



Escuela Técnica Superior de Ingenierías Informática y de
Telecomunicación

Universidad de Granada

Departamento de Arquitectura y Tecnología de Computadores

PROYECTO DE FIN DE CARRERA
INGENIERÍA INFORMÁTICA

SWADE: Editor WYSIWYG para SWAD

Alumno: David Medina Godoy
Director: Antonio Cañas Vargas
Fecha: 24 de septiembre de 2013

D. Antonio Cañas Vargas, profesor de la titulación de Ingeniería Informática, perteneciente al Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada, en calidad de director del proyecto fin de carrera de D. David Medina Godoy

INFORMA que el proyecto

SWADE: Editor WYSIWYG para SWAD.

Ha sido realizado y redactado por dicho alumno, y corresponde a la investigación realizada bajo su dirección, y con esta fecha autoriza su presentación.

Granada, Septiembre de 2013

Director del proyecto

Firmado: Antonio Cañas Vargas.

D. David Medina Godoy con DNI 53590723X, autorizo a que una copia de este

Proyecto Fin de Carrera, titulado “SWADE: Editor WYSIWYG para SWAD”, se deposite en la biblioteca de la facultad de la Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación para que pueda ser libremente consultada. Granada, Septiembre de 2013.

Firmado: David Medina Godoy.

A mis padres por esforzarse tanto y permitirme llegar hasta aquí.

A Liliana por aguantarme en mis momentos de estrés.

A mi hermano mayor por servirme siempre de inspiración.

A mi hermano pequeño por motivarme a dar lo mejor de mí.

UNIX es básicamente un sistema operativo simple, pero hay que ser un genio para entender la simplicidad.

Dennis Ritchie

Resumen

En el presente documento se introduce SWADE, el editor WYSIWYG para la plataforma SWAD. SWADE surge principalmente ante la necesidad de SWAD de permitir la introducción de fórmulas y texto enriquecido por parte del usuario. Los principios básicos en el desarrollo de SWADE han sido la usabilidad, la sencillez y la independencia de SWAD, intentando cubrir los objetivos propuestos con la solución más ligera posible.

El proyecto se puede descomponer en tres módulos bien diferenciados, que son la inserción de texto enriquecido, las fórmulas y el saneamiento posterior del código HTML generado. Para la inserción de texto enriquecido hemos desarrollado nuestro propio editor en JavaScript, evitando el uso de cualquier Framework, para así poder obtener una solución lo más ligera posible. Es necesario para la inserción de fórmulas que el usuario tenga unas nociones básicas de TeX. Para la visualización de las fórmulas y la generación posterior de los ficheros png hemos usado MathJax [1] y Texvc [2] respectivamente. Para el saneamiento del código HTML generado hemos usado HTMLpurifier [3].

Finalmente se ha implantado el editor en la plataforma OpenSWAD [4] donde se espera seguir realizando pruebas para su posterior implantación en la plataforma SWAD.

Índice general

1. Introducción	17
2. Análisis	21
2.1. Requisitos funcionales	21
2.2. Visión global de la solución	22
2.3. Integración en la Plataforma	23
3. Estado del arte	27
3.1. Plataformas	27
3.1.1. Wikipedia	27
3.1.2. Stackoverflow	29
3.2. Renderizado de Fórmulas	30
3.2.1. MathJax	30
3.2.2. Texvc	31
3.2.3. tex2png	31
3.3. Editores WYSIWYG	31
3.3.1. YUI Rich Text Editor	32
3.3.2. TinyMCE	32
3.3.3. NicEdit	32
3.3.4. CKeditor	34
3.3.5. openWYSIWYG	35
3.4. Conclusiones del Análisis	35

4. Diseño del editor	39
5. Resultados	45
6. Conclusiones	51
APÉNDICES	53
A. Instalación en el Servidor de OpenSWAD	55
B. Integración en la plataforma SWAD	59
C. Software Libre y Código Abierto	63
D. Licencias de distribución de Software Libre	65
D.1. MIT License	65
D.2. Mozilla Public License v2	66
D.3. BSD License	66
D.4. GNU General Public License	66
D.5. GNU Lesser General Public License	67
D.6. GNU Affero General Public License v3	67
E. Código del editor	69

Lista de Figuras

1.1. Lenguaje Wiki Markup	17
1.2. SWADE en mensajes	18
1.3. T _E X Editor	19
2.1. Secciones de info. asignaturas donde vamos a integrar SWADE	23
2.2. Menú de selección de método de edición.	24
2.3. Creación de una nueva actividad.	25
2.4. Creación de un nuevo test. Enunciado.	25
2.5. Creación de un nuevo test. Respuestas.	26
2.6. Mensajes de SWAD	26
3.1. Editor WYSIWYM de Wikipedia	28
3.2. Editor WYSIWYG de Wikipedia	29
3.3. Editor WYSIWYM de Stackoverflow	30
3.4. YUI Editor	33
3.5. Tinymce Editor	33
3.6. NicEdit Editor	34
3.7. Plugin WIRIS para CKEditor	35
3.8. CKEditor	36
3.9. openWYSIWYG Editor	36
4.1. Dialogo creado con SwadeDialog.	39
4.2. Editor SWADE bloqueado.	40

5.1. Página de pruebas sin SWADE	45
5.2. Página de pruebas con SWADE	46
5.3. Páginas de pruebas de limpieza HTML	47
5.4. Envío en un formulario de símbolos y fórmulas.	48
5.5. Limpieza del HTML en el servidor y renderizado del contenido enviado en el formulario.	48
5.6. SWADE en mensajes	49
5.7. SWADE en preguntas de Test	49
5.8. SWADE en respuestas de Test	50

Prólogo

La plataforma SWAD (Sistema Web de Apoyo a la Docencia) es un sistema de gestión docente, libre desde enero de 2010 que lleva usándose en la Universidad de Granada desde 2002. Con el paso de los años y la mejora de funcionalidades el uso de la plataforma se ha incrementado notablemente y en marzo de 2013 la instalación de SWAD en la Universidad de Granada albergaba 387 titulaciones (incluyendo grado y posgrado) con 6065 asignaturas, 83831 estudiantes y 2859 profesores [5]. Además de en la Universidad de Granada, la plataforma se usa también en la Universidad Nacional de Asunción (Paraguay) y en el portal OpenSWAD.org.

La plataforma integra funciones de apoyo al aprendizaje, docencia y gestión de datos de estudiantes y profesores. Entre las más destacadas se encuentra la gestión del material docente (planificación de actividades, descarga de material docente, subida de ejercicios, etc.), resolución de exámenes tipo test, gestión de estudiantes y profesores, mensajería interna entre usuarios, foros de discusión y consulta de calificaciones individualizada, entre otras. Algunas de sus principales funcionalidades están disponibles además en una aplicación para dispositivos móviles Android y iOS [6].

Introducción

A pesar de la extensa funcionalidad ofrecida por la plataforma SWAD en la actualidad, el usuario aún no puede dar formato al texto que introduce. Para solventar este problema se podría haber recurrido a un lenguaje de etiquetado intermedio, como BBCode, que posteriormente habría que procesar para obtener el HTML correspondiente. Además, esto presenta el inconveniente de que el usuario debería conocer la sintaxis de dicho lenguaje. Entonces, ¿por qué no permitir al usuario dar formato al texto que introduce usando HTML como lenguaje de marcado?

Editando «Transformada de Laplace»

No estás identificado con una cuenta de usuario. Si grabas los cambios, tu dirección IP quedará registrada públicamente en el historial de esta página. Para evitarlo, puedes crear una cuenta de usuario, lo cual tiene varias ventajas.

Es recomendable previsualizar antes de grabar. También puedes visitar la ayuda o el tutorial.

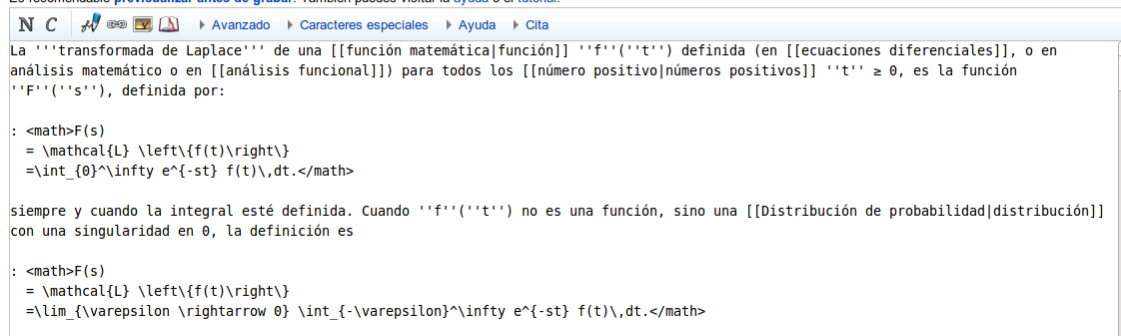


Figura 1.1: Lenguaje Wiki Markup

En los últimos años han aparecido en Internet diversos editores WYSIWYG que permiten al usuario modificar de forma inconsciente y en tiempo real el HTML del texto que introducen. Estos editores se sustentan en tecnologías como lenguajes interpretados en el lado del cliente, que ejecutados en un navegador, permiten modificar de forma dinámica el HTML del cuerpo del

texto. Si esto es así, ¿por qué se siguen usando lenguajes de etiquetado intermedio?

El problema de permitir al usuario introducir directamente código HTML, para ser renderizado posteriormente en la página, presenta un grave riesgo para la seguridad del sistema, es por eso que en un principio se adoptaron lenguajes como BBCode para evitar este tipo de ataques. A este tipo de ataques se les conoce como amenazas XSS, y hacen referencia habitualmente a la inyección de código malicioso en el código de la página.

El trabajo de limpieza del código HTML debe ser exhaustivo, lo que hace de esta una tarea tediosa, por lo que se adoptó el camino fácil del uso de lenguajes intermedios. Es de suma importancia que el análisis del HTML generado en busca de amenazas sea efectivo y se realice de forma correcta, y dada la complejidad de dicha tarea, implementarlo nosotros mismos dentro del marco temporal de este proyecto no es una opción. En lugar de eso existen diversas alternativas en la Web que podemos usar, habiéndonos decantado finalmente por HTMLpurifier [3].

El editor resultante es ligero, elegante, fácil de usar y cumple con los requisitos de edición requeridos por la plataforma SWAD. SWADE no busca la potencia de edición de otras alternativas Web como CKEditor [7] (al nivel de programas de ofimática como Word), para este tipo de tareas se usarán hojas de estilos CSS. En la figura 1.2 podemos ver cómo queda el editor integrado en SWAD en la sección mensajes.

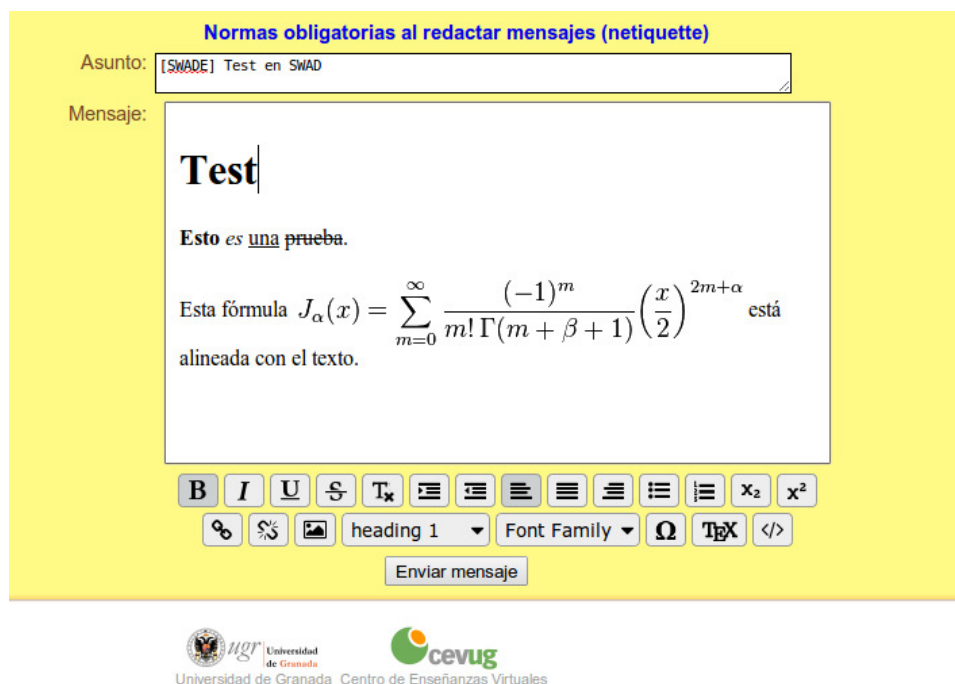


Figura 1.2: SWADE en mensajes

Otra funcionalidad de la que SWAD carece es un editor de fórmulas, de gran importancia cuando se trata de introducir preguntas de tipo test. En este caso nos decantaremos por un editor sencillo basado en LaTeX, que hará indispensable que el usuario tenga unas nociones básicas de dicho lenguaje. En nuestro editor de fórmulas el usuario podrá ver mientras escribe la fórmula que se está formando. Para llevar a cabo esta tarea usamos la herramienta MathJax, que se sustenta en AJAX (Asynchronous Javascript And XML) y hace un renderizado con HTML y CSS de dicha fórmula. Este proceso se realiza en el lado del cliente (navegador web). En la figura 1.3 podemos ver el editor de fórmulas.

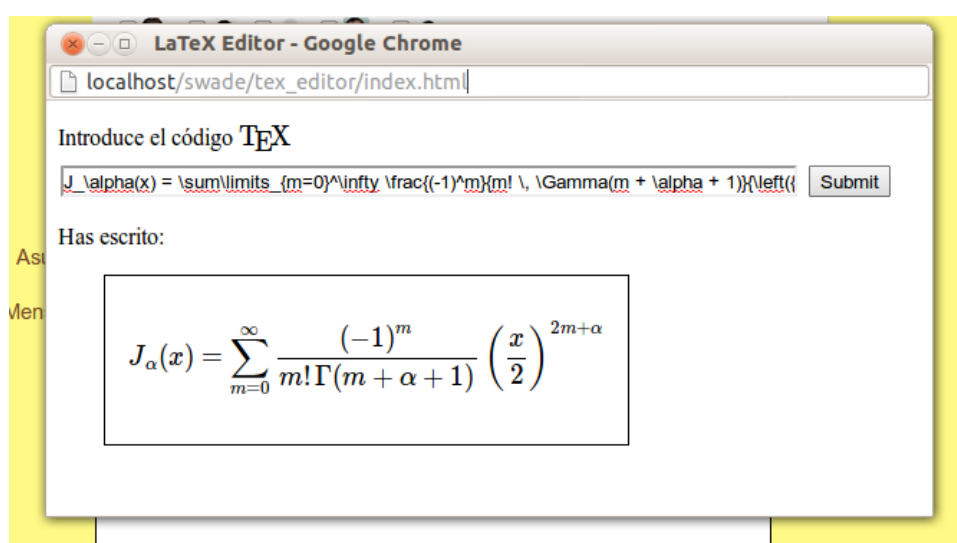


Figura 1.3: TeX Editor

Una vez introducido el código, el usuario confirmará que es la fórmula que quiere enviándose una petición al servidor donde se generará la imagen png correspondiente a la fórmula, usando para esta tarea la herramienta texvc [2], y devolviéndose finalmente el código HTML necesario para insertar dicha imagen. Esta petición al servidor se realiza de forma asíncrona usando AJAX, de forma que cuando se recibe la respuesta, el editor de fórmulas inserta el código de la imagen en el editor SWADE.

