



UNIVERSIDAD DE VALLADOLID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

TRABAJO FIN DE MÁSTER

MÁSTER UNIVERSITARIO EN INVESTIGACIÓN
EN TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

Desarrollo y evaluación de una solución tecnológica destinada a mejorar el acceso a contenidos digitales por parte de población excluida socialmente

Autor:

D. Saúl Álvaro Muñoz

Tutor:

Dr. D. Miguel López-Coronado Sánchez-Fortún

Valladolid, 11 de Julio de 2012

TÍTULO: Desarrollo y evaluación de una solución tecnológica destinada a mejorar el acceso a contenidos digitales por parte de población excluida socialmente.

AUTOR: D. Saúl Álvaro Muñoz

TUTOR: Dr. D. Miguel López Coronado

DEPARTAMENTO: Teoría de la señal y Comunicaciones e Ingeniería Telemática

TRIBUNAL

PRESIDENTE: Dr. D. Luis A. Bailón Vega

VOCAL: Dr. D. José Fernando Díez Higuera

SECRETARIO: Dr. D. Carlos Gómez Peña

FECHA: 11 de Julio de 2012

CALIFICACIÓN:

RESUMEN DEL PROYECTO

El objetivo del TFM, es el desarrollo, implantación y posterior evaluación de una novedosa herramienta llamada RSS_PROYECT, basada en la tecnología RSS de sindicación de contenidos. En la actualidad, existen decenas de sindicadores Web de contenidos, aunque aún no existe uno diseñado para la búsqueda de noticias relativas a mujeres en riesgo de exclusión social ni posee las características y grado de configuración como el del programa sindicador RSS_PROYECT, presentado en este paper.

Para obtener los contenidos, utilizaremos 2 filtros (genérico y selectivo), configurados por el administrador desde el módulo RSS_PROYECT instalado en Joomla.

El filtro genérico permitirá hacer una búsqueda de las palabras introducidas en una serie de fuentes indexadas por el usuario. Este filtro mostrará todas las fuentes que contengan esta palabra, sin excepción. En el filtro selectivo, la condición para que se muestren las fuentes será que todas las palabras del filtro selectivo estén contenidas en la fuente.

Para realizar este proyecto hemos utilizado diversos lenguajes como: PHP, MySQL, HTML, XML y la Application Program Interface (API) de Joomla! A su vez, hemos usado el programa Firebug, para medir la velocidad de respuesta del módulo en dos casos: con el filtro selectivo y el genérico. Los resultados han sido favorables para el filtro selectivo, y muy favorables para el filtro genérico, concluyendo que el tiempo de procesado ha sido bajo y se ejecuta con eficiencia.

Se obtuvieron mejores tiempos promedio para el módulo RSS_PROYECT con respecto a otros módulos analizados en Joomla!

Hoyen día, esta herramienta es usada por el Centro Integral de Ayuda a la Mujer (CIAM) de Valladolid, España.

PALABRAS CLAVE:

Joomala!, RSS, Buscador, exclusión social, sistemas web.

ABSTRACT

The aim of this TFM, is the development, implementation and subsequent performance evaluation of an innovative tool called RSS_PROYECT, based on the RSS technology for content syndication. At the present, there are several Web content syndicators; although there is no one so far specifically designed for the search of news related to women in risk of social exclusion. They do not even have the characteristics and configuration level of the program syndicator RSS_PROYECT, presented in this paper.

To obtain these contents, we used 2 filters (generic and selective), set up by the administrator from the module RSS_PROYECT installed in Joomla!.

The generic filter allowed us to do a search of the words included in a series of sources indexed by the user. This filter showed all the sources that contained this word, without exception. In the selective filter, the condition on which the sources would appear would be that all the words of the selective filter were contained in the source.

We used different languages such as PHP, MySQL, HTML, XML and Joomla's! Application Program Interface (API). Furthermore, we used the program, Firebug to measure the response speed of the module, both with the selective and generic filters. The results were favorable for the selective filter, and very favorable for the generic filter, concluding that the processing time was short and executed efficiently.

Better average times were obtained for RSS_PROYECT with respect to other modules analysed in Joomla!.

Nowadays, this tool is used by the Comprehensive Center for Women's Help (Centro Integral de Ayuda a la Mujer, CIAM) of Valladolid, Spain.

KEY WORDS:

Joomla!, RSS, search engine, social exclusion, Web-based.

*Manuscrito de Baden-Powell encontrado
meses después tras su muerte.*

... Si alguna vez han visto la obra de "Peter Pan", recordarán cómo el jefe de los piratas estaba siempre haciendo su último discurso de despedida por temor de que, posiblemente, cuando llegara la hora en que habría de morir, no fuera a tener tiempo para darlo a conocer. Así me sucede a mí, y aún cuando no me estoy muriendo en este momento, esto tendrá que suceder uno de estos días, y deseo decirles una palabra de despedida...

... la felicidad no proviene simplemente de la riqueza, ni de tener éxito en la carrera, ni dándose uno gusto a sí mismo. Un paso hacia la felicidad es hacerse uno sano y fuerte cuando niño, para poder ser útil y así poder gozar de la vida cuando se es hombre. El estudio de la naturaleza les enseñará cómo Dios ha llenado de cosas bellas y maravillosas este mundo para que lo puedan gozar. Estén satisfechos con lo que les haya tocado y saquen de ello el mejor partido que pueda. Vean siempre el lado bueno de las cosas y no el malo.

Pero la verdadera manera de obtener la felicidad es haciendo felices a los demás. Traten de dejar este mundo en mejores condiciones de como lo encontraron; de ésta manera, cuando les llegue la hora de morir, podrán hacerlo felices porque, por lo menos, no perdieron el tiempo e hicieron cuanto les fue posible por hacer el bien. "Estén Listos" en ésta forma, para gozar de una vida dichosa...

Su amigo

Lord Robert Stephenson Smith Baden-Powell
(Fundador del movimiento SCOUT)

AGRADECIMIENTOS

Son muchas las personas a las que quiero dar las gracias y poco el espacio, así que intentaré no desperdiciarlo.

A mis padres y mi hermano pequeño, que tanto les quiero y siempre han estado a mi lado guiándome y dando su amor y apoyo .Gracias a ellos soy como soy, y por eso, parte de este trabajo es suyo.

A mis compis del Máster, por echarme una mano siempre que lo he necesitado; por los buenos momentos, las 'queridas' cargas de trabajo... y todos esos momentos que ahora recuerdas con una sonrisa. :)

A todos mis amigos: los de Palencia, la facultad, fútbol, el Conservatorio, los del pueblo, los de Scouts, etc. Habéis estado siempre ahí, haciéndome sonreír, el mejor regalo que se le puede dar a un amigo.

A mi tutor Miguel y mi profesora Isabel, por el apoyo, la ayuda y las facilidades que me han prestado para hacer este TFM. No siempre es fácil encontrar gente así, y por eso lo valoro y os estoy muy agradecido a los dos.

A Socorro, Directora del centro CIAM, por confiarme su página WEB y permitirme crear esta herramienta. Gracias por brindarme la oportunidad de trabajar para un cliente real, con sus exigencias, mejoras, modificaciones, etc. Sin duda me han enseñado muchas cosas que en la carrera no se imparte.

A todos vosotros, os doy sinceramente las gracias, pues este camino que he recorrido hasta acabar la carrera nunca podría haberlo hecho solo.

GRACIAS!

ÍNDICE GENERAL

1. CAPITULO 1: INTRODUCCION.....	1
1. 1 Introducción	3
1. 2 Objetivos	4
1. 3 Medios Utilizados	6
1. 4 Organización de la memoria.....	7
2. CAPITULO 2: INTRODUCCIÓN A LOS SINDICADORES DE NOTICIAS RSS	9
2. 1 Introducción a los sindicadores de noticias RSS	11
2. 2 Conceptos básicos.....	11
2.2.1 RSS.....	11
2.2.2. ¿Qué es Joomla!?	12
2. 3 Sindicadores fuera de Joomla!.....	13
2.3.1 Google reader.....	13
2.3.2 Omega reader	13
2.3.3 Netvibes	14
2.3.4 FeedDemon	15
2. 4 Sindicadores Rss dentro de Joomla!	16
2.4.1 Introducción	16
2.4.2 Módulo RX RSS Sticker.....	17
2.4.3 Feed	18
2.4.4 RSS Browser 2.7	19
2.4.5 Slick RSS	22
2. 5 ¿Por qué Joomla!?	23

3. CAPÍTULO 4: MÓDULO RSS PROYECT	27
3.1 Introducción	29
3.2 Justificación	29
3.3 Características técnicas	31
3.3.1 Configurabilidad	30
3.3.2 Compatibilidad	34
4. CAPITULO 4: MÉTODOS DE EVALUACIÓN	37
4.1 Velocidad de respuesta del módulo RSS_PROYECT	39
4.2 Resultados de velocidad obtenidos de los sindicadores	42
4.2.1 Módulo Feed	42
4.2.2 Módulo RX RSS Sticker	43
4.2.3 Módulo Slick RSS	43
4.2.4 Módulo Browser 2.7	44
4.2.5 Módulo RSS_PROYECT	45
4.3 Comparativa de velocidad de los sindicadores	45
5. CAPITULO 5: DISCUSIÓN DE LOS RESULTADOS, CONCLUSIONES Y LÍNEAS FUTURAS	49
5.1 Discusión de los resultados	51
5.2 Conclusiones.....	51
5.2 Líneas futuras	52
6. BIBLIOGRAFÍA	55
7. PUBLICACIONES.....	59

8. ANEXO I: TESTEO DEL MÓDULO RSS_PROYECT	69
9. ANEXO II: TESTEO DE LOS MÓDULOS IMPLEMENTADOS.....	89
10.APÉNDICE I: CÓDIGO IMPLEMENTADO	101

INDICE DE FIGURAS

Figura 2.2.1	Logo RSS.....	13
Figura 2.3.1	Logo Google Reader.....	14
Figura 2.3.2.A	Logo Omega	14
Figura 2.3.2.B	Interfaz programa Omega.....	15
Figura 2.3.4	Interfaz FeedDemon	16
Figura 2.4.2.A	Back-End de RSS_Sticker.....	18
Figura 2.4.2. B	Front-End de RSS_Sticker	19
Figura 2.4.3A	Back-End del módulo Feed	19
Figura 2.4.3B	Front-End del módulo Feed.....	20
Figura 2.4.4A	Back-End del modulo RSS Browser 2.7	21
Figura 2.4.4B	Front-End del modulo RSS Browser 2.7	22
Figura 2.4.5A	Back-End del modulo Slick RSS.....	23
Figura 2.4.5B	Front-End del modulo Slick RSS	23
Figura 3.1A	Front-End del módulo.....	33
Figura 3.1B	Front-End del módulo con opción de eliminar.....	34
Figura 3.3.1A	Back-End configuración del módulo.....	35
Figura 3.3.1B	Back-End de los filtros	36
Figura 3.3.1C	Back-End de la configuración por defecto	37
Figura 3.3.1D	Tooltip	38
Figura 4.1.A	Gráfica con la velocidad de los datos obtenidos	44
Figura 4.1.B	Gráfica con la velocidad de los datos obtenidos del filtro selectivo	45
Figura 4.2.1	Capturas de pantalla con las velocidades registradas (...)	46
Figura 4.2.2	Capturas de pantalla con las velocidades registradas (...)	47
Figura 4.2.3	Capturas de pantalla con las velocidades registradas (...)	48
Figura 4.2.4	Capturas de pantalla con las velocidades registradas (...)	48
Figura 4.2.5	Capturas de pantalla con las velocidades registradas (...)	49
Figura 4.3.A	Velocidad de tiempo de respuesta para los módulos Joomla! analizados.....	50
Figura 4.3.B	Tempo promedio de los módulos al variar el número de fuentes feed.....	51

ANEXO I

Figura AI, gene	Ejemplo con el módulo sin ejecutar	71
Figura AI.a30	Ejemplo del módulo con 30 palabras	71
Figura AI.a20	Ejemplo del módulo con 20 palabras	72

Figura AI.a10	Ejemplo del módulo con 10 palabras	72
Figura AI.a1	Ejemplo del módulo con 1 palabra	73
Figura AI.b30	Ejemplo del módulo con 30 palabras	73
Figura AI.b20	Ejemplo del módulo con 20 palabras	74
Figura AI.b10	Ejemplo del módulo con 10 palabras	74
Figura AI.b1	Ejemplo del módulo con 1 palabra	75
Figura AI.c30	Ejemplo del módulo con 30 palabras	75
Figura AI.c20	Ejemplo del módulo con 20 palabras	76
Figura AI.c10	Ejemplo del módulo con 10 palabras	76
Figura AI.c1	Ejemplo del módulo con 1 palabra	77
Figura AI.d30	Ejemplo del módulo con 30 palabras	77
Figura AI.d20	Ejemplo del módulo con 20 palabras	78
Figura AI.d10	Ejemplo del módulo con 10 palabras	78
Figura AI.d1	Ejemplo del módulo con 1 palabra	79
Figura AI,selec.	Ejemplo con el módulo sin ejecutar	79
Figura AI.Sa10	Ejemplo del módulo con 10 filtros	80
Figura AI.Sa7	Ejemplo del módulo con 7 filtros	80
Figura AI.Sa4	Ejemplo del módulo con 4 filtros	81
Figura AI.Sa1	Ejemplo del módulo con 1 filtros	81
Figura AI.Sb10	Ejemplo del módulo con 10 filtros	82
Figura AI.Sb7	Ejemplo del módulo con 7 filtros	82
Figura AI.Sb4	Ejemplo del módulo con 4 filtros	83
Figura AI.Sb1	Ejemplo del módulo con 1 filtros	83
Figura AI.Sc10	Ejemplo del módulo con 10 filtros	84
Figura AI.Sc7	Ejemplo del módulo con 7 filtros	84
Figura AI.Sc4	Ejemplo del módulo con 4 filtros	85
Figura AI.Sc1	Ejemplo del módulo con 1 filtros	85
Figura AI.Sd10	Ejemplo del módulo con 10 filtros	86
Figura AI.Sd7	Ejemplo del módulo con 7 filtros	86
Figura AI.Sd4	Ejemplo del módulo con 4 filtros	87
Figura AI.Sd1	Ejemplo del módulo con 1 filtros	87

ANEXO II

Figura AII.1	Módulo FEED.....	91
Figura AII.2	Módulo FEED.....	91
Figura AII.3	Módulo FEED.....	92
Figura AII.4	Módulo FEED.....	92
Figura AII.5	Módulo RX RSS Sticker.....	93
Figura AII.6	Módulo RX RSS Sticker.....	93
Figura AII.7	Módulo RX RSS Sticker.....	94
Figura AII.8	Módulo RX RSS Sticker.....	94
Figura AII.9	Módulo Slick RSS	95
Figura AII.10	Módulo Slick RSS	95
Figura AII.11	Módulo Slick RSS	96
Figura AII.12	Módulo Slick RSS	96
Figura AII.13	Módulo Browner 2.7	97
Figura AII.14	Módulo Browner 2.7	97
Figura AII.15	Módulo Browner 2.7	98
Figura AII.16	Módulo Browner 2.7	98

Figura AII.17	Módulo RSS_PROYECT	99
Figura AII.18	Módulo RSS_PROYECT	99
Figura AII.19	Módulo RSS_PROYECT	100
Figura AII.20	Módulo RSS_PROYECT	100

APENDICE I

Figura 2A	FOREACH	104
Figura 2B	STRISTR	105
Figura 3	Query	106
Figura 4	Print	106
Figura 5	XML	107
Figura 6.A	JFactory	108
Figura 6.B	JTable.....	108
Figura 6.C	require_once	108

ÍNDICE DE TABLAS

Tabla 4.1.A	Tabla filtro genérico	40
Tabla 4.1.B	Tabla filtro específico	41

Introducción

1. 1	Introducción.....	3
1. 2	Objetivos	4
1. 3	Medios Utilizados.....	6
1. 4	Organización de la memoria	7

1.1 Introducción

En la actualidad la nueva configuración del mercado de trabajo (que afecta en mayor medida a mujeres y jóvenes), la existencia de factores políticos y/o socio-culturales, unido a la ausencia o disminución de apoyos y vínculos socio-familiares, puede desembocar en procesos de exclusión social. J. Estivill define la exclusión social como una acumulación de procesos concluyentes con rupturas sucesivas que, arrancando del corazón de la economía, la política y la sociedad, van alejando e interiorizando a personas, grupos, comunidades y territorios con respecto a los centros de poder, los recursos y los valores dominantes.[1][2][3]

Uno de los principales factores de exclusión sigue siendo tener o no tener un empleo, y la posibilidad de acceder a un empleo va a estar cada vez más condicionada por la capacidad de utilizar eficazmente las tecnologías de la información y del conocimiento.

La ONU, en el Informe "Human Development Report, 2001", analizó las oportunidades que las nuevas tecnologías podían representar para un adecuado desarrollo humano de los países, y como podían ser utilizadas para aumentar algunos de los principales factores necesarios para lograr crecimiento económico y un mayor bienestar social (especialmente en sanidad y educación).

En la Declaración de Principios (United Nation, 2004), se expuso que si realmente se deseaba una Sociedad de la Información para todos los individuos era necesario fomentar una Sociedad de la Información que tuviera en cuenta a la persona y que se encauzara el potencial de estas tecnologías para promover las metas de desarrollo establecidas en la Declaración del Milenio (United Nation, 2000).

Hoy en día vivimos en la era de la información. Cada día nos bombardean con cientos de noticias y contamos con una serie de medios (internet, prensa escrita, televisión, radio...) que hacen inabarcable la idea de poder asimilar tantas noticias. Antes, debíamos esperar a que las noticias llegasen a nosotros seleccionadas por unas personas o entidades que decidían que mostrarnos, cuando y como.

Con la aparición de internet, todo esto cambio, o por lo menos él cuando informarnos. Las personas que navegaban por la red podían acceder al contenido de las web desde cualquier sitio y a cualquier hora. Todo esto supuso un gran logro, pero contrajo a la par unas exigencias que hasta entonces no se imaginaban. Una de las mayores deficiencias que surgieron, es que el usuario no sabía cuando la web cambiaba o actualizaba su contenido, lo que suponía que un usuario podía visitar varias veces en un día una web sólo para comprobar si el contenido había variado. Esto suponía una pérdida de tiempo y de aumento del tráfico de internet, todo ello innecesario.

A partir de aquí surgió la idea de syndicar las web para mantenerse actualizado sin necesidad de estar conectándose todo el rato, y para ello se utilizaban programas externos al navegador en su mayoría y se estableció un convenio y una serie de protocolos con los que garantizar su universalidad.

RSS son las siglas de Really Simple Syndication, un formato XML para syndicar o compartir contenido en la web. Se utiliza para difundir información actualizada frecuentemente a usuarios que se han suscrito a la fuente de contenidos. El formato permite distribuir contenidos sin necesidad de un navegador, utilizando un software diseñado para leer estos contenidos RSS.

A pesar de eso, es posible utilizar el mismo navegador para ver los contenidos RSS. Las últimas versiones de los principales navegadores permiten leer los RSS sin necesidad de

software adicional. RSS es parte de la familia de los formatos XML, desarrollado específicamente para todo tipo de sitios que se actualicen con frecuencia y por medio del cual se puede compartir la información y usarla en otros sitios web o programas. A esto se le conoce como redifusión web o sindicación web.

Joomla! es un sistema de gestión de archivos, y entre sus principales virtudes está la de permitir editar el contenido de un sitio web de manera sencilla. Es una aplicación de código abierto, programada mayoritariamente en PHP bajo una licencia GPL. Este administrador de contenidos puede trabajar en Internet o Intranets y requiere de una base de datos MySQL, así como, preferiblemente, de un servidor HTTP Apache.

En Joomla! se incluyen características como: mejorar el rendimiento web, versiones imprimibles de páginas, flash con noticias, blogs, foros, calendarios, búsqueda en el sitio web e internacionalización del lenguaje. Su nombre es una pronunciación fonética para anglófonos de la palabra swahili “jumla”, que significa "todos juntos" o "como un todo". Se escogió como una reflexión del compromiso del grupo de desarrolladores y la comunidad del proyecto [5].

En la actualidad, existen decenas de sindicadores web de contenidos, aunque aún no existe uno diseñado para la búsqueda de noticias de mujeres en riesgo de exclusión social ni posee las características y grado de configuración como el del programa sindicador RSS_PROYECT.

Cabe destacar la importancia de esta novedosa herramienta, ya que según el informe del Gobierno de España del Ministerio de Igualdad y Asuntos Sociales del año 2006 [4], existe una población del 3,6% de mujeres maltratadas y en potencial riesgo de exclusión social. Esto abre las puertas a un nuevo área dentro de las TIC, dando paso a herramientas que puedan favorecer la inserción en la sociedad de estas personas, alejándolas de la exclusión.

La herramienta es un buscador que puede seleccionar noticias muy específicas de un número elevado (e indeterminado) de fuentes; y estas mismas se pueden visualizar a través de su plataforma WEB. Esta herramienta es autónoma y proporciona una continua actualización de su contenido, para ser reflejo de la actualidad. Y a su vez, proporciona al administrador mecanismos para configurar su interfaz y el control total sobre el contenido publicado. [5][6]

RSS_PROYECT es una herramienta web basada JOOMLA!, un potente gestor de contenidos web muy utilizado e implantado en internet. Esta herramienta posee una interfaz sencilla, que pueda administrarse de una manera fácil y que no requiere de conocimientos de programación a nivel profesional. De esta forma se fomenta la autogestión por parte del usuario con el fin de alcanzar un mayor éxito y grado de satisfacción por parte del cliente o usuario.

Por último, cabe destacar que aunque esta herramienta este siendo utilizada por un cliente particular, la Asociación de mujeres maltratadas CIAM, está programada de un modo abierto, de forma que cualquier usuario JOOMLA! 1.5 puede utilizarla y configurarla de manera individual para satisfacer sus fines y objetivos específicos de web, no exclusivamente los colectivos en riesgo de exclusión social.

1.2 Objetivos

El principal objetivo de nuestro trabajo, es la investigación de herramientas sindicadoras de contenido, así como buscadores u otros elementos software,

dentro del área de exclusión social, relativas a la mujer; así como la posterior creación, desarrollo y evaluación de una herramienta que favoreciese al colectivo en riesgo de exclusión social.

Uno de los objetivos secundarios es la búsqueda e investigación del dominio de exclusión social, y la búsqueda de un 'hueco' donde enclavar nuestra investigación y poder aportar un trabajo útil y de relevancia.

Otro de los objetivos secundarios de este proyecto, es la evaluación técnica del programa RSS_PROYECT, una herramienta novedosa que ofrece una solución al colectivo de personas en riesgo de exclusión social, facilitando la búsqueda de ofertas de empleo, y a su vez, definir una aplicación abierta, muy versátil y configurable, para que la puedan usar millones de usuarios JOOMLA! 1.5.

RSS_PROYECT plantea una solución dentro del área de sindicación RSS novedosa y hasta donde alcanza mi conocimiento, no resuelta por ninguna aplicación de manera total (más adelante, estudiaremos estas herramientas y veremos que dan una solución limitada y sólo a partes del problema.)

Este software es utilizado actualmente por la asociación de mujeres maltratadas CIAM. La asociación CIAM necesitaba una herramienta que pudiese integrarse en su plataforma web (JOOMLA!) y que pudiese syndicar noticias. El problema residía en que el contenido de estas noticias debía referirse a noticias de mujeres maltratadas, lo que suponía un problema, ya que los programas existentes de sindicación permitían controlar ciertos parámetros como: número de noticias a visualizar, mostrar imágenes, y en los más avanzados la búsqueda de noticias que contuvieran una palabra. La diferencia de RSS_PROYECT con el resto, es que se programó 2 bloques de búsqueda:

-Un filtro genérico en el cual el usuario ponía una palabra y solo mostraba las noticias que contuvieran esa palabra (existían varios programas con esta característica).

-Un filtro específico en el que se pusieran varias palabras y la condición para mostrar esa noticia es que debía tener ese conjunto de palabras específico (no se encontró ningún programa que resolviese ese problema).

Además de dar solución a este problema, cabe destacar otra característica incorporada a partir de las necesidades de la Asociación CIAM, que es la siguiente: ¿Qué sucede cuando el programa por error publica una noticia que nosotros no queremos y que resulta perjudicial? ¿Qué se puede hacer? En los programas RSS existentes anteriores a RSS_PROYECT, el usuario tenía dos opciones: Desactivar el programa o modulo para que no se viera la noticia, o bien no hacer nada y dejar la noticia publicada en su web con el trastorno que suponía. Ceder el control de publicar una noticia o no a las maquinas podía convertirse en un problema, y no existía ninguna herramienta que nos permitiese controlar de manera tan específica el contenido de nuestra web. A raíz de esto, se desarrollo una característica para RSS_PROYECT, la cual puedes activar como administrador, y te permite mediante un botón eliminar las noticias indeseadas. Gracias a esto, el control sobre el contenido que se publicase en la web de un usuario, sería por primera vez plenamente controlado.

1.3 Métodos y medios utilizados

A continuación iremos desmenuzando todos los medios: hardware, software, documentación internet, libros... que han sido utilizados a la hora de realizar el proyecto.

En primer lugar, hemos utilizado la documentación de revistas y artículos de investigación a través de sus repositorios en internet, accesibles desde la Universidad de Valladolid. Alguna de las paginas utilizadas han sido: www.ieeexplore.ieee.org, www.acm.org, www.sciencedirect.com...

Através de los métodos de indagación aprendidos en las diversas asignaturas del MUTIC, hemos ido ‘tirando’ de los hilos para quedarnos con los artículos más relevantes para nuestra investigación.

En segundo lugar, hemos necesitado un ordenador portátil ACER 5720G, sobre el que hemos programado nuestra herramienta. El sistema operativo del mismo es el Windows Vista que venía configurado de fábrica. A partir de aquí, el primer programa que hemos necesitado para poder instalar JOOMLA! ha sido XAMPP [3].

XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl. El programa está liberado y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. XAMPP sólo requiere descargar y ejecutar el archivo comprimido, con unas pequeñas configuraciones podremos usar sus componentes.

Gracias a este programa, creamos un servidor virtual en nuestra maquina y podemos programar, visualizar la web, etc. Sin necesidad de tener un dominio host, o conexión a internet. Así podemos trabajar en cualquier sitio, ya que todas las funciones y operaciones aritméticas se ejecutan en nuestro ordenador.

El siguiente software necesario es la versión de JOOMLA! v1.5. Descargada de la página oficial. JOOMLA! es un CMS (sistema gestor de contenidos). Lo siguiente que hacemos es instalarlo a través del programa XAMPP en el fichero “htdocs”, que en un servidor equivaldría a la raíz. Una vez instalado accedemos al Back-End, donde nos vamos familiarizando con el entorno.

Una vez verificada la instalación de JOOMLA! descargamos el programa ECLIPSE versión Helios. Este programa nos servirá para programar en PHP, HTML y XML nuestra aplicación. ECLIPSE es una herramienta muy útil a la hora de programar, ya que compila el código y es capaz de mostrarte los fallos de código que puedes tener a la hora de programar.

Por último, tendríamos que hablar de los diferentes métodos didácticos utilizados, que se podrían resumir en dos: 1º Toda la documentación extraída de internet de las paginas oficiales de JOOMLA!, PHP, HTML, etc. Y 2º la

documentación extraída de los libros de programación de la biblioteca pública de Palencia.

Todo ello haremos referencia en el ANEXO I y bibliografía.

1.4 Organización de la memoria

A la hora de realizar el Trabajo Fin de Master, hemos seguido unas pautas que vamos a relatar a través de esta memoria.

Capítulo 2: estado del arte de los sindicadores de contenido dentro y fuera de JOOMLA! en este capítulo se desarrolla un resumen del estado del arte de los sindicadores RSS, ya sean dentro de JOOMLA! a través de módulos, componentes o plugins. Y fuera de JOOMLA!, a través de programas independientes, como sindicación y procesan las noticias.

Capítulo 3: Módulo RSS PROYECT. En este capítulo explicaremos cuales fueron los motivos para elegir un módulo y no otro tipo de extensiones. También explicaremos las características del módulo, tanto administrativas como técnicas. Hablaremos de sus compatibilidades, velocidad de respuesta, todas sus funciones...

Capítulo 4: Métodos de evaluación. Explicaremos en detalle cuales han sido los métodos de evaluación de la herramienta RSS_PROYECT y justificaremos su uso.

Capítulo 5: Discusión de los resultados, conclusiones y líneas futuras. Detallaremos las conclusiones del programa, así como unos ejemplos de la repercusión y utilidades que tiene. Además propondremos líneas futuras de trabajo sobre el módulo, y nuevas mejoras e ideas con el que aumentar su valor.

Bibliografía: Aquí he escrito las fuentes más importantes o más recursivas que he utilizado. Como es lógico, no están todas ya que no todas han sido relevantes, y si tuviésemos que poner todas las Webs visitadas en 8 meses, tendríamos un libro de bibliografía.

Anexo I: Testeo del módulo. Aquí aparecen las capturas de pantalla de donde hemos tomado los datos, para demostrar que son datos reales obtenidos con el programa FIREBUG.

Anexo II: Testeo de otras herramientas con las que comparamos el módulo RSS_PROYECT.

Apéndice I: Código implementado. Pequeño resumen para que el lector tome una ligera idea de los códigos que se han utilizado para programar la herramienta [7][8].

Publicaciones I: ÁLVARO MUÑOZ, Saúl, et al. Development and Performance Evaluation of a New RSS Tool for a Web-Based System: RSS_PROYECT. *Journal of Network and Computer Applications*, no. 0. ISSN 1084-8045

Introducción a los sindicadores de noticias RSS

2.1	Introducción a los sindicadores de noticias RSS	11
2.2	Conceptos básicos.....	11
2.2.1	RSS.....	11
2.2.2	¿Qué es Joomla!?	12
2.3	Sindicadores fuera de Joomla!.....	13
2.3.1	Google reader	13
2.3.2	Omega reader	13
2.3.3	Netvibes	14
2.3.4	FeedDemon.....	15
2.4	Sindicadores RSS dentro de Joomla	16
2.4.1	Introducción	16
2.4.2	Módulo RX RSS Sticker	17
2.4.3	Feed.....	18
2.4.4	RSS Browser 2.7	19
2.4.5	Slick RSS.....	22
2.4.6	Conclusiones	22
2.5	¿Por qué Joomla!?	23

2.1 Introducción a los sindicadores de noticias

Hoy en día vivimos en la era de la información. Cada día nos bombardean con cientos de noticias y contamos con una serie de medios (internet, prensa escrita, televisión, radio...) que hacen inabarcable la idea de poder asimilar tantas noticias. Antes, debíamos esperar a que las noticias llegasen a nosotros seleccionadas por unas personas o entidades que decidían que mostrarnos, cuando y como.

Con la aparición de internet, todo esto cambio, o por lo menos él cuando informarnos. Las personas que navegaban por la red podían acceder al contenido de las web desde cualquier sitio y a cualquier hora. Todo esto supuso un gran logro, pero contrajo a la par unas exigencias que hasta entonces no se imaginaban. Una de las mayores deficiencias que surgieron, es que el usuario no sabía cuando la web cambiaba o actualizaba su contenido, lo que suponía que un usuario podía visitar varias veces en un día una web sólo para comprobar si el contenido había variado. Esto suponía una pérdida de tiempo y de aumento del tráfico de internet, todo ello innecesario.

A partir de aquí surgió la idea de syndicar las web para mantenerse actualizado sin necesidad de estar conectándose todo el rato, y para ello se utilizaban programas externos al navegador en su mayoría y se estableció un convenio y una serie de protocolos con los que garantizar su universalidad.

2.2 Conceptos básicos

2.2.1 RSS

RSS son las siglas de Really Simple Syndication, un formato XML para syndicar o compartir contenido en la web.[10]



Figura 2.2.1: Botón RSS.

Se utiliza para difundir información actualizada frecuentemente a usuarios que se han suscrito a la fuente de contenidos. El formato permite distribuir contenidos sin necesidad de un navegador, utilizando un software diseñado para leer estos contenidos RSS.

A pesar de eso, es posible utilizar el mismo navegador para ver los contenidos RSS. Las últimas versiones de los principales navegadores permiten leer los RSS sin necesidad de software adicional. RSS es parte de la familia de los formatos XML, desarrollado específicamente para todo tipo de sitios que se actualicen con frecuencia y por medio del cual se puede compartir la información y usarla en otros sitios web o programas. A esto se le conoce como redifusión web o sindicación web [6].

2.2.2 ¿Qué es JOOMLA!?

Joomla! es un sistema de gestión de archivos, y entre sus principales virtudes está la de permitir editar el contenido de un sitio web de manera sencilla. Es una aplicación de código abierto, programada mayoritariamente en PHP bajo una licencia GPL. Este administrador de contenidos puede trabajar en Internet o Intranets y requiere de una base de datos MySQL, así como, preferiblemente, de un servidor HTTP Apache.

En Joomla! se incluyen características como: mejorar el rendimiento web, versiones imprimibles de páginas, flash con noticias, blogs, foros, calendarios, búsqueda en el sitio web e internacionalización del lenguaje. Su nombre es una pronunciación fonética para anglófonos de la palabra swahili "jumla", que significa "todos juntos" o "como un todo". Se escogió como una reflexión del compromiso del grupo de desarrolladores y la comunidad del proyecto.[11]



Figura 2.2.2: Logo Joomla!

2.3 Sindicadores fuera de JOOMLA!

2.3.1 Google Reader

Google Reader es un lector de RSS y atom. Permite organizar y acceder rápidamente desde una interfaz Web a todas las noticias de las páginas configuradas en el sistema que soporten. Fue lanzado por Google el 7 de octubre de 2005 mediante Google Labs, Google Reader se incorporó a Google Labs el 17 de septiembre de 2007.

La interfaz es muy parecida a la bandeja de entrada de un lector de correos electrónico normal, a la izquierda se pueden visualizar las suscripciones a las cuales el usuario alta y a la derecha la lista de noticias nuevas, que puede ser vista de diversas formas.



Figura 2.3.1: Logo Google Reader

A la hora de compartir las noticias y elementos nuevos de interés para enviar por parte del usuario pueden ser compartidas de diversas formas, una opción es seleccionar enviar por correo, esta opción utiliza la cuenta de Gmail para enviar los elementos seleccionados, la siguiente opción es un espacio llamado Elementos Compartidos, aquí al igual que los elementos destacados pueden ser seleccionados y enviados a una página única por usuario estilo Blog, en donde se actualizan los elementos seleccionados.

En la organización, existe una opción para destacar algún elemento deseado, mediante la selección de la estrella a la izquierda del elemento, estas noticias de las suscripciones se guardan para futuras revisiones [12].

2.3.2 Omea Reader

Un lector de feeds RSS es agradable a la vista. Si no te gusta hacer juegos malabares múltiples aplicaciones, sitios y servicios, Omea Reader es una gran herramienta para la integración de elementos a leer.



Figura 2.3.2.A: Omega

Omea Reader es una perfecta mezcla de los canales RSS, páginas favoritas de Internet (que se puede leer en línea) y grupos de noticias Usenet, aunque

desafortunadamente, el correo electrónico y mensajería instantánea no se encuentran en Omea Reader.

La interfaz da una imagen muy personal y muy eficiente. No sólo se puede buscar elementos rápidamente en Omea Reader, también se puede guardar las búsquedas como carpetas o crear "las vistas" de una manera flexible.

Las banderas, anotaciones y categorías permiten organizar los elementos con más noticias y sin apenas un límite. Los filtros pueden hacer la mayoría de estas tareas de forma automática para usted - siempre y cuando usted encuentra los criterios adecuados a la hora de filtrar.

Los contras de este programa es que no ofrece las funciones de sincronización por sí mismo. La multitud de otras características, carpetas, filtros, puntos de acceso, puntos de vista y espacios de trabajo puede hacer de la interfaz de Omea Reader una pantalla muy saturada tan bien.

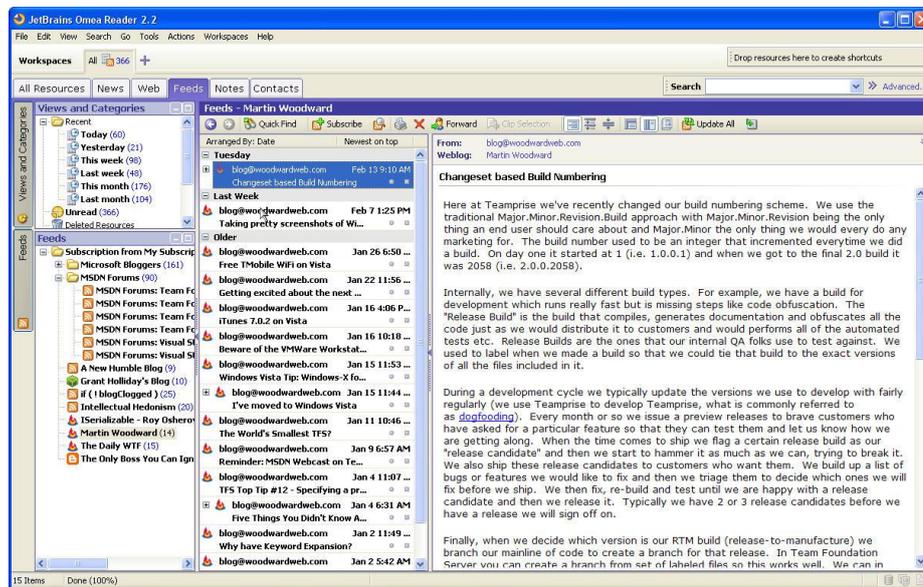


Figura 2.3.2.B: Interfaz del programa Omega.

