando de cara a la interoperabilidad que supondrá la adopción de sus especificaciones como un estándar de facto en la industria y en esa medida, el impulso que traduce para los Entornos Virtuales de Enseñanza y Aprendizaje en Internet.

Inicialmente la labor de IMS se desarrolló tomando como base la educación superior en EEUU, aunque hoy día sus especificaciones engloban gran variedad de contextos educativos, incluyendo formación corporativa y gubernamental.

Los primeros trabajos del IMS se centraron en la definición de un modelo y una arquitectura para los sistemas de aprendizaje distribuido. Sin embargo, sus esfuerzos se reorientaron rápidamente al percatarse de la necesidad de definir previamente un modelo de datos adecuado para describir los recursos, estructuras y demás elementos manejados por los componentes de la arquitectura.

A día de hoy, IMS define y desarrolla especificaciones interoperables usando XML para hacer posible el intercambio de contenidos educativos e información sobre los alumnos entre diferentes sistemas de enseñanza. Estas especificaciones se implementan con el objetivo de hacer más sencillo y más barato el desarrollo de material educativo. Podemos decir que las

especificaciones IMS son ya estándares *de facto* para la definición de requisitos educativos y para el desarrollo de productos y servicios relacionados con la educación.