

# La visualización de autores en un Repositorio Institucional a través del enfoque Model Driven con WebRatio

Jose Texier

*Universidad Nacional Experimental del Táchira (UNET), Venezuela*

`jtexier@unet.edu.ve`

Marisa R. De Giusti, Gonzalo L. Villarreal y Ariel J. Lira

*Servicio de Difusión de la Creación Intelectual, Universidad Nacional de La Plata (SEDICI), Argentina*

`{marisa.degiusti, gonzalo, ariel}@sedici.unlp.edu.ar`

**Resumen:** los repositorios institucionales a través de sus plataformas de software tiene distintas funcionalidades que en varias ocasiones no son implementadas. En este trabajo, se propone una metodología de desarrollo de software para implementar cualquier funcionalidad en los repositorios usando la herramienta WebRatio. En esta ocasión se implementó la funcionalidad de gestión de la entidad autores en el repositorio SEDICI para todos los recursos de tipo “Tesis”. El enfoque propuesto es conocido como Model-Driven y permitió abordar un problema en concreto con una solución en una determinada tecnología, además de ayudar a que las partes involucradas en un sistema de repositorios (desarrolladores, dueños del negocio y expertos del dominio) hablaran un mismo idioma sin importar el software de repositorios, de manera tal que se pueda encontrar una solución a un problema planteado.

**Palabras clave:** repositorios institucionales, model-driven, webratio, SEDICI, autores

**Abstract:** the institutional repositories through their software platforms can offer many features that are not repeatedly implemented. This paper proposes a methodology developed software to implement any functionality in the repositories using WebRatio tool. For reasons of space, this work only proposes the implement of functionality of the authors management entity in SEDICI repository for all resources of type thesis. The proposed approach is known as Model-Driven and discussed a particular problem with a solution in a given technology, in addition to helping the parties involved in a repository system (developers, business owners and domain experts) spoke a same language to find a solution the problem.

**Keywords:** institutional repositories, model-driven, webratio, SEDICI, authors

## **Introducción**

Los trabajos científicos, académicos y administrativos de instituciones se recopilan, catalogan, acceden, gestionan, difunden y preservan en el tiempo a través de sistemas informáticos conocidos como repositorios institucionales, a partir de la enumeración de un conjunto de datos específicos de dichos trabajos conocidos como metadatos (Jose Texier, 2013), además, están relacionados con los ideales y objetivos del Acceso Abierto (Björk & Solomon, 2012; Suber, 2012). Estos RI se desarrollan a través de plataformas de software que permiten gestionar los diferentes servicios que prestan. Algunas instituciones utilizan plataformas preexistentes, entre ellas las más usadas son DSpace, EPrints y Digital Commons (OpenDOAR, 2013; ROAR, 2013), mientras que otras realizan desarrollos propios, sobre las cuales se gestiona un gran porcentaje de los recursos depositados en el mundo, a saber: arXiv, CiteSeerX, Social Science Research Network, entre otras (CSIC, 2013).

Todas estas plataformas de software deben ser diseñadas e implementadas a través de alguna metodología de desarrollo de software. En este contexto, existen los factores que determinan el diseño e implementación de componentes de software en un repositorio para buscar soluciones a sus problemas. Estos factores incluyen: la diversidad de soluciones tecnológicas, tratamiento de diferentes tipologías de recursos, esquemas de metadatos, recomendaciones de almacenamientos de recursos, la preservación de recursos, recomendaciones de modelos conceptuales o de datos como: OAIS, FRBR, modelo 5S, modelo DELOS, Europeana Data Model, entre otros.

Estos factores se pueden contemplar dentro de metodologías de desarrollo dirigidas por modelos para generar una aplicación web que solucione el problema planteado. Por tanto, el objetivo de este trabajo es visualizar y modificar los autores de los recursos de tipo “Tesis” en el repositorio institucional de la Universidad Nacional de La Plata (UNLP), SEDICI, para dar una solución a partir de una metodología de desarrollo dirigida por modelos, MDSE, que permita abordar el problema en los RI sin tener que comprender la plataforma de software, de esta manera los actores de este sistema de software (desarrolladores, dueños del negocio y expertos del dominio) podrán hablar un mismo idioma, sin importar la plataforma de software usada.

Este artículo está organizado de la siguiente manera: una exposición del problema presentado; luego el detalle de metodología de desarrollo de software dirigida por modelos para llevar a cabo la implementación de la propuesta; después se presenta la propuesta, el diseño, el análisis y la implementación realizada; y, finalmente, en la última parte se presentan las conclusiones y trabajos futuros.

