

Edukalibre: Colaboración para la elaboración de material didáctico

Diego Chaparro, Luis López, Jesús M. González-Barahona,
{dchaparro,llopez,jgb}@gsync.escet.urjc.es
Grupo de Sistemas y Comunicaciones (GSyC). Universidad Rey Juan Carlos
Escuela Superior de Ciencias Experimentales y Tecnología.
C/ Tulipán s/n. 28933 - Móstoles (Madrid)

Abstract *The Libre Software collaborative development model has proved to be a powerful tool for the creation of software at many different scales and on multiple application domains. In this paper, we propose a system developed in the framework of the Edukalibre project, which intends to import and adapt this model for the creation and management of educational contents. This system supplies a set of tools for the creation of collaborative contents allowing version controlling, automatic conversions between multiple formats, diverse accessing methods and tools, etc. A fully functional version of the system is already available to download under a Libre Software license.*

1. Introducción

Las tecnologías web se han introducido en nuestro mundo de manera rápida invadiendo múltiples aspectos de nuestras vidas. Esta revolución está teniendo un impacto notable sobre el modo en que trabajamos, nos relacionamos y aprendemos. En la actualidad, es posible acceder a multitud de diferentes tipos de contenidos en cualquier instante y desde cualquier lugar. Por este motivo, numerosas instituciones, organizaciones y empresas del mundo de la educación están explorando las posibilidades de aplicar estas tecnologías de la información en sus procesos educativos [1, 16, 7].

En la actualidad, la mayoría de estos sistemas se utilizan como complemento a las técnicas tradicionales de enseñanza, por lo que se basan en el modelo tradicional profesor/instructor frente al de alumno/estudiante. Por este motivo, la mayor parte de los contenidos educativos que se pueden encontrar en la web han sido diseñados para un modelo de tele-enseñanza a través de tele-lectura y consisten en estructuras monolíticas y fijas, secuenciadas y limitadas que no están dotadas de flexibilidad para la realización de actualizaciones y que son difíciles de organizar y reutilizar [18].

Debido a esto, existe un gran interés en las comunidades científicas y académicas por concebir nuevos modelos y metodologías que permitan disponer de mecanismos telemáticos eficientes para la creación, compartición y reutilización de recursos educativos en la web. En particular, se ha detectado la necesidad de incidir en nuevos paradigmas que posibiliten el e-learning basándose en modelos más colaborativos y proactivos.

En este artículo proponemos un sistema de tele-enseñanza que desarrolla estas ideas utilizando para ello un conjunto de metodologías y técnicas

derivadas de las que se usan en el desarrollo de Software Libre. Estos métodos han modificado enormemente el modo en que el software se produce y se distribuye [2, 6]. La idea fundamental se basa en la presencia de una comunidad de participantes que comparten experiencias, conocimiento y código fuente en un proceso colaborativo en el que cada individuo ayuda con lo que puede. Ha sido aplicada con éxito en múltiples dominios incluyendo el desarrollo de sistemas operativos (Debian, FreeBSD, Fedora), entornos de escritorio (GNOME, KDE), navegadores y servidores web (Mozilla, Firefox, Apache) o aplicaciones ofimáticas (OpenOffice.org) [4, 5, 15, 11]. Por este motivo, hoy en día está aceptado que las comunidades del software libre han producido métodos revolucionarios en el ámbito de la ingeniería del software [14, 10].

Es importante destacar que existen multitud de similitudes entre el modelo de desarrollo del software libre y algunas metodologías modernas de aprendizaje basadas en la cooperación. De hecho, algunos autores proponen explícitamente que las actividades que tienen lugar en las comunidades del software libre pueden ser vistas como un proceso de aprendizaje en sí mismo, en el que las partes involucradas contribuyen y aprenden del resto [3]. Consiguientemente, es de esperar que la aplicación de dicho modelo puede tener un importante impacto sobre el modo en que las tecnologías web se utilizan para enseñar y aprender. Por este motivo, creemos que es extremadamente importante que las comunidades académicas y científicas que desarrollan o utilizan sistemas telemáticos para la enseñanza conozcan las potencialidades de este modelo.

