UNIVERSIDAD NACIONAL DE LA PLATA

FACULTAD DE INFORMÁTICA

TESINA DE LICENCIATURA

Titulo: Servicio de Recolección de Metadatos genérico para documentos
Autores: Julieta Paz Rodríguez Vuan
Director: Dra. Marisa Raquel De Giusti
Asesor profesional: Dr. Gonzalo Luján Villarreal
Asesor profesional: Lic. Ariel Jorge Lira
Carrera: Licenciatura en sistemas

Resumen

Esta tesina de grado detalla la implementación de una herramienta para permitir y optimizar el intercambio de información estructurada sobre recursos académicos y científicos proveniente de contextos que no necesariamente cumplen estándares de intercambio o esquemas de catalogación normalizados. Para realizar este trabajo, se analizaron y estudiaron las plataformas que publican artículos científicos, las distintas herramientas que éstos utilizan para comunicarse y finalmente los métodos tradicionales que se utilizan para compartir información (datos y metadatos). Con una idea formada se comenzó la creación de una herramienta que permite a los administradores de los repositorios institucionales (tanto CIC-Digital como SEDICI), realizar solicitudes a través de un formulario y que éste, como respuesta, realice la precarga del formulario de autoarchivo agilizando de esta manera la etapa de catalogación de materiales y con ello agilizar el poblamiento de los repositorios institucionales. Para dicha herramienta se estableció un método de extracción de información y un formato para el intercambio de metadatos.

Palabras Claves

Artículo científico, metadato, extracción de metadatos, interoperabilidad

Trabajos Realizados

Se desarrolló una herramienta capaz de realizar la extracción de metadatos de artículos científicos que tengan o no su información normalizada.
También, se desarrolló una extensión para el navegador web Chrome que autocompleta el formulario de carga de SEDICI con los metadatos que la herramienta extrae del artículo solicitado.

Conclusiones

A lo largo de esta tesina se explicaron las distintos motivos que dieron como objetivo la creación de esta herramienta, se detalló la investigación realizada del marco teórico junto con el análisis de los requerimientos funcionales para luego llevar al lector a través de la implementación de la herramienta y finalmente mostrar los casos concretos de uso de la herramienta.

Trabajos Futuros

Mejora de los metadatos que se recolectan; retornar resultados en distintos formatos como XML o CSV; agregar nuevos tipos de documentos para la extracción; utilización de la herramienta para carga de información correcta de los datos; capturar información de listados de links o listado de páginas web; módulo de caché; incrementar la velocidad de carga de datos en los sistemas de información académica

Fecha de la presentación: Octubre 2017

Resumen

Esta tesina de grado detalla la implementación de una herramienta para permitir y optimizar el intercambio de información estructurada sobre recursos académicos y científicos proveniente de contextos que no necesariamente cumplen estándares de intercambio o esquemas de catalogación normalizados. Para realizar este trabajo, se analizaron y estudiaron las plataformas que publican artículos científicos, las distintas herramientas que éstos utilizan para comunicarse y finalmente los métodos tradicionales que se utilizan para compartir información (datos y metadatos). Con una idea formada se comenzó la creación de una herramienta que permite a los administradores de los repositorios institucionales (tanto CIC-Digital como SEDICI), realizar solicitudes a través de un formulario y que éste, como respuesta, realice la precarga del formulario de autoarchivo agilizando de esta manera la etapa de catalogación de materiales y con ello agilizar el poblamiento de los repositorios institucionales. Para dicha herramienta se estableció un método de extracción de información y un formato para el intercambio de metadatos.

Capítulo 1 | Introducción

Motivación

Existe en la actualidad una amplia gama de plataformas que tienen como principal objetivo la difusión de recursos académicos y científicos, como ser artículos, congresos, libros, reportes y tesis de grado como de posgrado. Estas plataformas comparten la característica común de proporcionar acceso en línea a información descriptiva (metadatos) de los objetos que alojan (títulos, autores, resumen, palabras clave, etcétera). Estos metadatos pueden basarse en uno o más esquemas de metadatos predefinidos (Dublin Core [Dublin Core 17], MODS [MODS 17], EDT [EDT 17], etc), y también pueden organizarse bajo una representación interna específica, propia del software de gestión en uso. Algunas de estas plataformas permiten acceso a los metadatos de los objetos digitales bajo algún servicio web (ej. OAI PMH [Open Archives 17] o REST), y en muchos casos también exponen parte de la información de manera estandarizada (por ejemplo, campos META [W3C17] dentro del HTML como ser DC.title, DC.date o DC.type). Estos casos resultan particularmente interesantes ya que, al utilizar herramientas y/o formatos estandarizados, es relativamente sencillo extraer los metadatos descriptivos para ser utilizados en otros contextos y por otras aplicaciones. Sin embargo, el reto es realizar la extracción de metadatos a partir del código HTML utilizado para su exposición, en particular cuando este código no sigue un esquema estandarizado. El objetivo de este trabajo es precisamente proponer una herramienta que realice esta tarea de extracción de metadatos, a fin de poder hacer uso de las información descriptiva de los recursos incluso ante la falta de estandarización de dichos metadatos. Esta herramienta permite compartir metadatos entre contextos diferentes, como ser una editorial comercial y un repositorio digital, o un sistema de gestión y evaluación científica (current research information system, CRIS [USDA 17]) y un portal de revistas académicas. Esto puede utilizarse para agilizar la verificación de los datos puesto que se puede comparar la información guardada en, por ejemplo, un repositorio con la editorial donde pertenece el documento, o para automatizar la carga de contenido de un documento en los formularios de autoarchivo, entre otros usos posibles. Además, si bien aquí se trabaja con artículos en publicaciones periódicas, la herramienta propuesta ha sido diseñada para incluir a futuro otros tipos de documentos así como también otros espacios en la web de productores de dichos documentos.
El objetivo de esta tesina es estudiar técnicas de extracción de metadatos a partir de documentos HTML que no necesariamente siguen un esquema de catalogación estandarizado, e implementar un prototipo de servicio web capaz de analizar documentos HTML a fin de extraer sus metadatos y retornar estos a la plataforma que lo solicita. Para que el objetivo sea alcanzable en un marco de tiempo acotado, se trabajará sobre un tipo particular de objeto digital (artículos de revistas) y se seleccionará un conjunto de sitios web objetivo que servirán como ejemplo para otros casos similares; este conjunto incluirá sitios web de grandes editoriales, repositorios digitales, portales institucionales de revistas científicas y bases de datos internacionales.

Desarrollos propuestos

En primer lugar, el desarrollo propuesto constituye en un servicio web que recibe peticiones de aplicaciones en línea. Estas peticiones consisten en (al menos) una URL [RFC 17] de un

artículo digital. Al recibir estas peticiones, la aplicación recupera el documento HTML correspondiente y analiza su estructura interna a fin de identificar la mayor cantidad de metadatos posibles. Una vez completado el análisis y extracción de metadatos, generará una representación interna que utilizará para enviar como respuesta a la aplicación solicitante. En esta etapa inicial, para el desarrollo y validación del prototipo y su capacidad de análisis de documentos HTML, se toma una muestra de sitios web representativos, que incluye grandes grupos editoriales, responsables de más de la mitad de las publicaciones científicas en varios campos, portales de revistas académicas que funcionan sobre Open Journals System (OJS [OJS 17]), el software desarrollado por Public Knowledge Project (PKP) [PKP 17], y repositorios digitales que funcionan sobre herramientas ampliamente utilizadas como DSpace [DSpace 17].

El prototipo incluye también un ejemplo de cliente que sirve para integrar el servicio web con un sistema de terceros. En este caso, se ha generado una extensión implementada para un navegador web, que tomará las peticiones a través de un formulario donde se ingresará la URL de un artículo publicado en una página web, este recuperará el código HTML de dicha página e identificará los metadatos que contenga. A partir de esta información se realizará el análisis y extracción de los metadatos y se generará una representación interna, que se utilizará para autocompletar los campos en un formulario de carga de publicaciones en la aplicación solicitante.

Capítulo 2 | Estado del arte

Resumen

En este capítulo, se describe en primer lugar el formato de publicaciones que la herramienta utiliza como base en su prototipo de prueba. Para las evaluaciones, entonces, se seleccionó el formato artículo publicado en revistas científicas. Esta decisión fue tomada para completar la tesina en un tiempo acotado, y aprovechando el hecho de que estas revistas están alojadas en distintos sitios web que pertenecen al ámbito de estudio.

En segundo lugar, se detallan los distintos tipos de sitios donde la herramienta realiza la extracción de metadatos, a saber: sitios web de grupos editoriales comerciales, portales de revistas académicas y repositorios digitales. Para las pruebas se seleccionaron sitios web que pertenezcan a estas categorías.

En tercer lugar, se muestran las distintas herramientas que en la actualidad se utilizan para interoperar entre los sitios web que aplican normalización sus metadatos, lo que servirá para contextualizar este desarrollo y, principalmente, mostrar la importancia de la misma en contextos donde no existen dichas herramientas.

En cuarto lugar, se explica la técnica de *web scraping,* utilizada para recuperar, analizar y transformar los documentos HTML con los que se trabaja. Se detallan los web scrapers que existen actualmente y su forma de trabajo.

Finalmente se describe el concepto de metadato (acotado a la catalogación de recursos académicos), cuáles son sus usos y qué formatos existen.

Artículo Científico

Un artículo científico [Simon 13] es un trabajo de investigación publicado en una revista especializada en un cierto tema. Este tipo de documento tiene como objetivo difundir de manera clara y precisa los resultados de una investigación realizada sobre un área determinada de estudio.

Los artículos que se encuentran en las revistas científicas mantienen una estructura que comienza por el título del artículo, el nombre de sus autores, un resumen del trabajo y un esquema que contiene: introducción, materiales y métodos, resultados y discusión.

Tipos de artículos

Hay varios tipos de artículos científicos[Versione 13], los más habituales son:

- Las cartas o comunicados, que representan importantes hallazgos en investigación. Suelen publicarse más rápidamente puesto que se consideran urgentes.
- Las revisiones o síntesis sobre un tema en particular.
- Los artículos (o *papers*) que son una descripción completa de los resultados de una investigación original.
- El material suplementario, esta es una variante que no posee la estructura de un artículo sino que sirve para exponer la información experimental o gráfica obtenida de los artículos originales.

Revista científica

Una revista científica es una publicación, digital o impresa, en la que se difunde el progreso de la ciencia exponiendo los artículos que anteriormente se han expuesto. En general, las revistas científicas tienden a realizar publicaciones de artículos altamente especializados en un área, aunque algunas de las más antiguas (como por ejemplo *Nature* y *Science*) publican artículos en un amplio rango de campos científicos.

Proceso de edición

El proceso editorial de una publicación es el proceso típico que realizan las revistas científicas, portales de revistas y editoriales tanto institucionales como comerciales. Este proceso se puede dividir en cinco grandes etapas, en las que trabajan distintos actores. La tabla 1 se expone la relación entre ellos[Villarreal 17]:

ETAPA	ACTOR
1. Propuesta: flujo de artículos enviados. Comienza por el editor de la revista.	● Editor ● Autor

2. Revisión: el autor siempre conocerá el estado de su artículo, la revisión hecha por los pares y su aceptación o rechazo.	● Editor ● Evaluador ● Corrector ● Autor
3. Edición: los artículos son enviados a maquetación, diagramación, revisión de estilo y sintaxis.	● Editor ● Diagramador ● Maquetador
4. Publicación: los artículos son programados para ediciones presentes o futuras sin límite de tiempo.	● Editor
5. Difusión	● Editor

Tabla 1: Etapas del proceso de edición y los actores que realizan las labores dentro de las etapas de edición.

Dentro de los distintos de los roles que se muestran en la tabla 1 todos los tipos de actores que ayudan a este proceso aunque muchas veces el trabajo de corrector y diagramador pueden quedar incluidos en el trabajo de revisor y maquetador. Describiendo las tareas de cada uno de ellos:

- Administrador: es el encargado de gestionar y configurar el sistema en general, crear revistas, funciones administrativas, soporte técnico y diseño gráfico.
- Gestor: es el encargado de iniciar y configurar la publicación, adicionalmente maneja los usuarios y los roles en el proceso editorial.
- Editor: supervisa todo el proceso editorial, inicia el proceso mediante la asignación de los artículos que ingresan al sistema (enviados por los autores) a los coeditores para continuar su revisión, realizan la planeación de los números siguientes y el contenido de estos.
- Coeditor o asistente editorial: supervisa el envío, mediante su revisión y re envío a pares académicos y correctores de estilo, diagramación y normalización. Así mismo envía las novedades al autor para que conozca en todo momento del estado de su artículo.
- Evaluador: se encarga de la revisión analítica del artículo, su pertinencia y alcance investigativo. Es la persona encargada de dar la aprobación acerca del contenido y su calidad para ser publicado.

- Corrector de estilo: trabaja en la gramática y claridad para expresar las ideas del autor, realiza preguntas al autor para encontrar posibles errores e inconsistencias gramaticales, se asegura que el artículo cumpla con los lineamientos bibliográficos y de estilo requeridos por la publicación.

- Diagramador: transforma el documento final ya corregido en un artículo gráfico que cumple los lineamientos de imagen institucional predefinidos, en forma, fuentes, tamaños y colores. Crea los archivos de documentos finales (HTML, PDF, DOC) para su publicación electrónica.

- Maquetador: lee los documentos finales (pruebas de galeras) para encontrar errores tipográficos y de formato previos a la publicación.

Portales de Revistas

Los portales de revistas están orientados a la difusión de la investigación y al apoyo de la edición de revistas científicas, tanto en papel como electrónicas. En estos sitios se albergan una diversidad de revistas producidas por los grupos y programas de investigación de las diferentes áreas del ámbito académico. En el ámbito de las revistas académicas, una de las herramientas informáticas más utilizadas para la gestión y publicación de publicaciones periódicas es el software Open Journal System, que se detalla a continuación. Asimismo, en el ámbito privado, las editoriales suelen utilizar herramientas propias, realizadas a la medida de sus necesidades y objetivos.

Sistemas para la gestión de un portal de revistas: Open Journal System (OJS)

Según la iniciativa Public Knowledge Project (PKP, Proyecto de Conocimiento Público):

“Open Journal Systems (OJS) es un Sistema de Administración y publicación de revistas y documentos periódicos (Seriadas) en Internet. El sistema está diseñado para reducir el tiempo y energías dedicadas al manejo exhaustivo de las tareas que involucra la edición de una publicación seriada. Este sistema permite un manejo eficiente y unificado del proceso editorial, con esto se busca acelerar el acceso en la difusión de contenidos e investigación producido por las Universidades y centros de investigación productores del conocimiento. Así mismo, busca consolidarse como una herramienta con innovaciones que permite el acceso en texto completo de los documentos publicados.OJS es una solución de software libre que es desarrollado por el Public Knowledge Project (PKP), Canadá, que está dedicado al aprovechamiento y desarrollo de las nuevas tecnologías para el uso en investigación académica. PKP trabaja a través de sus esfuerzos, financiados con fondos federales, con el fin de expandir y mejorar el acceso a la investigación.”

Se seleccionó este software como ejemplo y como parte de prueba para el uso de la herramienta dada la cantidad de instalaciones y de instituciones que lo utilizan y porque su comunidad ha ido creciendo a lo largo de los últimos años. Como se muestra en el siguiente gráfico, al día de hoy se han registrado más de diez mil revistas que utilizan OJS [OJS Stats 17]:

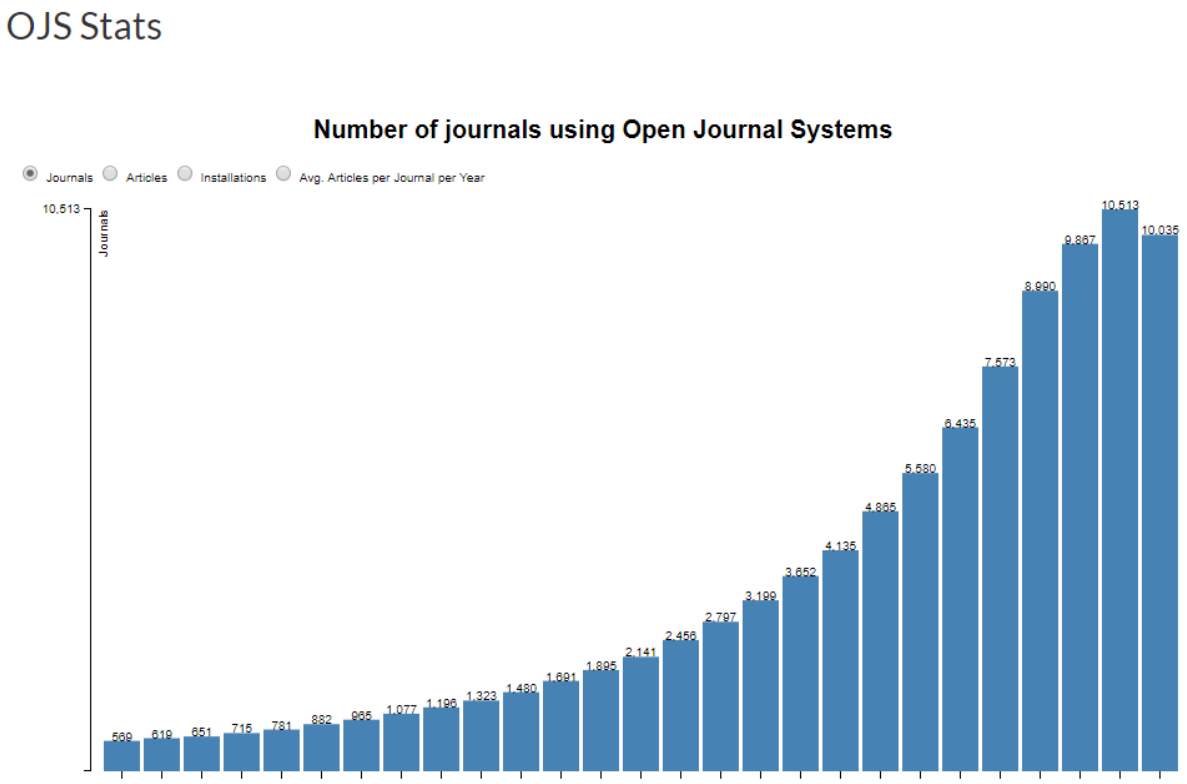


Figura 1: Número de revistas utilizando OJS. Se observa un crecimiento acelerado en cantidad de instalaciones de OJS en los últimos 10 años.

Ejemplo de uso del Software OJS

Para los efectos de prueba de la herramienta que se realiza en esta tesina, se ha tomado como ejemplo de uso del Portal de Revistas de la Universidad Nacional de La Plata, que utiliza el software OJS en su tercera versión, y el Portal de Revistas de la Universidad de Costa Rica, que utiliza el mismo software pero en su segunda versión. Se han tomado estos casos porque la versión 3 de OJS ha introducido cambios mayores en la presentación de artículos a los usuarios, con lo cual entre estos portales se cubren todas las alternativas de exposición web de artículos sobre dicha plataforma.

Portal de la Universidad Nacional de La Plata

Este Portal [Portal LP17] fue creado para: “facilitar la gestión de las revistas, su publicación y su difusión en línea. El portal sirve a revistas existentes que no cuentan con una versión digital, a revistas existentes que deseen informatizar sus procesos y a grupos que desean iniciar una nueva revista.”.

En el año 2008 la Universidad Nacional de La Plata y la Secretaría de Ciencia y Técnica realizaron un relevamiento para conocer cuántas y cuales son las revistas que se generan en la UNLP. A partir de este estudio se reveló la existencia de más de sesenta publicaciones, donde algunas cuentan con una versión digital además de la versión tradicional en papel.
A partir de esto, la UNLP ha creado el proyecto denominado Portal de Revistas de la UNLP, cuyo objetivo es servir a las revistas existentes que no cuentan con una versión digital, a revistas existentes que deseen informatizar sus procesos y a grupos que desean iniciar una

nueva revista. Actualmente el portal aloja alrededor de veinte revistas de distintas áreas pertenecientes a la UNLP y se ha actualizado a la última versión del software OJS..

Participantes y roles principales

El rol principal de la Secretaria de Ciencia y Tecnología es la de servir de puente entre los gestores de las revistas y el equipo técnico. Cada revista cuenta con un Gestor, quien se encarga de definir los parámetros globales propios de la revista (políticas, comité editorial, idiomas, plugins). El soporte técnico a los gestores de cada revista, formación de equipos editoriales, mantenimiento del servidor y gestión de la plataforma se realiza desde PREBI-SEDICI.

Portal de la Universidad de Costa Rica

Según el sitio web de la Universidad de Costa Rica [Portal CR17]: “Este portal fue diseñado por el equipo del proyecto UCRIndex con el fin de contribuir en la difusión de la ciencia que se publica en Costa Rica.”

Este portal, perteneciente a la Universidad de Costa Rica y también creado en el año 2008, comenzó con el objetivo de enfrentar las dificultades que tenían las revistas de la Universidad, entre ellas: el atraso en la impresión de las versiones de papel y el incumplimiento de la periodicidad de la publicación de las revistas. Para contrarrestar esas problemáticas la Universidad de Costa Rica tomó la iniciativa de utilizar un software que .

El portal ofrece entre otras cosas: espacio en el servidor, asistencia en el diseño gráfico de la revista, guía para usar el software OJS. Entre sus actividades se destacan:

- Brindar una versión digital de todas las revistas de la Universidad de Costa Rica.
- Dar propuestas para agilizar la gestión editorial de las revistas dentro de la Universidad.
- Experimentar con recursos tecnológicos, para que las revistas en línea no sean una “copia del papel” (video, multimedia; foros, interacción con el usuario, etc.) ¿Más que
- Implementar funciones para automatizar el proceso de gestión editorial
- Coordinar la evaluación de doble ciego
- Realizar la documentación sobre su proceso editorial.

Actualmente este portal aloja a más de cincuenta revistas de la Universidad de Costa Rica y está por realizar su actualización a la última versión del Software OJS.

Editorial

Una compañía editorial tiene como objetivo el de producir, difundir y distribuir obras, sobre las que realiza las tareas de producción y difusión. También este tipo de organizaciones fomentan la participación de la comunidad académica: investigadores, docentes, graduados y alumnos en la publicación y difusión de sus escritos, sabiendo que la diversidad de esta producciones ayudan a enriquecer la identidad de la Editorial.

Ejemplos de editoriales comerciales

Editoriales seleccionadas:

Según un artículo publicado en ABC Ciencia, cuyo trabajo fue dirigido por el investigador de la Universidad de Montreal Vincent Lariviére, se encontró que los grupos editoriales Reed-Elsevier, Taylor & Francis, Wiley-Blackwell, Springer y Sage controlan más de la mitad de la difusión científica: desde el año 2006 más del 50% de todas las publicaciones científicas son publicadas en revistas que se gestionan dentro de alguna de estas grandes editoriales. Se seleccionaron dos de estas grandes editoriales, Springer Nature (que actualmente aloja a más de tres mil revistas científicas) y Elsevier (que alberga casi la misma cantidad) para realizar las pruebas de extracción de metadatos desde sus portales web.

Springer Nature

Springer Nature [Springer 17] es una de las editoriales de revistas científicas más influyentes del mundo y pionera en el campo de la investigación abierta. Esta editorial posee sitios mundialmente conocidos en los que publica miles de artículos que ayudan a los investigadores a avanzar en las investigaciones en las que trabajan.

Springer publica revistas que pertenecen al área de la ciencia, la tecnología, la medicina y las ciencias sociales a través de miles de títulos específicos de sus disciplinas, donde proporciona revisiones de alto impacto de sus campos a través de títulos multidisciplinarios de Nature Research.

Elsevier

Según la página oficial de Elsevier España [Elsevier 17], la editorial se describe como: "proveedor de soluciones de información y contenidos actualizados, fiables y adaptados a las necesidades de investigadores, clínicos, docentes, estudiantes y demás miembros de la comunidad científica y sanitaria. Edita más de un centenar de revistas, entre las que se encuentran las cabeceras oficiales de más de 70 sociedades científico-médicas; cuenta con un amplio fondo editorial de libros de autores destacados, que conjuntamente con nuevas soluciones online, proporciona a los profesionales de la salud y la investigación científica conocimientos e información de alta calidad y amplia cobertura. Pertenece al grupo Elsevier, que con sede central en Ámsterdam y con más de 7.000 profesionales que trabajan en 24 países, es líder mundial en soluciones de información que mejoran el desarrollo de las ciencias, la tecnología y la salud, a la vez que ayuda a su profesionales a tomar las mejores decisiones y proporcionar los mejores cuidados y, a veces, realizar descubrimientos revolucionarios que ayudan a marcar nuevos límites del conocimiento y del desarrollo humano."

Repositorio institucional

Este tipo de sitio web se lo describe con un poco mas de detalle ya que, para demostrar la potencia de la herramienta, se desarrolló una extensión para el navegador web Chrome que se especializa en un software de este tipo de sitios.

