Proyecto Fin de Carrera
Ingeniería de Telecomunicación

DISEÑO DE UNA APLICACIÓN WEB PARA EL TRATAMIENTO DE ARCHIVOS PDF ONLINE

Autor: Francisco José Garrido Sannicolás

Tutor: **Pablo Aparicio Ruiz**

Luis Onieva Giménez

Dep. Organización Industrial y Gestión de Empresas II
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2015

Proyecto Fin de Carrera
Ingeniería de Telecomunicación

DISEÑO DE UNA APLICACIÓN WEB PARA EL TRATAMIENTO DE ARCHIVOS PDF ONLINE

Autor:

Francisco José Garrido Sannicolás

Tutor:

Pablo Aparicio Ruiz

Luis Onieva Giménez

Dep. Organización Industrial y Gestión de Empresas II

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2015

Proyecto Fin de Carrera: DISEÑO DE UNA APLICACIÓN WEB PARA EL TRATAMIENTO DE ARCHIVOS PDF ONLINE

Autor: Francisco José Garrido Sannicolás

Tutor: **Pablo Aparicio Ruiz**

Luis Onieva Giménez

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2015

El Secretario del Tribunal

A todos los que siguieron creyendo en mí, a pesar de los retrasos; y en especial a mis padres.

ÍNDICE

ÍNDICE **9**

ÍNDICE DE TABLAS **11**

ÍNDICE DE FIGURAS **12**

1 INTRODUCCIÓN **15**

2 OBJETIVOS Y ANÁLISIS **17**

2.1 OBJETIVOS DEL PROYECTO *17*

2.2 MOTIVACIÓN DEL PROYECTO *19*

2.3 ANÁLISIS *20*

2.3.1 ANÁLISIS DE ANTECEDENTES 20

2.3.2 APORTACIÓN REALIZADA 22

2.3.3 ANÁLISIS TEMPORAL Y DE COSTES 24

2.3.4 ANÁLISIS DE TECNOLOGÍAS INVOLUCRADAS 25

3 BASE DE DATOS **29**

3.1 DISEÑO *29*

3.1.1 NOMENCLATURA USADA POR EL FRAMEWORK SYMFONY 29

3.1.2 BASE DE DATOS 30

3.1.3 COPIA DE LA APLICACIÓN WEB ORIGINAL 31

3.1.4 COPIA DE LA BASE DE DATOS MODIFICADA 34

3.1.5 COMPARACIÓN DE AMBOS MODELOS 35

3.1.6 OTRAS MODIFICACIONES DE LA BASE DE DATOS 36

3.2 IMPLEMENTACIÓN *40*

4 PROCESADO AUTOMÁTICO DE NUEVOS ARTÍCULOS **46**

4.1 DISEÑO *46*

4.2 IMPLEMENTACIÓN *48*

4.2.1 MODIFICACIÓN DE LOS FORMULARIOS DE CREACIÓN DE UN NUEVO ARTÍCULO 48

4.2.2 MODIFICACIÓN DEL PROCESO DE CREACIÓN DE UN NUEVO ARTÍCULO 53
4.2.3 OTROS CAMBIOS REALIZADOS 59

5 PARSEO AUTOMÁTICO DE NUEVOS ARTÍCULOS 61

5.1 DISEÑO 61

5.2 IMPLEMENTACIÓN 63

5.2.1 FUNCIÓN DE PARSEO 64

5.2.2 REGLAS DE PARSEO 66

6 MANUAL DE USUARIO 75

6.1 NUEVO ARTÍCULO 75

6.2 NUEVA REGLA DE PARSEO 77

7 VALIDACIÓN Y PRUEBAS 80

7.1 VERIFICACIÓN GENERAL DE LA APLICACIÓN 80

7.2 REGLAS DE PARSEO CIO 2013 81

7.2.1 CIO2013_ENGLISH_TRACKS 81

7.2.2 Resumen de resultados de la regla de parseo CIO2013_ENGLISH_TRACKS 85

7.2.3 CIO2013_SPANISH_TRACKS 86

7.2.4 Resumen de resultados de la regla de parseo CIO2013_SPANISH_TRACKS 93

7.3 REGLAS DE PARSEO CIO 2014 94

7.3.1 CIO2014_FULL_PAPER_ENGLISH 94

7.3.2 Resumen de resultados de la regla de parseo CIO2014_FULL_PAPER_ENGLISH 98

7.3.3 CIO2014_FULL_PAPER_ESPAÑOL 98

7.3.4 Resumen de resultados de la regla de parseo CIO2014_FULL_PAPER_ESPAÑOL 103

7.3.5 CIO2014_EXTENDED_ABSTRACTS 104

7.3.6 Resumen de resultados de la regla de parseo CIO2014_EXTENDED_ABSTRACTS 108

8 CONCLUSIONES 109

9 FUTURAS MEJORAS Y AVANCES 111

BIBLIOGRAFÍA 113

REFERENCIAS ONLINE 114

ÍNDICE DE TABLAS

Tabla 2-1. Estimación temporal del Proyecto. 24

Tabla 2-2. Estimación de costes del Proyecto. 25

Tabla 3-1. Estructura predeterminada de archivos y directorios de los proyectos **SYMFONY**. 32

Tabla 5-1. Caracteres con un significado especial en expresiones regulares. 70

Tabla 7-1. Resumen de resultados de la regla de parseo CIO2013_ENGLISH_TRACKS. 86

Tabla 7-2. Resumen de resultados de la regla de parseo CIO2013_SPANISH_TRACKS. 93

Tabla 7-3. Resumen de resultados de la regla de parseo CIO2014_FULL_PAPER_ENGLISH. 98

Tabla 7-4. Resumen de resultados de la regla de parseo CIO2014_FULL_PAPER_ESPAÑOL. 104

Tabla 7-5. Resumen de resultados de la regla de parseo CIO2014_EXTENDED_ABSTRACTS. 108

Tabla 8-1. Resumen de los resultados obtenidos. 110

ÍNDICE DE FIGURAS

Figura 1-1. Logotipo de **ADINGOR**. 16

Figura 2-1. Diagrama del proceso genérico planteado inicialmente como solución. 18

Figura 2-2. **LogicalDOC**. 20

Figura 2-3. Diagrama del proceso genérico de la aplicación web original. 21

Figura 2-4. Diagrama del proceso genérico de la aplicación desarrollada. 23

Figura 2-5. Logotipo de **SYMFONY**. 26

Figura 3-1. Estructura predeterminada de archivos y directorios de los proyectos **SYMFONY**. 31

Figura 3-2. Copia de la aplicación web original en un entorno de desarrollo. 34

Figura 3-3. Diagrama del proceso genérico planteado inicialmente como solución. 37

Figura 3-4. Ejemplo de artículo en formato **PDF**. 44

Figura 3-5. Ejemplo de la presentación de un artículo en el "frontend". 45

Figura 4-1. Carpeta **LIB** de la estructura predeterminada de archivos. 47

Figura 4-2. Formulario para añadir un nuevo artículo. 50

Figura 4-3. Detalle del formulario para añadir un nuevo artículo. 52

Figura 4-4. Ejemplo de la presentación de un artículo en el "backend". 53

Figura 4-5. Nuevo menú **Gestor de Artículos** en el "backend". 60

Figura 5-1. Implementación de las reglas de parseo. 65

Figura 5-2. Ejemplo de artículo en formato **PDF**. 67

Figura 5-3. Plantilla de los artículos del **CIO 2014**. 68

Figura 6-1. Nuevo menú **Gestor de Artículos** en el "backend". 75

Figura 6-2. Formulario para añadir un nuevo artículo. 76

Figura 6-3. Formulario de verificación de nuevos artículos. 77

Figura 6-4. Detalle del formulario para añadir un nuevo artículo. 78

Figura 6-5. Implementación de las reglas de parseo. 78

Figura 7-1. Texto sin saltos de línea extraído de un artículo del **CIO 2013**. 86

Figura 9-1. Diagrama del proceso genérico planteado inicialmente como solución. 111

Figura 9-2. Diagrama del proceso genérico de la aplicación desarrollada. 111

1 INTRODUCCIÓN

El mundo actual vive las consecuencias de la conocida como **"Era de la Información"**; que se define formalmente como el periodo en el que el flujo de información se volvió más rápido que el propio movimiento físico. Comienza en la segunda mitad del siglo XIX con la invención de la telegrafía y el teléfono y culmina con la aparición de Internet. Una de estas consecuencias es la influencia del conocimiento como elemento fundamental para generar valor y riqueza por medio de su trasformación a información, creando un valor añadido en los productos y servicios en cuyo proceso de creación o transformación participa.

Aunque la importancia del conocimiento ya es reconocida en el siglo XIX, su influencia en el panorama económico no se hace notar de forma significativa hasta fechas muy recientes. En los análisis de la evolución de las economías de las últimas décadas es posible apreciar cómo hay una tendencia generalizada a depender cada vez más del conocimiento y de la información; convirtiéndose en motor del crecimiento económico y de la mejora de la productividad y, por tanto, en un claro elemento diferenciador *(Barceló Llauger, 2001)*.

Precisamente en este contexto se presenta este Proyecto; dada la importancia del conocimiento y la información para Empresas y Organizaciones, surge la necesidad de garantizar el acceso a dichos recursos de una manera cómoda y sencilla; a la vez que segura. En concreto es la **Asociación para el Desarrollo de la Ingeniería de Organización** (**ADINGOR**) la que busca una herramienta que le permita clasificar y publicar de forma ordenada la gran cantidad de documentación que se genera en los **Congresos de Ingeniería de la Organización** (**CIOs**).

La **Asociación para el Desarrollo de la Ingeniería de Organización** (**ADINGOR**) [1] es una entidad sin ánimo de lucro cuyo objetivo básico es contribuir al desarrollo y difusión de conocimientos teóricos y de aplicación práctica, relativos al diseño, instalación, funcionamiento, gestión, control y mejora de sistemas (industriales y de prestación de servicios, tales como Empresas y otras Organizaciones) integrados por personas, equipos, materiales, información, energía y recursos financieros, en un contexto de servicio a los usuarios, de atención a las necesidades e intereses de otros implicados y afectados, y de respeto al medio ambiente.

Figura 1-1. Logotipo de **ADINGOR**.

ADINGOR agrupa a los profesionales de la Gestión y de la I+D, tanto de las Empresas y Organizaciones como de las Universidades, que están de acuerdo con los objetivos definidos. En particular, los profesionales y profesores que actúen en el ámbito de la Ingeniería de Organización, con visión, formación y conocimientos de ingeniería, y que por ello, tengan presente en sus enfoques el papel de las distintas visiones (de gestión, socioeconómicas, de personal, tecnológicos, etc.) en el diseño y funcionamiento de los sistemas Empresa-Organizaciones.

Como objetivo operativo, la Asociación se ocupa de convocar y organizar los **Congresos de Ingeniería de la Organización** (**CIOs**) y de establecer canales de comunicación, sirviendo de nexo de comunicación y foro de intercambio y discusión entre sus miembros. Las publicaciones de dichos Congresos son reconocidas por su significativa contribución y sus autores son muy apreciados por la comunidad científica; sin embargo, es necesario generar un mayor impacto de esta importante fuente de conocimiento *(Aparicio Ruiz, 2012)*.

2 OBJETIVOS Y ANÁLISIS

2.1 OBJETIVOS DEL PROYECTO

Tal y como se ha mencionado en la Introducción, éste Proyecto tiene como principal finalidad diseñar e implementar una herramienta que permita el acceso a la documentación que se genera en los **Congresos de Ingeniería de la Organización** (**CIOs**) de una manera cómoda y sencilla; a la vez que segura. Esta documentación está formada por artículos de investigación en formato **PDF**.

Por lo tanto, el objetivo de este Proyecto se puede definir así:

Diseñar e implementar una herramienta que permita automatizar la clasificación y la publicación ordenada de los artículos de investigación generados en los ***Congresos de Ingeniería de la Organización (CIOs)*** *organizados por* ***ADINGOR***.

La Figura 2-1 muestra un diagrama que representa el proceso genérico planteado inicialmente como solución:

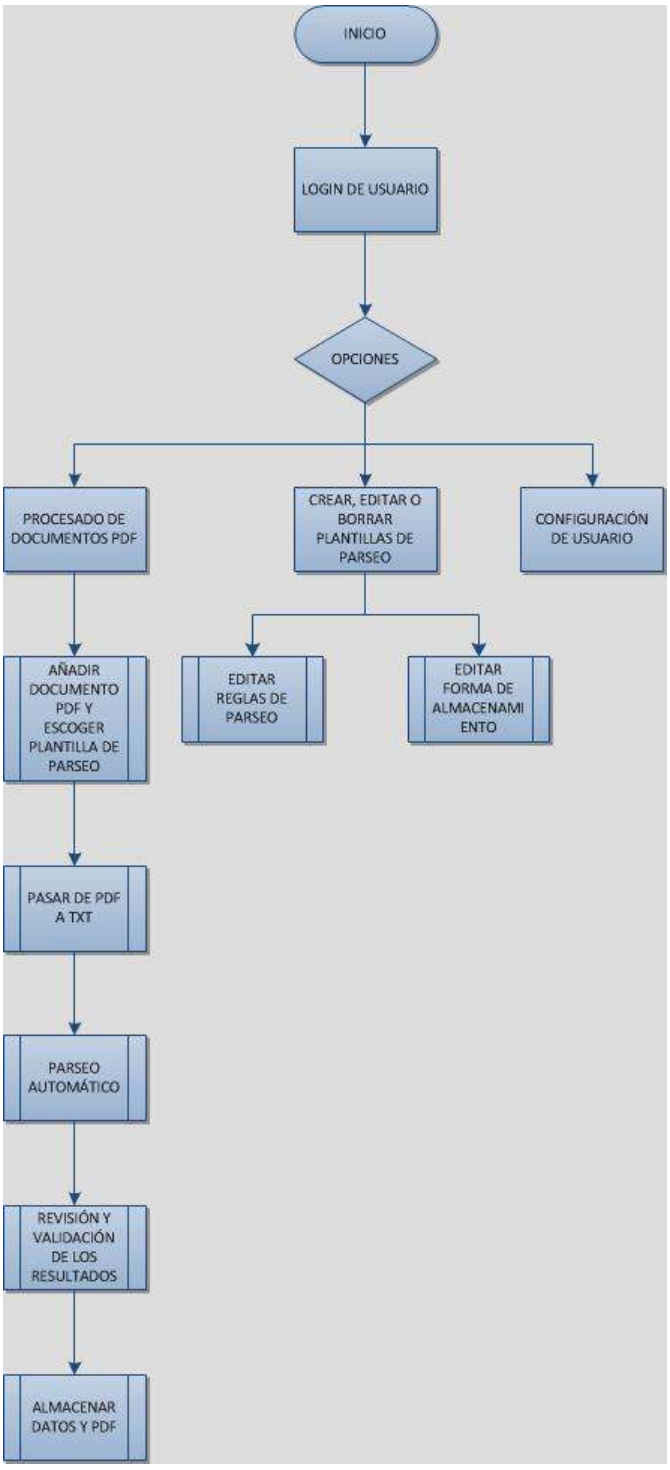


Figura 2-1. Diagrama del proceso genérico planteado inicialmente como solución.

El análisis del diagrama anterior nos ayuda a concretar los objetivos del trabajo a realizar:

La herramienta a desarrollar debe tener un acceso restringido. Únicamente los usuarios autorizados deben tener acceso. Estos usuarios deberán identificarse correctamente usando un nombre de usuario y una contraseña. De esta forma se pretende evitar el acceso de usuarios malintencionados.

Una vez correctamente autenticado, el usuario podrá acceder a las opciones que permitan el procesado de nuevos documentos, las opciones de creación y modificación de reglas de parseo y a un área de gestión

personal. Precisamente en el área de gestión personal será donde los usuarios podrán modificar los datos de su cuenta; para, por ejemplo, cambiar su contraseña.

Las dos opciones restantes (procesado de nuevos documentos y creación y modificación de reglas de parseo) serán realmente las encargadas de hacer que los artículos de investigación puedan ser clasificados y publicados automáticamente.

Si el usuario decide procesar un nuevo documento, es decir añadir un nuevo artículo; el primer paso será seleccionar el correspondiente archivo en formato **PDF** y la regla de parseo adecuada. A continuación, de forma automática y transparente para el usuario, la herramienta extraerá el texto del documento en **PDF**. Sobre este texto actuarán las reglas de parseo para obtener los datos necesarios para clasificar correctamente el artículo. Finalmente, la información obtenida debe presentarse al usuario para que este la verifique y la corrija si fuera necesario.

La tercera y última opción permitirá a los usuarios la creación de nuevas reglas de parseo y la modificación de las existentes. Estas reglas deben definir cómo encontrar de forma automática los datos que identifiquen a un artículo. Por ejemplo, deben definir cómo encontrar el título o los autores del mismo.

Si dicho proceso genérico se implementase; se obtendría, tal y como se desea, una herramienta que permitiría el acceso a la información de una manera cómoda y sencilla; a la vez que segura. Cómoda y sencilla porque todo el proceso de clasificación y publicación quedaría automatizado y segura porque únicamente los usuarios autorizados tendrán acceso a la misma.

2.2 MOTIVACIÓN DEL PROYECTO

El crecimiento y éxito de los últimos **Congresos de Ingeniería de la Organización** (**CIOs**) no ha ido acompañado de la difusión necesaria en Internet *(Aparicio Ruiz, 2012)*. Clasificar toda la documentación generada requiere una gran cantidad de tiempo y de trabajo, especialmente para agregar manualmente a la base de datos cada una de las características que presenta un artículo.

Además, como cada año el Congreso es desarrollado por una Universidad u Organización diferente, la plataforma encargada de la gestión del Congreso está separada de la plataforma que administra la documentación generada.

Este Proyecto surge precisamente con la intención de dar una solución a este problema. Se quiere diseñar e implementar una herramienta que automatice todo el proceso de clasificación de los artículos, limitando al máximo la interacción de la herramienta con los administradores y que dicha herramienta que sea compatible con la documentación generada en todos los Congresos.

2.3 ANÁLISIS

Durante las etapas iniciales del Proyecto se plantearon diferentes aproximaciones que pudieran dar respuesta a los requisitos definidos. Se valoró la posibilidad de desarrollar una aplicación web completamente nueva, partiendo de cero o dar continuidad a una ya existente. Incluso se realizó el análisis de alguna herramienta de gestión documental o **DMS** (**Document Management System**).

Figura 2-2. **LogicalDOC**.

De entre las herramientas de gestión documental que se analizaron cabría destacar **LogicalDOC** [2]. Se trata de una aplicación de software libre desarrollada en lenguaje **Java** y que funciona con **Tomcat**.

LogicalDOC se instaló en nuestro entorno de desarrollo y se hicieron varias pruebas para verificar su capacidad para indexar documentos en formato **PDF**.

Finalmente se decidió trabajar para mejorar una aplicación web ya existente. De esta manera se centraron todos los recursos en la búsqueda soluciones que cumpliesen las exigencias establecidas por los objetivos del Proyecto. Evitando tener que invertir esfuerzo en cuestiones que ya habían sido resueltas previamente; como por ejemplo, la gestión de los usuarios o la publicación de los artículos.

2.3.1 ANÁLISIS DE ANTECEDENTES

La aplicación web original que sirve de base a este Proyecto se había desarrollado para dar respuesta a los mismos objetivos que hemos definido previamente, aunque no llegó a implementar ni herramientas de procesado automático de nuevos documentos, ni herramientas de creación y modificación de reglas de parseo.

Por otra parte, la aplicación web original sí creó un magnifico entorno para la publicación de los artículos. Se habían desarrollado dos secciones completamente diferenciadas: una pública, a la que se accede sin ninguna restricción, y que únicamente permite consultar la información almacenada; y otra con acceso restringido, desde donde se gestiona todo el contenido.

Además, el sistema cumple con el estándar **OAI-PHM** y describe los registros con metadatos. **OAI-PHM** (**Open Archives Initiative - Protocol for Metadata Harvesting**) es un protocolo para la transmisión de contenidos en Internet. De este modo, se facilita la búsqueda y recuperación de sus contenidos. Asimismo, el sistema contiene información para **Google Académico** [3], mediante el uso de metainformación que permite a

los robots de búsqueda catalogar las comunicaciones *(Aparicio Ruiz, 2012)*.

De forma gráfica se puede representar el funcionamiento de la aplicación web original mediante el siguiente diagrama:

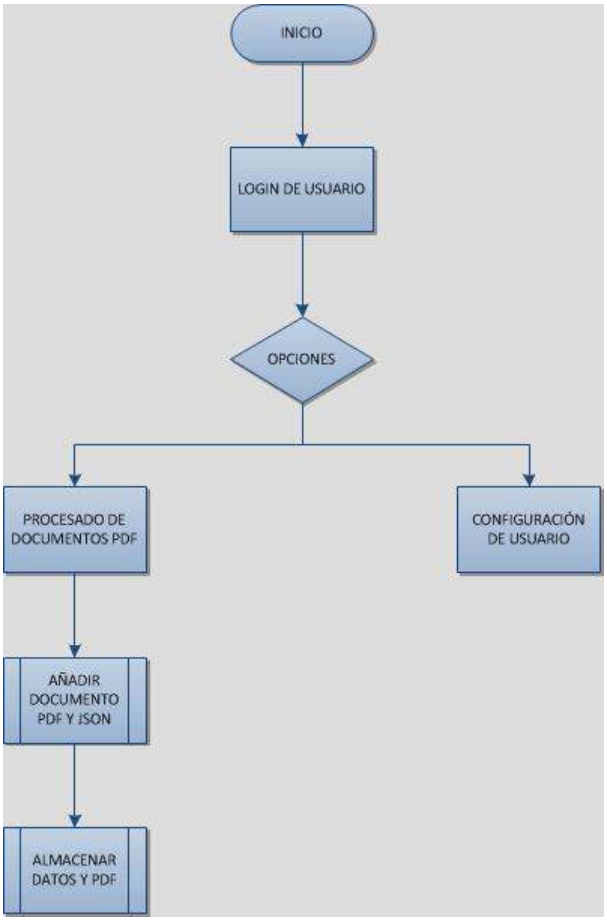


Figura 2-3. Diagrama del proceso genérico de la aplicación web original.

Aunque de forma integrada en la aplicación no se logró implementar el procesado automático de nuevos documentos, si se ideó una solución intermedia para evitar tener que introducir manualmente los datos necesarios para clasificarlos. Para añadir un nuevo documento a la herramienta era necesario subir el **PDF** acompañado de un archivo **JSON** que contenía toda la información que se almacenaba en la base de datos y que servía para clasificarlo.

JSON [4] (**JavaScript Object Notation**) es un formato diseñado para el intercambio de datos. Es simple de leer y escribir para los humanos, a la vez que es simple de interpretar y generar para las máquinas. Es un formato de texto completamente independiente del lenguaje, pero utiliza convenciones que son ampliamente conocidas por los programadores de la familia de lenguajes **C**, incluyendo **C**, **C++**, **C#**, **Java**, **JavaScript**, **Perl**, **Python**, y muchos otros.

A continuación, puede verse un ejemplo de archivo **JSON** al que se le han añadido saltos de línea para facilitar su lectura:

Extracto de un archivo JSON.

```
{“archivo”:”.\/administracion_de_empresas\/11-20.PDF”,
```

```
"titulo":"Modelo de coste unitario para los ensayos de aceros para armaduras pasivas realizados por laboratorios acreditados.\n",
"autores":["Sof\u00eda Estell\u00e9s-Miguel","Jos\u00e9 Miguel Albarrac\u00edn Guillem","Marta Palmer Gato","Teresa Barber\u00e1 Ribera"],
"emails":["soesm@omp.upv.es","jmalbar@omp.upv.es","marpalga@omp.upv.es","mabarri@omp.upv.es"],
"resumen":"El presente art\u00edculo nace en el marco de un convenio entre la Universidad Polit\u00e9cnica de Valencia y el Instituto Valenciano de la Edificaci\u00f3n (en adelante IVE). [...]",
"tags":["cost accounting","procesos","activity-based costing (ABC)"]}
```


Para construir los archivos **JSON** se creó una herramienta externa a la aplicación, un script escrito en lenguaje **PHP**. Básicamente este script se encargaba de extraer el texto del documento en **PDF** y lo guardaba como un archivo en formato **TXT**. A continuación, usaba reglas de parseo para recorrer dicho archivo buscado el título, el resumen, los autores con sus correspondientes direcciones de correo electrónico y las palabras clave del documento.

Cuando las reglas de parseo configuradas en el script no eran capaces de localizar los datos necesarios, el script permitía la interacción con el usuario mediante la línea de comandos para que éste señalara la ubicación del dato correspondiente.

2.3.2 APORTACIÓN REALIZADA

El trabajo realizado en este Proyecto se puede dividir en tres grandes bloques:

1. Adaptación de la aplicación web original para reflejar los cambios de la base de datos.
2. Adaptación de la aplicación web original para implementar el procesado automático de nuevos artículos.
3. Creación de reglas de parseo compatibles con la nueva aplicación.

Esta misma división servirá para estructurar esta Memoria; de forma que cada uno de los tres capítulos siguientes describe de forma exhaustiva las tareas realizadas en cada uno de estos bloques.

Como la base de datos sobre la que se construyó la aplicación web original había sido modificada, la primera tarea realizada consistió en identificar dichas modificaciones. Seguidamente, se hicieron los cambios necesarios para que la información manejada por la aplicación quedara equiparada con la de la base de datos.

Dentro de un segundo bloque de tareas se modificó la aplicación web original para incorporar el procesado automático de nuevos artículos. Como se ha descrito anteriormente, hasta ahora para añadir un nuevo artículo se debía seguir un proceso de dos pasos. El primero se realizaba de forma completamente independiente a la aplicación web y consistía en procesar el documento **PDF** para obtener el correspondiente archivo **JSON**. A continuación, de vuelta a la aplicación web, se daba de alta un nuevo artículo incorporando ambos archivos, tanto el **PDF** como el **JSON**. Gracias a las modificaciones realizadas estos dos pasos se reducen a uno solo.

De esta forma se puede añadir un nuevo artículo directamente usando la aplicación web.

Por último, se ha dedicado mucho esfuerzo en mejorar los resultados obtenidos durante el proceso de parseo. Se han definido nuevas reglas para obtener automáticamente los datos necesarios para clasificar un artículo con el objetivo de minimizar el número de errores.

