



Gestión de la información en Educación Superior. Una experiencia de evolución del software con sistemas gestores de contenidos

Information management in Higher Education. An experience of changes in software evolution applied to content management systems

Ana E. Gómez Codutti

Universidad Nacional del Nordeste- Argentina
anagomezcodutti@gmail.com

Sonia I. Mariño

Universidad Nacional del Nordeste- Argentina
simarinio@yahoo.com

Pedro L. Alfonzo

Universidad Nacional del Nordeste- Argentina
plalfonzo@hotmail.com

RESUMEN.

La Educación Superior ofrece un vasto dominio para el diseño y desarrollo de soluciones tecnológicas orientadas a la gestión de la información. Se presenta una experiencia fundada en conceptos teóricos de la evolución del software, en particular su mantenimiento, actividad que permite modificar el mismo conservando su integridad. Esta actividad continúa durante su tiempo de vida, dado que en la práctica sufren cambios para su permanencia y utilidad; existiendo diferentes factores que inciden en su ejecución, lo que permiten clasificarlos en mantenimiento correctivo y perfectivo, entre otros. Los Sistemas Gestores de Contenidos o CMS son ampliamente aplicados en una diversidad de soluciones para mediar la comunicación con los usuarios finales. En la propuesta se resume una práctica atinente a la evolución del software centrada en el mantenimiento perfectivo aplicado en un sistema gestor de contenidos de código abierto.

PALABRAS CLAVES.

Educación Superior, sistemas de gestión, Ingeniería del Software, mantenimiento, software libre.

ABSTRACT.

Higher Education offers a vast domain oriented to design and development technologies solutions oriented to information management. The paper illustrates practical application based on theoretical concepts of software evolution. Particularly maintenance is an activity that allows modifying the same preserving its integrity is presented. This activity continues during the lifetime, since in practice undergo changes to its permanence and utility; there are different factors that affect their execution, which allow their inclusion in corrective and perfective maintenance, among others. The Content Management Systems or CMS are widely applied in a variety of solutions to mediate communication with end users. In the



proposal the practices pertaining to the evolution of software-centric perfective maintenance management systems applied in one open source content are summarized.

KEY WORDS.

Higher Education, Management Systems, Software Engineering, maintenance, free software.

1. Introducción.

La Educación Superior ofrece un vasto dominio para el diseño y desarrollo de soluciones tecnológicas orientadas a la gestión de la información y a mediar procesos de enseñanza aprendizaje. En carreras informáticas como la Licenciatura en Sistemas de Información se promueve la adquisición de conocimientos fundamentados en teorías y metodologías y con una sostenida formación en la resolución de problemas del mundo real. Es decir, dominios de conocimientos para los cuales los profesionales del sector de la Industria del Software deberán responder con soluciones tecnológicas apropiadas y oportunas.

Por otra parte, la Industria del Software Libre evoluciona y dispone a la comunidad de desarrolladores y usuarios una diversidad de soluciones comprendidas en las Tecnologías de la Información y Comunicación. Además, estas soluciones se sustentan en la Ingeniería del Software donde uno de los temas abordables es la evolución del software, en particular su mantenimiento, actividad que permite modificar el mismo conservando su integridad.

En este trabajo se presenta el marco teórico y una experiencia de evolución de Gestores de Contenidos construidos bajo la filosofía del software libre a fin de plasmar conceptos tratados previamente.

1.1. La Educación Superior como escenario del desarrollo de sistemas informáticos.

En la sociedad del conocimiento las herramientas de las Tecnologías de la Información y Comunicación (TIC) ofrecen una amplia gama de posibilidades para gestionar procesos de administración como procesos educativos (Bayo & Vallejos, 2015; Gavilondo Rodríguez, 2017; Ruiz Tapia, 2016),

Para los actores del sistema educativo, las TIC se han constituido en un desafío ineludible dado que han debido aprender a utilizar con miras a implementar innovadoras estrategias orientadas a apoyar proceso de transformación de datos en información orientados a la toma de decisiones.

En el mercado existe una diversidad de sistemas informáticos de apoyo a procesos administrativos que abarcan una o varias situaciones de gestión. Algunos de ellos son particularmente diseñados mientras que otros son artefactos estándar que deben parametrizarse.