En la actualidad, existe un creciente interés por importar algunos aspectos de los modelos de desa-

rollo del software libre en los procesos de aprendizaje. Los primeros pasos los han dado algunas instituciones educativas de gran prestigio tales como el MIT [12], la Carnegie Mellon University [8] o Harvard [13]. Estas universidades han sobrepasado el esquema tradicional de pensamiento (de acuerdo con el que los materiales sólo están disponibles para los alumnos matriculados en los cursos correspondientes) y proporcionan un acceso completamente libre a su material educativo de alta calidad a través de la web. Algunos aspectos económicos o tecnológicos pueden haber influido en esta decisión, incluyendo beneficios de marketing, incremento de la reputación, mayor difusión de la innovación en la sociedad, posibilidad de obtener realimentación de profesionales y estudiantes de todo el mundo, etc. Esta idea de abrir los contenidos educativos se está desarrollando con gran entusiasmo en un importante número de proyectos entre los que se encuentran MIT OpenCourseware [12], Open Learning Initiative [8], etc.

La idea de hacer los contenidos universalmente accesible podría producir una revolución en la educación similar a la que se produjo en el mundo del software cuando se introdujo el software libre. Sin embargo, para que esta revolución realmente pueda tener lugar, simplemente hacer que los contenidos educativos sean libremente accesibles no es suficiente. El software libre no puede ser comprendido exclusivamente en función de la accesibilidad del código fuente. Su éxito se basa también en la existencia de mecanismos por los que los diferentes agentes se coordinan y obtienen un beneficio a través de la cooperación. El proceso por el que esta comunidad surge y se forman sinergias es complejo y no se comprende plenamente. Sin embargo, nuestra hipótesis es que suministrando facilidades similares a las que están disponibles para el desarrollo de código, se podría producir un fenómeno análogo en el ámbito de la generación de contenidos educativos. Es de destacar que esta hipótesis es razonable puesto que algunas iniciativas exitosas tales como Wikipedia [19] están basadas en la misma filosofía.

La idea de producir aprendizaje colaborativo en una comunidad no es nueva. Algunos autores [17, 9] describen una comunidad dinámica de aprendizaje como aquella en la que el control se distribuye entre miembros autónomos que pueden involucrarse de manera flexible en actividades de aprendizaje de modo negociado y dialogado, interactuando y colaborando con un claro compromiso hacia la generación y compartición de nuevos conocimientos. En estos trabajos, se reconoce que, a través de la promoción de la creatividad, la innovación, la colaboración y proporcionando mecanismos que permitan la modificación de contenidos, la comunidad es capaz de diagnosticar y solucionar la mayoría de las necesidades educativas que pue-

dan surgir. Sin embargo, no se establece de manera clara cómo es posible alcanzar estos objetivos

Nuestra propuesta es la de integrar la filosofía del software libre en la educación para lograrlo. Hasta hace poco, el estado de la tecnología hacía muy difícil soportar el desarrollo de contenidos educativos contruidos de manera colaborativa entre grandes grupos de profesores y alumnos. La comunidad del software libre ha creado multitud de herramientas que permiten que la colaboración pueda llevarse a cabo entre personas en diferentes lugares e instantes. Sin embargo, estas tecnologías no son fácilmente aplicables en el entorno educativo debido a diversos motivos. En primer lugar, no son lo suficientemente intuitivas para ser utilizadas por profesores y estudiantes medios. En segundo, están diseñadas para realizar tareas parciales, lo que se adapta al perfil de los desarrolladores que conocen y pueden utilizar múltiples herramientas complementarias, pero que no es satisfactorio en comunidades educativas.

En este contexto, está claro que son necesarias nuevas herramientas que permitan realizar de manera efectiva el desarrollo colaborativo de cursos educativos abiertos. Una solución basada en la web parece ser la elección más razonable para implementarlas por dos motivos. En primer lugar, esta tecnología ofrece la posibilidad de integrar todo tipo de contenidos y formatos. En segundo, está muy extendida y puede ser utilizada por un usuario medio sin ningún tipo de formación especializada. Más aun, estas herramientas deberían ser también software libre para garantizar que puedan ser adaptadas y desplegadas utilizando el mismo concepto abierto de colaboración. En este artículo, proponemos una aplicación que cumple todos esos requisitos y que permite la concepción de contenidos educativos colaborativos en la web. Esta aplicación ha sido desarrollada dentro del marco del proyecto Edukalibre, financiado por la Comisión Europea bajo el programa Socrates/Minerva¹. El proyecto comenzó en Octubre de 2003 y se espera que termine en Diciembre de 2005. Es coordinado por la Universidad Rey Juan Carlos e incluye como partners a la Universidad de Leeds, la Universidad de Porto, y la Universidad de Karlsruhe. La página web de proyecto se encuentra en <http://www.edukalibre.org>.

