ARQUITECTURA DE BÚSQUEDA PARA REPOSITORIOS DE OBJETOS DE APRENDIZAJE

SEARCH ARCHITECTURE FOR REPOSITORY OF LEARNING OBJECTS

Yosvanys Aponte Báez[1], Rafael Oliva Santos[2]
[1]Departamento de Informática, Universidad Agraria de La Habana, Cuba
[2]Facultad de Matemática y Computación, Universidad de La Habana, Cuba
E-mail: yaponte@isch.edu.cu

(*Enviado Julio 03, 2013; Aceptado Agosto 14, 2013*)

Resumen
Con el surgimiento de los Repositorios de Objetos de Aprendizaje, y para un uso más eficiente de los mismos, se han desarrollado diferentes motores de búsqueda que permitan a los usuarios encontrar fácil y efectivamente los materiales que necesiten. Existen muchos motores que son basados en palabras claves, pero estos limitan el área de búsqueda, algunos basados en ontologías y otros como el que se utilizó, basado en los metadatos que describen a los objetos de aprendizaje. Un motor muy potente es SOLR, éste se basa en la biblioteca LUCENE, el cual se ejecuta como una aplicación Web en Java, permitiendo que se puedan enviar los documentos a indexar vía HTTP y consultarlos utilizando peticiones HTTP GET. Los experimentos que se realizaron con la herramienta implementada basada en este motor, demuestran la eficacia y la calidad de los resultados con respecto a los demás motores.

Palabras Clave: *ROA, SOLR, XML, Indexación.*

Abstract
With the emergence of Repositories of Learning Objects, and for a more efficient use of them, different search engines have been developed that allow users to easily and effectively find the materials they need. There are many engines that are based on keywords, but these limit the search area, some based on ontologies and others like the one used, based on the metadata that describe learning objects. A very powerful engine is SOLR, this is based on the LUCENE library, which runs as a Web application in Java, allowing documents to be sent to be indexed via HTTP and consulted using HTTP GET requests. The experiments carried out with the tool implemented based on this engine demonstrate the efficiency and quality of the results with respect to the other engines.

Keywords: *ROA, SOLR, XML, Indexing.*

1 INTRODUCCIÓN

Existen en la actualidad varias definiciones de Repositorios de Objetos de Aprendizaje (ROA), comúnmente los ROA se definen como un tipo de bibliotecas digitales especializadas en recursos educativos que utilizan los estándares de metadatos propuestos para el proceso de *e-learning*. Estas bibliotecas están preparadas tecnológicamente para interoperar entre ellas y con otras aplicaciones de los entornos *e-learning*. También como describe [1] los ROA juegan un papel decisivo en el desarrollo de los sistemas de educación modernos, ya que permiten el intercambio y reutilización de los materiales educativos que se generan en una institución o los que están distribuidos geográficamente, lo que trae como consecuencia una disminución en los costos producción de cursos en las distintas modalidades.

El desarrollo y popularización cada vez mayor de los ROA, trae consigo la necesidad de crear motores de búsqueda con mejores desempeños que permitan a los usuarios encontrar fácil y efectivamente los materiales que necesiten. Al escoger un motor de búsqueda para cualquier Repositorio se debe tener en cuenta [2]:

- Eficacia: Aspecto muy relacionado con el desarrollo de la interfaz. La herramienta debe ser capaz de soportar consultas complejas y además que sea de fácil uso para usuarios de poca experiencia.
- Velocidad: Los usuarios deben obtener resultados de las búsquedas realizadas en el menor tiempo posible.
- Expansión de consultas: Consiste en mostrarle al usuario variaciones en las consultas realizadas, puede realizarse después de enviada la misma como posibles sugerencias.
- Recuperación efectiva: Aspecto ampliamente estudiado que considera como medidas fundamentales la precisión y la cobertura.

En este trabajo proponemos una arquitectura de búsqueda especializada para Repositorios de Aprendizaje que puede ser de gran utilidad para la labor metodológica de los docentes. La arquitectura que proponemos se basa en el motor SOLR.