Igual que antes, se puede representar el funcionamiento de la aplicación resultante de forma gráfica. Se simplifica así su comparación tanto con los objetivos iniciales como con la aplicación original:

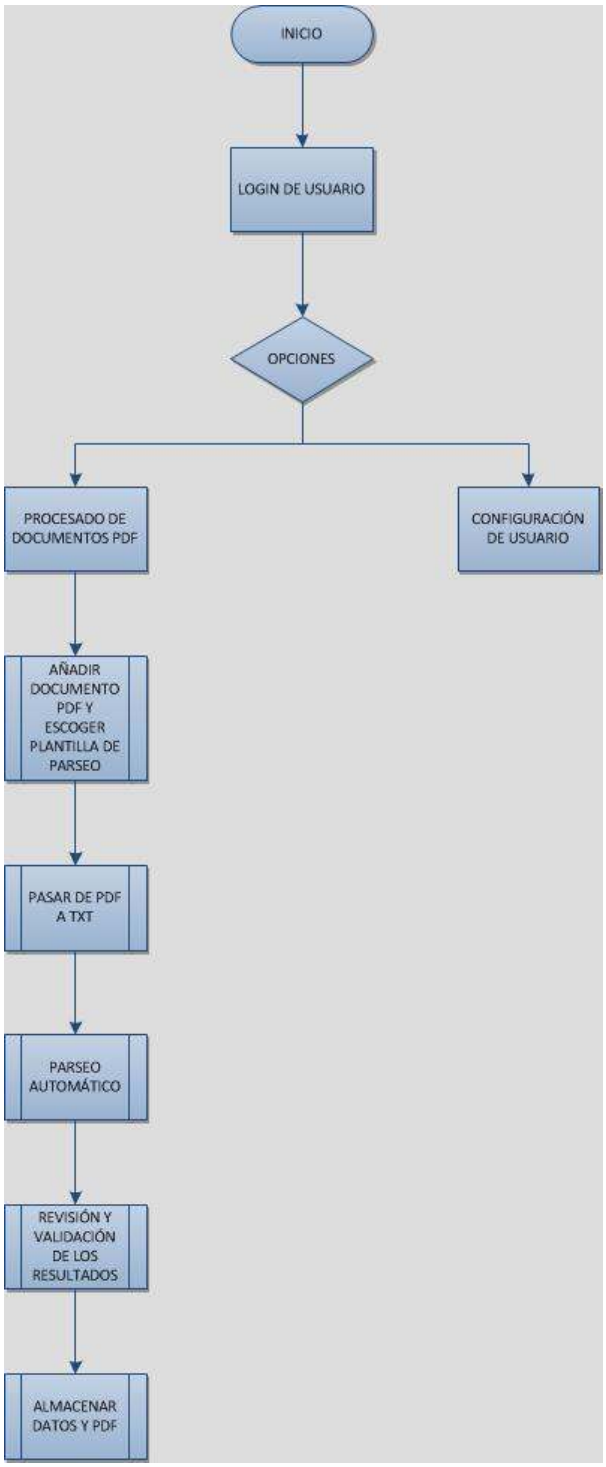


Figura 2-4. Diagrama del proceso genérico de la aplicación desarrollada.

Si comparamos el diagrama de la Figura 2-4 con el diagrama de la Figura 2-1 se observa que todas las

funcionalidades relativas al procesado de nuevos documentos han sido implementadas. No ocurre lo mismo con las herramientas de creación y modificación de reglas de parseo, que siguen sin estar integradas en la aplicación.

2.3.3 ANÁLISIS TEMPORAL Y DE COSTES

En esta sección se presenta una estimación tanto del tiempo como del coste que podría tener la realización de este Proyecto por una empresa dedicada al desarrollo de software. Para hacer esta estimación lo más cercana posible a la realidad se ha contado en la supervisión de **D. Sergio Alvarez Franco**, empresario del sector IT.

Respecto a la estimación temporal, la siguiente tabla muestra el número de jornadas necesarias para las diferentes fases del Proyecto:

Fase	Jornadas
Toma de requisitos	5
Diseño funcional y técnico	10
Desarrollo	20
Pruebas	10
Documentación y cierre	5
TOTAL	**50 Jornadas**

Tabla 2-1. Estimación temporal del Proyecto.

Esta estimación temporal ha tenido en cuenta la asignación de dos recursos: un jefe de proyectos que tendría una dedicación del 30% y una analista programador senior con una dedicación del 100%.

Recurso	Tarifa	Total
Jefe de proyectos *(Dedicación al 30%)*	40€/hora	4.800€
Analista programador *(Dedicación al 100%)*	32,5€/hora	13.000€

COSTE TOTAL SIN IVA	**17.800€**
IVA (21%)	3.738€
COSTE TOTAL CON IVA	**21.538€**

Tabla 2-2. Estimación de costes del Proyecto.

Con una tarifa de 40€ por hora (320€ por jornada) para el jefe de proyectos y de 32,5€ por hora (260€ por jornada) para el programador, el coste de contratar a una empresa especializada para realización de este Proyecto sería de 21.538€.

2.3.4 ANÁLISIS DE TECNOLOGÍAS INVOLUCRADAS

2.3.4.1 DESCRIPCIÓN DE LAS TECNOLOGÍAS INVOLUCRADAS

Las tecnologías usadas para la realización de este Proyecto prácticamente quedaron definidas cuando, en vez de crear una aplicación nueva, se decidió dar continuidad a la aplicación web original. La aplicación original se desarrolló usando **SYMFONY** [5] *(Bartosz Porebski, 2011) (Tim Bowler, 2009) (Zaninotto & Potencier, 2007)* como framework **PHP**, **Doctrine** como herramienta **ORM** [6] y **MySQL** como base de datos. Por este motivo, estas mismas herramientas han sido los pilares fundamentales sobre los que se ha realizado todo el desarrollo.

Como entorno de desarrollo se usó un ordenador con sistema operativo **Windows 7** donde se instaló el conocido software **XAMPP** [7]. **XAMPP** es una distribución de **Apache** completamente gratuita y fácil de instalar que contiene **MySQL**, **PHP** y **Perl**. Se usó la versión 1.7.4 de **XAMPP** que integra la versión 5.3.5 de **PHP**, la 2.2.17 de **Apache** y la 5.5.8 de **MySQL**.

Tras su sencilla instalación, se dispone de un entorno que cumple con todos los requisitos de **SYMFONY**: un servidor web (**Apache**), un motor de bases de datos (**MySQL**) y **PHP** 5.2.4 o superior.

2.3.4.2 SYMFONY

Se ha vuelto a usar **SYMFONY** como framework de **PHP**, pero con una versión más reciente, la 1.4 en vez de la 1.3. Se trata de la única versión clasificada como long-term support (LTS) publicada hasta el lanzamiento de la versión 2.3.

Figura 2-5. Logotipo de **SYMFONY**.

Aunque inicialmente se planteó la posibilidad de migrar la aplicación a la versión 2, esta opción se descartó por el gran número de cambios que dicha actualización implicaría. La versión 2 de **SYMFONY** no es exactamente una continuación de la versión 1, sino que supone una ruptura que implica importantes cambios incluso en la forma de trabajar con el framework.

Algunas diferencias significativas entre la versión 1 y la 2 son los cambios en la estructura predeterminada de archivos y directorios de la aplicación o la sustitución de los plugins por el nuevo concepto de bundles. La principal ventaja de la versión 2 de **SYMFONY** es que fue concebida desde el principio para ser rápida y para favorecer el rendimiento. A modo de comparación, la versión 2 de **SYMFONY** puede llegar a ser hasta 3 veces más rápida y consumir hasta 2 veces menos memoria.

Entre las ventajas de usar **SYMFONY** podemos destacar las siguientes:

- Su gran facilidad de uso, incluso para usuarios no expertos en el trabajo con frameworks. Su curva de aprendizaje es muy suave, permitiendo que incluso un usuario poco familiarizado con el desarrollo web (únicamente con nociones básicas de **HTML**) pueda construir poco a poco una compleja aplicación.
- La extensa documentación que puede encontrarse de forma gratuita en Internet. Merece una mención especial el excelente curso de introducción paso a paso conocido como **Jobeet** [8] y publicado en la página oficinal de **SYMFONY**.
- La abstracción que logra respecto a la estructura de la base de datos gracias a la incorporación dentro del propio framework de herramientas **ORM**: **Doctrine** y **Propel**.

Aunque es cierto que se requiere invertir cierto esfuerzo en conocer el funcionamiento de un framework como **SYMFONY**, las ventajas que aporta al desarrollador de aplicaciones de tamaño medio o grande hacen que su uso esté plenamente justificado.

Para finalizar, su característica más destacable (especialmente en situaciones como la de este Proyecto donde un desarrollador continúa un trabajo previo realizado por otra persona) es la normalización de los desarrollos. La estructura predeterminada de archivos y directorios permiten a cualquier desarrollador con algunos conocimientos del framework asumir fácilmente el mantenimiento de un proyecto.

2.3.4.3 PASO DE PDF A TXT

Por último, debido al relevante papel que tiene en el procesado automático de nuevos artículos, es importante hablar sobre la herramienta usada para extraer en forma de texto plano la información contenida en un archivo en formato **PDF**.

El script usado originalmente para crear los archivos **JSON** utilizaba el componente **PDFtotext** del proyecto de código abierto **XPDF** [9]. Los buenos resultados obtenidos con esta herramienta hicieron que inicialmente se plantease la posibilidad de seguir usándola en el nuevo desarrollo.

Incluso se llegó a desarrollar una herramienta de procesado de nuevos artículos en la que se usaba el componente **PDFtotext**. Para poder llamar a la herramienta dentro del código **PHP** se usó la función **shell_exec**. Esta función nos permite lanzar la ejecución de un comando de forma similar a cómo se haría usando la línea de comando:

```
shell_exec( PDFtotext.exe -f 1 -l 1 archivo.PDF archivo.txt );
```

Sin embargo, esta solución tuvo que ser desestimada porque la función **shell_exec** no puede usarse en muchos servidores en fase de producción puesto que representa un grave riesgo para la seguridad de la aplicación y del servidor.

Hubo que buscar soluciones alternativas que permitieran su integración con el código **PHP**. Se hicieron pruebas con una clase codificada en **PHP** llamada **class.PDF2text** [10]. Su principal ventaja era la posibilidad de integrarla dentro del código de la aplicación, puesto que la propia herramienta era en sí una clase en **PHP**. Para probarla se creó un sencillo script:

```
<?php
include('class.PDF2text.php');

$a = new PDF2Text();
$a->setFilename('FicheroPDF.PDF');
$a->decodePDF();

$f = fopen("FicheroTXT.txt", "w");
fwrite($f, $a->output());
fclose($f);
?>
```

Con esta nueva herramienta, sólo cuando el archivo en formato **PDF** era extraordinariamente sencillo se obtenían resultados aceptables. Se hicieron pruebas con artículos seleccionados aleatoriamente de varios Congresos y se pudo comprobar que los resultados contenían multitud de errores. En algunos casos se lograba extraer el texto del archivo pero se saltaba la primera página, en otros casos el fichero de texto resultante sólo contenía extraños caracteres e incluso con algunos ejemplos directamente aparecía un mensaje de error.

Finalmente se encontró otra libraría en **PHP** que incluía herramientas para extraer el texto de documentos en formato **PDF**, la librería **PDFParser** [11]. A pesar de que hasta fechas muy recientes la página web oficial de dicha libraría advertía que se trataba de un proyecto aún en construcción, las pruebas realizadas sí dieron buenos resultados.

Al igual que ocurría con la clase **class.PDF2text**, la gran ventaja de la librería **PDFParser** es su facilidad para integrarla dentro del propio código **PHP**. Como puede verse en el siguiente script, su uso también resulta bastante sencillo:

```
<?php
// Include Composer autoloader if not already done.
include 'vendor/autoload.php';

// Parse pdf file and build necessary objects.
$parser = new \Smalot\PdfParser\Parser();
$pdf    = $parser->parseFile('document.pdf');

$text = $pdf->getText();
echo $text;
?>
```

Finalmente en nuestro caso se optó por hacer la extracción del texto página a página:

```
<?php
// Include Composer autoloader if not already done.
include 'vendor/autoload.php';

// Parse pdf file and build necessary objects.
$parser = new \Smalot\PdfParser\Parser();
$pdf    = $parser->parseFile('document.pdf');

// Retrieve all pages from the pdf file.
$pages  = $pdf->getPages();

// Loop over each page to extract text.
foreach ($pages as $page)
        {
        echo $page->getText();
        }
?>
```

3 BASE DE DATOS

3.1 DISEÑO

En este capítulo empieza la descripción detallada del trabajo realizado a lo largo de todo el Proyecto. Comenzaremos con unas breves aclaraciones respecto a la nomenclatura de los proyectos desarrollados usando el framework **SYMFONY** y a continuación, se explicarán los cambios realizados en la base de datos.

La base de datos sobre la que se construyó la aplicación web original había sido modificada. Se habían incluido nuevas tablas y nuevos campos donde almacenar información que originalmente no se había tenido en cuenta; como por ejemplo, el idioma de los artículos. Precisamente por este motivo, lo primero que se hizo fue un análisis exhaustivo tanto de la aplicación web como de la base de datos, para identificar cada una de estas modificaciones. Seguidamente, se hicieron los cambios necesarios para que la información manejada por la aplicación quedara equiparada con la de la base de datos.

Para encontrar las diferencias entre el modelo de datos diseñado originalmente y el modificado se usaron muchas de las herramientas del framework **SYMFONY**.

De forma esquemática, estos fueron los pasos seguidos:

- Se hizo una copia de la aplicación web original en un entorno de desarrollo.
- Igualmente, se hizo una copia de la base de datos modificada en un entorno de desarrollo.
- Usando las herramientas de **SYMFONY** se obtuvieron ambos modelos de datos.
- Se compararon ambos modelos de datos para identificar las modificaciones.
- Por último, se hicieron los cambios necesarios en la aplicación.

3.1.1 NOMENCLATURA USADA POR EL FRAMEWORK SYMFONY

En la nomenclatura usada por el framework **SYMFONY** los proyectos están formados por aplicaciones que

comparten un mismo modelo de datos y que a su vez, están compuestas por módulos. Un módulo se define como un conjunto de código que representa una característica de la aplicación o como un conjunto de manipulaciones que el usuario puede hacer sobre un objeto del modelo.

De esta forma se diría que este Proyecto consta de dos aplicaciones; un "frontend" correspondiente a la vista pública, al que también llamaremos "papers" y un "backend" correspondiente a la vista de los administradores. El acceso al "frontend" no requiere usuario ni contraseña, es público, y únicamente permite consultar la información. A lo largo de todo el Proyecto, el "frontend" original sólo fue modificado para que se mostrasen los nuevos datos almacenados de cada artículo. Estas modificaciones se describirán más adelante. Por el contrario, el acceso al "backend" es restringido, requiere usuario y contraseña, y da acceso a todas las herramientas de gestión.

Sin embargo, a lo largo de esta Memoria el término aplicación también se usa para referirnos a la herramienta formada por el conjunto del "frontend" y del "backend". El contexto nos ayudará a diferenciar en cada caso si se hace referencia a la totalidad de la herramienta o únicamente a una de sus dos partes: "frontend" o "backend".

3.1.2 BASE DE DATOS

La aplicación web original se diseñó para trabajar con una base de datos formada por las siguientes tablas:

- **AreaTematica**
- **Articulo**
- **ArticulosCongreso**
- **Autor**
- **AutoresArticulo**
- **Comite**
- **Congreso**
- **Coordinador**
- **Editor**
- **EntidadOrganizadora**
- **OrganizadorComite**
- **PalabraClave**
- **PalabrasClaveArticulo**
- **Patrocinador**
- **Persona**
- **Publicacion**
- **InformacionCongreso**

Una vez que la herramienta original estaba en fase de producción, se vio la necesidad de almacenar más información y se modificó la base de datos. Estos cambios únicamente se realizaron en la base de datos. Nunca se llegó a adaptar la aplicación para publicar esta nueva información.

Como **SYMFONY** es un framework orientado a objetos, manipulamos los objetos en lugar de escribir sentencias **SQL** para recuperar los registros de la base de datos. Por este motivo, la información de la base de datos relacional debe ser mapeada a un modelo de objetos usando una herramienta **ORM**. En nuestro caso, **Doctrine**.

El **ORM** necesita una descripción de las tablas y sus relaciones para crear las clases relacionadas y esta es, precisamente, la función del archivo **schema.yml** que siguiendo las convenciones del framework se puede encontrar en la ruta: ***\config\doctrine\schema.yml***. Los archivos **schema.yml** se pueden crear a mano con un editor de texto o a partir de una base de datos existente usando las herramientas del propio framework.

3.1.3 COPIA DE LA APLICACIÓN WEB ORIGINAL

Para disponer de una réplica de la aplicación web original en un entorno de desarrollo, se copiaron los archivos de la aplicación web en producción.

Si recordamos, una de las grandes ventajas del framework **SYMFONY** es que todos los desarrollos comparten una estructura predeterminada de archivos y directorios. Para comenzar un nuevo desarrollo, **SYMFONY** dispone del siguiente comando:

```
SYMFONY generate:project nombre_del_proyecto
```

Este comando crea la estructura predeterminada de archivos y directorios comunes a todos los proyectos desarrollados con el framework.

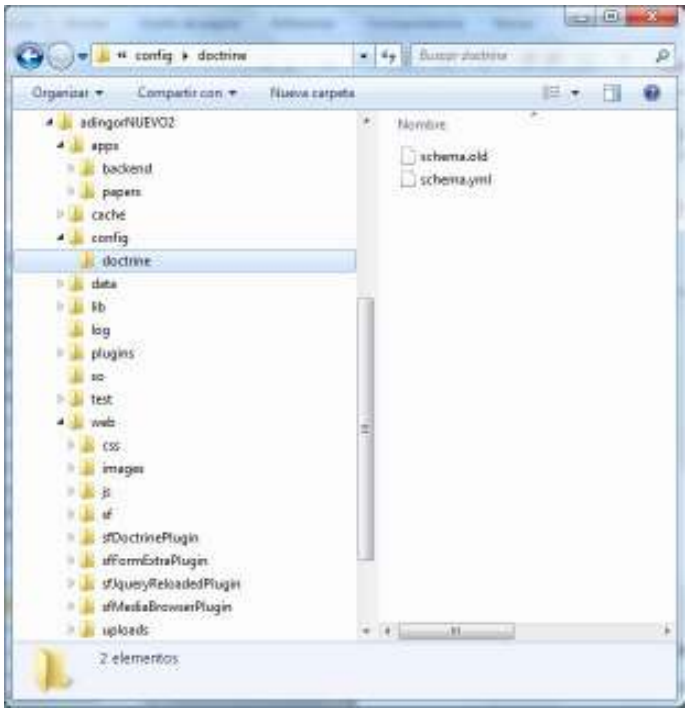


Figura 3-1. Estructura predeterminada de archivos y directorios de los proyectos **SYMFONY**.

Carpeta	Descripción
APPS	En esta carpeta se guardan todas aplicaciones del proyecto.
CACHE	Para los archivos en caché.
CONFIG	Los archivos de configuración del proyecto. Incluyendo el, ya comentado, **schema.yml** en la ruta: ***\config\doctrine\schema.yml***
LIB	Las bibliotecas y clases del proyecto.
LOG	Archivos de registro de eventos.
PLUGINS	Los plugins instalados.
TEST	Los archivos de pruebas.
WEB	El directorio raíz web.

Tabla 3-1. Estructura predeterminada de archivos y directorios de los proyectos **SYMFONY**.

Para crear una réplica, bastará con copiar el contenido todos los directorios de la aplicación en producción en un entorno de desarrollo.

Sin embargo, en nuestro caso, el contenido de las carpetas **LOG** y **CACHE** no es necesario copiarlo, al igual que ocurre con la subcarpeta **UPLOAD** del directorio **WEB** o la subcarpeta **VENDOR** del directorio **LIB**.

En la subcarpeta **UPLOAD** del directorio **WEB** se almacenan los documentos en formato **PDF** de todos los artículos que han sido agregados a la aplicación. Para nuestro desarrollo es suficiente con unos cuantos que usemos de a modo de ejemplo.

Por otra parte, en la subcarpeta **VENDOR** del directorio **LIB** es donde se sitúan los archivos propios del framework **SYMFONY**. Como originalmente se usaba la versión 1.3 pero se actualizará a la 1.4, el contenido original de la subcarpeta debe ser reemplazado por los archivos de la versión 1.4. Estos archivos se pueden descargar gratuitamente de la web de **SYMFONY** en Internet [5].

A continuación, se debe crear la base de datos y configurar la aplicación para que trabaje con ella. A la base de datos la llamaremos “adingor” y se crea con un sencillo comando que proporciona **MySQL**:

```
mysqladmin -uroot -p create adingor
```

Para definir con qué base de datos trabaja la herramienta se tiene que modificar el archivo **databases.yml**

Fichero \config\databases.yml.

```
all:
  doctrine:
    class: sfDoctrineDatabase
    param:
      dsn: 'mysql:host=localhost;dbname=adingor'
      username: root
      password: ****
```

La base de datos creada no tiene definido ningún modelo de datos, para ello se tendrán que usar las valiosas herramientas de **SYMFONY**. Únicamente se necesitan tres sencillas instrucciones:

```
SYMFONY doctrine:build --model
SYMFONY doctrine:build --sql
SYMFONY doctrine:insert-sql
```

El modelo creado en la base de datos tras la ejecución de estas tres instrucciones viene definido por el archivo **schema.yml**. Por este motivo, a pesar de que hemos partido de la descarga de la aplicación web en producción, el modelo de datos de la aplicación creada en el entorno de desarrollo se corresponde con el modelo de datos original. La clave está en recordar que los cambios se habían hecho únicamente en la base de datos en producción y no en la aplicación.

Llegados a este punto, se tiene una copia de la aplicación web original completamente activa en nuestro entorno de trabajo. Para poder visualizarla usando un navegador, hay que configurar el servidor web para publicar la carpeta **WEB**. Sólo es necesario permitir el acceso a carpeta **WEB**, publicar cualquier otra carpeta de la estructura de directorios de la aplicación puede suponer un importante riesgo de seguridad.

A modo de ejemplo, la configuración para publicar la carpeta **WEB** utilizando el servidor **Apache** de nuestro entorno de desarrollo sería la siguiente:

Extracto del fichero httpd.conf.

```
Listen 127.0.0.1:8087

<VirtualHost 127.0.0.1:8087>
  DocumentRoot “c:\dev\sfprojects\adingorNUEVO2\web”
  DirectoryIndex index.php
  <Directory “c:\dev\sfprojects\adingorNUEVO2\web”>
    AllowOverride All
    Allow from All
  </Directory>
[…]
</VirtualHost>
```

Ya se puede acceder a la réplica de la aplicación web original usando un navegador, aunque únicamente se presentará la información que previamente haya sido introducida en la base de datos.

Figura 3-2. Copia de la aplicación web original en un entorno de desarrollo.

3.1.4 COPIA DE LA BASE DE DATOS MODIFICADA

A continuación, se creó una nueva base de datos, a la que se llamó "adingor_modificada". En la sección anterior ya vimos cómo para definir una nueva base de datos se usa el siguiente comando:

```
mysqladmin -uroot -p create adingor_modificada
```

A diferencia de lo que se hizo con la base de datos "adingor", en "adingor_modificada" la estructura de la base de datos no se construyó usando las herramientas de **SYMFONY**, principalmente porque no se disponía del modelo de datos. Se creó usando comandos en lenguaje **SQL**, como por ejemplo:

```
CREATE TABLE IF NOT EXISTS `areatematica` (
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `alias` varchar(45) NOT NULL,
  `nombre` varchar(255) NOT NULL,
  `descripcion` text,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=116 ;
```

De esta forma, la estructura de la base de datos "adingor_modificada" es completamente similar a la de la base de datos de la aplicación en producción.

Además de las instrucciones que vimos anteriormente y que nos ayudaron a construir la base de datos a partir del modelo, el framework también dispone de una herramienta que nos proporciona el modelo a partir de la

base de datos.

Por lo tanto, si volvemos a modificar el archivo **databases.yml**, esta vez para establecer que la aplicación trabaje con la base de datos "adingor_modificada", se puede usar el siguiente comando para crear automáticamente el modelo de datos:

```
SYMFONY doctrine:build-schema
```

De esta forma, siguiendo un proceso completamente inverso del apartado anterior, se creará un nuevo archivo **schema.yml** con la definición del modelo de datos correspondiente a la base de datos de la aplicación en producción.

3.1.5 COMPARACIÓN DE AMBOS MODELOS

Ya disponemos de los archivos **schema.yml** correspondientes tanto a la aplicación diseñada originalmente como a la aplicación con la base de datos modificada. Una sencilla comparación entre ambos archivos nos permite identificar qué cambios habían sido realizados:

- Se había añadido una nueva tabla llamada **ArticuloIdioma** con cuatro campos: ***Id_articulo***, ***Idioma***, ***Titulo*** y ***Resumen***. Esta tabla se diseñó para almacenar información de un mismo artículo en varios idiomas.
- Se habían añadido los campos ***Idioma*** y ***DOI*** a la tabla **Articulo** para almacenar el idioma principal del artículo y su **DOI** [12] o **Digital Object Identifier** (Identificador Digital de Objeto).
- Un nuevo campo llamado ***Idioma*** se agregó en la tabla **PalabrasClaveArticulo**, para identificar el idioma de las palabras clave relacionadas con un artículo.
- Por último, se había añadido ***Institucion*** a la tabla **Autor** para reflejar la conexión de un autor con una Institución u Organización.

Extracto del fichero \config\doctrine\schema.yml correspondiente a la aplicación original.

```
Autor:
  connection: doctrine
  tableName: Autor
  columns:
    id:
      type: integer(11)
      unsigned: true
      primary: true
      autoincrement: true
    nombre:
      type: string(255)
      notnull: true
    email:
      type: string(255)
  relations:
    Articulos:
      class: Articulo
      refClass: AutoresArticulo
      local: Autor_id
      foreign: Articulo_id
```

Extracto del fichero \config\doctrine\schema.yml correspondiente a la base de datos modificada.

```
Autor:
  connection: doctrine
  tableName: Autor
  columns:
    id:
      type: integer(11)
      unsigned: true
      primary: true
      autoincrement: true
    nombre:
      type: string(255)
      notnull: true
    email:
      type: string(255)
    institucion:
      type: string()
  relations:
    Articulos:
      class: Articulo
      refClass: AutoresArticulo
      local: Autor_id
      foreign: Articulo_id
```

3.1.6 OTRAS MODIFICACIONES DE LA BASE DE DATOS

Durante el desarrollo del Proyecto se llegó a la conclusión de que era necesario realizar cambios adicionales en la base de datos. Estos cambios fueron motivados principalmente por la filosofía de trabajo de **SYMFONY**.

Si se revisa el diagrama que representa el proceso genérico planteado inicialmente como solución, se puede ver que en el procesado de documentos **PDF**, tras el parseo automático, aparece una fase de revisión y validación de los resultados.