2.3.3 Netvibes

Netvibes es un servicio web que actúa a modo de escritorio virtual señalizado, similar a la página principal personalizada de Google.



Figura 2.3.3: Netvibes

Visualmente está organizada en solapas o pestañas, donde cada solapa por lo general es en sí un agregado de diversos módulos y widget desplazables previamente definidos por el usuario. Estos módulos, a su

vez, actúan como pequeñas ventanas cuyo contenido es generado por otro servicio web o ser mini aplicaciones.

Algunos ejemplos de los servicios que puede contener son multi-buscadores para varios servidores (Google, Yahoo!etc), webmails tipo Gmail, fotos y videos de YouTube, Flickr y muchos servicios más, de los cuales algunos son verdaderos gadgets, como una pecera con peces virtuales o mini juegos como sudoku.

Hay módulos que permiten desplegar el contenido generado por otras páginas que funcionan como fuentes web de RSS y Atom; por ejemplo, los diarios y los blogs [13].

También existen pestañas que no son agregadores y que permiten acceder a un servicio completo de otra página, como Joomla!

Desde el punto de vista comercial, están los llamados Universos, que son páginas creadas principalmente por empresas o grupos musicales y en que se muestran diversas fuentes web, imágenes y otros materiales relacionados con el creador. En el momento de escribir esto, esta característica no está disponible para los usuarios independientes, aunque posiblemente se empiece a distribuir mediante invitaciones [14].

2.3.4 FeedDemon

Como otros lectores de RSS, permite suscribirnos a un número indeterminado de Feeds. Con ello conseguimos estar al día sobre las novedades que publican diversas fuentes de información que ofrecen contenidos en la web, las que seleccionamos.

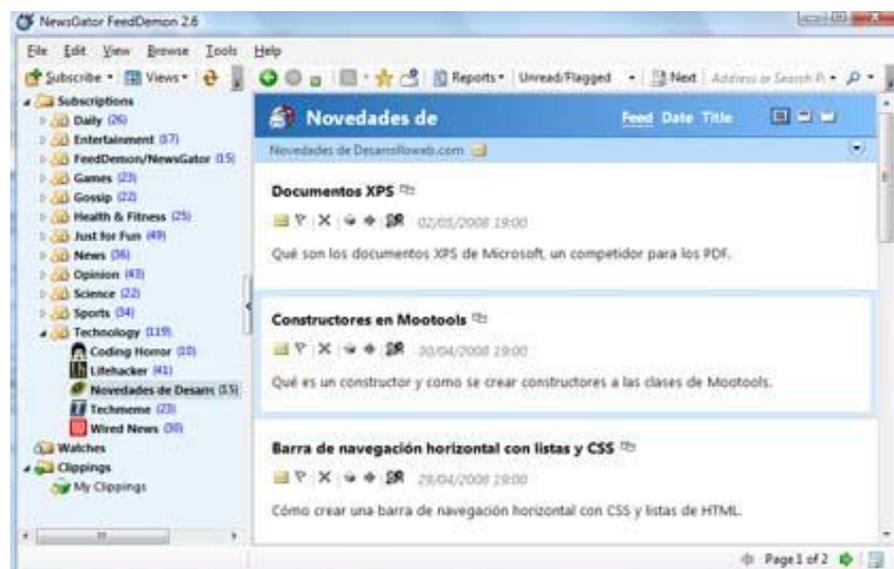


Figura2.3.4: FeedDemon interfaz.

Entonces, todas estas fuentes de información quedan centralizadas en FeedDemon, un único programa donde podemos entrar y ver de una sola vez todas las novedades de nuestros sitios preferidos.

FeedDemon ofrece diversas utilidades para poder administrar las fuentes de información RSS (feeds), como clasificación en varias categorías, avisos cuando surgen novedades en los sitios suscritos, diferentes vistas de lectura de los feed RSS, opciones particulares para cada feed suscrito, visualización de los posts en el propio programa, conexión con sitios de envío o votación de posts como del.icio.us o Digg, etc.

Todas estas funciones de NewsGator podrían estar disponibles en otros lectores RSS, pero tiene otra funcionalidad muy original y que puede ser bastante útil. Se trata de un sistema de sincronización con el lector de RSS News Gator Online, que es una aplicación web para administrar tus RSS, pero que no requiere ningún software, sólo tu navegador web. A través de NewsGator Online puedes administrar y mantener tu lista de RSS favoritos desde cualquier ordenador conectado a Internet. Así, si te vas de viaje y tienes que utilizar otros ordenadores, podrás tener en ellos tu lista de RSS de interés. Tanto la aplicación de escritorio como la aplicación web comparten tus configuraciones y estados de Feeds RSS [15].

En definitiva, se trata de un lector RSS muy potente y altamente personalizable. Gratuito y muy recomendable.

2.4 Sindicación RSS dentro de Joomla!

2.4.1 Introducción

En el siguiente apartado mostraremos algunos de los módulos y componentes estudiados a la hora de realizar el PFC antes de crear nuestro módulo. Realizaremos una presentación de las herramientas oficiales y gratuitas encontradas en la página oficial de Joomla! y de libre acceso.

Dado que el número de módulos de este tipo encontrados en la página oficial son cientos, seleccionaremos los 5 más importantes encontrados [16].

2.4.2 Módulo RX RSS Sticker

En este módulo solo se pueden suscribir a una noticia. Te permite la configuración a la hora de ver el número de noticias. Posee opciones como controlar la velocidad de transición en el Front-End y a la hora de colocar el ratón encima de la noticia. Una vez colocado el ratón la noticia es detenida y podemos leer una pequeña descripción de la misma.

En la figura anterior podemos observar la parte del Back-End de este módulo, con sus correspondientes cuadros de configuración.

The screenshot displays the configuration interface for the RSS Sticker module. At the top, there is a navigation bar with a 'Previsualizar' button, a counter showing '0' and '1', and a 'Cerrar sesi' button. Below this is a toolbar with icons for 'Guardar', 'Aplicar', 'Cancelar', and 'Ayuda'. The main configuration area is titled 'Parámetros' and contains a 'Parámetros del módulo' section with the following settings:

- Mootools: Habilitado Deshabilitado
- Speed: Slow Medium Fast Very fast
- Speed IE: Slow Medium Fast Very fast
- Direction: Right to left Left to right
- RSS Link:
- Count:
- Content separator:

Below the main configuration area, there are sections for 'Parámetros avanzados' and 'Otros parámetros'.

Figura 2.4.2.A: Back-End de RSS_Sticker.

Aparentemente es un módulo sencillo en el que sus fortalezas podríamos destacar el exponer de una forma llamativa las noticias en nuestra web, con un cierto control en su configuración. Sin embargo, en contra, podemos exponer que sólo permite una fuente, y no se pueden ver sus imágenes, ni su descripción, ni controlar el número de palabras a ver, etc. Es un módulo más visual que funcional realmente. En la figurar a de abajo se puede apreciar una captura de pantalla en el que claramente se puede apreciar que no se distingue el número de noticias, ya sean 2 o 22 (puesto que se ven una detrás de otra, y se pierde este sentido). A continuación podemos ver la respectiva captura de pantalla.



Figura 2.4.2.B: Front-End de RSS_Sticker.

2.4.3 Feed

El siguiente módulo, es el que se instala por defecto en el paquete Joomla! versión 1.5. Sus principales características desde el Back-End son: permitir ver una descripción de la noticia, incluyendo su imagen. En contra, no puedes seleccionar ver solo la imagen del artículo, sino que también la de la feed, produciendo en muchos casos una mala visualización poco atractiva y profesional.

A continuación vemos la configuración del Back-End:

Parámetros

▼ **Parámetros del módulo**

Sufijo de la clase del módulo	<input type="text"/>
URL del Feed	<input type="text" value="http://www.rtve.es/rss/temas_noticias.xml"/>
RTL Feed	<input checked="" type="radio"/> Sí <input type="radio"/> No
Título del Feed	<input checked="" type="radio"/> Sí <input type="radio"/> No
Descripción Feed	<input checked="" type="radio"/> Sí <input type="radio"/> No
Imagen del Feed	<input checked="" type="radio"/> Sí <input type="radio"/> No
Artículos	<input type="text" value="3"/>
Descripción del artículo	<input checked="" type="radio"/> Sí <input type="radio"/> No
Contador general	<input type="text" value="0"/>

► **Parámetros avanzados**

Figura 2.4.3.A: Back-End del modulo Feed.

Otro de los contras de este módulo es que sólo permite una fuente feed, en vez de varias, y también que no se puede controlar el contenido con el número de palabras a visualizar, pudiéndose dar un titular muy largo, desbordándose de la pantalla, con el resulta nada agradable que a continuación encontramos.

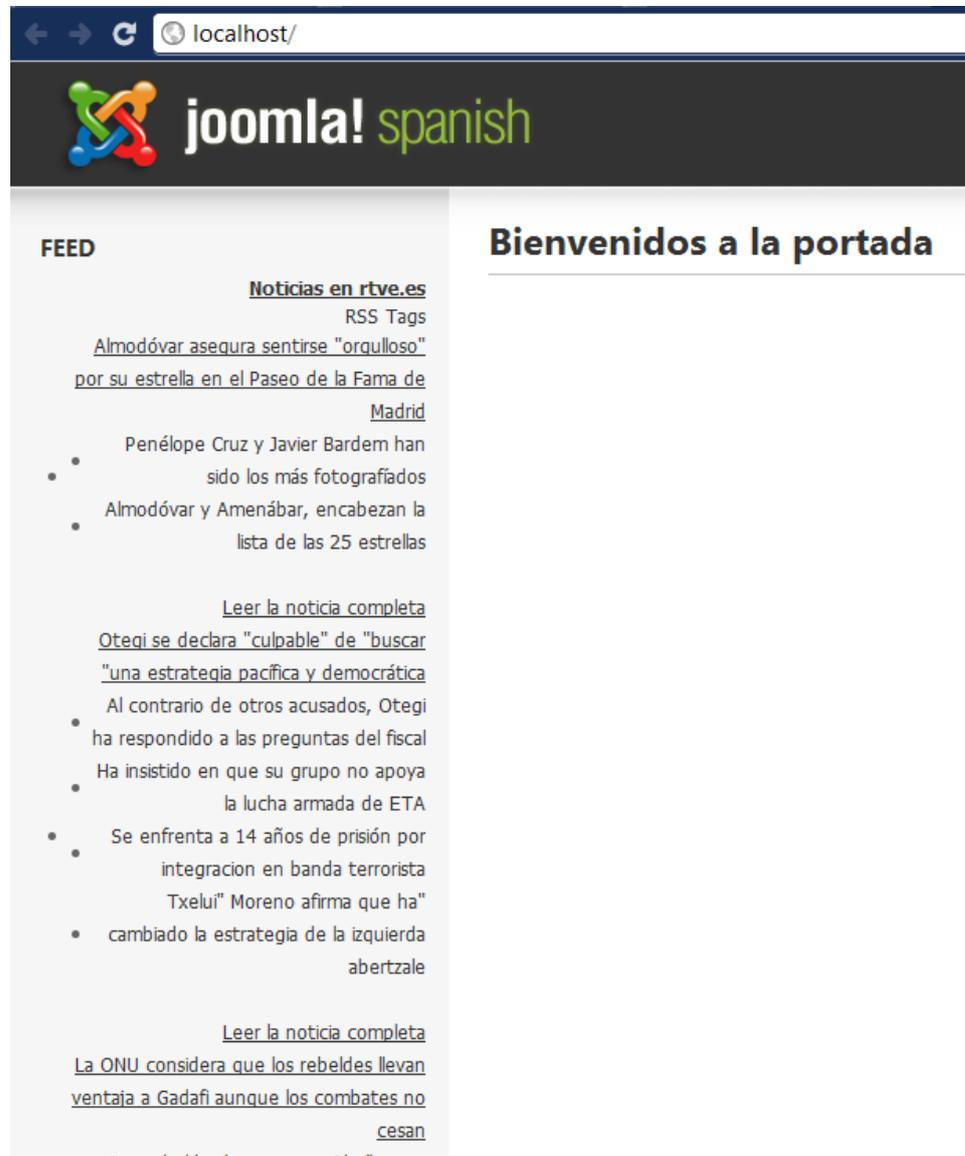


Figura 2.4.3.B: Front-End del Feed.

2.4.4 RSS Browser 2.7

Este módulo permite por primera vez de los que hemos visto, poder syndicar varias noticias. Es decir, poder suscribirte a varias fuentes de noticias. Entre sus ventajas y fortalezas se encuentran esta característica, además de poder controlar el número de artículos visualizados, como viene siendo normal en los anteriores. Pero sobretodo, este módulo añade

una característica importante, y es que se puede visualizar la información de la feed (no la del artículo), con lo que esto supone. Añade otra característica, y es que al posar el ratón encima de él la información de la noticia aumenta, mostrándonos una descripción del artículo. En la figura inferior se muestra nuestro Back-End.

Parámetros

▼ **Parámetros del módulo**

Sufijo de la clase del módulo	<input type="text"/>
Module Cache	<input checked="" type="radio"/> No <input type="radio"/> Sí
RSS Cache time (min.)	<input type="text" value="60"/>
RSS URLs	<input type="text" value="http://www.rtve.es/rss/temas_noticias.xml"/>
Feed Title	<input checked="" type="radio"/> Sí <input type="radio"/> No
Feed Description	<input checked="" type="radio"/> Sí <input type="radio"/> No
Imagen de la noticia(Feed)	<input checked="" type="radio"/> Sí <input type="radio"/> No
Items	<input type="text" value="5"/>
Char Count Title	<input type="text" value="0"/>
News Box Width	<input type="text" value="850"/>
News Box Height	<input type="text" value="650"/>
Item Description	<input type="radio"/> Sí <input checked="" type="radio"/> No
Enable Tooltip	<input checked="" type="radio"/> Sí <input type="radio"/> No
Enable Description in Tooltip	<input checked="" type="radio"/> Sí <input type="radio"/> No
Word Count Description	<input type="text" value="25"/>

Figura 2.4.4.A: Back_End del RSS Browser 2.7

A continuación vemos el resultado del Front-End, con las ventajas que hemos descrito, y su correcta visualización.

Podemos observar, con respecto a lo anterior, que los artículos se muestran correctamente, viéndose solo la información del Feed y no la del ítem.

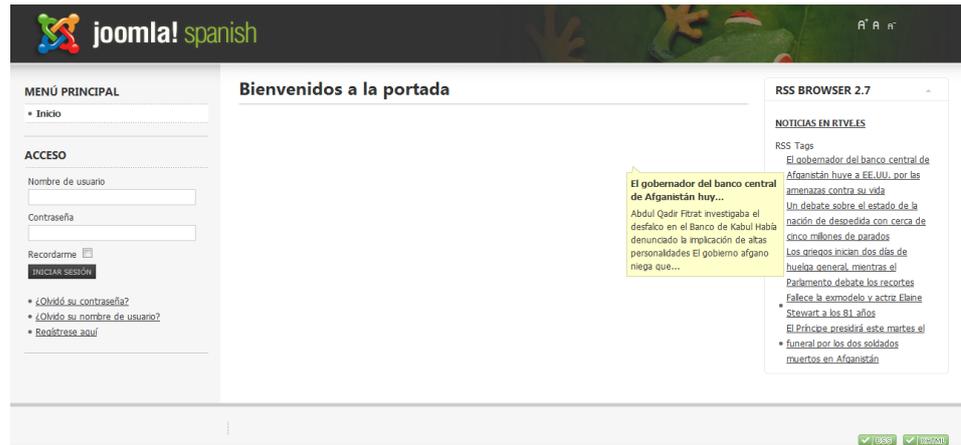


Figura 2.4.4.B: Front-End del RSS Browser 2.7.

2.4.5 Slick RSS

Este es uno de los módulos más completos de todos los encontrados. Entre sus principales ventajas y características:

- Control de las imágenes y contenido a visualizar, tanto feed, como del ítem.
- Control del número de palabras a visualizar del feed y del ítem.
- Permite abrir en una página nueva o en la misma.
- Configuración del tiempo de vida de la cache.
- Atributos no-follow.
- Habilitar Tooltip.

A pesar de todo esto, una de las principales desventajas de este módulo es que solo permite syndicate una fuente cada vez, por lo que al tener varias fuentes, es necesario syndicarlas una por una, construyendo un módulo cada vez.

A continuación mostraremos el Back-End donde aparecen todas las configuraciones descritas anteriormente.

Parámetros

▼ **Parámetros del módulo**

Sufijo de la clase del módulo	<input type="text"/>
RSS Cache time	<input type="text" value="3600"/>
RSS URL	<input type="text" value="http://www.rtve.es/rss/temas_noticias.xml"/>
Feed Title	<input type="radio"/> Sí <input checked="" type="radio"/> No
Feed Description	<input type="radio"/> Sí <input checked="" type="radio"/> No
Imagen de la noticia(Feed)	<input type="radio"/> Sí <input checked="" type="radio"/> No
Number of Items	<input type="text" value="5"/>
Item Title Word Count	<input type="text" value="15"/>
Item Description Text	<input type="radio"/> Sí <input checked="" type="radio"/> No
Item Description Word Count	<input type="text" value="25"/>
Item Description Text Images	<input type="radio"/> Sí <input checked="" type="radio"/> No
RSS Feed Link Target	<input checked="" type="radio"/> New Window <input type="radio"/> Same Window
Add rel='nofollow' to links	<input checked="" type="radio"/> No <input type="radio"/> Sí
Enable Tooltip	<input checked="" type="radio"/> Sí <input type="radio"/> No
Tooltip Title Word Count	<input type="text" value="25"/>
Tooltip Description Word Count	<input type="text" value="25"/>
Tooltip Description Text Images	<input type="radio"/> Sí <input checked="" type="radio"/> No

Figura 2.4.5.A: Back-End del Slick RSS.

Con su correspondiente visualización del Front-End en la imagen inferior.

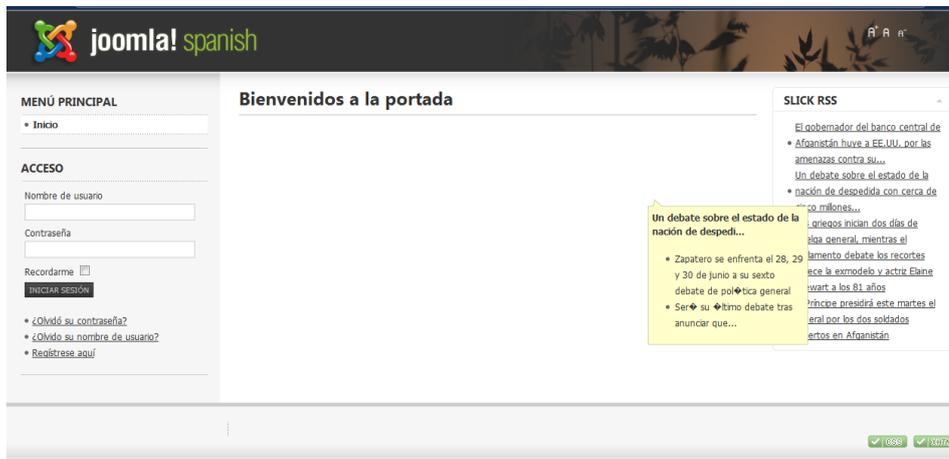


Figura 2.4.5.B: Front-End del Slick RSS.

2.4. 6 Conclusiones

En relación a los sindicadores dentro de Joomla!, RX RSS Ticker utiliza los métodos JFactory de Joomla!, la velocidad de procesamiento es mayor con respecto al modulo FEED que viene por defecto, debido a que el módulo de programación que ejecuta (RX RSS Ticker), implementa una serie de funcionalidades añadidas que requieren de un consumo de recursos más elevado que el módulo FEED. A favor, esta las opciones de visualización que ofrece y el grado de configuración mucho más completo y complejo que el modulo FEED por defecto. FEED se encuentra en el paquete por defecto de Joomla! 1.5. Este módulo utiliza las librerías de la API de Joomla! (JFactory::XML), esto permite un funcionamiento muy similar en cuanto a rapidez, en comparación con el uso de las librerías SimplePie.

Aunque por otro lado produce más fallos a la hora de syndicar noticias, en comparación con el uso de la librería SimplePie, ya que su nivel de desarrollo es más básico que esta, y no es capaz de reconocer muchos de las versiones ATOM que se utilizan en la actualidad para syndicar contenidos. RSS Browser 2.7 se basa en la librería SimplePie para obtener la información RSS de las noticias.

Esta librería aporta entre sus principales ventajas al módulo: rapidez similar de procesado que usando las API de Joomla! (debido a su optimización), mejor extracción de las fuentes RSS y ATOM (soporta hasta 8 protocolos, más que el paquete de Joomla! por defecto) y una mayor compatibilidad, ya que se puede integrar en más plataformas, aparte de Joomla! RSS Browser 2.7, permite poder syndicar varias fuentes. Por el contrario, el código de este módulo está programado en bucles anidados que se ejecutan siempre (a pesar de no ser requeridos desde la configuración back-end), por lo que su procesado es más lento en comparación con las otras herramientas estudiadas, independientemente de la forma en que se configure.

Slick RSS utiliza la librería SimplePie, con las ventajas aportadas anteriormente. Cabe destacar que esta librería está incorporada ya en la propia plataforma Joomla! por defecto desde la versión 1.5, lo que es un indicador sobre su utilidad y eficacia en el proceso de búsqueda de noticias. Otra ventaja es que está basada en PHP, por lo que su programación y lectura de código la hace más sencilla para los programadores en PHP. Utiliza las funciones de extracción de imágenes, títulos, etc. Por el contrario, en la estructuración del código, ejecuta diversas funciones y algoritmos a pesar de que luego no son utilizados, por lo que ralentiza su funcionamiento. Una mejora de la implementación del software la podría hacer mucho más eficiente, aumentando la velocidad del procesado.

Por otra parte, el módulo RSS_PROYECT sólo puede ejecutarse dentro de Joomla! 1.5, por lo que la comparación con el funcionamiento interno

de otros buscadores fuera de Joomla! no es posible, ya que no partiríamos de unos mismos criterios para poder realizar esta comparación.

2.5 ¿Por qué Joomla!?

El hecho de emplear Joomla!, fue debido a que este CMS te permite crear sitios web de una forma fácil, profesional y sin costo alguno. Joomla! es uno de los más populares Open Software disponibles. El código es libre de uso y disponible a todo el mundo.

Podemos afirmar que Joomla! es una evolución de Mambo. Después una discusión con la empresa propietaria por los derechos de propiedad, un grupo de programadores se separaron y quisieron seguir trabajando en el proyecto de manera libre.

¿Por qué Joomla! es tan popular?

Una de las razones principales es la propia esencia de este software: “free software”. Puede que estas palabras al traducir al español resultan un tanto ambiguas. Algunos la traducen como programa gratis, cuando el verdadero significado es programa libre.

Joomla! no tiene coste (de licencia), lo podemos bajar y usar sin tener que pagar nada. Pero lo más importante: podemos acceder al código libre, modificarlo, mejorarlo o adaptarlo a sus necesidades.

Se ha calculado que Joomla! vale más de 2 millones de dólares. Algunos desarrolladores ven en el software libre como Joomla una amenaza. Lo “gratis” para muchos equivale a “malo”. Nada más lejos de la realidad. De hecho, el costo de Joomla! está valorado en USD\$ 2, 194,486, [11] teniendo en cuenta que si una empresa decidiera llegar a un software similar, debería dedicar 11 programadores durante 18 meses continuos para alcanzar las 65.880 líneas de código que Joomla tiene escrito.

Es la gran fortuna del software libre: muchos de los mejores desarrolladores del planeta trabajando al tiempo, desde diferentes partes del mundo y aportando su conocimiento para construir entre todos el mejor software CMS del mundo.

Otro de los incentivos es que Joomla! ha obtenido diversos premios internacionales. Así fue considerado en el CMS Award de los dos últimos años (2006 y 2007): el mejor CMS de código abierto del mundo. Durante dos años consecutivos este prestigioso premio ha sido otorgado a Joomla, por encima de otros también importantes y grandes, como Drupal o PHP Nuke.

Otra razón de peso para utilizar Joomla! es su prestigio [11]. Las Naciones Unidas, la universidad de Harvard o la página oficial de Linux

utilizan este programa. Son miles de sitios los que hoy en día usan Joomla, sin embargo, es importante resaltar algunos casos exitosos, como el de las Naciones Unidas, que prefirió usar Joomla a una solución de administración de contenidos de Microsoft, lo que produjo un golpe de opinión en todo el mundo significativo, ya que organismo como Naciones Unidas base toda la plataforma multilingüe a software libre era un gran paso para la plataforma y otorgaba mucho prestigio.

Dentro de las principales ventajas de usar Joomla, encontramos:

- *Software libre:* usted es libre de usarlo, no necesita pagar costos de licenciamiento.
- *Posibilidad de modificar el código fuente:* los programadores o diseñadores podemos acceder a lo más íntimo del código y modificarlo según las necesidades de la organización.
- *Más de tres mil extensiones:* la gran mayoría de libre uso, que permiten ampliar las posibilidades y características de Joomla. Si el núcleo de Joomla no supe alguna necesidad específica, podemos buscar una extensión...
- *Instalación en servidores Linux, Mac y Windows:* no hay excusa, si usted tiene su propio servidor en Windows (pasa mucho en empresas colombianas), podrá instalarlo sobre Apache.
- *Velocidad de carga:* a diferencia de otras plataformas, Joomla permite una carga muy rápida de sus páginas gracias al sistema de caché.
- *Cumplimiento de estándares web:* la más reciente versión de Joomla se acerca al ideal de cumplimiento de los estándares del W3C. Gracias a su sistema de plantillas es posible separar la presentación del contenido y marcar semánticamente los documentos.
- *Facilidad en la creación de módulos personalizados:* gracias al código abierto.
- *Seguridad:* gracias a una comunidad de miles de usuarios es posible tener una vigilancia exhaustiva de posibles vulnerabilidades en el código. Y cuando aparece una, la solución es inmediata por parte de toda la comunidad.

La verdad, es que la seguridad dependerá del administrador de la página. Debe estar muy atento a las actualizaciones y parches que salgan (normalmente cada dos o tres meses). Pero si hay descuido por parte del Webmáster y nunca se actualiza, las vulnerabilidades estarán presentes y la puerta abierta para que la página sea atacada .

3

Módulo RSS Project

3.1 Introducción	29
3.2 Justificación	29
3.3 Características técnicas	30
3.3.1 Configurabilidad	30
3.3.2 Compatibilidad	34

3.1 Introducción

Nuestro modulo RSS_Proyect se le puede definir principalmente como un módulo capaz de suscribirte a un número ilimitado de fuentes feed, de las cuales puedes extraer todos los artículos y cotejarlos con una serie de filtros y una lista negra almacenada en nuestra base de datos que nos permite visualizarlos o no, según cumpla los criterios marcados por el administrador. A parte, podemos configurar un tipo de usuarios en nuestro modulo, para que les de permisos de eliminación de contenido; esto quiere decir, que a un usuario registrado le aparecerá unas pestañas con las que podrá eliminar una serie de artículos que no le interesan. De esta forma el control de la información que se tiene es total.

En la imagen inferior vemos a través de dos capturas de pantalla las distintas opciones y visualizaciones dependiendo si es un usuario registrado o un usuario huésped.

En la primera, vista como usuario, muestra todas las noticias que contengan la palabra 'los', indistintamente en mayúsculas que minúsculas, además de comprobar primeramente que no se encuentre la noticia en una lista negra.



Figura 3.1.A: Front-End del modulo.

En la figura inferior, se muestra la interfaz que se vería de un usuario registrado, como tienen a facultad de poder eliminar una noticia de los publicados, enviándola a la lista negra de la base de datos. De esta manera el control del administrador sobre el contenido de su portal web es total.



Figura 3.1.B: Front-End del módulo con la opción de eliminar.

En nuestro proyecto en concreto, la configuración viene especificada por nuestro cliente, la asociación CIAM de Valladolid, que la usará para incorporar a su página web noticias sobre trabajo y mujer, así como otros temas de interés para la asociación [17].

3.2 Justificación

Al principio, nuestro primer planteamiento fue desarrollar un componente con el modelo MVC, ya que conseguimos a través de internet encontrar algunas guías sobre como programar un componente para Joomla!

Realmente, una de las cosas más difíciles de este proyecto ha sido la hora de documentarse sobre el Framework de Joomla! (la información en español es mínima, y en algunos casos, descubrimos más adelante que no era del todo correcta ni fiable), como programarlo, sus estructuras, protocolos y convenio; por eso, a la hora de encontrar esta documentación nos dispusimos a estudiarla para programar en ella. Dados estos pasos, con un ligera idea por donde abordar los problemas, descubrimos en la API de Joomla! de la página oficial, que realmente existía una extensión que cumplía con nuestras necesidades y con una estructura relativamente sencilla: el modulo. [18-24]

A diferencia del componente, el modulo era más básico a la hora de estructurarse, lo que permitía una mejor lectura y seguimiento de las rutinas que se iban desarrollando en el programa, y por lo tanto, más sencillo de programar, ya que cada paso lo podéis invocar desde el programa ECLIPSE utilizado para programar, y te permitía ver los fallos, redundancia de procesos. En definitiva, mucho más efectivo.

Por otra parte surgió la pregunta de ¿cómo de largo se va a hacer el programa?, ya que una vez empezado no sabíamos las ramificaciones que podía tomar, y eso era algo que debíamos tener en cuenta, ya que sabíamos dónde iba a empezar el programa, pero no lo lejos que iba a llegar... por eso decidimos que la estructura sería de un módulo, aunque conscientes que el número de líneas de código se podía disparar de lo habitual (cosa que ocurrió al incorporar los requisitos de última hora que nuestro cliente nos propuso).

3.3 Características técnicas

3.3.1 Configurabilidad

A la hora de hablar de las características de configuración del módulo, lo haremos de dos partes; una la zona de los filtros, cache, etc.... y otra sobre todo lo relativo a las rss.

En la primera parte, como podemos ver a continuación hay una serie de parámetros que podemos acceder desde el Back-End para configurarle. Esta es una vista general para tomar perspectiva del programa.

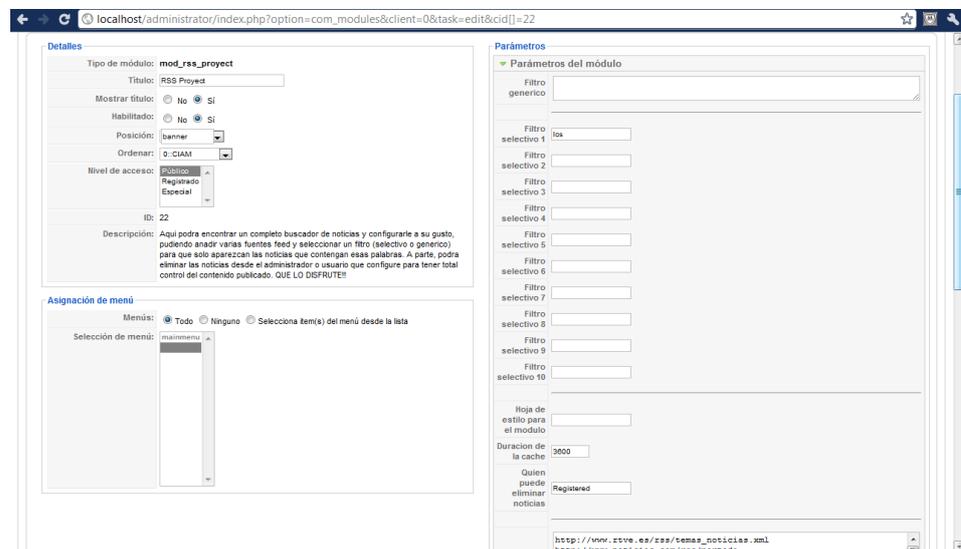


Figura 3.3.1.A: Back-End de configuración del módulo.

La zona que nos interesa es la de la derecha, que es la programada por nosotros. La columna de la izquierda es genérica para todos los módulos y no tenemos permisos en ella. Para empezar, explicaremos los campos en sentido descendente:

Existen dos tipos de filtros dentro del módulo: genérico y selectivo. Estos filtros están relacionados entre sí y la aparición de uno, ninguno o los dos tienen diferentes efectos sobre el módulo. El filtro específico es

prioritario; basta con que uno de los 10 este escrito para que no se ejecuten la información del resto. El genérico, para ejecutarse es necesario que el específico este en blanco. En caso de que los dos estén vacíos, se escogerán el número de noticias por orden de aparición, como si de un sindicador normal se tratase.

- **Filtro genérico**: Se puede poner un número ilimitado de palabras. Este filtro lo que hace es seleccionar individualmente cada una de ellas y busca una coincidencia dentro del titular de la noticia (es sensible a mayúsculas y minúsculas). Cuando esto ocurre, la noticia es mostrada por pantalla. (Siempre que cumpla otros requisitos como: no estar en la lista negra, no sobrepasar el número de noticias...).
- **Filtro selectivo**: Se pueden poner un número ilimitado de palabras. Debemos tener en cuenta que las condiciones de este filtro para que se cumpla es que coincidan todas las palabras del titular con todas las palabras del filtro selectivo. Un ejemplo: Si por ejemplo ponemos en el filtro 'Rafa Nadal'; nuestro filtro no seleccionara el titular que sala 'Rafa no podrá jugar por una lesión en su rodilla', pero sin embargo sí que seleccionará 'El tenista español Rafa Nadal no podrá disputar el torneo'. En este último caso sí que aparen las dos palabras del filtro por lo que sí que se ejecutará. Podemos poner hasta un máximo de 10 filtros selectivos con nuestras palabras clave.
- **Hoja de estilo para el módulo**: Podemos seleccionar un sufijo de las hojas de estilo en cascada. De este modo, el módulo podrá tomar un estilo diferente al predeterminado a toda la hoja.
- **Quién puede eliminar noticias**: Aquí pondremos unos de los nombre definidos por Joomla! en el ACL; de esta manera, al usuario que le otorguemos esos poderes, a la hora de registrarse podrá eliminar noticias que pasarán a la base de datos.

Figura 3.3.1.B: Back-End de los filtros.

URLS feed	<pre>http://www.rtve.es/rss/temas_noticias.xml http://www.noticias.com/rss/portada http://www.20minutos.es/rss/ http://www.nortecastilla.es/portada.xml http://www.abc.es/rss/feeds/abcPortada.xml http://www.elpais.com/rss/feed.html?feedId=1022 http://elmundo.feedsportal.com/elmundo/rss/portada.xml http://www.larazon.es/rss</pre>
Visualizar la fuente feed	<input type="radio"/> Si <input checked="" type="radio"/> No
Descripcion del feed	<input type="radio"/> Si <input checked="" type="radio"/> No
Imagen Feed	<input type="radio"/> Si <input checked="" type="radio"/> No
Numero de articulos a visualizar	<input type="text" value="5"/>
X palabras del titulo del articulo	<input type="text" value="15"/>
Descripcion del articulo	<input type="radio"/> Si <input checked="" type="radio"/> No
X palabras de la descripcion del articulo	<input type="text" value="25"/>
Ver las imagenes de cada articulo	<input type="radio"/> Si <input checked="" type="radio"/> No
Abrir la noticia	<input checked="" type="radio"/> En una ventana nueva <input type="radio"/> En la misma ventana
Añade el atributo 'no follow' a los enlaces	<input checked="" type="radio"/> No <input type="radio"/> Si

Figura 3.3.1.C: Back-End con las opciones por defecto.

En la figura 3.3.1.C, encontraremos parte de nuestra configuración feed, donde obtendremos un control muy amplia a la hora de configurara a nuestro gusto el módulo.

- **URLS feed:** Aquí podremos poner todas nuestras feed que queramos suscribirnos, el único requisito es que debemos dejar un espacio de separación, para que el programa reconozca que son URLS distintas.
- **Visualizar la fuente feed:** Permite ver el nombre del periódico, web... del que se ha extraído la información. Aparece una vez solo, y todos sus ítems debajo.
- **Descripción del feed:** Visualiza la información del periódico, web, etc. De la descripción que haya almacenado.
- **Imagen del feed:** Si la fuente feed posee una imagen, la muestra por pantalla. En caso contrario no ejecuta nada.

- **Número de artículos a visualizar:** seleccionamos un valor numérico, que será el número de artículos que se mostrarán. En caso de que se llegue al límite, el programa parará y mostrará por pantalla ese número de artículos. En el caso de que no se llegue al límite establecido, el programa se ejecutará hasta el final y mostrará los resultados obtenidos.
- **X palabras del artículo a visualizar:** Delimitas el número de palabras que se mostrarán del titular.
- **Descripción del artículo:** Habilitar o deshabilitar la descripción del artículo, en caso de que la posea.
- **X palabras de la descripción del artículo:** Delimitas el número de palabras que quieres que aparezcan tras el enlace del artículo. Es necesario que este activado el campo anterior para que este pueda ejecutarse.
- **Ver imágenes de cada artículo:** Puedes habilitar para que en caso de que el artículo posea imágenes, se vean a continuación del enlace (la mayoría de las rss no incorporan imágenes, por lo que es poco probable que se puedan visualizar).
- **Abrir la noticia:** Puedes controlar donde quieres que se vea la noticia. Una vez pinchado en el enlace, podemos ver la noticia dentro de nuestra página web, o en otra pestaña aparte.
- **Añade atributo 'no follow' a los enlaces:** útil cuando es una página institucional (como es el caso nuestro de la asociación CIAM), ya que evita que paginas de terceros se aprovechen de nuestro *page Rank* para fines indebidos.