El Sistema Edukalibre se describe en el resto de este artículo. En primer lugar describiremos la arquitectura básica del mismo. Acto seguido se presentarán los diferentes módulos que implementan cada una de las funcionalidades, así como las interfaces a través de las que interactúan. Finalmente se expondrán unas breves conclusiones.

¹http://europa.eu.int/comm/education/programmes/socrates/minerva/ind1a_en.html

2. Arquitectura modular del sistema

La arquitectura de la plataforma edukalibre se basa en un sistema de módulos intercambiables, de modo que cada uno de ellos puede ser substituido por otro de características similares sin necesidad de modificar el resto. También es posible añadir nuevos módulos de forma sencilla, permitiendo que estos se integren con el resto a través de una interfaz predefinida.

Los módulos se agrupan en varios niveles, cada uno de los cuales debe proporcionar una funcionalidad preestablecida. En la actualidad, el sistema está constituido en base a dos niveles. El nivel inferior, sobre el que se sustenta el resto de la aplicación, se basa en un sistema de control de versiones de ficheros. Este proporciona las funcionalidades básicas de gestión de la historia de los documentos. El nivel superior está formado por varios interfaces de usuario que permiten realizar diferentes tareas, que van desde las más básicas y habituales hasta las más avanzadas restringidas a usuarios expertos. La figura 2 muestra un esquema de la arquitectura y la interrelación entre los módulos.

La materia prima de la que se nutre el sistema son los documentos. Por tanto, para comprender en detalle cómo interaccionan los diferentes componentes de la aplicación, hay que comprender con precisión qué entendemos por un documento y cómo estos se gestionan. La sección siguiente está dedicada a este objetivo.

2.1. La noción de documento

El documento es la unidad funcional en la que se basa el sistema. En ellos, los usuarios pueden introducir y recuperar información. Para esto, se ofrecen un conjunto de operaciones que permiten manipularlos. Los documentos se pueden clasificar en varios tipos dependiendo de su formato

- *Documentos base*: son documentos que pueden ser editados por los usuarios para crear un nuevo documento, o generar una nueva versión de un documento preexistente. Para que un documento base sea aceptado por el sistema, su formato debe pertenecer al conjunto de formatos base. Los elementos de dicho conjunto han sido seleccionados gracias a sus propiedades y ventajas. En la actualidad se soportan los siguientes: DocBook, LaTeX y OpenOffice. Los dos primeros se representan mediante texto plano (no son ficheros binarios) lo que proporciona ventajas a la hora de almacenarlos en un sistema de control de versiones y también para extraer información de los mismos. El tercero, aunque en principio no se representa como un fichero de texto plano, en realidad es un fichero comprimido que contiene, entre otras

cosas, varios ficheros XML con toda la información del documento. Por tanto, también puede procesarse de manera similar a los dos anteriores.

- *Documentos finales*: Los documentos finales son aquellos cuyo formato se obtiene a partir de los documentos base. Una característica destacable del sistema es que se generan conversiones de los documentos base a los finales automáticamente cada vez que se introduce un nuevo documento o se genera una nueva versión de uno existente. Los formatos para los documentos finales se han elegido para facilitar la distribución y el uso de los mismos. Por ejemplo, algunos de los que se generan actualmente son:

- Aptos para ser impresos:
 - Postscript
 - PDF
- Aptos para ser publicados en la web:
 - HTML múltiple
 - HTML único
- Totalmente portables:
 - Texto plano

Es importante destacar que el sistema Edukalibre proporciona un nivel de abstracción superior al de otros sistemas similares debido a que la unidad de trabajo es el documento y no el simple fichero. En este sentido, un documento puede constar de múltiples ficheros que pueden contener el texto del documento, imágenes, hojas de estilo para realizar las conversiones de formatos, etc.

