

Diseño y implementación de un editor colaborativo de LaTeX vía web

Jesús Ojeda Contreras

Ingeniero Técnico en Informática de Gestión

Trabajo / Proyecto dirigido por: Josep Maria Ribó Balust

Departament d'informàtica i enginyeria industrial

Escola Politècnica Superior, c/Jaume II, 69 25001 Lleida

973702781 josepma@diei.udl.es

973203528 jojeda1@alumnes.udl.es

Resumen

Se pretende con este TFC el crear una aplicación web; esto es, accesible desde cualquier navegador de internet, que permita a múltiples usuarios la edición simultánea de documentos escritos bajo lenguaje LaTeX [1], un lenguaje de preparación de documentos ampliamente utilizado en el mundo científico y académico.

El resultado final es la propia aplicación totalmente funcional.

1. Introducción

Ya existían editores colaborativos previamente. Pero ninguno de ellos permite la edición con lenguaje LaTeX.

Para verlo de cerca, indagué un poco acerca de los editores más extendidos como MediaWiki [2], Zoho Writer [3], Writely [4], SynchroEdit [5], etc.

Este defecto o falta, impide o como mínimo complica ligeramente las cosas a aquellos equipos de científicos, matemáticos o en general investigadores que trabajen sobre un mismo artículo/escrito pero no se hallen en el mismo lugar (ni siquiera en el mismo huso horario). Es por esta razón que se decide hacer este trabajo. Así mismo, queríamos que todos ellos pudiesen trabajar concurrentemente y de una manera accesible desde cualquier lugar sin necesitar nada más que un simple navegador web. Dado que todos los editores existentes se basan en la edición de texto enriquecido (de manera parecida a como lo hacen editores como MS Word o OpenOffice.org), LaTeX necesita un tratamiento diferente al ser un lenguaje de marcas. Así pues, el editor que realicé se construyó partiendo de cero, aprovechando tan sólo porciones de código de programas como dhtmlXtree [6] o FCKEditor [7] dado que estos tenían licencias de software libre.

De la misma manera, todo el programa se libera sobre la licencia GPL. De esta manera aseguramos que el código

esté disponible de manera abierta y así cualquier persona pueda aportar mejoras o incluso continuar el proyecto si este quedase sin actividad en algún momento.

Por otro lado, y por interés personal, quería aprender sobre nuevas tecnologías ampliamente extendidas y bastante de moda últimamente, como son AJAX[8] y DHTML.

2. Desarrollo del trabajo

Dada la complejidad del proyecto, era necesario hacer un buen diseño inicial que nos permitiese hacer un mejor mantenimiento y ampliaciones más sencillas, sin necesidad de cambiar mucho código de una versión a otra.

Así pues, lo primero fue hacer un análisis de requerimientos. De esta manera, determinaríamos las necesidades, requisitos y funcionalidades esenciales de la aplicación. Dado el problema inicial de la edición de documentos de manera distribuida, debemos evitar la multiplicidad del documento y además los posibles errores que se pueden derivar de esto, como son incoherencias e inconsistencias en el sistema. Este problema se puede solventar de diversas maneras como pueden ser control de versiones o bloqueo de partes del documento. Para este trabajo se eligió esta segunda opción.

Esto llevó a la identificación de los usuarios, los documentos y las partes de los documentos como agentes que intervendrían en todo el proceso de edición, y de hecho serían en lo que se basaría todo el trabajo.

En el momento en que se conocen los elementos que intervendrían, se buscan las tecnologías y productos adecuados que nos permitan llevar a cabo el trabajo. Personalmente siempre se buscó soluciones de código abierto. Ya que hablamos de una aplicación web, necesitábamos un servidor, y dado que el lenguaje de programación deseado era Java[9], se utilizó el servidor Tomcat[10]. Así pues, se utilizó JSP[11] y el framework Struts[12] para facilitar ciertas acciones y mejorar la

legibilidad del código. Como sistema gestor de base de datos donde guardar toda la información se utilizó MySQL[13].

Así pues, y llegado a este punto, era el momento de modelizar el sistema y hacer el diseño.

Partiendo de los requisitos, se hizo el diseño de la base de datos y para el diseño del código del programa se optó por basarnos en uno de los patrones más utilizados para el diseño de cualquier aplicación que conste de interfaz gráfica, el patrón MVC (Modelo Vista-Controlador). Dado que nuestra aplicación es web, este patrón se adapta a este sistema mediante otro patrón, el Front Controller.

Así pues, definimos los documentos de una manera virtual ya que se guardarán separados por partes en la base de datos para facilitar el control de acceso y edición, y así evitar errores. Además, cada usuario tiene un directorio asignado, que también será virtual, el cual estará definido en un archivo XML y que manipularemos mediante JDOM[14] y XPATH[15]. La modificación de este archivo, y por tanto la estructura arbórea que verá el usuario, se decidió que fuese local a cada usuario, es decir, sólo el propio usuario puede modificar su directorio. Esto es así para impedir problemas de acceso concurrente al archivo. La limitación que esto implica es que el usuario no podrá ver que le han añadido o eliminado de la edición de un documento que no sea suyo, a menos que cierre la sesión y vuelva a entrar.

Por otro lado, los documentos se separan por partes como ya he dicho, y la forma de mantener su estructura es inherente en la base de datos, las claves identificadoras de cada parte ya identifican también cuál es la posición de esa parte dentro del texto. Así, para obtener el texto del documento completo, tan sólo hay que hacer un recorrido sobre las claves.

3. Resultados

De esta manera, y después de toda la programación requerida, la aplicación es totalmente funcional.

