

UNIVERSIDAD CARLOS III de MADRID

Máster en Bibliotecas y Servicios de Información Digital



Propuesta de un modelo estructurado de análisis de vocabularios SKOS

Juan Antonio Pastor Sánchez

Octubre 2013

Tutora: Eva María Méndez Rodríguez



ÍNDICE DE CONTENIDO

1. Introducción.....	7
1.1. Justificación.....	7
1.2. Objetivos.....	9
1.3. Metodología.....	10
2. Web Semántica y Linked Open Data.....	11
2.1. Arquitectura global de la Web Semántica.....	12
2.2. Principios de Linked Open Data.....	16
2.3. RDF.....	19
2.4. RDFS y OWL.....	22
2.5. SPARQL.....	26
3. SKOS.....	32
3.1. Descripción general de la ontología SKOS.....	32
3.2. Clases de recursos.....	34
3.3. Etiquetado.....	35
3.4. Documentación.....	37
3.5. Relaciones semánticas.....	39
3.6. Notaciones.....	40
3.7. Relaciones de mapeado.....	41
4. Análisis de vocabularios SKOS.....	43
4.1. Indicadores cuantitativos.....	43
4.2. Condiciones de integridad.....	52
4.3. Criterios de calidad.....	60
5. Modelo de evaluación.....	70
5.1. Grupos funcionales de análisis.....	70
5.2. Niveles de exigencia.....	71
5.3. Estructura del modelo de evaluación.....	71
5.4. Organización de indicadores cuantitativos.....	74
5.5. Puntualizaciones sobre el modelo.....	75
6. Conclusiones y líneas de trabajo futuras.....	77
7. Referencias.....	79
8. Sitios web citados.....	84



Índice de indicadores cuantitativos

[M1] Número de conceptos del vocabulario.....	44
[M2] Número de colecciones del vocabulario.....	44
[M3] Número de esquemas de conceptos del vocabulario.....	44
[M4] Idiomas utilizados y número de etiquetas preferentes por idioma.....	44
[M5] Idiomas utilizados y número de etiquetas alternativas por idioma.....	45
[M6] Idiomas utilizados y número de etiquetas ocultas por idioma.....	45
[M7] Tasa de equivalencia.....	45
[M8] Tasa de conexión.....	46
[M9] Tasa de reciprocidad.....	46
[M10] Tasa de enriquecimiento.....	47
[M11] Tasa de precoordinación.....	47
[M12] Tasa de anotación.....	48
[M13] Tasa de flexibilidad.....	48
[M14] Número de niveles.....	49
[M15] Tamaño de los grupos de conceptos.....	51
[M16] Tasa de ambigüedad.....	51
[M17] Tasa de mapeado.....	52

Índice de definiciones y condiciones de integridad normativas

[S4] Definición: el rango de skos:inScheme es la clase skos:ConceptScheme.....	53
[S5] Definición: el dominio de skos:hasTopConcept es la clase skos:ConceptScheme.....	53
[S6] Definición: el rango de skos:hasTopConcept es la clase skos:Concept.....	53
[S9] Condición de integridad: la clase skos:ConceptScheme es disyunta con skos:Concept	53
[S12] Definición: el rango para las propiedades skos:prefLabel, skos:altLabel y skos:hiddenLabel es un literal plano RDF.....	54
[S13] Condición de integridad: las propiedades skos:prefLabel, skos:altLabel y skos:hiddenLabel son disyuntas entre sí.....	54
[S14] Condición de integridad: un recurso no debe tener más de una etiqueta skos:prefLabel en un mismo idioma.....	54
[S19] Definición: el dominio de skos:semanticRelation es la clase skos:Concept.....	54
[S20] Definición: el rango de skos:semanticRelation is the class skos:Concept.....	55



[S27] Condición de integridad: la propiedad skos:related es disyunta con la propiedad skos:broaderTransitive.....	55
[S31] Definición: el dominio de la propiedad skos:member es la clase skos:Collection.....	56
[S32] Definición: el rango de la propiedad skos:member es la unión de las clases skos:Concept y skos:Collection.....	56
[S33] Definición: el dominio de skos:memberList es la clase skos:OrderedCollection.....	56
[S34] Definición: el rango de skos:memberList es la clase rdf:List.....	57
[S35] Definición: la propiedad skos:memberList es una instancia de la clase owl:FunctionalProperty.....	57
[S37] Definición: la clase skos:Collection es disyunta con las clases skos:Concept y skos:ConceptScheme.....	57
[S46] Condición de integridad: la propiedad skos:exactMatch es disyunta con skos:broadMatch y skos:relatedMatch.....	58
[S48] Definición: la clase skosxl:Label es disyunta con el resto de clases de la recomendación SKOS: skos:Concept, skos:ConceptScheme y skos:Collection.....	58
[S50] Definición: el dominio de skosxl:literalForm es la clase skosxl:Label.....	58
[S51] Definición: el rango de la propiedad skosxl:literalForm es un literal plano RDF.....	58
[S52] Definición: la cardinalidad de la propiedad skosxl:literalForm es exactamente 1.....	59
[S54] Definición: el rango de las propiedades skosxl:prefLabel, skosxl:altLabel y skosxl:hiddenLabel es la clase skosxl:Label.....	59
[S58] Definición: Las propiedades skosxl:prefLabel, skosxl:altLabel y skosxl:hiddenLabel son disyuntas entre sí.....	59
[S60] Definición: el dominio de skosxl:labelRelation es la clase skosxl:Label.....	60
[S61] Definición: el rango de la propiedad skosxl:labelRelation es la clase skosxl:Label.....	60

Índice de criterios de calidad

[C1] Etiquetas de idioma inválidas u omitidas.....	61
[C2] No se han etiquetado todos los literales en todos los idiomas recogidos por el vocabulario.....	61
[C3] Conceptos no documentados.....	62
[C4] Conceptos documentados parcialmente.....	62
[C5] Ausencia de etiquetado relevante.....	63
[C6] Existencia de espacios en blanco innecesarios en las etiquetas.....	63
[C7] Conceptos aislados (“huérfanos”).....	64



[C8] Grupos de conceptos desconectados.....	64
[C9] Relaciones jerárquicas cíclicas.....	64
[C10] Relaciones asociativas sin valor.....	65
[C11] Uso exclusivo de relaciones jerárquicas transitivas.....	65
[C12] Relaciones semánticas unidireccionales.....	66
[C13] Ausencia de conceptos cabecera.....	66
[C14] Conceptos cabecera con conceptos genéricos.....	66
[C15] Conceptos superiores no marcados como conceptos cabecera.....	67
[C16] Colisión de relaciones.....	67
[C17] Ausencia de enlaces entrantes.....	68
[C18] Ausencia de enlaces salientes.....	68
[C19] Enlaces salientes rotos.....	68
[C20] Sintaxis de URIs no válidas.....	68
[C21] Existencia de recursos RDF en el vocabulario con URIs no derreferenciables.....	68
[C22] Uso de elementos de SKOS no válidos.....	68
[C23] Uso de notaciones únicas en un mismo esquema de conceptos.....	69



Resumen

Este trabajo tiene como objetivo definir un modelo estructurado para la evaluación de vocabularios representados mediante SKOS (Simple Knowledge Organization System). Se analizan las principales tecnologías de la Web Semántica que intervienen en este proceso, principalmente RDF y SPARQL, junto con los diferentes elementos, definiciones y condiciones de integridad de la recomendación de SKOS. También se estudian diferentes propuestas de indicadores cuantitativos para vocabularios controlados y criterios de calidad para la evaluación de vocabularios SKOS. Asimismo se indican las oportunas consultas SPARQL y los pasos necesarios para realizar los correspondientes análisis. Como resultado se ha obtenido un modelo de evaluación que incorpora métricas, requisitos normativos y criterios de calidad organizados en una doble agrupación: a partir de facetas funcionales y de niveles de exigencia. El modelo propuesto es de fácil aplicación para el desarrollo de servicios de análisis y validación de vocabularios SKOS y puede ampliarse incorporando nuevos indicadores y criterios de calidad adicionales.

Palabras clave

SKOS; Web Semántica; Evaluación de vocabularios SKOS; SPARQL; Linked Open Data

Abstract

This work aims to define a structured model for evaluating SKOS vocabularies. The main Semantic Web technologies involved in this task, essentially RDF and SPARQL, are analyzed, along with the different elements, definitions and integrity conditions of SKOS recommendation. Several proposals of quantitative indicators and quality criteria for the evaluation of SKOS vocabularies are also studied. Furthermore, we pointed out SPARQL queries, steps and algorithms required to perform an appropriate analysis. The result is an evaluation model that incorporates metrics, normative requirements and quality criteria. This model organizes the analysis elements into a twofold classification: functional aspects and requirement levels. The proposed model is easy to apply for the development of analysis and validation services of SKOS vocabularies. New indicators and additional quality criteria can be added.

Keywords

SKOS; Semantic Web; SKOS vocabularies evaluation; SPARQL; Linked Open Data



1. Introducción.

1.1. Justificación.

La adecuada organización y clasificación de los recursos de información en la Web Semántica supone uno de los núcleos esenciales de su desarrollo. Uno de los aspectos en los que incide especialmente el Informe Final del Grupo Incubadora sobre Datos Enlazados Bibliotecarios (W3C, 2011) es precisamente el referido a los vocabularios controlados. Su importancia radica en la aplicación que se hace los mismos en la clasificación e indización para la descripción de recursos de información de la Web Semántica. En consecuencia se tratan de conjuntos de datos con un gran potencial de reutilización.

Linked Open Data (LOD) es uno de los ejemplos más claros y con mayor recorrido de aplicación de las tecnologías desarrolladas al amparo de la Web Semántica. En este contexto, el potencial de reutilización mencionado anteriormente se asocia directamente al grado de interoperabilidad de las tecnologías utilizadas para la descripción de los recursos de información y su consiguiente representación. De hecho podría decirse que LOD es el frente más activo de la Web Semántica en este sentido (Saorín y Peset, 2013).

En LOD se realiza una aplicación intensiva de taxonomías, clasificaciones y tesauros que aportan un mecanismo para la organización de los recursos, en función de la organización del dominio de conocimiento que representen. En consecuencia se precisan sistemas de organización del conocimiento que conformen una estructura a dos niveles: por un lado los propios recursos interconectados; por otro, vocabularios controlados con elementos que se estructuren semánticamente y que permitan describir el contenido de los recursos descritos. Sin embargo, este segundo nivel no debe concebirse como un conglomerado de vocabularios aislados, puesto que entre los elementos de los mismos es posible establecer conexiones que permitan su reutilización en un entorno distribuido. Así pues, los vocabularios controlados son conjuntos de datos muy valiosos cuya interoperabilidad marca el grado de madurez de LOD.

Una opción podría ser el desarrollo de ontologías especializadas que se ajusten a la realidad concreta de las necesidades de representación de cada vocabulario. Sin embargo, esta alternativa resulta compleja, tanto en el diseño de la ontología, como en su mantenimiento. Además, este enfoque conlleva dificultades para establecer conexiones de mapeado entre los vocabularios, puesto que cada ontología definiría sus propias clases y propiedades cuya compatibilidad no está asegurada. Es más adecuado contar con una propuesta que permita



representar una amplia variedad de casos a partir de elementos comunes.

La ontología más utilizada para el modelado de sistemas de organización del conocimiento en la Web es SKOS (Simple Knowledge Organization System). Se trata de una propuesta muy sencilla que parte de una generalización de este tipo de instrumentos. Posiblemente su sencillez ha facilitado su amplia difusión y aplicación en el modelado de todo tipo de vocabularios controlados como tesauros, clasificaciones, encabezamientos de materia, etc. Actualmente, existen más de medio centenar (Pastor, Martínez y Rodríguez, 2012) de vocabularios controlados representados mediante SKOS y registrados en *DataHub* en forma de LOD. Claros ejemplos de su aplicación los constituyen Europeana, DBpedia e incluso su adopción por parte de sistemas (tal es el caso de Drupal) para representar las taxonomías usadas en la organización de los contenidos de un sitio web..

Así pues, resulta relevante plantear dos cuestiones asociadas a la aplicación y reutilización de conjuntos de datos SKOS: el mantenimiento de dichos conjuntos de datos y el análisis de determinados aspectos que permitan evaluar su coherencia y adecuación semántica para un uso específico. La elección de alguno de estos vocabularios para su aplicación en los procesos de indización y descripción de recursos no es tarea fácil. Por este motivo, hay que considerar varios factores: el dominio de conocimiento abarcado, el nivel de profundidad, la actualización o la coherencia semántica y terminológica. Es preciso realizar determinadas operaciones de análisis destinadas a seleccionar el vocabulario más adecuado para su aplicación en un determinado contexto. La evaluación de vocabularios SKOS también supone un beneficio añadido a los diseñadores y editores de los mismos. Su análisis permite corregir posibles errores y carencias que sean producto de un modelado deficiente. Ofrecer un conjunto de datos SKOS libre de errores es esencial para asegurar su integridad y aplicación.

La recomendación de SKOS (W3C, 2009a) incluye una serie de requisitos que aseguran un nivel mínimo respecto a la consistencia de los vocabularios controlados que hacen uso de esta especificación. Algunos trabajos (Mader, Haslhofer e Isaac, 2012; Suominen y Hyvönen, 2012) proponen determinados criterios de calidad estructurales, de etiquetado, documentación y ciertos aspectos relacionados con la publicación LOD. Tanto la recomendación como las propuestas de criterios de calidad han sido recogidos por el módulo de validación/análisis de vocabularios de la herramienta PoolParty sin que existan detalles técnicos sobre su desarrollo al tratarse de un software propietario. Resulta de gran interés contar con una descripción pormenorizada del proceso de análisis de los requisitos de consistencia y de los criterios de calidad. Esto permitiría implementación de código abierto de herramientas de evaluación de



vocabularios SKOS. Por otra parte, también sería interesante contar con indicadores cuantitativos, que permitan obtener una visión general de la composición estructural y léxica de dichos vocabularios. Sin embargo, no todos los requisitos tienen el mismo nivel de importancia o exigencia en la creación y mantenimiento de vocabularios controlados.

Por todo ello en este trabajo se plantea un modelo para estructurar y cualificar esos requisitos, analizando diferentes aspectos clave en la creación y mantenimiento de esquemas de codificación de vocabularios, junto con distintos niveles de conformidad e indicadores cuantitativos que permita obtener una visión global de su estructura y composición.

1.2. Objetivos

El presente Trabajo Fin de Máster (TFM) se encuadra en el Máster de Bibliotecas y Servicios de Información Digital de la Universidad Carlos III de Madrid y supone la realización por parte del alumno de un proyecto, memoria, estudio o investigación original, en el que proyecte, proponga, aplique o desarrolle los conocimientos y competencias alcanzados en el Máster.

El objetivo general del este TFM es proponer un modelo de análisis y evaluación de vocabularios controlados representados con SKOS, estructurando los criterios de análisis en diferentes niveles de conformidad e incorporando indicadores estadísticos de la estructura semántica y terminológica del tesauro.

Para ello se definen una serie de objetivos específicos:

- Obtener una comprensión actualizada de aquellas tecnologías de la Web Semántica aplicables para la consecución del objetivo general.
- Identificar indicadores estadísticos de la estructura de los vocabularios representados mediante SKOS.
- Identificar los requisitos normativos de integridad que la recomendación SKOS establece para la correcta representación de vocabularios controlados.
- Identificar criterios de análisis de la calidad vocabularios representados con SKOS.
- Proponer un conjunto de facetas en torno a las cuales pueden organizarse los distintos criterios de análisis y los indicadores estadísticos.
- Definir una serie de niveles de conformidad que permita priorizar el cumplimiento de



los criterios de análisis previamente identificados o propuestos.

1.3. Metodología

Durante la realización de este trabajo se ha seguido la siguiente metodología:

- Para la delimitación del marco teórico se ha procedido a la consulta de la literatura básica sobre las principales tecnologías de la Web Semántica y su aplicación en la publicación de conjuntos de datos LOD. Igualmente se ha analizado la correspondiente documentación normativa e informativa sobre SKOS. De forma complementaria también se han consultado las normas ISO 5964:1985, 2788:1986 y 25964-1:2011 sobre tesauros. A continuación se han estudiado las propuestas más relevantes sobre el análisis de vocabularios SKOS basándose en criterios de calidad e integridad, junto con trabajos que proponen medidas estadísticas para el cálculo de indicadores cuantitativos.
- Se han analizado las tecnologías de la Web Semántica más relevantes para el modelo propuesto. Más concretamente se ha realizado un estudio de RDF, RDF Schema, OWL y SPARQL.
- A continuación se ha procedido a una revisión profunda de la recomendación normativa de SKOS publicada por el W3C. En esta revisión se ha incluido el análisis de casos de uso de SKOS y la extensión SKOS-XL. En esta fase, se han identificado las condiciones de integridad de la recomendación de SKOS.
- A partir de la delimitación del marco teórico se ha procedido a extraer o identificar indicadores cuantitativos, definiciones y condiciones de integridad normativas y criterios de evaluación de vocabularios SKOS.
- Finalmente se han clasificado los diferentes criterios en una serie de niveles de conformidad. De este modo se estructura la evaluación de un vocabulario en una serie de pasos progresivos, partiendo desde los requisitos imprescindibles y finalizando en los que aportan valor añadido al vocabulario.



2. Web Semántica y Linked Open Data

La Web Semántica es un término introducido por Berners-Lee, Hendler y Lassila (2001) para referirse a la representación del significado semántico de los datos, para que tanto las personas como las aplicaciones informáticas puedan explotar al máximo la información de la Web.

Bajo la Web Semántica se agrupan múltiples tecnologías, desarrollos y actividades de difusión de una serie de propuestas cuyo objetivo fundamental es la definición y reutilización del significado de los recursos de información disponibles en la web. De este modo las aplicaciones informáticas pueden intercambiar, conectar y procesar información de forma automática a partir de una semántica explícita y bien definida (Hendler, Berners-Lee y Miller, 2002). Resulta esencial el uso de estándares abiertos, puesto que permite definir un nuevo nivel de interoperabilidad, en el que las máquinas realizan de forma más eficiente una serie de procesos cuya misión es localizar y reutilizar la información disponible en recursos ubicados en la Web.

El W3C coordina los trabajos más importantes de desarrollo y despliegue de la Web Semántica. El sitio web que sobre la Web Semántica tiene el W3C indica: *“La Web Semántica proporciona un marco común que permite compartir y reutilizar datos entre aplicaciones, empresas y grupos de usuarios relacionados entre sí. Es un esfuerzo colaborativo, liderado por el W3C con la participación de un gran número de investigadores y socios industriales. Se basa en el uso de RDF.”*

La definición anterior gira en torno a tres ideas: la reutilización de la información, la participación colaborativa de diversos tipos de agentes y el uso de RDF como tecnología fundamental de la Web Semántica.

Halpin y Thompson (2006) apuntan que las propuestas de la Web Semántica son la solución al problema de la integración de datos. El uso de estándares públicos y abiertos aporta un alto grado de transparencia, participación y consenso entre investigadores, comunidades de usuarios, fabricantes y desarrolladores.

Desde el punto de vista de los procesos de descripción y organización del conocimiento la Web Semántica se asienta en el desarrollo y aplicación de esquemas de metadatos y ontologías web. Los esquemas de metadatos utilizados en la Web Semántica suelen ser de tipo descriptivo (Gilliland-Swetland, 2003) y se usan para caracterizar recursos u objetos de información. Las ontologías web, además de dicha función descriptiva, a través de clases y



propiedades, también cumplen una función de formalización lógica mediante la definición de relaciones y restricciones entre recursos.

Cabe señalar la modularización que realiza la Web Semántica entre modelo descriptivo, modelo de datos y formato de serialización. Los modelos descriptivos comprenden metadatos y ontologías, haciendo referencia al conjunto de elementos que permiten describir los atributos y relaciones entre objetos, así como a las posibles condiciones y restricciones que se definan en la asignación de valores a los atributos y en el establecimiento de relaciones entre objetos (Tangmunarunkit, Decker y Kesselman, 2003). El modelo de datos define una serie de elementos y operaciones con el objetivo de ofrecer una herramienta básica para la descripción y representación de objetos abstractos o reales. Por su parte, los formatos son los encargados de aportar la representación sintáctica y por lo tanto son claves para el almacenamiento de la información. Un caso concreto sería el modelo descriptivo de Dublin Core, la representación de descripciones de recursos mediante RDF y el almacenamiento de estos datos utilizando el formato RDF/XML.

Por su parte, la definición de relaciones semánticamente interoperables y formales desde un punto de vista lógico ofrecen dos aspectos de gran interés: la posibilidad de enlazar recursos de conjuntos de datos independientes y la ejecución de inferencias a partir de las relaciones lógicas entre recursos para descubrir información implícita. En términos generales, la interoperabilidad de la Web Semántica se centra en la aplicación de formatos estándares a nivel sintáctico, definir mecanismos homogéneos para la descripción de recursos a nivel estructural y establecer el significado de las propiedades y restricciones desde el punto de vista semántico.

2.1. Arquitectura global de la Web Semántica

Puesto que la Web Semántica es una extensión de la Web convencional, su arquitectura adopta una estructura multinivel en el que las tecnologías de más reciente creación se apoyan sobre otras desarrolladas anteriormente y consolidadas.

La figura 1 muestra la estructura funcional en la que la plataforma web tradicional es la base del desarrollo de la Web Semántica y en el que el protocolo utilizado a nivel de aplicación es HTTP. También se utiliza UNICODE para la transmisión de datos mediante la norma ISO-10646:2003 que permite codificar los caracteres de cualquier alfabeto. En este nivel también resulta clave el uso de referencias URI (Uniform Resource Identifier) para referirse a objetos o recursos de manera unívoca (Berners-Lee, Fielding, R. y Masinter, 2005). Mediante

las referencias URI es posible definir cadenas que definen espacios de identificadores que hacen mención a recursos, ya sean abstractos o estén directamente accesibles en Internet. Las referencias URI se generalizan en forma de IRI (Internationalized Resource Identifier) puesto que este último tipo de identificadores permite utilizar caracteres UNICODE para codificar identificadores de recursos (Duerst y Suignard, 2005).