2.2. Nivel 0: Núcleo del sistema

Este nivel está formado por un conjunto de módulos que proporcionan la funcionalidad principal al sistema. En él se almacenan todos los documentos, tanto los que están en formato base como los que se encuentran en formatos finales. También en este nivel se realizan las conversiones entre estos formatos y se extrae automáticamente un conjunto de informaciones que son relevantes para las capas superiores a la hora de interaccionar con los usuarios (p.e. título, autores, etc.) Cada una de estas funcionalidades es implementada por uno o varios módulos específicos. A continuación explicaremos en detalle en qué consiste cada uno de ellos.

2.2.1. Módulo: Repositorio de documentos base

Es el módulo principal de este nivel, y está formado básicamente por un sistema de control de

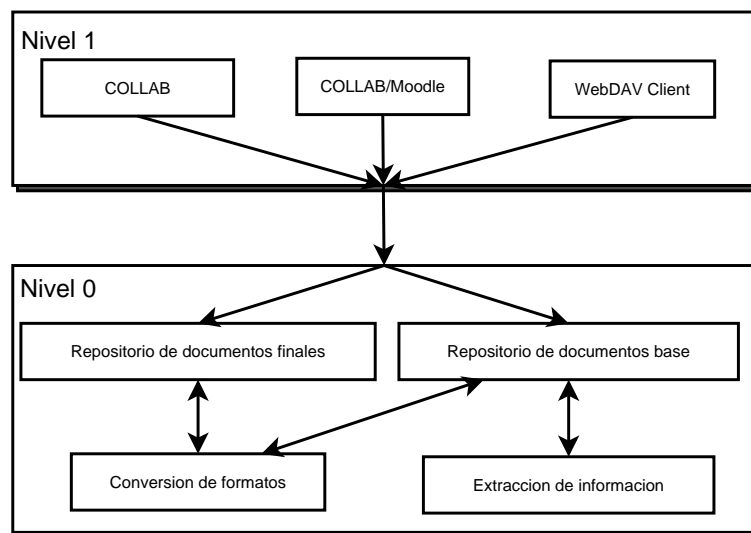


Figura 1: Arquitectura del sistema

versiones. En la versión actual se utiliza subversion² pero el sistema está preparado para adaptarse fácilmente a otro tipo de soluciones tales como CVS³ o GNUArch⁴. El repositorio es capaz de almacenar un registro que contiene todas y cada una de las diferentes versiones de cada uno de los documentos que se hayan insertado en el sistema. A través de su interfaz, se pueden obtener los datos necesarios para poder llevar a cabo tareas tales como recuperar estados anteriores, hallar las diferencias entre dos versiones, analizar el grado de contribución de los diferentes autores y determinar en qué partes del documento se ha producido dicha contribución, etc.

Además, el sistema de control de versiones implementa un mecanismo que permite añadir pequeños programas que se ejecutan cuando se realiza una determinada tarea sobre el repositorio. En la actualidad, estos programas se dividen en dos fases coincidiendo con dos de las operaciones que realiza subversion: el "pre-commit" el "post-commit".

- El conjunto de tareas asociadas a la *operación "pre-commit"* se ejecutan cuando se trata de subir un documento, pero éste no ha sido comprometido aún en el repositorio. En esta fase, se efectúan tareas de validación de documentos, que consisten básicamente en la comprobación sintáctica de los documentos base que se intentan introducir en el sistema. Por ejemplo, en el caso de documentos DocBook, la validación se producirá contra la DTD correspondiente. Esta tarea es importante, porque si un documento no respeta la sintaxis correcta de su especificación, hay otras tareas posteriores que no funcio-

narán correctamente como la conversión automática de formatos. En esta fase, se puede actuar de dos formas cuando un documento no es válido, o se aborta la operación mostrando el mensaje de error al usuario, o se continúa la operación y al final se le muestra un mensaje de aviso al usuario indicándole los posibles problemas.

- La segunda fase está asociada a la *operación "post-commit"* del repositorio, la cual ejecuta el código asociado cuando el documento ya se ha subido al sistema y ya no se puede abortar esta operación. Esta segunda fase se encarga de controlar el flujo de control del resto de módulos de este nivel, gestionando la ejecución de los mismos. Primero se ocupa de extraer información sobre el documento que ya se ha comprometido en el repositorio, y después activa el módulo encargado de realizar las conversiones de formatos sobre ese documento.