Se ha tenido en cuenta la posibilidad de edición de una fórmula, por lo que al hacer click en una se nos abrirá el editor de fórmulas permitiéndonos “editar” la fórmula. Hay que decir que la imagen no se edita, como era de esperar, sino que se elimina la imagen anterior y se posiciona en su lugar una nueva imagen generada.

Capítulo 2

Análisis

El proyecto surge a raíz de la necesidad de SWAD de un editor integrado, que permita el formato de los textos. SWAD es una plataforma muy completa en lo que a gestión docente se refiere, pero si hay un punto débil en esta es la incapacidad de un usuario de dar formato al contenido que genera o lo que es incluso más importante, insertar fórmulas matemáticas.

2.1. Requisitos funcionales

Los requisitos de nuestro editor son los siguientes:

- **Simplicidad** en su uso. El usuario deberá ser capaz de usar el editor sin una fase de entrenamiento previa, cometiendo como máximo un error por cada veinte minutos de uso.

Dado que SWAD es una plataforma usada por muchos profesores y alumnos de muy diversas titulaciones el editor debe ser lo mas simple e intuitivo posible, evitando en la representación del contenido notación HTML, bbcode o similares. Estos editores son los que se conoce como editores WYSIWYG (What You See Is What You Get).

- Fácil **integración** con SWAD. La integración del editor en la plataforma no requerirá más de tres líneas en el código de la plataforma. Nuestro editor debe ser lo mas independiente de SWAD posible, integrándose él mismo en el código de la página HTML.
- Soporte para **fórmulas** matemáticas. Nuestro editor permitirá insertar fórmulas matemáticas. Además, se contemplará la posibilidad de visualización de una página con múltiples

fórmulas matemáticas sin que se produzca un deterioro excesivo del rendimiento (Tiempo de carga medio de 10 fórmulas inferior a 1 segundo).

- Inserción de **símbolos**. Nuestro editor permitirá la inserción de símbolos tales como letras griegas y operadores matemáticos y algebraicos.
- Posibilidad de ver y editar el **código HTML** generado, así como generar un código HTML de calidad, controlando etiquetas no cerradas y bloqueando etiquetas no permitidas para que el contenido global de la página no se vea afectado y evitar ataques XSS respectivamente.
- **Formato simple de texto** y su estructura. Principalmente negrita, cursiva, subrayado, tachado y viñetas, subíndices, superíndices así como formato de cabeceras y cuerpo de texto y tipos de letra normal y mono-espaciada.
- El editor permitirá añadir **imágenes** y **enlaces**.
- **Disponibilidad** del servicio. Por eso no debemos usar ninguna herramienta online, ya que no tenemos ninguna garantía de la continuidad de estos servicios. Por el mismo motivo usaremos tecnologías poco susceptibles de cambio, para reducir de esta manera las posibles labores de mantenimiento.
- **Seguridad**. Permitir al usuario modificar directamente el HTML de la página conlleva un grave riesgo para la seguridad del sistema. Por ello, debemos filtrar el HTML generado con el fin de evitar las amenazas de inyección de código malicioso (XSS) en la plataforma.

2.2. Visión global de la solución

De lo anteriormente expuesto, podemos extraer que en el lado del cliente (Navegador Web) se dará formato al texto que introducimos, mientras que en el lado del servidor nos ocuparemos de generar las imágenes de las fórmulas y limpiar el código HTML generado.

Generando las imágenes de las fórmulas en el servidor nos aseguramos de que se cumple el requisito de tiempo impuesto a la carga de imágenes. Aunque puede producirse un pequeño deterioro en el rendimiento al cargar las imágenes, esto ocurrirá solo la primera vez ya que la caché del navegador hará que la carga las veces posteriores sea imperceptible.



Figura 2.1: Secciones de info. asignaturas donde vamos a integrar SWADE

2.3. Integración en la Plataforma

En esta sección analizaremos las distintas partes de la plataforma SWAD en las que es necesario el uso del editor SWADE. Es importante un análisis de los lugares de la plataforma en los que se insertará el editor para que el diseño de la solución se adapte de la forma mas simple y elegante posible a cada caso.

Usaremos el editor SWADE para introducir la información de las distintas asignaturas. Dados de alta como profesores, en la pestaña Asignatura, debemos usar el editor en las secciones Introducción, Guía docente, Programa teoría, Programa prácticas, Bibliografía, FAQ, Enlaces y en la pestaña Evaluación a la sección Sistema de Evaluación. En la figura 2.1 podemos ver las distintas secciones donde vamos a integrar el editor.

En todas estas opciones, en el menú de edición, se nos preguntará el método por el que vamos a editar dicha sección, donde podremos seleccionar nuestro editor, tal como podemos ver en la figura 2.2.

También será necesario integrar SWADE en el campo descripción de Actividades y Encuestas de las pestañas Evaluación y Estadísticas respectivamente. En la figura 2.3 podemos ver el formulario de creación de una nueva actividad. El formulario de creación de una nueva encuesta es muy similar.

Debemos contemplar también la posibilidad de creación de múltiples editores en un mismo documento HTML ya que, en la sección de Test de la pestaña Evaluación, debemos insertar un total de veintidós editores cada vez que creamos una nueva pregunta. Serían dos por cada posible respuesta, haciendo un total de veinte, más dos por el enunciado y la realimentación de este. En

Editor interno
(editor integrado, sólo para los temarios de teoría y prácticas)

Texto plano
(editor de texto plano)

Página web alojada aquí
(envíe un archivo HTML, o bien un archivo ZIP que contenga un archivo index.html/index.htm y archivos con imágenes)

Archivo: No se ha seleccionado ningún archivo

Página web alojada en otro servidor
(escriba la dirección web de una página que se encuentre alojada en otro servidor)

URL:

Fuente de información

- ☐ No ofrecer ninguna información
- ☐ Editor interno
- ☐ Texto plano
- ☐ Página web alojada aquí
- ☒ Página web alojada en otro servidor

Figura 2.2: Menú de selección de método de edición.

las figuras 2.4 e 2.5 podemos ver el formulario de creación de una nueva pregunta de test.

Finalmente debemos instanciar SWADE en los mensajes de SWAD, ya sea Mensajes entre usuarios o en los Foros. En principio no será necesario integrarlo en los Avisos. Tampoco es necesario en principio integrarlo en Convocatoria de examen o los Campos de las fichas. En la figura 2.6 podemos ver donde tendremos que insertar el editor para los mensajes.



Evaluación > Actividades
Actividades obligatorias o recomendadas en esta asignatura

Nueva actividad

Título:

Fecha inicial: 21 ▼ septiembre ▼ 2013 ▼ 23 h ▼ 03 ' ▼

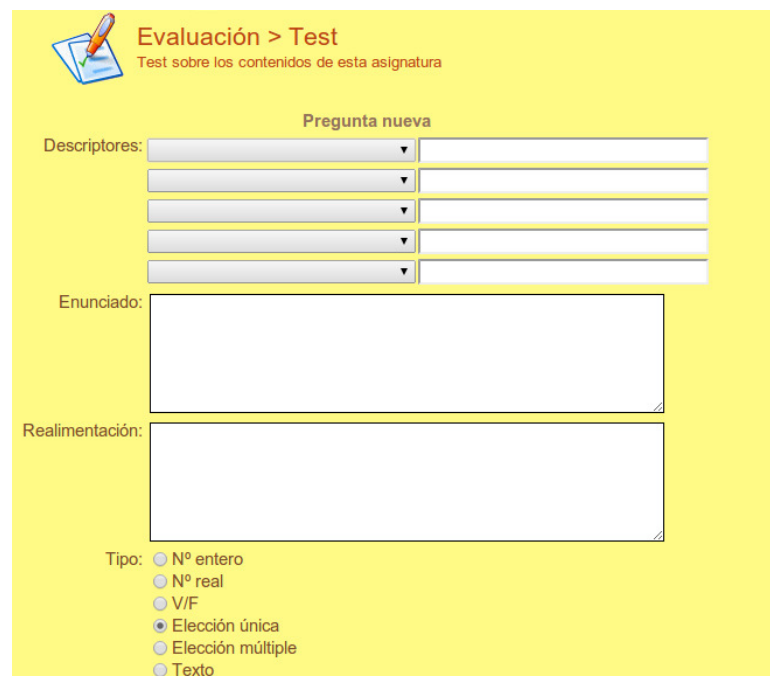
Fecha final: 21 ▼ septiembre ▼ 2013 ▼ 23 h ▼ 59 ' ▼

¿Enviar archivos?: ☐ Carpeta:

Descripción:

Grupos: ☒ Toda la asignatura Plataforma SWAD

Figura 2.3: Creación de una nueva actividad.



Evaluación > Test
Test sobre los contenidos de esta asignatura

Pregunta nueva

Descriptores:

Enunciado:

Realimentación:

Tipo: ☐ N° entero
☐ N° real
☐ V/F
☒ Elección única
☐ Elección múltiple
☐ Texto

Figura 2.4: Creación de un nuevo test. Enunciado.

	Respuesta	Realimentación
<input type="radio"/> <input type="checkbox"/> a)		
<input type="radio"/> <input type="checkbox"/> b)		
<input type="radio"/> <input type="checkbox"/> c)		
<input type="radio"/> <input type="checkbox"/> d)		
<input type="radio"/> <input type="checkbox"/> e)		
<input type="radio"/> <input type="checkbox"/> f)		

Figura 2.5: Creación de un nuevo test. Respuestas.

Normas obligatorias al redactar mensajes (netiquette)

Asunto:

Mensaje:

Figura 2.6: Mensajes de SWAD

Estado del arte

En todo proceso de investigación y desarrollo de software, es importante dedicar un tiempo a examinar en que punto se encuentra la investigación en dicho campo. Por tanto, antes de empezar a diseñar nuestra solución nos dedicaremos al análisis de diversas plataformas y editores que resuelven problemas similares al que nos ocupa.

Dedicaremos este capítulo al análisis de diversas plataformas online y editores WYSIWYG. Nos centraremos principalmente en Wikipedia, Stackoverflow y algunos editores como YUI Rich Text Editor [8], NicEdit [9], CKEditor [7], TinyMCE [10] y openWYSIWYG [11].

3.1. Plataformas

3.1.1. Wikipedia

Wikipedia usa un editor WYSIWYM (What You See IS What You Mean) bastante simple, usando solo las características que la plataforma necesita. Este tipo de editores se caracteriza por una total separación entre la presentación y el contenido del texto. Para el etiquetado usa un lenguaje propio llamado Wiki markup. En la figura 3.1 podemos ver el aspecto del editor de Wikipedia.

Como inconvenientes de este tipo de editores podemos destacar la complejidad en su uso. Tal y como dijimos anteriormente la plataforma SWAD es usada por alumnos y profesores de gran variedad de titulaciones, por lo que uno de los requisitos es la facilidad de uso, es por eso por lo que nos decantamos por un editor del tipo WYSIWYG (What You See Is What You Get), donde el usuario “obtiene lo que ve”, manipulando, de forma directa e inconsciente, el código HTML del

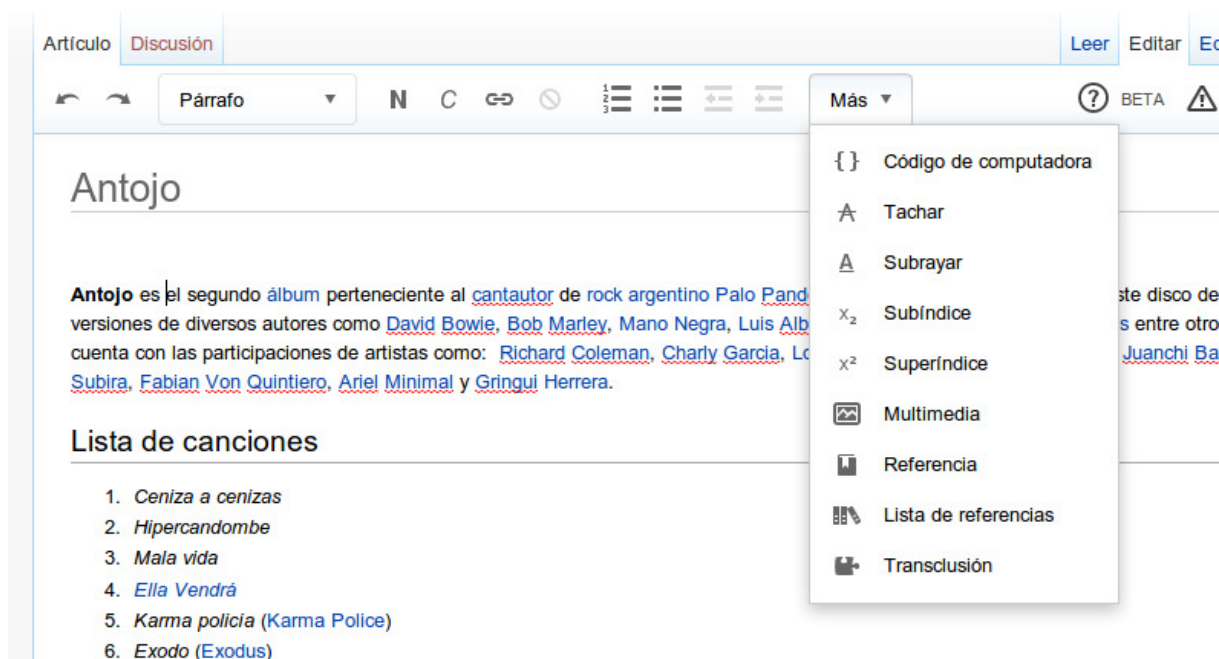


Figura 3.2: Editor WYSIWYG de Wikipedia

MathJax para interpretar las expresiones matemáticas toma más tiempo que Texvc.

Otra de las ventajas del uso de imágenes png en lugar de renderizar fórmulas con HTML y CSS es que las imágenes podrán ser almacenadas en cache, permitiendo que futuros accesos a la página sean mas rápidos.

Por otra parte, el uso de Texvc presenta el problema del tamaño de las fórmulas.