Clifford Lynch [Lynch 16] define a un repositorio institucional como:

“Un Repositorio Institucional universitario es un conjunto de servicios que ofrece la Universidad a los miembros de su comunidad para la dirección y distribución de materiales digitales creados por la institución y los miembros de esa comunidad. Es esencial un compromiso organizativo para la administración de estos materiales digitales, incluyendo la preservación a largo plazo cuando sea necesario, así como la organización y acceso o su distribución”.

Por otro lado, la organización SPARC (Scholarly Publishing and Academic Resources Coalition) destaca 3 características sobre los Repositorios Institucionales [CNI 13]:

- Pertenecen a una institución.
- Son de ámbito académico.
- Son acumulativos y perpetuos.

Un repositorio institucional es una base de datos compuesta de un grupo de servicios destinados a capturar, almacenar, ordenar, preservar y redistribuir la documentación académica en formato digital. Los repositorios institucionales tienen objetivos como: gestionar información sobre educación, investigación y recursos de forma más efectiva y transparente y que la investigación y la producción científica se encuentren disponibles, apoyando el desarrollo de nuevas relaciones entre los académicos y los centros de investigación, tanto nacionales como internacionales. Sirven como ventana al mundo para mostrar toda la producción intelectual generada por las instituciones, y en muchos casos permiten realizar la difusión de documentos generados por otras áreas de la institución, más allá de los resultados de investigaciones: resoluciones [Texier 13], entrevistas, objetos de aprendizaje, etc. [De Giusti 13], y en muchos casos brindan servicios de valor agregado para alcanzar a un público más amplio, como ser la difusión científica [De Giusti 15], el soporte a la enseñanza [Texier 12] y la adaptación de formatos para brindar mayor accesibilidad [De Giusti 16]

Este tipo de sitio web es entonces, un archivo electrónico de la producción científica de una institución, almacenada en un formato digital, en el que se permite la búsqueda y la recuperación para su posterior uso nacional o internacional. Los repositorios no se conciben como sistemas aislados, sino que son diseñados como plataformas con capacidad de interoperar con otros sistemas (otros repositorios, directorios, portales de revistas, etc.), y para ello incluyen herramientas que brindan estas capacidades [De Giusti 14] [De Giusti 13].

Software para gestión de Repositorios institucionales

Dentro de las plataformas más conocidas utilizadas como Repositorios Institucionales encontramos:

- DSpace
- EPrints [EPrints 17]
- Digital Commons [DigitalCommons 17]
- WEKO [WEKO 17]
- OPUS [OPUS 17]

- dLibra [dLibra 17]
- CONTENTdm [CONTENTdm 17]

Según datos recopilados por OpenDOAR, Directorio de Repositorios de Acceso Abierto (The Directory of Open Access Repositories), existen en la actualidad más de 3300 repositorios digitales de acceso abierto [OpenDOAR 17], y cerca del 80% de ellos utiliza alguna de las plataformas de software antes mencionadas:

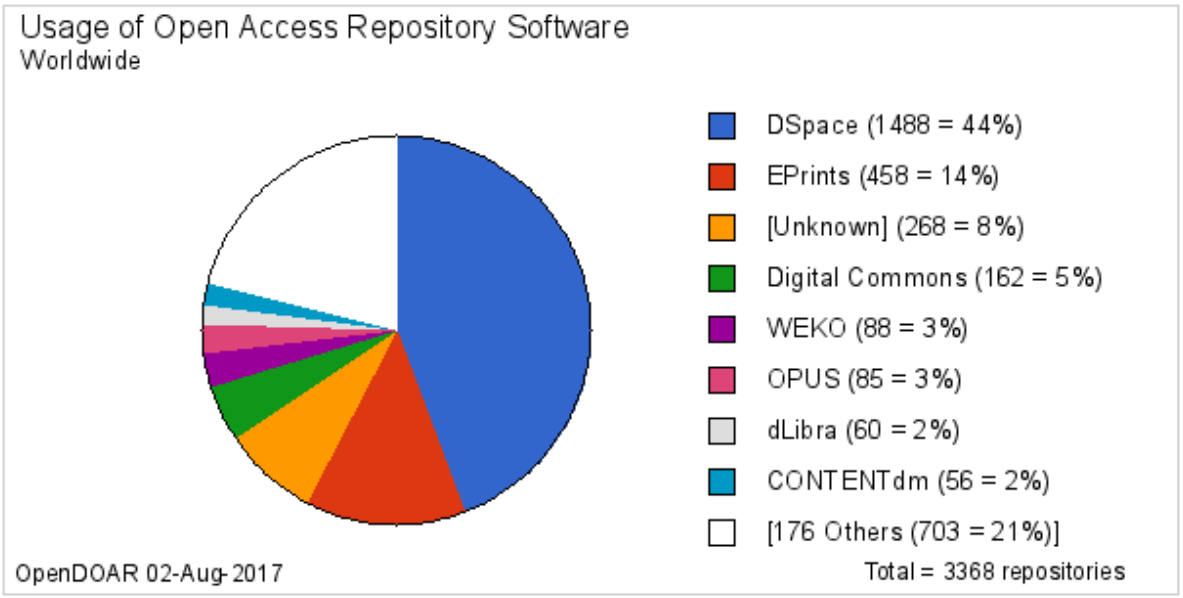


Imagen 1. Gráfico de uso de los distintos software para repositorios digitales de acceso abierto.

Como puede observarse en la imagen 1, la plataforma DSpace es la más utilizada para la implementación de repositorios digitales: 44% de los repositorios lo utilizan, con una tendencia ascendente en los últimos años. Su gran aceptación por la comunidad internacional, sumado al hecho de que tanto la Universidad Nacional de La Plata como la Comisión de Investigaciones Científicas de la Provincia de Buenos Aires, ambas instituciones en las que se enmarca este trabajo, utilizan DSpace para sus repositorios (SEDICI y CIC-Digital respectivamente), se ha tomado dicho software para realizar las pruebas en la herramienta aquí propuesta.

Dspace

DSpace es un software de código abierto pensado para la gestión de repositorios digitales que proporciona distintas herramientas y funcionalidades que permiten satisfacer las diferentes necesidades que requieren las instituciones.

En la actualidad, como se aprecia en el gráfico de la imagen 1, cerca de la mitad de los repositorios del mundo están desarrollados sobre DSpace. Esto, se debe a sus distintas características y funcionalidades que lo distinguen entre las herramientas pensadas para la misma finalidad. Entre estas características se encuentran:

- Posee una gran comunidad de usuarios y programadores a nivel mundial que lo mejoran y actualizan constantemente
- Es un software de código abierto (open source) disponible gratuitamente para cualquier persona. El código utiliza la licencia BSD (Berkeley Software Distribution) [FreeBSD 17] que permite a cualquier institución usarlo, modificarlo y agregar sus propias modificaciones al código.
- Es completamente personalizable a las necesidades de cada institución, cada una de ellas puede modificar la forma en la que se verá su repositorio, los formatos de metadatos que utilizará, la forma y los campos de búsqueda que permitirá, los protocolos y directrices de interoperabilidad que aplicará, entre otros.
- Puede almacenar y manejar todo tipo de contenido digital. DSpace puede reconocer y manejar distintos tipos de formatos de archivos y Mime Types [DuraSpace 15]

Ejemplo de uso de DSpace: SEDICI

Según su página web, el Servicio de Difusión de la Creación Intelectual, o simplemente SEDICI [SEDICI 17] es el repositorio institucional central de la Universidad Nacional de La Plata y como tal su misión es albergar, preservar, difundir y dar visibilidad a nivel mundial a toda la producción científica e intelectual de las distintas unidades académicas que la componen. Este repositorio institucional fue creado en el año 2003 con el propósito expuesto anteriormente y desde entonces sus colecciones han crecido de manera contínua. Además de artículos de revistas científicas donde publican los docentes e investigadores de la UNLP, y de todas las revistas académicas generadas en el marco de la UNLP a través de los grupos, cátedras, departamentos y secretarías, como también laboratorios, centros e institutos de investigación y desarrollo, materiales de particular interés para el desarrollo de la herramienta, este repositorio aloja todo tipo de producciones y materiales generados en todos los ámbitos de la institución: entrevistas de la Radio Universidad, libros, conferencias, objetos de aprendizaje, tesis de grado y de posgrado, series pictóricas, actas y resoluciones, convenios, etcétera.

Metadato

Un metadato [Senso 03] es información que se aplica sobre un recurso para facilitar su organización, es sencillamente un dato que describe otro dato, información estructurada para describir, explicar, localizar o facilitar la obtención, uso o administración de un recurso de información. La norma ISO 15489-1 del 2001 define los metadatos, en el contexto de la gestión de documentos, como:

"datos que describen el contexto, contenido y estructura de los documentos, así como su gestión a lo largo del tiempo (...) Como tales, los metadatos son información estructurada o semiestructurada que posibilita la creación, registro, clasificación, acceso, conservación y disposición de los documentos a lo largo del tiempo y dentro de un mismo dominio o dominios diferentes."

Debido a la gran diversidad y volumen de las fuentes y recursos en Internet, se hizo necesario establecer un mecanismo para etiquetar, catalogar, describir y clasificar los

recursos presentes en la World Wide Web con el fin de facilitar la posterior búsqueda y recuperación de la información. Este mecanismo los constituyen los llamados metadatos.

Existen distintos modelos de metadatos, cada uno de ellos con distintos esquemas de descripción. En los distintos modelos, cada objeto se describe por medio de una serie de atributos y el valor de estos atributos es el que puede servir para recuperar la información. Dependiendo de la clase de metadatos puede existir: información sobre elementos de datos o atributos, información sobre la estructura de los datos, información sobre un aspecto concreto, etc. De forma general, existen metadatos referidos a:

- el contenido (concepto)
- aspectos formales (tipo, tamaño, fecha, lengua, etc.)
- información del *copyright*
- información de la autentificación del documento o recurso
- información sobre el contexto (calidad, condiciones o características de acceso, uso, etc.)

Papel de los metadatos

Los metadatos se especializan para cumplir varios roles, entre ellos se destacan:

- Recuperación de la información
- Administración de documentos
- Gestión de derechos, autoría y propiedad intelectual
- Estado de archivo
- Control y descripción de procesos
- Seguridad y autentificación
- Valoración de contenidos
- Preservación y conservación
- Visibilidad de la información
- Actualización de la información

Tipo de metadatos

Como comentamos anteriormente, existen distintos tipos de metadatos, cada uno con un objetivo en particular [De Giusti 14a]. Se distinguen:

- Administrativos
- Estructurales
- Descriptivos

- Técnicos
- De uso
- De preservación

Se describen los usos y ejemplos de cada uno de estos tipos en la siguiente tabla:

TIPO	**USO**	**EJEMPLOS**
Administrativo	Usados en la identificación, gestión y administración de recursos de información	Adquisición de información Derechos y reproducción Requerimientos legales para el acceso Localización de información Criterios de selección para la digitalización Control de la versión
Descriptivo	Utilizados para representar recursos de información	Registros catalográficos Proporcionar ayuda en la búsqueda Índices especializados Hiperenlazar relaciones entre recursos Anotaciones de los usuarios
Preservación	Para salvaguardar los recursos de información	Informar sobre las condiciones de uso de los recursos físicos. Informar sobre las acciones llevadas a cabo para preservar versiones físicas y digitales de recursos
Técnico	Relativos a cómo funcionan los sistemas o el comportamiento de los metadatos	Documentación de hardware y software. Digitalización de la información (formato, ratio de comprensión, ..) Autentificación y datos de seguridad (encriptación, contraseña,..) Control de tiempo de respuesta de sistemas
Uso	Relativos al nivel y tipo de uso que se hace con los recursos informativos	Información sobre versiones. reutilización del contenido del recurso

Tabla 1. Descripción y usos de los distintos tipos de metadatos.

Metadatos Descriptivos

En los repositorios institucionales, portales de revistas y algunas editoriales, el tipo de metadato utilizado es el descriptivo. Por esta razón, se hace énfasis en este rol con la tabla aquí debajo donde se exponen:

- Objetivos de este tipo de metadato
- Elementos de muestra
- Implementaciones de este tipo de metadato

TIPO	**OBJETIVO**	**ELEMENTOS DE MUESTRA**	**IMPLEMENTACIONES DE MUESTRA**
Metadatos descriptivos	Descripción e identificación de recursos de información en el nivel local para permitir la búsqueda y la recuperación (por ejemplo, búsqueda de una colección de imágenes para encontrar pinturas con ilustraciones de animales); en el nivel Web, permite a los usuarios descubrir recursos (por ejemplo, búsqueda en la Web para encontrar colecciones digitalizadas sobre poesía).	identificadores únicos (PURL, Handle) atributos físicos (medios, condición de las dimensiones) atributos bibliográficos (título, autor/creador, idioma, palabras claves)	Handle PURL Dublin Core MARC HTML Meta Tags

Tabla 2. Objetivos, elementos de muestra e implementaciones de metadatos de tipo descriptivo.

Como se observa en la tabla 2, hay una gran variedad de implementaciones del tipo de metadato descriptivo. Específicamente, para nuestro caso de estudio, los estándares utilizados en los repositorios institucionales y los portales de revistas científicas son:

- DC: *Dublin Core Metadata Initiative.* Esta descripción se declara debajo ya que se da mayor énfasis por el hecho de ser el tipo de metadato utilizado en los portales de revista seleccionados para las pruebas y en los repositorios institucionales que utilizan el software Dspace y que también pertenecen a los ejemplos de prueba.
- METS: *Metadata Encoding and Transmission Standard.*Se trata de un esquema para describir objetos de bibliotecas digitales complejas que utiliza el lenguaje XML schema y asocia metadatos administrativos y descriptivos. El estándar es mantenido por la Network Development and MARC Standards Office de la Biblioteca del Congreso Permite describir separadamente archivos digitalizados (por ejemplo las distintas páginas de un libro).
- MODS: *Metadata Object Description Schema.* Es un esquema de metadatos descriptivo que se deriva del MARC 21 y que intenta permite crear la descripción de recursos originales o seleccionar los registros existentes en MARC 21. Utiliza el lenguaje y la sintaxis XML y puede utilizarse como un formato específico de la Próxima Generación de Z39.50 .
- EAD: *Encoded Archival Description.*Se trata de un proyecto internacional que desarrolla pautas para el marcado de textos electrónicos (novelas, obras de teatro, poesía, etc.) y se enfoca al campo de las humanidades.
- TEI: *Text Encoding Initiative* [TEI 17]
- IFLA: *Metadata Resources for Digital Libraries*[IFLA 17].
- CIMI: *Computer Interchange of Museum Information.* [CIMI 17](El Consorcio cerró sus operaciones en 2003)

Dublin Core

Dublin Core [Pinilla 14] comenzó y continúa como una organización abierta desde 1995. Su objetivo es el desarrollo de estándares de metadatos interoperables.

La primera versión del estándar se componía únicamente por un pequeño conjunto de descriptores con los que se puede describir de forma sencilla un recurso.

En 2001 el organismo de normalización de los Estados Unidos aprueba como norma estatal el conjunto de descriptores de Dublin Core, dando lugar a la norma Z39-85:2001 DUBLIN CORE METADATA ELEMENT SET [ANSI/NISO 01].

El estándar DCMI [DCMI 17], cuenta con un conjunto de 15 definiciones semánticas, descriptores, 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. Estos son:

CONTENIDO	**PROPIEDAD INTELECTUAL**	**INSTANCIACIÓN**

Title	Creator	Date
Subject	Publisher	Type
Description	Contributor	Format
Source	Rights	Identifier
Language		
Relation		
Coverage		

Tabla 4. Definiciones semánticas de Dublin Core.

A su vez, estos se agrupan en 3 grandes categorías: contenido, propiedad intelectual e instanciación como se muestra en la tabla superior. 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.

Dublin Core cualificado

El Dublin Core cualificado [DCMI Elements 17] es una extensión del Dublin Core donde algunos de sus elementos son acompañados de un cualificador que los hace más restrictivos como por ejemplo:

- Title –dc.title –dc.title.alternative
- Relation –dc. isVersionOf –dc. isPartOf
- Date –dc.date.created –dc.date.available

Este esquema de metadatos es utilizado por el repositorio institucional SEDICI, sitio que será utilizado en las pruebas realizadas en la herramienta.

Herramientas y estándares de interoperabilidad

En este capítulo,–se describen las distintas herramientas que pueden utilizarse para interoperar entre los distintos sitios estudiados en esta tesina.

Entre las definiciones que pueden encontrarse sobre interoperabilidad:

La IEEE define interoperabilidad como:

“la habilidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada”

El Marco Iberoamericano de Interoperabilidad utiliza la definición dada por la Comisión Europea, definiendo interoperabilidad como:

“la habilidad de organizaciones y sistemas dispares y diversos para interaccionar con objetivos consensuados y comunes y con la finalidad de obtener beneficios mutuos. La interacción implica que las organizaciones involucradas compartan información y conocimiento a través de sus procesos de negocio, mediante el intercambio de datos entre sus respectivos sistemas de tecnología de la información y las comunicaciones.”

Se describe en esta sección herramientas que siguen estándares que tienen como objetivo interoperar entre distintos sistemas como repositorios institucionales y portales de revistas.

Z39.50

Introducción

La ANSI/NISO Z39.50[Z39.50 17] (Application Service Definition and Protocol Specification), más conocido por Z39.50 es un protocolo para la recuperación de información basado en la estructura cliente/servidor que facilita la interconexión de sistemas informáticos.

El objetivo principal del cliente Z39.50 consiste en permitir al usuario realizar búsquedas en bases de datos que cuenten con un servidor Z39.50, sin tener que conocer para ello las sintaxis de búsqueda que utilicen dichos sistemas. Uno de los beneficios básicos que ha conseguido este protocolo, es en el ámbito de las bibliotecas y de los centros de documentación, donde hace posible la comunicación entre sistemas que utilizan diferente hardware y software. Hasta el momento ha habido tres versiones del protocolo:

- Versión 1: se aprobó en 1988, y ha quedado en desuso.
- Versión 2: creada en 1992 evita las discrepancias con el protocolo "Search and Retrieve" (ISO 10162 y 10163). Incluye dos nuevas operaciones: control de acceso de los clientes y control de recursos.
- Versión 3: aprobada en 1995 y aceptada como estándar ISO (ISO 23950) en 1997, incorpora mejoras, se mantiene compatible con la versión 2 y reconoce como medio de aplicación TCP/IP e Internet.

Servicios Z

Las funciones de las que este estándar se encarga son:

1. El preámbulo, en la que se establecen los parámetros fundamentales de la sesión que se va a iniciar entre el cliente y el servidor. Esta negociación incluye: versión del protocolo, operaciones que podrán efectuarse, juegos de caracteres, lenguas, segmentación y tamaño de la información, etc.

2. La búsqueda, es la funcionalidad más importante y que puede realizarse a múltiples bases de datos, agilizando la recuperación de información.

3. La recuperación de la información: una vez realizada la búsqueda, el cliente solicita al servidor los registros que quiere visualizar, que dependiendo del número solicitado, podrán aparecer segmentados en conjuntos de registros.

4. Otras: controlar el acceso, realizar búsquedas utilizando índices, ordenar la información recuperada, y poder acceder a información sobre el servidor y los servicios que ofrece. En el ámbito bibliotecario, son muy útiles los denominados servicios extendidos que permiten archivar las estrategias y resultados de las búsquedas, actualizar bases de datos, pedir documentos, y crear especificaciones de exportación.

Tarea del estándar

La Z39.50 reconoce que la recuperación de información posee dos componentes principales: la selección de información basada en ciertos criterios, y la recuperación de esta. Para realizar esta tarea el estándar proporciona un lenguaje común para ambas actividades.

El estándar Z39.50 normaliza la forma en que el cliente y el servidor se comunican e interoperan, aún cuando existan diferencias entre los sistema, motores de búsqueda y las bases de datos. El estándar Z39.50 considera una serie de mensajes iniciales entre el cliente y el servidor que son: establecer una conexión; iniciar una sesión Z39.50 y negociar las expectativas y las limitaciones de las actividades que ocurrirán. Luego de estos pasos el cliente puede enviar la consulta. El cliente Z39.50 traduce la consulta a una representación normalizada y la pasa al servidor Z39.50. El servidor ejecuta la búsqueda en la(s) base(s) de datos, y crea un set de resultados. Entonces, el cliente puede solicitar los registros del set de resultados o solicitar al servidor un procesamiento adicional dentro del set de resultados. Al recibir los registros, el cliente puede procesarlos y mostrar al usuario.

Funcionamiento Z39.50

Para conseguir interoperabilidad entre distintos sistemas, Z39.50 facilita un lenguaje común para realizar las dos operaciones básicas que garantizan la recuperación de información: selección de información y obtención de la misma. Por ello, Z39.50 contempla la estandarización tanto de los mecanismos de codificación (cómo deben codificarse los datos para ser transferidos), como de la semántica del contenido (modela los datos con una semántica común para cada comunidad específica).

Z39.50 es un estándar muy amplío que ofrece una gran funcionalidad y atiende muy diversos entornos, no sólo el bibliotecario. El modelo de arquitectura básico del estándar Z39.50 se apoya en este concepto de semántica en función del contenido. Es decir, cada servidor ofrece una visión de sus bases de datos en función del dominio o el ámbito en que se encuentre, una representación virtual de los registros que contiene, donde la estructura lógica real de la base de datos permanece oculta, y sólo se refleja la que corresponde a la semántica de ese dominio.

Adicionalmente a la representación virtual de la base de datos, se definen y registran para cada ámbito los puntos de acceso que se pueden emplear en las consultas (Attribute Sets) y las maneras de estructurar los datos al facilitarselos al cliente (Schemas). En el caso de la comunidad bibliográfica, se ha registrado el Bib-1 Attribute set y distintos formatos para presentar las respuestas, como los distintos formatos MARC, el formato propio del Catálogo de Acceso Público en Línea (Online Public Access Catalog | National Archives, OPAC) [OPAC 17], etc.

Empleo del estándar

Las aplicaciones más destacables a efectos de tareas bibliotecarias son:

- OPACs: acceso a las bases de datos más importantes del mundo, o a fuentes locales con una sola búsqueda.
- Catalogación: búsqueda y captura de registros bibliográficos, lo que supone un ahorro de tiempo y trabajo para las bibliotecas. También destaca la posibilidad de construir un catálogo colectivo virtual sin interferir en los métodos y procesos de la organización individual.
- Préstamo interbibliotecario: es la consecuencia inmediata de un catálogo colectivo virtual.
- Difusión Selectiva de la Información (DSI): El usuario puede especificar y grabar estamentos de búsqueda para ser ejecutados posteriormente, pudiéndose ejecutar las búsquedas cuando se quiera.
- Bases de datos comerciales: existen cientos de proveedores de servicios de información comercial disponibles (Dialog, Lexis Nexis, EBSCO, Chemical Abstracts.). El Z39.50 reduce la complejidad de las búsquedas en bases de datos diferentes.
- Búsqueda web y filtrado
- Actualización de bases de datos

Ventajas

Las ventajas del Z39.50 aplicado al entorno de las bibliotecas pueden ser muchas, y su importancia dependerá de cada usuario potencial y de sus necesidades ya sea que se trate de un usuario final o un bibliotecario. A grandes rasgos se destaca:

- Relacionar bases de datos diferentes.
- Realizar peticiones simultáneamente a diferentes bibliotecas, propiciando un ahorro de tiempo al realizar búsquedas de ítems poco comunes o que contengan muchos registros.
- Sencillez en la localización de la información sin que el usuario tenga la necesidad de aprender el manejo de los motores de búsquedas de diferentes sistemas y bases de datos.

- Compartir fuentes de información.
- Permite la localización de información en forma rápida y precisa evitando la compra de fuentes de información disponibles en otros centros.
- Catálogos colectivos virtuales.
- Permite realizar búsquedas en varias bases de datos de forma sencilla facilitando a los catalogadores intercambiar registros catalográficos ahorrando así recursos en la catalogación y clasificación de los materiales.
- El formato básico de intercambio es el formato MARC.
- Permite facilitar la interconexión entre usuarios de información y las bases de datos donde se encuentra la información que necesitan a partir de una interfaz común y de fácil manejo, independientemente del lugar en que las bases de datos se encuentren, que estructura de la base de datos y la forma de acceso.

Desventajas

Es importante destacar que ningún desarrollo comercial, ni particular, soporta el estándar completo definido para todos los entornos, aunque el propio estándar describe lo mínimo que deben cumplir todos los desarrollos para garantizar la interoperabilidad. Estas diferencias de un desarrollo a otro conlleva ciertos problemas. También este estándar tiende a ser bastante complejo en comparación de sus sucesores por lo que su uso está limitado a los conocedores del ámbito bibliotecario y de repositorios institucionales.