Cabe aclarar que estos artefactos software son diseñados, desarrollados y mantenidos siguiendo directrices de la Ingeniería del Software. Por lo expuesto, es viable establecer una vinculación entre la Ingeniería del Software y la Informática Educativa, dado que la primera proporciona los métodos y las herramientas para la construcción de artefactos tecnológicos dirigidos a apoyar procesos administrativos y de enseñanza – aprendizaje.



1.2 La Ingeniería del Software en el desarrollo de sistemas informáticos

La Ingeniería del Software (IS) como una de las disciplinas de la Ciencia Informática (RedUNCI, 2014), comprende los aspectos de la producción de software desde las etapas iniciales de la especificación de un sistema, hasta el mantenimiento de éste después de su implementación (Sommerville, 2011; Pigoski, 2015; April & Abran, 2008).

Asociaciones profesionales como la IEEE Computer Society (IEEE, 2015) y la Association for Computing Machinery (ACM, 2015) han aunado sus trabajos y desarrollaron una guía del Cuerpo de Conocimientos de la Ingeniería de Software o SWEBOK (Software Engineering Body of Knowledge).

La guía SWEBOK V3.0 (Bourque & Fairley, 2014) presenta actualizaciones en todas las áreas de conocimiento (KAS-por sus siglas en inglés-) a fin de reflejar los cambios de la Ingeniería de Software, previamente publicados en la guía SWEBOK2004.

Bourque & Fairley (2014) en la guía SWEBOK V3.0 exponen 15 áreas del conocimiento, identificadas como: Requisitos de Software, Diseño de software, Construcción de software, Pruebas de software, Mantenimiento del software, Gestión de la Configuración de Software, Gestión de Ingeniería de Software, Proceso de Ingeniería de Software, Modelos y Métodos de Ingeniería de Software, Calidad del software, Práctica Profesional de la Ingeniería de Software, Economía de Ingeniería de Software, Fundamentos de Computación, Fundamentos matemáticos, Fundamentos de Ingeniería.

El objetivo del mantenimiento del software (MS) es modificar el software existente conservando su integridad (Bourque & Fairley, 2014). El Instituto de Ingeniería Eléctrica y Electrónica (IEEE, 1993) define al mantenimiento del software como la modificación de un producto software, después de su entrega para corregir fallos, mejorar el rendimiento u otros atributos, o para adaptar el producto a otro entorno. Por ello, según April & Abran (2008); Polo et al. (2003); Sommerville (2011), el proceso de desarrollo del software continúa durante su tiempo de vida, dado que en la práctica sufren cambios para su permanencia y utilidad.

En SWEBOK se menciona que el proceso de mantenimiento, facilita las acciones necesarias y definidas para las entradas y salidas a esas actividades, las que se describen en las normas de mantenimiento del software como es la IEEE STD 1219 (IEEE, 1993) e ISO/IEC 14764 (ISO/IEC, 2006).

La norma IEEE 14764 (ISO/IEC, 2006) identifica las actividades primarias del mantenimiento de software, denominándolas como la implementación de procesos, análisis del problema y modificación, implementación de la modificación, revisión del mantenimiento y aceptación, migración y retiro (Bourque & Fairley, 2014).

Existen diferentes factores que inciden en la ejecución del Mantenimiento del Software. En IEEE (1993) se establecen los siguientes tipos: i) Mantenimiento adaptativo: modificación de un producto de software que se realiza después de su entrega para mantener un programa operativo mientras se cambia el entorno de producción; ii) Mantenimiento correctivo: modificación de un producto software realizado después de su entrega para corregir fallas encontradas; iii) Mantenimiento de emergencia: implica un mantenimiento correctivo no programado, ejecutado para mantener un sistema operacional; iv) Mantenimiento perfectivo: modificación de un producto software después de su entrega



para mejorar su rendimiento o mantenibilidad. Además, en la ISO/IEC (2006) se considera el mantenimiento preventivo, el cual consiste en modificar un producto de software después de entregado para detectar y corregir fallas latentes antes de que se conviertan en fallas funcionales.

