

EVALUACIÓN HEURÍSTICA DE SISTEMAS ADMINISTRADORES DE CONTENIDOS LIBRES¹

Sandra Lorena Anaya Díaz y Ulises Hernandez Pino²
sanaya@unicauca.edu.co

Vultur: Grupo de Aplicación de Software Libre
www.vultur.unicauca.edu.co
Universidad del Cauca
Popayán - Colombia

Con esta ponencia se busca presentar los avances realizados en el Proyecto eCMS «Evaluación Heurística de Sistemas Administradores de Contenidos Libres» resaltando fundamentalmente la metodología empleada para efectuar la exploración, evaluación y elección de un CMS (Content Management System) a partir de unos criterios iniciales y unos requerimientos específicos. La idea de realizar una evaluación heurística surgió con la intención de sistematizar y armonizar procesos informales e inconscientes de búsqueda en la web –muy usados en la implementación de soluciones con software libre–, con metodologías formales de investigación. El principal valor de esta trabajo radica en partir de un conjunto de criterios que se van re-evaluando en la medida que avanza el trabajo, y que por tanto, puede llegar a constituirse en una forma de realizar evaluaciones de soluciones de software libre disponibles en cualquier otro ámbito, enriqueciéndose en la medida en que se utilice, con nuevas formas de exploración y evaluación. Se espera que este trabajo contribuya en la elección de aplicaciones libres pertinentes y de calidad de acuerdo con los requerimientos particulares de un problema, y de esta forma aprovechar una de las características fundamentales que orienta el desarrollo del software libre: hacer programas que hagan una sola cosa, pero que la hagan bien.

1. Antecedentes

En todo ámbito académico, y particularmente en aquellas áreas denominadas ciencias aplicadas, las distintas disciplinas procuran mantener su nivel en la frontera de conocimiento, por lo que es frecuente que desde la Electrónica, Telecomunicaciones, Informática o afines, se planteen soluciones con la última tecnología cuando otras disciplinas se acercan para explorar en conjunto situaciones problemáticas. Pero lo que en muchas ocasiones no se observa, es que por lo general los problemas por fuera de estas áreas no requieren la última tecnología, ni un desarrollo desde cero, y algunas veces ni siquiera requieren una solución técnica, porque la problemática que se intenta abordar está en otro nivel.

1 Ponencia a ser presentada en la **IV Semana Linux Universidad Distrital**, Bogotá (Colombia), Octubre de 2005. Los autores entregan este documento con la Licencia Creative Commons Reconocimiento – Compartir Igual, para mayor información consultar: <http://creativecommons.org/licenses/by-sa/2.5/>

2 Ingenieros en Electrónica y Telecomunicaciones de la Universidad del Cauca, miembros del Grupo de Investigación en Educación y Comunicación – **GEC**, del Nodo Dinamizador de la Red de Investigación Educativa – **ieRed** y del Grupo GNU/Linux de la Universidad del Cauca - **GLUC**.

Particularmente en lo que tiene que ver con el software, muchas veces se plantea el dilema de si se requiere realizar un desarrollo propio a la medida, o si mejor se compra un producto. Situación que en cualquiera de los dos casos tiene sus pro y sus contra, pero evidencia muy bien el desconocimiento de una tercera posibilidad, y es la de considerar el uso de Software Libre.

Sin entrar, en principio, en todo el entramado político y social en el de que se desarrolla el Software Libre, vale la pena enfatizar las posibilidades que tiene cuando se trata de ofrecer soluciones software:

1. Existe una enorme cantidad de programas listos para usar, estables y para muchos tipos de situaciones que permiten hacer implementaciones rápidas y confiables. Esta característica supera con creces la posibilidad de realizar desarrollos propios o de mandarlos a hacer (a menos que no se encuentre un programa libre para lo que se necesita).
2. Son programas que se pueden adaptar, si se requiere, de acuerdo con necesidades específicas y en el momento en que se necesitan, lo que resulta muy conveniente cuando se parte de unos requerimientos iniciales, y solo en la medida en que se usa el software se van detectando nuevas necesidades (lo que generalmente sucede).
3. Son programas que por lo general no tienen costo, lo que favorece significativamente iniciativas que tienen presupuestos pequeños (lo que suele ser muy común en los proyectos académicos de nuestro país).