2.2.2. Módulo: Repositorio de documentos finales

Además del repositorio base, existe otro repositorio destinado a almacenar los documentos en sus formatos finales. Éste último se encuentra integrado con un servidor HTTP, que en la versión actual es Apache 1.3. De este modo, es posible acceder mediante este protocolo a toda la historia de cada uno de los documentos en su formato final.

El motivo por el que se ha separado el almacenamiento de documentos base y finales en dos repositorios diferentes es claro. Los documentos base son los únicos susceptibles de ser editados por los

²<http://subversion.tigris.org>

³<https://www.cvshome.org/>

⁴<http://www.gnuarch.org>

usuarios. Ya hemos visto que los formatos base se basan todos en ficheros de texto, lo que permite utilizar plenamente todas las funcionalidades de los sistemas de control de versiones. Por el contrario, los documentos finales suelen ser ficheros binarios, que el usuario final no puede modificar y que cambian de manera automática cuando sus correspondientes bases son alteradas. En este caso, es suficiente utilizar un servidor HTTP que permita descargar directamente los correspondientes ficheros bajo demanda. Por estos motivos, las funcionalidades que se requieren en ambos casos por parte del repositorio son diferentes y parece lógico aislar ambas alternativas.

2.2.3. Módulo: Conversión de formatos

La conversión de formatos es un módulo clave de este nivel, dado que genera directamente el producto que esperan los usuarios a través de un mecanismo automático que se activa cada vez que un nuevo documento o una nueva versión se introducen en el sistema.

El diseño de este módulo ha sido cuidadosamente estudiado, debido a que los procesos que se llevan a cabo en el mismo pueden suponer una carga computacional elevada. Dependiendo de la complejidad de los documentos que se traten, es posible encontrar que ciertas conversiones de formatos pueden llegar a tomar un tiempo respetable para poder llevarse a cabo. Este factor ha de ser tenido en cuenta para que no interfiera en la interacción con el usuario.

Las conversiones desde el formato base DocBook se realizan utilizando un procesador XSLT, para lo que se ha diseñado una hoja de estilo para cada uno de los formatos que generamos. Los procesadores de hojas de estilo utilizados son *xslt-proc* y *fop*. Usando este mecanismo, se generan los siguientes formatos finales: HTML único, HTML múltiple, texto plano, PDF, Postscript, LaTeX y OpenOffice. Algunas de las hojas de estilo XSL usadas están disponibles desde hace tiempo para los procesadores que hemos citado aunque, en algunos casos, estas se han modificado para adaptarlas a nuestras necesidades.

Las conversiones desde el formato base LaTeX se realizan utilizando las herramientas clásicas disponibles en el mundo del software libre para realizar conversiones desde LaTeX: *latex*, *dvipdf*, *dvips* y *hevea*. Con estas herramientas conseguimos los siguientes formatos finales: DVI, HTML único, HTML múltiple, PDF y Postscript.

Y las conversiones del formato base OpenOffice se realizan utilizando el propio editor OpenOffice.org en modo comando. Los formatos finales que es posible obtener en este caso son los siguientes: HTML único, PDF, Microsoft Word y texto plano.

Evidentemente, hemos descrito solamente una pequeña muestra de la gran colección de conversiones que se podrían realizar de forma sencilla

utilizando herramientas ya disponibles desde cada uno de los formatos base. El sistema permite añadir fácilmente nuevos formatos finales que se generan desde los base, e incluso formatos finales que se generan desde otros formatos finales. En la actualidad seguimos trabajando en ampliar la lista de formatos finales para cada uno de los formatos base lo máximo posible, de modo que se puedan satisfacer las necesidades de todos los usuarios.

2.2.4. Módulo: Extracción de información de documentos

Otra característica interesante del sistema es que proporciona a los usuarios información acerca de cada documento sin necesidad de tener que acceder al propio documento. Para ello, se extraen ciertos campos de cada documento a partir de los formatos base que se introducen en el sistema. Actualmente este proceso sólo se realiza sobre los documentos base de tipo DocBook/XML, que se someten a un parseado en el que se localizan las etiquetas necesarias en el fichero. La implementación de esta funcionalidad en los otros formatos base está lista en breve.