## **El problema**

Los recursos, dentro del contexto de los RI, son objetos físicos o digitales que se describen a partir de la enumeración de un conjunto de datos específicos (metadatos) que lo distinguen entre otros objetos. Estos metadatos son representados a través de distintos modelos, esquemas, formatos o estándares, que si bien comparten una sintaxis y estructura de la información, por lo general en XML, difieren atendiendo a los propósitos de la información que describen (Méndez, 2003). La entidad abstracta autor, en SEDICI, es el metadato de los recursos que permite identificar el creador intelectual del recurso, ya sea una persona o una organización. El autor se

considera entidad abstracta por poseer información descriptiva propia, utilizada en los procesos de catalogación de recursos como elemento de un vocabulario controlado (José Texier, De Giusti, Oviedo, Lira, & Villarreal, 2013). Ante esto, se entiende la importancia de poder identificar el autor de un recurso para identificarlo, a través de búsquedas o navegación en la plataforma de software para diferentes fines.

Los autores son catalogados de diferentes maneras en los software de repositorios, ya que existen diversas formas de normalizar dichos autores para el funcionamiento e interoperabilidad de los RI en el mundo. Por ello, es necesario poder tener un control completo de la entidad autores de manera que cualquier plataforma de software pueda gestionar la normalización de la forma que desee, para esto, es recomendable que un autor tenga un registro lo más completo posible, es decir, entre más datos y nombres completos tenga, será mejor para la gestión del RI.

WebRatio es una herramienta soportada por una metodología de desarrollo dirigida por modelos, permite visualizar y modificar los autores del Servicio de Difusión de la Creación Intelectual (SEDICI, 2013) y crear las funciones como el ABM (alta, baja y modificación) para dichos autores. SEDICI, usa como plataforma de software DSpace, la cual es una plataforma de código abierto con una arquitectura de tres capas (negocio, aplicación y almacenamiento) para la gestión de recursos digitales que se organizan a partir de comunidades jerárquicas y colecciones (DSpace, 2013). DSpace actualmente ofrece dos frontend, XMLUI y JSPUI, y deja abierta la gestión de la entidad autores en forma particular. SEDICI, decidió gestionar la entidad autores de la misma forma como la realizaba en la plataforma de software de la que migró en el 2012 y que esta basada en una base de datos MySQL. Entonces, WebRatio, brinda la posibilidad de presentar la propuesta para relacionar DSpace-SEDICI con la base de datos de autores en MySQL.

## **Metodología de desarrollo de software**

### ***MDSE***

La comunidad de desarrolladores de software ha comenzado a tomar conciencia de la necesidad de formalizar la construcción de componentes de software y guiar tales desarrollos a través de modelos definidos como un conjunto de elementos que sirven para demostrar la consistencia de una teoría, es decir, representan con detalle un sistema dado (Whittle, Clark, & Kühne, 2011). Este enfoque es conocido como Model-Driven, del cual nace el Model Driven Software Engineering (MDSE), que se define como una metodología de desarrollo de software para aplicar las ventajas del modelado en actividades de ingeniería de software (Brambilla, Cabot, & Wimmer, 2012). Esta metodología comprende los siguientes aspectos (Brambilla et al., 2012):

- **Conceptos:** componentes sobre los que se construyen la metodología. En este caso, los conceptos core de MDSE son: los modelos y las transformaciones.
- **Notación:** la forma en que se representan los conceptos, es decir, los lenguajes. En MDSE son llamados lenguajes de modelado, así como los programas computacionales necesitan los lenguajes de programación.
- **Procesos y reglas:** las actividades que llevan a generar un producto final, las reglas para su coordinación y control, y las aserciones sobre las propiedades deseadas de los productos o del proceso.
- **Herramientas:** aplicaciones que facilitan la ejecución de actividades o de su coordinación. En este caso serán los Entornos de Desarrollo Integrados o IDEs (Integrated

Development Environments) para la definición de los modelos y las transformaciones, y los compiladores o intérpretes para ejecutarlos y producir los artefactos de software finales.