Los Sistemas Gestores de Contenidos o CMS son ampliamente aplicados en una diversidad de soluciones para mediar la comunicación con los usuarios finales. En el mercado actual existen diversas herramientas comprendidas en esta categoría del software de libre distribución ampliamente utilizadas por la comunidad de programadores. Entre algunos se mencionan Joomla, Drupal, Moodle, PhpFusion. Además, se han reportado experiencias de su empleo en diversos dominios del conocimiento.

Se coincide con Chapin et al. (2010) citado en Ramírez et al. (2015), en que las modificaciones pueden realizarse a los efectos de adaptar el software a las necesidades específicas de los usuarios, incorporarlo a desarrollos más amplios o corregir errores; estas alternativas vinculan el estudio sobre FLOSS con la investigación sobre el mantenimiento y la evolución.

Las organizaciones se esfuerzan para maximizar el rendimiento de su inversión en el desarrollo de software, manteniendo el software operativo el mayor tiempo posible (Bourque & Fairley, 2014). Desde la Ingeniería del Software y particularmente desde la evolución como parte integral del ciclo de vida, es relevante disponer de software operativo el mayor tiempo posible, maximizando las inversiones de las organizaciones. Sin embargo, no todos los CMS como soluciones configurables son adecuadas a los usuarios finales.

En la propuesta se describe una experiencia atinente a la evolución del software centrada en el mantenimiento perfectivo aplicado en un sistema de contenidos FLOSS (F/OSS) diseñado para un contexto de Educación Superior.

2. Metodología.

En la Ingeniería del Software un tema de interés es el estudio de métodos y herramientas para la construcción y mantenimiento de aplicaciones informáticas. En una primera etapa se realizó investigación bibliográfica, centrándose en la evolución del software como un aspecto de la Ingeniería del Software.

En la comunidad del software libre se disponen de una diversidad de herramientas que promueven desde una perspectiva cooperativa y colaborativa el compartir e incrementar el conocimiento, éste se plasma en la diversidad de herramientas disponibles. Además, se difunde el uso de los mismos en soluciones tecnológicas adaptadas a diversos dominios.

Un claro ejemplo son los Gestores de Contenidos de libre distribución, en este trabajo se optó por Drupal. Para generar un prototipo adecuado a ciertos requerimientos de administración de información generada en espacios de Educación Superior y validar la propuesta, se estudió, analizó y aplicó el estándar IEEE 1219 (IEEE, 1993) adaptándose al caso objeto de la presente ponencia.





3. Propuesta de evolución del software

La industria del software libre ha promovido una diversidad de soluciones orientadas a distintos dominios, posibilitando la construcción e implementación de artefactos para mediar una diversidad de procesos.

Estos desarrollados –construidos por una comunidad orientada a promover y compartir sus ideaciones- son susceptibles de adecuar a distintos contextos. Por ello, en este trabajo se sintetiza la propuesta desarrollada.

En esta sección se describe la experiencia que ilustra conceptos teóricos de la evolución del software con la finalidad de atender a: la formación de recursos humanos en la temática, dado que se ha profundizado en aspectos teóricos tratados en cursos de grado y en la transferencia de tecnología. Esta última plasmada en la generación de artefactos software.

Por lo expuesto, en la Figura 1 se visualiza el proceso de modificación y las fases propuestas por el estándar IEEE 1219, documentándose en cada una de ellas los cambios realizados. En la Tabla 1, se presentan las actividades asociadas e implementadas en cada fase mencionada en la Figura 1 y sustentadas en el mencionado estándar que guía el desarrollo tecnológico de la propuesta.