La información que se extrae actualmente de los documentos es la siguiente:

- Título
- Abstract
- Autor

Esta información se almacena en el repositorio de documentos finales, junto con otros datos interesantes como la fecha y hora en la que se ha subido el documento al sistema, el usuario que lo ha subido, el nombre de la revisión del documento, etc.

2.3. Nivel 1: Interfaces del sistema

El sistema puede ser accedido por los usuarios desde diferentes interfaces y a través de diferentes protocolos de comunicaciones. Hemos desarrollado algunas interfaces de forma explícita para ser usadas sobre nuestro sistema; entre ellas COLLAB y COLLAB/Moodle que explicamos en detalle en esta sección. Sin embargo, también es posible utilizar interfaces genéricas y estandarizadas como el protocolo WebDAV, que permite acceder al repositorio de documentos base, o el HTTP que proporciona el repositorio de documentos finales.

De este modo, proporcionamos a los usuarios diversas alternativas a la hora de acceder al sistema. Por un lado, los usuarios medios pueden descargar los documentos finales utilizando navegadores web convencionales o pueden utilizar un cliente WebDAV para realizar tareas sencillas sobre los base. Por otro lado, existen mecanismos de acceso que requieren un nivel de conocimiento más técnico, como el interfaz que proporciona Subversion con sus herramientas de cliente tradicionales, que

pueden ser usadas como con un servidor de Subversion tradicional por usuarios más avanzados.

2.3.1. Módulo: COLLAB

Este es uno de los interfaces de usuario que hemos desarrollado para acceder al sistema y que permite a los usuarios realizar las tareas más comunes sobre el sistema Edukalibre de una forma sencilla.

Básicamente COLLAB es un interfaz web escrito en PHP que puede ser integrado de modo sencillo en cualquier portal escrito en este lenguaje. COLLAB proporciona la funcionalidad suficiente para que cualquier usuario pueda realizar las acciones más comunes sobre el sistema, tal y como se muestra a continuación. Los usuarios avanzados que requieran funcionalidad no contemplada en este interfaz, pueden usar otras herramientas más genéricas como se muestra en el apartado 2.3.3.

- **Autenticación:** COLLAB soporta un método de autenticación para acceder a ciertos aspectos de su funcionalidad restringida a usuarios registrados
- **Acceso a documentos:** mediante este interfaz podemos ver los documentos almacenados en el sistema e información sobre los mismos.
- **Historia de documentos:** se puede observar la historia de cada documento, mostrando cada una de las versiones, con su fecha, la persona que la subió al sistema, y enlaces para acceder a cada versión del documento en su formato base o en los finales.
- **Introducir nuevos documentos o actualizar documentos existentes:** mediante este interfaz se puede subir al sistema un nuevo documento o una nueva versión de un documento existente utilizando simplemente un formulario web.
- **Edición on-line:** Proporciona un sencillo interfaz para poder editar mediante el navegador los documentos (en formato base DocBook o LaTeX).

Puede verse la apariencia del interfaz COLLAB en la Figura 2 .

2.3.2. Módulo: COLLAB/Moodle

El uso de los Sistemas de Gestión de Contenidos (CMS) se está extendiendo por muchos centros educativos en la actualidad. Por un lado es utilizado para gestionar cursos de enseñanza a distancia, y por otro lado como apoyo a la gestión de los cursos presenciales tradicionales. Moodle es uno de los CMS que se está extendiendo en la comunidad educativa gracias a sus ventajas y funcionalidades y también gracias a la licencia GPL con la que se distribuye.

COLLAB/Moodle es una adaptación del interfaz COLLAB que puede ser usado como un recurso dentro de los cursos de Moodle. Se ha desarrollado de forma que pueda ser instalado de forma sencilla, ya que se empaqueta como un módulo Moodle y está totalmente integrado con el CMS tanto en funcionalidad y gestión de la base de datos, como en la apariencia, gracias a las hojas de estilo de Moodle.

En la Figura 2 puede verse la apariencia del interfaz COLLAB/Moodle.

2.3.3. Interfaces genéricas

Uno de los objetivos perseguidos en el diseño del sistema es que pueda ser accesible desde el mayor número de herramientas que sea posible. Para lograrlo, además de implementar varios interfaces de usuario como COLLAB y COLLAB/Moodle, el sistema soporta la utilización de protocolos de comunicaciones estándar.