3.1.2. Stackoverflow

La web Stackoverflow usa un editor WYSIWYM, al estilo del que usa Wikipedia, que combina un lenguaje de etiquetado intermedio con un subconjunto del lenguaje de etiquetado HTML. Este editor posee un campo que muestra, al mismo tiempo que editamos, el resultado final del contenido que generamos. Además, tal cómo se muestra en la figura 3.3, posee un menú de ayuda desplegable que permite consultarlo de forma muy cómoda mientras se escribe. Estos dos aspectos hacen que este editor sea mucho mas sencillo de usar que el editor WYSIWYM que posee Wikipedia, aunque, todo sea dicho, el editor de Wikipedia contempla funcionalidad, como el soporte de fórmulas LaTeX entre otros, que este no posee.

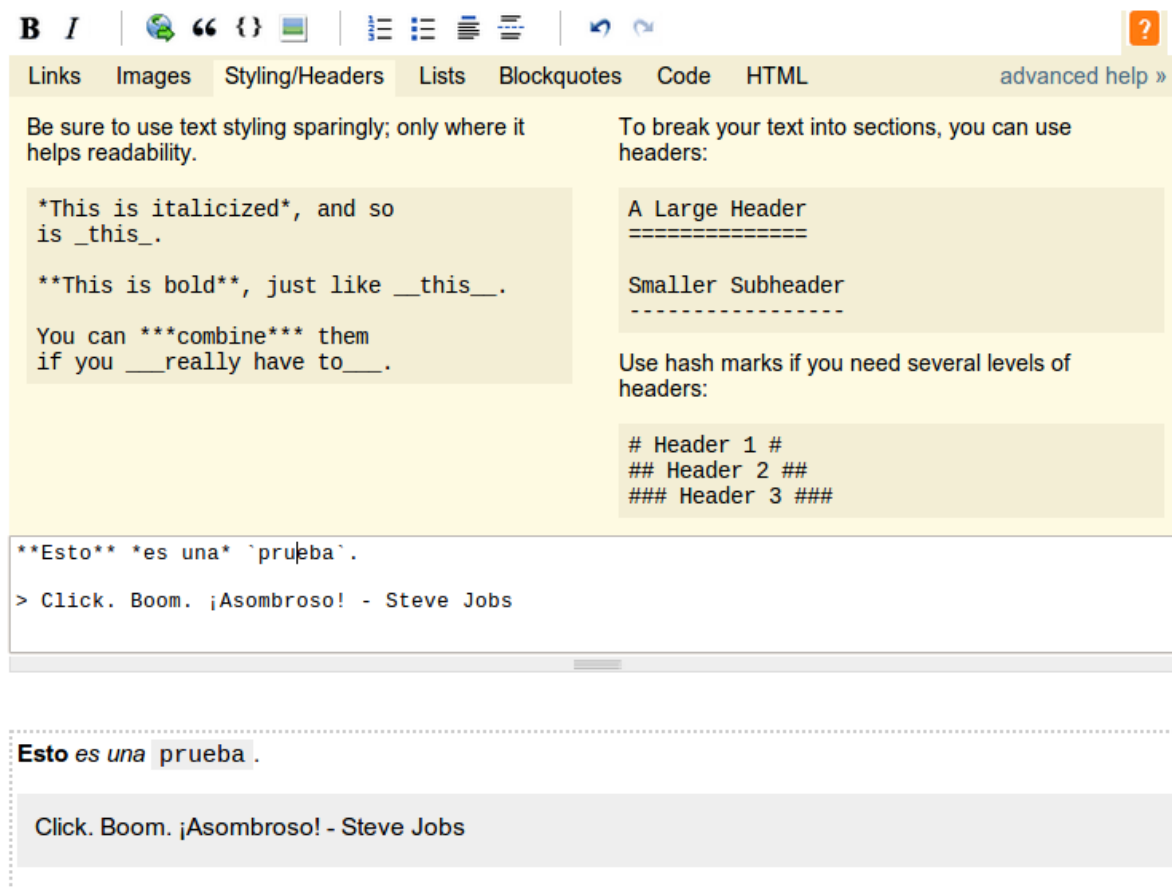


Figura 3.3: Editor WYSIWYM de Stackoverflow

3.2. Renderizado de Fórmulas

3.2.1. MathJax

MathJax, tal y como se define en su página, es un motor de visualización JavaScript para fórmulas matemáticas y de código abierto que funciona en todos los navegadores. Dicha aplicación, como ya hemos comentado previamente, nos permite renderizar las fórmulas \LaTeX mediante HTML y CSS.

MathJax analiza el contenido de la página buscando una fórmula en \LaTeX y cuando la encuentra la sustituye por un conjunto de etiquetas que compondrán la fórmula renderizada.

Esta herramienta permite dos modos de uso. La primera es enlazando directamente la aplicación a través de la Red de Distribución de Contenido (CDN) de MathJax. De esta forma MathJax siempre estará actualizado y además te ahorras su instalación en el servidor.

Sin embargo, tal cómo dijimos antes, busquemos herramientas instalables en el servidor, ya que no podemos estar sujetos a la disponibilidad de servicios de terceros. Por este motivo, si nos decantamos a usar MathJax, optaremos por la descarga e instalación de MathJax en el servidor.

3.2.2. Texvc

Texvc (Tex validator and converter) es la herramienta usada por Wikipedia para validar y generar imágenes png a partir del código en formato ams-latex.

Dicha herramienta tiene una serie de dependencias que tendremos que satisfacer. Estas dependencias son ocaml, el lenguaje en la que está implementada y latex, dvips, gs y convert, usadas en el proceso de renderizado de la imagen de la fórmula.

A pesar de la escasa documentación disponible de esta herramienta, su extendido uso ha dado lugar a un completo FAQ con problemas de uso y soluciones para dichos problemas. El motivo por el que dicha herramienta se usa tanto es porque pertenece a la extensión math de mediawiki.

3.2.3. tex2png

Tex2png es una herramienta simple que cumple perfectamente con nuestro propósito. El autor la describe en su página [21] como una herramienta para pasar fácilmente del formato $\text{T}_{\text{E}}\text{X}$ y $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ a PNG. Es un simple envoltorio de *latex* y *dvipng* que fue originalmente escrito como *backend* de la web donde se aloja para insertar fácilmente fórmulas matemáticas.

Tex2png es un script bash, por lo que su instalación consistirá únicamente en copiarlo a la carpeta desde donde lo vamos a llamar (y en la instalación de *latex* y *dvipng*).

La documentación de esta herramienta es bastante escasa y se han encontrado algunos problemas de renderizado con algunas fórmulas.

3.3. Editores WYSIWYG

Hay muchas personas muy inteligentes que ya han dedicado mucho tiempo a trabajar en este tema, pero esto no quiere decir que todo esté hecho. Debemos plantearnos cómo podemos mejorar estas soluciones. Obviamente, el concepto “mejorar” es muy difuso, puede significar que algo funcione más rápido, de una forma más novedosa o que sea más barato que el sistema original. En nuestro caso es tremendamente importante que se integre a la perfección con la plataforma SWAD y que cumpla una serie de requisitos muy concretos. En esta sección analizaremos las

características de los editores WYSIWYG más usados en la Web, así como los puntos a favor y en contra de cada uno.

También tendremos en mente las licencias bajo las que se distribuyen, pues en caso de usar parte de algún editor en nuestra solución, su licencia debe ser compatible con la licencia Affero GPL de SWAD. En la sección Apéndices podemos consultar las características generales de las distintas licencias. En caso de querer ampliar dicha información, podemos acudir a la Bibliografía.

3.3.1. YUI Rich Text Editor

Basado en la librería YUI de Yahoo y liberado bajo licencia [13], este editor sustituye el textarea estándar de HTML por un editor bastante completo. Permite dar un formato de texto enriquecido, incluyendo elementos para modificar la estructura del texto como listas, elementos para dar formato e inclusión de imágenes drag-and-drop y redimensionado de estas. Permite soporte para plugins y tiene dos modos, normal y simple, el cual tiene con un conjunto de características más reducido.

Podemos decir que se trata de un editor de alta complejidad, que se apoya en un completo Framework como es la librería YUI de Yahoo. La Web de este editor es <http://yuilibrary.com/yui/docs/editor/> y allí podremos encontrar gran cantidad de información y una amplia documentación tanto de este como del Framework en el que se apoya.

3.3.2. TinyMCE

Este popular editor WYSIWYG transforma nuestro tradicional textarea por un conjunto de elementos HTML para construir el menú de formato, esta herramienta es muy usada por diversos CMS (Content Management System) Open Source, es muy fácil de integrar, permite la personalización de botones en la barra de herramientas y tiene opciones para configurar el menú de formato limitando el número de botones que aparecen. Como ventaja comentar que se distribuye bajo la licencia de código abierto LGPL [14] y que es fácil de integrar y personalizar. Para probar este editor o ampliar esta información se recomienda consultar la web <http://www.tinymce.com/>.

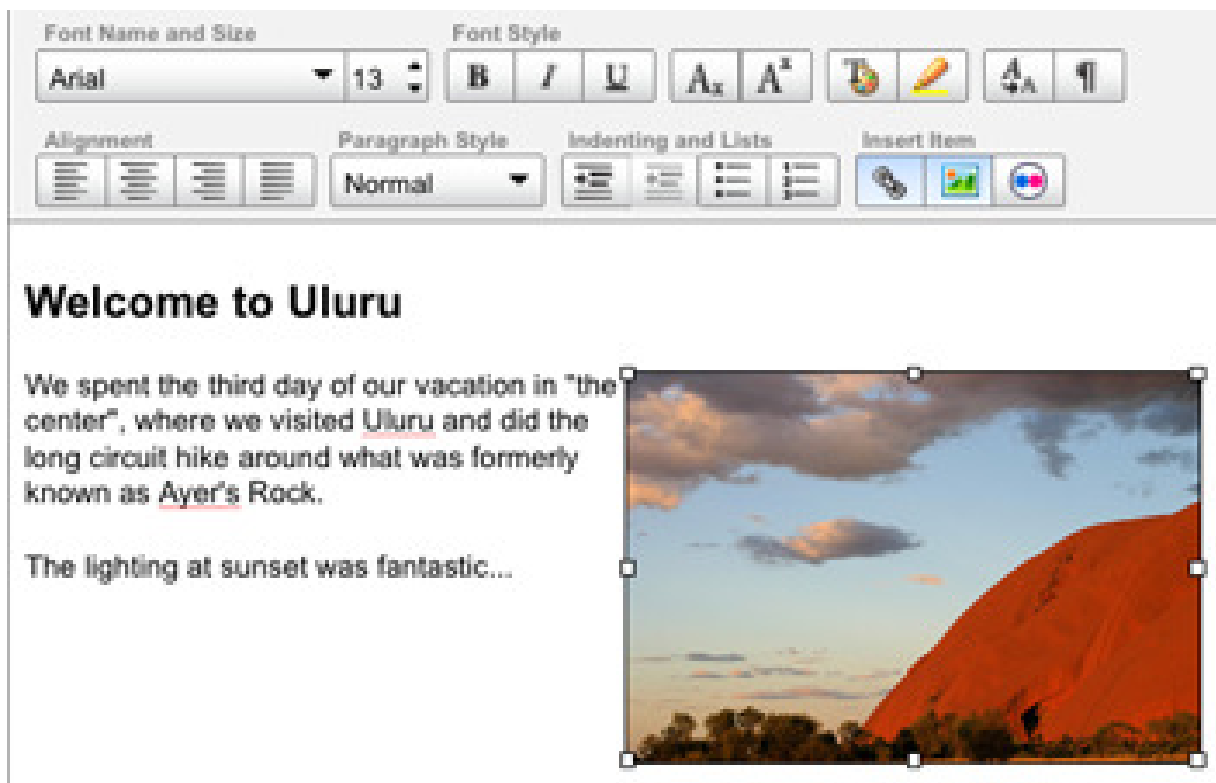


Figura 3.4: YUI Editor

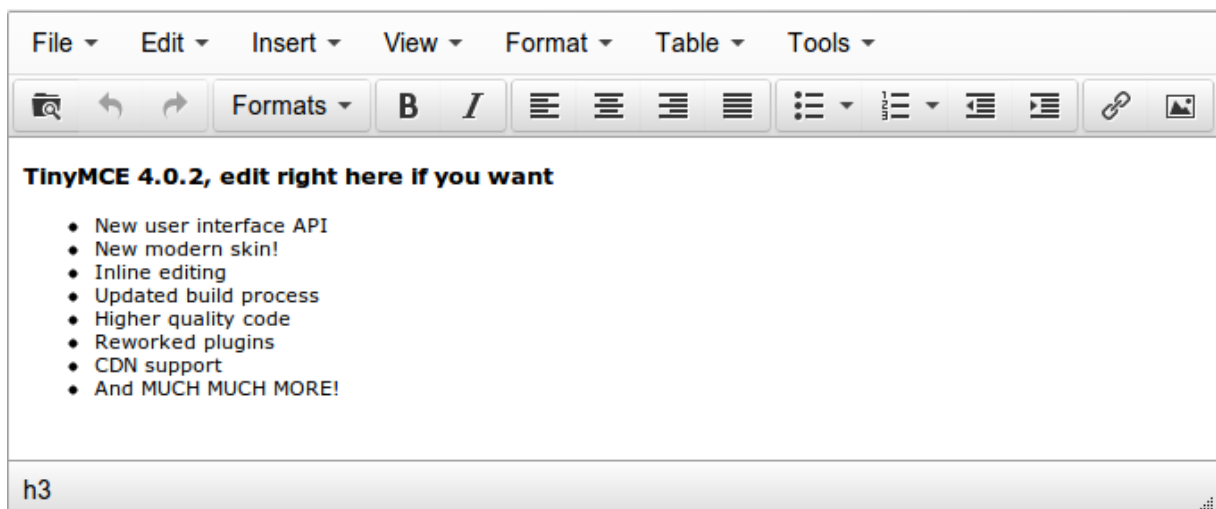


Figura 3.5: Tnymce Editor

3.3.3. NicEdit

Este ligero editor WYSIWYG, distribuido bajo Licencia MIT [15], se caracteriza por su flexibilidad, sencillez y capacidad de personalización. Permite transformar cualquier elemento

div o textarea en un editor de texto enriquecido. Se trata de un editor simple pero efectivo. Al contrario que otros editores, no viene con distintos skins o plugins para ampliar su funcionalidad, ni tiene soporte nativo para fórmulas.

Permite limitar la funcionalidad mediante la selección de los elementos que aparecerán en la barra de herramientas y, a excepción del editor de fórmulas, implementa toda la funcionalidad que requerirá nuestro editor. De fácil instalación y uso es una opción muy a tener en cuenta. La integración en la página es muy simple y puede llevarse a cabo con un par de líneas de código. Para más información visitar su web: <http://nicedit.com/>.