En cuanto a las desventajas, que son relativas dependiendo de la perspectiva desde la cual se aborden, es importante considerar:

1. Por lo general no se cuenta con una empresa que ofrezca soporte, y si la hay, sucede lo mismo que con los programas no libres: hay que pagar por el soporte. Generalmente las implementaciones que se realizan con Software Libre debe prever la conformación de un equipo de personas dispuestas aprender de forma autodidacta, que generalmente encuentran apoyo (pero que es diferente a soporte) por comunidades de usuarios y desarrolladores que giran en torno a estos programas.
2. En el mundo del Software Libre, generalmente no se encuentra un programa dominante por área, sino muchos programas, cada uno especializado en una tarea en particular, por lo que escoger el programa adecuado -cuando no se tiene este conocimiento previamente-, puede requerir mucho tiempo de búsqueda y pruebas de acuerdo a lo que se necesite.

Pero también es necesario considerar que la utilización de Software Libre no sólo tiene unas implicaciones técnicas y económicas, sino que también tiene unas implicaciones sociales y políticas, ya que el modelo de desarrollo del Software Libre te involucra con una forma de ser y de actuar hacia la colaboración y la solidaridad. Con el Software Libre, un programa no es sólo un producto para comercializar, sino que es un instrumento que puede ayudar a las personas a compartir, y ese debería ser el fundamento de una mejor sociedad.

En este contexto, vale la pena mencionar que el Software Libre puede realizar grandes contribuciones en el ámbito académico, de un lado porque permite apoyar y dar solución a problemas concretos en los que se requiera Software, de otro lado porque permite dar soluciones rápidas y estables, y finalmente, porque posibilita promover el valor de compartir, tan necesario de rescatar en el mundo académico.

Estas consideraciones y algunas coyunturas que se mencionan a continuación, posibilitaron plantear el proyecto eCMS «Evaluación de Sistemas Administradores de Contenidos Libres»:

- En primer lugar el Grupo de Investigación en Educación y Comunicación de la Universidad del Cauca – GEC, grupo que lidera la Red de Investigación Educativa – ieRed, había implementado los servicios básicos de Correo Electrónico, Disco Web y Chat Público, pero la dinámica de la red, empezó a requerir de un sistema que permitiera publicar cierto tipo de información.
- En segundo lugar la conformación del Grupo de Estudio Vultur en el marco del anterior Grupo de Investigación y del Grupo GNU/Linux de la Universidad del Cauca, con el propósito abordar el conocimiento de tecnologías relacionadas con el Software Libre, pero a partir de su utilización en contextos específicos.
- En tercer lugar la pertenencia de algunos miembros del Grupo Vultur, a otros Grupos, particularmente en el campo de la Educación, ha permitido ir introduciendo paulatinamente el discurso del Software Libre en otros espacios diferentes al meramente técnico, con lo cual, el siguiente paso era consolidar procesos que permitieran demostrar con acciones, las ventajas y posibilidades que permiten este tipo de aplicaciones.

2. Planteamiento del Proyecto

Con base en las características y necesidades que se observaron al interior de la Red de Investigación Educativa - ieRed, de sus particularidades propias, tanto en la infraestructura telemática como en el tipo de usuarios y en el propósito que persigue esta Red, se tomaron como punto de partida algunos requerimientos que debía satisfacer el Sistema Administrador de Contenidos a elegir:

- **Soporte de español o facilidad para españolizar:** debido a que los usuarios que potencialmente utilicen el CMS tendrán poco o nulo dominio en otros idiomas diferentes al español y además escaso manejo del computador.
- **Posibilidad de construir comunidades virtuales:** ya que la principal estrategia de consolidación de la Red de Investigación Educativa - ieRed esta centrada en el apoyo a diversos grupos, promoviendo inicialmente el intercambio de información y experiencias a nivel de los grupos como tales, y luego la realización de estos intercambios en ámbitos más abiertos.
- **Licencia GPL o compatible:** la preferencia por este tipo de licencia se dio principalmente para tener confianza en que el producto se siga manteniendo libre en el mediano plazo.