El resto del trabajo se estructura en otros cinco apartados numerados y un apartado para presentar las referencias. En el apartado 2 se exponen los principales conceptos y trabajos relacionados con la presente investigación, en el 3 se presenta la arquitectura de búsqueda para ROA. En el 4 se muestran los resultados obtenidos y finalmente en el 5 se exponen las conclusiones.

2 CONCEPTOS Y TRABAJOS RELACIONADOS

Para la descripción de los Objetos de Aprendizaje (OAs) se han desarrollado varios estándares, uno de ellos es el IEEE *Learning Object Metadata* (LOM) [3] del que parten importantes iniciativas para la estandarización del *e-learning*. En LOM se especifica la sintaxis y la semántica de los atributos necesarios para describir los OAs. Este estándar está compuesto por nueve categorías de metadatos, que agrupan elementos con los que se ha pretendido una descripción completa de los recursos educativos. En nuestro caso de estudio centramos la atención en *Virtual Meta Data* (VMD). VMD no es un estándar, sino un sistema de metadatos genérico que agrupa los diferentes estándares existentes más empleados (IMS-MD, LOM, SCORM, IMS-LD, CanCore, Dublin Core). La utilización de VMD permite la importación de OAs descritos en cualquier formato o incorporación de nuevos objetos, así como la exportación de OAs descritos en cualquier marco de los estándares mencionados [4].

Con el uso de diferentes estándares de metadatos se busca, además de la organización, la reutilización de recursos y la interoperabilidad entre los sistemas involucrados con el uso de contenidos. Para esto es necesario que los metadatos estén representados a través de lenguajes abiertos [5] como XML (*eXtended Markup Language*) [6], ya que se considera que los metadatos basados en tecnología XML son un elemento clave para la administración de repositorios digitales, con esta alianza se puede llevar a cabo el intercambio de información y de contenidos, entre plataformas y entre los ROAs, de forma transparente para el usuario [7].

No cabe duda que el XML se está convirtiendo rápidamente en uno de los más importantes formatos de datos. Una de las principales ventajas del uso de XML es que puede ser utilizado para representar datos estructurados y guardar la estructura sintáctica del documento, y en el campo del *e-learning* es muy importante porque facilita una búsqueda más eficaz y efectiva de los OAs. La búsqueda con buscadores tradicionales no satisface a la mayoría de los usuarios. Debido a esto se realizó un estudio profundo de las herramientas que existen en la actualidad para la indexación y búsqueda sobre documentos XML.

La mayoría de los motores de búsqueda son basados en palabras clave, estos presentan limitaciones en la calidad de sus resultados [8]. Debido a esto muchos autores han propuesto motores de búsqueda especializados, mediante la introducción de técnicas para construir taxonomías para clasificar documentos en un tópico en específico.

Una restricción importante para la recuperación de recursos a través de sistemas automatizados es la incapacidad de búsquedas semánticas, problema que sigue causando gran número de respuestas fallidas en los buscadores más potentes, ya que los motores carecen de inteligencia y aún no procesan el significado de las palabras. Muchas investigaciones han introducido elementos semánticos para mejorar las búsquedas, tal es el caso de *XSearch* [9], el cual presenta un lenguaje sencillo de consultas y devuelve fragmentos de documentos semánticamente relacionados que satisfacen las consultas. Otras herramientas combinan técnicas de la Web Semántica con motores de búsqueda especializados [10].

Las bases de datos nativas en XML son también muy utilizadas, entre ellas se encuentran: *XIndice* y *eXist* donde su principal beneficio es que no hay que preocuparse por la correspondencia entre los documentos XML con otras estructuras.

LUCENE [11], es una biblioteca escalable de alto rendimiento para la recuperación de información, que permite agregar capacidades de indexación y búsqueda a las aplicaciones. Puede indexar y buscar cualquier dato que pueda ser convertido a formato textual. Se trata de un API de indexación y búsqueda con simplicidad en su utilización que requiere básicamente que el usuario aprenda el uso de sus clases.

Otro motor muy potente es SOLR, el cual funciona con la biblioteca de búsqueda de LUCENE mencionada anteriormente. Éste se ejecuta como una aplicación Web en Java, permitiendo que se puedan pasar los documentos a indexar vía HTTP y consultarlos utilizando peticiones HTTP GET. Esta interacción por medio del protocolo HTTP, permite que las aplicaciones que utilicen esta herramienta, no necesariamente deban estar escritas en Java, sino que pueden estar en cualquier otro lenguaje.