Fases	Actividades
Identificación del problema	<ul style="list-style-type: none"> Identificar el problema. Se planteó la necesidad de diseñar interfaces que faciliten el despliegue de información específica. Asignar prioridad. Se estableció como primacía la redefinición de la interfaz inicial y la localización de información específica. Obtener aprobación de la solicitud de modificación y las tareas a llevar a cabo. Se analizó la solicitud y se procedió a su aceptación. Estimar inicialmente los recursos necesarios para modificar el sistema existente. Básicamente se asociaron a la disponibilidad de una computadora personal y conexión a la web para su despliegue en la mencionada interfaz.
Análisis	<ul style="list-style-type: none"> Se identificaron los elementos a modificar. A través de prototipos se plantearon soluciones alternativas y como estos cambios impactan en la información contenida en la base de datos y en el funcionamiento del sistema. Los perfiles de usuarios definidos se ilustran en la Figura 2, que impactaron en algunos cambios introducidos. Evaluar el impacto. Evaluar los costos. Estudiar la viabilidad y el alcance de las modificaciones. Desarrollar un plan preliminar de diseño. implementación, pruebas y liberación del software. Desarrollar estrategias de prueba.
Diseño	<ul style="list-style-type: none"> Diseño de las interfaces a partir de componentes del CMS y de las modificaciones planteadas.
Implementación	<ul style="list-style-type: none"> Desarrollar y probar las modificaciones realizadas. Codificar y generar pruebas unitarias. Integrar el software modificado con el sistema existente. Analizar el riesgo. Revisar la preparación para las pruebas.

Tabla 1. Fases y actividades relacionadas con estándar IEEE 1219 (Fuente: elaboración propia).





Fases	Actividades
Pruebas del Sistema	<ul style="list-style-type: none"> Realizar pruebas sobre el sistema modificado. Revisar integridad. Obtener aprobación
Pruebas de Aceptación	<ul style="list-style-type: none"> Realizar pruebas sobre el sistema completamente integrado. Se desarrollan incrementalmente y en paralelo al mantenimiento perfectivo aplicado en el módulo seleccionado.
Liberación del Producto	<ul style="list-style-type: none"> Desarrollar un plan. Notificar a los usuarios. Realizar una copia de seguridad de la versión del sistema. Realizar la instalación y capacitar a los usuarios. En proceso, se elabora plan para la liberación del producto. En referencia a su difusión, se entiende este trabajo como un producto derivado de actividades de I+D y se planifica un curso de capacitación en la temática.

Tabla 1. Fases y actividades relacionadas con estándar IEEE 1219 (Fuente: elaboración propia).

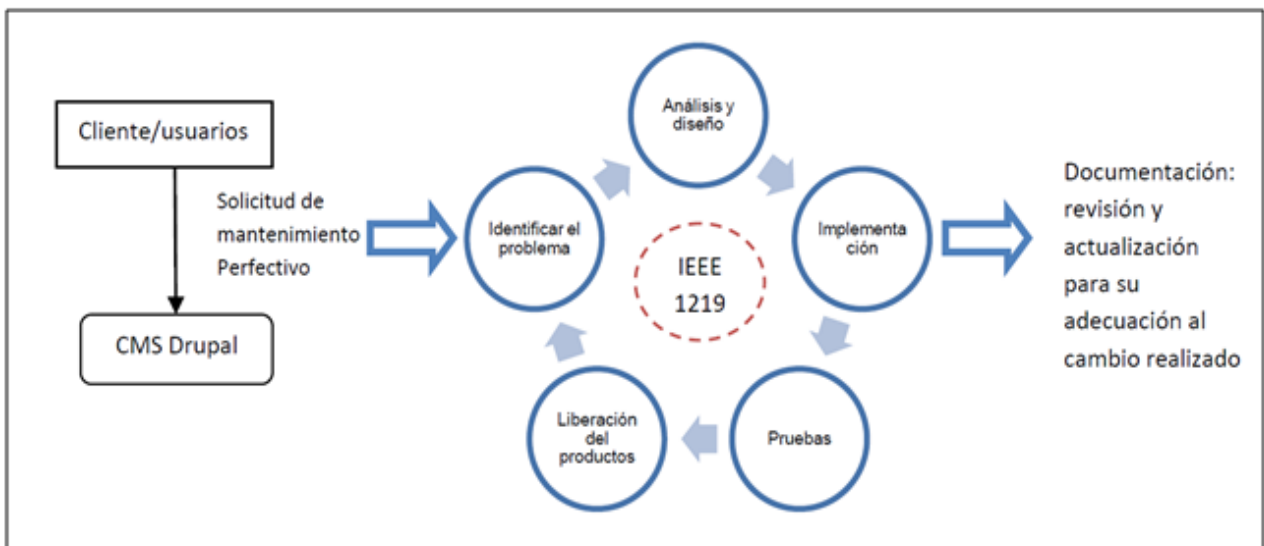


Figura 1. Proceso de solicitud de modificación (Fuente: elaboración propia).