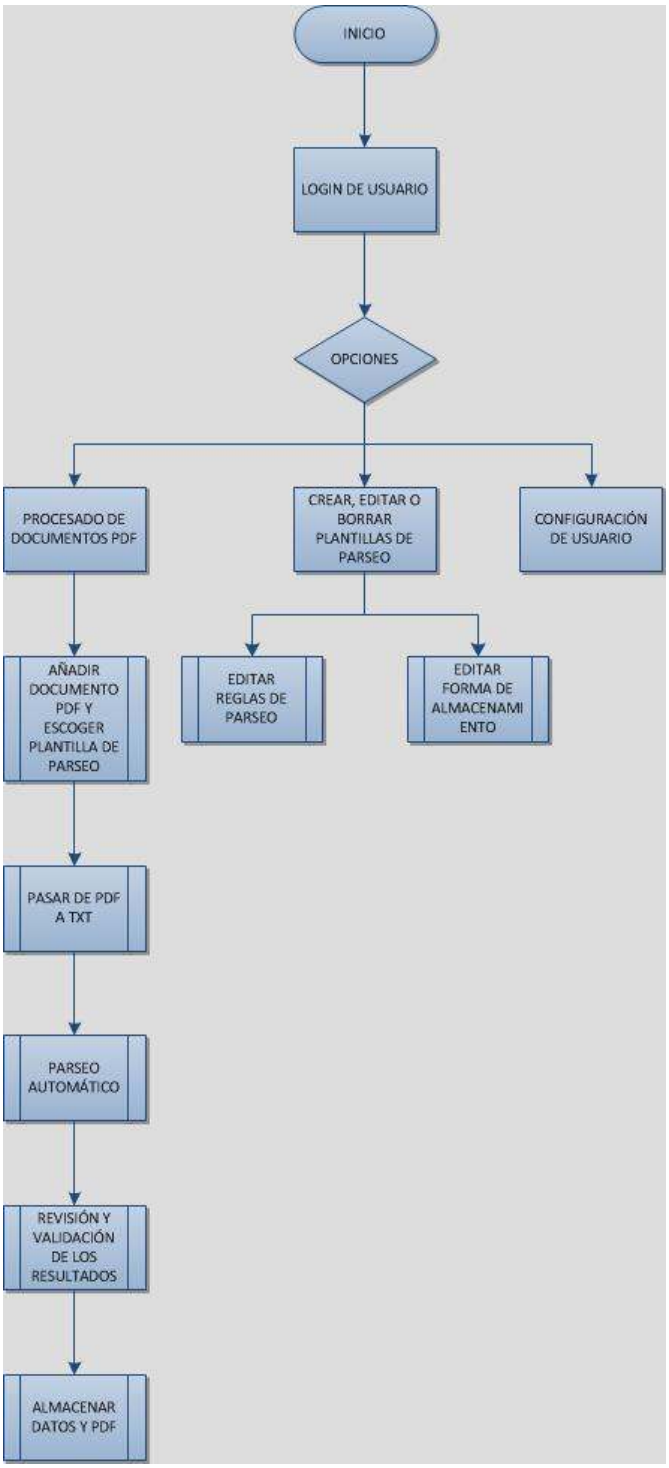


Figura 3-3. Diagrama del proceso genérico planteado inicialmente como solución.

Cuando la herramienta procesa un nuevo documento crea un objeto de tipo **ARTICULO**, obtiene del parseo toda la información necesaria y la presenta para su revisión y validación. El problema es que parte de la información obtenida no se almacena en el propio objeto **ARTICULO** y los datos que se almacenan en otros objetos no pueden ser modificados.

Por ejemplo, si tras el procesado de un nuevo artículo se identifica como autor a "*Francisco Garrido*", en la revisión y validación de los resultados se puede eliminar la relación entre el nuevo artículo y el autor o incluso establecer una nueva relación con otro autor de la base de datos, pero no se pueden hacer modificaciones sobre

los propios datos del autor. Es decir, en la pantalla de revisión no podríamos cambiar el nombre para que en vez de "*Francisco Garrido*" apareciese "*Francisco José Garrido*".

Por este motivo, se creó un nuevo campo en la tabla **Autor**, llamado ***Revisado***. Así, cuando el procesado de un nuevo artículo tiene como consecuencia la creación de un nuevo autor, el campo ***Revisado*** indicará que este autor ha sido creado automáticamente por el sistema y necesita ser validado por los administradores.

***Extracto del fichero \config\doctrine\schema.yml* modificado.**

```
  Autor:
    connection: doctrine
    tableName: Autor
    columns:
      id:
        type: integer(11)
        unsigned: true
        primary: true
        autoincrement: true
      nombre:
        type: string(255)
        notnull: true
      email:
        type: string(255)
      institucion:
        type: string(255)
      revisado: { type: boolean, notnull: true, default: 1 }
    relations:
  [...]
```

El mismo problema que acabamos de ver con los autores a la hora de revisar y validar los resultados ocurre si durante el procesado de un nuevo artículo se identifica el título y el resumen en más de un idioma. Originalmente habría que almacenar esta información en nuevo objeto **ARTICULOIDIOMA**, que no podría ser corregida en la pantalla de revisión.

Por este motivo, se decidió modificar el objeto **ARTICULO**, para que fuera capaz de almacenar el título y el resumen en dos idiomas diferentes. Por lo tanto, se eliminó la tabla **ArticuloIdioma**, y se modificó la tabla **Articulo**. Si ya se habían añadido los campos ***Idioma*** y ***DOI***; ahora sumamos ***Idioma2***, ***Titulo_idioma2*** y ***Resumen_idioma2***.

La última modificación de la tabla **Articulo** fue añadir el campo ***Regla_parseo***. Sin necesidad de profundizar en las características de las reglas de parseo, es evidente que los artículos de un mismo **CIO** presentan diferencias significativas. Por eso, se vio la necesidad de crear diferentes reglas de parseo para un mismo Congreso. El campo ***Regla_parseo*** se diseñó para identificar qué regla de parseo debe usarse para el procesado del artículo.

Extracto del fichero \config\doctrine\schema.yml modificado

```
Articulo:
  connection: doctrine
  tableName: Articulo
  columns:
    id:
      type: integer(11)
      unsigned: true
      primary: true
      autoincrement: true
    titulo:
      type: string(500)
      notnull: true
    titulo_idioma2:
      type: string(500)
    resumen:
      type: string()
    resumen_idioma2:
      type: string()
    capturaPDF:
      type: string(255)
    rutaPDF:
      type: string(255)
    firstPage:
      unsigned: true
      type: integer(5)
    lastPage:
      unsigned: true
      type: integer(5)
    idioma:
      type: string(255)
    idioma2:
      type: string(255)
    doi:
      type: string(255)
    regla_parseo:
      type: string(255)
  relations:
    […]
```

La principal ventaja de realizar estos cambios es la posibilidad de corregir los datos en la fase de revisión y validación de los resultados, posterior al procesado automático del documento. En cambio, perdemos cierta flexibilidad, puesto que ahora únicamente podemos almacenar información de un artículo en dos idiomas; mientras que usando la tabla **ArticuloIdioma**, no teníamos esta limitación.

3.2 IMPLEMENTACIÓN

Cuando se presentó la nomenclatura usada por el framework **SYMFONY**, ya se explicó que este Proyecto constaba de dos aplicaciones: el "frontend" o "papers" y el "backend". En el "frontend" se publica la información y su acceso es libre, mientras que en el "backend" se realizan las tareas de gestión y su acceso está limitado a los administradores.

Una vez establecido el nuevo modelo de datos, se tendrán que actualizar ambas aplicaciones: "frontend" y "backend". Por suerte, al disponer del nuevo modelo de datos definido en un archivo **schema.yml** podremos usar las herramientas del framework para actualizar ambas aplicaciones de una forma muy sencilla. Sólo será necesario modificar el código fuente para personalizar la presentación de la información que realiza el "frontend".

Una vez verificado que los archivos **schema.yml** y **databases.yml** sean los correctos, se ejecuta el siguiente comando:

```
SYMFONY doctrine:build --all --no-confirmation
```

Este comando vuelve a definir todos los módulos que forman tanto el "frontend", como el "backend"; también crea la estructura de tablas y campos de la base de datos conforme a los definido por el archivo **schema.yml**. Además, tiene la gran de ventaja de que al ejecutarlo no se pierde la personalización de las plantillas usadas para la presentación de la información que se muestra a los usuarios.

De esta forma han quedado actualizadas ambas aplicaciones: "frontend" y "backend". Quedan pendientes las modificaciones del código fuente necesarias para personalizar la presentación de la información que realiza el "frontend". Los cambios que necesitamos hacer son:

- Incluir el **DOI** en la referencia del artículo, si éste está definido.
- Publicar el título y el resumen en el segundo idioma, si estos están definidos.
- Ordenar las palabras clave por idioma.

El framework **SYMFONY** sigue el patrón de diseño **MVC** o **Model View Controller** (Modelo, Vista y Controlador) para organizar el código. De esta forma el código queda separado en tres capas:

- La capa Modelo define la lógica, la base de datos pertenece a esta capa.
- Vista es la capa con la que interactúa el usuario. Está principalmente definida por plantillas **PHP**, almacenadas en varias subcarpetas llamadas **TEMPLATES**.
- Por último, el Controlador es el código encargado de llamar al Modelo para obtener los datos que se

pasan a la Vista para su presentación al usuario.

Todo el código que se necesita modificar pertenece al módulo **ARTICULO** de la aplicación "frontend" o "papers"; por lo tanto los archivos se encontrarán en la siguiente ruta: ***\apps\papers\modules\articulo***. En dicha ruta hay dos subcarpetas: **TEMPLATES** y **ACTIONS**. La subcarpeta **TEMPLATES** está relacionada con la capa Vista, mientras que la subcarpeta **ACTIONS** está relacionada con la capa Modelo.

Con el siguiente extracto de código se incluye el campo **DOI** en la referencia del artículo. El bucle *if* verifica que aparezca únicamente si el campo **DOI** está definido.

Extracto del fichero \apps\papers\modules\articulo\templates\detalleSuccess.php.

```
<?php
if( $articulo->getDOI() != null)
        {
        echo ("DOI:".$articulo->getDOI().".");
        }
?>
```

De forma completamente similar se haría con el título del artículo en el segundo idioma.

En el caso del resumen, se hace una comprobación previa, de forma que si el segundo idioma es el inglés, el texto del resumen irá precedido de la palabra "Abstract".

Extracto del fichero \apps\papers\modules\articulo\templates\detalleSuccess.php.

```
<?php if( $articulo->getResumenIdioma2() != null) :?>

        <?php if( $articulo->getIdioma2() == 'Inglés') :?>

        <div class="articulo-detalle-resumen">
        <h3><?php echo image_tag('icons/page_white_text','class=title_img'); ?>Abstract</h3>
        <p><?php echo str_replace("\\n","<br />", $articulo->getResumenIdioma2()); ?></p>
        </div>

        <?php else :?>

        <div class="articulo-detalle-resumen">
        <h3><?php echo image_tag('icons/page_white_text','class=title_img'); ?>Resumen</h3>
        <p><?php echo str_replace("\\n","<br />", $articulo->getResumenIdioma2()); ?></p>
        </div>

        <?php endif; ?>

<?php endif; ?>
```

Por último, para ordenar las palabras clave por idioma no basta con hacer modificaciones en la capa Vista. También será necesario modificar el código del Modelo. En el archivo **actions.class.php** de la subcarpeta **ACTIONS** se usarán dos variables diferentes (*tags* y *tags2*) para almacenar las palabras clave de cada idioma.

Extracto del fichero \apps\papers\modules\articulo\actions\actions.class.php.

```
$this->tags = array();
$this->tags2 = array();

[…]

foreach($this->articulo->getPalabrasClaveArticulo() as $palabra_clave)
{
        if ($palabra_clave->getIdioma() == $this->articulo->getIdioma())
                {
                $this->tags[] = $palabra_clave->getPalabraClave();
                }
        elseif ($palabra_clave->getIdioma() == $this->articulo->getIdioma2())
                {
                $this->tags2[] = $palabra_clave->getPalabraClave();
                }
}
```

Estas variables son automáticamente accesibles desde el archivo **detalleSuccess.php** de la subcarpeta **TEMPLATES.**

Al igual que ocurría en el caso del resumen, se hace una comprobación previa, de forma que si el segundo idioma es el inglés, la lista de palabras irá precedida de la palabra “Keywords”.

Extracto del fichero \apps\papers\modules\articulo\templates\detalleSuccess.php.

```
<?php if(count($tags) > 0): ?>

        <?php if( $articulo->getIdioma() == 'Inglés') :?>

                <div class=“articulo-detalle-tags”>
                <h3><?php echo image_tag('icons/note','class=title_img'); ?>Keywords</h3>
                <ul>
                <?php foreach($tags as $tag): ?>
                        <li><?php echo $tag->getPalabra(); ?></li>
                <?php endforeach; ?>
                </ul>
                </div>

        <?php else :?>

                <div class=“articulo-detalle-tags”>
                <h3><?php echo image_tag('icons/note','class=title_img'); ?>Palabras Clave</h3>
                <ul>
                <?php foreach($tags as $tag): ?>
                        <li><?php echo $tag->getPalabra(); ?></li>
                <?php endforeach; ?>
                </ul>
```

```
                </div>

        <?php endif; ?>
<?php endif; ?>

<?php if(count($tags2) > 0): ?>

        <?php if( $articulo->getIdioma2() == 'Inglés') :?>

                <div class="articulo-detalle-tags">
                <h3><?php echo image_tag('icons/note','class=title_img'); ?>Keywords</h3>
                <ul>
                <?php foreach($tags2 as $tag2): ?>
                        <li><?php echo $tag2->getPalabra(); ?></li>
                <?php endforeach; ?>
                </ul>
                </div>

        <?php else :?>

                <div class="articulo-detalle-tags">
                <h3><?php echo image_tag('icons/note','class=title_img'); ?>Palabras Clave</h3>
                <ul>
                <?php foreach($tags2 as $tag2): ?>
                        <li><?php echo $tag2->getPalabra(); ?></li>
                <?php endforeach; ?>
                </ul>
                </div>

        <?php endif; ?>
<?php endif; ?>
```

Para terminar este capítulo, veremos un ejemplo donde se ponen de manifiesto todos los cambios realizados. La Figura 3-4 muestra la primera página de un artículo. Se puede comprobar como el idioma principal es el castellano y el secundario el inglés. Aparecen en ambos idiomas tanto el título, como el resumen y las palabras clave.

Figura 3-4. Ejemplo de artículo en formato **PDF**.

La Figura 3-5 muestra cómo quedaría la presentación de dicho artículo en el "frontend". Efectivamente se observa el título en castellano, seguido del título en inglés. En la siguiente línea aparece la referencia del artículo donde se puede observar que aparece el **DOI**. Igualmente se puede ver cómo se presenta el resumen y el "abstract", las palabras clave y las "keyworks".

Figura 3-5. Ejemplo de la presentación de un artículo en el "frontend".

4 PROCESADO AUTOMÁTICO DE NUEVOS ARTÍCULOS

4.1 DISEÑO

En el capítulo anterior se hizo una copia de la aplicación web original en un entorno de desarrollo, se actualizó el framework **SYMFONY** de la versión 1.3 a la 1.4, se modificó el modelo de datos y se hicieron los cambios necesarios para que la información manejada por la aplicación quedara equiparada con la de la base de datos. En este nuevo capítulo continuará la descripción detallada del trabajo realizado para implementar el procesado automático de nuevos artículos.

En los objetivos iniciales del Proyecto se estableció que para agregar un nuevo artículo a la herramienta, el primer paso sería añadir el documento en **PDF** y seleccionar la regla de parseo adecuada. A continuación, y ya de forma automática y transparente para el usuario, la aplicación extraería el texto del documento. Sobre dicho texto actuarían las reglas de parseo para obtener los datos necesarios para poder clasificar correctamente el artículo. Finalmente, la información obtenida debía presentarse al usuario para que este la revisara y corrigiese si fuese necesario.

En este capítulo se definirá el formulario apropiado para añadir el documento en **PDF** y seleccionar la regla de parseo adecuada. También crearemos el formulario que nos permita verificar que la información obtenida en el parseo es correcta. Además de los formularios, se establecerá todo el procesamiento tanto previo como posterior al parseo automático de los nuevos artículos.

Ya sabemos que es en el "frontend" donde se publica la información y su acceso es libre, mientras que en el "backend" se realizan las tareas de gestión y su acceso está limitado a los administradores. Añadir un nuevo artículo, es precisamente una de dichas tareas de gestión. Por lo tanto, en este capítulo todo el trabajo se realizará en la aplicación "backend".

En una primera aproximación al diseño de la solución, se pensó crear un nuevo módulo **ARTICULO**. El módulo **ARTICULO** original no había sido creado usando las herramientas estándar del framework para la creación automática de módulos, sino que había sido creado cómo módulo de administración y se creía que esto podía suponer algún tipo de limitación. Finalmente, tras varias pruebas, se comprobó que no era necesario porque independientemente de cómo se creara el módulo, todas las opciones que se necesitaban estaban disponibles.

Los principales archivos sobre los que se trabajará pertenecen a la carpeta **LIB**, donde se definen las bibliotecas y clases.

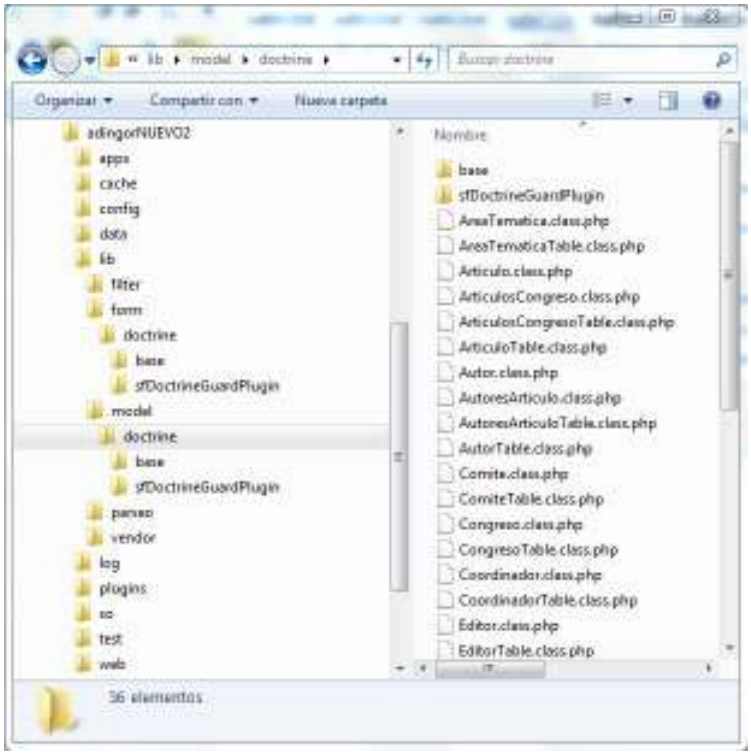


Figura 4-1. Carpeta **LIB** de la estructura predeterminada de archivos.

En la función **configure**, de la clase **ArticuloForm**, del archivo ***\lib\form\doctrine\ArticuloForm.class.php*** se definen los formularios para añadir un nuevo artículo y para modificarlo o corregirlo.

Extracto del fichero \lib\form\doctrine\ArticuloForm.class.php.

```
class ArticuloForm extends BaseArticuloForm
{
        public function configure()
        {
        if ($this->getObject()->isNew())
                // Definimos el formulario para crear un nuevo artículo
                {
                }
        else
                // Definimos el formulario para modificar un artículo
                {
                }
        }
```

```
}
```

Por otra parte, en la función **save**, de la clase **Articulo**, del archivo ***\lib\model\doctrine\Articulo.class.php*** se definen todas las tareas que se deben realizar cuando se añade un nuevo artículo.

Extracto del fichero \lib\model\doctrine\Articulo.class.php.

```
class Articulo extends BaseArticulo
{
	public function save(Doctrine_Connection $conn = NULL)
	{
		if ($this->isNew())
		{
		}
		return parent::save($conn);
	}
}
```

4.2 IMPLEMENTACIÓN

Los detalles de la implementación de la solución adoptada para procesar nuevos artículos de forma automática se van a presentar divididos en tres bloques. El primero se centrará en definir los nuevos formularios; el segundo propiamente en el procesado del artículo y el tercero en otros cambios realizados.

4.2.1 MODIFICACIÓN DE LOS FORMULARIOS DE CREACIÓN DE UN NUEVO ARTÍCULO

Si recordamos, la definición de los formularios para la gestión de los artículos se realiza en la función **configure**, de la clase **ArticuloForm**, del archivo ***\lib\form\doctrine\ArticuloForm.class.php***. Esta función establece el diseño del formulario usado para añadir nuevos artículos en la aplicación y también, el diseño del formulario usado para modificar o corregir los datos de un artículo previamente añadido. Por este motivo dentro de la función, un bucle *if*, con la condición *$this->getObject()->isNew()*, nos ayudara a diferenciar un caso del otro.

Extracto del fichero \lib\form\doctrine\ArticuloForm.class.php.

```
class ArticuloForm extends BaseArticuloForm
{
	public function configure()
```

```
        {
        if ($this->getObject()->isNew())
                // Definimos el formulario para crear un nuevo artículo
                {
                }
        else
                // Definimos el formulario para modificar un artículo
                {
                }
        }
}
```

La lista de los parámetros con los que se puede trabajar será común a ambos formularios. Al nombrarlos individualmente fijamos el orden en que serán presentados al usuario.

Extracto del fichero \lib\form\doctrine\ArticuloForm.class.php.

```
public function configure()
{
$this->useFields(array(
        'rutaPDF',
        'regla_parseo',
        'idioma',
        'titulo',
        'resumen',
        'idioma2',
        'titulo_idioma2',
        'resumen_idioma2',
        'autores_list',
        'palabras_clave_list',
        'congresos_list',
        'areas_tematicas_list',
        'firstPage',
        'lastPage',
        'doi'
        ))
```

También será común a ambos formularios la definición de aquellos parámetros que se presentan al usuario de una forma especial. Por ejemplo, el Congreso y el Área Temática se deben seleccionar de una lista desplegable.

Extracto del fichero \lib\form\doctrine\ArticuloForm.class.php.

```
public function configure()
{
        […]
        $this->widgetSchema['congresos_list']->setOption('renderer_class', 'sfWidgetFormSelect');
        $this->widgetSchema['areas_tematicas_list']->setOption('renderer_class', 'sfWidgetFormSelect');

        $this->validatorSchema['congresos_list'] = new sfValidatorAnd(array(
        $this->validatorSchema['congresos_list'],
        new sfValidatorDoctrineChoice(array('multiple' => true, 'model' => 'Congreso', 'required' => true)),
```

```
));

$this->validatorSchema['areas_tematicas_list'] = new sfValidatorAnd(array(
$this->validatorSchema['areas_tematicas_list'],
new sfValidatorDoctrineChoice(array('multiple' => true, 'model' => 'AreaTematica', 'required' => true)),
));
```

Para añadir un nuevo artículo, el formulario debe pedirnos 5 datos:

- El documento con el artículo en formato **PDF**.
- La regla de parseo que debe usar el sistema.
- El Congreso.
- El Área temática.
- Y, si existe, el **DOI**.

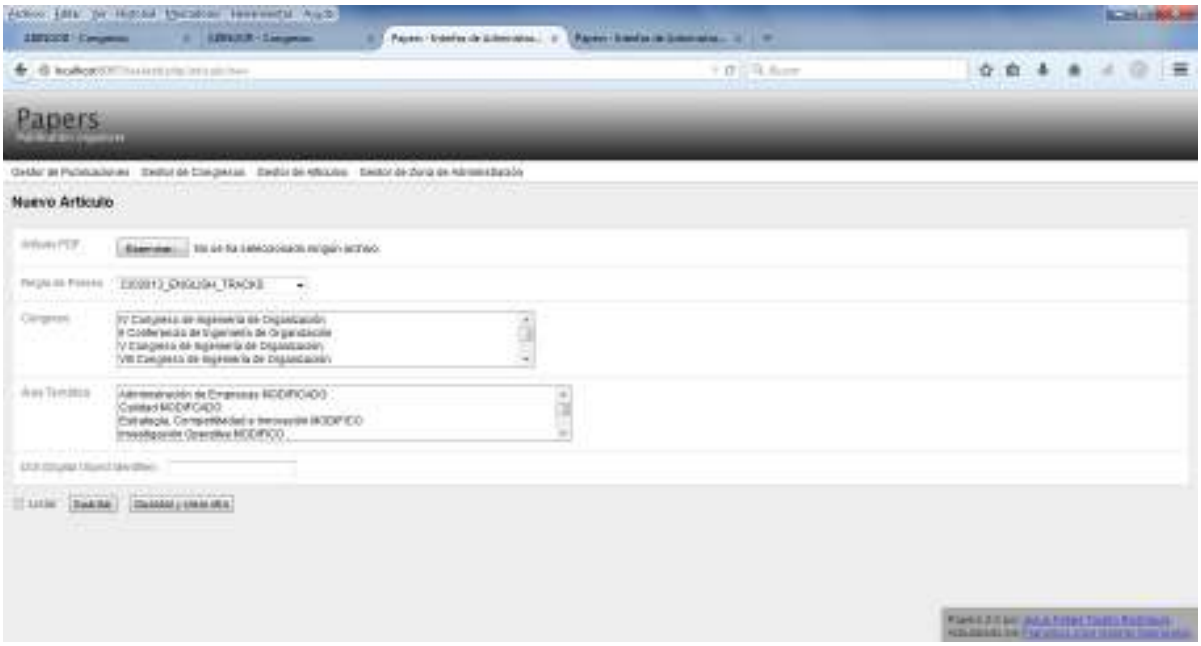


Figura 4-2. Formulario para añadir un nuevo artículo.

Por lo tanto, una vez verificado que el formulario que se debe presentar al usuario es el correspondiente a la opción de añadir nuevos artículos, tenemos que ocultar el resto de los parámetros. El contenido de estos parámetros será proporcionado por el parseo.

Extracto del fichero \lib\form\doctrine\ArticuloForm.class.php.

```
if ($this->getObject()->isNew())
        // Definimos el formulario para dar de alta un nuevo artículo
        {
        unset($this['idioma']);
```

```
unset($this['titulo']);
unset($this['resumen']);
unset($this['idioma2']);
unset($this['titulo_idioma2']);
unset($this['resumen_idioma2']);
unset($this['autores_list']);
unset($this['palabras_clave_list']);
unset($this['firstPage']);
unset($this['lastPage']);
```

Además, para seleccionar el documento con el artículo en formato **PDF**, necesitamos usar una herramienta que nos permita seleccionar el artículo desde su ubicación en nuestro ordenador, copiarlo y almacenarlo en el servidor de la aplicación. Precisamente de esta tarea se encarga el siguiente extracto de código:

Extracto del fichero \lib\form\doctrine\ArticuloForm.class.php.

```
if ($this->getObject()->isNew())
        // Definimos el formulario para dar de alta un nuevo artículo
        {
        […]
        $this->widgetSchema['rutaPDF'] = new sfWidgetFormInputFile(array());

        $this->validatorSchema['rutaPDF'] = new sfValidatorFile(array(
        'required'   => true,
        'path'       => sfConfig::get('sf_upload_dir').'/cio',
        ));
```

En el caso de la elección de la regla de parseo, volvemos a tener una lista desplegable.

