UNIVERSIDAD NACIONAL DE LA PLATA

FACULTAD DE INFORMÁTICA

TESINA DE LICENCIATURA

Título: Ingesta asistida de contenidos en repositorios digitales - Un framework para DSpace
Autores: Carmine, Fernando Daniel
Director: Bibbo, Luis Mariano
Codirector: Fernández, Alejandro
Carrera: Licenciatura en Sistemas

Resumen

La proliferación de documentos digitales inició una nueva etapa en la organización de la información. En este contexto se tornó relevante el uso de soluciones tecnológicas que permitiesen a las instituciones implementar sus propios repositorios digitales. En este marco, el uso de los metadatos ofrece amplias posibilidades para representar de manera estandarizada la información digital. Los metadatos son la información descriptiva que se aplica sobre un recurso para facilitar su organización, recuperación y preservación. Su adopción en los repositorios digitales permitió dotarlos de interoperabilidad y son el fundamento de toda recuperación de la información.

Motivados por las potencialidades y usos de los metadatos, y ante la necesidad de contar con una herramienta de software integrada a un repositorio digital capaz de asistir en la ingesta de contenidos, automatizando la extracción de metadatos de calidad. Se pretende, por medio de la realización del presente trabajo, desarrollar una herramienta de software que tenga como principal funcionalidad la automatización de la extracción de metadatos de artículos científicos, potenciando la interoperabilidad entre los repositorios institucionales. A su vez se buscará mejorar la calidad de los metadatos creando estrategias de extracción, por medio de uso de algoritmos inteligentes, que trabajen directamente sobre los datos definidos por los autores de artículos científicos.

Palabras Claves

Extracción automática, repositorios digitales, DSpace, metadatos, Dublin Core.

Trabajos Realizados

Se estudió el estado del arte de los conceptos a abordar, analizando bibliografía, publicaciones científicas, revistas y demás producciones digitales disponibles en la Web.
Se desarrolló un framework que dispone de un conjunto de algoritmos capaces de extraer un grupo de metadatos de artículos científicos, entre los que se encuentran el título, autor, idioma, palabras claves y resumen.
Se modificó el workflow de carga de contenidos de DSpace con el fin de poder ofrecer, dentro de la herramienta, la opción de extracción automática de los metadatos.

Conclusiones

El presente trabajo concluye con el desarrollo de un framework capaz de proveerle a DSpace la extracción automática de un grupo de metadatos de artículos científicos.
Gracias a la modificación introducida en el workflow de carga de contenidos, a la extensión realizada en la definición de los campos de los formularios de carga y a la utilización de extractores automáticos de metadatos, se enriqueció y flexibilizo la experiencia de carga de artículos científicos por parte de los usuarios de la herramienta DSpace.

Trabajos Futuros

Extender las estrategias de extracción y ampliar el espectro de análisis con el fin de dar soporte a otros metadatos y formatos de estándares de publicaciones científicas.
Mejorar la estrategia de extracción de autores entrenando los modelos utilizados para la búsqueda.
Crear estrategias para recuperación de textos en otros formatos de archivo.
Integrar el framework desarrollado en un proceso de curación de metadatos.

Fecha de la presentación: Diciembre 2014

Contenido

Contenido I
Índice de ilustraciones IV
Índice de tablas VI
Listado de abreviaturas VII
Capítulo 1 - Introducción 1
1.1 Motivación 1
1.2 Objetivos 3
1.3 Aportes 3
1.4 Estructura de la tesina 3
Capítulo 2 - Estándares 5
2.1 Necesidad de estándares 5
2.2 Definición de estándares 6
2.2.1 IEEE LOM (Learning Object Metadata) 7
2.2.2 SCORM (Sharable content object reference model) 10
2.2.3 IMS LD (LEARNING DESIGN) 12
2.2.4 Dublin Core 15
Capítulo 3 - Metadatos 20
3.1 Definición de metadatos 20
3.2 Tipos de metadatos 20
3.2.1 Metadatos descriptivos 21
3.2.2 Metadatos estructurales 21
3.2.3 Metadatos administrativos 21
3.2.4 Otras clasificaciones de tipos de metadatos 21
3.3 Estructura de los Metadatos 21
3.4 Evolución de los Metadatos 22
3.4.1 Protocolos para la Recuperación de Información 23
3.4.2 Lenguajes de Metadatos 24
3.4 Conjunto de Metadatos Dublin Core 27
3.5 Funciones que desempeñan los metadatos 28
3.6 Problemática de la creación de metadatos 28
Capítulo 4 - Artículos científicos 29
4.1 Categorías de artículos 29
4.2 Estándares de los artículos 31
4.2.1 Artículos para TRANSACCIONES y PERIÓDICOS del IEEE 31
4.2.2 Estandar LNCS/LNAI – Springer 34
4.3 Secciones de los artículos 37
Capítulo 5 - Repositorios Digitales 39

5.1 Acceso abierto 40
5.2 Repositorio institucional de acceso abierto 41
5.2.1 Tipos de Acceso Abierto 43
5.3 Herramientas de código abierto 43
5.4 Caracterización DSpace 45
5.4.1 Características de DSpace 46
5.4.2 Modelo de desarrollo comunitario 46
5.4.3 Modelo de datos 47
5.4.4 Arquitectura de DSpace 47
5.4.5 Interfaces de DSpace 48
Capítulo 6 – Técnicas de recuperación de información 53
6.1 Minería de Texto 54
6.2 Extracción de Información 55
6.3 Procesamiento de lenguaje natural 55
6.4 Técnicas automáticas en la RI 57
Capítulo 7 - Herramientas para la extracción de metadatos 58
7.1 Introducción 58
7.2 Herramientas existentes 58
Capítulo 8 - Herramienta desarrollada 60
8.1 Metadatos de interés 60
8.2 Flujo de trabajo para la Extracción de metadatos 60
8.3 Arquitectura 61
8.3.1 Capa de Servicios 63
8.3.2 Capa de Modelo de Dominio 66
8.3.3 Capa Algorítmica 67
8.4 Integración con XMLUI 77
8.4.1 Arquitectura de alto nivel 77
8.4.2 Configuración del proyecto como dependencia 79
8.4.3 Configuración del camino de carga (submission step) 79
8.4.4 Configuración de los extractores por campos (fields) 81
8.4.5 Especificación de la etapa de extracción automática 81
8.4.6 Pantallas de carga en XMLUI 82
8.5 Consideraciones 86
8.5.1 Formato de los documentos soportados 86
8.5.2 Idioma de los documentos 86
8.5.3 Seguridad de los documentos 87
8.6 Pruebas realizadas 87
8.6.1 El corpus 88
8.6.2 Diseño de experimentos 88
8.6.3 Resultados del corpus completo 89

8.7 Comparación entre los metadatos obtenidos de forma manual y automática 90
8.7.1 Validación de autores 90
8.7.2 Validación de títulos 92
8.7.3 Validación de los idiomas 93
8.7.4 Validación de palabras clave 94
8.7.5 Validación del resumen 95
Capítulo 9 - Conclusión y trabajo futuro 105
9.1 Contribuciones 105
9.2 Experiencias realizadas 105
9.3 Trabajo futuro 106
Referencias 107

Índice de ilustraciones

Ilustración 1 Jerarquía completa de DCMI 16
Ilustración 2 Ejemplo de estructura de documento IEEE 31
Ilustración 3 Ejemplo de estructura de documento IEEE (continuación) 32
Ilustración 4 Definición de título, institución y nombre de autores de documento IEEE 32
Ilustración 5 Definición de resumen de documento IEEE 32
Ilustración 6 Definición de índice de términos de documento IEEE 33
Ilustración 7 Definición de introducción de documento IEEE 33
Ilustración 8 Definición de referencias de documento IEEE 33
Ilustración 9 Ejemplo de estructura de documento LNCS 34
Ilustración 10 Ejemplo de estructura de documento LNCS (continuación) 35
Ilustración 11 Definición de título de documento LNCS 35
Ilustración 12 Definición de institución y nombre de autores de documento LNCS 35
Ilustración 13 Definición de resumen de documento LNCS 36
Ilustración 14 Definición de palabras claves de documento LNCS 36
Ilustración 15 Definición de contacto de documento LNCS 36
Ilustración 16 Definición de referencias de documento LNCS 37
Ilustración 17 Software de Repositorios de Acceso Abierto usados en la Argentina 44
Ilustración 18 Software de Repositorios de Acceso Abierto usados en el mundo 44
Ilustración 19 Modelo de datos de DSpace 47
Ilustración 20 Arquitectura de DSpace 48
Ilustración 21 Workflow de carga de contenidos 49
Ilustración 22 Configuración de los campos para una página 50
Ilustración 23 Especificación de las etapas para carga de contenidos 52
Ilustración 24 interfaz JSPUI 52
Ilustración 25 Flujo de carga modificado 61
Ilustración 26 Arquitectura en capas de alto nivel 62
Ilustración 27 Interfaz de los servicios de los extractores 64
Ilustración 28 Implementación de los servicios de los extractores 65
Ilustración 29 Implementación de los servicios de los extractores (continuación) 66
Ilustración 30 Diagrama de clases del framework 67
Ilustración 31 Codificación de la estrategia del extractor de títulos 68
Ilustración 32 Codificación de la estrategia del extractor del resumen 70
Ilustración 33 Codificación de la estrategia del extractor del resumen (continuación) 71
Ilustración 34 Formula de probabilidad de características 71
Ilustración 35 Codificación del extractor para la detección de idiomas 72
Ilustración 36 Codificación del extractor de palabras clave 73
Ilustración 37 Codificación del extractor de palabras clave (continuación) 74
Ilustración 38 Codificación del extractor de autores 75
Ilustración 39 Codificación del extractor de texto 76
Ilustración 40 Arquitectura de alto nivel 78
Ilustración 41 Diagrama de interacción con el framework de extracción 78
Ilustración 42 Integración de dspace-extractor a XMLUI (pom.xml del proyecto XMLUI) 79

Ilustración 43 Definición del workflow de carga 80
Ilustración 44 Configuración del campo autor para la extración automática 81
Ilustración 45 Configuración de la etapa de extracción automática 81
Ilustración 46 Aceptación de licencia de DSpace 82
Ilustración 47 Selección de carga del archivo en DSpace 83
Ilustración 48 Visualización de los metadatos extraídos en DSpace 84
Ilustración 49 Visualización del ítem con sus metadatos completos en DSpace 85
Ilustración 50 Completitud de la carga en DSpace 86
Ilustración 51 Comparación de los resultados en términos de precisión y recuperación 90

Índice de tablas

Tabla 1 Categorías del esquema base de LOM versión 1.0 9
Tabla 2 Clasificación de los elementos DublinCore 16
Tabla 3 Elementos Dublin Core Simple 27
Tabla 4 Diferencias entre la recuperación de datos y la recuperación de información 53
Tabla 5 Resumen del método de evaluación 88
Tabla 6 Resultados obtenidos en la evaluación de los extractores 89
Tabla 7 Resultados de precisión y recuperación obtenidos en la evaluación de los extractores 89
Tabla 8 Comparativa de extracción manual y automática de autores 90
Tabla 9 Comparativa de extracción manual y automática de títulos 92
Tabla 10 Comparativa de extracción manual y automática de idiomas 93
Tabla 11 Comparativa de extracción manual y automática de palabras clave 94
Tabla 12 Comparativa de extracción manual y automática de resumen 95

Listado de abreviaturas

ADL	*Advanced Distributed Learning*
AICC	*Aviation Industry Computed Based-Training Comitee*
API	*Application Programming Interface*
ARIADNE	*Foundation for the European Knowledge Pool*
BOAI	*Budapest Open Access Initiative*
BSD	*Berkeley Software Distribution*
CCF	*The Comon Communication Format*
CICyT	*Consejo Interinstitucional de Ciencia y Tecnología*
CSS	*Cascading Style Sheets*
DC	*Dublin Core*
DCMI	*Dublin Core Metadata Initiative*
DRI	*Digital Repository Interface*
EAD	*Encoded Archival Description*
GML	*Generalized Markup Language*
HTML	*HyperText Markup Language*
IEEE	*Institute of Electrical and Electronics Engineers*
ISBD	*International Standard Bibliographic Description*
ISO	*International Organization for Standardization*
JSP	*JavaServer Pages*
KEA	*Automatic Keyphrase Extraction*
LD	*Learning Design*
LDAP	*Lightweight Directory Access Protocol*
LNCS	*Lecture Notes in Computer Science*
LOM	*Learning Object Metadata*
LTSC	*Learning Technology Standards Committee*
MARBI	*Machine-Readable Bibliographic Information*
MARC	*MAchine-Readable Cataloging*
NCSA	*National Center for Supercomputing Applications*
OAI	*Open Access Iniciative*
ORE	*Object Reuse and Exchange*
PMH	*Protocol for Metadata Harvesting*
OCLC	*Online Computer Library Center*
OIS	*Open Society Institute*
OpenDOAR	*Directory of Open Access Repositories*
RDF	*Resource Description Framework*
RSS	*Really Simple Syndication*
SCO	*Sharable Content Objects*
SCORM	*Sharable Content Object Reference Model*
SEDICI	*Servicio de Difusión de la Creación Intelectual*
SGML	*Standard Generalized Markup Language*
SNRD	*Sistema Nacional de Repositorios Digitales*
SOA	*Service Oriented Architecture*
TIC	*Tecnologías de la información y la comunicación*

UNESCO	*Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura*
USMARC	*United States MAchine-Readable Cataloging*
W3C	*World Wide Web Consortium*
WebDAV	*Web Distributed Authoring and Versioning*
XHTM	*eXtensible HyperText Markup Language*
XML	*Extensible Markup Language*
XSLT	*Extensible Stylesheet Language Transformations*

Agradecimientos

A mis padres Daniel y Mabel, mis hermanos Alejandro y Mauricio y a mis abuelos por estar todos estos años apoyándome como si fuera el primer día.

A mi sobrino Tiziano, que con su llegada trajo alegría a mi vida, y me dio motivos para terminar mi carrera.

A mis amigos, que me aguantaron todo este tiempo, en especial a Lautaro Maza, por decirme "vamos a La Plata a estudiar" y a Nicolás Alonso, que con sus consejos y opiniones me ayudaron a terminar la tesina.

A Luis Mariano Bibbo y Alejandro Fernández, por guiarme en el desarrollo de esta tesina.

A Leandro Antonelli, por los consejos y ayuda brindada durante la realización de esta tesina.

A mis compañeros de trabajo, que me soportaron todos estos años.

Son muchas las personas que han formado parte de mi vida a las que me encantaría agradecerles su amistad, consejos, apoyo, ánimo y compañía en los momentos más difíciles de mi vida. Algunas están aquí conmigo y otras en mis recuerdos y en mi corazón, sin importar en donde estén quiero darles las gracias por formar parte de mí, por todo lo que me han brindado.

Capítulo 1 - Introducción

La necesidad de albergar, preservar, difundir y dar visibilidad a la propia producción científica e intelectual es cada vez mayor en cualquier institución que realiza tareas de investigación, desarrollo y transferencia. Para alcanzar estas metas, muchas instituciones implementan repositorios institucionales centralizados y de acceso abierto.

Un repositorio tiene como propósito brindar un conjunto de servicios a una comunidad, destinados a recopilar, gestionar, difundir y preservar contenidos a través de una colección organizada y accesible en abierto que debe estar provista de facilidades que le permiten interoperar con otros repositorios similares.

Se considera repositorio digital a la colección de documentos digitales científicos y tecnológicos de una institución de investigación y desarrollo organizada de modo de permitir la búsqueda y recuperación de los mismos. Un repositorio digital debe, por ello, proveer mecanismos de identificación, almacenamiento, preservación, recuperación y exportación de documentos científicos y tecnológicos desde un portal libremente accesible desde Internet.

Dentro del contexto de la sociedad actual, las tecnologías de la información y la comunicación (TIC) se presentan cada vez más como una necesidad en donde los rápidos cambios, el aumento, demanda y actualización de la información y de nuevos conocimientos se convierten en una exigencia permanente.
Uno de los cambios más importantes derivado directamente de las TIC es el aumento de nuestra capacidad de generar, almacenar y gestionar datos en formato digital.

La proliferación de documentos digitales inició una nueva etapa en la organización de la información. En este contexto se tornaron relevantes la creación y uso de soluciones tecnológicas que permitiesen a las instituciones implementar sus propios repositorios digitales. Muchas propuestas para la codificación de la información digital surgieron para dar solución a las diferentes necesidades de las instituciones. En este marco, el uso de los metadatos ofrecieron amplias posibilidades para representar de manera estandarizada la información digital. Los metadatos son la información descriptiva que se aplica sobre un recurso para facilitar su organización, recuperación y preservación. Su adopción en los repositorios institucionales permitió dotarlos de interoperabilidad y son el fundamento de toda recuperación de la información.

1.1 Motivación

Las instituciones educativas y centros de investigación organizan anualmente distintos tipos de reuniones académicas y científicas. Estas reuniones abarcan congresos, simposios, conferencias, seminarios, presentaciones, etcétera. En ellas se busca preservar y difundir toda la creación intelectual generada a partir de los mismos, generalmente en forma presentaciones, ponencias, posters e informes, tradicionalmente conocidos como trabajos o artículos científicos (denominados *paper* por su anglicismo).

La forma de difundir la creación intelectual y dar a conocer los resultados de los trabajos científicos es por medio de las actas, las cuales son recopilaciones de dichos trabajos, editados, en general, por la entidad organizadora.

Entre las tareas llevadas a cabo por los organizadores de las reuniones están las de recopilar las versiones finales de los trabajos y generar las actas. Esta tarea puede ser realizada manualmente o por medio de la instrumentación de algún producto de software, la cual, en mayor o menor medida, implica una asistencia manual. La difusión de las actas puede implicar una posterior incorporación de las mismas a un repositorio institucional. Las tareas de recopilación de la producción intelectual de las reuniones, su incorporación al repositorio institucional, su correcta catalogación y difusión requieren desglosar manualmente cada uno de los trabajos de las actas ya generadas. Estas tareas pueden ser llevadas por distintos equipos de trabajo, por lo que en este sentido se hace necesario el establecimiento de un flujo de trabajo que permita, con el fin de asegurar la preservación de la producción intelectual, una mutua colaboración, minimizando el re-trabajo por parte de los integrantes de los equipos intervinientes.

En Argentina existe un Sistema Nacional de Repositorios Digitales (SNRD), **[1]** el cual es una iniciativa del Ministerio de Ciencia, Tecnología e Innovación Productiva conjuntamente con el Consejo Interinstitucional de Ciencia y Tecnología (CICyT) **[2]** a través de sus representantes en el Consejo Asesor de la Biblioteca Electrónica de Ciencia y Tecnología, quien promueven que "*Los repositorios digitales institucionales deberán ser compatibles con las normas de interoperabilidad adoptadas internacionalmente, y garantizarán el libre acceso a sus documentos y datos a través de Internet u otras tecnologías de información que resulten adecuadas a los efectos, facilitando las condiciones necesarias para la protección de los derechos de la institución y del autor sobre la producción científico-tecnológica*"

En este contexto y ante la ya mencionada proliferación de documentos digitales, el uso y adopción de metadatos fue lo que le permitió a los repositorios digitales dotarse de interoperabilidad.

Actualmente existen distintos estándares de metadatos tales como *DublinCore* **[3]** e *IEEE LOM* **[4]**, que utilizan distintas categorías no sólo para describir el contenido del recurso (título, autor, palabras claves, idioma, etc.) sino también permiten describir aspectos educacionales de los mismos (nivel educativo, complejidad, etc.). Sin embargo, en la mayoría de los casos, la calidad de la información cargada en estos metadatos en los distintos repositorios, suele ser insuficiente o incompleta. Esto se debe a que la carga de los metadatos es una tarea que suele ser tediosa e implica y un consume un tiempo considerable. Es en este punto donde se comienza a vislumbrar la necesidad de contar con una herramienta de software integrada a un repositorio digital, capaz de asistir en la ingesta de contenidos automatizando la extracción de metadatos de calidad **[5]**.

1.2 Objetivos

Motivados por las potencialidades y usos de los metadatos, y ante la necesidad de contar con una herramienta de software integrada a un repositorio digital capaz de asistir en la ingesta de contenidos, automatizando la extracción de metadatos de calidad. Se pretende, por medio de la realización del presente trabajo, desarrollar una herramienta de software que tenga como principal funcionalidad la automatización de la extracción de metadatos de artículos científicos, potenciando la interoperabilidad entre los repositorios institucionales. A su vez se buscará mejorar la calidad de los metadatos creando estrategias de extracción, por medio de uso de algoritmos inteligentes, que trabajen directamente sobre los datos definidos por los autores de artículos científicos.

Con el fin de transferir el uso de la herramienta, la misma será integrada y evaluada en uno de los software de repositorios digitales adheridos al SNRD, buscando producir una automatización en la carga de metadatos dentro del proceso de ingesta de contenidos. De esta forma, mediante una configuración en el repositorio, se podrá optar por el uso de la extracción automática sobre cada metadato, brindando flexibilidad en el uso de la herramienta.

1.3 Aportes

Considerando que la difusión de la creación intelectual y los resultados de los trabajos científicos, articulados con el uso de repositorios digitales, constituyen una fuente de conocimiento y de almacenamiento de datos para estudios y estadísticas vinculadas a la producción científica. Visto el valor de contar con una fuente de datos a partir de la cual obtener indicadores cualitativos y cuantitativos que faciliten la gestión institucional y la toma de decisiones en este rubro, así como el logro de una mayor visibilidad y difusión de la producción académica y científica. Y contemplando la problemática que persigue que los grandes volúmenes de información digitales almacenados en repositorios necesitan contar con metadatos precisos para poder ser localizados y recuperados conllevan al aporte brindado por la aplicación de la realización de este trabajo, la cual por medio del uso del *framework* desarrollado, se introducirán mejoras en la ingesta de contenidos almacenados en repositorios digitales, que estén basados en la herramienta *DSpace*, brindando una precisión en los metadatos y potenciando la interoperabilidad.

1.4 Estructura de la tesina

La tesina será organizada manteniendo la siguiente estructura:

Capítulo 1: Se describe el marco introductorio generando un contexto en el cual fue desarrollado el trabajo. Se plantea el tema central del trabajo con sus objetivos y aportes. Se detalla la organización de sus partes.

Capítulo 2: Se hace una introducción a la necesidad de los estándares sobre los contenidos digitales. Se define el concepto de estándar, y se describen algunos de ellos.

Capítulo 3: Se define el concepto de metadato, sus tipos y la estructura. Se menciona la evolución de los metadatos en la que se describen los protocolos para la recuperación de la información y los lenguajes de metadatos. Se define el estándar *Dublin Core* junto a sus elementos. Se explican las funciones que desempeñan los metadatos y se introduce a la problemática en la creación de los mismos.

Capítulo 4: Se introduce el concepto de artículos científicos describiendo sus categorías, estándares y las principales secciones del documento.

Capítulo 5: Se introduce la definición de repositorio digital, junto con las características que definen el acceso abierto a la información y los tipos de repositorios institucionales. Se detallan las herramientas de código abierto para luego realizar una descripción de la herramienta de software DSpace, utilizada como repositorio digital con el fin de entender su funcionamiento y aplicar el desarrollo de esta tesis.

Capítulo 6: En este capítulo se describirán los primeros intentos por tratar el contenido de los documentos; hablamos entonces de la definición y técnica de Recuperación de Información, en especial la Minería de Texto, la Extracción de Información y Texto, el procesamiento de lenguaje natural, debido a que son las tecnologías afines a la recuperación de Información que están más relacionadas con nuestro problema. Y para finalizar se introduce el concepto de técnicas automáticas en la recuperación de la información.

Capítulo 7: Se describen algunas herramientas existentes para la extracción de metadatos introduciendo la problemática que presentan y la elección de desarrollar estrategias sin hacer uso de estas herramientas.

Capítulo 8: Se especifica la herramienta desarrollada, desde la elección de los metadatos a extraer, el nuevo flujo en la carga de contenidos necesario para aplicar las estrategias de extracción y el desarrollo de la arquitectura en capas. Partiendo de la arquitectura propuesta se describen las configuraciones que fueron necesarios para la utilización del *framework* desde *DSpace*. Se enumeran las consideraciones necesarias para poder acotar el problema de la extracción de metadatos. Se detallan las pruebas realizadas para la validación de la herramienta.

Capítulo 9: Se realiza la conclusión en base a los objetivos planteados y se detallan posibles trabajos futuros sobre el desarrollo.

Capítulo 2 - Estándares

2.1 Necesidad de estándares

Hasta hace relativamente poco tiempo, la historia de la humanidad, estaba confinada a las grandes bibliotecas, tanto académicas como públicas, dentro de estantes, donde se colocan los documentos y un número variable de copias, sean libros, revistas informes, periódicos u otros. La creciente producción de información exige un espacio cada vez mayor para su almacenamiento. Asimismo, se requiere de una difusión casi instantánea de la información como resultado del llamado proceso de globalización o internacionalización del conocimiento **[6]**. Probablemente, los factores que más han incidido en el tránsito de las publicaciones hacia nuevos soportes son la necesidad de una distribución inmediata de la información y el conocimiento, los crecientes costos de las ediciones impresas y el papel, la flexibilidad, accesibilidad y economía de los medios más modernos, así como la falta de espacio para su almacenamiento.

La tendencia actual de las publicaciones es la sustitución del formato impreso por el medio electrónico, un proceso acelerado, que comenzó en la década de los años 1990, acompañado por el desarrollo y la masificación de la Internet. Estos avances produjeron un incremento notable de la difusión del conocimiento, con formas muchas veces incontrolables, sin una estructura informática diseñada específicamente para estos fines. Este proceso se acompañó de una proliferación de documentos digitales que inició una nueva etapa en la organización de la información.

Hoy en día tenemos a nuestro alcance la biblioteca más grande del mundo, gran parte de la música, las películas, los libros, las fotografías y casi todo lo que ha producido el hombre en el último siglo. Ahora, ¿es esta información homogénea?, la respuesta es no. Esta información es de lo más heterogénea, esto responde claro a que es de los tipos más variados (incluyendo libros, música, películas, herramientas de software, etc.), pero también responde a que las herramientas que se utilizaron para crearla o distribuirla son también muy diferentes entre sí. Gracias a éstas, una persona que no posea grandes conocimientos técnicos puede desarrollar contenidos que pueden ser distribuidos electrónicamente. Sin embargo este amplio rango de herramientas, provistas por diferentes fabricantes, hace que tales recursos no compartan un mecanismo común respecto a la forma en la que serán posteriormente encontrados y utilizados. Si estos recursos no comparten un mecanismo de adquisición y uso común, no podrán ser compartidos entre diferentes repositorios de contenidos digitales. Es aquí donde se hace necesario y evidente el uso de estándares.

Según se expresa en una publicación de la Universidad de Alcalá, "*el principal objetivo de un estándar es el establecimiento de un lenguaje común que permita la colaboración en un determinado ámbito de la actividad humana. Los estándares han sido uno de los pilares del progreso de todos los campos de la industria, haciendo posible, por ejemplo, que Internet sea hoy una realidad, y el propio término "Internet" hace referencia a un estándar*" **[7]**.

Cuando hablamos de estándares orientados a contenidos digitales, estamos hablando de un conjunto de reglas aplicables al desarrollo de tecnología. Estas reglas permiten, entre otras cosas, la reutilización del contenido y su interoperabilidad. Este concepto se refiere a la capacidad de diferentes sistemas informáticos, aplicaciones y servicios para comunicar, compartir e intercambiar datos, información y conocimiento de una forma precisa, efectiva y consistente; para funcionar de forma correcta con otros sistemas, aplicaciones y servicios, así como para integrarse con otros sistemas, aplicaciones y servicios, y ofrecer nuevos productos electrónicos.

Los estándares tecnológicos permiten diferentes tipos de interoperabilidad constituyen una mayor dimensión con más aproximaciones tradicionales engranando hacia la interoperabilidad de metadatos en los repositorios digitales. Hoy en día contamos con una buena cantidad de estándares que abarcan diversos aspectos como accesibilidad, calidad, descripción de los recursos de aprendizaje, arquitectura, etc. A continuación se describen brevemente aquellos estándares que han abordado uno de los principales problemas que es la descripción, distribución y uso de los contenidos digitales orientados a la producción y aprendizaje.

2.2 Definición de estándares

La estandarización de los contenidos digitales toma sus definiciones de los conceptos de la creación de recursos u objetos de aprendizaje. Estos buscan en primer término establecer relaciones y competencias específicas de cada parte, de modo de lograr interoperabilidad, reusabilidad, manejabilidad, flexibilidad, accesibilidad, durabilidad y escalabilidad.

Uno de los estándares propuestos como primer abordaje al problema de la heterogeneidad de la información y la dificultad que supone su descubrimiento y aprovechamiento es IEEE LOM (*Learning Object Metadata*). IEEE LOM es un estándar versátil y flexible, que permite la descripción de los recursos de aprendizaje a través de la especificación de sus metadatos.

Si vamos un paso más allá, el solo hecho de poder describir adecuadamente los recursos de aprendizaje resulta insuficiente. Piense que para enseñar un concepto determinado se utilizan distintos recursos (como pueden ser libros y actividades), también se necesitará poder distribuirlos y, al mismo tiempo, seguramente existirá algún orden determinado para su abordaje que es necesario dejar plasmado. Por ejemplo, no se puede comenzar con una actividad sin antes haber adquirido los conceptos teóricos que se presentaron en el libro; además un recurso puede ser útil para más de un propósito. Con esta cuestión en mente surge la especificación SCORM **[8]** (*Sharable Content Object Reference Model*, o Modelo de Referencia de Objetos de Contenido Compartible) que, como su nombre lo indica, constituye un modelo de referencia y abarca el empaquetamiento y la distribución de los recursos de aprendizaje.

Ahora, poder describir y distribuir los recursos de aprendizajes es algo muy bueno, pero también resulta insuficiente. Según la corriente de pensamiento constructivista, la comprensión y los conceptos se construyen en la mente del que aprende. Es decir, se sabe que los aprendices construyen sus propios conocimientos, pero se habla y actúa como si fueran los educadores quienes entregan los conocimientos a sus estudiantes.

En la educación presencial las intervenciones de los educadores pueden resolver esta contradicción. A través de un sinfín de intervenciones didácticas, muchas de las cuales son espontáneas e indocumentadas, ayudan a los aprendices a construir su propia comprensión. Esto lleva a la idea de presentar al conocimiento como un objeto y es lo que permite que, por ejemplo, un libro se pueda reutilizar durante varios cursos para enseñar una determinada materia a muchos estudiantes. Pero en el ámbito de la educación no tradicional, en cambio, estas intervenciones suelen desaparecer, ya que la infraestructura para la relación profesor-alumno es generalmente pobre, o, incluso, inexistente. Así, se puede decir que los recursos tienen un carácter educativo en virtud de su uso por alumnos en actividades educativas, y no por sus cualidades internas. Es por esto que nace el estándar IMS LD (*IMS Learning Design*) **[9]**, aún en fases de desarrollo. Este estándar permite, a través de la definición de un lenguaje genérico, expresar distintas pedagogías.

En concordancia con los estándares presentados y agrupando la funcionalidad de los mismos, surge uno de los estándares más utilizados y reconocidos que DublinCore. Este constituye un mecanismo básico de descripción que puede usarse en todos los dominios, para todo tipo de recursos, sencillo pero potente, que puede extenderse fácilmente y puede trabajar conjuntamente con otras soluciones específicas.

Se describirán cada uno de ellos, pero antes es necesario desarrollar un concepto que es fundamental para su correcta interpretación y entendimiento, que es el concepto de "metadato".

2.2.1 IEEE LOM (Learning Object Metadata)

2.2.1.1 Orígenes

Las bases para este estándar fueron sentadas por dos importantes organizaciones, por un lado el "Consorcio de Aprendizaje Global IMS" (IMS Global Consortium) y por otro la fundación ARIADNE (ARIADNE "*Foundation for the European Knowledge Pool*"). Estas dos organizaciones presentan en el año 1998 una propuesta conjunta ante el "*Comité de Estándares de Tecnología de Aprendizaje del IEEE*" (*IEEE Learning Technology Standards Committee*, o simplemente *LTSC*), consistente de una especificación que más tarde se convertiría en el borrador para el desarrollo del estándar mismo.

El proyecto IMS fue iniciado por la organización sin fines de lucro EDUCOM, hoy EDUCASE. Con este proyecto se intentó desarrollar estándares abiertos de mercado para aprendizaje online. Pero no puede dejar de mencionarse otra importante organización que también contribuyó a la creación del estándar actual, como es el Instituto Nacional de Estándares y Tecnología estadounidense (NIST), que en el año 1997 comienza desarrollos similares al proyecto IMS y contribuye con este último.
Así, el Comité de Estándares y Tecnología de Aprendizaje del IEEE, desarrolla el estándar para Metadatos de Objetos de Aprendizaje ("*IEEE Learning Object Metadata*" o IEEE LOM por sus siglas en inglés). Se trata de un estándar multi-partes: la primera de ellas consiste en un esquema conceptual de datos; y las partes dos, tres y cuatro, corresponden a formas de encapsulamiento de ese esquema a través de ISO/IEC 11404, XML y RDF respectivamente.

En junio del año 2002, el IEEE aprueba el esquema conceptual de datos de LOM como IEEE 1484.12.1-2002 "*Estándar para Metadatos de Objetos de Aprendizaje*" y el modelo de encapsulamiento en XML para dicho esquema como IEEE 1484.12.3 "*Estándar para Lenguaje de Marcado Extensible*", en el año 2005.

2.2.1.2 El estándar

En la primera parte del estándar, cuya publicación está identificada como IEEE 1484.12.1-2002, se especifica el esquema conceptual de datos que define la estructura de una instancia de metadatos para un objeto de aprendizaje, pero no define como un sistema de tecnología de aprendizaje representa o usa dicha instancia de metadatos.

Para este estándar, un objeto de aprendizaje se define como cualquier entidad (digital o no) que puede ser usada para aprendizaje, educación o entrenamiento.
Una instancia de metadatos para un objeto de aprendizaje describe características relevantes del objeto de aprendizaje al cual aplica. Dichas características se pueden resumir en: "general", "ciclo de vida", "meta-metadatos", "educativo" (o "pedagógico"), "técnicos", "derechos", "relación", "anotación" y "categorías de clasificación".

Según se especifica en el estándar, el esquema conceptual de datos propuesto permite la diversidad lingüística tanto de los objetos de aprendizaje como de las instancias de metadatos que los describen.

El propósito del estándar IEEE LOM es facilitar la búsqueda, evaluación, adquisición y uso de objetos de aprendizaje para instructores, aprendices o procesos de software automatizados. También busca facilitar el intercambio y compartimiento de objetos de aprendizaje, a través del desarrollo de catálogos e inventarios, al mismo tiempo que tiene en cuenta la diversidad de contextos culturales y lingüísticos en los cuales se reutilizan los objetos de aprendizaje y sus metadatos.

2.2.1.3 Esquema conceptual de datos

El esquema conceptual de datos descripto en el estándar de IEEE, es un esquema jerárquico. Este esquema sigue la típica estructura de árbol, donde un elemento que contiene otros elementos "hijos" es llamado "rama" y un elemento que no contiene otros elementos es una "hoja".