Para comprender el mundo del Model-Driven, en la Figura 1 se observa los diferentes acrónimos que se usan como referencia en el área que pueden parecer sinónimos básicos y oscuros (Brambilla et al., 2012), pero cada uno de ellos representa una sub-área específica que se detalla a continuación:

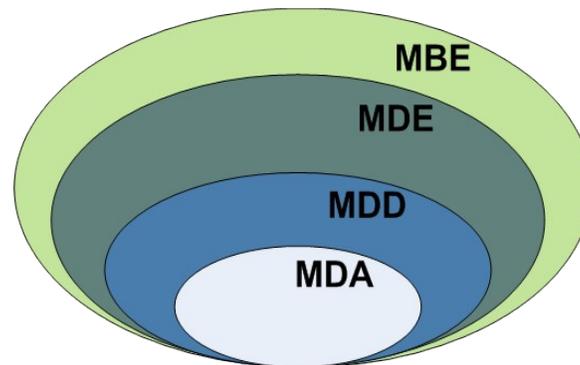


Figura 1. Relaciones entre los diferentes acrónimos del Model Driven star. (Brambilla, 2012)

- MBE (Model-Based Engineering): Tom Demarco en 1979 introdujo este concepto, que se destacó por la construcción de un sistema de software precedida por la construcción de un modelo, tal y como se realiza en otros sistemas ingenieriles.
- MDD (Model-Driven Development): paradigma de construcción de software cuyas motivaciones principales son la independencia de los productores a través de estandarizaciones y la portabilidad de los sistemas (Pons, Giandini, & Pérez, 2010). El objetivo de MDD es separar el diseño del sistema de la arquitectura de las tecnologías, para que puedan ser modificados independientemente.
- MDA (Model-Driven Architecture): propuesta de MDD definida por Object Management Group (OMG). En ocasiones, el término MDA se intercambia con el de MDD, ya que MDA se refiere a las actividades que llevan a cabo los desarrolladores, mientras que MDD se enfoca en su definición formal (Navarro, Cristóbal, Fernández-Chamizo, & Fernández-Valmayor, 2011).
- MDE (Model-Driven Engineering): todas las variantes de “Model Driven Lo-que-sea” (Model Driven Whatever) son frecuentemente referidas al acrónimo MD\* (Model Driven star), a pesar de la enorme variedad de acrónimos que se pueden encontrar en la literatura, por ejemplo: MDSE (Model Driven Software Engineering), MDPE (Model Driven Product Engineering), MDWE (Model Driven Web Engineering), etc. Entonces, MDE puede ser visto como el superconjunto de todas estas variantes, por ello, cualquier enfoque MD\*E podría caer bajo el paraguas MDE.

MDSE se puede aplicar en diferentes niveles de abstracción para poder proporcionar una visión integral en el desarrollo del sistema de software. En esta metodología los modelos pueden transformarse en otros modelos sucesivamente hasta obtener una representación final (artefactos de software o código fuente), gracias a la definición de un lenguaje que proporciona la

posibilidad de describirlos adecuadamente (Figura 2). Este lenguaje está a su vez definido a partir de un metalenguaje, lo cual genera la idea de sucesivas capas recursivas (Navarro et al., 2011).

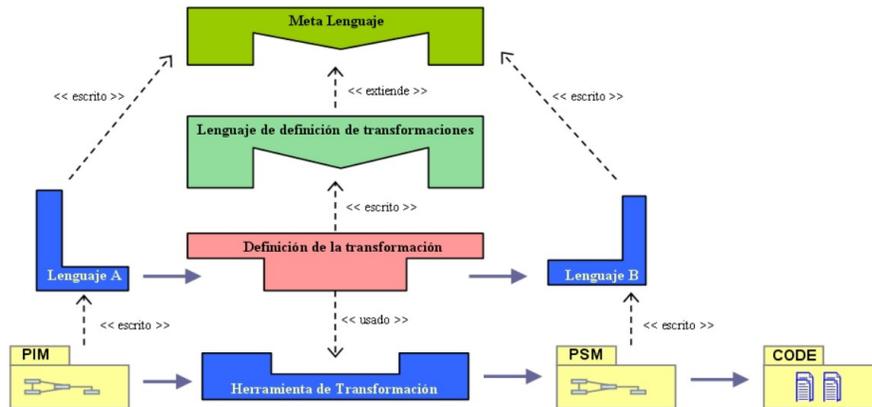


Figura 2. Esquema de la Transformación. (Kleppe, Warmer, & Bast, 2003)

## Ingeniería Web

Es una disciplina que propone extensiones y adecuaciones a los métodos y modelos tradicionales, incluye un proceso que permite crear, implementar y mantener aplicaciones Web de alta calidad. El desarrollo de este tipo de aplicaciones se ha incrementado con el auge de Internet, volviéndose un desafío para los ingenieros de software (Pressman, 2002). A raíz de esto se crearon enfoques y metodologías tomando en cuenta el aporte tecnológico. Por ello, nace el modelado conceptual de aplicaciones Web, más específicamente el Model Driven Web Engineering (MDWE), que se encuentra dentro del contexto MDE (Brambilla et al., 2012). En MDWE se pueden identificar 5 grandes grupos (Liddle, 2011):