Nos encontramos ante un sistema en el que el usuario se registra (la aplicación le envía un correo electrónico mediante JavaMail [16] para recordarle los datos que puso en el registro).

Cuando el usuario ya está registrado, puede acceder a la aplicación. Y lo que encuentra es un gran editor de texto a la derecha y una barra de herramientas a la izquierda además del directorio virtual y otro par de paneles más. Mediante esta barra de herramientas puede crear y eliminar carpetas y documentos en el directorio virtual. También puede mover elementos de una carpeta a otra. Y finalmente, puede obtener la versión final de un documento.

En el momento de crear un documento debemos asignarle un título. También podemos asignarle colaboradores. Pueden ser con acceso a modo de escritura, o sólo lectura. De esta manera, nos podría interesar que nuestro

documento lo pudiesen leer diferentes personas pero sólo pudiesen modificarlo unas pocas. Por supuesto, este listado de personas que tienen acceso al documento, sea como lectura o como escritura, puede cambiarse. De hecho, hasta el título puede cambiarse.

Los documentos LaTeX se compilan, así pasan de texto plano a un formato enriquecido. Esto es totalmente transparente al usuario. Y únicamente cuando hay algún error en este proceso de compilación es notificado al usuario. Si todo va bien, el usuario obtiene el documento PDF resultante.

Los dos paneles que hay en la página, permiten :

a) subir imágenes (comúnmente denominadas figuras) para poder incluirlas en los documentos. Todo esto gracias a Jakarta FileUpload [17].

b) elegir la parte del documento que queremos editar.

Cuando seleccionamos una parte a editar, el texto relacionado se carga en el editor. Esa parte es bloqueada impidiendo el acceso a escritura a cualquier otro usuario. No obstante, los demás usuarios sí podrán acceder en modo lectura, de manera que podrán saber que se está escribiendo.

La aplicación es totalmente compatible con los navegadores actuales, entre ellos Mozilla Firefox e Internet Explorer.

4. Conclusiones

Una vez acabado este trabajo, me ha permitido darme cuenta de varias cosas:

- Un análisis y diseño previos correctos son realmente importantes si se quiere llevar a cabo un proyecto de cierta magnitud y que llegue a buen puerto.
- El trabajo colaborativo es sumamente importante. Pero tiene una serie de problemas asociados, que son la sincronización en el acceso a los recursos y control de inconsistencias dichos recursos (bases de datos, archivos, etc.).
- La deslocalización es gratificante. En el sentido que el trabajador/colaborador puede acceder desde cualquier lugar, siempre que tenga una conexión a internet.
- Es posible crear aplicaciones web que tengan unos tiempos de respuesta razonablemente buenos como para competir con aplicaciones típicas de escritorio. Gracias al uso de AJAX y otras técnicas similares es como se consigue. De esta manera, los ordenadores desde los cuales se necesita trabajar no necesitan unos requerimientos tan extensos, sino que tan sólo necesitan de la conexión a internet y un navegador actual.
- Las aplicaciones web, como ésta, están en constante desarrollo. Esto se debe a que el acceso a ellas por parte del cliente se hace siempre de la

misma manera y por tanto la actualización y mejora de los servicios se hace de manera transparente a él. Esto lleva a una constante mejora de este tipo de aplicaciones sin molestias importantes para el usuario.

Por supuesto esta aplicación se puede mejorar (como todas). Lo más destacable sería la internacionalización y traducción, además de incluir un sistema de mensajería interno que permita a los usuarios relacionarse sin necesitar herramientas externas. Otras posibles mejoras entrarían a contar en el cambio de ciertos aspectos que conseguirían una mejor escalabilidad y respuesta del programa, como la implementación de otras tecnologías como Comet [18] (una variante mejorada de AJAX) o el uso de múltiples hilos de ejecución que permitan una mejor eficiencia en ciertos fragmentos de código.

5. Agradecimientos

Por supuesto, tengo que dar las gracias a mi familia por todo el apoyo y la comprensión. Y a mis amigos, por ayudarme a comprenderme mejor a mi mismo y no dejar que me rinda.

Referencias

- [1] Lamport, L.: *LaTeX. A document preparation system*. Addison-Wesley Professional. 2nd edition. June, 1994.
- [2] Página principal MediaWiki
<http://www.mediawiki.org>
- [3] Página principal Zoho Writer
<http://www.zohowriter.com/>
- [4] Página principal Writely
<http://www.writely.com/>
- [5] Página principal SynchroEdit
<http://www.synchroedit.com/>
- [6] Documentació dhtmlXtree
<http://scbr.com/docs/products/dhtmlxTree/>
- [7] Documentació FCKEditor
<http://wiki.fckeditor.net/>
- [8] Tutorial AJAX
<http://www.w3schools.com/ajax/>
- [9] Portal de Java. Sun Microsystems.
<http://java.sun.com>
- [10] Página de Apache Tomcat
<http://tomcat.apache.org>
- [11] JSP API
<http://tomcat.apache.org/tomcat-5.5-doc/jspapi/index.html>
- [12] Tutorial Apache Struts
http://www.programacion.com/java/tutorial/joa_struts/
- [13] Página de MySQL
<http://www.mysql.org>
- [14] JDOM
<http://www.jdom.org>
- [15] Tutorial XPATH
<http://www.w3schools.com/xpath>
- [16] Página JavaMail
<http://java.sun.com/products/javamail>
- [17] Página Jakarta FileUpload
<http://jakarta.apache.org/commons/fileupload>
- [18] Descripción Comet (inglés)
[http://en.wikipedia.org/wiki/Comet_\(programming\)](http://en.wikipedia.org/wiki/Comet_(programming))