- **Niveles de acceso y usuarios:** Dentro del sistema se desea tener diferentes tipos de usuarios tanto para los permisos de realizar publicaciones, como para el acceso a los archivos de los diferentes grupos que se soportarán.
- **Buen manejo de archivos y directorios:** el requerimiento explícito de este servicio se debe a que es uno de los principales servicios que utilizan los usuarios actuales de la plataforma telemática de ieRed, y servirá como estrategia de migración.
- **Usabilidad, transparencia para el usuario final:** debido al poco nivel de conocimiento y de habilidades que tienen los usuarios potenciales y que hacen parte de la Red de Investigación Educativa.
- **Proyecto Maduro o con una versión estable reciente:** para asegurar que existe una comunidad interesada en mantener el producto, que se preocupa por lanzar versiones con cierta frecuencia y poder prestar un servicio desde ieRed con una alta estabilidad y confiabilidad.
- **Buena documentación:** debido a que tanto la instalación, configuración y operación se debe hacer por autoaprendizaje.

Para iniciar la labor, además de los requerimientos que se acaban de mencionar, trazamos unos objetivos y un plan de trabajo³:

Objetivos

- Definir un conjunto de criterios para evaluar Sistemas Administradores de Contenidos (CMS - Content Management System) de acuerdo con los requerimientos de la Red de Investigación Educativa - ieRed.
- Identificar los Sistemas Administradores de Contenidos Libres que cumplan con la mayor parte de los requerimientos, y seleccionar algunos para evaluarlos.
- Seleccionar un Sistema Administrador de Contenidos e implementarlo para la Red de Investigación Educativa - ieRed.
- Migrar la información que se viene manejando en los proyectos asociados a ieRed en al Sistema Administrador de Contenidos implementado.

Plan de Trabajo

Metodología: Exploración Tecnológica Iterativa

1. Montar un servidor réplica de ieRed como equipo de pruebas.
2. Realizar una primera definición de criterios de búsqueda a partir de los requerimientos.
3. Búsqueda de Proyectos de CMSs Libres y Activos que cumpla con la mayor cantidad de requerimientos.

3 Página del Proyecto eCMS: http://gluc.unicauca.edu.co/wiki/index.php/Proyecto_eCMS

4. Selección y Evaluación de un pequeño grupo de CMSs.
5. Definición de los criterios finales para realizar la evaluación y escogencia de un CMS.
6. Implementación del CMS seleccionado en el servidor de ieRed.
7. Migración de la información al nuevo portal de ieRed.

Se planteó además realizar una búsqueda de manera heurística, que nos permitiera reevaluar paulatinamente las directrices trazadas, sirviéndonos de los sitios de Software Libre más reconocidos, los repositorios de proyectos y todas las herramientas de las que pudiéramos obtener información imparcial y consistente, esto principalmente por el deseo de hallar una solución propia y no guiada por recomendaciones externas respecto al CMS de moda o al que “X” persona o institución ha utilizado, por lo tanto se desechó desde el principio la opción de consultar Foros o Chats para los propósitos del trabajo.

La idea de realizar una exploración de manera heurística, surge precisamente de la inexperiencia en el ámbito de la evaluación de plataformas telemáticas y otro tipo de herramientas software, y del deseo de hacer un trabajo personalizado teniendo como base unas necesidades puntuales, un contexto específico, unas dinámicas propias y la posibilidad de elegir entre cientos de propuestas que se encuentran disponibles en el momento, pero con la idea clara de optar por Software Libre. Textualmente la Real Academia de la Lengua define como Heurístico: “Técnica de la indagación y del descubrimiento. En algunas ciencias, manera de buscar la solución de un problema mediante métodos no rigurosos, como por tanteo, reglas empíricas, etc.”⁴, siendo este el lineamiento que guió el proceso que se llevó adelante durante tres semanas y cuya experiencia deseamos compartir.