- *Enfoques orientados a los datos:* tales como RMM, WebML y Hera. Tienen su origen en sistemas de base de datos y se centran en aplicaciones web de datos intensivos.
- *Métodos orientados al hipertexto:* tales como HDM, HDM-lite, WSDM y W2000, originados de trabajos en diseños de hipermedia y manejo de la naturaleza hipertextual de las aplicaciones web.
- *Enfoques orientados a objeto:* siguen la tradición del modelo orientado a objetos y se incluyen los métodos OOHDM, UWE, OOWS y OO-H.
- *Métodos orientados al software:* toman el enfoque tradicional de desarrollo de software. Web Application Extension (WAE) y su extensión WAE2 ejemplifican este enfoque.
- *Métodos orientados al MDE:* explícitamente tomado del enfoque model-driven para el desarrollo de aplicaciones Web y hacen hincapié en la generación de código fuente a partir de modelos de aplicaciones Web. Ejemplos de esta categoría incluye Wobile, WebSA, MIDAS y Netsilon.

## WebML

Es una metodología de modelado Web orientada a datos (Acerbis et al., 2008; Ceri, Fraternali, & Bongio, 2000), que se encuentra dentro de la definición MDWE y corresponde a MDE. En esta metodología se definen 4 modelos que acompañan al proceso del desarrollo del software: datos, hipertexto, gestión de contenidos e hipertexto avanzado (Ceri et al., 2003; Ceri, Fraternali,

& Matera, 2002). Este trabajo se enfocará solo en los modelos de datos y de hipertexto en WebRatio.

### **WebRatio**

El desarrollo de aplicaciones con base en WebML es asistido por la herramienta WebRatio, la cual es una herramienta comercial (con una licencia para desarrollo personal) para el diseño y la implementación de aplicaciones web, y está soportado por el IDE Eclipse (WebRatio, 2013). Los requisitos se expresan a través de un modelo de alto nivel y el código de la aplicación se genera automáticamente (mediante el uso de reglas). El código resultante es una aplicación Java basada en estándares Web, y consta de tres pasos básicos: construcción de los modelos (en particular, el modelo de la aplicación se realiza utilizando WebML), personalización de las reglas y generación de la aplicación (Brambilla & Fraternali, 2013). WebRatio también soporta los estándares BPM y el más reciente IFML aprobado por la OMG.

La arquitectura de WebRatio (Figura 3) consiste en dos capas: una capa de diseño que provee funciones para la edición visual de las especificaciones y una capa de ejecución (run-time) que implementa los servicios básicos para la ejecución de las unidades WebML sobre un framework de aplicaciones Web estándar (Brambilla, Comai, Fraternali, & Matera, 2008).

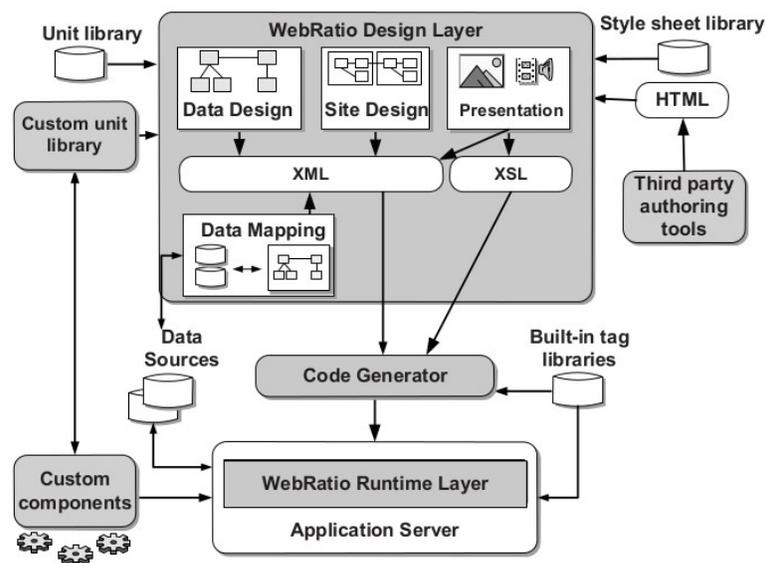


Figura 3. Arquitectura WebRatio. Fuente: Brambilla, 2012

La capa de diseño, generador de código y la capa de ejecución tienen una arquitectura de plugin: los nuevos componentes de software pueden ser acoplados con descriptores XML y hacerlos disponibles para la capa de diseño como las unidades WebML personalizadas, el generador de código se puede ampliar con las reglas XSL adicionales para producir el código necesario para unirlos con los componentes definidos por el usuario y los propios componentes pueden ser desplegados en el marco de la aplicación en tiempo de ejecución.

