ThesaurusAPI: una API para la manipulación de tesauros

ThesaurusAPI: an API for thesaurus manipulation

María Mercedes MARTÍNEZ-GONZÁLEZ (1), María MUÑOZ NIETO (2)

Departamento de Informática, Universidad de Valladolid (España)
(1) mercedes@infor.uva.es (2) maria.munoz@gmail.com

Resumen

ThesaurusAPI es una API para la manipulación de tesauros. A nivel interno, los tesauros se representan con SKOS, y se utilizan almacenes RDF para su almacenamiento. En este artículo se presenta la API y se comentan algunas características: funcionalidad, tratamiento de la integridad, uso de SKOS, así como las ventajas e inconvenientes de utilizar SKOS. La API está orientada específicamente a la manipulación de tesauros, lo cual la diferencia de las API orientadas a SKOS. Se trata de una API pensada para facilitar el desarrollo de aplicaciones que manipulan tesauros que pueden estar almacenados localmente. Una implementación de libre distribución está disponible.

Palabras clave: Tesauros. SKOS. API.

Abstract

ThesaurusAPI is an API for managing thesauri. Internally, thesauri are represented with SKOS, and RDF stores are used to store them. The functionality and integrity management of the API is discussed, and the advantages and disadvantages of using SKOS are commented. This API has been specially created for thesauri management, which is a significant difference with other APIs, SKOS-oriented. It is intended for use from applications that manage thesauri stored in the same server in which the application runs. A distribution is available for free download.

Keywords: Thesauri. SKOS. API.

1. Introducción

Los tesauros son herramientas conceptuales (*KOS*, *Knowledge Organization Systems*) que permiten modelar el vocabulario de un dominio. Estándares internacionales como la norma ISO 25964-1:2011 (IS0, 2011), heredera a su vez de ISO 2788-1986 (ISO, 1985; ISO, 1986), proporcionan las directrices para crear y organizar tesauros. El valor de los tesauros como herramientas para mejorar la recuperación de información en los sistemas de información ha sido reconocido desde hace tiempo. Por otro lado, la Web y, más concretamente los avances en el marco de la Web Semántica, han traído consigo un renovado interés en los tesauros como herramientas conceptuales facilitadoras de la explotación de semántica en los sistemas de información (Blocks, Cunliffe y Tudhope, 2006; García Marco, 2007). Incluso puede afirmarse que el propio concepto de tesauro ha evolucionado para adaptarse mejor a los modelos de representación de la Web semántica, evolucionando de un modelo terminológico a un modelo conceptual (García Marco, 2007; García Torres, Pareja Lora y Pradana López, 2008; Pastor-Sánchez, Martínez Méndez y Rodríguez-Muñoz, 2009). También en relación con la Web Semántica, encontramos estándares como SKOS (World Wide Web Consortium, 2009), que se utilizan para representar tesauros.

La utilización de los tesauros en formato electrónico requiere la disponibilidad de herramientas que permitan construir y editar tesauros, o consultarlos y utilizarlos desde aplicaciones diversas. En este contexto se sitúa la necesidad, habitual en los tratamientos informáticos, de paquetes de software genéricos que faciliten su labor a los desarrolladores de aplicaciones. Las *Application Programming Interfaces* (API) facilitan esta tarea. Una API proporciona una interfaz de programación general que puede ser reutilizada desde diversas aplicaciones. Así pues, una API para manipular tesauros facilita la construcción de herramientas que manipulan tesauros.

Existen diversas herramientas para manipular tesauros —se pueden consultar estudios comparativos en (Moya Martínez y Gil Leiva, 2001; Pérez-León y Martínez-González, 2009; Wiil, 2010) y encontrar amplias recopilaciones en (Miles y Bechofer, 2009)—, e incluso APIs adaptadas a estas herramientas, o a tesauros específicos. Sin embargo, cuando se planteó desde el grupo de trabajo de las Universidades de Valladolid y de León la búsqueda de una herramienta que permitiese estos desarrollos, no se encontró ninguna de tipo genérico, independien-

te de tesauro, que tuviese toda la funcionalidad requerida (Martínez-González, Pérez-León y Alvite-Díez, 2009), razón por la cual se abordó el diseño de una API genérica y el desarrollo del paquete software que la implementase. No obstante, las API encontradas se tuvieron en cuenta como una referencia en el diseño de la que aquí se presenta. Igualmente, se han tomado en consideración algunas ideas aportadas en Pastor-Sánchez (2009).

ThesaurusAPI es una API genérica para tesauros desarrollada en la Universidad de Valladolid. Sus características más destacables son: es una API genérica para tesauros, independiente del formato de representación de los tesauros manipulados, se proporciona como un paquete de libre distribución, desarrollado en Java, y soporta una amplia funcionalidad, que permite tanto la consulta como la edición de tesauros, así como la importación de tesauros externos y su exportación. Por otro lado, se trata de una API pensada para facilitar el desarrollo de aplicaciones que manipulan tesauros que se almacenan localmente, de los cuales, por tanto, se tiene control y capacidad para modificarlos según las propias necesidades. El objetivo es permitir el desarrollo de aplicaciones de usuario diferentes, que acceden a los mismos tesauros, pero que son independientes de modificaciones en los niveles inferiores del software: cambios de almacén, cambios en los esquemas, etc.