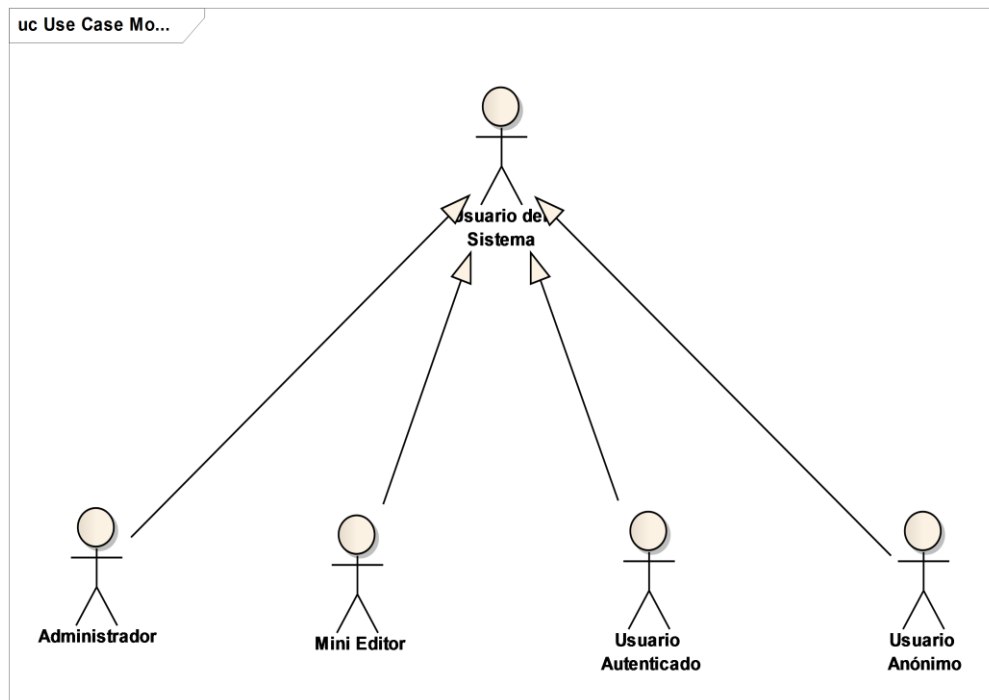


Figura 2. Diagrama de jerarquía de usuarios del sistema (Fuente: elaboración propia)

Drupal, como CMS de libre distribución puede ser mejorado por la comunidad de desarrolladores. En este trabajo, se ilustra el mantenimiento perfectivo aplicado en un desarrollo informático de gestión de adscripciones basado en el mencionado CMS.

Por defecto, el CMS Drupal incorpora los siguientes roles de usuario:

- Anónimo: puede visualizar o buscar los contenidos sin autenticarse.
- Autenticado: puede incluir comentarios sobre los productos.
- Administrador: responsable técnico del sistema, pudiendo acceder a cualquiera de los servicios del mismo, para configurar o modificar cualquier parámetro de éste.

Además, con miras a asegurar el acceso seguro a la información se incluyeron distintos perfiles de usuarios, representados en la Figura 2. Así se observó la necesidad de adaptar estos parámetros iniciales que incorpora el CMS de acuerdo a las características de los roles de usuario requeridos por un sistema informático aplicado al dominio de la Educación Superior en cuestión (Figura 2), con el propósito de incorporar funcionalidades al sistema que facilitarían su mantenimiento.

De esta manera, se incorporó el rol “Mini Editor”, quien gestiona los contenidos del sitio – particularmente, de las adscripciones - quien no necesariamente debe poseer nociones técnicas del funcionamiento del sistema. En la Figura 3 se visualizan los roles de usuarios personalizados. El mencionado cambio implicó la alteración de la tabla de la base de datos correspondiente a los roles de usuario, donde se agregó el registro “Mini Editor”.



A partir de los roles creados, el sistema permite dar de alta nuevos usuarios, categorizarlos y asignarles permisos para realizar determinadas operaciones.