Extracto del fichero \lib\form\doctrine\ArticuloForm.class.php.

```
if ($this->getObject()->isNew())
        // Definimos el formulario para dar de alta un nuevo artículo
        {
        […]
        $this->widgetSchema['regla_parseo'] = new sfWidgetFormChoice(array(
        'choices'  => Doctrine_Core::getTable('Articulo')->getReglas(),
        'expanded' => false,
        ));
```

En la Figura 4-3 se puede ver cómo la elección de la regla de parseo se limita a un desplegable:

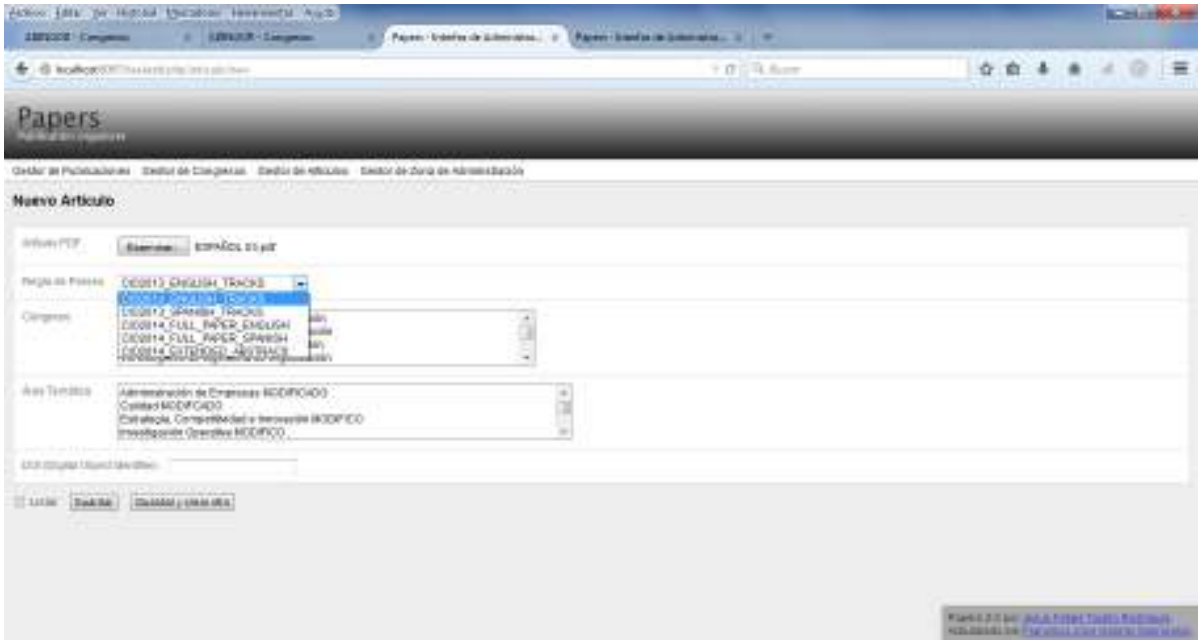


Figura 4-3. Detalle del formulario para añadir un nuevo artículo.

En el caso contrario, cuando en vez de añadir un nuevo artículo lo que se pretende es modificar o corregir los datos de un artículo previamente añadido, los parámetros que se tienen que ocultar son el documento con el artículo en formato **PDF** y la regla de parseo.

Extracto del fichero \lib\form\doctrine\ArticuloForm.class.php.

```
else
        // Definimos el formulario para modificar un artículo
        {
        unset($this['rutaPDF']);
        unset($this['regla_parseo']);
```

Los autores y las palabras clave se presentan con lo que se conoce como una lista doble. Puede verse un ejemplo en la Figura 4-4.

Extracto del fichero \lib\form\doctrine\ArticuloForm.class.php.

```
else
        // Definimos el formulario para modificar un artículo
        {
        [...]
        $this->widgetSchema['autores_list']->setOption('renderer_class', 'sfWidgetFormSelectDoubleList');

        $this->widgetSchema['palabras_clave_list']->setOption('renderer_class', 'sfWidgetFormSelectDoubleList');
```

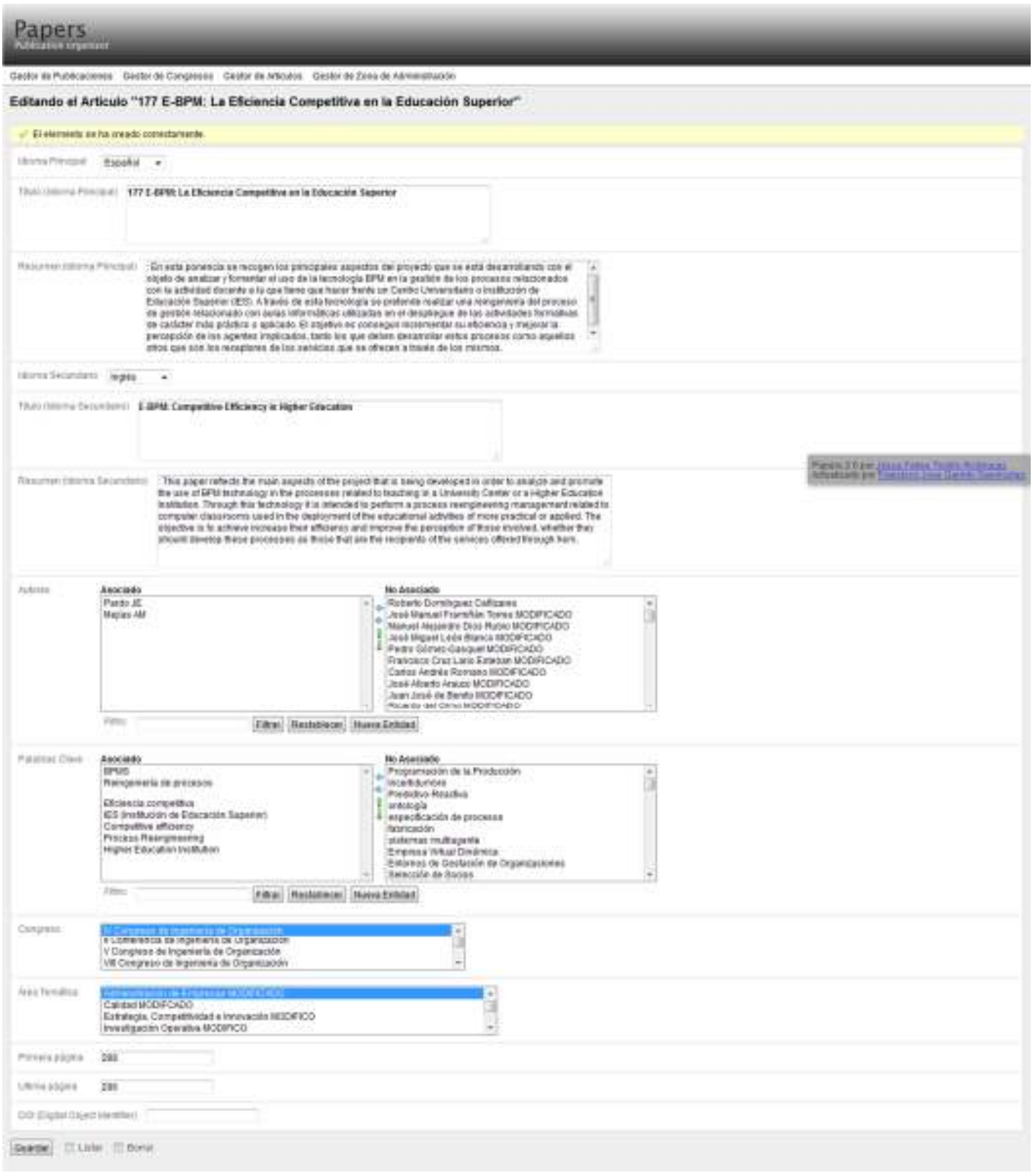


Figura 4-4. Ejemplo de la presentación de un artículo en el "backend".

4.2.2 MODIFICACIÓN DEL PROCESO DE CREACIÓN DE UN NUEVO ARTÍCULO

En este caso será la función **save**, de la clase **Articulo**, del archivo ***\lib\model\doctrine\Articulo.class.php*** la

responsable de definir todas las tareas que se realizan durante el proceso de creación de un nuevo artículo.

Extracto del fichero \lib\model\doctrine\Articulo.class.

```
<?php

require_once(dirname(__FILE__).'/../../parseo/PdfParser/Parser.php');
require_once(dirname(__FILE__).'/../../parseo/ReglasDeParseo/ReglasDeParseo.php');

class Articulo extends BaseArticulo
{
public function save(Doctrine_Connection $conn = NULL)
        {
        return parent::save($conn);
        }
}
```

Necesitamos hacer uso de las herramientas que proporcionan tanto la clase **Parser**, para extraer el texto del documento **PDF**, como la clase **ReglasDeParseo**, precisamente porque esta es la clase que hemos definido para extraer los datos que necesitamos para clasificar los artículos. Estas dos clases tienen que ser cargadas previamente, para ello usamos el comando **require_once**.

La función **save**, de la clase **Articulo** es bastante compleja, precisamente por la gran cantidad de tareas que define. Por este motivo, se han creado secciones separadas por comentarios dentro del código que nos ayudan a organizarlo.

De forma esquemática estás son las tareas de la función **save**:

- Comienza con una serie de tareas previas, necesarias para clasificar los artículos correctamente.
- Extrae el texto del documento **PDF** original.
- Prepara la llamada a la función encargada del parseo.
- Ejecuta la llamada a la función encargada del parseo.
- Por último, procesa los datos obtenidos en el parseo.

Además, como esta función también se ejecuta cuando se modifica un artículo, tenemos que asegurar que las tareas definidas sólo afecten a la creación de nuevos artículos, no a su actualización. Para ello, usamos un bucle *if* con la confición *if ($this->isNew())*.

Extracto del fichero \lib\model\doctrine\Articulo.class.

```
public function save(Doctrine_Connection $conn = NULL)
        {
        if ($this->isNew())
                {
```

```
                /*********************************/
                /* PASAMOS A TXT ****************/
                /*********************************/
                […]
                /*********************************/
                /* PASOS PREVIOS PARSEO *********/
                /*********************************/
                […]
                /*********************************/
                /* PARSEO ***********************/
                /*********************************/
                […]
                /*********************************/
                /* RESULTADOS PARSEO ************/
                /*********************************/
                […]
                /*********************************/
                /* AUTORES & EMAILS *************/
                /*********************************/
                […]
                /*********************************/
                /* PALABRAS CLAVE IDIOMA 1 ******/
                /*********************************/
                […]
                /*********************************/
                /* PALABRAS CLAVE IDIOMA 2 ******/
                /*********************************/
                […]

                }
        return parent::save($conn);
        }
```

La primera tarea de la función consiste en identificar el Congreso y el Área Temática del artículo. Estos datos son necesarios para almacenar los documentos **PDF** originales en la ruta apropiada.

Extracto del fichero \lib\model\doctrine\Articulo.class.

```
    $congreso_id = $this->ArticulosCongreso[0]->Congreso_id;
    $area_id = $this->ArticulosCongreso[0]->AreaTematica_id;

    $query_congreso = Doctrine::getTable('Congreso') ->createQuery('congresoalias') ->where('congresoalias.id = ?', $congreso_id);
    $congreso = $query_congreso->fetchOne();

    $query_area = Doctrine::getTable('AreaTematica') ->createQuery('areaalias') ->where('areaalias.id = ?', $area_id);
    $area = $query_area->fetchOne();

    $fichero_origen = dirname(__FILE__).'/../../../web/uploads/cio//'.$this->getRutaPDF();
    $fichero_destino = dirname(__FILE__).'/../../../web/uploads/cio//'.$congreso->getAlias().'\\'.$area->getAlias().'\\'.$this-
>getRutaPDF();
```

A continuación, realizamos la extracción del texto usando las herramientas de la librería **PDFParser** [11]. La extracción del texto se hace página a página usando el bucle *foreach*.

Extracto del fichero \lib\model\doctrine\Articulo.class.

```
/*********************************/
/* PASAMOS A TXT ****************/
/*********************************/
// Parse pdf file and build necessary objects.
$parser = new Parser();
$filename= dirname(__FILE__).'/../../../web/uploads/cio//'.$this->getRutaPDF();
$document = $parser->parseFile($filename);

// Retrieve all details from the pdf file.
$details = $document->getDetails();

$fd = fopen (dirname(__FILE__).'/../../../web/uploads//'.$this->getRutaPDF().'.txt', 'w');

// Retrieve all pages from the pdf file.
$pages  = $document->getPages();
$paginas_totales = 0;

// Loop over each page to extract text.
foreach ($pages as $page)
        {
        set_time_limit(0);
        $texto = ($page->getText());
        fwrite ($fd, $texto);
        $paginas_totales ++;
        }

fclose($fd);
```

En la sección previa al parseo se crean y se inicializan las variables necesarias.

Extracto del fichero \lib\model\doctrine\Articulo.class.

```
/*********************************/
/* PASOS PREVIOS PARSEO *********/
/*********************************/
$datos_parseados = new ReglasDeParseo();

$titulo_parseado                = '';
$resumen_parseado               = '';
$titulo_idioma2_parseado        = '';
$resumen_idioma2_parseado       = '';
$autores_parseados              = array();
$emails_parseados               = array();
$tags_parseados                 = array();
$tags__idioma2_parseados        = array();
```

```
$autores_parseados = array();
$emails_parseados = array();
$first_page = 0;
$idioma = 'N/A';
$idioma2 = 'N/A';
```

Se realiza la llamada a la función **buscaDatos**, de la clase **ReglasDeParseo**, implementada en el archivo ***\lib\parseo\ReglasDeParseo\ReglasDeParseo.php***. En el siguiente capítulo se describe minuciosamente el comportamiento de dicha función.

Extracto del fichero \lib\model\doctrine\Articulo.class.

```
/*********************************/
/* PARSEO ***********************/
/*********************************/
$datos_parseados->buscaDatos(
        $this->getRutaPDF(),
        $this->getReglaParseo(),
        $titulo_parseado,
        $resumen_parseado,
        $titulo_idioma2_parseado,
        $resumen_idioma2_parseado,
        $autores_parseados,
        $emails_parseados,
        $tags_parseados,
        $tags_idioma2_parseados,
        $first_page,
        $idioma,
        $idioma2
        );
```

Una vez devuelto el control a la función **save**, las variables que hemos pasado a la función de parseo por referencia tendrán almacenados todos los datos que necesitamos para clasificar el nuevo artículo.

Antes de centrarnos en los resultados obtenidos en el parseo, movemos el documento **PDF** original y el documento con el texto extraído del mismo a su ruta definitiva. Dicha ruta viene determinada por el Congreso y el Área Temática.

Extracto del fichero \lib\model\doctrine\Articulo.class.

```
// Movemos el fichero PDF a la carpeta correspondiente al Congreso y al Área Temática
mkdir (dirname($fichero_destino), 0777, true);
rename( $fichero_origen, $fichero_destino);

// Movemos el fichero de texto a la misma carpeta que el PDF
$texto_origen = dirname(__FILE__).'/../../../web/uploads//'.$this->getRutaPDF().'.txt';
$texto_destino = dirname(__FILE__).'/../../../web/uploads/cio//'.$congreso->getAlias().'\\'.$area->getAlias().'\\'.$this->getRutaPDF().'.txt';
rename( $texto_origen, $texto_destino);
```

Finalmente, llegamos al procesado de los datos obtenidos en el parseo. En la mayoría de los casos es suficiente con asignar directamente el valor de la variable al campo correspondiente del objeto **ARTICULO**.

Extracto del fichero \lib\model\doctrine\Articulo.class.

```
/*******************************/
/* RESULTADOS PARSEO ************/
/*******************************/
$this->setRutaPDF($area->getAlias().'/'.$this->getRutaPDF());
$this->setCapturaPDF($area->getAlias().'/'.$this->getRutaPDF().'.jpg');

$this->setTitulo($titulo_parseado);
$this->setResumen($resumen_parseado);
$this->setTituloIdioma2($titulo_idioma2_parseado);
$this->setResumenIdioma2($resumen_idioma2_parseado);
$this->setFirstPage($first_page);
$this->setLastPage($first_page+$paginas_totales-1);
$this->setIdioma($idioma);
$this->setIdioma2($idioma2);
```

Sin embargo, tanto los autores y sus direcciones de correo electrónico, como las palabras clave requieren un procesado más complejo. Principalmente porque esta información no se almacena en el propio objeto **ARTICULO**.

Para cada uno de los autores identificados en el parseo tendremos que comprobar si dicho autor ya existe en nuestra base de datos. Si efectivamente dicho autor existe, estableceremos la relación entre el autor y el artículo definiendo una prioridad. Esta prioridad es importante para conocer quiénes son los autores principales de los artículos. En cambio, si dicho autor no existe tendremos que añadirlo.

De esta forma, dentro de un bucle que recorre todos los autores identificados por el parseo, tendremos un código similar al siguiente:

Extracto del fichero \lib\model\doctrine\Articulo.class.

```
$query_autor = Doctrine::getTable('Autor')->createQuery('autor')->where('autor.nombre = ?',$autor_parseado);
$autor = $query_autor->fetchOne();
$autor_id = NULL;

if($autor != NULL)
        {
        $autor_id = $autor->getId();
        }
else
        {
        $autor = new Autor();
```

Cuando el autor no está definido, creamos un nuevo objeto **AUTOR**, le asignamos el nombre identificado en

el parseo, la correspondiente dirección de correo electrónico y usamos el campo ***Revisado*** para indicar que el autor ha sido creado automáticamente por el sistema y necesita ser validado por los administradores.

Si el parseo no proporciona una dirección de correo electrónico para dicho autor, se le asigna la dirección de correo electrónico del autor principal del artículo, para de disponer de un contacto los más próximo posible al autor.

Extracto del fichero \lib\model\doctrine\Articulo.class.

```
$autor = new Autor();
$autor->nombre = $autor_parseado;
if ($emails_parseados[$indice_autores] == NULL)
        {
        //Si el autor no tienen su correspondiente email se le asigna la primera dirección de correo encontrada
        $autor->email = $emails_parseados[0];
        }
else
        {
        $autor->email = $emails_parseados[$indice_autores];
        }
$autor->revisado = 0;
$autor->save();
$autor_id = $autor->getId();
```

De una forma completamente similar se procesan las palabras clave; un bucle recorre todas las palabras clave identificadas por el parseo tanto en el idioma principal como en el idioma secundario, comprobando si dicha palabra clave ya existe en nuestra base de datos. Si efectivamente dicha palabra existe, estableceremos la relación entre la palabra clave y el artículo definiendo un idioma. Si dicha palabra clave no existe tendremos que añadirla.

Si todo el proceso de creación de un nuevo artículo termina correctamente, se le mostrará al usuario la página de revisión y validación de los datos obtenidos con una barra dorada informado de la correcta creación del mismo, como pudo verse en la Figura 4-4.

4.2.3 OTROS CAMBIOS REALIZADOS

Además de las definir los nuevos formularios y el procesado previo y posterior al parseo se han realizado unas sencillas modificaciones en la interfaz del "backend", principalmente para simplificar el acceso a la herramienta que permite el procesado de nuevos documentos.

Hasta ahora, una vez autenticados, los administradores tenían que seleccionar la opción **Artículos** del menú **Gestor de Artículos** antes de poder acceder a la opción de añadir documentos. Ahora, la opción **Nuevo Artículo** está directamente disponible desde el menú **Gestor se Artículos**.

Además de simplificar el acceso, con este cambio evitamos que se hagan consultas innecesarias a la base de

datos, ayudando a mejorar el rendimiento de la aplicación.

El nuevo menú puede verse en la Figura 4-5:

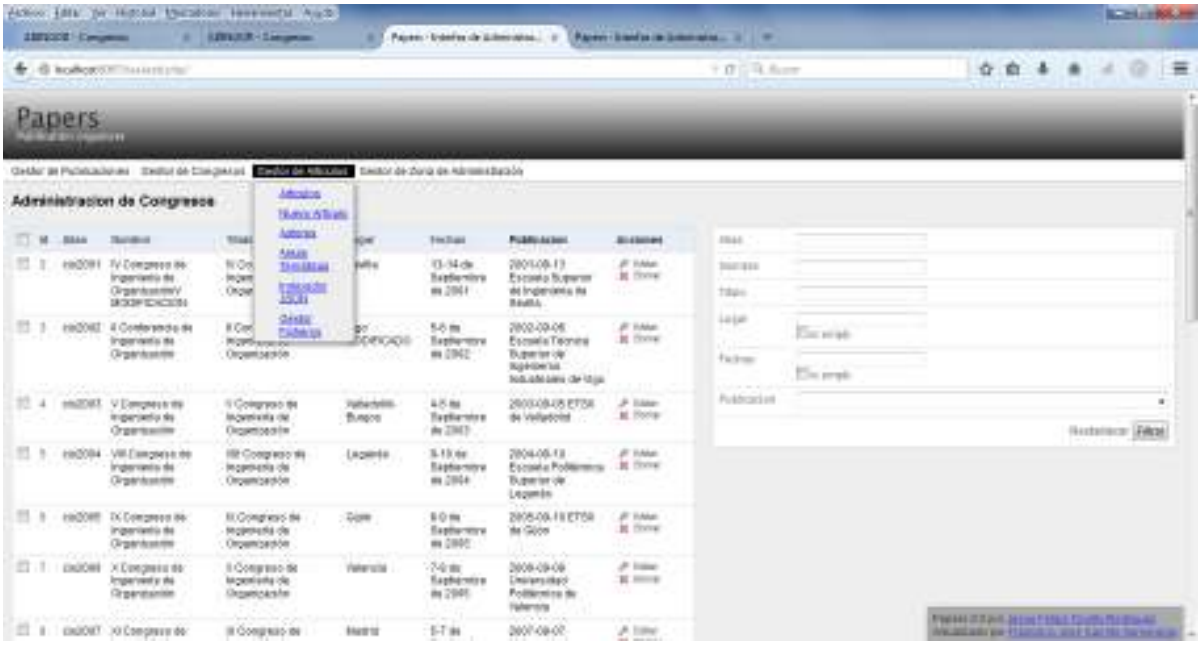


Figura 4-5. Nuevo menú **Gestor de Artículos** en el "backend".

Para implementar este cambio se agregó una línea en el archivo ***\apps\backend\templates\layout.php***.

Extracto del fichero \apps\backend\templates\layout.php.

```
<li id="handler-articulos" class="main-nav-element" onclick="toggleMenu('articulos')">
Gestor de Articulos
<ul id="menu-articulos" class="nav-element-menu-container">
<li class="nav-element"><?php echo link_to('Articulos', 'articulo'); ?></li>
<li class="nav-element"><?php echo link_to('Nuevo Articulo', 'articulo/new'); ?></li>
<li class="nav-element"><?php echo link_to('Autores', 'autor'); ?></li>
<li class="nav-element"><?php echo link_to('Areas Temáticas', 'area_tematica'); ?></li>
<li class="nav-element"><?php echo link_to('Indexador JSON', 'indexer/index'); ?></li>
<li class="nav-element"><?php echo link_to('Gestor Ficheros', 'sf_media_browser/index'); ?></li>
</ul>
</li>
```

Para cerrar este capítulo conviene mencionar que también se hicieron pequeñas modificaciones en las **CCS** [13] o **Cascading Style Sheets** (Hojas de Estilo en Cascada), principalmente para que la presentación del título y el resumen en el idioma principal quedara equiparada con la del idioma secundario.

5 PARSEO AUTOMÁTICO DE NUEVOS ARTÍCULOS

5.1 DISEÑO

En este tercer y último capítulo dedicado a la descripción detallada del trabajo realizado a lo largo de todo el Proyecto nos centraremos en el parseo automático. Ya sabemos que para agregar un nuevo artículo a la herramienta solamente necesitamos el documento con el artículo en formato **PDF**, seleccionar la regla de parseo, el Congreso, el Área Temática y, si existe, el **DOI**. El objetivo del parseo automático es localizar en el texto extraído del archivo en formato **PDF** el resto de datos que caracterizan al documento.

Para identificar esta información se usan palabras o posiciones de referencia. Por ejemplo, supongamos que los artículos de un congreso se escriben usando una plantilla en la que la lista de autores comience por la palabra "*Authors*". Además, sabemos que cada autor se separa del siguiente usando un punto y coma, y que la lista completa de autores termina en un punto. De esta forma, si localizamos la cadena "*Authors"* dentro del texto, podremos identificar los autores del artículo de forma automática.

Ya se ha comentado anteriormente que el parseo automático está implementado en la función **buscaDatos**, de la clase **ReglasDeParseo**. Esta función es invocada durante el procesado de nuevos artículos desde la función **save**, de la clase **Articulo**. En el siguiente extracto de código se puede volver a ver cómo se realiza esta llamada:

Extracto del fichero \lib\model\doctrine\Articulo.class.php.

```
class Articulo extends BaseArticulo
{
        public function save(Doctrine_Connection $conn = NULL)
                {
                if ($this->isNew())
                        {
                        […]
                        $datos_parseados->buscaDatos(
                        $this->getRutaPDF(),
                        $this->getReglaParseo(),
```

```
                                $titulo_parseado,
                                $resumen_parseado,
                                $titulo_idioma2_parseado,
                                $resumen_idioma2_parseado,
                                $autores_parseados,
                                $emails_parseados,
                                $tags_parseados,
                                $tags_idioma2_parseados,
                                $first_page,
                                $idioma,
                                $idioma2
                                );
                                [...]
                                }
        return parent::save($conn);
        }
}
```

Las dos primeras variables que recibe la función **buscaDatos** son *$this->getRutaPDF()* y *$this->getReglaParseo()*. Estas variables se usan para conocer el nombre del fichero que contiene el texto extraído del artículo en formato **PDF** y la regla de parseo a usar respectivamente.

El resto de variables que recibe la función **buscaDatos** son pasadas por referencia, precisamente para que la función pueda modificar su valor. De esta forma cuando se termine de ejecutar la función **buscaDatos**, las variables *$titulo_parseado*, *$resumen_parseado*, *$titulo_idioma2_parseado*, *$resumen_idioma2_parseado*, *$autores_parseados*, *$emails_parseados*, *$tags_parseados*, *$tags_idioma2_parseados*, *$first_page*, *$idioma* e *$idioma2* contendrán la información obtenida en el parseo.

Por lo tanto, la función **buscaDatos** proporciona las siguientes características del artículo:

- El título.
- El resumen.
- El título en el idioma secundario.
- El resumen en el idioma secundario.
- Los autores y sus direcciones de correo electrónico.
- Las palabras clave tanto en el idioma principal del artículo como en el idioma secundario.
- El número de la primera página del documento.
- El idioma principal.
- Y por último, el idioma secundario.