La herramienta SOLR fue la que se utilizó para realizar la implementación del motor de búsqueda propuesto en este trabajo, para verificar su correcta elección, se realizaron pruebas con una muestra de 25 profesores de la carrera de Agronomía sobre un ROA de la Facultad de Agronomía de la Universidad Agraria de La Habana.

3 METODOLOGÍA

3.1 Herramienta Solr

Como se mencionaba anteriormente SOLR es una herramienta de búsqueda, subproyecto de LUCENE. La

misma se ejecuta como una aplicación Web en Java, permitiendo que se puedan enviar los documentos a indexar vía HTTP y consultarlos utilizando peticiones HTTP GET y HTTP POST.

La arquitectura de SOLR está conformada por tres capas fundamentales: interface de administración (que radica sobre un servidor web), núcleo de SOLR y LUCENE. En la Figura 1 se muestra gráficamente como está compuesta esta arquitectura.

Hay que destacar en esta arquitectura lo siguiente:

- Las configuraciones se realizan por medio de ficheros XML.
- Manipuladores de respuestas con diferentes formatos de salidas. (XML, JSON, etc.)
- Implementaciones de réplicas fuera de la máquina virtual de Java.
- Producto que es una extensión de LUCENE, utiliza muchas de sus terminologías. Los índices creados por SOLR son compatibles con el motor de búsqueda de LUCENE.
- Presenta API para Ruby, PHP, Java, Python, JSON, entre otras.
- Soporta toda la sintaxis de búsqueda de LUCENE, incluidos los operadores de Internet de consulta (+, -, "").
- Opciones para el manejo de consultas personalizadas.
- Sinónimos y palabras de paradas configurables en archivos de texto.

Tanto en SOLR como en LUCENE el índice es construido por uno o más documentos. Un documento consiste en uno o más campos. Un campo consiste en nombre, contenido y metadato que hacen que LUCENE pueda manipular el contenido. Este índice es invertido donde los términos son los que tienen el rol principal y cada término se refiere a los documentos que lo contienen.

Una de las principales fortalezas de esta herramienta es que soporta índice incremental, que muchas bibliotecas de recuperación de información no son capaces de soportar. Mientras algunas de estas bibliotecas necesitan reindexar todo el corpus cuando un nuevo documento es agregado al índice, LUCENE no necesita hacerlo, después que cada documento es agregado al índice, su contenido puede ser buscado en cualquier momento. El hecho de que LUCENE soporta esta característica la hace más apropiada para entornos que tratan grandes volúmenes de información donde la reindexación sería inmanejable.

Tanto LUCENE como SOLR son fácilmente reutilizables para aplicaciones que requieran agregar la capacidad de búsqueda en una fuente de datos definida y su información se obtenga como texto. Una de las mejores ventajas de SOLR es que al ser un proyecto de código abierto y de rápida configuración, ofrece la posibilidad de agregar nuevas funcionalidades, desarrollando nuevos componentes.

3.2 Arquitectura de búsqueda para un ROA

Existen tres procesos fundamentales que no se pueden dejar de mencionar para el total funcionamiento de un motor de búsqueda, tales son: el análisis (modificar contenidos de la aplicación antes de ser indexada), la indexación (consiste en tomar entradas y pasarlas al índice) y la búsqueda (recuperación de información por parte del usuario).

Se definió una arquitectura para cualquier ROA (Figura 2), la misma cuenta con tres capas: cliente, servidor web y servidor SOLR. Pueden encontrarse en una misma máquina, así como pueden estar distribuidos por diferentes. Esto es posible ya que la comunicación entre estas capas es realizada a través del protocolo HTTP, ventaja que facilita SOLR. La interfaz web es el único componente de color diferente, está hecho con el objetivo de recalcar que es el que se debe implementar completamente. Los demás componentes solo se deben configurar dependiendo del repositorio en que se vaya a trabajar.

Esta interfaz web se puede implementar en cualquier lenguaje de programación, en este caso se utilizó Java. La debe tener una estrecha relación con el ROA, incluso puede estar incluida dentro del mismo, en dependencia como el usuario decida.