El CMS dispone de escasas funcionalidades para la búsqueda de información, no contemplando filtros o categorías que acoten los resultados. A fin de mejorar esta opción, en esta sección se muestran las modificaciones efectuadas.

Se incorporaron determinados criterios de filtrado para facilitar y agilizar la búsqueda de contenidos. Para implementar esta funcionalidad, se debió indexar el contenido del sitio web y brindar al usuario un mecanismo para ejecutar dicha búsqueda. La generación del mecanismo requirió la creación de una "Vista" o View, un constructor de consultas, seleccionando en la configuración su origen, campos que se visualizarán, criterios de filtrado y de ordenación, entre otros elementos. Con respecto a la indexación del sitio, se efectúa ejecutando el módulo "cron", un planificador automático de tareas que incorpora por defecto el CMS. En la Figura 4, se observa el formulario de búsqueda modificado.

A los fines de incrementar mejoras en el rendimiento del CMS, específicamente en cuanto al despliegue de información, se incorporó la generación de reportes, proporcionando diferentes opciones de visualización dinámica de la información.

Como formato estándar, el CMS dispone de un módulo compuesto de varias secciones (Figura 5), donde la configuración de sus parámetros contribuye a determinar el aspecto y comportamiento de la Vista. Así, en la sección Formato se estableció que los resultados sean estructurados en una tabla. Se configuró la sección "Campos" a fin de seleccionar los campos de la base de datos a mostrar. Con respecto a la sección "Criterios de Filtrado", se eligieron diversos criterios de acceso a la información que permiten visualizar los reportes de manera dinámica, pudiendo el usuario optar por uno o más criterios.

En la sección "Criterios de ordenación", se estableció el despliegue de los resultados en orden descendente respecto al año de adscripción. Se modificó la opción "Comportamiento si no hay resultados", estableciendo el mensaje de error a desplegar al usuario en caso de carecer de resultados que coincidan con el Criterio de Filtrado elegido.

Otra sección modificada fue "Paginador", donde se indicó un paginado compuesto por 5 elementos. La Figura 6 ilustra los cambios implementados.





NOMBRE	OPERACIONES	
+ usuario anónimo (bloqueado)	editar permisos	
+ usuario autenticado (bloqueado)	editar permisos	
+ administrator	editar rol	editar permisos
+ minieditor	editar rol	editar permisos
<input type="text"/>	<input type="button" value="Añadir rol"/>	
<input type="button" value="Guardar pedido"/>		

Figura 3. Roles de usuarios del sistema (Fuente: elaboración propia).

Inicio Adscripciones Régimen de Adscripciones Contacto

Navegación

- Buscar adscripción
- Reportes

Inicio » Búsqueda de Adscripciones

Búsqueda de Adscripciones

Ingrese apellido del adscripto

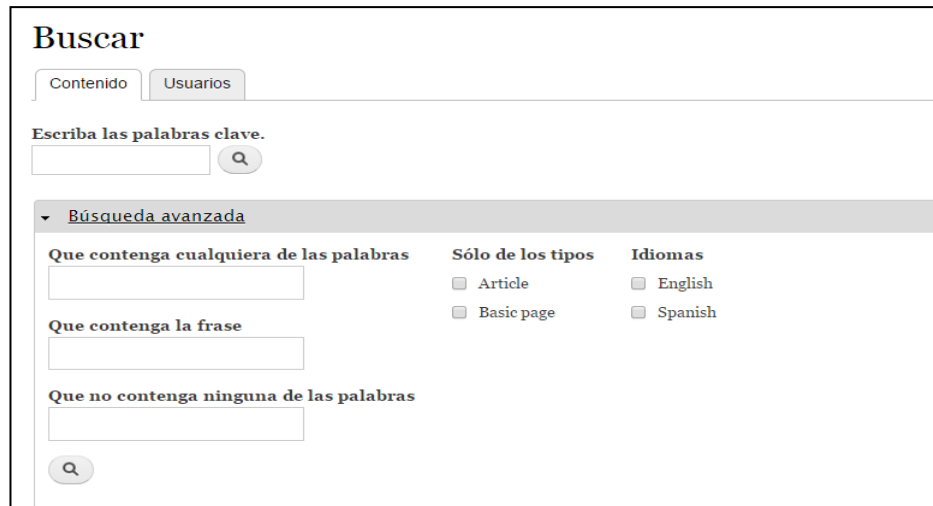
Ingrese apellido del Director

Ingrese DNI del Adscripto

Ingrese LU del Adscripto

Ingrese Resolución

Figura 4. Formulario de búsqueda personalizado para usuarios (Fuente: elaboración propia).