Para terminar, hablaremos del Tooltip, que es una herramienta que se activa al pasar por encima de la noticia el ratón y que podemos activar distintas funciones.

Habilitar Tooltip	<input checked="" type="radio"/> Si <input type="radio"/> No
X palabras del titulo tooltip	<input type="text" value="25"/>
X palabras de la descripcion tooltip	<input type="text" value="25"/>
Imagen en el tooltip	<input type="radio"/> Si <input checked="" type="radio"/> No

Figura 3.3.1.D: Tooltip.

- **Habilitar Tooltip:** Permite activar o desactivar esta función. Los siguientes campos no surgirán efecto si no está activada esta opción.

- **X palabras del Tooltip**: Permite configurar el número de palabras a visualizar del titular de la noticia en el Tooltip.
- **X palabras de la descripción del Tooltip**: Configuramos el número de palabras de la descripción que se va a ver.
- **Imagen del Tooltip**: Si queremos que se muestre la imagen o no dentro del Tooltip.

3.3.2 Compatibilidad

La versión programada de este módulo es compatible en cualquier versión 1.5 de Joomla! Tras una serie de testeos, podemos afirmar que todas las funcionalidades permanecen estables, tanto para la primera, como la última versión de Joomla! 1.5.

Sin embargo, instalar la versión 1.6 de Joomla! y como ya habíamos leído en foros y paginas de Joomla! oficiales, el módulo no es compatible. Al intentar instalarlo en esta versión el módulo no da ningún problema y se instala con normalidad, es reconocido por el sistema y todo es correcto, aparentemente; pero a la hora de configurar el módulo para luego verlo en pantalla, el módulo aparece vacía la columna del Back-End de la derecha, no permitiendo configurar nada en absoluto. A pesar de dar los permisos para visualizarse, al ver el Front-End el módulo no se muestra, sale vacío.

Esta podría ser una de las desventajas de este módulo, pero como ya hemos comentado antes, a pesar de todas las mejoras de la versión 1.6, no está teniendo el éxito que se esperaba, debido a dos causas: que no es compatible hacia atrás, por lo que solo el 7% de los módulos sirven, y que la versión 1.7 tiene previsto ver la luz en Julio de 2011 (solo 6 meses de vida), por lo que la mayoría de los usuarios están manteniendo la versión 1.5 de sus plataformas (lleva ya una amplia trayectoria de 3 años) para migrar a la versión 1.7 más adelante si les es favorable.

Métodos de evaluación

4.1	Velocidad de respuesta del módulo RSS_PROYECT	39
4.2	Resultados de velocidad obtenidos de los indicadores	42
4.2.1	Módulo Feed	42
4.2.2	Módulo RX RSS Sticker	43
4.2.3	Módulo Slick RSS	43
4.2.4	Módulo Browser 2.7	44
4.2.5	Módulo RSS_PROYECT	45
4.3	Comparativa de velocidad de los indicadores	45

4.1 Velocidad de respuesta del módulo RSS_PROYECT

Realizaremos una prueba real en la página web de la asociación CIAM, que es AMAREX, donde utilizaremos el programa FIREBUG, con el que realizaremos unos test y unas mediciones de respuesta, antes y después de nuestro módulo. La conexión utilizada es de 6Mbits downstream y 512 upstream del proveedor de internet.

A parte, realizaremos un estudio para ver el promedio de los tiempos que tardan en procesar los servidores, varían las variables de:

- Numero de feed.
- Número de palabras filtro.
- Número de artículos a visualizar.

Con todo esto, veremos cuáles son los puntos fuertes y deduciremos los procesos críticos en los que se requieren más recursos. Toda esta información podrá ser utilizada en futuros líneas de trabajo, que busquen soluciones para aumentar el rendimiento.

También hay que decir, que las dos últimas semanas de trabajo sobre el código, se llevaron a cabo una reestructuración y ‘escapes’ de los bloques con los que subjetivamente se apreciaban mejoras en los tiempos de procesado significativas. Las primeras versiones, estaban dirigidas principalmente a que el programa funcionase de manera correcta, independientemente de la redundancia de los bucles que se pudiesen dar y del tiempo en que tardase en procesar el servidor los parámetros a calcular.[24-28]

Para poder tener una referencia real, hemos realizado una medición antes de implantar el módulo y así tener una referencia del retardo que puede introducir el retardo. La velocidad en cargar la página sin el módulo ha sido de 1,07 segundos. Todo esto y las capturas de pantalla las podremos ver en el ANEXO.

Estudiaremos el primer caso, que se trata del filtro genérico y sus respuestas en segundos con respecto a los cambios en los parámetros de URLS y palabras del filtro.

Debemos tener en cuenta que en la configuración Back-End se ha configurado para que salgan los 100 primeros artículos, utilizando para ello la web sobre la que vamos a utilizar este modulo (se adjunta más información y capturas de pantalla en el ANEXO).

URL\Palabras	1 palabra	10 palabras	20 palabras	30 palabras
3	1,86	1,96	2,23	1,79
6	2,17	2,24	2,4	2,62
9	2,72	3	3,05	3,17
12	2,98	3,14	3,86	4,28

Tabla 4.1.A: Filtro genérico.

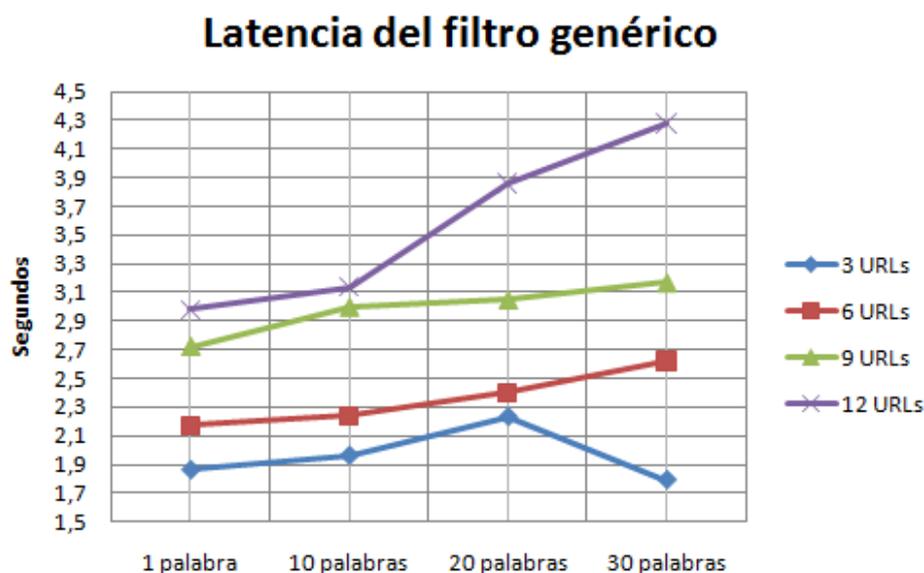


Figura 4.1.A: Gráfica con los datos de velocidad obtenidos.

Como podemos observar, al aumentar en número de fuentes feed que tenemos que obtener noticias, el retardo se acentúa, ya que el cálculo es mayor. Sin embargo, podemos ver como el peor tiempo de respuesta del programa se encuentra en 4,28 segundos, lo que significa que en el peor de los casos, el tiempo sería de 3,21 segundos en cargar; tiempo no excesivamente grande teniendo en cuenta que es la peor situación. La mejor respuesta la obtenemos, como es lógico, con 3 URLs, en 1,79 segundos, 0,6 segundos más lento, lo cual indica que el procesado ha sido rápido, en parte por la limitación de la muestra.

El tiempo promedio de todo el estudio se encuentra en torno a 2,8 segundos, lo que significa 1,8 segundos de retraso que introduciría de promedio nuestro módulo en caso genérico; lo cual resulta aceptable. Debemos tener en cuenta factores como: el tráfico de la red, alcanzar el límite de artículos, el número de feed. Cabe mencionar, que al aumentar el número de palabras, el tiempo de carga aumenta, al igual que si aumentamos el número de fuentes feed, como se muestra las tendencias en la gráfica.

Estudiaremos el caso del filtro específico. El tiempo de carga sin módulo obtenido es de 2,46 (véase ANEXO). Es un tiempo superior al del mismo caso que el anterior, pero debemos tener en cuenta que no solo se mide la rapidez del módulo, sino que depende del estado de la red y del servidor, variables que condicionan estos parámetros y que no podemos controlar.

Mostramos a continuación los resultados obtenidos, para el caso de 2 artículos, configuración de nuestro cliente:

URL\Palabras	1 palabra	4 palabras	7 palabras	10 palabras
6	3,74	3,62	4,4	5
12	3,33	3,39	3,5	5,75
18	8,05	17,93	9,39	12,25
24	17,98	18,13	8,48	12,44

Tabla 4.1.B: Tabla de filtro específico.

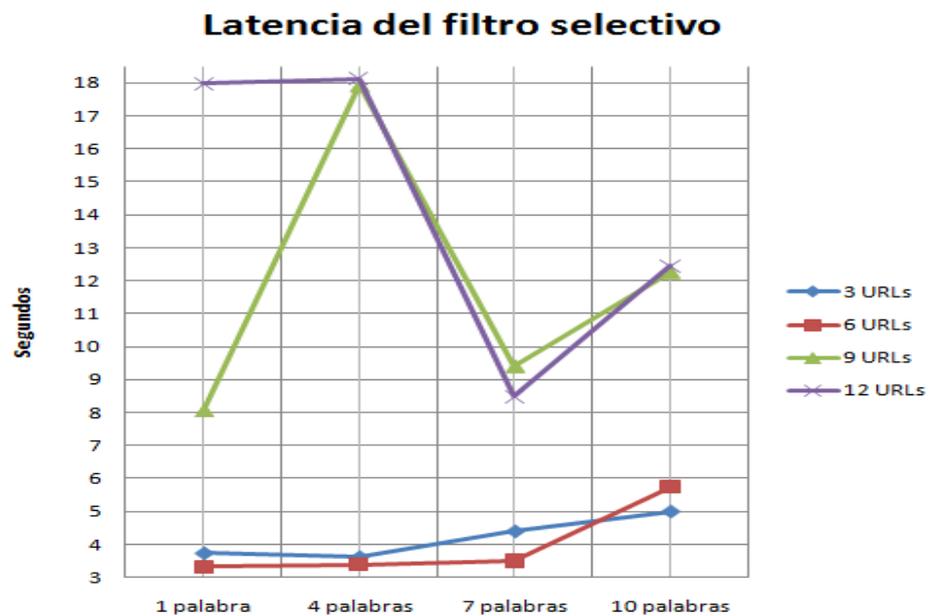


Figura 4.1.B: Gráfica con los datos del filtro selectivo de velocidad.

Como se ve a simple vista, en este caso los tiempos de retardo que introduce el módulo son críticos. El peor caso es con 24 URLs y 4 palabras. Puede que sea un poco extraño que se obtenga con 4 palabras, en vez de con 10; pero si observamos detenidamente la gráfica, observamos que esto sucedía en los caso que íbamos reduciendo las feed. Esto quiere decir, que cuando eliminábamos ciertas palabras a partir del 4 filtro (en nuestro caso) el programa se ejecutaba entero, ya que no encontraba el número de artículos máximo, y por eso tardaba más, ya que se tenía que escanear todos los artículos de la feed.

En el mejor de los casos, el retardo introducido ha sido de 3,74; por encima de la media del filtro genérico. Esto se debe a la naturaleza del algoritmo, ya que en este filtro selectivo, el número de operaciones crece exponencialmente con respecto al genérico.

El promedio final de este filtro selectivo es de 8,4 segundos, lo que representa casi 6 segundos más lento al cargar que sin el módulo. Es por esto que conviene estudiar detenidamente el número de filtros selectivos a poner y sus fuentes, ya que pueden hacer de la ejecución, que sea crítica, a ser navegable. Como conclusión sacamos, que a pesar de no ser

intuitivo, al aumentar el número de palabras filtros mejoramos la respuesta del programa con cierta independencia de las URLs.

Como conclusión, el estudio realizado de su velocidad en ejecución, han confirmado lo que se intuía a la hora de programar los algoritmos del modulo. A mayores fuentes feed, mayor carga computacional. Cabe destacar la singularidad ocurrida en el filtro selectivo, que bien podría deberse al orden de las palabras y el mero azar de su colocación.[29][30]

4.2 Resultados de velocidad obtenidos de los sindicadores

Utilizando la herramienta de software FIREBUG , hemos realizado una serie de medidas para poder comparar con las distintas herramientas y posteriormente poder realizar su evaluación. Dado las especificaciones desarrolladas en los capítulos 2 y 3, hemos decido realizar 10 capturas por pantalla, quedándonos como referencia la promedio de todos los casos, con el fin de evitar valores incorrectos o aleatorios en los resultados.

4.2.1 Módulo feed

Se trata del módulo que viene por defecto instalado en el paquete de instalación de Joomla! Las funcionalidades de este módulo son muy básicas, ya que lo único que hace es mostrar por pantalla las noticias sindicadas. La motivación principal para analizar este módulo, es ofrecer al lector una referencia desde la cual poder comparar.

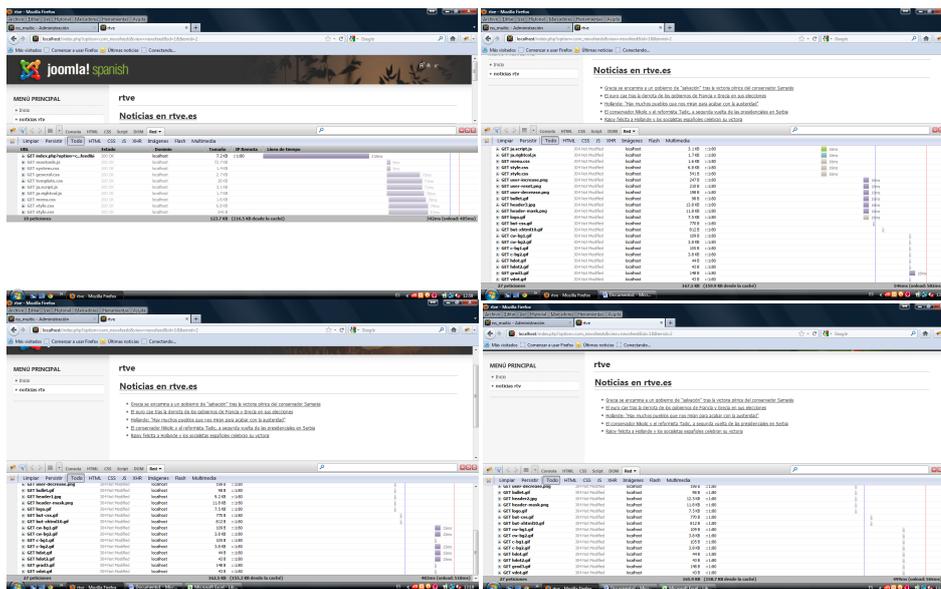


Figura 4.2.1: Capturas de pantallas con las velocidades registradas, por orden: Esquina superior izquierda 342ms, esquina superior derecha 546ms, esquina inferior izquierda 483ms, esquina inferior derecha 499ms.

Los datos obtenidos de este módulo son los más pequeños, como era de esperar. En el siguiente capítulo detallaremos la explicación de estos resultados y su relevancia. Todo esto lo podemos observar en la figura 4.2.1.

4.2.2 Módulo RX RSS Sticker

Se trata de un módulo que ya realiza procesamiento de los datos obtenidos, de ahí que los tiempos sean superiores a los del FEED. La velocidad promedio del módulo es de 620ms. En la figura 4.2.2, observamos parte de los valores obtenidos en la recolecta de datos.

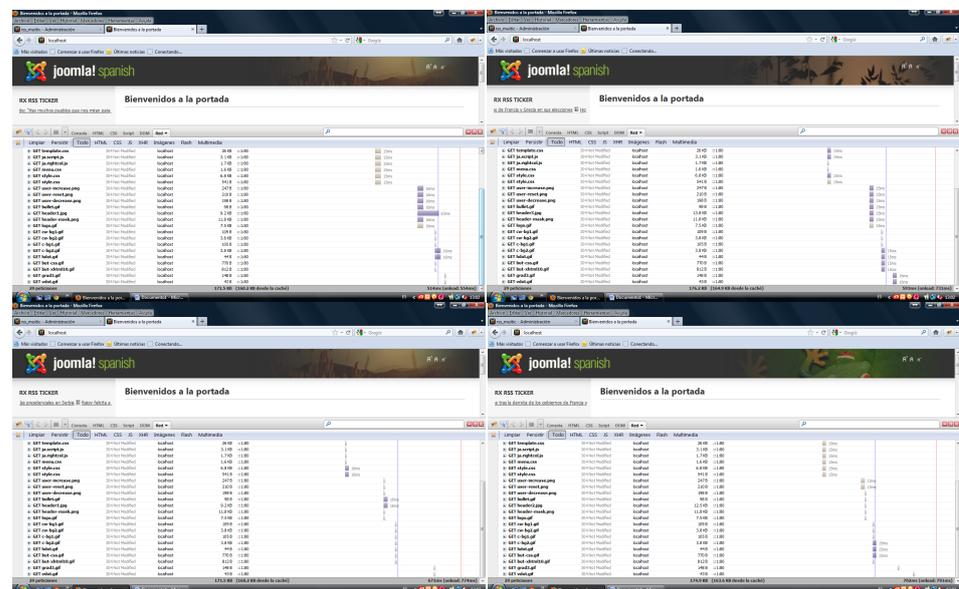


Figura 4.2.2: Capturas de pantallas con las velocidades registradas, por orden: Esquina superior izquierda 514ms, esquina superior derecha 593ms, esquina inferior izquierda 671ms, esquina inferior derecha 702ms.

4.2.3 Módulo SLICK RSS

Al igual que su predecesor, este modulo realiza procesamiento de datos, sin embargo, los datos obtenidos son ligeramente mejores que el RX RSS Sticker, por lo que se desprende una posible mejor optimización en el procesado. La velocidad promedio del módulo es de 592,75ms. En la figura 4.2.3, observamos parte de los valores obtenidos en la recolecta de datos.

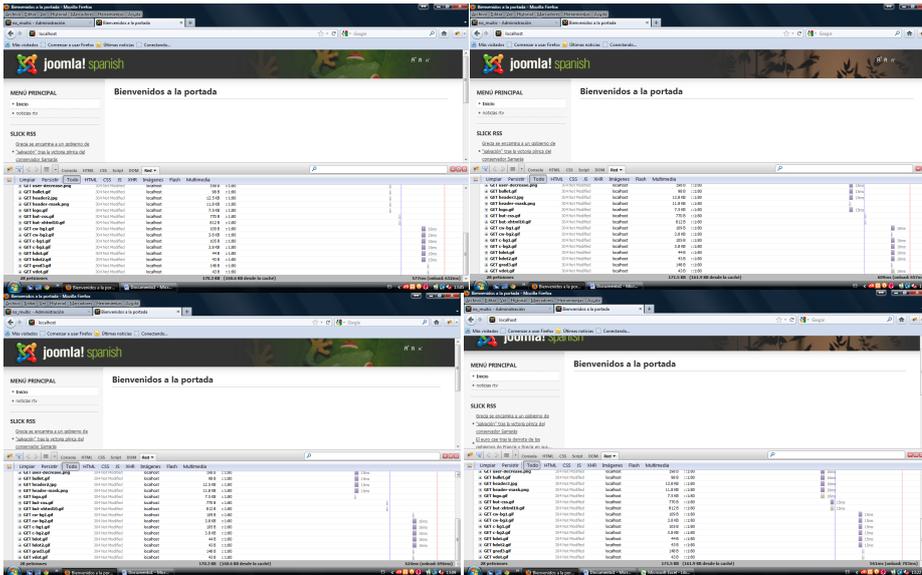


Figura 4.2.3: Capturas de pantallas con las velocidades registradas, por orden: Esquina superior izquierda 507ms, esquina superior derecha 609ms, esquina inferior izquierda 609ms, esquina inferior derecha 524ms.

4.2.4 Módulo BROWNSER 2.7

El último de los módulos estudiados que realizan procesado de los datos obtenidos y el que peores resultados obtiene en comparación con los otros dos anteriores que realizan procesado de datos. La velocidad promedio del módulo es de 651ms. En la figura 4.2.4, observamos parte de los valores obtenidos en la recolecta de datos.

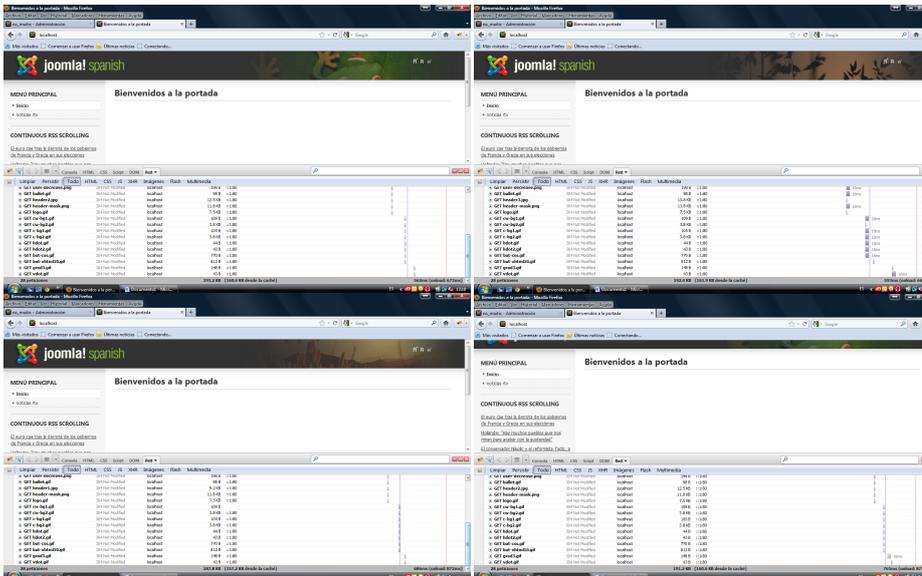


Figura 4.2.4: Capturas de pantallas con las velocidades registradas, por orden: Esquina superior izquierda 562ms, esquina superior derecha 593ms, esquina inferior izquierda 686ms, esquina inferior derecha 765ms.

4.2.5 Módulo RSS PROTECT

A continuación mostramos los resultados obtenidos por nuestro módulo. A pesar de que realiza mayor número de procesado y de iteraciones en comparación con las funcionalidades de los otros módulos, observamos que los tiempos promedio son ligeramente inferiores a los registrados en los anteriores casos. La velocidad promedio del módulo es de 565,25ms. En la figura 4.2.5, observamos parte de los valores obtenidos en la recolecta de datos.

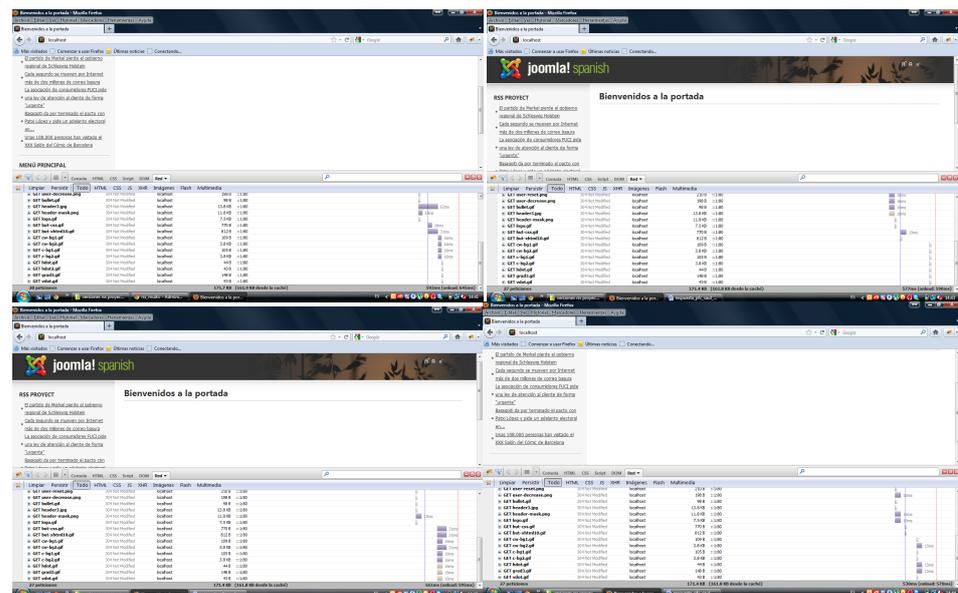


Figura 4.2.5: Capturas de pantallas con las velocidades registradas, por orden: Esquina superior izquierda 562ms, esquina superior derecha 593ms, esquina inferior izquierda 686ms, esquina inferior derecha 765ms.

4.3 Comparativa de velocidad de los sindicadores

La relevancia del buscador RSS_PROYECT es la capacidad de realizar búsquedas más rápidas en comparación con las herramientas investigadas dentro de Joomla! 1.5, optimizando su velocidad de procesamiento y añadiendo funcionalidades de administración y configuración de las noticias.

Esta herramienta permite al administrador un control total sobre la publicación de los artículos, permitiendo eliminar artículos encontrados por el buscador (se almacenan en la base de datos llamada black_list) y que considere que no deben ser publicados en la Web. Además, permita una asignación de atributos (visualizar, ocultar, eliminar...) en base a la

jerarquía del usuario (administrator, superadministrator, gestor, publicador...) nunca antes utilizada en las herramientas RSS de Joomla! 1.5, y que la diferencia del resto de herramientas de su área.

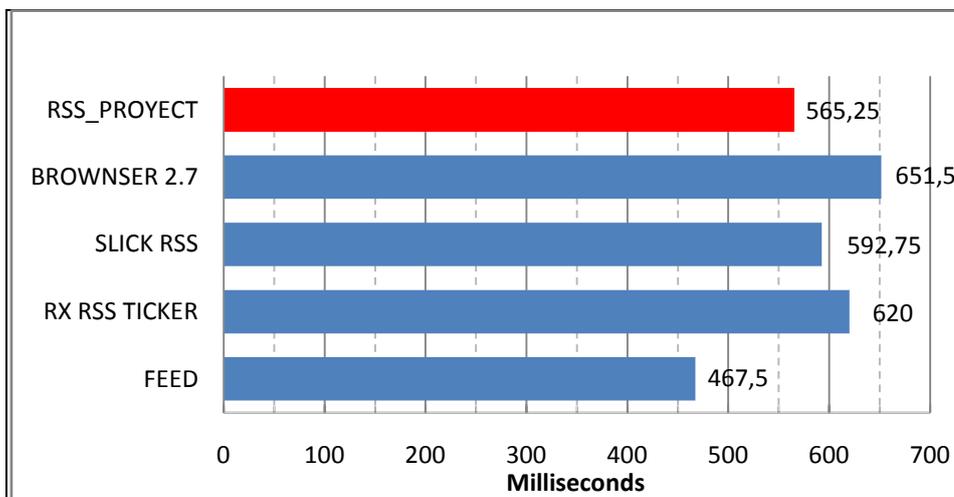


Figura 4.3.A: Velocidad de tiempos de respuesta para los módulos Joomla! analizados.

En la figura 4.3.A, observamos como los tiempos promedio son mejores para RSS_PROYECT, en comparación con los módulos estudiados en Joomla! 1.5. Hemos incluido a modo de referencia el buscador FEED que se encuentra incluido en el paquete de instalación de Joomla! por defecto, para ilustrar al lector sobre los tiempos de procesado. A la hora de evaluar los resultados, debemos tener en cuenta los criterios de desarrollo e implementación descritos en capítulo 3.

En la figura 4.3.B, se muestra una gráfica con los tiempos promedio que tardan en ejecutarse las búsquedas por los distintos módulos. Vemos como en este caso concreto, la velocidad de ejecución de RSS PROYECT es relativamente menor incluso que el módulo FEED; esto se debe a casos particulares en los que el módulo RSS_PROYECT encuentra el valor requerido antes de tiempo y abandona el bucle de búsqueda, por lo que al aumentar el número de búsquedas en fuentes, la velocidad se mantiene, mientras que la tendencia de los otros módulos es ejecutarse hasta el final, por lo que aumenta el tiempo de procesado al aumentar el volumen de información.

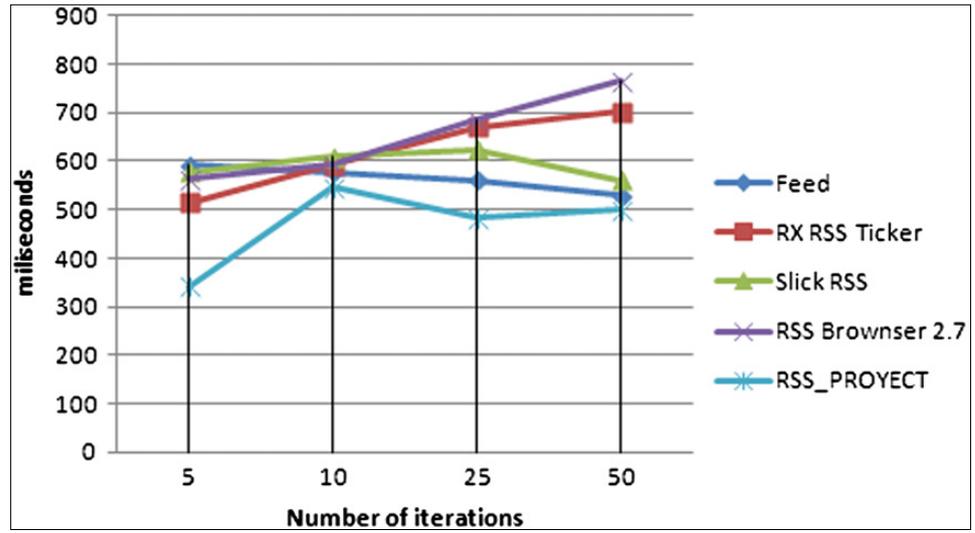


Figura 4.3.B: Tiempos promedio de los módulos al variar el número de fuentes feed a syndicar.

***Discusión de los resultados, conclusiones
y líneas futuras***

5.1	Discusión de los resultados	51
5.2	Conclusiones	51
5.3	Líneas futuras	52

5.1 Discusión de los resultados

A la hora de analizar los resultados, debemos tener en cuenta una serie de factores condicionantes que pueden alterar el valor de los resultados, y una serie de condicionantes por los que es difícil replicarlos y obtener los mismos valores exactos. No obstante, a pesar de las circunstancias que describiremos a continuación, cabe destacar que los resultados relativos que se obtengan de las replicas de las mediciones si que coincidirán con los expuestos en este trabajo fin de master, así como los resultados publicados en la revista ‘Computer and network applications’ [31].

La razón de que los resultados obtenidos no se puedan replicar exactamente es la variabilidad de los contenidos que extraemos, ya que son contenidos de información dinámicos en el tiempo, y no estáticos; por lo que al relizar la ejecución del módulo en un determinado instante y realiarla posteriormente puede darse el caso de que nueva información haya sido indexda en las fuentes, por lo que una misma búsqueda en distintos instantes de tiempo dará diferentes resultados, a pesar de mantener los parámetros de configuración.

Otro de los factores que influyen en el procesado de los datos, es el estado de la red, no tanto por la velocidad de subida y bajada, sino por la velocidad de procesado y la saturación de los servidores, ya que todo este trabajo de procesado se realiza del lado del servidor, y no del lado del usuario, por lo que es altamente variable.

No obstante, como solución a estas variables dinámicas que nos afectan a la hora de replicar nuestro trabajo, sería interesante proponer un repositorio universal para todo aquel que trabaje con contenido RSS, ya que sería la mejor forma para poder replicar y validar los datos (todo esto lo desarrollaremos en la sección de líneas futuras en detalle).

Los resultados obtenidos, no dejan luar al duda de que el módulo RSS_PROYECT supone un avance significativo, tanto en velocidad, en la que supera entre un 5% y 15% a los anteriores módulos, teniendo en cuenta que las funcionalidades del módulo son entre 3 y 5 veces mayores que estos, por lo que supone una gran mejora en el servicio y una mejor optimización de los recursos.

5.2 Conclusiones

Las conclusiones que hemos alcanzado en el aspecto técnico, es decir, la velocidad de carga del módulo, hemos podido observar que la carga computacional que suponen los algoritmos del módulo es baja, y con muy poco peso (400Kb), por lo que la hace idónea también para visualizarlo a través de un dispositivo móvil. (Cabe destacar, que para el caso de filtro selectivo, con más de 20 feed, la carga computacional sí que es significativa, por lo que habría que

realizar un estudio sobre la conveniencia o no, tanto para dispositivos fijos, como móviles).

Otra de las conclusiones alcanzadas, es que la herramienta desarrollada en nuestra investigación es superior en velocidad de procesamiento, así como en funcionalidades, con respecto a las herramientas más avanzadas en su momento. En concreto, un 15% mayor de velocidad de procesado con respecto al peor resultado obtenido.

El hecho de que posea 2 tipos de filtros, uno selectivo y otro genérico, constituye una potente herramienta y una característica diferenciadora sin duda de los otros módulos que podemos encontrar dentro del segmento; como señalamos en el capítulo 2, sin duda alguna, este módulo aporta mayores ventajas que los de su segmento, lo que podría diferenciarle y lanzarle al éxito.

Los niveles de seguridad implementados, son los máximos que se pueden implementar a día de hoy dentro de un módulo, por lo que nuestra herramienta podemos asegurar que no va a ser constituir un agujero de seguridad dentro de nuestra web y va a poder estar bien protegido. Esto se debe a que los comandos utilizados han sido a través de la API de Joomla!, por lo que el módulo no es vulnerable.

Además, pensamos que este módulo puede tener éxito en la web, ya que no hay nada parecido en Joomla! (al menos de manera gratuita). Por lo que sería interesante poder colgarle en la página oficial de Joomla!, para que usuarios puedan descargarlo y adaptarlo a sus necesidades.

Por último, como fruto de este esfuerzo, hemos conseguido desarrollar una herramienta de utilidad para la asociación CIAM, y tan bien y muy importante, una herramienta que pudiesen instalar y disfrutar millones de usuarios de Joomla! en todo el mundo, en un mercado emergente, como es la plataforma Joomla!, en la que hemos podido ver un mercado en expansión. Son cada vez más, las extensiones de Joomla! que empiezan a ser de pago, profesionalizando el mercado.

5.3 Líneas futuras

A pesar de ser una herramienta bastante completa dentro de los módulos de Joomla!, siempre hay espacio para la mejora y para las líneas futuras. A continuación pondré posibles líneas y su explicación:

- Repositorio común de noticias RSS: sin duda la creación de un repositorio común ayudaría al testeo y poder dilucidar si la herramienta es mejor o peor que otras existentes, de manera unívoca, por lo que se replicación y posterior validación por terceros resultaría mucho más sencilla.

- Compatibilidad con Joomla! 2.5: una de las principales líneas de trabajo que trataría, aunque tenemos que tener en cuenta que la versión 2.5 de Joomla. Esta versión (1.5) seguirá siendo utilizada por unos cuantos años, dado que con la nueva evolución presente no es compatible hacia atrás, por lo que dejarían de funcionar los módulos si se migrase a las últimas versiones.
- Añadir noticias: Sería interesante incluir una opción en la que el administrador pudiese añadir noticias que encontrase de su interés, ya que el programa hace unas búsquedas sobre las web que le indiques, pero se podrían poner alguna de su interés particular.
- Escala de valores de las noticias: Otra de las mejoras que se podría hacer, es que si por ejemplo las noticias de una web que se sindicán , son de un tipo que se actualizan lentamente, se podría poner una escala, para que los usuarios las valorasen, y por ejemplo no se quedase desactualizada la noticia.
- Aumentar el tamaño de las URLS buscadas: >consistiría en buscar una función o un algoritmo que diese en dos las URLS muy largas para así poder procesarlas, ya que una de las limitaciones que posee el programa, es que las URLS muy largas (más de 150 caracteres), no es capaz el módulo de eliminarlas, si de procesarlas y almacenarlas en la base de datos, pero no es capaz de compararlas, ya que por decirlo de alguna manera, estas URLS se ‘desbordan’ y no es capaz de procesarlas la sentencia lógica.
- Algoritmo para eliminar de la base de datos noticias: Aunque realmente los artículos eliminados no suponen un peso bits significativo, ni en memoria, ni en ejecución, se podría poner una especie de temporizador a las noticias que se almacenan en la base de datos (lista_negra), porque a lo largo de los años, aumentarían el cálculo del programa y podrían ir más lento (recuerda que las noticias se van a actualizando, por lo que desaparecen y no tiene sentido seguir almacenándolas).
- Filtros selectivos infinitos: En nuestro caso hemos puesto 10 filtros selectivos, ya que con esto satisfacíamos las necesidades de nuestro cliente, pero puede darse la circunstancia que un usuario necesitase más aun. Pues esta podría ser una posible línea de estudio y trabajo.
- Mejora en los buscadores: Podríamos mejorar los algoritmos de búsqueda para que por ejemplo, sean capaces de buscar palabras de otros alfabetos, o sepan interpretar palabras con signos de interrogación, exclamación... dotándoles de una inteligencia artificial capaz de saber distinguir este tipo de símbolos.