De este modo, los repositorios pueden ser accedidos usando los protocolos HTTP (Hyper Text Transfer Protocol), y WebDAV (Web-based Distributed Authoring and Versioning) que es una extensión de HTTP que permite editar y gestionar ficheros en servidores remotos.

- **Accesos mediante HTTP:** Se puede acceder al sistema mediante el protocolo HTTP directamente sobre el repositorio de documentos base y sobre el repositorio de documentos finales. Mediante este acceso podemos realizar tareas de lectura sobre estos repositorios; es decir, podemos acceder a los documentos base y finales, pero solo en modo lectura. No podemos modificar los documentos ni subir otros nuevos.
- **Accesos mediante WebDAV:** Mediante el acceso WebDAV podemos realizar las mismas acciones que utilizando HTTP. Sin embargo, en estecaso, además es posible llevar a cabo otras operaciones utilizando las ampliaciones que proporciona WebDAV. Así, es posible acceder a propiedades de los ficheros (método PROPFIND), crear directorios (método MKCOL), copiar y mover ficheros (métodos COPY y MOVE), acceder a la historia de un documento o mezclar varias versiones de un documento.

Para acceder mediante el protocolo HTTP, podemos usar cualquier cliente que lo soporte, incluyendo navegadores web (Mozilla, Netscape, etc.) o editores que permitan leer los documentos desde un servidor HTTP (casi cualquier editor lo permite), pero recordando que sólo podemos acceder en modo lectura, es decir, no podremos guardar el documento en el repositorio usando este protocolo.

El acceso mediante WebDAV se puede realizar mediante un explorador de ficheros que lo soporte

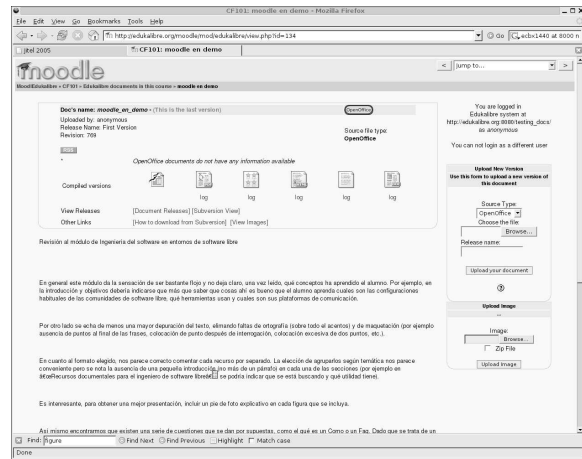
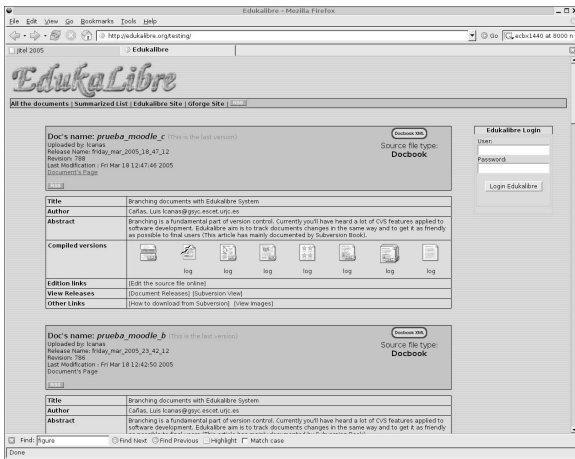


Figura 2: Página principal de COLLAB e interfaz de COLLAB/Moodle

(como Nautilus en GNOME o cadaver⁵ que es un cliente WebDAV en modo texto). También puede realizarse utilizando directamente un editor compatible como OpenOffice.org⁶ o emacs⁷.

Además, el sistema tiene disponible un interfaz del tradicional ViewCVS,⁸ que presenta mediante un interfaz web información sobre los documentos almacenados en el repositorio de documentos base, mediante el que se pueden mostrar diferencias entre versiones de un fichero, historia de versiones de un fichero, etc. Una de las diferencias entre este interfaz y el interfaz que proporciona COLLAB es que el primero está centrado en ficheros, mientras que el segundo trabaja sobre la abstracción de documentos, que a su vez pueden estar compuestos de uno o varios ficheros.