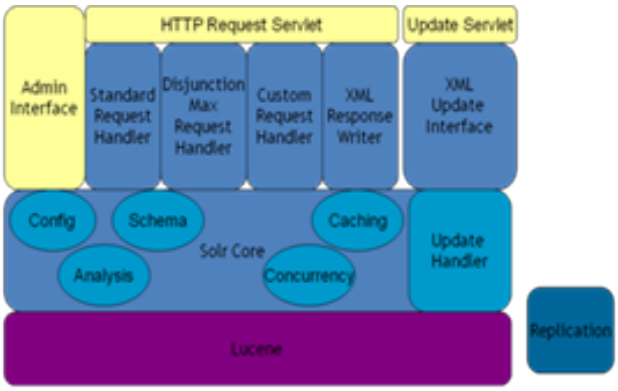

Figura 1 Arquitectura de SOLR.

3.3 Proceso de Análisis

El proceso de análisis se puede ejecutar para modificar contenidos de la aplicación antes de ser indexada. En SOLR un analizador (*analizer*) consiste en un marcador (*tokenizer*) y en uno o más filtros de marcas (*tokenfilters*). El marcador es responsable de producir marcas que en la mayoría corresponden con las palabras a indexar. Los filtros de marcas escogen de las marcas realizadas y las modifica o las elimina antes de ser indexadas. Además se pueden ejecutar estos analizadores en el momento de realizar las consultas.

Otros tipos de análisis se pueden llevar a cabo como: obtención de derivados, expansión de sinónimos, eliminación de palabras en blanco, entre otros.

3.4 Proceso de Indexación

El proceso de indexación consiste en tomar entradas y enviarlas al índice de SOLR mediante mensajes XML por vía HTTP POST. Se pueden procesar cuatro formas de peticiones:

- *add/update* (adiciona o modifica un documento)
- *commit* (prepara el índice para poder permitir la búsqueda)
- *optimize* (reestructura los ficheros del índice de LUCENE para un mejor funcionamiento del proceso de búsqueda)
- *delete* (elimina un documento)

Un aspecto muy importante en el proceso de indexación de SOLR es la normalización de contenidos, es transformar al estándar SOLR todos los campos que se desean indexar. Para esto se realizó una selección de los campos más propicios para ser indexados, qué tipo de datos guarda ese campo, si presenta múltiples valores, descripción del mismo y si debe ser guardado en el índice o no.

3.5 Proceso de Búsqueda

Una vez que son agregados todos los documentos al índice pueden ser buscados en cualquier momento, esto es posible mediante mensajes por vía HTTP GET y HTTP POST.

La interfaz de búsqueda realizada especifica campos de búsqueda como catálogo, entrada, título, idioma, descripción, palabras clave, entre otras. Se pueden especificar otros parámetros de búsqueda como operadores lógicos, número de documentos a devolver, resaltar los resultados y ordenar por determinados campos.

Una vez enviada la consulta al servidor, el resultado se muestra en un fichero XML, con los campos que SOLR almacenó en el índice. Esto se puede mostrar en formato HTML mediante el uso de plantillas XSLT, incluso se puede hacer un vínculo al repositorio que pertenezca dicho objeto, para poder verlo completamente.

4 RESULTADOS OBTENIDOS

En este apartado se describen las pruebas experimentales que se realizaron con la herramienta implementada, para comprobar la factibilidad de la arquitectura que se planteó con respecto a los buscadores tradicionales (Google para nuestro caso). Y para esto se les orientó a los profesores realizar las siguientes búsquedas:

- Búsqueda sencilla utilizando Google en el dominio del Repositorio de Objetos de Aprendizaje de la carrera de Agronomía.
- Búsqueda avanzada de Google sobre el mismo dominio mencionado anteriormente.
- Búsqueda con la herramienta implementada basada en SOLR, sobre el mismo dominio mencionado anteriormente.