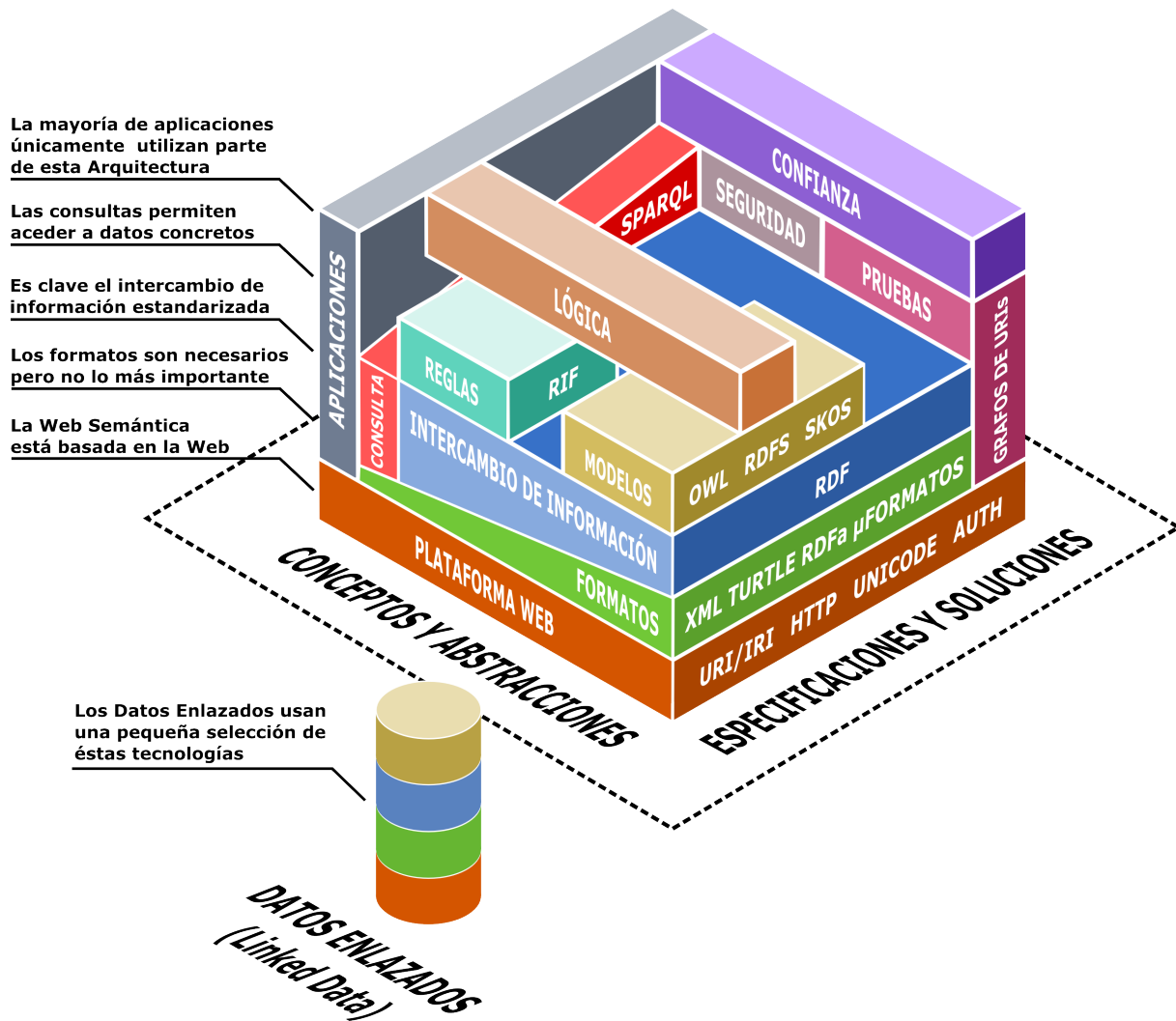


Figura 1. Arquitectura Tecnológica de la Web Semántica. Traducido y adaptado de:
<http://bnode.org/blog/2009/07/08/the-semantic-web-not-a-piece-of-cake>

Es necesario distinguir entre IRI y URL (Uniform Resource Locators) que es aplicable para identificar recursos que además estén disponibles en Internet. Más adelante, en la sección dedicada a LOD, se verá que el uso de identificadores no conlleva necesariamente que estos puedan utilizarse para acceder al recurso. El uso de este tipo de identificadores es muy



importante en Internet, puesto que permite distinguir de un modo sencillo e intuitivo diferentes recursos de forma descentralizada en el contexto de un sistema de naturaleza distribuida.

Los formatos permiten la representación de la información de forma que sea posible su almacenamiento y su consiguiente transmisión y procesamiento en un entorno digital de red. La familia de especificaciones de XML (eXtensible Markup Language) permite la definición de formatos utilizando una sintaxis jerárquica procesable por aplicaciones informáticas. Aunque RDF/XML es tal vez la opción más utilizada en el ámbito de la Web Semántica, otras opciones, como Turtle o Notation3 cada vez son más utilizadas.

Los espacios de nombres XML (XML namespaces) son una parte fundamental de la infraestructura de la Web Semántica. Según la propia recomendación del W3C (W3C, 2009b) un espacio de nombres XML se identifica mediante una referencia URI en el que se definen nombres de elementos y atributos de un determinado documento XML. Siguiendo la misma recomendación, se define el nombre expandido de un elemento o atributo como la URI formada por la unión del espacio de nombres de dicho elemento o atributo junto con el nombre local del mismo.

Por ejemplo, el espacio de nombres de los términos de metadatos de Dublin Core es <http://purl.org/dc/terms/> y el nombre expandido para el elemento `title` de dicho esquema de metadatos sería <http://purl.org/dc/terms/title>.

Esta tecnología es de una importancia vital, puesto que se aplica en el desarrollo de vocabularios descriptivos y en la representación de conjuntos de datos. Además, mediante los espacios de nombres XML es posible que varios desarrolladores utilicen el mismo nombre para representar entidades de vocabularios diferentes. De este modo se resuelven los posibles conflictos entre los nombres de elementos. Cualquier vocabulario de un modelo descriptivo o conjunto de datos se identifica mediante un espacio de nombres XML, que a su vez permite definir los nombres expandidos a partir de una serie de convenciones determinadas por los propios desarrolladores.

Las declaraciones de espacios de nombres permiten al diseñador de conjuntos de metadatos definir el contexto para un elemento con un nombre en particular. Esto asegura que dicho nombre tiene una única definición dentro de los límites del espacio de nombres declarado. Una de las características más importantes de los espacios de nombres XML es que facilitan enormemente la reutilización de elementos o recursos al ofrecer un mecanismo sencillo para



referirse a ellos de forma inequívoca.

El núcleo esencial de la Web Semántica lo constituye RDF (Resource Description Framework). Se trata de un lenguaje que permite realizar descripciones de recursos mediante sentencias compuestas de tres partes: sujeto, predicado y objeto. El sujeto hace referencia al recurso descrito, el predicado es el atributo o relación que sobre dicho recurso se está describiendo y el objeto es el valor del atributo o el recurso con el que se relaciona el recurso sujeto. Estas declaraciones también se denominan tripletas (triplets). Las relaciones entre recursos y la asociación de valores de atributos conforman estructuras de grafos dirigidos. Un grafo dirigido es una estructura, utilizada en matemáticas, que abarca un conjunto de objetos llamados vértices (o nodos) unidos por aristas (o arcos) que apuntan en una determinada dirección.

RDF es muy simple y efectivo en la descripción de recursos de información. Sin embargo, si se desea utilizar RDF para la definición de un vocabulario descriptivo (con clases de recursos, atributos y relaciones personalizadas) es preciso utilizar RDFS (RDF Schema). Mediante RDFS es posible definir jerarquías de clases y propiedades de recursos definiendo estructuras que suelen conocerse como “ontologías ligeras” (Volz, Oberle y Studer, 2003).

OWL (Web Ontology Language) es un lenguaje para la representación de ontologías complejas. Se utiliza conjuntamente con RDF y RDFS y permite definir aspectos semánticos formales de un modo mucho más preciso y detallado: transitividad o reflexividad en las relaciones, definición de clases disyuntas, cardinalidad de relaciones, etc. El soporte a la ejecución de inferencias mediante procesos de razonamiento lógico está garantizado gracias a que OWL establece una serie de equivalencias con operaciones basadas en la lógica de primer orden y la lógica descriptiva. En este sentido, RIF (Rule Interchange Format) establece un formato para el intercambio de reglas que puedan expresarse en diversos lenguajes de representación de reglas lógicas.

Por su parte, SKOS es una ontología OWL, desarrollada por el W3C, para representar cualquier tipo sistema de organización del conocimiento en el ámbito de la Web Semántica mediante RDF. Es posible aplicar SKOS para representar cualquier tipo de vocabulario controlado como clasificaciones, tesauros, encabezamientos de materia, taxonomías, tesauros, etc. Su importancia

Teniendo en cuenta que la Web Semántica utiliza un modelo de representación de datos basado en grafos (RDF) se precisa de un lenguaje de interrogación que se adapte a esta



estructura. Al igual que SQL se utiliza en el modelo relacional, RDF hace uso de SPARQL (SPARQL Protocol and RDF Query Language). Las consultas de SPARQL definen patrones de sentencias RDF para realizar búsquedas en conjuntos de datos. SPARQL también define una serie de operaciones además de un formato XML para devolver los resultados. SPARQL suele aplicarse para recuperar sentencias RDF, disponibles en almacenes RDF (triplestore) generalmente a través de servicios web denominados SPARQL Endpoint que pueden usarse manualmente por personas o directamente por aplicaciones informáticas.

El resto de la arquitectura de la Web Semántica se centra en el uso de mecanismos que permitan comprobar los niveles de confianza y veracidad de los recursos de información recuperados. Para ello se aplica en todo el proceso tecnologías de registro de la procedencia de los conjuntos de datos, firma digital y encriptación de datos que permitan la transmisión segura de los datos y la verificación de su autenticidad.

A lo largo de esta sección se realiza un análisis de RDF, SKOS y SPARQL, puesto que son las principales recomendaciones aplicadas en el presente trabajo. También se realiza un análisis básico de RDFS y OWL con el fin de ofrecer una contextualización más completa.

2.2. Principios de Linked Open Data

LOD hace referencia a un conjunto de técnicas dirigidas a la publicación de datos estructurados e interoperables en la Web. El objetivo de LOD supone la reutilización de datos gestionados por diferentes entidades para su lectura, interpretación y procesado por parte de aplicaciones informáticas. Estos programas, utilizan diversos estándares y formatos abiertos, junto con el protocolo HTTP, para acceder a conjuntos de datos con la finalidad de su reutilización y su análisis para descubrir conexiones con otros conjuntos de datos a partir de los enlaces que se hayan podido definir.

Basándose en la propuesta original realizada por Berners-Lee (2006) y algunos otros autores (Haslhofer y Schandl, 2010) es posible definir una serie de reglas de lo que implica LOD:

1. Utilizar URIs derreferenciables como mecanismo para identificar o acceder a objetos o recursos. Esto es de gran importancia para facilitar la definición de enlaces entre diferentes conjuntos de datos. Un ejemplo de URI derreferenciable sería una URL que además de identificar inequívocamente un recurso también proporciona una dirección para acceder al mismo.
2. A partir del uso de URIs derreferenciables es posible utilizar el protocolo HTTP para



la transmisión de datos entre cliente y servidor, pudiendo indicar el primero el formato en el que éste último debe suministrar los datos. En unos casos el servidor detectará que la petición se está realizando a través de un navegador web, en cuyo caso la información se suministrará al usuario en formato HTML/XHTML. En otros la petición será realizada por una aplicación informática y se utilizará un formato legible por máquina para suministrar los datos. Esta técnica se conoce como negociación de contenido.

3. Utilizar estándares y tecnologías abiertas para acceder a los datos y proceder a su recuperación. Tim Berners-Lee no hace mención explícita a RDF, pero resulta lógico que en la Web Semántica se haga uso de él. En general, este principio abarca cualquier tecnología abierta (SKOS, OWL, SPARQL, etc) dejando de lado los formatos propietarios que puedan dificultar el acceso a los datos.
4. Incluir enlaces a URIs de recursos externos. De esta forma los procesos de publicación de datos son mucho más eficientes. Al reutilizar fuentes externas y conectándolas con las propias se evitan duplicidades o procesos de replicación. Esto permite descubrir nuevos recursos de información externos al conjunto de datos que inicialmente se esté consultando.
5. Es recomendable que las instituciones que publiquen LOD, ofrezcan un SPARQL Endpoint para construir consultas que permitan recuperar, de un modo más selectivo, únicamente aquellos datos que se precisen.
6. Desde el punto de vista de la apertura de los datos, es importante que estos dispongan de una licencia libre que no esté sujeta a ningún tipo de copyright y que permita su reutilización flexible, sin contraprestaciones de ningún tipo y no restrictiva.

De esta forma es posible crear plataformas para compartir y procesar datos de forma interoperable, facilitando la publicación, distribución y conexión de datos. Es preciso tener en cuenta que no necesariamente todos los datos publicados bajo las recomendaciones LOD deben cumplir con todos los principios enumerados anteriormente.

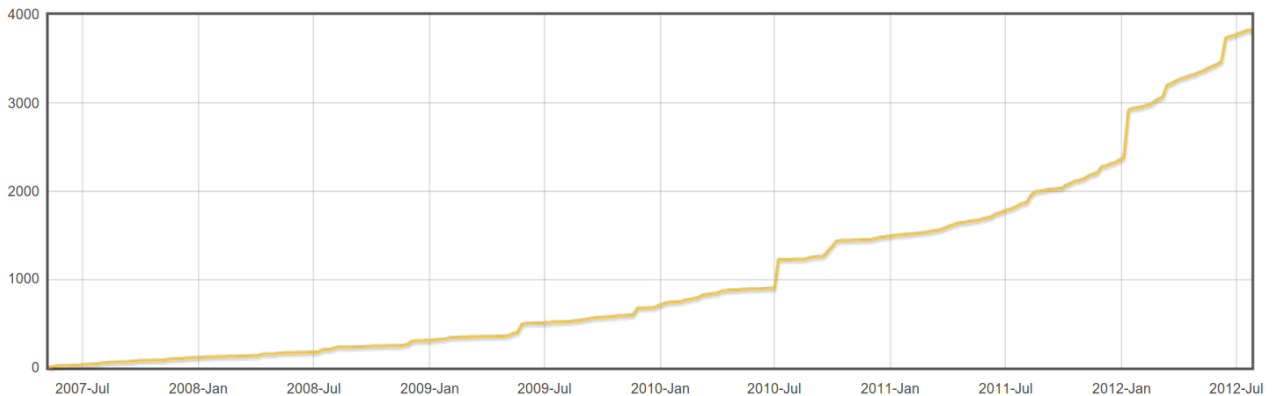
En *Datahub* puede encontrarse un ejemplo de la expansión de LOD. Se trata de un catálogo colaborativo con información descriptiva de conjuntos de datos disponibles en Internet. *Datahub* contiene información sobre la localización, tipo de contenido, forma de acceso a los datos, volumen o relaciones con conjuntos de datos externos entre otras características. A día de hoy este catálogo contiene información descriptiva sobre más de 11.700 conjuntos de datos



y en las estadísticas de este servicio (figura 2) puede verse el crecimiento en la publicación de conjuntos de datos a partir de julio de 2010 y sobre todo en enero de 2012.

Pese a que resulta conveniente seguir los principios LOD para la publicar vocabularios controlados representados con SKOS, no todos los conjuntos de datos asociados a los mismos cumplen con estos principios (Pastor, Martínez y Rodríguez, 2012). En especial la derreferenciación de URIs y el uso de SPARQL facilita enormemente el acceso y la reutilización de estos conjuntos de datos. Concretamente para este trabajo se ha considerado el uso de SPARQL para los procesos de análisis y evaluación.

Total number of Datasets



Revisions to Datasets per week

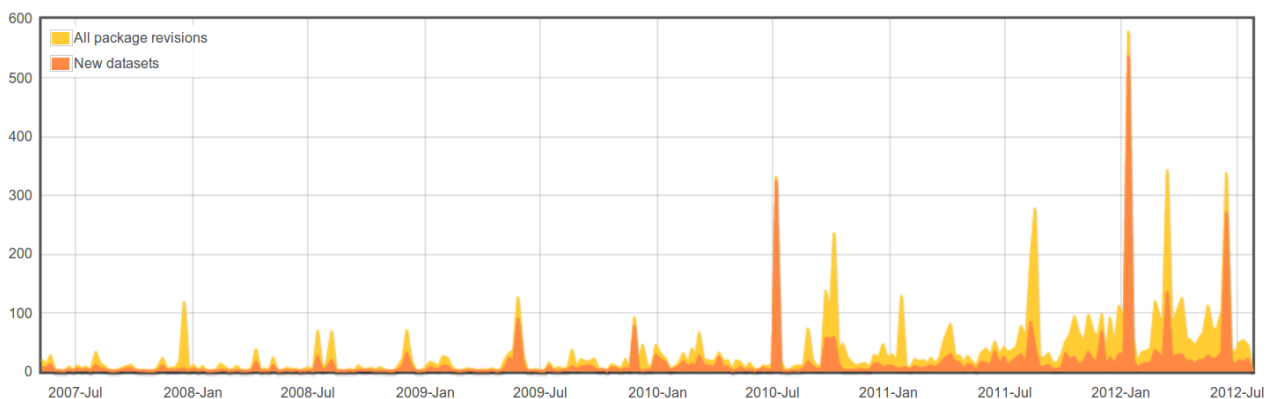


Figura 2. Evolución en el número de datasets (arriba) y número de nuevos datasets y revisiones de descripciones (abajo). Fuente: <<http://web.archive.org/web/20120628023622/http://thedatahub.org/en/stats>>.



2.3. RDF

RDF (Resource Description Framework) es un lenguaje para la descripción de recursos de información en Internet (W3C, 2004a) que aporta una semántica básica de descripción de recursos (W3C, 2004e). Para ello, dichas descripciones toman la forma de tripletas (o sentencias) con tres elementos: sujeto, predicado y objeto. El sujeto es el recurso descrito. El predicado es una propiedad que hace referencia a un atributo de dicho recurso o a una relación del mismo con otro recurso. El objeto es el valor asignado a un atributo o el recurso con el que establece la relación.

RDF supera la estructura jerárquica de XML en favor de un modelo de representación basado en grafos. Además RDF es versátil, simple de entender, escalable e interoperable (W3C, 2004b). Trabajar con RDF supone operar con:

- **Recursos:** cualquier objeto capaz de ser descrito mediante una sentencia RDF: una página web (o un sitio completo), una persona, un lugar, un concepto intangible, etc. No es preciso que el recurso descrito esté accesible a través de la Web. Todos los recursos son identificados mediante una referencia URI aunque no es obligatoria su derreferenciación en una URL.
- **Propiedades:** característica, aspecto, atributo o relación usada para describir un recurso. Cada propiedad tiene un significado específico, define sus valores posibles, los tipos de recursos sobre los que se puede aplicar para su descripción, así como su relación con otras propiedades. Las propiedades forman parte de un determinado vocabulario de descripción y se identifican de forma unívoca mediante una URI.

RDF usa referencias URI y por tanto espacios de nombres XML, tanto para identificar los recursos como los elementos de los vocabularios descriptivos. Esto previene que pueda producirse algún tipo de colisión cuando se combinen múltiples vocabularios RDF, en caso de existir elementos de diferentes ámbitos con el mismo nombre.

Una de las principales características de RDF es su capacidad de representación visual mediante grafos, tal y como se muestra en la figura 3. En este caso, sujeto y objeto adoptan la forma de nodos unidos por un arco dirigido (que representa el predicado) desde el sujeto al objeto.

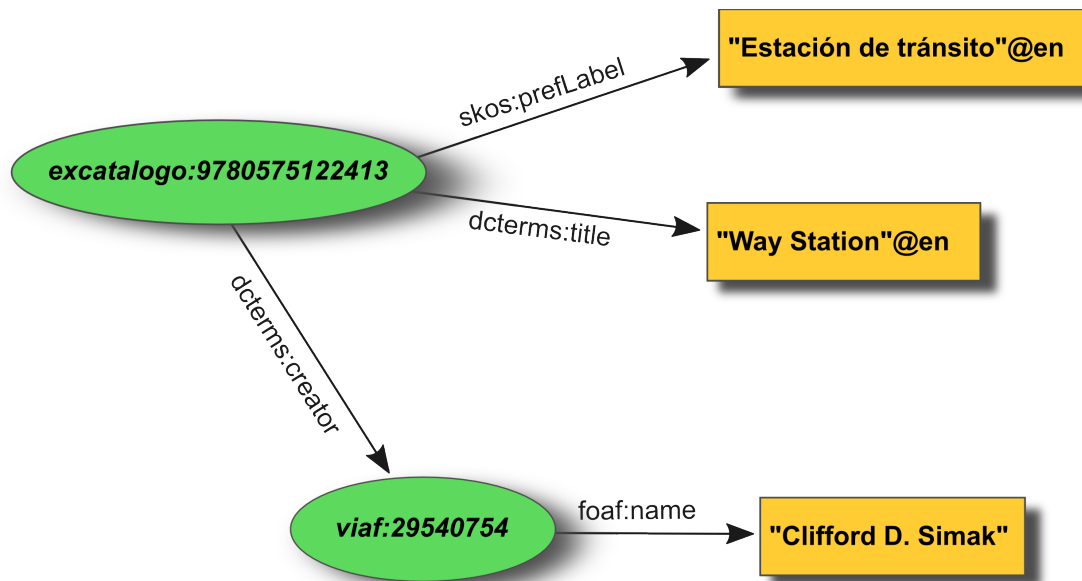


Figura 3. Ejemplo de grafo RDF. Fuente: elaboración propia.