La estructura y multiplicidad de cada elemento está definida en la citada primera parte del estándar (IEEE 1484.12.1-2002), por lo que solo se dará aquí una breve descripción de cada uno de los elementos principales del esquema, sugiriéndose consultar dicha publicación para mayores detalles.

2.2.1.4 Estructura básica de los metadatos

Los elementos de datos que describen un objeto de aprendizaje son agrupados en categorías. El esquema base de LOM en su versión 1.0 define nueve categorías que se describen sucintamente en la tabla presentada a continuación:

Tabla 1 Categorías del esquema base de LOM versión 1.0

Categoría	Sub-elementos
General: agrupa la información general que describe el objeto de aprendizaje como un todo. Tag: <general>	1. Identificador a. Catálogo b. Entrada 2. Título 3. Idioma 4. Descripción 5. Palabra clave 6. Cobertura 7. Estructura 8. Nivel de agregación
Ciclo de vida: agrupa las características relacionadas con la historia y el estado actual del objeto de aprendizaje y aquellos quienes lo han afectado durante su evolución. Tag: <lifeCicle>	1. Versión 2. Estado 3. Contribución a. Rol b. Entidad c. Fecha
Meta-metadatos: agrupa información respecto de la instancia de metadatos en sí misma (describe los metadatos que describen al objeto de aprendizaje). Tag: <metaMetadata>	1. Identificador a. Catálogo b. Entrada 2. Contribución a. Rol b. Entidad c. Fecha 3. Esquema
Técnico: agrupa los requerimientos y características técnicas del objeto de aprendizaje. Tag: <technical>	1. Formato 2. Tamaño 3. Ubicación 4. Requisito a. Composición i. Tipo ii. Nombre iii. Versión mínima iv. Versión máxima 5. Observaciones de instalación 6. Otros requisitos de plataforma 7. Duración
Educativo: agrupa las características pedagógicas y educativas claves del objeto de aprendizaje. Tag: <educational>	1. Tipo de interactividad 2. Tipo de recurso de aprendizaje 3. Nivel de interactividad 4. Densidad semántica 5. Rol de usuario final al que está destinado. 6. Contexto 7. Rango etario típico 8. Dificultad 9. Tiempo de aprendizaje típico 10. Descripción 11. Idioma
Derechos: agrupa los derechos de propiedad intelectual y condiciones de uso del objeto de aprendizaje. Tag: <rights>	1. Costo 2. Derechos de copia y otras restricciones 3. Descripción

Relación: agrupa características que definen la relación entre el objeto de aprendizaje y otros objetos de aprendizaje. Tag: <relation>	1. Clase 2. Recurso a. Identificador i. Catálogo ii. Entrada iii. Descripción
Anotación: provee observaciones respecto del uso educativo del objeto de aprendizaje así como también información respecto de quién y cuándo creó cada observación. Tag: <annotation>	1. Entidad 2. Fecha 3. Descripción
Clasificación: describe el objeto de aprendizaje de acuerdo a un sistema particular de clasificación. Tag: <classification>	1. Propósito 2. Camino taxonómico a. Origen b. Taxón i. Identificador ii. Entrada 3. Descripción 4. Palabra clave

Tal como se dijo anteriormente, tener un mecanismo común para describir apropiadamente un recurso es algo muy bueno pero, dependiendo del recurso y de su complejidad, y sumado al hecho de que un recurso puede ser utilizado para más de un propósito, surge la necesidad de poder distribuirlos apropiadamente, presentar un ordenamiento determinado para su utilización, etc., así surge la iniciativa SCORM, la cual se presenta a continuación.

2.2.2 SCORM (Sharable content object reference model)

2.2.2.1 Orígenes

En 1997 el Departamento de Defensa de los EEUU y la Oficina de Políticas de Ciencias y Tecnología (OSTP) de la Casa Blanca impulsan la "Iniciativa de Aprendizaje Distribuido" (*Advanced Distributed Learning Initiative* ó ADL por sus siglas en inglés). Esta iniciativa está dirigida a establecer un nuevo entorno de aprendizaje distribuido que permita la interoperabilidad de herramientas de aprendizaje y de contenidos en una escala global.

ADL notó que la industria ya tenía muchos estándares que intentaban alcanzar esas metas (o partes de ellas), por lo que creó un modelo de referencia denominado "*Sharable Content Object Reference Model*" (SCORM) o "Modelo de Referencia de Objetos de Contenido Compartible" que, básicamente, referencia esos estándares existentes y le dice a los desarrolladores como usarlos juntos apropiadamente.

2.2.2.2 El estándar

SCORM promueve la creación de contenidos de aprendizaje reutilizables (denominados "objetos instruccionales") dentro un *framework* técnico común para el aprendizaje basado en computadoras y la Web.

Se trata de un "modelo de referencia" que reúne estándares y especificaciones para el empaquetamiento y secuenciamiento de contenidos de aprendizaje reutilizables y compartibles. Las fuentes primarias para SCORM son especificaciones de organizaciones tales como el "Comité de Entrenamiento Basado en Computadoras de la Industria de Aviación" ("*Aviation Industry Computed Based-Training Comitee*" o AICC), el "Consorcio de Aprendizaje Global IMS" y los estándares del "*Comité de Estándares de Tecnología de Aprendizaje*" (LTSC) del IEEE. Esos estándares y especificaciones referenciados están organizados como "libros" que componen la especificación, así SCORM sería una "biblioteca" que contiene esos libros. Específicamente, esos libros son cuatro: "Perspectiva General", "Entorno de Tiempo de Ejecución", "Modelo de Agregación de Contenidos" y "Secuenciamiento y Navegación". El primero define al modelo como un todo y los tres siguientes describen capacidades y características específicas con detalle técnico.

2.2.2.3 Libro "Modelo de agregación de contenidos" (CAM BOOK)

Este libro describe los componentes usados en la experiencia de aprendizaje, como empaquetarlos para poder intercambiarlos entre sistemas, como describirlos para permitir su localización y como definir la información de navegación para esos componentes.

El CAM está compuesto entonces por:

- Modelo de Contenidos
- Metadatos
- Empaquetamiento de Contenidos

Que podríamos considerarlos como "secciones" del libro CAM.

Modelo de Contenidos

El "Asset" o recurso es el formato básico de los recursos de aprendizaje. Un Asset puede ser texto, imagen, audio, video, animación flash, archivo pdf o cualquier otra forma de recurso digital. Varios Assets pueden ser agrupados constituyendo un nuevo Asset.

Los "Objetos de Contenido Compartible" ("*Sharable Content Objects*" o SCOs) son una colección de uno o más Assets. A diferencia de un Asset, un SCO se puede comunicar con el LMS usando el Entorno de Ejecución (que será ampliado más adelante). Un SCO debe tener un medio para localizar la instancia de API provista por el LMS y debe incluir, como mínimo, los métodos "LMSInitialize('' '')" y "LMSTerminate('' '')". Para lograr la reusabilidad, interoperabilidad y durabilidad buscada, los SCOs deben ser independientes de su contexto de aprendizaje.

Metadatos

Los metadatos consisten en información adicional añadida a un recurso para describir sus atributos de una forma común. SCORM recomienda fuertemente el uso del estándar IEEE LOM para describir los contenidos de los componentes del módulo.

Empaquetamiento de Contenidos

Esta sección define un mecanismo común para empaquetar Assets y SCOs. Los paquetes de contenidos pueden ser intercambiados entre distintos LMSs o herramientas. Dentro de cada paquete se encuentra un archivo XML especial, llamado archivo de manifiesto (imsmanifest.xml). El manifiesto describe no solo forma en la que se encuentran ensamblados los recursos sino también la forma en la que son presentados.

2.2.2.4 Libro "Entorno de tiempo de ejecución" (O RTE BOOK)

Este libro define un mecanismo para ejecutar objetos de contenido común, define además un mecanismo de comunicación entre éstos y los LMSs y, por último, un modelo de datos común para el seguimiento de la experiencia de aprendizaje con estos objetos.

2.2.2.5 Libro "secuenciamiento y navegación"

Este libro define reglas para controlar el flujo lógico, o secuencia, de objetos de contenido compartible en una experiencia de aprendizaje.

Hasta aquí se introdujeron dos mecanismos, uno que permite describir los recursos de aprendizaje a través de sus metadatos (IEEE LOM) y otro que define formas de empaquetar, distribuir y navegar dichos recursos (SCORM). La mayoría de los pensadores sostiene que la importancia o la relevancia que tiene un recurso determinado para el aprendizaje viene dada por el uso que se haga del mismo, con un enfoque pedagógico determinado, más que por sus cualidades internas. Así surge IMS LD, como un nuevo mecanismo que intenta reflejar distintas pedagogías aplicadas al uso de los recursos.

2.2.3 IMS LD (LEARNING DESIGN)

2.2.3.1 Orígenes

La iniciativa "Diseño de Aprendizaje" de IMS (IMS Learning Design, ó simplemente IMS LD) remonta sus orígenes al año 1997. En ese año la Universidad Abierta de Holanda decide transformar todos sus cursos en cursos online. A través de esta experiencia identificó que los diferentes cursos hacían uso de variados enfoques pedagógicos y, al mismo tiempo, cada profesor tenía su propia visión pedagógica. Por esta razón, si se quería confeccionar una plantilla representativa de cada estilo pedagógico, se iban a necesitar tantas plantillas como docentes tenía la universidad, lo cual sin dudas resultaba inviable. Sin embargo, se determinó inteligentemente que, sin importar el enfoque pedagógico del cual se tratase, en la práctica todos consistían en combinaciones de tres elementos básicos: recursos educativos, múltiples personas actuando en varios roles, y actividades pedagógicas.

Esta experiencia dio lugar al denominado "Lenguaje de Modelado Educativo" (*Educational Modelling Language*), que tuvo lugar tras la comparación y el examen exhaustivo de un amplio rango de enfoques pedagógicos y de sus actividades de aprendizaje asociadas, y tras varias iteraciones del lenguaje desarrollado para obtener un buen balance entre generalidad y expresividad pedagógica conformando un meta-lenguaje relativamente conciso.

El enfoque del meta-lenguaje desarrollado tiene la gran ventaja de que, en lugar de intentar capturar la terminología de cada enfoque pedagógico, lo que implicaría lidiar con un vocabulario infinitamente grande, o un conjunto de vocabularios, se usa un vocabulario relativamente pequeño para expresar lo que, en términos concretos, cada uno de los enfoques solicita de los aprendices y del personal de apoyo involucrado.

De esta manera el lenguaje, al permitir un buen grado de expresividad en lugar de ser prescriptivo de una pedagogía determinada, alienta el desarrollo de nuevas pedagogías.

2.2.3.2 El estándar

IMS Learning Design representa una integración entre el "Lenguaje de Modelado Educativo" y especificaciones existentes de IMS.

Como se señaló, los recursos adquieren carácter educativo cuando los alumnos los emplean para aprender; y se puede afirmar que IMS LD posibilita que se pueda convertir tales recursos en objetos reutilizables. Ya no se considera que hay conocimientos en el recurso que se pueden transmitir a un estudiante receptor, sino que los recursos son un elemento esencial, pero no suficiente, en un proceso más amplio y complejo de aprendizaje. El recurso solamente puede ser usado para aprendizaje cuando se combina con actividades realizadas por personas en roles, es decir (en términos de IMS LD), cuando forma parte de una unidad de aprendizaje (UoL).

2.2.3.3 Unidad de aprendizaje (UNIT OF LEARNING – UOL)

IMS define a la "Unidad de Aprendizaje" como un término abstracto usado para referir a cualquier pieza delimitada de educación o entrenamiento, como puede ser un curso, un módulo, una lección, etc. Al mismo tiempo aclara: representa más que solo una colección de recursos ordenados para aprender, incluye una variedad de actividades prescriptas (como actividades de resolución de problemas, actividades de investigación, actividades de discusión, etc.), evaluaciones, servicios y apoyo provistos por maestros, entrenadores o miembros del staff. Tales actividades, recursos, roles y flujo de trabajo dependen del "diseño de aprendizaje" que es parte integral de la unidad de aprendizaje.

IMS define "diseño de aprendizaje" como la descripción de un método que posibilita a los aprendices alcanzar ciertos objetivos de aprendizaje realizando determinadas actividades de aprendizaje en un orden estipulado en el contexto de un cierto ambiente de aprendizaje. Se podría resumir diferentes personas actuando en diferentes roles con determinados recursos y en el marco de una actividad pedagógica dada.

La UoL puede tener muchas formas distintas incluso con el mismo recurso. Por ejemplo, un texto que describa una ciudad puede formar parte de una UoL tratando de historia, o de literatura, o de lengua, o de formación de educadores, entre otros temas, y en cada contexto tendría actividades y/o roles distintos. Las diferentes UoLs serían reutilizables por distintos profesores, con distintas cohortes de alumnos.

Como se puede observar, una UoL es un elemento mucho más complejo que un objeto de SCORM, y puede ofrecer oportunidades de aprendizaje mucho más ricas.

2.2.3.4 Objetivo de la especificación de IMS LD

El objetivo de la especificación IMS LD es proveer un *Framework* de contención de los elementos que describen cualquier diseño de un proceso de enseñanza-aprendizaje de una manera formal. Más específicamente, IMS LD reúne los siguientes requisitos:

1. Completitud: la especificación debe ser capaz de describir completamente el proceso de enseñanza-aprendizaje en una unidad de aprendizaje, incluyendo referencias a los objetos de aprendizaje y servicios, digitales y no digitales, necesarios durante el proceso. Esto incluye:

   - Integración de las actividades, tanto de aprendices como de instructores.
   - Integración de los recursos y servicios usados durante el aprendizaje.
   - Soporte de una amplia variedad de enfoques para aprendizaje.
   - Soporte para modelos de aprendizaje de usuarios simples o múltiples.
   - Soporte de modo mixto (aprendizaje semi-presencial) así como también de aprendizaje online puro.

2. Flexibilidad pedagógica: la especificación debe ser capaz de expresar la funcionalidad y el significado de diferentes elementos de datos dentro del contexto de una unidad de aprendizaje. Debe ser además flexible en la descripción de todas las diferentes clases de pedagogías y no obligar a adoptar ningún enfoque pedagógico específico.

3. Personalización: la especificación debe ser capaz de describir los aspectos de personalización dentro de un diseño de aprendizaje, de manera tal que el contenido y las actividades dentro de una unidad de aprendizaje puedan ser adaptadas de acuerdo a las preferencias, conocimientos previos, necesidades educativas, y circunstancias actuales de los usuarios.

4. Formalización: la especificación debe describir un diseño de aprendizaje en el contexto de una unidad de aprendizaje de manera formal para que el procesamiento automático sea posible.

5. Reproducibilidad: la especificación debe describir el diseño de aprendizaje abstraído de manera tal que su ejecución bajo distintas configuraciones y con diferentes personas sea posible.

6. Interoperabilidad: la especificación debe soportar interoperabilidad de diseños de aprendizaje.

7. Compatibilidad: la especificación usa los estándares y especificaciones disponibles donde sea posible, principalmente IMS Content Packaging, IMS Question and Test Interoperability, IMS/LOM Meta-Data e IMS Simple Sequencing.

8. Reusabilidad: la especificación debe hacer posible identificar, aislar, de-contextualizar e intercambiar artefactos de aprendizaje útiles y reutilizarlos en otros contextos.

2.2.4 Dublin Core

2.2.4.1 Orígenes

Creado en 1995 por iniciativas de las asociaciones de bibliotecarios americanos, y patrocinado por la OCLC (*On Line Computer Library Center*), tiene su origen en un círculo intelectual de Dublin, en el estado de Ohio en Estados Unidos. La primera reunión para tratar aspectos relacionados con el Dublin Core la convocó la OCLC y el NCSA (*National Center for Supercomputing Applications*); en ella participaron 52 investigadores expertos en el campo de la bibliotecología, ciencias de la computación, codificadores de textos y áreas afines, con el objetivo de impulsar el desarrollo de los registros descriptivos de recursos de información en línea.

Muchas son las personalidades e instituciones que se han interesado y han participado en el desarrollo de este formato. Su progreso ha ocurrido aparejado al desarrollo del XML y del RDF; en octubre del 2001, se logró convertir el conjunto de elementos del vocabulario de Dublin (DCMES, *Dublin Core Metadata Element Set*) en un estándar formal, ANSI/NISO Z39.852001.1

2.2.4.2 El estándar

El estándar DCMI [DCMI, 2010], cuenta con un conjunto de 15 definiciones semánticas que permiten la descripción y organización de la información, así como también la definición de las propiedades de objetos para sistemas que se encarguen de la búsqueda de recursos basados en la Web. Los 15 elementos que componen el estándar son: contribuidor, cobertura, creador, fecha, descripción, formato, identificador, lenguaje, editor, relación, derechos, fuente, tema, título y tipo (ver Figura 1). A su vez, estos se agrupan en 3 grandes categorías: **contenido**, **propiedad intelectual** e **instanciación**.

Este sistema de definiciones fue diseñado específicamente para proporcionar un vocabulario de características "base", capaces de proporcionar la información descriptiva básica sobre cualquier recurso, sin que importe el formato de origen, el área de especialización o el origen cultural.

Este estándar de metadatos no está restringido a un perfil de aplicación específico, y es altamente usado en el mundo en diferentes disciplinas de estudio. Muchos repositorios lo han adoptado para etiquetar sus recursos de material educativo (por ejemplo, SEDICI, Rephip, Corciencia, Universidad Nacional de Colombia, Universidad Javeriana). Así mismo, DCMI puede ser utilizado sobre cualquier sistema de información y, a su vez, permite que dicho sistema sea interoperable con otros sistemas de información que ofrezcan sus contenidos según las etiquetas.

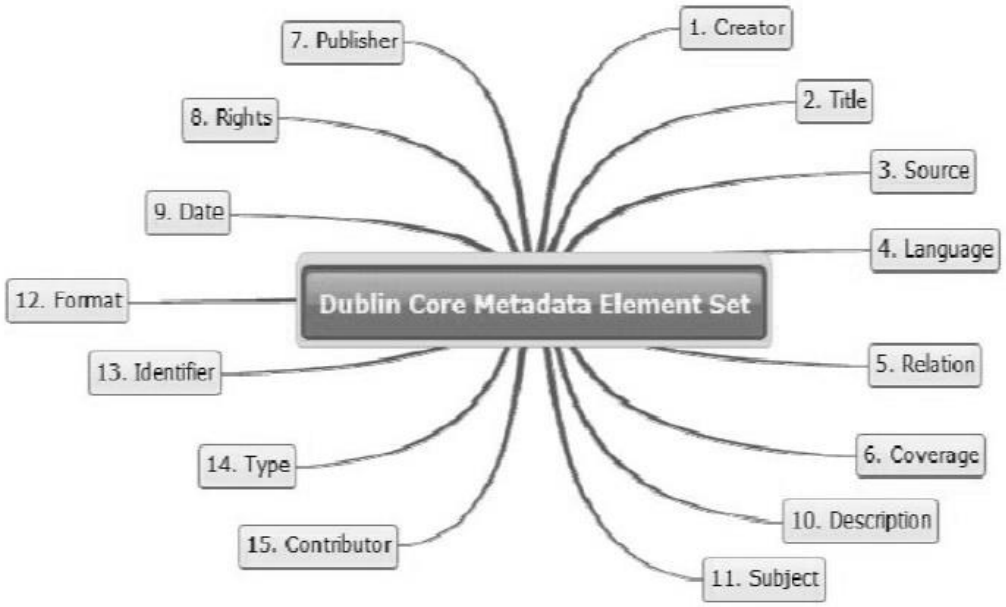


Ilustración 1 Jerarquía completa de DCMI

2.2.4.3 Estructura de los metadatos

En general, podemos clasificar estos elementos en tres grupos que indican la clase o el ámbito de la información que se guarda en ellos:

- Elementos relacionados principalmente con el contenido del recurso.
- Elementos relacionados principalmente con el recurso cuando es visto como una propiedad intelectual.
- Elementos relacionados principalmente con la instanciación del recurso.

Dentro de cada clasificación encontramos los siguientes elementos [24]:

Tabla 2 Clasificación de los elementos DublinCore

Etiqueta: DC. Source, Fuente	Referencia al recurso del que deriva el documento actual. El recurso actual puede derivar de un recurso Fuente en todo o en parte. La práctica mejor recomendada en este caso es identificar el recurso referenciado por medio de un string o número conforme con un sistema de identificación formal. Ejemplo: "Imagen de pag. 54 de la edición 1922 de Romeo and Julieta"
Etiqueta: DC. Lenguaje, Lengua	El idioma del contenido del recurso. Se recomienda usar la RFC 3066 [RFC3066] que, en conjunción con la norma ISO639 [ISO639]), define dos y tres etiquetas de idioma principal con subetiquetas opcionales. Ejemplo: se incluye "en" o "eng" para el inglés.

Etiqueta: DC. Relation, Relación	Una referencia a un recurso relacionado con el contenido. Se recomienda identificar los recursos referenciados mediante un string [conjunto de caracteres manipulados como un grupo] o un número conforme a un sistema de identificación formal. Ejemplo: Una relación de versiones Título="AACR2 Electrónica" Relación="Reglas de Catalogación Angloamericanas, 2da edición"
Etiqueta: DC. Coverage, Cobertura	Refiere a la magnitud o el alcance del contenido de un recurso. Puede tratarse de una especificación geográfica, temporal o legal, es decir, la cobertura incluirá la localización espacial (un nombre de un lugar o unas coordenadas geográficas), período temporal (una expresión que identifica un período, fecha o rango de fecha) o jurisdicción (por ejemplo una denominación de una entidad administrativa). Ejemplo: "1995-1996" ; “Siglo XVII” ; “Buenos Aires, AR”
Etiqueta: DC. Creador, Autor	Responsable de la creación del contenido. Puede ser una entidad, una persona o un servicio. Ejemplo: "Shakespeare, William" ; “Universidad Nacional de Misiones”
Etiqueta: DC. Publisher, Editor	Responsable de que el recurso se encuentre disponible. Una persona, una organización, o un servicio. Normalmente, el nombre de un editor debe utilizarse para indicar la entidad. Ejemplo: “Universidad Nacional de Misiones”
Etiqueta: DC. Contributor, Colaborador	Responsable de hacer colaboraciones al contenido del recurso. Una persona, una organización o un servicio. Normalmente el nombre de un colaborador debe utilizarse para indicar una entidad. Ejemplo: “Alvarado, Julián, tr.”
Etiqueta: DC. Rights, Derechos	Información sobre los derechos de la propiedad intelectual del recurso como por ejemplo el copyright. Normalmente, los derechos contendrán una declaración de gestión de derechos para el recurso, o referenciarán un servicio que proporcione dicha información. La información sobre los derechos normalmente abarca los derechos de Propiedad Intelectual (IPR), Copyright, y varios derechos relacionados con la propiedad. Si no consta los elementos de Derechos, no se deben hacer asunciones sobre ningún derecho contenido en el recurso o entorno a él. Ejemplo: Acceso limitado a integrantes del equipo" ; "http://cs-tr.cs.cornell.edu/Dienst/Repository/2.0/Terms& quot ; “Todos los derechos reservados, UNAM”
Etiqueta: DC. Date, Fecha	Fecha asociada a la creación o modificación del recurso. Se suele seguir la notación AAAA-MM-DD Ejemplo: “2007-12-19”

Etiqueta: DC. Type, El tipo o categoría del contenido	Palabras claves de un vocabulario que describen la naturaleza del recurso. Se refiere a términos que describen categorías generales, funciones, géneros o niveles de agregación del contenido. La práctica recomendada en este sentido, es seleccionar un valor de un vocabulario controlado (por ejemplo, del Vocabulario de Tipos de la DCMI [DCT1]). Para describir la manifestación física o digital de un recurso, se usa el elemento FORMAT. Ejemplo: “Imagen” ; “Sonido” ; “Texto” ; “Software” ; “Recurso interactivo”
Etiqueta: DC. Format, Formato	Descripción física del recurso, como su tamaño, duración, dimensiones, etc. si son aplicables. El formato puede usarse para identificar el Software, hardware, u otros equipamientos necesarios para visualizar/presentar u operar el recurso. Se recomienda seleccionar el valor de un vocabulario controlado (por ejemplo, la lista de Tipos de Medios en Internet [MIME] que define los formatos de medios informáticos). Ejemplo: Titulo="Icono Dublin Core" Identificador="http://purl.org/metadata/dublin_core/images/dc2.gif& quot; Tipo="Image" Formato="image/gif" Formato="4 kB"
Etiqueta: DC. Identifier, Identificación	Referencia unívoca para el contenido del recurso. Identificar el recurso por medio de un string [serie de caracteres manipulados como un grupo] o número adaptado a un sistema formal de identificación. Algunos formatos de identificación formal de recursos son, entre otros, el Identificador Uniforme de Recursos (URI) (que incluye el Localizador Uniforme de Recursos (URL)), el Identificador de Objetos Digitales (DOI) y el Número Internacional Normalizado de Libros (ISBN). Ejemplo: "ISBN:0385424728"; “http://purl.org/metadata/dublin_core/images/dc2.gif&quot”;

2.2.4.4 Características del formato

Entre las principales características de este formato pueden citarse **[10]**:

Alcance internacional: se ha traducido a más de 20 idiomas, esto es posible como resultado de la participación en el proyecto de representantes de prácticamente todos los continentes, que ha garantizado que el formato considere la naturaleza multilingüe y multicultural del universo de la información que alberga hoy Internet.

Simplicidad: es un simple, pero eficaz conjunto de elementos descriptivos, pensado, desde su inicio, para su uso, tanto por profesionales como por cualquier autor que desee describir su recurso con el objetivo de hacerla más visible. Todos los elementos del formato son opcionales y repetibles.

Flexibilidad: todos los elementos son opcionales y repetibles, lo que permite a cada autor poder escoger los elementos del formatos que considere necesarios en correspondencia con las necesidades descriptivas de sus recursos de información. La disposición de los elementos puede tener cualquier orden.

Interoperabilidad semántica: establece vínculos y relaciones con otras normas, sin sacrificar su autonomía.

Extensibilidad: los creadores del formato *Dublin Core* han creado mecanismos que permiten ampliar el conjunto de sus elementos, y esto permite que las diferentes comunidades que utilizan o desean utilizar el formato puedan formular y fundamentar propuestas de agregación de modificaciones y nuevos elementos al formato, según una necesidad descriptiva concreta.

Capítulo 3 - Metadatos

3.1 Definición de metadatos

Aunque el término metadato se relacionó inicialmente con el campo de la bibliotecología, actualmente se ha extendido a los recursos digitales. Fue acuñado por Jack Myers en la década de los año 60 (Caplan, 1995) **[11]**, para describir conjuntos de datos. La primera acepción que se le dio (y actualmente la más extendida) fue la de dato sobre el dato, ya que su intención era proporcionar la información mínima necesaria para identificar un recurso. Teniendo en cuenta esta definición y el contexto actual, se puede considerar el alcance de la catalogación como un proceso de generación de metadatos, que convoca no sólo a profesionales de la información, sino también, a informáticos, diseñadores de programas, técnicos de sistemas, etc.

El avance y desarrollo de las tecnologías de la información y las comunicaciones ha provocado cambios revolucionarios en la tarea bibliotecaria, la transformación y evolución asociada a Internet produjo dentro de las instituciones gestoras de información un cambio de paradigma en relación con la selección, procesamiento y recuperación de la información, dado que en Internet, los recursos de información están al alcance de todos, siempre y cuando seamos capaces de desarrollar mecanismos que permitan su localización.

En este contexto el uso normalizado de metadatos se presenta como una solución que permite una estructura adecuada para la descripción normalizada de documentos digitales y que posibiliten la localización y recuperación selectiva de la información en los entornos digitales y la red Internet **[12]**.

Los metadatos son datos secundarios debidamente estructurados, correspondientes a: autor, título, palabras clave, resumen, fecha, y/u otros y que sirven para describir los recursos de información con el objetivo de ayudar en su identificación y ulterior localización, tanto por parte de las personas como de las computadoras.

3.2 Tipos de metadatos

Los tipos y funciones de metadatos existentes son múltiples, dependiendo de diversos factores: el tipo de información que describen, el nivel de estructuración de esta información, el lugar donde se encuentren los metadatos, su ámbito de aplicación, el tipo de usuarios que los utiliza y sus finalidades, entre otros.
A los fines prácticos, los tipos y funciones de los metadatos pueden clasificarse en tres amplias categorías -descriptivos, estructurales y administrativos- con límites no siempre bien definidos y a veces superpuesto. Por ejemplo, los metadatos administrativos pueden incluir una amplia gama de información que podría considerarse como metadatos descriptivos y estructurales **[13]** (Tim Berners-Lee, 2000).

3.2.1 Metadatos descriptivos

Los metadatos descriptivos son aquellos que sirven para la descripción e identificación de los recursos de información, permiten la búsqueda y recuperación de la información, como también distinguir un recurso de otro y entender el asunto o contenido del mismo. Se realizan mediante los estándares como Dublin Core; MARC; Meta tags, HTML, etc.

3.2.2 Metadatos estructurales

Estos tipos de metadatos son los que más influyen en la recuperación de la información electrónica, facilitan la navegación y presentación de los recursos electrónicos. Así, ofrecen la información sobre la estructura interna de los recursos, estableciendo las relaciones entre ellos, de manera que pueden incluso unir los archivos de imagen y textos que están relacionados. Los estándares más difundidos para ellos son SGML y XML/RDF; EAD (*Encoded Archival Description* - Descripción de Archivo Codificado para Archivos).

3.2.3 Metadatos administrativos

Los metadatos administrativos son de carácter más técnico porque incluyen datos sobre la creación y control de calidad, datos sobre la gestión de derechos, requisitos del control de acceso y utilización, información sobre la preservación y permiten la gestión a largo y corto plazo. Ejemplo de estos metadatos son: tipo y modelo de escáner utilizado, resolución, limitaciones de reproducción, etc.

3.2.4 Otras clasificaciones de tipos de metadatos

Además de esta clasificación por tipos y funciones, también se pueden analizar sus características según otros atributos, como por ejemplo:

- el método de creación de los metadatos que puede ser manual (una persona) o automático (una máquina),
- la estructura que puede ser simple o estructurada,
- la semántica que puede seguir un vocabulario libre o controlado, etc.

3.3 Estructura de los Metadatos

Típicamente, los elementos que conforman un metadato están definidos por algún estándar o perfil, donde los usuarios que deseen compartir metadatos están de acuerdo con el significado preciso de cada elemento. Conforme al nivel de información que brinden sobre un conjunto de datos documentado, los metadatos pueden ser mínimos o detallados:

- Los metadatos mínimos sólo se restringen a los componentes más importantes e involucra las siguientes secciones:
    - Identificación: Información básica sobre el conjunto de datos (título, autoría, propósito, resumen, temática, localización etc.).
    - Calidad: Evaluación general de la calidad de un conjunto de datos.
    - Distribución: Datos del distribuidor y medios para obtener el conjunto de datos.

- En el detallado, además de las secciones arriba mencionadas, se componen con otras secciones como:
    - Entidades y atributos: Información sobre los objetos involucrados y sus atributos.
    - Referencia del metadato: Actualidad de la información del metadato y de sus responsables.
    - Citación: Datos de soporte sobre las referencias citadas dentro del conjunto de datos.
    - Contacto: Información de soporte sobre personas y organizaciones asociadas al conjunto de datos.

3.4 Evolución de los Metadatos

En el área de la organización de la información, el uso de metadatos, como parte de la identificación y clasificación es muy anterior a la era de la informática. Los primeros catálogos de libros impresos correspondían a listas ordenadas alfabéticamente sin criterios de clasificación sofisticados. Un avance importante en cuanto a esquemas de clasificación se desarrolla alrededor del año 1900, cuando los catálogos de libros son reemplazados completamente por tarjetas, de las cuales una de sus propiedades es que pueden ser actualizadas. En la década del sesenta, con el surgimiento de la tecnología, se facilitaron los proyectos de organización documental automatizada. La automatización se aplicó con mayor frecuencia a la organización de la información bibliográfica y esto dio como resultado las primeras bases de datos bibliográficas. Los métodos de producción en masa, hicieron necesario disponer de múltiples copias de los catálogos existentes. Es allí cuando surgen masivas colecciones distribuidas de libros y los catálogos de tarjetas no logran satisfacer los nuevos requerimientos. Fue necesario entonces, desarrollar estándares de codificación, llamados hoy en día metadatos.

Los primeros metadatos digitales y sus bases se desarrollan a finales del siglo XX, cuando emergen, como se dijo anteriormente, múltiples estándares de codificación, así como también variados lenguajes y protocolos, los cuales se utilizan en la generación y uso de catálogos.

La automatización aplicada a las bibliotecas, la generación de programas y las posibilidades de cooperación bibliotecaria basada en la automatización provocaron el interés de organizaciones como la UNESCO, IFLA, FID e ISO, por generar la sistematización de la información bibliográfica orientada al uso de normas en el marco internacional y en los avances de tecnologías de información y telecomunicaciones. El resultado de dicho interés se relaciona con los formatos MARC, USMARC, UNIMARC y CCF, las normas ISO aplicadas a la documentación, las Reglas de Catalogación Angloamericanas (2da. edición) y las ISBD (*International Standard Bibliographic Description*).

La función de los formatos bibliográficos, como los señalados anteriormente, han sido un soporte metodológico en la representación estructural en ambiente automatizado de registros bibliográficos, para su intercambio entre unidades de información, y en la orientación del diseño de las bases de datos bibliográficas. **[14]**

En las últimas décadas, el desarrollo tecnológico trajo aparejada la posibilidad de presentar y transmitir textos electrónicos a través de redes de telecomunicación, o de incorporar al texto electrónico imagen, sonido y movimiento, lo que se le conoce hoy en día como hipertexto. A partir de este avance, aparecen nuevas formas para la representación de la información electrónica. Surgieron entonces los "formatos digitales", como HTML (*HyperText Markup Language*), SGML (*Standard Generalized Markup Language*) y XML (*Extensible Markup Language*), lenguajes que proponen diferentes sintaxis en la que puede ser representada la información de carácter electrónico. **[15]**

A continuación, se describen dos de los primeros protocolos para la recuperación de información: el protocolo *Machine Readable Cataloguing* (MARC) y el Z39.50. Luego, se exponen diferentes lenguajes involucrados en la evolución del marcado de metadatos en la era digital.

3.4.1 Protocolos para la Recuperación de Información

3.4.1.1 Machine Readable Cataloguing (MARC)

El formato MARC **[16]** es un conjunto de normas que permite almacenar información en registros de cualquier tipo, para posteriormente, poder tratarla, localizarla, intercambiarla o ponerla a disposición del usuario. Fue desarrollado para ayudar a las bibliotecas en el uso, desarrollo y mantenimiento de sus bases de datos, y precisamente dicho desarrollo ha hecho realidad la catalogación compartida y la automatización de bibliotecas.

La Biblioteca del Congreso de los E.E.U.U. inició a finales de los años cincuenta la investigación para desarrollar un formato legible por "máquina" para los registros bibliográficos. Otras bibliotecas comenzaron a cooperar en este proyecto que incluía el desarrollo y uso del MARC I. La Biblioteca del Congreso (*Library of Congress*) distribuía registros MARC I a los miembros del proyecto y las bibliotecas trataban estos registros en sus ordenadores locales.

El proyecto piloto MARC I aportó información para establecer una norma para registros legibles por máquina. El formato MARC I fue refinado y extendido a partir del año 1967 en el formato conocido hoy como MARC II, concebido para intercambio de datos, capaz de almacenar información bibliográfica sobre toda clase de materiales.