En este artículo se presenta esta API y se comentan algunas de sus características más relevantes. La funcionalidad abarca la edición de tesauros, su consulta y la importación y exportación. La gestión de la integridad es otro aspecto al que se dedicó especial atención. Por último, se comentan las ventajas e inconvenientes de haber elegido un estándar de la Web Semántica, SKOS/RDF, para representar los tesauros almacenados. El uso de SKOS conlleva la utilización de un almacén RDF para guardar los tesauros.

2. Tesauros y Web Semántica

2.1. Utilidad de los tesauros en el marco de la Web Semántica

Los tesauros tienen una importante utilidad en el marco de la Web Semántica. Además de proporcionar unas relaciones predefinidas (*Broader Than*, *Narrower Than*, *Related To*), cuya semántica es común para cualquier usuario de tesauros, tienen el valor de que los conceptos representados también están validados por una comunidad, amplia en algunos casos, lo cual supone en realidad un acuerdo respecto a la semántica de los conceptos y del vocabulario (o términos) usados para referirse a ellos. Al ser el resultado del acuerdo de una comunidad sobre el vocabulario, o conceptos, utilizados de modo común por esa comunidad, su valor como herramienta para manipular semántica es muy alto. Sirven pues, para facilitar búsquedas semánticas, donde distintos proveedores comparten su conocimiento de un dominio.

Por esta razón, tesauros tan importantes como Eurovoc, Agrovoc, etc., se han adaptado a la Web Semántica y sirven de soporte de búsquedas en sus correspondientes sistemas de información. La figura 1 muestra una búsqueda en el sistema de información EUR-Lex de la Unión Europea usando Eurovoc.

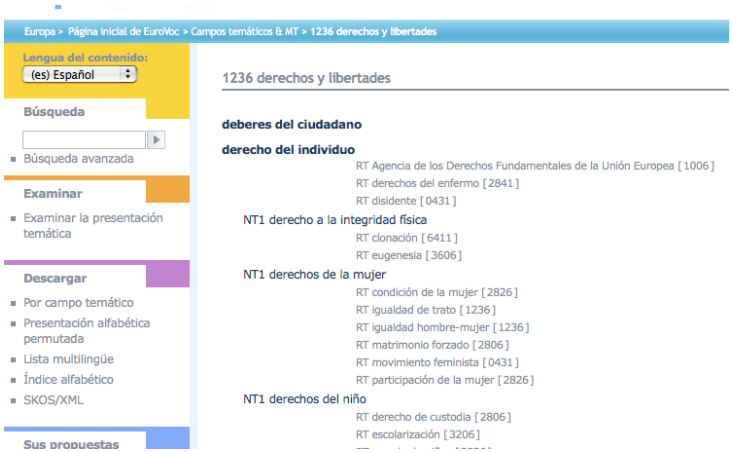

Figura 1. Uso del tesauro Eurovoc en EUR-Lex

2.2. Estándares para la representación de tesauros

La expansión de la tecnología y estándares asociados a la Web Semántica ha proporcionado como estándar de representación de los tesauros el lenguaje SKOS (World Wide Web Consortium, 2009). Según la propia definición provista en la Recomendación, SKOS es un 'modelo de datos diseñado para compartir sistemas de organización de conocimiento en la Web'. En SKOS se representan conceptos, lo cual nos sitúa en el contexto de tesauros basados en conceptos. A cada concepto se le asocian *etiquetas*, las cuales pueden ser *preferidas* (descriptores) o *no preferidas* (sinónimos). Asimismo, están también presentes las relaciones semánticas típicas de los tesauros: *narrower term (NT)*, *broader term (BT)*, *related* term (RT). SKOS es actualmente una Recomendación estable del W3C. La figura 2 es un ejemplo sencillo, extraído del tesauro Eurovoc, el cual se ha simplificado para mostrar la representación de un concepto y los distintos términos o etiquetas

utilizados para referirse a él; en este ejemplo se muestran únicamente los descriptores o *etiquetas preferidas* elegidos en español (‘*España*’) y francés (‘*Espagne*’), y un sinónimo o *etiqueta no preferida* (‘*Reino de España*’), con la cual nos estaríamos refiriendo a este mismo concepto o idea.

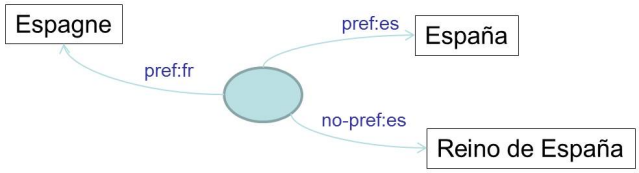

Figura 2. Grafo RDF simplificado en el que se representa un concepto

SKOS es una aplicación de RDF, lo cual supone que cuando se representa un tesauro con SKOS se está construyendo en realidad un grafo RDF, el cual se puede manipular con cualquiera de las herramientas provistas al efecto. Además es posible usar cualquiera de los formatos disponibles para representar grafos RDF: XML, N3, Turtle, etc. Si bien SKOS no está pensado exclusivamente para representar tesauros, es cierto que su utilización con los tesauros es una de sus aplicaciones más importantes. De hecho, la propia guía introductoria a SKOS, *SKOS Primer* (Isaac y Summers, 2009), proporciona en uno de sus apéndices una tabla comparativa que indica las correspondencias más importantes entre los conceptos cubiertos en la norma ISO 2788 y sus equivalentes en SKOS. Diversos grupos se han preocupado por proponer metodologías para representar tesauros con SKOS (Van Assem, Malaisé, Miles y Schreiber, 2006).