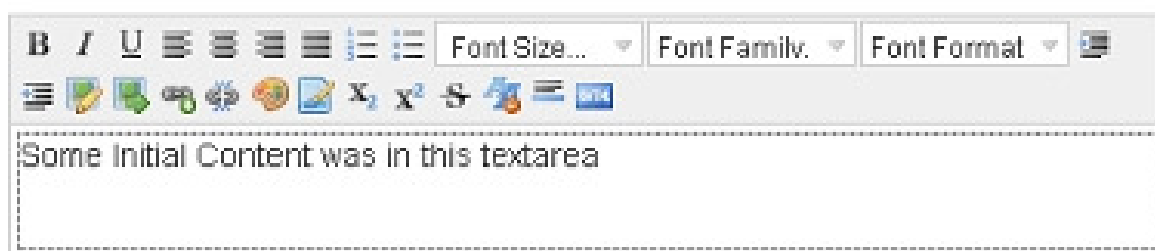


Figura 3.6: NicEdit Editor

3.3.4. CKeditor

Este editor WYSIWYG es muy potente y estable a la vez que flexible. Ampliamente usado, cuenta con una trayectoria de 10 años en el mercado. Además genera código XHTML de calidad, permitiendo su edición y parsea el contenido para que este no afecte a la página que contiene al editor. Permite el uso de plugins desarrollados por terceros y con estos amplía notablemente la funcionalidad del editor. La apariencia está muy cuidada y hay disponibles varios skins que podemos usar. Distribuido bajo las licencias de código abierto GPL [16], LGPL [14] y MPL [17].

Presenta la ventaja de tener soporte para fórmulas mediante plugins. Los plugins con esta finalidad disponibles son dos: Math Editor y WIRIS math & science editor. El primero funcionaba correctamente en pruebas online, pero al probarlo en nuestro servidor no hemos conseguido que funcione en ningún navegador, muy probablemente por incompatibilidades con la versión de CKEditor. El segundo plugin cumple con su cometido, aunque desde mi punto de vista es demasiado complejo. Tras seleccionar la opción de fórmulas, tal y como se muestra en la Figura 3.7, se abre una nueva ventana que contiene un completo editor donde podemos escribir nuestra fórmula. Una vez confirmada la operación se inserta en nuestro editor la imagen correspondiente

a la fórmula escrita. Pese a estos pequeños fallos en los plugins, que recordemos han sido desarrollados por terceros, el editor en sí es muy completo y permite entre otras cosas la edición de texto en línea al hacer click sobre este.

Para más información podemos consultar su web: <http://ckeditor.com/>

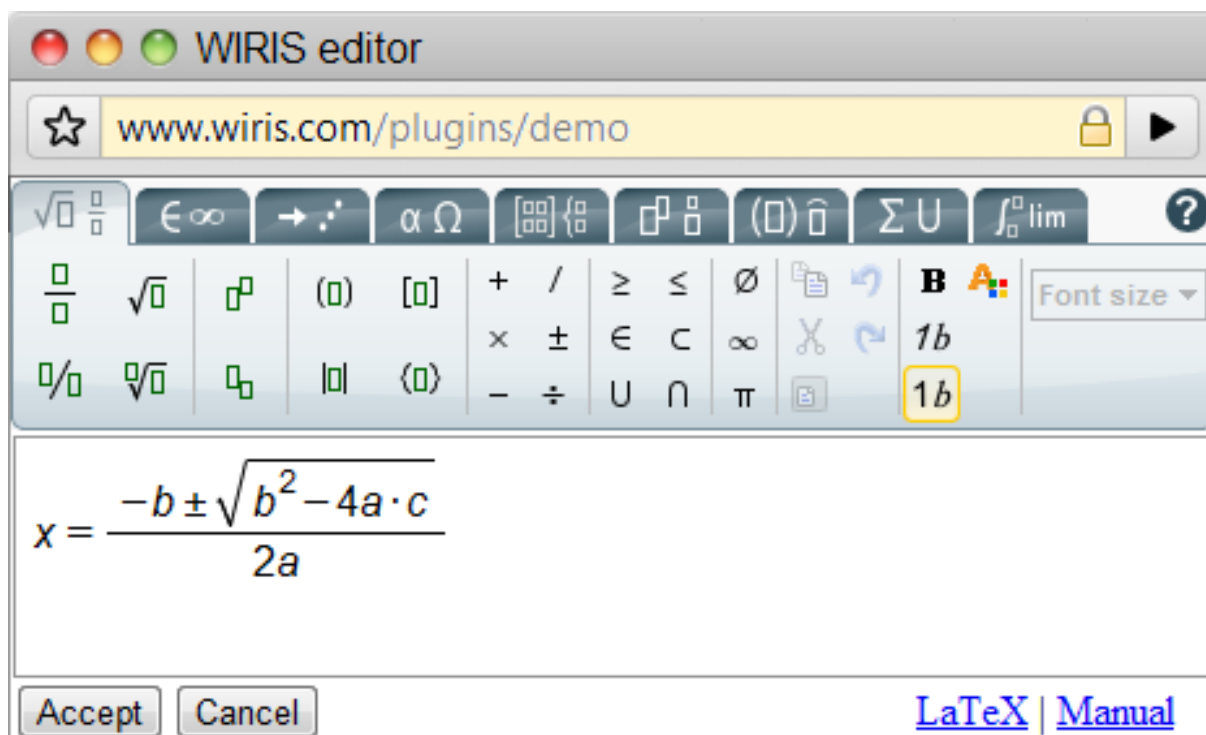


Figura 3.7: Plugin WIRIS para CKEditor

3.3.5. openWYSIWYG

<http://www.openwebware.com> Este sencillo editor, distribuido bajo licencia de código abierto LGPL [14], se presenta como el editor ideal para mejorar tu CMS. Como los anteriores reemplaza tarea por un editor de texto. Parece ser que en la actualidad se encuentra obsoleto y presenta incompatibilidades con los navegadores actuales. Su web es <http://www.openwebware.com>.

3.4. Conclusiones del Análisis

En este capítulo hemos analizado y reflexionado acerca de diversas plataformas y herramientas de la Web. A continuación veremos cuales de estas se ajustan más a nuestros objetivos, teniendo siempre en mente los requisitos de nuestro editor y buscando una solución sencilla y de calidad.

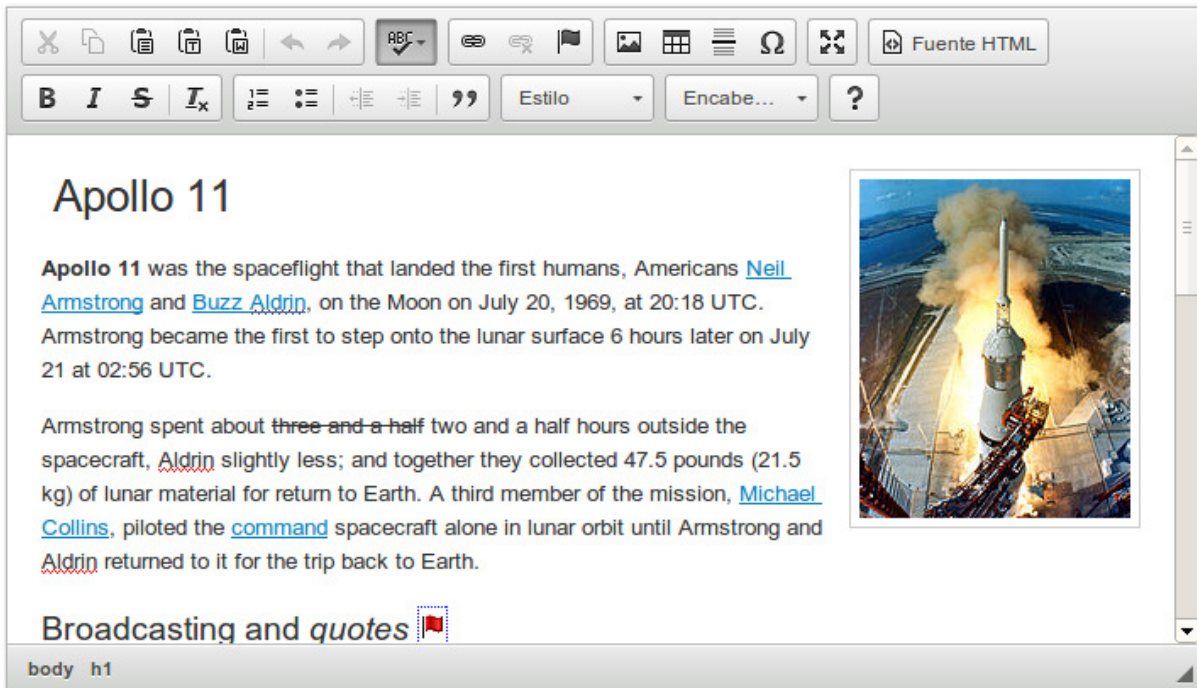


Figura 3.8: CKEditor

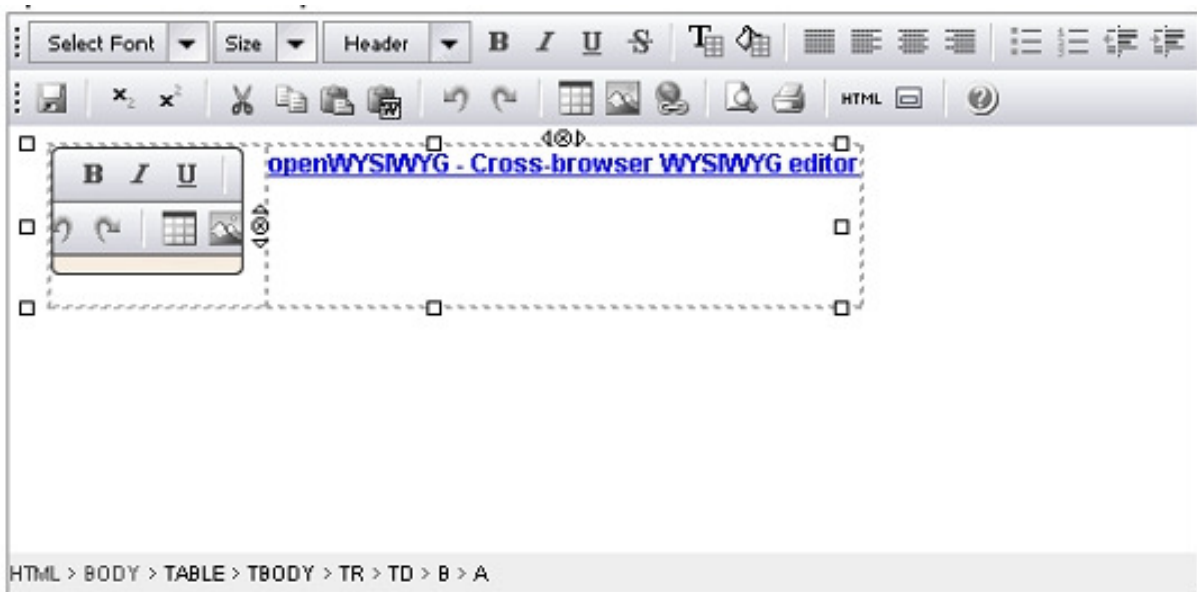


Figura 3.9: openWYSIWYG Editor

En lo referente a los editores descartamos el uso de *openWYSIWYG* debido a los problemas de compatibilidad con los distintos navegadores que ha presentado, ya que parece ser que dicho editor se encuentra obsoleto.

Por otra parte hay algunos editores que soportan de forma nativa (o con el uso de plugins) el uso de fórmulas. Este es el caso de *CKEditor* y *YUI Rich Text Editor*.

CKEditor es un proyecto maduro con 10 años en el mercado. Esta madurez se nota en todos los aspectos. El acabado de CKEditor es impecable, no tanto lo es el de sus plugins, desarrollados por terceros. A pesar de esto CKEditor es una opción muy a tener en cuenta. El problema que presenta, sin duda, es que esta complejidad de la que hace gala puede convertirse en un problema a la hora de estudiar y modificar su código fuente para adaptarlo a nuestras necesidades. Comentar también la posibilidad de usar cualquiera de las licencias [16], LGPL [14] y MPL [17], lo que nos aporta gran libertad.

TinyMCE supone una alternativa de calidad a *CKEditor*. A diferencia de este último *TinyMCE* se distribuye solo bajo licencia LGPL, lo que algunos casos podría ser una restricción. Sin embargo, en este caso concreto, no es importante ya que SWAD se encuentra liberado bajo una licencia AGPLv3 y no presentaría incompatibilidades con la licencia LGPL del editor.

YUI Rich Text Editor presenta una potente alternativa, con el respaldo de una gran empresa como es YAHOO. Como ya comentamos antes, se basa en el framework YUI (Yahoo User Interface), una serie de bibliotecas escritas en JavaScript para la construcción de aplicaciones de Internet enriquecidas (RIA) y liberadas bajo licencia BSD por parte de la compañía Yahoo. Sin embargo, para nuestro editor, evitaremos el uso de frameworks y nos basaremos en algún editor “standalone”.

NicEdit es una solución simple, aunque carece de soporte nativo para fórmulas, funcionalidad que tendríamos que ampliar con alguna otra herramienta, como puede ser MathJax o Texvc. A excepción de esto último, NicEdit implementa toda la funcionalidad necesaria por nuestro editor.

Al parecer, las mejores opciones para nuestro editor son NicEdit y CKEditor. CKEditor es mas pesado que NicEdit aunque mucho mas completo y con muchos plugins desarrollados por terceros. Dada la simplicidad del problema que nos ocupa, la ligereza de NicEdit es un gran punto a su favor, haciéndose innecesarios la gran mayoría de los plugins que tiene CKEditor. Por otra parte, el plugin WIRIS de fórmulas para CKEditor también está disponible para NicEdit.

El problema de modificar un editor ya existente es que requiere una compleja labor de estudio de su código fuente para poder eliminar aquella funcionalidad que no necesitamos y añadir la funcionalidad de la que carece, así como adaptarlo a nuestros requisitos.

Tras un completo análisis del código fuente de los editores se ha decidido implementar nosotros mismos el editor WYSIWYG, evitando el uso de cualquier Framework y desarrollando la solución

lo más simple posible, cumpliendo cada uno de nuestros requisitos. Aunque no tomemos código fuente de ninguno de los editores anteriormente comentados, nos inspiraremos en NicEdit, el ligero editor que más se ajustaba a nuestras necesidades.

Diseño del editor

En este capítulo nos centraremos en los detalles concretos del diseño del editor. El editor resultante se compone de tres módulos bien diferenciados que son el núcleo del editor, el editor de fórmulas y el purificador de HTML.

El núcleo del editor es un fichero Javascript que será el responsable de llevar a cabo la mayor parte de las funciones del editor. Este fichero encapsula la funcionalidad de la aplicación en la clase SwadeManager, SwadeDialog y SwadePanel. La primera de ellas es una clase donde todos los métodos son estáticos y usamos para las distintas operaciones que componen la funcionalidad del editor.

SwadeDialog es una clase que nos permite crear Dialog propios como el de la figura 4.1 y ofrece principalmente métodos para mostrarlos, ocultarlos y destruirlos.