Todos estos datos deben ser localizados en el texto, excepto el idioma principal y el idioma secundario que son definidos por la propia regla de parseo.

5.2 IMPLEMENTACIÓN

Toda la implementación del parseo automático se encuentra en la clase **ReglasDeParseo**, del archivo ***\lib\parseo\ReglasDeParseo\ReglasDeParseo.php***.

Antes de poder usar las herramientas que proporciona esta nueva clase, es necesario cargarla, usando el comando **require_once**:

```
require_once(dirname(__FILE__).'/../../parseo/ReglasDeParseo/ReglasDeParseo.php');
```

Dentro de la clase, se ha definido la función, **buscaDatos**. Su definición puede verse a continuación:

Extracto del fichero \lib\parseo\ReglasDeParseo\ReglasDeParseo.php.

```
class ReglasDeParseo
{
	protected $objects = array();

	public function __construct()
	{
	}

	public function buscaDatos(
			$ruta_PDF,
			$regla_parseo,
			&$titulo,
			&$resumen,
			&$titulo_idioma2,
			&$resumen_idioma2,
			&$autores,
			&$emails,
			&$tags,
			&$tags_idioma2,
			&$first_page,
			&$idioma,
			&$idioma2
			)
			{
			[...]
			}
}
```

5.2.1 FUNCIÓN DE PARSEO

El código de la función **buscaDatos** se puede dividir en tres partes. Un primera y común a todas las reglas de parseo sirve para cargar el texto extraído del artículo en formato **PDF**. En cambio, la segunda parte es propia de cada regla. Por último, la tercera parte de la función vuelve a ser común a todas las reglas y, básicamente, son una serie de comprobaciones.

Aunque del archivo en formato **PDF** se extrae la totalidad del texto, para el parseo automático únicamente seleccionamos los 6.000 primeros caracteres. Se está suponiendo que la información que necesitamos localizar se encuentra en la primera o, como mucho, en la segunda página del documento original. Esta decisión también contribuye a mejorar el rendimiento de la aplicación porque se limita el texto sobre el que se aplican las búsquedas, con el consiguiente ahorro de procesamiento.

Extracto del fichero \lib\parseo\ReglasDeParseo\ReglasDeParseo.php.

```
$fichero = file_get_contents(dirname(__FILE__).'/../../../web/uploads//'.$ruta_PDF.'.txt');

$texto = substr($fichero,0,6000);
```

A continuación una sucesión de bucles *if/elseif* determina la regla de parseo que se debe usar:

Extracto del fichero \lib\parseo\ReglasDeParseo\ReglasDeParseo.php.

```
/*********************************/
/* REGLA DE PARSEO: *************/
/*********************************/
/* 'CIO2013_ENGLISH_TRACKS' *****/
/*********************************/
if ($regla_parseo == 'CIO2013_ENGLISH_TRACKS')
{
[...]
}
/*********************************/
/* REGLA DE PARSEO: *************/
/*********************************/
/* 'CIO2013_SPANISH_TRACKS' *****/
/*********************************/
elseif ($regla_parseo == 'CIO2013_SPANISH_TRACKS')
{
[...]
}
[...]
```

La Figura 5-1 muestra el documento que implementa de las distintas reglas de parseo:

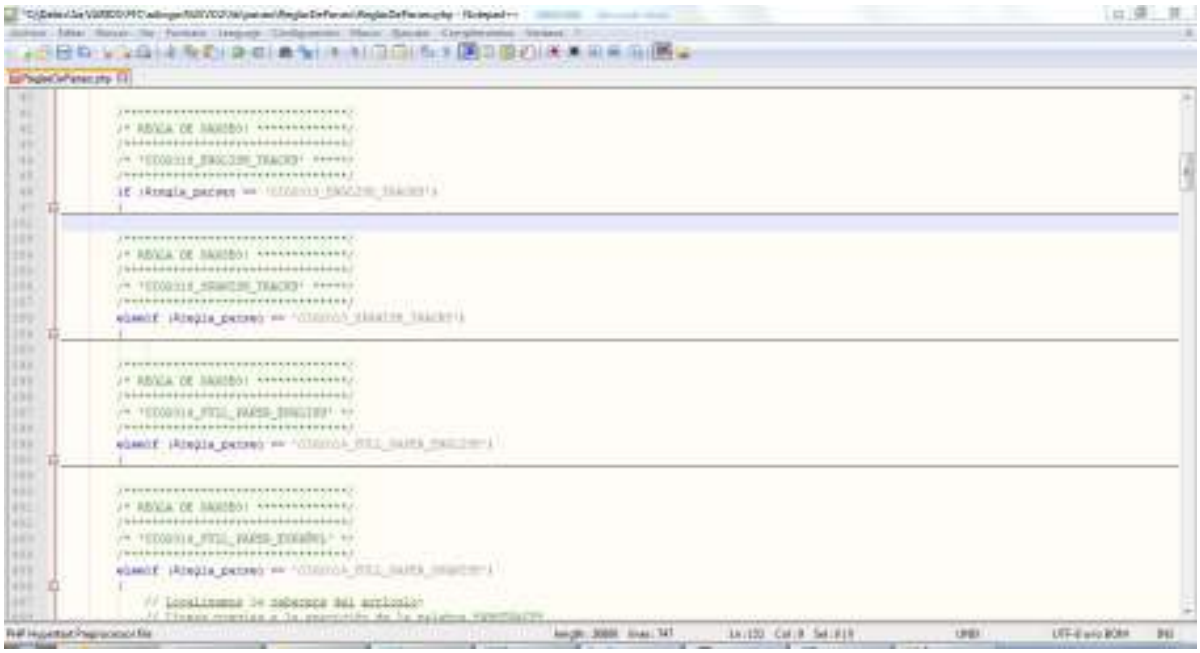


Figura 5-1. Implementación de las reglas de parseo.

La tercera y última parte de la función, que recordamos que vuelve a ser común a todas las reglas, sirve para hacer una serie de comprobaciones y ajustes de los resultados.

Lo primero que se comprueba son los resultados obtenidos en el parseo del título y del resumen. Dada la importancia de estos dos campos, si el parseo no ha proporcionado ningún resultado, se incluyen un mensaje que informa del resultado al administrador. Si por el contrario, el parseo ha encontrado un título o un resumen se utiliza la función **preg_replace** para que todos los espacios con más de un hueco (“#\s+#”) se sustituyan por un solo espacio (“ “). A continuación, la función **trim** elimina los espacios en blanco que haya al principio o del final de la cadena de la cadena de texto.

Extracto del fichero \lib\parseo\ReglasDeParseo\ReglasDeParseo.php.

```
/*********************************/
/* AJUSTES COMUNES **************/
/*********************************/

if ($titulo == “”)
{
$titulo = “TITULO NO ENCONTRADO SIGUIENDO EL PATRÓN DE PARSEO”;
}
else
{
$titulo = preg_replace(“#\s+#”,” “, $titulo);
$titulo = trim($titulo);
}
```

De forma completamente similar se comprueba que no aparezcan espacios con más de un hueco en blanco, ni espacios al principio o al final de la cadena de texto identificada como un autor o una dirección de correo electrónico.

Extracto del fichero \lib\parseo\ReglasDeParseo\ReglasDeParseo.php.

```
if ($autores!=NULL)
{
	foreach($autores as &$autor)
	{
		$autor = preg_replace("#\s+#"," ", $autor);
		$autor = trim($autor);
	}
}

if ($emails!=NULL)
{
	foreach($emails as &$email)
	{
		$email = preg_replace("#\s+#"," ", $email);
		$email = trim($email);
	}
}
```

Antes de terminar la función, comprueba si el parseo ha podido localizar el número de la primera página del documento. En caso contrario, para evitar devolver un valor vacío que pudiera generar problemas en la aplicación, asignamos a la variable el valor 0.

Extracto del fichero \lib\parseo\ReglasDeParseo\ReglasDeParseo.php.

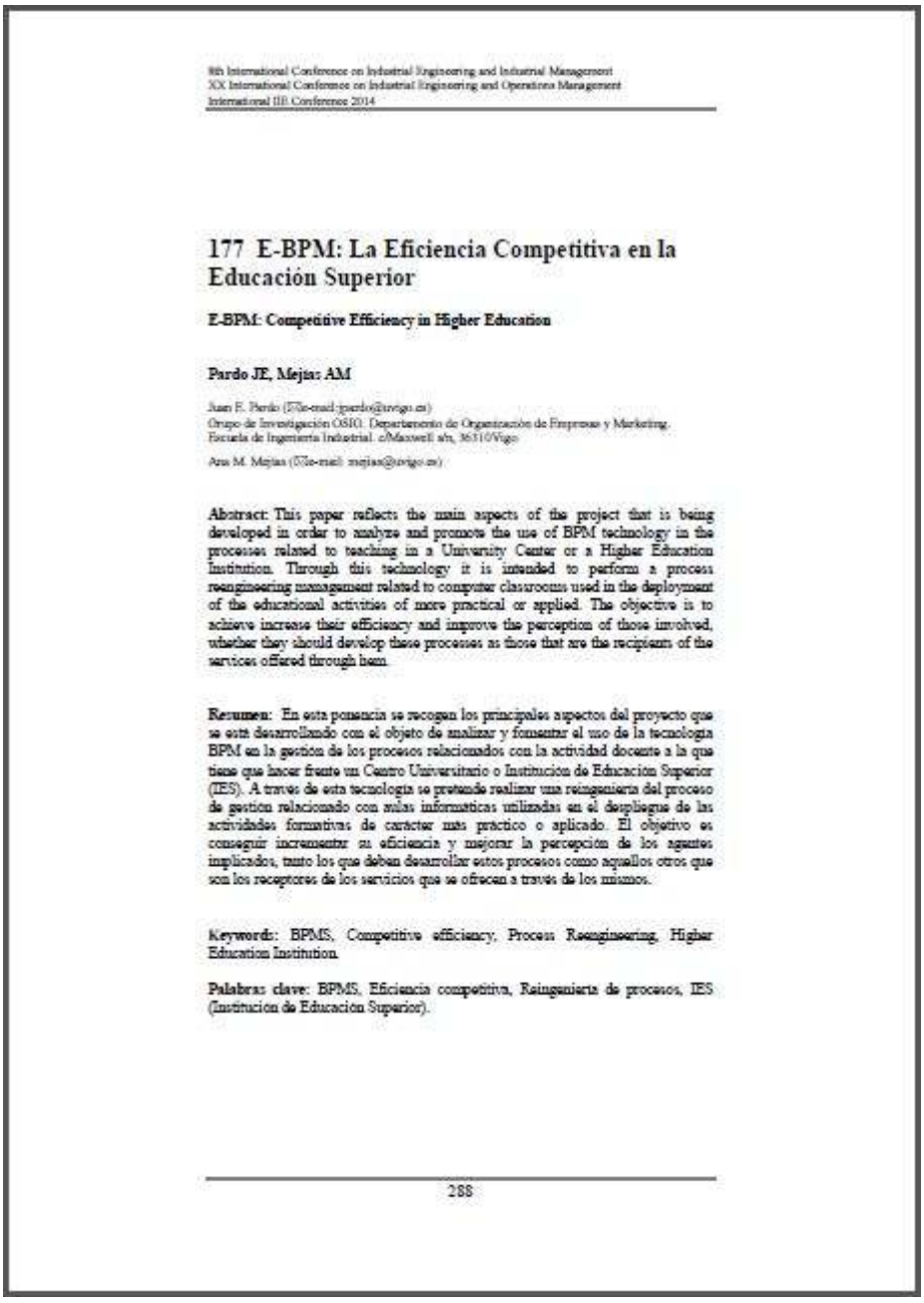
```
if ($first_page == "") {
$first_page = 0;
}

return;
}
```

5.2.2 REGLAS DE PARSEO

En este apartado nos centraremos en identificar las características que necesitamos para clasificar el artículo. Se tomará como ejemplo la regla de parseo diseñada para los artículos del **CIO 2014** escritos en castellano. Esta es una de las reglas más completas de cuantas se han implementado, precisamente porque está diseñada para artículos que tienen información en dos idiomas. Localiza el título, el resumen y las palabras clave tanto en castellano como en inglés.

En la Figura 5-2 se puede ver un ejemplo de un artículo en formato **PDF** al que se le aplicaría esta regla de parseo.



8th International Conference on Industrial Engineering and Industrial Management
XX International Conference on Industrial Engineering and Operations Management
International III Conference 2014

177 E-BPM: La Eficiencia Competitiva en la Educación Superior

E-BPM: Competitive Efficiency in Higher Education

Pardo JE, Mejías AM

Juan E. Pardo (Ele-mail: jpardo@uvigo.es)
Grupo de Investigación OSIO. Departamento de Organización de Empresas y Marketing.
Escuela de Ingeniería Industrial. c/Maxwell s/n, 36310 Vigo

Ana M. Mejías (Ele-mail: mejias@uvigo.es)

Abstract: This paper reflects the main aspects of the project that is being developed in order to analyze and promote the use of BPM technology in the processes related to teaching in a University Center or a Higher Education Institution. Through this technology it is intended to perform a process reengineering management related to computer classrooms used in the deployment of the educational activities of more practical or applied. The objective is to achieve increase their efficiency and improve the perception of those involved, whether they should develop these processes as those that are the recipients of the services offered through them.

Resumen: En esta ponencia se recogen los principales aspectos del proyecto que se está desarrollando con el objeto de analizar y fomentar el uso de la tecnología BPM en la gestión de los procesos relacionados con la actividad docente a la que tiene que hacer frente un Centro Universitario o Institución de Educación Superior (IES). A través de esta tecnología se pretende realizar una reingeniería del proceso de gestión relacionado con aulas informáticas utilizadas en el despliegue de las actividades formativas de carácter más práctico o aplicado. El objetivo es conseguir incrementar su eficiencia y mejorar la percepción de los agentes implicados, tanto los que deben desarrollar estos procesos como aquellos otros que son los receptores de los servicios que se ofrecen a través de los mismos.

Keywords: BPMS, Competitive efficiency, Process Reengineering, Higher Education Institution.

Palabras clave: BPMS, Eficiencia competitiva, Reingeniería de procesos, IES (Institución de Educación Superior).

288

Figura 5-2. Ejemplo de artículo en formato **PDF**.

Para implementar una nueva regla de parseo lo primero que se debe hacer es analizar el mayor número posible de artículos sobre los que queremos aplicar dicha regla. Es posible que no sea viable usar una única regla para todos los artículos y sea necesario desarrollar más de una. En el caso de los artículos del **CIO 2014** se tuvieron que implementar tres reglas diferentes; una para los artículos escritos en castellano, otra para los artículos escritos en inglés y otra para una categoría especial de documentos que llamados “extendend abstracts”.

Una vez que sabemos sobre qué artículos se aplicará la regla se debe hacer un segundo análisis. En este caso el objetivo es comenzar a identificar patrones que sean comunes a todos los documentos. Por ejemplo, buscando en qué orden aparece la información o que palabras preceden a cada uno de los datos que buscamos.

En el caso concreto de los artículos del **CIO 2014** contamos con la ventaja adicional de disponer de la plantilla que debían usar los autores para escribir el artículo:

Figura 5-3. Plantilla de los artículos del **CIO 2014**.

Por lo tanto, sabemos que primero nos encontraremos con un encabezado con los datos del Congreso, seguido del título en castellano y del título en inglés. Tras el título en inglés viene la lista completa de autores, aunque sus direcciones de correo electrónico tendremos que localizarlas al final de la página. El resumen en inglés viene precedido de la palabra "Abstract" y el resumen en castellano de "Resumen". De forma similar las palabras clave en vienen precedidas de "Keywords" y "Palabras clave" respectivamente.

Con toda esta información se puede comenzar la implementación de la regla de parseo. Ya sabemos cómo dentro de la función **buscaDatos** se seleccionaba la regla de parseo a usar mediante una serie de bucles encadenados:

Extracto del fichero \lib\parseo\ReglasDeParseo\ReglasDeParseo.php.

```
/*********************************/
/* REGLA DE PARSEO: ************/
/*********************************/
/* 'CIO2014_FULL_PAPER_ESPAÑOL' */
/*********************************/
elseif ($regla_parseo == 'CIO2014_FULL_PAPER_SPANISH')
```

```
{
[…]
}
```

Ya tenemos la regla definida, ahora comenzaremos a localizar datos. Primero nos centraremos en los más sencillos. Por ejemplo el resumen en inglés y el resumen en castellano. El resumen en inglés será el texto localizado entre las palabras "Abstract" y "Resumen". Mientras que el resumen en castellano estará limitado por "Resumen" y "Keywords".

Para seleccionar el texto deseado usaremos la función **preg_match_all** [14]. Esta función busca dentro del texto indicado por el segundo parámetro todas las coincidencias de la expresión regular [15] dada por el primero.

El siguiente extracto muestra el código que implementa el parseo del resumen en inglés. Para el resumen en castellano sería completamente similar.

Extracto del fichero \lib\parseo\ReglasDeParseo\ReglasDeParseo.php.

```
/*****************************/
/* ABSTRACT******************/
/*****************************/
preg_match_all("#(abstract)[^\r]+?(resumen)#i",$texto,$resumen_idioma2);

$resumen_idioma2 = preg_replace ("#(abstract)#i","",$resumen_idioma2[0][0]);
$resumen_idioma2 = preg_replace ("#(resumen)#i","",$resumen_idioma2);
$resumen_idioma2 = preg_replace ("#-#i" ,"",$resumen_idioma2);              //Saltos de línea
$resumen_idioma2 = preg_replace ("#[\r|\n]#"," ",$resumen_idioma2);
```

La expresión regular utilizada ha sido:

```
"#(abstract)[^\r]+?(resumen)#i".
```

Al limitar la expresión por corchetes seguidos de una i, estamos indicando que no se tenga en cuenta si las letras aparecen en mayúscula o en minúscula. Algunos de los caracteres y cuantificadores más comunes que tienen un significado especial al usarlos dentro de expresiones regulares se explican en la siguiente tabla:

Carácter	Descripción
\s	Espacio en blanco.
\r	Retorno de carro.
\n	Nueva línea.
n*	Coincide con cualquier cadena que contiene cero o más ocurrencias de

	n.
n+	Coincide con cualquier cadena que contiene una o más ocurrencias de n.
n?	Coincide con cualquier cadena que contiene cero o una ocurrencia de n.
^n	Coincide con cualquier cadena con n al comienzo de la misma.

Tabla 5-1. Caracteres con un significado especial en expresiones regulares.

Igual de sencilla es la identificación de las palabras clave. En el siguiente extracto de código se puede ver como usando la misma función que antes, **preg_match_all**, se selecciona la línea que contiene las palabras clave:

Extracto del fichero \lib\parseo\ReglasDeParseo\ReglasDeParseo.php.

```
/*****************************/
/* KEYWORDS ******************/
/*****************************/

// Keywords: Up to 5 (for papers written in English or Spanish);
// Palabras clave: Máximo 5 (only for papers written in Spanish)
preg_match_all("#(keywords|key[\s]*words)[^\r]+?(palabras[\s]*clave|palabra[\s]*clave|8th)#i",$texto,$linea_tags_idioma2);

$linea_tags_idioma2 = preg_replace ("#(keywords|key[\s]*words|8th)#i","",$linea_tags_idioma2[0][0]);
$linea_tags_idioma2 = preg_replace ("#(palabras[\s]*clave|palabra[\s]*clave)#i","",$linea_tags_idioma2);
$linea_tags_idioma2 = preg_replace ("#[:]#i","",$linea_tags_idioma2);
$linea_tags_idioma2 = preg_replace ("#[-]#i","",$linea_tags_idioma2);                    //Saltos de línea
$linea_tags_idioma2 = preg_replace ("#[\r|\n]#"," ",$linea_tags_idioma2);
$linea_tags_idioma2 = trim($linea_tags_idioma2);

$tags_idioma2 = preg_split("/[,;]+/",$linea_tags_idioma2, -1, PREG_SPLIT_NO_EMPTY);
```

En este caso la expresión regular utilizada ha sido:

```
#(keywords|key[\s]*words)[^\r]+?(palabras[\s]*clave|palabra[\s]*clave|8th)#
```

Puede resultar curiosa pero las pruebas realizadas nos hicieron contemplar no sólo "Keywords", también "Key words" o "Palabra clave" en vez de "Palabras clave". Una vez localizada la línea que contiene las palabras

clave se usa la función **preg_split** para separarlas. En este caso se considera que las palabras estarán separadas por comas o puntos y comas.

Si nos volvemos a fijar en los extractos de código anteriores, se puede ver que una vez identificado el texto deseado, se usa la función **preg_replace** en repetidas ocasiones. Esta función se usa para ajustar el resultado del parseo. Por ejemplo, se usa para evitar que el texto seleccionado como resumen, comience por la propia palabra "Resumen" o para eliminar los guiones que indican que una palabra continúa en la siguiente línea.

Hasta ahora se ha podido localizar el resumen en inglés y el resumen en castellano, así como las palabras clave en inglés y en castellano. Queda identificar el título en los dos idiomas, los autores y sus direcciones de correo y el número de la primera página del artículo.

Continuaremos viendo exactamente cuál es el texto que se extrae del artículo en formato **PDF**. Nos fijaremos en dos cosas principalmente: si al extraer el texto se han conservado los saltos de línea y cómo ha quedado el pie de página. Por ejemplo, para el artículo de la Figura 5-2 el resultado es el siguiente:

```
8th International Conference on Industrial Engineering and Industrial Management XX International Conference on Industrial Engineering and Operations Management  International IIE Conference 2014




288


177  E-BPM: La Eficiencia Competitiva en la
Educación Superior
E-BPM: Competitive Efficiency in Higher Education
Pardo JE, Mejías AM
Juan E. Pardo (?e-mail:jpardo@uvigo.es) Grupo de Investigación OSIG. Departamento de Organización de Empresas y Marketing. Escuela de Ingeniería Industrial. c/Maxwell s/n, 36310Vigo.
Ana M. Mejías (?e-mail: mejias@uvigo.es)
Abstract […]
```

Efectivamente, se puede verificar que los saltos de línea se han conservado y como el pie de página ha quedado justo entre el encabezado y el cuerpo de la página.

Excepto las direcciones de correo electrónico, todos los datos que todavía no hemos localizado se encuentran antes de la primera ocurrencia de la palabra "Abstract". Por lo tanto, se trabajará precisamente con esa parte del texto. El siguiente código muestra cómo se extrae y cómo se divide en líneas usando las conocidas funciones **preg_match_all** y **preg_split**.

Extracto del fichero \lib\parseo\ReglasDeParseo\ReglasDeParseo.php.

```
// Localizamos la cabecera del artículo:
// Líneas previas a la aparición de la palabra "ABSTRACT"
//
// Estas líneas nos proporcionan título, autores y primera página

preg_match_all("#([^\r]+?(abstract))#i",$texto,$cabecera);
```

```
$lineas_cabecera = preg_split("#[\n]+#", $cabecera[0][0]);          //Separación de líneas
$numero_de_lineas_cabecera = count($lineas_cabecera);
```

El número de la primera página del artículo será la primera línea que comience por uno o varios dígitos numéricos seguidos de espacios en blanco o salto de línea. Imponemos la condición de que los dígitos numéricos estén seguidos de espacios en blanco o salto de línea para evitar confusiones con el título del Congreso "8th International". Esta condición se implementa de la siguiente forma:

Extracto del fichero \lib\parseo\ReglasDeParseo\ReglasDeParseo.php.

```
// La primera línea que comience por números nos proporciona la página
for ($contador = 0; $contador < $numero_de_lineas_cabecera; $contador++)
{
        if (preg_match_all("#^[0-9]+?[\s]*$#", $lineas_cabecera[$contador], $linea_numerica))
                {
                $indice_linea_numerica = $contador;
                $first_page = $lineas_cabecera[$contador];
                }
}
```

Seguimos localizando la línea que contiene la lista de los autores. Se trata de una línea que contiene una sucesión de palabras separadas por comas o puntos y comas.

Extracto del fichero \lib\parseo\ReglasDeParseo\ReglasDeParseo.php.

```
// Localizamos la línea que nos proporciona los autores
// Suponemos que la línea de los autores está después del título y que título ocupará al menos dos líneas
for ($contador2 = $indice_linea_numerica+4; $contador2 < $numero_de_lineas_cabecera; $contador2++)
{
        if (preg_match_all("#[,;]+#", $lineas_cabecera[$contador2], $linea_autores))
                {
                $indice_linea_autores = $contador2;
                $todos_los_autores = $lineas_cabecera[$indice_linea_autores];
                break;
                }
        }
```

Si volvemos a revisar el texto que se extrae del artículo en formato **PDF**, vemos como entre la línea con el número de la primera página del artículo y la línea que contiene la lista de los autores estarán tanto el título en castellano como el título en inglés.

```
288

177  E-BPM: La Eficiencia Competitiva en la
Educación Superior
```

```
E-BPM: Competitive Efficiency in Higher Education
Pardo JE, Mejías AM
```

Como la línea con el número de la primera página del artículo y la línea que contiene la lista de los autores ya las tenemos localizadas, se puede seleccionar el texto que contiene tanto el título en castellano como el título en inglés. Las líneas sin texto son excluidas.

Extracto del fichero \lib\parseo\ReglasDeParseo\ReglasDeParseo.php.

```
// El título nos lo proporcionas las líneas que hay entre la página y los autores.
$numero_de_lineas_titulo=0;
for ($contador3 = $indice_linea_numerica+1; $contador3 < $indice_linea_autores; $contador3++)
{
        if (preg_match_all("#^[\s]*$#",$lineas_cabecera[$contador3],$autores))
                {
                // LINEA VACIA
                }
        else
                {
                $titulos[$numero_de_lineas_titulo] = $lineas_cabecera[$contador3];
                $numero_de_lineas_titulo++;
                }
}
```

Como no disponemos de ninguna referencia adicional, supondremos que la mitad de las líneas seleccionadas corresponderán al título en castellano y la otra mitad al título en inglés. Cuando el número de líneas seleccionadas sea impar al título en castellano se le asignará una línea más que al título en inglés.

Únicamente nos queda identificar las direcciones de correo electrónico de los autores. Por suerte, las direcciones de correo electrónico siguen un patrón muy particular que nos permite localizaras sin usar ningún tipo de referencia externa, solamente aplicando la expresión regular apropiada.

Extracto del fichero \lib\parseo\ReglasDeParseo\ReglasDeParseo.php.

```
/*************************/
/* EMAILS ****************/
/*************************/
preg_match_all("#([\s]*)([_a-zA-Z0-9-]+(\.[_a-zA-Z0-9-]+)*([ ]+|)@([ ]+|)([a-zA-Z0-9-]+\.)+([a-zA-Z]{2,}))([\s]*)#i",$texto,$correos);
$emails = $correos[0];
```

Para terminar el código que implementa la regla de parseo se agregan una serie de comprobaciones y ajustes propios de la regla.