3. Desarrollo del Proyecto y Aprendizajes

Con los requerimientos en mente, se realizó una exploración preliminar en la web, que permitiera sentar las bases conceptuales relacionadas con los Sistemas Administradores de Contenidos.

“En informática, un sistema administrador de contenidos (CMS) es un sistema usado para organizar y facilitar la creación colaborativa de documentos y otros tipos de contenidos. Un CMS es frecuentemente una aplicación web usada para gestionar websites y contenido web, aunque en algunos casos, un sistema administrador de contenidos requiere un software cliente especial para editar y construir artículos. El mercado de los sistemas administradores de contenidos permanece fragmentado, con varias soluciones disponibles tanto de código abierto como propietarias⁵”.

4 Real Academia de la Lengua Española. Sito Web: <http://www.rae.es>

5 WIKIPEDIA. Content management system. [consultado en Septiembre de 2005]. Disponible en Internet: http://en.wikipedia.org/wiki/Content_management_system

En esta exploración se encontró una Lista de Sistemas Administradores de Contenidos tanto libres como open source en Wikipedia, que se tomó en su totalidad para comenzar la exploración de proyectos⁶, inicialmente realizando la búsqueda de cada uno en los repositorios más importantes de proyectos de Software Libre como son: Savannah, Sourceforge y Freshmeat⁷ –para enterarnos fundamentalmente del estado del proyecto (activo/inactivo), fecha del último lanzamiento, versión, licencia, plataformas, lenguajes, etc. Se exploraron alrededor de 100 proyectos, lo cual permitió además de todo tener una idea general de las características fundamentales implementadas y el estado de este tipo de tecnologías en la actualidad y de las nuevas iniciativas que cada día surgen.

Además de la información de los repositorios se realizó una comparación de la mayoría de los CMSs valiéndonos de la matriz de comparación CMS Matrix⁸ por tratarse de una valiosa herramienta que permite establecer comparaciones entre este tipo de proyectos con base en características importantes como son: requerimientos del sistema, seguridad, soporte, facilidad de uso, desempeño, gestión, interoperabilidad, flexibilidad, aplicaciones construidas (blogs, chat, listas, etc.), y soporte para aplicativos comerciales. Aunque los resultados de esta matriz no se analizaron en profundidad principalmente por falta de tiempo, los aspectos que se tuvieron en cuenta tiene que ver con requerimientos del sistema y aplicaciones construidas de manera particular, esto basados en los requerimientos iniciales que se buscaban.

Durante el proceso se fueron reevaluando los criterios establecidos, en la medida en que se tenía mayor conocimiento de los CMS y también por el hecho de no encontrar información de los proyectos suficiente o consistente en los repositorios, debiendo entonces acudir a la página principal del proyecto con el fin de realizar una exploración más detallada que nos diera luces del estado real de un desarrollo determinado. En algunos casos se encontró la posibilidad de acceder a un Demo de la herramienta o a Screen-Shots de las páginas de gestión de la misma, dándonos un nuevo elemento a tener en cuenta para la exploración, ya que nos daba una mejor idea de las interfaces que presenta la aplicación, la usabilidad de las mismas, funcionalidad requerida y esperada y la valiosa oportunidad de probar un aplicativo aunque sea en su funcionalidad más inmediata. Aunque no todos los proyectos pre-seleccionados ofrecían estas facilidades, se procuró tener en cuenta aquellos de los que se pudo realizar algún tipo de prueba preliminar basados en las mismas.

A medida que se avanzó en la indagación se realizó un primer documento que contenía el nombre del proyecto, su dirección web, la tecnología utilizada en su desarrollo (lenguaje de programación, tipo de base de datos utilizada), y un comentario propio del evaluador con miras a obtener un primer conjunto de proyectos pre-seleccionados. Paralelo a esto se realizó la comparación matricial de la mayoría de los CMSs, todo lo cual alimentó el documento en mención. Como consecuencia de esta primera fase se eligieron 19 proyectos iniciales, también enmarcados en un segundo documento con

6 WIKIPEDIA. List of content management systems. [consultado en Septiembre de 2005]. Disponible en Internet: http://en.wikipedia.org/wiki/List_of_content_management_systems