Existen además diversas propuestas de representación de variados tesauros bajo SKOS, algunas de ellas accesibles en la web. Así, se pueden encontrar representaciones RDF/SKOS de los tesauros Agrovoc, UKAT, GEMET, etc., en los sitios web de los respectivos tesauros, o en recopilaciones de implementaciones con SKOS sostenidas por el W3C —ver (World Wide Web Consortium, 2012) o Miles y Bechofer (2009).

En agosto de 2011, se aprobó la primera parte de un nuevo estándar ISO para tesauros: *ISO 25964-1:2011 Information and documentation – Thesauri and interoperability with other vocabularies – Part 1: Thesauri for information retrieval* (ISO, 2011), que consta de dos partes. La segunda parte, que se espera trate los aspectos relacionados con interoperabilidad entre tesauros y ontologías, está actualmente en desarrollo. Dado lo reciente de este estándar, hasta donde conocemos, aún no están disponibles tesauros representados conforme a sus reglas.

3. ThesarusAPI: una API para tesauros

ThesaurusAPI es una API genérica para manipular tesauros, independiente del dominio. Su utilidad principal, al igual que la de cualquier API, es facilitar la tarea de los desarrolladores de aplicaciones, en este caso desarrolladores de aplicaciones que manipulen tesauros. Su especificación, así como el software que la implementa, están disponibles en http://www.infor.uva.es/~mercedes/ThesaurusAPI/. La figura 3 muestra una captura de pantalla de la página de bienvenida. También están disponibles los manuales de instalación y utilización, así como una pequeña aplicación desde la cual se utiliza, cuyo valor principal es servir como modelo para los desarrolladores de aplicaciones sobre el modo de invocar sus métodos. En la figura 4 se muestra una captura de pantalla de esta aplicación, cuya interfaz de usuario traslada de modo prácticamente directo las interfaces de los métodos proporcionados por la API.

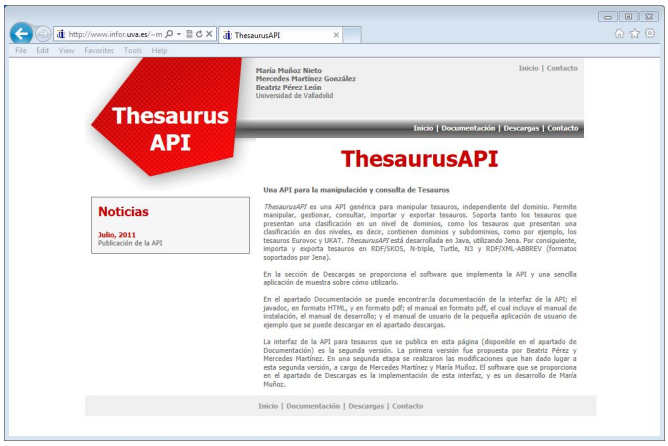

Figura 3. ThesaurusAPI: página de bienvenida

Esta API está orientada a tesauros basados en conceptos, lo cual supone que sus usuarios trabajan con *dominios*, *conceptos*, *etiquetas preferidas*, *etiquetas no preferidas*, etc. Se considera un tesauro compuesto de *elementos*, que pueden ser de tres tipos: *dominio*, *concepto* o *colección*. Su interfaz es independiente del formato elegido para la representación de los tesauros, y del tesauro utilizado. La funcionalidad incluye capacidades de creación y edición de tesauros, consultas en los tesauros, e importación y exportación de tesauros a los formatos o lenguajes vinculados actualmente con la Web Semántica, tales como RDF/XML, N3 o Turtle. Se permite manipular tesauros monolingües o

multilingües y se proporciona, como se ha adelantado en este mismo párrafo, soporte para *colecciones*. También se permite asociar metadatos a los tesauros.