Buscar

Contenido Usuarios

Escriba las palabras clave.

Búsqueda avanzada

Que contenga cualquiera de las palabras

Que contenga la frase

Que no contenga ninguna de las palabras

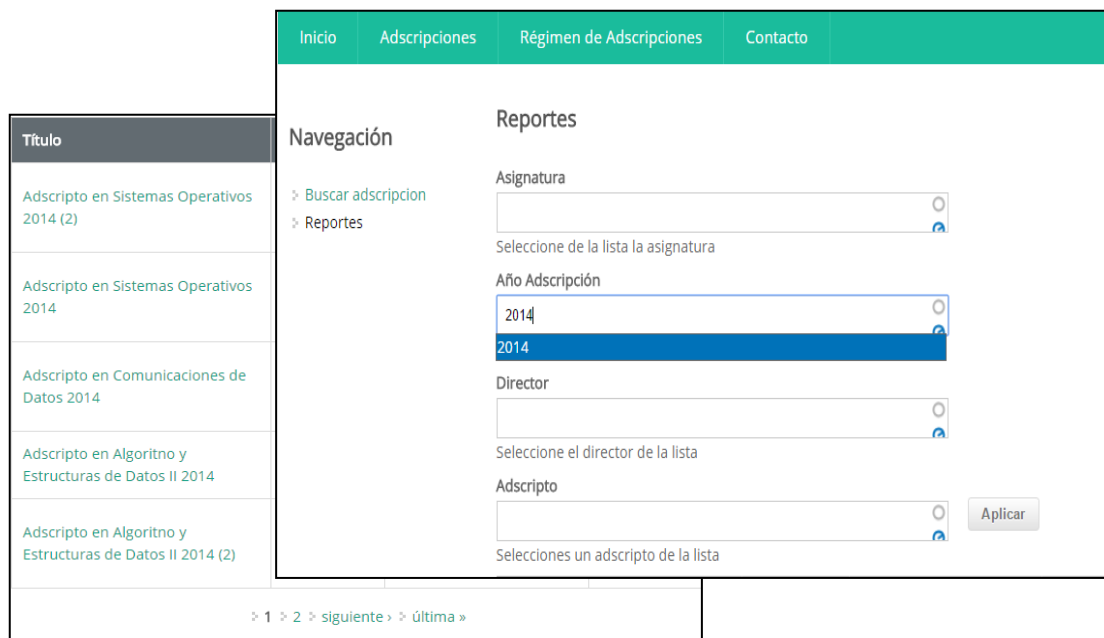
Sólo de los tipos

- Article
- Basic page

Idiomas

- English
- Spanish

Figura 5. Formulario estándar de acceso a la información (Fuente: elaboración propia).



Inicio Adcripciones Régimen de Adcripciones Contacto

Título

- Adscripto en Sistemas Operativos 2014 (2)
- Adscripto en Sistemas Operativos 2014
- Adscripto en Comunicaciones de Datos 2014
- Adscripto en Algoritmo y Estructuras de Datos II 2014
- Adscripto en Algoritmo y Estructuras de Datos II 2014 (2)

Navegación

- Buscar adscripcion
- Reportes

Reportes

Asignatura

Seleccione de la lista la asignatura

Año Adscripción

2014

Director

Seleccione el director de la lista

Adscripto

Selecciones un adscripto de la lista

Aplicar

1 2 siguiente última

Figura 6. Formulario personalizado para despliegue de información (Fuente: elaboración propia).

4. Conclusiones y futuros trabajos.

El incremento de soluciones tecnológicas, el paradigma del código abierto y los cambiantes requerimientos de los usuarios, se ha reflejado en la necesidad de localizar nuevas estrategias para brindar artefactos software de real utilidad.