SRU & SRW

Las herramientas SRW/U (Search & Retrieve Web Service) [SRW7U17] son, según la traducción realizada en el sitio web de OCLC Research [OCLC 17]: "es parte de un esfuerzo colaborativo internacional para desarrollar una interfaz estándar de búsqueda de texto basada en la web. Se basa en gran medida en los modelos abstractos y la funcionalidad de Z39.50, pero elimina gran parte de la complejidad. SRW se construye utilizando herramientas comunes de desarrollo web (WSDL [WSDL 01], SOAP [SOAP 00], HTTP [HTTP 17] y XML [XML 17]) y el desarrollo de interfaces SRW a los repositorios de datos es significativamente más fácil que para Z39.50. Además, estos formatos de registro arcano como MARC y GRS-1 han sido reemplazados por XML".

SRU (Servicio de búsqueda y recuperación de URL) es una alternativa basada en la URL a SRW. Los mensajes se envían a través de HTTP utilizando el método GET y los componentes de la solicitud SRW SOAP (Simple Object Access Protocol) se asignan a parámetros HTTP simples. La respuesta a una petición de SRU es idéntica a la respuesta a una petición SRW, con el paquete de SOAP eliminado. SRU se convirtió en un estándar de OASIS (Organización para el Avance de Estándares de Información Estructurada) en febrero de 2013. El sitio web oficial de SRU, alojado por la Biblioteca del Congreso, proporciona acceso al estándar publicado (incluyendo esquemas XML asociados).

SRW/U se está desplegando como la API de búsqueda para la iniciativa DSpace. Está siendo considerada como la API de búsqueda estándar por un número de comunidades, incluyendo la meta-búsqueda y las comunidades de búsqueda geoespacial.

Búsqueda remota: SRU

Se caracteriza por enviar la expresión de búsqueda (y cualquier otra indicación) dentro de una URL.

Esto es, todos los comandos necesarios para que el servidor entienda una petición y lleve a cabo las acciones pertinentes se envían dentro de la URL misma acciones pertinentes, se envían dentro de la URL misma de la petición.

Búsqueda remota: SRW

Al igual que su semejant SRU, trabaja sobre tecnologías actuales y muy difundidas: XML y HTTP, pero presenta una importante diferencia: el envío de la petición se una importante diferencia: el envío de la petición se realiza mediante un POST al servidor, en el que se envía un documento XML que contiene todas las envía un documento XML que contiene todas las instrucciones y datos correspondientes. Esto es, la consulta al servidor se "empaqueta" en XML y se envía, recibiendo XML como respuesta (al igual que se envía, recibiendo XML como respuesta (al igual que en el caso de SRU).

Las reglas y restricciones utilizadas para armar e interpretar el paquete XML están dadas por el protocolo SOAP. SOAP fue creado y es mantenido por la W3C, en el área de los Web Services. SOAP es un protocolo estándar y muy difundido. Casi cualquier lenguaje de programación moderno tiene librerías para trabajar con SOAP.

Desventajas

En primer lugar, se debe conocer el lenguaje CQL para realizar consultas al servidor. Esto hace que los que utilicen el sistema tengan que tener conocimientos de programación.

En segundo lugar es obligatorio utilizar librerías SOAP en el sistema tanto receptor como emisor.

En tercer lugar, los resultados se devuelven en XML dando por obligatorio normalizar los datos a ese tipo de documento.

En último lugar, el protocolo SRU-SRW tiene una esquematización precisa, delimitada por campos obligatorios y campos opcionales, tanto para la formulación de las preguntas y los parámetros de respuesta. Por lo que no es posible su comunicación con otros tipos de servidores puesto que estos puede que no tengan la información solicitada para comunicarse.

Open Archives Initiative – Protocol for Metadata Harvesting (OAI-PMH)

OAI (Open Archives Initiative) es una iniciativa para desarrollar y promover estándares de interoperabilidad que faciliten la difusión de contenidos así como el intercambio de formatos bibliográficos entre distintos repositorios digitales y portales de revistas. La idea básica que fomenta OAI es crear una forma de intercambiar información entre repositorios o portales de revistas heterogéneos que alberguen cualquier objeto que contenga metadatos asociados. Esta iniciativa surgió a partir de los servidores de documentos en acceso abierto que habían aparecido en distintas disciplinas científicas: arXiv [arXiv 17] en Física, RePEc [RePEc 17]

en Economía, CogPrints [Cogprints 17] en Psicología, NCSTAR [ncst rl17] en Informática y NDLTD [ndltd 17] para tesis. Su objetivo inicial fue estudiar la interoperabilidad de los distintos servidores con objeto de facilitar el intercambio de datos entre los mismos. La iniciativa comenzó en la Convención de Santa Fe. La iniciativa se concretó en el desarrollo del protocolo OAI-PMH para permitir el intercambio de estos metadatos, y cuya primera versión apareció en Enero de 2001.

Aunque inicialmente se creó para ser aplicado a depósitos de documentos en acceso abierto, rápidamente se vio que podía implementarse sobre cualquier material almacenado en soporte electrónico que requiriese la comunicación de metadatos. La importancia de OAI-PMH se puede resumir en como: "OAI-PMH está llamado a ser a las bibliotecas digitales lo que HTTP es hoy al web".

Es importante señalar que OAI-PMH trata exclusivamente de la comunicación de metadatos, no de los textos completos de los documentos que son referenciados. De este se pueden ver tres características importantes:

- Simplicidad: Los creados del protocolo tuvieron en cuenta los problemas de implementación que habían tenido otras iniciativas como Z39.50.
- Normalización: basado en estándares ampliamente utilizados en Internet como son el protocolo HTTP (HyperText Transport Protocol) para la transmisión de datos y órdenes y XML (eXtended Markup Language) para la codificación de los metadatos.
- Recolección: frente a otros sistemas de agregación de contenidos como la búsqueda distribuida (Z39.50), OAI-PMH ha optado por la recolección de metadatos. En este caso, existe una entidad que pone a disposición de los interesados información bibliográfica sobre los documentos que almacena. Estos, normalmente agregadores de contenidos, recogen periódica y sistemáticamente todos o parte de los metadatos expuestos para, localmente, implementar servicios de valor añadido.

OAI-PMH sigue el principio de que existen múltiples proveedores de datos (“Data Providers”) que comparten información con múltiples proveedores de servicios (“Service Providers”) a través de un protocolo común. Los primeros son los depósitos de documentos que proporcionan los metadatos de los documentos que almacenan. Por otra parte, el rol de “Service Provider” implica recolectar información académica desde distintos repositorios, con el objetivo de incorporar algún valor añadido y almacenarla en motores de bases de datos o indexadores que admitan una buena performance y un costo de mantenimiento/optimización adecuado. Entre los valores añadidos que se pueden ofrecer se encuentran: sistema de búsqueda e identificación, filtrado, alertas temáticas, medición del uso e impacto de los documentos, etc. El rol de Service Provider es la cara visible al usuario final, y de acuerdo a estudios internacionales hay una relación 1 a 5 entre la cantidad de Service y Data Providers, tal número muestra que representa una innovación en cuanto a los servicios que debe prestar una biblioteca digital y especialmente, una biblioteca digital temática. Para minimizar los problemas derivados de conversiones entre múltiples formatos, OAI-PMH requiere que todos los proveedores de datos expongan sus recursos utilizando mínimamente el esquema Dublin Core sin calificar, descripto en el capítulo anterior. Además de este formato, cada servidor es libre de ofrecer los registros en otro/s formatos adicionales (como por ejemplo el formato MARCXML). Por lo tanto, el protocolo OAI-PMH

puede combinarse con otros protocolos y normas de bibliotecas digitales para facilitar un amplio rango de funcionalidades. De esta forma, este protocolo se convierte en una opción viable y sencilla para que los proveedores de datos puedan poner sus metadatos a disposición de diferentes servicios de información, utilizando para ello estándares abiertos como el HTTP (Hypertext Transport Protocol) y XML (eXtensible Markup Language).

Desventajas

Esta iniciativa, es utilizada por una gran cantidad de sitios como por ejemplo softwares de repositorios digitales y portales de revistas, trae aparejada ciertas desventajas como:

- Recolección de recursos: Cuando se recolectan recursos desde múltiples repositorios, se presentan varios problemas.
- Políticas de catalogación independientes
- Diferencia de formatos de metadatos
- Diferencia de formatos de metadatos (y por lo tanto de especificidad de la información)
- Múltiples términos para el mismo concepto (ej .: idiomas)
- Uso de múltiples vocabularios controlados (tesauros, sistemas de clasificación, etc)
- La gran mayoría expone sus recursos sólo en Dublin Core por lo que sitios que utilizan otra implementación de metadatos no pueden utilizar la iniciativa para interoperar con esta mayoría.

Ejemplo de uso: OAI PMH en SEDICI

El repositorio institucional SEDICI utiliza, entre otros estándares de interoperabilidad, OAI-PMH para comunicarse con "consumidores típicos" como: Sistema Nacional de Repositorios Digitales, SNRD [SNRD 17], Base-Search [base-search 17] y OPAC de ISTEC [ISTEC 17].

También, SEDICI utiliza el estándar OAI-PMH para comunicarse con portales de revista que también utilizan la iniciativa como por ejemplo OJS en su tercera versión.

Simple Web-service Offering Repository Deposit (SWORD)

SWORD [SWORD 17], antes de su aparición en el año 2007, no existía ninguna interfaz estándar para etiquetado, empaquetamiento o herramientas de autoría para cargar objetos en un repositorio, ni tampoco para la transferencia de objetos digitales entre repositorios. No había manera de realizar un depósito desde el exterior de un repositorio, ni de depositar en más de uno a la vez. La ausencia de un estándar de depósito llevó a que JISC [JISC 17] (Joint Information Systems Committee) propusiera una solución denominada SWORD (Simple Web-service Offering Repository Deposit). La misma se trata de un protocolo liviano diseñado para facilitar el depósito interoperable de recursos principalmente en repositorios, pero potencialmente en cualquier sistema en el que se pretenda recibir contenido de fuentes remotas. El acrónimo de las siglas que lo conforman es el siguiente:

- Simple: liviano, ágil y apropiado para sus fines

- Web-service (Servicio Web): independiente del software propietario, soporta interfaces estándar
- Offering (de Oferta): el cliente brinda contenido al servidor.
- Repository (Repositorio): o cualquier otro sistema en el que se quiera depositar o recibir contenido
- Deposit (Depositar): poner, enviar, registrar o añadir, es un paso en el flujo del consumo.

Fue creado para: facilitar la interoperabilidad entre las aplicaciones; simplificar el proceso de identificación, hallar la opción apropiada de contribución y colocación de metadatos mínimos e intentar dotar a las herramientas comunes usadas por el usuario para la creación de materiales digitales, de las capacidades de contribución con los RI.

Soportado por:	Clientes para:	Librerías para:
● Dspace ● Eprints ● Fedora ● arXiv	● Open Journal System ● Moodle ● Microsoft Word ● Bibapp.org ● más	● PHP ● Java ● Python ● Creación de clientes

Tabla 5. Software de repositorios institucionales, clientes y librerías que soportan el protocolo SWORD

Uso de SWORD

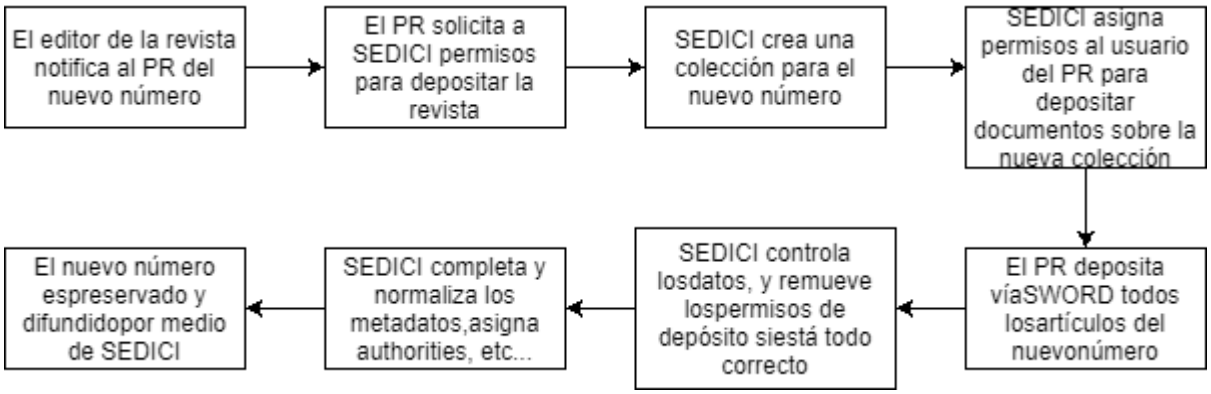
El depósito es un proceso de dos etapas dentro de APP y SWORD. En primer lugar, una solicitud de un usuario autenticado se envía a la implementación para lo que APP llama el 'documento de servicio', esto devuelve detalles de las colecciones que el usuario está autorizado a depositar en el repositorio. En este punto, el usuario puede depositar su archivo en la colección elegida. En esta etapa del proceso algunas cosas pueden llegar a fallar como, por ejemplo falta de credenciales de autenticación, formato de archivo inaceptable o una suma de comprobación MD5 corrupta. El repositorio enviará una respuesta indicando el éxito, o no del depósito.

Ejemplo de uso: Portal de Revistas a SEDICI

Cada nuevo número de las revistas del portal de revistas de la UNLP es enviado hacia el repositorio SEDICI a través de SWORD. Se observa en la figura debajo la cooperación preestablecida entre:

- Editores de revistas (editor)
- Equipo de soporte del Portal de Revistas (PR)

- Administradores de SEDICI (SEDICI)



Para que este flujo de trabajo y el mapeo de metadatos funcione, el plugin SWORD de OJS del portal de revistas de la Universidad Nacional de La Plata fue adaptado [De Giusti 14b]. Con esta adaptación en su momento se logró: depositar 115 documentos, pertenecientes a 9 números de 6 revistas; que el depósito vía SWORD de un número tome menos de un minuto y de un número con 12 artículos tarde 5 minutos.

En el sentido inverso de flujo de información se observa que muchas revistas existen desde hace muchos años y varias de ellas se encuentran en SEDICI, incluyendo sus artículos a texto completo. Algunas de estas revistas han migrado recientemente al portal de revistas y por lo general, quieren exponer todos sus artículos en su nuevo portal. Una solución posible para esto fue generar desde Dspace un archivo .zip que contiene los documentos en forma de AIP (Archival Information Packages). Cada .zip se corresponde con un número de una revista. Se debe, entonces, ejecutar un proceso batch que genere la revista en OJS a partir del AIP generado. Se identifican las etapas del proceso batch en :

- descomprimir el archivo .zip
- crear el nuevo número de la revista
- Para cada AIP
    - identificar los metadatos "de interés" para OJS (Dspace guarda mucha más información que OJS)
    - transformar estos metadatos al formato de OJS
    - insertar los metadatos transformados dentro de la BBDD de OJS
    - identificar el/los archivos correspondientes al artículo
    - copiar los archivos al directorio de uploads del nuevo número
    - asociar los archivos al paper

Desventajas

Como se puede ver, la extensión de SWORD ~~tuvo que ser~~ fue ~~adaptado~~ adaptada para su funcionamiento con SEDICI. Esto quiere decir, que se requiere una personalización para su correcto funcionamiento. También se demuestra que, en el flujo inverso de las revistas científicas, se deben realizar tareas incómodas para su interoperabilidad dado que, por ejemplo: el software OJS no posee un servidor SWORD.

Digital Object Identifier (DOI)

DOI [DOI 17] es una cadena de caracteres utilizada para identificar la propiedad intelectual en el ambiente digital. Constituye un identificador único y permanente de un recurso y un mecanismo para acceder a ese contenido. Se lo considera el Número Internacional Normalizado del Libro (International Standard Book Number, ISBN [ISBN 17]) web porque brinda un sistema similar dentro del ambiente digital. La International DOI Foundation define al DOI como un acrónimo de "identificador de objeto digital", es decir un "identificador digital de un objeto". Proporciona un sistema de identificación permanente e interoperable de información.

El sistema DOI es administrado por la International DOI Foundation (IDF), organización sin fines de lucro, que es la responsable de otorgar la licencia a las agencias de registro DOI, de establecer políticas del sistema y de fomentar su desarrollo futuro. Para obtener los servicios DOI se crearon organizaciones encargadas de esta tarea. En la actualidad son: Airiti, Inc., China National Knowledge Infrastructure (CNKI), CrossRef, DataCite, Entertainment Identifier Registry (EIDR), The Institute of Scientific and Technical Information of China (ISTIC), Japan Link Center (JaLC), Multilingual European DOI Registration Agency (mEDRA), Publications Office of the European Union (OP) y R.R. Bowker (ésta última a partir de septiembre del 2013 pasará a formar parte de CrossRef).

Formato DOI

El sistema DOI está formado por:

1. Una sintaxis de numeración estandarizada que esta compuesta de:
    a. un prefijo, asignado al editor (editorial) por una agencia registradora.
    b. Un sufijo, seguido de un guión.
    Un ejemplo podría ser el de un libro.Éste puede tener un DOI, mientras que un capítulo de ese libro puede tener otro DOI.
2. Servicio de resolución: El servicio de resolución es el proceso en el que un identificador constituye una entrada a un servicio de red para obtener información acerca de un objeto digital. El sistema DOI posee un directorio central. Cuando un usuario hace clic sobre un DOI, un mensaje es enviado al directorio central donde una dirección web es asociada con dicho DOI. Esta ubicación se envía nuevamente al navegador del usuario con un mensaje especial que le dice al sistema que se dirija a "esa dirección particular de Internet". Cuando un objeto es movido a un nuevo servidor o cuando el propietario vende el producto a otra compañía, el cambio de URL es grabado en el directorio y todos los usuarios son direccionados al nuevo

sitio web. De esta manera el cambio de URL de un documento sólo se hace en el directorio y no en todas las referencias hacia él.

3. Modelo de datos: Los metadatos brindan información acerca del objeto y pueden incluir nombres, identificadores, descripciones, clasificaciones, tipos, lugares, tiempos, medidas, relaciones y cualquier otro tipo de información relacionada con el objeto digital.

4. Mecanismo de aplicación: El enlace que provee DOI posee las siguientes características:

    a. Accionable: a través del mecanismo de resolución de números (Handle System [Handle.Net 17]) el DOI enlaza al recurso.

    b. Persistente: mientras que la información acerca de un Palabra Clave (La Plata). ISSN 1853-9912 Volumen 3, número 1, mayo-octubre 2013, p. 12-29 16 objeto puede cambiar con el tiempo, el código DOI es permanente.

    c. Interoperable: a través de un modelo de interoperabilidad semántica el sistema DOI permite que la información originada en un contexto sea utilizada en otros contextos automatizados.

Ejemplo de agencia de registro DOI: Crossref

Crossref [Crossref 17] (antes llamado CrossRef) es una agencia oficial de registro de identificadores de objetos digitales (DOI) de la Fundación Internacional DOI. Está dirigido por Publishers International Linking Association Inc. (PILA) y fue lanzado a principios de 2000 como un esfuerzo cooperativo entre editores para permitir la publicación de referencias cruzadas persistentes entre editores en revistas académicas en línea .

Crossref es una asociación sin fines de lucro, incluye editores con modelos de negocio variados, incluyendo aquellos con políticas de acceso abierto y de suscripción. Crossref no proporciona una base de datos de contenido científico de texto completo. Más bien, facilita los enlaces entre contenido distribuido alojado en otros sitios. Esta agencia enlaza millones de artículos de una variedad de tipos de contenido, incluyendo revistas, libros, actas de conferencias, documentos de trabajo, informes técnicos y conjuntos de datos.

Además de la tecnología DOI que enlaza referencias académicas, Crossref permite un contrato de enlace común entre sus participantes. Los miembros acuerdan asignar DOIs a su contenido actual de la revista y también aceptan vincular desde las referencias de su contenido al contenido de otros editores. Esta reciprocidad es un componente importante de lo que hace que el sistema funcione.

Las organizaciones que no son editores pueden participar en Crossref al convertirse en afiliados. Dichas organizaciones incluyen bibliotecas, hosts de revistas en línea, proveedores de servicios de enlace, proveedores de bases de datos secundarios, motores de búsqueda y proveedores de herramientas de descubrimiento de artículos

Servicios

Además de asignar DOIs al contenido académico, Crossref proporciona servicios adicionales tales como detección del plagio y búsqueda por los financiadores.

Ejemplo de uso DOI: Portal de revistas de la UNLP

En una publicación en el sitio de la Universidad Nacional de La Plata se dio a conocer el uso de esta tecnología. Citando parte del artículo se comentó que: “En un programa sostenido de promoción de los archivos institucionales de acceso abierto, la Universidad Nacional de La Plata adquirió su membrecía Crossref, organización que, entre otros servicios, provee DOIs https://www.crossref.org/about/ y de esta forma, progresivamente, la producción científica editada por la UNLP quedará identificada nacional e internacionalmente con este sistema.”

Handle

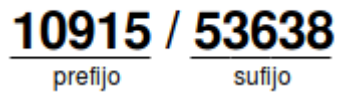
Los handles son identificadores persistentes que surgen para solucionar los problemas que se crean cuando se cambia la ubicación y/o nombre de un objeto digital. El objetivo de un identificador persistente es el de redireccionar a los documentos, aunque estos hayan cambiado su ubicación en la red (cambio de URL).

Uno de los beneficios de utilizar handles es que garantizan la citación correcta de los objetos digitales, puesto que su URN (Nombre Uniforme de Recurso) siempre es el mismo aunque haya sufrido un cambio de ubicación a otro servidor o directorio.

El Sistema Handle, es el sistema que permite la asignación de este tipo de identificadores persistentes a objetos digitales existentes en Internet (artículos, revistas, imágenes, etc.). Desarrollado por CNRI (Corporation for National Research Initiatives), su estructura tiene dos partes:

- Prefijo (Prefix): identifica al productor del identificador (universidad, editorial, revista, etc.)
- Sufijo (Suffix): identifica a cada uno de los documentos u obras digitales (artículo, libro, capítulo, etc.)

La suma del prefijo y el sufijo conforma el identificador persistente, en este caso llamado “handle” como se puede ver en la imagen de aquí debajo



Ejemplo de uso: SEDICI

Este sistema de identificación persistente es el que se usa en el repositorio institucional SEDICI para identificar cada uno de los ítems y documentos que contiene. En el siguiente ejemplo: http://sedici.unlp.edu.ar/handle/10915/53638 el prefijo sería 10915, que corresponde a SEDICI como productor de handles, y el sufijo sería 53638 , que señala el documento específico dentro de SEDICI, en este caso una publicación de un artículo.

Web Scraping

Se puede describir esta técnica [SiteLab 16] como la forma de extraer de manera automática datos de un sitio web donde luego de la extracción se tratan esos datos como información. Representan la capacidad de comunicación entre dos softwares: el que nos brinda la información y el que la recolecta.

Esta tarea se puede representar en etapas, éstas se observan en la siguiente imagen:

En la primera fase del flujo de extracción se examinan los datos que se quieren obtener de un documento HTML-XML. En el siguiente paso se realiza la extracción propiamente dicha de la información seleccionada del documento. Una vez realizada la obtención, se comienza por estructurar los datos al modelo que posee el solicitante de la información. En la fase siguiente los datos son expuestos con la estructura solicitada.

En la última etapa se se realizan las verificaciones para conocer si fue correcta la extracción.

Algunas herramientas para web scraping

Web Scraper (Chrome)

Esta herramienta [Web Scraper 17] es una extensión del navegador web Chrome construida para la extracción de datos de páginas web. Utilizando esta extensión, puede crear un plan (mapa del sitio) sobre cómo debe recorrer un sitio web y qué debe extraerse. Utilizando estos sitemaps el Web Scraper navega el sitio en consecuencia y extrae los datos. Los datos extraídos posteriormente se pueden exportar como CSV.

Esta es una herramienta gratuita que puede ser agregada sin ningún paso previo

Mendeley Web Importer

Esta herramienta [Mendeley 17] realiza la importación de documentos, páginas web y otros documentos directamente a su biblioteca (Mendeley) de referencia de los motores de búsqueda y bases de datos académicos. Mendeley Web Importer está disponible para todos los principales navegadores web: Firefox, Chrome y Safari.

Capítulo 3 | Práctica

Resumen

Como se mencionó en el Capítulo 1, en la actualidad, numerosas plataformas como repositorios institucionales, portales de revistas y editoriales académico-científicas tienen entre sus funciones la difusión de trabajos académicos y científicos como: artículos, congresos, libros, reportes y tesis de grado como de posgrado que se encuentran disponibles en línea.