Bibliografía

- [01] B, Hull. Telematics and Informatics, 2003. *ICT and Social Exclusion: The Role of Libraries* , pp. 20- 131.
- [02] BOURAS, Christos; POULOPOULOS, Vassilis and TSOBKAS, Vassilis. Adaptation of RSS Feeds Based on the User Profile and on the End Device. *Journal of Network and Computer Applications*, 7, 2010, vol. 33, no. 4, pp. 410-421. ISSN 1084-8045.
- [03] J, Estivill. *Concepts and Strategies for Combating Social Exclusion: An Overview*. . J, Estivill ed., Porugal: International Labour Office, 2003.
- [04] MINISTRY OF HEALTH, SocialPolicyandEquality,SpanishGovernment. *I Annual Report about Violence*. , 2007.
- [05] NING, Xiaomin; JIN, Hai and WU, Hao. RSS: A Framework Enabling Ranked Search on the Semantic Web. *Information Processing & Management*, 3, 2008, vol. 44, no. 2, pp. 893-909. ISSN 0306-4573.
- [06] SAMPER, Juan J., et al. NectaRSS, an Intelligent RSS Feed Reader. *Journal of Network and Computer Applications*, 11, 2008, vol. 31, no. 4, pp. 793-806. ISSN 1084-8045.
- [07] Andy Harris, *Programación con PHP 6 y MySQL*. Anaya Multimedia 2009
- [08] Luis Miguel Granado cabezas, *PHP 6, manual imprescindible*. Anaya Multimedia 2009
- [09] Apache friends, página web del servidor Apache donde está el XAMPP. <<http://www.apachefriends.org/en/xampp.html>>. Última consulta 16 de Noviembre de 2010.
- [10] Información sobre las fuentes RSS, < <http://www.rss-specifications.com/>> Última consulta 30 de junio de 2012.
- [11] Información concerniente a Joomla!, sus versiones, cronología y principales características. < <http://www.joomla.org/>>. Última consulta 30 de junio de 2012.
- [12] Información sobre el gestor de RSS Google Reader y su funcionamiento. < <http://www.google.es/reader>>. Última consulta 30 de junio de 2012
- [13] Información sobre el Sindicador de noticias RSS Netvibes. http://faq.netvibes.com/what_can_i_use_with_my_netvibes_page/_rss_or_web_feeds. Última consulta 25 de junio de 2011
- [14] Página de la wikipedia, con la historia y desarrollo de la herramienta Netvibes. <<http://www.netvibes.com/es>>. Última consulta 23 de junio de 2011.

- [15] Programa sindicador de feed, Feeddemon. <http://www.desarrolloweb.com/articulos/feeddemon.html>. Última consulta 23 de junio de 2011.
- [16] Pagina oficial de la web Joomla! <<http://www.joomla.org/>>, Última consulta 28 de junio de 2011.
- [17] CIAM Website. Available from: <http://www.josefinas-trinitarias.org/amarex/index.php?option=com_content&view=article&id=7&Itemid=8S>.
- [18] Mejoras y evolución de Joomla! 1.6. <http://www.nosolocodigo.com/5-razones-por-las-cuales-deberias-usar-joomla-en-lugar-de-wordpress>. Última consulta 29 de junio de 2011.
- [19] Mejoras y evolución de Joomla! 1.6. <http://www.joomlapanama.org/index.php?option=com_content&view=article&id=2:porque-joomla&catid=3:informacion&Itemid=19>. Última consulta 29 de junio de 2011.
- [20] Pagina oficial de Joomla! en español. <<http://ayuda.joomlaspanish.org/content/view/382/114/>>. Última consulta 29 de junio de 2011.
- [21] Actualizaciones y versiones estables de Joomla! 1.6. <<http://ayudajoomla.com/joomla/noticias/347-joomla-16-o-joomla-15-icual-uso.html>>. Última consulta 29 de junio de 2011.
- [22] Pagina oficial de Joomla! en español, ayuda y soporte de código. <<http://www.joomlaspanish.org/foros/showthread.php?t=65997>>. Última consulta 26 de junio de 2011.
- [23] Historia y versiones ejecutables estables de Joomla! <<http://cmsfactory.com/joomla-breve-historia-y-caracteristicas/>>. Última consulta 29 de junio de 2011.
- [24] Documentación relativa a código fuente de Joomla! y sus características. <<http://www.webempresa.com/creacion-web-joomla/caracteristicas-de-joomla.html>>. Última consulta 29 de junio de 2011.
- [25] Código Joomla! <<http://www.joomlaos.net/caracteristicas-de-joomla/>>. Última consulta 2 de julio de 2011.
- [26] Futuro de Joomla! 1.6 y la versión 1.7. <<http://ayuda.joomlaspanish.org/content/view/388/82/>>. Última consulta 29 de junio de 2011.

[27] Tutorial sobre el uso y transferencia de modulo a plugins. <<http://www.edujoomla.es/manuales-joomla-15/plugins-herramientas/142-los-plugins-en-joomla-15>>. Última consulta 1 de julio de 2011.

[28] Modificación de Plugins. <<http://www.nosolocodigo.com/tutorial-crear-plugins-joomla-15>>. Última consulta 27 de junio de 2011

[29] Corrección de error en el filtro selectivo. <<http://es2.php.net/manual/es/control-structures.foreach.php>>. Última consulta 3 de julio de 2011.

[30] Uso del comparador de cadenas en PHP 6. <<http://es2.php.net/manual/es/function.stristr.php>>. Última consulta 3 de julio de 2011.

[31] ÁLVARO-MUÑOZ, Saúl, et al. Development and Performance Evaluation of a New RSS Tool for a Web-Based System: RSS_PROYECT. *Journal of Network and Computer Applications*, no. 0. ISSN 1084-8045.

PUBLICACIONES

ARTICLE IN PRESS

Journal of Network and Computer Applications



Contents lists available at SciVerse ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca



Development and performance evaluation of a new RSS tool for a Web-based system: RSS_PROYECT

Isabel de la Torre-Díez^{a,*}, Saúl Álvaro-Muñoz^{a,1}, Miguel López-Coronado^{a,1}, Joel Rodríguez^{b,2}

^a Department of Signal Theory and Communications, University of Valladolid, Paseo de Belén, 15 47011-Valladolid, Spain

^b Instituto de Telecomunicações, University of Beira Interior, Rua Marquês d'Ávila e Bolama, 6201-001 Covilhã, Portugal

ARTICLE INFO

Article history:
Received 3 February 2012
Received in revised form
16 May 2012
Accepted 9 June 2012

Keywords:
Joomla!
RSS
Search engine
Social exclusion
Web-based

ABSTRACT

The aim of this paper is the development, implementation and subsequent performance evaluation of an innovative tool called RSS_PROYECT, based on the RSS technology for content syndication. At the present, there are several Web content syndicators; although there is no one so far specifically designed for the search of news related to women in risk of social exclusion. They do not even have the characteristics and configuration level of the program syndicator RSS_PROYECT, presented in this paper.

To obtain these contents, we used 2 filters (generic and selective) set up by the administrator from the module RSS_PROYECT installed in Joomla!. The generic filter allowed us to do a search of the words included in a series of sources indexed by the user. This filter showed all the sources that contained this word, without exception. In the selective filter, the condition on which the sources would appear would be that all the words of the selective filter were contained in the source.

We used different languages such as PHP, MySQL, HTML, XML and Joomla!'s Application Program Interface (API). Furthermore, we used the program, Firebug to measure the response speed of the module, both with the selective and generic filters. The results were favorable for the selective filter, and very favorable for the generic filter, concluding that the processing time was short and executed efficiently. Better average times were obtained for RSS_PROYECT with respect to other modules analysed in Joomla!. Nowadays, this tool is used by the Comprehensive Center for Women's Help (Centro Integral de Ayuda a la Mujer, CIAM) of Valladolid, Spain.

© 2012 Published by Elsevier Ltd.

1. Introduction

At the present, the new configuration of the labor market (which mostly affects women and young people), the existence of political and/or socio-cultural factors, joined to the absence or decrease of family support and ties, may end up in processes of social exclusion. Estivill, (2003) defined social exclusion as an accumulation of conclusive processes with successive breaks that, starting from the heart of economy, politics and society, gradually increases the distance between persons, groups, communities and territories with regard to the centers of power, the resources and the dominant values (Estivill, 2003).

One of the most important factors of exclusion still arises from the fact of having or not having a job. In this regard, the possibility of

having access to a job is going to be increasingly determined by the ability to make efficient use of the Information and Communication Technologies (ICT) (Hull, 2003; Warren, 2007). The Declaration of Principles of the United Nations (2004) argued that if an "Information Society" for all the people was really desired, it was necessary to promote an Information Society that relied on the individual as well as to harness the potential of these technologies to promote the development goals established in the Millennium Declaration (United Nations Development Programme, 2001).

Nowadays we are living in the information age. Every day we are bombarded with hundreds of news and have access to a variety of media (Internet, written press, television, radio, etc.) that makes it impossible for us to be able to process so much information. In the past, we had no choice but to wait for the news previously filtered by a few people or entities who decided what, when and how the information was presented. With the advent of the Internet all this changed, or at least "when" the news is given. People who surfed the net could now access the Web contents anywhere and anytime. All that meant a great achievement, but at the same time it demanded a few requirements that had not been thought of until then. The updating of

* Corresponding author. Tel.: +34 983423000x3703; fax: +34 983423667.

E-mail addresses: isato@tel.uva.es (I. de la Torre-Díez),

salvman@iberate.uva.es (S. Álvaro-Muñoz),

miglop@tel.uva.es (M. López-Coronado), joeljr@ieee.org (J. Rodríguez).

¹ Tel.: +34 983423000x3703; fax: +34 983423667.

² Tel.: +351 275 319 891; fax: +351 275 319 899.

1084-8045/\$ - see front matter © 2012 Published by Elsevier Ltd.

http://dx.doi.org/10.1016/j.jnca.2012.06.004

Please cite this article as: de la Torre-Díez I, et al. Development and performance evaluation of a new RSS tool for a Web-based system: RSS_PROYECT. Journal of Network and Computer Applications (2012), http://dx.doi.org/10.1016/j.jnca.2012.06.004

ARTICLE IN PRESS

2

I. de la Torre-Díez et al. / Journal of Network and Computer Applications ■ (■■■) ■■■–■■■

the information made the users consult the Web constantly. This meant a loss of time and an increase of Internet traffic, which was unnecessary.

From this came the idea of organizing the Web to keep updated with no need to be connected all the time. Most of the programs used for this were external to the browser and an agreement and a series of protocols were established to guarantee their universality. RSS is an XML format to syndicate and share content on the Web. It is employed for spreading frequently updated and personalized information among users subscribed to the source of content (Ning et al., 2008; Samper et al., 2008; Bouras et al., 2010). The format allows for the distribution of content without the need of a browser, using software designed to read these RSS contents. Likewise, Joomla! is a file management system and its main strength is to let you edit the contents of a Web site in an easy way. It is an open source application, programmed mostly in PHP under a GPL license. The content administrator can be used at Internet or Intranets and requires a MySQL database and, preferably, an Apache HTTP server.

It is important to emphasize the significance of this novel tool (RSS_PROYECT), as according to the Spanish Government, namely the Ministry of Equality and Social Affairs (Ministry of Health, Social Policy and Equality, 2007), 3.6% of Spanish women were battered and in risk of suffering social exclusion in 2011.

This opens up a new area within ICT giving way to some tools that can encourage the integration of these people into society, keeping them away from social exclusion. The tool presented in this paper is a search engine that can select very specific news from a high and indefinite number of sources and display them through its Web platform. It is a stand-alone tool that provides a continuous update of its content to reflect the current news.

In this way, RSS_PROYECT is a Web-based tool in Joomla! It has an easy interface to administrate in a simple way. It does not require professional programming knowledge. This will encourage self-management by the user in order to achieve greater success and satisfaction.

The rest of this paper is organized as follows: first, we will talk about similar applications that we can find in and out of Joomla!. Next, we will explain the technique that we have used and the results (performance evaluation and development module). Finally, the conclusions and future related works will be shown.

2. Related work

When looking for content, it is important to know the relevance it may have for us, although we cannot know it without knowing the prior impact of this and we do not have tools to calculate it.

In spite of that, we cannot establish a relevancy of our RSS source items; we have to provide a mechanism that allows our user to publish things when they want to. In the following paragraph, we will study some of the tools in and out of Joomla! with the highest impact, observing the number of users, allowing us to control our RSS items.

2.1. Syndicators outside of Joomla!

2.1.1. Google Reader

Google Reader is an Atom and RSS reader. It allows us to organise and to quickly access, from a Web interface, to all the news of the pages that are supported. Google launched it on October 7, 2005 by means of Google Labs, Google Reader incorporated to Google Labs on September 17, 2007. The interface is very similar to a normal email inbox reader. On the left side you can see the subscriptions, which the user has applied for

membership, and to the right, the list of new pieces of news, which can be seen in several ways (Google Reader RSS, 2011).

2.1.2. Omea reader

Omea Reader is a tool for reading. It is versatile and simple to set up for Feeds. Omea Reader is a perfect mixture of the RSS channels, favorites of the Internet (that can be read online) and newsgroups. The e-mail and instant messenger are not in Omea Reader. The interface gives a very personal and efficient impression. It allows you to search and to save the Web sites. You can save these searches in unlimited categories. The disadvantage is that it does not offer synchronization.

2.1.3. Netvibes

Netvibes is a Web service that acts like a virtual desktop, similar to a personalized Google main page. Visually, it is organized in tabs, where each tab is typically an aggregate of different modules and the user previously defines a scrollable widget. These modules, in turn, act as small windows which content is generated by another Web service or mini application. As other readers of generic RSS, they allow us to sign up for an undetermined number of Feeds. With it, we can be up to date on the news that publish diverse sources of information, which offers its contents on the Web, which we have selected, but without controlling the content (Netvibes, RSS, 2011).

2.1.4. FeedDemon

FeedDemon is a program where we can enter and see all of the news of our favorite sites at once. It provides various utilities to manage data sources RSS (feeds) like: ranking in several categories, notices when there are new sites entered, different views of reading RSS feeds, options specific to each subscribed feed, display posts in the program itself, connection with sites of sending or posts voting (Feeddemon RSS, 2011).

2.1.5. NewsGator

NewsGator is an RSS reader, but it has another very original and useful functionality. It is a synchronization system with the RSS reader, NewsGator Online. It is a Web application to administer your RSS, but that does not need any software, only your Web browser. With NewsGator Online you can administer and support your list of favorite RSS from any computer connected to the Internet. This way, if you go away on a trip and have to use other computers, you will be able to have your RSS list of interest with you. Both the desktop and Web applications share your configurations and RSS Feed's conditions.

2.2. Syndicators inside of Joomla!

We will give an overview of the most valued and downloaded Joomla! Modules 1.5 (free ones) by users (Joomla! 1.5 Extensions, 2011; Joomla! 1.5, 2011; Joomla! API, 2011).

2.2.1. RX RSS ticker module

In this module, you can only subscribe to a Feed source. It has options such as the transition speed controls in the Front-End, and when you place the mouse on the news is when it stops.

Seemingly, it is a simple module whose strengths can stand out in a striking display of the news in our Web, with some control in its configuration. Nevertheless, in contrast, we can state that it only allows for one source, and neither we can see the images, nor the description, nor control the number of words we see, etc. It is a more visual module and it is really functional.

Please cite this article as: de la Torre-Díez I, et al. Development and performance evaluation of a new RSS tool for a Web-based system: RSS_PROYECT. Journal of Network and Computer Applications (2012), <http://dx.doi.org/10.1016/j.jnca.2012.06.004>

ARTICLE IN PRESS

2.2.2. Feed

The next module, which is installed by default in the Joomla! version 1.5. Its principal characteristics from the Back-End are: to allow seeing a description of the news, including its image. In contrast, you cannot select to see only the image of the article, but also those of the feed, producing in many cases a bad visualization, less attractive and professional.

Other cons of this module is that it only offers one feed source, instead of several, and also that it cannot control the content with the number of words on display, which might result in very long titles and overflows in the screen.

2.2.3. RSS Browser 2.7

This module is the first that we have seen that allows syndication of several news. With this, you can subscribe to several news sources. Between its advantages and strengths we find this characteristic, in addition to being able to control the number of visualized articles, as normally happens in the previous ones. But above all, this module adds an important characteristic, and that is possible to visualize the information of the feed (not that of the article), with the supposed problems. It adds another characteristic: if you put the mouse over the news description, the information of the news increase, showing the items description.

2.2.4. Slick RSS

It is one of the most complete modules of all that we have found. Among its principal advantages and characteristics there is the control of images and visualization content, including on the feed, or the control of the number of words visualized in the article. It opens in a new page or in the same one, time configuration of the cache life, no-follow attributes and enables Tooltip. In spite of this, one of the principal disadvantages of this module is that only allows a source to syndicate every time. If you have several sources, it is necessary for them to syndicate one by one, building a module every time.

Regarding the syndicators within Joomla!, RX RSS Ticker uses the Joomla! J Factory methods, which provide higher processing speed with respect to the default FEED module, as the program module (RX RSS Ticker) to run implements a series of additional features that require a higher resource consumption than in the FEED module.

In its favour we find its display options and far more comprehensive and complex settings than in the default FEED module. FEED is found in the Joomla!'s default 1.5 Package. This module uses the libraries from the Joomla! API (JFactory::XML), which enables a similarly fast operation to the use of SimplePie libraries, although it also leads to more malfunctions when syndicating news compared to the use of the SimplePie library, as it features a more basic level of development and is not able to recognise many of the ATOM versions that are currently used to syndicate content. RSS Browser 2.7 is based on the SimplePie library to retrieve the RSS feeds of the news. The main advantages that this library provides to the module are: a similar processing speed to the Joomla! API (due to its optimisation), enhance retrieval of the RSS and ATOM (it supports up to 8 protocols, outperforming Joomla!'s default package) and greater compatibility as it can be integrated into more platforms (apart from Joomla! RSS Browser 2.7), and it allows the user to syndicate several feeds. Conversely, the code of this module is based on nested loops that are always run (in spite of not being required by the Back-End setting), resulting in a slower processing speed compared to the other analysed tools, no matter how it is set up.

Slick RSS uses the SimplePie library, with the advantages mentioned above. It needs to be noted that this library is already

incorporated in the Joomla!'s default platform from the 1.5 version to prove useful and efficient in the news search process. A further benefit is that it is based on PHP, resulting in more user-friendly code programming and reading for programmers in PHP. It uses the functions for the extraction of images, headlines, etc. However, in the structuring of the code, the system executes several functions and algorithms that remain unused, which causes the system to run slowly. A better implementation of the software could make it more efficient by increasing the processing speed.

The difference of the RSS_PROYECT with the rest, is that we have developed 2 blocks of search:

- Generic filter, in which the user typed a single word and that only showed the news that contained this word (several programs exist with this characteristic).
- Specific filter, in which they typed several words and it only showed the news that had the specific set of words (We did not find any program that solved this problem).

3. Methods

The technologies used for the development of the RSS_PROYECT are the following:

PHP. The version used at the moment of programming is PHP 6, in spite that in Joomla! we only need a version of the PHP 5.3, in this way, it would support future improvements without having to rewrite the code.

Database MySQL is a code used by Joomla! It is for making queries in the databases. It has been necessary to use this module, for adding and searching data in the information table of the "black list", used to store the URLs eliminated by the user administrator.

HTML. It is used in the file 'index.php', which is responsible for displaying the processed results. The structures are defined in JOOMLA! and partly programmed in HTML so it's not very big.

XML. Package installer is written in XML, any error of code does not only cause an error, but also it does not allow for the installation of the module. It is a critical part of the code, since besides all the file and roots of the program, it contains the part of the variables that are going to interfere from the Back-End. A bad programming will cause the loss of data or configurations from our Back-End.

Joomla!'s API 1.5.23 (Joomla! API, 2011) is the source from which we obtained all the methods and Joomla! sentences for executing the code correctly and to avoid instabilities and safety gaps in the program (CIAM Website, 2012). The software used for the evaluation is Firebug. It is an extension of Mozilla searcher.

4. Results

Later we will show the results obtained from the front-end part of the user and the performance evaluation.

4.1. RSS_PROYECT module

The RSS_PROYECT module can be define principally as a module capable of subscribing to an unlimited number of feed sources. It can extract all the articles and compare them with a series of filters and a black list stored in our database. This allows us to visualize them or not, as well as to fulfill the criteria marked by the administrator. In addition, we can form a type of users in our module, giving permissions to eliminate content. It goes without saying that a registered user will find tags with which

Please cite this article as: de la Torre-Díez I, et al. Development and performance evaluation of a new RSS tool for a Web-based system: RSS_PROYECT. Journal of Network and Computer Applications (2012), <http://dx.doi.org/10.1016/j.jnca.2012.06.004>

1	he will be able to delete a series of articles that are not interesting for him. By this way, the user has total control.		
3	We will talk about the characteristics of the configuration of the module in two parts: one, the zone of the filters, cache, etc.		
5	The first part consists of a series of configurable parameters. This one is a general sight for creating a perspective on the program.		
7	The zone we are interested in is on the right, which is specific to the RSS_PROYECT. The column on the left side is generic for all the modules and we do not have permissions in this. Then, we will explain the fields in a descending order: two types of filters exist inside the module: generic and selective. Generic filter: it is possible to put an unlimited number of words. This one selects individually each of the words and searches for a correlation within the news title (it is sensitive to capital letters and lower-case letters). When this happens, the news is shown on the screen. Selective Filter: They can put in an unlimited amount of words. The condition for executing it is that they all coincide with the words of the title and with the words of the selective filter.	4.2. Performance evaluation	67
9	These filters are connected between the appearance of one, two, or none; and they have different effects on the module. The specific filter has priority; it is enough if one of the 10 fields has something written and they do not execute the information of the rest. For executing the generic, it is necessary that the specific is empty. In case that two are empty, the number of news will be chosen in order of appearance, as if a normal syndicator was searching itself. Later we will explain briefly the configurable fields of the RSS_PROYECT module.		69
11	In the field "template style for the module," we can select a suffix of cascading style sheet. Thus, the module will be able to take a different style to factory default. To grant elimination permissions to the users for deleting news, we will give some of them a name defined by Joomla! in the Access Control Level (ACL). Moreover, the user that we grant these permissions at the moment of registering will be able to delete news that will go on to the database.		71
13	In Fig. 1, we will find a part of our configuration feed, where we will obtain a wide range control at the moment to configure our module. In the URLs feed, we will be able to put all of our feeds that we want to subscribe to; the only requirement is that we must make a separation space, because the program can add different URLs. To visualize the source feed we will mark the field that permits seeing the newspaper name, the Web site where we extract the information. It appears only once, and all of its articles below. With the description of the feed we can visualize the information in the newspaper, Web, etc.		73
15	In the description that it has stored. By activating the field <i>Feed Image</i> we see the image of the source feed. To control the <i>Number of articles seen</i> , we select a numeric value, which will be the number of articles that will appear. In the case that it is near the limit, the program will stop and will show this number of articles on the screen. And if it does not come near to the established limit, the program will execute until the end and will show the obtained results. With the field <i>X words of the article to see</i> we can decrease the limit of the number of words that will appear in the title. In <i>Paper Description</i> we can enable or disable the item description, in case that there is one.		75
17	In <i>X words of the article description</i> , you delimit the number of words that you want and they will appear after the link of the article. It is necessary that this is activated in the previous field in order for this one to be executed. With the field <i>See images for each article</i> you can enable individually the article that has images, they are later seen on the link (the majority of the RSS do not incorporate images, because it is slightly probable that they could be visualized).		77
19	Finally, we will speak about the <i>Tooltip</i> . It is a tool that activates after placing the mouse over the news icon, and with it we can activate different functions. The field <i>Entitled Tooltip</i> allows us to activate or to deactivate this function. The following fields will not be affected if this option is not activated. With <i>X Words of Tooltip</i> it allows you to form the number of words to visualizing of the news title in the <i>Tooltip</i> . The field <i>X Words of Tooltip description</i> forms the number of the description words that is going to be seen.		79
21			81
23			83
25			85
27			87
29			89
31			91
33			93
35			95
37			97
39			99
41			101
43			103
45			105
47			107
49			109
51			111
53			113
55			115
57			119
59			121
61			123
63			125
65			127
			129
			131
			133

1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
33
35
37

URLS feed

http://www.xtve.es/rss/temas_noticias.xml
 http://www.noticias.com/rss/portada
 http://www.20minutos.es/rss/
 http://www.nortecastilla.es/portada.xml
 http://www.abc.es/rss/feeds/abcPortada.xml
 http://www.elpais.com/rss/feed.html?feedid=1022
 http://elmundo.feedsportal.com/elmundo/rss/portada.xml
 http://www.laazon.es/rss

Visualizar la fuente feed Si No

Descripción del feed Si No

Imagen Feed Si No

Numero de articulos a visualizar

X palabras del titulo del articulo

Descripción del articulo Si No

X palabras de la descripción del articulo

Ver las imagenes de cada articulo Si No

Abrir la noticia En una ventana nueva En la misma ventana

Añade el atributo 'no follow' a los enlaces No Si

67
69
71
73
75
77
79
81
83
85
87
89
91
93
95
97
99
101
103

Fig. 1. Back-End.

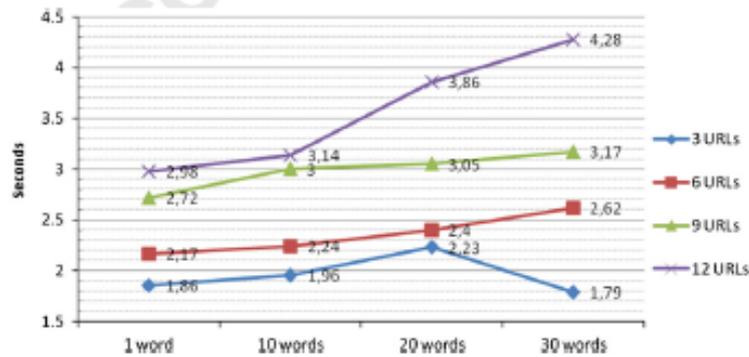


Fig. 2. Response times for generic filter.

57
59
61
63

to the generic one. The final average of this selective filter is 8.4 s, which represents almost 6 s slower loading that without the module. This is why we should consider carefully the number of selective filters and selected sources, since their implementation becomes critical when navigating.

65

The relevance of the search engine RSS_PROYECT lies in its ability to provide faster searching compared to the tools analysed within Joomla! 1.5, thereby optimising processing speed and adding news management and setup functions. This tool gives

the administrator full control over the publication of articles, which allows him/her to delete articles found by the search engine (they are stored in the database known as black_list) but that he/she considers that they should not be published on the Web. It also enables attribute assignment (display, hide, remove) depending on the user hierarchy (administrator, superadministrator, manager, publisher...), differentiating it from the rest of tools in its field. Fig. 4 shows better average times for RSS_PROYECT with respect to other modules analysed in Joomla!.

Please cite this article as: de la Torre-Díez I, et al. Development and performance evaluation of a new RSS tool for a Web-based system: RSS_PROYECT. Journal of Network and Computer Applications (2012), <http://dx.doi.org/10.1016/j.jnca.2012.06.004>

ARTICLE IN PRESS

L. de la Torre-Díez et al. / Journal of Network and Computer Applications (2012) 35–40

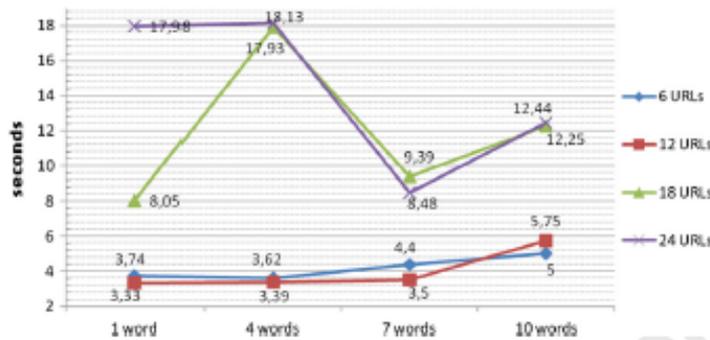


Fig. 3. Response times for specific filter.

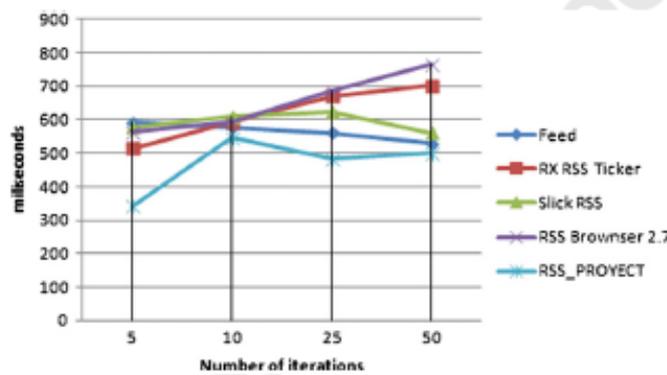


Fig. 4. Average times for different Joomla! modules.

5. Conclusions and future work

One of the conclusions achieved is that this module is successful in the Web, as there is nothing similar in Joomla! (at least free). For this, it would be interesting to be able to publish in the official page of Joomla!, so that users could download and adapt it to their needs. It has two types of filters, selective and generic, which constitutes a powerful tool and clearly a distinct characteristic, because this module is different from the other Joomla! modules. RSS_PROYECT provides solutions to problems that other RSS tools are not able to solve. This makes the RSS_PROYECT different from other tools.

In relation with the efficiency of the filters, the conclusions are: when we increase the number of words in the filters we also increase the processing time, as in the case of the URLs.

In conclusion, the study conducted about the speed execution has confirmed what we felt at the moment of programming the algorithms of the module. When the number of feed sources increases, the computational efficiency increases too. It is necessary to emphasize the singularity of the selective filter. We must bear in mind the random factor, as the flow chart of RSS_PROYECT is optimized. Therefore, in the moment that it achieves the goals it stops processing (it is the singularity observed in the selective filter, where by increasing the number of URLs the goals are reached and it does not explore all the possible iterations).

The security levels implemented are the maximum that can be implemented today within a module, so we ensure that our tool will not be a security gap within our site and we will be well

protected. This is because the commands have been used through the API of Joomla!, so the module is not vulnerable.

In the technical aspect, we have observed that the computational efficiency posed by the algorithms of the module is low, and very small (400 kb), ideal for a mobile device. (It is important to point out that for the case of selective filters, with more than 20 feeds, the computational efficiency is significant. It would be necessary to conduct a study on the convenience for both fixed devices and mobiles). In spite of being a complete tool within Joomla! modules, there is always room for an improvement in the future. Next, we propose some possible future lines of study and their explanation:

Added news: It would be interesting to include an option in which the administrator could add news, since the program performs a few searches on the Web that you have indicated, but you might want to add something else.

Scale of values of the news: Another improvement that could be done is that if, for example, news are updated slowly, the user would be able to rate them and, for example, the news that were out of date could be deleted.

Algorithm to eliminate news on the database: Despite the eliminated articles do not pose a significant size in bits, neither in memory, nor in execution, a timing device could be installed for the news that are stored in the database (Black_List), because throughout the years they would increase the calculation of the program and it might go slower (remember that the news are being updated, so they are also being removed and it does not make sense to continue storing them).

Please cite this article as: de la Torre-Díez L, et al. Development and performance evaluation of a new RSS tool for a Web-based system: RSS_PROYECT. Journal of Network and Computer Applications (2012), <http://dx.doi.org/10.1016/j.jnca.2012.06.004>

ARTICLE IN PRESS

1 Search algorithms based on the number of times that a word is
2 repeated in an article, but we must bear in mind the environment
3 where our tool is being used, as the Web scraping is carried out on
4 a series of sites in RSS format indexed by the user.

5 Infinite selective filters: In our case we have made 10 selective
6 filters, with this we are meeting the needs of our client, but
7 circumstances may arise where a user needs even more. This could
8 be a possible line of study and work. Improvements to the searcher:
9 We might improve the algorithms of search so that they are capable
10 of looking for words of other alphabets, or interpreting words with
11 question marks, exclamation marks, etc. providing them with an
12 artificial intelligence to distinguish this type of symbols.

13
14
15 **Acknowledgements**

16 The authors wish to express our gratitude to the Comprehensive
17 Center for Women's Help (*Centro Integral de Ayuda a la Mujer*,
18 CIAM) of Valladolid (Spain) for their collaboration in this work.

19 Part of this work has been supported by the *Instituto de*
20 *Telecomunicações*, Next Generation Networks and Applications
21 Group (NetGNA), Portugal and by National Funding from the
22 FCT—*Fundação para a Ciência e a Tecnologia*.

23
24
25 **References**

26 Bouras C, Pouloupoulos V, Tsoykas V. Adaptation of RSS feeds based on the user
27 profile and on the end device. *Journal of Network and Computer Applications*
28 2010;22:410–21.
29 CIAM Website. <http://www.josefinas-trinitarias.org/jam/arex/index.php?option=com_content&view=article&id=7&Itemid=8> (last accessed: May 2012).

30 Btivill J. Concepts and Strategies for Combating Social Exclusion: An Over-
31 view Portugal: International Labour Office; 2003.
32 Feedem on RSS. <<http://www.desarrolloWeb.com/articulos/feedem-on-rss/>
33 (last accessed: June 2011).
34 Google Reader RSS. <http://es.wikipedia.org/wiki/Google_Reader>, (last accessed:
35 June 2011).
36 Hull B. ICT and social exclusion: the role of libraries. *Telematics and Informatics*
37 2003;20:131–42.
38 Joomla! 1.5 Extensions. <[http://extensions.joomla.org/extensions/news-display/](http://extensions.joomla.org/extensions/news-display/rss-readers)
39 [rss-readers](http://extensions.joomla.org/extensions/news-display/rss-readers)>, (last accessed: June 2011).
40 Joomla! 1.5. <[http://www.joomla.org/actualidad/200-ediciones-del-pack-](http://www.joomla.org/actualidad/200-ediciones-del-pack-joomla-spanish-y-joomla-423-liberada-joomla-1524-senu-talia-ama-nalid-)
41 [joomla-spanish-y-joomla-423-liberada-joomla-1524-senu-talia-ama-nalid-](http://www.joomla.org/actualidad/200-ediciones-del-pack-joomla-spanish-y-joomla-423-liberada-joomla-1524-senu-talia-ama-nalid-)
42 [html](http://www.joomla.org/actualidad/200-ediciones-del-pack-joomla-spanish-y-joomla-423-liberada-joomla-1524-senu-talia-ama-nalid-)>, (last accessed: May 2011).
43 Joomla! API. <<http://api.joomla.org/joomla-Framework/Application/ApiApplication.html>
44 (last accessed: June 2011).
45 Ministry of Health, Social Policy and Equality, Spanish Government. I. Annual
46 report about violence, 2007.
47 Ning X, Jin H, Wu H. RSS: a framework enabling ranked search on the semantic
48 web. *Information Processing and Management* 2008;44:893–909.
49 Netvibes RSS. <[http://faq.netvibes.com/what_can_i_use_with_my_netvibes_page/](http://faq.netvibes.com/what_can_i_use_with_my_netvibes_page/rss_or_web_feeds/)
50 [rss_or_web_feeds](http://faq.netvibes.com/what_can_i_use_with_my_netvibes_page/rss_or_web_feeds/)>, (last accessed: June 2011).
51 Samper JJ, Castillo PA, Araujo I, Merino JJ, Cordon O, Tricas F. NectarRSS, an
52 intelligent RSS feed reader. *Journal of Network and Computer Applications*
53 2008;31:799–806.
54 United Nations Development Programme. Human development report, making
55 new technologies work for human development, 2001.
56 Warren M. The digital vicious cycle: links between social disadvantage and digital
57 exclusion in rural areas. *Telecommunications Policy* 2007;31:374–88.

Please cite this article as: de la Torre-Díez I, et al. Development and performance evaluation of a new RSS tool for a Web-based system: RSS_PROYECT. *Journal of Network and Computer Applications* (2012), <http://dx.doi.org/10.1016/j.jnca.2012.06.004>

ANEXO I

Testeo del módulo RSS_PROYECT

A continuación mostraremos las capturas de pantalla de donde hemos obtenido los datos de velocidad de procesado del módulo a través del programa FIREBUG.