7 <http://savannah.gnu.org>, <http://sourceforge.net>, y <http://freshmeat.net>

8 CMS Matrix. Sitio Web: <http://cmsmatrix.org>

sus características más importantes, funcionalidad y prestaciones, con el fin de argumentar su elección y tener un punto de partida para acotar aún más la elección de la plataforma definitiva. Los CMSs pre-seleccionados fueron: Ariadne, Campsite, CMSimple, ezContents, ezPublish, InfoGlue, LucidCMS, Magnolia, Mambo, MediaWiki, MKDoc, MySource, OpenACS, Plone, SPIP, TikiWiki, Bitweaver, WebGUI, XOOPS.

Dentro de la evaluación realizada de manera general y en esta pre-selección en particular encontramos que existen proyectos con características, prestaciones y funcionalidades diversas, para soportar diferentes tipos de sitios con mucha estabilidad y robustez. Así, como hay CMSs para grandes sitios, los hay también para los más pequeños y personales, para periódicos, revistas y magazines, para sitios comerciales y empresariales, etc. Y a manera de muestra de aquello que se encontró quisiéramos resaltar las características más interesantes de algunos proyectos:

- Ariadne <http://www.ariadne-cms.org> Ariadne es un Servidor de Aplicaciones Web y un Sistema de Gestión de Contenidos multilinguaje construido con PHP. Tiene una rica interfaz de usuario que incluye wizards, menús pulldown, y un editor HTML WYSIWYG. Los datos son almacenados en un almacén estructurado de objetos y pueden ser accedidos a través de llamadas al sistema (ls, find, get). Soporta MySQL, PostgreSQL, y MSSQL.
- Campsite <http://campsite.campware.org> Bonito, amigable, multilinguaje, buen manejo de usuarios, es como para periódicos ya que la interfaz de usuario fue construida con el usuario final en mente (periodista, redactor, editor), y fue configurada para un diferentes perfiles y usuarios finales. Campsite sigue el paradigma de publicación print-newspaper, así que estructura los sitios por defecto en Publicaciones > Temas > Secciones> Artículos. Campsite trabaja mejor para publicaciones de tamaño mediano a grande, pero es capaz de manejar sitios más ágiles también.
- CMSimple <http://www.cmsimple.dk> Soporte español, buena documentación, liviano, sencillo. Es un CMS para pequeños sitios privados o comerciales, fácil de instalar y modificar ya que usa un archivo HTML para guardar el sitio entero, el cual puede ser editado con su editor favorito.
- ezContents <http://ezcontents.org> Para sitios medianos, buenas características, multilinguaje. Las características incluyen: representación frame/frameless; tres niveles de menús; seguridad granular; autores múltiples; módulos plug-in tales como un libro de visitantes, noticias, enlaces, menús y sub-menús, permisos y workflow.
- eZPublish http://ez.no/products/ez_publish_cms Idóneo para publicaciones de noticias, e-commerce (B2B y B2C), portales, y sitios web corporativos, intranets, y extranets. Por su notable modelo de contenidos completamente adaptable, y extensible es conveniente como plataforma general de desarrollos web.
- InfoGlue <http://www.infoglue.org> Es un CMS/Plataforma de Portales avanzado, robusto y escalable escrita 100% en Java. Es apropiado para un amplio rango de aplicaciones y organizaciones. Usos típicos incluyen websites públicos, portales, intranets y extranets.