3. Conclusiones

En este artículo hemos descrito el Sistema EdukLibre, una nueva aplicación que permite el desarrollo colaborativo de contenidos educativos y que se basa en las metodologías de desarrollo de software libre. Hemos presentado las principales características del sistema así como su arquitectura básica. Hemos mostrado que este tipo de soluciones satisface plenamente las necesidades que pueden surgir a la hora de crear este tipo de contenidos. Por este motivo, proponemos que la herramienta puede tener aplicaciones interesantes dentro del ámbito de la tele-educación, por lo que creemos que la comunidad de la Ingeniería Telemática debe conocer su existencia. El sistema ha sido desarrollado utilizando herramientas libres y ha sido liberado bajo una licencia también libre. Por tanto, puede ser descargado, utilizado y distribuido con todas las ventajas asociadas a este tipo de modelo.

⁵<http://webdav.org/cadaver/>

⁶<http://www.openoffice.org/>

⁷<http://www.gnu.org/software/emacs/>

⁸<http://viewcvs.sourceforge.net/>

Agradecimientos

El trabajo descrito en este artículo ha sido financiado en parte por el programa Socrates/Minerva de la Comisión Europea, bajo el número de ayuda 110330-CP-1-2003-IES-MINERVA-M.

Referencias

- [1] H.H. Adelsberger, B. Collis, and J.M. Pawlowski. *Handbook on Information Technologies for Education and Training*. Springer, 2002.
- [2] Nikolai Bezroukov. Open source software development as a special type of academic research. *First Monday*, 4(10), October 1999.
- [3] K. Edwards. Epistemic communities, situated learning and open source software development. <http://opensource.mit.edu/papers/kasperedwards-ec.pdf>.
- [4] Daniel M. German. The GNOME project: a case study of open source, global software development. *Software Process Improvement and Practice*, pages 201–215, August 2003.
- [5] Jesús M. González-Barahona, Luis López, and Gregorio Robles. Community structure of modules in the apache project. In *Proceedings of the 4th Workshop on Open Source Software Engineering. 26th International Conference on Software Engineering*, Edinburgh, Scotland, UK, May 2004.

- [6] Ahmed E. Hassan, Michael W. Godfrey, and Richard C. Holt. Software engineering research in the bazaar. In *Proceedings of the 2nd Workshop on Open Source Software Engineering at the 24th International Conference on Software Engineering*, May 2001.
- [7] W. Horton. *Designing Web-Based Training*. Wiley Computer Publishing, 2000.
- [8] Carnegie Mellon Open Learning Initiative. <http://www.cmu.edu/oli/>.
- [9] P. Irvine and P. Brna. Growing an internet-based community for lifelong self-learners: empowering the community. *International Journal of Continuing Engineering Education and Lifelong Learning*, 13(1/2):21–21, 2003.
- [10] Stefan Koch, editor. *Free/Open Source Software Development*. Idea Group, Inc., 2004.
- [11] Christoph Lameter. Debian gnu/linux: The past, the present and the future. In *Free Software Symposium 2002*, October 2002.
- [12] MIT OpenCourseWare. <http://ocw.mit.edu/index.html>.
- [13] Harvard University Library Open Collections Program. <http://ocp.hul.harvard.edu>.
- [14] Eric S. Raymond. The cathedral and the bazaar. *First Monday*, 1997. http://www.firstmonday.dk/issues/issue3_3/raymond/.
- [15] Christian Robottom Reis and Renata Pontin de Mattos Fortes. An overview of the software engineering process and tools in the Mozilla project. In *Workshop on Open Source Software Development*, February 2002.
- [16] J.M. Rosenberg. *E-Learning*. McGraw-Hill, 2001.
- [17] M. Simonson. Dynamic learning communities: an alternative to designed instructions. In *Proceedings of Selected Research and Development Presentations, Washington D.C.: Association for Educational Communications and Technology*, pages 800–809.
- [18] V. Uskov. A 3rd generation web-based instructional tool for education and lifelong training. *International Journal of Continuing Engineering Education and Lifelong Learning*, 13(1/2):110–131, 2003.
- [19] Wikipedia. http://en.wikipedia.org/wiki/Main_Page.