Una característica común en estas plataformas es que proporcionan información en línea en la que se describen los objetos que alojan (título, autor, resumen, palabras clave, áreas temáticas, idiomas, instituciones, etc.). Estos metadatos pueden seguir internamente un esquema de catalogación existente (Dublin Core [DCMI 17], MODS [Metadata MODS 17], etc.) o utilizar una representación interna propia del software de gestión en uso. Algunas de estas plataformas se sirven del acceso a metadatos de los objetos digitales bajo algún servicio web (ej. OAI PMH [OAI-PMH 17]), o exponen toda o parte de la información de manera estructurada (por ejemplo, campos META dentro del HTML [HTML-CSS 17] como ser DC.title, DC.date o DC.type). En estos casos, la extracción y mapeo de metadatos resulta, por lo general, más simple de realizar. Sin embargo, el reto es realizar la extracción de metadatos a partir de los documentos HTML utilizados para su exposición, en particular cuando el código HTML no sigue un esquema estandarizado o no utiliza las etiquetas descriptivas correctas. Si esta información pudiera ser extraída y mapeada hacia esquemas de metadatos normalizados, podría integrarse como *input* hacia otros sistemas externos que pudieran requerirla para completar su tarea. De aquí surge entonces la idea de desarrollar una herramienta que realice esta tarea de manera que se pueda mitigar el problema de la falta de estandarización de los metadatos y, especialmente, aprovechar la vasta cantidad de recursos académicos y científicos en línea disponibles en la red que no pueden ser "consumidos" por sistemas tradicionales de procesamiento de metadatos.

Desarrollo de la herramienta

Introducción

En este capítulo se detallan los lenguajes, tecnologías y patrones utilizados para la creación de la herramienta. Luego se detallan las distintas etapas de desarrollo del trabajo donde se exponen: los formatos de documentos que poseen los sitios de prueba donde se realizaron las extracciones, las prueba realizadas en ellos, y los resultados que se obtuvieron.

Finalmente se expone el desarrollo de la extensión para el navegador web Chrome que realiza la carga automatizada de los campos del formulario del repositorio SEDICI.

Lenguaje y Framework

Ruby

Ruby [Ruby 17] es un lenguaje multiplataforma, interpretado y orientado a objetos y su implementación oficial es distribuida bajo licencia de software libre. Combina una sintaxis inspirada en Python y Perl con características similares a Smalltalk. Comparte también funcionalidades con otros lenguajes de programación como Lisp, Lua, Dylan y CLU.

Ruby on rails

Se presenta como un entorno de desarrollo web de código abierto para el lenguaje de programación Ruby. Este framework, además, sigue el paradigma del patrón Modelo Vista Controlador (MVC) que será comentado en la siguiente sección. Posee herramientas que permiten escribir un buen código evitando que te repitas y favoreciendo la convención antes que la configuración.

El lenguaje de programación Ruby permite la metaprogramación. Esto, consiste en escribir programas que escriben o manipulan otros programas (o a sí mismos) como por ejemplo

datos, o que hacen en tiempo de compilación parte del trabajo que, de otra forma, se haría en tiempo de ejecución. Esta característica es utilizada por Rails, lo que resulta en una sintaxis que muchos de sus usuarios encuentran muy legible.

Tecnología utilizada: Gemas

Una gema es un paquete que tiene como lenguaje de programación Ruby. Para obtener una gema se debe utilizar RubyGems. Este, es un gestor de paquetes que proporciona un formato estándar y autocontenido (llamado gem o gemas en español) que sirve para poder distribuir programas o bibliotecas en Ruby.

Nokogiri

Es un analizador de HTML, XML, SAX y Reader. Entre las muchas características de Nokogiri se encuentra la capacidad de buscar documentos a través de selectores XPath [XPath 17] o CSS3 [CSS 17]. Esta herramienta puede ser también utilizada como una librería en muchos sistemas operativos como Windows Ubuntu/Debian y MacOS entre otros.

Ejemplos de uso

Análisis gramatical de documentos HTML / XML :

```
html_doc = Nokogiri::HTML("<html><body><h1>Mr. Belvedere Fan
Club</h1></body></html>")
xml_doc  =
Nokogiri::XML("<root><aliens><alien><name>Alf</name></alien></alie
ns></root>")
```

Ejemplo 1. Uso de Nokogiri en documentos XML y HTML

En el ejemplo 1 se encuentran dos variables llamadas `html_doc` y `xml_doc`, estas son documentos Nokogiri que se crean al momento de analizar un HTML o XML. Los documentos Nokogiri poseen muchas propiedades que pueden utilizarse para realizar análisis gramatical dentro del objeto, como por ejemplo:

- NONET - Evita las conexiones de red durante el análisis. Recomendado para analizar documentos no confiables.
- RECOVER - Intenta recuperarse de los errores. Recomendado para analizar documentos malformados o no válidos.
- NOBLANKS - Eliminar nodos en blanco
- NOENT - Sustituir entidades
- NOERROR - Suprimir informes de error
- STRICT - Análisis estricto; generar un error al analizar documentos malformados

- DTDLOAD y DTDVALID - Permite validar si se quiere el DTD
- HUGE - Se utiliza para omitir los límites codificados en cuanto al tamaño del documento o la profundidad del DOM.

Otra de las características importantes de Nokogiri es que provee la búsqueda dentro de documentos XML y HTML.

```
[shows.xml]
<root>
  <sitcoms>
    <sitcom>
      <name>Stranger things</name>
      <characters>
        <character>Winona Ryder</character>
        <character>David Harbour </character>
        <character>Finn Wolfhard</character>
      </characters>
    </sitcom>
    <sitcom>
      <name>Outlander</name>
      <characters>
        <character>Caitriona Balfe</character>
        <character>Sam Heughan</character>
      </characters>
    </sitcom>
  </sitcoms>
  <dramas>
```

```
    <drama>
      <name>House of cards</name>
      <characters>
        <character>Kevin Spacey</character>
        <character>Robin Wright </character>
      </characters>
    </drama>
  </dramas>
</root>
```

Ejemplo 2. Documento XML para ejemplìficar la búsqueda con Nokogiri

Tenemos en el ejemplo 2 un documento XML que tiene los actores de distintas series de televisión. Se quiere obtener del documento XML el listado de los actores:

```
@doc = Nokogiri::XML(File.open("shows.xml"))
@doc.xpath("//character")
# => ["<character>Winona Ryder</character>",
#     "<character>David Harbour </character>",
#     "<character>Finn Wolfhard</character>",
#     "<character>Caitriona Balfe</character>",
#     "<character>Sam Heughan</character>",
#     "<character>Kevin Spacey</character>",
#     "<character>Robin Wright </character>",]
```

Ejemplo 2a. Método de búsqueda dentro de un documento XML utilizando Nokogiri

En el ejemplo 2a se convierte el documento XML en un documento Nokogiri. Una vez convertido, se aplica el método xpath el cual devuelve un NodeSet. Esto hace que la variable doc pueda actuar como si este fuese una matriz y pueda solicitarle el resultado de todos los nodos coincidentes a “//character” en el documento.

Open-uri

Esta gema [OpenURL 17] nos permite ingresar a una URL enviada a través de HTTP, HTTPS o FTP como si se tratase de un archivo.

```
open("http://www.ruby-lang.org/")                                    {|f|
```

```
  f.each_line {|line| p line}

  p f.base_uri         # <URI::HTTP:0x40e6ef2 URL:http://www.ruby-lang.org/en/>

  p f.content_type     # "text/html"

  p f.charset          # "iso-8859-1"

  p f.content_encoding # []

  p f.last_modified    # Thu Dec 05 02:45:02 UTC 2002


}
```

Ejemplo 3. Uso de open-uri

En el ejemplo 3 se usa open-uri para convertir una URL en archivo. Este archivo abierto tiene varios métodos para obtener su metainformación, ya que el método es extendido por OpenURI :: Meta. En el archivo se encuentran por ejemplo: base_uri, content_type, charset, content_encoding, last_modified.

Ejemplo de uso de las gemas en conjunto:

En el ejemplo 3 se muestra la sintaxis del lenguaje Ruby y el uso de las gemas open uri y nokogiri.

```ruby
#! /usr/bin/env ruby

require 'nokogiri'
require 'open-uri'

# Obtener y analizar el documento HTML
doc =
Nokogiri::HTML(open('http://www.nokogiri.org/tutorials/installing_nokogi
ri.html'))

puts "### Buscar nodos a través de css"
doc.css('nav ul.menu li a', 'article h2').each do |link|
  puts link.content
end
```

Ejemplo 3. Uso del lenguaje Ruby y framework Rails que utiliza las gemas Open-Uri y Nokogiri.

El código realiza la transformación de una URL que aloja un documento HTML a un documento Nokogiri para poder ser analizado. Para esto, se utiliza la gema open-uri que convierte la URL donde está la información a analizar en un archivo.

Una vez que se tiene el documento Nokogiri se utiliza el método css para realizar la búsqueda de las etiquetas `'nav ul.menu a'` y `"article h2"`.

PostgreSQL

PostgreSQL [Postgres 17] es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código abierto. Utiliza un modelo cliente/servidor y usa *multiprocesos* en vez de *multihilos* para garantizar la estabilidad del sistema. Esto significa que, cuando un proceso falla no afecta el resto de los procesos, por lo que el sistema continuará funcionando.

La gema llamada pg, es la interfaz de Ruby para PostgreSQL. Esta gema funciona con PostgreSQL 8.4 y versiones posteriores.

Puma

Puma [Puma 17] fue creado por Evan Phoenix a fines de 2011 como derivado de Mongrel (servidor web para Ruby)aunque la mayor parte del código Mongrel original se ha reescrito excepto por el analizador. Puma es un servidor simple, rápido y altamente concurrente HTTP 1.1 para aplicaciones Ruby. Está diseñado para su uso en entornos de desarrollo y producción.

Cómo trabaja

Puma procesa las solicitudes utilizando una extensión Ragel, compilador de máquina de estado finito y generador de análisis, optimizada en C que proporciona un análisis rápido y preciso del protocolo HTTP 1.1.Luego sirve la solicitud en un hilo de un grupo de hilos interno. Como cada solicitud se sirve en un hilo por separado, las implementaciones de Ruby realmente simultáneas usarán todos los núcleos de CPU disponibles.

Para instalar y comenzar a utilizar puma sólo se debe instalar y especificar la aplicación que va a usarlo:

```
$ gem install puma
$ puma <any rackup (*.ru) file>
```

Puma vs. otros servidores

Puma en su sitio web compara el uso de memoria [Puma 17a] que utilizan algunos servidores web:

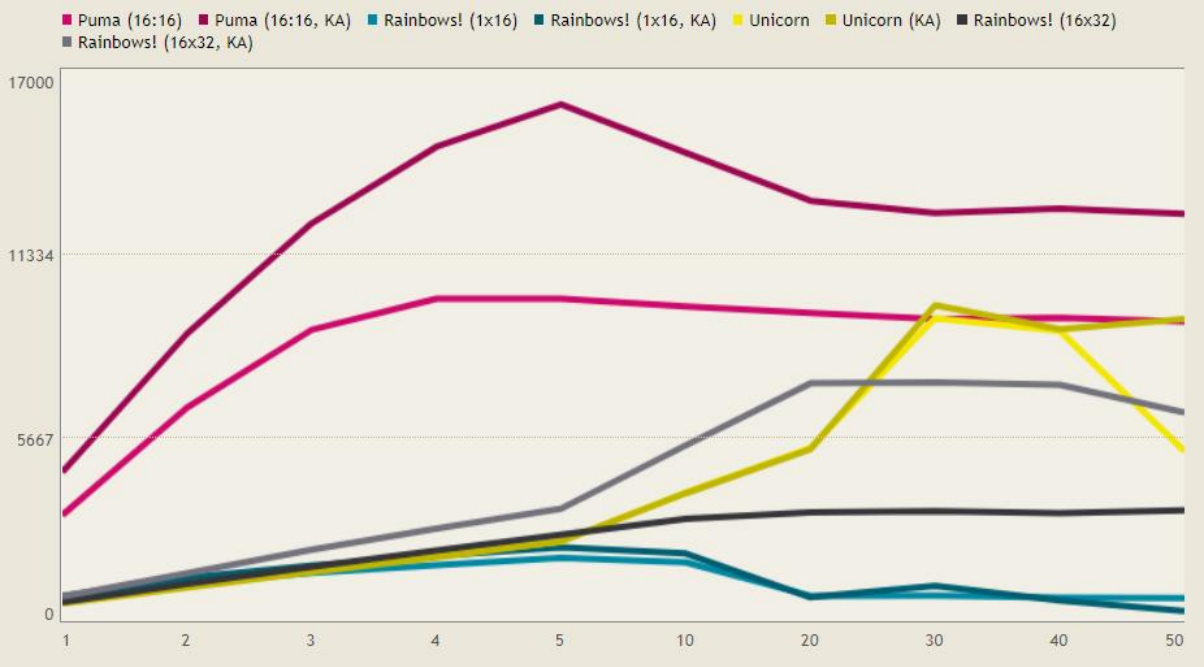
PUMA Puma- 78 Mb

RAINBOWS! Rainbows!(1X16) - 120 Mb

UNICORN Unicorn - 1076 Mb

RAINBOWS! Rainbows!(16X32) - 1138 Mb

El gráfico muestra el uso de memoria a lo largo del tiempo y la comparación de velocidad



Javascript

JavaScript [JavaScript 17] (abreviado comúnmente JS) es un lenguaje de programación interpretado, orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Principalmente se utiliza lado del cliente (client-side), aunque también, existe una forma de JavaScript del lado del servidor(Server-side JavaScript o SSJS). Implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

JavaScript se diseñó con una sintaxis similar a C, aunque adopta nombres y convenciones del lenguaje de programación Java, aunque Java y JavaScript tienen semánticas y propósitos diferentes.

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías como AJAX. JavaScript se interpreta en el agente de usuario al mismo tiempo que las sentencias van descargando junto con el código HTML.

Tipos de documentos utilizados

Hyper Text Markup Language (HTML)

Este es un lenguaje cuyo objetivo es el desarrollo de las páginas web, indicando cuales son los elementos que la compondrán, orientando hacia cuál será su estructura y también su contenido. El código HTML se organiza en dos grandes secciones, encabezado o HEAD, y cuerpo o BODY. Dentro del encabezado, se incluye todo el código que no hace a la estructura del documento, pero que permite que se visualice correctamente: título, hojas de estilo, código javascript y, de particular interés en este trabajo, campos meta o meta tags. Los meta tags se usan normalmente para resumir el contenido de la página para buscadores y navegadores web. Es decir, describen la página para que pueda ser entendida por diferentes servicios web. Otro de sus usos es el de especificar información que los navegadores web utilizan para mostrar la página. La información puede ser por ejemplo como: el grupo de caracteres usado, tiempo de expiración del contenido, posibilidad de dejar la página en cache o calificar el contenido del sitio y también utilizando meta tags de Dublin Core (Metadato explicado anteriormente).

Ejemplo

```
<head>
  <meta charset="UTF-8">
  <meta name="description" content="Tesina de grado">
  <meta name="keywords" content="HTML,CSS,XML,JavaScript">
  <meta name="author" content="Julieta Rodriguez Vuan">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
```

Ejemplo 2. Uso de tag meta en documento HTML

XML

XML (Extensible Markup Language) es un lenguaje de marcado de uso general. Creado en 1998 por W3C, es un estándar internacional libre y gratuito. Es un lenguaje de programación que utiliza marcas o etiquetas para definir la estructura, presentación y/o formato de los textos. La palabra marcado sigue la idea de que el lenguaje permite añadir etiquetas al contenido original del texto, y éstas permiten que los programas informáticos puedan procesar o interpretar adecuadamente los textos.

El objetivo fundamental de XML es intercambiar datos estructurados entre sistemas de información. Se trata de un formato de texto plano, lo que facilita la transferencia de información.

Para que los documentos XML sean procesables deben estar bien formados lo que implica que deben cumplir estrictas normas sintácticas. El modelo de datos de los documentos XML es jerárquico y está formado por dos estructuras principales: elementos y atributos.

La importancia del lenguaje XML está dada por ser la base de numerosos procesos y técnicas. Por ejemplo, XML se utiliza para marcar documentos de carácter variado como por ejemplo en bibliotecas digitales, artículos, transferencia de información. Asimismo, su uso en las bases de datos se ha incrementado notoriamente, no sólo como soporte para la transferencia de datos sino como formato de almacenamiento como es el caso de los repositorios institucionales.

Aquí debajo se detalla un ejemplo de una estructura simple de un documento XML

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

Ejemplo 3. Estructura de documento HTML

Notación de Objetos de JavaScript (JSON)

JSON (JavaScript Object Notation) es un formato de intercambio de datos. Tiene como base un subconjunto del Lenguaje de Programación JavaScript, aunque puede utilizarse de manera independiente del lenguaje con el que se ha desarrollado, aun que utiliza convenciones que son ampliamente conocidas como por ejemplo: C, C++, C#, Java, JavaScript, Perl, Python,etc.

JSON está constituído por dos partes:

1. Una colección de pares nombre-valor. Esta primer parte es implementada por los distintos lenguajes como: un registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.

2. Una lista ordenada de valores. Esta segunda estructura puede ser implementada como: arreglo, vector, lista o secuencia.

```
{
  "Equipo": "SEDICI",
  "ciudad": "La Plata",
  "fecha_creacion": 2016,
  "Base": "Ex Liceo",
  "activado": true,
  "miembros": [
    {
```

```
      "nombre": "Julieta Rodriguez Vuan",
      "edad": 27,
      "puesto": "Tesista",
      "trabajo": [
        "Desarrollo",
        "Autor",
      ]
    },
    {
      "nombre": "Gonzalo Villarreal",
      "edad": 35,
      "puesto": "Asesor Profesional",
      "trabajo": [
        "Asesor",
      ]
    },
  ]
}
```

Ejemplo 4. Estructura en formato JSON

Diseño

Patrones utilizados

Según Christopher Alexander, uno de los autores del libro “Patrones de Diseño”, describe los patrones como: “cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, así como la solución a ese problema, de tal modo que se pueda aplicar esta solución un millón de veces, sin hacer lo mismo dos veces”. Un patrón, abstrae e identifica los aspectos claves de una estructura de diseño común, lo que los hace útiles para crear un diseño orientado a objetos reutilizable. El patrón de diseño identifica las clases e instancias participantes, sus roles y colaboraciones, y la distribución de responsabilidades. Cada patrón de diseño se centra en un problema concreto, describiendo cuando aplicarlo y si tiene sentido hacerlo teniendo en cuenta otras restricciones de diseño, así como las consecuencias y ventajas e inconvenientes de uso.

Para el diseño de la herramienta se tomaron dos patrones que mejoran la calidad y permite que sea escalable y pueda soportar los cambios en el futuro.

Cadena de responsabilidad

Este patrón se utiliza para establecer una cadena de objetos receptores por donde se pasa la petición realizada por un objeto emisor. La idea de este patrón es que cualquiera de los objetos receptores puede responder a la petición en función de un criterio establecido. Además ayuda a que el usuario no necesite conocer la forma en que se procesa la petición.

El modelo UML aquí debajo representa el patrón utilizado:

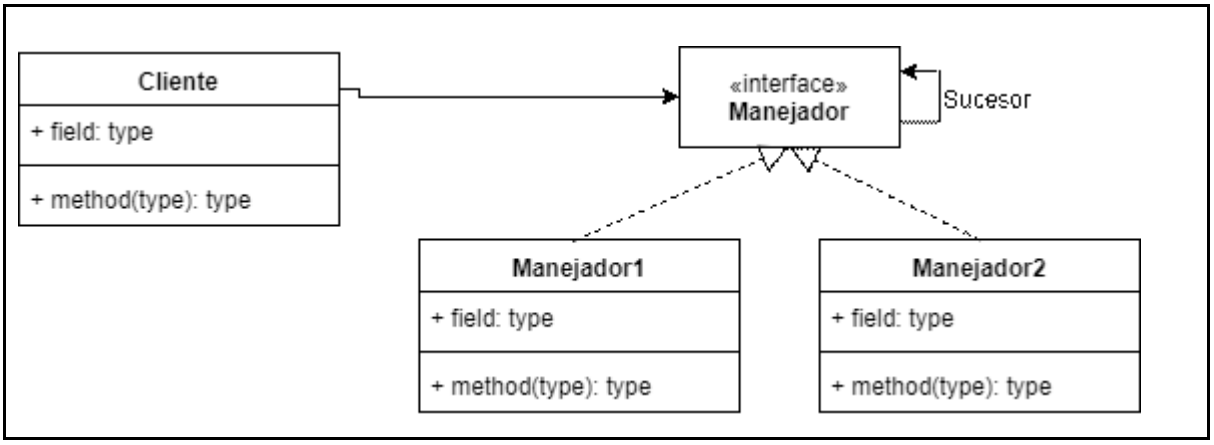


Figura 2. Patrón de diseño Cadena de Responsabilidad.

Donde:

- Manejador: Interfaz que define las operaciones necesarias para tratar los mensajes y propagarlos si corresponde.
- Manejador 1/Manejador 2: Implementan la interfaz Manejador. Son los encargados de procesar un tipo de mensaje concreto o propagar el mensaje a otro miembro de la cadena en caso de que el mensaje no sea de dicho tipo.
- Cliente: Trata de enviar un mensaje a un destino enviándolo mediante un Manejador conocido.

Este patrón se utilizó en la implementación de la herramienta para decidir que tipo de *parseo* se debía realizar en los metadatos del artículo seleccionado.

Estrategia

La idea de este patrón es la de encapsular algoritmos en objetos, permitiendo que éstos puedan ser re utilizados e intercambiables. En base a un parámetro, que puede ser cualquier objeto, permite a una aplicación decidir el algoritmo que debe ejecutar.

La base de este patrón es la de proporcionar múltiples variantes de un algoritmo o comportamiento que pueden ser encapsulados en clases separadas. Esto, permite cambiar o agregar algoritmos, independientemente de la clase que lo utiliza.

El modelo UML aquí debajo representa el patrón utilizado:

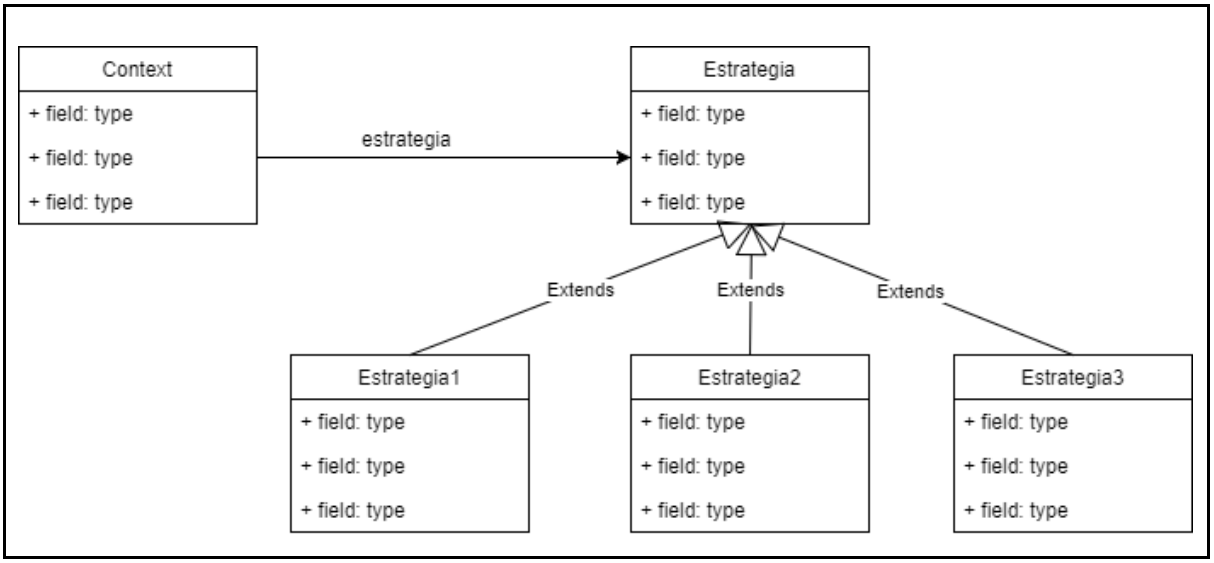


Figura 3. Patrón de diseño Estrategia.

- Estrategia: declara una interfaz común a todos los algoritmos soportados.
- Estrategia 1-2-3: implementan un algoritmo utilizando la interfaz Estrategia. Es la representación de un algoritmo.
- Context: mantiene una referencia a Estrategia y según las características del contexto, optará por una estrategia determinada. Solicita un servicio a Estrategia y este debe devolver el resultado de una Estrategia concreta (1,2,3..).

El patrón se utilizó en la implementación de la herramienta para poder seleccionar el tipo de schema con el que se quiere realizar la extracción de metadatos.

Arquitectura de la aplicación

En esta sección se explica la arquitectura y patrón utilizados en el desarrollo de la herramienta. En el caso de la arquitectura se decidió utilizar el tipo cliente-servidor y el diseño de la arquitectura fue utilizado el patrón MVC.