El grafo anterior hace uso de prefijos para abreviar las referencias URI. En la tabla 1 se muestran los prefijos utilizados en el grafo, así como otros que suelen utilizarse de forma común para la descripción de recursos mediante RDF.

Prefijo	Espacio de nombres	Descripción
excatalogo	http://catalogo.ejemplo.org/	Espacio de nombres de un catálogo imaginario, utilizado a modo de ejemplo
foaf	http://xmlns.com/foaf/0.1/	Ontología FOAF (“Fried-of-a-Friend”) que describe propiedades y relaciones de personas, organizaciones, grupos y proyectos.
viaf	http://viaf.org/viaf/	Espacio de nombres para el fichero internacional de autoridades VIAF (“Virtual International Authority File”)
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#	Espacio de nombres en el que se definen todos los elementos de RDF
rdfs	http://www.w3.org/2000/01/rdf-schema#	Espacio de nombres en el que se definen todos los elementos de RDF Schema
dcterms	http://purl.org/dc/terms/	Usado para referenciar elementos de los términos de metadatos de Dublin Core
owl	http://www.w3.org/2002/07/owl#	Elementos de OWL para la definición de ontologías



xsd	http://www.w3.org/2001/XMLSchema#	Definición del esquema XML usado principalmente para definir tipos de datos XML.
skos	http://www.w3.org/2004/02/skos/core#	Espacio de nombres utilizado para referenciar SKOS, de aplicación para representar sistemas de organización del conocimiento.

Tabla 1. Algunos ejemplos (reales y ficticios) de prefijos para espacios de nombres. Fuente: elaboración propia.

En consecuencia, el grafo anterior se serializaría del siguiente modo:

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dcterms="http://purl.org/dc/terms/"
xml:base="http://excatalogo.ejemplo.org/">
  <rdf:Description rdf:about="9780575122413">
    <dcterms:title xml:lang="es">Estación de tránsito</dcterms:title>
    <dcterms:title xml:lang="en">Way Station</dcterms:title>
    <dcterms:creator rdf:resource="http://viaf.org/viaf/29540754" />
  </rdf:Description>
</rdf:RDF>
```

Ejemplo 1. Serialización RDF/XML del grafo de la figura 3. Fuente: elaboración propia.

Otros formatos son menos engorrosos que RDF/XML, como N-Triples, Notation3 o Turtle (Terse RDF Triple Language), ya que ofrecen una sintaxis más compacta y un modo más sencillo para expresar las referencias URI. No obstante, RDF/XML tiene la ventaja de poder transformarse mediante el uso conjunto de XSLT y XPATH (W3C, 1999; Vrandečić et al, 2009). El siguiente ejemplo utiliza Notation3 para representar las mismas sentencias RDF del ejemplo anterior. A lo largo de este TFM se utilizará Notation3 como formato de serialización de RDF en los ejemplos de sentencias RDF.

```
@Prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@Prefix dcterms: <http://purl.org/dc/elements/1.1/> .
@Prefix viaf: <http://www.ejemplo.org/terms/> .
@Prefix excatalogo: <http://excatalogo.ejemplo.org/> .
excatalogo:9780575122413 dcterms:title "Estación de tránsito"@es ;
dcterms:title "Way Station"@en ;
dcterms:creator viaf:29540754 .
```

Ejemplo 2. Serialización en Notation3 del grafo de la figura 2. Fuente: elaboración propia.



Otras características que ofrece RDF son las siguientes (W3C, 2004b):

- **Nodos en blanco:** En RDF es posible representar un recurso que no tenga asociada ninguna referencia URI. Este tipo de elemento de RDF se denomina nodo en blanco (blank node) y los recursos representados de esta forma se denominan recursos anónimos. En consecuencia la identificación y acceso a dichos recursos se restringe únicamente al nivel local del grafo en el que está incluida la sentencia de declaración del nodo en blanco.
- **Literales tipados:** los valores de los atributos que describen un recurso pueden expresarse utilizando un determinado tipo de datos como cadenas de texto, fechas, valores numéricos, etc. El conjunto de tipos de datos disponible viene dado por la recomendación de XML Schema (W3C, 2012b). También es posible indicar el idioma de una cadena literal especificando alguno de los códigos recogidos por la especificación RFC 3066 (Alvestrand, 2001).
- **Reificación:** es posible expresar sentencias sobre otras sentencias RDF. En realidad son “meta-sentencias” que permiten realizar descripciones de aspectos específicos de una tripleta.
- **Contenedores y colecciones:** es posible definir agrupaciones de recursos utilizando una amplia variedad de estructuras del tipo de listas abiertas (ordenadas o no) de recursos o literales, listas de selección para especificar una serie de recursos o literales de los cuales solo podrá asociarse uno mediante una propiedad a un recurso. RDF también permite representar conjuntos cerrados (colecciones en la terminología de RDF) de recursos o literales, ordenados o no, en los que no pueden existir duplicados.

2.4. RDFS y OWL

El lenguaje RDF permite realizar descripciones de recursos web de un modo muy sencillo. Sin embargo, por sí solo no es suficiente para elaborar descripciones más complejas y formales desde un punto de vista lógico. Para un uso avanzado de RDF se precisa la definición de clases y subclases de objetos adaptadas a modelos descriptivos específicos.

Esta función es realizada por RDFS (RDF Schema) que se trata de un lenguaje de definición de vocabularios RDF. Mediante RDFS es posible definir clases y subclases de recursos, así como propiedades y subpropiedades para la descripción de los mismos.



Los principales elementos de RDF/RDFS para la definición de vocabularios RDF son (W3C, 2004d):

- **rdfs:Resource**. Clase utilizada para representar recursos web.
- **rdf:Property**. Clase a partir de la cual es posible definir nuevas propiedades para la descripción de recursos.
- **rdfs:Class**. clase fundamental utilizada para definir otras clases.
- **rdf:type**. Propiedad para establecer la clase a la que pertenece un recurso (instancia de una clase).
- **rdfs:subClassOf**. Propiedad que permite definir jerarquías de clases. Una clase puede ser subclase de otras subclases. Para los recursos cuyo tipo es una determinada subclase, puede inferirse que también son del tipo de la clase superior.
- **rdfs:subPropertyOf**. Propiedad para la definición de jerarquías de propiedades. Una propiedad puede ser subpropiedad de otras propiedades. Para los recursos descritos mediante una supropiedad a la que se ha asignado un determinado valor, puede inferirse que la propiedad superior también tiene asignado dicho valor.
- **rdfs:domain**. Permite definir el dominio de una propiedad, es decir las clases de los recursos sujeto sobre los que puede aplicarse.
- **rdfs:range**. Permite definir el rango de una propiedad, es decir sus posibles valores o clases de los recursos objeto a los que puede hacer referencia.

En el caso de querer definir la jerarquía de clases y subclases Artista → Escritor → Novelista, se podrían utilizar las sentencias RDFS del siguiente ejemplo:

```
ex:Artista rdf:type rdfs:Class .
ex:Escritor rdf:type rdfs:Class .
ex:Novelista rdf:type rdfs:Class .
ex:Escritor rdfs:subClassOf ex:Artista .
ex:Novelista rdfs:subClassOf ex:Escritor .
```

Ejemplo 3. Definición de una estructura de clases mediante RDF Schema.

RDFS proporciona un mecanismo para la definición de ontologías ligeras. Sin embargo, cuando se precisa el uso de una semántica bien definida para el diseño y ejecución de los procesos de inferencia es preciso utilizar OWL (*Web Ontology Language*). Se trata de un lenguaje desarrollado por el W3C que extiende RDFS, cuyo objetivo es publicar y compartir



en la Web ontologías que definen los términos y relaciones de un vocabulario en la Web.

Un análisis detallado de OWL excedería en mucho los objetivos del presente trabajo, sin embargo resulta conveniente ofrecer una visión general de los elementos de modelado que ofrece OWL (W3C, 2012c), y que complementan los que ofrece RDFS:

- Clases primordiales para representar el conjunto de todos los individuos (`owl:Thing`) y el conjunto vacío (`owl:Nothing`).
- Clases y propiedades para la definición de aserciones para definir conjuntos de clases y propiedades por extensión (`owl:members`), igualdad/desigualdad de individuos (`owl:sameAs`, `owl:differentFrom`, `owl:AllDiferent`)
- Propiedades para definir clases a partir de la intersección (`owl:intersectionOf`), unión (`owl:unionOf`) complemento (`owl:complementOf`) de clases o la enumeración de individuos específicos (`owl:oneOf`).
- Propiedades para definir restricciones de asignación de tipos objetos y valores en una propiedad: cuantificadores (`owl:allValuesFrom`, `owl:someValuesFrom`, `owl:hasValue`), reflexividad local (`owl:hasSelf`) y especificaciones de cardinalidad (`owl:cardinality`, `owl:qualifiedCardinality`, `owl:maxCardinality`, `owl:minCardinality`, `owl:maxQualifiedCardinality` y `owl:minQualifiedCardinality`).
- Propiedades para definir axiomas de clases equivalentes (`owl:equivalentClass`) disyuntas (`owl:disjointWith` y `owl:AllDisjointClasses`).
- Clases y propiedades para definir axiomas de expresiones de propiedades, tales como propiedades equivalentes (`owl:equivalentProperty`), disyuntas (`owl:propertyDisjointWith` y `owl:AllDisjointProperties`), inversas (`owl:inverseOf`), funcionales (`owl:FunctionalProperty`, `owl:InverseFunctionalProperty`), transitivas (`owl:TransitiveProperty`), asimétricas/simétricas (`owl:SymmetricProperty`, `owl:AsymmetricProperty`) y reflexivas/irreflexivas (`owl:ReflexiveProperty`, `owl:IrreflexiveProperty`).

OWL se articula en tres sublenguajes con distintos niveles de expresividad adaptados a los requerimientos de diseño de ontologías:

- **OWL Lite:** Orientado a la elaboración de clasificaciones jerárquicas con restricciones



simples, ofreciendo un modo sencillo para representar tesauros, taxonomías y otro tipo de esquemas conceptuales.

- **OWL DL:** Que ofrece una expresividad máxima garantizando que la obtención de resultados en los procesos de inferencia puedan realizarse en un tiempo finito y condicionando el uso de elementos de OWL a la verificación de ciertas restricciones.
- **OWL Full:** Permite la máxima expresividad pero no garantiza que los resultados puedan calcularse en un tiempo finito, haciendo posible incrementar el significado de un vocabulario ya predefinido.

La versión de OWL 2 ha mejorado considerablemente este lenguaje, manteniendo la compabilidad con OWL y añadiendo nuevos elementos basados en la lógica descriptiva. OWL 2 se ofrece en diferentes versiones reducidas que incluyen determinadas restricciones que simplifican la complejidad de cálculo. Dichas versiones se denominan perfiles. La nueva versión, además de OWL 2 Full y OWL 2 DL (similares a la primera versión de OWL) ofrece tres perfiles adicionales (W3C, 2012a):

- **OWL 2 EL:** Aplicable en ontologías que precisen de un bajo nivel de complejidad de cálculo durante los procesos de inferencia. Es útil para representaciones simples y es la opción más eficiente y escalable para ontologías de gran tamaño a costa de la expresividad lógica.
- **OWL 2 QL:** Se usa para ontologías con un gran número de instancias y la ejecución de procesos que requieren la obtención de respuestas a una consulta utilizando motores de razonamiento. Aunque tiene una baja potencia es muy adecuado para representar modelos conceptuales como UML y Entidad-Relación.
- **OWL 2 RL:** Se utiliza en aplicaciones que precisen la ejecución de procesos de inferencia de manera escalable y conservando la mayor parte de la capacidad expresiva de OWL 2 aplicando razonadores y lenguajes de reglas. Sin embargo, no se garantiza que dichos procesos puedan ser computables en la práctica.

La principal ventaja de OWL 2 con respecto a OWL es que las propiedades y perfiles del primero permiten especializar el uso de los elementos de OWL según el tipo de aplicación y el volumen de clases, entidades o sentencias con las que se opere.



2.5. SPARQL

Anteriormente se ha visto que el modelo de representación de datos de RDF se basa en la construcción de grafos dirigidos. Desde el punto de vista del acceso y reutilización a conjuntos de datos, publicados en forma de LOD, bastaría con la obtención completa del mismo para su análisis y la consiguiente extracción de la información deseada. Sin embargo, esta técnica resulta ineficiente para aquellos conjuntos de datos de gran volumen. Resulta mucho más adecuado realizar una búsqueda directamente sobre el conjunto de datos y recuperar únicamente aquellos que cumplan con las condiciones especificadas en la consulta.

Precisamente SPARQL cumple con dicha función. Se trata de un lenguaje de consulta para RDF, al tiempo que un formato y un protocolo tanto para el envío de consultas a un servidor como para los resultados obtenidos. Desde la versión de SPARQL 1.1 también se han desarrollado nuevas funcionalidades para la consulta y actualización de datos, integración con el protocolo HTTP, consultas federadas, procesos de inferencia y se ha ampliado el conjunto de formatos soportados por la recomendación (W3C, 2013a).

SPARQL se basa en la consulta mediante la especificación de patrones de sentencias RDF y en la búsqueda de concordancias de los mismos en un grafo. En SPARQL existen cuatro formas de consulta:

- **SELECT:** permite asignar a una variable el valor de alguno de los elementos (sujeto, predicado u objeto) de aquellas tripletas que cumplan con las condiciones indicadas por el patrón del grafo.
- **CONSTRUCT:** devuelve un grafo (conjunto de sentencias RDF) a partir de una plantilla determinada.
- **DESCRIBE:** permite recuperar las sentencias RDF relativas a un recurso concreto.
- **ASK:** indica si un patrón de consulta tiene solución o no en el grafo y expresa el resultado con un valor booleano (true / false).

Para ilustrar el uso de las diferentes formas de consulta consideremos el siguiente conjunto de sentencias RDF:



```
PREFIX excatalogo: <http://catalogo.ejemplo.org/> .  
PREFIX dcterms: <http://purl.org/dc/terms/> .  
excatalogo:Book003243 dcterms:title "Moby Dick"@en ;  
  dcterms:creator "Herman Melville" .  
excatalogo:Book002678 dcterms:title "Cien Años de Soledad"@es ;  
  dcterms:creator "Gabriel García Márquez"@es .  
excatalogo:Book004458 dcterms:title "Soledades"@ES ;  
  dcterms:creator "Luis de Góngora" .  
excatalogo:Book078091 dcterms:title "La Nuit des temps"@fr ;  
  dcterms:creator "René Barjavel" .  
excatalogo:Book000234 dcterms:title "La vida de Lazarillo de Tormes  
y de sus fortunas y adversidades"@es .
```

Ejemplo 4. Conjunto de datos de ejemplo en Notation3.

Una consulta en un SPARQL para obtener los títulos y el nombre de sus correspondientes autores a partir del conjunto de datos del ejemplo 4 podría ser la siguiente:

```
PREFIX dcterms: <http://purl.org/dc/terms/> .  
SELECT ?titulo ?autor WHERE {  
  ?libro dcterms:title ?titulo ;  
    dcterms:creator ?autor .  
}
```

Ejemplo 5. Consulta SPARQL básica con SELECT.

La consulta anterior incluye una combinación de concordancias múltiples, de forma que los datos recuperados deben estar asociados a sentencias que cumplan ambos patrones de búsqueda. Se identifican aquellas sentencias RDF que cumplan con el primer patrón de búsqueda, en este caso que el predicado sea `dcterms:title`.

A continuación se intenta determinar si además existen otras sentencias con el mismo recurso sujeto pero con el predicado `dcterms:creator`. En caso afirmativo se recuperan los valores objeto de dichas sentencias y se asocian a las variables `?titulo` y `?autor` respectivamente. De este modo, la consulta anterior devuelve el siguiente resultado en formato XML (W3C, 2013b):



```
<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="titulo"/>
    <variable name="autor"/>
  </head>

  <results>

    <result>
      <binding name="titulo">
        <literal xml:lang="en">Moby Dick</literal>
      </binding>
      <binding name="autor">
        <literal>Herman Melville</literal>
      </binding>
    </result>

    <result>
      <binding name="titulo">
        <literal xml:lang="es">Cien Años de Soledad</literal>
      </binding>
      <binding name="autor">
        <literal>Gabriel García Márquez</literal>
      </binding>
    </result>

    <result>
      <binding name="titulo">
        <literal xml:lang="es">Soledades</literal>
      </binding>
      <binding name="autor">
        <literal>Luis de Góngora</literal>
      </binding>
    </result>

    <result>
      <binding name="titulo">
        <literal xml:lang="fr">La Nuit des temps</literal>
      </binding>
      <binding name="autor">
        <literal>René Barjavel</literal>
      </binding>
    </result>

  </results>
</sparql>
```

Ejemplo 6. Resultado de la consulta del ejemplo anterior según la el formato XML para SPARQL.

Puede observarse que dentro de la etiqueta <literal> se incluye el atributo lang para especificar el idioma correspondiente, siempre que se haya especificado en la sentencia RDF. Expresado en forma de tabla los resultados serían los siguientes:



Titulo	autor
Moby Dick	Herman Melville
Cien Años de Soledad	Gabriel García Márquez
Soledades	Luis de Góngora
La Nuit des temps	René Barjavel

Tabla 2. Resultados de la consulta SPARQL del ejemplo 6.

Puede verse que no se recupera el título referido a *El Lazarillo de Tormes* debido a que no cumple la condición del segundo patrón de búsqueda, es decir, que exista una sentencia cuyo predicado se `dcterms:creator`.

También es posible utilizar filtros y funciones que permiten una recuperación más selectiva de las sentencias RDF. La consulta anterior podría incluir un filtro para recuperar únicamente las obras en español con un autor definido:

```
PREFIX dcterms: <http://purl.org/dc/terms/> .
SELECT ?titulo ?autor WHERE {
  ?libro dcterms:title ?titulo ;
        dcterms:creator ?autor .
  FILTER (langMatches(lang(?titulo),"es"))
}
```

Ejemplo 7. Ejemplo de consulta SPARQL con SELECT aplicando un filtro.

Obteniendo el siguiente resultado:

Titulo	autor
Cien Años de Soledad	“Gabriel García Márquez”
Soledades	“Luis de Góngora”

Tabla 3. Resultados de la consulta SPARQL del ejemplo 7.

La función `lang` permite extraer el código del idioma asignado al literal. Por su parte la función `langMatches` permite realizar una comparación que sea insensible a mayúsculas/minúsculas o a variantes regionales de un mismo idioma. En los resultados anteriores se muestra que pese a realizar la comparación con el código de idioma del español en minúscula (“es”) también se recupera el registro correspondiente a “Soledades” etiquetado con el código en mayúscula (“ES”).



En SPARQL es posible utilizar operaciones de agregación clásicas en otros lenguajes de interrogación, como son el cálculo de valores máximos, mínimos, medias aritméticas, recuento de registros recuperados, etc. La agrupación se realiza mediante la expresión GROUP BY y es posible utilizar HAVING para especificar determinadas condiciones que debe cumplir el conjunto de sentencias agrupadas para la selección del valor resultante (de un modo similar a FILTER que opera sobre sentencias no agrupadas). Un ejemplo de uso de estas operaciones podría ser el siguiente:

```
PREFIX dcterms: <http://purl.org/dc/terms/> .  
SELECT (COUNT(?idioma) AS ?total) ?idioma WHERE {  
  ?libro dcterms:title ?titulo ;  
    dcterms:creator ?autor .  
  FILTER (langMatches(lang(?titulo),"es"))  
} GROUP BY (lang(?titulo) AS ?idioma)
```

Ejemplo 8. Ejemplo de consulta SPARQL con SELECT aplicando un filtro y agrupando los resultados.

De este modo se obtendría el número de libros cuyo título está expresado en Español, obteniendo los siguientes resultados:

total	idioma
2	"es"

Tabla 4. Resultados de la consulta SPARQL del ejemplo 8.

La consulta anterior únicamente es posible plantearla a partir de la recomendación SPARQL 1.1 en la que se introduce la posibilidad de vincular el resultado de la evaluación de funciones a una variables específico y las operaciones de agregación antes indicadas.

La forma de consulta CONSTRUCT permite expresar una plantillas para grafos de sentencias RDF a partir de las condiciones expresadas en la consulta. Por ejemplo, para mapear la autoría de las obras del conjunto de datos de ejemplo con el elemento gnd:author de la ontología GND, podría utilizarse la siguiente consulta:

```
PREFIX gnd: <http://d-nb.info/standards/elementset/gnd#> .  
CONSTRUCT ?libro gnd:author ?autor WHERE {  
  ?libro dcterms:creator ?autor .  
}
```

Ejemplo 9. Ejemplo de consulta SPARQL con CONSTRUCT.



Obteniéndose un grafo RDF conformado por las siguientes sentencias:

```
ex:Book003243 gnd:author "Herman Melville" .  
ex:Book002678 gnd:author "Gabriel García Márquez"@es .  
ex:Book004458 gnd:author "Luis de Góngora" .  
ex:Book078091 gnd:author "René Barjavel" .
```

Ejemplo 10. Sentencias RDF obtenidas mediante la consulta del ejemplo 9.

Por su parte, DESCRIBE permite recuperar aquellas sentencias referidas a un determinado recurso, por ejemplo, la consulta:

```
PREFIX excatalogo: <http://catalogo.ejemplo.org/> .  
PREFIX dcterms: <http://purl.org/dc/terms/> .  
DESCRIBE excatalogo:Book003243 .
```

Ejemplo 11. Ejemplo de consulta SPARQL con DESCRIBE.

Recupera las siguientes sentencias RDF:

```
excatalogo:Book003243 dcterms:title "Moby Dick"@en ;  
dcterms:creator "Herman Melville" .
```

Ejemplo 12. Sentencias RDF obtenidas mediante la consulta del ejemplo 10.

Por último, ASK permiten averiguar si existe alguna sentencia RDF (sin que en el resultado se especifique cual) que cumpla el patrón y las condiciones expresadas en la consulta. Para averiguar si existe en el conjunto de datos del catálogo de ejemplo alguna obra de Gabriel García Márquez se utilizaría la siguiente consulta

```
PREFIX excatalogo: <http://catalogo.ejemplo.org/> .  
PREFIX dcterms: <http://purl.org/dc/terms/> .  
ASK {?obra dcterms:creator "Gabriel García Márquez" . }
```

Ejemplo 13. Ejemplo de consulta SPARQL con DESCRIBE.

Obteniendo el resultado yes (verdadero), lo que indica que existe al menos una sentencia RDF que responde a la consulta planteada.

3. SKOS

3.1. Descripción general de la ontología SKOS

SKOS es una ontología OWL Full que proporciona un modelo para la descripción de todo tipo de sistemas de organización del conocimiento. Dentro de este tipo de sistemas pueden encuadrarse tesauros, clasificaciones, encabezamientos de materia, glosarios y cualquier tipo de vocabulario controlado cuyos requisitos de representación sean relativamente sencillos. Los elementos de SKOS definen un conjunto de clases y propiedades para la representación de conceptos, etiquetado terminológico y definición de relaciones semánticas. Además, desde el punto de vista normativo, SKOS presenta numerosas correspondencias con las normas ISO-2788:1986 sobre tesauros monolingües e ISO-5964:1985 de tesauros multilingües.

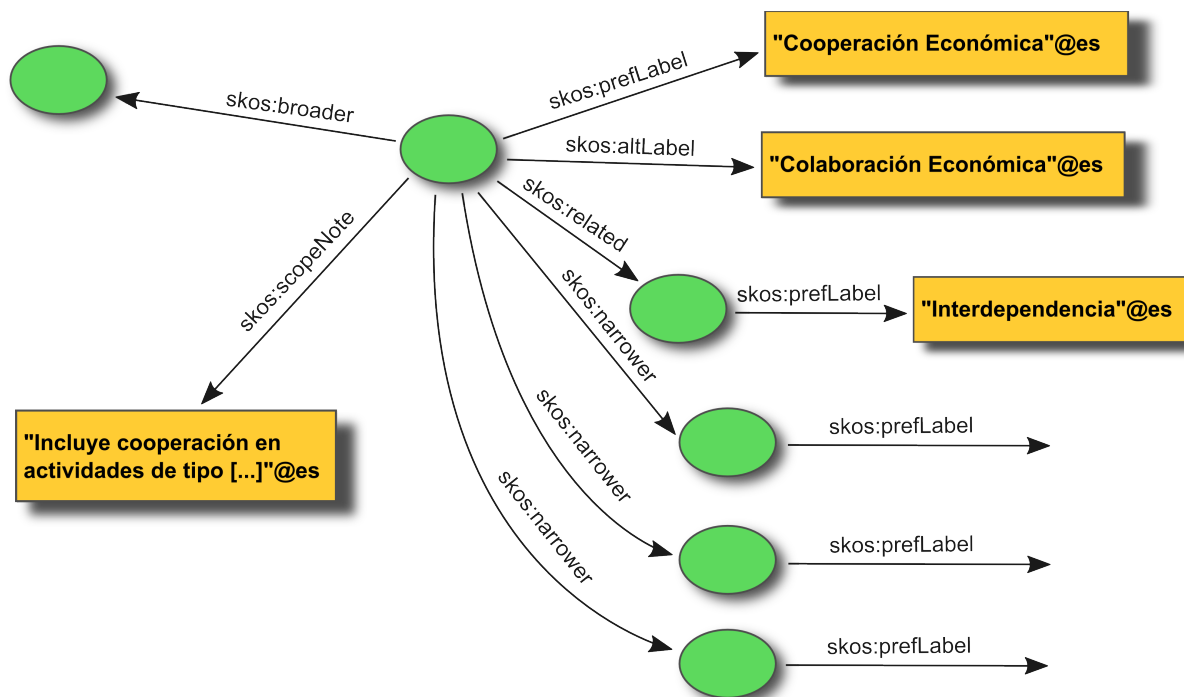


Figura 4: Ejemplo de representación de elementos de un vocabulario con SKOS.

Adaptado de : <[http://www.w3.org/People/Ivan/CorePresentations/SW_Advanced/Slides.html#\(73\)](http://www.w3.org/People/Ivan/CorePresentations/SW_Advanced/Slides.html#(73))>

En SKOS los principales elementos que conforman un vocabulario se definen como conceptos. Entre dichos conceptos es posible establecer relaciones semánticas, tanto asociativas como jerárquicas simples o transitivas. Los conceptos pueden etiquetarse en diferentes idiomas mediante etiquetas preferentes, alternativas y ocultas. Haciendo una

analogía con los tesauros, las etiquetas preferentes se corresponden con los descriptores y las alternativas con los no-descriptores. Por su parte las etiquetas ocultas se asignan pensando en su procesamiento por aplicaciones informáticas y no para su consulta por usuarios. Un concepto únicamente puede asociarse con una etiqueta preferente en un mismo idioma, mientras que puede tener múltiples etiquetas alternativas u ocultas. Existe una extensión de SKOS para definir relaciones entre etiquetas denominada SKOS-XL.

SKOS también ofrece esquemas de conceptos y colecciones. Los esquemas de conceptos suelen identificar el vocabulario en su conjunto o bien un área de conocimiento o campo semántico específico. Es posible definir puntos de acceso a partir de conceptos ubicados en la cabecera de las estructuras jerárquicas. Las colecciones pueden aplicarse para conformar facetas o grupos a partir de criterios semánticos o de otro tipo.

Es posible definir relaciones semánticas de mapeado entre conceptos de diferentes esquemas. Esto permite establecer que un concepto de un vocabulario se considera similar a un concepto de otro vocabulario. Esto es de gran importancia puesto que la nueva norma de tesauros ISO-25964-2:2013 también contempla este tipo de relaciones y por otro lado la publicación LOD precisa de este mecanismo de mapeado entre distintos vocabularios (Méndez y Greenberg, 2012).

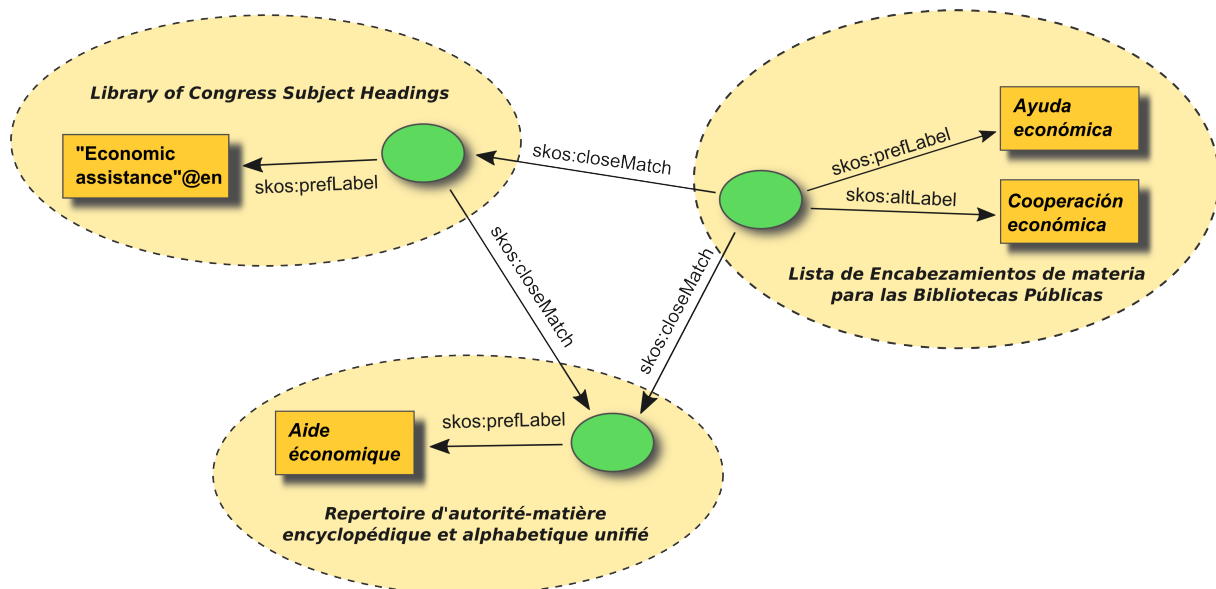


Figura 5: Interoperabilidad de vocabularios entre las LEM, LCSH y RAMEAU. Fuente: elaboración propia.



3.2. Clases de recursos

La recomendación de SKOS establece cuatro clases para la descripción de recursos que se definen como instancias de `owl:Class`:

- **skos:Concept** (conceptos).
- **skos:ConceptScheme** (esquemas de conceptos).
- **skos:Collection** (colecciones).
- **skos:OrderedCollection** (colección ordenada).

Los conceptos son elementos que permiten representar una idea, una noción o una unidad de pensamiento. Es preciso destacar que un concepto es una entidad abstracta independiente del término o términos que se utilicen para etiquetarlo y por lo tanto para expresarlo en uno o varios idiomas. Los esquemas de conceptos y las colecciones permiten representar estructuras de nivel superior y en torno a las cuales se agrupan los conceptos. Un tipo especial de colección es el de las colecciones ordenadas que permiten agrupar conceptos siguiendo una secuencia. De hecho `skos:OrderedCollection` es una subclase de `skos:Collection`.

La recomendación SKOS indica explícitamente que `skos:Concept`, `skos:ConceptScheme` y `skos:Collection` son clases mutuamente excluyentes. Esto significa que un recurso que se defina como un concepto no puede definirse también como un esquema de conceptos o una colección (y a la inversa). Igualmente una colección tampoco puede definirse como un esquema de conceptos.

Las propiedades que definen relaciones entre conceptos, colecciones y esquemas de conceptos son las siguientes:

- **skos:inScheme**. Esta propiedad se suele utilizar para asociar un concepto a un esquema de conceptos. Es una instancia de la clase `owl:ObjectProperty` cuyo rango es `skos:ConceptScheme`. Un aspecto interesante es la indeterminación del dominio, lo que permite cierta flexibilidad para asociar cualquier tipo de recurso a un esquema de conceptos, por lo que en el ámbito de SKOS también podría utilizarse para asociar una colección a un esquema de conceptos.
- **skos:hasTopConcept**. Se utiliza para definir aquellos conceptos que conforman los puntos de acceso a las estructuras jerárquicas de un esquema (conceptos cabecera). Es una instancia de `owl:ObjectProperty` cuyo dominio es `skos:ConceptScheme` y su



rango es `skos:Concept`.

- **skos:topConceptOf**. Se trata de la propiedad inversa (`owl:inverseOf`) de `skos:hasTopConcept` y se utiliza para indicar que un concepto es un concepto cabecera de un esquema. Se trata de una subpropiedad de `skos:inScheme`, aunque en este caso el dominio está definido (`skos:Concept`).
- **skos:member**. Determina los conceptos que conforman una colección. Es una instancia de `owl:ObjectProperty`. Su dominio es `skos:Collection` y su rango es la unión de las clases `skos:Concept` y `skos:Collection`. De este modo es posible definir una colección como miembro de una colección (jerarquías de colecciones).
- **skos:memberList**. Esta propiedad se utiliza exclusivamente para indicar la lista de recursos que conforman una colección ordenada. Por lo tanto, su dominio es `skos:OrderedCollection` y su rango es `rdf:List` (la clase que permite definir una lista de recursos RDF). Es importante señalar que se trata de una propiedad funcional (instancia de `owl:FunctionalProperty`), lo que significa que una colección ordenada únicamente puede asociarse a una única lista de recursos RDF. La recomendación de SKOS también especifica que cualquier elemento incluido en una lista RDF asociada a una colección ordenada puede considerarse miembro de una colección normal.

El siguiente ejemplo ilustra el uso de los elementos de SKOS analizados en esta sección.

```
ex:Esquema rdf:type skos:ConceptScheme ;
  skos:hasTopConcept ex:Concepto_1 .

ex:Concepto_1 rdf:type skos:Concept ;
  skos:inScheme ex:Esquema ;
  skos:topConceptOf ex:Esquema .

ex:Concepto_2 rdf:type skos:Concept ;
  skos:inScheme ex:Esquema ;

ex:Coleccion rdf:type skos:Collection ;
  skos:member ex:Concepto_1, ex:Concepto_2 .
```

Ejemplo 14. Aplicación de principales clases y propiedades de SKOS para la definición de conceptos, esquemas de conceptos y colecciones.

3.3. Etiquetado

En la descripción general del modelo SKOS se ha apuntado que la ontología ofrece diversos



tipos de etiquetado. Más concretamente las propiedades disponibles son:

- **skos:prefLabel**. Para etiquetas preferentes.
- **skos:altLabel**. Para etiquetas alternativas.
- **skos:hiddenLabel**. Para etiquetas ocultas para los usuarios.

Las tres propiedades son instancias de `owl:AnnotationProperty` y subpropiedades de `rdfs:label`. Su rango es la clase `rdfs:Literal`, es decir, cadenas de texto alfanuméricas. Es interesante apuntar que no se define un dominio específico, por lo que las propiedades de anotación de SKOS pueden utilizarse de un modo muy flexible, incluso para anotar recursos que no formen parte de vocabularios controlados representados mediante SKOS. La recomendación SKOS establece que un recurso no puede etiquetarse con más de una etiqueta preferente en un mismo idioma (no existe limitación alguna con las etiquetas alternativas u ocultas). Además también se indica que los valores literales asociados a las etiquetas deben ser disyuntos entre las tres propiedades de etiquetado. Esto significa que una propiedad `skos:prefLabel` no puede tener asociado para un mismo idioma el mismo valor literal que una propiedad `skos:altLabel` o `skos:hiddenLabel` y otro tanto debe cumplirse entre `skos:altLabel` y `skos:hiddenLabel`.

```
ex:Concepto_1 rdf:type skos:Concept ;
skos:prefLabel "Biotecnología"@es ;
skos:prefLabel "Biotechnology"@en ;
skos:prefLabel "Biotechnologie"@fr ;
skos:altLabel "Bioingeniería"@es ;
skos:altLabel "Ingeniería Biológica"@es ;
skos:altLabel "Bioengineering"@en ;
skos:altLabel "Biological engineering"@en ;
skos:altLabel "Bioingénierie"@fr ;
skos:altLabel "Ingénierie génétique"@fr ;
skos:hiddenLabel "Bio-tecnología"@es .
```

Ejemplo 15. Aplicación de principales clases y propiedades de SKOS para el etiquetado de conceptos.

Para establecer relaciones entre etiquetas léxicas debe utilizarse la extensión SKOS-XL. Mientras que las sentencias RDF que hacen uso de las propiedades de etiquetado normales tienen como objeto una cadena de texto literal, aquellas que hacen uso de SKOS-XL tienen como objeto un recurso RDF (nodo). Esto resulta imprescindible, ya que según el modelo RDF es imposible definir relaciones entre valores literales, ya que esto únicamente puede hacerse entre recursos. Partiendo de las consideraciones anteriores se definen los siguientes elementos del vocabulario de SKOS-XL:



- **skosxl:Label**. Es la clase que debe asignarse a los recursos RDF asociados a las etiquetas entre las que se establece algún tipo de relación. Esta clase es mutuamente excluyente con `skos:Concept`, `skos:ConceptScheme` y `skos:Collection`.
- **skosxl:literalForm**. Propiedad mediante la que se expresa el valor literal de una etiqueta SKOS-XL. Es una instancia de `owl:DatatypeProperty` cuyo dominio es `skosxl:Label` y, al igual que con el etiquetado normal de SKOS, su rango es `rdfs:Literal`. Un recurso definido como `skos:Label` deberá tener una y únicamente una relación `skosxl:literalForm` (restricción de cardinalidad exacta 1).
- **skosxl:prefLabel**, **skosxl:altLabel** y **skosxl:hiddenLabel**. Estas propiedades permiten vincular recursos de cualquier tipo (puesto que su dominio no está definido) con otros de la clase `skosxl:Label` que constituye su rango de valores. Cumplen idéntica función que las propiedades equivalentes de SKOS `skos:prefLabel`, `skos:altLabel` y `skos:hiddenLabel`. Igualmente, los valores literales asociados a las propiedades de etiquetado de SKOS-XL han de ser mutuamente excluyentes en un mismo idioma.
- **skosxl:labelRelation**. Mediante esta propiedad es posible definir relaciones entre etiquetas de SKOS-XL. Se trata de una propiedad simétrica definida como instancia de `owl:ObjectProperty` cuyo dominio y rango son los recursos de la clase `skosxl:Label`. Una de las características más interesantes es la posibilidad de definir subpropiedades personalizadas a partir de esta propiedad. El siguiente ejemplo ilustra el uso de esta y otras propiedades de SKOS-XL.

```
@Prefix skosxl: <http://www.w3.org/2008/05/skos-xl#> .
ex:Etiqueta_1 rdf:type skosxl:Label ;
  skosxl:literalForm "Inteligencia Artificial"@es .
ex:Etiqueta_2 rdf:type skosxl:Label ;
  skosxl:literalForm "IA"@es .
ex:acronimo rdfs:subPropertyOf skos:labelRelation .
ex:Etiqueta_1 ex:acronimo ex:Etiqueta_2 .

ex:Concepto_3 rdf:type skos:Concept ;
  skosxl:prefLabel ex:Etiqueta_1 ;
  skosxl:altLabel ex:Etiqueta_2 .
```

Ejemplo 16. Aplicación de principales clases y propiedades de SKOS-XL para el etiquetado de conceptos.

3.4. Documentación

Las propiedades de documentación de SKOS permiten incluir información acerca del



significado de relaciones, conceptos, etiquetas, su evolución a lo largo del tiempo, notas de publicación, ejemplos, notas de alcance y otro tipo de aclaración relevante. SKOS ofrece una tipología que permite cierto grado de formalización. Las distintas propiedades para ello son:

- **skos:note**. Propiedad genérica de documentación a partir del cual se derivan el resto de propiedades de este tipo.
- **skos:changeNote**. Permite documentar los cambios realizados en un recurso durante los procesos de gestión y mantenimiento del vocabulario. Resulta de gran utilidad para registrar los cambios de ubicación de un concepto dentro de la estructura de relaciones.
- **skos:definition**. Propiedad para incluir definiciones.
- **skos:editorialNote**. Permite describir tareas administrativas relacionadas con la edición y publicación del vocabulario.
- **skos:example**. Suministra ejemplos de aplicación de un concepto.
- **skos:historyNote**. Descripción de cambios importantes en el significado de un concepto durante el ciclo de vida de un vocabulario.
- **skos:scopeNote**. Permite incluir información del significado de un concepto en un contexto de uso, por ejemplo, el modo de utilizar un determinado concepto durante la indización.

Todas las propiedades son instancias de `owl:AnnotationProperty`. Un ejemplo de aplicación de `skos:scopeNote` se muestra a continuación.

```
ex:Concepto_3 rdf:type skos:Concept ;
skos:prefLabel "Inteligencia Artificial"@es ;
skos:scopeNote "Capacidad de un instrumento mecánico de efectuar
operaciones normalmente propias de la inteligencia
humana."@es .
```

Ejemplo 17. Aplicación de principales clases y propiedades de SKOS-XL para el etiquetado de conceptos.

Tanto en la descripción como en el ejemplo anteriores se hace referencia al uso de las propiedades de documentación para la descripción de conceptos. Sin embargo, la recomendación de SKOS indica de manera explícita que su uso puede extenderse a cualquier tipo de recurso RDF, puesto que el dominio de las propiedades no está definido. Por otro lado, el hecho de que su rango tampoco lo esté implica que pueden utilizarse indistintamente



literales o URIs de recursos para asignar un valor a dichas propiedades.

3.5. Relaciones semánticas

La recomendación de SKOS define un conjunto de relaciones semánticas que permiten definir vínculos entre conceptos en función de su significado. Todas estas propiedades son instancias de `owl:ObjectProperty`:

- **skos:semanticRelation.** Es la propiedad genérica a partir de la cual se derivan las principales relaciones semánticas de SKOS. El dominio y el rango de esta propiedad es la clase `skos:Concept` y por lo tanto las subpropiedades que se deriven de ella también.
- **skos:broaderTransitive** y **skos:narrowerTransitive.** Permiten representar las relaciones genéricas y específicas respectivamente y crear de esta manera jerarquías de conceptos en función de su significado. Son subpropiedades de `skos:semanticRelation`, inversas entre sí (`owl:inverseOf`) e instancias de `owl:TransitiveProperty`. Esto último significa que si un concepto A es específico de B y otro concepto B es específico de C, entonces se puede inferir que el concepto C es específico de A.
- **skos:broader** y **skos:narrower.** Son subpropiedades de `skos:broaderTransitive` y `skos:narrowerTransitive` respectivamente, aunque en este caso no se tratan de relaciones transitivas, por lo que las sentencias RDF con esta propiedad no pueden utilizarse en procesos de inferencia. Son propiedades inversas entre sí (`owl:inverseOf`).
- **skos:related.** Permite relacionar dos conceptos cuyos significados tengan algún tipo de asociación semántica. Es una subpropiedad de `skos:semanticRelation` y una instancia de `owl:SymmetricProperty`, por lo que si un concepto A está relacionado con un concepto B, entonces se puede inferir que el concepto B está relacionado igualmente con A. Por otro lado se define una condición de integridad que indica que esta propiedad es disyunta con las propiedades de las distintas relaciones semánticas jerárquicas y en consecuencia no deben relacionarse dos conceptos simultáneamente con `skos:related` y `skos:broader/skos:narrower`.