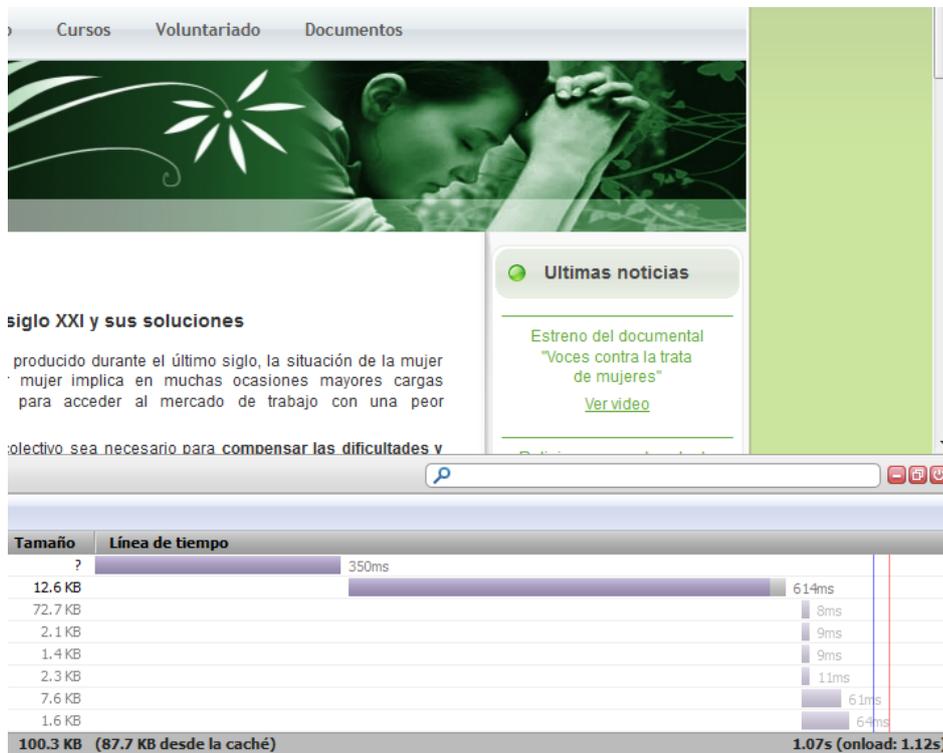


Figura ANEXO I genérico: Ejemplo sin el módulo ejecutándose

Todas las gráficas mostradas a continuación tienen en común que en el Back-End está configurado para 100 artículos a visualizar. Se trata del filtro genérico, e iremos indicando el número de fuentes feed utilizadas, así como el número de palabras que se han introducido en dicho filtro:

A) Con 12 feed en la casilla URLs:

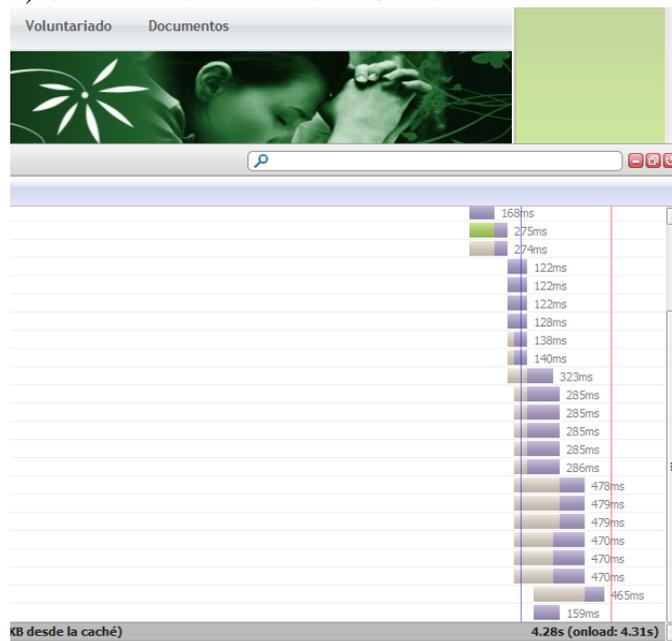


Figura ANEXO I, a30: Ejemplo con el módulo y 30 palabras.

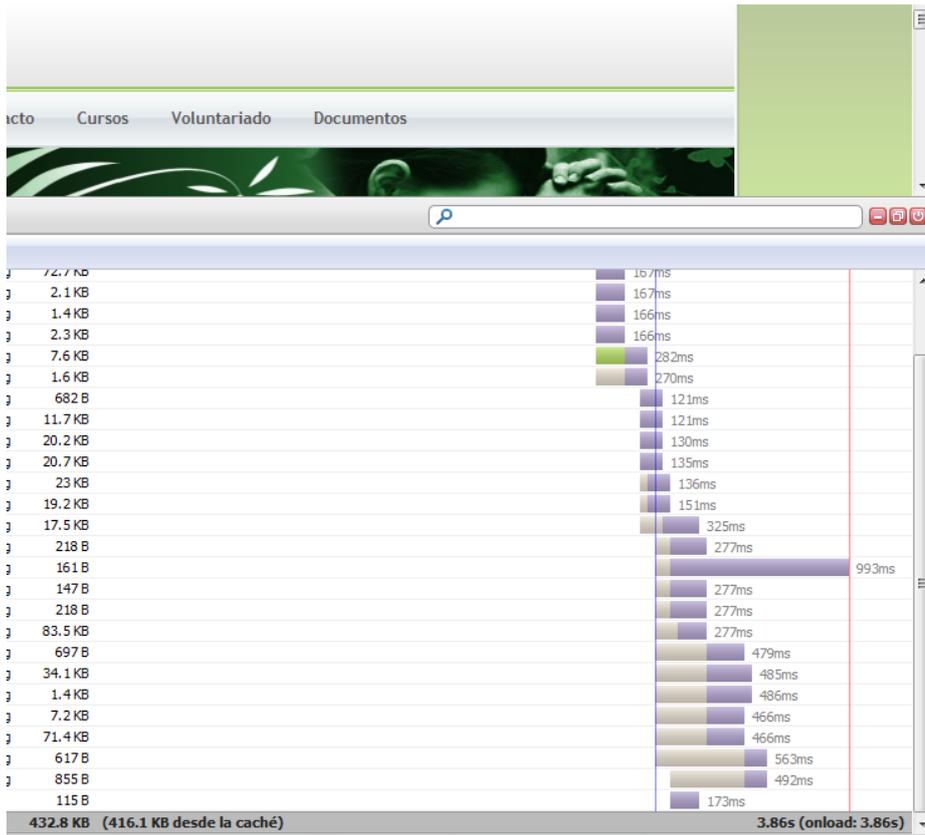


Figura ANEXO I, a20: Ejemplo con el módulo y20 palabras

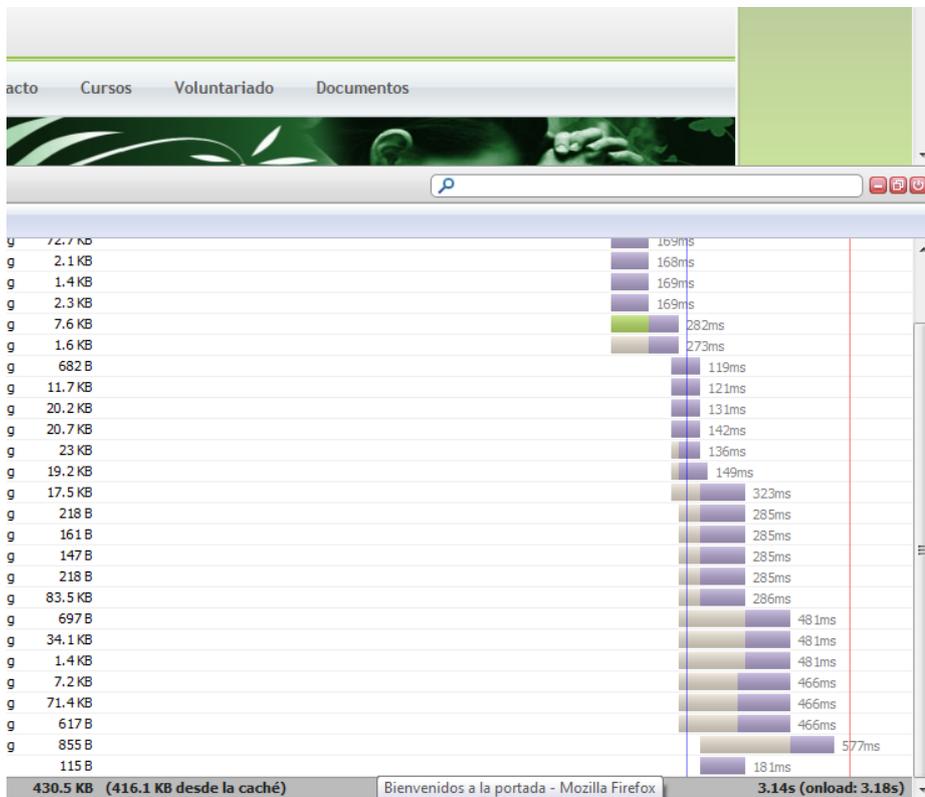


Figura ANEXO I, a10: Ejemplo con el módulo y10 palabras

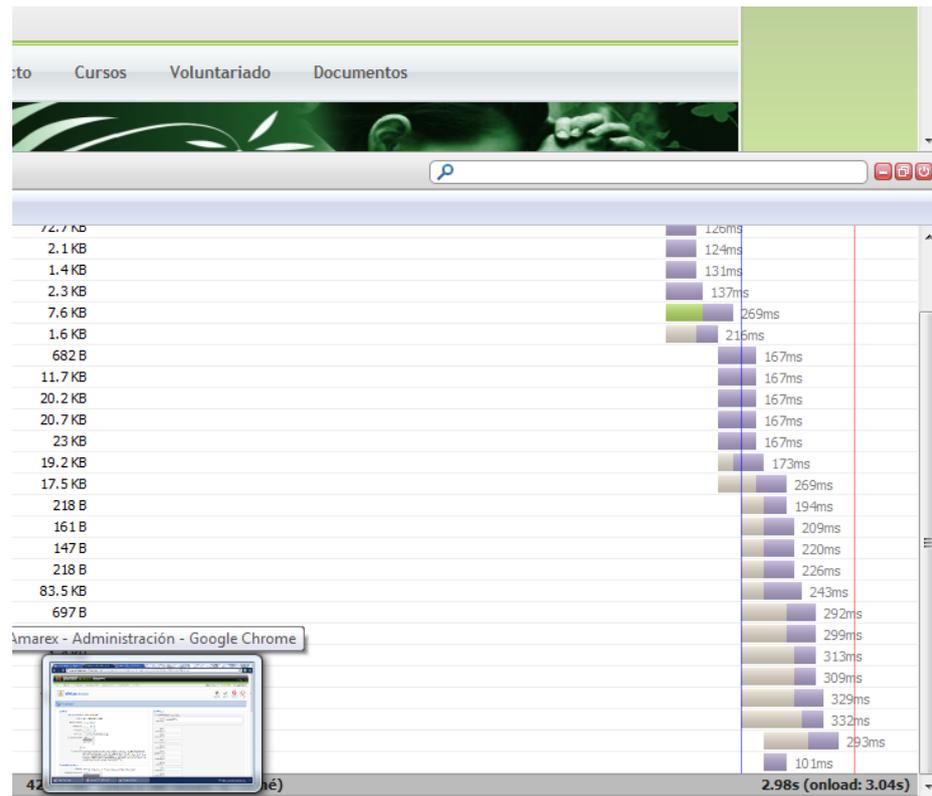


Figura ANEXO I, a1: Ejemplo con el módulo y 1 palabra

B) Con 9 feed en la casilla del Back-End URLs:

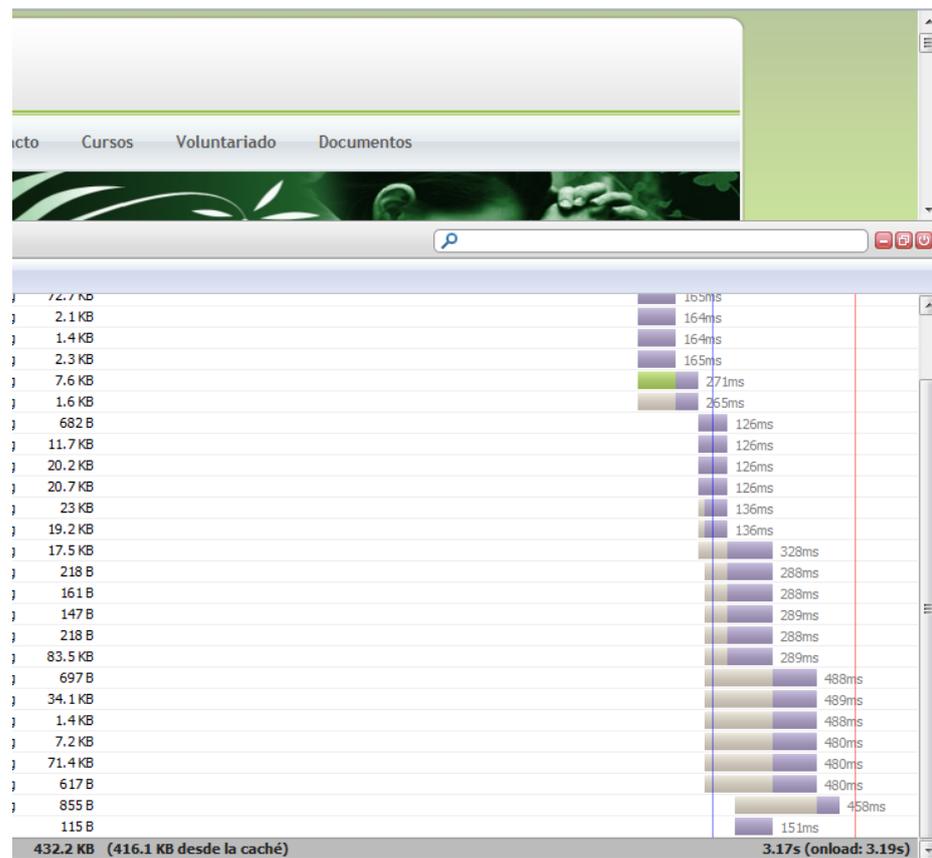


Figura ANEXO I, b30: Ejemplo con el módulo y 30 palabras

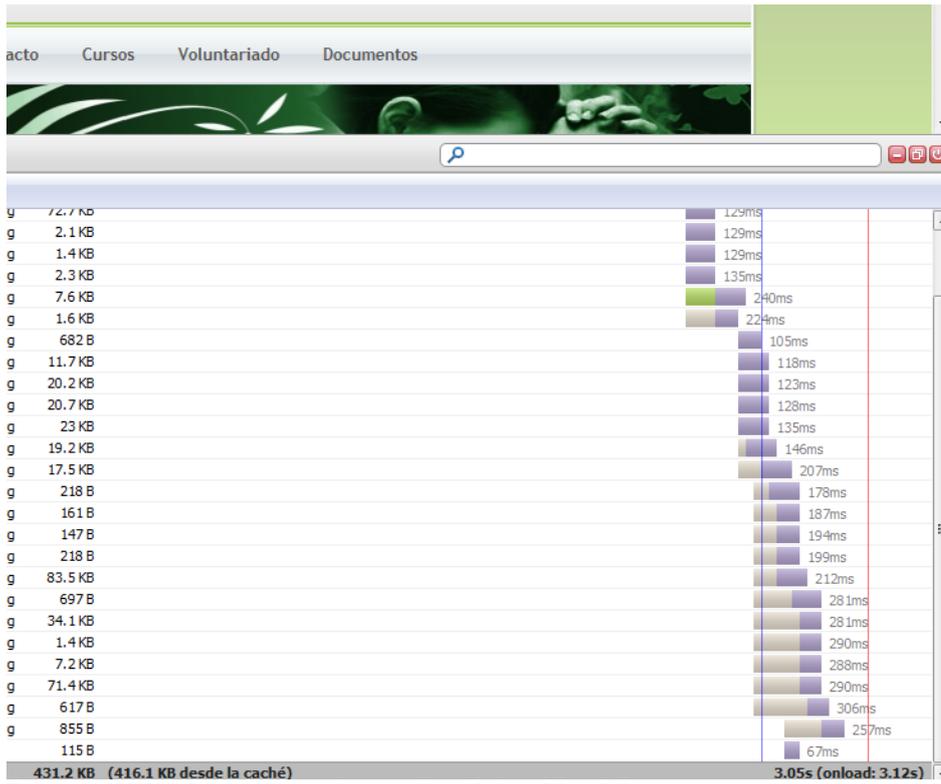


Figura ANEXO I, b20: Ejemplo con el módulo y 20 palabras

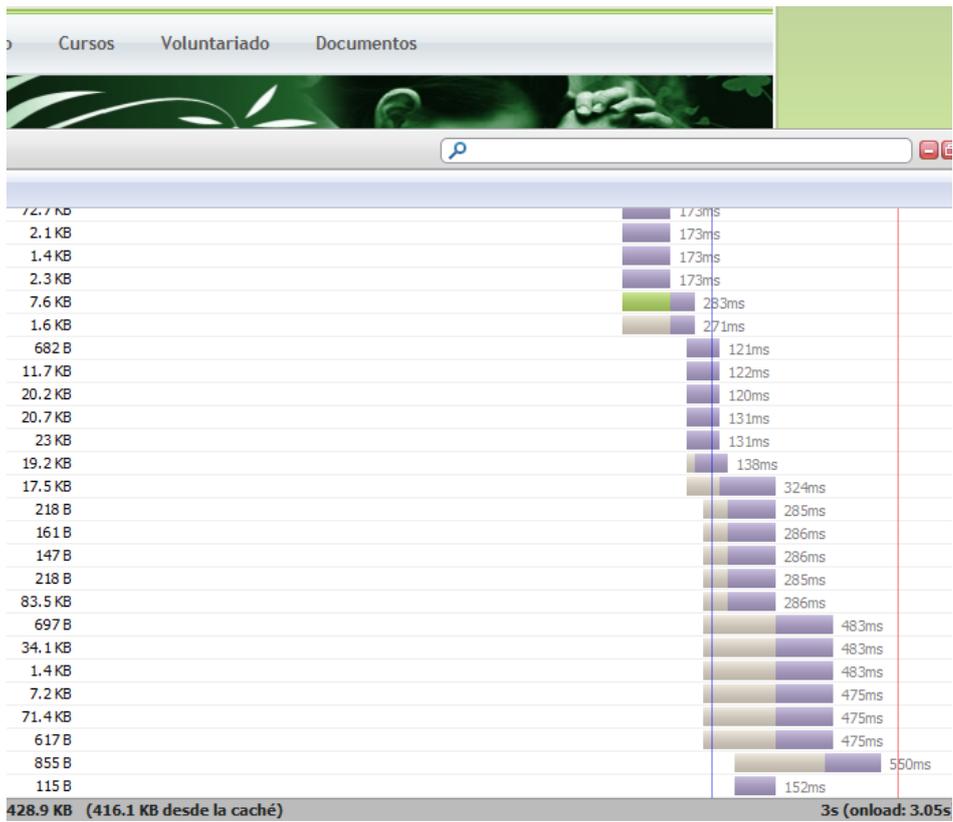


Figura ANEXO I, b10: Ejemplo con el módulo y 10 palabras

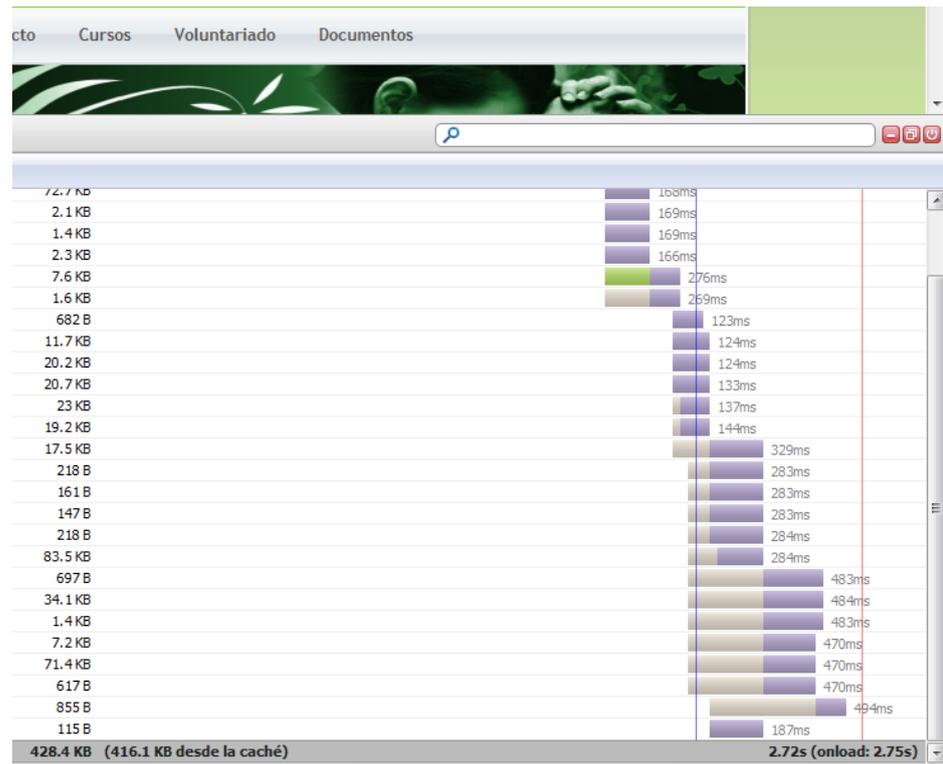


Figura ANEXO I, b1: Ejemplo con el módulo y 1 palabra

C) Con 6 feed en la casilla URLs:

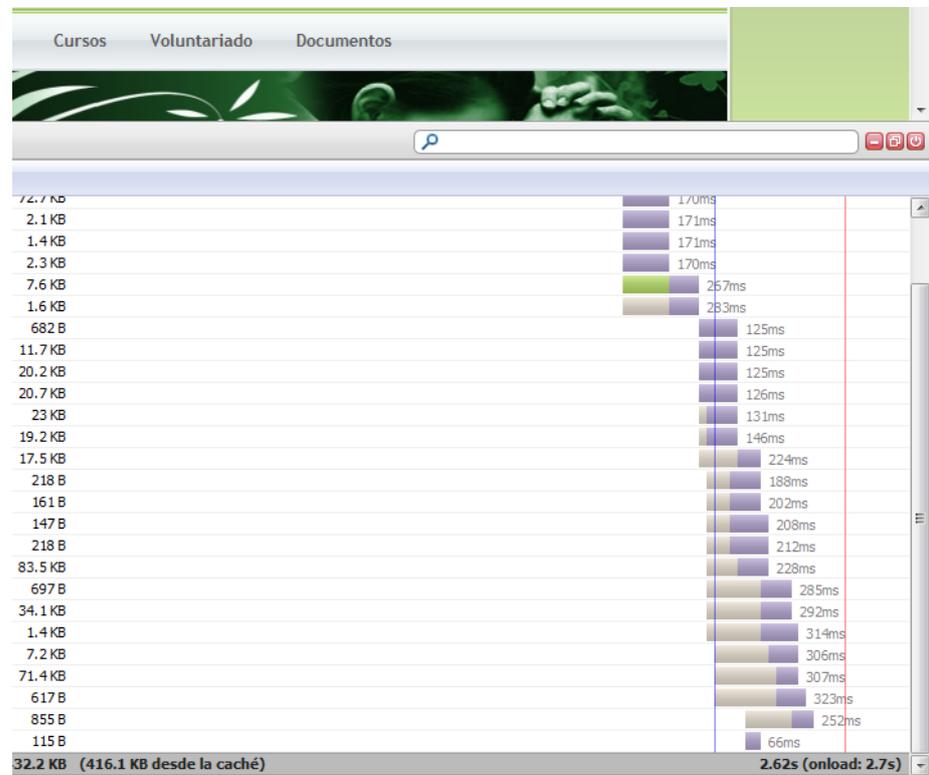


Figura ANEXO I, c30: Ejemplo con el módulo y 30 palabras

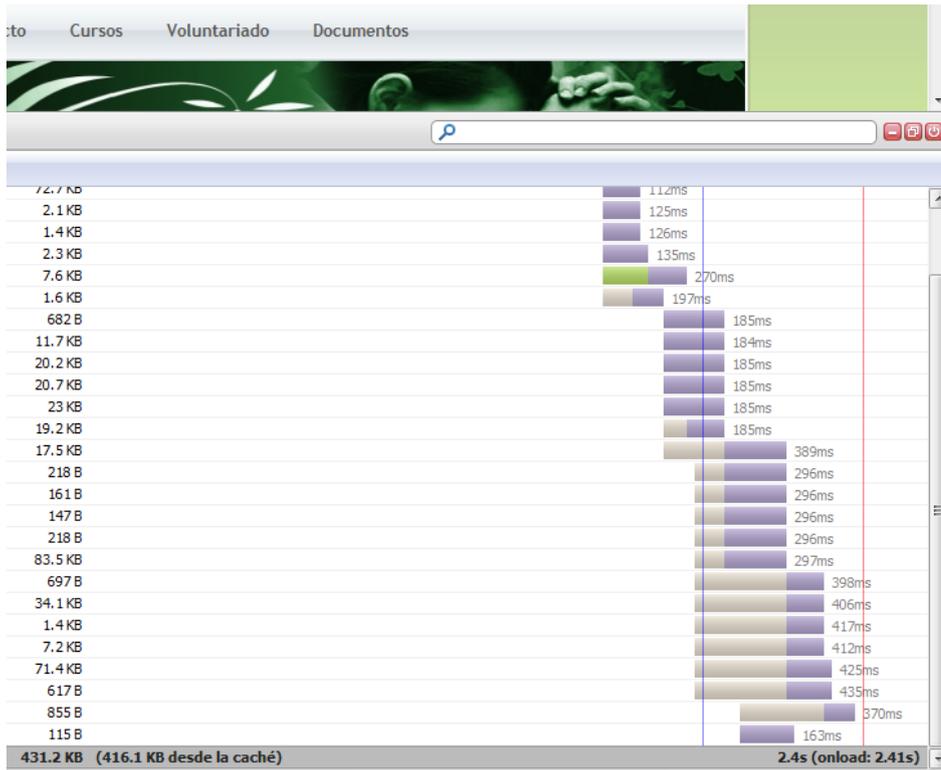


Figura ANEXO I, c20: Ejemplo con el módulo y 20 palabras

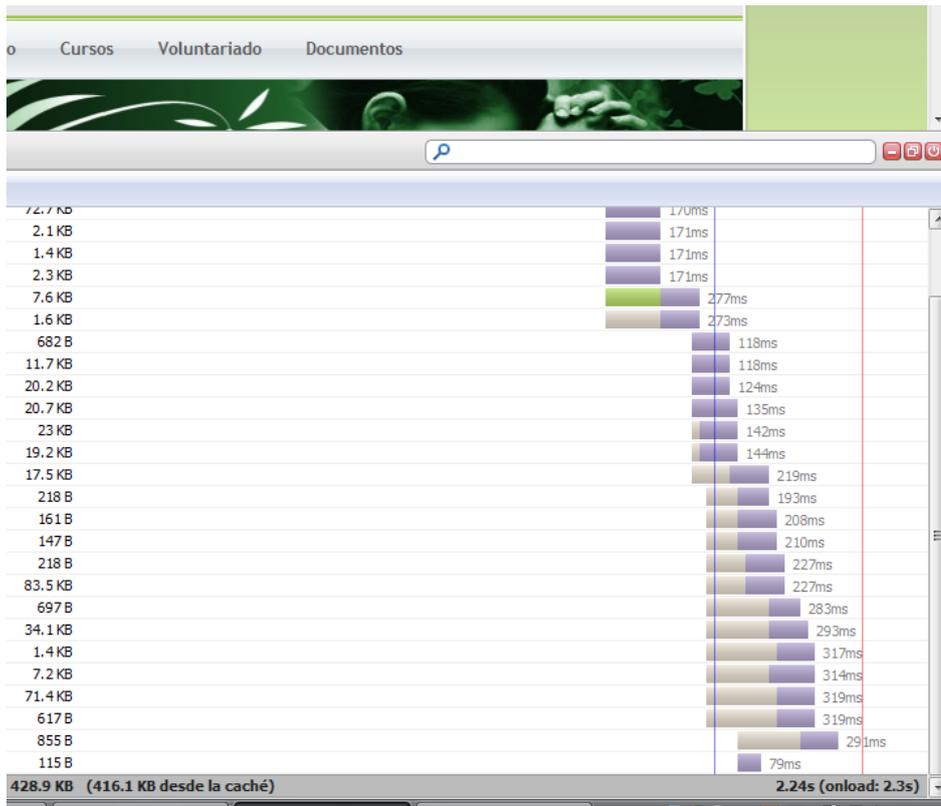


Figura ANEXO I, c10: Ejemplo con el módulo y 10 palabras

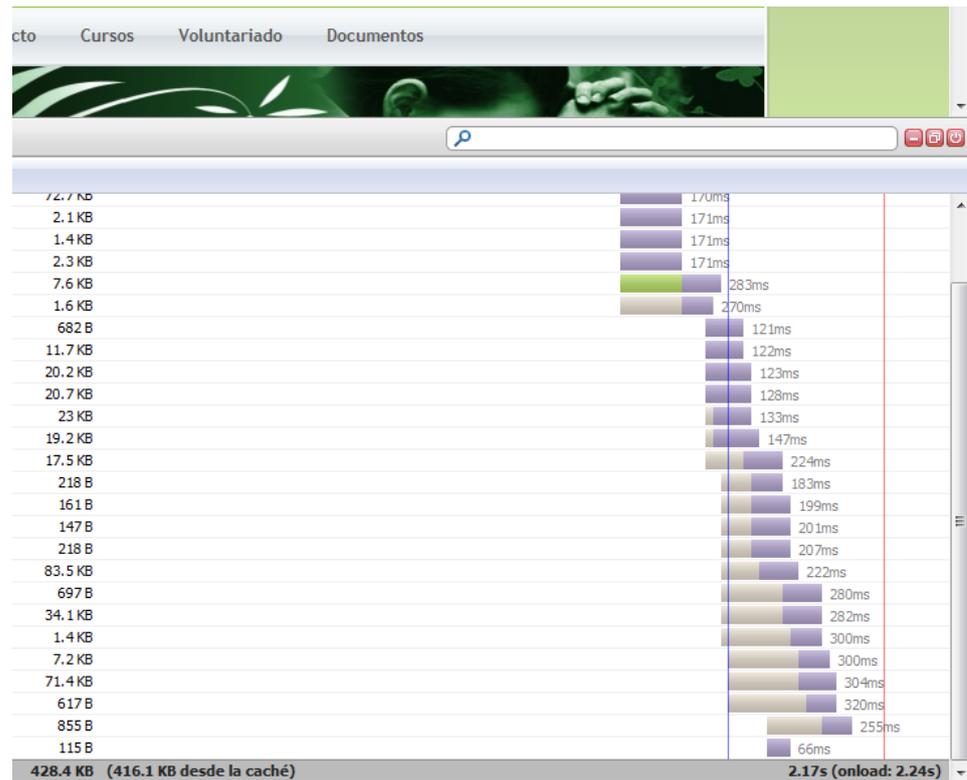


Figura ANEXO I, c1: Ejemplo con el módulo y1 palabra

D) Con 3 feed en la casilla Back-End URLs:

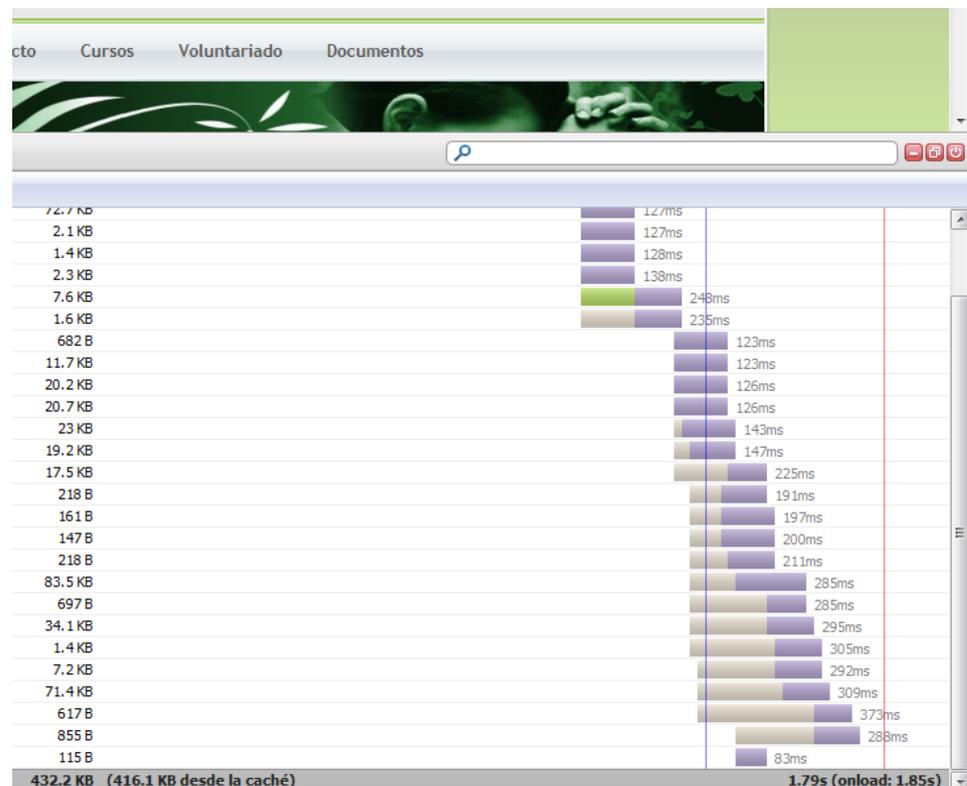


Figura ANEXO I, d30: Ejemplo con el módulo y 30 palabras

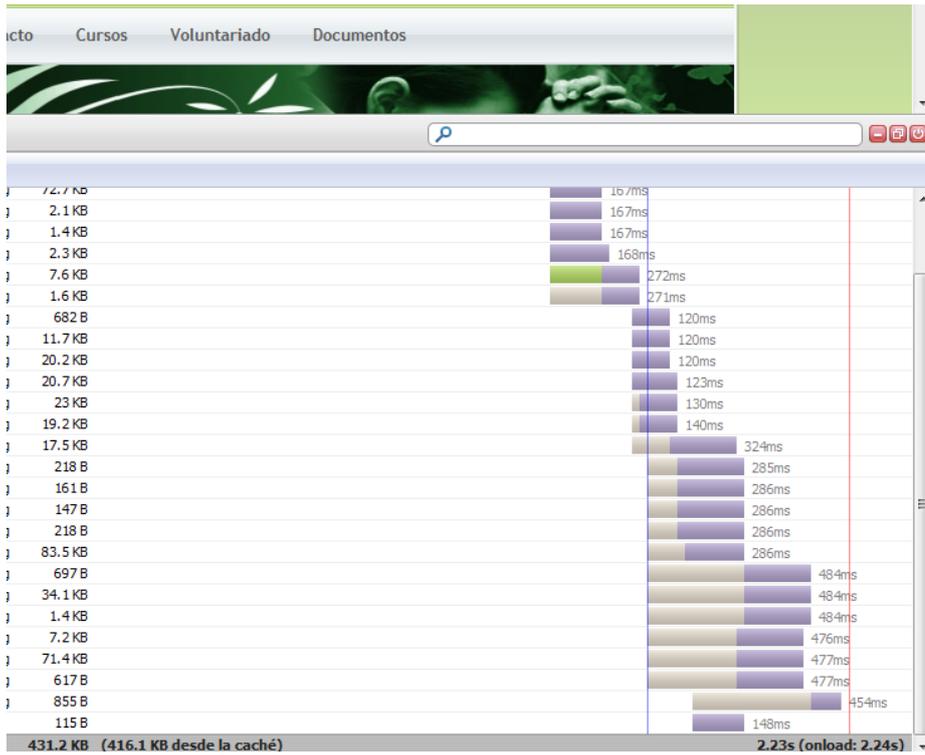


Figura ANEXO I, d20: Ejemplo con el módulo y 30 palabras

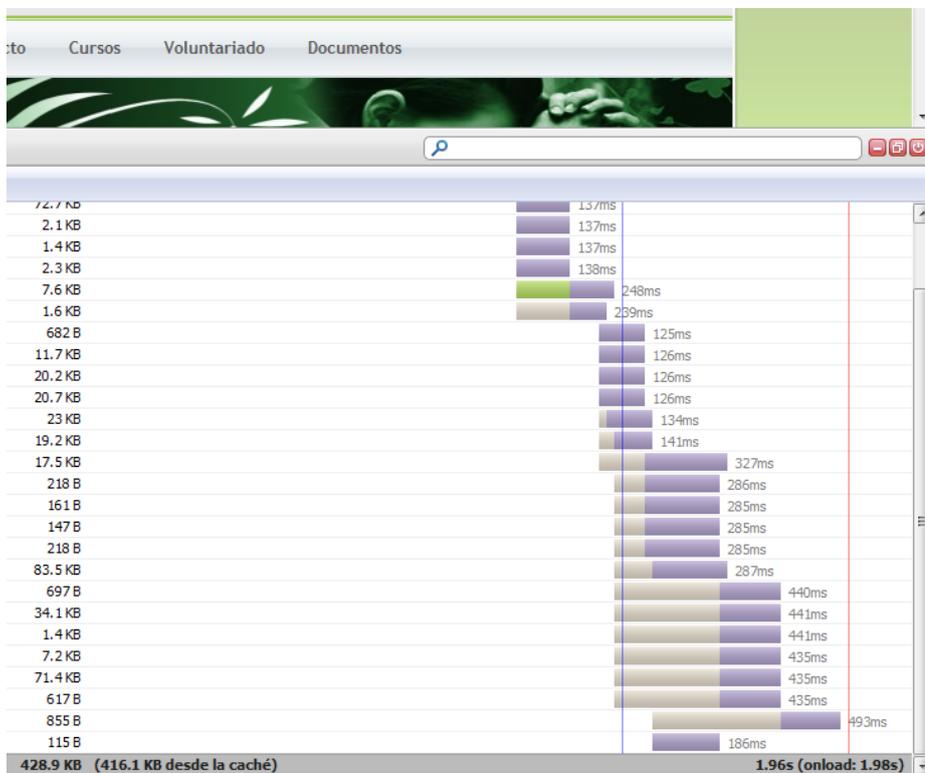


Figura ANEXO I, d10: Ejemplo con el módulo y 30 palabras

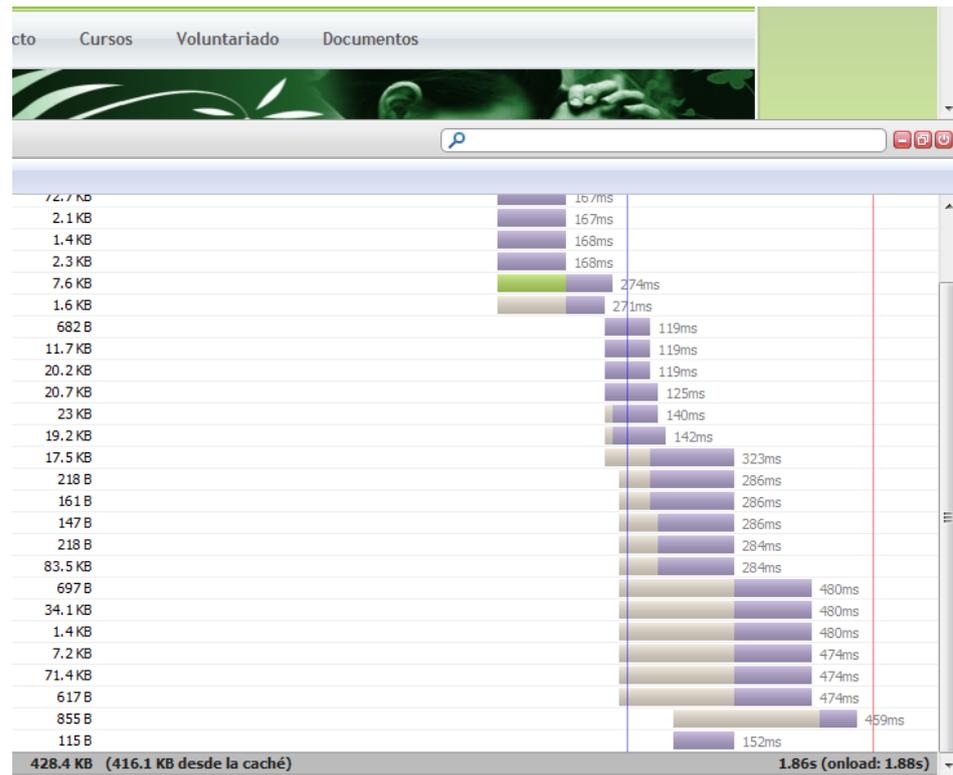


Figura ANEXO I, d1: Ejemplo con el módulo y 1 palabra

A continuación, realizaremos los mismos pasos, pero para el buscador específico. Primero realizaremos un test de prueba para ver cómo funciona. Debemos tener en cuenta que se realiza para la búsqueda de 5 artículos en este caso.