- LucidCMS <http://lucidcms.net> Es un CMS simple y flexible para sitios personales u organizaciones que desean gestionar un conjunto de páginas web sin la sobrecarga y complejidad de otros CMSs disponibles, por su sencillez y usabilidad.
- Magnolia <http://www.magnolia.info/en/magnolia.html> Para sitios grandes, muy intuitivo, multilinguaje, las interfaces son muy bonitas. Es un CMS basado en JSR-170 –el API estándar para repositorios de contenidos java (JCR)-, disponible en 15 idiomas. Combina las ventajas del software basado en web (con cero instalación de clientes) con la usabilidad del software de escritorio. Todas las funciones pueden ser activadas fácilmente en menús de contexto o a través de drag'n'drop (arrastrar y soltar).
- Mambo <http://www.mamboserver.com> Mambo es una herramienta dinámica de contenido web que es capaz de construir sitios de varias páginas a varios miles de páginas. Viene completa en 10 módulos, un editor WYSIWYG, estadísticas del sitio, interfaz de administrador, soporte multilinguaje, soporte de módulos personalizables, gestión y aplicativo para subir medios (archivos), RSS, gestor de anuncios (banners), gestión de acceso jerárquico, etc.
- MKDoc <http://www.mkdoc.com> La principal característica que nos llamó la atención de este sitio es la posibilidad de construir comunidades. Además es un constructor de sitios Web y CMS que fue diseñado para animar el uso de buenas arquitecturas de información y la producción de sitios web accesibles a través de diferentes formas para que el público interactúe y navegue entre los documentos, incluyendo un mapa del sitio, facilidad de búsquedas y versión impresa de páginas. Toda la gestión, la creación de documentos, la edición, y la organización se realiza a través de un browser web, y las vistas se controlan usando plantillas Petal. Se usa Unicode/UTF-8 para soportar todos los idiomas, incluyendo aquellos que se escriben de derecha a izquierda.
- MySource <http://mysource.squiz.net> Tiene buena documentación, es modular y soporta principalmente sitios pequeños. Está diseñado para permitir a usuarios inexpertos y sin conocimientos tecnológicos construir y mantener sus propias soluciones online con seguridad, profesionalmente, y a bajo costo ya que una vez instalado no se requieren habilidades de programación para operar el sistema.
- OpenACS <http://openacs.org> OpenACS (Open Architecture Community System) es un toolkit para construir aplicaciones web escalables y orientadas a comunidades.
- Plone <http://plone.org> Plone es un CMS simple de instalar, mantener y modificar. Está diseñado un sistema de gestión de contenido corporativo. Es ideal como un servidor en intranet y extranet, como un sistema de publicaciones documento/Web, y como una herramienta groupware para trabajo colaborativo entre entidades localizadas en sitios dispersos.
- SPIP <http://www.spip.net> Es un sistema open source para publicaciones en Internet, que apunta principalmente a individuos, grupos informales y asociaciones sin ánimo de lucro. Permite la gestión y escritura colaborativa de sitios web que tienen la estructura de un magazín (por ejemplo artículos e historias cortas contenidas en secciones jerarquizadas), mientras no necesiten habilidades en HTML (exepcto para

definir la estructura de las plantillas). Hay varias características adicionales: foros públicos, peticiones, keywords, sindicación (RSS), un motor de búsquedas integrado, etc.

- TikiWiki <http://tikiwiki.org> Completo, actualizable, multilanguage, galeria de imagenes y archivos, documentación en español. Es un poderoso CMS y Groupware open-source que puede ser usado para crear cualquier tipo de aplicaciones web, sitios, portales, intranets, y extranets.
- WebGUI <http://www.plainblack.com/spreadwebgui> Es un CMS y una aplicación framework, multilinguaje. WebGUI es una estructura de gestión de contenidos construido para permitir que usuarios medios de negocios construyan y mantengan sitios web complejos. Es modular, pluggable, e independiente de la plataforma. Fué diseñado para permitir que las personas manejen el contenido online. WebGUI viene con un completo foro de discusión con la funcionalidad de phpBB o FUD Forum, más calendario de eventos, galería de fotos, blog, FAQ y gestión de listas de enlaces, y un sistema de perfiles y privilegios de usuario configurables.
- XOOPS <http://www.xoops.org> Es un programa que permite a los administradores la creación de sitios web dinámicos con gran contenido y muchas características excepcionales. Es ideal para desarrollar sitios web comunitarios pequeños y grandes, portales al interior de las compañías, portales corporativos, weblogs y mucho más. Puede ser instalado en un host con PHP –con un servidor web (por ejemplo, Apache) y una base de datos (por ejemplo MySQL).