Este nuevo formato, es un formato de comunicaciones, cuyo uso principal es permitir que distintas bibliotecas y organizaciones, independientemente de sus sistemas, puedan transmitirse registros entre ellas para ser usados en un sistema automatizado. La estructura del formato se aceptó por la Organización Internacional de Normalización convirtiéndose en norma ISO 2709. Esta norma, junto a la norma ANSI Z39.2, norma nacional americana, definen la estructura del formato MARC.

Desde sus inicios, MARC ha sido sometido a continuos perfeccionamientos para adaptarse a las necesidades de las bibliotecas y sus usuarios. La Biblioteca del Congreso coordina la investigación y proyectos sobre MARC, sirve como última autoridad sobre estos formatos y es la editora de la documentación de USMARC (hoy en día MARC 21). Otras organizaciones con gran influencia en el desarrollo del MARC son el Comité de la American Library Association, "MARBI", y el MARC Advisory Comité.

3.4.1.2 Z39.50

"Z39.50" es el nombre de un estándar definido por ANSI/NISO, basado en la estructura cliente/servidor, que permite comunicar sistemas que funcionan en distinto hardware y usan distinto software. Fue diseñado para solucionar los problemas asociados a la búsqueda en múltiples bases de datos con diferentes lenguajes y procedimientos. **[17]** En este sentido, el protocolo permite tanto la realización de búsquedas simultáneas a múltiples bases de datos, utilizando una única interfaz de usuario, así como también recuperar la información, ordenarla, y exportar los registros bibliográficos **[18]**.

Con el Z39.50 el proceso de consulta de la información es más sencillo y ágil, por eso también son más fluidas otras funciones y servicios habituales en las bibliotecas y centros de documentación, como los trabajos de referencia e información bibliográfica, puesto que una misma interfaz puede ser utilizada con diferentes motores de búsqueda y bases de datos.

También se facilitan la catalogación cooperativa, ya que el protocolo permite la descarga a menudo gratuita de registros MARC de distintas fuentes, y el préstamo interbibliotecario de un documento, solicitado a partir de los datos de ejemplares suministrados por un servidor Z. **[19]**

La primera versión del estándar se liberó en el año 1988. Dos años después se formaron dos grupos de trabajo que garantizaron el desarrollo y evolución de la norma: un grupo de implementadores ZIG (Z39.50 Implementors Group) y una agencia para el soporte del estándar (Z39.50 Maintenance Agency), fruto de su trabajo se aprueba la versión 2 en 1992 que, además de numerosas mejoras, evita las incompatibilidades con el protocolo de ISO "*Search and Retrieve*" SR (ISO 10162 y 10163).

En el año 1995 se aprobó la versión 3, que fue aceptada como estándar de la ISO (ISO 23950) en marzo del año 1997. Las nuevas facilidades que se han ido añadiendo tienen un carácter modular, y pueden irse implementando progresivamente de forma independiente. Los sistemas de gestión documental, evolucionan gradualmente hacia la versión 3, manteniendo además la compatibilidad con versiones anteriores del estándar. **[20]**

En la actualidad, Z39.50 es un estándar maduro, con una amplia presencia en la comunidad bibliotecaria, y se lo puede considerar como la norma más importante para el mundo de las bibliotecas y la documentación desde la aparición del formato MARC.

3.4.2 Lenguajes de Metadatos

Para que los metadatos se materialicen es necesaria la existencia de lenguajes que permitan especificar las sintaxis en la que se definen las estructuras, además de proveer medios para las especificaciones semánticas necesarias (que especifiquen lo que las expresiones sintácticas significan en términos de un modelo). Estos modelos y sintaxis son los que permiten representar las expresiones, hechos, reglas y consultas sobre las descripciones.

A continuación se detallarán los lenguajes HTML (*HyperText Markup Language*), SGML (*Standard Generalized Markup Language*) y XML (*Extensible Markup Language*) los cuales proponen diferentes sintaxis en la que puede ser representada la información de carácter electrónico.

3.4.2.1 Standard Generalized Markup Language (SGML)

Es un lenguaje estándar generalizado para marcado de documentos, que unifica la aplicación de los conceptos de anotación estructural. Sus raíces se remontan a 1969 cuando en los laboratorios de IBM se desarrolló el *Generalized Markup Language* (GML), lenguaje que fue evolucionando hasta 1974 donde pasó a llamarse SGML. La *International Organization for Standardization* (ISO) aprobó y publicó dicho lenguaje en el año 1984 con el nombre de estándar ISO 8879.

SGML no es un formato de almacenamiento ni un procesador de texto. Por el contrario, se trata de un metalenguaje con el que se pueden definir lenguajes de anotación que permitan almacenar y procesar texto.
Este estándar internacional consta de un conjunto de reglas para describir la estructura de un documento de tal forma que puedan ser intercambiados a través de diferentes plataformas. SGML es extremadamente flexible y es la base de los lenguajes de marcado más utilizados hoy en día.

En SGML un documento está definido en función de la estructura de las entidades que lo conforman. Estas entidades se organizan en una estructura lógica de manera jerarquizada determinando la estructura de los elementos del documento. Las entidades pueden ser compartidas por distintos documentos. El marcado se lleva a cabo mediante delimitadores y etiquetas las cuales pueden estar anidadas y se representan mediante el conjunto de caracteres básicos de acuerdo al estándar ISO 8879.

En el contexto histórico de los metadatos, la introducción de SGML jugó un papel fundamental, pues estableció un nuevo paradigma, en que los datos dejan de ser sólo datos. Los documentos SGML contienen separadamente (en el sentido lógico) los contenidos, la estructura y el formato.

3.4.2.2 HyperText Markup Language (HTML)

El hipertexto es un método de organización de la información en el cual los diferentes elementos se enlazan a través de otros elementos del propio texto. En pocas palabras, hipertexto significa texto almacenado en forma electrónica con vínculos de referencias cruzadas entre páginas, y tiene como característica que en lugar de leer un texto siguiendo una estructura rígida y lineal (como un libro), es posible avanzar de un punto a otro fácilmente, y desplazarse (navegar) por el texto.

El HTML es un “lenguaje de marcas”, que permite construir documentos hipertexto, es decir, se añaden marcas a los documentos que definen la presentación gráfica de los mismos y los enlaces entre sus páginas. Los recursos de la presentación incluyen resaltados, separación de párrafos, negrita, subrayado, etc.

Este lenguaje se basa en la teoría de que todos los documentos tienen ciertos elementos en común como son los títulos, los párrafos, las listas y las ilustraciones.

Los documentos HTML no son más que documentos de texto con una serie de etiquetas, las cuales le son de utilidad al navegador para interpretar la forma en que tiene que ser representado el texto, las imágenes o los sonidos en la pantalla.

3.4.2.3 eXtensible Markup Language

A medida que el número de materiales disponible en soporte digital aumentaba, también se hacían mayores las dificultades para acceder a los mismos. Para solucionar este problema, se comenzó a trabajar a favor de la normalización de formatos, con el propósito de diseñar un lenguaje de marcas optimizado uniendo la simplicidad de HTML con la capacidad expresiva de SGML.

Tal normalización llevó al surgimiento de XML, siglas en inglés de *eXtensible Markup Language* (lenguaje de marcas extensible), un metalenguaje extensible de etiquetas desarrollado por el *World Wide Web Consortium* (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

Dentro de los objetivos que persigue XML podemos nombrar la necesidad de distinguir el contenido y la estructura de los documentos de su presentación en papel o en pantalla; de hacer explícita su estructura y sus contenidos informativos; y crear documentos portables, que puedan intercambiarse y procesarse con facilidad en sistemas informáticos heterogéneos.

Para lograr estos objetivos XML propone un formato de documentos en texto plano (evitando las complejidades de los documentos binarios) e intercalar marcas con el objetivo de distinguir las distintas partes o elementos estructurales que conforman cada tipo de documento.

Una vez definidos los objetivos de XML podemos decir que la funcionalidad de XML consiste en representar y distribuir tanto documentos como información textual; intercambiar datos e información estructurada a través de Internet y la World Wide Web; integrar datos procedentes de fuentes heterogéneas; y eliminar la barrera entre información estructurada e información textual.

Presenta las siguientes ventajas:

- Es extensible, lo que quiere decir que una vez diseñado un lenguaje y puesto en producción, es posible extenderlo con la adición de nuevas etiquetas de manera de que los usuarios de la vieja versión todavía puedan entender el nuevo formato.
- El analizador es un componente estándar, por lo que no es necesario crear un analizador específico para cada lenguaje. De esta manera se evitan bugs y se acelera el desarrollo de la aplicación.
- Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarlo. Esto mejora la compatibilidad entre aplicaciones.

3.4 Conjunto de Metadatos Dublin Core

Ante el advenimiento de la información digital, el estándar de Metadatos Dublin Core (DCMI) se ha convertido en un simple pero eficaz conjunto de elementos que sirven para describir una amplia gama de recursos de Internet. Actualmente es la iniciativa de catalogación más extendida en el mundo electrónico, al tiempo que es considerada un estándar internacional (ISO-15836-2003).

Existe una serie de características propias de este estándar, las cuales pueden resumirse en:

- Simplicidad: Puede ser utilizado tanto por bibliotecarios como por cualquier autor que desee describir sus documentos y aumentar su visibilidad.
- Interoperatibilidad Semántica: Contiene un conjunto de descriptores que permiten la unificación con otros estándares de datos.
- Reconocimiento Internacional.
- Extensibilidad: Permite la elaboración de descripciones de modelos tales como el MARC completo. Cuenta también con suficiente flexibilidad y extensibilidad para limitar la estructura, además de una semántica más elaborada y un amplio estándar de descripción.
- Flexibilidad: Nada es obligatorio, todos los elementos son opcionales y repetibles, así el usuario elige la profundidad de una descripción.

La norma Dublin Core (DC) promueve dos niveles de codificación: simple y cualificado. El Dublin Core simple comprende quince elementos; el Dublin Core cualificado implica el mismo número de elementos más un subgrupo de éstos denominados cualificadores, que refinan la semántica de los primeros a fin de recuperar y localizar de mejor modo los recursos en Internet.

Cada elemento del conjunto es opcional y repetible, y pueden clasificarse en tres tipos: los que tienen que ver con el contenido del recurso, los referentes a la propiedad intelectual y los relacionados con la creación e identidad del material, tal como aparece en la tabla a continuación.

Tabla 3 Elementos Dublin Core Simple

<table>
<tr><th colspan="2">Elementos Dublin Core Simple</th></tr>
<tr><td rowspan="2">Contenido
Título
Tema
Descripción
Fuente
Lengua
Relación
Cobertura</td><td>Propiedad intelectual
Creador
Editor o editorial
Colaborador
Derechos</td></tr>
<tr><td>Creación e identidad
Fecha
Tipo
Formato
Identificador</td></tr>
</table>

Comúnmente los repositorios institucionales utilizan el esquema de metadatos DC para describir el contenido de sus objetos, estándar que se ha generalizado en la medida que se ha vuelto indispensable para cumplir el protocolo OAI-PMH, el cual se describirá más adelante, dado que promueve la interoperabilidad entre repositorios estructurados, debido a que es un formato aceptado globalmente.

Lo interesante de la Iniciativa de Metadatos Dublin Core es que permite establecer formas normalizadas para matizar cada uno de sus elementos a partir del uso y promoción de esquemas de codificación y vocabularios. Sin embargo, DC sigue presentando cierta ambigüedad al momento de codificar información en elementos como Título, Creador, Colaborador y Editor, que curiosamente no presentan ningún esquema que ayude a la codificación y asignación de los metadatos.

3.5 Funciones que desempeñan los metadatos

Una vez expuesto el concepto de metadato y los estándares más relevantes, se destacan varias razones que resaltan la importancia de los sistemas de metadatos:

- Incrementan el acceso: la existencia de un conjunto de metadatos que describa correctamente uno o varios Objetos, aumenta la posibilidad de acceder a ellos. Además, los metadatos hacen posible la búsqueda de información en múltiples bancos a la vez. Con una única ecuación de búsqueda, es posible consultar bases de datos que utilicen diferentes sistemas de metadatos para describir sus Objetos.
- Expandir el uso de la información: los metadatos facilitan la difusión de versiones digitales de un único Objeto.
- Control de versiones: aplica no sólo en lo que se refiere a gestionar la vida de un Objeto, sino también en lo que tiene que ver con su difusión. Es decir, se generan diferentes metadatos con distintas cantidades o tipos de información sobre un mismo Objeto, con el fin de distribuirlo a un público heterogéneo.
- Precisión en los procesos de búsqueda y recuperación: la correspondencia entre los descriptores usados en la búsqueda y los metadatos del Objeto, permite aumentar la precisión en la mayoría de búsquedas en Internet.

3.6 Problemática de la creación de metadatos

Muchos de los metadatos estructurales y administrativos básicos los provee el personal técnico que se encarga de la digitalización o creación de un objeto digital, o son creados a través de un proceso automatizado. En cambio, algunas categorías de metadatos, como los descriptivos, es preferible que sea el personal creador del recurso quien provea la información, en especial cuando se trata de conjuntos de datos científicos, donde el creador es quien tiene mayor comprensión del tema y de los usos que se le pueden dar a la información.

Sin embargo, si los autores o creadores de los datos no tienen el tiempo o las habilidades necesarias para hacerlo, es preferible que sea el personal técnico quien se encargue de su creación consultando con especialistas para confirmar su consistencia y corrección.

Capítulo 4 - Artículos científicos

Un artículo científico es un Informe original, escrito y publicado, que plantea y describe resultados experimentales, nuevos conocimientos o experiencias que se basan en hechos conocidos. Su finalidad es poder compartir y contrastar estos resultados con el resto de la comunidad científica, y una vez validados, se incorporen como recurso bibliográfico a disponibilidad de los interesados.

Existen múltiples definiciones del artículo científico, las cuales han aportado varios autores a lo largo de la historia de la investigación científica, pero de una manera precisa o sencilla puede decirse que el artículo científico es un documento cuyo objetivo es difundir de forma clara y precisa, en una extensión regular, los resultados de una investigación realizada sobre un área determinada del conocimiento.

La UNESCO **[21]** define el artículo científico como uno de los métodos inherentes al trabajo de la ciencia, cuya finalidad esencial es la de comunicar los resultados de investigaciones, ideas y debates de una manera clara, concisa y fidedigna. Al mismo tiempo, dicha organización considera los estudios recapitulativos como investigaciones realizadas sobre un tema determinado, en las que se reúnen, analizan y discuten informaciones ya publicadas, por lo que se pueden clasificar, a veces, como publicaciones secundarias o terciarias.

Sin embargo algunos autores estiman que el artículo de revisión es una forma de investigación o trabajo original que se realiza en una biblioteca y no en un laboratorio o unidad asistencial, y cuya diferencia fundamental será el tipo de información y la unidad de análisis y no los principios científicos que se aplican.

Las revisiones de la literatura en forma de artículos de revisión son de suma importancia hoy día, debido al incremento del número de las publicaciones científicas, lo que impide a los investigadores y especialistas poder leer toda la información publicada por razones de accesibilidad a las numerosas revistas, la falta de tiempo y su excesivo costo. Por ello, las revisiones son una solución que tienen los profesionales para mantenerse actualizados acerca de los últimos conocimientos y tendencias sobre una determinada materia. En este sentido, algunos autores señalan que cada cierta cantidad de artículos se necesita una revisión para consolidar la información existente y dar una respuesta clara y actualizada sobre un tema.

4.1 Categorías de artículos

La UNESCO define las siguientes categorizaciones de artículos:

Memorias científicas originales
En este tipo de artículos se informa sobre los resultados obtenidos, se describen métodos, técnicas y aparatos, se presentan nuevas ideas, etc.

Esta es la principal categoría de colaboraciones primarias destinadas a publicaciones periódicas.

Además de los artículos completos y las monografías, las notas preliminares y la exposición subsiguiente en forma de anotación desempeñan un papel importante en la publicación primaria.

Un texto pertenece a la categoría de "publicaciones originales" cuando contribuye a ampliar considerablemente el conocimiento o la comprensión de un problema y está redactado de tal manera que un investigador competente pueda repetir los experimentos, observaciones, cálculos o razonamientos teóricos del autor y juzgar sus conclusiones y a precisión de su trabajo.

Publicaciones secundarias y servicios de información
Por lo general, estos sistemas son administrados por importantes organismos comerciales o gubernamentales y se ocupan de la elaboración de resúmenes y el indizado de publicaciones primarias, así como del almacenamiento y recuperación de la información contenida en ellas. El autor de memorias científicas necesita estos sistemas para obtener resúmenes analíticos y grupos de palabras clave.

Estudios recapitulativos
Un estudio recapitulativo es una investigación realizada sobre un tema determinado, en a que se reúnen, analizan y discuten informaciones ya publicadas. Su alcance depende de la publicación a la que se destina. El estudio recapitulativo es considerado, a veces, como una publicación secundaria e, incluso, terciaria; de hecho, los compiladores creativos de este tipo de estudio a menudo lo complementan con actitudes considerables de información primaria.
El autor de un estudio recapitulativo debe tener en cuenta todos los trabajos publicados que han hecho avanzar el tema, o que lo habrían hecho avanzar si se hubiesen tomado en consideración.

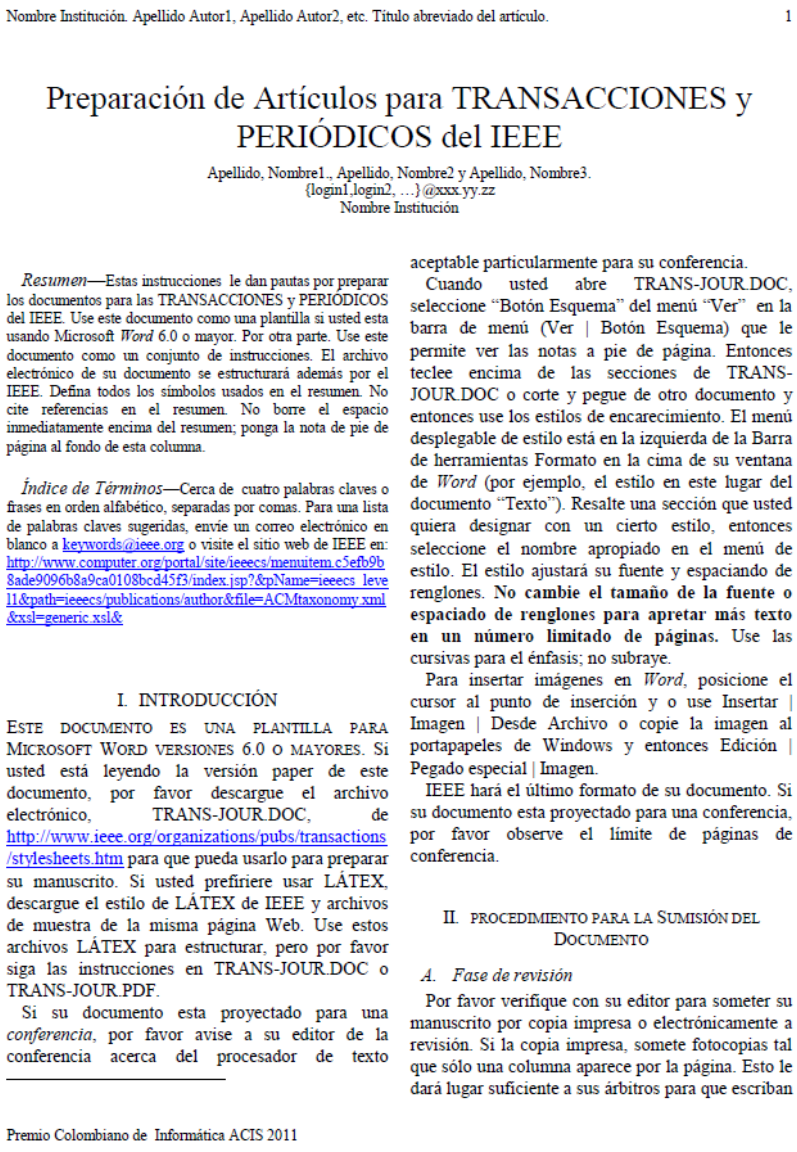
Por otro lado un *paper* es un artículo científico relativamente corto, en algunos casos monográficos, escrito con el fin de publicarse en revistas especializadas, de acuerdo con reglas específicas definidas de manera autónoma por los consejos y comités editoriales de las mismas.

El *paper* debe ser cuidadosamente redactado con el fin de que se haga fácilmente entendible y logre expresar de un modo claro y sintético lo que se pretende comunicar y para que contenga las citas y referencias necesarias. En la universidad los *papers* son artículos científicos que exponen síntesis de informes o tesis de mayor envergadura presentados en esta forma para facilitar al trabajo de quienes puedan estar interesados en consultar la obra original. La palabra inglesa *paper* tiene un sentido ligeramente más amplio, pues ella incluye también a lo que se suele llamar una ponencia. Los artículos científicos también se publican a veces como capítulos o partes independientes de ciertos libros, en los que algún estudioso, que asume el papel de compilador reúne varios trabajos de autores diferentes pero que tratan una materia común **[22]**.

4.2 Estándares de los artículos

Los artículos pueden ser escritos respetando algún estándar, dependiendo de su tipo, del congreso a ser presentando etc. Entre los más utilizados para la confección de los mismos se encuentra IEEE y LNCS.

4.2.1 Artículos para TRANSACCIONES y PERIÓDICOS del IEEE



Nombre Institución. Apellido Autor1, Apellido Autor2, etc. Título abreviado del artículo. 1

Preparación de Artículos para TRANSACCIONES y PERIÓDICOS del IEEE

Apellido, Nombre1., Apellido, Nombre2 y Apellido, Nombre3.
{login1,login2, …} @xxx.yy.zz
Nombre Institución

Resumen—Estas instrucciones le dan pautas por preparar los documentos para las TRANSACCIONES y PERIODICOS del IEEE. Use este documento como una plantilla si usted esta usando Microsoft *Word* 6.0 o mayor. Por otra parte. Use este documento como un conjunto de instrucciones. El archivo electrónico de su documento se estructurará además por el IEEE. Defina todos los símbolos usados en el resumen. No cite referencias en el resumen. No borre el espacio inmediatamente encima del resumen; ponga la nota de pie de página al fondo de esta columna.

Índice de Términos—Cerca de cuatro palabras claves o frases en orden alfabético, separadas por comas. Para una lista de palabras claves sugeridas, envíe un correo electrónico en blanco a keywords@ieee.org o visite el sitio web de IEEE en: http://www.computer.org/portal/site/ieeecs/menuitem.c5efb9b8ade9096b8a9ca0108bcd45f3/index.jsp?&pName=ieeecs_level1&path=ieeecs/publications/author&file=ACMtaxonomy.xml&xsl=generic.xsl&

I. INTRODUCCIÓN

ESTE DOCUMENTO ES UNA PLANTILLA PARA MICROSOFT WORD VERSIONES 6.0 O MAYORES. Si usted está leyendo la versión paper de este documento, por favor descargue el archivo electrónico, TRANS-JOUR.DOC, de http://www.ieee.org/organizations/pubs/transactions/stylesheets.htm para que pueda usarlo para preparar su manuscrito. Si usted prefiriere usar LÁTEX, descargue el estilo de LÁTEX de IEEE y archivos de muestra de la misma página Web. Use estos archivos LÁTEX para estructurar, pero por favor siga las instrucciones en TRANS-JOUR.DOC o TRANS-JOUR.PDF.

Si su documento esta proyectado para una *conferencia*, por favor avise a su editor de la conferencia acerca del procesador de texto aceptable particularmente para su conferencia.

Cuando usted abre TRANS-JOUR.DOC, seleccione "Botón Esquema" del menú "Ver" en la barra de menú (Ver | Botón Esquema) que le permite ver las notas a pie de página. Entonces teclee encima de las secciones de TRANS-JOUR.DOC o corte y pegue de otro documento y entonces use los estilos de encarecimiento. El menú desplegable de estilo está en la izquierda de la Barra de herramientas Formato en la cima de su ventana de *Word* (por ejemplo, el estilo en este lugar del documento "Texto"). Resalte una sección que usted quiera designar con un cierto estilo, entonces seleccione el nombre apropiado en el menú de estilo. El estilo ajustará su fuente y espaciando de renglones. **No cambie el tamaño de la fuente o espaciado de renglones para apretar más texto en un número limitado de páginas.** Use las cursivas para el énfasis; no subraye.

Para insertar imágenes en *Word*, posicione el cursor al punto de inserción y o use Insertar | Imagen | Desde Archivo o copie la imagen al portapapeles de Windows y entonces Edición | Pegado especial | Imagen.

IEEE hará el último formato de su documento. Si su documento esta proyectado para una conferencia, por favor observe el límite de páginas de conferencia.

II. PROCEDIMIENTO PARA LA SUMISIÓN DEL DOCUMENTO

A. Fase de revisión

Por favor verifique con su editor para someter su manuscrito por copia impresa o electrónicamente a revisión. Si la copia impresa, somete fotocopias tal que sólo una columna aparece por la página. Esto le dará lugar suficiente a sus árbitros para que escriban

Premio Colombiano de Informática ACIS 2011

Ilustración 2 Ejemplo de estructura de documento IEEE

documento; las normas de prueba son más altas cuando se reportan resultados extraordinarios o inesperados.

Porque la repetición se requiere para el progreso científico, documentos sometidos a la publicación deben proporcionar información suficiente para permitirles a los lectores realizar experimentos similares o cálculos y usar los resultados informados. Aunque no todo necesita ser descubierto, un documento debe contener nueva, usada, e información totalmente descubierta. Por ejemplo, la composición química de un espécimen necesita que no se informe si el propósito principal de un documento es introducir una nueva técnica de la medida. Los autores deben esperar ser desafiados por críticos si los resultados no son soportados por los datos adecuados y los detalles críticos.

Documentos que describen el trabajo continuo o anuncian el último logro técnico que es conveniente para la presentación en una conferencia profesional no pueden ser apropiados para la publicación en TRANSACCIONES o PERIÓDICOS

IX. Conclusiones

Una sección de conclusiones no se requiere. Aunque una conclusión puede repasar los puntos principales del documento, no reproduzca lo del resumen como conclusión. Una conclusión podría extender la importancia del trabajo o podría hacer pensar en aplicaciones y extensiones.

Apéndice

Los apéndices, si son necesarios, aparecen antes del reconocimiento.

Reconocimiento

La ortografía preferida de la palabra "acknowledgment" en inglés americano es sin una "e" después de la "g." Use el título singular aun cuando usted tiene muchos reconocimientos. Evite las expresiones como "Uno de nosotros (S.B.A.) gustaría agradecer... ." En cambio, escriba "F. A. agradecimentos del autor... ." **reconocimientos a patrocinador y de apoyo financieros se ponen en la nota a pie de página de la primera página sin numerar.**

References

[1] G. O. Young, "Synthetic structure of industrial plastics (Book style with paper title and editor)," in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.

[2] W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

[3] H. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1985, ch. 4.

[4] J. U. Duncombe, "Infrared navigation—Part I: An assessment of feasibility (Periodical style)," *IEEE Trans. Electron Devices*, vol. ED-11, pp. 34–39, Jan. 1959.

[5] S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," *IEEE Trans. Neural Networks*, vol. 4, pp. 570–578, July 1993.

[6] R. W. Lucky, "Automatic equalization for digital communication," *Bell Syst. Tech. J.*, vol. 44, no. 4, pp. 547–588, Apr. 1965.

[7] E. H. Miller, "A note on reflector arrays (Periodical style—Accepted for publication)," *IEEE Trans. Antennas Propagat.*, to be published.

[8] S. P. Bingulac, "On the compatibility of adaptive controllers (Published Conference Proceedings style)," in *Proc. 4th Annu. Allerton Conf. Circuits and Systems Theory*, New York, 1994, pp. 8–16.

[9] G. R. Faulhaber, "Design of service systems with priority reservation," in *Conf. Rec. 1995 IEEE Int. Conf. Communications*, pp. 3–8.

[10] W. D. Doyle, "Magnetization reversal in films with biaxial anisotropy," in *1987 Proc. INTERMAG Conf.*, pp. 2.2-1–2.2-6.

[11] G. W. Juette and L. E. Zeffanella, "Radio noise currents n short sections on bundle conductors (Presented Conference Paper style)," presented at the IEEE Summer power Meeting, Dallas, TX, June 22–27, 1990, Paper 90 SM 690-0 PWRS.

[12] J. G. Kreifeldt, "An analysis of surface-detected EMG as an amplitude-modulated noise," presented at the 1989 Int. Conf. Medicine and Biological Engineering, Chicago, IL.

[13] J. Williams, "Narrow-band analyzer (Thesis or Dissertation style)," Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993.

[14] N. Kawasaki, "Parametric study of thermal and chemical nonequilibrium nozzle flow," M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993.

[15] B. Smith, "An approach to graphs of linear forms (Unpublished work style)," unpublished.

[16] J. P. Wilkinson, "Nonlinear resonant circuit devices (Patent style)," U.S. Patent 3 624 12, July 16, 1990.

[17] A. Harrison, private communication, May 1995.

[18] *IEEE Criteria for Class IE Electric Systems* (Standards style), IEEE Standard 308, 1969.

[19] *Letter Symbols for Quantities*, ANSI Standard Y10.5-1968.

[20] R. E. Haskell and C. T. Case, "Transient signal propagation in lossless isotropic plasmas (Report style)," USAF Cambridge Res. Lab., Cambridge, MA Rep. ARCRL-66-234 (II), 1994, vol. 2.

Ilustración 3 Ejemplo de estructura de documento IEEE (continuación)

Nombre Institución. Apellido Autor1, Apellido Autor2, etc. Título abreviado del artículo.

Ilustración 4 Definición de título, institución y nombre de autores de documento IEEE

Resumen—Estas instrucciones le dan pautas por preparar los documentos para las TRANSACCIONES y PERIÓDICOS del IEEE. Use este documento como una plantilla si usted esta usando Microsoft *Word* 6.0 o mayor. Por otra parte. Use este documento como un conjunto de instrucciones. El archivo electrónico de su documento se estructurará además por el IEEE. Defina todos los símbolos usados en el resumen. No cite referencias en el resumen. No borre el espacio inmediatamente encima del resumen; ponga la nota de pie de página al fondo de esta columna.

Ilustración 5 Definición de resumen de documento IEEE

Índice de Términos—Cerca de cuatro palabras claves o frases en orden alfabético, separadas por comas. Para una lista de palabras claves sugeridas, envíe un correo electrónico en blanco a keywords@ieee.org o visite el sitio web de IEEE en: http://www.computer.org/portal/site/ieeecs/menuitem.c5efb9b8ade9096b8a9ca0108bcd45f3/index.jsp?&pName=ieeecs_level1&path=ieeecs/publications/author&file=ACMtaxonomy.xml&xsl=generic.xsl&

Ilustración 6 Definición de índice de términos de documento IEEE

I. INTRODUCCIÓN

ESTE DOCUMENTO ES UNA PLANTILLA PARA MICROSOFT WORD VERSIONES 6.0 O MAYORES. Si usted está leyendo la versión paper de este documento, por favor descargue el archivo electrónico, TRANS-JOUR.DOC, de http://www.ieee.org/organizations/pubs/transactions/stylesheets.htm para que pueda usarlo para preparar su manuscrito. Si usted prefiriere usar LÁTEX, descargue el estilo de LÁTEX de IEEE y archivos de muestra de la misma página Web. Use estos archivos LÁTEX para estructurar, pero por favor siga las instrucciones en TRANS-JOUR.DOC o TRANS-JOUR.PDF.

Si su documento esta proyectado para una *conferencia*, por favor avise a su editor de la conferencia acerca del procesador de texto

Ilustración 7 Definición de introducción de documento IEEE

REFERENCIAS

[1] G. O. Young, "Synthetic structure of industrial plastics (Book style with paper title and editor)," in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.

[2] W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

[3] H. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1985, ch. 4.

[4] J. U. Duncombe, "Infrared navigation—Part I: An assessment of feasibility (Periodical style)," *IEEE Trans. Electron Devices*, vol. ED-11, pp. 34–39, Jan. 1959.

[5] S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," *IEEE Trans. Neural Networks*, vol. 4, pp. 570–578, July 1993.

[6] R. W. Lucky, "Automatic equalization for digital communication," *Bell Syst. Tech. J.*, vol. 44, no. 4, pp. 547–588, Apr. 1965.

[7] E. H. Miller, "A note on reflector arrays (Periodical style—Accepted for publication)," *IEEE Trans. Antennas*

Ilustración 8 Definición de referencias de documento IEEE

4.2.2 Estandar LNCS/LNAI – Springer

Author Guidelines for the Preparation of Contributions to Springer Computer Science Proceedings

Alfred Hofmann[1,*], Ralf Gerstner[1], Anna Kramer[1], and Frank Holzwarth[2]

[1] Springer-Verlag, Computer Science Editorial, Heidelberg, Germany
{alfred.hofmann,ralf.gerstner,anna.kramer}@springer.com
[2] Springer-Verlag, Technical Support, Heidelberg, Germany
frank.holzwarth@springer.com

Abstract. The abstract is a mandatory element that should summarize the contents of the paper and should contain at least 70 and at most 150 words. Abstract and keywords are freely available in SpringerLink.

Keywords: We would like to encourage you to list your keywords here. They should be separated by middots.

1 Introduction

You will find here Springer's guidelines for the preparation of proceedings papers to be published in one of the following series, in printed and electronic form:

- Lecture Notes in Computer Science (LNCS), incl. its subseries Lecture Notes in Artificial Intelligence (LNAI) and Lecture Notes in Bioinformatics (LNBI), and LNCS Transactions;
- Lecture Notes in Business Information Processing (LNBIP);
- Communications in Computer and Information Science (CCIS);
- Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST);
- IFIP Advances in Information and Communication Technology (IFIP AICT), formerly known as the IFIP Series;
- Proceedings in Information and Communication Technology (PICT).

Your contribution may be prepared in LaTeX or Microsoft Word. Technical Instructions for working with Springer's style files and templates are provided in separate documents which can be found in the respective zip packages on our website.

* No academic titles or descriptions of academic positions should be included in the addresses. The affiliations should consist of the author's institution, town, and country.

Ilustración 9 Ejemplo de estructura de documento LNCS

Appendix: Springer Author Discount

Authors contributing to any of Springer's Computer Science proceedings publications are entitled to a 33.3% discount off all Springer products when placing an order through springer.com. To make use of this discount, please access the following page: `http://www.springer.com/gp/authors-editors/book-authors-editors/springertoken-request-for-springer-authors/4090`. You will be requested to give full details of your Springer publication and will be given a so-called SpringerToken. This token is a number that must be entered when placing an order through www.springer.com, in order to obtain the discount.

Contact Us

If you have any further questions regarding the preparation of your paper, then please do not hesitate to get in touch with us.

- For all questions related to our LaTeX style files, your contact person is: Mr. Frank Holzwarth, e-mail: `frank.holzwarth@springer.com`.
- For overall technical questions concerning the preparation of LNCS/LNAI/LNBI papers, please contact Ms. Anna Kramer, e-mail: `lncs@springer.com`.
- For the LNBIP series, please contact Ms. Viktoria Meyer, e-mail: `lnbip@springer.com`.
- For the CCIS series, please contact Ms. Leonie Kunz, e-mail: `ccis@springer.com`.
- For the LNICST series, please contact Mr. Peter Strasser, e-mail: `lnicst@springer.com`.
- For the IFIP AICT series, please contact Ms. Erika Siebert-Cole, e-mail: `ifip@springer.com`.