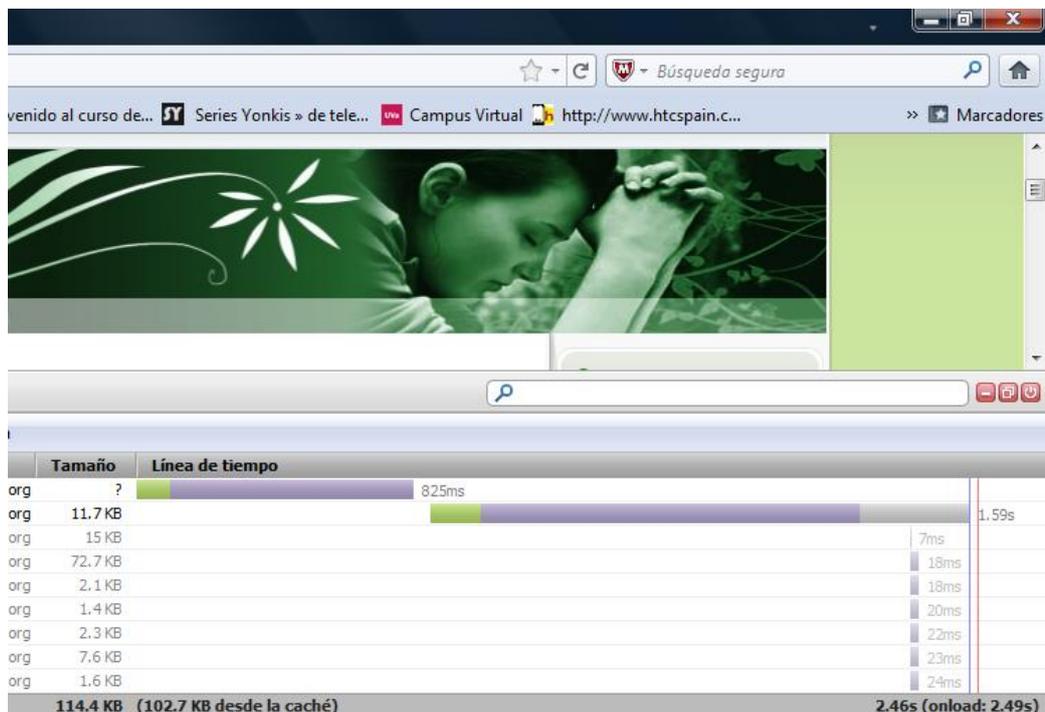


Figura ANEXO I selectivo: prueba sin activar el módulo

A continuación comenzaremos el testeo con diferentes feed y palabras filtro:

A) Con 24 feed Rss:

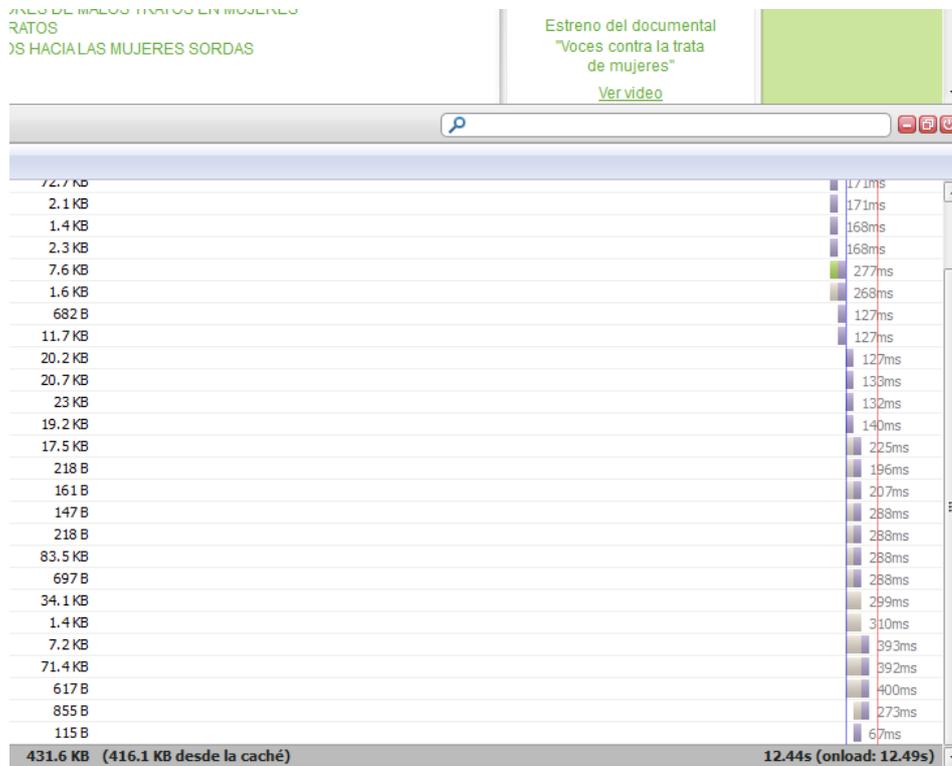


Figura ANEXO I, selec. A10: Ejemplo con el módulo y 10 filtros

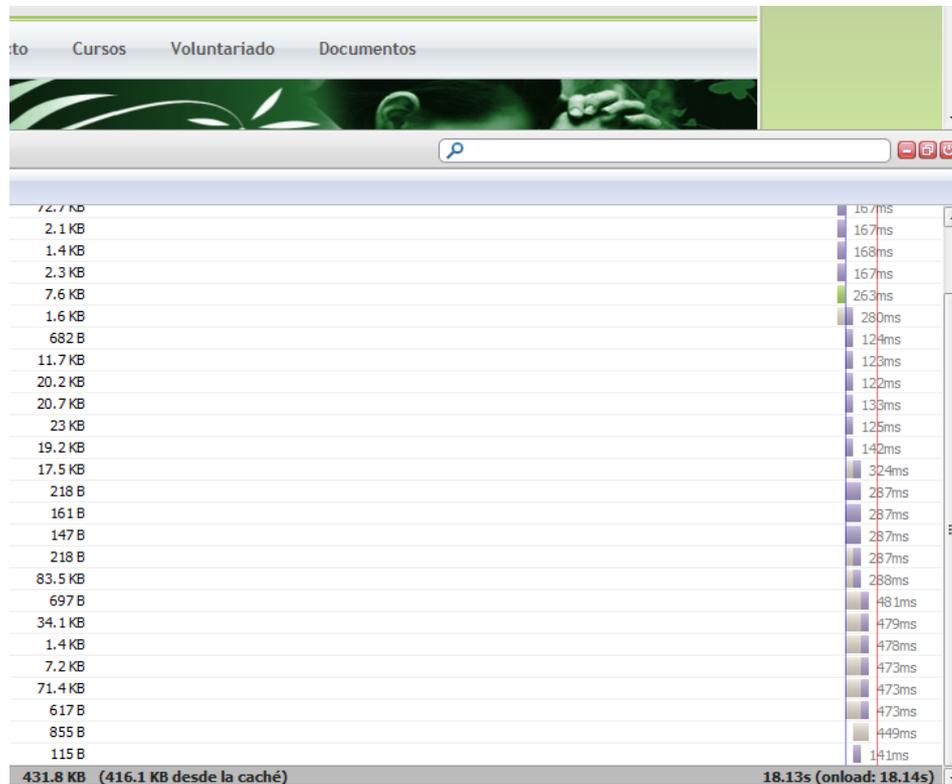


Figura ANEXO I, selec. A7: Ejemplo con el módulo y 7 filtros

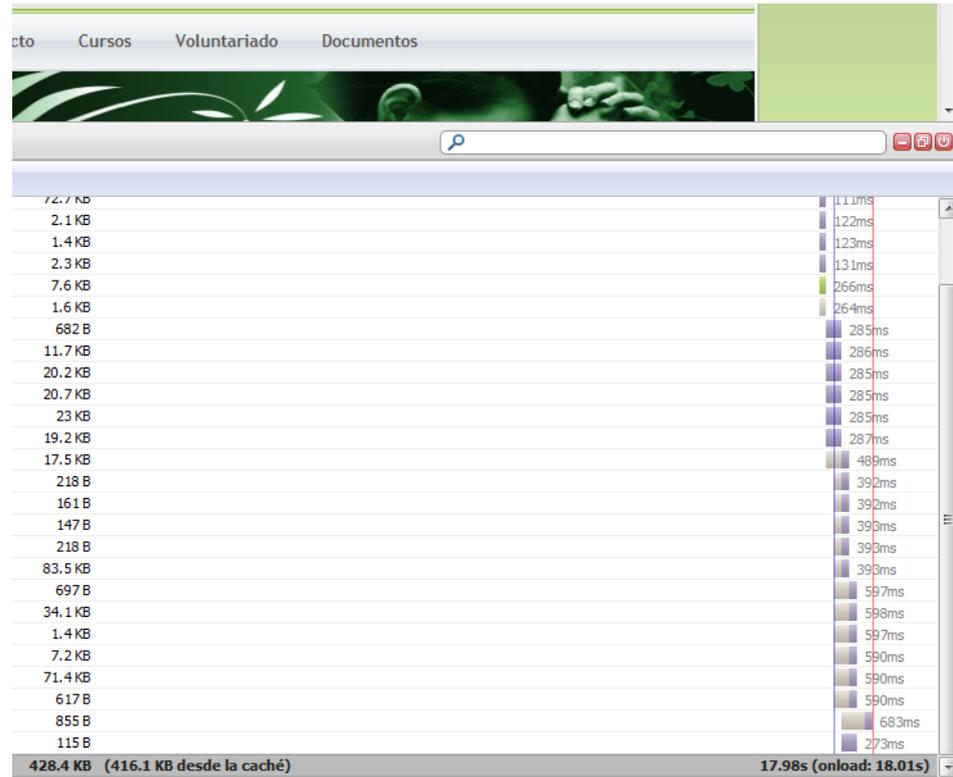


Figura ANEXO I, selec. A4: Ejemplo con el módulo y 4 filtros

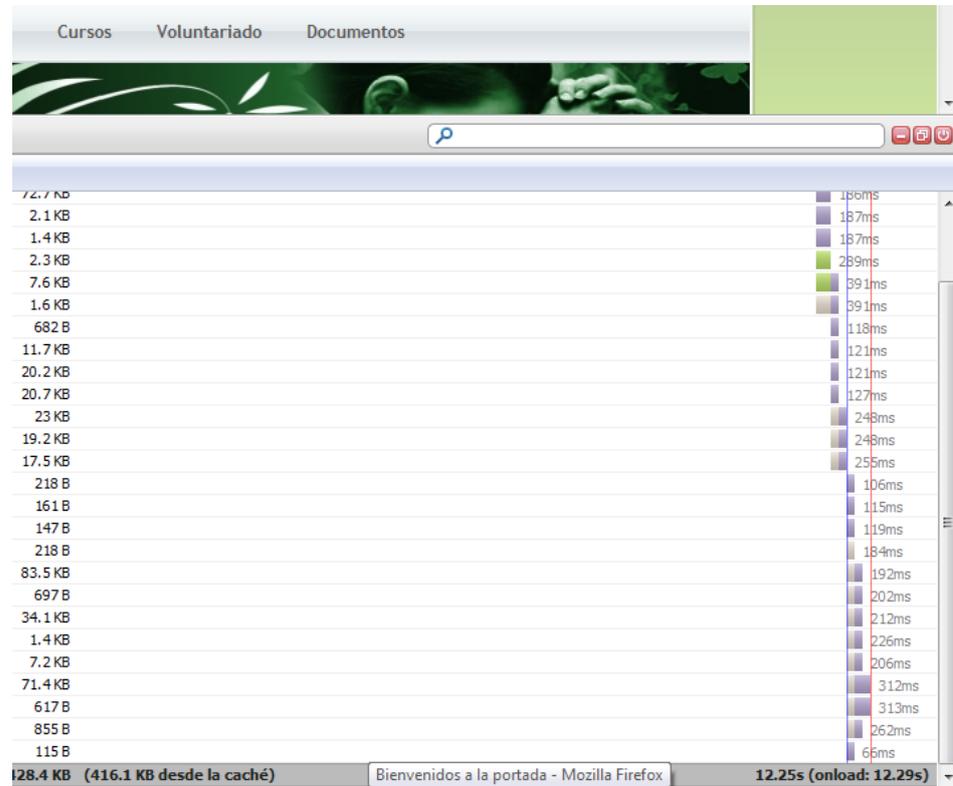


Figura ANEXO I, selec. A1: Ejemplo con el módulo y 1 filtro

B) Con 18 feed URLs:

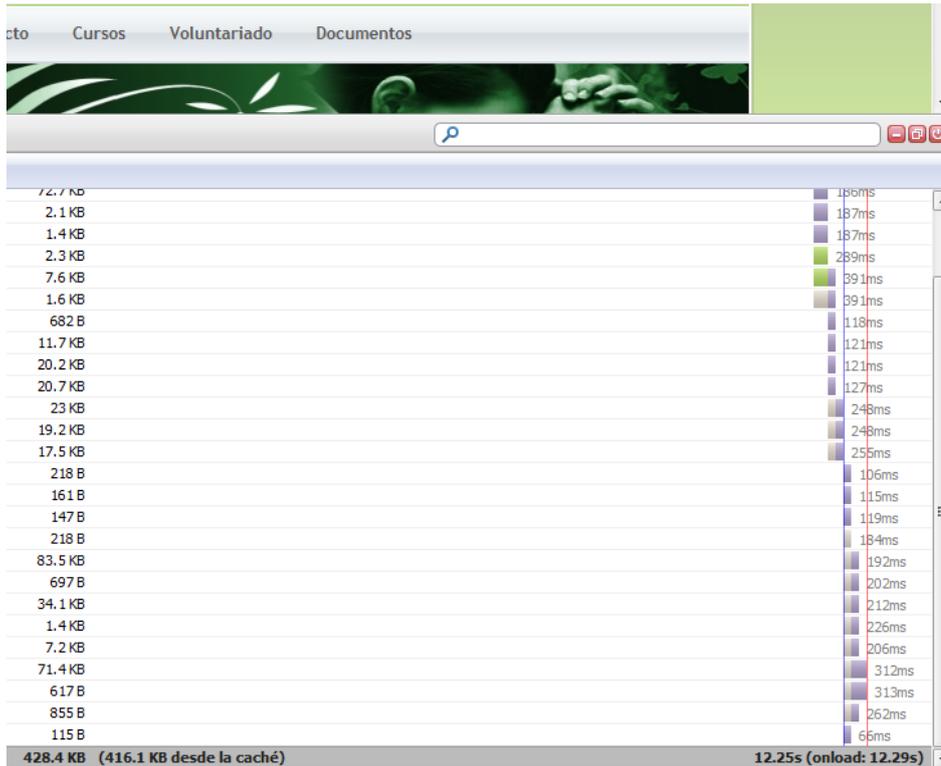


Figura ANEXO I, selec. B10: Ejemplo con el módulo y 10 filtros

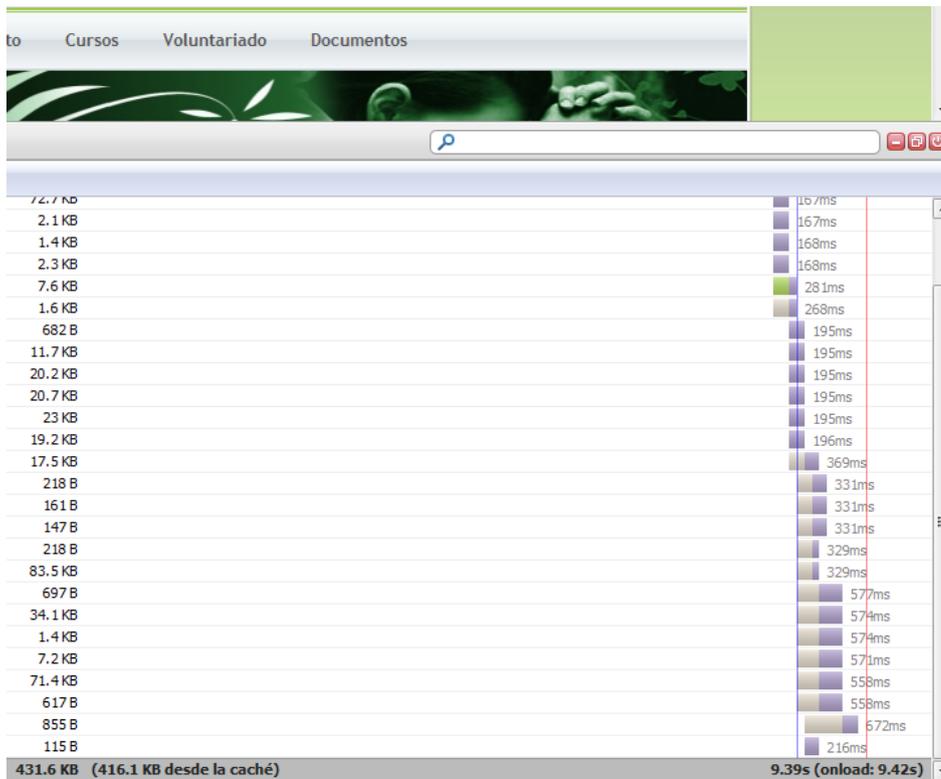


Figura ANEXO I, selec. B7: Ejemplo con el módulo y 7 filtros

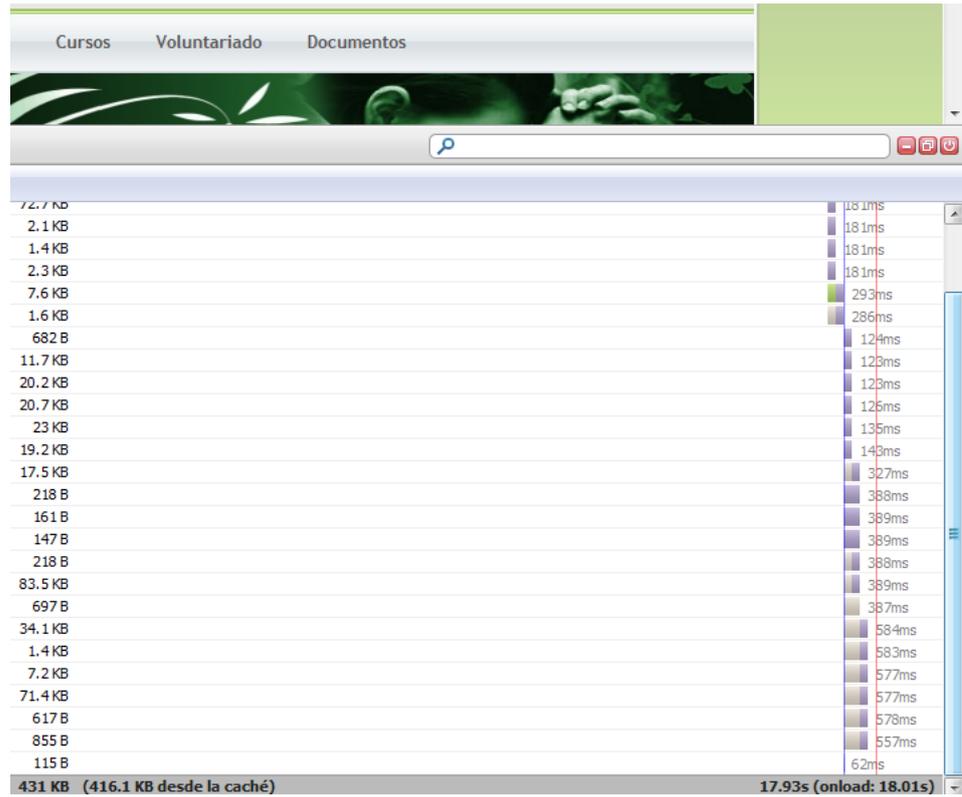


Figura ANEXO I, selec. B4: Ejemplo con el módulo y47 filtros

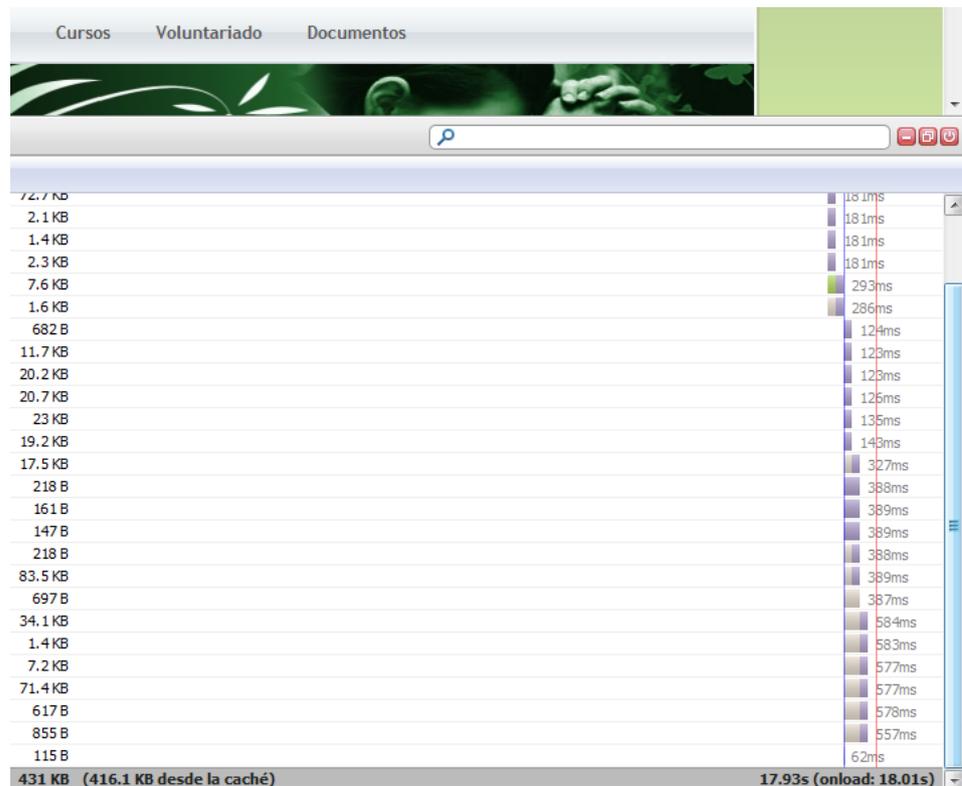


Figura ANEXO I, selec. B1: Ejemplo con el módulo y 1 filtro

C) Con 12 feed URLs Back-End:

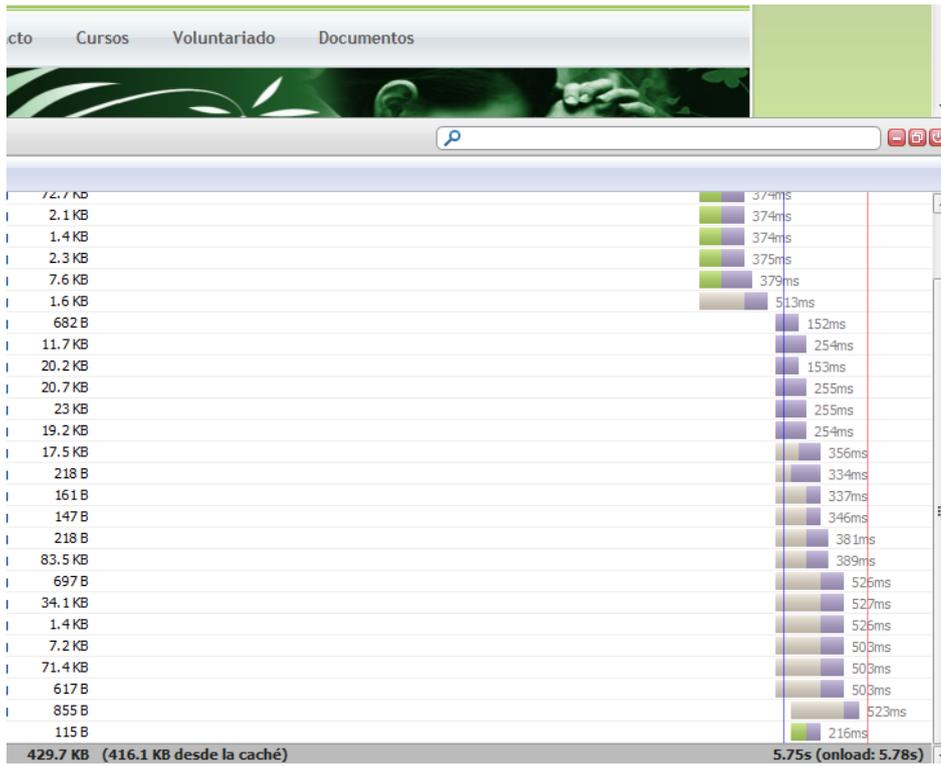


Figura ANEXO I, selec. C10: Ejemplo con el módulo y 10 filtros

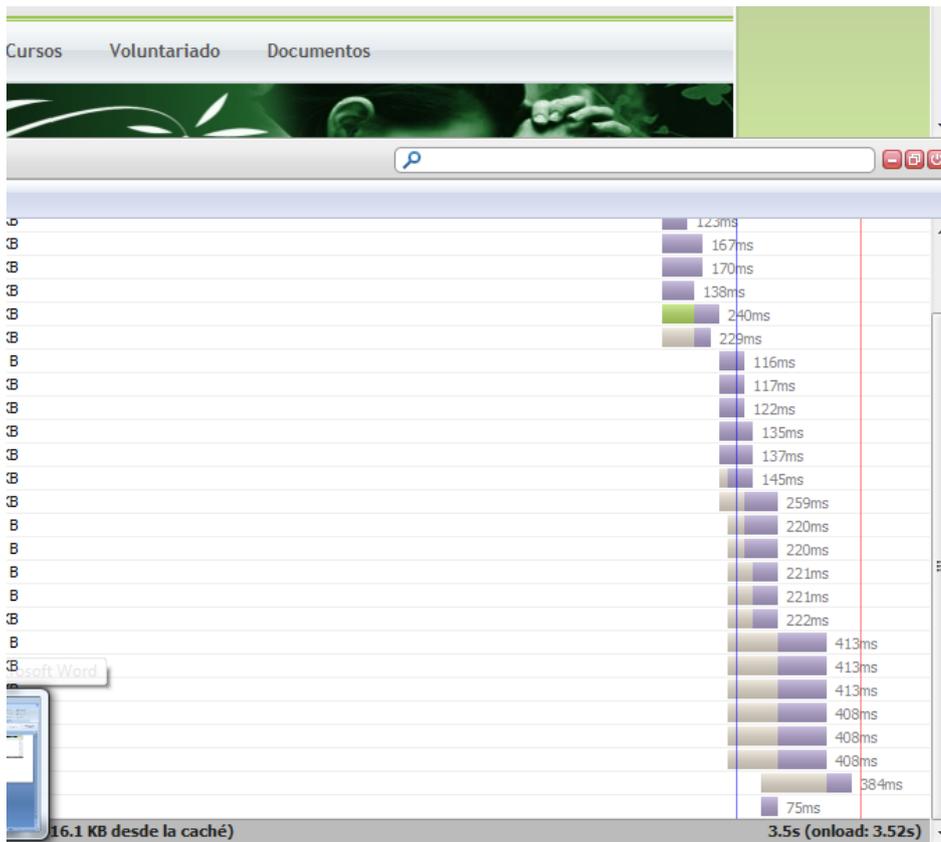


Figura ANEXO I, selec. C7: Ejemplo con el módulo y 7 filtros

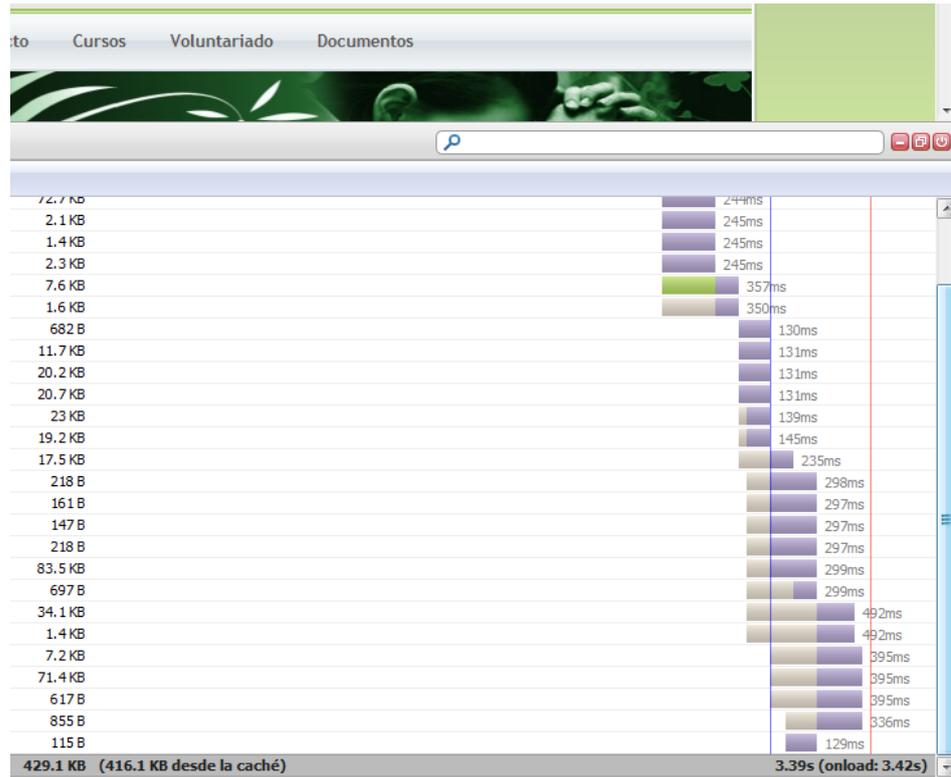


Figura ANEXO I, selec. C4: Ejemplo con el módulo y 4 filtros

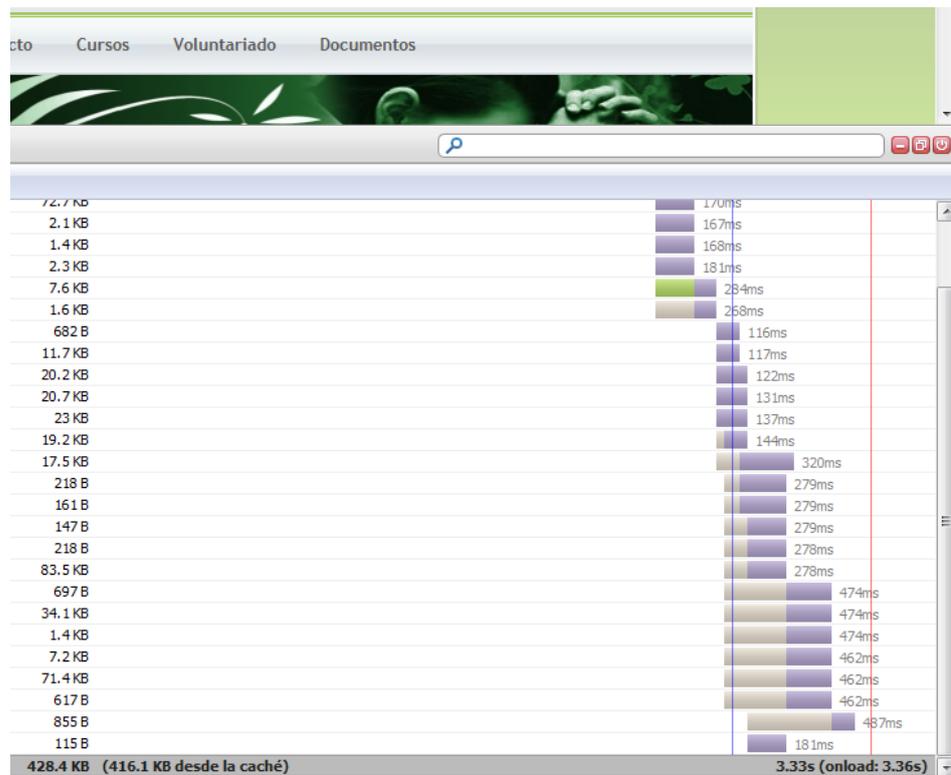


Figura ANEXO I, selec. C1: Ejemplo con el módulo y 1 filtro

D) Con 6 feed URLs configuradas:

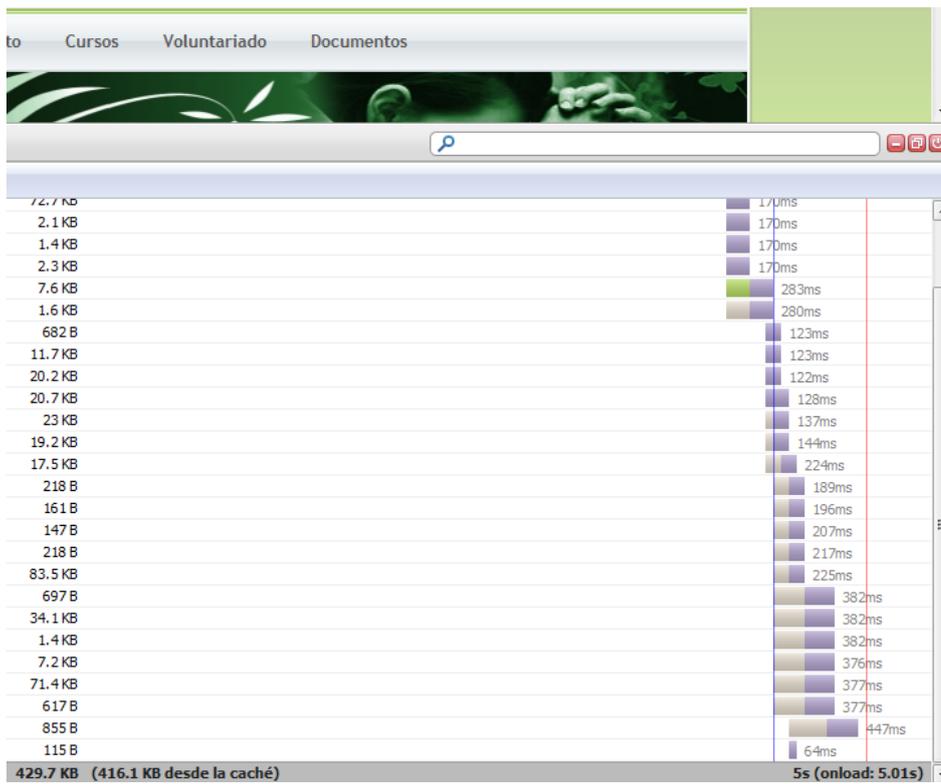


Figura ANEXO I, selec. D10: Ejemplo con el módulo y10 filtros

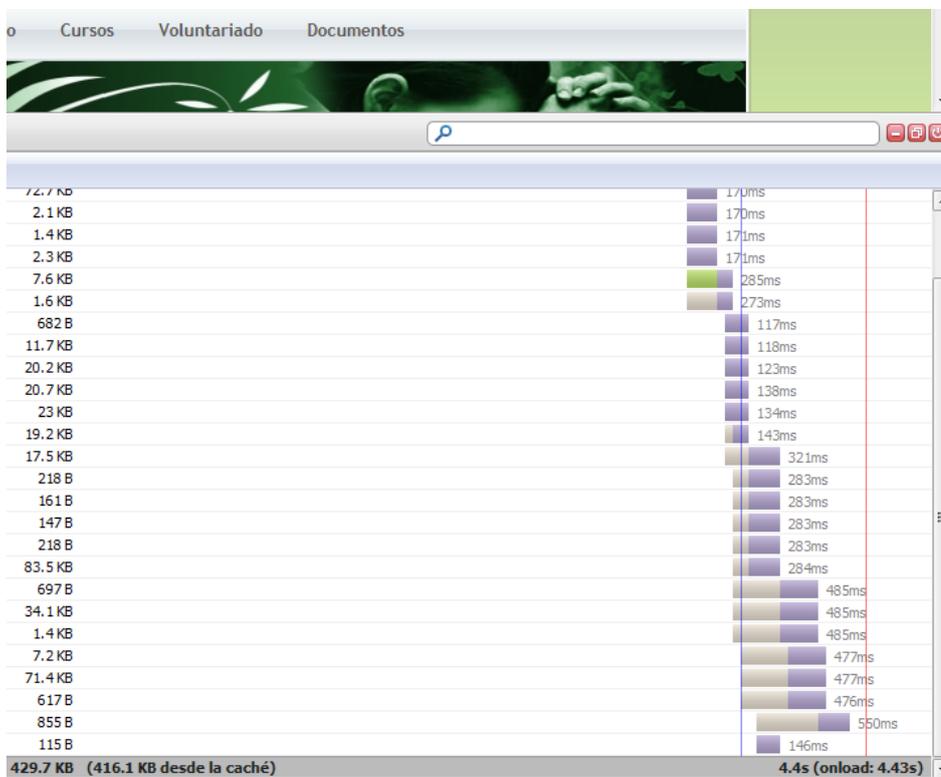


Figura ANEXO I, selec. D7: Ejemplo con el módulo y 7 filtros

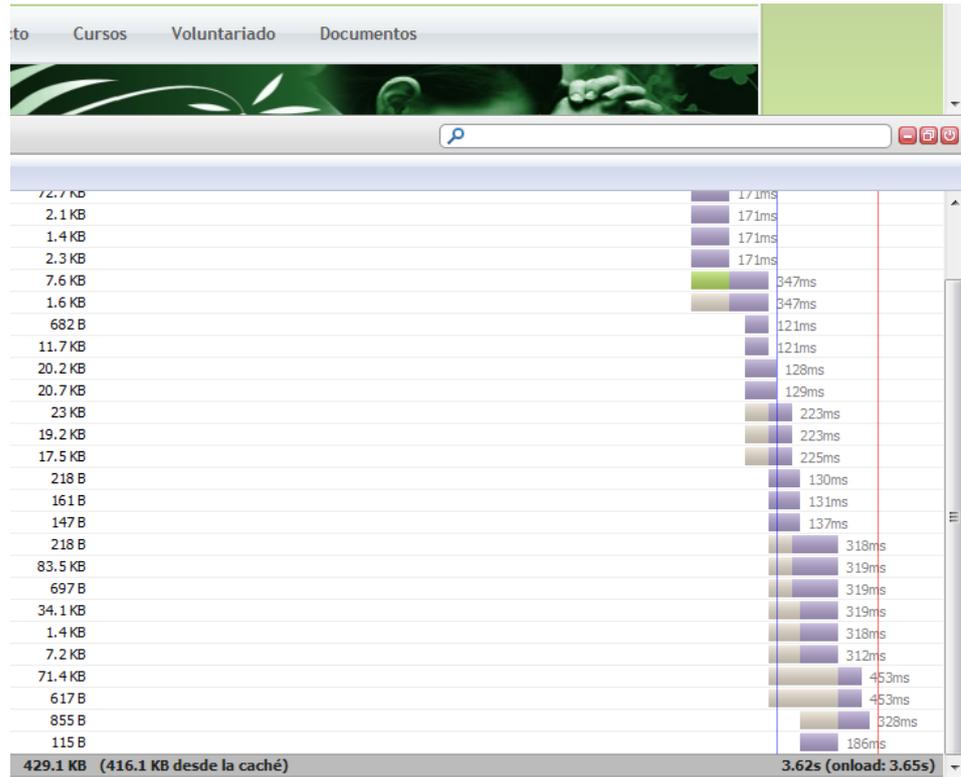


Figura ANEXO I, selec. D4: Ejemplo con el módulo y47 filtros

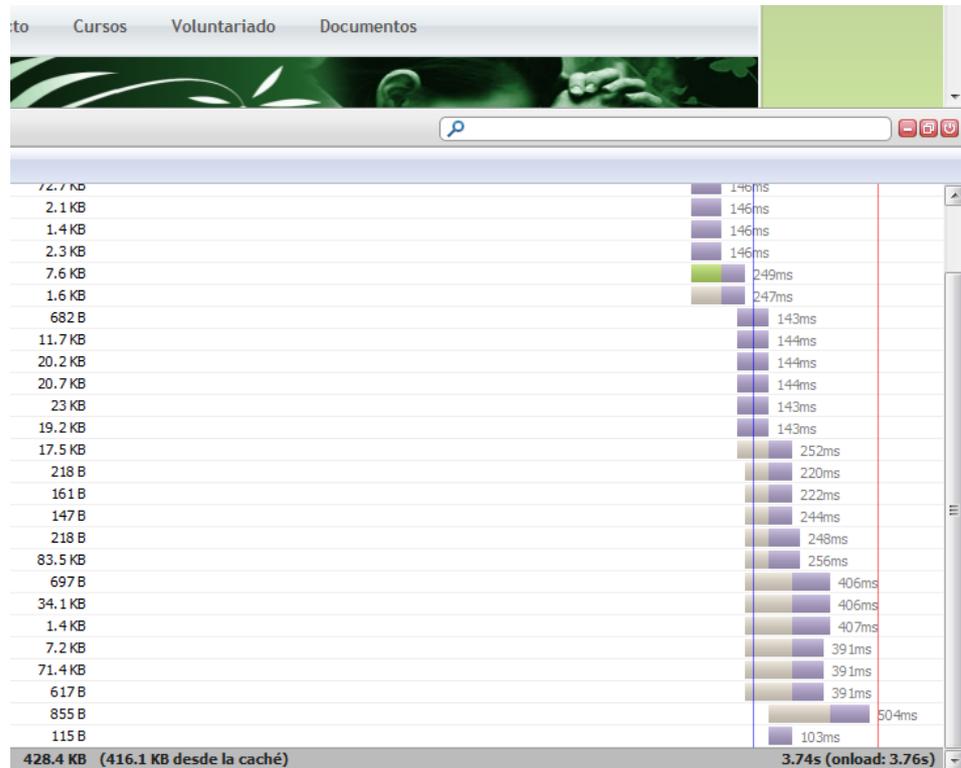


Figura ANEXO I, selec. D1: Ejemplo con el módulo y 1 filtro

ANEXO II

Testeo de los módulos implementados

FEED

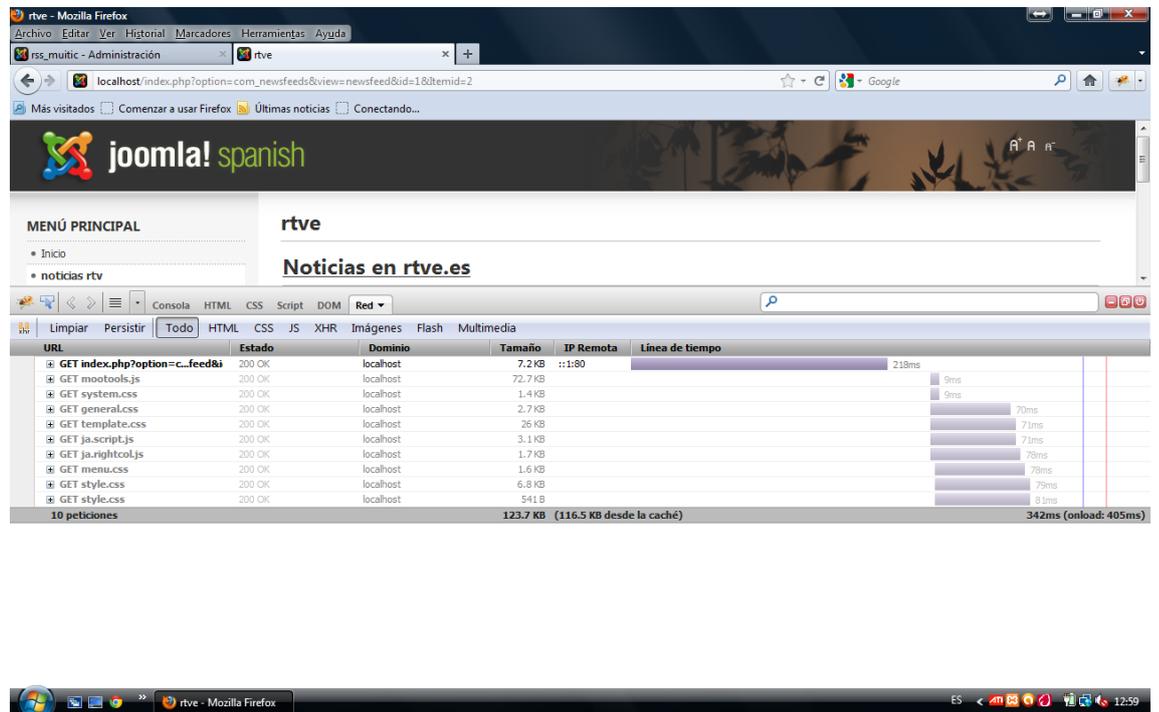


Figura ANEXO II.1: Módulo FEED, 342 ms.

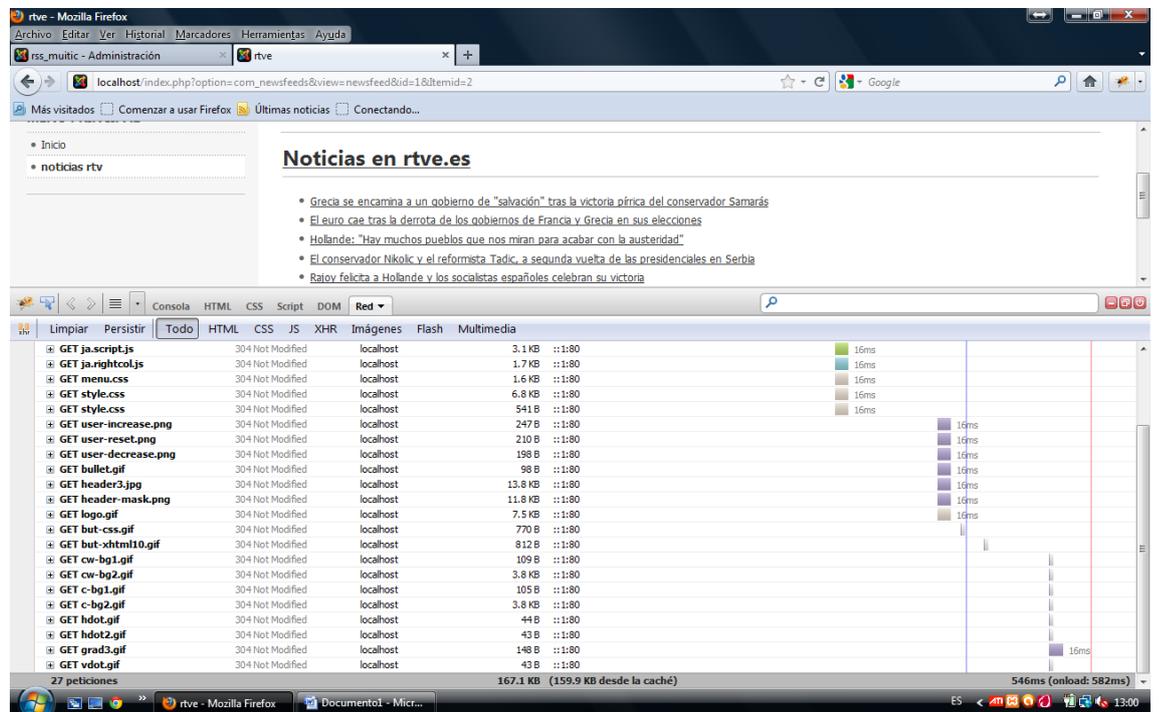


Figura ANEXO II.2: Módulo FEED, 546 ms.

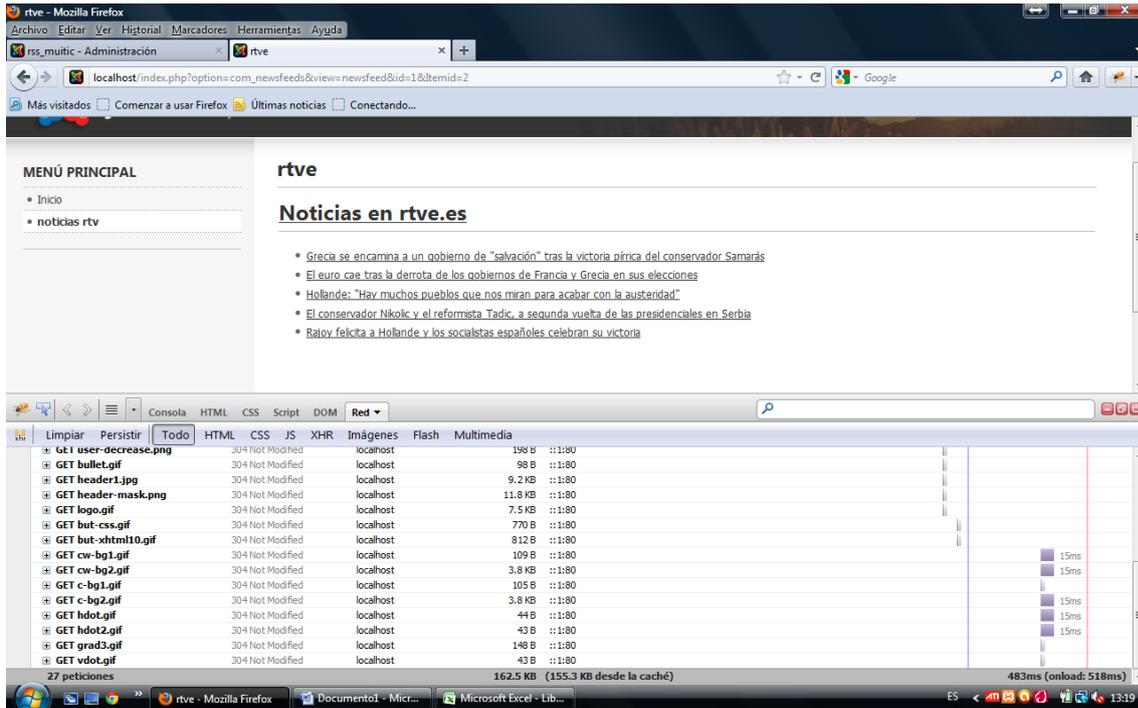


Figura ANEXO II.3: Módulo FEED, 483 ms.

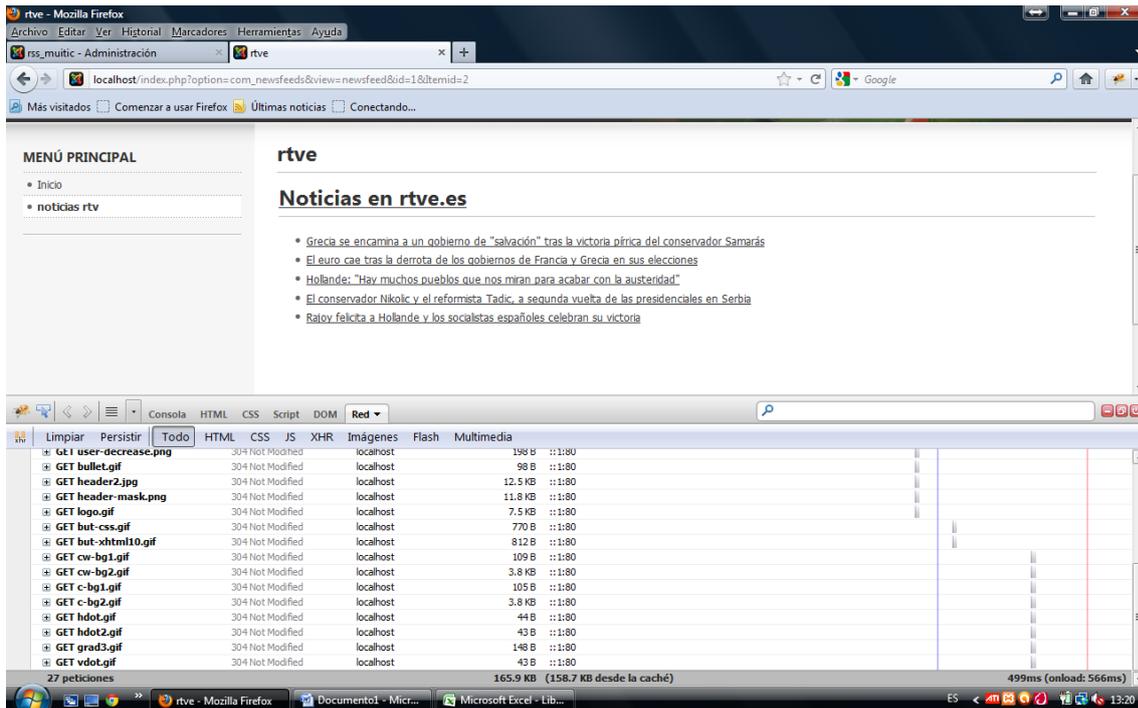


Figura ANEXO II.4: Módulo FEED, 499 ms.

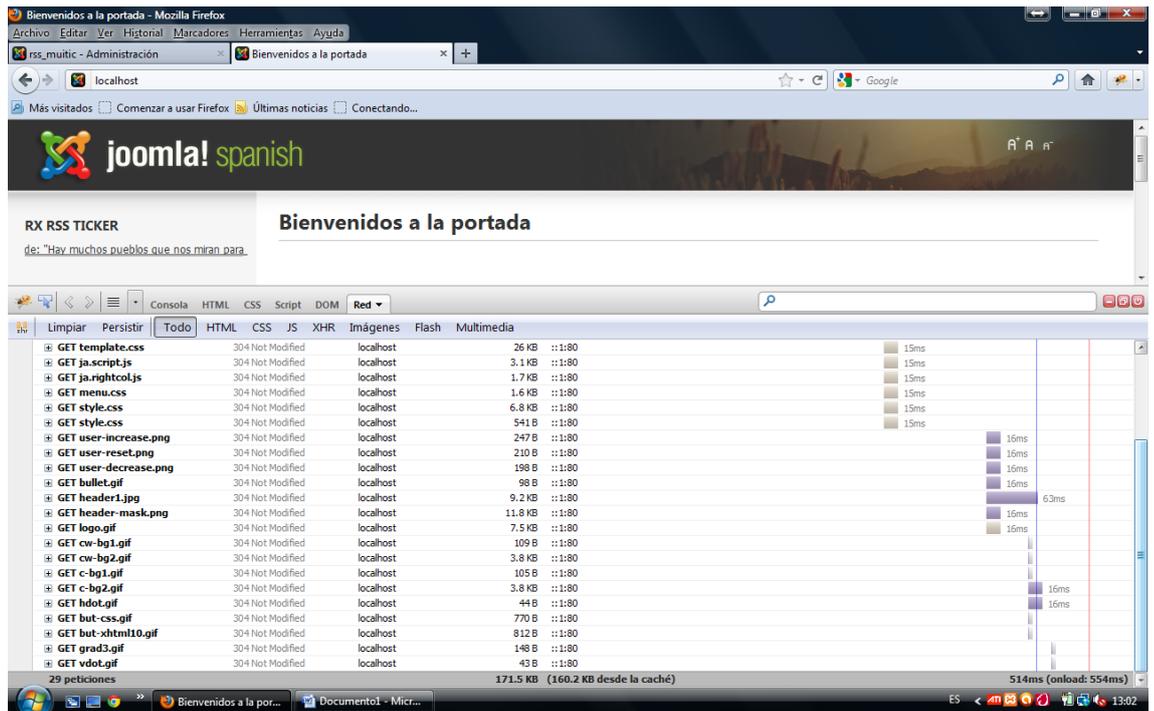


Figura ANEXO II.5: Módulo RX RSS Sticker, 514 ms.

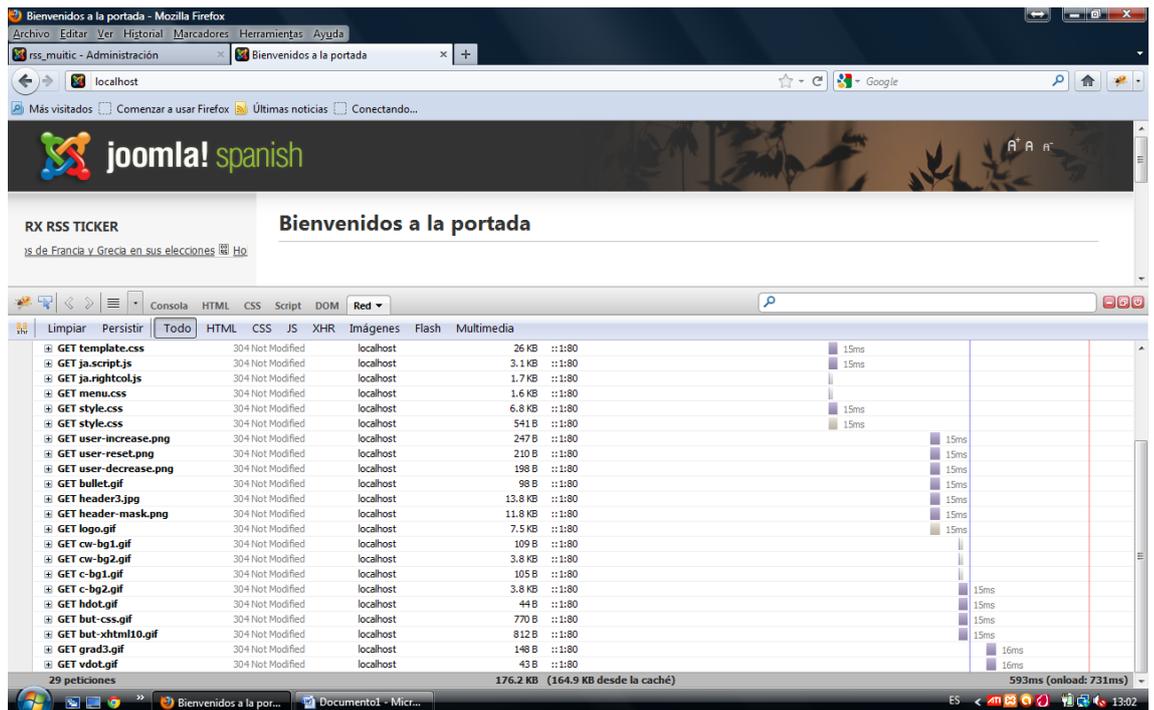


Figura ANEXO II.6: Módulo RX RSS Sticker, 593 ms.

ANEXO II: TESTEO DE LOS MÓDULOS IMPLEMENTADOS

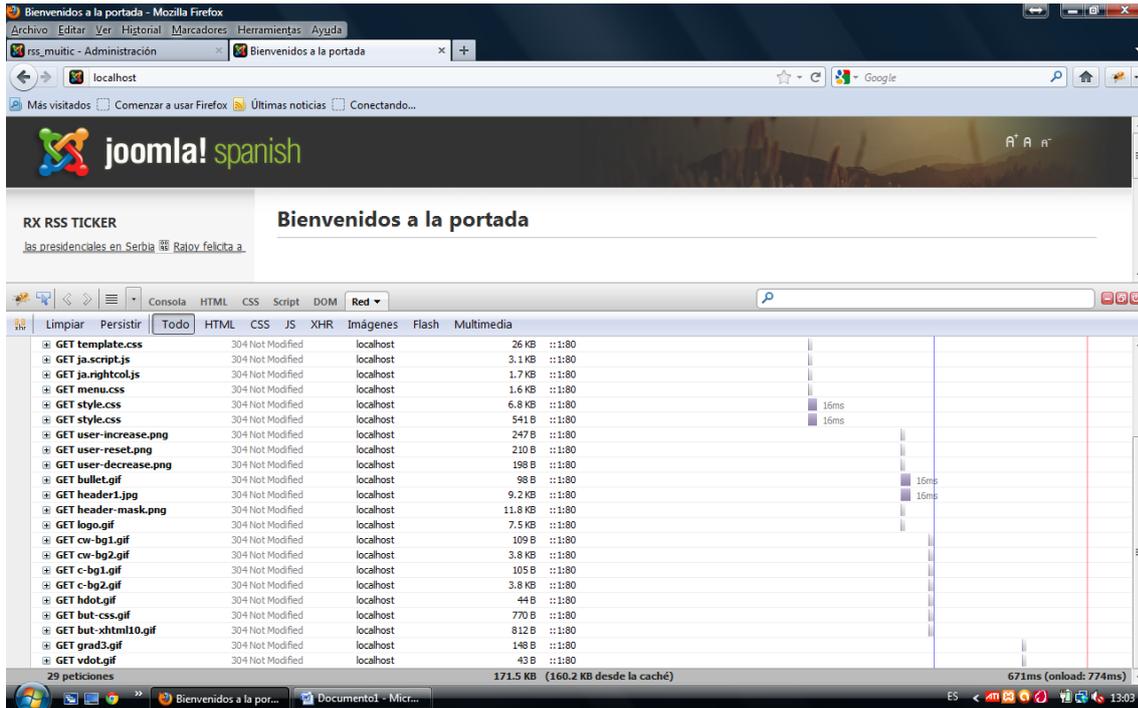


Figura ANEXO II.7: Módulo RX RSS Sticker, 671 ms.

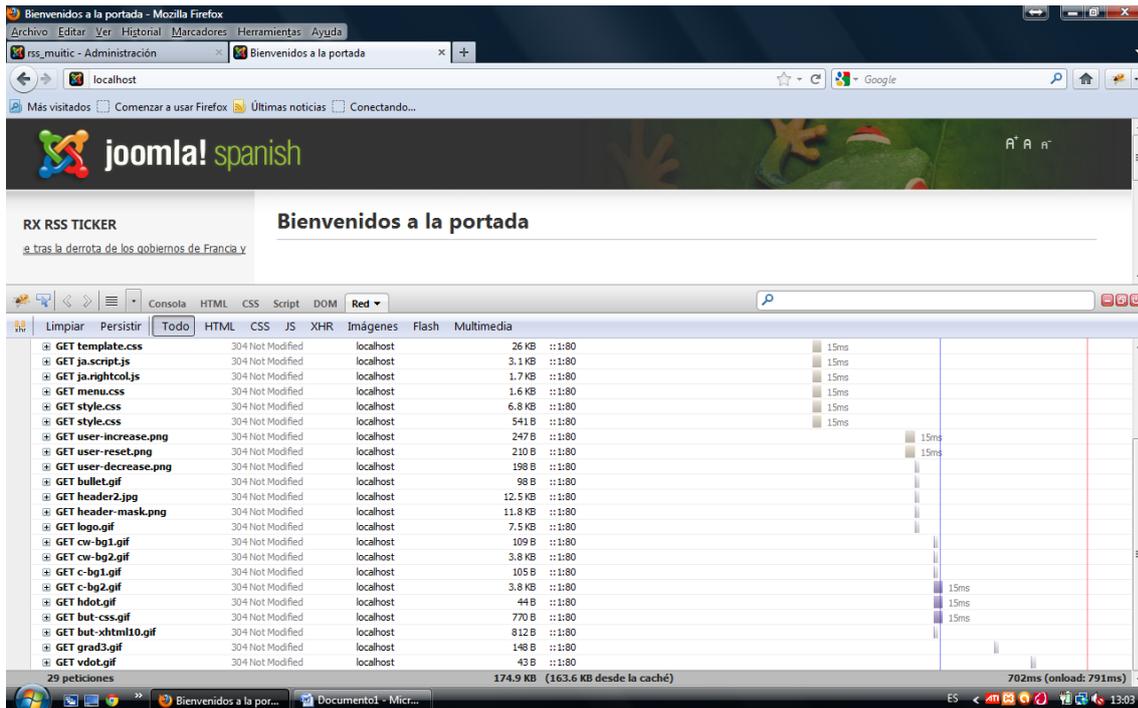


Figura ANEXO II.8: Módulo RX RSS Sticker, 702 ms.

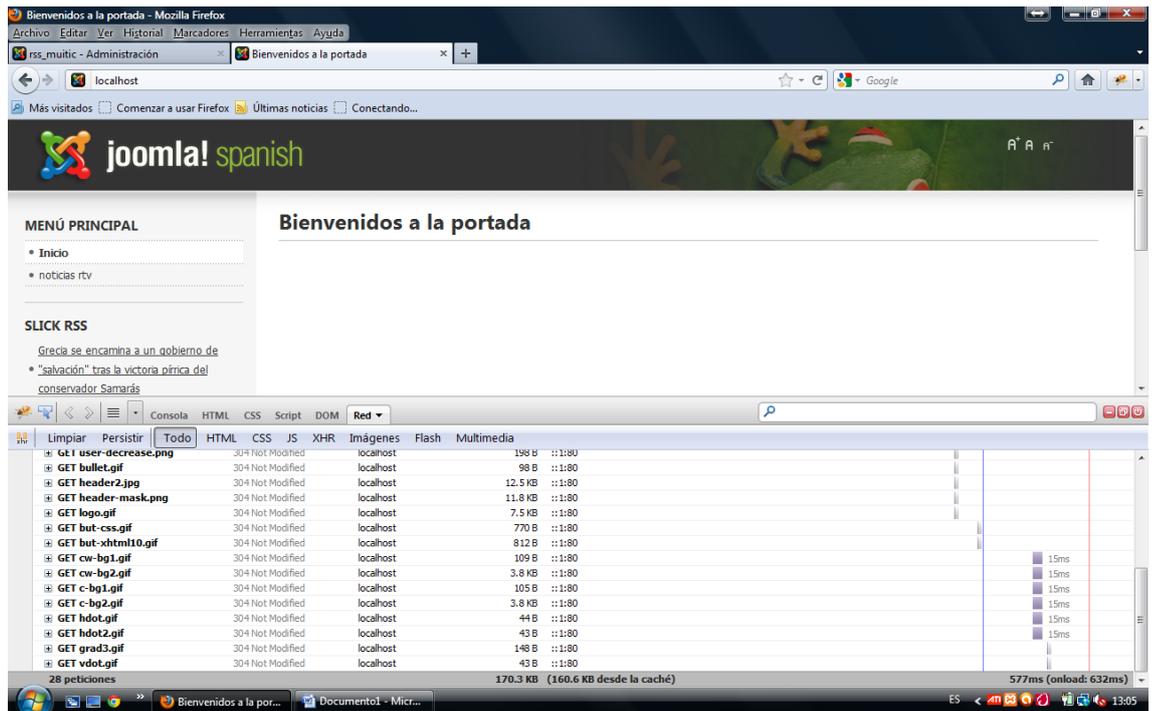


Figura ANEXO II.9: Módulo Slick RSS ,577 ms.

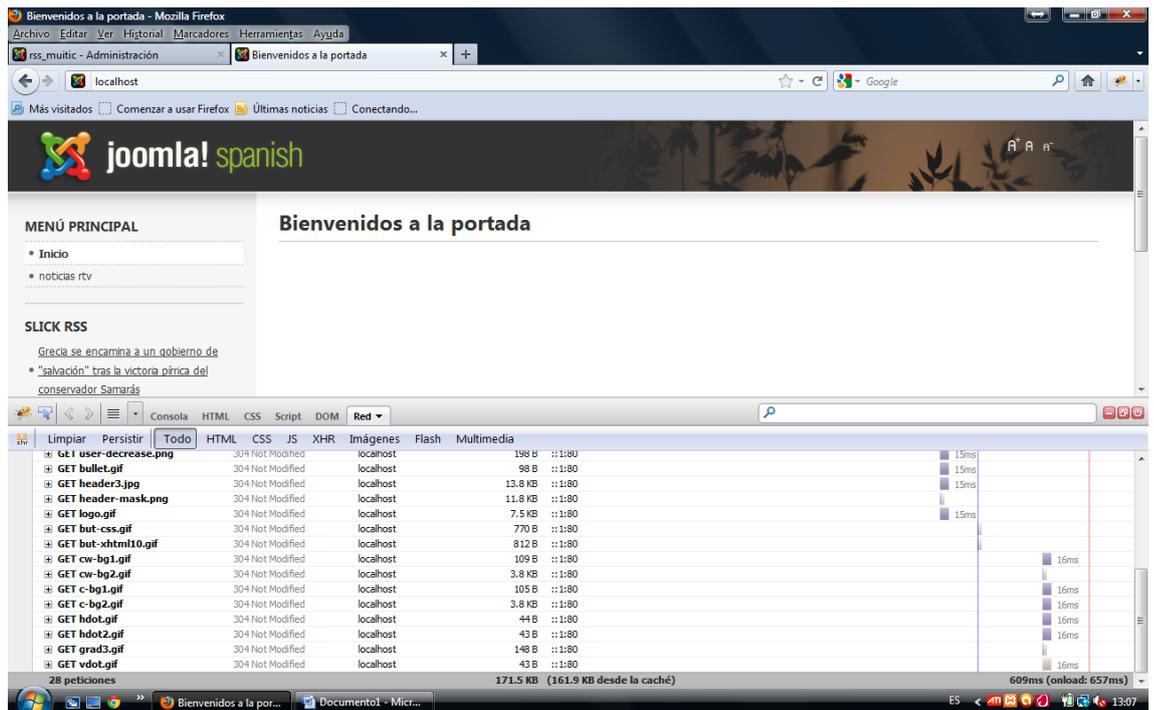


Figura ANEXO II.10: Módulo Slick RSS ,609 ms.

ANEXO II: TESTEO DE LOS MÓDULOS IMPLEMENTADOS

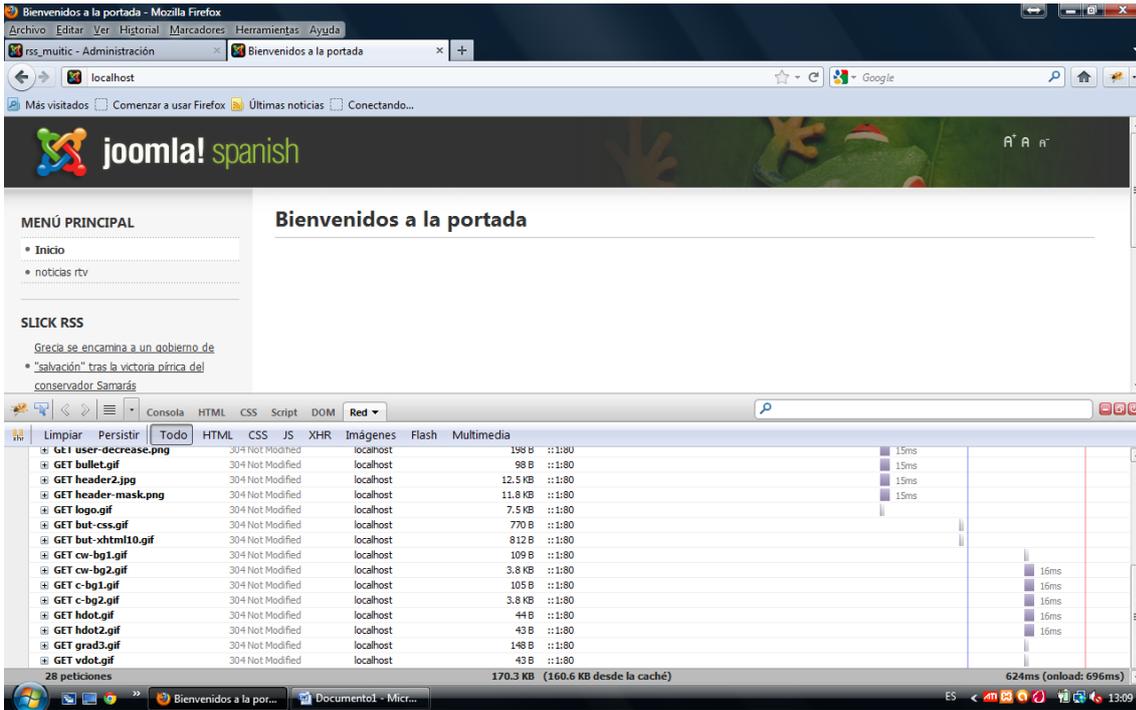


Figura ANEXO II.11: Módulo Slick RSS ,624 ms.