Ya con esto en las manos se definieron y evaluaron unas características adicionales y definitivas en las plataformas pre-seleccionadas, a partir de requisitos más bien técnicos que funcionales, pero que fueron determinantes en la elección que se realizó; puntualmente se requirió que el proyecto estuviera contenido dentro de la Paquetería de Debian⁹, y que cumpliera con el estándar W3C¹⁰, para lo cual se exploró la página principal de cada uno de los proyectos sometiénola a la evaluación de dicho estándar y así mismo se actualizaron los repositorios para el servidor en el cual se montaría la plataforma y se indagó la existencia de cada uno dentro de la paquetería. El resultado final ha sido la elección de Plone como CMS que servirá de plataforma telemática a la Red de Investigación Educativa de la Universidad del Cauca, por ser el único que cumplió a cabalidad estos dos últimos requisitos.

Ahora bien, inicialmente se planteó realizar una exploración iterativa que nos permitiera decantar cada vez una pre-selección, hasta lograr elegir el CMS según nuestros criterios, pero por cuestiones de tiempo de las personas involucradas en el proyecto, no se pudo realizar la exploración de esta manera, debiéndonos limitar sólo a la información que se obtuvo de la primera revisión que se hizo.

9 Paquetes disponibles en Debian GNU/Linux. Sitio Web: <http://www.debian.org/distrib/packages>

10 Servicio de Validación de Marcado de la W3C. Sitio Web: <http://validator.w3.org>

4. Aprendizajes y Perspectivas

La idea fundamental desde el comienzo fue no realizar la selección de un CMS con base en las iniciativas o experiencias ajenas o en las recomendaciones de los foros u otros medios de este estilo, sino tener la posibilidad de ver un panorama de los proyectos partiendo desde una iniciativa libre de prejuicios al respecto, y evaluar con nuestros propios criterios todas las iniciativas encontradas, teniendo como base las necesidades del sitio que se pretende soportar. Ahora bien, un proceso de estas características se ha ido transformando poco a poco a medida que se ha avanzado en la indagación, pues se han ido encontrando más elementos que permiten explorar de mejor forma las características y prestaciones de los proyectos, dando como resultado muy seguramente la pérdida de información en los primeros proyectos evaluados, así como el desconocimiento de nuevos trabajos que están surgiendo con muy buenas características y prestaciones, pero que no estaban presentes en la lista sobre la cual se trabajó desde el comienzo. Además de todo, los elementos adicionales que se dieron a lo largo de la exploración y que no se aplicaron a los proyectos iniciales permitieron que la filtración se hiciera menos rigurosa con la posibilidad de profundizar en algunos proyectos y encontrar funcionalidad coherente con los requerimientos.

Ya con los proyectos pre-seleccionados lo que zanjó la decisión tuvo que ver fundamentalmente con las dos características técnicas que se evaluaron en cada uno de los 19, y donde solo Plone las cumplió a cabalidad. ¿Es por esto el mejor CMS del momento?, ¿el más robusto, adecuado o pertinente?. Realmente no se puede responder categóricamente, pues solo fue el resultado de nuestra propia exploración, de una experiencia que se construyó paso a paso, revaluando los lineamientos, las características esperadas, los factores a evaluar y todo aquello que nos diera herramientas para intentar obtener un CMS que bajo nuestra perspectiva fuera adecuado a unos propósitos determinados. Seguramente si realizáramos de nuevo esta experiencia empezaríamos por priorizar de diferente forma los requerimientos esperados del CMS -idea que surgió luego de socializar los resultados iniciales del proyecto al interior del GLUC, y que no habíamos contemplado desde el comienzo-, o la exploración partiría desde los Foros o con base en los proyectos utilizados en ámbitos similares al nuestro, o de repente nos daríamos la oportunidad de evaluar otro tipo de proyectos menos maduros o con versiones de prueba, pero realmente lo que tenemos es una elección que nos satisface fruto de un trabajo arduo y juicioso.

De todo esto se deriva el enriquecimiento obtenido a partir del trabajo realizado, que ayudan a conseguir soluciones de gran calidad que aunque no sean las más populares responden adecuadamente a necesidades específicas, y a aprovechar en toda la dimensión todas aquellas iniciativas que surgen en la comunidad del software libre y que alimentan de múltiples formas el mundo académico y empresarial.