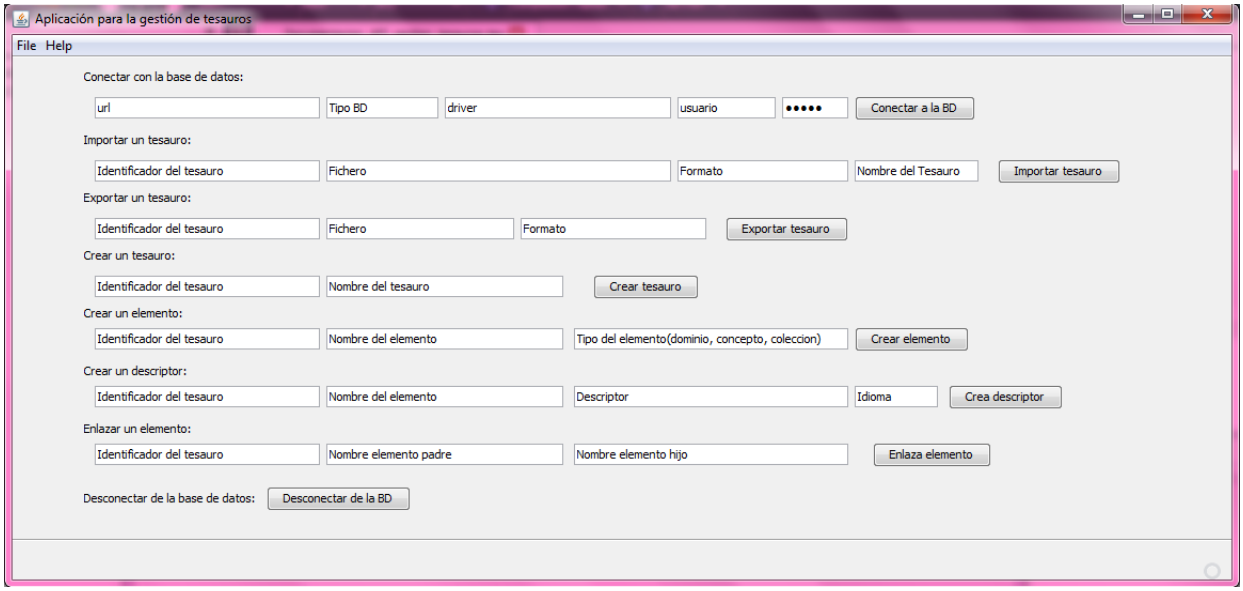

Figura 4. Una aplicación de ejemplo que usa ThesaurusAPI

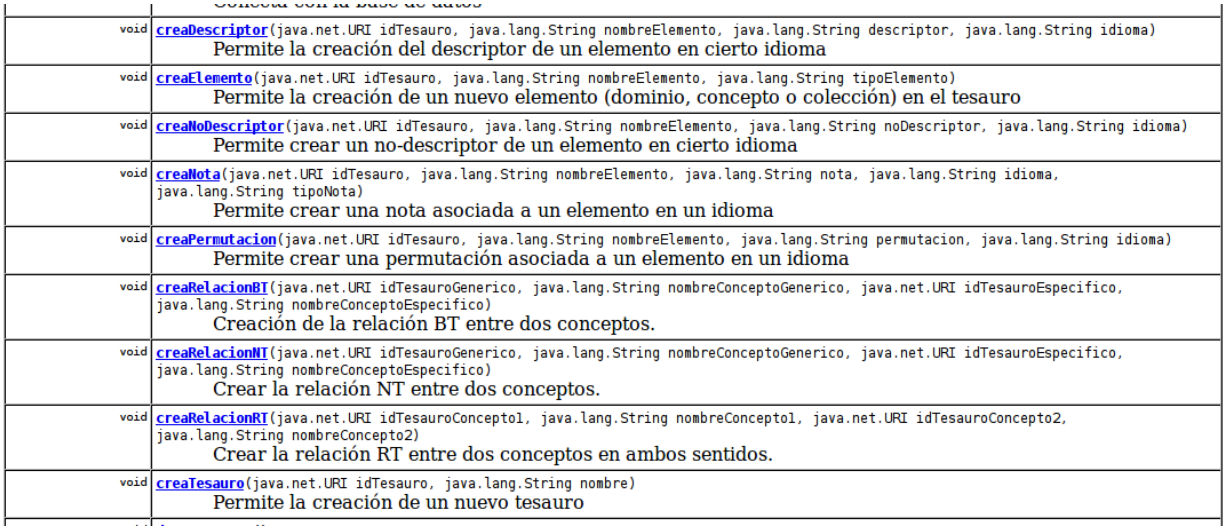

Figura 5. Un extracto de la interfaz de los métodos de creación de tesauros y elementos de tesauros

3.1. Funcionalidad

Como se ha indicado, la funcionalidad de la API permite un tratamiento integral de los tesauros. Es posible editar tesauros, consultar los tesauros almacenados, importar y exportar en varios formatos.

La edición comprende métodos para crear tesauros, asociar metadatos Dublin Core a los tesauros, crear *elementos* (dominio, concepto o colección), crear descriptores, no descriptores, notas y variantes léxicas (permutaciones). Es posible establecer un concepto como *top concept* de un determinado dominio, enlazar un elemento en la jerarquía del tesauro (por ejemplo, indicar que un concepto pertenece a un dominio concreto), y crear relaciones BT, NT o RT entre conceptos. Por supuesto, también están disponibles métodos para realizar las operaciones inversas: eliminar tesauros, borrar metadatos de un tesauro, elementos, descriptores, etc. La figura 5 muestra un extracto de la interfaz de algunos métodos de creación de tesauros, o de elementos de un tesauro. El con-

junto de interfaces de sus métodos se puede encontrar en la Javadoc disponible en su web, bajo la clase *ThesaurusAPI*.

En lo que se refiere a recuperación y búsqueda se han previsto métodos para recuperar los identificadores de los tesauros almacenados, sus metadatos, la información completa de un elemento dado, los conceptos genéricos, o los específicos, de uno dado, o aquellos con los que está relacionado. Se puede recuperar de modo independiente el descriptor de un concepto, sus no descriptores, o notas. Las búsquedas son una de las capacidades más notables de esta API, por la flexibilidad de la que se la ha dotado a la hora de restringir las búsquedas. Se han considerado búsquedas de elementos, restringiendo la búsqueda de la cadena de entrada en los descriptores, no descriptores, notas o permutaciones. También permite restringirla por posición en la jerarquía del tesauro, de modo que, por ejemplo, sólo se busque entre los descendientes de un elemento dado.