References

1. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. J. Mol. Biol. 147, 195–197 (1981)
2. May, P., Ehrlich, H.C., Steinke, T.: ZIB Structure Prediction Pipeline: Composing a Complex Biological Workflow through Web Services. In: Nagel, W.E., Walter, W.V., Lehner, W. (eds.) Euro-Par 2006. LNCS, vol. 4128, pp. 1148–1158. Springer, Heidelberg (2006)
3. Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, San Francisco (1999)
4. Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid Information Services for Distributed Resource Sharing. In: 10th IEEE International Symposium on High Performance Distributed Computing, pp. 181–184. IEEE Press, New York (2001)
5. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid: an Open Grid Services Architecture for Distributed Systems Integration. Technical report, Global Grid Forum (2002)
6. National Center for Biotechnology Information, `http://www.ncbi.nlm.nih.gov`

Ilustración 10 Ejemplo de estructura de documento LNCS (continuación)

Author Guidelines for the Preparation of Contributions to Springer Computer Science Proceedings

Ilustración 11 Definición de título de documento LNCS

Alfred Hofmann[1,*], Ralf Gerstner[1], Anna Kramer[1], and Frank Holzwarth[2]

[1] Springer-Verlag, Computer Science Editorial, Heidelberg, Germany
{alfred.hofmann,ralf.gerstner,anna.kramer}@springer.com
[2] Springer-Verlag, Technical Support, Heidelberg, Germany
frank.holzwarth@springer.com

Ilustración 12 Definición de institución y nombre de autores de documento LNCS

Abstract. The abstract is a mandatory element that should summarize the contents of the paper and should contain at least 70 and at most 150 words. Abstract and keywords are freely available in SpringerLink.

Ilustración 13 Definición de resumen de documento LNCS

Keywords: We would like to encourage you to list your keywords here. They should be separated by middots.

Ilustración 14 Definición de palabras claves de documento LNCS

1 Introduction

You will find here Springer's guidelines for the preparation of proceedings papers to be published in one of the following series, in printed and electronic form:

- Lecture Notes in Computer Science (LNCS), incl. its subseries Lecture Notes in Artificial Intelligence (LNAI) and Lecture Notes in Bioinformatics (LNBI), and LNCS Transactions;
- Lecture Notes in Business Information Processing (LNBIP);
- Communications in Computer and Information Science (CCIS);
- Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST);
- IFIP Advances in Information and Communication Technology (IFIP AICT), formerly known as the IFIP Series;
- Proceedings in Information and Communication Technology (PICT).

Your contribution may be prepared in LaTeX or Microsoft Word. Technical Instructions for working with Springer's style files and templates are provided in separate documents which can be found in the respective zip packages on our website.

Ilustración 15 Definición de introducción de documento LNCS

Contact Us

If you have any further questions regarding the preparation of your paper, then please do not hesitate to get in touch with us.

- For all questions related to our LaTeX style files, your contact person is: Mr. Frank Holzwarth, e-mail: `frank.holzwarth@springer.com`.
- For overall technical questions concerning the preparation of LNCS/LNAI/LNBI papers, please contact Ms. Anna Kramer, e-mail: `lncs@springer.com`.
- For the LNBIP series, please contact Ms. Viktoria Meyer, e-mail: `lnbip@springer.com`.
- For the CCIS series, please contact Ms. Leonie Kunz, e-mail: `ccis@springer.com`.
- For the LNICST series, please contact Mr. Peter Strasser, e-mail: `lnicst@springer.com`.
- For the IFIP AICT series, please contact Ms. Erika Siebert-Cole, e-mail: `ifip@springer.com`.

Ilustración 16 Definición de contacto de documento LNCS

References

1. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. J. Mol. Biol. 147, 195–197 (1981)
2. May, P., Ehrlich, H.C., Steinke, T.: ZIB Structure Prediction Pipeline: Composing a Complex Biological Workflow through Web Services. In: Nagel, W.E., Walter, W.V., Lehner, W. (eds.) Euro-Par 2006. LNCS, vol. 4128, pp. 1148–1158. Springer, Heidelberg (2006)
3. Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, San Francisco (1999)
4. Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid Information Services for Distributed Resource Sharing. In: 10th IEEE International Symposium on High Performance Distributed Computing, pp. 181–184. IEEE Press, New York (2001)
5. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid: an Open Grid Services Architecture for Distributed Systems Integration. Technical report, Global Grid Forum (2002)
6. National Center for Biotechnology Information, `http://www.ncbi.nlm.nih.gov`

Ilustración 17 Definición de referencias de documento LNCS

4.3 Secciones de los artículos

Título

Se presenta con mayúsculas, sin subrayar. El título de la investigación es la primera referencia que el lector tiene con documento, es decir es la primera impresión. Por lo tanto, el título induce el interés o desinterés del lector, un mal título puede hacer abortar la lectura del documento.

Autores

El nombre del autor o autores va centrado, debajo del título del trabajo. El nombre va con letra ordinario. No se incluyen los grados académicos. Debajo del nombre va la institución académica a la que pertenece.

Resumen o abstract

La redacción de este resumen debe realizarse de acuerdo o las recomendaciones formulados en la "Guía para la preparación y publicación de resúmenes analíticos destinados a la publicación", de la UNESCO. El resumen tiene una triple finalidad:

- Ayudar a quienes interesa el tema del artículo para que se puedan decidir si lo leerán por completo.
- Dar al lector para quien el artículo sólo ofrezca un interés marginal, la mayor cantidad de datos posibles a fin de que no tengan necesidad de leerlo íntegramente.
- Acelerar el trabajo de las revistas de resúmenes analíticos permitiéndoles reproducir inmediatamente el resumen, lo que contribuirá al mejoramiento general de los servicios de información científica.

El resumen debe ser un esbozo sucinto pero explícito del contenido y de las conclusiones del artículo y debe mencionar toda nueva información que figure en él. No debe contener datos o afirmaciones que no figuren en el texto del artículo y no debe recargarse con detalles de interés secundario.

Se recomienda presentar simultáneamente el resumen traducido al inglés, lo que facilitará la consulta por especialistas de esa lengua, así como, la indización y la codificación del artículo por bibliotecas o centros de documentación del extranjero.

Palabras clave

Las palabras clave (keywords) son una lista de cuatro a ocho términos clave relacionados con el contenido del artículo, generalmente se colocan después del resumen (abstract). Estas palabras son usadas por los servicios bibliográficos (Biological Abstracts, etc.) para clasificar el trabajo bajo un índice o tema particular.

Capítulo 5 - Repositorios Digitales

La sociedad actual está inmersa en un mundo de productos tecnológicos, este hecho está directamente relacionado con la revolución científico-técnica iniciada en el siglo XX y que se profundiza en el siglo XXI [Halaban, 2010]. **[23]**

Las tecnologías se presentan cada vez más como una necesidad en el contexto de la sociedad donde los rápidos cambios, el aumento, demanda y actualización de la información y de nuevos conocimientos se convierten en una exigencia permanente.

Es necesario tener presente las transformaciones que como sociedad hemos enfrentado en las últimas décadas, algunas de las cuales persisten hoy en día. Una de estas transformaciones, que continúa motivando cambios en nuestros hábitos y costumbres, ha tenido lugar gracias a la incorporación de nuevas y más sofisticadas Tecnologías de la Información y la Comunicación (TIC).

Uno de los cambios más importantes derivado directamente de las TIC es el aumento de nuestra capacidad de generar, almacenar y gestionar datos.

A medida que incorporamos más y más tecnología, crece también el volumen de datos que se generan. Actualmente, hasta los sistemas de información y sistemas informatizados más triviales subyacen sobre un enorme conjunto de datos, proveniente de operaciones simples y rutinarias, pero que si se los tratan y analizan cuidadosamente, se pueden obtener valiosos conocimientos. Estamos hablando de los datos masivos, conocidos como Big Data, que consiste en analizar y explotar grandes conjuntos de datos para crear nuevos productos o mejorar la competitividad y la productividad.

Los datos masivos son grandes conjuntos de datos digitales que requieren de grandes sistemas informáticos para su captura, almacenamiento, gestión y visualización y que, de hecho, no pueden ser analizados usando procesos y herramientas tradicionales. El aumento del número de fuentes y la potenciación de la captura implícita y explícita de datos han planteado una nueva situación en referencia a su tratamiento.

En la actualidad, el Big Data está revolucionando la gestión de la salud, la administración pública, las telecomunicaciones o las predicciones climáticas, entre muchos otros campos. También las universidades y los grupos de investigación se ven abocados a adentrarse en el tratamiento de la ingente cantidad de datos generados, tanto en el ámbito académico como científico. En este sentido, la creación y uso de repositorios digitales, podrían ayudar a generar Big datas para la investigación del desarrollo.

Uno de los muchos sectores que más datos recopila es el científico. La mayor parte de ellos sirven de apoyo a la publicación de artículos que son publicados mayormente en revistas científicas de los que, con suerte, hasta un 25% de media a escala mundial pasan a formar parte de los repositorios institucionales (Ginsparg, 2011; Björk, 2010; Gargouri, 2012). **[24]**

Un amplio número de universidades y centros de investigación de todo el mundo, incluido nuestro país, disponen ya de repositorios institucionales que almacenan los resultados de investigación de sus miembros (principalmente artículos, comunicaciones a congresos y tesis doctorales).

Los repositorios de datos de investigación sirven, entre otros fines, para validar resultados de investigación y, por tanto, deben estar vinculados de alguna manera a las publicaciones científicas en donde se muestra para qué fueron utilizados esos datos, por lo que algunos de los problemas se podrán abordar de forma conjunta, tanto para los repositorios institucionales como para los repositorios de datos de investigación.

El concepto de repositorio digital o repositorio de datos se amplía más de la digitalización y almacenamiento de documentos. Cualquier contenido digital, tal como una imagen, un archivo de audio, un documento textual-multimedial, un documento digitalizado, un libro electrónico, un sitio HTML, pueden formar parte de un "repositorio digital".

Podríamos definirlo como un depósito o archivo de un sitio web centralizado donde se almacena y mantiene información digital, habitualmente bases de datos o carpetas informáticas. Pueden contener los archivos en su servidor o referenciar desde su web al alojamiento originario. Pueden ser de acceso público, o pueden estar protegidos y necesitar de una autentificación previa. Los depósitos más conocidos son los de carácter académico e institucional **[25]** y tienen por objetivo la preservación y reutilización de contenido **[26]**, acceso permanente y mayor visibilidad, y facilidad de la búsqueda y recuperación **[27]** mediante el uso de etiquetas que los identifican y permiten su intercambio entre diferentes sistemas de información.

Por otra parte el Sistema Nacional de Repositorios Digitales **[28]** los repositorios digitales son colecciones digitales de la produccion cientifico-tecnologica de una institución, en las que se permite la búsqueda y recuperación de información para su posterior uso nacional e internacional. Un repositorio digital contiene mecanismos para importar, identificar, almacenar, preservar, recuperar y exportar un conjunto de objetos digitales. Esos objetos son descriptos mediante metadatos, datos que describen otros datos, que facilitan su recuperación. A su vez, los repositorios digitales son abiertos e interactivos, pues cumplen con protocolos internacionales que permiten la interoperabilidad entre ellos.

5.1 Acceso abierto

El término "Acceso abierto" (*Open Access*) se utiliza para definir plataformas de acceso a fuentes de información científica como pueden ser monografías, revistas, tesis, manuales de práctica clínica y cualquier otra modalidad de publicaciones editadas en formato electrónico. El acceso se realiza a través de Internet, permitiendo realizar búsquedas, lectura, recuperación de documentos, copia, impresión, distribución y enlace a accesos directos de textos completos de las fuentes, mediante un sistema de acceso libre, directo, permanente y gratuito.

El movimiento, *The Open Access Iniciative* (OAI) se encuentra comprometido con la calidad de los contenidos, con la garantía de accesibilidad a la información científica, con el mantenimiento de archivos que preserven el conocimiento, con la eliminación de la obligatoriedad de cesión del copyright de los artículos publicados, y por último, con todos los principios éticos relacionados con la investigación y la publicación científica de documentos.

El OAI tiene su origen lejano en iniciativas tendentes a almacenar información de calidad de distintas colecciones y fuentes de conocimiento del *Open Society Institute* (OIS), **[29]** fundación creada en 1993 por el investigador y filántropo George Soros, al objeto de promover sociedades, que al amparo de políticas gubernamentales, permitieran dar soporte a proyectos relacionados con la educación, multimedia, salud pública, derechos de la mujer, reformas legales, sociales y económicas.

Sin embargo, el verdadero lanzamiento del OAI surge en Budapest en una reunión realizada en diciembre del año 2012. En esta reunión, continuidad de otras anteriores, se marcaba como principal objetivo unir esfuerzos y aglutinar iniciativas separadas dentro de un plan estratégico que permitiese generar archivos exhaustivos de fuentes de conocimiento y de los logros recientes alcanzados, estableciendo como pieza central la utilización las fuentes de información de OIS como herramienta fundamental del OAI. Esta iniciativa aporta a la comunidad investigadora un nuevo poder, al facilitar el uso la literatura científica relevante, y que da a los autores y a sus trabajos una nueva visibilidad mesurable, eliminando las barreras, especialmente las del precio.

Así, el 14 de febrero de 2002, nace *Budapest Open Access Initiative* (BOAI) en la que se define acceso abierto (AA). Por “acceso abierto” a esta literatura queremos decir su disponibilidad gratuita en Internet público, permitiendo a cualquier usuario leer, descargar, copiar, distribuir, imprimir, buscar o usarlos con cualquier propósito legal, sin ninguna barrera financiera, legal o técnica, fuera de las que son inseparables de las que implica acceder a Internet mismo. La única limitación en cuanto a reproducción y distribución y el único rol del copyright en este dominio, deberá ser dar a los autores el control sobre la integridad de sus trabajos y el derecho de ser adecuadamente reconocidos y citados **[30]**.

Es también en Budapest donde se establecen las dos principales estrategias para alcanzar el acceso abierto: a través de repositorios (también llamado " acceso abierto verde") y a través de las revistas (también llamado " acceso abierto dorado").

5.2 Repositorio institucional de acceso abierto

La diversidad de formatos y tipos de documentos digitales desarrollados, la multitud de funciones demandadas por las instituciones y la gran posibilidad de aplicaciones ofrecidas por el software aplicado hace replantearse las definiciones de “Repositorio Institucional” enmarcadas dentro del Movimiento Open Access. Para empezar, sería necesario volver a recordar cómo surgen los RI, para qué y por qué.

Si volvemos a la Declaración de Budapest (2002) **[31]**, recordaremos que los "Repositorios Institucionales" surgen como una respuesta de las instituciones, en especial las académicas, hacia la política inflacionista de las revistas científicas tradicionales, tendente hacia la subida constante de precios, y la necesidad de las instituciones de conservar, preservar y poner a disposición de su comunidad académica e investigadora su patrimonio intelectual. En respuesta a estas dos realidades de la publicación científica comercial surgen los dos ejes fundamentales en donde debe basarse todo proyecto de repositorio institucional: la difusión de la investigación y el acceso abierto a los *e-documentos*.

Lo importante es que deben ser resultados de investigación y ser de acceso abierto. El formato de presentación del documento digital no sería determinante. Por ejemplo: tesis, artículos de revistas, fotos, mapas, documentos económicos o financieros, estadísticas, etc., pero también ideas, propuestas, hipótesis, experimentos, datos, informe de resultados, etc., entrarían dentro de un OA RI siempre si son el resultado de una investigación de la institución. Estos serían los factores clave identificadores de un "Repositorio Institucional", a diferencia de cualquier otro tipo de sistema de gestión de contenidos cuyas características pueden ser totalmente diferentes y, por supuesto, estar más orientado hacia la docencia /aprendizaje. Por ejemplo: un *Learning Object* utilizado para la docencia no debería formar parte de un OA RI, aunque existan iniciativas desarrolladas en este punto. En este último caso, puede tratarse más de un sistema de gestión de contenidos.

El concepto de repositorio institucional adolece de cierta imprecisión y puede llegar a utilizarse para designar realidades muy diferentes tanto en cuanto a objetivos como a implementaciones. Por ello consideramos que es fundamental acotar el campo de aplicación en el que vamos a utilizarlo para este trabajo.

Han sido múltiples las definiciones que se han dado de repositorio institucional. Sánchez y Melero **[32]** [Sánchez and Melero, 2006] han hecho una recopilación de las mismas. Entre ellas, se describen las siguientes:

- SPARC [Crow, 2002] lo define de forma muy escueta como una colección digital que agrupa y preserva la producción intelectual de una o varias universidades. Establece un objeto de aplicación muy amplio, el conjunto de la producción intelectual de la institución, en el que cabría más allá de los resultados de investigación y objetos de aprendizaje cualquier otro producto de carácter cultural producido por la institución **[33]**.

- Clifford Lynch [Lynch, 2003] define repositorio institucional como un conjunto de servicios que una institución ofrece a su comunidad para la gestión, y difusión de los contenidos digitales generados por los miembros de esa comunidad. Es, en su nivel más básico, un compromiso organizativo para el control de esos materiales digitales, incluyendo su preservación, su organización, acceso y distribución **[34]**.

- Cat S. McDowell [McDowell, 2007] considera que un repositorio debe cumplir al menos las siguientes características **[35]**:

- Es un servicio institucional abierto a toda la comunidad universitaria y a todo tipo de temáticas.
- Su objetivo debe ser reunir, preservar y dar acceso a, entre otras cosas, la producción de los investigadores y docentes en múltiple formatos. Se excluiría cualquier repositorio que ponga límites a esta producción, por ejemplo, sólo objetos de aprendizaje o sólo de tesis, etc.
- Debe recibir contenidos de forma activa bien a través de un formulario web o simplemente a través de correo electrónico.

- López Medina (2007) otorga a los repositorios institucionales las siguientes funciones **[36]**:
  - Es una herramienta común de gestión de contenidos digitales de la institución y de apoyo a la investigación y el aprendizaje.
  - Es un vehículo proactivo del "Open Access".
  - Es un lugar de almacenamiento y preservación.

5.2.1 Tipos de Acceso Abierto

Como se mencionó anteriormente el acceso abierto se consigue a través de dos rutas: "la dorada" la dorada" y "la verde". Por un lado tenemos la ruta dorada que va a contribuir al *open access* a través de la creación de las revistas de acceso abierto, y que se han desarrollado en el ámbito de algunas disciplinas científicas especializadas (biomedicina,) y en entornos geográficos específicos, como por ejemplo en Iberoamérica y el Caribe, destacando el proyecto SciELO (BIREME & FAPESP, 2013) que es la suma de revistas en acceso abierto en Ciencias de la Salud.

Y por otro lado, tenemos la ruta verde que desarrolla el acceso abierto a través de los Repositorios, pudiendo añadir un volumen importante de registros mayor de información en acceso abierto y de una manera más rápida que la que se consigue a través de la ruta dorada.
Así hay que apostar por la creación de repositorios institucionales como repositorios digitales de toda la cultura y el patrimonio de un pueblo en este caso, que lo hace propio y característico.

5.3 Herramientas de código abierto

La creación de repositorios digitales en una institución requiere llevar adelante el proceso de selección de una herramienta informática que atienda a criterios de calidad, fiabilidad y prestaciones. Desde hace varios años se vienen publicando informes que comparan aplicaciones para tal fin, es el caso de Nixon (2003) **[37]**, Crow (2004), **[38]** Han (2004) **[39]** y **[40]**, Kim (2005) **[41]**, Prudlo (2005) **[42]**, Tramullas Garrido (2005), **[43]**realizaron extensas descripciones sobre las prestaciones de Software, entre ellos los más destacados son entre las herramientas más populares disponibles actualmente para la creación de colecciones digitales se encuentran los siguientes Software: EPrints **[44]**, Dspace **[45]**, Fedora **[46]** y Greenstone **[47]**.

Según OpenDOAR (*Directory of Open Access Repositories*), las plataformas más utilizadas para crear repositorios digitales son las de software libre DSpace, E-Print, Fedora. América del Sur comprende el 8.7% de la cantidad total de repositorios en el mundo. Argentina actualmente tiene 34 repositorios registrados, de los cuales los softwares más utilizados son DSpace y Greenstone, en primer lugar y luego le sigue EPrints, todos ellos de software libre.
Greenstone, en realidad es una biblioteca digital, a diferencia de Dspace y EPrints que son repositorios digitales. A su vez, Dspace, es el software más utilizado a nivel mundial, con un 42 %, siguiéndole Eprints con el 14 %.

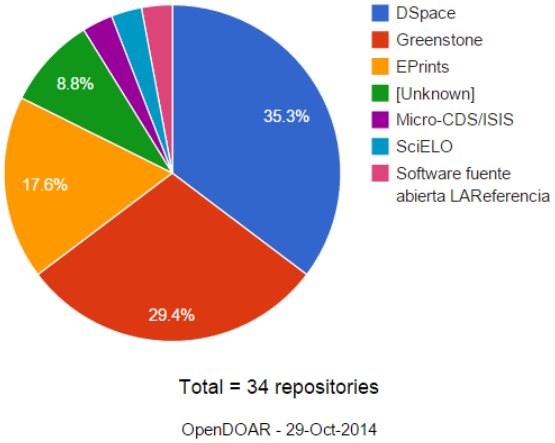


Ilustración 18 Software de Repositorios de Acceso Abierto usados en la Argentina (Fuente OpenDOAR)

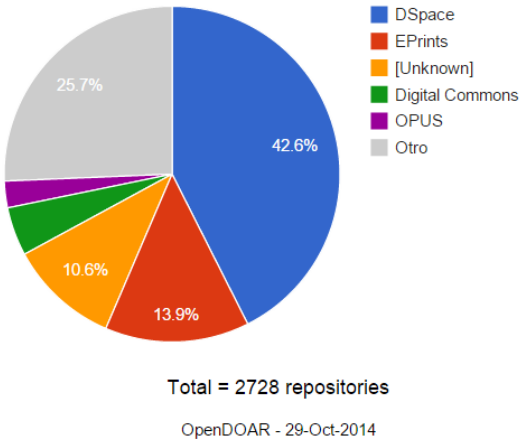


Ilustración 19 Software de Repositorios de Acceso Abierto usados en el mundo (Fuente OpenDOAR)

La amplia proliferación de las herramientas para crear colecciones digitales **[48]** hace que su selección requiera de un proceso de análisis antes de escoger la que se utilizará en cada institución, según sus necesidades y las bondades que ofrezca dicha herramienta. La selección debe centrarse principalmente en los siguientes aspectos **[49]**:

- Interfaz: La forma de presentación al usuario final, así como la presentación a la persona que se ocupa del procesamiento.
- Flexibilidad: Adaptación de la herramienta, según las características institucionales.
- Lenguaje: Idiomas del ambiente de procesamiento y de la interfaz de recuperación.

- Contenidos: Formato de los documentos que acepta en sus colecciones.
- Procesamiento: Facilidades para procesar los documentos para una recuperación efectiva.
- Recuperación: Formas que tiene el usuario de acceder a los documentos.
- Requerimientos de sistema: Características de las computadoras que soportarán la herramienta y de las que harán uso de las colecciones.
- Servidor Web: Requerimientos de los servidores en los que se soportará la herramienta.
- Licencia: Si es libre o privada.
- Costo: Gratis o pago.

5.4 Caracterización DSpace

DSpace es el software de código abierto más usado para gestión de repositorios. Provee herramientas para la administración de colecciones digitales, y comúnmente es usado como solución de repositorio institucional. Conserva y permite el acceso fácil y abierto a todo tipo de contenido digital, incluyendo texto, imágenes, videos y conjuntos de datos.

Fue liberado en el 2002, como producto de una alianza de HP y el MIT bajo una licencia BSD que permite a los usuarios personalizar o extender el software según se necesite. En el año 2005 fue lanzado DSpace 1.3. En marzo de 2008, la comunidad DSpace libera la versión 1.5, en marzo de 2010, fue liberada la versión 1.6. El 17 de julio de 2007, HP y MIT anunciaron conjuntamente la formación de la DSpace Foundation, una organización sin ánimo de lucro que proporcionará liderazgo y soporte a la comunidad DSpace. Finalmente el 12 de mayo de 2009, Fedora Commons y la Fundación DSpace crean una organización conjunta sin ánimo de lucro para perseguir una misión común. La llamaron DuraSpace. La misión es proporcionar liderazgo e innovación en tecnologías *open source* y basadas en nube (*cloud-based*) principalmente para bibliotecas, universidades, centros de investigación, y organizaciones de patrimonio cultural.

La plataforma DSpace es utilizado por numerosas instituciones de investigación y universitarias para los que se desarrolló inicialmente. Actualmente está siendo utilizado por museos, archivos, bibliotecas, repositorios de revistas, consorcios, administraciones públicas y privadas, y sociedades mercantiles para la gestión de sus activos digitales.

Según datos del SNRD, de los 27 repositorios registrados, el SEDICI **[50]** Servicio de Difusión de la Creación Intelectual y el Repositorio Institucional del Ministerio de Educación de la Nación **[51]**, son los que se encuentran en la primera y segunda posición en cuanto a cantidad de recursos disponibles, utilizando ambos a DSpace como repositorio central.

Un claro ejemplo de la confianza que se le tiene a DSpace como repositorio de contenidos es el caso de SEDICI que actualmente dispone de más de 25000 recursos entre los que encuentran Tesinas de grado y post-grado, publicaciones en revistas científicas, ponencias realizadas en congresos y conferencias, libros digitalizados, *e-books*, entre otros.

5.4.1 Características de DSpace

DSpace está desarrollado en Java bajo la licencia BSD, además de las características mencionadas previamente, posee la capacidad para ser personalizado, permitiendo satisfacer las necesidades de cualquier institución. Algunas de las principales formas en las que se puede personalizar la aplicación son:

Interfaz de usuario. Esto permite adaptar la apariencia del sitio web a la imagen corporativa de su institución. DSpace ofrece dos opciones de interfaz de usuario: la interfaz tradicional, JSPUI (basado en JSP), y XMLUI (basado en XML), que ofrece varios "temas" con diferentes diseños.

Personalizar los metadatos. DublinCore es el formato de metadatos nativo dentro de DSpace, sin embargo se puede agregar o cambiar cualquier campo. También es posible incluir esquemas de metadatos jerárquicos como MARC y MODS.

Estándares compatibles. DSpace cumple con muchos protocolos estándar para el acceso, importación y exportación de datos. Los estándares que soporta DSpace son: OAI-PMH, OAI-ORE, SWORD, WebDAV, OpenSearch, OpenURL, RSS, ATOM.

Configuración de la navegación y el motor de búsqueda. Se puede decidir qué campos se desean mostrar para la navegación, tales como autor, título, fecha, idioma, etc. De igual forma se pueden configurar los campos de metadatos que se requieran en el interfaz de búsqueda. DSpace permite la indexación a texto completo de cualquier ítem para permitir las búsquedas por contenido si se considera oportuno.

Capacidad de utilización de los mecanismos de autenticación local. DSpace incorpora plugins para la mayoría de los métodos de autenticación, incluyendo: LDAP (LDAP y jerárquico), Shibboleth, X.509, basada en IP. Además, DSpace cuenta con su propio método de autenticación interna, o también puede ser configurado para utilizar varios métodos de autenticación a la vez.

Base de datos configurable. DSpace puede funcionar con dos bases de datos: PostgreSQL **[52]** y Oracle **[53]**.

Idioma. DSpace está disponible en más de veinte idiomas por lo que se puede configurar DSpace para que sea compatible con varios idiomas a la vez.

5.4.2 Modelo de desarrollo comunitario

La comunidad DSpace ha intentado basar su estructura formal en la misma línea que en el modelo de desarrollo de la Apache Software Foundation. Es decir, hay una base de usuarios dentro de la cual hay un subconjunto de desarrolladores, varios de los cuales son contribuidores de la base de código núcleo. Los desarrollos de estos contribuidores son entonces añadidos a la distribución bajo la depuración de un equipo de committers, cuyo trabajo es garantizar que el código cumple las pautas de la documentación de desarrollo, y que contribuye efectivamente en la dirección del desarrollo de DSpace. SourceForge presta servicio tecnológico mediante una base de desarrollo, y varias listas de correo para preguntas técnicas y discusiones de desarrollo, así como una lista general para miembros comunitarios no-técnicos.

5.4.3 Modelo de datos

La información que almacena DSpace está estructurada en 5 componentes básicos:

Comunidades/Subcomunidades: El modelo de datos de DSpace se divide en comunidades, pudiendo ser divididas en sub-comunidades reflejando la estructura universitaria, departamento, centro de investigación, o de un laboratorio.

Colecciones: Comunidades contienen colecciones, que son agrupaciones de contenidos relacionados. Una colección puede aparecer en más de una comunidad.

Ítems: Cada colección se compone de items, que son los elementos básicos de archivo en el repositorio. Cada item es propiedad de una colección. Además, un elemento puede aparecer en colecciones adicionales; sin embargo cada item tiene una y sólo una colección de propietaria.

Bundles: Los items son subdivididos en paquetes de archivos (bundles) que a su vez estos agrupan un conjunto de archivos (bitstreams).

Bitstreams: Los bitstreams son, como su nombre indica, cadena de bits, por lo general los archivos físicos, como por ejemplo archivos HTML, PDF, etc.

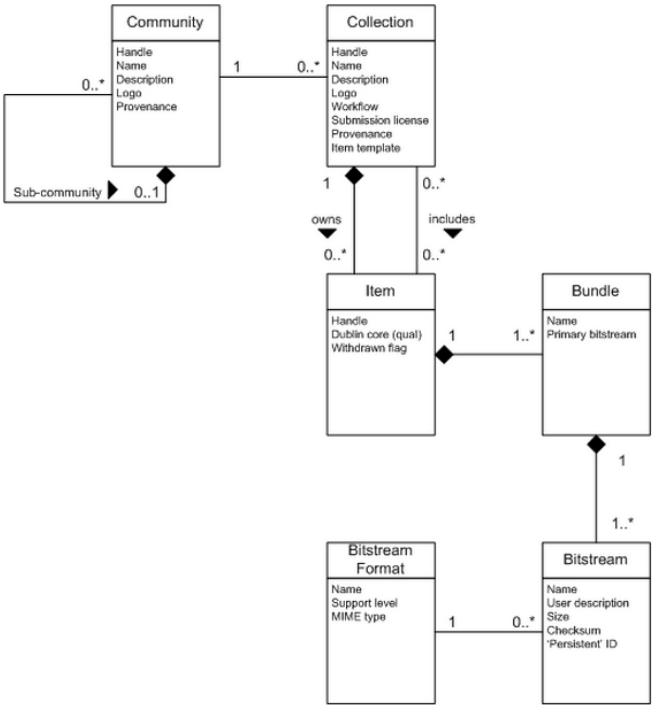


Ilustración 20 Modelo de datos de DSpace (Fuente: https://wiki.duraspace.org)

5.4.4 Arquitectura de DSpace

La arquitectura de DSpace está dividida en tres capas: capa de almacenamiento, capa de lógica de negocio, y capa de aplicación, cada una de las cuales ofrece servicios a la capa superior por medio de APIs, y utiliza los servicios de la capa inferior.

La capa de almacenamiento es donde se guarda la información de los grupos de usuarios, organización y metadatos del contenido; flujos de trabajo; autorizaciones, índices y los objetos digitales en el sistema de gestión de archivo.

Una capa lógica de gestión o administración de la plataforma. Proporciona los medios para leer o modificar los contenidos de la capa de almacenamiento y un sistema de autorización que provee el flujo de entradas y salidas.

Por último una capa de aplicaciones o interfaz de usuario donde los usuarios acceder vía navegador. Es el componente principal en esta capa y su presentación de cara al exterior.
Cuenta con un conjunto de módulos que permiten la interacción con el mundo exterior, entre ellos se destaca JSPUI y XMLUI.

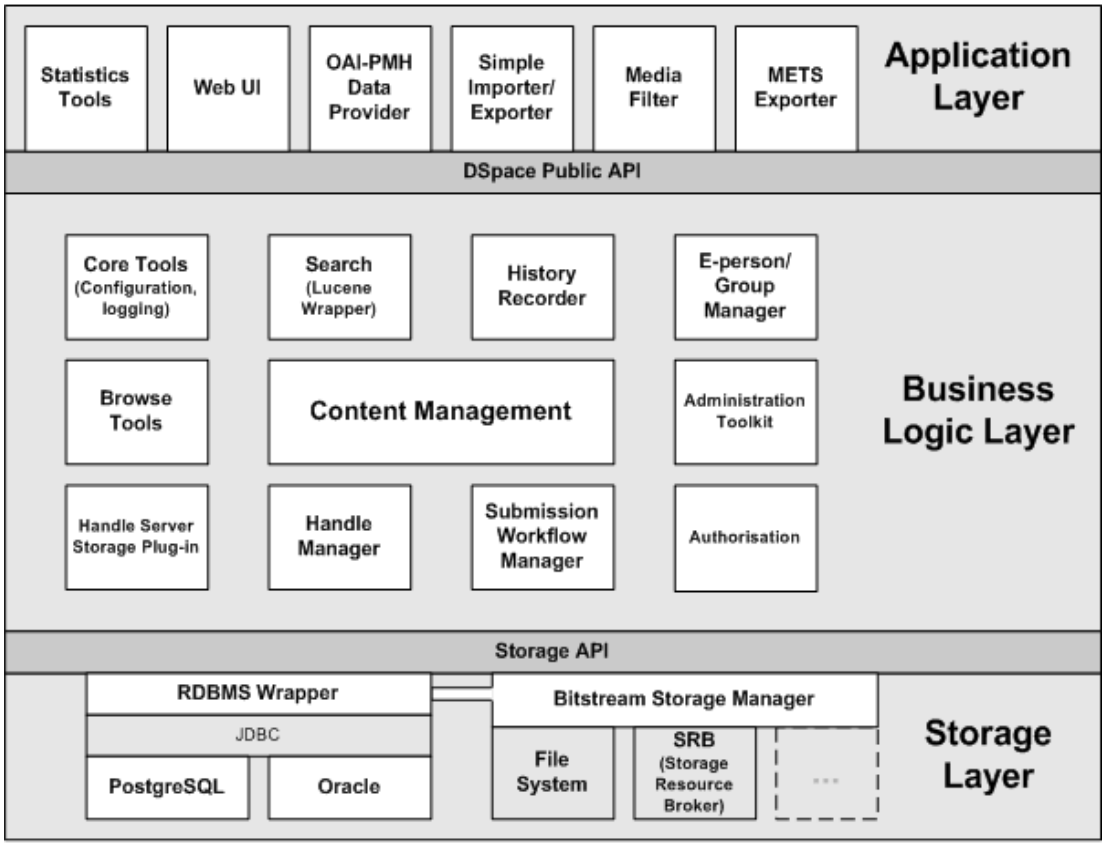


Ilustración 21 Arquitectura de DSpace fuente: https://wiki.duraspace.org/display/DSDOC4x/Architecture

5.4.5 Interfaces de DSpace

El repositorio digital DSpace soporta dos interfaces de usuario: una basada en JavaServer Pages (JSP) y otra basada en Apache Cocoon (XMLUI).