## Desarrollo de la propuesta

El ideal para los actores involucrados (desarrolladores, dueños del negocio y expertos del dominio) en el diseño y desarrollo de un sistema para repositorios, es que exista un lenguaje neutral y de alto nivel que permita que las partes se pongan de acuerdo y que les sirva de apoyo para describir, discutir y negociar las funciones (destinados a recopilar, catalogar, almacenar, gestionar, acceder, difundir y preservar) que el repositorio debe ofrecer. El enfoque *Model-Driven* brinda el marco que permite a los interesados compartir sus puntos de vista y manipular directamente las representaciones de las entidades de este dominio. Además, este paradigma ofrece algunas ventajas como: incremento en la productividad (errores, costos, código), adaptación a cambios tecnológicos, reuso de software, mejora en la comunicación con usuarios y desarrolladores, asignación de roles, entre otros (Pons et al., 2010). Por ello, a partir de un repositorio funcionalmente activo (por ejemplo SEDICI) y siguiendo la metodología MDSE, desde la herramienta WebRatio, se pueden desarrollar algunas de las siguientes funcionalidades:

- visualizar de los elementos de una representación de recursos del repositorio,
- interoperar entre las plataformas de software,
- evaluar correlación de entidades con cualquier esquema de metadatos,
- analizar los metadatos que garantizan la preservación de los recursos,
- almacenamiento de las entidades en cualquier tipo de base de datos,
- poder establecer métricas relacionadas a entidades deseada (p.e.: descargas, visitas, búsquedas),
- lograr sistemas científicos más inteligentes,
- mantener coherencia con los conceptos de la norma ISO 14721, que ofrece una recomendación a la administración de documentos para poder especificar la representación de un recurso dentro de un RI,
- realizar tareas de curation de forma rápida y sencilla de un RI.

Estas funcionalidades pueden implementarse a partir de un enfoque MDSE con independencia tecnológica, es decir, puede adaptarse a diferentes plataformas de software. Por razones de espacio, este trabajo se centró solamente en la visualización y modificación de la entidad *Autor* en SEDICI para los recursos de tipo *tesis* (grado, especialización, maestría y doctorado) pertenecientes a la UNLP. Por tanto, se crearon las siguientes funciones con componentes de WebRatio para la entidad autores:

- Funciones básicas de crear, leer, actualizar y borrar (ABM).
- Exportar e importar registros en archivos XML, a partir de esquemas de definición en XML (XSD).
- Modificación por lotes de la entidad autores, a través de consultas SQL.
- Búsquedas gracias a sentencias SQL.
- Control de acceso de los usuarios para modificar, gracias al soporte de seguridad que ofrece WebRatio.

### **Análisis**

A partir de WebRatio, se diseñaron los diferentes modelos involucrados en el proceso, que se compartieron con los diferentes actores de SEDICI, quien cuenta con aproximadamente 30.000 recursos digitales (en noviembre de 2013) que han sido catalogados uno por uno por expertos en el área desde el 2003. El desarrollo actual de SEDICI se encuentra en DSpace pero la base de datos de los autores se encuentra en MySQL debido a que la versión anterior del repositorio de la

UNLP era un desarrollo propio (Celsius DL) que se migró a DSpace en el 2012, aprovechando así el soporte que brinda DSpace para un control de autoridades abierto. Por tanto, se tienen dos bases de datos en WebRatio, una en PostgreSQL (DSpace) y una en MySQL (Celsius DL).

### ***Diseño e implementación***

WebRatio, exige dos modelos básicos para generar la aplicación Web que se desea: un modelo de datos y un modelo de hipertexto, pero antes de ellos, es conveniente realizar un diseño de cómo se quiere el frontend de la aplicación, por ello se diseñaron 3 mockups correspondientes a la visualización de los autores, tesis y ABM de los autores. Por cuestiones de espacio solamente se mostrará el mockup de autores (Figura 4).