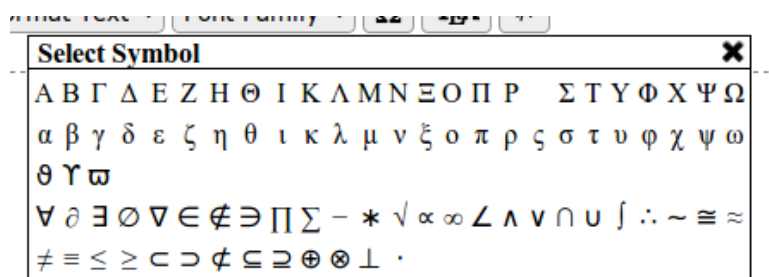


Figura 4.1: Dialogo creado con SwadeDialog.

SwadePanel es la clase que encapsula los métodos propios de cada editor. Estos métodos son principalmente para bloquear un editor cuando no se ha seleccionado y para el refresco del formato del texto seleccionado 4.2.



Figura 4.2: Editor SWADE bloqueado.

Para usar SWADE debemos incluir el fichero javascript con el código siguiente.

```
<script src="path-to-swade/swade.js" type="text/javascript"></script>
```

Este script convierte uno o varios textarea o div en un editor SWADE. Para instanciar un editor en un textarea o div tenemos tres métodos. Por id, por clase o realizando una consulta. En el fragmento de código siguiente vemos un ejemplo de cada uno de ellos.

```
<script>
  SwadeManager.setOnDOMLoaded(function(){
    SwadeManager.setSwadeByQuery("textarea.swade"); //1
    SwadeManager.setSwadeById("swade-id");           //2
    SwadeManager.setSwadeByClassName("swade-class");//3
  }
);
</script>
```

La función `SwadeManager.setOnDOMLoaded` aplica la función pasada por parámetro cuando se ha construido completamente lo que se conoce por Document Object Model, aunque las

imágenes aún estén en carga. A esta función le pasaremos los métodos para instanciar el editor en las distintas partes de la página. Con el primero de ellos instanciamos un editor en aquellos textarea con la clase “swade”, con el segundo lo haremos para los elementos div o textarea con id “swade-id” y con el último instanciamos el editor para los elementos div o textarea con clase “swade-class”.

Todas estas funciones se apoyan en las distintas funciones de selección de Javascript y la función setSwade, que instancia un editor para cada elemento de un vector de elementos. En esta función se tiene en cuenta si el objetivo es un div o un textarea, siendo este último el caso más complejo. Si el objetivo es un div se añade a este un panel al que añadimos los distintos eventos de pulsación de los distintos botones. Por el contrario, si el objetivo es un textarea, debemos crear nosotros mismo el elemento div, asignarle como contenido el valor del textarea, ocultar dicho textarea y comprobar si formaba parte de un formulario. Si el textarea formaba parte de un formulario, debemos interceptar la petición del formulario para establecer el valor del textarea al valor del div que estamos editando.

El editor de fórmulas es una página HTML que usa la herramienta Mathjax para renderizar la fórmula en tiempo real usando HTML y CSS. Este proceso se lleva a cabo mientras que el usuario la escribe dicha fórmula.

Cuando el usuario finaliza la inserción de la fórmula, enviamos una petición asíncrona al servidor al script tex2png.php que nos devolverá el elemento HTML correspondiente a la imagen generada. El fragmento de código siguiente es el responsable de hacer la petición al servidor de forma asíncrona, de insertar la imagen devuelta por este en SWADE y de cerrar la ventana del editor de fórmulas.

```
function showFormula(str)
{
    var xmlhttp;
    if (window.XMLHttpRequest)
    {
        // code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp=new XMLHttpRequest();
    }
    else
```

```

{
  // code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
xmlhttp.onreadystatechange=function()
{
  if (xmlhttp.readyState==4 && xmlhttp.status==200)
  {
    opener.tex = xmlhttp.responseText;
    window.returnValue = xmlhttp.responseText;
    top.close();
    return false;
  }
}
xmlhttp.open("POST","tex2png.php",true);
xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");
xmlhttp.send("MathInput="+encodeURIComponent(str));
}

```

Para el editor de fórmulas, en la parte del servidor, encontramos el script `tex2png.php` (que no tiene nada que ver con la herramienta `tex2png` descrita en el capítulo *Estado del arte*).

Este script se ocupa de extraer el valor de la fórmula de la variable POST, de generar la imagen correspondiente con `texvc` y de devolver el elemento HTML necesario para insertar dicha fórmula en nuestro editor SWADE. El fragmento de código siguiente recoge dicha funcionalidad.

```

...
$latex_formula = "'".$_POST["MathInput"]."'";
$cadena = "./texvc $path_tmp $path_dest $latex_formula";
exec($cadena,$output,$return);
...
echo "<img src=\"${cwd}/${path_dest}${name}.png\" alt=\"Formula inválida\"
data-tex-swade = $latex_formula

```

```
onclick=\"swade_app.tex_code_selected = $latex_formula;\n
swade_app.selected_tex_object = this;\n
style=\"vertical-align:middle;\n/>\";
```

Podemos observar como se han añadido algunos atributos al elemento HTML para insertar la imagen, aparte de la ruta de esta. El atributo `data-tex-swade` y `onclick` son para poder editar la fórmula haciendo click sobre ella. El atributo `style` es para alinear la fórmula con el texto que la rodea.

El último modulo de nuestro editor es la herramienta `HTMLPurifier` [3], la cual se ocupará del filtrado del HTML generado por el usuario. Dicha herramienta está escrita en php y nos permitirá evitar ataques XSS en la página de la plataforma. En el siguiente fragmento de código se muestra un ejemplo de uso de dicha herramienta.

```
...
require_once '../htmlpurifier/library/HTMLPurifier.auto.php';
$config = HTMLPurifier_Config::createDefault();
$config->set('Core.Encoding', 'UTF-8');
$config->set('HTML.Doctype', 'HTML 4.01 Transitional');
$purifier = new HTMLPurifier($config);
...
$parsed_content = $purifier->purify($_POST['editor3']);
...
```

Para usar dicha herramienta debemos instanciar el purificador con una configuración determinada y finalmente aplicar el método `purify` sobre el texto que queremos purificar. Dado que la plataforma usa la codificación ISO-8859-1, en este punto tendremos que tener en mente una serie de consideraciones recogidas en el apéndice *Integración en la plataforma SWAD*.

Capítulo 5

Resultados

El editor resultante es un editor ligero y fácil de usar que puede insertarse en una página con apenas dos líneas de código (ver apéndice Integración en la plataforma SWAD). Se han realizado una serie de pruebas sobre el editor y aunque aún no se ha incorporado totalmente en la plataforma se han obtenido muy buenos resultados.

Durante el desarrollo del editor se han ido realizando pruebas en la página de test de la figura 5.1.

Editor 1 (Div)

Editor 2 (Div)

Editor 3 (TextArea)

Enviar

Figura 5.1: Página de pruebas sin SWADE

Esta página posee dos elementos div y un textarea, identificados por el atributo HTML clase “swade”. Insertando en la cabecera de la página las líneas de código siguientes:

```
<script src="../../swade.js" type="text/javascript"></script>
<script>
  SwadeManager.setOnDOMLoaded(function(){
    SwadeManager.setSwadeByClassName("swade");});
</script>
```

Nuestro editor hará el resto del trabajo, transformando los tres elementos citados en editores SWADE. Ver figura 5.2.



Figura 5.2: Página de pruebas con SWADE

Esta página contiene también un formulario cuyos únicos elementos son el textarea inferior y el botón de submit, responsable de enviar el contenido del textarea. Nuestro editor se ocupará de interceptar el evento de pulsación del botón para asignar el contenido del editor al textarea antes de ser enviado. De esta forma se enviará el código HTML que contiene el editor.

Para probar el filtrado HTML se ha usado la página de la figura 5.3. Esta página es un script php que recoge el valor enviado por el formulario y muestra su valor filtrado y sin filtrar,

Test de Content Parser

Código enviado:

```
<span onclick="alert('XSS1');" style="font-weight: bold;javascript:alert(XSS2);font-style: italic;">Editor 3 (TextArea)&nbsp;</span>
```

Código parseado:

```
<span style="font-weight:bold;font-style:italic;">Editor 3 (TextArea)&#160;</span>
```

Contenido parseado:

Editor 3 (TextArea)

Figura 5.3: Páginas de pruebas de limpieza HTML

además de renderizar el HTML filtrado.

En las figuras 5.4 y 5.5 podemos ver el envío de símbolos y fórmulas.

Una vez testada la funcionalidad del editor, hicimos algunas pruebas de integración en las secciones descritas en el capítulo de Análisis.

En la figura 5.6 podemos ver el editor integrado en la sección de envío de mensajes y en las figuras 5.7 y 5.8 lo vemos en la sección de Test.

Figura 5.4: Envío en un formulario de símbolos y fórmulas.

TEST

Símbolos $\alpha\beta\gamma \oplus \otimes \oslash$ y **fórmulas matemáticas** $J_{\alpha}(x) = \sum_{m=0}^{\infty} \frac{(-1)^m}{m! \Gamma(m + \alpha + 1)} \left(\frac{x}{2}\right)^{2m+\alpha}$

Figura 5.5: Limpieza del HTML en el servidor y renderizado del contenido enviado en el formulario.



 Universidad de Granada Centro de Enseñanzas Virtuales

Figura 5.7: SWADE en preguntas de Test

Figura 5.7: SWADE en preguntas de Test

Respuesta	Realimentación
<p><input type="radio"/> a) 2 m</p> <div> <div> <div> <div>B</div> <div>I</div> <div>U</div> <div>S</div> <div>T_x</div> </div> <div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> </div> <div> <div> <div>x₂</div> <div>x²</div> <div>∞</div> <div>∫</div> </div> <div> <div>Format Text ▾</div> <div>Font Family ▾</div> <div>Ω</div> </div> <div> <div>T_x</div> <div></></div> </div> </div> </div> </div>	<div> <div> <div>B</div> <div>I</div> <div>U</div> <div>S</div> <div>T_x</div> </div> <div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> </div> <div> <div> <div>x₂</div> <div>x²</div> <div>∞</div> <div>∫</div> </div> <div> <div>Format Text ▾</div> <div>Font Family ▾</div> <div>Ω</div> </div> <div> <div>T_x</div> <div></></div> </div> </div> </div>
<p><input type="radio"/> b) 0.2 m</p> <div> <div> <div> <div>B</div> <div>I</div> <div>U</div> <div>S</div> <div>T_x</div> </div> <div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> </div> <div> <div> <div>x₂</div> <div>x²</div> <div>∞</div> <div>∫</div> </div> <div> <div>Format Text ▾</div> <div>Font Family ▾</div> <div>Ω</div> </div> <div> <div>T_x</div> <div></></div> </div> </div> </div></div>	<div> <div> <div>B</div> <div>I</div> <div>U</div> <div>S</div> <div>T_x</div> </div> <div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> </div> <div> <div> <div>x₂</div> <div>x²</div> <div>∞</div> <div>∫</div> </div> <div> <div>Format Text ▾</div> <div>Font Family ▾</div> <div>Ω</div> </div> <div> <div>T_x</div> <div></></div> </div> </div> </div>
<p><input type="radio"/> c) $\sqrt{2}$ m</p> <div> <div> <div> <div>B</div> <div>I</div> <div>U</div> <div>S</div> <div>T_x</div> </div> <div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> </div> <div> <div> <div>x₂</div> <div>x²</div> <div>∞</div> <div>∫</div> </div> <div> <div>Format Text ▾</div> <div>Font Family ▾</div> <div>Ω</div> </div> <div> <div>T_x</div> <div></></div> </div> </div> </div></div>	<div> <div> <div>B</div> <div>I</div> <div>U</div> <div>S</div> <div>T_x</div> </div> <div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> <div> <div> <div></div> <div></div> </div> <div> <div></div> <div></div> </div> </div> </div> <div> <div> <div>x₂</div> <div>x²</div> <div>∞</div> <div>∫</div> </div> <div> <div>Format Text ▾</div> <div>Font Family ▾</div> <div>Ω</div> </div> <div> <div>T_x</div> <div></></div> </div> </div> </div>

Figura 5.8: SWADE en respuestas de Test

Capítulo 6

Conclusiones

Existe un mundo de posibilidades en el mercado laboral en lo que a aplicaciones web se refiere. Aunque con los conocimientos adquiridos en la carrera se puede aprender una tecnología concreta en un tiempo razonable, si bien es verdad que al inicio de tu vida laboral te piden experiencia en ciertas herramientas que en muchos casos no has adquirido.

Por eso, una de mis motivaciones para la elección de este proyecto de fin de carrera era aprender todo lo posible acerca de aplicaciones web y las tecnologías actuales para su desarrollo. En este punto me siento muy satisfecho con el trabajo desarrollado, ya que a lo largo del año he adquirido conocimientos de tecnologías concretas como son HTML, PHP y Javascript y metodologías de desarrollo como AJAX entre otras. Además del aprendizaje y la experiencia de tecnologías específicas, también he aprendido mucho acerca de aplicaciones web, lo que facilitará mi aprendizaje futuro de otras herramientas.

En una fase inicial me dediqué al aprendizaje de las distintas herramientas para el desarrollo del editor. He aprendido mucho de la web <http://w3schools.com/> y debo mucho a la comunidad de desarrolladores de <http://stackoverflow.com/>, conocimiento que espero poder devolver en algún momento.

Mi segunda motivación era la resolución de un problema real, construir algo que sirviera a la gente, y que no se quedara en el CDROM de la estantería de un departamento de la ETSIIT. Por este motivo me puse en contacto con Antonio Cañas, pues soy consciente de la amplia difusión de SWAD en la UGR, y me ofrecí para llevar a cabo alguna de las ideas que tuviera en mente para mejorar dicha plataforma.

Es distinto el desarrollo de software para unas prácticas de una asignatura que desarrollarlo

para su uso en una plataforma real, que será usada al día por cientos de personas. Lo que en unas prácticas es secundario (para subir nota), aquí es primordial y la atención a los pequeños detalles lo es todo. He pasado muchas horas solucionando pequeños errores sutiles o trabajando para que el editor funcione sin problemas en los distintos navegadores más usados. También he pasado muchas horas leyendo foros y FAQs y trabajando en la configuración de las herramientas externas como texvc y HTMLpurifier para conseguir que funcionaran correctamente. Los detalles de dicha configuración pueden encontrarse en los apéndices.

Puede probarse la versión final del editor en una página de test en <http://openswad.org/swade/test/>. La solución que propongo aún necesita pasar algunas pruebas antes de estar completamente integrada en la plataforma, pero continuaré colaborando junto con Antonio Cañas para su completa implantación. Es por este motivo por el que he liberado el código fuente de SWADE con la licencia Affero GPL y se puede descargar de <https://github.com/asce/swade>, de forma que cualquiera que esté interesado podrá colaborar en la mejora del editor.