Finalmente encontramos las posibilidades de importación y exportación de tesauros. Se pueden importar tesauros cuyos formatos sean RDF/XML, N-Triple, Turtle, N3 y RDF/XML Abbrev. Igualmente se pueden exportar los tesauros almacenados a estos formatos. Asimismo se ha considerado la posibilidad de importar fragmentos de un tesauro dentro de uno dado: los fragmentos que se importan son subárboles cuya raíz es un elemento (dominio o concepto).

3.2. Tratamiento de la integridad

La gestión de la integridad es un aspecto importante en la manipulación de tesauros, al cual se ha dedicado una atención especial durante el diseño y desarrollo de esta API. Cuando la herramienta que manipula tesauros asume la comprobación y gestión de las reglas de integridad que debe cumplir un tesauro bien formado, sus usuarios se ven liberados de tareas que pueden ser arduas. La comprobación de las reglas básicas definidas por los estándares, de modo que cada vez que se edite un tesauro, sea el software quien avise de los posibles errores que se están cometiendo, permite a los creadores de tesauros tener una mayor seguridad de que los tesauros resultantes son correctos y no se han deslizado errores que conlleven incoherencias que tengan repercusión posterior en las consultas y accesos al tesauro.

En *ThesaurusAPI* se ha previsto la gestión automática de las reglas de integridad aplicables a los tesauros conformes a las normas ISO 2788 e ISO 5964. De este modo, cuando se crea una relación BT, NT o RT entre dos conceptos se crea automáticamente la relación inversa o simétrica. Por ejemplo, ante la creación de la relación '<*derecho del individuo*> BT <*derecho de la integridad física*>' automáticamente se creará en el tesauro otra relación equivalente: '<*derecho de la integridad física*> NT <*derecho del individuo*>'. Además se comprueban también aquellas situaciones en las que se llegaría a una incoherencia, como por ejemplo, la incompatibilidad de una relación BT o NT entre dos conceptos con una relación RT entre ellos mismos. Existen más comprobaciones de la integridad en *ThesaurusAPI*. Para más detalle se puede consultar la interfaz de la API, en cuyos métodos se indica la actuación que se realiza al respecto.

Merece la pena destacar que algunas de estas reglas no están cubiertas en SKOS, que, por ser un estándar que da cobertura a la representación de KOS adicionales a los tesauros, no restringe tanto la estructura permitida como debe hacerse en un tesauro. Uno de estos casos se da con los ciclos en las relaciones semánticas, algunos de los cuales están permitidos en SKOS mientras que en un tesauro suponen una incoherencia. Por ejemplo, en un tesauro es incompatible la presencia simultánea de las dos afirmaciones siguientes: '<*derecho del individuo*> BT <*derecho de la integridad física*>' y '<*derecho de la integridad física*> BT <*derecho del individuo*>'. Sin embargo, la norma SKOS permite este tipo de situaciones (véase el Apéndice de (Isaac, 2009) para más detalles). Así pues, la gestión de este tipo de integridad supone un elemento diferenciador respecto a otras API orientadas a SKOS (Jupp, 2009) o a ontologías (Horridge y Bechofer, 2011).

3.3. Uso de SKOS

Existen algunas decisiones que merece la pena mencionar respecto a determinados aspectos de diseño y desarrollo. En este apartado se tratará la decisión de optar por SKOS como estándar de representación para los tesauros almacenados y del tratamiento de la integridad.

Para la representación de los tesauros se eligió SKOS. Las razones para hacerlo fueron varias. En primer lugar, como se ha indicado en la Introducción, se valoró que SKOS es un estándar del W3C, ampliamente difundido y utilizado para representar tesauros en la web. Esto suponía que la API debía ser capaz de manipular tesauros representados con este lenguaje. La decisión sobre el lenguaje de representación utilizado para el almacenamiento está directamente relacionada con la decisión acerca del tipo de gestor o almacén de datos que se utilizará. Utilizar SKOS suponía utilizar RDF y tener, por tan-

to, la posibilidad de recurrir a almacenes RDF para el almacenamiento de los tesauros.

Para representar los tesauros se han tenido en cuenta las pautas y consideraciones aportadas en diversos autores (Isaac y Summers, 2009; Matthews, Alistair y Wilson, 2001; Miles, Mathews, Beckett, Brickley, Wilson y Rogers, 2005; Pástor-Sánchez, 2009; eFoundations, 2011). Las decisiones tomadas se han basado en el trabajo iniciado con la representación del tesauro Eurovoc con SKOS, documentadas en Alvite Díez, Pérez León y Martínez González (2010). Se ha utilizado SKOS Core.

Para los tesauros y dominios se utiliza la clase *skos:ConceptScheme*. Los conceptos tienen su propia clase, *skos:Concept*. La pertenencia de un elemento a un tesauro o dominio se modela con la relación *skos:inScheme*. A su vez, los distintos tipos de etiquetas, preferidas, no preferidas, y variantes léxicas, se corresponden con las clases *skos:prefLabel*, *skos:altLabel* y *skos:hiddenLabel* respectivamente. Asimismo, las relaciones RT, NT y BT están reflejadas en las equivalentes clases *skos:related*, *skos:narrower* y *skos:broader*.