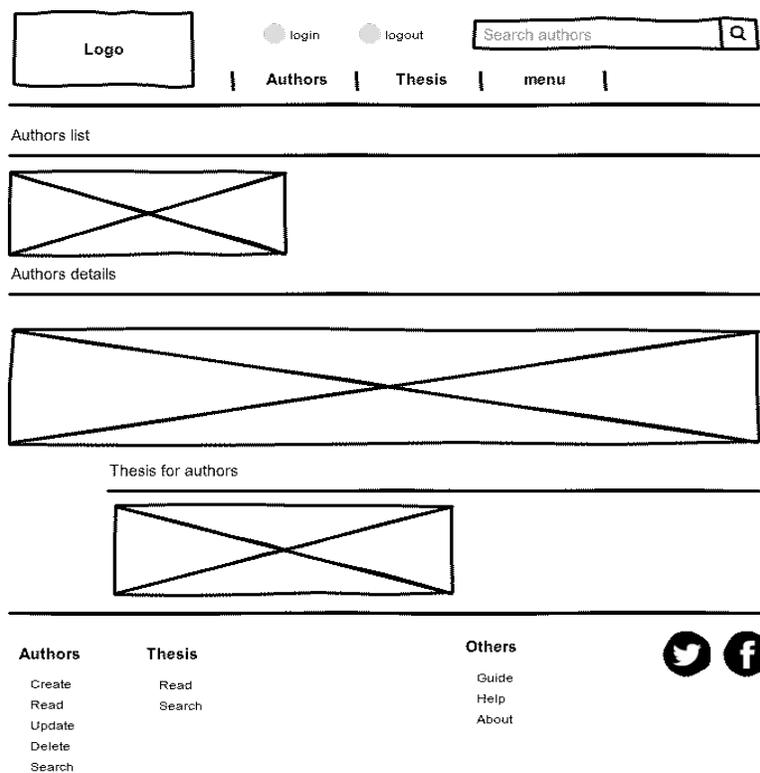


Figura 4. Mockup de autores

En cuanto al modelo de datos, se importaron las bases de datos de DSpace-SEDICI y SEDICI-OLD (Celsius DL) que contiene los registros de los 30.000 recursos y 19.000 autores y usuarios, respectivamente. A continuación se muestra una parte del modelo de hipertexto desarrollado (Figura 5) conocido en WebRatio como *SiteView*, en caso de querer profundizar en los modelos y el proyecto WebRatio, pueden visitar el proyecto GitHub (Jose Texier, 2014).

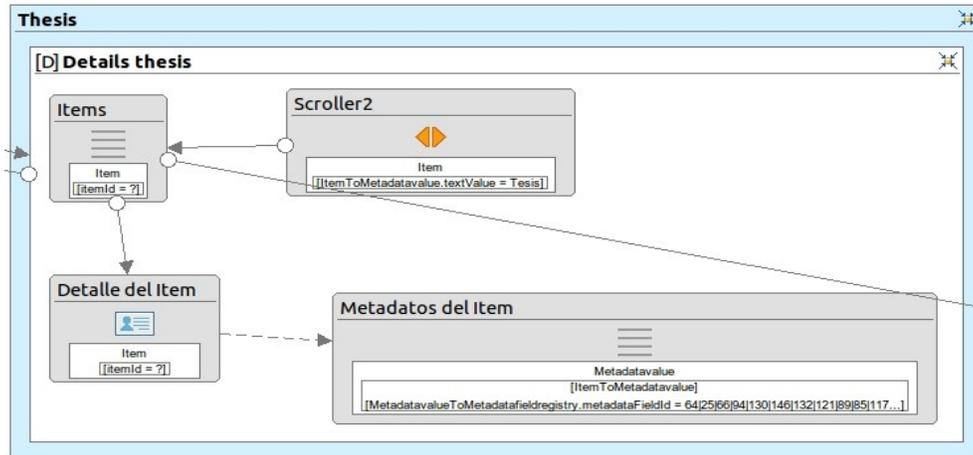


Figura 5. Modelo hipertexto

Luego de la sincronización de las bases de datos a través del modelo de datos y de hipertexto (SiteView), el desarrollador del sistema solamente debe dar clic al botón de generación de código y levantar el servidor de aplicaciones Web. Luego, cualquier usuario dentro del contexto de la red donde se encuentra montada la aplicación, despliega el browser Web donde se observa la aplicación elaborada (ver Figura 6):