5.4.5.1 XMLUI

XMLUI (aka Manakin) es la interfaz de usuario basada en Cocoon *Framework* **[54]**, provee un modelo extensible en capas, las cuales implementan un orden de complejidad creciente:

Estilos: permite el uso de css y XHTML para customizar el template de XMLUI.

Temas: permite usar XSLT, XHTML y CSS para crear nuevos y más complejos templates para XMLUI.

Aspectos: permite usar el Cocoon *Framework* y Java (o XSLT) para crear nuevas funcionalidades y generar nuevos contenidos para DRI **[55]**(*Digital Repository Interface*).

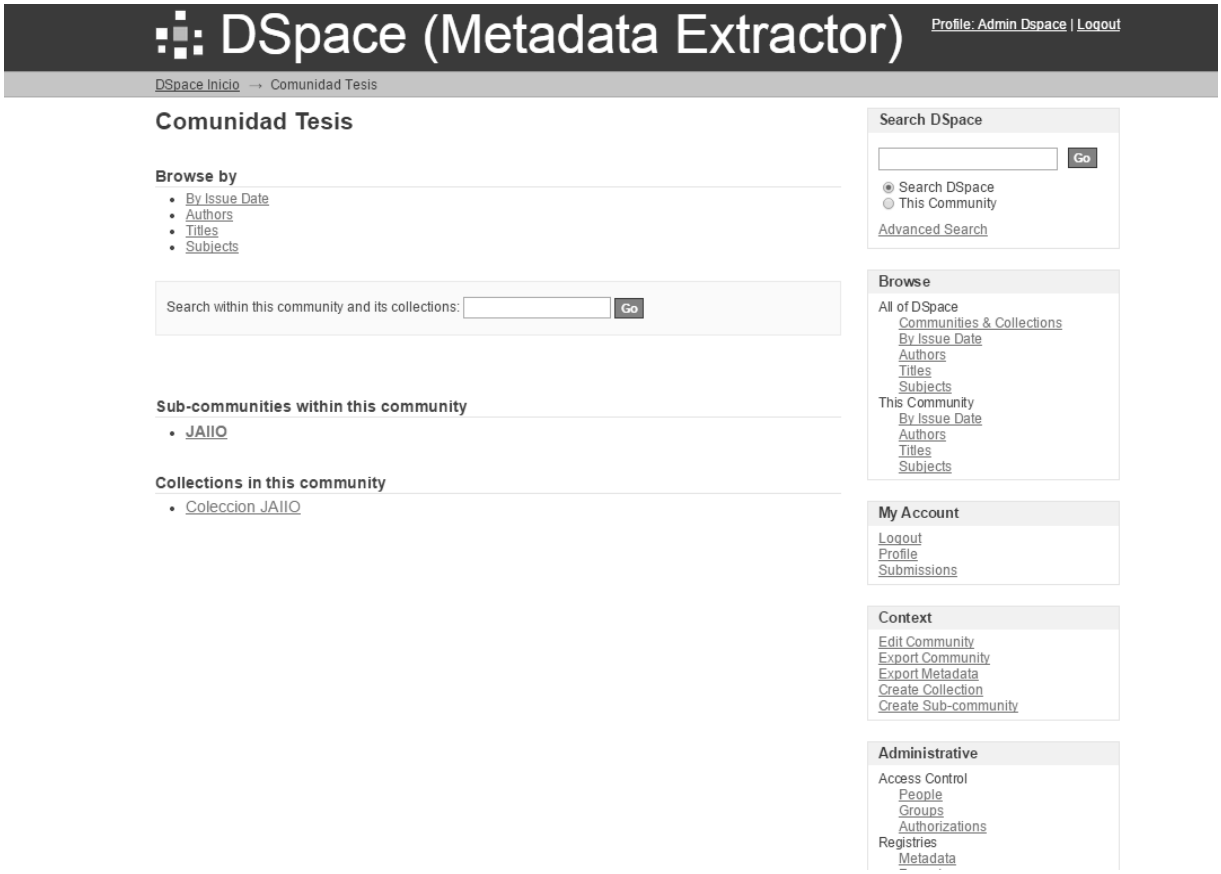


Ilustración 22 Workflow de carga de contenidos

Los workflows configurables son una característica opcional que puede ser habilitada para su uso sólo en DSpace XMLUI. El enfoque principal de la estructura de flujo de trabajo es crear una solución más flexible para que el administrador configure, e incluso para permitir que un desarrollador de aplicaciones para implementar medidas personalizadas, que se pueden configurar en el flujo de trabajo para la recogida a través de un archivo de configuración simple. El concepto detrás de este enfoque se inspira en el sistema de presentación configurable ya presente en DSpace.

Configuración del workflow

Desde el archivo item-submission.xml DSpace permite la configuración del *submission-process* o caminos de carga. Este consiste en una serie de "pasos", en los que se define:

- **processing-class**: Ruta completa de la clase Java encargada de procesar este paso. Esta clase debe llevar a cabo el procesamiento primario de cualquier información obtenida en esta etapa, tanto para el XMLUI y JSPUI.
- **jspui-binding**: Ruta completa de clase Java para la vista de JSPUI. Esta clase debería inicializar y llamar a la JSP apropiada para mostrar la interfaz de usuario del paso.

- **xmlui-binding**: Ruta completa de la clase Java de la vista de XMLUI. Esta clase debería generar el Manakin XML (documento DRI) necesarios para generar la interfaz de usuario del paso.
- **workflow-editable**: Define si este paso puede ser editado durante el proceso de edición de metadatos. Los valores posibles son true y false.

Por defecto, el proceso de carga de DSpace incluye los siguientes pasos, en este orden:

- **Seleccione de la Colección**: el usuario debe seleccionar una colección para depositar el contenido.
- **DescribeStep**: Aquí es donde el usuario puede introducir metadatos descriptivos sobre el contenido. Este paso puede consistir en una o más páginas de entrada de metadatos. De forma predeterminada, hay dos páginas. Cada página, contiene una secuencia de campos y cada campo define un metadato de Dublin Core junto con información necesaria para mostrar los campos en la vista, como por ejemplo: obligatoriedad, mensaje de error, etc.

```
<page number="1">
  <field>
    <dc-schema>dc</dc-schema>
    <dc-element>contributor</dc-element>
    <dc-qualifier>author</dc-qualifier>
    <repeatable>true</repeatable>
    <label>Authors</label>
    <input-type>name</input-type>
    <hint>Enter the names of the authors of this item below.</hint>
    <required></required>
  </field>

  <field>
    <dc-schema>dc</dc-schema>
    <dc-element>title</dc-element>
    <dc-qualifier></dc-qualifier>
    <repeatable>false</repeatable>
    <label>Title</label>
    <input-type>onebox</input-type>
    <hint>Enter the main title of the item.</hint>
    <required>You must enter a main title for this item.</required>
  </field>

  <field>
    <dc-schema>dc</dc-schema>
    <dc-element>title</dc-element>
    <dc-qualifier>alternative</dc-qualifier>
    <repeatable>true</repeatable>
    <label>Other Titles</label>
    <input-type>onebox</input-type>
    <hint>If the item has any alternative titles, please enter them below.</hint>
    <required></required>
  </field>

  <field>
    <dc-schema>dc</dc-schema>
    <dc-element>date</dc-element>
    <dc-qualifier>issued</dc-qualifier>
    <repeatable>false</repeatable>
    <label>Date of Issue</label>
    <input-type>date</input-type>
    <hint>Please give the date of previous publication or public distribution
                  below.  You can leave out the day and/or month if they aren't
                  applicable.</hint>
    <required>You must enter at least the year.</required>
  </field>
```

Ilustración 23 Configuración de los campos para una página

Cada campo (*field*) contiene los siguientes elementos, en cada uno se indica si es requerido para su definición.

- **dc-schema** (requerido): Nombre del esquema del metadato, por ejemplo *dc* para Dublin Core.
- **dc-element** (requerido): Nombre del elemento de Dublin Core que va a ser ingresado en el campo, por ejemplo *contributor*.
- **dc-qualifier**: Calificador del elemento de Dublin Core, por ejemplo, cuando el campo es contributor.advisor el valor de este elemento sería asesor. Dejando este atributo en blanco significa que el campo es para un elemento DC no cualificado.
- **repeatable**: El valor es “true” cuando se permiten múltiples valores de este campo, “false” en caso contrario.
- **label** (requerido): Texto a mostrar como la etiqueta del campo, describe lo que se va a ingresar, por ejemplo, "Nombre de Autor".
- **input-type** (Required): Define el tipo de widget asociado para el elemento de Dublin Core. El contenido debe ser una de las siguientes palabras clave:
    - **onebox**: caja de texto simple.
    - **twobox**: par de onebox, es para facilitar la carga de campos repetibles.
    - **textarea**: texto largo.
    - **name**: 2 inputs, uno para apellido y otro para nombre. Se guardan como 'Apellido, Nombre'.
    - **date**: Fecha. En campos requeridos, obliga a poner al menos el año.
    - **series**: 2 cajas de texto, una para título de la serie y otro para número.
    - **dropdown**: selector que despliega un listado de value-pairs.
    - **qualdrop_value**: caja doble con selector de calificador (en base a un value-pairs) y una caja de texto.
    - **list**: conjunto de checkboxes (campos repetibles) o radio buttons (campos no repetibles). Los valores se toman de un value-pairs.
- **hint** (requerido): El contenido es el texto que aparecerá como una "sugerencia", o instrucciones, al lado de los campos de entrada. Se pueden dejar vacíos.
- **required**: Cuando este elemento se incluye, se marca el campo como una entrada requerida. Si el usuario intenta salir de la página sin necesidad de introducir un valor para este campo, se muestra un mensaje de advertencia. Por ejemplo, <required> Debe introducir un título </ required>. Dejando el elemento requerido vacío no marcará al campo como obligatorio, por ejemplo: <Required></ required>
- **visibility**: Este es un campo opcional, cuando se incluye con un valor, se restringe la visibilidad del campo al alcance definido por ese valor. Si el elemento no se encuentra o está vacío, el campo es visible en todos los ámbitos.
- **UploadStep**: Aquí es donde el usuario puede cargar uno o más archivos para ser asociados al contenido.
- **ReviewStep**: Aquí es donde el usuario puede revisar toda la información anteriormente ingresada, y hacer correcciones cuando sea necesario.
- **Licencia**: Aquí es donde el usuario debe estar de acuerdo con la licencia de distribución del repositorio con el fin de completar el depósito. Esta licencia se define en el archivo [dspace]/config/default.license. También puede ser personalizado por cada colección.

- **Completado**: El depósito del contenido fue completado. Este estará disponible inmediatamente o podrá someterse a un proceso de aprobación.

```
<!--This "traditional" process defines the DEFAULT item submission process-->
<submission-process name="traditional">

 <step>
     <processing-class>org.dspace.submit.step.SkipInitialQuestionsStep</processing-class>
 </step>


  <!--Step 2 will be to Describe the item.-->
  <step>
    <heading>submit.progressbar.describe</heading>
    <processing-class>org.dspace.submit.step.DescribeStep</processing-class>
    <jspui-binding>org.dspace.app.webui.submit.step.JSPDescribeStep</jspui-binding>
    <xmlui-binding>org.dspace.app.xmlui.aspect.submission.submit.DescribeStep</xmlui-binding>
    <workflow-editable>true</workflow-editable>
  </step>

  <!--Step 3 will be to Upload the item-->
  <step>
    <heading>submit.progressbar.upload</heading>
    <processing-class>org.dspace.submit.step.UploadStep</processing-class>
    <jspui-binding>org.dspace.app.webui.submit.step.JSPUploadStep</jspui-binding>
    <xmlui-binding>org.dspace.app.xmlui.aspect.submission.submit.UploadStep</xmlui-binding>
    <workflow-editable>true</workflow-editable>
  </step>

  Step 4 will be to Verify/Review everything
  <step>
    <heading>submit.progressbar.verify</heading>
    <processing-class>org.dspace.submit.step.VerifyStep</processing-class>
    <jspui-binding>org.dspace.app.webui.submit.step.JSPVerifyStep</jspui-binding>
    <xmlui-binding>org.dspace.app.xmlui.aspect.submission.submit.ReviewStep</xmlui-binding>
    <workflow-editable>true</workflow-editable>
  </step>


  <!--Step 6 will be to Sign off on the License -->
  <step>
    <heading>submit.progressbar.license</heading>
    <processing-class>org.dspace.submit.step.LicenseStep</processing-class>
    <jspui-binding>org.dspace.app.webui.submit.step.JSPLicenseStep</jspui-binding>
    <xmlui-binding>org.dspace.app.xmlui.aspect.submission.submit.LicenseStep</xmlui-binding>
    <workflow-editable>false</workflow-editable>
  </step>
```

Ilustración 24 Especificación de las etapas para carga de contenidos

5.4.5.2 JSPUI

La interfaz JSPUI se implementa utilizando Java Servlets para manejar la lógica de negocio, y *JavaServer Pages* (JSP) que proporciona las páginas HTML enviadas a un usuario final.

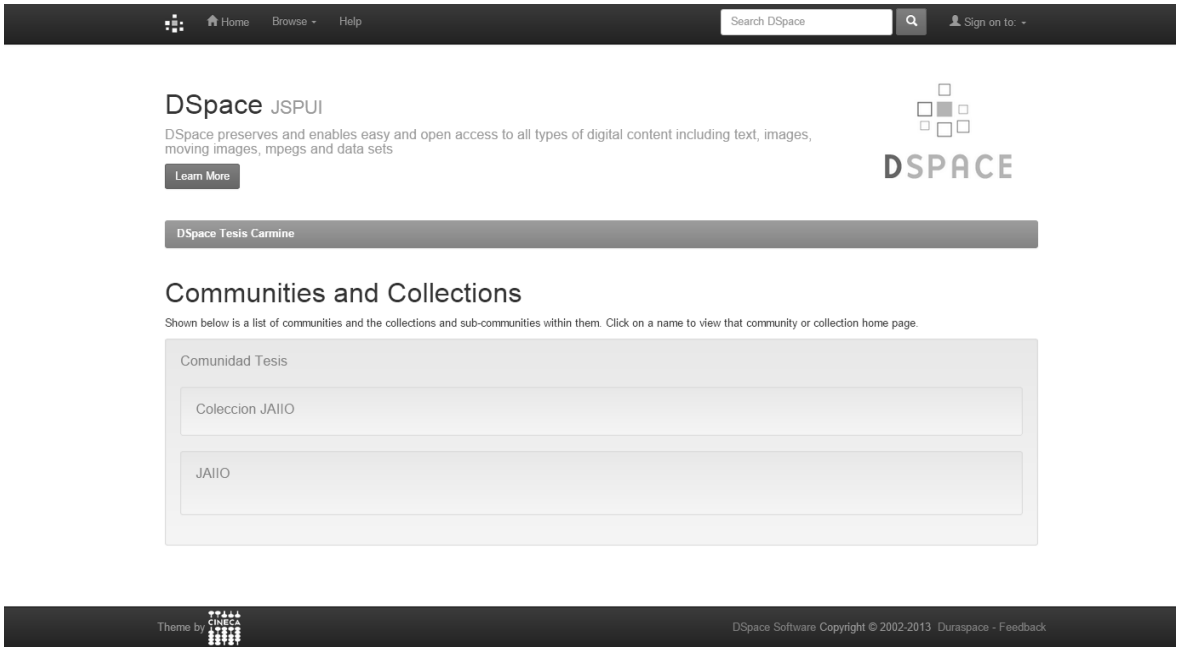


Ilustración 25 interfaz JSPUI

Capítulo 6 – Técnicas de recuperación de información

La recuperación de información (*IR Information Retrieval*) es un término amplio, y en algunos casos poco definido y en este contexto se refiere solamente a los sistemas automáticos de recuperación de información. Lancaster proporciona una definición: "*Un sistema de recuperación de información no informa (es decir cambia el conocimiento) al usuario del propósito de su pregunta. Este informa simplemente de la existencia (o la no existencia) y paradero de documentos referentes a su petición"* **[56]** [Lancaster, F. W.].

Jacobs y Rau definen la recuperación de información como una tarea que, dado un conjunto de documentos y una consulta de usuarios, encuentra los documentos relevantes. Las aplicaciones de IR requieren velocidad, consistencia, precisión, y facilidad de uso en la recuperación de textos relevantes para satisfacer las consultas de los usuarios [Jacobs y Rau] **[57]**.

Según [Moreno, A.], la IR "*se ocupa de tomar la consulta de un usuario a una base de datos y elegir entre todos los textos que se tienen archivados aquellos que mejor responda a las condiciones de búsqueda planteadas. Cuanto mayor sea el número de textos y más diversos sean los temas de los que tratan, más difícil será responder con exactitud*". De aquí surge la necesidad de "entender" realmente la pregunta y reconocer el contenido del documento **[58]**.

Rijsbergen [van Rijsbergen. C.J.] Resume las diferencias entre la recuperación de datos (DR Data Retrieval) y la recuperación de información (IR Information Retrieval), **[59]**La siguiente tabla marca las diferencias fundamentales existentes entre recuperación de datos y recuperación de información a juicio de Rijsbergen **[60]**[RIJ, 1999]:

Tabla 4 Diferencias entre la recuperación de datos y la recuperación de información

Propiedad	Data Retrieval (DR)	Information Retrieval (IR)
Matching	Match exactos	Match parciales, el mejor match
Inferencia	Deducción	Inducción
Modelo	Determinístico	Probabilístico
Clasificación	Monothetic	Polythetic
Lenguaje de Consulta	Artificial	Natural
Especificación de Consulta	Completa	Incompleta
Items requeridos	Matching	Relevantes
Respuesta ante error	Sensible	Insensible

Según la tabla anterior la recuperación de datos busca generalmente un *matching* exacto, es decir, se verifica si un item está o no está presente en el archivo. Este tipo de búsqueda puede, ser a veces de interés para la recuperación de información, pero generalmente se trata de hacer corresponder parcialmente estos items con la consulta y después se seleccionan los mejores.

La inferencia usada en la recuperación de datos es de la clase deductiva simple, es decir, aRb y bRc entonces aRc. En la recuperación de información es más común utilizar inferencia inductiva; las relaciones se especifican solamente con el grado de certeza o incertidumbre y por lo tanto la confianza en la inferencia es variable. Esta distinción conduce a describir a la recuperación de datos como una recuperación determinista y a la recuperación de información como probabilística.

El lenguaje de consulta para DR generalmente es artificial, con una sintaxis y vocabulario restringido; en IR se prefiere utilizar el lenguaje natural. En DR la consulta es una especificación completa de lo que se desea, mientras que en IR generalmente es incompleta. Esta última diferencia se presenta debido a que en la IR se están buscando lo documentos relevantes, en lugar de items que correspondan exactamente. Por último la DR es más sensible al error en el sentido que, un error en el *matching* no recuperara el item deseado lo que implica una falla total del sistema. Los pequeños errores en los *matching* de IR generalmente no afectan de manera perceptible el funcionamiento del sistema.

Aunque la recuperación de información se puede subdividir de muchas maneras, hay tres campos de investigación principales, ellos son: análisis del contenido, estructuras de la información, y evaluación [van Rijsbergen. C.J.]. El primero tiene que describir abreviadamente el contenido de los documentos en una forma conveniente para el tratamiento a través de una computadora; el segundo se refiere a explotar relaciones entre los documentos para mejorar la eficacia de las estrategias de la extracción; el tercero con la medida de la eficacia de la extracción.

A partir de las técnicas básicas de recuperación de información, se han originado diferentes tecnologías entre ellas las denominadas Minería de Texto y Extracción de Información.

6.1 Minería de Texto

Se trata de un grupo de herramientas destinadas a la recuperación de información en los sistemas. Este grupo de herramientas implementa una tecnología llamada "Minería" sirve para identificar y extraer valores importantes de los datos, descubrir conocimiento, informaciones ocultas, asociaciones, patrones y características **[61]**. Existen dos enfoques, la minería de datos y la minería de texto. La minería inteligente de datos permite minar la estructura de los datos almacenados en una base de datos convencional. En cambio, en la minería inteligente de texto la fuente de información proviene de textos en lenguaje natural, tales como correspondencia de los clientes, servicios de noticias, correo electrónico, páginas web, etc.

La minería de texto permite extraer patrones de texto, documentos organizados por sujetos, temas predominantes en la colección de documentos y búsqueda de documentos relevantes [IBM-2]. Los servicios que presta la minería de texto son diversos:

- Asignar documentos a categorías predefinidas, obteniendo una lista de nombres de categorías y niveles.
- Dividir documentos en grupos, llamados "clusters" para facilitar el proceso de exploración y encontrar información similar o relacionada. Aquí se pueden identificar relaciones ocultas entre los documentos, y se descubren documentos repetidos.
- Extraer características (nombres, terminologías, abreviaciones, etc.) automáticamente.
- Identificar el idioma del texto, indexar por el lenguaje y restringir las búsquedas en un lenguaje particular.
- Búsquedas por texto, realizan un análisis lingüístico para procesar las consultas en lenguaje natural y procesar los diccionarios.

6.2 Extracción de Información

La extracción de información (IE *Information Extraction*) analiza colecciones de textos. Luego los transforma en información que es procesada y analizada más fácilmente. Esto identifica los fragmentos de textos relevantes, extrae la información relevante de los fragmentos, y con estas piezas organiza la información requerida en una estructura coherente. Se trata de reconocer la información importante contenida en los documentos y trasladarla a un formato predefinido para que pueda ser tratada y recuperada con mayor facilidad.
El objetivo de los investigadores de IE es construir sistemas que encuentren items que puedan ser de interés para el análisis humano a partir de documentos.
Además de la información relevante deben conseguirse las relaciones entre ellos, mientras que se ignora la información irrelevante. Hoy día, sin embargo, los sistemas de IE tratan solamente con tipos específicos de textos y solo tienen buenos resultados en algunos componentes **[62]**.

6.3 Procesamiento de lenguaje natural

El Procesamiento del Lenguaje Natural (NLP Natural Language Processing), originalmente desarrollado a comienzos de la Guerra Fría **[63]**[Locke y Booth] como el mecanismo que usaban los físicos Soviéticos para la traducción de documentos, es uno de los primeros objetivos computacionales más investigados. Estos esfuerzos prematuros, por analizar y modelar el lenguaje humano, fueron caracterizados por una técnica sin conocimiento lingüístico y por el bajo rendimiento computacional de la época.

El lenguaje es uno de los aspectos fundamentales no sólo del comportamiento humano, sino de su propia naturaleza. En su forma escrita nos permite guardar un registro del conocimiento que se transmite de generación en generación, y en su forma hablada constituye el principal medio de comunicación en nuestro día a día.

El Procesamiento del Lenguaje Natural (NLP, Natural Language Processing) es la rama de las ciencias computacionales encargada del diseño e implementación de los elementos software y hardware necesarios para el tratamiento computacional del lenguaje natural, entendiendo como tal todo lenguaje humano, en contraposición a los lenguajes formales propios del ámbito lógico, matemático, o computacional.

El objetivo último que se persigue es el de la comprensión del lenguaje humano por parte de la computadora. La consecución de un objetivo tan ambicioso, del que todavía se está muy lejos, supondría una auténtica revolución. Por una parte, las computadoras podrían tener por fin acceso al conocimiento humano, y por otra, una nueva generación de interfaces, en lenguaje natural, facilitaría en grado sumo la accesibilidad a sistemas complejos.

6.3.1 OpenNlp

OpenNLP **[64]** es un proyecto de la Apache Software Foundation de uso libre y *open source*, que consiste en una suite de herramientas para el procesamiento del lenguaje natural basadas en el aprendizaje de máquinas. Como la mayoría de las herramientas basadas en técnicas inductivas, se requiere entrenar los distintos componentes para luego utilizarlos sobre texto nuevo. Se pueden descargar sets de datos pre-entrenados para diversos idiomas, entre ellos el español.

OpenNLP tiene una variedad de herramientas en Java basadas en las técnicas de NLP, que permiten la detección de oraciones, tokenization, pos-tagging, chunking, parsing y detección del nombre-entidad.

Los métodos de análisis de OpenNLP se basan en la utilización de ficheros de "patrones" en los que se contienen millones de ejemplos analizados que se utilizan como base. Utiliza "hidden markov models". ""Hidden markov models" (HMM) es un modelo estadístico en el que asume que el sistema a modelar es un proceso de Markov de parámetros desconocidos. El objetivo es determinar los parámetros desconocidos a partir de los parámetros observables **[65]**." Entonces, OpenNLP utiliza este método para clasificar morfológicamente las palabras a partir de los patrones, ya que como hemos dicho anteriormente, contienen ejemplos manualmente analizados.

OpenNLP dispone de un conjunto de herramientas para el procesamiento de lenguaje natural, entre las más destacadas podemos destacar las siguientes:

Detección de entidades nombradas (NE)
El buscador de nombres puede detectar entidades citadas en el texto. Para ser capaz de detectar las entidades el nameFinder necesita un modelo. El modelo depende del tipo de idioma y la entidad que fue entrenado. OpenNLP dispone una serie de modelos de nombres pre-formados.

Sentence Detection
Esta tarea se encarga de realizar la separación de los párrafos entregados por la tarea anterior en oraciones o sentencias. Esto no es una operación trivial, ya que si bien se podría pensar en dividir el párrafo en sentencias cada vez que ocurre un punto, es probable que debido a las ambigüedades del texto natural esto puede producir una equivocación.

Separación de Palabras (Tokenizer)
Esta tarea está encargada de dividir una sentencia en un conjunto de palabras. De la misma manera que la tarea anterior, esta operación no es trivial debido al uso de caracteres especiales en el texto natural

6.3.2 Utilidad en el proyecto

En el proyecto se utilizará para analizar el texto obtenido en lenguaje natural identificando sentencias, palabras, nombre de personas, etc. Los algoritmos desarrollados descomponen el texo en sentencias utilizando "SentenceDetector" para luego realizar los análisis sobre estas sentencias. En otros casos se utiliza el Tokenizer para descomponer una sentencia en tokens o unidades indivisibles y analizar el token en búsqueda de algún patrón que identifique la presencia de algún metadato.

6.4 Técnicas automáticas en la RI

Con la frase técnicas automáticas se hace referencia al conjunto de procedimientos y recursos que se aplican para que el sistema explote capacidades que el usuario no posee, lo alivie de las tareas rutinarias y trabajosas, o complemente y amplíe sus capacidades.
Aplicando este criterio amplio, muchas de las cosas que se tratan en la bibliografía sobre RI son técnicas. Por ejemplo, la base de datos es un recurso de almacenamiento y recuperación de información que supera ampliamente a la memoria humana. Una interfaz de búsqueda gráfica es un procedimiento que permite el acceso temático a una colección de miles de documentos en solo 2 o 3 pantallas. Todas estas técnicas son empleadas para lograr una interacción exitosa entre un usuario, que puede ser humano o máquina, concierta necesidad informativa; y una masa de información variada, registrada en documentos digitales o no, susceptible de satisfacer dicha necesidad y que puede o no haber sido sometida a algún proceso de descripción previo. Sin embargo, en este trabajo se acota el concepto de técnicas automáticas de recuperación a la clasificación propuesta por K.Sparck Jones [1, p.305], **[66]**quien sostiene, no sin cierta dificultad, que las técnicas se pueden agrupar en:

- Técnicas de indización
- Técnicas de búsqueda

Las técnicas de indización tienen que ver con la construcción de la representación del documento y de la representación de la necesidad de información del usuario en el sistema de recuperación. Las técnicas de búsqueda tienen que ver con la manera en que el archivo de documentos es examinado y los ítems son extraídos de acuerdo a la interrogación que se formuló.

Capítulo 7 - Herramientas para la extracción de metadatos

7.1 Introducción

Anteriormente se mencionó la importancia de la extracción automática de metadatos como una forma de garantizar la calidad de la información que se almacena en los repositorios y que será utilizada durante la ejecución de las búsquedas, y de alguna manera, asistirá a los usuarios en la selección de contenidos relevantes a sus preferencias y necesidades.
Esto, sumado a la estandarización de metadatos y el auge de los repositorios institucionales y del acceso abierto al conocimiento, da como resultado el fundamento necesario para comprender la verdadera importancia del desarrollo de nuevos algoritmos para la extracción automática a partir de documentos en repositorios institucionales.

Al momento de elegir o diseñar una herramienta para la extracción automática de metadatos es importante tener en cuenta estos aspectos:

Tipo de archivo (PDF, DOC, HTML, etc.).
Metadatos a extraer, punto fundamental en el desarrollo de las estrategias de extracción.
Técnicas y recursos utilizados para realizar la extracción, por ejemplo herramientas para el procesamiento de lenguaje natural (NLP), ontologías, etc.

7.2 Herramientas existentes

No hay muchos trabajos centrados en la extracción automática de metadatos, cada herramienta extrae distintos tipos de metadatos y poseen diferentes tipos de arquitecturas. Muchos de estos productos de extracción no están disponibles como herramientas libres y es complejo poder integrarlas a otros productos de software.
La mayoría de las herramientas extraen metadatos desde la meta-información del archivo y no desde el texto.

Por ejemplo KEA *Automatic Keyphrase Extraction* es la implementación en JAVA del algoritmo KEA [16]. La herramienta extrae automáticamente frases claves del texto completo a partir del documento a analizar. El conjunto de todas las frases seleccionadas en un documento se identifican utilizando procesamiento léxico rudimentario. Utiliza técnicas de machine-learning para generar un clasificador que determina qué frases candidatas deben ser asignadas como frases clave. Esta herramienta no nos serviría para el objetivo propuesto en la tesis, dado a que la palabras clave están presenten en el peper y no es necesario generarlas.

Alchemy API **[67]** es una plataforma de minería de texto la cual proporciona un conjunto de herramientas que permiten el análisis semántico utilizando técnicas de procesamiento de lenguaje natural. Provee un conjunto de servicios que permiten analizar de forma automática documentos de texto plano o HTML. La herramienta expone varios servicios a partir de su RESTful API, entre los que se encuentran: Extracción de Autor, Entidades, Palabras Claves, Categorización del Contenido e Identificación del Idioma. En su versión gratuita, el servicio presenta una limitación de 1000 consultas diarias y un límite por consulta de 150 kbs.

Mr. DLib [17] es una biblioteca digital que proporciona acceso a varios millones de artículos de texto completo y sus metadatos en formato XML y JSON a través de un servicio web RESTful. En su etapa beta de desarrollo sus funcionalidades son utilizadas por terceros y permite extraer Título y Autores [18].

En [20] **[68]** se realiza un análisis de varias herramientas para la extracción automática llegando a las siguientes conclusiones:

Herramientas analizadas: SAFEX, TWYS, Looking4LO, MAGIC

a) No se contemplan gran cantidad de formatos de archivos; esto puede llegar a limitar la funcionalidad y utilidad del repositorio.

b) No se extraen de manera significativa metadatos educacionales, que resultan ser sumamente importantes a la hora de la recuperación de los OA mediante ejecución de búsquedas.

c) No siguen un estándar de metadatos, lo que puede conducir a la diversidad y baja calidad en la información descriptiva asociada a los OA.

d) Si bien actualmente hay algunos estudios y sistemas para la extracción automática de metadatos, falta mucho por hacer, en parte, porque implica la aplicación de estrategias de inteligencia artificial.

Teniendo en cuenta el análisis que se realizó de los cuatro sistemas extractores de metadatos, es evidente que en lo que respecta a esta tecnología es mucho el camino que queda por recorrer, tanto en aquellos tipos de archivos conocidos y comunes como documentos de texto y PDF. Esto es así no sólo en el momento de ser autoarchivados y clasificados, sino también en el de ser incluidos en los resultados de las búsquedas, precisamente porque los metadatos asociados a los mismos no son claramente identificados. Es por esto que nace la necesidad de desarrollar estrategias precisas que se ajusten a un universo particular de contenidos con el fin de brindar calidad en los resultados.

Capítulo 8 - Herramienta desarrollada

Como se mencionó en el capítulo anterior la necesidad de desarrollar estrategias aplicadas a la recuperación de metadatos sobre un espacio particular de documentos originó el desarrollo del *framework* de extracción. El *framework* y DSpace como repositorio de contenidos son aliados perfectos para poder aplicar las técnicas de extracción.

Desde el inicio, el *framework* se pensó como una solución desacoplada de DSpace con el objetivo de mantener independencia funcional y de no alterar el código del repositorio, pudiendo así ser incorporado en instalaciones en funcionamiento con simples configuraciones. Su arquitectura permite agrupar las estrategias de extracción de metadatos como un módulo de DSpace y contemplando la posibilidad de extender la funcionalidad con el fin de ampliar el conjunto de metadatos soportados.

8.1 Metadatos de interés

Para el desarrollo del *framework* de extracción, con el fin de acotar el universo de metadatos existentes se optó por desarrollar las estrategias sobre un subconjunto de ellos presenten en la mayoría de los *papers* de investigación.

- Título
- Palabras Clave
- Idioma
- Resumen
- Autores

8.2 Flujo de trabajo para la Extracción de metadatos

Para facilitar la carga de contenidos digitales en el repositorio se propone modificar el flujo de carga estándar de la plataforma DSpace y diseñar una nueva etapa para la extracción automática de algunos metadatos. Esto permitirá ayudar al usuario en este proceso disminuyendo su trabajo y mejorando la cantidad y calidad de los metadatos cargados.

Para poder incluir el *framework* de extracción automática de metadatos, se reestructuró el flujo de carga, reordenando y modificando los pasos en el depósito de contenidos digitales. La ilustración 26 muestra los pasos propuestos para el nuevo flujo de carga. Las modificaciones principales consisten en:

1. Selección de la comunidad: En esta etapa se selecciona la comunidad contenedora del contenido.
2. Elección de la colección: Se selecciona la colección en la cual será depositado el contenido.
3. Aceptación de la licencia institucional: Este paso se agrega al principio del workflow de carga ya que en el caso de que ésta no se acepte, el depósito se cancela y los otros pasos no son necesarios.

4. Carga de los archivos asociados al objeto: Se realiza antes de completar los formularios de descripción, de manera que los metadatos configurados para ser extraídos de forma automática puedan completarse con ayuda del *framework* de extracción.

5. Extracción automática y descripción de los metadatos: En este paso es donde se incorpora la nueva etapa para la extracción automática. Para lograr la extracción, se utiliza el archivo que fue cargando en la etapa anterior y se invocan a los servicios del *framework*. Los metadatos que fueron obtenidos por *framework* de extracción son presentados al usuario para su validación en los campos correspondientes del formulario. Permitiendo ser modificados y ampliados. En esta etapa también se le permite al usuario completar el resto de los metadatos que no recuperados de forma automática.

6. Verificación de los metadatos: Se presentan los metadatos recuperados automáticamente y los generados por el usuario, permitiendo la modificación de cada uno de ellos.

Ilustración 26 Flujo de carga modificado

8.3 Arquitectura

El desarrollo de la herramienta se basó en una arquitectura de sistemas moderna, de acceso amplio y orientado a servicios utilizando el concepto de Arquitectura Orientada a Servicios de Clientes (*Service Oriented Architecture*, SOA).

SOA es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos de negocios. La utilización de una arquitectura SOA permite que los sistemas sean altamente escalables brindando a su vez una forma bien definida de exposición e invocación de servicios.