4. Ventajas e inconvenientes de usar SKOS

Existen varios almacenes RDF disponibles, tanto de libre distribución como propietarios: Jena (Carroll, Dickinson, Dollin, Reynolds, Seaborne y Wilkinson, 2004; McBride, 2002), 3store/4store/5store (Harris y Shadbolt, 2005), Virtuoso (Erling y Mikhailov, 2007), AllegroGraph (Franz Inc., 2012), etc, ampliamente probados, que eran por tanto candidatos óptimos para facilitar este almacenamiento.

Cualquiera de ellos facilita la manipulación y consulta de datos RDF, que es en última instancia un tesauro representado con RDF. Pero tienen además la ventaja añadida de soportar la conversión entre formatos asociados a RDF. De este modo, en el desarrollo de la API, se utilizan directamente las herramientas de conversión proporcionadas por el almacén en lugar de desarrollar conversores específicos para los distintos formatos. En este caso concreto, el almacén elegido durante el desarrollo del paquete software que implementa la API es Jena, lo cual ha supuesto que los formatos RDF soportados por el paquete distribuido son los mismos que soporta Jena: RDF/XML, N3, Turtle, RDF/XML-ABBREV.

Otro aspecto valioso de utilizar RDF como lenguaje para la manipulación interna está en la posibilidad de abstraerse de la sintaxis XML concreta utilizada para representar cada tesauro. El lenguaje de consulta asociado a RDF es SPARQL (World Wide Web Consortium, 2008), una de cuyas virtudes más destacables para un desarrollador es que las consultas no son dependientes de la sintaxis XML de los datos sobre los que se realizan, sino que se basan en el modelo de datos. Conociendo el modelo SKOS es posible realizar consultas sobre cualquier tesauro representado con él. Esto significa, en un desarrollo como éste, que las consultas previstas dentro de cada método de la API siguen siendo válidas aunque se cambie el almacén RDF utilizado y la sintaxis XML de los datos RDF almacenados. La figura 6 muestra un extracto de la consulta SPARQL que permite recuperar el descriptor (etiqueta preferida) de un elemento del tesauro.

```
PREFIX skos: <http://www.w3.org/2004/skos/core#>
SELECT ?descriptor
WHERE {
       <idElemento> skos:prefLabel ?descriptor .
       FILTER (lang(?descriptor) = "idioma") .
       }
```

Figura 6. Una consulta SPARQL

Obviamente, existen otras soluciones para representar los tesauros que manipulan las herramientas. Por ejemplo, una de las que ha sido más utilizada son las bases de datos relacionales, con el consiguiente uso del modelo relacional, basado en relaciones o tablas (Ballew, Duncan y Blasingame, 2009). Esta opción es de hecho la que utilizan algunas herramientas de manipulación de tesauros como *TemaTres* (Ferreyra, 2011). Es bien conocido que las bases de datos relacionales son sistemas robustos y eficientes, ampliamente probados, siendo de hecho por esta razón, una de las opciones más ampliamente utilizadas por los almacenes RDF, que 'ocultan' a sus usuarios los detalles sobre las tablas utilizadas para proporcionarles un acceso basado en todo momento en el modelo de datos RDF. Lógicamente, introducir una capa adicional de software, como es el caso de estos almacenes, supone aceptar una reducción de eficiencia respecto a la utilización de una base de datos relacional. Así pues, puede afirmarse que la elección de un almacén RDF supone estar dispuesto a renunciar al más alto grado de eficiencia en las consultas.

Por último cabe tomar en consideración que, si bien RDF, SKOS, SPARQL y los almacenes RDF, son estándares y tecnologías que están alcanzando rápidamente un amplio nivel de

aceptación entre la comunidad que manipula datos en la web, no han alcanzado aún un nivel de consolidación suficientemente alto como para ser conocidos por cualquier desarrollador de aplicaciones. Su elección puede suponer, pues, un cierto coste de aprendizaje, cosa que no ocurre, por ejemplo, con las bases de datos relacionales, cuya amplia trayectoria hace que sean conocidas por un amplio número de potenciales desarrolladores.

5. Conclusión

Se ha presentado *ThesaurusAPI*, una API genérica para manipular tesauros, cuyo objetivo es facilitar la labor de los desarrolladores de aplicaciones. La API completa se puede consultar en el sitio web previsto al efecto: http://www.infor.uva.es/~mercedes/ThesaurusAPI. Junto a ella, se ofrece una implementación de libre distribución, disponible como un paquete Java, y una aplicación cuyo objetivo es servir de modelo a los desarrolladores interesados en utilizarla.

Existen otras API para tesauros o KOS. En algunos casos se trata de API orientadas a la manipulación de KOS representados con SKOS (Jupp, Bechofer y Stevens, 2009). Otras están pensadas específicamente para tesauros, pero sólo están accesibles para los clientes de las herramientas a las que se han asociado (Koller, 2009). Si bien los tesauros representados con SKOS se pueden manipular, como es lógico, a través de una API de estas características, existen diferencias entre SKOS y los tesauros, especialmente en los aspectos relacionados con la integridad, que deben ser tenidas en cuenta a la hora de construir una aplicación que utiliza uno u otro tipo de API (Isaac y Summers, 2009). Hay además otras API pensadas específicamente para facilitar las consultas sobre tesauros a través de servicios web. Algunos ejemplos son las API diseñadas para acceder a los tesauros GEMET (GEMET, 2010) o AGROVOC. En la página web *KOS-based web services* (http://hypermedia.research.glam.ac.uk/kos/terminology-services/links/ (2012-03-10)) se puede encontrar una recopilación suficientemente amplia. En (Tudhope, Koch y Heery, 2006) se puede encontrar también una revisión de los protocolos y APIs disponibles en el momento en que se realizó. Asimismo se pueden consultar las referencias (Binding y Tudhope, 2004; Gerbe y Kerheve, 2010; Tuominen, Frosterus, Viljanen y Hyvönen, 2009).