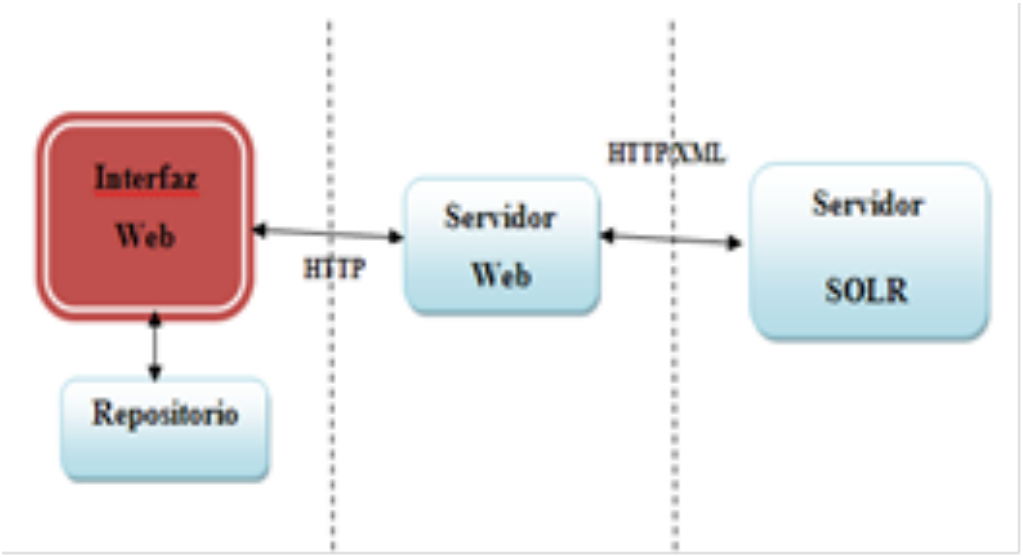

Figura 2 Arquitectura de búsqueda para cualquier Repositorio.

El total de la muestra fue de 25 profesores de la carrera de Agronomía, los cuales debían buscar materiales con temas relacionados con su línea de investigación y con las clases que imparten. Entre los parámetros que se definieron para medir los resultados fue la precisión, que se basa en la fórmula: Precisión = Número documentos devueltos que satisfacen al profesor / Número de documentos devueltos, es decir va a estar dada por la división de los documentos que satisfacen al profesor entre los documentos totales que devolvió la herramienta, mientras más se acerque la división a "1", mayor será la precisión de la herramienta y mayor la utilidad de estos objetos de aprendizaje al profesor. Los demás parámetros son los siguientes:

- Realizar la consulta.
- Medición de la calidad de los objetos devueltos.
- Reutilización de los objetos en otros entornos de aprendizaje.
- Identificación de estructuras de aprendizaje.

Una vez definidos los tipos de búsqueda y los parámetros a medir se le asignaron a cada profesor 7 tipos de consulta que debían de formular para cada tipo de búsqueda. Las consultas son las siguientes:

1º. Objetos de aprendizaje relacionados con su tema de investigación.
2º. Objetos de aprendizaje en formato pdf relacionados con su tema de investigación.
3º. Objetos de aprendizaje en idioma inglés relacionados con su tema de investigación.
4º. Objetos de aprendizaje que se relacionan con uno de los Objetos de aprendizaje encontrados en consultas anteriores.
5º. Objetos de aprendizaje destinados a estudiantes de la carrera de Agronomía.
6º. Objetos de aprendizaje creados por un autor específico.
7º. Objetos de aprendizaje que solo sean utilizados en sistemas operativos Windows.

Los resultados de estas búsquedas realizadas por los profesores se analizaron estadísticamente y se conformó el gráfico que se muestra en la Figura 3.

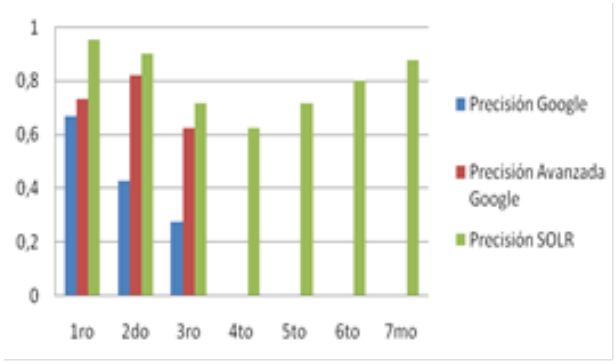

Figura 3 Precisión de los tres tipos de búsqueda.