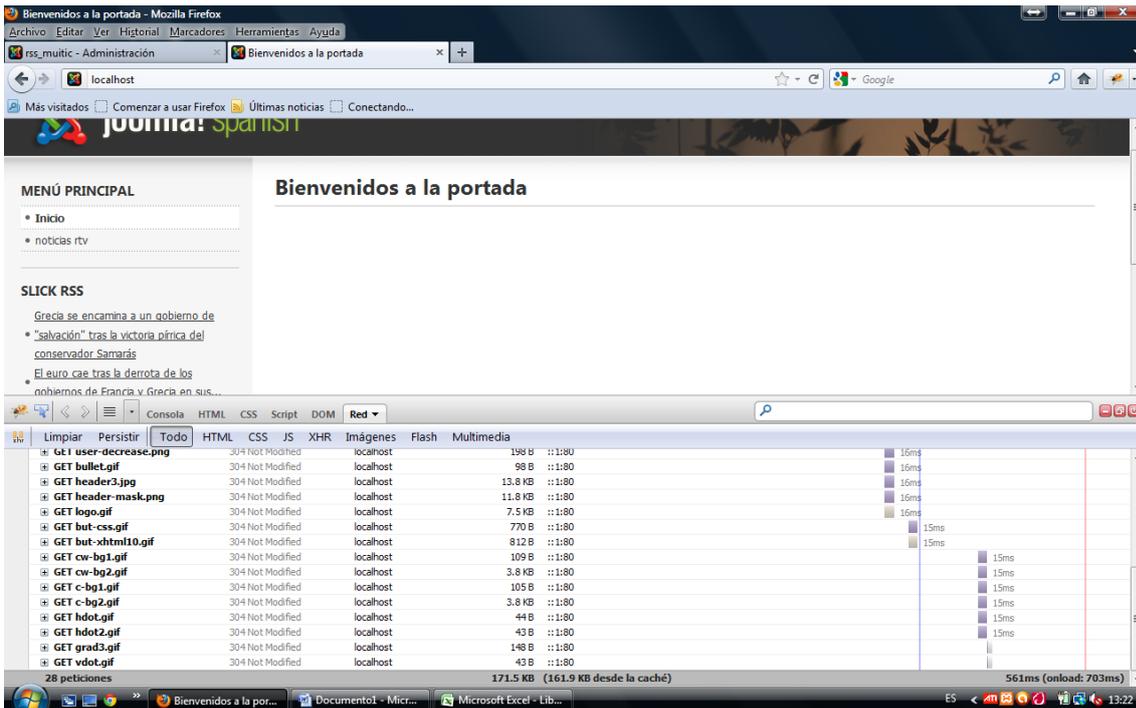


Figura ANEXO II.12: Módulo Slick RSS ,561 ms.

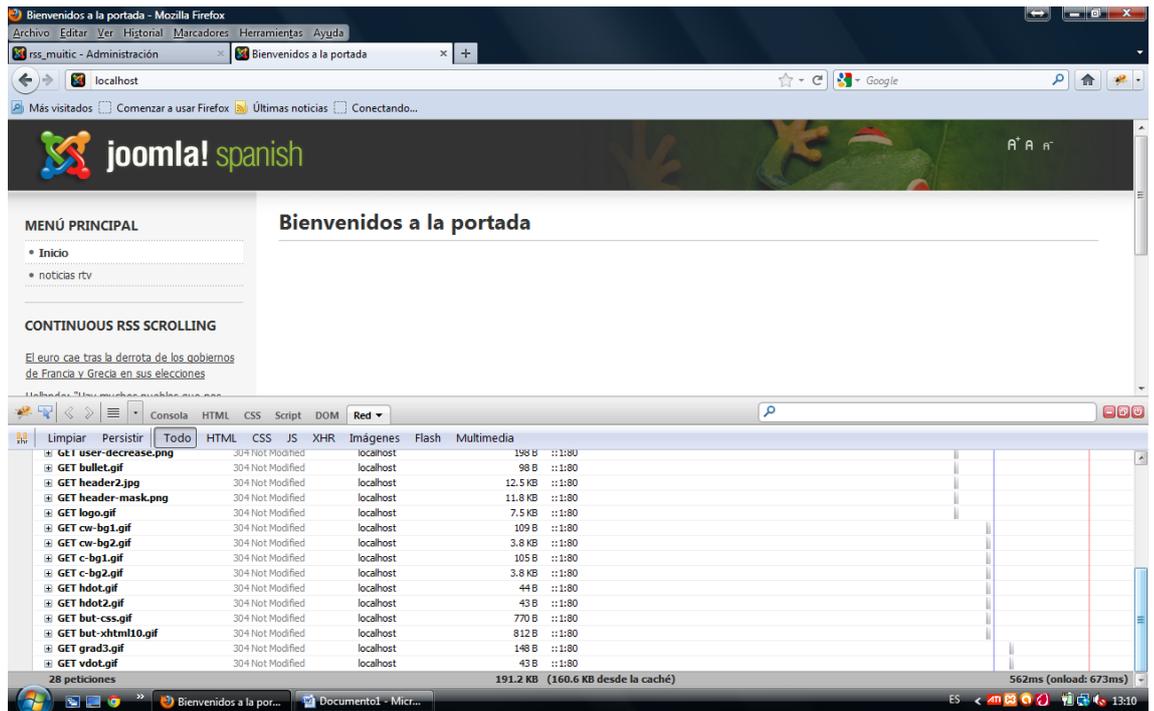


Figura ANEXO II.13: Módulo Browser2.7 562 ms.

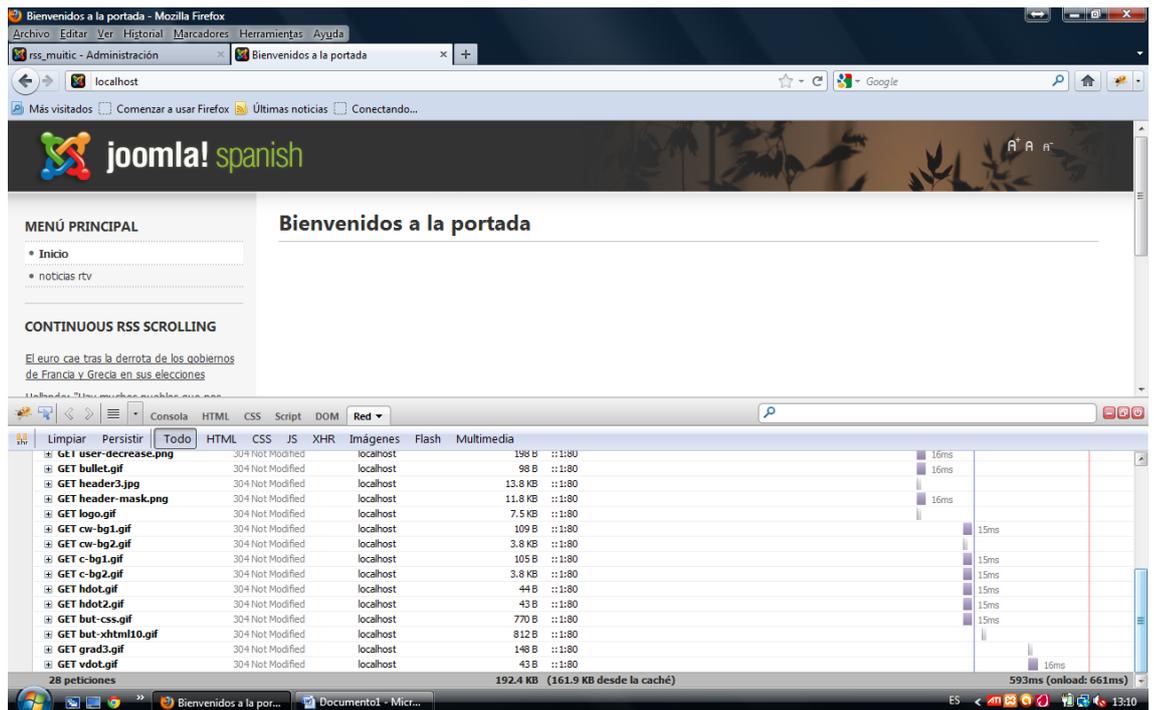


Figura ANEXO II.14: Módulo Browser2.7 593 ms.

ANEXO II: TESTEO DE LOS MÓDULOS IMPLEMENTADOS

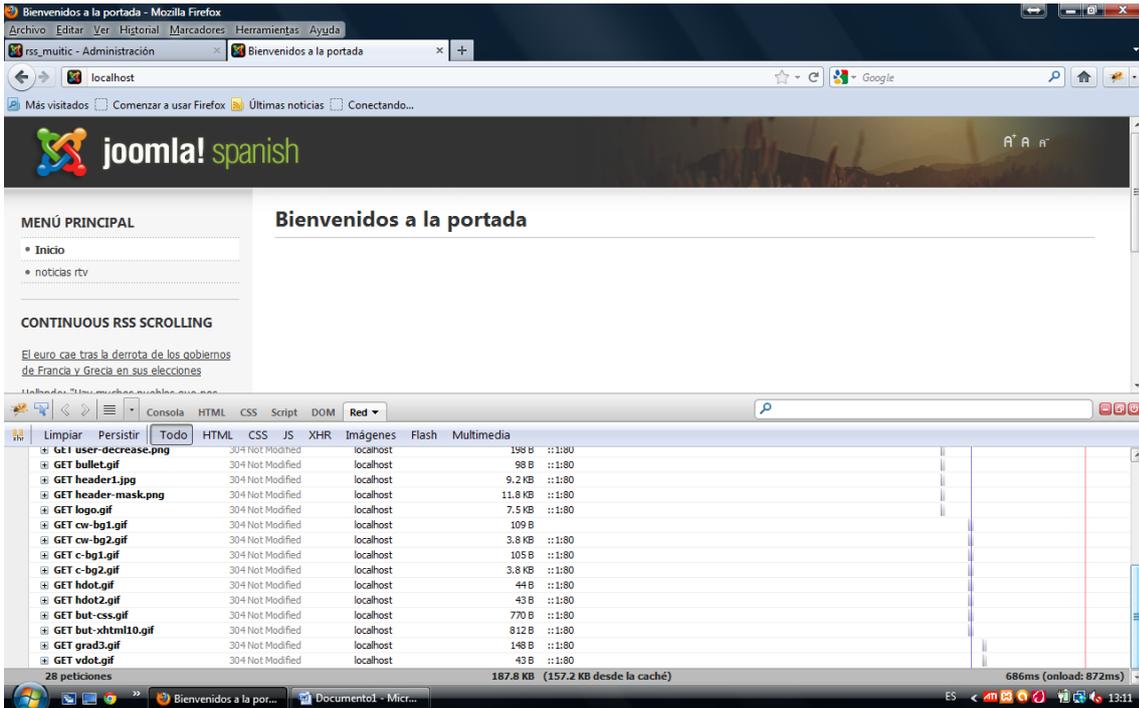


Figura ANEXO II.15: Módulo Browser2.7 686 ms.

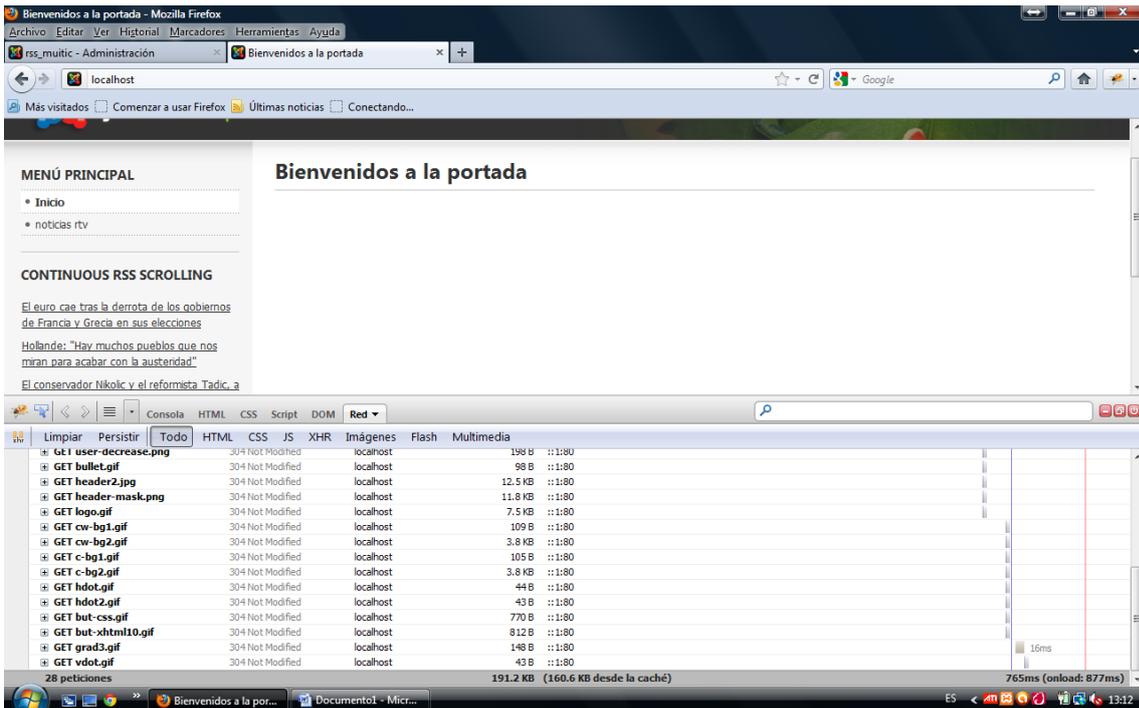


Figura ANEXO II.16: Módulo Browser2.7 765 ms.

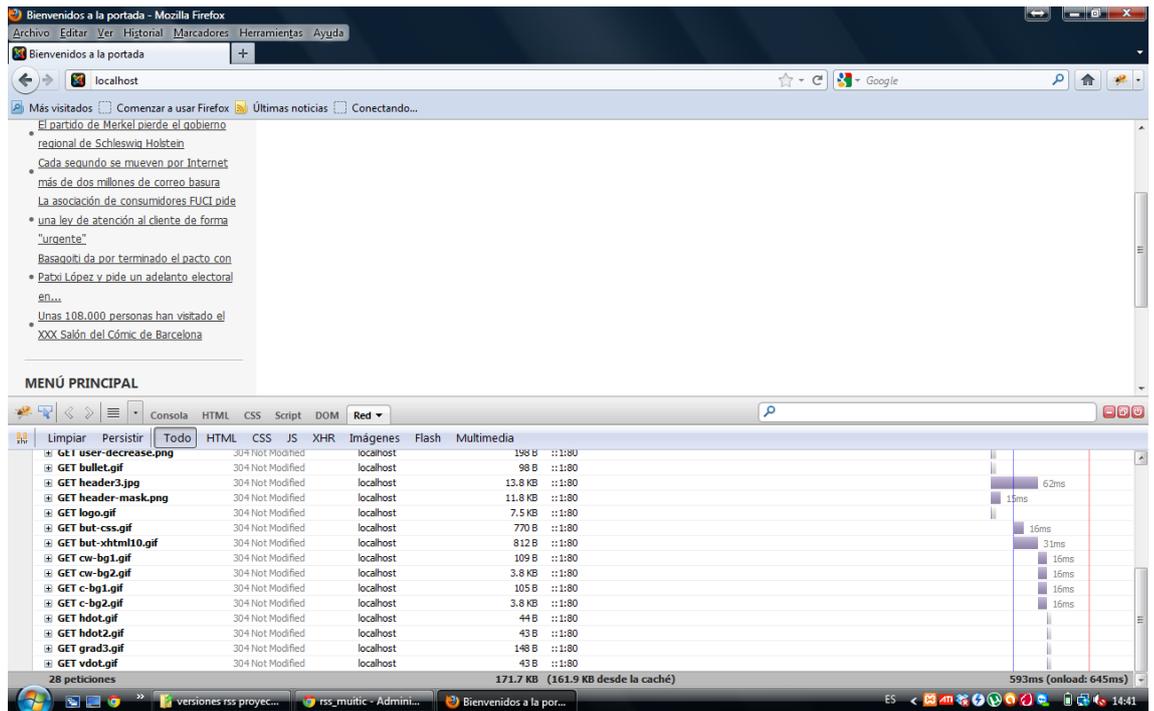


Figura ANEXO II.17: Módulo RSS_PROYECT 593 ms.

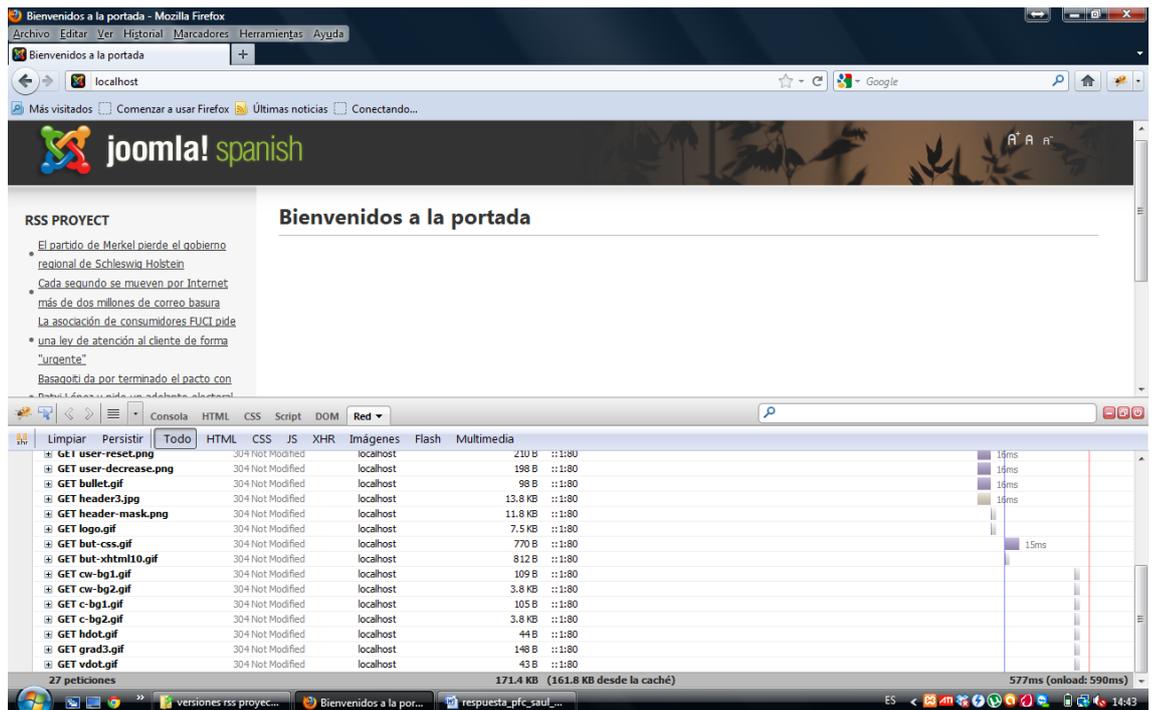


Figura ANEXO II.18: Módulo RSS_PROYECT 577 ms.

ANEXO II: TESTEO DE LOS MÓDULOS IMPLEMENTADOS

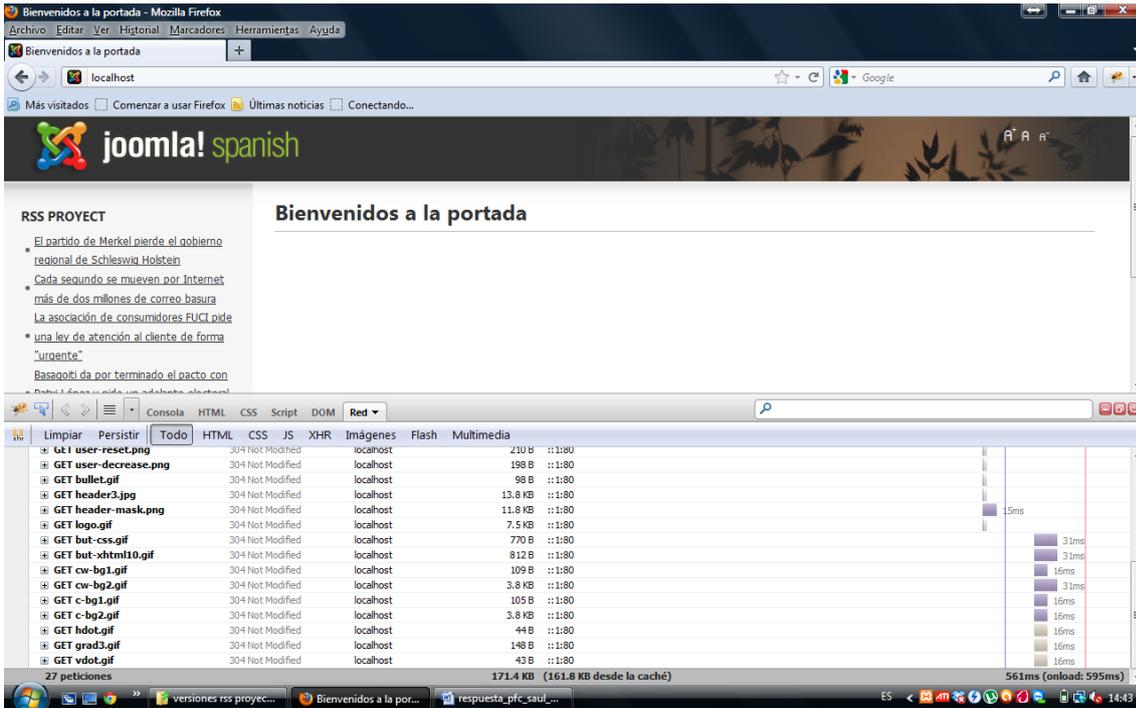


Figura ANEXO II.19: Módulo RSS_PROYECT 561 ms.

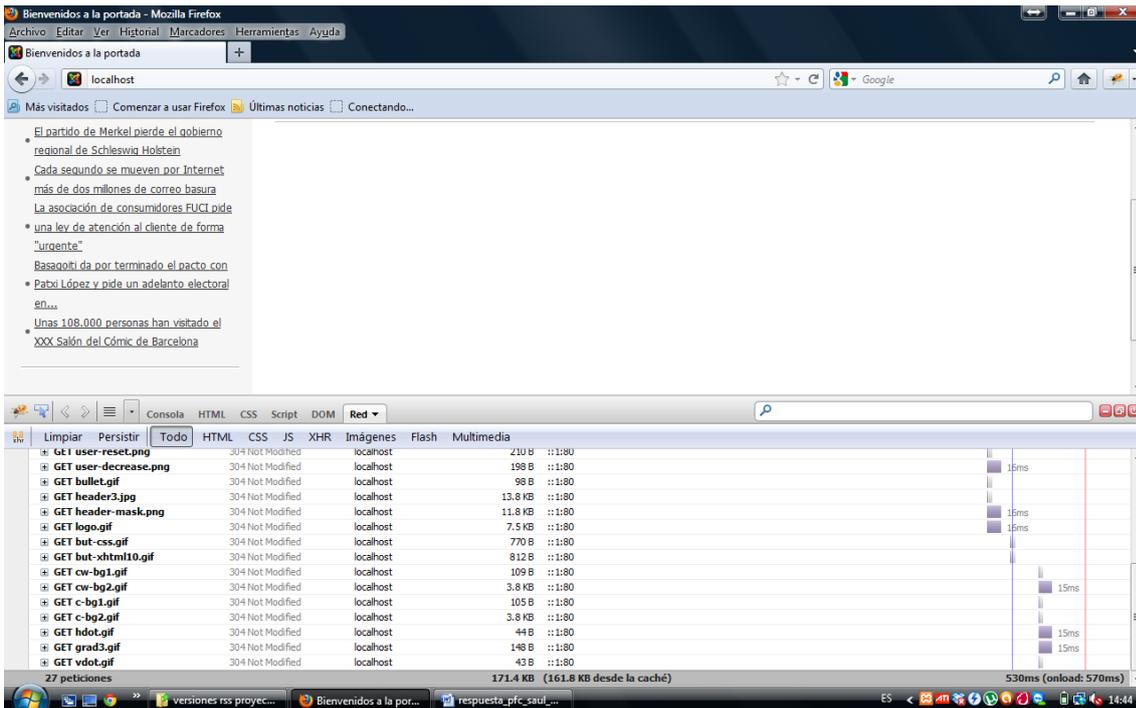


Figura ANEXO II.20: Módulo RSS_PROYECT 583 ms.

Apéndice I

Código implementado

1 Introducción

Uno de las principales dificultades a la hora de realizar este PFC, ha sido la de aprender códigos de programación, algunos muy utilizados hoy en día, de los que algunos de ellos solo conocíamos el nombre.

Cabe destacar, con diferencia, que el mayor reto a la hora de programar este módulo, ha sido la de aprender el API (Application Programme Interface) de Joomla!, ya que apenas había código, salvo el de la página oficial, que por desgracia no presentaba ningún ejemplo, o unas pautas que seguía el CMS. Simplemente exponía: métodos, objetos, clases... que los definía, pero no explicaba los protocolos que debíamos seguir en su utilización ni nada; hecho que sin duda supuso el mayor reto de este PFC, entender las rutinas de Joomla! y cómo se podía poner el código, y sobretodo que código específico de él.

2 Código PHP

Es sin duda el de mayor uso y más líneas reciben a lo largo del proyecto. Salvo un archivo, todos los demás llevan implementado en mayor o menor medida el código PHP. La versión utilizada a la hora de programar es PHP 6, a pesar de que Joomla! sólo es necesario una versión PHP 5.3, nosotros quisimos utilizar una más actual, para que así el código pudiera ser utilizado en futuras mejoras sin que se tuviese que reescribir por completo todo él.

A continuación, sin entrar en detalle, daremos unas pinceladas en las partes del código más importantes, explicándolas brevemente su funcionamiento.

La sentencia FOREACH, es sin duda una de las más utilizadas a lo largo del programa. Su funcionamiento es similar al de un bucle FOR, salvo que esta realiza una iteración de una variable y va almacenando sus resultados en el nombre de otra que le demos en forma de array. La hemos utilizado a la hora de navegar por los artículos de cada noticia. Un ejemplo del código a continuación [23].

```
// Obtención del logo feed (imagen de la fuente)
if ($feed->get_image_url() && $rssimagen)
{
    $rss_project['items'][$rssproject_item]['image_url']= $feed->get_image_url();
    $rss_project['items'][$rssproject_item]['image_title']= $feed->get_image_title();
}

/*****
* FOREACH: cojo cada feed y obtengo sus articulos uno a uno
*****/
foreach ($feed->get_items(0,$total_noticias) as $cont_art)
{
    //ESCAPE
    if ($j<$rssitem)
    {

        //Variable que me dice si la URL esta en la lista negra
        $permisos=0;
```

Apéndice I, 2A: FOREACH

Otras de las funciones más utilizadas han sido la familia de las funciones de procesado de cadenas. Estas son el núcleo empleado en los filtros buscadores, tanto el selectivo, como el genérico.

En el ejemplo anterior vemos la aplicación del al función STRISTR que lo que hace es comparar dos cadenas, independientemente sean en minúsculas o mayúsculas, y devuelve el valor TRUE en caso de ser cierto.

```
{

//////////PRIMER FILTRO SELECTIVO//////////
if ($selec1!=NULL)
{
    $acierto=0;
    $selec1_s = trim($selec1);
    $sell=explode(' ', $selec1_s);
    $con_s1=count($sell);
    for($m=0; $m<$con_s1;$m++)
    {
        if(stristr($arti_titulo, $sell[$m]) == TRUE)
        {
            $acierto++;
        }
    }
    if ($acierto==$m)
    {
        $comprobacion_filtro='OK';
    }
}
```

Apéndice I, 2B: STRISTR

El resto de funciones utilizadas: IF, FOR, ELSE... no hablaremos de ellas, puesto que son relativamente genéricas y conocidas a nivel básico de programación [24].

3 Código Bases de datos MySQL

Se trata de un código utilizado por Joomla! a la hora de realizar consultas a la base de datos. Ha sido necesaria nuestra utilización en este módulo, a la hora de añadir y buscar en la tabla de datos Lista_negra, utilizada para almacenar las URLs eliminadas por usuario administrador.

A pesar de no haber más 40 líneas de código MySQL, este se ha visto fuertemente limitado por la API de Joomla! que para poder ejecutarlo, requería una serie de métodos Joomla!, ya que, de la manera tradicional de programarlo, el CMS los detectaba como que fuesen una intrusión todas estas consultas, y su sistema de seguridad las denegaba.

A continuación veremos un ejemplo de esta aplicación:

```
// Creo y llamo al objeto JTable para poder añadir y ver campos de la BBDD
JTable::addIncludePath(JPATH_BASE.DS.'modules'.DS.'mod_rss_project');

//Instancia al archivo lista_negra.php donde se obtiene el objeto
$listanegra =& JTable::getInstance('lista_negra', 'Table');

//Obtengo la variable del boton submit pulsado y la almaceno en boton (en nuestro caso la URL c
$boton= JRequest::getVar("valores");

// Aqui creamos la tabla de la base de datos de la LISTA NEGRA, y si existe se queda igual
$db= &JFactory::getDBO();
$query1 = 'CREATE TABLE #_lista_negra (id INT NOT NULL AUTO INCREMENT, url VARCHAR(400), PRIME
$db->setQuery($query1);
$db->query();

// Aqui selecciono la tabla usuarios y luego proceso los campos nombre_usuario y tipo_usuario
$query3 = 'SELECT * FROM #_users ';
$db->setQuery($query3);
$tipos = $db->loadObjectList();
$sp=0;
```

Apéndice I, 3: Query

A través de consultas llamadas *\$query* en nuestro programa, conseguíamos utilizando métodos de Joomla! para que se pudiesen ejecutar. En este ejemplo en concreto se puede ver la creación de la base de datos en caso de que no existiese, y una consulta más abajo para obtener todos los datos de los usuarios almacenados en nuestra base de datos.

4 HTML

En este caso, la utilización de HTML ha sido prácticamente nula, salvo para las salidas del módulo 'default.php', que se encarga de la visualización de los resultados procesados.

Al programar en Joomla!, realmente la parte más débil o que menos programación necesita es HTML, porque las estructuras ya están predefinidas, y salvo algo muy concreto, es la parte que menos problemas da. Sólo debemos dar un repaso a los apuntes o tutorial sobre etiquetados, columnas, etc.

A continuación veremos en la siguiente captura de pantalla el código utilizado en nuestro módulo, junto con código PHP.

```

rs_proyect['items']}{
  h($rss_proyect['items'] as $item)

  // Título del feed
  if($params->get(rstitulo, 1) && $item['title_link'] && $item['a'])
  {
    print '<div><a href="'. $item['title_link'].'" style ="text-decoration:none" target="'. $rss_proyect['target']
  }

  // Descripción del feed
  if($params->get(rssdesc, 1) && $item['get_description'] && $item['a'])
  {
    print '<div class="slick-rss-desc">'. $item['get_description']. '</div>';
  }

  // Imagen del feed
  if($params->get(rssimagen, 0) && $item['image_url'] && $item['a'])
  {
    print '';
  }
  if($tooltips== true)
  {
    $title = $item['tooltip']['title'] . ':::'. $item['tooltip']['description'];
  }

```

Apéndice I, 4: Print

5 XML

Una de las partes básicas del módulo, es la que contiene el código XML. Si esta parte está mal, no sólo no se ejecutará, sino que no podremos instalarlo nuestro módulo, ya que la parte XML contiene principalmente el código del instalable.

Como ya hemos dicho es una parte crítica del código, ya que además de todos los fichero y raíces del programa, contiene la parte de las variables que se van a introducir desde el Back-End, por lo que una mala programación provocara la perdida de entrada de datos o configuraciones desde nuestro Back-End.

Una de las dificultades que hemos tenido a la hora de programar nuestro módulo, es que, al igual que la API de Joomla!, la información sobre este apartado es mínima. Como curiosidad, diremos que tardamos casi un día entero en conseguir que nuestro programa se instalase una versión básica. Esto se debía a que en los archivos XML no está permitidos acentos ni algún tipo de caracteres (como por ejemplo la ‘ñ’). Nuestro programa no daba errores a la hora de programarlo, pero al intentar instalarlo, Joomla! nos mostraba un mensaje de error en el archivo instalable. Conseguimos encontrar la solución a base de descargarnos una docena de módulos y estudiar sus paquetes instalables para averiguar cuál podía ser el error.

Para terminar, mostraremos parte del código XML contenido en el archivo ‘mod_RSS_Project.xml’:

Apéndice I, 5: XML

6 API de Joomla!

Podríamos afirmar con total certeza, que este ha sido el cuello de botella del proyecto. A la hora de encontrar información a nivel de programación de su API, ha sido tarea poco más que difícil. Es cierto que a través de su web encontrábamos una serie de definiciones de métodos, funciones, etc. Pero también es cierto que la información de la serie de convenios en los que se basa la versión 1.5 sobre la que se programaba eran inexistentes, y que la forma de encontrar y entender las herencias que se hacen en Joomla!, o los convenios a la hora de llamar un archivo, o enviar una variable de un fichero a otro, ha sido a base de foros de páginas en inglés, y programas que pudieran tener alguna función, o consultas, o formularios, ya que un tutorial sobre su API era inexistente.

Parte de las dificultades que nos ha dado esta situación, es que por ejemplo a la hora de eliminar una noticia mediante un botón en la interfaz, nos daba error, ya que utilizábamos código PHP directamente, sin usar las funciones específicas de Joomla!, lo que el programa lo detectaba como una intrusión, y denegaba el acceso (en parte esto es bueno, ya que quiere decir que tienen un cierto nivel de seguridad nuestra web frente a intrusiones no deseadas) lo que suponía un verdadero quebradero de cabeza, ya que a la hora de estudiar el Framework (en inglés) de la página oficial de Joomla!, no existían ejemplos de sus funciones y métodos, y a parte, muchas de ellas eran un tanto ambiguas, ya que podías utilizar dos métodos diferentes para obtener una cosa, pero luego dependiendo del que utilizase te dejaba realizar una serie de tareas en un sentido o en otro.

Un ejemplo era ver esta de forma más clara sería los métodos: JTable y JFactory. En principio son métodos diferentes, pero con los dos podemos obtener una tabla de la base de datos (JTable: GetDBO y JFactory: GetDBO), pero sin embargo si queremos introducir datos en la base de datos, el método JFactory ya no nos

sirve (a pesar de que sí que es útil para leer una tabla, a través de la herencia de GetDBO). Para introducir datos en una BB.DD. necesitamos utilizar el método JTable si o si, construyendo una función que llame al Objeto JTable, para luego, a través de métodos, poder realizar las consultas, inserciones o modificaciones.

A continuación veremos los dos ejemplos utilizados en el módulo.

```
// Aquí creamos la tabla de la base de datos de la LISTA NEGRA, y si existe se q
$db= JFactory::getDBO();
$query1 = 'CREATE TABLE #__lista_negra (id INT NOT NULL AUTO_INCREMENT, url VARCHAR(255));
$db->setQuery($query1);
$db->query();
```

Apéndice I, 6.A: JFactory

```
// Creo y llamo al objeto JTable para poder añadir y ver campos de la BBD
JTable::addIncludePath(JPATH_BASE.DS.'modules'.DS.'mod_rss_project');

//Instancia al archivo lista_negra.php donde se obtiene el objeto
$listanegra =& JTable::getInstance('lista_negra', 'Table');

//Obtengo la variable submit pulsado y la almaceno en boton (en
$boton= JRequest::getVar("valores");
```

Apéndice I, 6.B: JTable

Estos son solo unos de los ejemplos utilizados para poder tener acceso a través de Joomla!

Esto es una de las características de Joomla!, que para bien o para mal, está diseñado así. Las ventajas son evidentes: alejas a todos los posibles intrusos levantando un nivel de seguridad en el que todo debe pasar a través de Joomla! Lo malo es que necesitamos conocer las rutinas de Joomla! lo que es un problema porque no son intuitivas y muchas veces las aplicaciones encontradas en su WEB oficial resultan ambiguas.

Por último, haremos mención a una serie de comando imprescindible para la ejecución de Joomla! que veremos en la figura inferior.

```
<?php

/* Punto de entrada Joomla!, al ejecutar el módulo, Joomla! buscará este
 * archivo como punto de entrada, desde aquí ejecutaremos los metodos de la
 * clase helper (donde se encuentra el grueso del código de programación PHP)
 */

// Código de seguridad Joomla!, actua como proteccion ante ataques externos
defined('_JEXEC') or die('Restricted access');

// Incluye el archivo (sino lo ha sido ya) 'helper.php', dentro de la ruta del directorio padre (root)
require_once (dirname(__FILE__).DS.'helper.php');

// Obtiene los datos del helper y los almacena la variable para poder utilizarlos con posterioridad
$rs = modRssProjectHelper::getRss($params);

/* Ejecuta el metodo de la clase JModuleHelper que se encarga del Front-End de nuestro template almacena
 * en la carpeta tmp con el nombre por defecto 'default.php'. (Todo esto es código específico de la AP.
 * necesario para la correcta visualización y configuración posteriormente.)
 */
require(JModuleHelper::getLayoutPath('mod_rss_project'));
```

Apéndice I, 6C: Require_once

La función `require_once` nos indica el archivo que lleva la lógica del programa, es donde se va a empezar a ejecutar todo. Es necesario declararlo, aunque por defecto siempre va a buscar los archivos `'helper.php'`, pero se le podría dar otro nombre, aunque no es recomendable. La siguiente fila indica la variable que toma, heredada desde la función `$params` y que utilizaremos en la ejecución del programa, pasándosela al archivo `default` que finalmente las publicará en pantalla.