La API que se ha presentado en este artículo es una API orientada específicamente a la manipulación de tesauros, lo cual la diferencia de las API orientadas a SKOS. Como hemos apuntado en el párrafo anterior, una diferencia importante está en el tratamiento de la integridad, que en nuestro caso está adaptado específicamente para tesauros. Por otro lado, *ThesaurusAPI* es una API genérica, independiente del tesauro, que no está pensada para facilitar la consulta a uno o varios tesauros, o a tesauros almacenados en sitios de terceros, sino que tiene como objetivo facilitar el desarrollo de aplicaciones que manipulan tesauros que pueden estar almacenados localmente. Introduce una capa intermedia que permite al desarrollador de la aplicación de usuario un trabajo más rápido e independiente de los detalles de la manipulación de los tesauros. En esto también difiere de API y servicios web orientados a este fin como son los proporcionados junto a los tesauros GEMET y Agrovoc, o con la herramienta TemaTres.

Existen varias posibilidades de trabajo futuro. En primer lugar está la lógica revisión de la API, y del paquete que la implementa, en base a la experiencia de utilización. En segundo lugar, se contempla la posibilidad de extenderla con soporte para la norma ISO 25964, cuya primera parte se aprobó recientemente y en cuya segunda parte trabaja el comité correspondiente.

Referencias

Alvite Díez, M. Luisa; Pérez León, Beatriz; Martínez González, M. Mercedes (2010). Propuesta de representación del tesauro Eurovoc en SKOS para su integración en sistemas de información jurídica. // Scire: representación y organización del conocimiento. 16:2 (jul.-dic. 2010) 47-51.

Ballew, Randy; Duncan, Thomas; Blasingame, Mike (1999). Relational Data Structures for Implementing Thesauri. University of California, 1999.

Binding, Ceri; Tudhope, Douglas (2004). KOS at your Service: Programmatic Access to Knowledge Organisation Systems. // Journal of Digital Information. 4:4 (2004). http://journals.tdl.org/jodi/article/view/110.

Blocks, Dorothee; Cunliffe, Daniel; Tudhope, Douglas (2006). A Reference Model for User-System Interaction in Thesaurus-Based Searching. // Journal of the American Society for Information Science and Technology. 57:12 (2006) 1655-1665.

Carroll, J.; Dickinson, J.; Dollin, C.; Reynolds, D.; Seaborne, A.; Wilkinson, K. (2004). Jena: implementing the Semantic Web recommendations. // WWW (Alternate Track Papers & Posters). 2004. 74-83.

eFoundations, 2011. Term-based thesaurus and SKOS (Part 1). http://efoundations.typepad.com/efoundations/2011/... (2012-03-10)

Erling, O.; MIkhailov, I. (2007). RDF support in the Virtuoso DBMS. // CSSW (Conference on Social Semantic Web). 2007. 59-68.

Ferreyra, D. (2011). TemaTres: gestión de vocabularios controlados. // http://www.r020.com.ar/tematres/wiki/doku.php?id=tematres:tematres_view. (2012-03-10)

Franz Inc. (2012). AllegroGraph RDFStore Web 3.0's Database. http://www.franz.com/agraph/allegrograph/ (2012-03-10).

Harris, S.; Shadbolt, N. (2005). SPARQL Query Processing with Conventional Relational Database Systems. // Lecture Notes in computer Science. 3807 (2005) 235-244.

Koller, Andreas (2009). SKOS Thesaurus Management based on Linked Data. // ESTC 2009, Third Annual European Semantic Technology Conference, 2 y 3 de diciembre de 2009. Vienna, Austria. 2009. http://poolparty.punkt.at/wp-content/uploads72009/10/2009_SKOS_ThesaurusManagement_Linked Data.pdf (2012-03-10).

García Marco, Francisco Javier (Coord). (2007). Proyectos internacionales de reforma y ampliación de las normas sobre tesauros para su adaptación a los nuevos contextos de integración e interoperabilidad en el entorno digital. // VIII Congreso ISKO – España. León: Universidad de León, 2007. 389-398.

García Torres, Alberto; Pareja Lora, Antonio; Pradana López, Daniel (2008). Reutilización de tesauros: el documentalista frente al reto de la Web semántica. // El Profesional de la Información. 17:1 (2008) 8-21.

GEMET (2010). GEMET webservice API. http://svn.eionet.europa.eu/projects/Zope/wiki/GEMETWebServiceAPI. 2010 (2012-03-10).

Gerbe, Olivier; Kerheve, Brigitte (2010). A Model-Driven Approach to SKOS Implementation. // International Conference on Internet and Web Applications and Services. ISBN 978-0-7695-4022-1. 0 (2010) 484-488. DOI= http://doi.ieeecomputersociety.org/10.1109/ICIW.2010.79 (2012-03-10).

Horridge, Matthew; Bechofer, Sean (2011). The OWL API: A Java API for OWL ontologies. // Semantic Web (2011). DOI = 10.3233/SW-2011-0025. Online Date: Friday, February 25, 2011.