APÉNDICES

Instalación en el Servidor de OpenSWAD

El servidor donde se encuentra OpenSWAD usa una instalación mínima del SO Linux CentOS 5.7 en su versión de 64 bits. SWADE usa distintas herramientas libres para desarrollar parte de su funcionalidad. Estas herramientas son HTMLpurifier [3], texvc [2] y MathJax [1], ya comentadas anteriormente. A continuación describiremos el proceso de instalación de dichas herramientas.

En primer lugar instalamos mediawiki y la última versión de php en el servidor. Mediawiki se usará para una futura implementación de una wiki para OpenSWAD, lo cuál no pertenece al ámbito de este documento. La instalación de php, sin embargo, era necesaria ya que actualmente no se usa php en la plataforma y nos será de utilidad para algunas de las funciones que vamos a desarrollar.

HTMLpurifier se usa para filtrar el código HTML generado y protegernos de la inyección de posible código malicioso. Deberá instalarse en el fichero raíz de SWADE como `htmlpurifier`. La página de dicha herramienta es: <http://htmlpurifier.org/>. Será necesario configurar la codificación de caracteres y doctype en el fichero `content_parser.php`. Existe un ejemplo de uso de la función de filtrado en `test/form.php`.

Para usar la caché de HTMLpurifier será necesario dar permisos de escritura a los directorios que cuelgan de `library/HTMLPurifier/DefinitionCache/Serializer`. Para ello usaremos el comando:

```
chmod -R 0755 library/HTMLPurifier/DefinitionCache/Serializer
```

Texvc se usa para generar las imágenes de las formulas LaTeX y forma parte de la extensión `math` de mediawiki. Antes de compilar dicha herramienta debemos instalar sus dependencias,

que son:

- ocaml
- LaTeX
- dvipng

La última ya se encuentra instalada así que nos centramos en la instalación de las otras dos.

Para instalar ocaml descargamos las fuentes de su página oficial (= <http://caml.inria.fr/download.en.html>) y a continuación la compilamos e instalamos con los siguientes comandos:

```
./configure
make world
make opt
sudo make install
make clean
```

LaTeX podemos instalando usando el gestor de paquetes yum mediante el comando:

```
yum install tetex
```

Finalmente para la compilación de la herramienta texvc, una vez resueltas sus dependencias, nos bajamos la extensión math de mediawiki de su página: http://www.mediawiki.org/wiki/Special:ExtensionDistributor?extdist_extension=Math. Una vez descargados los ficheros fuente, accedemos al directorio math/ y ejecutamos el comando:

```
make
```

Será necesario copiar el binario generado dentro del directorio tex_editor, que se encuentra en el directorio raíz de SWADE.

Deberemos compilar dicha herramienta e incluirla en la carpeta tex_editor (o en el path). Para la generación de las imágenes de las fórmulas debemos dar permisos de escritura a las carpetas tex_editor/img y tex_editor/tmp. Podemos modificar la localización de estas carpetas en el script php tex_editor/tex2png.php. Para más información acerca de texvc consultar: <http://www.mediawiki.org/wiki/Texvc>

En el caso de no generarse correctamente las imágenes de las fórmulas debemos consultar si tenemos instalados las herramientas latex, dvips, gs y convert (ImageMagick).


```
which latex
which dvips
which gs
which convert
```

En nuestro caso tuvimos que instalar ImageMagick con:

```
yum install ImageMagick
```

Si se generan los archivos temporales .tex pero no se genera la imagen el problema puede solucionarse copiando a la carpeta de los archivos temporales el fichero latex.fmt. Este fichero puede estar en varios sitios, en nuestro caso se encontraba en `/.texmf-var/web2c/latex.fmt`. En caso de no aparecer en dicha ruta usar el comando find.

```
find ~ -name latex.fmt
```

Para cualquier otro problema en el uso de texvc consultar http://www.mediawiki.org/wiki/Manual:Enable_TeX/problems.

Para la visualización de las fórmulas LaTeX mientras el usuario escribe su código se usa la herramienta MathJax. Dicha herramienta deberá instalarse en la carpeta tex_editor de SWADE. La página de dicha herramienta es <http://www.mathjax.org/>

Integración en la plataforma SWAD

Insertar SWADE en una página web es tan sencillo como incluir el fichero swade.js en la página y elegir uno de los tres métodos que ofrece para instanciar este en los textarea o div.

```

<script src="path-to-swade/swade.js" type="text/javascript"></script>
<script>
SwadeManager.setOnDOMLoaded(function(){
    SwadeManager.setSwadeByQuery("textarea.swade"); //1
    SwadeManager.setSwadeById("swade-id");           //2
    SwadeManager.setSwadeByClassName("swade-class");//3
}
);
</script>

```

Con el primero de ellos instanciamos un editor en aquellos textarea con la clase “swade”, con el segundo lo haremos para los elementos div o textarea con id “swade-id” y con el último instanciamos el editor para los elementos div o textarea con clase “swade-class”.

Sin embargo, a la hora de integrar SWADE en la plataforma SWAD uno de los mayores problemas al que nos enfrentamos es la codificación de caracteres. Nuestro editor permite la introducción de caracteres Unicode (UTF-8) mientras que SWAD usa una codificación ISO Latin 1 (iso-8859-1) por lo que tendremos que tener especial cuidado en el tratamiento del código HTML generado por el usuario.

Si queremos enviar el contenido de un editor usando un formulario deberemos asignarle al atributo `accept-charset` el valor `"iso-8859-1"`. De esta forma nos aseguramos que envíe como entidad los caracteres Unicode que ISO Latin 1 no reconoce.

```
<form action="form.php" accept-charset="iso-8859-1">
...
</form>
```

De la misma manera, en el servidor, tendremos que codificar los caracteres desconocidos con entidades cuando accedamos al valor que hemos enviado en el formulario. Pero antes de realizar esta tarea debemos asegurarnos de sanear el HTML para evitar ataques XSS [18].

Para realizar esta tarea debemos crear el purificador:

```
require_once '../htmlpurifier/library/HTMLPurifier.auto.php';
$config = HTMLPurifier_Config::createDefault();
$config->set('Core.Encoding', 'ISO-8859-1'); // replace with your encoding
$config->set('HTML.Doctype', 'HTML 4.01 Transitional'); // replace with your doctype
$config->set('Core.EscapeNonASCIICharacters', true);
//ISO-8859-1 -> UTF-8 -> clean html -> UTF-8 -> EscapeNonASCII -> ISO-8859-1
// if we don't add that option the purifier will mess our entities
$purifier = new HTMLPurifier($config);
```

Debemos habilitar la opción `EscapeNonASCIICharacters` para visualizar correctamente los símbolos introducidos.

Y para purificar usaremos:

```
$content = $purifier->purify($_POST['editor3'])
```

En este punto podremos acceder al contenido:

```
//htmlentities(string,flags,character-set,double_encode)
$content = htmlentities($content,0,'ISO-8859-1',TRUE);
```

Como ya hemos dicho, usamos `htmlentities` para codificar los símbolos como entidades.

Y una vez saneado el código, es seguro renderizarlo como HTML.

```
<div id="content" style="border:1px solid black;width:800px;min-height:250px;">
<?php echo html_entity_decode($content);
/*Decodificamos entidades para mostrarlas*/
?>
</div>
```

El ejemplo utilizado podemos encontrarlo en el directorio `src/test/`.

Software Libre y Código Abierto

Aunque para muchos es lo mismo, ambos términos defienden movimientos distintos. Software Libre (Free Software en inglés) defiende la libertad de los usuarios en el uso del software. Software libre no solo implica que el usuario puede acceder al código fuente del software, sino que garantiza una serie de libertades como son ejecutar, copiar, distribuir, y estudiar el mismo, e incluso modificar el software y distribuirlo modificado. Software Libre es un movimiento ético más que práctico.

La expresión software de Código Abierto se propuso originalmente para evitar un posible malentendido con el término free software (software libre), pero pronto se asoció con posiciones filosóficas diferentes a las del movimiento del software libre. Tal como Richard Stallman comenta en su artículo [20], el término fue adoptado para vender el modelo de desarrollo de software libre a las empresas, indicándole las ventajas de este, como son la robustez, estabilidad y calidad del software desarrollado pero omitiendo las obligaciones éticas que conlleva el software libre. Por otra parte, el término código abierto tampoco queda exento de ambigüedad ya que lo que refleja es aquel software que permite visualizar su código.

En la práctica, el código abierto sostiene criterios un poco más débiles que los del software libre. Por lo que sabemos, todo el software libre existente se puede calificar como código abierto. Casi todo el software de código abierto es software libre, con algunas excepciones. Algunas licencias de código abierto son demasiado restrictivas, por lo que no se las puede calificar como licencias de software libre. Afortunadamente esas licencias no se usan en muchos programas.

En el citado artículo de Stallman se cuenta como algunas empresas usan software libre en dispositivos cuya licencia no permite la modificación de este software por parte del usuario,

conociéndose este hecho como Tivoización. Aunque el código fuente sea libre, estos ejecutables no lo son. Según los criterios del código abierto, esto no es un problema; sólo les interesa la licencia del código fuente.

Licencias de distribución de Software Libre

En este anexo se incluirá el texto de las distintas licencias de Software Libre bajo las que se distribuyen los distintos editores WYSIWYG analizados anteriormente. En el caso de que la licencia sea demasiado extensa, se añade un resumen de sus implicaciones. Este texto es solo orientativo, para el uso final de dichas licencias se recomienda la lectura del contenido oficial de estas, cuyos enlaces se encuentran en la bibliografía.

D.1. MIT License

El texto diferencia tres puntos:

- Condiciones, la condición es que la nota de copyright y la parte de los derechos se incluya en todas las copias o partes sustanciales del Software. Esta es la condición que invalidaría la licencia en caso de no cumplirse.
- Derechos, los derechos son muchos: sin restricciones; incluyendo usar, copiar, modificar, integrar con otro Software, publicar, sublicenciar o vender copias del Software, y además permitir a las personas a las que se les entregue el Software hacer lo mismo.
- Limitación de responsabilidad, finalmente se tiene un disclaimer o nota de limitación de la responsabilidad habitual en este tipo de licencias.

Puede encontrar el texto completo de la licencia en [\[15\]](#).

D.2. Mozilla Public License v2

La licencia MPL cumple completamente con la definición de software de código abierto de la Open Source Initiative (OSI) y con las cuatro libertades del software libre enunciadas por la Free Software Foundation (FSF). Sin embargo la MPL deja abierto el camino a una posible reutilización no libre del software, si el usuario así lo desea, sin restringir la reutilización del código ni el relicenciamiento bajo la misma licencia.

Puede encontrar el texto completo de la licencia en [\[17\]](#).

D.3. BSD License

La licencia BSD es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Es una licencia de software libre permisiva como la licencia de OpenSSL o la MIT License. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

El principal problema de esta es la clausula de publicidad, modificada en una posterior revisión.

Puede encontrar el texto completo de la licencia en [\[13\]](#).

D.4. GNU General Public License

Es la licencia más ampliamente usada en el mundo del software y garantiza a los usuarios finales la libertad de usar, estudiar, compartir (copiar) y modificar el software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

Esta es la primera licencia copyleft para uso general. Copyleft significa que los trabajos derivados sólo pueden ser distribuidos bajo los términos de la misma licencia. Bajo esta filosofía, la licencia GPL garantiza a los destinatarios de un programa de ordenador los derechos-libertades reunidos en definición de software libre (free software definition) y usa copyleft para asegurar que el software está protegido cada vez que el trabajo es distribuido

Puede encontrar el texto completo de la licencia en [\[16\]](#).

D.5. GNU Lesser General Public License

La principal diferencia entre GPL y LGPL es que la primera condiciona a que, dada una librería distribuida bajo GPL, solo puede ser usada por un software distribuido bajo GPL. La segunda aporta una serie de permisos adicionales.

Puede encontrar el texto completo de la licencia en [\[14\]](#).

D.6. GNU Affero General Public License v3

AGPLv3 es la licencia bajo la que se distribuye Swad. Es una licencia copyleft derivada de la licencia GPLv3 y compatible con ella, diseñada específicamente para asegurar la cooperación con la comunidad en el caso de software que corra en servidores de red.

Puede encontrar el texto completo de la licencia en [\[19\]](#).

APÉNDICE E

Código del editor

```
1
2 /*
3  SWADE is a lightweight WYSIWYG editor developed for SWAD platform
4  .
5  Copyright (C) 2013 - David Medina Godoy - asce88@gmail.com
6
7  This program is free software: you can redistribute it and/or
8  modify
9  it under the terms of the GNU Affero General Public License as
10  published by
11  the Free Software Foundation, either version 3 of the License, or
12  (at your option) any later version.
13
14  This program is distributed in the hope that it will be useful,
15  but WITHOUT ANY WARRANTY; without even the implied warranty of
16  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
17  GNU Affero General Public License for more details.
18
19  You should have received a copy of the GNU Affero General Public
20  License
21  along with this program. If not, see <http://www.gnu.org/
```

```
        licenses/>.
18
19 */
20
21
22 var swade_app = {};
23
24 function SwadeManager(){
25 }
26
27 SwadeManager.init = function(swade_app_object){
28
29     var scripts= document.getElementsByTagName('script');
30     for(var i = 0; i < scripts.length; i++){
31         var script_str = scripts[i].getAttribute('src');
32         if(script_str){
33             if(script_str.match('swade.js')){
34                 swade_app_object.path = scripts[i].src.split('?')[0];        //
35                 remove any ?query
36                 swade_app_object.path = swade_app_object.path.split('/').slice
37                     (0, -1).join('/')+'//'; // remove last filename
38             }
39         }
40
41         swade_app_object.focus = null;
42         swade_app.initialized = true;
43         swade_app_object.swade_data = new Array();
44         swade_app.targets = new Array();
45         swade_app_object.msie_selection = null;
46         document._swade_onclick = document.onclick;
```

```
46
47
48     swade_app_object.onclick_function = function(){
49 if(this.navigator == 'MSIE')
50     this.msie_selection = SwadeManager.saveSelection();
51 var elem = document.activeElement;
52 var index = elem.getAttribute('data-swade-index');
53 if(index){//Si se ha seleccionado editor
54     if(this.focus!=elem){
55         if(this.focus)
56             this.swade_data[this.focus.getAttribute('data-swade-index')]
57                 .panel.lock();
58         this.swade_data[index].panel.unlock();
59         this.focus = elem;
60     }
61 }else{//Si no se ha seleccionado editor
62     if(this.focus){//Si hay un editor previamente
63         seleccionado
64         if(this.swade_data[this.focus.getAttribute('data-swade-index')]
65             ].panel.dom_panel!=this.clicked_panel){
66             this.swade_data[this.focus.getAttribute('data-swade-index')]
67                 .panel.lock();
68             this.focus = null;
69         }
70     }
71 this.clicked_panel=null;
72 }
```

```

73
74 document.onclick = function(e){swade_app_object.
    onclick_function(e);if(this._swade onclick)this.
    _swade onclick();}
75 swade_app_object.clicked_panel=null;
76 var div_dom_elem = document.createElement('div');
77 div_dom_elem.innerHTML = '<link rel="stylesheet" type="text/css
    " href="'+swade_app_object.path+'swade.css"/>';
78 document.head.appendChild(div_dom_elem.firstChild);
79
80
81 var d = document.createElement('div');
82 d.innerHTML='<head><title>Select Symbol</title></head><table><
    tr><td>&#913;</td><td>&#914;</td><td>&#915;</td><td>&#916;</
    td><td>&#917;</td><td>&#918;</td><td>&#919;</td><td>&#920;</
    td><td>&#921;</td><td>&#922;</td><td>&#923;</td><td>&#924;</
    td><td>&#925;</td><td>&#926;</td><td>&#927;</td><td>&#928;</
    td><td>&#929;</td><td></td><td>&#931;</td><td>&#932;</td><td>
    &#933;</td><td>&#934;</td><td>&#935;</td><td>&#936;</td><td>
    &#937;</td></tr><tr><td>&#945;</td><td>&#946;</td><td>
    &#947;</td><td>&#948;</td><td>&#949;</td><td>&#950;</td><td>
    &#951;</td><td>&#952;</td><td>&#953;</td><td>&#954;</td><td>
    &#955;</td><td>&#956;</td><td>&#957;</td><td>&#958;</td><td>
    &#959;</td><td>&#960;</td><td>&#961;</td><td>&#962;</td><td>
    &#963;</td><td>&#964;</td><td>&#965;</td><td>&#966;</td><td>
    &#967;</td><td>&#968;</td><td>&#969;</td><tr><td>&#977;</td>
    <td>&#978;</td><td>&#982;</td></tr></tr><tr><td>&#8704;</td>
    <td>&#8706;</td><td>&#8707;</td><td>&#8709;</td><td>
    &#8711;</td><td>&#8712;</td><td>&#8713;</td><td>&#8715;</td>
    <td>&#8719;</td><td>&#8721;</td><td>&#8722;</td><td>
    &#8727;</td><td>&#8730;</td><td>&#8733;</td><td>&#8734;</td>