Se verifica que tanto el campo del título como el del resumen en inglés no estén vacíos. Si estos datos no hubieran sido localizados en el parseo se añade un mensaje indicándolo. Si recordamos, una verificación completamente similar se realiza para el título y el resumen en castellano como parte de las comprobaciones comunes a todas las reglas de parseo.

Extracto del fichero \lib\parseo\ReglasDeParseo\ReglasDeParseo.php.

```
/************************/
/* AJUSTES REGLA PARSEO **/
/************************/
if ($titulo_idioma2 == "")
{
        $titulo_idioma2 = "TITULO NO ENCONTRADO SIGUIENDO EL PATRÓN DE PARSEO";
}

if ($resumen_idioma2 == "")
{
        $resumen_idioma2 = "RESUMEN NO ENCONTRADO SIGUIENDO EL PATRÓN DE PARSEO";
}
```

Las dos últimas líneas de la implementación de la regla de parseo definen tanto el idioma principal como el secundario.

Extracto del fichero \lib\parseo\ReglasDeParseo\ReglasDeParseo.php.

```
$idioma         = 'Español';
$idioma2        = 'Inglés';
```

6 MANUAL DE USUARIO

Este capítulo es un manual dirigido a los administradores de la aplicación que explica paso a paso cómo añadir tanto un nuevo artículo como una nueva regla de parseo.

6.1 NUEVO ARTÍCULO

Las tareas de gestión se realizan desde el "backend", que tiene un acceso restringido. Por lo tanto, el primer paso será identificarse correctamente usando el nombre de usuario y la contraseña. Una vez autenticados seleccionamos la opción **Nuevo Artículo** del desplegable **Gestor de Artículos** como puede verse en la Figura 6-1.

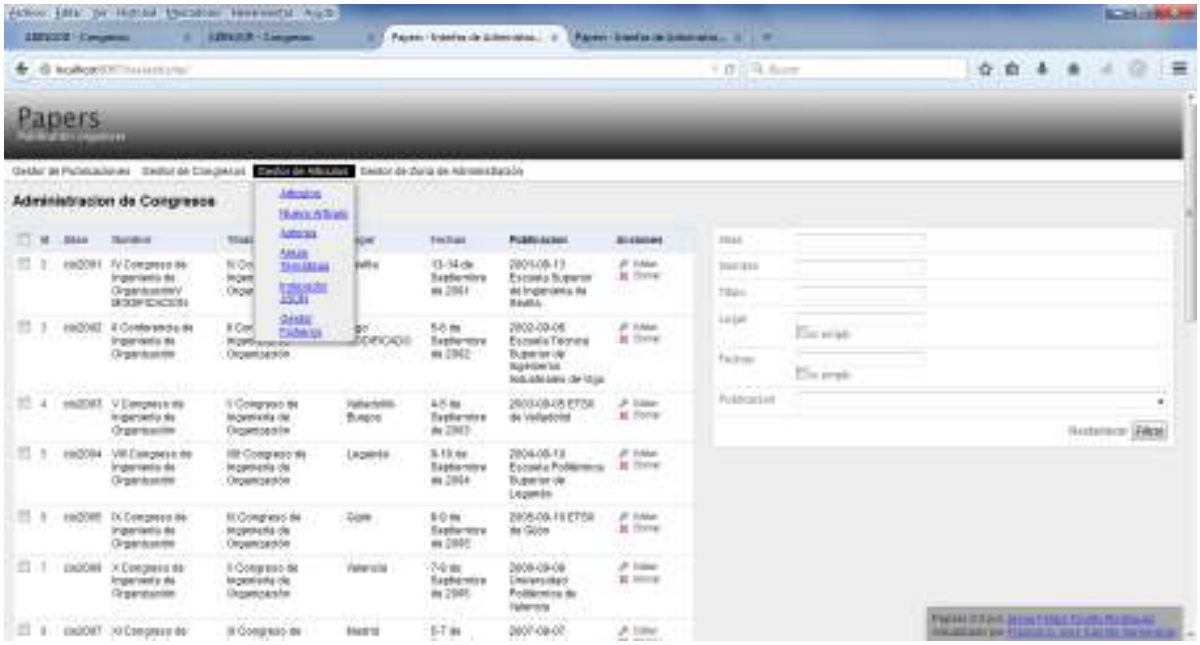


Figura 6-1. Nuevo menú **Gestor de Artículos** en el "backend".

Aparecerá el formulario diseñado para añadir nuevos artículos. Como se puede ver sólo son necesarios 5 datos:

- El documento con el artículo en formato **PDF**.

- La regla de parseo que debe usar el sistema.
- El Congreso.
- El Área temática.
- Y, si existe, el **DOI**.

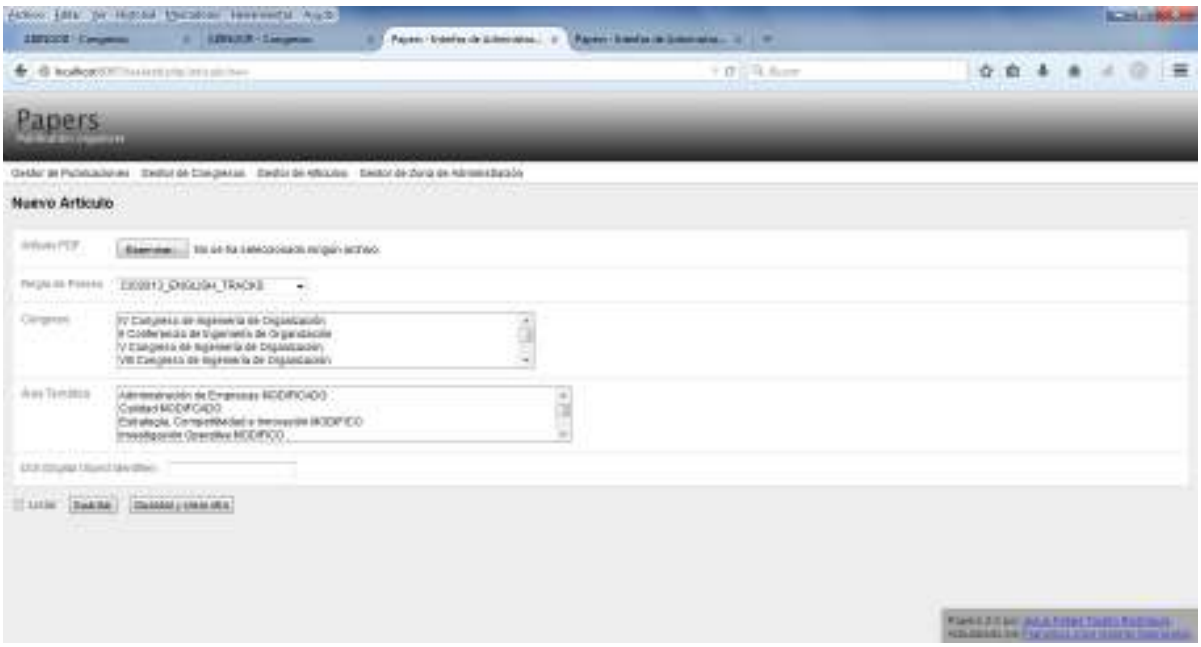


Figura 6-2. Formulario para añadir un nuevo artículo.

Para cargar el documento con el artículo en formato **PDF** pulsamos sobre el botón **Examinar** y se abrirá una nueva ventana que nos permitirá movernos por las carpetas de nuestro ordenador hasta localizar el archivo correspondiente.

La regla de parseo, el Congreso y el Área temática se seleccionan de la lista correspondiente que presenta el formulario. En el menú desplegable **Gestor de Congresos** se encuentran las herramientas para definir nuevos Congresos y Áreas temáticas.

Una vez definidos todos los datos, pulsamos sobre el botón **Guardar**. En ese momento comenzará el procesado del artículo. Una vez terminado, podremos ver un nuevo formulario donde revisar los datos que se han obtenido automáticamente, y si fuese necesario corregirlos.

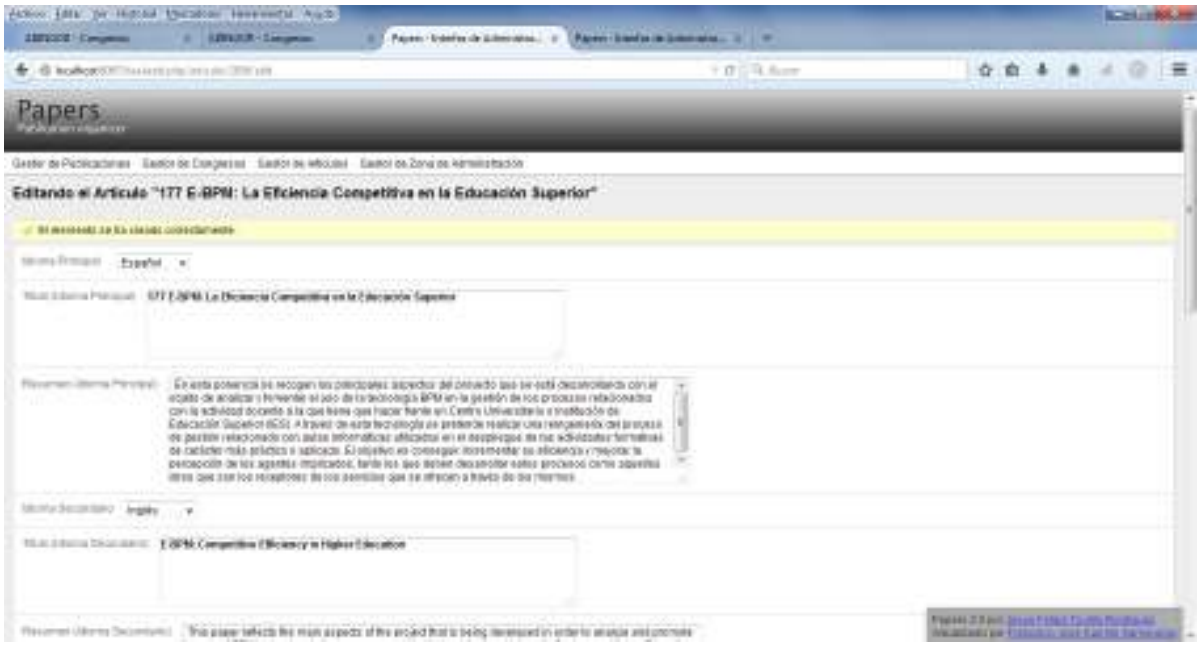


Figura 6-3. Formulario de verificación de nuevos artículos.

En este punto el artículo estaría correctamente incorporado.

Ahora bien, si queremos agilizar el proceso; una vez completado el formulario para añadir un nuevo artículo, en vez de pulsar sobre el botón **Guardar**, podemos hacerlo sobre el botón **Guarda y crear otro** que hay justo a la derecha. Al seleccionar esta opción, cuando el artículo termina de ser procesado, en vez de aparecer el formulario de verificación, aparecerá de nuevo el formulario de creación de nuevos artículos. Esta opción es útil para añadir artículos rápidamente, posponiendo la revisión y posible corrección de los datos obtenidos automáticamente en el parseo.

6.2 NUEVA REGLA DE PARSEO

Ni la creación ni la modificación de reglas de parseo están integradas en la herramienta, por lo tanto cualquier cambio se tiene que realizar modificando el código fuente de la aplicación. Para incluir una nueva regla de parseo en el formulario, como las que pueden verse en la Figura 6-4, el primer paso es definirla en el archivo ***\lib\model\doctrine\ArticuloTable.class.php***.

Extracto del fichero \lib\model\doctrine\ArticuloTable.class.php.

```
static public $reglas = array
(
'CIO2013_ENGLISH_TRACKS'          => 'CIO2013_ENGLISH_TRACKS',
'CIO2013_SPANISH_TRACKS'          => 'CIO2013_SPANISH_TRACKS',
'CIO2014_FULL_PAPER_ENGLISH'      => 'CIO2014_FULL_PAPER_ENGLISH',
'CIO2014_FULL_PAPER_SPANISH'      => 'CIO2014_FULL_PAPER_SPANISH',
'CIO2014_EXTENDED_ABSTRACTS'      => 'CIO2014_EXTENDED_ABSTRACTS',
'NUEVA_REGLA_DE_PARSEO'           => 'NUEVA_REGLA_DE_PARSEO',
```

```
);
```

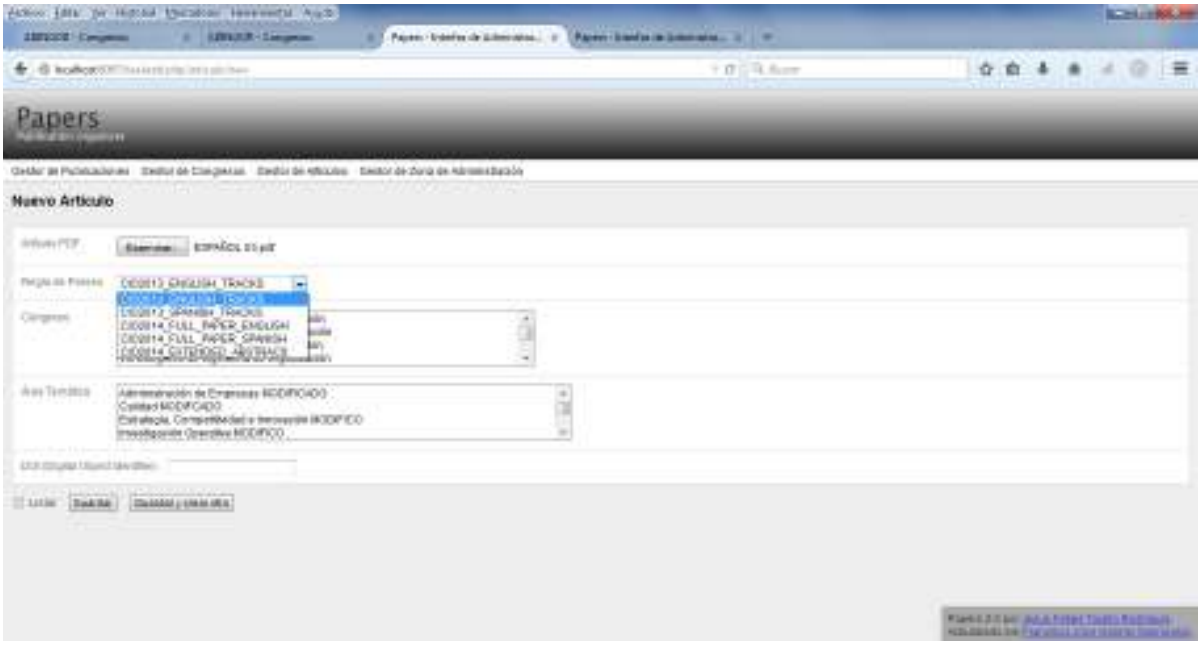


Figura 6-4. Detalle del formulario para añadir un nuevo artículo.

Las tareas que realiza la regla de parseo se definen en la función **buscaDatos**, de la clase **ReglasDeParseo**, del archivo ***\lib\parseo\ReglasDeParseo\ReglasDeParseo.php***. En esta función se definen todas las reglas de parseo. Nos ayudamos de comentarios para organizar el código y hacerlo más legible.

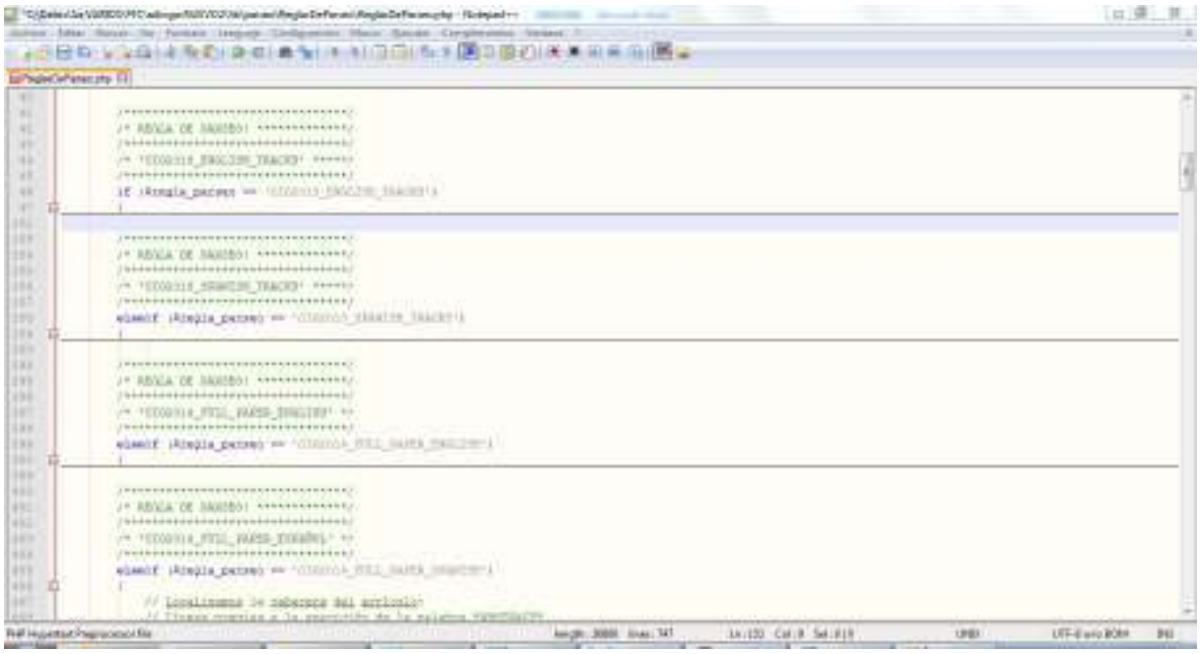


Figura 6-5. Implementación de las reglas de parseo.

Para incluir una nueva regla basta con añadir una nueva comprobación *elseif* a la función **buscaDatos**.

Extracto del fichero \lib\parseo\ReglasDeParseo\ReglasDeParseo.php

```
/*********************************/
/* REGLA DE PARSEO: *************/
/*********************************/
/* 'NUEVA_REGLA_DE_PARSEO' *****/
/*********************************/
elseif ($regla_parseo == 'NUEVA_REGLA_DE_PARSEO')
{
[…]
}
```

Todo el código necesario para localizar los datos que caracterizan un artículo estará incluido dentro del bucle *elseif*. A modo de ejemplo, el siguiente extracto de código muestra una nueva regla de parseo donde los datos de los artículos son definidos por la propia regla.

Extracto del fichero \lib\parseo\ReglasDeParseo\ReglasDeParseo.php

```
/*********************************/
/* REGLA DE PARSEO: *************/
/*********************************/
/* 'NUEVA_REGLA_DE_PARSEO' *****/
/*********************************/
elseif ($regla_parseo == 'NUEVA_REGLA_DE_PARSEO')
{
        $titulo                 = “TITULO”;
        $resumen                = “RESUMEN”;

        $titulo_idioma2         = “TITLE”;
        $resumen_idioma2 =       “ABSTRACT”;

        autores[0]              = “FRANCISCO JOSE GARRIDO”;
        $emails[0]              = “fcojosegarrido@gmail.com”;

        $tags[0]                = “PRUEBA”;
        $tags[1]                = “EJEMPLO”;
        $tags_idioma2[0]        = “TRY”;
        $tags_idioma2[1]        = “EXAMPLE”;

        $first_page             = 1982;

        $idioma                 = 'Español';
        $idioma2                = 'Inglés';
}
```

7 VALIDACIÓN Y PRUEBAS

Una vez finalizado el diseño y la implementación de los cambios descritos en los capítulos anteriores se realizaron varias pruebas para verificar el correcto funcionamiento de la herramienta.

Primero se comprobó el funcionamiento general de la aplicación. Se probó que la presentación de todas las páginas tanto del "frontend" como del "backend" fuera correcta con independencia del navegador utilizado. También se probó que los enlaces funcionan correctamente.

A continuación, se realizó una extensa batería de pruebas para medir el número de errores obtenidos en el parseo automático de los artículos de los **CIO 2013** y **CIO 2014**.

Para evaluar la calidad del parseo de cada una de las reglas se analiza el resultado de 5 artículos seleccionados aleatoriamente.

7.1 VERIFICACIÓN GENERAL DE LA APLICACIÓN

En esta primera fase de pruebas se comprobó que la aplicación se presentaba correctamente usando varias versiones de los navegadores **Internet Explorer**, **Mozilla Firefox** y **Google Chrome**.

Sólo se encontró un error significativo. En el "frontend", en la página que muestra los detalles de un artículo concreto, se observó que si el artículo había sido definido usando el nuevo procesado automático no funcionaba el enlace que nos permite descargar el documento con el artículo en formato **PDF**.

El problema era causado por la información que almacenaba en el campo **RutaPDF**. Básicamente un artículo dado de alta en la aplicación web original almacenaba en el campo **RutaPDF** la cadena: ***Alias_Area_Tematica*****Nombre_del_Documento.PDF***, mientras que un artículo procesado automáticamente sólo almacenaba ***Nombre_del_Documento.PDF***.

La aplicación almacena los documentos **PDF** en la ruta ***uploads*****cio**\, clasificados por Congreso y

posteriormente por Área Temática. De forma que la ruta completa al documento sería ***\uploads\cio\Alias_Congreso\Alias_Area_Tematica \Nombre_del_Documento.PDF***.

El código fuente de la llamada al documento **PDF** es el siguiente:

```
<div class="articulo-detalle-acciones">
<a class="papers-Button" href="<?php echo public_path('uploads/cio/'.$congreso->getAlias().'/'.$articulo->getRutaPDF()); ?>"><?php echo image_tag('icons/zoom','class=papers-Button-img'); ?>Ver art&iacute;culo</a>
</div>
```

Vemos como la ruta de la llamada al documento es ***\uploads\cio\Alias_Congreso***, por lo tanto para poder llegar a identificar el documento, necesitamos tanto la subcarpeta correspondiente al Área temática como el nombre del documento.

La solución adoptada fue cambiar la información que el procesado automático almacena en el campo **RutaPDF** para hacerla compatible con el sistema original. Para ello simplemente se tuvo que cambiar una línea de la función **save**, de la clase **Articulo**, del archivo ***\lib\form\doctrine\ArticuloForm.class.php***:

```
$this->setRutaPDF($area->getAlias().'/'.$this->getRutaPDF());
```

Tras la modificación se comprobó que efectivamente ahora sí era posible descargar el documento **PDF** de un artículo creado usando el procesado automático.

7.2 REGLAS DE PARSEO CIO 2013

Para los artículos del **7th International Conference on Industrial Engineering and Industrial Management** y **XVII Congreso de Ingeniería de Organización**, al que de forma abreviada denominaremos **CIO 2013**, se crearon dos reglas de parseo diferentes; una para los artículos escritos en inglés y otra para los artículos escritos en castellano.

Los errores detectados se señalarán subrayados en rojo o en amarillo. Cuando se detecte un error grave en el resultado del parseo se subrayará en rojo y automáticamente el campo correspondiente pasará a ser considerado como erróneo. En cambio, los errores que no sean graves se subrayarán en amarillo. Estos errores afectan más al estilo y a la correcta legibilidad que a la propia información localizada. Estos errores no hacen que el campo correspondiente pase a ser considerado como erróneo.

7.2.1 CIO2013_ENGLISH_TRACKS

La principal característica de esta regla de parseo es que está diseña para artículos escritos en inglés que no incluyen datos en otros idiomas. Por lo tanto, no se busca información ni del título, ni del resumen ni de las palabras clave en otros idiomas.

Ejemplo 1

En este primer ejemplo se puede verificar que todos los datos obtenidos por el parseo automático son correctos.

TITULO: Calculation of the Approaches to CSL in Continuous Review Policy (s,Q) from an Analogy of a Periodic Review Policy (R,S).

RESUMEN: This paper presents two new approximations to compute the cycle service level (CSL) in a continuous review policy (s, Q) not only for the backordering case but also for the lost sales one. In order to develop these approximations we focus on transforming a (R,S) model to a (s,Q) model. As a result, the analogy and the transformation proposed in this paper are different from Silver classical model. Due to huge complexity of the exact CSL calculus the approximate methods are needed.

AUTORES:

Estelles-Miguel S
Albarracin J.M
Cardós M
Guijarro E

EMAILS:

soesmi@omp.upv.es
jmalbarr@omp.upv.es
mcardos@doe.upv.es
eguitar@upvnet.upv.es

TAGS:

Cycle Service Level
Management Stock Policy

PRIMERA PAGINA: 534

Ejemplo 2

Este ejemplo, es otro caso de artículo correctamente parseado. Este artículo sólo incluye la dirección de correo del autor principal. En estos casos, a los nuevos autores que se incorporen a la base de datos se les asignará el correo electrónico del autor principal del artículo.

TITULO: Economic Performance and Financial Profitability: Two Study Cases in F&B Industry

RESUMEN: In this paper we describe some economic and financial ratios in order to reflect about the need to introduce some standards in annual account about innovation and intangible assets. For this, we compare two innovative companies with data from Food & Beverage (F&B) industry, and we show some differences that have to be explained.

AUTORES:

Santandreu-Mascarell C
Canós-Darós L
Vidal-Carreras PI
Valero-Herrero M

EMAILS:

crisanma@omp.upv.es

TAGS:

Food and Beverage Industry
Economic Performance
Financial Profitability
Innovation
Intangible Assets

PRIMERA PAGINA: 859

Ejemplo 3

En este ejemplo tenemos errores en el parseo del título y de los autores. El parseo identifica una sección de texto que contiene tanto el título como la lista de autores. Dentro de esta sección no aparece ninguna cadena de caracteres que pueda servir de referencia, por lo tanto se optó por asignar al título todas las líneas excepto la última. Suponiendo que esta última línea contendría la lista de autores. En este caso, la lista de autores ocupa dos líneas; por lo tanto, la primera línea de la lista de autores ha sido asignada erróneamente al título y solamente se han reconocidos los autores de la segunda línea.

Otro error que se puede observar en este artículo es que en el texto del resumen donde debería aparecer "2006-2009" aparece "20062009". Esto se debe al procesado posterior del texto identificado como resumen y que se realiza usando la función **preg_replace** en repetidas ocasiones. En concreto, los guiones se eliminan porque en el texto original cuando aparecen suele ser para indicar que la última palabra de la línea no está completa y continúa en la siguiente.

TITULO: High-growth Firms: Qualitative Analysis Via Case Study **Insunza Aranzeta G1, Basañez Llantada A2, Ruiz de Arbulo López P3,**

RESUMEN: This work, which forms part of more extensive research into highgrowth firms in the Autonomous Community of the Basque Country during the period **20062009**, attempts to analyse the reasons for high growth firms by four firms identified within it. This growth will be discussed from a qualitative standpoint based on extensive literature regarding highgrowth firms and using the case study as the main methodology. Interviews with managers from these firms together with a partial analysis of the context within

which they were found during the period analysed (sectorial analysis, general economic context, etc.) enable us to interpret the results obtained using the SABI (Sistema de Análisis de Balances Ibéricos) data base in a more dynamic and contextualised way.