Los *framework*s tienen como objetivo ofrecer una funcionalidad definida, auto contenida, siendo construidos usando patrones de diseño, y su característica principal es su alta cohesión y bajo acoplamiento. Es en esta sección donde se describe el *framework* de extracción. En particular, para la construcción del *framework*, se utilizó una arquitectura de n-capas distribuida, utilizando el lenguaje Java como la tecnología dominante para el soporte de los servicios.

La separación en capas propuesta permite aislar los distintos aspectos del desarrollo de la herramienta, trabajando de esta manera las capas inferiores proveen servicios a su capa inmediata superior. Esta forma de separación, además de lo descripto y mediante el uso de interfaces permite que las capas puedan reemplazarse teniendo un costo e impacto mínimo respecto de las capas con las que se comunica.

Las capas en las que se separó la arquitectura del *framework*, ordenadas desde la capa superior a la inferior fueron las siguientes:

- Servicios de Negocios
- Modelo de Dominio
- Algorítmica

En la ilustración 27 se describe la arquitectura definida para el *framework* en interacción con la arquitectura de alto nivel de DSpace.

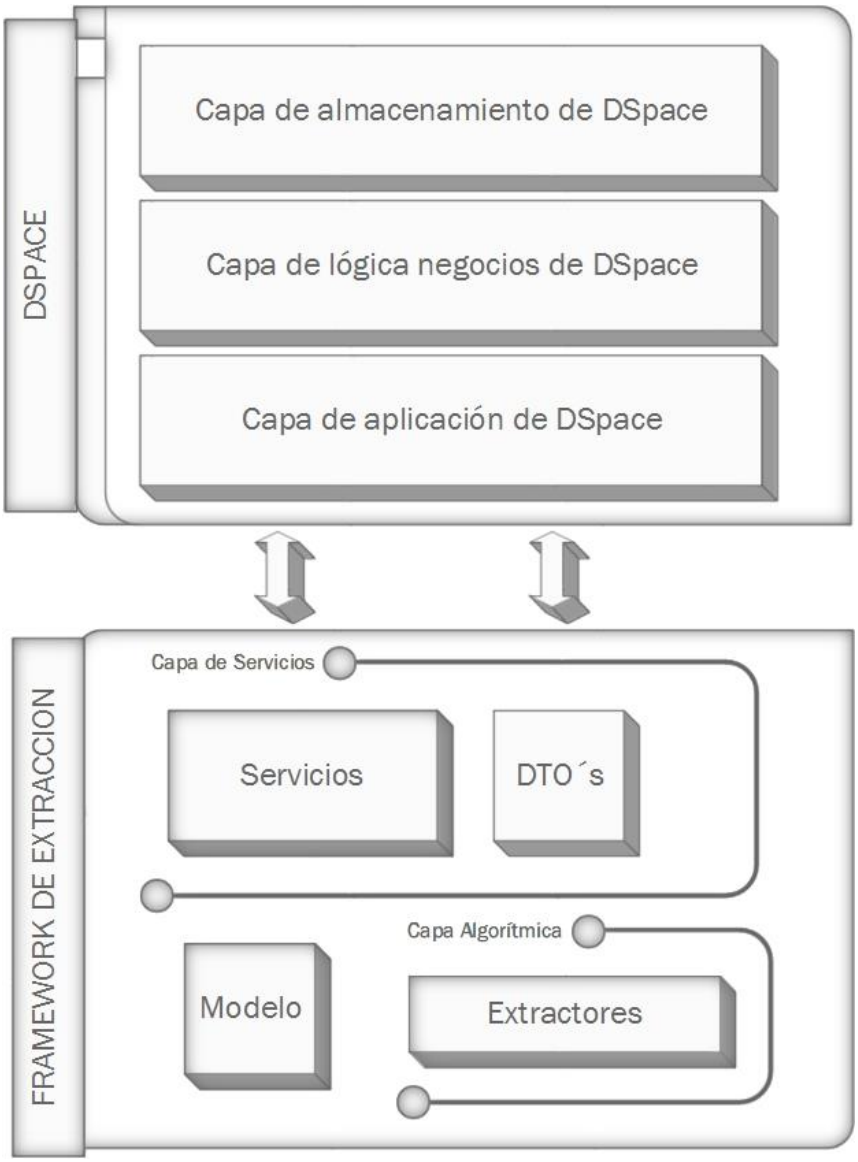


Ilustración 27 Arquitectura en capas de alto nivel

8.3.1 Capa de Servicios

Esta capa es central en la arquitectura de la aplicación, porque es la capa que lleva a caracterizarse como *Service Oriented Architecture* (SOA). Permite abstraer todas las complejidades propias del dominio del sistema y brindársela como funciones de grano grueso para que sean utilizadas por la capa de presentación desde los controladores de XMLUI.

Cada servicio de esta capa se encarga de integrar cada una de las capas inferiores y delegar al modelo de dominio la responsabilidad de la lógica de negocios.

Adicionalmente en esta capa se hace un uso extensivo de patrones arquitecturales como Fachada (*Session Facade*) y Servicio de Aplicaciones (*Application Service*), donde Fachada tiene como objetivo el exponer ante los usuarios componentes de la capa de negocio evitando que accedan directamente a los objetos de esta capa. Mientras que el patrón Aplicación de Servicio cumple la finalidad de centralizar la lógica de negocio y apartarla de los objetos de negocio. Otro patrón que se utilizó fue el DTO, que se aplica creando objetos que conducen los datos a través de XMLUI y el *framework* de extracción, con el objetivo principal de minimizar el acoplamiento entre la capa de presentación y el modelo de dominio.
El uso de estos patrones es para realizar una correcta organización de la lógica y su buena performance de acceso.

Como muestra la ilustración 28 **DspaceMetadataExtractoService** expone los métodos que pueden ser usados por XMLUI. Su implementación se realiza en la clase **DspaceMetadataExtractoServiceImpl**, cada método se encarga de comunicarse con la estrategia correspondiente, esperar el resultado y encapsularlo dentro del DTO. Existen dos tipos de DTO, el **SimpleValueResultDto** el cual es utilizado para retornar valores simples como pueden ser un título o un idioma y el **CompositeValueResultDto** que puede retornar valores múltiples, como por ejemplo una lista de keywords o una lista de autores. Ambos DTO poseen campos para indicar el nombre del metadato extraído así como también otro campo para indicar errores en la extracción.

```
public interface DspaceMetadataExtractoService {
        /**
         * Método que extrae el título de un PDF presente en el texto.
         * @param file archivo PDF
         * @return
         * @throws LangDetectException
         */
        public SimpleValueResultDto extractTitle (File file);

        /**
         * Método que extrae las keyword de un PDF presentes en el texto.
         * @param file archivo PDF
         * @return Lista de keywords
         */
        public CompositeValueResultDto  extractKeywords (File file);

        /**
         * Metodo que extrae el Abstract de un PDF
         * @param file archivo PDF
         * @return el abstract de un documento
         */
        public SimpleValueResultDto  extractAbstract (File file);

        /**
         * Método que determina el idioma de un PDF
         * @param file archivo PDF
         * @return idioma del PDF
         * @throws LangDetectException
         */
        public SimpleValueResultDto  extractLanguage(File file) throws  LangDetectException;

        /**
         * Método que extrae los autores de un PDF contenido en el texto
         * @param file
         * @return lista de autores
         */
        public CompositeValueResultDto  extractAuthor(File file);
}
```

Ilustración 28 Interfaz de los servicios de los extractores

```
@Service
public class DspaceMetadataExtractoServiceImpl implements DspaceMetadataExtractoService {
    @Override
    public SimpleValueResultDto extractTitle (File file) {
        String title = "";
        SimpleValueResultDto simpleValueResultDto = new SimpleValueResultDto("", "", "");
        simpleValueResultDto.setMetadataName(Constants.DC_TITLE);
        try {
            title = TitleEngine.extractTitle(file);
            simpleValueResultDto.setValue(title);
        } catch (LangDetectException e) {
            simpleValueResultDto.setError(e.getMessage());
        }
        return simpleValueResultDto;
    }

    @Override
    public CompositeValueResultDto  extractKeywords (File file){
        List<String> keywords= new ArrayList<String>();
        CompositeValueResultDto    compositeValueResultDto    =    new    CompositeValueResultDto(new
        ArrayList<String>(), "","");
        compositeValueResultDto.setMetadataName(Constants.DC_SUBJECT);
        try {
            keywords = KeywordEngine.extractKeywordsFromFile(file);
            compositeValueResultDto.setValue(keywords);
        } catch (Exception e) {
            compositeValueResultDto.setError(e.getMessage());
        }
        return compositeValueResultDto;
    }

    @Override
    public SimpleValueResultDto extractAbstract(File file) {
        String resume = "";
        SimpleValueResultDto simpleValueResultDto = new SimpleValueResultDto("", "", "");
        simpleValueResultDto.setMetadataName(Constants.DC_DESCRIPTION);
        try {
            resume = AbstractEngine.extractAbstract(file);
            simpleValueResultDto.setValue(resume);
        } catch (Exception e) {
            simpleValueResultDto.setError(e.getMessage());
        }
        return simpleValueResultDto;
    }
```

Ilustración 29 Implementación de los servicios de los extractores

```
@Override
public SimpleValueResultDto extractLanguage(File file) throws LangDetectException {
        String language = "";
        SimpleValueResultDto simpleValueResultDto = new SimpleValueResultDto("", "", "");
        simpleValueResultDto.setMetadataName(Constants.DC_LANGUAGE);
        try {
                language = LanguageEngine.detectLang(file);
                simpleValueResultDto.setValue(language);
        } catch (IOException e) {
                simpleValueResultDto.setError(e.getMessage());
        }
        return simpleValueResultDto;

}

@Override
public CompositeValueResultDto extractAuthor(File file) {
        AuthorEngine authorEngine = new AuthorEngine();
        CompositeValueResultDto compositeValueResultDto = new      CompositeValueResultDto(new
        ArrayList<String>(), "","");
         compositeValueResultDto.setMetadataName(Constants.DC_CONTRIBUTOR);
        List<String> authors = new ArrayList<String>();
        try {
                authors = authorEngine.extractAuthors(file);
                compositeValueResultDto.setValue(authors);
        } catch (Exception e) {
                compositeValueResultDto.setError(e.getMessage());
        }
        return compositeValueResultDto;

}
```

Ilustración 30 Implementación de los servicios de los extractores (continuación)

8.3.2 Capa de Modelo de Dominio

El modelo de dominio contiene una red de objetos interconectados donde cada objeto encapsula la estructura y el comportamiento de cada estrategia de extracción. Como su nombre lo indica esta capa contiene las entidades del dominio. En la ilustración 31 se muestra el diagrama de clases del *framework* de extracción.

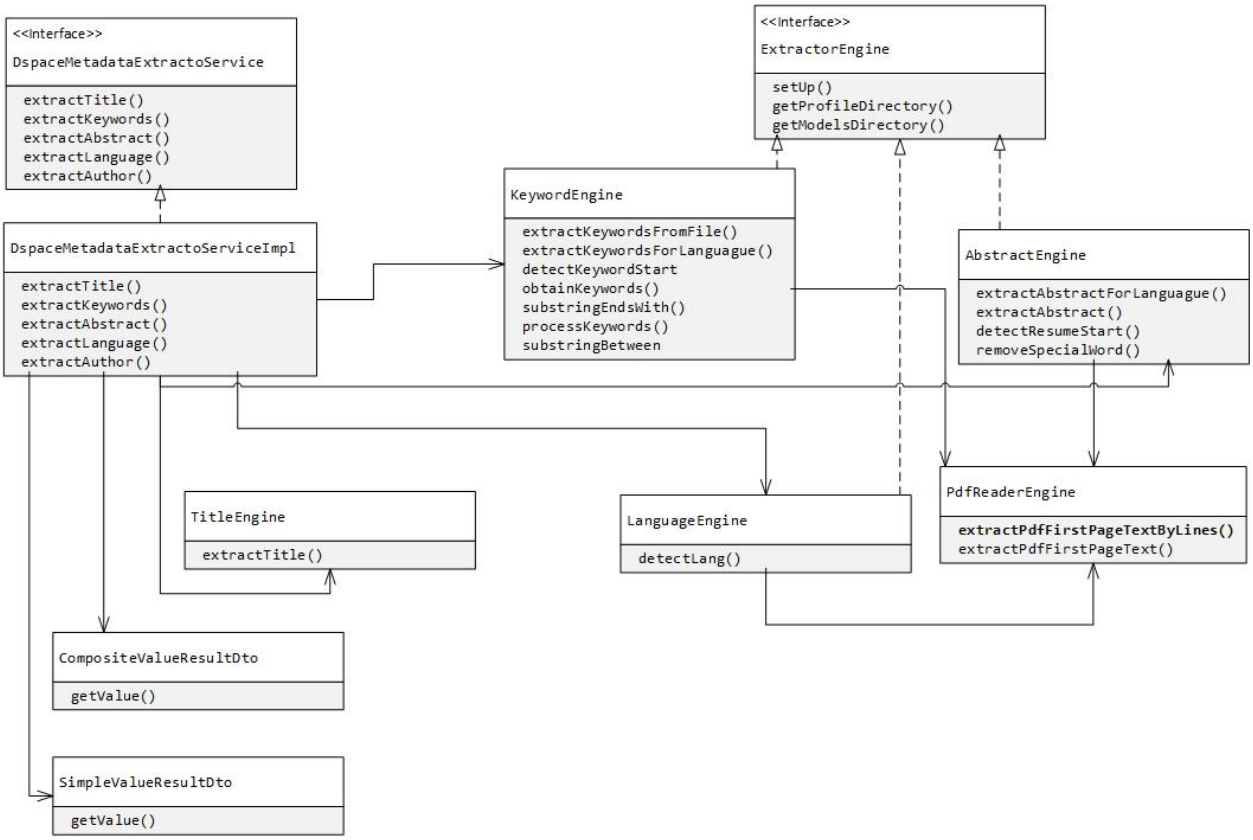


Ilustración 31 Diagrama de clases del framework

8.3.3 Capa Algorítmica

La funcionalidad que provee esta capa es la de llevar a cabo las estrategias de detección de metadatos sobre los archivos. Su implementación fue realizada completamente en lenguaje Java. Las diferentes estrategias fueron agrupadas en clases, cada una concentra la lógica para la detección, recuperación y parseo de un metadato. Esta separación se debe principalmente a una correcta división modular de las distintas funcionalidades.

8.3.3.1 Estrategias de extracción

Dado a que los metadatos de interés no están presentes como parte del archivo físico, fue necesario utilizar distintas técnicas de procesamiento de texto y de lenguaje natural para detectarlos.

Cada metadato a extraer está asociado a una estrategia, la cual implementa la forma de identificar el metadato dentro del documento, procesarlo y entregar los valores a la capa de servicios para luego ser retornados a DSpace. Cada estrategia posee una forma diferente de detección, según sea el metadato. En algunos casos se hace uso de librerías para que junto con la lógica desarrollada puedan realizar la extracción. A continuación se describen las distintas estrategias desarrolladas.

TitleEngine

Esta estrategia se concentra en extraer el título del *paper*, para lograr la extracción se hizo uso de Docear's PDF Inspector [11], una librería en Java que extrae títulos de un PDF no presentes desde los metadatos del PDF, sino de su texto completo. Más precisamente, Docear's PDF Inspector extrae el texto completo de la primera página de un PDF y busca el texto más grande en el tercio superior de la página. Este texto se devuelve como título. Por supuesto, esto no siempre retorna el título correcto (por ejemplo, se detectó en papers antiguos que el tamaño de la fuente de los autores era más grande que el título de un artículo), pero en la mayoría de los casos obtendrá el correcto.

La clase TitleEngine se describe a continuación:

```
public class TitleEngine  {
        /**
         * Método que extrae el título de un PDF
         * @param file archivo PDF
         * @return titulo del PDF
         * @throws LangDetectException
         */
    public static String extractTitle(File file) throws LangDetectException {

        PdfDataExtractor extractor = new PdfDataExtractor(file);
                try {
                        String title = extractor.extractTitle();
                        return title;
                } catch (IOException e) {
                        e.printStackTrace();
                }
                return null;
    }
}
```

Ilustración 32 Codificación de la estrategia del extractor de títulos

Características de Docear

- Extrae títulos de los archivos PDF con buena precisión (~ 70%) y con tiempo de ejecución excelente (pocos milisegundos por PDF en modo batch).

- Se puede utilizar como una librería de Java (otras herramientas como gestores de referencias pueden ser integrados fácilmente a Docear's PDF Inspector para extraer títulos de los archivos PDF).

- Se puede usar como una aplicación por línea de comandos (devuelve el título de los archivos PDF en la línea de comandos).

- Se puede utilizar en el modo por lotes (almacena los títulos extraídos en un archivo CSV)

Lee todas las versiones de PDF (otras herramientas como SciPlore Xtract o ParsCit están utilizando PDFBox para el procesamiento de los archivos PDF Sin embargo, PDFBox a veces tiene problemas para extraer texto de archivos PDF no son 100% compatibles con el estándar PDF -. PDF Inspector de Docear se basa en jPod, que es más tolerante)

Escrito completamente en JAVA 1.6. Por lo tanto, PDF Inspector de Docear funciona en cualquier sistema operativo, incluyendo Windows, Linux, y Mac OS, sin otras herramientas necesarias.

Totalmente independiente de otras herramientas - sólo necesita PDF Inspector de Docear, (por ejemplo SciPlore Xtract requiere instalar pdftohtml).

Publicado bajo la Licencia Pública General de GNU (GPL) 2 o posterior, lo que significa que es de uso completamente gratuito y su código fuente puede ser descargado y modificado por cualquier persona.

AbstractEngine

Esta estrategia es la encargada de extraer el “Abstract” (resumen) de un paper. AbstractEngine recibe el archivo a procesar y realiza las siguientes etapas:

- PdfReaderEngine recupera las líneas del texto de la primera página.
- Se procede a la extracción del Abstract.
    - Se analiza cada línea en búsqueda de algún patrón que identifique el inicio el Abstract. Como por ejemplo: "resumen", "abstract", "resume".
    - Se recupera el Abstract, procesando el texto hasta encontrar el patrón de fin como puede ser un doble salto de línea o el inicio de las “palabras clave” (Keywords).
    - Retorna el Abstract al servicio DspaceMetadataExtractoService, el cual encapsula el resultado dentro del SimpleValueResultDto.

Las funciones principales de ésta clase son:

```
/**
        * Método que detecta y extrae el abstract de una lista de líneas de texto
         * @param text
         * @return
         * @throws InvalidFormatException
         * @throws IOException
*/
private     static     String     extractAbstractMetadata(List<String>     text)     throws
InvalidFormatException, IOException {
               String initWord;
               boolean keyLoc = false;
               boolean endResume = false;
               int numberLine = 0;
               StringBuilder stringBuilder = new StringBuilder();
               String line;
               String newLine;
               while ((!keyLoc) && (numberLine < text.size())) {
                       // busco si en la línea esta la palabra clave
                       initWord = detectResumeStart(text.get(numberLine));
                       if (!initWord.isEmpty()) {
                               keyLoc = true;
                               // remuevo las palabras resume/abstract
                               line = removeSpecialWord(text.get(numberLine));
                               numberLine++;
                               while ((!endResume) && (numberLine < text.size())) {
                                       if (!line.equals("")) {
                                               stringBuilder.append(line);
                                       }
                                       line = text.get(numberLine);
                                       if     (line.equals("")     ||     startsWithText(line,
                                       othersPatterns)          ||          startsWithText(line,
                                       initKeywordsPatterns)) {
                                               newLine            =            detectWord(line,
initKeywordsPatterns);
                                               if            ((newLine.equals(""))            &&
                                               (!startsWithText(line, initKeywordsPatterns))) {
                                                       newLine         =         detectWord(line,
othersPatterns);
                                               }
                                               stringBuilder.append(newLine);
                                               endResume = true;
                                       } else {
                                               numberLine++;
                                       }
                               }
                       } else {
                               numberLine++;
                       }
```

Ilustración 33 Codificación de la estrategia del extractor del resumen

```
/**
* Método que determina si una sentencia empieza con alguno de los  identificadores para el
abstract
         *
         * @param sentence
         * @return
*/
        private static String detectResumeStart(String sentence) {
                for (String keyStart : initStringsPatterns) {
                                if         ((sentence.toLowerCase().contains(keyStart))         &&
                        (sentence.toLowerCase().startsWith(keyStart))) {
                                return keyStart;
                        }
                }
                return "";
```

Ilustración 34 Ilustración 32 Codificación de la estrategia del extractor del resumen (continuación)

LanguageEngine

LanguageEngine recibe el texto a analizar y con ayuda de la librería "Language Detection" de Cybozu Labs [12], procede a detectar el idioma. Esta librería provee un método que dadas las características de ortografía calcula las probabilidades de los diferentes idiomas. Esto se logra utilizando el algoritmo clasificador de bayes, una técnica de clasificación descriptiva y predictiva basada en la teoría de la probabilidad del análisis de Tomas Bayes la cual busca correlaciones entre atributos.

Inicialmente clasifica los documentos en categorías de lenguajes, como por ejemplo inglés, japonés, chino, etc. Luego actualiza las probabilidades de acuerdo a las características del texto dado.

$$p(C_k|X)^{(m+1)} \propto p(C_k|X)^{(m)} \cdot p(X_i|C_k)$$

Donde: C_k= categoría, X= documento, X_i= característica de documento.

Ilustración 35 Formula de probabilidad de características

Finaliza el proceso de detección si la probabilidad máxima (normalizada) es mayor a 0.99999. Este algoritmo tiene una precisión del 90%, y tiene bajas probabilidades de detectar lenguajes como el japonés, el chino tradicional, el ruso y el persa, debido al sesgo y el ruido que generan las características que poseen los mismos. Por esto la librería de Cybozu Labs implementó una mejora que añade un filtro de ruido y la normalización de caracteres. Primero se eliminan caracteres de idiomas independientes (cifras numéricas, símbolos, URL's y direcciones de correo electrónico), caracteres latinos en textos no latinos y latinos (acrónimos, nombres de personas, etc.). Para la normalización, clasifica frecuencias similares de texto y normaliza cada grupo en una representación determinada. Gracias a esto, esta librería, es capaz de detectar 49 lenguajes diferentes con una precisión de 99.8%.

Una vez detectado el idioma, el resultado es retornado al servicio **DspaceMetadataExtractoService**, el cual encapsula el resultado dentro del **SimpleValueResultDto**.

La clase LanguageEngine se describe a continuación:

```
public class LanguageEngine extends ExtractorEngine {
        /**
         * Método que determina el idioma del texto
         * @param text
         * @return
         * @throws LangDetectException
         * @throws IOException
         */
    public static String detectLang(File file) throws LangDetectException, IOException {

        String text = PdfReaderEngine.extractPdfFirstPageText(file);
        if (DetectorFactory.getLangList().size()<=0)
                DetectorFactory.loadProfile(getProfileDirectory());
         Detector detector = DetectorFactory.create();
         detector.append(text);
         return detector.detect();
    }
}
```

Ilustración 36 Codificación del extractor para la detección de idiomas

KeywordEngine

Esta estrategia se inicia detectando el idioma del texto de entrada y junto con OpenNLP proceden a extraer las palabras clave (keywords). Para determinar las sentencias que luego son procesadas en búsqueda del identificador de las palabras claves, se utilizó OpenNLP con dos modelos en-sent.bin y es-sent.bin, el primero para el idioma ingles y el segundo para el español.

- PdfReaderEngine recupera el texto de la primera página.
- LanguageEngine determina el idioma del texto.
- Se procede a la extracción de las keywords (palabras clave)
    - En base al idioma, se separa el texto en sentencias.
    - Se analiza cada sentencia en búsqueda de algún patrón que identifique el inicio de la especificación de las keywords. Como por ejemplo: “palabras claves:", "palabras clave:", "palabras claves", "palabras clave", "keywords:.
    - Se recuperan las keywords, hasta encontrar el patrón de fin como puede ser un doble salto de línea o el inicio de la “Introducción 1”.
    - Se procesa el resultado generando una lista de keywords, eliminando caracteres especiales, saltos de línea, etc.
- Retorna la lista de palabras clave al servicio **DspaceMetadataExtractoService**, el cual encapsula el resultado dentro del **CompositeValueResultDto**.

Las funciones principales de esta clase son:

```
/**
* Método que extrae las palabras clave de un archivo PDF. Se extraen las palabras clave
suponiendo que están en la primera página.
         *
         * @param file archivo PDF
         * @return lista de palabras clave
         * @throws Exception
 */
        public static List<String> extractKeywordsFromFile(File file) throws Exception {

                String text = PdfReaderEngine.extractPdfFirstPageText(file);

                if (LanguageEngine.detectLang(file).equals("es")) {
                        return extractKeywordsForLanguague(text, "es-sent.bin");
                }
                if (LanguageEngine.detectLang(file).equals("en")) {
                        return extractKeywordsForLanguague(text, "en-sent.bin");
                }
                return null;
        }
```

Ilustración 37 Codificación del extractor de palabras clave

```
/**
         * Método que dependiendo del idioma procesa el texto en búsqueda de palabras clave
         * @param sentence
         * @param sentBin
         * @return lista de palabras clave
         * @throws FileNotFoundException
         * @throws IOException
         * @throws InvalidFormatException
         */
        private static List<String> extractKeywordsForLanguague(String sentence, String
        sentBin) throws FileNotFoundException, IOException,
                        InvalidFormatException {

                InputStream modelSentence = new FileInputStream(new File(getModelsDirectory(),
sentBin));
                SentenceModel model = new SentenceModel(modelSentence);
                SentenceDetectorME sentenceDetector = new SentenceDetectorME(model);
                String sentences[] = sentenceDetector.sentDetect(sentence);
                modelSentence.close();

                String initWord;
                boolean keyLoc = false,endKeys = false;
                int si = 0;
                List<String> keywords = new ArrayList<>();
                while ((!keyLoc) && (si < sentences.length)) {
                        initWord = detectKeywordStart(sentences[si]);
                        if (!initWord.isEmpty()) {
                                while ((!endKeys) && (si < sentences.length)) {
                                        keywords = obtainKeywords(sentences[si].toLowerCase(),
initWord);
                                        if (keywords.size()>0){
                                                endKeys=true;

                                        }
                                        else{
                                                si++;
                                        }
                                }
                                keyLoc=true;
                                }
                        si++;
                }
```

Ilustración 38 Codificación del extractor de palabras clave (continuación)

AuthorEngine

Para lograr la extracción de autores de un *paper* fue necesario contar con otra herramienta que posee OpenNlp, el nameFinder. Como se describió anteriormente esta herramienta puede extraer nombres de personas basándose en modelos pre entrenados. Al igual que en la estrategia de extracción de palabras claves se utilizaron dos modelos, un en inglés y otro en español.

El proceso de recuperación de autores contempla las siguientes etapas:

- PdfReaderEngine recupera el texto de la primera página.
- LanguageEngine determina el idioma del texto.
- Se procede a la extracción de los autores.
    - En base al idioma, se separa el texto en sentencias.

- Se separa cada sentencia en Tokens.
- NameFinder busca en los Tokens nombres de personas.

- Retorna la lista de autores al servicio **DspaceMetadataExtractoService**, el cual encapsula el resultado dentro del **CompositeValueResultDto**.

Las funciones principales de esta clase son:

```
/**
         * Método encargado de extraer nombres y apellidos de autores
         *
         * @param sentence
         *            texto a extraer los autores
         * @return lista de autores
         * @throws Exception
*/
        public List<String> extractAuthors(File file) throws Exception {
                String text = PdfReaderEngine.extractPdfFirstPageText(file);
                List<String> authors = new ArrayList<>();

                this.setModelPerson("es-ner-person.bin");
                this.setModelSentence("es-sent.bin");
                authors.addAll(this.extract(text));

                this.setModelPerson("en-ner-person.bin");
                this.setModelSentence("en-sent.bin");
                authors.addAll(this.extract(text));
                return authors;
        }
        private   List<String>   extract(String   sentence)   throws   FileNotFoundException,
        IOException, InvalidFormatException {
                InputStream modelSentence = new FileInputStream(new File(getModelsDirectory(),
                this.getModelSentence()));
                SentenceModel model = new SentenceModel(modelSentence);
                SentenceDetectorME sentenceDetector = new SentenceDetectorME(model);
                String sentences[] = sentenceDetector.sentDetect(sentence);
                TokenNameFinderModel     modelPerson     =     new     TokenNameFinderModel(new
                File(getModelsDirectory(), this.getModelPerson()));
                NameFinderME nameFinder = new NameFinderME(modelPerson);
                List<String> namedEntities = new ArrayList<String>();
                for (int si = 0; si < sentences.length; si++) {

                        List<TextAnnotation>          allTextAnnotations          =          new
ArrayList<TextAnnotation>();
                        String[] tokens = tokenizer.tokenize(sentences[si]);

                        Span[] spans = nameFinder.find(tokens);
                        double[] probs = nameFinder.probs(spans);
                        for (int ni = 0; ni < spans.length; ni++) {
                                allTextAnnotations.add(new  TextAnnotation("person",  spans[ni],
probs[ni]));
                        }
                        if (!allTextAnnotations.isEmpty()) {
                                removeConflicts(allTextAnnotations);
                                convertTextAnnotationsToNamedEntities(tokens,
allTextAnnotations, namedEntities);
                        }
                        nameFinder.clearAdaptiveData();
                }
                return namedEntities;
        }
```

Ilustración 39 Codificación del extractor de autores

PdfReaderEngine

PdfReaderEngine es la estrategia de recuperación de texto dentro un archivo PDF. Esta estrategia es utilizada como un colaborador por el resto de las estrategias, junto con la librería pdfbox [13] brindan la lógica necesaria para poder abrir un archivo PDF y recuperar el texto contenido en algunas de las páginas, áreas del mismo o recuperar las líneas que forman el PDF. Para lograr la extracción, la estrategia debe recibir el archivo desencriptado.

Las funciones de esta clase son:

```
/**
         * Método que extrae el texto de la primera página de un PDF
         *
         * @param file
         * @return
         * @throws IOException
         */
        public static String extractPdfFirstPageText(File file) throws IOException {
                PDDocument pdf = PDDocument.load(file);
                PDFTextStripper stripper = new PDFTextStripper("UTF-8");
                stripper.setStartPage(1);
                stripper.setEndPage(1);

                String plainText = stripper.getText(pdf);
                pdf.close();
                return plainText;
        }
        /**
         * Método que extrae las líneas del texto dela primera página de PDF
         * @param file
         * @return
         * @throws IOException
         */
        public static List<String> extractPdfFirstPageTextByLines(File file) throws IOException
{
                PDDocument pdf = PDDocument.load(file);
                PDFTextStripper stripper = new PDFTextStripper("UTF-8");
                stripper.setStartPage(1);
                stripper.setEndPage(1);
                String plainText = stripper.getText(pdf);
                List<String> ans = Arrays.asList(plainText.split("\r\n"));
                pdf.close();
                return ans;
        }
```

Ilustración 40 Codificación del extractor de texto

8.3.4 Tecnologías adicionales

Además de las tecnologías utilizadas en cada una de las capas anteriormente descriptas, la herramienta también utiliza:

PDFBOX:

PDFBox es una librería Java de código abierto que permite trabajar con documentos en formato PDF. Estas librerías permiten acciones relativas a la creación, manipulación y extracción del contenido de documentos en formato PDF.

Algunas de las funcionalidades concretas que ofrece esta librería son las siguientes:

- Extraer el texto contenido en archivos PDF.
- Unir ficheros PDF.
- Encriptación/desencriptación de documentos PDF.
- Crear un PDF a partir de un fichero de texto.
- Crear imágenes a partir de ficheros PDF.

Además, PDFBox incluye varias utilidades para línea de comandos que permiten realizar algunas de estas acciones. Este tutorial se centrará en la utilización de dichas utilidades, por lo que no son necesarios conocimientos profundos de Java. No será necesario realizar ningún programa en Java y solamente serán precisas nociones básicas sobre ejecución de programas Java en consola.

Docear's PDF Inspector:
Como se mencionó en la estrategia para la extracción de títulos se hace uso de las librerías de Docear's PDF Inspector. Dado a que Docear's PDFInspector provee de un archivo ejecutable .jar para la extracción de títulos por línea de comandos, fue necesario incorpora el proyecto Docear's como dependencia al proyecto *dspace-extractor* y así poder hacer uso de los métodos que dispone en forma correcta desde la estrategia de extracción de títulos.

8.4 Integración con XMLUI

XMLUI puede hacer uso del *framework* de extracción, gracias a la posibilidad que dispone DSpace de crear nuevos procesos de carga (*submission step*) y poder integrarlo a la capa de presentación de XMLUI.

En esta sección se detallan los pasos necesarios para contar con la extracción automática de metadatos, partiendo por la configuración de dependencias entre los proyectos XMLUI y dspace-extractor, continuando por la modificación del proceso de carga y finalizando con la descripción de la clase encargada de utilizar cada extractor (AutomaticExtractStep).

8.4.1 Arquitectura de alto nivel

Aquí se describen los componentes principales de DSpace y el modo en que interactúan entre sí para satisfacer cada uno de los requisitos. Como se muestra la ilustración 41, el proyecto XMLUI conoce al *framework* de extracción dspace-extractor, y este a su vez cuenta con el proyecto para la extracción de títulos dspace-docenar-inspector.