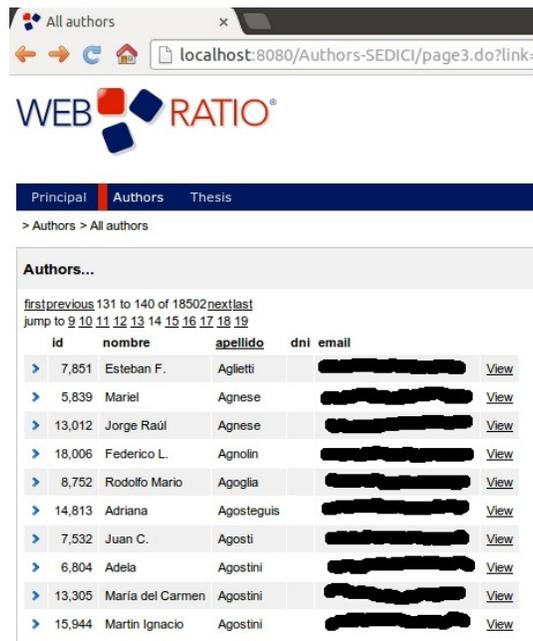


Figura 6. Site Web generado automáticamente

## Conclusiones

- La investigación realizada permitió cumplir con el propósito de poder visualizar y modificar la entidad autores en SEDICI a través del enfoque Model Driven, es decir, un

problema en concreto solucionado con una determinada tecnología. Este enfoque ayuda a que las partes involucradas en un sistema de repositorios (desarrolladores, dueños del negocio y expertos del dominio) hablan un mismo idioma sin importar la plataforma de software de RI a través de la elaboración de modelos de datos e hipertexto que provee la herramienta WebRatio. Por ello, la metodología desarrollada puede ser aplicable sobre cualquier repositorio, porque en WebRatio se puede sincronizar y modificar el esquema de la base de datos del repositorio y permitir el diseño del modelo de hipertexto que después genera la aplicación Web.

- Se observó el potencial que puede tener WebRatio en los repositorios institucionales, ya que se pueden implementar diversas funcionalidades sin importar la arquitectura de software en la que se encuentran el repositorio.
- Una de las ventajas de MDSE, es que se cuenta con un entorno de desarrollo integrado sobre la base de Eclipse y soportada por la metodología WebML, ya que los paradigmas de construcción de software necesitan contar con herramientas de software amenas y que estén en sintonía con las nuevas tecnologías. De igual manera, WebRatio cuenta con una página Web que orienta su uso, un foro para consultas, una wiki con los principales problemas categorizados por nivel de dificultad y un canal de youtube.
- La propuesta se enfocó en la plataforma de software DSpace en SEDICI como campo de pruebas, pero esta propuesta puede aplicarse a otros RI con otras plataformas de software, gracias a la metodología de desarrollo de software dirigida por modelos.

## Trabajos futuros

- Aplicar la metodología desarrollada en plataformas de RI reconocidas y realizar un análisis comparativo que el trabajo realizado.
- Realizar un estudio de las diferentes alternativas de herramientas de desarrollo de aplicaciones Web bajo el enfoque MDE.
- Implementar otras funcionalidades que permitan tener un criterio más certero del alcance real de WebRatio, así como también estudiar la integración de dichas funcionalidades con los estándares BPM y IFML que se encuentran en WebRatio.

## Bibliografía

- Acerbis, R., Bongio, A., Brambilla, M., Butti, S., Ceri, S., & Fraternali, P. (2008). Web Applications Design and Development with WebML and WebRatio 5.0. In R. F. Paige & B. Meyer (Eds.), *Objects, Components, Models and Patterns* (pp. 392–411). Springer Berlin Heidelberg. Retrieved from [http://link.springer.com/chapter/10.1007/978-3-540-69824-1\\_22](http://link.springer.com/chapter/10.1007/978-3-540-69824-1_22)
- Björk, B.-C., & Solomon, D. (2012). Open access versus subscription journals: a comparison of scientific impact. *BMC Medicine*, *10*(1), 73. doi:10.1186/1741-7015-10-73
- Brambilla, M., Cabot, J., & Wimmer, M. (2012). *Model-Driven Software Engineering in Practice*. Morgan & Claypool.
- Brambilla, M., Comai, S., Fraternali, P., & Matera, M. (2008). Designing Web Applications with Webml and Webratio. In G. Rossi, O. Pastor, D. Schwabe, & L. Olsina (Eds.), *Web Engineering: Modelling and Implementing Web Applications* (pp. 221–261). Springer London. Retrieved from [http://link.springer.com/chapter/10.1007/978-1-84628-923-1\\_9](http://link.springer.com/chapter/10.1007/978-1-84628-923-1_9)