AUTORES:

Landeta Manzano B
González Laskibar X

EMAILS:

gaizka.insunza@ehu.es

TAGS:

High Growth Firms
Gazelles
Case studies

PRIMERA PAGINA: 85

Ejemplo 4

Ejemplo correctamente parseado. Este es otro ejemplo donde sólo aparece una dirección de correo electrónico.

TITULO: Information Capability in Basque Country Quality Award Winners

RESUMEN: Given the global environment that companies have to compete in nowadays, changes are so frequent that companies have to adopt a proactive attitude by trying to anticipate those changes. Using quality information while making decisions has become a critical factor for success, and nobody disputes the importance of having this quality information, which comes from the efficient use and management of information. Companies that have such quality information will have a competitive advantage and improve their results. Under the RBV theory, this efficient use and management of information could be considered a capability of a company. The aim of this paper is to explore the degree to which certain companies have developed this information capability. We focused the study on companies committed to Total Quality Management models because, due to the nature of these information intensive models, such companies can be expected to have developed information capability. The findings confirm this fact, although there are still opportunities for improvement.

AUTORES:

Zárraga-Rodríguez M
Álvarez MJ

EMAILS:

mzarraga@unav.es

TAGS:

RBV
Information Capability
Information Practices
EFQM

PRIMERA PAGINA: 693

Ejemplo 5

Ejemplo correctamente parseado.

TITULO: Logistic Management in a Fresh Food Firm: A Case Study

RESUMEN: An efficient and effective logistic system is a strategic objective in any business. This paper presents a real case study of routing problem on a food industry firm. The simplest case of route optimization is the traveling salesman problem. In this paper there are capacity restrictions and different demands at each node. Then, the problem is classified as a capacity vehicle routing problem. In this paper the Neural Network and Tabu Search algorithms, based on previous literature, are used to solve the problem.

AUTORES:

García Márquez F.P
Peña García-Pardo I
Trapero Arenas J.R

EMAILS:

faustopedro.garcia@uclm.es
isidro.pena@uclm.es
juanramon.trapero@uclm.es

TAGS:

CVRP
Neural network
Tabu Search
Logistic Management

PRIMERA PAGINA: 655

7.2.2 Resumen de resultados de la regla de parseo CIO2013_ENGLISH_TRACKS

	Ejemplo 1	Ejemplo 2	Ejemplo 3	Ejemplo 4	Ejemplo 5
Título	**OK**	**OK**	**KO**	**OK**	**OK**
Resumen	**OK**	**OK**	**OK**	**OK**	**OK**
Autores	**OK**	**OK**	**KO**	**OK**	**OK**

Emails	OK	OK	OK	OK	OK
Tags	OK	OK	OK	OK	OK
Primera página	OK	OK	OK	OK	OK

Tabla 7-1. Resumen de resultados de la regla de parseo CIO2013_ENGLISH_TRACKS.

De los 30 datos localizados por la regla de parseo han fallado 2. Se han producido errores de parseo en menos de un 7% de los datos.

7.2.3 CIO2013_SPANISH_TRACKS

Esta regla está diseña para artículos escritos en castellano con información adicional en inglés. En este caso el parseo plantea la dificultad adicional de no poder usar los saltos de línea como referencia. Cuando se extrae el texto de los artículos en formato **PDF** a los que se les aplica esta regla, se pierden los saltos línea. Probablemente como consecuencia de algún error en el formato de la plantilla usada para escribir los documentos originales.

La Figura 7-1 muestra el texto sin saltos de línea que se ha extraído de un artículo:

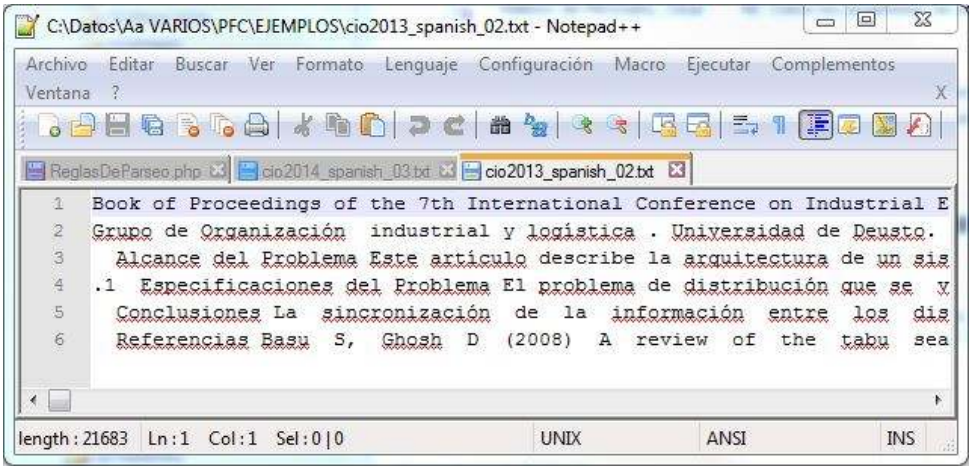


Figura 7-1. Texto sin saltos de línea extraído de un artículo del **CIO 2013**.

Lógicamente, la principal consecuencia es una mayor dificultad para lograr identificar los datos que necesitamos en el proceso de parseo.

También es importante mencionar que, además de perder los saltos de línea al extraer el texto de los artículos

en formato **PDF**, aumenta significativamente el número de palabras en las que erróneamente se incluye un espacio blanco.

Ejemplo 1

En este primer ejemplo se ponen de manifiesto perfectamente los problemas de esta regla de parseo. Se han parseado incorrectamente el título en el idioma principal, el título en el idioma secundario y los autores. Estos errores se repetirán prácticamente en todos los artículos a los que se le aplique esta regla.

Si se revisa el texto extraído del artículo; se puede observar cómo primero aparece el encabezado del congreso y el número de página, seguido del título en castellano. El problema es que justamente a continuación, sin ninguna separación, aparece el título en inglés y el primer autor. Hasta que no aparece la palabra "Abstract" (marcada en negrita) no se dispone de ninguna referencia.

Book of Proceedings of the 7th International Conference on Industrial Engineering and Industrial Management - XVII Congreso de Ingeniería de Organización. 1007 Sistema Integrado de Planificación de la Producción y Distribución para la Gestión de Excepciones An Integrated Production and Distribution Planning System for Exceptions Handling Álvarez E 1, Villalón L 2, Osaba E 3, Díaz F 4 **Abstract** In an International *[…]*

En este ejemplo también puede verse el mensaje que aparece cuando el parseo no logra encontrar ningún valor para el título.

TITULO: Sistema Integrado de Planificación de la Producción y Distribución para la Gestión de Excepciones **An Integrated Production and Distribution Planning System for Exceptions Handling Álvarez E**

RESUMEN: En un entorno internacional tan competitivo, donde las empresas **pu eden** vender sus productos por todo el mundo, la comunicación y el intercambio de

TITULO IDIOMA 2: **TITULO NO ENCONTRADO SIGUIENDO EL PATRÓN DE PARSEO**

RESUMEN IDIOMA 2: In an international competitive environment, where companies can sell their products all over the world, communication and information exchange with other companies is of utmost importance. Nevertheless, most businesses are **relu ctant** to share demand information or future plans with others. This paper describes an integrated production and distribution planning system in a supply chain with three stages: a manufacturer, its suppliers and customers. This system tries to **pr ovide** solutions to improve the performance of the supply chain by identifying the information that must be exchanged between nodes and developing methodologies for production and distribution planning and scheduling in a coordinated manner. Moreover, a number of exceptions have been considered to make the model as **r ealistic** as possible.

AUTORES:

Villalón L
Osaba E
Díaz F

EMAILS:

varez@deusto.es
lvillalon@deusto.es
Eneko.osaba@gmail.com

Fernando.diaz@deusto.es

TAGS:

Gestión de Excepciones
Integración de la Cadena de Suministro
Coordinación en la Cadena de Suministro

TAGS IDIOMA 2:

Exceptions Handling
Supply Chain Management
Coordination in the Supply Chain

PRIMERA PAGINA: 1007

Ejemplo 2

En este ejemplo se han parseado incorrectamente el título en el idioma principal, el título en el idioma secundario y los autores.

Además, se pueden observar otros errores menos importantes provocados porque la herramienta usada para extracción del texto, no identifica apóstrofes, ni comillas y los sustituye por un signo de admiración.

TITULO: Open E-Government y Cambios Organizativos en las Administraciones Públicas Españolas **Open E-Government and Organizational Changes in the Spanish Public A d-ministrations. Martínez Núñez M**

RESUMEN: La crisis y la disminución de confianza en las Administraciones Publicas (AAPP) están produciendo que los gobiernos estén apostando por nuevos sistemas de gobernanza y organización de las AAPP. Los medios sociales han abierto **nu evas** posibilidades sin precedentes de participación del público en la labor del **g obierno** obteniéndose el concepto Open EGovernment. En España se empiezan a promover leyes de transparencia y mayor participación de los ciudadanos como

TITULO IDIOMA 2: **Pérez Aguiar WS**

RESUMEN IDIOMA 2: The crisis and the declining confidence in Public Administrations (PA) are leading Governments to promote new governance and management methods of the Pas. Social media has opened up innovative and unprecedented opportunities for public participation in **governments?** activity and the concept Open E Government has arisen. Laws to promote **g reater** transparency and **citizens?** participation are being proposed in Spain as a way to improve confidence in public **o rganizations**. But Social Media still are a new technology that needs to be better understood in terms of benefits, risks, barriers and strategic use. This study has two objectives: Firstly, to provide an overview of the social media use in the **di fferent** levels of the Spanish public administration. Secondly, to identify the **orga nizational** changes that are taking place assessing indepth the Social Media real impact through the analysis of barriers and organizational efficiency. Results show that social media are improving transparency, but not the concept of collective **di alogue**. The main organizational barriers in implementing an Open E Government lie in the reluctance to change of a part of employees and that the **i ncreasing** complexity of the tasks carried out are not accompanied by incentives to increase production nor by the training required.

AUTORES:

```
                Martin -Fernandez L

EMAILS:
                margarita.martinez@upm.es

TAGS:
                Barreras organizacionales
                Gobierno Electrónico Abierto
                Medios Sociales
                Administraciones Públicas

TAGS IDIOMA 2:
                Organizational Barriers
                Open EGovernment
                Social Media
                Public Administrations

PRIMERA PAGINA: 951
```

Ejemplo 3

En este ejemplo se han parseado incorrectamente el título en el idioma principal, el título en el idioma secundario y los autores.

Este ejemplo nos sirve para comentar otra casuística. La palabra clave "VRP" aparece asociada tanto al idioma principal como al idioma secundario. La aplicación no permite que un mismo artículo se asocie dos veces a la misma palabra, aunque cambie el idioma. Por ese motivo "VRP" sólo estará en lista de palabras clave del idioma principal.

Esta limitación viene como consecuencia de la definición del modelo de datos. Las palabras clave se almacenan en una tabla que contiene únicamente el identificador y la palabra clave en sí. En dicha la tabla no se hace referencia al idioma de la misma.

Es en la tabla **PalabrasClaveArticulo** donde se establece la relación entre el identificador del artículo, el identificador de la palabra clave y donde se define el idioma. Almacenar la información del idioma de las palabras clave en esta tabla fue de unas modificaciones que habían sido realizadas sobre la base de datos original.

Extracto del fichero \config\doctrine\schema.yml correspondiente a la base de datos modificada.

```
PalabrasClaveArticulo:
  connection: doctrine
  tableName: PalabrasClaveArticulo
  columns:
    Articulo_id:
      type: integer(11)
      unsigned: true
```

```
      primary: true
    PalabraClave_id:
      type: integer(11)
      unsigned: true
      primary: true
    idioma:
      type: string(255)
  relations:
    Articulo:
      local: Articulo_id
      foreign: id
    PalabraClave:
      local: PalabraClave_id
      foreign: id
```

Por lo tanto, como en tabla **PalabrasClaveArticulo** no se permite que mismo artículo pueda estar relacionado dos veces con la misma palabra, no se puede establecer la relación de un artículo con una misma palabra en dos idiomas.

TITULO: La Mejora Dinámica del Rutado de Vehículos: Eventos de Reoptimización **The Dynamic Improvement of Vehicle Routing: Reoptimization Events Escudero A**

RESUMEN: Los entornos estáticos de optimización no son todo lo eficientes que se esperaría en situaciones donde existe incertidumbre. El rutado de vehículos es una caso común donde existe incertidumbre, por ejemplo en el tiempo de tránsito, **m otivado** principalmente por los diferentes niveles de congestión existentes. La **reo ptimización** dinámica se ha mostrado más eficaz en este tipo de sistemas. **Determ inar** en que momentos realizar la reoptimización es fundamental en la eficiencia de este tipo de sistemas.

TITULO IDIOMA 2: **Muñuzuri J**

RESUMEN IDIOMA 2: The static environments of optimization are not efficient when there is Uncertainty. The vehicle routing is a common case of it. For example, there **usua lly** is uncertainty in the transit time due to the congestion, traffic jam, etc. Dynamic optimization has been more efficient in these environments. To determinate when a reoptimization have to be run is fundamental.

AUTORES:

Cortés P

Aparicio P

EMAILS:

alejandroescudero@etsi.us.es

TAGS:

Incertidumbre

VRP

Acarreo

Dinamismo

TAGS IDIOMA 2:

Uncertainty

Drayage
Dynamism

PRIMERA PAGINA: 1034

Ejemplo 4

En este ejemplo se han parseado incorrectamente el título en el idioma principal, el título en el idioma secundario y los autores.

Este es otro ejemplo en el que una misma palabra clave aparece asociada tanto al idioma principal como al idioma secundario.

TITULO: Técnicas de Predicción Cuantitativas Aplicadas a la Cadena de Suministro. Un Caso de Estudio **Quantitative Models for Supply Chain Forecasting. A Case Study Trapero Arenas J.R**

RESUMEN: Las ventas sujetas a promociones se pueden pronosticar mediante una técnica estadística univariante (predicción del sistema), que posteriormente se **m odifica** de acuerdo a la opinión de los expertos de la compañía. Este trabajo tiene dos objetivos: en primer lugar, se pretende analizar la precisión de los expertos cuando predicen las ventas y en segundo lugar, se investigan modelos **cuantitat ivos** que puedan reducir o sustituir el ajuste realizado por los expertos. Los **result ados** muestran que bajo ciertas condiciones los expertos consiguen mejorar las **pr edicciones** automáticas del sistema. No obstante, la utilización de modelos

TITULO IDIOMA 2: **TITULO NO ENCONTRADO SIGUIENDO EL PATRÓN DE PARSEO**

RESUMEN IDIOMA 2: Demand forecasting is a complex topic due to different factors like promotions. Generally, promotions may be forecast by using an univariate statistical approach (system forecast) that is judgmentally adjusted by company experts. The present work reports an analysis of the managerial adjustments accuracy when promotions are taking place . Additionally, quantitative models will be assessed as an alternative to judgmental adjustments when referring to forecast promotions. The results show that judgmentally adjusted forecasts on promotion periods may enhance system forecasts, but not systematically and more importantly, multivariate models based on past promotions information might achieve lower forecasting errors than system and judgmentally adjusted forecasts.

AUTORES:

García Márquez F.P

EMAILS:

juanramon.trapero@uclm.es
faustopedro.garcia@uclm.es

TAGS:

Predicción
Funciones de Transferencia
Predicción experta
Marketing

Promociones

TAGS IDIOMA 2:

Forecasting
Transfer Functions
Judgmental Forecasting

Promotions

PRIMERA PAGINA: 1016

Ejemplo 5

Exactamente igual que en los ejemplos anteriores, en este caso también han sido incorrectamente parseados el título en el idioma principal, el título en el idioma secundario y los autores.

Además, no ha podido pasearse completamente el resumen en castellano porque se ha alcanzado el final de la primera página.

TITULO: La Calidad de E-Servicio en Portales Web BC: Evaluación Mediante Sistemas de Inferencia **Borrosos T he E-Service Quality in BC Websites: Evaluation by means of Fuzzy Infe r-ence Systems. Castro A**

RESUMEN: **El presente artículo propone un modelo para evaluar la calidad de eservicio en portales web B2C. Para ello, se ha considerado un conjunto de dime n**

TITULO IDIOMA 2: **TITULO NO ENCONTRADO SIGUIENDO EL PATRÓN DE PARSEO**

RESUMEN IDIOMA 2: The present paper proposes a model to evaluate B2C websites. It has been considered a set of dimensions that have influence in the evaluation **accor ding** to the literature review and a study of reliability . Once validated the model, it makes use of fuzzy inference systems in order to reduce the uncertainty associated with the **decisionmaking** process. As a result of this research, a model capable of designing a simple and intuitive knowledge base is obtained (experts I users) in the evaluation of B2C websites, thus allowing a further optimization of the results obtained.

AUTORES:

Puente J
de la Fuente D
Parreño J
Lozano J

EMAILS:

adriancastrolopez@gmail.com
jpuente@uniovi.es
david@uniovi.es
parreno@uniovi.es
lozano@uniovi.es

TAGS:

Sistemas de Inferencia Borrosos
Eservicio
Calidad de Eservicio
Evaluación de Portales B2C

TAGS IDIOMA 2:

Fuzzy Inference Systems
Eservice
Quality Service
B2C Websites Evaluation

PRIMERA PAGINA: 1145

7.2.4 Resumen de resultados de la regla de parseo CIO2013_SPANISH_TRACKS

	Ejemplo 1	Ejemplo 2	Ejemplo 3	Ejemplo 4	Ejemplo 5
Título	KO	KO	KO	KO	KO
Resumen	OK	OK	OK	OK	KO
Título Idioma 2	KO	KO	KO	KO	KO
Resumen Idioma 2	OK	OK	OK	OK	OK
Autores	KO	KO	KO	KO	KO
Emails	OK	OK	OK	OK	OK
Tags	OK	OK	OK	OK	OK
Tags Idioma 2	OK	OK	OK	OK	OK
Primera página	OK	OK	OK	OK	OK

Tabla 7-2. Resumen de resultados de la regla de parseo CIO2013_SPANISH_TRACKS.

De los 45 datos localizados por la regla de parseo han fallado 16. Se han producido errores de parseo en un 36% de los datos. Más de un tercio de los datos localizados son erróneos.

7.3 REGLAS DE PARSEO CIO 2014

Para los artículos del **8th International Conference on Industrial Engineering and Industrial Management** y **XX International Conference on Industrial Engineering and Operations Management**, al que de forma abreviada denominaremos **CIO 2014**, se crearon tres reglas de parseo diferentes; una para los artículos escritos en inglés, otra para los artículos escritos en castellano y otra diferentes para los "extended abstract".

Exactamente igual que en caso anterior, los errores detectados se señalarán subrayados en rojo o en amarillo. Cuando se detecte un error grave en el resultado del parseo se subrayará en rojo y automáticamente el campo correspondiente pasará a ser considerado como erróneo. En cambio, los errores que no sean graves se subrayarán en amarillo. Estos errores afectan más al estilo y a la correcta legibilidad que a la propia información localizada. Estos errores no hacen que el campo correspondiente pase a ser considerado como erróneo.

7.3.1 CIO2014_FULL_PAPER_ENGLISH

Esta regla de parseo está diseña para artículos escritos en inglés que no incluyen datos en otros idiomas. Por lo tanto, no se busca información ni del título, ni del resumen ni de las palabras clave en otros idiomas.

Ejemplo 1

Ejemplo correctamente parseado. Únicamente se puede señalar que el texto que debería aparecer como "Smart Cities" lo hace cómo ?Smart Cities?, debido a las limitaciones para identificar apóstrofes y comillas de la herramienta usada para extraer el texto de los artículos en formato **PDF**.

TITULO: 012 ESCOs Formation as key factor for smart cities: Spain case analysis

RESUMEN: Cities have reached such a magnitude that they represent platforms for wealth, employment and competitiveness creation but also involve an enormous amount of complexity that emphasized their management challenges. Cities evolution is a trend towards development of more efficient and livable cities called **?Smart Cities?** where key topics are linked to how efficiently they use resources. This document describes the development in Spain of Energy Service Companies (ESCOs), a type of organization focused on promoting and managing projects related to the efficient use of energy, being their business success linked to energy savings achieved by their clients. Outcome of this study provide information of interest to understand current development of ESCO projects in Spain, barriers they faced and how collaboration between organizations can facilitate energy efficiency management, which is linked to future development of **?Smart Cities?** initiatives.

AUTORES:

Morcillo Bellido J
Prida Romero B

EMAILS:

morcillo@ing.uc3m.es
bprida@ing.uc3m.es

TAGS:

Smart Cities
ESCO
collaborative relationship
services management
collaboration

PRIMERA PAGINA: 17

Ejemplo 2

Ejemplo correctamente parseado.

TITULO: 130 Integration of Discrete-event Simulation Model and Optimisation Method for Solving Stochastic Job Shop Dynamic Scheduling Problem

RESUMEN: This work presents an integration of a simulation model with an optimisation method in order to solve the stochastic job shop dynamic scheduling problem. The proposed model integration is accomplished using **outofprocess** components, through the ActiveX Automation technology and the Visual Basic for Application, in which a simple Genetic Algorithm runs as a freestanding application. Results were compared with some common dispatching rules and show that simulation optimization method can solve the scheduling problem efficiently, achieving results up to 40% better compared with common dispatching rules.

AUTORES:

Silva M
Grassi F
Pereira F

EMAILS:

marilda.silva@uninove.br
flaviograssi.fg@gmail.com
fabiohp@uninove.br

TAGS:

Simulation optimization
stochastic scheduling problem
job shop

PRIMERA PAGINA: 204

Ejemplo 3

Ejemplo correctamente parseado.

TITULO: 151 The relationship among Order Picking, Logistics and Supply Chain Management: a reference model to configure an order picking system

RESUMEN: The order picking activity, one of the warehousing activities, is responsible for representative costs in the warehousing. These costs could reach up to 75% of the total warehousing costs (Coyle et al., 1996 apud Petersen and Aase, 2004). The model was developed based on the bibliographic research method, in order to keep the scientific approach of the study. The model was presented in process and activities and it was considered the Logistics and SCM (Supply Chain Management) premises of time, quality and cost. It was also presented a matrix in order to point out the main decisions to be taken in the model, considering four scenarios of **SKU?s** quantity and variety. The main result of this research was the proposal of a method with a managerial approach never studied before.

AUTORES:

Bozutti D F
Costa M A B

EMAILS:

dfbozuti@terra.com.br
mbcosta@ufscar.br

TAGS:

Picking
Logistics
Warehouse Management
Supply Chain Management

PRIMERA PAGINA: 246

Ejemplo 4

Ejemplo correctamente parseado.

TITULO: 273 Competency mapping as a tool to aid organizational learning, and innovation process

RESUMEN: This paper presents a literature review of the main models for competency mapping, revealing the importance of these tools to support the organizational learning process. The study is relevant to present the most used methods, highlighting the advantages and disadvantages of each. It provides grants that can be used as guide for the process of choosing the most appropriate technique to the company profile. Identify the specific skills for the innovations development, and make them, hard core in the company is essential to obtain competitive and comparative advantages, in the market.

AUTORES:

Dias R

Souza Cabral A

EMAILS:

raqueldias2006@gmail.com

cabral@ita.br

TAGS:

Mapping Skills

Organizational Learning

Management

PRIMERA PAGINA: 404

Ejemplo 5

Ejemplo correctamente parseado.

TITULO: 306 Transport logistics supporting the development of high added value supply chains: an analysis of the cellular phones industry in Brazil

RESUMEN: This article presents a discussion about air transport importance for agile supply chains development at Manaus Industrial Pole. Good development of agile supply chains, according Uncertainty Supply Chain Model (Lee, 2002), is important to keep updated industrial park on emergent economies, in order to keep attractiveness for products of technological frontier. Air transport is fundamental for supply logistics because considers agility which is necessary for products of technological frontier. The results obtained on this discussion and analysis confirms the necessity of good development on air transport infrastructure as attractiveness and competitive factor on Manaus Industrial Park, Manaus Industrial Pole (PIM), as case study.

AUTORES:

Oliveira Fabiana L

Oliveira Aristides R

EMAILS:

flucenaoliveira@gmail.com

ristides.jr@homail.com

TAGS:

air Transport

Uncertainty Supply chain Model

Agile supply chain

Manaus industrial pole

logistics

PRIMERA PAGINA: 460

7.3.2 Resumen de resultados de la regla de parseo CIO2014_FULL_PAPER_ENGLISH

	Ejemplo 1	Ejemplo 2	Ejemplo 3	Ejemplo 4	Ejemplo 5
Título	**OK**	**OK**	**OK**	**OK**	**OK**
Resumen	**OK**	**OK**	**OK**	**OK**	**OK**
Autores	**OK**	**OK**	**OK**	**OK**	**OK**
Emails	**OK**	**OK**	**OK**	**OK**	**OK**
Tags	**OK**	**OK**	**OK**	**OK**	**OK**
Primera página	**OK**	**OK**	**OK**	**OK**	**OK**

Tabla 7-3. Resumen de resultados de la regla de parseo CIO2014_FULL_PAPER_ENGLISH.

Los 30 datos localizados por la regla de parseo son correctos. No se han producido errores de parseo.

7.3.3 CIO2014_FULL_PAPER_ESPAÑOL

Esta regla está diseña para artículos escritos en castellano con información adicional en inglés.

Ejemplo 1

Ejemplo correctamente parseado.

TITULO: 061 Una Aplicación de la planificación Docente basada en Competencias en la Dirección de Operaciones

RESUMEN: El Espacio Europeo de Educación Superior supone un cambio en el modo de entender la docencia universitaria, ya que se centra en la formación en competencias y no en contenidos. Eso implica que el alumnado no sólo ha de saber (conocimientos) sino que también ha de saber hacer (habilidades) y saber estar (actitudes). Para ello, es necesario planificar las asignaturas desde la óptica de las competencias, seleccionando los contenidos y las metodologías de aprendizaje que garanticen el desarrollo de dichas competencias. En este trabajo se analiza cómo la propuesta curricular de la asignatura Diseño de Sistemas Productivos y Logísticos del Grado en Ingeniería de Organización Industrial gira en torno a la adquisición de las competencias del título.

TITULO IDIOMA 2: An Application of the Learning Plan based on Competencies to Operations Management

RESUMEN IDIOMA 2: The European Higher Education Area has introduced a new philosophy of university education focused on **competencybased** learning. It differs from other approaches in that students not only should know (knowledge) but also should develop the knowhow (skills) and know how to be (attitude). To do this, lecturers need to plan their courses from the skills perspective, selecting the contents and learning methodologies to ensure the development of these competencies. This paper discusses the curriculum of Design of Production and Logistics Systems in the Industrial Engineering degree and shows how it focuses on the **competencybased** learning.