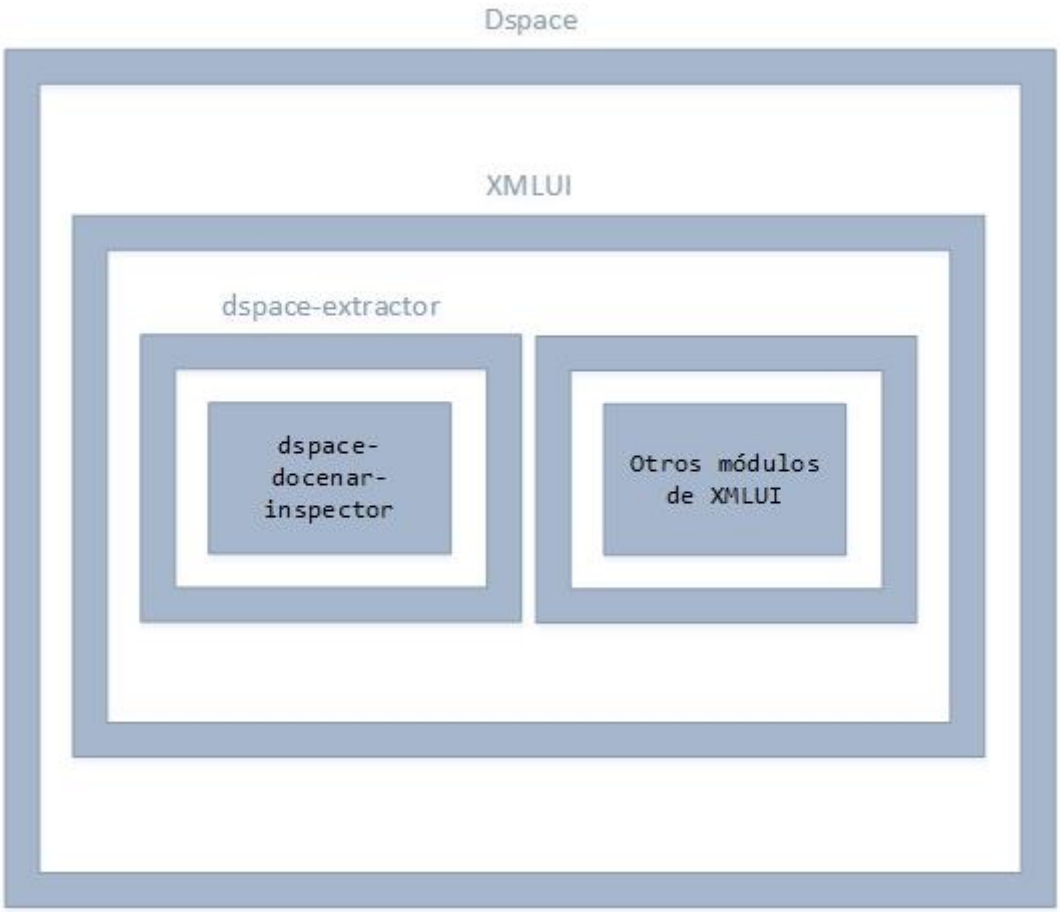


Ilustración 41 Arquitectura de alto nivel

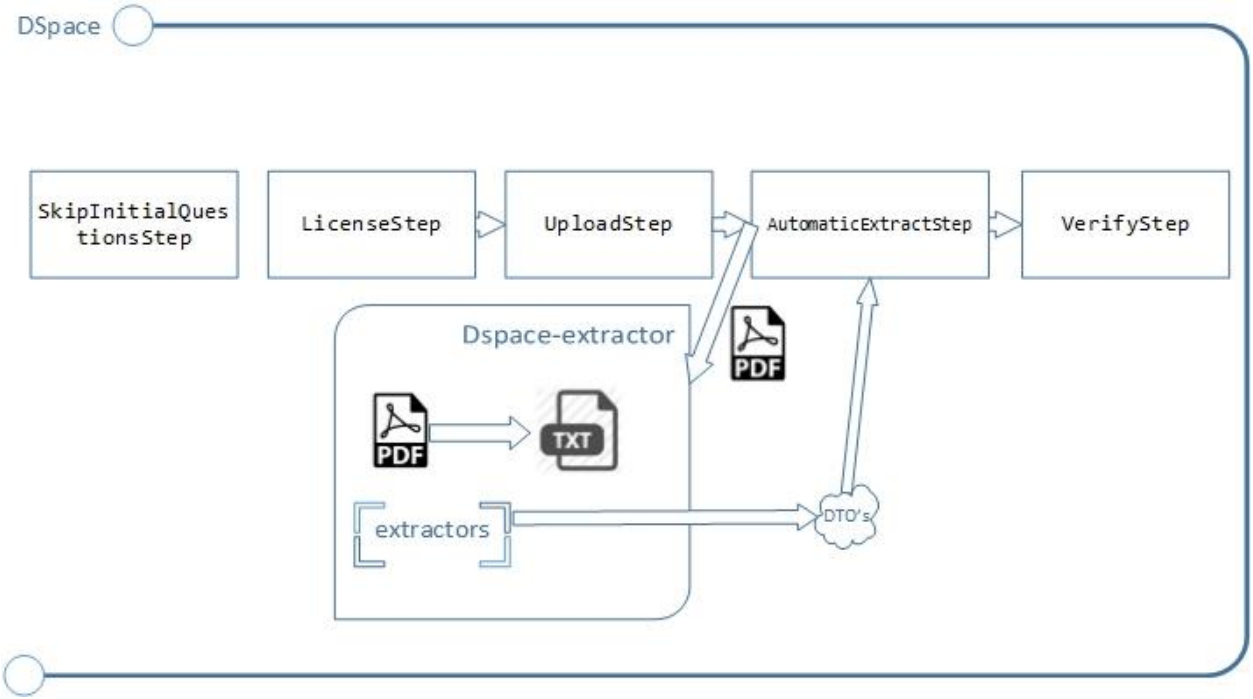


Ilustración 42 Diagrama de interacción con el framework de extracción

8.4.2 Configuración del proyecto como dependencia

Para poder conectar XMLUI con el *framework* se configuró desde el POM *master* de Dspace la dependencia con el proyecto de extracción (*dspace-extractor*). Para lograrlo fue necesario el uso de un *framework* Maven [Maven 2012]. El *framework* Maven fue utilizado en las capas anteriormente descriptas como forma de organización, enumerando y gestionando las distintas bibliotecas de software libre que fueron necesarias para desarrollar algunas de las características de la herramienta, ejemplo de esto son las bibliotecas para leer un PDF (PDFBOX). En esta capa, además de la función que cumplió en las capas previas fue necesario realizar una compilación e instalación del *framework* de extracción debido a que para que XMLUI pueda utilizar sus servicios, este debe ser parte de su repositorio local.

```
<dependency>
    <groupId>org.dspace</groupId>
    <artifactId>dspace-api</artifactId>
</dependency>
<dependency>
    <groupId>org.dspace</groupId>
    <artifactId>dspace-services</artifactId>
</dependency>

<dependency>
    <groupId>org.dspace.tesis</groupId>
    <artifactId>dspace-extractor</artifactId>
    <version>0.0.1</version>
</dependency>
```

Ilustración 43 Integración de dspace-extractor a XMLUI (pom.xml del proyecto XMLUI)

8.4.3 Configuración del camino de carga (submission step)

Como se mencionó anteriormente fue necesario modificar el proceso de carga para extraer los metadatos automáticamente. Para esto se creó un nuevo proceso llamado *automaticExtraction* en el cual se define el orden necesario para subir un contenido al repositorio, extraer los metadatos, revisarlos, extenderlos y corregirlos en caso que sea necesario.

La especificación del nuevo proceso de carga es el siguiente:

```
<submission-map>
        <name-map collection-handle="default" submission-name="automaticExtraction" />
</submission-map>

<submission-process name="automaticExtraction">
        <step>
                <processing-class>org.dspace.submit.step.SkipInitialQuestionsStep</processing-class>
        </step>

        <!--Step 1 will be to select a Creative Commons License -->

        <step>
                <heading>submit.progressbar.license</heading>
                <processing-class>org.dspace.submit.step.LicenseStep</processing-class>
                <jspui-binding>org.dspace.app.webui.submit.step.JSPLicenseStep</jspui-binding>
                <xmlui-binding>
                                        org.dspace.app.xmlui.aspect.submission.submit.LicenseStep
                        </xmlui-binding>
                <workflow-editable>false</workflow-editable>
        </step>

        <!--Step 2 will be to Upload the item -->

        <step>
                <heading>submit.progressbar.upload</heading>
                <processing-class>org.dspace.submit.step.UploadStep</processing-class>
                <jspui-binding>org.dspace.app.webui.submit.step.JSPUploadStep</jspui-binding>
                <xmlui-binding>
                                        org.dspace.app.xmlui.aspect.submission.submit.UploadStep
                        </xmlui-binding>
                <workflow-editable>true</workflow-editable>
        </step>

        <!--Step 2 will be to Describe the item. -->

        <step>
                <heading>submit.progressbar.describe</heading>
                <processing-class>org.dspace.submit.step.DescribeStep</processing-class>
                <jspui-binding>
                                        org.dspace.app.webui.submit.step.JSPDescribeStep
                        </jspui-binding>
                <xmlui-binding>
                                        org.dspace.app.xmlui.aspect.submission.submit.AutomaticExtractStep
                        </xmlui-binding>
                <workflow-editable>true</workflow-editable>
        </step>

        <!--Step 3 will be to Verify/Review everything -->

        <step>
                <heading>submit.progressbar.verify</heading>
                <processing-class>org.dspace.submit.step.VerifyStep</processing-class>
                        <jspui-binding>org.dspace.app.webui.submit.step.JSPVerifyStep</jspui-binding>
                <xmlui-binding>
                                        org.dspace.app.xmlui.aspect.submission.submit.ReviewStep
                        </xmlui-binding>
                <workflow-editable>true</workflow-editable>
        </step>
</submission-process>
```

Ilustración 44 Definición del workflow de carga

8.4.4 Configuración de los extractores por campos (fields)

Como se mencionó anteriormente, los formularios para la carga de contenidos son configurados desde el archivo *input-forms.xml*. A cada campo (*field*) del formulario se le debe indicar si se opta por la extracción automática o la carga manual, para esto se extendió la definición de los campos añadiendo una nueva propiedad <automatic-extract>, la cual puede tomar dos valores booleanos, *true* o *false*.

<automatic-extract>**true**</automatic-extract> indica que el campo utilizará el extractor automático.

<automatic-extract>**false**</automatic-extract> indica que el campo se cargará de forma manual.

```
<field>
                <dc-schema>dc</dc-schema>
                <dc-element>contributor</dc-element>
                <dc-qualifier>author</dc-qualifier>
                <repeatable>true</repeatable>
                <label>Authors</label>
                <input-type>name</input-type>
                <hint>Enter the names of the authors of this item below.</hint>
                <required></required>
                <automatic-extract>true</automatic-extract>
</field>
```

Ilustración 45 Configuración del campo autor para la extración automática

Configuración para el campo Contributor (autor) con extracción automática:

8.4.5 Especificación de la etapa de extracción automática

Como podemos ver en la ilustración 46, se incorpora para XMLUI la clase **AutomaticExtractStep,** contenedora de la lógica capaz de procesar el archivo PDF. Esta es la responsable de invocar a los extractores y persistir el ítem junto a los metadatos extraídos.

```
        <step>
                <heading>submit.progressbar.describe</heading>
                <processing-class>org.dspace.submit.step.DescribeStep</processing-class>
                <jspui-binding>org.dspace.app.webui.submit.step.JSPDescribeStep</jspui-binding>
                <xmlui-binding>
                        org.dspace.app.xmlui.aspect.submission.submit.AutomaticExtractStep
                </xmlui-binding>
                <workflow-editable>true</workflow-editable>
        </step>
```

Ilustración 46 Configuración de la etapa de extracción automática

Esta clase, además de contener la lógica para hacer uso de los extractores, tiene la responsabilidad de brindar a la vista de XMLUI los elementos necesarios para la visualización de los campos y la validación de los mismos.

El funcionamiento **AutomaticExtractStep** es el siguiente:

- Se recupera el Item creado en los pasos anteriores del *workflow*. Este contiene el archivo que fue cargado en el **UploadStep.**
- Se recuperan los inputs para el formulario de carga, que fueron definidos en el archivo inputs-forms.xml.
- Por cada input, dependiendo del calificador (*qualifier*) de cada input y si se configuró el input para ser cargado automáticamente, se llama al método correspondiente del servicio de extracción con el archivo PDF que fue incorporado a DSpace en la etapa anterior del *workflow*. Una vez que se obtienen los resultados, los metadatos son guardados en el item.
- Se finaliza la etapa de extracción automática.

8.4.6 Pantallas de carga en XMLUI

Luego de describir el desarrollo y las modificaciones necesarias para utilizar el *framework* de extracción, se mostraran los pasos para la carga de un contenido.

Una vez indicada la Comunidad y la Colección que tendrá lugar el contenido se procede con las siguientes etapas del *workflow*:

Se acepta la licencia de distribución:

Ilustración 47 Aceptación de licencia de DSpace

Se selecciona el archivo a subir:

Ilustración 48 Selección de carga del archivo en DSpace

Se visualizan los metadatos extraídos automáticamente:

Ilustración 49 Visualización de los metadatos extraídos en DSpace

Se visualiza el ítem completo:

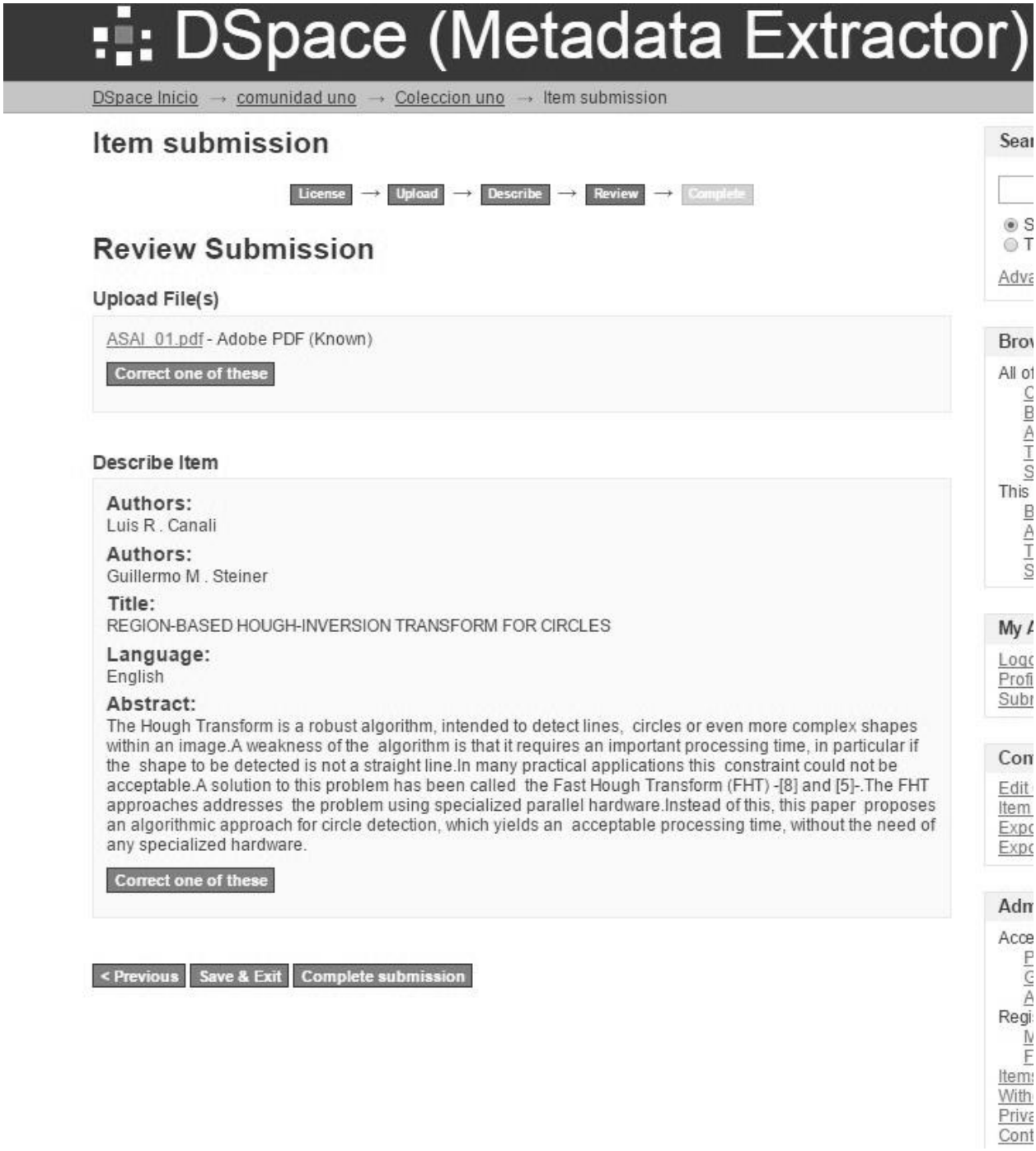


Ilustración 50 Visualización del ítem con sus metadatos completos en DSpace

La carga se completa:

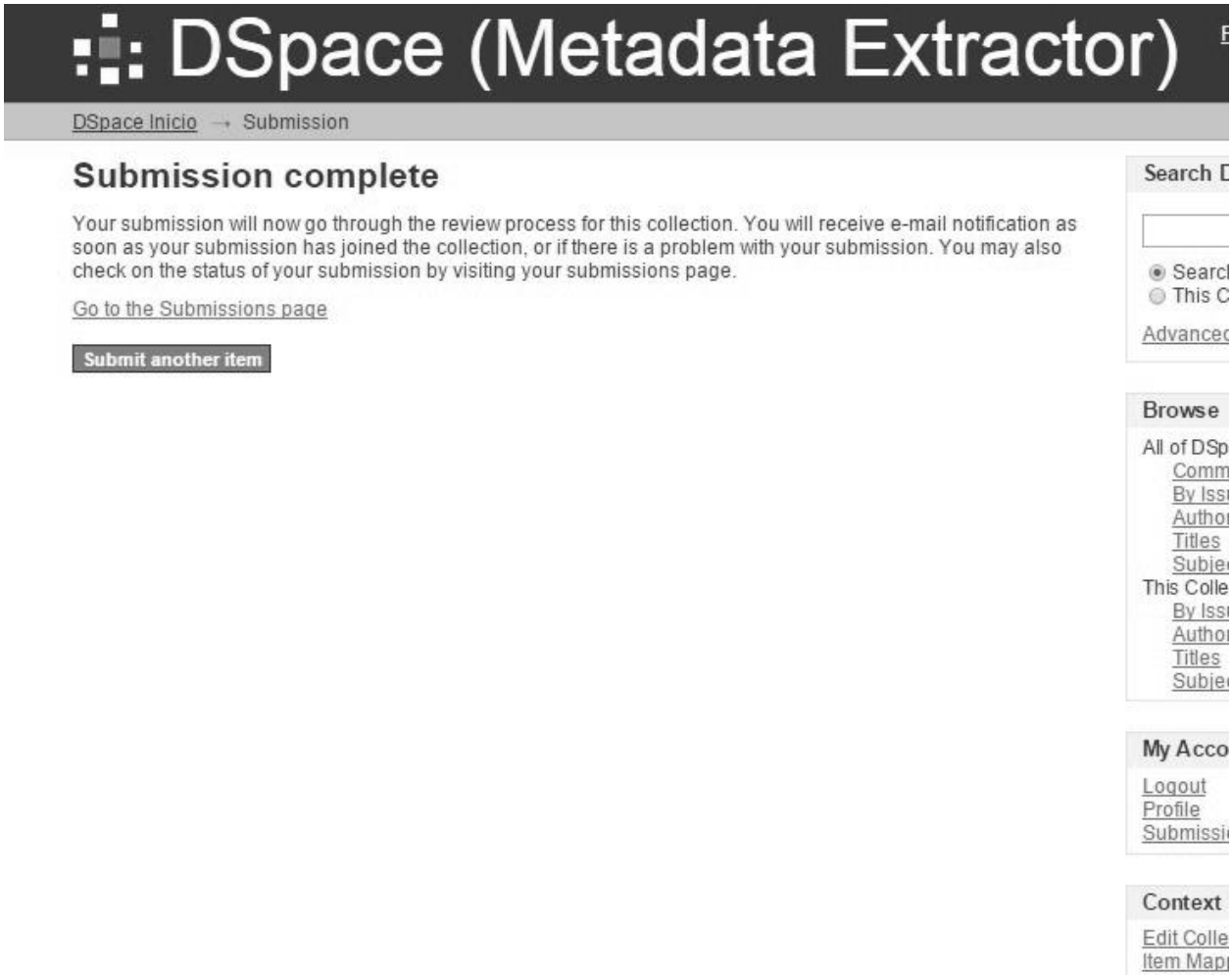


Ilustración 51 Completitud de la carga en DSpace

8.5 Consideraciones

En esta sección se describen las consideraciones que fueron necesarias tomar para reducir el universo de los tipos de archivos e idiomas que soportan los repositorios, así como también no procesar documentos en los cuales se encuentra algún tipo de seguridad que impida el procesamiento del documento.

8.5.1 Formato de los documentos soportados

Seleccionando un formato específico podemos reducir el espacio del problema a uno más manejable. Más específicamente, se espera que la herramienta pueda manejar para los contenidos cargados en el repositorio, el formato PDF (cuyo uso es adoptado ampliamente en todos los repositorios, bibliotecas y archivos digitales).

8.5.2 Idioma de los documentos

El *framework* soporta el idioma inglés y español.

8.5.3 Seguridad de los documentos

Con el fin de no violar la seguridad de los archivos en los que el autor aplicó algún tipo de restricción, el *framework* trabaja sobre documentos PDF no encriptados y sin passwords.

8.6 Pruebas realizadas

En esta sección, se describe la experimentación realizada y los resultados obtenidos en la evaluación de la herramienta desarrollada. Comenzamos con una descripción de los documentos utilizados como corpus y terminamos con un resumen de los resultados obtenidos.

Uso de las métricas de “precisión” y “recuperación” para evaluar las estrategias de extracción

Para evaluar la lista de metadatos recuperados por cada extractor, se va a comparar esta lista contra la lista de metadatos generada manualmente. Las medidas usadas para la evaluación vienen del área de recuperación de información las cuales son: la “precisión” y la “recuperación”.

La “precisión” es una métrica utilizada para evaluar la capacidad que tiene el método de búsqueda para no obtener documentos no pertinentes. Es el cociente entre el total de documentos pertinentes obtenidos y el de documentos obtenidos. Por lo tanto, puede interpretarse como la probabilidad de que un documento obtenido sea pertinente **[69]**. En nuestro caso, la “precisión” es la proporción de SÓLO los conceptos relevantes recuperados.

La “recuperación” es una métrica utilizada para evaluar la capacidad que tiene el método de búsqueda para obtener documentos pertinentes, es decir, considerados útiles por quien hizo la consulta. Es el cociente entre el total de documentos pertinentes obtenidos y el de documentos pertinentes de la base. Por lo tanto, puede interpretarse como la probabilidad de que un documento pertinente sea obtenido. En nuestro caso, la “recuperación” es la proporción de TODOS los conceptos recuperados, incluso aquellos recuperados que son irrelevantes.

Para evaluar los conceptos extraídos por las herramientas, es necesario primeramente clasificar los conceptos en la categoría de relevantes o irrelevantes. La evaluación llevada a cabo puede ser resumida en la siguiente tabla, tabla 5

Tabla 5 Resumen del método de evaluación

	Concepto clasificado como pertinente por el humano	Concepto clasificado como no pertinente por el humano
Concepto clasificado como pertinente por la herramienta	a	b
Concepto clasificado como no pertinente por la herramienta	c	d

La variable "a" representa el número de conceptos generados por el humano que coinciden con los conceptos extraídos por la herramienta. Las variables "b" y "c" representan el número de veces o de conceptos en que el humano y la herramienta no concordaron en la fase de clasificación. La variable "d" representa el número de veces en que el humano y la herramienta concordaron en evaluar un concepto como irrelevante. De esta manera, las fórmulas de "precisión" y de "recuperación" son presentadas a continuación:

Precisión = $a \div (a + b)$

Recuperación = $a \div (a + c)$

8.6.1 El corpus

Los experimentos presentados en este artículo están basados en 100 *papers* de presentados en distintas jornadas JAIIO **[70]** entre los años 2002 y 2013. Estos documentos corresponde a pequeños artículos científicos compuestos de aproximadamente quince páginas cada uno y respetan el estandar LNCS. El corpus utilizado para la evaluación de las estrategias se encuentra en el idioma español e inglés y no cuentan con restricciones de seguridad.
En la siguiente sección, se presentan los pasos seguidos para efectuar la evaluación de las diferentes estrategias así como los resultados obtenidos

8.6.2 Diseño de experimentos

El primer paso para la evaluación de la herramienta consiste en la creación de una lista de referencia, la cual nos servirá como base para la comparación de la eficiencia de cada una de las estrategias. Esta primera lista es completada manualmente. La lista de referencia contiene los metadatos necesarios para evaluar las diferentes estrategias de extracción que fueron desarrolladas.
El segundo paso para la evaluación de las herramientas consiste en la comparación de los conceptos extraídos por cada estrategia contra los conceptos extraídos manualmente y que se encuentran en la lista de referencia

El tercer paso es el análisis de los valores resultantes al hacer uso de la herramienta sobre el corpus.

Los valores que analizaremos en este paso son:

- El número total de conceptos extraídos por la herramienta,
- El número total de conceptos extraídos por la herramienta y que aparecen en la lista de referencia creada manualmente,
- El número total de conceptos extraídos por la herramienta y que no aparecen en la lista de referencia creada manualmente,
- El número total de conceptos extraídos manualmente y que no aparecen en la lista generada por la herramienta.

8.6.3 Resultados del corpus completo

La tabla 6 es el resultado de contar el número de conceptos extraídos al igual que las diferencias con la lista de referencia.

Tabla 6 Resultados obtenidos en la evaluación de los extractores

	Titulo	Idioma	Abstract	Keywords	Autores
Número total de conceptos extraídos	100	100	81	80	80
Número total de conceptos presentes en la lista de referencia (a)	93	100	75	71	25
Número total de conceptos ausentes en la lista de referencia (b)	7	0	6	9	55
Número total de conceptos no extraídos (c)	2	0	15	17	40

Por último para poder hacer la comparación entre las estrategias, los resultados obtenidos en el tercer paso son aplicados a las métricas de “precisión” y “recuperación”. Por ejemplo, para el corpus de 100 *papers* obtendríamos los siguientes resultados (Tabla 7):

Tabla 7 Resultados de precisión y recuperación obtenidos en la evaluación de los extractores

	Titulo	Idioma	Abstract	Keywords	Autores
Precisión	0.93	1	0.92	0.89	0.31
Recuperación	0.98	1	0.83	0.80	0.38

Para este caso, con los resultados obtenidos podríamos decir que la máxima “precisión” es la obtenida por la estrategia para la detección del idioma (LanguageEngine) con un 100%, es decir que detecto todos los idiomas de los *papers* de forma correcta.

Luego la estrategia para la detección de títulos (TitleEngine) con un 93%, la estrategia para la recuperación del abstract (AbstractEngine) con un 92%, 89% para el caso de la estrategia para las palabras claves (KeywordEngine) y por último la estrategia con más dificultades al momento de recuperar de forma correcta los valores el AuthorEngine con un 31%.

En términos de "recuperacion" LanguageEngine obtiene el mejor valor, puede recuperar todos los idiomas. La estrategia TitleEngine con un 98% tiene la capacidad de recuperar un gran número de valores. Luego AbstractEngine con un 83% de recuperación, KeywordEngine con un 80% y AuthorEngine con un 38%.

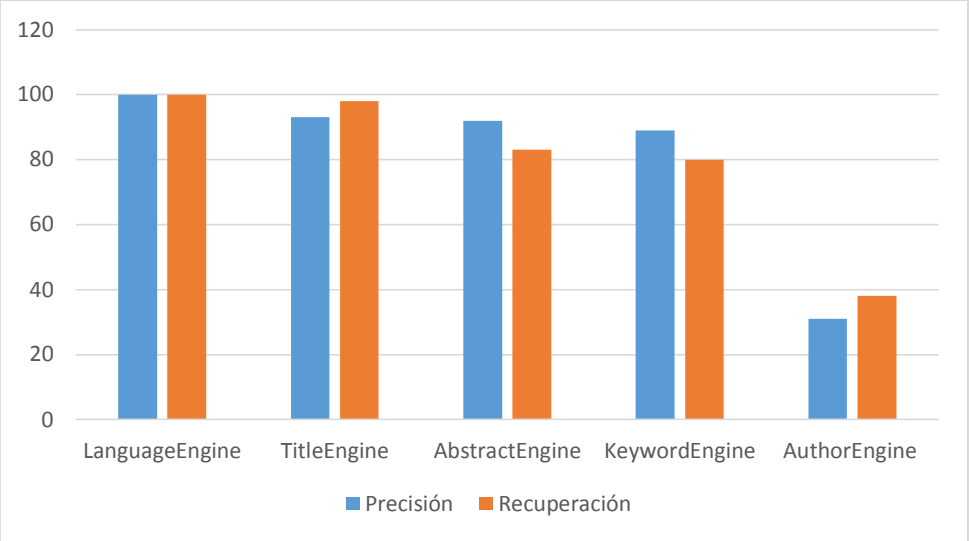


Ilustración 52 Comparación de los porcentajes obtenidos en términos de precisión y recuperación

En la Ilustracion 52, presentamos la diferencia entre la "precisión" y la "recuperación" obtenida por cada estrategia. En este caso, LanguageEngine obtuvo el 100% de "precisión" y "recuperación". Esto significa que LanguageEngine extrae todos los idiomas de forma correcta. Para la estrategia TitleEngine obtuvimos un 93% de "precisión" contra 98% de "recuperación" lo cual significa que con muy baja probabilidad la estrategia obtuvo valores que no correspondían al título. AbstractEngine obtuvo un 83% de "recuperación" y un 93% de "precisión" lo cual significa que la mayoría de los valores extraídos por la estrategia son conceptos que aparecen en la lista de referencia y por tanto son pertinentes. Al igual que AbstractEngine, KeywordEngine con valores que corresponden al 89% en la "precisión" y 80% en la "recuperación" de los valores obtenidos gran parte son pertinentes. AuthorEngine con una "precisión" de 31% y 38% de "recuperación" denota que esta estrategia puede detectar muy pocos nombre de personas.

8.7 Comparación entre los metadatos obtenidos de forma manual y automática

Las siguientes tablas detallan los valores obtenidos de forma manual y los recuperados por el *framework* de extracción para cada uno de los metadatos soportados.

8.7.1 Validación de autores

Tabla 8 Comparativa de extracción manual y automática de autores

Manual	Automático
María Celeste Carignano **Silvio Gonnet** **Horacio Leone**	María Celeste Carignano Silvio Gonnet Horacio Leone
Susana Romaniz **Juan Carlos Ramos** **Marta Castellaro** **Ignacio Ramos**	Catalogación Juan Carlos Ramos Marta Castellaro Ignacio Ramos Facultad Regional Santa Fe
Manuel Imaz, PhD	Manuel Imaz
Wilson Pádua **Thiago R. V. Anjos** **Daniela C. C. Peixoto**	Wilson Pádua Synergia Thiago R . V . Anjos
P. Lafortune **G. Houzeaux** **M. Vazquez** **R. Ar´ıs**	P . Lafortune G . Houzeaux
David Monge **Jirí Belohradský** **Carlos García Garino** **Filip Zelezný**	
Ana Julia Villar **Juan Miguel Santos**	Juan Miguel Santos
Pablo R. Rinaldi **Enzo A. Dari** **Marcelo J. Vénere** **Alejandro Clausse**	Marcelo J Moore
Mario H. González **Ricardo H. Medel** **Sergio D. Canchi**	Mario H Ricardo H
Luis Olsina **Mario Diván**	Luis Olsina Santa Rosa General Pico
Silvio Gonnet **Horacio Leone** **Ana Sofía Zalazar**	Silvio Gonnet Horacio Leone Service Level Agreement Cloud Computing Santa Fe Avellaneda

8.7.2 Validación de títulos

Tabla 9 Comparativa de extracción manual y automática de títulos

Manual	Automático
Razonamiento y Reutilización en el Diseño de Arquitecturas de Software: Prácticas en la Industria Argentina	Razonamiento y Reutilización en el Diseño de Arquitecturas de Software: Prácticas en la Industria Argentina
Catalogación como Apoyo al Uso de Patrones de Seguridad	Catalogación como Apoyo al Uso de Patrones de Seguridad
Aspects of BPM/SOA: Processes, Use Cases and Concerns	Aspects of BPM/SOA: Processes, Use Cases and Concerns
A UML reuse framework and tool for Requirements Engineering	A UML reuse framework and tool for Requirements Engineering
Parallel Electromechanical model of the heart	Parallel Electromechanical model of the heart
A Performance Prediction Module for Work ow Scheduling	
Q-Function Kernel Smoother: a New Approach for Opened Issues in Huge State-Action Spaces	Q-Function Kernel Smoother: a New Approach for Opened Issues in Huge State-Action Spaces
Fluid Simulation with Lattice Boltzmann Methods Implemented on GPUs Using CUDA	Fluid Simulation with Lattice Boltzmann Methods Implemented on GPUs Using CUDA
Propuesta de una Arquitectura de Software para Aplicaciones de Publicidad para Televisión Digital Interactiva	Propuesta de una Arquitectura de Software para Aplicaciones de Publicidad para Televisión Digital Interactiva
Vista de Proceso del Enfoque Integrado de Procesamiento de Flujos de Datos centrado en Metadatos de Mediciones	Vista de Proceso del Enfoque Integrado de Procesamiento de Flujos de Datos centrado en Metadatos de Mediciones

Un Modelo para Contratos de Cloud Computing	Un Modelo para Contratos de Cloud Computing
Sistema de Percepción para un Vehículo Autónomo Submarino	Sistema de Percepción para un Vehículo Autónomo Submarino
Recomendador de usuarios en una plataforma colaborativa en base a su perfil y reputación	14th Argentine Symposium on Articial Intelligence, ASAI 2013

8.7.3 Validación de los idiomas

Tabla 10 Comparativa de extracción manual y automática de idiomas

Manual	Automático
Español	Español
Español	Español
Ingles	Ingles
Ingles	Ingles
Ingles	Ingles
Ingles	Ingles
Ingles	Ingles
Ingles	Ingles
Español	Español
Español	Español
Español	Español
Español	Español

8.7.4 Validación de palabras clave

Tabla 11 Comparativa de extracción manual y automática de palabras clave

Manual	Automático
Encuesta **Arquitectura** **Software** **Razonamiento** **Reutilización**	encuesta arquitectura software razonamiento reutilización
Concerns **aspects** **use cases,** **AOP** **cognitive semantics**	concerns aspects use cases aop cognitive semantics
Requirements Engineering **UML framework**	requirements engineering uml framework
Computational Electrophysiology **Computational Solid Mechanics** **Cardia Mechanics** **Parallelization**	computational electrophysiology computational solid mechanics cardia mechanics parallelization
Workow **Scheduling** **Performance Prediction** **Nonpara- metric Regression**	workow scheduling performance prediction nonpara- metric regression
Reinforcement learning **Q-function approximation** **Q-function approximation** **robot learning**	reinforcement learning q-function approximation kernel smooth- ing robot learning
CUDA **GPU Computing** **Lattice Boltzmann Methods**	cuda gpu computing lattice boltzmann methods
Televisión digital interactiva **ISDB-Tb** **Ginga** **arquitectura de software** **publicidad interactiva**	televisión digital interactiva isdb-tb ginga arquitectura de software

Procesos **Medición** **Flujo de Datos** **C-INCAMI**	procesos medición flujo de datos c-incami
Cloud Computing **Acuerdo de Nivel de Servicio**	cloud computing acuerdo de nivel de servicio
AUV **sistema de percepción** **arquitectura de software**	auv sistema de percepción arquitectura de software
Arreglos – Antenas – Síntesis de Fourier – Smart Grid – Comunicación – Wireless – Sistema Eléctrico – Red inteligente.	como consecuencia de esto la cantidad de cortes en la red eléctrica en horarios de alta demanda se ven incrementados