Arquitectura cliente-servidor

La arquitectura cliente-servidor [Valle 05] consiste en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta. Hoy en día las aplicaciones con esta arquitectura son el soporte de la mayor parte de la comunicación por redes.

En esta arquitectura la computadora que utilizan los usuarios (cliente), produce una petición de información a la o las computadoras que proporcionan información (servidores). Éstas, responden a la demanda del cliente que la produjo.

Los clientes y los servidores pueden estar conectados a una red local o una red global, como la que se implementa en una empresa o lo que es la Internet.

Partes que componen el sistema

Cliente: Este, como comentamos anteriormente, es el que envía una petición al servidor y se queda esperando por una respuesta. Su tiempo de vida es finito una vez que son servidas sus solicitudes, termina el trabajo.

Servidor: Éste o éstos, son los que ofrecen un servicio que se puede obtener en una red tanto local como global. Su trabajo es, si esta libre, aceptar la petición enviada por el cliente desde la red, realizar el servicio solicitado y devolver el resultado al solicitante. Su tiempo de vida o de interacción es “infinito” puesto que esta siempre a la espera de nuevas peticiones.

Patrón MVC

Modelo Vista Controlador [MVC 17] es un patrón de diseño que mejora la arquitectura de software. Éste, separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

Para este tipo de arquitectura, el patrón MVC propone el desarrollo de tres componentes esenciales que son el modelo, la vista y el controlador. Es decir, por un lado define componentes para la representación de los datos, y por otro lado para la interacción del usuario.

Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

Componentes

- Modelo:Este componente se puede decir que es el central del patrón. Expresa el comportamiento de la aplicación en términos del dominio del problema, independientemente de la interfaz de usuario. Gestiona directamente los datos, la lógica y las reglas de la aplicación.
- Vista: Este componente puede ser cualquier representación de salida de información, tal como un gráfico o un diagrama. Son posibles múltiples vistas de la misma información, como un gráfico de barras para la gestión y una vista tabular para los contadores.
- Controlador: Este tercer componente acepta la entrada, como una petición y la convierte en comandos para el modelo o la vista.

Interacciones

Además de dividir la aplicación en tres tipos de componentes, el patrón de diseño MVC define las interacciones entre los componentes anteriormente detallados.

Una de las interacción es la realiza el modelo, donde almacena los datos que se recuperan de acuerdo con los comandos del controlador y se muestran en la vista.

Siguiendo esta línea, la vista genera una nueva salida para el usuario basada en cambios en el modelo.

Finalmente, el controlador puede enviar comandos al modelo para actualizar el estado del modelo (por ejemplo, editar un documento). También puede enviar comandos a su vista asociada para cambiar la presentación de la vista del modelo (por ejemplo, desplazarse a través de un documento, movimiento de documento).

A continuación se expone una explicación gráfica de los componentes y las interacciones entre ellos:

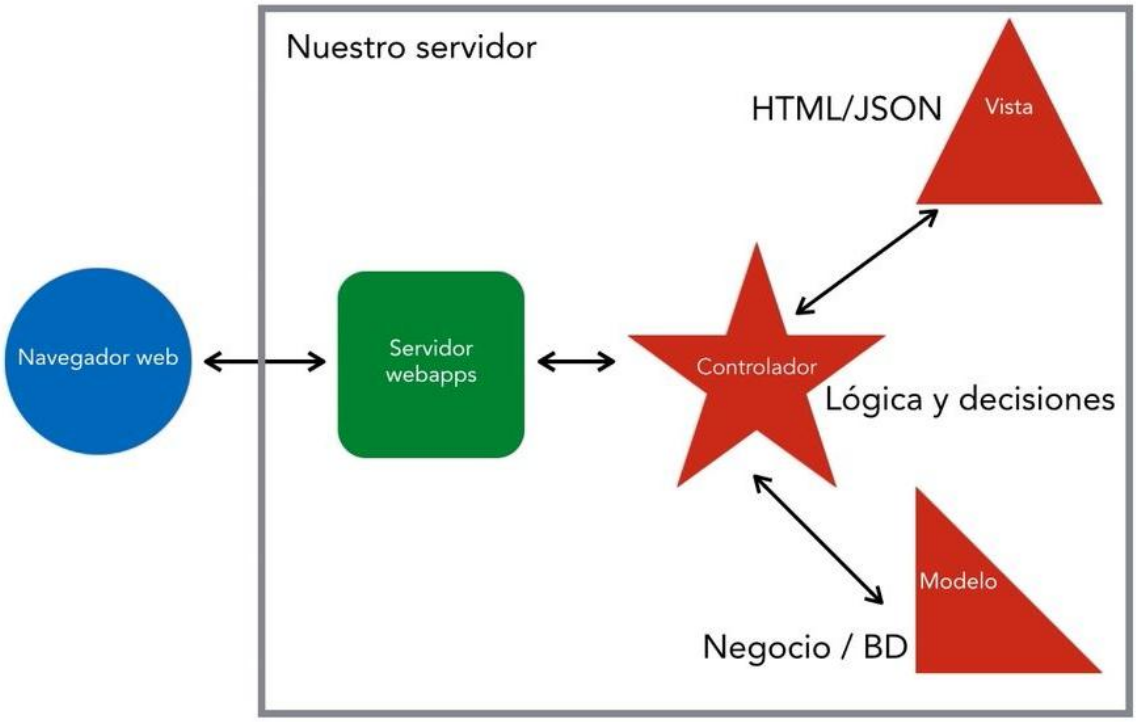


Figura 4. Patrón de diseño Modelo-Vista-Controlador.

Etapas de modelado

Para el desarrollo del lenguaje, se propusieron distintos prototipos hasta llegar al modelo final. Cada una fue puesta a prueba encontrando sus deficiencias y mejoras a realizar. Para dar una visión de la transición se detalla el modelo de la aplicación y un diagrama de secuencia para exponer su funcionamiento.

Prototipo y modelo final

Versión 1

Modelo: En este primer prototipo se tomó la idea del patrón de cadena de responsabilidades como la base de la herramienta. En este caso los métodos de extracción están separados por tipo de *parseo.*

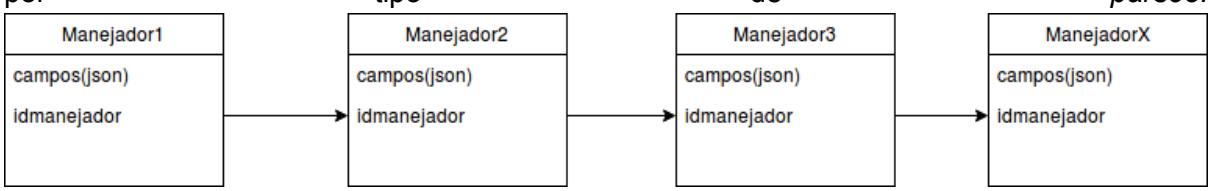


Figura 5. Modelo de la herramienta en su primera versión.

Diagrama de secuencia: En esta primera versión se describe el envío de una petición por parte de un usuario para realizar la extracción de los metadatos de un artículo científico enviado a través de un formulario alojado en la interfaz. Como se observa en el diagrama, existe un objeto manejador por cada tipo de manejo de metadato. La petición se va pasando por los manejadores hasta que se encuentre el indicado para análisis.

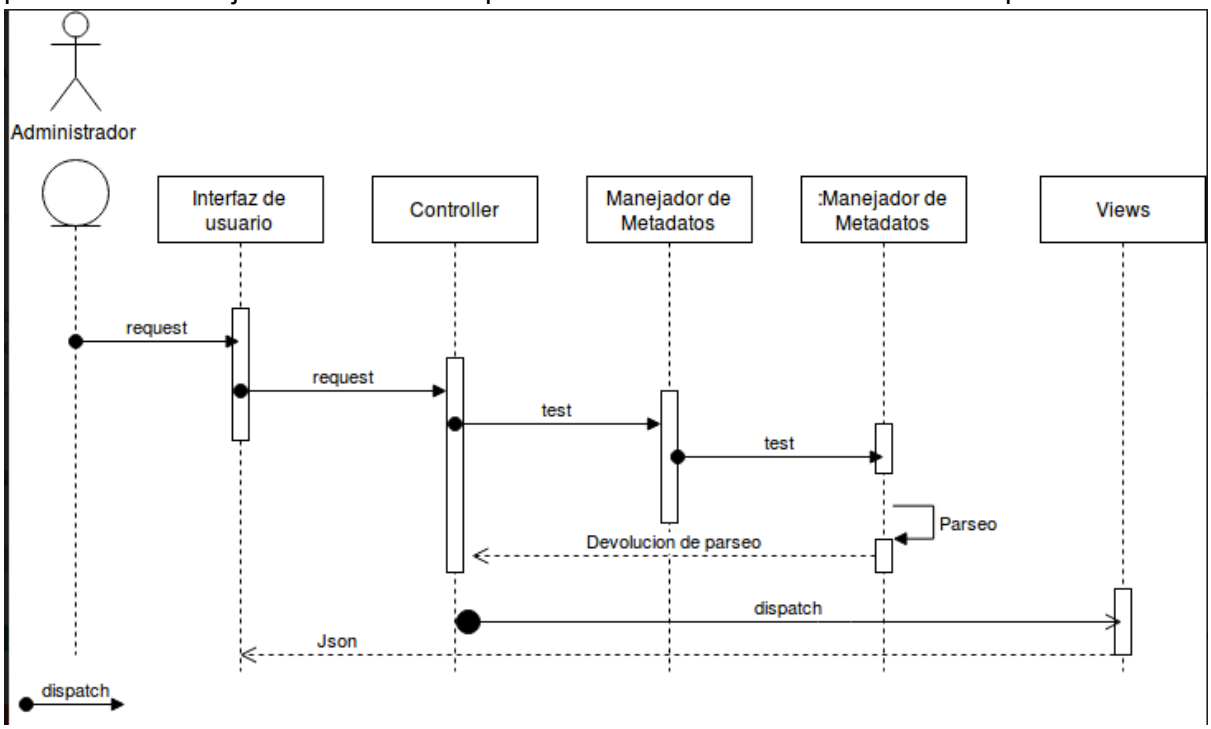


Diagrama 1. Diagrama de secuencia que se obtuvo en la primera versión de la herramienta.

Desventajas: En esta versión se tenía la información separada por cada una de los objetos manejador, esto hacia que la escalabilidad sea compleja ya que:

- si se quisiese agregar una nueva forma de manejar metadatos hay que agregar un objeto nuevo y su formato de parseo
- Solo una persona con conocimiento del lenguaje podría agregar un nuevo manejador puesto que es nuevo código en la herramienta
- También existe mucha repetición de código puesto que es probable que distintos tipos de manejador utilicen la misma forma de *parseo*.

Versión 2

Modelo: En este segundo prototipo se agrega el patrón estrategia y se modifica el uso de la cadena de responsabilidades. Como se puede observar en el modelo se observan:

Site: Este es un objeto que se crea por cada sitio que se quiere agregar a la herramienta. En este contexto, cada sitio se corresponde con cada una de las plataformas web a partir de

las que se extraerán metadatos: Elsevier, Springer Nature, Portal de Revistas de la UNLP, etc. Este objeto tiene una referencia al objeto Parser.

- Parser: Este es un objeto abstracto que tiene el método parse que será implementado por sus hijo.
- META/HTML: Estos objetos, hijos de Parser, implementan el método parse.

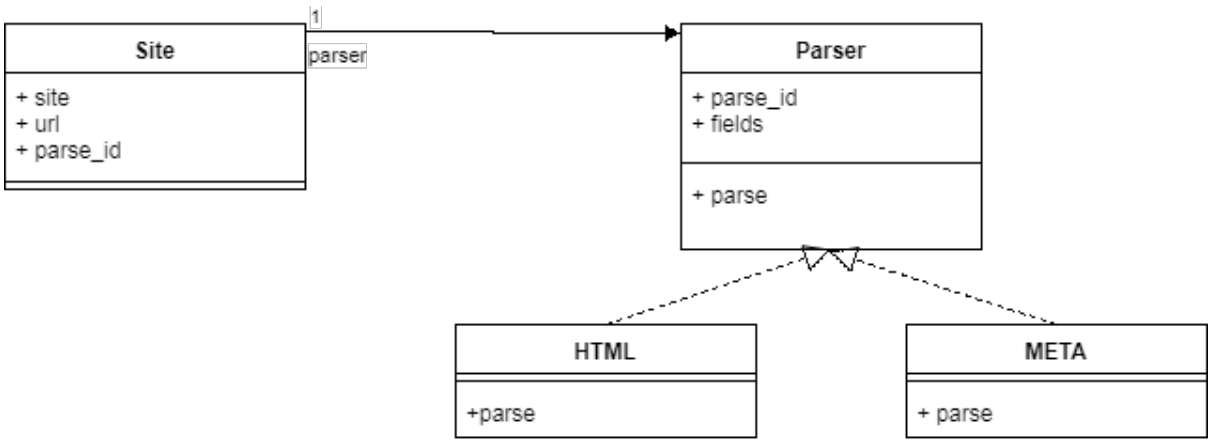


Figura 6. Modelo de la herramienta en su segunda versión.

Diagrama de secuencia: En esta segunda versión se describe el envío de una petición por parte de un usuario para realizar la extracción de los metadatos de un artículo científico enviado a través de un formulario alojado en la interfaz. Como se observa en el diagrama, se cuenta con un controlador, ScrapeController, que realizará una búsqueda entre todos los sitios que tiene alojados y enviará la petición de *parseo* al sitio si es que lo encuentra. Este Site utilizará el objeto Parser que tiene asociado y le pedirá que realice la extracción de información y de ahí se devuelve el resultado.

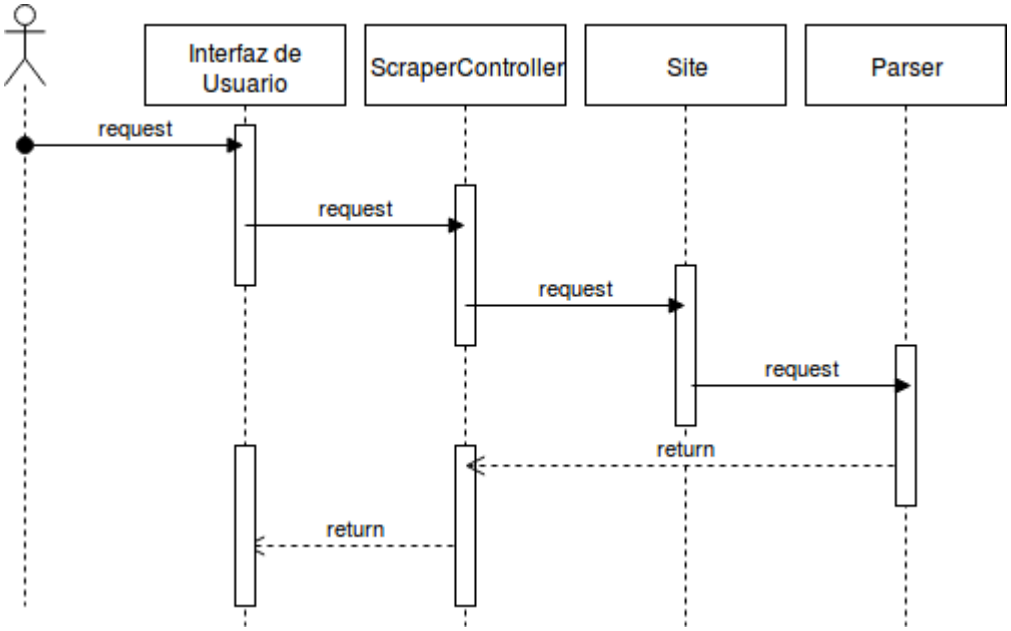


Diagrama 2. Diagrama de secuencia que se obtuvo en la segunda versión de la herramienta.

Desventajas: En este prototipo los problemas se basan en que:

- Se debe tener alojado en la base de datos el sitio que se quiere *scrapear*. Esto produce que se tenga repetición de código puesto que que es si alguno de ellos comparte el mismo software van a tener el mismo tipo de *parseo*.
- Se tienen separadas las formas de extracción de metadatos por lo que puede haber muchos datos que no sean extraídos.

Versión 3

Modelo: En este tercer prototipo se elimina el objeto sitio y se declaran únicamente dos objetos. En este caso se incluyen:

Fields: Esto es un objeto que tiene el listado de todos los campos que posee un artículo, entre ellos: título, autor, resumen, palabras claves, etc. El objeto Fields posee además una colección de objetos Methods.

- Methods: Este es un objeto que posee las formas con las que se puede *parsear* un campo, como por ejemplo: meta name=”dc.title”.

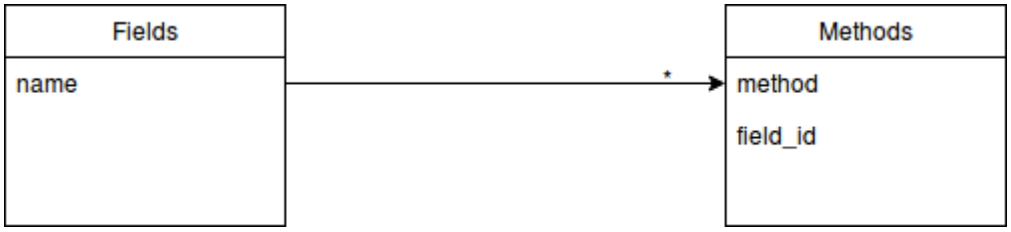


Figura 7. Modelo de la herramienta en su tercera versión.

Diagrama de Secuencia: En esta tercer versión se describe el envío de una petición por parte de un usuario para realizar la extracción de los metadatos de un artículo científico enviado a través de un formulario alojado en la interfaz. Como se observa en el diagrama se observa un controlador, ScrapeController, que recorre todos los campos que están en Field. Este Field recorre su colección de ScrapeMethods para *parsear* el documento con los métodos cargados. Si encuentra un método que logre *parsear* el campo devuelve el valor al controller. Esta secuencia se repite con todos los campos que están alojados en la base de datos y luego se retorna el resultado.

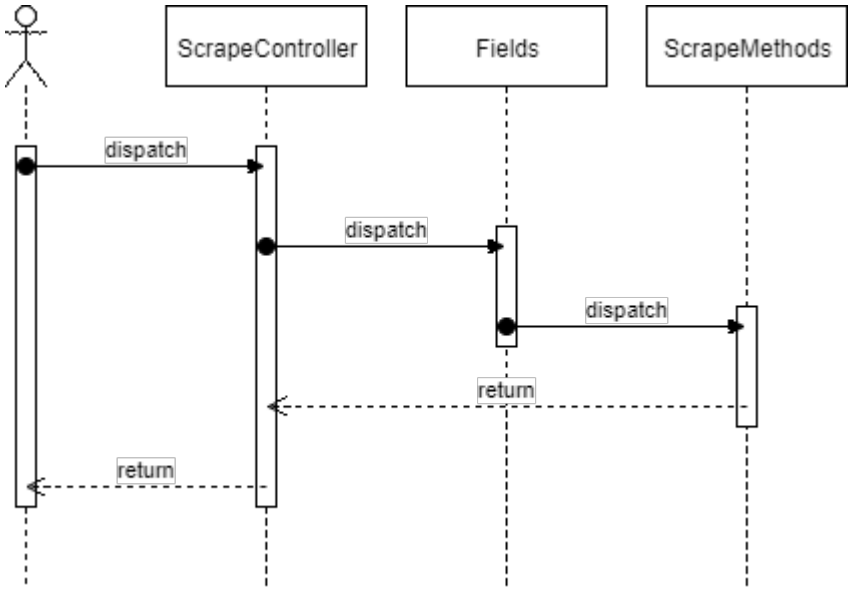


Diagrama 3. Diagrama de secuencia que se obtuvo en la tercera versión de la herramienta.

Desventajas: En esta versión se encontró que:

- Se pierde la “prioridad” de búsqueda de método de parseo ya que al tener todos los métodos de extracción en colecciones puede suceder que se esté extrayendo con un método que no obtenga la información correcta.
- No se puede determinar que tipo de schema tiene la página a *scrapear*.
- Repetición de información.

Versión 4 (versión final)

Modelo: En este cuarto y último prototipo se unen los dos patrones comentados anteriormente y se agrega el objeto Schema. Las clases son:

- Schema: Este objeto posee un identificador que identifica el tipo de esquema de metadatos (metadata schema) que el artículo posee. Una colección de objetos Fields, en este caso solo se tienen los Fields que el artículo realmente posee.
- Field: Esto es un objeto que tiene el listado de todos los campos que posee un artículo, entre ellos: título, autor, resumen, palabras claves, etc. El objeto Fields posee además una colección de objetos Methods.
- Method: Este es un objeto que posee las formas con las que se puede *parsear* un campo, como por ejemplo: meta name=”dc.title”.

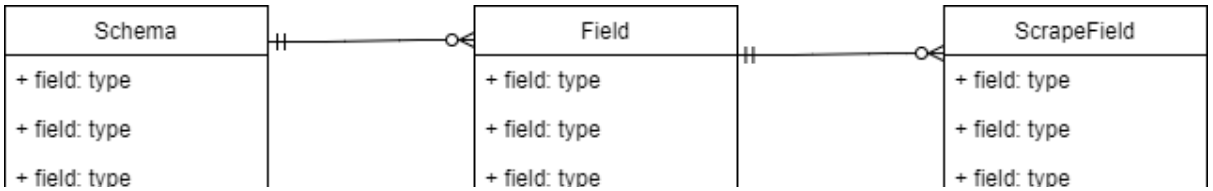


Figura 8. Modelo de la herramienta en su versión final.

Diagrama de Secuencia: En esta cuarta y última versión se describe el envío de una petición por parte de un usuario para realizar la extracción de los metadatos de un artículo científico enviado a través de un formulario alojado en la interfaz. Como se observa en el diagrama, se tiene un controlador, ScrapeController, que a través de un identificador que se extrae del documento a *scrapear* se reconoce el tipo de Schema que utiliza el artículo. Una vez obtenido el Schema, se recorren todos los objetos Field que posee. Este Field, a su vez, selecciona su ScrapeMethods y solicita el *parseo* en el documento. Esta secuencia se repite con todos los campos que posee el Schema y luego se retorna el resultado. Si no se tiene un schema, la herramienta utiliza una forma de extracción genérica de datos con tags CSS.

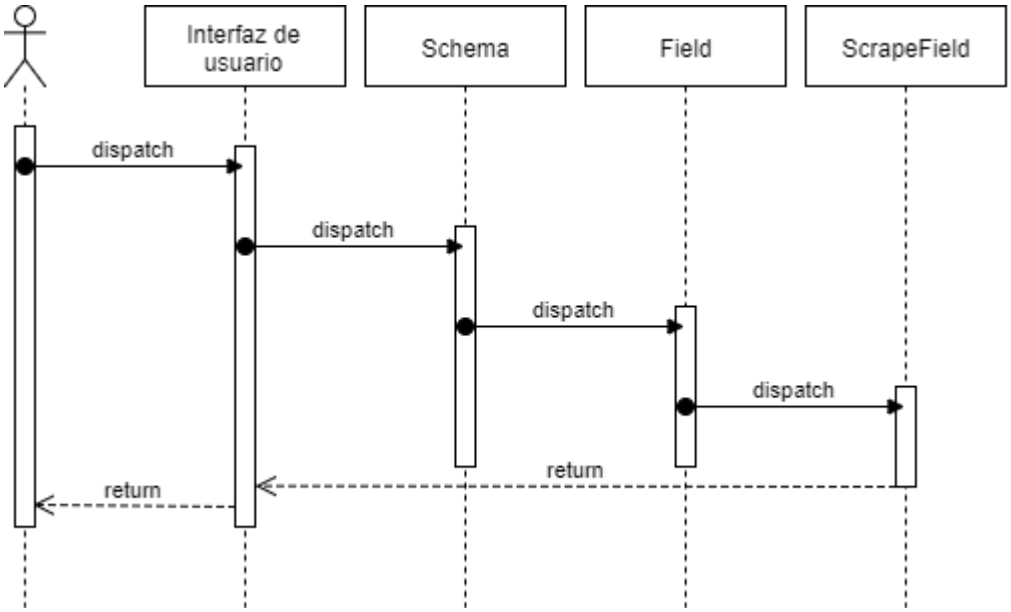


Diagrama 4. Diagrama de secuencia que se obtuvo en la versión final de la herramienta.

Estilo de la herramienta

Para la interfaz de la herramienta se optó por solo un formulario con un input en el que se ingrese la URL donde esta alojado el artículo al que se le quiere realizar la extracción de metadatos.

Imagen 1. Vista inicial de la herramienta donde se tiene un formulario para el ingreso de la URL del artículo.

También se agregaron las vistas que contienen los Campos y los Métodos de extracción para que el usuario de la herramienta pueda cargar los datos que quisiese.

En las capturas siguientes se observan las opciones de Field:

el listado de Campos existentes con las opciones de edic ión, borrado.