La evolución del Sector de Servicios Informáticos y los recursos humanos formados en ámbitos universitarios se refleja en la adopción de estrategias orientadas a desarrollar métodos más eficientes de mantenimiento de sus soluciones tecnológicas implementadas para la captura, el almacenamiento, el procesamiento y la difusión de la información.

En este trabajo se expuso un caso de mantenimiento o evolución del software, principalmente diseñado y ejecutado desde espacios de Educación Superior con miras a mejorar las funcionalidades previstas en sistemas gestores de contenidos de libre distribución.

Se continuará trabajando en esta línea de la Ingeniería del Software integrándose a cuestiones de la Educación Superior a fin de contribuir con experiencias que fortalezcan la generación de artefactos desde la Industria del Software y la transferencia de estos artefactos al contexto de la influencia universitaria.

Referencias bibliográficas.

- ACM. (2015). *Association for Computing Machinery*. Recuperado de: <https://www.acm.org>
- April, A. & Abran, A. (2008). *Software Maintenance Management: Evaluation and Continuous Improvement*. USA: IEEE Computer Society.
- Bayo, M. & Vallejos, O. (2015). La Tecnicatura Universitaria en Software Libre de la Universidad Nacional del Litoral: fundamentos, lineamientos curriculares y desafíos de su implementación. *Simposio Argentino sobre Tecnología y Sociedad (STS) – 44 Jornadas Argentinas de Informática e Investigación Operativa*.
- Bourque, P. y Fairley, R E. (2014). *SWEBOK Guide to the Software Engineering Body of Knowledge, Version 3.0*. IEEE Computer Society. Recuperado de: www.swebok.org
- Gavilondo Rodríguez, C. E. (2017). Las TIC en el proceso de enseñanza: reflexiones acerca de su aplicación. *Question*, 53(1), 256-273.
- Chapin N., Hale J E., Fernandez-Ramil J. y Tan, W, G. (2001). Types of software evolution and software maintenance. *J. Softw. Maint. Evol. Res. Pract.*, 13(1), 3–30.
- Drupal. *Open source CMS*. Recuperado de: <https://drupal.org/>
- GNU (2016). *Operating System. What is free software?* Recuperado de: <http://www.gnu.org/philosophy/free-sw.en.html>
- IEEE (2015). Computer Society. Recuperado de: <https://www.acm.org/>
- IEEE (1993). STD 1219:1993. Standard for Software Maintenance. USA: IEEE Computer Society Press.
- ISO/IEC Standard 14764: 2006. *International Organisation for Standardization*. Software Engineering – Software Life Cycle Processes – Maintenance.
- Joomla (2016). Recuperado de: <http://www.joomla.org/>
- Moodle (2016). Recuperado de: <https://moodle.org/?lang=es>





- PhpFusion (2016). Recuperado de: <https://www.php-fusion.co.uk/home.php>
- Pigoski, T M. (2015). SWEBOK Knowledge Area Description for Software Evolution and Maintenance. Recuperado de: [http://kopustas.elen.ktu.lt/~rsei/Swebok/KA_Description_Software_Evolution_Maintenance\(Version_0_5\).PDF](http://kopustas.elen.ktu.lt/~rsei/Swebok/KA_Description_Software_Evolution_Maintenance(Version_0_5).PDF)
- Polo, M., Piattini, M. & Ruiz, F. (2003). *Advances in Software Maintenance Management: Technologies and Solutions*. Ed. Idea Group Inc.
- Ramirez J., Reyes C., Gil G. & Durgam F. (2015). Evolución y reusabilidad en FLOSS. *XVII Workshop de Investigadores en Ciencias de la Computación*. Recuperado de: http://sedici.unlp.edu.ar/bitstream/handle/10915/45896/Documento_completo.pdf?sequence=1
- REDUNCI. (2006). Red de Carreras de Grado en Informática. *Propuesta de Currícula*. Recuperado de: <http://redunci.info.unlp.edu.ar/docs/propuesta.doc>.
- Ruiz Tapia, J. A., Martínez Ávila, M. & Sánchez Paz, M. (2016). El impacto de las TICs en la calidad de la educación superior, *Revista de Investigación en Ciencias Contables y Administrativas*, 1(1), 28-44.
- Sommerville, I. (2011). *Ingeniería del Software*. México: Pearson Educación.