8.7.5 Validación del resumen

Tabla 12 Comparativa de extracción manual y automática de resumen

Manual	Automático
En los últimos años el diseño de arquitecturas de software ha cobrado notoria importancia tanto en el ámbito industrial como en el ámbito de investigación debido al valor que se le atribuye dentro del proceso de desarrollo de un sistema de software. **En este contexto, se ha prestado especial atención a la documentación del razonamiento realizado por los arquitectos durante un diseño arquitectónico, resaltando las ventajas y desventajas de dicha actividad.** **El presente trabajo intenta brindar una visión de las prácticas de los arquitectos de la industria argentina con respecto a la documentación del razonamiento y su posterior utilización y acceso.**	En los últimos años el diseño de arquitecturas de software ha cobrado notoria importancia tanto en el ámbito industrial como en el ámbito de investigación debido al valor que se le atribuye dentro del proceso de desarrollo de un sistema de software.En este contexto, se ha prestado especial atención a la documentación del razonamiento realizado por los arquitectos durante un diseño arquitectónico, resaltando las ventajas y desventajas de dicha actividad.El presente trabajo intenta brindar una visión de las prácticas de los arquitectos de la industria argentina con respecto a la documentación del razonamiento y su posterior utilización y acceso.
La principal característica de un software seguro reside en la natura-leza de los procesos y las prácticas utilizadas para especificar, diseñar,	La principal característica de un software seguro reside en la natura-leza de los procesos y las prácticas utilizadas para especificar, diseñar,

desarro- llar y desplegar el software.La atención temprana de la seguridad tiene que ver con la adopción de un conjunto de actividades que hacen posible la integración de la misma en el ciclo de vida de desarrollo de software.Los patrones de segu- ridad aplican el concepto de patrón al dominio de la seguridad, describiendo un problema particular de seguridad recurrente que ocurre en un contexto específi- co y presentando una solución probada, permitiendo una transferencia eficiente de experiencia y de conocimientos.La descripción de un patrón debe ayudar a capturar de manera inmediata su esencia: cuál es el problema al que atiende y cuál es la solución propuesta.Los diferentes formatos existentes para su des- cripción y la multiplicidad de fuentes donde se encuentran disponibles, hacen que su descubrimiento demande esfuerzo que desalienta el uso sistemático por parte de los potenciales destinatarios.En este trabajo se presenta el prototipo de un catálogo que busca establecer un puente entre el conocimiento y la experien- cia desarrollados por expertos en seguridad y las necesidades de conocimiento de los equipos de desarrollo de software."	desarro- llar y desplegar el software.La atención temprana de la seguridad tiene que ver con la adopción de un conjunto de actividades que hacen posible la integración de la misma en el ciclo de vida de desarrollo de software.Los patrones de segu- ridad aplican el concepto de patrón al dominio de la seguridad, describiendo un problema particular de seguridad recurrente que ocurre en un contexto específi- co y presentando una solución probada, permitiendo una transferencia eficiente de experiencia y de conocimientos.La descripción de un patrón debe ayudar a capturar de manera inmediata su esencia: cuál es el problema al que atiende y cuál es la solución propuesta.Los diferentes formatos existentes para su des- cripción y la multiplicidad de fuentes donde se encuentran disponibles, hacen que su descubrimiento demande esfuerzo que desalienta el uso sistemático por parte de los potenciales destinatarios.En este trabajo se presenta el prototipo de un catálogo que busca establecer un puente entre el conocimiento y la experien- cia desarrollados por expertos en seguridad y las necesidades de conocimiento de los equipos de desarrollo de software."
"In this paper we show how BPM/SOA avoid the increas- ing complexities added by the aspect- oriented programming (AOP) ap- proach, mainly in relation to functional concerns.From the beginnings of object-orientation, some difficulties derived from the uses cases model have been detected, as they are the root of scattering and tangling.This is the question that AOP addresses even if it uses its own jargon: con- cerns in place of use cases.The present	"In this paper we show how BPM/SOA avoid the increas- ing complexities added by the aspect- oriented programming (AOP) ap- proach, mainly in relation to functional concerns.From the beginnings of object-orientation, some difficulties derived from the uses cases model have been detected, as they are the root of scattering and tangling.This is the question that AOP addresses

analysis of the problem is based on concepts of Cognitive Semantics (CS) that allow to explain some odd questions, such as the way of presenting the classical UML architecture as a '4 + 1' –in place of 5– views.Some CS concepts, such as perspective, focusing and profiling help to clarify some phenomena that have been analyzed from a very general notion of view that needs, evidently, to be refined in order to build more useful ideas about software engineering."	even if it uses its own jargon: concerns in place of use cases.The present analysis of the problem is based on concepts of Cognitive Semantics (CS) that allow to explain some odd questions, such as the way of presenting the classical UML architecture as a '4 + 1' –in place of 5– views.Some CS concepts, such as perspective, focusing and profiling help to clarify some phenomena that have been analyzed from a very general notion of view that needs, evidently, to be refined in order to build more useful ideas about software engineering."
"Requirement Engineering (RE) activities are manual and critical by nature.Providing some automated support for the RE tasks helps analysts to reduce manual labor, and in consequence, reduce defects rates and increase reuse and motivation.In this paper, we introduce a UML framework and tool support which automates part of the RE process.Using UML stereotypes concepts as the core of this solution, we created a set of integrated tools composed by: (1) a reusable framework that models some common RE behavior patterns that are typically present in information system projects; (2) a function that allows the reuse of information provided by entity modeling; (3) a tool that automates the generation of application prototypes; and (4) a tool that analyzes specific types of defects.Our preliminary findings indicate that the framework and the automated support are effective at RE modeling and review.In addition, they in-	"Requirement Engineering (RE) activities are manual and critical by nature.Providing some automated support for the RE tasks helps analysts to reduce manual labor, and in consequence, reduce defects rates and increase reuse and motivation.In this paper, we introduce a UML framework and tool support which automates part of the RE process.Using UML stereotypes concepts as the core of this solution, we created a set of integrated tools composed by: (1) a reusable framework that models some common RE behavior patterns that are typically present in information system projects; (2) a function that allows the reuse of information provided by entity modeling; (3) a tool that automates the generation of application prototypes; and (4) a tool that analyzes specific types of defects.Our preliminary findings indicate that the framework and the automated support are effective at RE

crease motivation and promote team engagement, through elimination of repeti- tive activities."	modeling and review.In addition, they in- crease motivation and promote team engagement, through elimination of repeti- tive activities."
"in this paper, we present a high performance computational electrome- chanical model of the heart, coupling between electrical activation and mechan- ical deformation and running efficiently in up to thousands of processors.The electrical potential propagation is modelled by FitzHugh-Nagumo or Fenton- Karma models, with fiber orientation.The mechanical deformation is treated us- ing anisotropic hyper-elastic materials in a total Lagrangian formulation.Several material models are assessed, such as models based on biaxial tests on excised myocardium or orthotropic formulations.Coupling is treated using the Cross- Bridges model of Peterson.The scheme is implemented in Alya, which run sim- ulations in parallel with almost linear scalability in a wide range computer sizes, up to thousands of processors.The computational model is assessed through sev- eral tests, including those to evaluate its parallel performance."	"in this paper, we present a high performance computational electrome- chanical model of the heart, coupling between electrical activation and mechan- ical deformation and running efficiently in up to thousands of processors.The electrical potential propagation is modelled by FitzHugh-Nagumo or Fenton- Karma models, with fiber orientation.The mechanical deformation is treated us- ing anisotropic hyper-elastic materials in a total Lagrangian formulation.Several material models are assessed, such as models based on biaxial tests on excised myocardium or orthotropic formulations.Coupling is treated using the Cross- Bridges model of Peterson.The scheme is implemented in Alya, which run sim- ulations in parallel with almost linear scalability in a wide range computer sizes, up to thousands of processors.The computational model is assessed through sev- eral tests, including those to evaluate its parallel performance."
"Through the years, scienti c applications have demanded more powerful and sophisticated computing environments and manage- ment techniques.Work ows facilitated the design and management of scienti c applications.The complexity of today's work ows demand a	"Through the years, scienti c applications have demanded more powerful and sophisticated computing environments and manage- ment techniques.Work ows facilitated the design and management of scienti c applications.The complexity

high amount of resources and mechanisms for provisioning them.The execution of scienti c work ow applications is a complex task and depends on how the resources are assigned.Scheduling is the name given to the process that assigns computing resources to the tasks comprised in a work ow.This work presents a scheduling algorithm (PPSA) for work ows tightly coupled to a performance prediction module (PEM).A set of experiments was developed for measuring the performance of the algorithm using the information provided by the proposed performance module.The proposed algorithm is compared with an algorithm included in the well-known work ow middlewares Condor DAGMan and ASKALON."	of today's work ows demand a high amount of resources and mechanisms for provisioning them.The execution of scienti c work ow applications is a complex task and depends on how the resources are assigned.Scheduling is the name given to the process that assigns computing resources to the tasks comprised in a work ow.This work presents a scheduling algorithm (PPSA) for work ows tightly coupled to a performance prediction module (PEM).A set of experiments was developed for measuring the performance of the algorithm using the information provided by the proposed performance module.The proposed algorithm is compared with an algorithm included in the well-known work ow middlewares Condor DAGMan and ASKALON."
"Reinforcement learning is a method where agents learn how to map states to actions.They have to interact with their environment and then, they receive reinforcements from it.Q-learning is, probably, the most used technique in reinforcement learning and it needs to compute by successive estimations an action-value function (ó-function).In a large number of problems, the environments have huge state-action spaces and, therefore it is necessary to compute ó for countless pairs of states and actions.A previous work has shown a way to do that using only one approximator of ó, a kernel smoother, over the state-action space, where resolution depends on the frequency of visits during the learning.This paper extends the previous proposal presenting	"Reinforcement learning is a method where agents learn how to map states to actions.They have to interact with their environment and then, they receive reinforcements from it.Q-learning is, probably, the most used technique in reinforcement learning and it needs to compute by successive estimations an action-value function (ó-function).In a large number of problems, the environments have huge state-action spaces and, therefore it is necessary to compute ó for countless pairs of states and actions.A previous work has shown a way to do that using only one approximator of ó, a kernel smoother, over the state-action space, where resolution depends on the frequency of visits during the learning.This paper extends the

improvements on some issues related to its use.To validate the proposed ideas, experiments by simulations were performed and their results are presented."	previous proposal presenting improvements on some issues related to its use.To validate the proposed ideas, experiments by simulations were performed and their results are presented."
"This work presents a parallel shared-memory implementation of a Lattice Boltzmann Model for Computational Fluid Dynamics. The model was specially coded for a Graphic Processing Unit using NVIDIA Compute Unified Device Architecture. Two-dimensional backwards-facing step flow simulation results were validated against a proved Navier-Stoke Finite Element solver. We obtain very promising speed-up results up to 40 times on a GeForce 8800 GT, compared with a normal "single core" PC-CPU. "	"This work presents a parallel shared-memory implementation of a Lattice Boltzmann Model for Computational Fluid Dynamics.The model was specially coded for a Graphic Processing Unit using NVIDIA Compute Unified Device Architecture.Two-dimensional backwards-facing step flow simulation results were validated against a proved Navier-Stoke Finite Element solver.We obtain very promising speed-up results up to 40 times on a GeForce 8800 GT, compared with a normal "single core" PC-CPU."
"La Televisión Digital interactiva (TVDi) es una tecnología de trans-misión digital de contenidos televisivos que, a diferencia de la televisión analó-gica tradicional, transmite la información codificada en forma binaria, lo que hace posible una óptima calidad del video y sonido y el envío de software que puede ser ejecutado en el aparato receptor.Si bien tanto a nivel mundial como a nivel regional la implementación de la TVDi continúa su avance, con numero-sos países que ya han realizado por completo el cambio al sistema digital y la mayoría de los países de la región Latinoamericana adoptando el estándar brasi-lero ISDB-Tb y su middleware Ginga, existe aún una ausencia notable de soft-	"La Televisión Digital interactiva (TVDi) es una tecnología de trans-misión digital de contenidos televisivos que, a diferencia de la televisión analó-gica tradicional, transmite la información codificada en forma binaria, lo que hace posible una óptima calidad del video y sonido y el envío de software que puede ser ejecutado en el aparato receptor.Si bien tanto a nivel mundial como a nivel regional la implementación de la TVDi continúa su avance, con numero-sos países que ya han realizado por completo el cambio al sistema digital y la mayoría de los países de la región Latinoamericana adoptando el estándar brasi-lero ISDB-Tb y su middleware Ginga,

ware para la TVDi y en particular que aproveche el enorme potencial de la pu- blicidad interactiva.En este trabajo presentamos una propuesta de una arquitec- tura de software para aplicaciones de publicidad interactiva en el marco del es- tándar ISDB-Tb, establecida teniendo en cuenta los requerimientos de los acto- res involucrados y las restricciones impuestas por la plataforma."	existe aún una ausencia notable de soft- ware para la TVDi y en particular que aproveche el enorme potencial de la pu- blicidad interactiva.En este trabajo presentamos una propuesta de una arquitec- tura de software para aplicaciones de publicidad interactiva en el marco del es- tándar ISDB-Tb, establecida teniendo en cuenta los requerimientos de los acto- res involucrados y las restricciones impuestas por la plataforma."
"El enfoque integrado de procesamiento de flujos de datos centrado en metadatos de mediciones, es un gestor de flujos de datos sustentado en un mar- co de medición y evaluación, el cual incorpora comportamiento detectivo y predictivo, mediante el empleo de las mediciones y metadatos asociados.Este trabajo discute la formalización de los procesos asociados con el funcionamien- to del gestor de flujos de datos, como así también la interacción entre ellos.La formalización de los procesos se realiza en base a SPEM, y se consideran las actividades comprendidas entre la configuración de las fuentes de datos involu- cradas en un proceso de medición y evaluación, hasta aquellas asociadas con la emisión de las alarmas detectivas o predictivas.Esto permite hacer comunicable el aspecto de procesos del enfoque, y adicionalmente, abre la posibilidad para medir y evaluar los propios procesos formalizados del enfoque, como medio pa-	"El enfoque integrado de procesamiento de flujos de datos centrado en metadatos de mediciones, es un gestor de flujos de datos sustentado en un mar- co de medición y evaluación, el cual incorpora comportamiento detectivo y predictivo, mediante el empleo de las mediciones y metadatos asociados.Este trabajo discute la formalización de los procesos asociados con el funcionamien- to del gestor de flujos de datos, como así también la interacción entre ellos.La formalización de los procesos se realiza en base a SPEM, y se consideran las actividades comprendidas entre la configuración de las fuentes de datos involu- cradas en un proceso de medición y evaluación, hasta aquellas asociadas con la emisión de las alarmas detectivas o predictivas.Esto permite hacer comunicable el aspecto de procesos del enfoque, y adicionalmente, abre la posibilidad para medir y evaluar los propios procesos

ra monitorear cuantitativamente su salud funcional en línea."	formalizados del enfoque, como medio pa- ra monitorear cuantitativamente su salud funcional en línea."
"Cloud computing es un paradigma de negocio que se gestiona a través de internet, donde diferentes proveedores ofrecen sus recursos informáti- cos de manera de servicios, utilizando las ventajas de virtualización.Estos ser- vicios son adquiridos por el consumidor bajo demanda y acordando previamen- te los acuerdos de nivel de servicio.Aunque este paradigma está ganado popu- laridad, todavía no existe una definición unificada de los servicios y estándares para la creación de estos acuerdos.Generalmente, los proveedores formulan sus contractos de forma estática, ambigua y predefinida, protegiendo sus propios in- tereses.En este trabajo se propone un modelo del entorno de cloud computing, con la finalidad de facilitar la captura de datos sobre los servicios y para que los consumidores cuenten con los criterios suficientes para evaluar y comparar di- ferentes contratos de servicios, utilizando métricas.Además, el modelo pro- puesto puede servir como base para los sistemas de monitoreo de nivel de servi- cio, considerando el análisis de parámetros funcionales, monetarios y de cali- dad."	"Cloud computing es un paradigma de negocio que se gestiona a través de internet, donde diferentes proveedores ofrecen sus recursos informáti- cos de manera de servicios, utilizando las ventajas de virtualización.Estos ser- vicios son adquiridos por el consumidor bajo demanda y acordando previamen- te los acuerdos de nivel de servicio.Aunque este paradigma está ganado popu- laridad, todavía no existe una definición unificada de los servicios y estándares para la creación de estos acuerdos.Generalmente, los proveedores formulan sus contractos de forma estática, ambigua y predefinida, protegiendo sus propios in- tereses.En este trabajo se propone un modelo del entorno de cloud computing, con la finalidad de facilitar la captura de datos sobre los servicios y para que los consumidores cuenten con los criterios suficientes para evaluar y comparar di- ferentes contratos de servicios, utilizando métricas.Además, el modelo pro- puesto puede servir como base para los sistemas de monitoreo de nivel de servi- cio, considerando el análisis de parámetros funcionales, monetarios y de cali- dad."
"La necesidad de la industria off-shore para compartir información y recursos de energía a través de cables y tuberías submarinas conduce a un cre-	"La necesidad de la industria off- shore para compartir información y recursos de energía a través de cables y tuberías submarinas conduce a un cre-

ciente despliegue de infraestructuras sumergidas.Esto requiere de un posterior mantenimiento preventivo.Los vehículos autónomos submarinos (AUVs) re- presentan una alternativa para llevar a cabo esta tarea.En base a la percepción, estos vehículos deben estar equipados con distintos dispositivos de sensores como cámaras de vídeo, dispositivo rastreador electromagnético, sonar de ba- rrido lateral, ecosonda muti-haz, como así también dispositivos de ubicación como sistema de posicionamiento global, sistema de navegación inercial, brúju- la, entre otros.Cada uno de estos dispositivos hay que tratarlos por separado pa- ra la captura e interpretación de datos, pero en conjunto para la búsqueda de co- nocimiento útil que modifique el comportamiento on-line de un AUV.Este do- cumento presenta el diseño y desarrollo de una arquitectura de software de un sistema de percepción para un AUV (PS-AUV).La arquitectura emplea como entrada datos provenientes de dispositivos de sensores interconectados aplican- do distintos procesos, y como salida, conocimiento que alimentará al modelo del mundo del robot que se encuentra implementado en forma de un sistema ba- sado en conocimiento."	ciente despliegue de infraestructuras sumergidas.Esto requiere de un posterior mantenimiento preventivo.Los vehículos autónomos submarinos (AUVs) re- presentan una alternativa para llevar a cabo esta tarea.En base a la percepción, estos vehículos deben estar equipados con distintos dispositivos de sensores como cámaras de vídeo, dispositivo rastreador electromagnético, sonar de ba- rrido lateral, ecosonda muti-haz, como así también dispositivos de ubicación como sistema de posicionamiento global, sistema de navegación inercial, brúju- la, entre otros.Cada uno de estos dispositivos hay que tratarlos por separado pa- ra la captura e interpretación de datos, pero en conjunto para la búsqueda de co- nocimiento útil que modifique el comportamiento on-line de un AUV.Este do- cumento presenta el diseño y desarrollo de una arquitectura de software de un sistema de percepción para un AUV (PS-AUV).La arquitectura emplea como entrada datos provenientes de dispositivos de sensores interconectados aplican- do distintos procesos, y como salida, conocimiento que alimentará al modelo del mundo del robot que se encuentra implementado en forma de un sistema ba- sado en conocimiento."
En el contexto del análisis de movimiento en secuencias de im´agenes, una medida de movimiento computada localmente puede ser	

tanto nula como diferente de cero. En consecuencia, puede ser importante considerar expl´ıcitamente que dicha cantidad toma valores discretos o simb´olicos expresando la ausencia de movimiento, o bien, valores continuos relacionados con las medidas de movimiento en sí. Nuevas representaciones de este tipo de información han sido recientemente presentadas en la forma de campos aleatorios de Markov con estados mixtos, en un enfoque puramente espacial. En este art´ıculo, introducimos nuevos modelos de estados mixtos para el modelado temporal de texturas dinámicas o texturas de movimiento. Se propone una formulación innovadora de lo que llamaremos cadenas de Markov mixtas asumiendo dependencia causal como un primer enfoque de estudio de la evolución temporal de las mismas. Luego, y basados en este modelo, proponemos un método de segmentación de texturas de movimiento, el cual demuestra el buen desempe˜no de estos modelos en el an´alisis de escenas reales. dvips -o	

Capítulo 9 - Conclusión y trabajo futuro

En este capítulo, la sección 9.1 presenta una recapitulación de las contribuciones realizadas con este trabajo. A continuación, en la sección 9.2 se presenta un resumen de las experiencias realizadas basadas en esta propuesta y en la sección 9.3 describe las propuestas de trabajo futuro.

9.1 Contribuciones

El objetivo principal de este trabajo consistía en facilitar la carga de contenidos en repositorios digitales, automatizando la extracción de metadatos. Por medio de la utilización del *framework* implementado, se logró mejorar el proceso de ingesta de contenidos en la herramienta DSpace.

La modificación del flujo de datos propuesto en la sección 8.2 le permitió al *workflow* de carga de DSpace disponer del documento en etapas iniciales del proceso de carga para poder aplicarles las estrategias de extracción y entregarle a la etapa de descripción los metadatos recuperados para ser verificados y corregidos por el usuario.

El desarrollo del *framework* como un módulo independiente de DSpace facilita su uso en repositorios sin necesidad de realizar modificaciones complejas en el código fuente y gracias a la extensión de los campos del formulario propuesta en la sección 8.4.4 se le puede indicar a DSpace sobre qué metadato se desea aplicar la extracción automática, dejando a elección del administrador del repositorio el uso de determinados extractores.

Las configuraciones y guías aportadas en las secciones 8.4.3, 8.4.4 y 8.4.5 brindan especificaciones técnicas de la forma en que DSpace hace uso del *framework* desarrollado. Mediante estas especificaciones se permite la escalabilidad del *framework* posibilitando, a futuro, la agregación de nuevos extractores de metadatos.

9.2 Experiencias realizadas

Al momento de iniciar el relevamiento sobre herramientas existentes que permitiesen cumplir con los objetivos propuestos relacionados con la extracción automática de metadatos, se encontraron diversos trabajos de investigación que no contaban con una implementación bien definida y que trabajaban sobre metadatos que eran extraídos de los atributos del archivo no contemplando los definidos por el autor. Ejemplo de esto se daba cuando mostraban la extracción de las palabras claves, las cuales no eran las que el autor definía en su trabajo sino que eran inferidas en base a la frecuencia en que aparecían las palabras en el texto.

Gran parte de las herramientas existentes para la extracción de metadatos eran propietarias, lo que dificultaba en primera instancia la evaluación de las mismas. De las herramientas libres analizadas, resultaba complejo poder integrarlas en repositorios en producción dado a que se hacía necesario realizar costosas modificaciones del código fuente. Y en algunos casos las herramientas se disponían únicamente para ser utilizadas de forma *on-line*, en las cuales se obtenían elevados tiempos de respuesta debido a la sobrecarga de los sitios que las alojaba.

Al momento de la extracción de autores se presentaron problemas de detección dada la gran diversidad de formatos y nacionalidades dado a que OpenNLP utiliza modelos para el reconocimiento de personas en base a su idioma.

9.3 Trabajo futuro

El *framework* desarrollado puede ser ampliado y/o mejorado considerando los aspectos expuestos a continuación:

1. **Mejora en la extracción de autores**. Para lograr una correcta extracción de autores utilizando OpenNLP se propone entrenar los modelos para la recuperación de nombres. Este entrenamiento se podrá realizar con datos de entrada de autores presentes en una base de datos de un repositorio en producción que cuente con nombres correctos. Este entrenamiento mejorará la cantidad y calidad de los autores extraídos dado a que estos datos de entrada tienen un origen igual al que luego se aplicaran las extracciones.

2. **Ampliación de las estrategias de extracción**. Con el fin de dar soporte a otros metadatos, como por ejemplo referencias, institución de origen, etc. Se propone extender el diseño de las estrategias de extracción. Para esto se debe crear una nueva clase, por ejemplo *ReferencesEngine* que contenga la lógica capaz de localizar las referencias y extraerlas. Luego se debe crear un nuevo método en la capa de servicios para hacer uso de la nueva estrategia desde el repositorio. En DSpace es necesario agregar la configuración en el archivo *input-forms.xml* para indicar que el campo se extrae automáticamente, además se debe extender la lógica de la clase *AutomaticExtractStep* para que en el momento de procesar los campos utilice el nuevo servicio y así obtener el valor del metadato.

3. **Extensión de la aplicación a otros formatos de archivo y estándares de publicaciones científicas**. Dando soporte a otros tipos de archivos como por ejemplo .TXT, .DOC, .ODP, la herramienta podrá procesar un conjunto más amplio de archivos. Para lograrlo se propone desarrollar una nueva clase similar a la *PdfReaderEngine* que contenga la lógica necesaria para descomponer el texto en líneas. Luego, dentro de las clases que representan los extractores se debe determinar el tipo archivo y hacer uso de la nueva clase. Siguiendo esta lógica se propone también incorporar nuevos estándares de publicaciones científicas extendiendo el análisis a formatos tales como el IEEE.

4. **Curación de metadatos**. En un repositorio en producción donde ya se dispone de contenidos con sus metadatos asociados, se los podría mejorar, aplicando un proceso de curación de metadatos. En este caso se propone desarrollar un proceso que utilice los servicios de extracción del *framework* para volver a recuperar los metadatos del documento y proceder a una curación.

Referencias

[1] «Servicio Nacional de Repositorios Digitales» [En línea]. Disponible en: http://repositorios.mincyt.gob.ar/.

[2] «CICyT» [En línea]. Disponible en: www.cicyt.mincyt.gob.ar.

[3] [En línea]. Disponible en: dublincore.org. [Último acceso: Marzo 2013].

[4] I. LOM. [En línea]. Disponible en: www.ieee.org. [Último acceso: Marzo 2013].

[5] C. D. C. B. S. F. y. C. S. Ana Casali, Asistente para el Depósito de Objetos en Repositorios con Extracción Automática de Metadatos.

[6] L. Carrizo, «Internacionalización del conocimiento: El impacto de la globalización en la educación superior.» Abril 2005. [En línea]. Disponible en: http://www.rsu.uninter.edu.mx/doc/antecedentes_contexto/InternacionalizaciondelConocimiento.pdf.

[7] J. Hilera González y R. Hoya Marín, Estándares de e-learning: Guía de Consulta., Universidad de Alcalá, 2010.

[8] «SCORM» [En línea]. Disponible en: http://scorm.com/scorm-explained/.

[9] «IMS» [En línea]. Disponible en: http://www.imsglobal.org/learningdesign/.

[10] N. E. García y S. E. Jaroszczuk, «Objetos digitales: una experiencia de representación con metadatos Dublín Core.,» de *Encuentro Nacional de Catalogadores (1°: 2008: Buenos Aires). I Encuentro Nacional de Catalogadores: experiencias en la organización y tratamiento de la información e*, vol. I, Buenos Aires, Biblioteca Nacional, 2009, pp. Vol. 1 (pág. 193-206).

[11] P. Caplan, You Call It Corn, "We Call It Syntax-Independent Metadata for Document-Like Objects." The Public-Access Computer Systems Review 6, 6 ed., University Libraries, University of Houston. All Rights Reserved, 1995.

[12] M. I. D. Founier, «Organización y recuperación de información en Internet: teoría de los metadatos» 2006. [En línea]. Disponible en: http://bvs.sld.cu/revistas/aci/vol14_5_06/aci06506.htm. [Último acceso: Junio 2010].

[13] T. Berners-Lee, «Semantic Web-XML» 2000. [En línea]. Disponible en: http://www.w3.org/2000/Talks/1206-xml2k-tbl.

[14] R. Garduño Vera, “Organización de la información documental y su utilidad social.» La Información en el inicio de la era electrónica : Organización del conocimiento y sistemas de información., vol. I, México : UNAM, Centro Universitario de Investigaciones Bibliotecológicas, 1998.

[15] P. y. J. S. B. Martínez Ortega, Los Metadatos y la información digital, México: Dirección General de Bibliotecas, Universidad Nacional Autónoma de México, Área de la Investigación Científica..

[16] M. Standars, «Library of Congress Network Development and MARC Standards Office» 29 Agosto 2012. [En línea]. Disponible en: http://www.loc.gov/marc/.

[17] F. CORMENZA, Normas y estructuras para automatizar la información, resumen sobre el protocolo Z39.5.

[18] A. Carrión Gútiez, «De las virtudes del catálogo virtual. Dossier 2. Boletín de la SEDIC. p 2-3.» [En línea]. Disponible en: http://www.sedic.es/z3950.pdf.

[19] M. E. Arango, El Z39.50 En el Ambiente de Transferencia y Recuperación de Información, Bogotá: Universidad pontificia Javeria.

[20] H. Benitez Sanchez y F. Robayo Romero, «Protocolo Z39.50 una Herramienta Importante en la Recuperación de Información, 2007. pp.1-17.» [En línea]. Disponible en: http://hdl.handle.net/10760/9556. [Último acceso: 29 Agosto 2012].

[21] «UNESCO» [En línea]. Disponible en: http://unesdoc.unesco.org/images/0005/000557/055778SB.pdf.

[22] D. C. Sabino, Caracas: Panapo, 1994.

[23] P. Halaban, La comunicación virtual en educación a distancia, un estudio en interacciones comunicacionales y procesos pedagógicos en internet., CICCUS, 2010.

[24] 2. Ginsparg, 2. Björk y 2. Gargouri.

[25] M. Sonntag., «*Metadata in E-Learning Applications: Automatic Extraction and Reuse», in Christian Hofer, Gerhard Chroust (Eds.): IDIMT-2004. 12th Interdisciplinary Information Management Talks, pp. 219-231, Universitätsverlag Rudolf Trauner, Linz, Austria, 2004..*

[26] J. Akeroyd, Information management and e-learning.some perspectives," Aslib procs: New information perspectives, 2005.

[27] S. B. e. al., A need analysis framework for the design of digital repositories in higher education., 2008.

[28] «Repositorios Digitales,» 2013. [En línea]. Disponible en: http://repositorios.mincyt.gob.ar/. [Último acceso: 19 Diciembre 2013].

[29] «Open Society Institute and Soros Foundations Network,» [En línea]. Disponible en: http://www.soros.org/.

[30] «Budapest Open Access Initiative (BOAI). Iniciativa de Acceso Abierto de Budapest. GeoTrópico online, 1 (1), 2003,» 2003. [En línea]. Disponible en: http://www.geotropico.org/files/PDFBoai_Espanol_1-1.pdf. [Último acceso: Marzo 2010].

[31] [En línea]. Disponible en: http://www.budapestopenaccessinitiative.org/translations/spanish-translation.

[32] S. a. M. R. Sánchez, «La denominación y el contenido de los repositorios institucionales en acceso abierto : base teórica para la "ruta verde".» 2006. [En línea]. Disponible en: http://eprints.rclis.org/6368/.

[33] R. Crow, «The case for institutional repositories: A sparc position paper.» *Technical Report 223,* 2002.

[34] C. Lynch, «Institutional repositories: Essential infrastructure for scholarship in the digital age.» *Technical Report 226,* 2003.

[35] C. S. McDowell, «institutional repository deployment in american academe since early 2005 repositories by the numbers, part 2.» *D-Lib Magazine,* nº 13, 2007.

[36] A. LÓPEZ MEDINA, Guía para la puesta en marcha de un repositorio institucional., Madrid: SEDIC, 2007.

[37] W. Nixon, «Daedalus: initial experiences with Eprints and DSpace at the University of Glasgow» 2003. [En línea]. Disponible en: http://www.ariadne.ac.uk/issue37/nixon/intro.htm.

[38] R. Crow, A guide to institutional repository software, New York: OpenSociety Institute, 2004.

[39] Y. Han, «Digital content management: the search for a content manage-ment system», Library hi tech, 2004.

[40] Y. Han, Space data and information transfer systems–open archival informa-tion system–reference model, Ginebra: International Organization for Standardization, 2003.

[41] J. Kim, «Finding documents in a digital institutional repository: DSpaceand Eprints. Proceedings 68th Annual meeting of the American Society for Information Science and Technology» 2005. [En línea]. Disponible en: http://eprints.rclis.org/archive/00005189/01/Kim_Finding.pdf. [Último acceso: 22 Enero 2012].

[42] M. Prudlo, «E–archiving: an overview of some repository managementsoftware tools» 2005. [En línea]. Disponible en: http://www.ariadne.ac.uk/issue43/prudlo/intro.html. [Último acceso: 16 Enero 2013].

[43] J. Tramullas y P. Garrido, «Los estudios de usuario en proyectos de bi-blioteca digital: una revisión de técnicas» de *Actas de las 9as Jornadasespañolas de documentación Infogestión, pp. 169–179.*, 2005.

[44] «Open Access and Institutional Repositories with EPrints» [En línea]. Disponible en: http://www.eprints.org/.

[45] «DSPACE» [En línea]. Disponible en: http://www.dspace.org/. [Último acceso: Mayo 2010].

[46] «Fedora Commons Repository Software Open source tecnologies to manage, preserve,and link your digital content» [En línea]. Disponible en: http://www.fedora.info/.

[47] «Greenstone digital library Software» [En línea]. Disponible en: http://www.greenstone.org/index_es.

[48] Y. Sarduy Dominguez y P. Urra Gonzalez, «Herramientas para la creación de colecciones» 2006. [En línea]. Disponible en: http://bvs.sld.cu/revistas/aci/vol14_5_06/aci19506.htm. [Último acceso: Mayo 2010].

[49] H. Kuna, S. Jaroszczuk y D. Caballero, Herramientas para la creación de colecciones digitales. EN: Miranda, M; Kuna, H; Prevosti, N; García, N; Oria, M; Jaroszczuk,S;Caballero, D. Informe final 2008. "Iniciativas de acceso abierto para la conformación de repositorios institucionales", Secretaría de Investigación y Posgrado; UNaM.

[50] «SEDICI» [En línea]. Disponible en: http://sedici.unlp.edu.ar/.

[51] «Repositorio Institucional del Ministerio de Educación de la Nación» [En línea]. Disponible en: http://repositorio.educacion.gov.ar/.

[52] «PostgreSQL» [En línea]. Disponible en: http://www.postgresql.org.es/.

[53] «ORACLE» [En línea]. Disponible en: http://www.oracle.com/es/index.html.

[54] «Framework Cocoon» [En línea]. Disponible en: http://cocoon.apache.org/.

[55] «Digital Repository Interface» [En línea]. Disponible en: https://wiki.duraspace.org/display/DSDOC4x/DRI+Schema+Reference.

[56] F. Lancaster, Information Retrieval Systems: Characteristics, Testing and Evaluation, Wiley, New York (1968).

[57] P. y. R. L. Jacobs, "Innovations in text interpretation". Artificial Intelligence 63, 1993.

[58] A. Moreno Sandoval, "Lingüística Computacional. Introducción a los modelos simbólicos, estadísticos y biológicos"., Madrid: Editorial Sintesis, 1998.

[59] C. Rijsbergen, Information Retrieval., Glasgow, University, 1999.

[60] C. J. van Rijsbergen, "Information Retrieval" Second Edition, London: Butterworths: London: Butterworths, 1979.

[61] «Intelligent Miner for Text "Text Analysis Tools version 2.2",» Segunda Edicion, Junio 1998..

[62] J. y. L. W. Cowie, "Information Extraction", Communications of the ACM, vol. 39, 1996.

[63] L. W. y. B. A.D., «"Machine Translation of Languages"» *Technology Press of MIT ans Wiley, Cambridge, Mass.,* 1955.

[64] «Apache Software Foundation opennlp 2000,» [En línea]. Disponible en: http://opennlp.apache.org.

[65] «Wikipedia - Definición de Hidden Markov Model» [En línea]. Disponible en: http://es.wikipedia.org/wiki/Modelo_oculto_de_M%C3%A1rkov.

[66] K. y. W. P. SPARCK JONES, Readings in information retrieval. San Francisco: Morgan Kaufmann, 1997..

[67] «Alchemy API» [En línea]. Disponible en: http://www.alchemyapi.com/.

[68] SEDICI. [En línea]. Disponible en: http://sedici.unlp.edu.ar/bitstream/handle/10915/41759/Documento_completo.pdf?sequence=1.

[69] G. S. M. McGill, Introduction to modern information retrieval, McGraw-Hill Book Company, 1983.

[70] «JAIIO» [En línea]. Disponible en: http://www.sadio.org.ar/.