```
ex:Concepto_3 rdf:type skos:Concept ;
skos:prefLabel "Inteligencia Artificial"@es ;
skos:narrower ex:Concepto_4 ;
skos:related ex:Concepto_5 .

ex:Concepto_4 rdf:type skos:Concept ;
skos:prefLabel "Sistema experto"@es ;
skos:broader ex:Concepto_3 .

ex:Concepto_5 rdf:type skos:Concept ;
skos:prefLabel "Cibernética" ;
skos:related "ex:Concepto_3" .
```

Ejemplo 18. Ejemplo de aplicación de las relaciones semánticas jerárquicas y asociativa de SKOS.

Un ejemplo de inconsistencia de la condición de integridad planteada para `skos:related` sería el siguiente:

```
ex:Concepto_6 rdf:type skos:Concept ;
skos:prefLabel "Delfín"@es ;
skos:broader ex:Concepto_7 .

ex:Concepto_7 rdf:type skos:Concept ;
skos:prefLabel "Cetáceos"@es ;
skos:broader ex:Concepto_8 ;
skos:narrower ex:Concepto_6;

ex:Concepto_8 rdf:type skos:Concept ;
skos:prefLabel "Mamíferos"@es ;
skos:narrower ex:Concepto_7 ;
skos:related ex:Concepto_6 .
```

Ejemplo 19. Ejemplo de inconsistencia según la condición de integridad definida para `skos:related`.

3.6. Notaciones

Con SKOS es posible asociar una notación a un concepto, un código que suele identificar un elemento en el vocabulario controlado. Los valores asociados a las notaciones suelen definirse con un literal tipificado que se construye generalmente a partir de tipos de datos definidos en XML Schema. Un concepto puede tener asignadas varias notaciones. Sin embargo, tal como establece la recomendación de SKOS, una notación solo debería ser asignada a un único concepto puesto que en este caso el sistema de notación presentaría ambigüedades.



3.7. Relaciones de mapeado

Uno de los aspectos que definen la publicación LOD es la interconexión de distintos conjuntos de datos. En el caso de SKOS es posible establecer relaciones semánticas entre conceptos de diferentes esquema de conceptos. A este respecto, la segunda parte de la norma ISO-25964 aborda la interoperabilidad de los tesauros con otros vocabularios. Dicha norma establece una serie de equivalencias (Pastor, 2013):

1. Exacta: conceptos que representan exactamente la misma idea. Por ejemplo: “enfermedad de las vacas locas”, “encefalopatía espongiiforme bovina”.
2. Inexacta: conceptos equivalentes en determinados contextos o con diferencias de connotación. Por ejemplo: “sillas” y “asientos”.
3. Parcial: conceptos con un significado muy cercano pero ligeramente más genérico o específico. Por ejemplo: “Aeronave” y “Aeroplanos”.
4. Jerárquica: un concepto es claramente más genérico o específico que otro. Por ejemplo: “Países Bajos” es más genérico que “Holanda”.
5. Asociativas: conceptos no equivalentes pero relacionados entre sí. Por ejemplo: “Fotografías” y “Fotógrafos”.

La misma norma indica que las equivalencias dos conceptos de tesauros diferentes (equivalencia simple) o entre un concepto de un tesoro y varios de otro (equivalencia compuesta), en cuyo caso la equivalencia puede establecerse mediante la intersección o la unión del significado de los conceptos del tesoro de destino. La recomendación actual de SKOS únicamente contempla la posibilidad de definir relaciones de equivalencia simples a partir de las siguientes propiedades que se definen como instancias de owl:ObjectProperty:

- **skos:mappingRelation**. Es una subpropiedad de skos:semanticRelation a partir de la que se derivan el resto de propiedades de mapeado.
- **skos:closeMatch**. Mediante esta propiedad se representan equivalencias inexactas o parciales entre conceptos de diferentes esquemas.
- **skos:exactMatch**. Es una subpropiedad de skos:closeMatch y se utiliza para definir equivalencias exactas entre conceptos. Es una propiedad transitiva (instancia de owl:TransitiveProperty). Es decir con esta propiedad, el concepto <A> se relaciona con y este con <C>, es posible inferir <A> skos:exactMatch <C>.



- **skos:broadMatch** y **skos:narrowMatch**. Son subpropiedades de **skos:broader** y **skos:narrower** respectivamente y se usan para las relaciones de equivalencia jerárquica.
- **skos:relatedMatch**. Esta propiedad se utiliza para representar las equivalencias asociativas y es una subpropiedad de **skos:related**.

Veamos un ejemplo de uso de estas propiedades a partir de la figura 5:

```
@Prefix lem: <http://id.sgcb.mcu.es/Autoridades/> .
@Prefix rameau: <http://stitch.cs.vu.nl/vocabularies/rameau/ark:/> .
@Prefix lcsh: <http://id.loc.gov/authorities/subjects/> .

lem:LEM201001188 rdfs:type skos:Concept ;
  skos:prefLabel "Ayuda económica"@es ;
  skos:altLabel "Cooperación económica"@es ;
  skos:closeMatch rameau:12148/cb13318537z ;
  skos:closeMatch lcsh:sh85040767 .

rameau:12148/cb13318537z rdfs:type skos:Concept ;
  skos:prefLabel "Aide économique"@fr ;

lcsh:sh85040767 rdfs:type skos:Concept ;
  skos:prefLabel "Economic assistance"@en ;
  skos:closeMatch rameau:12148/cb13318537z .
```

Ejemplo 20. Ejemplo de uso de relaciones de mapeado entre conceptos de diferentes vocabularios.

Las propiedades **skos:relatedMatch**, **skos:closeMatch** y **skos:exactMatch** son instancias de **owl:SymmetricProperty**, por lo que a partir de una sentencia RDF que relacione dos conceptos con alguna de estas propiedades se puede inferir otra con la misma propiedad pero permutando los conceptos objeto y sujeto.

La recomendación SKOS también establece que la propiedad **skos:exactMatch** es disyunta con **skos:relatedMatch**, **skos:broadMatch** y **skos:narrowMatch**. Por este motivo, no es posible definir al mismo tiempo entre dos conceptos una relación mediante **skos:exactMatch** y alguna de las otras tres propiedades mencionadas.



4. Análisis de vocabularios SKOS

En esta sección se ofrece una doble perspectiva para la evaluación de vocabularios modelados con SKOS. Por un lado se ofrece un conjunto de indicadores cuantitativos que permiten determinar determinados aspectos estadísticos en relación a la estructura y composición de los vocabularios. Por otro, se muestran una serie de condiciones y recomendaciones asociadas aspectos cualitativos que tratan de estimar tanto la integridad como la calidad de los vocabularios.

En aquellos casos que ha sido posible se han detallado las consultas en SPARQL necesarias para realizar los procesos de recopilación de datos, análisis y evaluación. Cuando ha sido preciso, se han especificado los pasos o algoritmos necesarios para ello.

Se ha evitado el uso de la recomendación SPARQL 1.1, restringiendo la propuesta de consultas a la recomendación de 2008. Sin embargo, en algunos casos no ha sido posible cumplir este criterio, debido a que ha sido necesario utilizar en algún momento los operadores EXIST, HAVING o BIND que únicamente ofrece SPARQL 1.1.

4.1. Indicadores cuantitativos.

La incorporación de indicadores como parte del modelo de evaluación de vocabularios SKOS, parte de la noción de que la estructura y composición de los tesauros, medida en términos cuantitativos de los tesauros, son elementos fundamentales que inciden directamente en los procesos de indización y recuperación de la información (Loukachevitch y Dobrov, 2002).

La evaluación cuantitativa de los lenguajes documentales, en especial de los tesauro, ha sido abordado por diversas propuestas como las de Gil (1998), Lancaster (2002:172-174), Loose (2007), y Martínez et al. (2010 y 2011). Nuestra propuesta reúne algunos de dichos indicadores, adaptándolos al ámbito de SKOS y proponiendo alguna otra medida adicional relacionada con el mapeado de esquemas de conceptos.

Es preciso puntualizar que estas medidas, tal como apunta Lancaster (2001:182-193) pueden resultar subjetivas. En el presente trabajo no se entra a analizar este hecho, sino únicamente se indica como se realizaría su cálculo en vocabularios representados mediante SKOS.



- **[M1] Número de conceptos del vocabulario:** número de recursos de la clase `skos:Concept`. Se obtendría el indicador N_C que se obtiene a partir de la siguiente consulta SPARQL:

```
SELECT COUNT(?concept) AS ?numberconcepts WHERE {  
  GRAPH ?graph { ?concept rdf:type skos:Concept. }  
}
```

- **[M2] Número de colecciones del vocabulario:** número de recursos de la clase `skos:Collection`. Se obtendría el indicador N_{COL} que se obtiene a partir de la siguiente consulta SPARQL:

```
SELECT COUNT(?collection) AS ?numbercollections WHERE {  
  GRAPH ?graph { ?collection rdf:type skos:Collection. }  
}
```

- **[M3] Número de esquemas de conceptos del vocabulario:** número de recursos de la clase `skos:ConceptScheme`. El indicador N_{CS} se determina a partir de la siguiente consulta SPARQL:

```
SELECT COUNT(?conceptscheme) AS ?numberconceptschemes WHERE {  
  GRAPH ?graph { ?conceptscheme rdf:type skos:ConceptScheme. }  
}
```

- **[M4] Idiomas utilizados y número de etiquetas preferentes por idioma** en el etiquetado de conceptos: pares de código de idioma-etiquetas preferentes del idioma. Se expresa como un conjunto de pares $LAB_{PREF} = \{(L_1, PREF_1), (L_2, PREF_2), \dots, (L_n, PREF_n)\}$, donde en cada par L_n se refiere al idioma expresado según lo especificado en la RFC 3066 (Alverstrand, 2001), y $PREF$ es el número de etiquetas preferentes en dicho idioma. El conjunto de pares se recupera con la siguiente consulta SPARQL:

```
SELECT ?idioma (COUNT(?idioma) AS ?pref) WHERE {  
  { ?s rdf:type skos:Concept .  
    ?s skos:prefLabel ?label .  
  } UNION  
  { ?s rdf:type skos:Concept .  
    ?s skosxl:prefLabel ?literal .  
    ?literal skosxl:literalForm ?label .  
  }  
} GROUP BY (lang(?label) AS ?idioma)
```



- **[M5] Idiomas utilizados y número de etiquetas alternativas por idioma** en el etiquetado de conceptos: pares de código de idioma-etiquetas alternativas del idioma. Se expresa como un conjunto de pares $LAB_{ALT} = \{(L_1, ALT_1), (L_2, ALT_2), \dots, (L_n, ALT_n)\}$, donde en cada par L_n se refiere al idioma y ALT es el número de etiquetas alternativas en dicho idioma. El conjunto de pares se recupera con la siguiente consulta SPARQL:

```
SELECT ?idioma (COUNT(?idioma) AS ?pref) WHERE {
  { ?s rdf:type skos:Concept .
    ?s skos:altLabel ?label .
  } UNION
  { ?s rdf:type skos:Concept .
    ?s skosxl:altLabel ?literal .
    ?literal skosxl:literalForm ?label .
  }
} GROUP BY (lang(?label) AS ?idioma)
```

- **[M6] Idiomas utilizados y número de etiquetas ocultas por idioma** en el etiquetado de conceptos: pares de código de idioma-etiquetas ocultas del idioma. Se expresa como un conjunto de pares $LAB_{HID} = \{(L_1, HID_1), (L_2, HID_2), \dots, (L_n, HID_n)\}$, donde en cada par L_n se refiere al idioma y HID es el número de etiquetas ocultas en dicho idioma. El conjunto de pares se obtiene con la siguiente consulta SPARQL:

```
SELECT ?idioma (COUNT(?idioma) AS ?pref) WHERE {
  { ?s rdf:type skos:Concept .
    ?s skos:hiddenLabel ?label .
  } UNION
  { ?s rdf:type skos:Concept .
    ?s skosxl:hiddenLabel ?literal .
    ?literal skosxl:literalForm ?label .
  }
} GROUP BY (lang(?label) AS ?idioma)
```

- **[M7] Tasa de equivalencia:** Tradicionalmente, este indicador establece la relación entre términos descriptores y no-descriptores de un tesoro. Indica en qué medida puede incrementarse el grado de coherencia y precisión del proceso de indización. En el caso de SKOS podría determinarse un indicador similar con respecto a las etiquetas preferentes y etiquetas alternativas, siendo de aplicación para cada idioma. Se define la tasa de equivalencia REQ_n para cada idioma L_n como:

$$REQ_{L_n} = ALT_{L_n} / PREF_{L_n}$$



- **[M8] Tasa de conexión:** Este indicador se basa en la propuesta del Bureau Marcel Van Dijk (Lancaster, 2002) para los tesauros que relaciona el total de descriptores y los descriptores aislados (aquellos sobre los que no se ha definido ninguna relación). Desde el punto de vista de SKOS, la equivalencia con los descriptores aislados serían aquellos conceptos en los que no se ha definido ningún tipo de relación con alguna de las subpropiedades nativas de `skos:semanticRelation`. Estos conceptos, como se verá más adelante, también se denominan conceptos huérfanos. Se define $NORPH_C$ como el número total de conceptos aislados que puede calcularse a partir del recuento de los conceptos obtenidos a partir de la siguiente consulta SPARQL:

```
SELECT ?concept WHERE {
  ?concept rdf:type skos:Concept .
  OPTIONAL {?concept skos:narrower ?n .}
  OPTIONAL {?concept skos:broader ?b .}
  OPTIONAL {?concept skos:related ?r .}
  OPTIONAL {?concept skos:narrowerTransitive ?nt .}
  OPTIONAL {?concept skos:broaderTransitive ?bt .}
  FILTER (!bound(?n) && !bound(?b) && !bound(?r) && !bound(?nt)
  && !bound(?bt)) } GROUP BY ?concept
```

En consecuencia, se define la tasa de conexión como

$$RCON = (N_C - NORPH_C) / N_C$$

- **[M9] Tasa de reciprocidad:** indica la proporción de relaciones inversas o recíprocas con respecto a las relaciones jerárquicas y asociativas definidas. En primer lugar se define el número de relaciones semánticas NR_{bro} , NR_{nar} y NR_{rel} como los totales de relaciones semánticas `skos:broader`, `skos:narrower` y `skos:related` respectivamente, que se calculan mediante las correspondientes consultas SPARQL:

```
SELECT (count(?s) as ?nrbro) WHERE {
  ?s skos:broader ?o . }
```

```
SELECT (count(?s) as ?nrnar) WHERE {
  ?s skos:narrower ?o . }
```

```
SELECT (count(?s) as ?nrrel) WHERE {
  ?s skos:related ?o . }
```

Por su parte, el número de relaciones recíprocas para cada tipo de relación semántica se define como $NREC_{bro}$, $NREC_{nar}$ y $NREC_{rel}$ con sus respectivas consultas SPARQL:



```
SELECT (count(?s) as ?nrecbro) WHERE {  
  ?s skos:broader ?o .  
  ?o skos:narrower ?s . }  
  
SELECT (count(?s) as ?nrecnar) WHERE {  
  ?s skos:narrower ?o .  
  ?o skos:broader ?s . }  
  
SELECT (count(?s) as ?nrecrel) WHERE {  
  ?s skos:related ?o .  
  ?o skos:related ?s . }
```

Y en consecuencia las tasas de reciprocidad para `skos:broader`, `skos:narrower` y `skos:related` se definen como:

$$RREC_{bro} = NREC_{bro} / NR_{bro}$$

$$RREC_{nar} = NREC_{nar} / NR_{nar}$$

$$RREC_{rel} = NREC_{rel} / Nr_{rel}$$

- **[M10] Tasa de enriquecimiento:** en el ámbito de los tesauros este indicador mide la proporción entre el número relaciones jerárquicas y asociativas con el número de descriptores. En SKOS la proporción se establece entre dichas relaciones y el número de conceptos. El número de conceptos N_C viene determinado por [M1], mientras que el número de relaciones jerárquicas y asociativas, NR_{bro} , NR_{nar} y NR_{rel} , son valores obtenidos en [M9]. En consecuencia la tasa de enriquecimiento se calcula de la siguiente forma:

$$RENr = (NR_{bro} + NR_{nar} + NR_{rel}) / N_C$$

- **[M11] Tasa de precoordinación:** en SKOS puede calcularse este indicador para cada idioma dividiendo el número de palabras significativas (no vacías) que componen las etiquetas preferentes en un idioma L_n , NW_{L_n} entre el número de conceptos que componen el vocabulario N_C . Por lo tanto, se define este indicador como:

$$RPRE_{L_n} = NW_{L_n} / N_C$$

La siguiente consulta SPARQL para recuperar la etiquetas preferentes en un determinado idioma, sustituyendo en cada caso el segundo parámetro de la función `langMatches`:



```
SELECT ?label WHERE {  
  {?s skos:prefLabel ?label } UNION  
  {?s skosxl:prefLabel ?literal .  
   ?literal skosxl:literalForm ?label . }  
} FILTER (langMatches(lang(?label),"es"))
```

A partir de los resultados obtenidos se recorre la lista de etiquetas preferentes recuperadas. En cada iteración se segmenta la etiqueta preferente en palabras, y se contabilizan aquellas que no se encuentren en un fichero de palabras vacías específico para cada idioma. El total de palabras significativas de la etiqueta se incrementa hasta obtener el valor de NW_{L_n} .

- **[M12] Tasa de anotación:** esta medida es equivalente a la tasa de definición o notas de aplicación (Gil, 1998). Se define como la relación entre conceptos con algún tipo de anotación y el total de conceptos. En SKOS existen varios tipos de propiedades de anotación. Si se exceptúan aquellas cuya función es la anotación de diferentes aspectos del proceso editorial se tienen las siguientes propiedades: `skos:scopeNote`, `skos:definition` y `skos:example`. El número de conceptos con alguna de estas propiedades de anotación se define como $NCANOT$ y se obtiene a partir de la siguiente consulta:

```
SELECT (count(?s) as ?total) WHERE {  
  { ?s skos:scopeNote ?o . }  
  UNION  
  { ?s skos:definition ?o . }  
  UNION  
  { ?s skos:example ?o . }  
}
```

Y en consecuencia se calcula la tasa de anotación:

$$RANOT = NCANOT / N_C$$

- **[M13] Tasa de flexibilidad:** se define para cada idioma L_n del vocabulario como la proporción entre el número de palabras únicas significativas utilizadas en los descriptores compuestos (precoordinados) NKC_{L_n} que a su vez se encuentran como descriptores o no-descriptores no compuestos (simples) NKS_{L_n} . No debe confundirse NKC_{L_n} con el número de palabras significativas NW_{L_n} de la tasa de precoordinación ([M11]), ya que en este caso se contabilizan palabras únicas. Para realizar este cálculo es necesario recuperar la lista (lista_1) de etiquetas preferentes que contienen al menos una palabra:



```
SELECT ?label WHERE {  
  { ?s skos:prefLabel ?label . } UNION  
  { ?s skosxl:prefLabel ?literal .  
    ?literal skosxl:literalForm ?label . }  
  FILTER (regex(?label, "\s+", "i")) }
```

Igualmente también deben recuperarse las etiquetas preferentes o alternativas que contienen una sola palabra (lista_2)

```
SELECT ?label WHERE {  
  {{ ?s skos:prefLabel ?label . } UNION  
  { ?s skos:altLabel ?tlabel . } UNION  
  { ?s skosxl:prefLabel ?literal .  
    ?literal skosxl:literalForm ?label . } UNION  
  { ?s skosxl:altLabel ?literal .  
    ?literal skosxl:literalForm ?label }}  
  FILTER (regex(?label, "[a-zA-Z0-9]+$", "i"))  
}
```

Teniendo en cuenta las consultas anteriores se procedería del siguiente modo:

1. Recorrer la lista_1 y en cada iteración segmentar la etiqueta preferente analizada en palabras y contabilizar una sola vez aquellas que no se encuentren en el fichero de palabras vacías del idioma. Añadir una sola vez la palabra significativa a la lista_3. Con este paso, tras recorrer la lista_1 se obtiene el valor de NKC_{Ln} .
2. A continuación se recorre la lista_3 (palabras significativas únicas) comprobando en cada iteración si existe alguna etiqueta preferente o alternativa de la lista_2 que contenga la palabra significativa única. Con este paso, tras recorrer la lista_3 se obtiene el valor de NKS_{Ln} .

De este modo se calcula la tasa de flexibilidad:

$$RFLEX = NKS_{Ln} / NKCLn$$

- **[M14] Número de niveles:** este indicador permite establecer el grado de especificidad del vocabulario. Según Gil (1998) es un “valor relativo en función de la especialización de la materia”. Nuestra propuesta determina el número de niveles como el valor máximo de una matriz de distancias entre los conceptos sin relaciones jerárquicas genéricas y los conceptos sin relaciones jerárquicas específicas.

Para ello se crea una lista_1 con los m conceptos que no tienen ningún concepto superior, es decir:



```
SELECT ?concept WHERE {  
  ?concept rdf:type skos:Concept .  
  OPTIONAL {?concept skos:broader ?b .}  
  OPTIONAL {?concept skos:broaderTransitive ?bt .}  
  FILTER (!bound(?b) && !bound(?bt))} GROUP BY ?concept
```

Igualmente se construye una lista_2 con n los conceptos que no tienen ningún concepto específico:

```
SELECT ?concept WHERE {  
  ?concept rdf:type skos:Concept .  
  OPTIONAL {?concept skos:narrower ?b .}  
  OPTIONAL {?concept skos:narrowerTransitive ?bt .}  
  FILTER (!bound(?n) && !bound(?nt))} GROUP BY ?concept
```

La composición de la matriz de distancias $M_{m \times n}$ tiene como filas a los elementos de la lista_1 y como columnas a los de la lista_2. Para cada par de conceptos obtenido por la combinación fila/columna se calcula la correspondiente distancia mediante la siguiente función recursiva, para la que se establecen los valores level=0 y updown=true para la primera iteración:

```
Dist(itemlista_1, itemlista_2, level, updown) {  
  
  Si updown=true ejecutar la sentencia SPARQL  
  SELECT ?concept WHERE {  
    ?concept rdf:type skos:Concept .  
    itemlista_1 ?p ?concept .  
    FILTER (?p=skos:narrower || ?p=narrowerTransitive)  
  }  
  
  Si updown=false ejecutar la sentencia SPARQL  
  SELECT ?concept WHERE {  
    ?concept rdf:type skos:Concept .  
    itemlista_1 ?p ?concept .  
    FILTER (?p=skos:broader || ?p=broaderTransitive)  
  }  
  
  Si no hay resultados finalizar función, devolver int(0)  
  
  Si hay resultados para cada valor de ?concept  
  
    Si ?concept=itemlista_2 finalizar función, devolver int(level+1)  
  
    /* La siguiente condición indica que existe un ciclo en la  
    estructura de relaciones jerárquicas */  
    Si ?concept=itemlista_1 finalizar la función y devolver int(-1)  
  
    Si no se dan las dos condiciones anteriores evaluar  
    Dist(?concept,itemlista_2,level+1,updown)  
}
```



Una vez evaluadas las distancias para todos los pares fila/columna es posible determinar el valor máximo de la matriz, calculando el número de niveles:

$$NLEVEL = \max(M_{m \times n})$$

- **[M15] Tamaño de los grupos de conceptos:** los grupos de conceptos están compuestos por el conjunto de conceptos que forman parte de una misma estructura jerárquica. Requiere el recorrido del grafo RDF partiendo de los conceptos sin relaciones jerárquicas genéricas recogidos en la lista_1 de [M14]. Para cada elemento de dicha lista se evalúa la siguiente función:

```
Cluster(itemlista_1, group) {  
    Ejecutar la sentencia SPARQL  
    SELECT ?concept WHERE {  
        ?concept rdf:type skos:Concept .  
        itemlista_1 ?p ?concept .  
        FILTER (?p=skos:narrower || ?p=narrowerTransitive)  
    }  
    Si no hay resultados finalizar función, devolver array(group)  
    Si hay resultados para cada valor de ?concept  
        Si ?concept no está presente en group añadirlo al array  
        Evaluar Cluster(?concept, group)  
}
```

La función anterior devuelve, para cada concepto TC_n sin relaciones jerárquicas genéricas, un conjunto G_n de conceptos, por lo que el tamaño del grupo correspondiente S es la cardinalidad de dicho conjunto, es decir:

$$S(TC_n) = |G_n|$$

- **[M16] Tasa de ambigüedad:** esta medida indica la proporción entre el número de conceptos ambiguos que carecen de notas de alcance o relaciones jerárquicas, $NAMB$, y el total de conceptos N_C definido en [M1]. $NAMB$ se obtiene mediante el recuento de conceptos obtenidos a través de la siguiente consulta:

```
SELECT ?concept WHERE {  
    ?concept rdf:type skos:Concept .  
    OPTIONAL {?concept skos:narrower ?n .}  
    OPTIONAL {?concept skos:broader ?b .}  
    OPTIONAL {?concept skos:scopeNote ?sn .}  
    FILTER (!bound(?sn) || (!bound(?b) && !bound(?n))) } GROUP  
BY ?concept
```



Se define la tasa de ambigüedad del siguiente modo:

$$RAMB = NAMB / N_C$$

- **[M17] Tasa de mapeado:** este indicador mide la proporción de conceptos que establecen alguna relación de mapeado en relación al total de conceptos del vocabulario. Se define, $NNOTMAP_C$ como el número total de de conceptos sobre los que no se ha definido ninguna relación mediante alguna de la subpropiedades de `skos:mappingRelation` y que se calcula a partir del recuento de los conceptos obtenidos mediante la siguiente consulta SPARQL:

```
SELECT ?concept WHERE {
  ?concept rdf:type skos:Concept .
  OPTIONAL {?concept skos:narrowMatch ?nm .}
  OPTIONAL {?concept skos:broadMatch ?bm .}
  OPTIONAL {?concept skos:relatedMatch ?rm .}
  OPTIONAL {?concept skos:exactMatch ?em .}
  OPTIONAL {?concept skos:closeMatch ?cm .}
  FILTER (!bound(?nm) && !bound(?bm) && !bound(?rm) &&
    !bound(?em) && !bound(?cm)) } GROUP BY ?concept
```

En consecuencia, se define la tasa de mapeado del siguiente modo:

$$RMAP = (N_C - NNOTMAP_C) / N_C$$

4.2. Condiciones de integridad.

Uno de los aspectos esenciales de cualquier vocabulario SKOS es el cumplimiento de las condiciones de integridad de la recomendación de SKOS (W3C, 2009a). Además de dichas condiciones, existen una serie de definiciones de clases y propiedades que también deben verificarse para determinar si un vocabulario hace un uso correcto de la recomendación.

En esta sección se identifican única y explícitamente dichas condiciones y definiciones y en su caso se propone la correspondiente consulta SPARQL para su verificación. Lo realmente interesante no es determinar únicamente cuando un vocabulario cumple con la conformidad con la recomendación SKOS, sino también identificar de forma precisa los elementos del mismo que no lo hacen.

Por este motivo, las consultas propuestas se basan en la forma CONSTRUCT que permite construir grafos para identificar los elementos que incumplen una condición o definición específica. En caso de obtener un grafo vacío el vocabulario analizado cumple con la correspondiente condición o definición.



- **[S4] Definición: el rango de `skos:inScheme` es la clase `skos:ConceptScheme`:** para averiguar los casos en los que se incumple esta definición se construye el correspondiente grafo utilizando la siguiente consulta:

```
CONSTRUCT { ?s skos:inScheme ?o }  
WHERE {  
  ?c skos:inScheme ?o .  
  ?o rdf:type ?type .  
  FILTER (?type!=skos:ConceptScheme)  
}
```

- **[S5] Definición: el dominio de `skos:hasTopConcept` es la clase `skos:ConceptScheme`:** de modo similar a la definición anterior se crea el siguiente grafo:

```
CONSTRUCT { ?s skos:hasTopConcept ?o }  
WHERE {  
  ?s skos:hasTopConcept ?o .  
  ?o rdf:type ?type .  
  FILTER (?type!=skos:ConceptScheme)  
}
```

- **[S6] Definición: el rango de `skos:hasTopConcept` es la clase `skos:Concept`:** de forma similar a [S4] y [S5] se crea el siguiente grafo:

```
CONSTRUCT { ?s skos:hasTopConcept ?o }  
WHERE {  
  ?s skos:hasTopConcept ?o .  
  ?o rdf:type ?type .  
  FILTER (?type!=skos:Concept)  
}
```

- **[S9] Condición de integridad: la clase `skos:ConceptScheme` es disyunta con `skos:Concept`,** es decir, un mismo recurso no puede ser de ambas clases al mismo tiempo. La siguiente consulta permite averiguar qué recursos incumplen esta condición:

```
CONSTRUCT {?s rdf:type skos:Concept .  
  ?s rdf:type skos:ConceptScheme . }  
WHERE {  
  ?s rdf:type skos:Concept .  
  ?s rdf:type skos:ConceptScheme .  
}
```



- **[S12] Definición: el rango para las propiedades `skos:prefLabel`, `skos:altLabel` y `skos:hiddenLabel` es un literal plano RDF** (sin ningún tipo de dato asociado). Se comprueba mediante la siguiente consulta:

```
CONSTRUCT { ?s ?p ?label }
WHERE {
  ?s ?p ?label
  FILTER ((!isLiteral(?label)) &&
    (?p=skos:prefLabel || ?p=skos:altLabel ||
    ?p=skos:hiddenLabel))
}
```

- **[S13] Condición de integridad: las propiedades `skos:prefLabel`, `skos:altLabel` y `skos:hiddenLabel` son disyuntas entre sí.** Es decir, no puede asignarse el mismo literal, en el mismo idioma, a un mismo recurso con más de una de estas propiedades de etiquetado. El grafo adecuado para verificar esta condición se define a partir de la siguiente consulta:

```
CONSTRUCT { ?s ?p1 ?label1 .
  ?s ?p2 ?label2 }
WHERE {
  ?s ?p1 ?label1 .
  ?s ?p2 ?label2 .
  FILTER(lang(?label1)=lang(?label2) && ?label1=?label2 &&
    ((?p1=skos:prefLabel && ?p2=skos:altLabel) ||
    (?p1=skos:prefLabel && ?p2=skos:hiddenLabel) ||
    (?p1=skos:altLabel && ?p2=skos:hiddenLabel)))
}
```

- **[S14] Condición de integridad: un recurso no debe tener más de una etiqueta `skos:prefLabel` en un mismo idioma.**

```
CONSTRUCT { ?s skos:prefLabel ?label1 .
  ?s skos:prefLabel ?label2 . }
WHERE {
  ?s skos:prefLabel ?label1 .
  ?s skos:prefLabel ?label2 .
  FILTER(lang(?label1)=lang(?label2) && ?label1!=?label2)
}
```

- **[S19] Definición: el dominio de `skos:semanticRelation` es la clase `skos:Concept`.** Esto significa que cualquier propiedad definida a partir de `skos:semanticRelation` también tiene el mismo dominio.

Esta comprobación también debe extenderse a las relaciones de mapeado, puesto que `skos:mappingRelation` es una subpropiedad de `skos:semanticRelation`. Para



comprobar esta definición se define el siguiente grafo:

```
CONSTRUCT { ?s ?p ?o .
  ?s rdf:type ?type }
WHERE {
  ?s ?p ?o .
  ?s rdf:type ?type.
  FILTER (?type!=skos:Concept && (?p=skos:broader ||
    ?p=skos:narrower || ?p=skos:related ||
    ?p=skos:broaderTransitive ||
    ?p=skos:narrowerTransitive || ?p=skos:exactMatch ||
    ?p=skos:closeMatch || ?p=skos:broadMatch ||
    ?p=skos:narrowMatch || ?p=skos:relatedMatch ||
    ?p=skos:semanticRelation || ?p=skos:mappingRelation ||
    ?p=skos:exactMatch || ?p=skos:closeMatch ||
    ?p=skos:broadMatch || ?p=skos:narrowMatch ||
    ?p=skos:relatedMatch ))
}
```

- **[S20] Definición: el rango de skos:semanticRelation is the class skos:Concept.**

Al igual que en el caso anterior, esto significa que cualquier propiedad definida a partir de skos:semanticRelation también tiene el mismo rango. Al igual que en el caso anterior también hay que considerar a las relaciones de mapeado. La verificación se realiza mediante el grafo siguiente:

```
CONSTRUCT { ?s ?p ?o .
  ?s rdf:type ?type }
WHERE {
  ?s ?p ?o .
  ?o rdf:type ?type.
  FILTER (?type!=skos:Concept && (?p=skos:broader ||
    ?p=skos:narrower || ?p=skos:related ||
    ?p=skos:broaderTransitive ||
    ?p=skos:narrowerTransitive || ?p=skos:exactMatch ||
    ?p=skos:closeMatch || ?p=skos:broadMatch ||
    ?p=skos:narrowMatch || ?p=skos:relatedMatch ||
    ?p=skos:semanticRelation || ?p=skos:mappingRelation ||
    ?p=skos:exactMatch || ?p=skos:closeMatch ||
    ?p=skos:broadMatch || ?p=skos:narrowMatch ||
    ?p=skos:relatedMatch ))
}
```

- **[S27] Condición de integridad: la propiedad skos:related es disyunta con la propiedad skos:broaderTransitive.** Las disyunción se extiende a las subpropiedades de esta última relación, así como a las propiedades y subpropiedades inversas.

Es decir, tal y como se ha indicado en la sección 3.5, dos conceptos no pueden



relacionarse simultáneamente con `skos:related` y `skos:broader/skos:narrower`.

```
CONSTRUCT { ?s ?p ?o .
  ?s skos:related ?o }
WHERE {
  ?s rdf:type skos:Concept .
  ?s ?p ?o ; skos:related ?o.
  FILTER (?p=skos:narrower || ?p=skos:broader ||
    ?p=skos:narrowerTransitive || ?p=skos:broaderTransitive)
}
```

- [S31] **Definición:** el dominio de la propiedad `skos:member` es la clase `skos:Collection`. La siguiente consulta permite averiguar en que casos se incumpliría esta definición:

```
CONSTRUCT { ?s skos:member ?o .
  ?s rdf:type ?type }
WHERE {
  ?s skos:member ?o .
  ?s rdf:type ?type .
  FILTER (?type!=skos:Collection)
}
```

- [S32] **Definición:** el rango de la propiedad `skos:member` es la unión de las clases `skos:Concept` y `skos:Collection`. Es decir, únicamente los conceptos y las colecciones pueden ser miembros de otras colecciones. Para determinar en qué casos no se cumple esta definición se define el siguiente grafo:

```
CONSTRUCT { ?s skos:member ?o .
  ?o rdf:type ?type }
WHERE {
  ?s skos:member ?o .
  ?o rdf:type ?type .
  FILTER (?type!=skos:Concept && ?type!=skos:Collection)
}
```

- [S33] **Definición:** el dominio de `skos:memberList` es la clase `skos:OrderedCollection`.

```
CONSTRUCT { ?s skos:memberList ?o .
  ?s rdf:type ?type }
WHERE {
  ?s skos:member ?o .
  ?s rdf:type ?type .
  FILTER (?type!=skos:OrderedCollection)
}
```




- **[S34] Definición: el rango de skos:memberList es la clase rdf:List**, es decir, una lista RDF. Esto puede verificarse con la siguiente consulta:

```
CONSTRUCT { ?s skos:memberList ?o .  
             ?o rdf:type ?type }  
WHERE {  
  ?s skos:member ?o .  
  ?o rdf:type ?type .  
  FILTER (?type!=rdf:List)  
}
```

- **[S35] Definición: la propiedad skos:memberList es una instancia de la clase owl:FunctionalProperty**. Un recurso del tipo skos:OrderedCollection puede tener asociada solamente una propiedad skos:memberList. La verificación de esta condición se hace con el siguiente grafo:

```
CONSTRUCT { ?s rdf:type skos:OrderedCollection .  
             ?s skos:memberList ?list1 .  
             ?s skos:memberList ?list2 . }  
WHERE {  
  ?s rdf:type skos:OrderedCollection .  
  ?s skos:memberList ?list1 .  
  ?s skos:memberList ?list2 .  
  FILTER (?list1!=?list2)  
}
```

- **[S37] Definición: la clase skos:Collection es disyunta con las clases skos:Concept y skos:ConceptScheme**. Un recurso definido como colección no puede ser al mismo tiempo un concepto o un esquema de conceptos. Al mismo tiempo hay que tener en cuenta que esta definición también afecta a skos:OrderedCollection:

```
CONSTRUCT { ?s rdf:type ?o1 .  
             ?s rdf:type ?o2 . }  
WHERE {  
  ?s rdf:type ?o1 .  
  ?s rdf:type ?o2 .  
  FILTER ((?o1=skos:Collection || ?o1=skos:OrderedCollection)  
          && (?o2=skos:Concept || ?o2=skos:ConceptScheme))  
}
```



- **[S46] Condición de integridad: la propiedad `skos:exactMatch` es disyunta con `skos:broadMatch` y `skos:relatedMatch`, tal y como se indica en la sección 3.7:**

```
CONSTRUCT { ?s skos:exactMatch ?o .
  ?s ?p ?o . }
WHERE {
  ?s skos:exactMatch ?o .
  ?s ?p ?o .
  FILTER (?p=skos:broadMatch || ?p=skos:relatedMatch)
}
```

- **[S48] Definición: la clase `skosxl:Label` es disyunta con el resto de clases de la recomendación SKOS: `skos:Concept`, `skos:ConceptScheme` y `skos:Collection`.** Con una sola consulta es posible construir el grafo que contiene los recursos que incumplen esta definición:

```
CONSTRUCT { ?s rdf:type skosxl:Label .
  ?s rdf:type ?o . }
WHERE {
  ?s rdf:type skos:Label .
  ?s rdf:type ?o .
  FILTER (?o=skos:Collection || ?o=skos:OrderedCollection ||
    ?o=skos:Concept || ?o=skos:ConceptScheme)
}
```

- **[S50] Definición: el dominio de `skosxl:literalForm` es la clase `skosxl:Label`.**

```
CONSTRUCT { ?s skos:literalForm ?o .
  ?s rdf:type ?type }
WHERE {
  ?s skos:member ?o .
  ?s rdf:type ?type .
  FILTER (?type!=skosxl:Label)
}
```

- **[S51] Definición: el rango de la propiedad `skosxl:literalForm` es un literal plano RDF (sin ningún tipo de dato asociado). Es similar a la definición [S12] pero en el ámbito de SKOS-XL.**

```
CONSTRUCT { ?s ?p ?label .
  ?label skosxl:literalForm ?literal }
WHERE {
  ?s ?p ?label .
  ?label skosxl:literalForm ?literal .
  FILTER (!isLiteral(?literal))
}
```



- **[S52] Definición:** la cardinalidad de la propiedad `skosxl:literalForm` es exactamente 1. Por lo tanto, un recurso de la clase `skosxl:Label` debe tener asociada una y solamente una propiedad `skosxl:literalForm`. Esta comprobación precisa de dos consultas, la primera de ellas para determinar si existe un recurso de la clase `skosxl:Label` sin una propiedad `skosxl:literalForm` asociada:

```
CONSTRUCT { ?s rdf:type skosxl:Label }  
WHERE {  
  ?s rdf:type skosxl:Label .  
  OPTIONAL { ?s skosxl:literalForm ?literal . }  
  FILTER (!bound(?literal))  
}
```

La segunda consulta determina si existe más de una propiedad `skosxl:literalForm` asociada al recurso:

```
CONSTRUCT { ?s rdf:type skosxl:Label .  
  ?s skosxl:literalForm ?literal1 .  
  ?s skosxl:literalForm ?literal2 . }  
WHERE {  
  ?s rdf:type skosxl:Label .  
  ?s skosxl:literalForm ?literal1 .  
  ?s skosxl:literalForm ?literal2 .  
  FILTER (?literal1!=?literal2)  
}
```

- **[S54] Definición:** el rango de las propiedades `skosxl:prefLabel`, `skosxl:altLabel` y `skosxl:hiddenLabel` es la clase `skosxl:Label`. Es similar a la definición de [S12] pero en el ámbito de SKOS-XL y complementando a [S51]:

```
CONSTRUCT { ?s ?p ?label }  
WHERE {  
  ?s ?p ?label  
  FILTER ((?label!=skosxl:Label)) &&  
    (?p=skosxl:prefLabel || ?p=skosxl:altLabel ||  
    ?p=skosxl:hiddenLabel))  
}
```

- **[S58] Definición:** Las propiedades `skosxl:prefLabel`, `skosxl:altLabel` y `skosxl:hiddenLabel` son disyuntas entre sí. Esta definición es similar a [S13] pero aplicada al ámbito de SKOS-XL y en consecuencia la consulta para construir el correspondiente grafo resulta algo más compleja, debido a que la forma literal no se conecta directamente con el concepto, sino que lo hace a través de la propiedad `skosxl:literalForm`:



```
CONSTRUCT { ?s ?p1 ?label1 .
  ?label1 skosxl:literalForm ?literal1 .
  ?s ?p2 ?label2 .
  ?label2 skosxl:literalForm ?literal2 }
WHERE {
  ?s ?p1 ?label1 .
  ?label1 skosxl:literalForm ?literal1 .
  ?s ?p2 ?label2 .
  ?label2 skosxl:literalForm ?literal2 .
  FILTER(lang(?literal1)=lang(?literal2) &&
    ?literal1=?literal2 &&
    ((?p1=skosxl:prefLabel && ?p2=skosxl:altLabel) ||
    (?p1=skosxl:prefLabel && ?p2=skosxl:hiddenLabel) ||
    (?p1=skosxl:altLabel && ?p2=skosxl:hiddenLabel)))
}
```

- **[S60] Definición: el dominio de `skosxl:labelRelation` es la clase `skosxl:Label`.**

```
CONSTRUCT { ?s skosxl:labelRelation ?o .
  ?s rdf:type ?type }
WHERE {
  ?s skosxl:labelRelation ?o .
  ?s rdf:type ?type .
  FILTER (?type!=skosxl:Label)
}
```

- **[S61] Definición: el rango de la propiedad `skosxl:labelRelation` es la clase `skosxl:Label`.**