Isaac, Antoine; Summers, Ed. (Eds.). (2009). SKOS Simple Knowledge Organization System Primer. W3C Working Group Note 18 August 2009. World Wide Web Consortium. (Aug. 2009).

ISO (1986). ISO 2788:1986 - Guidelines for the establishment and development of monolingual thesauri. Geneva: International Organization for Standardization (ISO).

ISO (1985). ISO 5964:1985 Documentation – Guidelines for the establishment and development of multilingual thesauri. Geneva: International Organization for Standardization (ISO).

ISO (2011). ISO 25964-1:2011 Information and documentation – Thesauri and interoperability with other vocabularies – Part 1: Thesauri for information retrieval. http://www.iso.org/iso/fr/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?ics1=01&ics2=140&ics3=20&csnumber=53657 (2012-03-10).

Jupp, Simon; Bechofer, Sean; Stevens, Robert (2009). A Flexible API and Editor for SKOS. // Lecture Notes in Computer Science. 5554 (2009) 506-520.

Martínez-González, M. Mercedes; Pérez-León, Beatriz; Alvite-Díez, M. Luisa (2009). SKOS en la integración de conocimiento en los sistemas de información jurídica. // Actas de los Talleres de las Jornadas de Ingeniería del Software y Bases de Datos. 3:6 (2009) 56-62.

McBride, B. (2002). Jena: A Semantic Web Toolkit. // IEEE Internet Computing. 6:6 (2002) 55-59.

Matthews, Brian; Miles, Alistair; Wilson, Michael (2001). Modelling Thesauri for the Semantic Web (Workpackage 8 Deliverable 8.1/8.2 SWAD-Europe). http://www.w3c.rl.ac.uk/SWAD/thesaurus/tif/deliv81/final.html (Archived by WebCite® at http://www.webcitation.org/5m2lmCyQY) (2012-03-10).

Miles, Alistair; Bechofer, Sean (eds.) (2009). SKOS Implementation Report. May 19th 2009. World Wide Web Consortium, 2009. http://www.w3.org/2006/07/SWD/SKOS/reference/20090315/implementation.html (2012-03-10).

Miles, Alistair; Matthews, Brian; Beckett, Dave; Brickley, Dan; Wilson, Michael; Rogers, Nikki (2005). SKOS: A language to describe simple knowledge structures for the web. // XTech 2005: XML, the Web and beyond. Amsterdam (2005). http://ids.snu.ac.kr/w/images/f/f1/SC18.pdf (2012-03-10).

Moya Martínez, G.; Gil Leiva, I. (2001). Evaluación de software de gestión de tesauros. // Ciencias de la Información. 32:3 (2001) 3-23.

Pastor-Sánchez, Juan-Antonio; Martínez Méndez, Francisco Javier; Rodríguez-Muñoz, José Vicente (2009). Advantages of thesaurus representation using the Simple Knowledge Organization System (SKOS) compared with proposed alternatives. // Information Research. 14:4 (Dic. 2009). http://informationr.net/ir/14-4/paper422.html (2012-03-10).

Pastor-Sánchez, Juan-Antonio (2009). Diseño de un sistema colaborativo para la creación y gestión de tesauros en Internet basado en SKOS. Tesis doctoral. Universidad de Murcia (España): Facultad de Comunicación y Documentación.

Pérez-León, Beatriz; Martínez-González, M. Mercedes (2010). A comparative study of thesauri tools: A perspective from Integrability in Information Systems. // Filipe, Joaquim y Cordeiro, José (eds.). Proceedings of the 6th International Conference on Web Information Systems and Technologies, WEBIST 2010. Valencia: INSTICC, 2010. 203-206.

Tudhope, Douglas; Koch, T.; Heery, R. (2006). Terminology Services and Technology: JISC state of the art review. http://www.jisc.ac.uk/media/documents/programmes/capital/terminology_services_and_technology_review_sep_06.pdf (2006) (2012-03-10).

Tuominen, Jouni; Frosterus, Matias; Viljanen, Kim; Hyvönen, Eero (2009). ONKI SKOS Server for Publishing and Utilizing SKOS Vocabularies and Ontologies as Services. // Lecture Notes in Computer Sience. 5554 (2009) 768-780.

Van Assem, Mark; Malaisé, Véronique; Miles, Alistair; Schreiber, Guus (2006). A Method to Convert Thesauri to SKOS. // Lecture Notes in Computer Science. 4011 (2006) 95-109.

Will, L. (2010). Software for Building and Editing Thesauri, http://www.willpowerinfo.co.uk/thessoft.htm (2012-03-10).

World Wide Web Consortium. SPARQL Query Language for RDF. W3C Recommendation 15 January 2008. World Wide Web Consortium, 2008. http://www.w3.org/TR/owl-guide (2012-03-10).

World Wide Web Consortium. SKOS Simple Knowledge Organization System Reference. W3C Recommendation 18 August 2009. World Wide Web Consortium, 2009. http://www.w3.org/TR/2009/REC-skos-reference-20090818/ (2012-03-10).

Word Wide Web Consortium (2012). SKOS/Datasets. http://www.w3.org/2001/sw/wiki/SKOS/Datasets (2012-03-10).

Enviado: 2012-04-04.
Aceptado: 2012-07-05.