- Brambilla, M., & Fraternali, P. (2013). Large-scale Model-Driven Engineering of web user interaction: The WebML and WebRatio experience. *Science of Computer Programming*. doi:10.1016/j.scico.2013.03.010
- Ceri, S., Fraternali, P., & Bongio, A. (2000). Web Modeling Language (WebML): a modeling language for designing Web sites. *Computer Networks*, 33(1–6), 137–157. doi:10.1016/S1389-1286(00)00040-2
- Ceri, S., Fraternali, P., Bongio, A., Brambilla, M., Comai, S., & Matera, M. (2003). *Designing Data-Intensive Web Applications*. Morgan Kaufmann.
- Ceri, S., Fraternali, P., & Matera, M. (2002). Conceptual modeling of data-intensive Web applications. *IEEE Internet Computing*, 6(4), 20–30. doi:10.1109/MIC.2002.1020321
- CSIC. (2013). Ranking Web of Repositories. Retrieved March 8, 2013, from <http://repositories.webometrics.info/>
- DSpace. (2013). DSpace. Retrieved April 12, 2013, from <http://www.dspace.org/>
- Kleppe, A., Warmer, J., & Bast, W. (2003). *MDA Explained: The Model Driven Architecture(TM): Practice and Promise*. Addison-Wesley Professional. Retrieved from <http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/032119442X>
- Liddle, S. W. (2011). Model-Driven Software Development. In D. W. Embley & B. Thalheim (Eds.), *Handbook of Conceptual Modeling* (pp. 17–54). Springer Berlin Heidelberg. Retrieved from [http://link.springer.com/chapter/10.1007/978-3-642-15865-0\\_2](http://link.springer.com/chapter/10.1007/978-3-642-15865-0_2)
- Méndez, E. (2003). Tratamiento de los objetos de información en los archivos: retos y estándares para la descripción basada en metadatos. Book Chapter. Retrieved October 13, 2013, from <http://eprints.rclis.org/handle/10760/12691#.UAAZFuzfgM>
- Navarro, A., Cristóbal, J., Fernández-Chamizo, C., & Fernández-Valmayor, A. (2011). Architecture of a multiplatform virtual campus. *Software: Practice and Experience*. doi:10.1002/spe.1130
- OpenDOAR. (2013). OpenDOAR - Home Page - Directory of Open Access Repositories. Retrieved October 21, 2013, from <http://www.opendoar.org/>
- Pons, C., Giandini, R., & Pérez, G. (2010). *Desarrollo de Software Dirigido por Modelos*. La Plata, Argentina: Mc Graw Hill.
- Pressman, R. (2002). *Ingeniería del Software* (Quinta edición.). Mc Graw Hill.
- ROAR. (2013). Registry of Open Access Repositories (ROAR). Retrieved October 21, 2013, from <http://roar.eprints.org/>
- SEDICI. (2013). SEDICI - Repositorio de la Universidad Nacional de La Plata. Retrieved October 14, 2012, from <http://sedici.unlp.edu.ar/>
- Suber, P. (2012). Ensuring open access for publicly funded research. *BMJ: British Medical Journal*, 345. doi:10.1136/bmj.e5184
- Texier, J. (2013). Los repositorios institucionales y las bibliotecas digitales: una somera revisión bibliográfica y su relación en la educación superior (p. 9). Presented at the 11th Latin American and Caribbean Conference for Engineering and Technology - 2013, Cancun, Mexico: LACCEI. Retrieved from <http://eprints.rclis.org/19925/>
- Texier, J. (2014). WebRatio · GitHub. Retrieved January 20, 2014, from <https://github.com/dantexier/WebRatio>
- Texier, J., De Giusti, M. R., Oviedo, N., Lira, A. J., & Villarreal, G. L. (2013). La representación de recursos en los repositorios institucionales. El caso de estudio: SEDICI. Presented at the III Conferencia de Bibliotecas y Repositorios Digitales de América Latina (BIREDIAL) y VIII Simposio Internacional de Bibliotecas Digitales (SIBD) (Costa Rica, 2013). Retrieved from <http://hdl.handle.net/10915/30111>

WebRatio. (2013). WebRatio. Retrieved January 20, 2014, from <http://www.webratio.com/portal/content/en/home>

Whittle, J., Clark, T., & Kühne, T. (Eds.). (2011). *Model Driven Engineering Languages and Systems*. Wellington, New Zealand,: MODELS 2011. Retrieved from <http://www.springerlink.com/content/978-3-642-24484-1#section=971719&page=1>