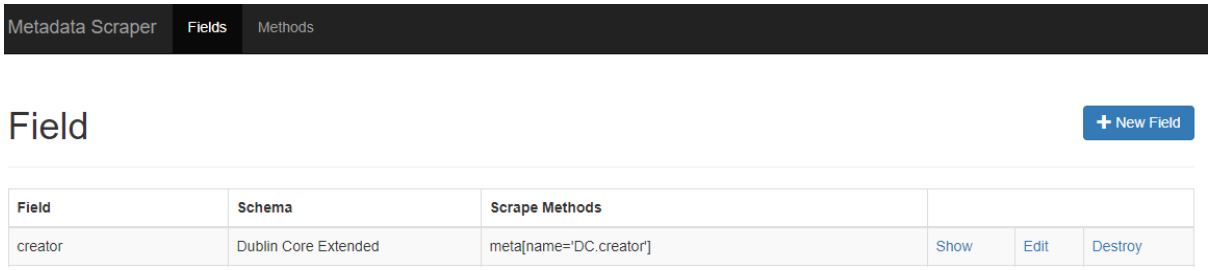


Imagen 2. Vista donde se listan los Fields, esquema al que pertenece, método de extracción y opciones para ver, editar y borrar

y carga de un nuevo campo.

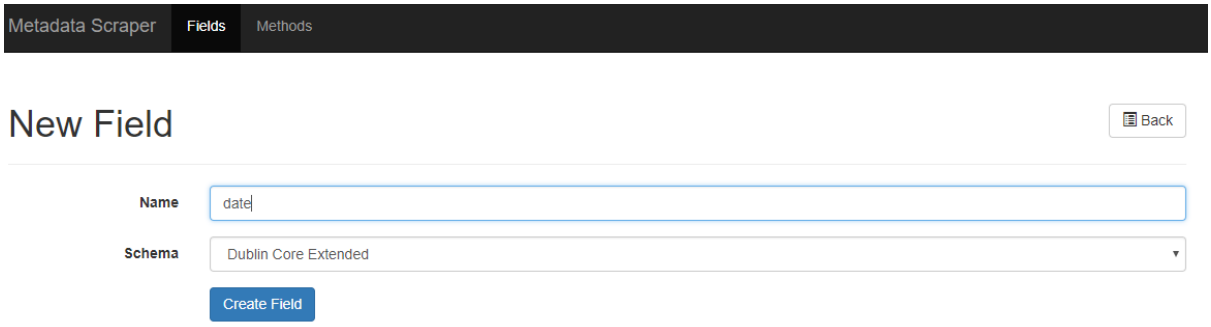


Imagen 3. Vista para cargar o editar un Field

Por último se exponen las capturas de pantalla de las opciones pertenecientes a los métodos de extracción:

el listado de Métodos existentes donde se puede observar a qué campo y esquema pertenece el método y las opciones de edición, borrado.

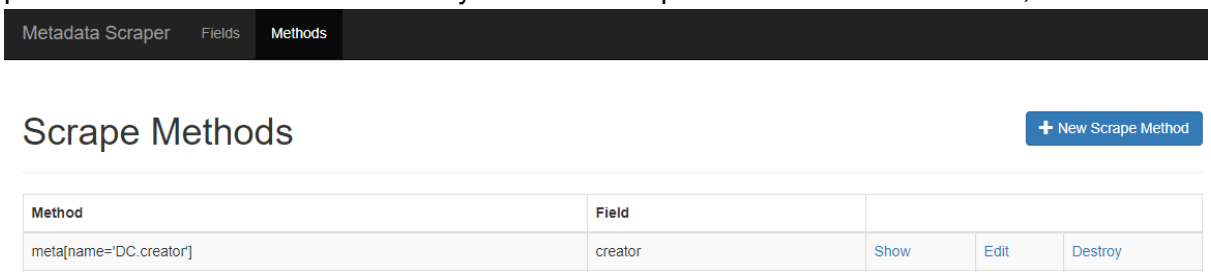


Imagen 4.Listado de Métodos de extracción

carga de un nuevo método especificando a qué campo pertenece.

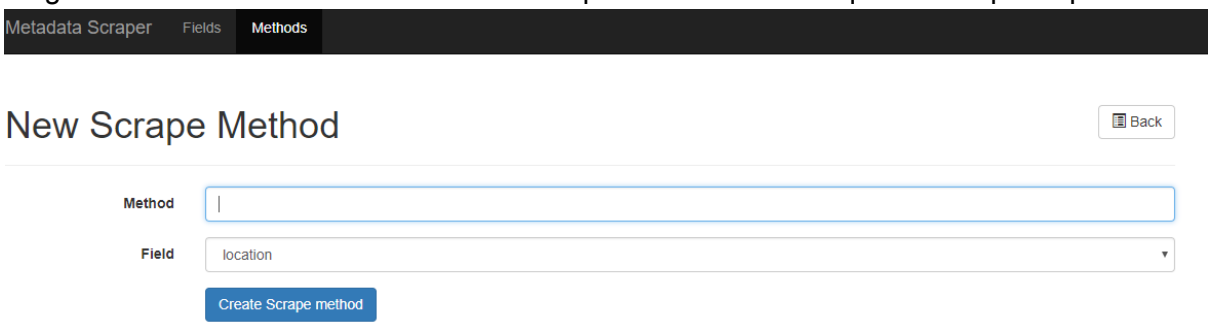


Imagen 5.Vista de carga o edición de un método de extracción

Resultados/Experimentación

En esta sección se exponen los resultados obtenidos de ingresar una URL de un artículo científico en la herramienta de extracción de metadatos. Como se comentó anteriormente, se seleccionaron varias plataformas que alojan artículos científicos. Estos sitios fueron: SEDICI, Portal de Revistas de la UNLP, Elsevier, Springer, Portal de Revistas UCR.

En primer lugar se exponen las tablas en donde se observa qué etiqueta se utiliza para especificar un metadato, de esta manera se conoce de donde se obtendrá la información. En segundo lugar, se eligieron artículos específicos de cada una de las plataformas. Con esto se presentan las tablas en donde se tiene por cada etiqueta los datos que se van a extraer. Finalmente, se ingresan los artículos a través de su URL en la herramienta y se muestran los resultados en formato JSON.

Exposición de información de los distintos sitios de prueba

En esta sección se exponen los metadatos que pertenecen a las plataformas seleccionadas y la etiqueta que se utilizará para extraer la información.

Tabla Dspace extraído de SEDICI:

Metadato	**Etiqueta**

Autor	DC.creator
Título	DC.title
Subtitulo	DCTERMS.alternative
Fecha de publicación	DCTERMS.issued
Localización Física	DCTERMS.spatial
Alcance temporal	DCTERMS.temporal
Extensión	DCTERMS.extent
Entidad de origen	DC.identifier
Resumen	DCTERMS.abstract
Palabras clave	citation_keywords
Localización electrónica	DC.identifier
Materia	DC.subject
Descritores	citation_keywords
ISSN	citation_issn
otros identificadores	citation_doi
Título de la Serie	citation_journal_title
Volumen y Número de la serie	citation_volume
Nombre del evento	DC.subject

Tabla OJS 2 extraído de Portal de Revistas académicas de la Universidad de Costa Rica:

Metadato	**Etiqueta**
Autor	DC.Creator.PersonalName
Título	DC.Title
Subtitulo	no posee
Fecha de publicación	DC.Date.dateSubmitted
Localización Física	no posee
Alcance temporal	no posee
Extensión	DC.Identifier.pageNumber
Entidad de origen	DC.Source
Resumen	DC.Description
Palabras clave	citation_keywords
Localización electrónica	DC.Identifier.URI
Materia	DC.Type.articleType
Descritores	citation_keywords
ISSN	DC.Source.ISSN
otros identificadores	DC.Identifier.DOI
Título de la Serie	DC.Source
Volumen y Número de la serie	DC.Source.Volume

Nombre del evento	DC.subject

Tabla editorial (no normalizado) extraído de Elsevier España:

Metadato	**Etiqueta**
Autor	elsevierItemAutores
Título	elsevierItemTitulo
Subtitulo/título alternativo	elsevierItemTitulosAlternativos
Fecha de publicación	elsevierItemFechas
Localización Física	no posee
Alcance temporal	no posee
Extensión	no posee
Entidad de origen	pag
Resumen	elsevierItemsResumenes
Palabras clave	elsevierItemPalabrasClaves
Localización electrónica	pag
Materia	no posee
Descritores	elsevierItemPalabrasClaves
ISSN	no posee
otros identificadores	tituloRevista caja

Título de la Serie	tituloRevista caja
Volumen y Número de la serie	pag
Nombre del evento	navAnt

Tabla OJS 3 extraído del Portal de Revistas de la UNLP

Metadato	**Etiqueta**
Autor	authors
Título	page_title
Subtitulo/título alternativo	no posee
Fecha de publicación	published
Localización Física	no posee
Alcance temporal	no posee
Extensión	no posee
Entidad de origen	no posee
Resumen	abstract
Palabras clave	no posee
Localización electrónica	pdf
Materia	issue
Descritores	no posee
ISSN	no posee

otros identificadores	doi
Volumen y Número de la serie	issue
Nombre del evento	no posee

Tabla Springer

Metadato	**Etiqueta**
Título de la revista	JournalTitle
Año de publicación	ArticleCitation_Year
Volumen	ArticleCitation_Volume
Número	ArticleCitation_Issue
Extensión	ArticleCitation_Pages
Título del artículo	MainTitleSection h1
Nombre de los autores	authors__name
Información de contacto	authors__contact
Resumen	Abstract
Palabras clave	Keyword
Licencia	ArticleCopyright
Cómo citar	citethis-text
DOI	doi-url

Nombre de la editorial	`publisher-name`
ISSN Físico	`print-issn`
ISSN Online	`electronic-issn`

Resultados de la extracción

En esta sección se muestran las tablas con la información de los artículos seleccionados, pertenecientes a las plataformas anteriormente comentadas, donde se extrae la información y las capturas de pantalla de los resultados de la herramienta en formato JSON.

Dspace SEDICI

Registro de información

Se seleccionó la url de uno de los artículos alojados en el repositorio institucional SEDICI, se puede ingresar a través de la URL:

http://sedici.unlp.edu.ar/handle/10915/53638. La información total extraíble de éste es:

`Metadato`	**Información a Extraer**
`dc.date.accessioned`	2016-07-01T13:59:21Z
`dc.date.available`	2016-07-01T13:59:21Z
`dc.date.issued`	2016-07
`dc.identifier.uri`	http://hdl.handle.net/10915/53638
`dc.description.abstract`	El objetivo de este trabajo es describir alternativas incorporadas en el formato EPUB3 para promover el acceso a la producción académica y científica de las instituciones por parte de personas con discapacidades visuales. Como punto de partida se toma la figura del repositorio institucional como espacio que alberga y difunde esta producción, y cuyos objetivos incluyen darle mayor visibilidad y

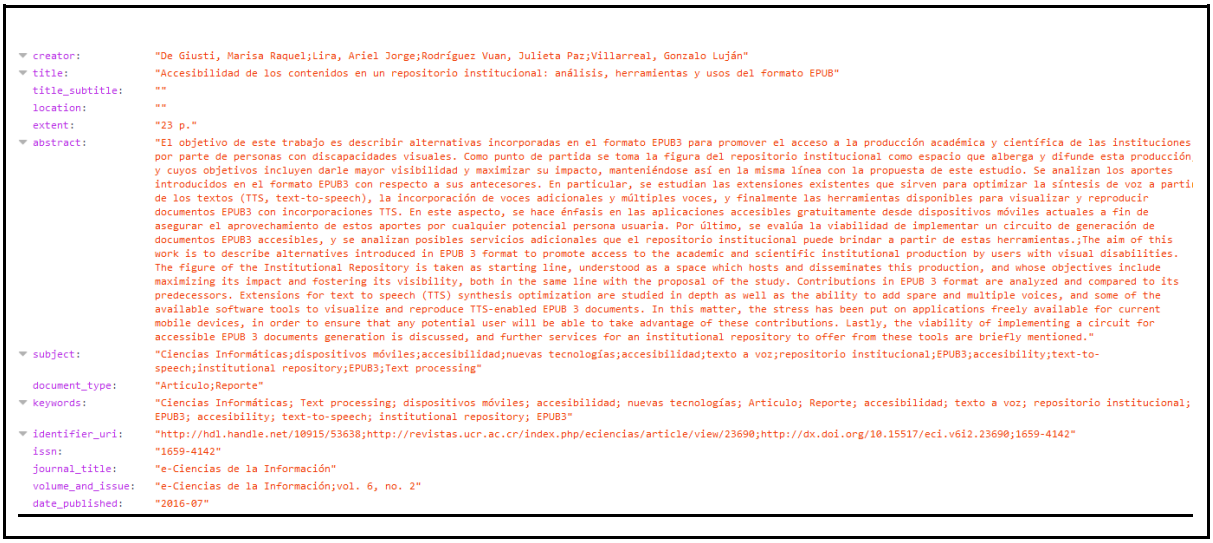
	maximizar su impacto, manteniéndose así en la misma línea con la propuesta de este estudio. Se analizan los aportes introducidos en el formato EPUB3 con respecto a sus antecesores. En particular, se estudian las extensiones existentes que sirven para optimizar la síntesis de voz a partir de los textos (TTS, text-to-speech), la incorporación de voces adicionales y múltiples voces, y finalmente las herramientas disponibles para visualizar y reproducir documentos EPUB3 con incorporaciones TTS. En este aspecto, se hace énfasis en las aplicaciones accesibles gratuitamente desde dispositivos móviles actuales a fin de asegurar el aprovechamiento de estos aportes por cualquier potencial persona usuaria. Por último, se evalúa la viabilidad de implementar un circuito de generación de documentos EPUB3 accesibles, y se analizan posibles servicios adicionales que el repositorio institucional puede brindar a partir de estas herramientas.
dc.description.abstract	The aim of this work is to describe alternatives introduced in EPUB 3 format to promote access to the academic and scientific institutional production by users with visual disabilities. The figure of the Institutional Repository is taken as starting line, understood as a space which hosts and disseminates this production, and whose objectives include maximizing its impact and fostering its visibility, both in the same line with the proposal of the study. Contributions in EPUB 3 format are analyzed and compared to its predecessors. Extensions for text to speech (TTS) synthesis optimization are studied in depth as well as the ability to add spare and multiple voices, and some of the available software tools to visualize and reproduce TTS-enabled EPUB 3

	documents. In this matter, the stress has been put on applications freely available for current mobile devices, in order to ensure that any potential user will be able to take advantage of these contributions. Lastly, the viability of implementing a circuit for accessible EPUB 3 documents generation is discussed, and further services for an institutional repository to offer from these tools are briefly mentioned.
dc.format.extent	23 p.
dc.language	es
dc.title	Accesibilidad de los contenidos en un repositorio institucional: análisis, herramientas y usos del formato EPUB
dc.type	Artículo
sedici.identifier.uri	http://revistas.ucr.ac.cr/index.php/ecienciais/article/view/23690
sedici.identifier.doi	http://dx.doi.org/10.15517/eci.v6i2.23690
sedici.identifier.issn	1659-4142

sedici.creator.person	De Giusti, Marisa Raquel;Lira, Ariel Jorge; Rodríguez Vuan, Julieta Paz; Villarreal, Gonzalo Luján
sedici.subject.materias	Ciencias Informáticas
sedici.subject.other	dispositivos móviles; accesibilidad; nuevas tecnologías
sedici.subject.keyword	accesibilidad; texto a voz; repositorio institucional; EPUB3; accesibility; text-to-speech; institutional repository; EPUB3
sedici.description.fulltext	true
mods.originInfo.place	Servicio de Difusión de la Creación Intelectual (SEDICI)
sedici.subtype	Reporte
sedici.rights.license	Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0)
sedici.rights.uri	http://creativecommons.org/licenses/by-nc-nd/4.0/
sedici.description.peerReview	peer-review
sedici.relation.journalTitle	e-Ciencias de la Información
sedici.relation.journalVolumeAndIssue	vol. 6, no. 2

`sedici.subject.acmcss98`	Text processing

Los resultados ingresando la URL a la herramienta fueron:



OJS2 (Revista e-ciencias, Portal de Revistas de la Universidad de Costa Rica):

Se seleccionó la url de uno de los artículos alojados en el repositorio institucional SEDICI, se puede ingresar a través de la URL:

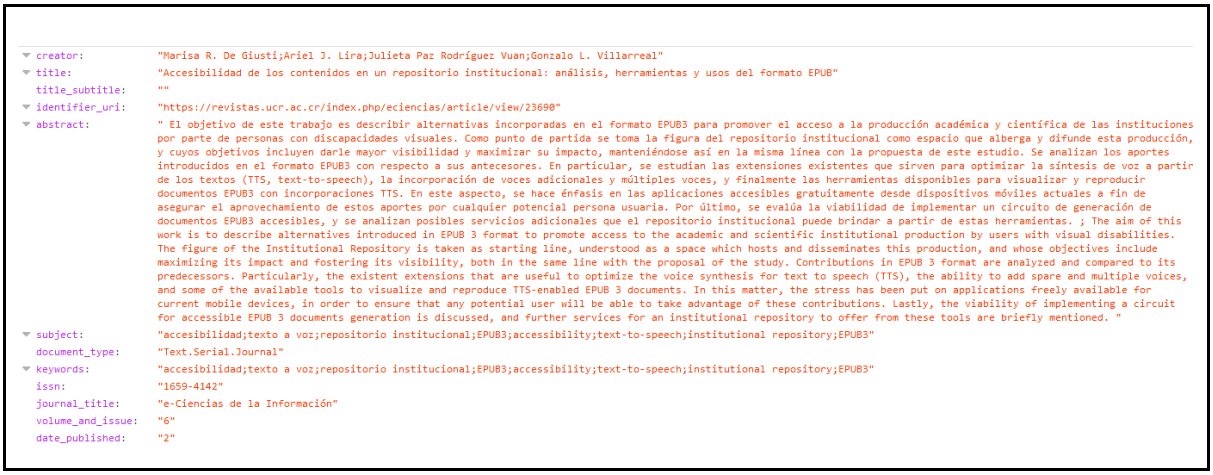
https://revistas.ucr.ac.cr/index.php/eciencias/article/view/23690. la información total extraíble de éste es:

Metadato	**Información a extraer**
`Título`	Accesibilidad de los contenidos en un repositorio institucional: análisis, herramientas y usos del formato EPUB
`Creador/a`	-Marisa R. De Giusti; Universidad Nacional de La Plata. Proyecto de Enlace en Bibliotecas. Servicio de Difusión de la Creación Intelectual. Comisión de Investigaciones Científicas de la Provincia.; Argentina -Ariel J. Lira; Universidad Nacional de La Plata. Proyecto de Enlace en Bibliotecas. Servicio de Difusión de la Creación Intelectual.; Argentina -Julieta Paz Rodríguez Vuan; Universidad Nacional de La Plata. Proyecto de Enlace

	en Bibliotecas. Servicio de Difusión de la Creación Intelectual.; Argentina -Gonzalo L. Villarreal; Universidad Nacional de La Plata. Proyecto de Enlace en Bibliotecas. Servicio de Difusión de la Creación Intelectual.; Argentina
Materia	-accesibilidad; texto a voz; repositorio institucional; EPUB3
Descripción	El objetivo de este trabajo es describir alternativas incorporadas en el formato EPUB3 para promover el acceso a la producción académica y científica de las instituciones por parte de personas con discapacidades visuales. Como punto de partida se toma la figura del repositorio institucional como espacio que alberga y difunde esta producción, y cuyos objetivos incluyen darle mayor visibilidad y maximizar su impacto, manteniéndose así en la misma línea con la propuesta de este estudio. Se analizan los aportes introducidos en el formato EPUB3 con respecto a sus antecesores. En particular, se estudian las extensiones existentes que sirven para optimizar la síntesis de voz a partir de los textos (TTS, text-to-speech), la incorporación de voces adicionales y múltiples voces, y finalmente las herramientas disponibles para visualizar y reproducir documentos EPUB3 con incorporaciones TTS. En este aspecto, se hace énfasis en las aplicaciones accesibles gratuitamente desde dispositivos móviles actuales a fin de asegurar el aprovechamiento de estos aportes por cualquier potencial persona usuaria. Por último, se evalúa la viabilidad de implementar un circuito de generación de documentos EPUB3 accesibles, y se analizan posibles servicios adicionales que el repositorio institucional puede brindar a partir de estas herramientas.

Editorial	University of Costa Rica. School of Library and Information Science
Colaborador/a	Servicio de Difusión de la Creación Intelectual, Universidad Nacional de La Plata
Fecha	2016-06-30
Tipo	Artículo revisado por pares
Formato	EPUB, PDF
Identificador	https://revistas.ucr.ac.cr/index.php/eciencias/article/view/23690 https://doi.org/10.15517/eci.v6i2.23690
Fuente	e-Ciencias de la Información; Volumen 6, número 2: julio-diciembre 2016
Idioma	es
Relación	Curriculum Vitae de Marisa R. De giusti (650 KB) Curriculum Vitae Gonzalo L. Villarreal (125KB) Curriculum Vitae Ariel J. Lira (265 KB) Curriculum Vitae Julieta Rodriguez Vuan (75KB) Carta originalidad (307 KB)

Los resultados ingresando la URL a la herramienta fueron:



Elsevier

Registro de información

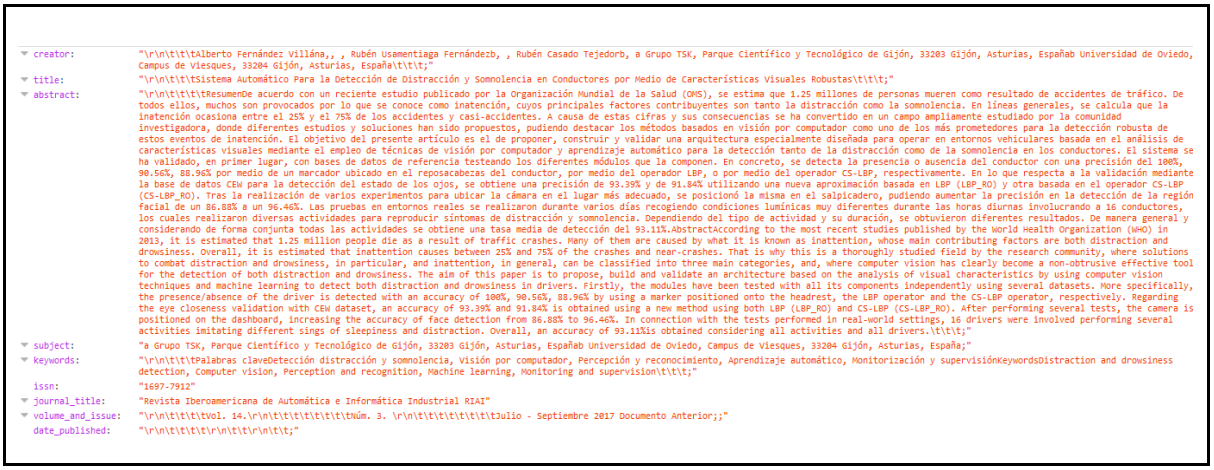
Se seleccionó la url de uno de los artículos alojados en la editorial Elsevier España, se puede ingresar a través de la URL:

http://www.elsevier.es/es-revista-revista-iberoamericana-automatica-e-informatica-331-articulo-sistema-automatico-para-deteccion-distraccion-S1697791217300183. La información total extraíble de éste es:

Metadato	**Información a extraer**
Titulo	Sistema Automático Para la Detección de Distracción y Somnolencia en Conductores por Medio de Características Visuales Robustas
Titulo Alternativo	Automatic System to Detect Both Distraction and Drowsiness in Drivers Using Robust Visual Features
Autores	Alberto Fernandez Villána,, , Rubén Usamentiaga Fernández, , Rubén Casado Tejedor,
Fechas	Julio-Septiembre 2017
Resumen	De acuerdo con un reciente estudio publicado por la Organización Mundial de la Salud (OMS), se estima que 1.25 millones de personas mueren como resultado de accidentes de tráfico. De todos ellos, muchos son provocados por lo que se conoce como inatención, cuyos principales factores contribuyentes son tanto la distracción como la somnolencia. En líneas generales, se calcula

	que la inatención ocasiona entre el 25% y el 75% de los accidentes y casi-accidentes. A causa de estas cifras y sus consecuencias se ha convertido en un campo ampliamente estudiado por la comunidad investigadora, donde diferentes estudios y soluciones han sido propuestos, pudiendo destacar los métodos basados en visión por computador como uno de los más prometedores para la detección robusta de estos eventos de inatención. El objetivo del presente artículo es el de proponer, construir y validar una arquitectura especialmente diseñada para operar en entornos vehiculares basada en el análisis de características visuales mediante el empleo de técnicas de visión por computador y aprendizaje automático para la detección tanto de la distracción como de la somnolencia en los conductores. El sistema se ha validado, en primer lugar, con bases de datos de referencia testeando los diferentes módulos que la componen. En concreto, se detecta la presencia o ausencia del conductor con una precisión del 100%, 90.56%, 88.96% por medio de un marcador ubicado en el reposacabezas del conductor, por medio del operador LBP, o por medio del operador CS-LBP, respectivamente. En lo que respecta a la validación mediante la base de datos CEW para la detección del estado de los ojos, se obtiene una precisión de 93.39% y de 91.84% utilizando una nueva aproximación basada en LBP (LBP_RO) y otra basada en el operador CS-LBP (CS-LBP_RO). Tras la realización de varios experimentos para ubicar la cámara en el lugar más adecuado, se posicionó la misma en el salpicadero, pudiendo aumentar la precisión en la detección de la región facial de un 86.88% a un 96.46%. Las pruebas en entornos reales se realizaron durante varios días recogiendo condiciones lumínicas muy diferentes durante las horas diurnas involucrando a 16 conductores, los cuales realizaron diversas actividades para reproducir síntomas de distracción y somnolencia. Dependiendo del tipo de actividad y su duración, se obtuvieron diferentes resultados. De manera general y considerando de forma conjunta todas las actividades se obtiene una tasa media de detección del 93.11%.
Palabras Clave	Detección distracción y somnolencia, Visión por computador, Percepción y reconocimiento, Aprendizaje automático, Monitorización y supervisión
Volumen y Número	Vol. 14. Núm. 3.