```



```

      ><td>##8736;</td><td>##8743;</td><td>##8744;</td><td>
      >##8745;</td><td>##8746;</td><td>##8747;</td><td>##8756;</td>
      ><td>##8764;</td><td>##8773;</td><td>##8776;</td></tr><tr><
      td>##8800;</td><td>##8801;</td><td>##8804;</td><td>##8805;</
      td><td>##8834;</td><td>##8835;</td><td>##8836v;</td><td>
      >##8838;</td><td>##8839;</td><td>##8853;</td><td>##8855;</td>
      ><td>##8869;</td><td>##8901;</td></tr></table>';

83
84     var a = d.getElementsByTagName('td');
85     for(i=0;i<a.length;i++){
86         if(a[i].innerHTML){
87
88             a[i].setAttribute('onclick','this.parentNode.parentNode.
                parentNode.parentNode.parentNode.style.visibility="hidden
                ";SwadeManager.restoreSelection(swade_app.selection);
                SwadeManager.pasteHtmlAtCaret(this.innerHTML);');
89             a[i].style.cursor='pointer';
90         }
91
92         swade_app_object.symbols_dialog = new SwadeDialog('Select
                Symbol',d.innerHTML,40,40);
93     if(navigator.userAgent.search('MSIE')>=0)
94         swade_app_object.navigator = 'MSIE';
95     else
96         swade_app_object.navigator = 'other';
97     }
98
99     swade_app_object.hash = {};
100     swade_app_object.hash['formatBlock'] = {}
101     swade_app_object.hash['formatBlock']['p'] = 1;
102     swade_app_object.hash['formatBlock']['pre'] = 2;

```

```
103     swade_app_object.hash['formatBlock']['h6'] = 3;
104     swade_app_object.hash['formatBlock']['h5'] = 4;
105     swade_app_object.hash['formatBlock']['h4'] = 5;
106     swade_app_object.hash['formatBlock']['h3'] = 6;
107     swade_app_object.hash['formatBlock']['h2'] = 7;
108     swade_app_object.hash['formatBlock']['h1'] = 8;
109
110     swade_app_object.hash['fontName'] = {};
111     swade_app_object.hash['fontName']['serif'] = 1;
112     swade_app_object.hash['fontName']['monospace'] = 2;
113
114 }
115
116 SwadeManager.makeSwade = function(dom_elem,target_index)
117 {
118     //If the DOM element we want to edit exists
119     if(dom_elem)
120     {
121         try {
122             document.execCommand("styleWithCSS", 0, true);
123         } catch (e) {
124             try {
125                 document.execCommand("useCSS", 0, false); //Revisar
126                     en internet explorer
127             } catch (e) {
128                 try {
129                     document.execCommand('styleWithCSS', false,
130                         true);
131                 }
132             }
133         }
134     }
135 }
```

```

132         }
133     }
134
135
136     dom_elem.style.border='1px solid black';
137     dom_elem.style.backgroundColor='white';
138
139     var buttons_container = document.createElement('div');
140
141     buttons_container.innerHTML=''+
142         '<div class="swade-buttons-panel" style="display: block
143             ;text-align:center;margin:auto;padding-top:5px;max-
144             width:500px;">'+
145         '<button class="swade-button" data-tag="bold" title="Click to
146             Bold"></
147             button>'+
148         '<button class="swade-button" data-tag="italic" title="Click
149             to Italic"></button>'+
151         '<button class="swade-button" data-tag="underline" title="
152             Click to Underline"></button>'+
154         '<button class="swade-button" data-tag="strikeThrough" title
155             ="Click to Strikethrough"></button>'+
157         '<button class="swade-button" data-tag="removeFormat" title="
158             Remove Format" ></button>'+
160
161         '<button class="swade-button" data-tag="indent" title="Click to
162             Indent Text"></button>' +
150   '<button class="swade-button" data-tag="outdent" title="Click
      to Outdent Text"></button>' +
151   '<button class="swade-button" data-tag="justifyLeft" title="
      Left Align"></button>' +
152   '<button class="swade-button" data-tag="justifyCenter" title
      ="Center Align"></button>' +
153   '<button class="swade-button" data-tag="justifyRight" title="
      Right Align"></button>' +
154   '<button class="swade-button" data-tag="insertUnorderedList"
      title="Insert Unordered List" ></button>' +
155   '<button class="swade-button" data-tag="insertOrderedList"
      title="Insert Ordered List" ></button>' +
156   '<button class="swade-button" data-tag="subscript" title="
      Click to Subscript"></button>' +
157   '<button class="swade-button" data-tag="superscript" title="
      Click to Superscript"></button>' +
158   '<button class="swade-button" data-tag="createLink" title="
      Add Link"></
      button>' +
159   '<button class="swade-button" data-tag="unlink" title="Remove
      Link"></
      button>' +

```

```

160     '<button class="swade-button" data-tag="insertImage" title="
        Insert Image"></button>' +
161     '<span class="swade-select" ><select style="color:black;"
        data-tag="formatBlock">' +
162     '<option value="p" style="display:none">Format Text</option>'
        +
163     '<option value="p">paragraph</option>' +
164     '<option value="pre">pre</option>' +
165     '<option value="h6">heading 6</option>' +
166     '<option value="h5">heading 5</option>' +
167     '<option value="h4">heading 4</option>' +
168     '<option value="h3">heading 3</option>' +
169     '<option value="h2">heading 2</option>' +
170     '<option value="h1">heading 1</option>' +
171     '</select></span>' +
172     '<span class="swade-select" ><select style="color:black;"
        data-tag="fontName">' +
173     '<option value="serif" style="display:none">Font Family</
        option>' +
174     '<option value="serif">Serif</option>' +
175     '<option value="monospace">Monospace</option>' +
176     '</select></span>' +
177     '<button class="swade-button" data-tag="insertSymbol" title="
        Insert HTML Symbol"></button>' +
178     '<button class="swade-button" data-tag="insertLatex"
        title="Insert TeX Formula"></button>' +
179     '<button class="swade-button" data-tag="viewHTML" title="Edit
        HTML"></

```

```
        button>' +
180    '</div>';
181
182    buttons_container.firstChild.setAttribute('data-swade-
        index', target_index);
183    dom_elem.setAttribute('data-swade-index', target_index);
184
185    //dom_elem.panel = buttons_container.firstChild;
186    swade_app.swade_data[target_index].panel = new SwadePanel(
        buttons_container.firstChild, dom_elem);
187
188    var parent = dom_elem.parentNode;
189
190    if(parent.lastchild == dom_elem)
191    {
192        parent.appendChild(buttons_container.firstChild); //
        parent.appendChild(buttons_container);
193    }
194    else
195    {
196        parent.insertBefore(buttons_container.firstChild,
        dom_elem.nextSibling); //parent.insertBefore(
        buttons_container, dom_elem.nextSibling);
197    }
198
199
200    //For each of them...
201    for(var i=0, l=swade_app.swade_data[target_index].panel.buttons.
        length; i<l; i++){
202        //We bind the click event
203        swade_app.swade_data[target_index].panel.buttons[i].onclick =
```

```
function(){return false;};
204 swade_app.swade_data[target_index].panel.buttons[i].
    addEventListener('click',function(e){
205
206     if(!e) e = window.event;
207     swade_app.clicked_panel = this.parentNode;
208
209
210     //Code here...
211     var tag = this.getAttribute('data-tag');
212     switch(tag)
213     {
214
215     case 'createLink':
216
217         var sel = SwadeManager.saveSelection();
218         var link = prompt('Please specify the link.');
```

SwadeManager.restoreSelection(sel);

```
document.execCommand('createLink', false, link);
221         break;
222     case 'unlink':
223         document.execCommand('unlink', false, false);
224         break;
225
226     case 'insertImage':
227         var src = prompt('Please specify the link of the image.');
```