4 CONCLUSIONES

Con todos los resultados de las pruebas realizadas con las diferentes herramientas, se llegaron a las siguientes conclusiones.

Con la búsqueda simple de Google y la avanzada no se pueden medir todos los parámetros de calidad de los resultados de la herramienta porque Google es un buscador de carácter general y no de propósito bien determinado como él que se basa en la arquitectura que presentamos en este trabajo, además Google tiene un carácter comercial. A pesar de todo esto, Google tiene una popularidad muy grande y la comparación con él aunque nos permita hacer una comparación más detallada sí podría indicarnos la utilidad y comportamiento de la implementación realizada.

El parámetro precisión utilizando la herramienta basada en SOLR es mayor que en las otras, debido a que los objetos están descritos mediante el uso de estándares.

El uso de los metadatos permite aumentar nuestra fuente de búsqueda, ya que describen muchas características que nos conducen fácilmente al objeto de aprendizaje. Con la búsqueda utilizando la herramienta basada en SOLR se pueden medir todos los parámetros de calidad de los resultados de la herramienta.

Con la herramienta basada en SOLR el profesor puede buscar los materiales más específicos para su trabajo docente y reutilizar el objeto de aprendizaje para otros entornos de aprendizaje que existan. También puede identificar cual objeto de aprendizaje representa una estructura de aprendizaje y que tipo es.

Como conclusión general se debe decir que el uso de herramientas de búsqueda en XML repercute grandemente en la calidad de los resultados, ya que un objeto de aprendizaje descrito mediante el uso de estándares, es el paso esencial para poder explotar toda la valiosa información que poseen los Repositorios de Objetos de Aprendizaje. Y que la aplicación creada debe considerarse como punto de partida para el estudio y ampliación de la potencia de SOLR como herramienta de recuperación de información. Como trabajo futuro pretendemos agregar el uso de ontologías para mejorar la calidad y la eficiencia de la herramienta, y así agregarle inteligencia y poder procesar mejor el significado de las palabras.

5 REFERENCIAS

[1] Iriarte, L.; Marco, M.; Morón, D.; Pernias, P. Architecture Oriented towards the management of Learning Objects Repositories (LOR@). Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies. 2006.
[2] Bruce Croft, W. What Do People Want from Information Retrieval? D-Lib , 1. 1995.
[3] IEEE Standards Department. Draft Standard for eXtensible MarkupLanguage (XML) Binding for Learning Object Metadata Data Model. 2002.

[4] Hernández, H.; Saiz, M. Ontologías mixtas para la representación conceptual de objetos de aprendizaje. Procesamiento del Lenguaje Natural. 2007.
[5] Toshniwal, R.; Agrawal, D. P. Tracing the roots of markup languages. Commun. ACM , pp. 95 - 98. 2004
[6] Bray, T.; Paoli, J. Sperberg-McQueen, C. M., Maler, E., & Yergeau, F. (2004). Extensible Markup Language (XML) 1.0 (Third Edition). URL: http://www.w3.org/TR/2004/REC-xml-20040204/
[7] Ahmed, K.; Ayers, D.; Birbeck, M.; Cousins, J.; Dodds, D.; Lubell, J. Professional XML Metadata. UK: Wrox Press Ltd. 2001.
[8] Arrigo, M.; Gentile, M.; Taibi, D.; Pagoto, G. Specialized Search Engines for E-learning. Recent Research Developments in Learning Technologies. 2005.
[9] Cohen, S.; Mamou, J.; Kanza, Y.; Sagiv, Y. XSEarch: A Semantic Search Engine for XML. Proceedings of 29th International Conference on Very Large Data Bases, pp. 45-56. Berlín: Alemania. 2003.
[10] Taibi, D.; Gentile, M.; Seta, L. A Semantic Search Engine for Learning Resources. 3rd International Conference on Multimedia and Information & Communication Technologies in Education. Cáceres: España. 2005.
[11] Goetz, B. The Lucene search engine: Powerful, flexible, and free. 2000.
[12] URL: http://www.javaworld.com/javaworld/jw-09-2000/jw-0915-lucene.html.