Los resultados ingresando la URL a la herramienta fueron:



OJS 3 Portal de Revistas de la UNLP:

Registro de información:

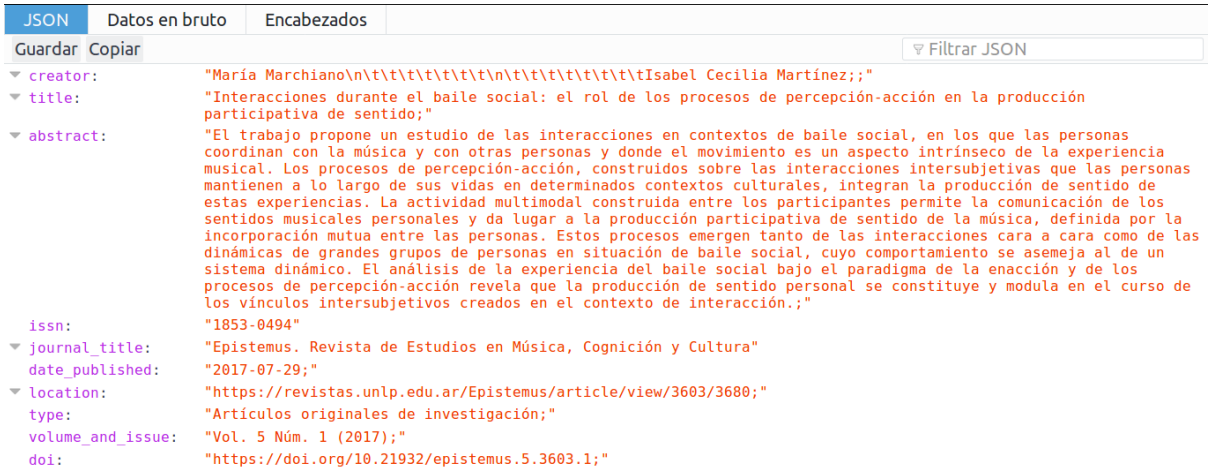
Se seleccionó la url de uno de los artículos alojados en el Portal de Revistas de la UNLP, específicamente de la revista Epistemus. Se puede ingresar a través de la URL:

https://revistas.unlp.edu.ar/Epistemus/article/view/3603. La información total extraíble de éste es:

Metadato	**Información a extraer**
Titulo de la artículo	Interacciones durante el baile social: el rol de los procesos de percepción-acción en la producción participativa de sentido
Título del revista	Epistemus. Revista de Estudios en Música, Cognición y Cultura
ISSN	1853-0494
Autor	María Marchiano; Isabel Cecilia Martínez
Fecha de publicación	2017/07/29
Volumen de la revista	5
Número de la revista	1
DOI	10.21932/epistemus.5.3603.1
Identificador	https://revistas.unlp.edu.ar/Epistemus/article/view/3603

PDF identificador	https://revistas.unlp.edu.ar/Epistemus/article/download/3603/3680

Los resultados ingresando la URL a la herramienta fueron:



Springer

Registro de información

Se seleccionó la url de uno de los artículos alojados en la editorial Elsevier, específicamente de la revista Hydrobiologia. Se puede ingresar a través de la URL:

https://link.springer.com/article/10.1007/s10750-006-0421-8. La información total extraíble de éste es:

Metadato	**Información a extraer**
Título de la revista	Hydrobiologia
Año de publicación	February 2007
Volumen	Volume 577
Número	Issue 1
Extensión	pp 107–125

Título del artículo	Hydrographic, geomorphologic and fish assemblage relationships in coastal lagoons
Nombre de los autores	Angel Pérez-Ruzafa;Mª Carmen Mompeán Concepción Marcos
Resumen	In this study, 40 Atlanto-Mediterranean coastal lagoons were analyzed in order to evaluate the extent to which their ecological characteristics depend on hydrographic, trophic or geomorphologic features. Fish species richness increases with lagoon volume and the openness parameter, which characterizes the potential influence of the sea on general lagoon hydrology and is related to the total transversal area of the inlets, which connect the lagoon to the sea. On the other hand, the number of species decreases exponentially with the phosphate concentration in water. The fishing yield increases with the chlorophyll a concentration in the water column and exponentially with shoreline development. With respect to the fish assemblage composition, geomorphologic features alone explain 22% of the variance in the canonical analyses and an additional 75% when including the hydrographic and trophic characteristics of the lagoon, the latter on its own explaining only 3% of the observed differences
Palabras clave	Coastal lagoon Fish assemblages Typification Species richness Fishing yield
Licencia	
Cómo citar	Pérez-Ruzafa, A., Mompeán, M.C. & Marcos, C. Hydrobiologia (2007) 577: 107. https://doi.org/10.1007/s10750-006-0421-8
DOI	https://doi.org/10.1007/s10750-006-0421-8
Nombre de la editorial	Kluwer Academic Publishers
ISSN Físico	0018-8158
ISSN Online	1573-5117

Los resultados ingresando la URL a la herramienta fueron:

JSON | Datos en bruto | Encabezados

Guardar Copiar | Filtrar JSON

how_to_cite: "Pérez-Ruzafa, A., Mompeán, M.C. & Marcos, C. Hydrobiologia (2007) 577: 107. https://doi.org/10.1007/s10750-006-0421-8;"
doi: "https://doi.org/10.1007/s10750-006-0421-8;"
publisher_name: "Kluwer Academic Publishers;"
creator: "Angel Pérez-RuzafaMª Carmen MompeánConcepción Marcos;;;"
title: "Hydrographic, geomorphologic and fish assemblage relationships in coastal lagoons;"
abstract: "In this study, 40 Atlanto-Mediterranean coastal lagoons were analyzed in order to evaluate the extent to which their ecological characteristics depend on hydrographic, trophic or geomorphologic features. Fish species richness increases with lagoon volume and the openness parameter, which characterizes the potential influence of the sea on general lagoon hydrology and is related to the total transversal area of the inlets, which connect the lagoon to the sea. On the other hand, the number of species decreases exponentially with the phosphate concentration in water. The fishing yield increases with the chlorophyll a concentration in the water column and exponentially with shoreline development. With respect to the fish assemblage composition, geomorphologic features alone explain 22% of the variance in the canonical analyses and an additional 75% when including the hydrographic and trophic characteristics of the lagoon, the latter on its own explaining only 3% of the observed differences.;"
keywords: "Coastal lagoon Fish assemblages Typification Species richness Fishing yield ;;;;;"
issn: "1573-5117;"
issn_fisico: "0018-8158;"
journal_title: "HydrobiologiaRapp Comm int Mer Medit 32 fascOceanologia ActaFAO Studies and ReviewsICES Journal of Marine ScienceRapp Comm int Mer MeditAcqua and AriaEcologyCahiers de Biologie MarineEcological ModellingAtti del VII Congreso AIOLPublicaciones Especiales del Instituto Español de OceanografíaTopics in Marine Biology (Scientia Marina)HydrobiologiaPublicaciones Especiales del Instituto Español de OceanografíaFisheries Management and EcologyRapp Comm int Mer MeditRapp Comm int Mer MeditRapp Comm int Mer MeditJournal of Fish BiologyLebanese Science BulletinJournal de Recherche Oceanographique III bolletinActa Botánica MalacitanaHydrobiologiaAnnales du Musée d'historie naturelle de Marseille -ZoologieTravaux du laboratoire de géologieJournal de recherche oceanographiqueUnesco Technical papers in Marine ScienceVie et MilieuVie et MilieuVie MilieuFAO Studies and ReviewsGeneral Fisheries Council for the Mediterranean Studies and Reviews 2 n°Scientia MarinaFAO Studies and ReviewsAnné BiologiqueFAO Studies and ReviewsCybiumBoletín del Instituto Español de OceanografíaEstuarine, Coastal and Shelf ScienceScientia MarinaMarine EcologyVie et MilieuFisheries ResearchRapp Comm int Mer MeditPublicaciones Especiales del Instituto Español de OceanografíaHydrobiologiaJournal of Fish BiologyHydrobiologiaMarine Pollution BulletinVie MilieuEstuarine, Coastal and Shelf ScienceMarine BiologyPSZN I:Marine EcologyRapp Comm int Mer MeditFAO Studies and ReviewsRapp Comm int Mer MeditBull Mus Hist Nat MarseilleFreshwater BiologyFAO Studies and ReviewsFAO Studies and ReviewsRapp Comm int Mer MeditVerh Internat Verein LimnolBrazilian Journal of BiologyEcologyRapp Comm int Mer MeditUnesco Technical Papers in Marine ScienceUNESCO Technical Papers in Marine ScienceHydrobiologiaAn Centro Cienc del Mar y Limnol Univ Nal Autón México;;"
volume_and_issue: "Volume 577,;"
date_published: "February 2007,;"
extent: "pp 107–125;"
contact: "Email author;"
license: "© Springer Science+Business Media B.V. 2007© Springer Science+Business Media B.V. 2007;;"

Análisis de los resultados

A partir de las pruebas realizadas se observó en primera medida que los sitios en los que se encontraron metadatos mal formados fueron los que alojaban su información en el cuerpo del documento como son los casos de Elsevier y el Portal de Revistas de la UNLP.

Los sitios que tenían su información expuesta en el cuerpo del documento tienden a guardar también en los campos meta. Esto hizo que la forma de extraer información utilizara los dos métodos de extracción para que el resultado de la extracción sea más completa.

Los sitios que tenían su información expuesta en el cuerpo del documento muchas veces no identificaban los campos. Esto hace que el método de extracción tuviese formas complejas como en el caso del Portal de Revistas de la UNLP donde para poder conseguir el valor de "Volume and Issue" se debió cargar en la herramienta como método de extracción: `.issue div[1] div[class='value']`.

La identificación del schema puede traer conflictos puesto que las páginas que normalizan sus datos tienden a identificar varios esquemas de metadatos. Esto significa que la herramienta puede llegar a utilizar, por la prioridad establecida, un esquema que no fue el que realmente se quería utilizar.

Plugin especializado en el formulario de SEDICI

Para ejemplificar un poco el uso de la herramienta se desarrolló de una extensión sobre el navegador Web Chrome. Esta extensión tiene como objetivo-agilizar la carga de un artículo científico en el repositorio SEDICI. Para lograr esto se utilizó el formulario de carga de

documentos de este repositorio. La tarea entonces de la extensión es a través de la URL del artículo científico, conectarse con la herramienta, extraer sus metadatos y devolver el resultado en los campos del formulario de SEDICI. Para el desarrollo de esta extensión se utilizó el lenguaje de programación Javascript y el manual de desarrollo de extensiones del Navegador Chrome.

Desarrollo

Para el desarrollo de esta extensión se creo un formulario para el ingreso de la URL donde esta alojado el artículo que se quiere cargar dentro del repositorio SEDICI, la extensión se conecta con la herramienta que realiza la extracción de metadatos y devuelve la información cargando el formulario de manera automática.

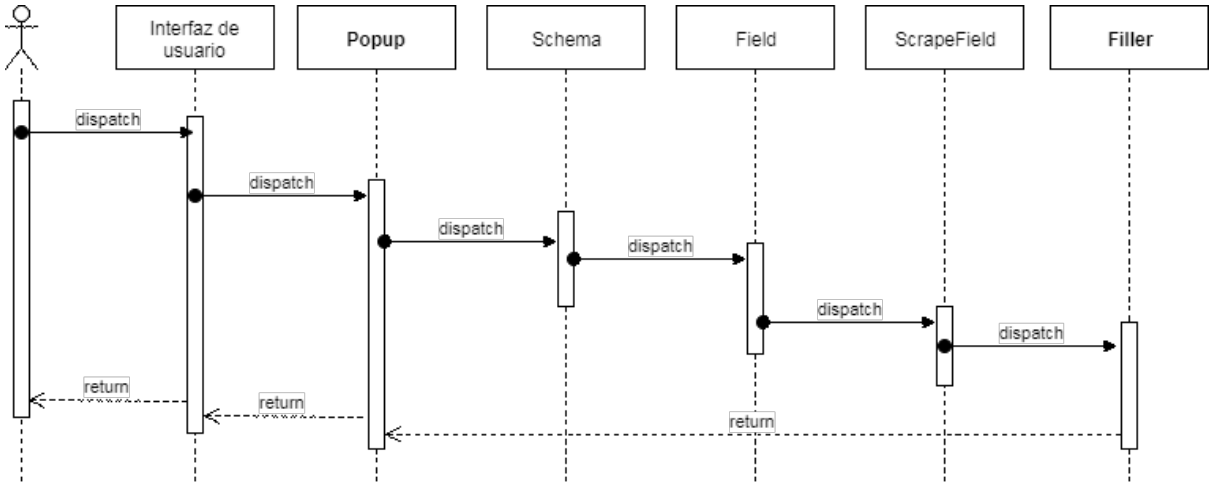


Diagrama 5. Flujo de una petición para extracción de metadatos utilizando la extensión en navegador web Chrome

En el código siguiente se expone el código que realiza la carga de los campos del formulario con la información recolectada por la herramienta:

```
var fields =
['aspect_submission_StepTransformer_field_sedici_creator_person',

'aspect_submission_StepTransformer_field_dc_title',
'aspect_submission_StepTransformer_field_sedici_title_subtitle',

'aspect_submission_StepTransformer_field_mods_location',

'aspect_submission_StepTransformer_field_dc_format_extent',
'aspect_submission_StepTransformer_field_dc_description_abstract',
'aspect_submission_StepTransformer_field_sedici_subject_keyword',
'aspect_submission_StepTransformer_field_sedici_identifier_uri',
'aspect_submission_StepTransformer_field_mods_originInfo_place',
'aspect_submission_StepTransformer_field_dc_type',
'aspect_submission_StepTransformer_field_sedici_identifier_issn',
'aspect_submission_StepTransformer_field_sedici_relation_journalTitle',
'aspect_submission_StepTransformer_field_sedici_relation_journalVolumeAn
```

```
dIssue',
'aspect_submission_StepTransformer_field_dc_date_issued_year',
'aspect_submission_StepTransformer_field_dc_date_issued_month']


var year = response.date_published.split('-')[0]

var month = response.date_published.split('-')[1]

console.log(response.document_type)

var values = [response.creator,

              response.title,

              response.title_subtitle,

              response.location,

              response.extent,

              response.abstract,

              response.keywords,

              response.identifier_uri,

              response.subject,

              response.document_type,

              response.issn,

              response.journal_title,

              response.volume_and_issue,

              year,

              month

              ]


for (var i = 0;i < fields.length; i++) {

  elem = document.getElementById(fields[i])

  if (elem && values[i]) elem.value = values[i]

}
```

Ejemplo 5. Código de la extensión para el navegador web Chrome que se conecta con la herramienta desarrollada

Empaquetado de la extensión

Para poder ingresar la extensión a google chrome [Google 17] se deben seguir ciertas normativas que especifican la forma de los documentos de la aplicación, carpetas, archivo de configuración, etc.

Extension Chrome Diseño

Este plugin está publicado en el Chrome Web Store, y puede descargarse a través del link: https://chrome.google.com/webstore/detail/autofillmagic3000/hnccjnjemjnmkecellcgbjfkllbaaimb?utm_source=chrome-app-launcher-info-dialog

A continuación se muestra su diseño visto desde la vista de extensiones del navegador (chrome://extensions/)

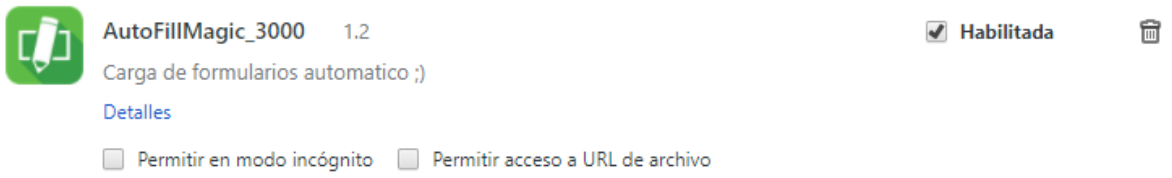
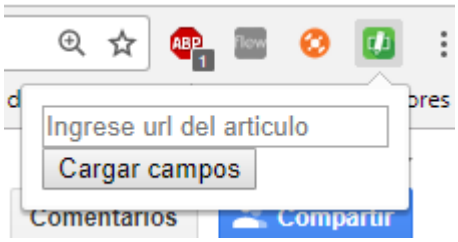


Imagen 6. Captura de pantalla de la extensión una vez agregada al navegador

En la siguiente captura, se expone el diseño de la extensión en uso



Ejemplo 7. Vista inicial de la extensión

Formulario de SEDICI

Como se puede ver en la captura de pantalla, el formulario de SEDICI está dividido en cuatro etapas para la carga de un nuevo artículo. En la captura se solicita la carga de uno de los ejemplos de artículos anteriormente utilizados. http://sedici.unlp.edu.ar/handle/10915/53638

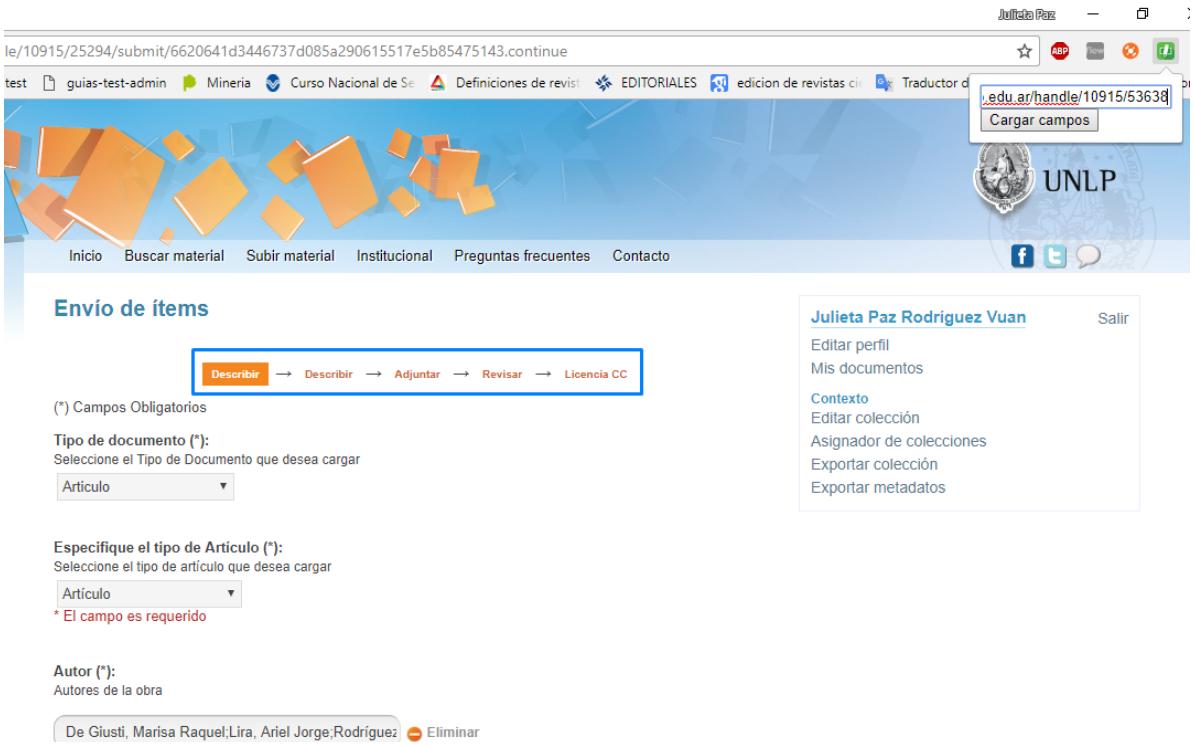


Imagen 8. Captura de pantalla de los resultados obtenidos del uso de la extensión.

Resultados

Utilizando la extensión, los campos del formulario de SEDICI fueron cargados con éxito a partir del JSON que la herramienta retorna como respuesta. Pero, hay cuestiones que deben mejorarse.

En primer lugar, hay campos como es el caso de Autor que, como se observan en la captura, es cargado con todos los autores del artículo separados por ";" y esto no es correcto. Los autores son cargados en campos Autor por separado que van agregándose "en el momento".

En segundo lugar, se debe solicitar los datos de carga a la extensión en cada una de las etapas. Esto es así, ya que la URI del formulario cambia por cada etapa quitando la posibilidad de identificar en qué parte del formulario se esta utilizando la extensión.

Por último, los campos del formulario que son selectores hacen complejo el mapeo para saber que seleccionar al momento de la carga. Este campo, es posible cargarlo con la extensión cuando el artículo, cuando proviene de repositorios institucionales que utilizan DSpace ya que coinciden con los selectores utilizados en SEDICI.

Capítulo 4 | Conclusiones y trabajos futuros

Conclusiones

A lo largo de esta tesina se explicaron las distintos motivos que dieron como objetivo la creación de esta herramienta, se detalló la investigación realizada del marco teórico junto con el análisis de los requerimientos funcionales para luego llevar al lector a través de la

implementación de la herramienta y finalmente mostrar los casos concretos de uso de la herramienta.

Un ejemplo que demuestran los beneficios obtenidos por la herramienta se expresa en el flujo de trabajo que se realiza en el Repositorio institucional SEDICI. Diariamente, se realizan cargas de ítems dentro de este repositorio de manera manual por parte de los administradores. La carga manual de metadatos trae algunas desventajas, como por ejemplo la introducción de errores humanos al momento de duplicar la información en el ingreso de datos, un mayor gasto de tiempo en el ingreso de cada ítem, la limitación en las plataformas con las que los administradores pueden interoperar, entre otras. Con la herramienta desarrollada se ha logrado realizar la transformación de los metadatos recolectados en una plataforma al tipo de metadato que se utiliza dentro del repositorio, disminuye las desventajas anteriormente expuestas. Esto es así dado que por ejemplo: con la extensión para el navegador web Chrome las tareas de carga ahora pueden realizarse de manera automatizada disminuyendo el tiempo de carga por ítem, mejorar la calidad de los objetos digitales al poder compararse la información guardada con el sitio donde se recuperó la información, se ha aumentado la interoperabilidad con otros sitios que no realizan una normalización estándar.

Mejoras en la herramienta actual

Una de las cosas que la herramienta no tuvo en cuenta fue que, como lo comentamos anteriormente, algunos de los datos que se recuperan dentro del cuerpo del documento no tienen un formato correcto. Esto podría resolverse mejorando los metadatos que se recolectan como por ejemplo, eliminando los espacios en blanco.

Otras de las mejoras que se pueden realizar a la herramienta actual es darle al usuario la posibilidad de que se retornen resultados en distintos formatos como XML o CSV.

Trabajo futuros

Existen diversas líneas de mejoras que se pueden realizar a la herramienta desarrollada en esta tesina, algunas de ellas fueron detectadas durante la realización de la misma, mientras que otras fueron planteadas ante el surgimiento de nuevas necesidades. A continuación se detallan las distintas líneas de mejoras previamente mencionadas con una explicación de su utilidad y necesidad.