var sel = document.selection;

```
if (sel) {
230     var textRange = sel.createRange();
231     document.execCommand('insertImage', false, src);
232     textRange.collapse(false);
```

```
233     textRange.select();
234     } else {
235     document.execCommand('insertImage', false, src);
236     }
237
238     break;
239 case 'viewHTML':
240     if(swade_app.navigator=='MSIE'){
241     var args = new Object();
242     args.opener = window;
243     args.target = swade_app.swade_data[this.parentNode.
        getAttribute('data-swade-index')].panel.editor;
244     args.editor_content = dom_elem.innerHTML;
245     var sFeatures = '';
246     window.showModalDialog(swade_app.path+'viewHTML.html', args ,
        sFeatures)
247
248     }else{
249
250     var dataitem = SwadeManager.popupwindow(swade_app.path+'
        viewHTML.html', 'Edit Source', 500, 500);
251     dataitem.opener = window;
252     dataitem.target = swade_app.swade_data[this.parentNode.
        getAttribute('data-swade-index')].panel.editor;
253     dataitem.editor_content=dom_elem.innerHTML;
254
255     }
256
257
258     break;
259 case 'insertLatex':
```



```
260         SwadeManager.insertLatex("", false);
261
262         break;
263     case 'insertSymbol':
264
265         swade_app.selection = SwadeManager.saveSelection();
266         swade_app.symbols_dialog.alignWithElem(this.
            getBoundingClientRect());
267         swade_app.symbols_dialog.change_visibility();
268
269         break;
270     case 'indent':
271         if(swade_app.navigator=='MSIE')
272             SwadeManager.restoreSelection(swade_app.msie_selection);
273         document.execCommand('indent', false, null);
274         break;
275     case 'outdent':
276         if(swade_app.navigator=='MSIE')
277             SwadeManager.restoreSelection(swade_app.msie_selection);
278         document.execCommand('outdent', false, null);
279         break;
280     default:
281         var value = null;
282
283         document.execCommand(tag, false, value);
284     }
285     swade_app.swade_data[this.parentNode.getAttribute('data-swade-
        index')].panel.refreshSelection();
286
287     if(e.preventDefault)
288         e.preventDefault();
```

```
289     else e.returnValue = false;
290
291     },false);
292
293 }
294     for(var i=0, l=swade_app.swade_data[target_index].panel.
        selects.length; i<l; i++){
295
296         swade_app.swade_data[target_index].panel.selects[i].
            addEventListener('click',function(e){
297 if(!e) e = window.event;
298         swade_app.clicked_panel = this.parentNode.
            parentNode;
299
300         if(e.preventDefault)
301             e.preventDefault();
302     else e.returnValue = false;
303     },false);
304
305         swade_app.swade_data[target_index].panel.selects[i].
            addEventListener('change',function(e){
306 if(!e) e = window.event;
307 swade_app.clicked_panel = this.parentNode.parentNode;
308 var tag = this.getAttribute('data-tag');
309         switch(tag)
310         {
311             case 'formatBlock':
312                 document.execCommand('formatBlock', false, '<'+this.value+
                    '>');
313                 break;
314             case 'fontName':
```

```
315     document.execCommand('fontName', false, this.value);
316     break;
317 }
318
319     if(e.preventDefault)
320         e.preventDefault();
321     else e.returnValue = false;
322
323     },false);
324
325 }
326
327
328     dom_elem.setAttribute('contenteditable', true);
329
330
331 }
332     return dom_elem;
333 };
334
335 SwadeManager.insertLatex = function(formula, bool_edit){
336
337
338     if(swade_app.navigator=='MSIE'){
339
340     var args = new Object();
341     args.opener = window;
342
343     args.formula = formula;
344     args.sel = SwadeManager.saveSelection();
345
```

```
346   var sFeatures = '';
347   var tex_img = window.showModalDialog(swade_app.path+'tex_editor/
      index.html',args , sFeatures);
348   if(bool_edit){
349       var d = document.createElement('div');
350       d.innerHTML = tex_img;
351       d = d.firstChild;
352       swade_app.selected_tex_object.parentNode.insertBefore(d,
          swade_app.selected_tex_object);
353       swade_app.selected_tex_object.parentNode.removeChild(
          swade_app.selected_tex_object);
354       delete swade_app.selected_tex_object;
355   }else
356       SwadeManager.pasteHtmlAtCaret(tex_img);
357
358
359   }else{
360   var dataitem = SwadeManager.popupwindow(swade_app.path+'
      tex_editor/index.html', 'LaTeX Editor', 800, 500);
361   dataitem.opener = window;
362   dataitem.formula = formula;
363
364   dataitem.sel = SwadeManager.saveSelection();
365
366   dataitem.onbeforeunload = function(){
367       if(window.tex){
368       if(bool_edit){
369           var d = document.createElement('div');
370           d.innerHTML = window.tex;
371           d = d.firstChild;
372           swade_app.selected_tex_object.parentNode.insertBefore(d,
```

```
        swade_app.selected_tex_object);
373     swade_app.selected_tex_object.parentNode.removeChild(
        swade_app.selected_tex_object);
374     delete swade_app.selected_tex_object;
375 }else
376     SwadeManager.pasteHtmlAtCaret(window.tex);
377
378 }
379
380     window.tex='';
381 };
382 }
383 }
384
385
386
387 SwadeManager.saveSelection = function() {
388     if (window.getSelection()) {
389         sel = window.getSelection();
390         if (sel.getRangeAt && sel.rangeCount) {
391             var ranges = [];
392             for (var i = 0, len = sel.rangeCount; i < len; ++i) {
393                 ranges.push(sel.getRangeAt(i));
394             }
395             return ranges;
396         }
397     } else if (document.selection && document.selection.createRange
        ) {
398         return document.selection.createRange();
399     }
400     return null;
```

```
401 }
402
403 SwadeManager.restoreSelection = function(savedSel) {
404     if (savedSel) {
405         if (window.getSelection()) {
406             sel = window.getSelection();
407             sel.removeAllRanges();
408             for (var i = 0, len = savedSel.length; i < len; ++i) {
409                 sel.addRange(savedSel[i]);
410             }
411         } else if (document.selection && savedSel.select) {
412             savedSel.select();
413         }
414     }
415 }
416
417 SwadeManager.pasteHtmlAtCaret = function(html) {
418     var sel, range;
419     if (window.getSelection()) {
420         // IE9 and non-IE
421         sel = window.getSelection();
422         if (sel.getRangeAt && sel.rangeCount) {
423             range = sel.getRangeAt(0);
424             range.deleteContents();
425
426             // Range.createContextualFragment() would be useful
427             // here but is
428             // non-standard and not supported in all browsers (IE9,
429             // for one)
430             var el = document.createElement("div");
431             el.innerHTML = html;
```

```
430         var frag = document.createDocumentFragment(), node,
            lastNode;
431         while ( (node = el.firstChild) ) {
432             lastNode = frag.appendChild(node);
433         }
434         range.insertNode(frag);
435
436         // Preserve the selection
437         if (lastNode) {
438             range = range.cloneRange();
439             range.setStartAfter(lastNode);
440             range.collapse(true);
441             sel.removeAllRanges();
442             sel.addRange(range);
443         }
444     }
445     } else if (document.selection && document.selection.type != "
        Control") {
446         // IE < 9
447         document.selection.createRange().pasteHTML(html);
448     }
449 }
450
451 SwadeManager.popupwindow = function(url, title, w, h) {
452     var left = (screen.width/2)-(w/2);
453     var top = (screen.height/2)-(h/2);
454     return window.open(url, title, 'toolbar=no, location=no,
        directories=no, status=no, menubar=no, scrollbars=yes,
        resizable=yes, copyhistory=no, width='+w+', height='+h+',
        top='+top+', left='+left);
455 }
```

```
456
457 SwadeManager.setSwadeByClassName = function(class_name){
458
459     var targets = document.getElementsByClassName(class_name);
460     SwadeManager.setSwade(targets);
461
462 }
463
464 SwadeManager.setSwadeById = function(id){
465
466     var target = document.getElementById(id);
467     var targets = new Array();
468     targets.push(target);
469     SwadeManager.setSwade(targets);
470
471 }
472
473 SwadeManager.setSwadeByQuery = function(query){
474
475     var targets = document.querySelectorAll(query);
476     SwadeManager.setSwade(targets);
477
478 }
479
480
481 SwadeManager.setSwade = function(selected_targets){
482     //console.log(targets);
483     if(!selected_targets) return;
484     if(!swade_app.initialized)
485     SwadeManager.init(swade_app);
486
```



```
487     var targets = new Array();
488     //Anadimos a targets los objetos no anadidos previamente
489     for(var i = 0; i < selected_targets.length; i++){
490     if(swade_app.targets.indexOf(selected_targets[i])==-1){
491         targets.push(selected_targets[i]); //Si el target no se ha
492         anadido previamente lo anadimos
493         swade_app.targets.push(selected_targets[i]);
494     }
495     }
496     var keydown_function = function(e) {
497
498     //if(swade_app.navigator != 'MSIE'){
499     if(!e) e = window.event;
500     swade_app.swade_data[this.getAttribute('data-swade-index')].panel
501     .refreshSelection(); //Actualizar seleccion en panel
502     if(e.which == 9) { //tab_keydown, problem with IE
503         document.execCommand('indent', false, null);
504
505         if(e.preventDefault)
506             e.preventDefault();
507         else e.returnValue = false;
508     }
509     //}
510     };
511
512     var click_function = function(e) {
513     if(!e) e = window.event;
514     //swade_app.focusEditor(this.getAttribute('data-swade-index'));
515     if(swade_app.tex_code_selected){
516         //console.log(swade_app.tex_code_selected);
```

```
516     SwadeManager.insertLatex(swade_app.tex_code_selected, true);
517     delete swade_app.tex_code_selected;
518 }
519
520
521 swade_app.swade_data[this.getAttribute('data-swade-index')].panel
    .refreshSelection(); //Actualizar seleccion en panel
522 }
523
524
525     for(var i = 0; i < targets.length; i++){
526 if(targets[i].tagName=='TEXTAREA' || targets[i].tagName=='DIV'){
527     swade_app.swade_data[i] = new Object();
528     var r = targets[i].getBoundingClientRect();
529 }
530 if(targets[i].tagName=='TEXTAREA'){
531     var divElem = document.createElement('div');
532     divElem.innerHTML = targets[i].innerHTML;
533     divElem.style.textAlign = targets[i].style.textAlign;
534     divElem.style.padding = '10px';
535     divElem.style.margin = 'auto';
536     targets[i].style.display='none';
537     targets[i].parentNode.insertBefore(divElem, targets[i]);
538     divElem.style.width = r.width+'px';
539     divElem.style.minHeight = r.height+'px';
540
541
542     var elem = targets[i].parentElement;
543     while(elem){
544 if(elem.tagName=="FORM"){
545     swade_app.swade_data[i].form = elem;
```

```
546     var forms_input = elem.getElementsByTagName('input');
547     for(j = 0;j < forms_input.length; j++){
548     if(forms_input[j].type=='submit')
549         swade_app.swade_data[i].input_submit = forms_input[j];
550     }
551
552     if(swade_app.swade_data[i].form)
553     if(swade_app.swade_data[i].input_submit){
554         swade_app.swade_data[i].input_submit.setAttribute('
                    onclick','return SwadeManager.setTextAreaBeforeSubmit(
                    '+i+');');
555     }
556 }
557 elem = elem.parentElement;
558 }
559 SwadeManager.makeSwade(divElem,i);
560 divElem.addEventListener('keydown', keydown_function, false);
561 divElem.addEventListener('click', click_function, false);
562 swade_app.swade_data[i].panel.lock();
563 }else if(targets[i].tagName=='DIV'){
564
565     targets[i].style.margin='auto';
566     targets[i].style.padding='10px';
567     targets[i].style.width = r.width+'px';
568     targets[i].addEventListener('keydown', keydown_function,false
        );
569     targets[i].addEventListener('click', click_function, false);
570     SwadeManager.makeSwade(targets[i],i);
571     swade_app.swade_data[i].panel.lock();
572 }
573
```

```
574     }
575
576 }
577
578
579 SwadeManager.setTextAreaBeforeSubmit = function(i){
580
581     swade_app.targets[i].innerHTML = swade_app.swade_data[i].panel.
        editor.innerHTML;
582     return true;
583 }
584
585
586 SwadeManager.setOnDOMLoaded = function(fun){
587
588     //window.onload = fun;
589
590     SwadeManager.bindReady(fun);
591
592     return true;
593 }
594
595 SwadeManager.bindReady = function(handler){
596
597     var called = false
598
599     function ready() {
600         if (called) return
601         called = true
602         handler()
603     }
```

```
604
605     if ( document.addEventListener ) { // native event
606 document.addEventListener( "DOMContentLoaded", ready, false )
607     } else if ( document.attachEvent ) { // IE
608
609     try {
610         var isFrame = window.frameElement != null
611     } catch(e) {}
612
613     // IE, the document is not inside a frame
614     if ( document.documentElement.doScroll && !isFrame ) {
615         function tryScroll(){
616             if (called) return
617             try {
618                 document.documentElement.doScroll("left")
619                 ready()
620             } catch(e) {
621                 setTimeout(tryScroll, 10)
622             }
623         }
624         tryScroll()
625     }
626
627     // IE, the document is inside a frame
628     document.attachEvent("onreadystatechange", function(){
629         if ( document.readyState === "complete" ) {
630             ready()
631         }
632     })
633     }
634
```

```
635 // Old browsers
636 if (window.addEventListener)
637     window.addEventListener('load', ready, false)
638 else if (window.attachEvent)
639     window.attachEvent('onload', ready)
640 else {
641     var fn = window.onload // very old browser, copy old onload
642     window.onload = function() { // replace by new onload and call
        the old one
643         fn && fn()
644         ready()
645     }
646 }
647 }
648
649
650 function SwadeDialog(title, content, left, top){
651     d = document.createElement('div');
652     d.innerHTML="<div class='dialog' onclick='SwadeManager.
        restoreSelection(swade_app.selection);'><div class='titlebar
        '><div class='dialog_title'>"+title+"</div><div class='
        close_dialog_button' onclick='this.parentNode.parentNode.
        style.visibility=\"hidden\";SwadeManager.restoreSelection(
        swade_app.selection);'></div></div><div class='
        dialog_content'>"+content+"</div><div style='clear:both'></
        div></div>";
653     d = d.firstChild;
654     document.body.appendChild(d);
655     d.style.left = left+'px';
656     d.style.top = top+'px';
657     d.style.visibility = 'hidden';
```

```
658     this._dialog = d;
659
660 }
661 SwadeDialog.prototype.show = function(){
662
663     this._dialog.style.visibility = 'visible';
664     return this;
665 }
666 SwadeDialog.prototype.close = function(){
667
668     this._dialog.style.visibility = 'hidden';
669
670 }
671 SwadeDialog.prototype.remove = function(){
672
673     this._dialog.parentNode.removeChild(this._dialog);
674
675 }
676 SwadeDialog.prototype.change_visibility = function(){
677
678     this._dialog.style.visibility = (this._dialog.style.visibility
679                                     == 'hidden'? 'visible': 'hidden');
680 }
681
682 SwadeDialog.prototype.alignWithElem = function(rect){
683     var rect_width = rect.right - rect.left;
684
685     var dialog_width;
686     if(this._dialog.getBoundingClientRect){
687         var rect_dialog = this._dialog.getBoundingClientRect();
```

```
688     dialog_width = rect_dialog.right - rect_dialog.left;
689     }else{
690     dialog_width = this._dialog.offsetRight - this._dialog.offsetLeft
        ;
691     }
692     this._dialog.style.left = rect.left - (dialog_width -
        rect_width)/2+'px';
693     this._dialog.style.top = rect.bottom+1+'px';
694 }
695
696 function SwadePanel(dom_panel,div_editor){
697     this.dom_panel = dom_panel;
698     this.buttons = this.dom_panel.querySelectorAll("button[data-tag
        ]");
699     this.selects = this.dom_panel.querySelectorAll("select[data-tag
        ]");
700     this.editor = div_editor;
701     for(var i = 0; i < this.buttons.length; i++){
702     this.buttons[i].onmousedown = function(){this.style.
        backgroundColor='white';};
703     this.buttons[i].onmouseup = function(){this.style.backgroundColor
        ='#EEE';};
704     }
705
706 }
707
708 SwadePanel.prototype.refreshSelection = function(){
709     var hash = {};
710     var dataTag;
711     var value, valueBool;
712     for(var i = 0; i < this.buttons.length; i++){
```



```
713     dataTag = this.buttons[i].getAttribute('data-tag');
714     try{
715         valueBool = document.queryCommandState(dataTag);
716     }catch(err){
717         valueBool = false;
718     }
719     hash[dataTag] = valueBool;
720
721     }
722
723     var button;
724
725     for (var k in hash) {
726
727     if (hash.hasOwnProperty(k)) {
728
729         var query = "[data-tag="+k+"]";
730         if(swade_app.navigator != 'MSIE'){
731             button = this.dom_panel.querySelector(query);
732             if(button){
733                 //console.log(k+'==>'+hash[k]);
734                 button.style.backgroundColor=(hash[k]==true?'#CCD0D0': '#EEE
735                     ');
736             }
737             }else{//IE ...
738             var index = this.dom_panel.getAttribute('data-swade-index');
739             var buttons = document.querySelectorAll(query);
740             buttons[index].style.backgroundColor=(hash[k]==true?'#CCD0D0': '#
741                 #EEE');
```

```
742     }
743   }
744   //////////////////////////////////////
745   for(var i = 0; i < this.selects.length; i++){
746     dataTag = this.selects[i].getAttribute('data-tag');
747     try{
748       value = document.queryCommandValue(dataTag);
749     }catch(err){
750       value = null;
751     }
752     if(value==null)
753       this.selects[i].selectedIndex = 0;
754     else this.selects[i].selectedIndex = swade_app.hash[dataTag][
       value];
755   }
756
757 }
758 SwadePanel.prototype.lock = function(){
759   this.dom_panel.style.opacity=0.4;
760   this.editor.style.border = "1px dashed gray";
761   childNodes = this.dom_panel.querySelectorAll('[data-tag]');
762   for(i = 0; i < childNodes.length; i++)
763     childNodes[i].disabled = true;
764
765 }
766 SwadePanel.prototype.unlock = function(){
767   this.dom_panel.style.opacity=1;
768   this.editor.style.border = "1px solid black";
769   childNodes = this.dom_panel.querySelectorAll('[data-tag]');
770   for(i = 0; i < childNodes.length; i++)
771     childNodes[i].disabled = false;
```

772 }

Bibliografía

- [1] <http://www.mathjax.org/>.
- [2] <http://www.mediawiki.org/wiki/TeXvc>.
- [3] <http://htmlpurifier.org/>.
- [4] <http://openswad.org/>.
- [5] <http://swad.ugr.es/stat/>.
- [6] <http://openswad.org/info/>.
- [7] <http://cksource.com/license/ckeditor>.
- [8] <http://developer.yahoo.com/yui/editor/>.
- [9] <http://nicedit.com/>.
- [10] <http://www.tinymce.com/>.
- [11] <http://www.openwebware.com/>.
- [12] http://en.wikipedia.org/wiki/Help:Displaying_a_formula.
- [13] <http://opensource.org/licenses/bsd-license.php>.
- [14] <http://www.gnu.org/licenses/lgpl.html>.
- [15] <http://opensource.org/licenses/MIT>.
- [16] <http://www.gnu.org/licenses/gpl.html>.

- [17] <http://www.mozilla.org/MPL/2.0/>.
- [18] http://en.wikipedia.org/wiki/Cross-site_scripting.
- [19] <http://www.gnu.org/licenses/agpl-3.0.html>.
- [20] R. Stallman. Por qué el código abierto pierde el punto de vista del software libre. Website.
<http://www.gnu.org/philosophy/open-source-misses-the-point.es.html>.
- [21] Xyne. Website. <http://xyne.archlinux.ca/projects/tex2png/>.