AUTORES:

Cardós M
Guijarro E
Babiloni E
Vicens E

EMAILS:

mcardos@doe.upv.es
esguitar@upvnet.upv.es
mabagri@doe.upv.es
evicens@omp.upv.es

TAGS:

planificación docente
aprendizaje basado en competencias
coordinación
propuesta curricular

TAGS IDIOMA 2:

learning plan
competencybased learning
coordination
curriculum

PRIMERA PAGINA: 108

Ejemplo 2

Ejemplo correctamente parseado. Este es otro ejemplo en el que una misma palabra clave aparece asociada tanto al idioma principal como al idioma secundario.

TITULO: 107 Propuesta de un marco conceptual para el análisis comparativo de las redes de distribución de dos supermercados online

RESUMEN: En este artículo se analiza y compara la configuración de red para la preparación y distribución de pedidos online de supermercado de dos distribuidores británicos. Para este fin se propone un marco conceptual que comprende los siguientes aspectos clave: configuración de la red, gestión del transporte y localización de la demanda. Como resultado no resulta evidente determinar el grado ideal de centralización de la red de distribución para cada caso. Finalmente se sugiere el futuro desarrollo de una herramienta analítica que ayude a escoger el modelo de distribución más adecuado.

TITULO IDIOMA 2: Proposal of a conceptual framework for the comparative analysis of the distribution networks of two online supermarkets

RESUMEN IDIOMA 2: In this article the network configuration for fulfillment and distribution of online orders of two British retailers is analyzed and compared. For this purpose, it is proposed a conceptual framework that consists of the key following aspects: network configuration, transportation management and location of demand. As a result is not obvious to determine the ideal centralization degree in each case. Finally, it is suggested the future development of an analytic tool that helps to choose the most appropriate model.

AUTORES:

Guerrero-Lorente J
Ponce-Cueto E
Blanco EE

EMAILS:

guerreloja@gmail.com

TAGS:

fulfillment
online
supermercados
red
distribución

TAGS IDIOMA 2:

supermarkets
network
distribution

PRIMERA PAGINA: 180

Ejemplo 3

Ejemplo correctamente parseado. Este es otro ejemplo en el que una misma palabra clave aparece asociada tanto al idioma principal como al idioma secundario.

TITULO: 177 E-BPM: La Eficiencia Competitiva en la Educación Superior

RESUMEN: En esta ponencia se recogen los principales aspectos del proyecto que se está desarrollando con el objeto de analizar y fomentar el uso de la tecnología BPM en la gestión de los procesos relacionados con la actividad docente a la que tiene que hacer frente un Centro Universitario o Institución de Educación Superior (IES). A través de esta tecnología se pretende realizar una reingeniería del proceso de gestión relacionado con aulas informáticas utilizadas en el despliegue de las actividades formativas

de carácter más práctico o aplicado. El objetivo es conseguir incrementar su eficiencia y mejorar la percepción de los agentes implicados, tanto los que deben desarrollar estos procesos como aquellos otros que son los receptores de los servicios que se ofrecen a través de los mismos.

TITULO IDIOMA 2: E-BPM: Competitive Efficiency in Higher Education

RESUMEN IDIOMA 2: This paper reflects the main aspects of the project that is being developed in order to analyze and promote the use of BPM technology in the processes related to teaching in a University Center or a Higher Education Institution. Through this technology it is intended to perform a process reengineering management related to computer classrooms used in the deployment of the educational activities of more practical or applied. The objective is to achieve increase their efficiency and improve the perception of those involved, whether they should develop these processes as those that are the recipients of the services offered through hem.

AUTORES:

Pardo JE
Mejías AM

EMAILS:

jpardo@uvigo.es
mejias@uvigo.es

TAGS:

BPMS
Eficiencia competitiva
Reingeniería de procesos
IES (Institución de Educación Superior)

TAGS IDIOMA 2:

Competitive efficiency
Process Reengineering
Higher Education Institution

PRIMERA PAGINA: 288

Ejemplo 4

Ejemplo correctamente parseado.

TITULO: 214 Importancia de los modelos de conducta en la intención emprendedora en estudiantes de ingeniería

RESUMEN: El emprendimiento de base tecnológica es un tema crítico para la generación de crecimiento económico, por lo que conocer los determinantes de la intención emprendedora de estudiantes de universidades técnicas adquiere una especial relevancia. En este estudio se analiza la importancia de los modelos de conducta en la intención emprendedora entre los estudiantes de carreras técnicas. Los resultados muestran que los estudiantes de padres empresarios tienen una intención emprendedora superior a la media, mientras que los de hijos de padres funcionarios están por debajo de ella. Igualmente, la ausencia de modelos de conducta emprendedora en el entorno cercano disminuye la intención emprendedora, mientras que la ausencia de funcionarios no la hace aumentar.

TITULO IDIOMA 2: The importance of role modelsfor the entrepreneurial intention of technical degrees students

RESUMEN IDIOMA 2: **Technologybased** entrepreneurship is a critical issue for the generation of economic growth. For this reason, the determinants of entrepreneurial intention of students from technical universities are of particular relevance. This study analyzes the importance of role models for the entrepreneurial intention among students of technical degrees. The results show that students whose parents are entrepreneurs score higher than the average in entrepreneurial intention. Contrarily, those whose parents are civil servants are below average in entrepreneurial intention. Furthermore, the absence of role models for entrepreneurship in the near environment decreases entrepreneurial intention, while the lack of role models of civil servants does not influence it.

AUTORES:

Morales-Alonso G
Pablo-Lerchundi I
Vargas-P. A M

EMAILS:

gustavo.morales@upm.es
iciar.depablo@upm.es
ana.vargas@upm.es

TAGS:

emprendimiento
empresas de base tecnológica
influencia de los padres
transferencia de conocimiento
estudiantes de ingeniería

TAGS IDIOMA 2:

entrepreneurship
technologybased companies
parental influence
knowledge transference
engineering students

PRIMERA PAGINA: 338

Ejemplo 5

Ejemplo correctamente parseado. Este es otro ejemplo en el que una misma palabra clave aparece asociada tanto al idioma principal como al idioma secundario.

TITULO: 298 Utilización del mapa de la cadena de valor en un entorno sanitario: un caso de estudio

RESUMEN: El mapa de de la cadena de valor es una de las herramientas fundamentales para poder planificar la implantación de herramientas lean en cualquier entorno. Existe abundante literatura sobre su utilización en entornos industriales incluso y en

entornos de **?oficina?** (lean office), incluso a pesar de la creciente extensión del lean en el sector sanitario la herramienta todavía no se ha usado extensamente. En el presente artículo se pretende abordar una adaptación de la herramienta que permita visualizar en un único mapa los movimientos de información y los movimientos de pacientes que permitan poder entender el funcionamiento de un proceso asistencia en su estado actual, y a partir de ahí, planificar las mejoras.

TITULO IDIOMA 2: Value Stream Map utilization in healthcare environment: a case study

RESUMEN IDIOMA 2: The value stream mapping (VSM) is one of the main tools in order to plan the implementation of lean tools in any environment. It can be found a lot of literature on its use in industrial environments and even in office environments **?office?** (lean office), even though the growing extension of lean in healthcare tool has not yet been widely used there. This paper is intended to address an adaptation of the tool to be displayed on a single map information and patientflow that allow sanitary staff to understand the functioning of a process in its current state, and from there, plan improvements.

AUTORES:

Garcia-Sabater J J
Maheut J
Vidal-Carreras P I

EMAILS:

jugarsa@omp.upv.es
juma2@upv.es
pivicar@omp.upv.es

TAGS:

Mapa de la cadena de valor
cuidado de la saludo
producción ajustada
mejoras
hospital

TAGS IDIOMA 2:

Value Stream map
healthcare
lean manufacturing
improvement

PRIMERA PAGINA: 444

7.3.4 Resumen de resultados de la regla de parseo CIO2014_FULL_PAPER_ESPAÑOL

	Ejemplo 1	Ejemplo 2	Ejemplo 3	Ejemplo 4	Ejemplo 5
Título	**OK**	**OK**	**OK**	**OK**	**OK**
Resumen	**OK**	**OK**	**OK**	**OK**	**OK**

Título Idioma 2	OK	OK	OK	OK	OK
Resumen Idioma 2	OK	OK	OK	OK	OK
Autores	OK	OK	OK	OK	OK
Emails	OK	OK	OK	OK	OK
Tags	OK	OK	OK	OK	OK
Tags Idioma 2	OK	OK	OK	OK	OK
Primera página	OK	OK	OK	OK	OK

Tabla 7-4. Resumen de resultados de la regla de parseo CIO2014_FULL_PAPER_ESPAÑOL.

Los 45 datos localizados por la regla de parseo son correctos. No se han producido errores de parseo.

7.3.5 CIO2014_EXTENDED_ABSTRACTS

Esta regla es un caso excepcional, porque no está diseñada exactamente para artículos, sino para "extendend abstracts". Estos documentos tienen una estructura similar a la de los artículos, pero no incluyen una explicación detallada del estudio realizado, solo un resumen extendido. Suelen ocupar una única hoja y tampoco incluyen palabras clave para su clasificación.

La regla se ha diseñado para documentos escritos en inglés que no incluyen ningún dato en otro idioma.

Ejemplo 1

Ejemplo correctamente parseado.

TITULO: 048 Analysis of credit constraints influence on **companies?** investments from Brazilian electricity sector

RESUMEN: This study aimed to understand the behavior of the companies from Brazilian electricity sector as for the use of their own resources to provide capital for investments when these firms are affected by credit constraints, taking into account their size and degree of financial leverage (DFL). For this purpose, a derivative of the empirical model proposed by Fazzari, Hubbard and Petersen (1988) was applied. Following the study, the regression panel data technique was employed in a database comprising financial information from 16 companies that compose the Electric Power Index (IEE) from BM&FBOVESPA. The results obtained,

when the degree of financial leverage (DFL) was used as the criterion to separate the sample to represent credit constraints, confirm Fazzari, Hubbard and **Petersen?s** (1988) hypothesis. Such results are due to the dependence observed between the cash flow and investment variables for those companies with an unfavourable financial leverage degree, classified as constrained firms. For the unconstrained ones though, with a favourable degree of financial leverage, this dependence was not confirmed. Nevertheless, using the size of companies (based on the value of total assets) as a criterion to separate the sample to designate credit constraints, dependence between cash flow and investment in the large companies, classified as unconstrained firms, was observed. For the small companies, classified as constrained firms, this dependence was not observed. This conclusion contradicts Fazzari, Hubbard and Petersen (1988), but confirms the trend presented by Kaplan and Zingales (1997), who argue that cash flow influence on investment does not grow monotonically with the credit constraint degree in a given company. According to Oliveira and Cunha (2012), the size of the companies is a good criterion to designate credit constraints. Yet, the result obtained in this study differs from their proposition, which is frequently referred to in academic papers.

AUTORES:

Cotomacio A
Rossetti N
Meirelles J

EMAILS:

andrecotomacio@gmail.com
nara@ufscar.br
jorgeluis@ufscar.br

PRIMERA PAGINA: 545

Ejemplo 2

Ejemplo correctamente parseado.

TITULO: 056 Maturity of Performance Measurement Systems for Supply Chain Management

RESUMEN: The Supply chains have gained important focus after **2000?s** due to not only the opportunities that it can provide but also the complexity involved on its management. Into this context, the implementation of different management practices can help companies to achieve better performance in their supply chains. As part of these practices it can be considered the Performance Measurement Systems (PMSs). The theory for PMSs has been concentrated on only organization perspective. At the same time the theory about PMSs for Supply Chain Management (SCM) is focused more on scope of measurement leaving a lack with regards other dimensions that must be considered on the PMS maturity development. Therefore, this research aimed to answer the following research question which dimensions should be considered for the maturity management of PMSs for **SCM?**From the main findings obtained through a systematic literature review it was possible to identify eleven PMSs for SCM which shows focus only on measurement scope. Also, two maturity models for PMSs were found out showing more dimensions to be considered on the maturity management of PMSs beyond scope of measurement. As the main contribution from this research a theoretical model presenting the alignment between the maturity of PMSs and PMSs for SCM is proposed to help practitioners and researchers on the maturity management of PMSs for SCM.

AUTORES:

Frederico G
Martins R

EMAILS:

guilherme.frederico@ufpr.br

ram@dep.ufscar.br

PRIMERA PAGINA: 547

Ejemplo 3

En este ejemplo tenemos un error en el parseo del título y de los autores. El parseo identifica una sección de texto que contiene tanto el título como la lista de autores. Dentro de esta sección no aparece ninguna cadena de caracteres que pueda servir de referencia, por lo tanto se optó por suponer que el título ocupa dos líneas. En este caso, ocupa sólo una, por lo tanto, la primera línea de la lista de autores ha sido asignada erróneamente al título y solamente se han reconocidos los autores de la segunda línea.

TITULO: 117 Categorizing after learning from a database **Lozano J, Gómez A, Puente J, De la Fuente D**

RESUMEN: We start having data over 10 numerical variables and one symbolical category, which forms a record, in a CSV database, up to a total of 1100 records. Then, we first evaluate with crossvalidation the data, which takes in 10 steps 110records and check if it matches the symbolical real category from the inference of the other data. This forms an array of confusion in which the deviations from the diagonal are errors. If the training quality index, an error complementary coefficient, is under 0.9 (maximum 1) the data could contain potential for categorizing new data, but not in real world applications. After that, we could do training over the database, called A, to categorize another one, and called B. As a result, a decision tree constructed from A is applied to data B showing the probability of each category. In the test conducted, we used the module Algorithm Decision Tree version 2.2, in language Perl (www.cpan.org), and both the process of evaluation and the categorization takes few minutes each.

AUTORES:

Jesús Lozano (? e-mail: lozano@uniovi.es)

Alberto Gómez

Javier Puente

David de la Fuente Dpto. de Administración de Empresas. Escuela de Politécnica de Ingeniería de Gijón. Universidad de

EMAILS:

lozano@uniovi.es

PRIMERA PAGINA: 550

Ejemplo 4

Este ejemplo está correctamente parseado. No se ha podido parsear ninguna dirección de correo electrónico

porque el documento original no las incluía.

TITULO: 055 Modeling for Performance Evaluation of Beef Slaughterhouses

RESUMEN: The objective of this study was to propose a modeling to assess the performance of beef slaughter houses regarding its production costs, contributing to identify critical processes that adds value and increases competitiveness. The methodology comprehended, at first, a bibliographical research on accounting, costs management, strategic management and performance assessment, which pointed out that the **activitybased** costing method known as UP (Units of Production) and the concepts regarding to the Key Performance Indicators (**KPI?s**) could be considered as the most appropriated methods to construct the indicators of evaluation. The proposed modeling is composed by fourteen steps Definition of the production areas (1); Definition of the costs centers (2); Creation of the operative posts (3); Supervision distribution (4); Equipment cadastre (5); Determination of families of products (6); Market creation (7); Creation of groups of products (8); Cadastre of feedstock (9); Cadastre of products (10); Cadastre of productions (11); Allocation of expenses by cost centre (12); Allocation and division of indirect costs (13); and Allocation of expenses of indirect costs (14), returning in the end a structured tool to assess the industrial cost. In order to verify a practical application, we tested the modeling in a slaughterhouse located in Brazil, where the indicators Operational cost; operational cost per animal; operational cost/kg produced; **UP?s** produced; **UP?s** value; and **UP?s** produced per animal; were evaluated for six months. The results were compared to previous established goals, allowing the managers to measure the efforts necessary to manufacture each product and enabling faster and more efficient solutions to make decisions.

AUTORES:

Costa RP
Siluk JCM
Soliman M
Neuenfeldt Júnior AL
Machado CM
De Paris SR

EMAILS:

PRIMERA PAGINA: 546

Ejemplo 5

Ejemplo correctamente parseado. Este es un caso excepcional porque el título aparece en castellano y en inglés.

TITULO: 118 El análisis DAFO como herramienta estratégica de la Planificación Urbana **The SWOT analysis as strategic tool for Urban Planning**

RESUMEN: One main urban planning objective is to improve the quality of life in cities. The environmental circumstances affect the urban areas, conditioning the city and territory planning and management. Urban planning needs tools to understand the different planning phases and stages, and facilitate the further development and implementation of **plan?s** determinations. This paper focuses on the SWOT analysis as a useful tool to study the cities and urban planning status.

AUTORES:

Ros-McDonnell D
de la Fuente-Aragón MV
Ros-McDonnell L

EMAILS:

diego.ros@upct.es

marivi.fuente@upct.es

lorenzo.ros@upct.es

TAGS IDIOMA 2:

7.3.6 Resumen de resultados de la regla de parseo CIO2014_EXTENDED_ABSTRACTS

	Ejemplo 1	Ejemplo 2	Ejemplo 3	Ejemplo 4	Ejemplo 5
Título	**OK**	**OK**	**KO**	**OK**	**OK**
Resumen	**OK**	**OK**	**OK**	**OK**	**OK**
Autores	**OK**	**OK**	**KO**	**OK**	**OK**
Emails	**OK**	**OK**	**OK**	**OK**	**OK**
Primera página	**OK**	**OK**	**OK**	**OK**	**OK**

Tabla 7-5. Resumen de resultados de la regla de parseo CIO2014_EXTENDED_ABSTRACTS.

De los 25 datos localizados por la regla de parseo han fallado 2. Se han producido errores de parseo en un 8% de los datos.

8 CONCLUSIONES

La normativa relativa a Proyectos Fin de Carrera (PFC) de la **Escuela Técnica Superior de Ingeniería (ETSI)** de la **Universidad de Sevilla (US)** define la naturaleza del mismo como:

> *"un trabajo [...] que tiene como finalidad la aplicación por parte de aquel [cada alumno] de los conocimientos y de las habilidades adquiridas y de sus dotes de análisis y síntesis, para dar solución a un trabajo de corte igual o similar a los que pueda desarrollar en el ejercicio de su profesión como ingeniero."*

La aplicación web diseñada e implementada y toda la documentación que la acompaña encajan perfectamente en la definición anterior. Se ha solucionado un problema existente aplicando tanto conocimientos y habilidades previamente adquiridas como otras capacidades que ha sido necesario desarrollar a lo largo del Proyecto.

Concretamente destacaría como los conocimientos previos sobre programación simplificaron el proceso de aprendizaje de un framework completamente nuevo, como **SYMFONY**. Demostrando ser capaz de aprender a trabajar con nuevos lenguajes de programación rápidamente.

Si recordamos, el objetivo definido inicialmente para este Proyecto era:

> *Diseñar e implementar una herramienta que permita automatizar la clasificación y la publicación ordenada de los artículos de investigación generados en los **Congresos de Ingeniería de la Organización (CIOs)** organizados por **ADINGOR**.*

Efectivamente, la aplicación web diseñada e implementada también cumple con el objetivo definido inicialmente.

Para ayudarnos a evaluar la aplicación se analizarán los resultados obtenidos durante la fase de pruebas. Estos resultados se pueden resumir en la siguiente tabla:

Regla de parseo	Datos localizados	Errores	Porcentaje
CIO2013_ENGLISH_TRACKS	30	2	7%
CIO2013_SPANISH_TRACKS	45	16	36%
CIO2014_FULL_PAPER_ENGLISH	30	0	0%

CIO2014_FULL_PAPER_ESPAÑOL	45	0	0%
CIO2014_EXTENDED_ABSTRACTS	25	2	8%
TOTAL	**175**	**20**	**11,5%**

Tabla 8-1. Resumen de los resultados obtenidos.

De los 175 datos obtenidos automáticamente durante el parseo automático, 20 han sido erróneos. Es decir, se han producido errores de parseo en el 11,5% de los datos. Esto significa que casi el 90% de los datos se identifican correctamente, con el consiguiente beneficio que esto implica.

Además, conviene resaltar que si no se tienen en cuenta los resultados obtenidos con la regla de parseo CIO2013_SPANISH_TRACKS (donde motivos excepciones que ya se han comentado nos llevaron a obtener un número de errores muy elevado), el porcentaje de datos identificadores erróneamente sería menor al 4%.

Estos datos nos llevan a dos conclusiones: la primera es la confirmación del buen funcionamiento de la aplicación y, en concreto del parseo automático; y la segunda es la importancia de que los autores de los artículos tomen conciencia de las ventajas que aporta el correcto uso de las plantillas que se proporcionan en los Congresos.

A nivel personal, resulta muy gratificante comprobar que el tiempo, el esfuerzo y el sacrificio invertido en este Proyecto realmente han servido para simplificar una tarea que resultaba ardua y pesada para los administradores de la herramienta. Gracias al trabajo realizado se ha logrado mejorar algunos de los puntos débiles de la aplicación web original.

Además, esta satisfacción a la hacemos referencia nos motiva a seguir adelante, a no querer conformamos con la herramienta que tenemos. Queremos aportar soluciones a más problemas, avanzar. Por este motivo, en el último capítulo comentaremos posibles mejoras que pueden implementarse en el futuro.

9 FUTURAS MEJORAS Y AVANCES

Partimos del ambicioso objetivo que se había definido inicialmente y que gráficamente se correspondía con el diagrama de la Figura 9-1:

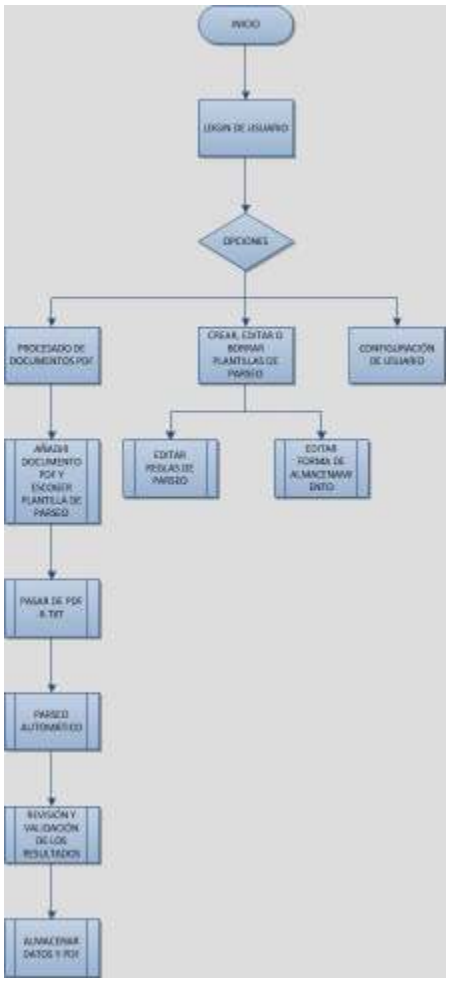


Figura 9-1. Diagrama del proceso genérico planteado inicialmente como solución.

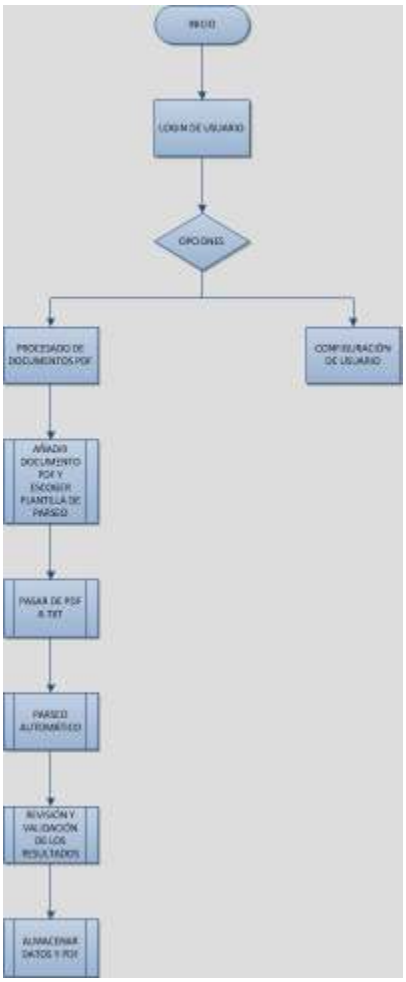


Figura 9-2. Diagrama del proceso genérico de la aplicación desarrollada.

Si lo comparamos con el diagrama de Figura 9-2, se observa que todas las funcionalidades relativas al procesado de nuevos documentos han sido implementadas. No ocurre lo mismo con las herramientas de

creación y modificación de reglas de parseo, que se siguen sin integrarse en la aplicación. Por este motivo, de cara al futuro, se debe trabajar para solucionar esta carencia.

Para terminar, también es importante plantear la actualización de la versión de **SYMFONY**. Ya se ha comentado que pasar de una versión 1.x a una 2.x del framework no es una tarea trivial. Posiblemente requiera la modificación de muchos de los módulos y bastante trabajo, pero es la única manera de evitar que "el esqueleto" sobre el que se construye la aplicación quede obsoleto.

BIBLIOGRAFÍA

Aparicio Ruiz, P., Guadix Martín, J., Onieva Giménez, L., & Arango Pastrana, C. (2012). Modernización de la difusión de los Congresos de Ingeniería de Organización. En *XVI Congreso de Ingeniería de Organización* (págs. 1452-1459). Vigo.

Barceló Llauger, M. (2001). *Hacia una economía del conocimiento.* ESIC Editorial.

Bowler, T., & Bancer, W. (2009). *Symfony 1.3 Web Application Development.* Packt Pub.

Porebski, B., Przystalski, K., & Nowak, L. (2011). *Building PHP Applications with Symfony, CakePHP, and Zend Framework.* Wiley.

Zaninotto, F., & Potencier, F. (2007). *The Definitive Guide to symfony.* Apress.

REFERENCIAS ONLINE

[1] *ADINGOR*. (s.f.). Obtenido de http://www.adingor.es/

[2] *LogicalDOC*. (s.f.). Obtenido de http://www.logicaldoc.com/es.html

[3] *Google Académico,*. (s.f.). Obtenido de https://scholar.google.es/

[4] *JSON*. (s.f.). Obtenido de http://json.org/

[5] *Symfony*. (s.f.). Obtenido de https://symfony.com/

[6] *Databases and Doctrine*. (s.f.). Obtenido de http://symfony.com/doc/current/book/doctrine.html

[7] *XAMPP*. (s.f.). Obtenido de https://www.apachefriends.org/es/index.html

[8] *Jobeet*. (s.f.). Obtenido de http://symfony.com/legacy/doc/jobeet?orm=Doctrine

[9] *XPDF*. (s.f.). Obtenido de http://www.foolabs.com/xpdf/

[10] *class.PDF2text.* (s.f.). Obtenido de http://webcheatsheet.com/php/reading_clean_text_from_pdf.php

[11] *PDFParser*. (s.f.). Obtenido de http://www.pdfparser.org/

[12] *DOI*. (s.f.). Obtenido de http://www.doi.org/

[13] *CSS*. (s.f.). Obtenido de http://www.w3.org/Style/CSS/Overview.en.html

[14] *PregMatchAll*. (s.f.). Obtenido de http://php.net/manual/es/function.preg-match-all.php

[15] *Expresiones Regulares*. (s.f.). Obtenido de http://php.net/manual/es/reference.pcre.pattern.syntax.php