En primer lugar, una de las mejoras que se pueden realizar en la herramienta es la idea de agregar nuevos tipos de documentos para la extracción de metadatos. Esto se podría realizar agregando un nuevo tipo de objeto a la versión final que se llega en esta tesina:

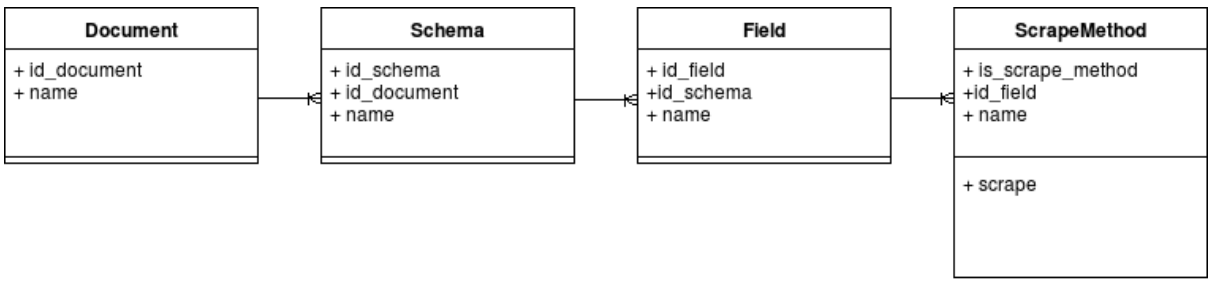


Figura 9. Ejemplo de UML extendiendo la herramienta a nuevos tipos de documentos.

En la figura 9 se expone como con simplemente agregar un nuevo objeto al modelo, este puede extenderse a distintos tipos de documentos a manera de ejemplo podrían ser: videos de conferencias, ponencias, clases; audios de distintas índoles; imágenes, etcétera.

También, podemos extender la vista, para que se puedan agregar a través de una interfaz nuevos documentos según se requiera.

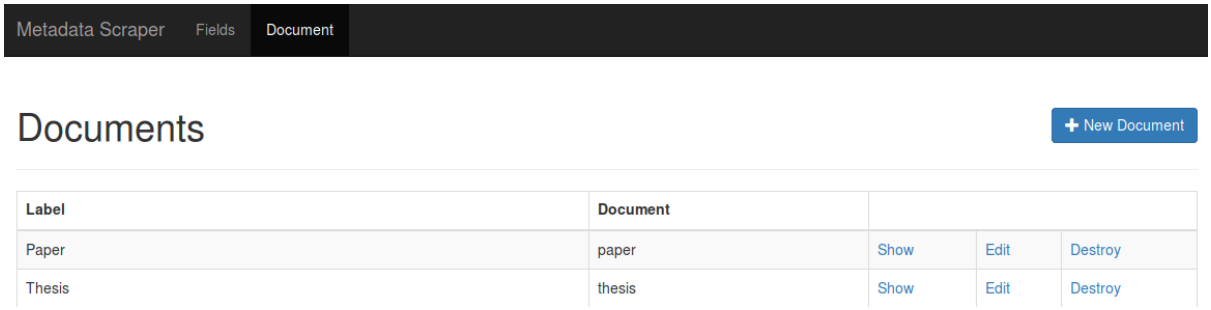


Imagen 9. Ejemplo de cómo podría ser una vista de un listado de documentos con opciones de alta, baja y borrado.

En segundo lugar, se pueden realizar con la herramienta es la de proveer una carga de información correcta de los datos. Con este servicio se pueden realizar correcciones a los recursos que se tienen depositados en el sistema de almacenamiento; se puede agregar información que antes no se tuvo en cuenta, o se pueden verificar los datos almacenados anteriormente. Esto podría realizarse con utilizando el JSON:

Imagen 10. Metadatos alojados en el repositorio SEDICI acerca de un artículo, comparativa con la extracción resultante de la herramienta.

En tercer lugar, capturar información de listados de links o listado de páginas web que contengan índices y subíndices dará un rápido crecimiento en la velocidad en la que se

cargan los datos. Un ejemplo podría ser la tabla de contenidos del Directory of Open Access Journals (DOAJ).

Esto en ruby es muy sencillo de hacer con la gema Nokogiri. Un ejemplo es la escritura de un script que realice el análisis gramatical buscando links y en en cada ingreso ejecutar la herramienta.

```
lista = doc.css('div.heat a').map { |link| link['href'] }
```

Ejemplo 5. Línea de código que realiza la búsqueda de URLs utilizando la herramienta Nokogiri

En cuarto lugar, se podría realizar un módulo de caché. Su trabajo sería el de actualizar los metadatos en el caso de que hubiese alguna modificación posterior a la extracción realizada.

Finalmente, se puede incrementar la velocidad de carga de datos en los sistemas de información académica mediante la recolección de información del usuario a partir del uso del servicio en los distintos sistemas de gestión de la información. Un ejemplo podría ser, autocompletar un sistema CRIS como SIGEVA a través de una extensión como se realizó para el formulario de SEDICI.

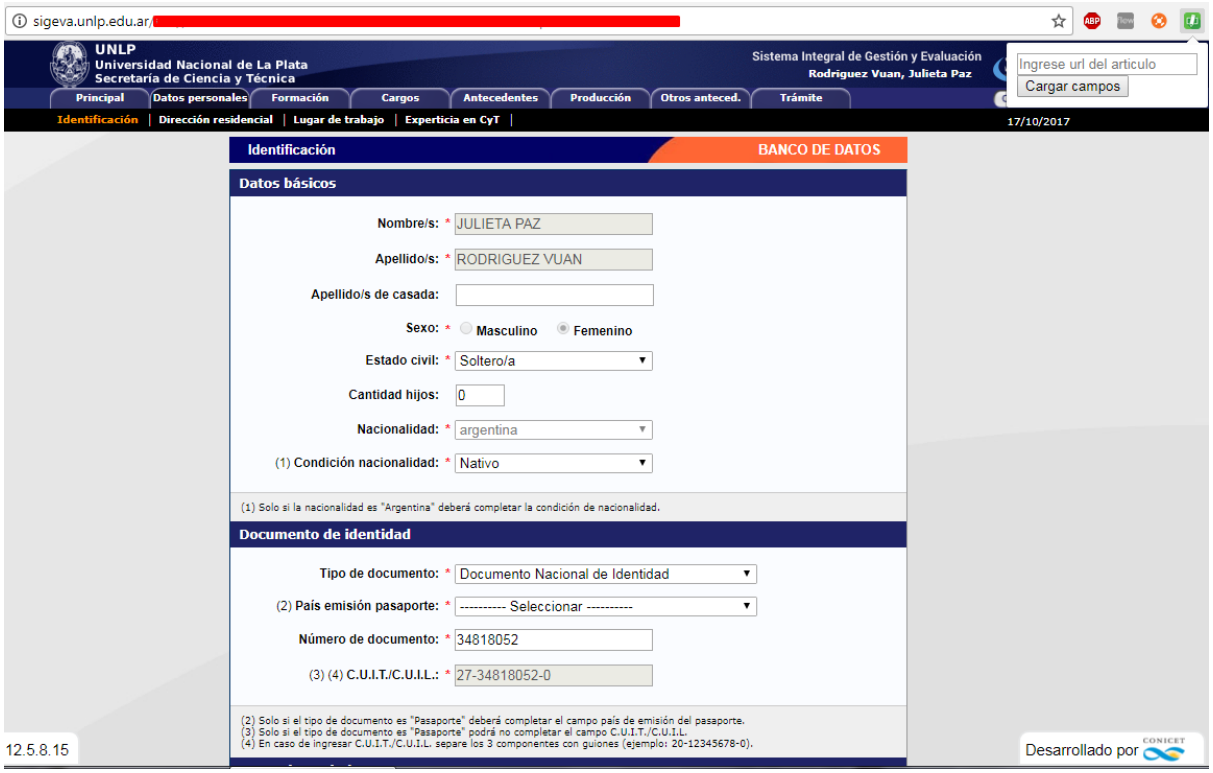


Imagen 10. Autocompletado de formulario SIGEVA a través de una extensión web

Capítulo 5 | Bibliografía

[ANSI/NISO01] NISO, 2001 "ANSI/NISO Z39.85- 2001, The Dublin Core Metadata Element Set."http://www.niso.org/apps/group_public/download.php/6577/z39-85-2001_dublin_core.pdf. Fecha de acceso 24 septiembre 2017.

[ApacheSolr17]The Apache Software Foundation, 2017 "Apache Solr-Apache Lucene" http://lucene.apache.org/solr/. Fecha de acceso 24 septiembre 2017.

[arXiv 17] Cornel University Library,2017 "arXiv." https://arxiv.org/. Fecha de acceso 24 septiembre 2017.

[base-search 17] Bielefeld Academic Search Engine, 2017 "Suchmaschine BASE." https://www.base-search.net/. Fecha de acceso 24 septiembre 2017.

[CIC-Digital 17] Comisión de investigaciones científicas de la Provincia de Buenos Aires, 2017 "¿Qué es CIC-Digital?." https://digital.cic.gba.gob.ar/page/que-es-cic-digital_es. Fecha de acceso 18 septiembre 2017.

[CIMI 17] Society of American Archivists, 2017 "Consortium for the Computer Interchange of Museum Information" https://www2.archivists.org/glossary/terms/c/consortium-for-the-computer-interchange-of-museum-information. Fecha de acceso 14 septiembre 2017.

[CNI13] Lynch C. A., 2003 "Institutional Repositories, Infrastructure for Scholarship" https://www.cni.org/publications/cliffs-pubs/institutional-repositories-infrastructure-for-scholarship. Fecha de acceso 14 septiembre 2017.

[Cogprints 17] Cognitive Sciences EPrint Archive, 2017 "Cogprints." http://cogprints.org/. Fecha de acceso 24 septiembre 2017.

[CONTENTdm 17] OCLC , 2017 "Build, showcase and preserve your digital collections." http://www.oclc.org/en/contentdm.html. Fecha de acceso 14 septiembre 2017.

[Crossref 17] Crossref , 2017 "Crossref" https://www.crossref.org/. Fecha de acceso 14 septiembre 2017.

[CSS 17]W3C, 2017 "CSS3 Introduction" https://www.w3schools.com/css/css3_intro.asp Fecha de acceso 14 septiembre 2017.

[DCMI 17] Dublin Core Metadata Initiative, 2017 "DCMI: Home." http://dublincore.org/. Fecha de acceso 24 septiembre 2017.

[DCMI Elements 17]Dublin Core Metadata Initiative, 2005 "DCMI: Using Dublin Core - Dublin Core Metadata Initiative." http://www.dublincore.org/documents/usageguide/elements/. Fecha de acceso 14 septiembre 2017.

[De Giusti 13] De Giusti M; Villarreal, G.; Terruzzi, A.; Lira, A. 2013 "Interoperabilidad entre el Repositorio Institucional y servicios en línea en la Universidad Nacional de La Plata", http://sedici.unlp.edu.ar/handle/10915/27406 Fecha de acceso 11 de octubre 2017

[De Giusti 14] De Giusti M.; Adorno, F; Lira, J., 2014 "Repositorios DSpace con múltiples contextos OAI-PMH" http://sedici.unlp.edu.ar/handle/10915/44572, Fecha de acceso 11 de octubre 2017

[De Giusti 14a] De Giusti, M. 2014, "Una metodología de evaluación de repositorios digitales para asegurar la preservación en el tiempo y el acceso a los contenidos", http://sedici.unlp.edu.ar/handle/10915/43157 Fecha de acceso 11 de octubre 2017

[De Giustib] De Giusti M.; Lira A., Villarreal, G.; Terruzzi A. “Interoperabilidad con el repositorio institucional” 2014 http://sedici.unlp.edu.ar/bitstream/handle/10915/34200/Presentaci%C3%B3n__diapositivas_.pdf?sequence=1 Fecha de acceso 11 de octubre 2017

[De Giusti 15] De Giusti, M.; “¿Por qué conviene construir un Repositorio Institucional?” 2013 http://sedici.unlp.edu.ar/handle/10915/31682 Fecha de acceso 11 de octubre 2017

[De Giusti 16] De Giusti, M.; Lira, A.; Rodríguez Vuan, J.; Villarreal, G.; 2016 “Accesibilidad de los contenidos en un repositorio institucional: análisis, herramientas y usos del formato EPUB” , http://sedici.unlp.edu.ar/handle/10915/53638 Fecha de acceso 11 octubre 2017

[DigitalCommons 17]Bepress, 2017 "Digital Commons" https://www.bepress.com/products/digital-commons/. Fecha de acceso 14 sep.. 2017.

[dLibra 17] PCSS, 2017 "Strona domowa systemów dArceo, dLab, dLibra oraz dMuseion." http://dlibra.psnc.pl/. Fecha de acceso 14 sep.. 2017.

[DOI 17]DOI Registration Agency, 2017 "Digital Object Identifier System." https://www.doi.org/. Fecha de acceso 14 septiembre 2017.

[DSpace 17]DSpace Organization, 2017 "DSpace institutional repository application.." http://www.dspace.org/. Fecha de acceso 14 septiembre 2017.

[DuraSpace 15] DuraSpace, 2015, “DuraSpace Wiki." https://wiki.duraspace.org/display/samvera/2015-05-22. Fecha de acceso 14 septiembre 2017.

[EDT17] Microsoft, 2017, "EdtRealMetadata Properties" https://msdn.microsoft.com/en-us/library/microsoft.dynamics.ax.framework.services.metadata.contracts.edtrealmetadata_properties.aspx. Fecha de acceso 14 septiembre 2017.

[Elsevier 17] Elsevier, 2017 "Elsevier: Zona de Lectura." http://www.elsevier.es/es. Fecha de acceso 14 septiembre 2017.

[EPrints 17] EPrints Organization, 2017, "EPrints Services." http://www.eprints.org/. Fecha de acceso 14 septiembre 2017.

[FreeBSD 17] Berkeley Software Distribution, 2013, "Qué es BSD - FreeBSD." https://www.freebsd.org/doc/es_ES.ISO8859-1/articles/explaining-bsd/article.html. Fecha de acceso 14 septiembre 2017.

[Google17] Google Support, 2017 "Tutorial: cómo crear una aplicación de Chrome - Google Support." https://support.google.com/chrome/a/answer/2714278?hl=es. Fecha de acceso 23 septiembre 2017.

[Handle.Net17] Corporation for National Research Initiatives, 2017 "Handle.Net Registry." https://www.handle.net/. Fecha de acceso 14 septiembre 2017.

[HTML-CSS 17] W3C, 2017 “HTML & CSS“ https://www.w3.org/standards/webdesign/htmlcss Fecha de acceso 14 septiembre 2017

[HTTP 17] W3C, 2006, "HTTP - Hypertext Transfer Protocol Overview." https://www.w3.org/Protocols/. Fecha de acceso 14 sep.. 2017.

[IFLA 17] **I**nternational **F**ederation of **L**ibrary **A**ssociations and Institutions, 2017 "Digital Libraries: Metadata Resources." https://www.ifla.org/node/9337. Fecha de acceso 14 septiembre 2017.

[ISBN17]International ISBN Agency, 2017, "International ISBN Agency." https://www.isbn-international.org/. Fecha de acceso 24 septiembre 2017.

[ISTEC 17] Ibero-American Science and Technology Education Consortium, 2017 "ISTEC" https://www.istec.org/. Fecha de acceso 24 septiembre 2017.

[JavaScript 17] JavaScript 2017, "JavaScript." https://www.javascript.com/. Fecha de acceso 22 septiembre 2017.

[JISC 17] Jisc, 2017, "Jisc." https://www.jisc.ac.uk/. Fecha de acceso 24 septiembre 2017.

[Lynch 16] Lynch C. “Coalition for Networked Information." 2016, https://www.cni.org/about-cni/staff/clifford-a-lynch. Fecha de acceso 24 septiembre 2017.

[Mendeley 17] Mendeley Web Importer, 2017, "Cite Websites with a Browser Plugin" https://www.mendeley.com/reference-management/web-importer. Fecha de acceso 19 septiembre 2017.

[Metadata MODS 17] Library of Congress, 2017, “Metadata Object Description Schema” http://www.loc.gov/standards/mods/ Fecha de acceso 19 septiembre 2017.

[Milliot 15] ABC Ciencia, 2015, "Estas cinco editoriales controlan más de la mitad de la publicaciones ." http://www.abc.es/ciencia/20150612/abci-control-publicaciones-cientificas-201506120943.html. Fecha de acceso 10 octubre 2017.

[MODS 17] Library of Congress, 2017, "Metadata Object Description Schema" http://www.loc.gov/standards/mods/. Fecha de acceso 14 septiembre 2017.

[MVC 17] Yii Framework, 2017, "Fundamentals: Best MVC Practices" http://www.yiiframework.com/doc/guide/1.1/en/basics.best-practices. Fecha de acceso 23 septiembre 2017.

[ncst 17] Networked Computer Science Technical Reference Library, 2017 ,"nsctrl." http://www.ncstrl.org/. Fecha de acceso 24 septiembre 2017.

[ndltd 17] Networked Digital Library of Theses and Dissertations, 2017 "ndltd." http://www.ndltd.org/. Fecha de acceso 24 septiembre 2017.

[OAI-PMH 17] Open Archives Initiative Protocol for Metadata Harvesting, 2017, “OAI-PMH” https://www.openarchives.org/pmh/ Fecha de acceso 24 septiembre 2017

[OCLC 17] Online Computer Library Center, 2017, "Research - OCLC.org." http://www.oclc.org/research.html. Fecha de acceso 24 septiembre 2017.

[OJS 17] Public Knowledge Project, 2017, "Open Journal Systems." https://pkp.sfu.ca/ojs/. Fecha de acceso 14 septiembre 2017.

[OJ Stats 17] Public Knowledge Project, 2017, "OJS Stats" https://pkp.sfu.ca/ojs/ojs-usage/ojs-stats/. Fecha de acceso 14 septiembre 2017.

[OPAC 17] Online Public Access Catalog, 2017 "National Archives." https://www.archives.gov/research/alic/tools/online-public-access-catalog.html. Fecha de acceso 24 septiembre 2017.

[Open Archive 17] Open Archives Initiative, 2017 "OAI-PMH - Open Archives Initiative." https://www.openarchives.org/pmh/. Fecha de acceso 14 septiembre 2017.

[OpenDOAR 17] OpenDOAR, 2017, "OpenDOAR." http://www.opendoar.org/. Fecha de acceso 14 septiembre 2017.

[OpenURL 17]Tanaka Akira, 2017, Module: OpenURL (Ruby 2.1.0) de https://ruby-doc.org/stdlib-2.1.0/libdoc/open-uri/rdoc/OpenURI.html Fecha de acceso 14 septiembre 2017.

[OPUS 17] Kooperativer Bibliotheksverbund Berlin, 2017, "OPUS 4 - Repository Software" http://www.kobv.de/entwicklung/software/opus-4/. Fecha de acceso 14 septiembre 2017.

[PKP17] Public Knowledge Project, 2017 "Public Knowledge Project." https://pkp.sfu.ca/. Fecha de acceso 14 septiembre 2017.

[Pinilla 14] Pinilla A., Gutiérrez M.; Ballejos L.; 2014 “Extracción automática de metadatos a partir de objetos de aprendizaje en un repositorio institucional” http://43jaiio.sadio.org.ar/proceedings/STS/640%20-%20Pinilla%20et%20al.pdf Fecha de acceso 11 de octubre 2017

[PortalUCR17] Portal de revistas académicas de la Universidad de Costa Rica, 2017

"Portal de revistas académicas" https://revistas.ucr.ac.cr/. Fecha de acceso 24 septiembre 2017.

[Portal UNLP 17] Portal de revistas de la UNLP, 2017, "Portal de revistas de la UNLP." https://revistas.unlp.edu.ar/. Fecha de acceso 24 septiembre 2017.

[Postgres 17] RubyGems.org, 2016, "pg, alojamiento de gemas para la comunidad." https://rubygems.org/gems/pg/versions/0.18.4?locale=es. Fecha de acceso 20 septiembre 2017.

[Puma 17] RubyGems.org, 2016,"puma, your community gem host." https://rubygems.org/gems/puma/versions/3.4.0. Fecha de acceso 20 septiembre 2017.

[Puma 17a] Puma, 2017, “PUMA VS. OTHER WEBSERVERS” 2017 http://puma.io Fecha de acceso 20 septiembre 2017.

[RePEc 17] Research Papers in Economics, 2017, "RePEc." http://repec.org/. Fecha de acceso 24 septiembre 2017.

[RFC 17] Internet Engineering Task Force, 2005,"RFC 3986 - Uniform Resource Identifier (URI): Generic." https://tools.ietf.org/html/rfc3986. Fecha de acceso 14 septiembre 2017.

[Ruby 17] Ruby, 2017, "Ruby Tutorial" http://rubytutorial.wikidot.com/introduccion. Fecha de acceso 19 septiembre 2017.

[SEDICI 17] Servicio de Difusión de la Creación Intelectual es el Repositorio Institucional de la Universidad Nacional de La Plata, 2017, "SEDICI" http://sedici.unlp.edu.ar/. Fecha de acceso 14 septiembre 2017.

[Senso 03]Senso, J..;Pinero R., 2003 "El concepto de metadato: algo más que descripción de recursos electrónicos" http://www.scielo.br/scielo.php?pid=S0100-19652003000200011&script=sci_abstract&tlng=es. Fecha de acceso 14 septiembre 2017.

[Simon 13]Simon CC, 2017, "El artículo científico" http://edicionesdigitales.info/Manual/Manual/defartcient.html. Fecha de acceso 14 septiembre 2017.

[SiteLab 16] Sitelabs, 2016, "Qué es el Web scraping? Introducción y herramientas " https://sitelabs.es/web-scraping-introduccion-y-herramientas/. Fecha de acceso 14 septiembre 2017.

[SNRD 17]Sistema Nacional de Repositorios Digitales, 2017, "Sistema Nacional de Repositorios Digitales" http://repositorios.mincyt.gob.ar/. Fecha de acceso 24 septiembre 2017.

[SOAP 00] Simple Object Access Protocol, 2000, "Simple Object Access Protocol (SOAP) https://www.w3.org/TR/2000/NOTE-SOAP-20000508/. Fecha de acceso 14 septiembre 2017.

[Springer 17]Springer, 2017, "Springer." http://www.springer.com/. Fecha de acceso 14 septiembre 2017.

[SRW7U17] Online Computer Library Center, 2017, "SRW/U" http://www.oclc.org/research/software/srw. Fecha de acceso 14 septiembre 2017.

[SWORD 17] Simple Web-service Offering Repository Deposit, 2017, "About SWORD" http://swordapp.org/about/. Fecha de acceso 14 septiembre 2017.

[TEI 17] Text Encoding Initiative, 2017, "TEI: Text Encoding Initiative." http://www.tei-c.org/. Fecha de acceso 14 septiembre 2017.

[Texier 12] Texier, J.; 2012, “El uso de repositorios y su importancia para la educación en Ingeniería” http://sedici.unlp.edu.ar/handle/10915/22943 Fecha de acceso 11 de octubre 2017

[Texier 13] Texier, J; De Giusti, M.; Lira, A, Oviedo, N.; Villarreal, G. 2013, "DSpace como herramienta para un repositorio de ... “, http://sedici.unlp.edu.ar/handle/10915/44555. Fecha de acceso 11 oct.. 2017.

[USDA 17] United States Department of Agriculture, 2017, "USDA/Current Research Information System." https://cris.nifa.usda.gov/. Fecha de acceso 10 oct.. 2017.

[Valle 05] Valle J., 2005 "Definición arquitectura cliente servidor" http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml. Fecha de acceso 29 septiembre 2017.

[Versione 13]Versione L., 2012, "Repertorio Bibliográfico Dekk'Educazione Linguistica in Italia", http://www.unive.it/media/allegato/centri/CRDL/BELI_Repertorio_1960-2012.pdf. Fecha de acceso 25 septiembre 2017.

[Villarreal 17] Villarreal G; García, D., 2017, "Taller de Revistas Académicas", http://sedici.unlp.edu.ar/handle/10915/61817, Fecha de acceso 11 septiembre 2017

[W3C17]W3C, 2017, "W3C HTML - World Wide Web Consortium." https://www.w3.org/html/. Fecha de acceso 14 septiembre 2017.

[Web Scraper 17] Google, 2016, "Web Scraper" https://chrome.google.com/webstore/detail/web-scraper/jnhgnonknehpejjnehehllkliplmbmhn. Fecha de acceso 19 septiembre 2017.

[WEKO17] "インストール - WEKO." http://weko.at.nii.ac.jp/index.php?page_id=17. Fecha de acceso 14 septiembre 2017.

[WSDL01]W3c, 2001, "Web Service Definition Language", https://www.w3.org/TR/wsdl. Fecha de acceso 14 septiembre 2017.

[XML 17] World Wide Web Consortium, 2016, "Extensible Markup Language (XML)" https://www.w3.org/XML/. Fecha de acceso 14 septiembre 2017.

[XPath17] W3C, 2017, "XPath Tutorial - W3Schools", https://www.w3schools.com/xml/xpath_intro.asp, Fecha de acceso 8 septiembre 2017,

[Z39.50 17] Library of Congress, " Z39.50", 2015 https://www.loc.gov/z3950/agency/. Fecha de acceso 14 sep.. 2017.