```
CONSTRUCT { ?s skos:labelRelation ?o .
  ?o rdf:type ?type }
WHERE {
  ?s skosxl:labelRelation ?o .
  ?o rdf:type ?type .
  FILTER (?type!=skosxl:Label)
}
```

4.3. Criterios de calidad

Además de las definiciones y condiciones normativas de la recomendación, es necesario considerar otros aspectos relacionados con la calidad de los vocabularios representados con SKOS. En los últimos años son numerosos los trabajos sobre este tema. Mader y Haslhofer (2013) han definido una serie de criterios de calidad con respecto a los vocabularios SKOS. Dichos criterios se estructuran en una serie de áreas de análisis (Mader, 2012) que han dado lugar a numerosos estudios sobre la aplicación de los mismos en diferentes representaciones de vocabularios SKOS (Mader, Haslhofer e Isaac, 2012; Suominen y Mader, 2013) y como se



obtienen mejores resultados con respecto al modelado y a la aplicación de dichos vocabularios en la recuperación de información (Suominen y Hyvönen, 2012).

A partir de los trabajos mencionados anteriormente se ha procedido a identificar los principales criterios de calidad, al tiempo que se ha descrito el modo en el que desarrollar la evaluación de los mismos, preferentemente mediante consultas SPARQL.

- **[C1] Etiquetas de idioma inválidas u omitidas.** Los vocabularios controlados contienen literales con expresiones en lenguaje natural pero sin incluir la información sobre el idioma o que no corresponde con lo establecido en la especificación RFC3066. Aunque esta comprobación debería extenderse a todas los predicados que sean subpropiedades de `rdfs:label` y `skos:note` se propone la siguiente consulta SPARQL que realiza dicha comprobación únicamente a las propiedades de etiquetado definidas en la recomendación de SKOS con el objeto de no recurrir a un motor de inferencia:

```
CONSTRUCT { ?s ?p ?label }
WHERE {
  ?s ?p ?label .
  FILTER
    (!regex(lang(?label), "[a-z]{2,3}(-[A-Z]{2,3})?$", "i") &&
    (?p=rdfs:label || ?p=skos:prefLabel || ?p=skos:altLabel ||
    ?p=skos:hiddenLabel || ?p=skosxl:literalForm ||
    ?p=skos:note || ?p=skos:scopeNote ||
    ?p=skos:definition || ?p=skos:example ||
    ?p=skos:editorialNote || ?p=skos:changeNote ||
    ?p=skos:historyNote))
}
```

- **[C2] No se han etiquetado todos los literales en todos los idiomas recogidos por el vocabulario.** Se trata de un criterio difícil de evaluar, puesto que es posible que algunos conceptos tengan etiquetas alternativas u ocultas en un idioma que sean innecesarias en otro. Algo similar puede suceder con las propiedades de anotación. Pensamos que la aplicación de este criterio de calidad únicamente puede aplicarse a las etiquetas preferentes. Los pasos para evaluar este criterio son los siguientes:

Paso 1: En primer lugar se determinan los idiomas del vocabulario, tal y como se realiza en el indicador [M4]. De este modo se obtiene el número de idiomas, *NIDIOMAS*, utilizado en el vocabulario.

Paso 2: A continuación se ejecuta la siguiente consulta SPARQL para determinar qué conceptos no han sido etiquetados con todos los idiomas:



```
SELECT ?concept WHERE { {  
  ?concept rdf:type skos:Concept .  
  ?concept skos:prefLabel ?o .  
}} GROUP BY ?concept HAVING (COUNT(?concept)<NIDIOMAS)
```

Paso 3: Se recorre toda la lista de conceptos y se realizan las correspondientes consultas SPARQL recuperando las etiquetas preferentes de los conceptos identificados en el paso anterior y determinando aquellos que no están etiquetados en alguno de los idiomas del vocabulario establecidos en el paso 1.

- **[C3] Conceptos no documentados.** Es conveniente utilizar alguna de las subpropiedades de `skos:note` para documentar convenientemente el vocabulario. Los conceptos sin ningún tipo de anotación podrían localizarse mediante la siguiente consulta:

```
SELECT ?concept WHERE {  
  ?concept rdf:type skos:Concept .  
  OPTIONAL {?concept skos:scopeNote ?sn .}  
  OPTIONAL {?concept skos:definition ?de .}  
  OPTIONAL {?concept skos:example ?ex .}  
  OPTIONAL {?concept skos:editorialNote ?en .}  
  OPTIONAL {?concept skos:historyNote ?hn .}  
  OPTIONAL {?concept skos:changeNote ?cn .}  
  OPTIONAL {?concept skos:note ?no .}  
  FILTER (!bound(?sn) && !bound(?de) && !bound(?ex) &&  
    !bound(?en) && !bound(?hn) && !bound(?cn) &&  
    !bound(?no)) }
```

- **[C4] Conceptos documentados parcialmente.** Es conveniente utilizar como mínimo una vez cada una de las subpropiedades de `skos:note` para documentar completamente el vocabulario. Se trata de un refinamiento del criterio anterior y que es posible evaluar mediante la siguiente consulta:

```
SELECT ?concept WHERE {  
  ?concept rdf:type skos:Concept .  
  OPTIONAL {?concept skos:scopeNote ?sn .}  
  OPTIONAL {?concept skos:definition ?de .}  
  OPTIONAL {?concept skos:example ?ex .}  
  OPTIONAL {?concept skos:editorialNote ?en .}  
  OPTIONAL {?concept skos:historyNote ?hn .}  
  OPTIONAL {?concept skos:changeNote ?cn .}  
  FILTER (!bound(?sn) || !bound(?de) || !bound(?ex) ||  
    !bound(?en) || !bound(?hn) || !bound(?cn)) }
```



- **[C5] Ausencia de etiquetado relevante.** Resulta aconsejable realizar un etiquetado correcto de cada uno de los recursos de un vocabulario: conceptos, colecciones y esquemas de conceptos. El tipo de etiquetado varía en función del tipo de recurso. Los conceptos deben estar etiquetados al menos con una propiedad `skos:prefLabel`, mientras que la práctica más habitual con respecto a las colecciones y esquemas de conceptos es utilizar al menos el elemento `dcterms:title` de Dublin Core. La propuesta de Mader (2012) indica que es suficiente con el uso de alguna subpropiedad de `rdfs:label`, pero pensamos que es necesario utilizar una propiedad relevante de etiquetado o usada de forma habitual tal y como se ha indicado anteriormente. Para la evaluación de este criterio de calidad se plantean tres consultas. La primera de ellas determina los conceptos sin ninguna etiqueta preferente:

```
SELECT ?concept WHERE {  
  ?concept rdf:type skos:Concept .  
  OPTIONAL {?concept skos:prefLabel ?label .}  
  FILTER (!bound(?label)) }
```

La segunda consulta permite localizar las colecciones sin ningún tipo de etiquetado relevante:

```
SELECT ?collection WHERE {  
  ?collection rdf:type ?o .  
  OPTIONAL {?collection skos:prefLabel ?label .}  
  OPTIONAL {?collection dcterms:title ?label .}  
  FILTER ((?o=skos:Collection || ?o=skos:OrderedCollection)  
    && !bound(?label)) }
```

La tercera consulta hace lo propio para los esquemas de conceptos:

```
SELECT ?conceptscheme WHERE {  
  ?conceptscheme rdf:type ?o .  
  OPTIONAL {?conceptscheme skos:prefLabel ?label .}  
  OPTIONAL {?collection dcterms:title ?label .}  
  FILTER ((?o=skos:conceptScheme && !bound(?label)) }
```

- **[C6] Existencia de espacios en blanco innecesarios en las etiquetas.** Es posible que dentro de las etiquetas preferentes, alternativas u ocultas la separación entre palabras se haya realizado con más de un espacio. Incluso es posible que existan espacios al principio o al final del literal de la etiqueta. Esto puede dificultar los procesos de recuperación de información basados en la búsqueda de cadenas de texto en las etiquetas del vocabulario (Suominen y Mader, 2013). Es conveniente identificar dichas



etiquetas y proceder a su corrección en la medida de lo posible. El siguiente grafo contiene dichas etiquetas:

```
CONSTRUCT { ?s ?p ?label } WHERE {  
  ?s ?p ?label .  
  FILTER (regex(?label, "\\s{2,}", "i") && (?p=skos:prefLabel ||  
    ?p=skos:altLabel || ?p=skos:hiddenLabel)) }
```

- **[C7] Conceptos aislados (“huérfanos”).** Su existencia se ha abordado anteriormente en el indicador cuantitativo [M8]. Son conceptos que no tienen relaciones jerárquicas o asociativas con otros conceptos del mismo vocabulario. La consulta SPARQL para hallar estos conceptos es la misma que la utilizada en [M8].
- **[C8] Grupos de conceptos desconectados.** Aunque la literatura citada al principio de esta sección aborda este criterio, no se detalla claramente a qué se refiere dicha desconexión. Una posible interpretación es la existencia de grupos de conceptos que conforman estructuras de relaciones semánticas jerárquicas que carecen de término cabecera. En estos casos el grupo de conceptos no tiene un punto de acceso definido para el inicio de la navegación por la estructura jerárquica. Por lo tanto se establecen los siguientes pasos para la evaluación de este criterio:

Paso 1: determinar los conceptos sin relaciones jerárquicas genéricas (pero con relaciones jerárquicas específicas) que no están definidos como términos cabecera:

```
SELECT ?concept WHERE {  
  ?concept rdf:type skos:Concept .  
  OPTIONAL {?concept skos:narrower ?n .}  
  OPTIONAL {?concept skos:broader ?b .}  
  OPTIONAL {?concept skos:topConceptOf ?t .}  
  FILTER (!bound(?b) && bound(?n) && !bound(?t)) }
```

Paso 2: Utilizar la función Cluster utilizada en [M15] partiendo de los conceptos recuperados en el paso anterior, identificando de esta forma la totalidad de conceptos que forman parte del grupo.

- **[C9] Relaciones jerárquicas cíclicas.** Es posible que durante el proceso de construcción de un vocabulario se defina una estructura jerárquica cíclica: <A> skos:narrower , skos:narrower <C>, <C> skos:narrower <A>. Este tipo de estructuras cíclicas puede tener sentido ascendente (skos:broader) o descendente (skos:narrower). Aunque son consistentes con SKOS reflejan una organización deficiente que puede afectar al proceso de consulta de un vocabulario.



Para localizar estos ciclos se utiliza la *Dist* definida en el indicador cuantitativo [M14], para cada concepto del vocabulario y en ambos sentidos. Aquellos conceptos en los que la función *Dist* devuelve el valor -1 están dentro de un ciclo. De esta forma resulta sencillo ubicar el ciclo dentro del conjunto de relaciones jerárquicas del vocabulario.

- **[C10] Relaciones asociativas sin valor.** Esta situación se produce cuando entre dos conceptos que tienen el mismo concepto genérico o específico (conceptos “hermanos”) se define una relación asociativa. A este respecto la norma ISO-25964-1:2011 indica que “Un par de términos preferentes cuyos significados se solapan a menudo puede ser hermanos, es decir, compartir un término genérico común. Sin embargo, no es necesario interrelacionar todos los términos hermanos de esta forma [mediante una relación asociativa]”. Por este motivo es preciso identificar estas situaciones:

```
CONSTRUCT { ?concept1 ?p ?concept2 .
  ?concept1 skos:related ?concept2 }
WHERE {
  ?concept1 rdf:type skos:Concept .
  ?concept2 rdf:type skos:Concept .
  ?concept1 ?p ?concept2 .
  ?concept1 skos:related ?concept2 .
  FILTER (?p=skos:narrower || ?p=skos:broader ||
    ?p=skos:narrowerTransitive || ?p=skos:broaderTransitive) }
```

- **[C11] Uso exclusivo de relaciones jerárquicas transitivas.** Dichas relaciones únicamente han de utilizarse para la ejecución de inferencias. Para la organización jerárquica de conceptos desde el punto de vista de su significado han de utilizarse las relaciones jerárquicas generales. Es decir, si dos conceptos se relacionan mediante *skos:narrowerTransitive* o *skos:broaderTransitive*, no hay que dejar de hacer lo propio con las correspondientes propiedades *skos:narrower* y *skos:broader*. Para ello se identifican los casos de relaciones transitivas que carecen de relaciones jerárquicas generales mediante la siguiente consulta:

```
CONSTRUCT { ?s1 skos:broaderTransitive ?o1 .
  ?s2 skos:narrowerTransitive ?o2 . } WHERE {
  { ?s1 skos:broaderTransitive ?o1 .
    FILTER NOT EXISTS { ?s1 skos:broader ?o1 . } } UNION
  { ?s2 skos:narrowerTransitive ?o2 .
    FILTER NOT EXISTS { ?s2 skos:narrower ?o2 . } }
  }
```



- **[C12] Relaciones semánticas unidireccionales.** Las relaciones semánticas deben realizarse en ambas direcciones. Es decir, una sentencia $\langle A \rangle$ `skos:broader` $\langle B \rangle$ debe ir acompañada $\langle B \rangle$ `skos:narrower` $\langle A \rangle$. Lo mismo ha de aplicarse a las relaciones asociativas, puesto que tienen carácter simétrico. Para localizar las relaciones semánticas unidireccionales se utiliza el siguiente grafo:

```
CONSTRUCT { ?s1 skos:broader ?o1 .
  ?s2 skos:narrower ?o2 .
  ?s3 skos:related ?o3 } WHERE {
  { ?s1 skos:broader ?o1 .
    FILTER NOT EXISTS { ?o1 skos:narrower ?s1 . } } UNION
  { ?s2 skos:narrower ?o2 .
    FILTER NOT EXISTS { ?o2 skos:broader ?s2 . } } UNION
  { ?s3 skos:related ?o3 .
    FILTER NOT EXISTS { ?o3 skos:related ?s3 . } }
}
```

- **[C13] Ausencia de conceptos cabecera.** Los vocabularios deben proporcionar puntos de acceso a partir de los que es posible consultar la estructura de conceptos y relaciones. En SKOS los esquemas de conceptos han de relacionarse con los conceptos cabecera a través de la relación `skos:hasTopConcept`. La siguiente consulta identifica los esquemas de conceptos que carecen de conceptos cabecera:

```
SELECT ?conceptscheme WHERE {
  ?conceptscheme rdf:type skos:ConceptScheme .
  OPTIONAL { ?conceptscheme skos:hasTopConcept ?concept . }
  OPTIONAL { ?concept skos:topConceptOf ?conceptscheme . }
  FILTER (!bound(?concept))
} GROUP BY ?conceptscheme
```

- **[C14] Conceptos cabecera con conceptos genéricos.** Por definición, un concepto cabecera no debe tener ningún concepto más genérico que él mismo, puesto que son los términos que ocupan el lugar más alto en la estructura de relaciones jerárquicas del vocabulario y por lo tanto son un punto de acceso para la consulta del mismo.

La definición simultánea sobre un concepto de las propiedades `skos:topConceptOf` y `skos:broader` supone una ambigüedad en la organización del vocabulario y puede ocasionar problemas durante su consulta. Este hecho se determina con el siguiente grafo:



```
CONSTRUCT { ?concept skos:topConceptOf ?conceptscheme .  
  ?concept skos:broader ?broader } WHERE {  
  ?concept rdf:type skos:Concept .  
  ?concept skos:topConceptOf ?conceptscheme .  
  ?concept skos:broader ?broader .  
}
```

- **[C15] Conceptos superiores no marcados como conceptos cabecera.** Los conceptos no huérfanos con algún concepto inferior, pero sin concepto superior alguno, son en realidad conceptos cabecera y deben definirse expresamente como tales. Este problema está relacionado directamente con [C8]. La lista de dichos conceptos que se encuadran en esta situación se obtiene mediante la consulta:

```
SELECT ?s WHERE {  
  ?s rdf:type skos:Concept .  
  ?s skos:narrower ?narrower .  
  OPTIONAL { ?s skos:broader ?broader . }  
  OPTIONAL { ?s skos:topConceptOf ?conceptscheme . }  
  FILTER (!bound(?conceptscheme) && !bound(?broader))  
} GROUP BY ?s
```

- **[C16] Colisión de relaciones.** Tal y como indican Suonimen y Mader (2013), entre dos conceptos que forman parte de un grupo de conceptos relacionados jerárquicamente no debe establecerse una relación asociativa. En realidad se trata de una generalización de la condición de integridad [S27]. La comprobación de la existencia de tales colisiones se realiza en tres pasos:

Paso 1: Se recuperan todos los pares de conceptos entre los que se ha establecido una relación asociativa mediante la consulta:

```
SELECT ?s ?o WHERE {  
  ?s skos:related ?o .  
}
```

Paso 2: Se unifica la lista de pares, eliminando los pares inversos que se han recuperado debido a que `skos:related` es una relación semántica simétrica.

Paso 3: A continuación para cada par de conceptos y en ambas direcciones (ascendente y descendente) se evalúa la función *Dist*. En los casos en los que se obtenga un valor distinto de cero se determina que existe una ruta jerárquica que conecta ambos conceptos y por lo tanto una colisión con la relación asociativa.



- **[C17] Ausencia de enlaces entrantes.** Este criterio de calidad se relaciona con la visibilidad de un vocabulario controlado SKOS en el ámbito LOD. Debido a que estos enlaces se establecen desde conjuntos de datos externos no es posible utilizar los datos del propio vocabulario para determinar su ausencia o existencia. Es preciso utilizar servicios como SINDICE para buscar sentencias RDF cuyos objetos sean referencias URI que contengan el espacio de nombres del vocabulario controlado.
- **[C18] Ausencia de enlaces salientes.** Resulta de gran valor definir conexiones con conjuntos de datos externos. Es posible determinar si un conjunto de datos tiene enlaces salientes buscando sentencias RDF cuyos objetos sean referencias URI que NO contengan el espacio de nombres del vocabulario controlado.
- **[C19] Enlaces salientes rotos.** Una vez localizados los enlaces salientes a partir de [C18] es posible realizar peticiones HTTP a las referencias URI identificadas. Para las conexiones en las que sea imposible obtener como respuesta el código de estado “200 OK” (tras resolver las respuestas “303 See other”) se podrá afirmar que se trata de un enlace roto.
- **[C20] Sintaxis de URIs no válidas.** Para los enlaces salientes rotos identificados en [C19] es posible determinar si la imposibilidad de la conexión se ha debido a una sintaxis incorrecta de la URI, simplemente comprobando las indicaciones de la especificación RFC3986 (Berners-Lee, Fielding y Masinter, 2005).
- **[C21] Existencia de recursos RDF en el vocabulario con URIs no derreferenciables.** Para que un vocabulario representado con SKOS pueda incorporarse al entorno LOD es preciso que los diferentes recursos posean referencias URI que puedan utilizarse para acceder a los mismos. Los recursos que no cumplen dicha condición se identifican mediante la siguiente consulta SPARQL:

```
SELECT ?s WHERE {  
  ?s ?p ?o .  
  FILTER (!regex(?s,"^(http|https)://","i"))  
} GROUP BY ?s LIMIT 10
```

- **[C22] Uso de elementos de SKOS no válidos.** La recomendación de SKOS establece un conjunto de clases y propiedades dentro de un espacio de nombres determinado. Sin embargo, es posible que en algún momento un vocabulario representado con SKOS utilice elementos que no se atengan a la recomendación. Esto puede ocurrir si el editor del vocabulario define elementos personalizados utilizando el mismo espacio de



nombres de SKOS (<http://www.w3.org/2004/02/skos/core#>) o de SKOS-XL (<http://www.w3.org/2008/05/skos-xl#Label>) o si utiliza elementos obsoletos previos a la recomendación definitiva. Para comprobar esto únicamente hay que recuperar las sentencias cuyo predicado u objeto haga referencia a una URI que contenga el espacio de nombres de SKOS o de SKOS-XL y comprobar si en todos los casos se hace referencia a un elemento válido de la recomendación.

- **[C23] Uso de notaciones únicas en un mismo esquema de conceptos.** La recomendación SKOS indica que dos conceptos de un mismo esquema no deben tener la misma notación. Para verificar este criterio se construye el siguiente grafo:

```
CONSTRUCT { ?s1 skos:notation ?n1 .  
             ?s2 skos:notation ?n2 } WHERE {  
  ?s1 rdf:type skos:Concept .  
  ?s2 rdf:type skos:Concept .  
  ?s1 skos:inScheme ?cs .  
  ?s2 skos:inScheme ?cs .  
  ?s1 skos:notation ?n1 .  
  ?s2 skos:notation ?n2 .  
  FILTER (?s1!=?s2 && ?n1=?n2)  
}
```



5. Modelo de evaluación

Los diferentes indicadores cuantitativos, definiciones, condiciones de integridad y criterios de calidad expuestos anteriormente pueden estructurarse siguiendo una serie de criterios. Dichos criterios responden a la naturaleza de los elementos que intervienen en el análisis y por lo tanto a diferentes aspectos de modelado y gestión del vocabulario.

Así pues, por ejemplo, la condición de integridad [S13] está relacionada con aspectos del etiquetado del vocabulario, mientras que la condición [S27] se asocia a la estructura de relaciones semánticas del mismo.

Por otro lado, el nivel de exigencia con respecto al cumplimiento de los criterios de calidad no es el mismo en todos los casos. Mientras que las definiciones y condiciones de integridad son de obligado cumplimiento para un modelado correcto conforme a la recomendación SKOS, no sucede lo mismo en todos los criterios de calidad expuestos anteriormente. Por este motivo se ha establecido una doble clasificación tanto para las definiciones y condiciones de integridad normativas como para los criterios de calidad.

5.1. Grupos funcionales de análisis

La clasificación elaborada para este trabajo parte de la propuesta de Mader y Haslhofer (2011) en donde se identifican cinco grupos de criterios: Estructura de los grafos RDF, Linked Data, Conformidad con SKOS, Etiquetado y Dominio del vocabulario / otros. En este trabajo se refina dicha clasificación y se proponen los siguiente grupos funcionales de análisis:

- **Tipología de recursos del vocabulario:** aspectos normativos y de calidad referidos a la disyunción de las clases de recursos de SKOS establecidos en la recomendación.
- **Etiquetado y documentación:** uso adecuado de las propiedades de etiquetado, así como de las diferentes propiedades de anotación y documentación del vocabulario.
- **Macro-estructura:** definición y uso correcto de las clases y propiedades asociadas a la definición y manejo de esquemas de conceptos y colecciones.
- **Relaciones semánticas:** uso de las propiedades para la definición de relaciones jerárquicas (incluidas las transitivas) y asociativas.
- **Mapeado y Linked Open Data:** mapeado de conceptos a través de las propiedades correspondientes y definición de relaciones con conjuntos de datos externos al



vocabulario.

5.2. Niveles de exigencia

Aunque las definiciones y condiciones de integridad de la recomendación SKOS son de obligado cumplimiento para una correcta representación de un vocabulario, no ocurre lo mismo con todos los criterios de calidad. En este sentido, los niveles de exigencia propuestos expresan los requisitos de modelado del vocabulario en los siguientes términos:

- **Nivel 1:** incluye todos los aspectos normativos de la recomendación de SKOS, junto con aquellos criterios de calidad que aseguran un correcto modelado y organización del dominio de conocimiento del vocabulario. El incumplimiento de alguno de los elementos de análisis de este nivel implica que el vocabulario presenta deficiencias que supongan un grave inconveniente para consulta, gestión o mantenimiento.
- **Nivel 2:** agrupa los criterios de calidad que determinan si un vocabulario está representado de forma que permita un acceso claro tanto por parte de personas como por parte de aplicaciones informáticas. El incumplimiento de los criterios de calidad de este nivel puede conllevar ciertos obstáculos para la consulta y acceso a los elementos del vocabulario.
- **Nivel 3:** los criterios de calidad de este nivel suponen la incorporación de aspectos de representación que aportan valor añadido a la representación del vocabulario SKOS. El incumplimiento de los criterios de calidad de este nivel implica la ausencia de cierta información adicional y de interconexión con vocabularios externos.

5.3. Estructura del modelo de evaluación

A partir de los indicadores cuantitativos, elementos normativos y criterios de calidad analizados en la sección 4 se propone la siguiente estructuración:

Tipología de recursos

Nivel 1

- **[S9]** Condición de integridad: La clase skos:ConceptScheme es disyunta con skos:Concept.
- **[S37]** Definición: La clase skos:Collection es disyunta con las clases skos:Concept y skos:ConceptScheme.
- **[S48]** Definición: la clase skosxl:Label es disyunta con el resto de clases de la



recomendación SKOS: skos:Concept, skos:ConceptScheme y skos:Collection.

- [C22] Uso de elementos de SKOS no válidos.

Etiquetado y documentación

Nivel 1

- [S12] Definición: el rango para las propiedades skos:prefLabel, skos:altLabel y skos:hiddenLabel es un literal plano RDF.
- [S13] Condición de integridad: Las propiedades skos:prefLabel, skos:altLabel y skos:hiddenLabel son disyuntas entre sí.
- [S14] Condición de integridad: Un recurso no debe tener más de una etiqueta skos:prefLabel en un mismo idioma.
- [S50] Definición: el dominio de skosxl:literalForm es la clase skosxl:Label.
- [S51] Definición: el rango de la propiedad skosxl:literalForm es un literal plano RDF (sin ningún tipo de dato asociado).
- [S52] Definición: la cardinalidad de la propiedad skosxl:literalForm es exactamente 1.
- [S54] Definición: el rango de las propiedades skosxl:prefLabel, skosxl:altLabel y skosxl:hiddenLabel es la clase skosxl:Label.
- [S58] Definición: Las propiedades skosxl:prefLabel, skosxl:altLabel y skosxl:hiddenLabel son disyuntas entre sí.
- [S60] Definición: el dominio de skosxl:labelRelation es la clase skosxl:Label.
- [S61] Definición: el rango de la propiedad skosxl:labelRelation es la clase skosxl:Label.
- [C1] Etiquetas de idioma inválidas u omitidas.
- [C22] Uso de elementos de SKOS no válidos.

Nivel 2

- [C2] No se han etiquetado todos los literales en todos los idiomas recogidos por el vocabulario.
- [C5] Ausencia de etiquetado relevante.

Nivel 3

- [C3] Conceptos no documentados.
- [C4] Conceptos documentados parcialmente.
- [C6] Existencia de espacios en blanco innecesarios en las etiquetas.



Macro-estructura

Nivel 1

- [S4] Definición: el rango de skos:inScheme es la clase skos:ConceptScheme.
- [S5] Definición: el dominio de skos:hasTopConcept es la clase skos:ConceptScheme.
- [S6] Definición: el rango de skos:hasTopConcept es la clase skos:Concept.
- [S31] Definición: el dominio de la propiedad skos:member es la clase skos:Collection.
- [S32] Definición: el rango de la propiedad skos:member es la unión de las clases skos:Concept y skos:Collection.
- [S33] Definición: El dominio de skos:memberList es la clase skos:OrderedCollection.
- [S34] Definición: El rango de skos:memberList es la clase rdf:List, es decir, una lista RDF.
- [S35] Definición: la propiedad skos:memberList es una instancia de la clase owl:FunctionalProperty.
- [C14] Conceptos cabecera con conceptos genéricos.
- [C22] Uso de elementos de SKOS no válidos.

Nivel 2

- [C8] Grupos de conceptos desconectados.
- [C13] Ausencia de conceptos cabecera.
- [C15] Conceptos superiores no marcados como conceptos cabecera.

Relaciones semánticas

Nivel 1

- [S19] Definición: El dominio de skos:semanticRelation es la clase skos:Concept.
- [S20] Definición: el rango de skos:semanticRelation is the class skos:Concept.
- [S27] Condición de integridad: la propiedad skos:related es disyunta con la propiedad skos:broaderTransitive.
- [C7] Conceptos aislados (“huérfanos”).
- [C9] Relaciones jerárquicas cíclicas.
- [C14] Conceptos cabecera con conceptos genéricos.



- [C16] Colisión de relaciones.
- [C22] Uso de elementos de SKOS no válidos.

Nivel 2

- [C10] Relaciones asociativas sin valor.
- [C12] Relaciones semánticas unidireccionales.

Nivel 3

- [C11] Uso exclusivo de relaciones jerárquicas transitivas.

Mapeado y Linked Open Data

Nivel 1

- [S46] Condición de integridad: La propiedad skos:exactMatch es disyunta con skos:broadMatch y skos:relatedMatch.
- [C20] Sintaxis de URIs no válidas.
- [C22] Uso de elementos de SKOS no válidos.

Nivel 2

- [C19] Enlaces salientes rotos.

Nivel 3

- [C17] Ausencia de enlaces entrantes.
- [C18] Ausencia de enlaces salientes.
- [C21] Existencia de recursos RDF en el vocabulario con URIs no derreferenciables.

5.4. Organización de indicadores cuantitativos

Una de las características del modelo es la integración de los indicadores cuantitativos en función de su relación con los correspondientes grupos funcionales. A partir de la estructura anterior se propone la siguiente distribución:

Tipología de recursos

- [M1] Número de conceptos del vocabulario .
- [M2] Número de colecciones del vocabulario.
- [M3] Número de esquemas de conceptos del vocabulario.

Etiquetado y documentación

- [M4] Idiomas utilizados y número de etiquetas preferentes por idioma .
- [M5] Idiomas utilizados y número de etiquetas alternativas por idioma .



- [M6] Idiomas utilizados y número de etiquetas ocultas por idioma .
- [M7] Tasa de equivalencia .
- [M11] Tasa de precoordinación .
- [M12] Tasa de anotación .
- [M13] Tasa de flexibilidad .
- [M16] Tasa de ambigüedad.

Macro-estructura

- [M8] Tasa de conexión .
- [M15] Tamaño de los grupos de conceptos .

Relaciones semánticas

- [M8] Tasa de conexión .
- [M9] Tasa de reciprocidad
- [M10] Tasa de enriquecimiento
- [M14] Número de niveles
- [M15] Tamaño de los grupos de conceptos .

Mapeado y Linked Open Data

- [M17] Tasa de mapeado

5.5. Puntualizaciones sobre el modelo

Se ha considerado el criterio de calidad [C22] como relevante al nivel 1 de todos los grupos funcionales de análisis, puesto que el uso correcto del espacio de nombres de SKOS y SKOS-XL es indispensable para que un vocabulario cumpla cualquiera de los aspectos de análisis.

La práctica no suele aplicar de forma intensiva la documentación de los conceptos de los vocabularios SKOS. Por este motivo los criterios de calidad [C3], [C4] y [C6] se han ubicado en el nivel 3 de exigencia.

Un vocabulario en el que los conceptos cabecera tengan conceptos genéricos sigue siendo coherente con el modelo de SKOS. Sin embargo, desde el punto de vista del modelado de vocabulario supone una grave incoherencia, motivo por el que el criterio [C14] está encuadrado en el nivel 1 de exigencia.

Por otro lado, se ha decidido incluir el criterio [C11] dentro del nivel de exigencia 3, puesto que la recomendación no indica explícitamente que el uso exclusivo de relaciones jerárquicas



transitivas conlleve problemas para la consulta y uso del vocabulario. Los criterios [C17], [C18] y [C21] al respecto de la definición de enlaces Linked Data, se ha considerado como un valor añadido al vocabulario, motivo por el que se ubican en el nivel 3 de exigencia.

	NIVEL 1	NIVEL 2	NIVEL 3
TIPOLOGÍA DE RECURSOS M1 M2 M3	S9 S37 S48 C22		
ETIQUETADO Y DOCUMENTACIÓN M4 M5 M6 M7 M11 M12 M13 M16	S12 S13 S14 S50 S51 S52 S54 S58 S60 S61 C1 C22	C2 C5	C3 C4 C6
MACRO-ESTRUCTURA M8 M15	S4 S5 S6 S31 S32 S33 S34 S35 C14 C22	C85 C13 C15	
RELACIONES SEMÁNTICAS M8 M9 M10 M14 M15 M16	S19 S20 S27 C7 C9 C14 C16 C22	C10 C12	C11
MAPEADO Y LINKED OPEN DATA M17	S46 C20 C22	C19	C17 C18 C21

M Indicador cuantitativo
S Integridad normativa
C Criterio de calidad



Tabla 5: Resumen de la estructura del modelo propuesto para la evaluación de vocabularios SKOS.

6. Conclusiones y líneas de trabajo futuras

A lo largo del este TFM se ha mostrado la posibilidad de evaluación de vocabularios SKOS, utilizando una serie de elementos de análisis y utilizando fundamentalmente para ello una tecnología propia de la Web Semántica como es SPARQL. Los elementos se organizan en un modelo de evaluación que posee una estructura bi-dimensional: grupo funcionales de análisis y niveles de exigencia.

Una de las primeras conclusiones que se pueden extraer de dicho modelo es su naturaleza ampliable, flexible y modular: la adición de nuevos elementos o la creación de nuevas componentes funcionales o de exigencia puede realizarse fácilmente, reorganizando el modelo y reutilizando los elementos de análisis ya existentes. De hecho una posible línea de trabajo en el futuro es la incorporación de nuevos criterios de calidad que actualmente están en proceso de definición por parte de Christian Mader, relacionados con el manejo de los cualificadores en las unidades léxicas y por lo tanto asociados al etiquetado del vocabulario.

El uso de SPARQL se ha demostrado eficaz tanto en la obtención de indicadores, como en la verificación de las definiciones y condiciones de integridad normativas y en la evaluación de los criterios de calidad. Pensamos que esta tecnología es una de las claves en los procesos de evaluación de vocabularios SKOS. Por otro lado, pese a que la recomendación SPARQL 1.1 del lenguaje de consulta tiene total vigencia, en estos momentos hay que intentar diseñar las consultas con la recomendación SPARQL de 2008, previendo que el despliegue de SPARQL 1.1 se está produciendo actualmente.

Sin embargo, esto no quiere decir que deba abandonarse el uso de SPARQL 1.1, todo lo contrario: hay que seguir profundizando en su aplicación, en especial en lo referente a nuevos operadores, agregación, subconsultas, rutas de propiedades y expresiones regulares de sentencias. Una línea de trabajo futura sería la ampliación o matización de muchas de las consultas SPARQL planteadas en este trabajo, redefiniéndolas en la medida de lo posible con SPARQL 1.1 para obtener una mayor eficiencia e interoperabilidad de las operaciones de evaluación de los vocabularios SKOS.

La aplicación de motores de inferencia en la evaluación resulta esencial: la capacidad de estos sistemas para considerar nuevas subpropiedades y subclases de elementos durante este proceso es clave. Esto es de vital importancia, sobre todo teniendo en cuenta que, aunque



SKOS no haya sufrido actualizaciones desde 2008, es posible que los desarrolladores definan nuevos elementos a partir de los ya existentes para adaptarse a proyectos concretos o a nuevas necesidades. Un ejemplo de ello es el desarrollo de la extensión ISO-THES, para definir equivalencias entre la norma de tesauros 25964 y SKOS. La incorporación de ISO-THES a los procesos de evaluación de vocabularios podría realizarse de un modo más eficiente utilizando motores de inferencia.

Con respecto a los indicadores cuantitativos existentes actualmente, cabe señalar su escasa presencia en relación a los grupos funcionales de Macro-estructura. Esto se debe a que estas macro-estructuras (micro-tesauros, facetas, arrays) todavía no cuentan con un modelado estándar mediante los elementos que ofrece SKOS. Se precisan por tanto delimitar un conjunto de buenas prácticas (o prácticas más comunes) para el modelado de macro-estructuras con SKOS, de forma que puedan elaborarse indicadores cuantitativos estándar.

Por otro lado, también hay que destacar la ausencia prácticamente total de indicadores cuantitativos en el grupo funcional de Mapeado y Linked Open Data. La relativa novedad que supone el auge de LOD en los últimos años y la progresiva incorporación de relaciones de mapeado, o interconexiones entre conjuntos de datos, demanda una mayor profundización en este aspecto.

Como conclusión final, es necesario desarrollar servicios de evaluación de vocabularios SKOS, basados en tecnologías interoperables como SPARQL, y cuyos procesos se definan de forma abierta y transparente, ajenos a soluciones específicas de software propietario. De este modo se contribuirá a una mejora de los conjuntos de datos correspondientes, cuya constante mejora es un factor clave para su aplicación eficiente en el entorno presente y futuro de LOD.



7. Referencias

- Alvestrand, H. (2001). RFC3066: Tags for the identification of Languages. Recuperado el 03-09-2013 de <http://www.ietf.org/rfc/rfc3066.txt>
- Berners-Lee, T. (2006). Linked data. Recuperado el 12-07-2013 de <http://www.w3.org/DesignIssues/LinkedData.html>
- Berners-Lee, T., Fielding, R. & Masinter, L. (2005). RFC3986: Uniform Resource Identifier (URI): Generic Syntax. Recuperado el 03-09-2013 de <http://www.ietf.org/rfc/rfc3986.txt>
- Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific american*, 284(5), 28-37.
- Berners-Lee, T., Masinter, L. & McCahill, M. (1994). RFC1738: Uniform Resource Locators (URL). Recuperado el 03-09-2013 de <http://www.ietf.org/rfc/rfc1738.txt>
- Duerst, M., Suignard, M. (2005). RFC3987: Internationalized Resource Identifiers (IRIs). Recuperado el 03-09-2013 de <http://www.ietf.org/rfc/rfc3987.txt>
- Gil Urdiciain, B. (1998). Evaluación semántica y estructural de tesauros. *Revista general de información y documentación*, 8(2), 193-199.
- Gilliland-Swetland, A. (2003). Metadata - Where Are We Going? En G.E. Gorman (ed.) *International yearbook of library and information management 2003-2004: Metadata applications and management* (pp. 17-33). London: Facet Publishing.
- Halpin, H. & Thompson, H. S. (2006). One document to bind them: combining XML, web services, and the semantic web. En *Proceedings of the 15th International Conference on World Wide Web*, Edinburgh, Scotland, May 23 - 26, 2006 (pp. 679-686). Recuperado el 23-08-2013 de <http://www2006.org/programme/files/pdf/5060.pdf>
- Haslhofer, B. & Schandl, B. (2010). Interweaving OAI-PMH data sources with the linked data cloud. *International Journal of Metadata, Semantics and Ontologies*, 5(1), 17-31. Recuperado 01-05-2013 de <http://dx.doi.org/10.1504/IJMSO.2010.032648>



- Hendler, J., Berners-Lee, T., & Miller, E. (2002). Integrating Applications on the Semantic Web. *Journal of the Institute of Electrical Engineers of Japan*, 122(10), 676-689. Recuperado el 03-09-2013 de <http://www.w3.org/2002/07/swint.html>
- ISO 5964:1985. Guidelines for the establishment and development of multilingual thesauri. Geneva: International Organization for Standardization.
- ISO 2788:1986. Guidelines for the establishment and development of monolingual thesauri. Geneva: International Organization for Standardization.
- ISO 25964-1:2011. Thesauri and interoperability with other vocabularies. Part 1: Thesauri for information retrieval. Geneva: International Organization for Standardization, 2011.
- ISO 25964-2:2013. Thesauri and interoperability with other vocabularies. Part 2: Interoperability with other vocabularies. Geneva: International Organization for Standardization, 2013.
- Lancaster, F. W. (2001). Elaboración y mantenimiento tesauros. En F.W. Lancaster, F.W. y M. Pinto Molina (coords.) *Procesamiento de la información científica*. Madrid: Arco libros.
- Lancaster, F. W. (2002). *El control del vocabulario en la recuperación de la información*. 2a ed. Valencia: Universidad de Valencia.
- Losee, R. M. (2007). Decisions in Thesaurus Construction and Use. *Information Processing & Management*, 43(4), 958-968.
- Loukachevitch, N. V. & Dobrov, B. V. (2002). Evaluation of Thesaurus on Sociopolitical Life as Information-Retrieval Tool. En *LREC, European Language Resources Association*, 2002, 115-121.
- Mader, C. (2012). Quality assurance in collaboratively created Web vocabularies. En *The Semantic Web: Research and Applications* (pp. 870-874). Springer Berlin Heidelberg.
- Mader, C. & Haslhofer, B. (2011). Quality Criteria for Controlled Web Vocabularies. En *International Conference on Theory and Practice of Digital Libraries 2011, NKOS Workshop*, 1-5.



- Mader, C. & Haslhofer, B. (2013). Perception and relevance of quality issues in web vocabularies. En *I-SEMANTICS'13, Proceedings of the 9th International Conference on Semantic Systems*, 9-16 .
- Mader, C., Haslhofer, B., & Isaac, A. (2012). Finding quality issues in SKOS vocabularies. En P. Zaphiris; F. Loizides & E. Rasmussen (eds.) *Theory and Practice of Digital Libraries* (pp. 222-233). Springer Berlin Heidelberg.
- Martínez, A.M. et al. (2010). Concepto, forma y longitud de los términos preferentes del tesauro: una propuesta de indicadores de calidad. *Anales de documentación*, 13, 185-195.
- Martínez, A.M. et al. (2011). La estructura sistemática del tesauro: indicadores para evaluar su calidad. *Revista española de documentación científica*, 34(1), 29-43.
- Méndez, E. & Greenberg, J. (2012). Linked Data for Open Vocabularies and HIVE's global framework. *El Profesional de la Información*, 21(3), 236-244.
- Pastor-Sánchez, J. A., Martínez-Méndez, F. J., & Rodríguez-Muñoz, J. V. (2012). Aplicación de SKOS para la interoperabilidad de vocabularios controlados en el entorno de linked open data. *El profesional de la información*, 21(3), 245-253.
- Pastor Sánchez, J. A. (2013). ISO-THES: ampliando Skos a partir de la norma de tesauros ISO 25964. *Anuario ThinkEPI*, 7, 189-193.
- Saorín, T., Peset, F. & Ferrer-Sapena, A. (2013). Factores para la adopción de linked data e implantación de la web semántica en bibliotecas, archivos y museos. *Information Research*, 18(1) paper 570. <http://InformationR.net/ir/18-1/paper570.html>.
- Suominen, O. & Hyvönen, E. (2012). Improving the quality of SKOS vocabularies with skosify. En *EKAW'12 Proceedings of the 18th international conference on Knowledge Engineering and Knowledge Management*, 383-397.
- Suominen, O. & Mader, C. (2013). Assessing and Improving the Quality of SKOS Vocabularies. *Journal of Data Semantics*. DOI: 10.1007/s13740-013-0026-0



- Tangmunarunkit, H., Decker, S., & Kesselman, C. (2003). Ontology-based resource matching in the Grid—the Grid meets the semantic web. En *The Semantic Web-ISWC 2003*, pp. 706-721.
- Gil Urdiciain, B. (1998). Evaluación semántica y estructural de tesauros. *Revista general de información y documentación*, 8(2), 193-199.
- Volz, R., Oberle, D., & Studer, R. (2003). Implementing views for light-weight web ontologies. En *Database Engineering and Applications Symposium, 2003. Proceedings. Seventh International*, 160-169.
- Vrandečić, D., Dengler, F., Rudolph, S., & Erdmann, M. (2009). RDF syntax normalization using XML validation. En *Proceedings of the Workshop Semantics for the Rest of Us at ISWC2009*. Recuperado el 03-09-2013 de <http://www.aifb.kit.edu/images/7/72/SemRUs2009SyntaxNorm.pdf>
- W3C (1999). World Wide Web Consortium. XSL Transformations (XSLT). W3C Recommendation 16 November 1999 (1999). J. Clark, (ed.). Disponible en: <http://www.w3.org/TR/xslt>
- W3C (2004a). World Wide Web Consortium. RDF Primer. W3C Recommendation 10 February 2004. F. Manola, & E. Miller (eds.). Disponible en: <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>
- W3C (2004b). World Wide Web Consortium. Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation 10 February 2004. G. Klyne & J. J. Carroll (eds.). Disponible en: <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>
- W3C (2004c). World Wide Web Consortium. RDF/XML Syntax Specification (Revised). W3C Recommendation 10 February 2004. D. Beckett, (ed). Disponible en: <http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/>
- W3C (2004d). World Wide Web Consortium. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation 10 February 2004. D. Brickley, & R.V. Guha (eds). Disponible en: <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>



- W3C (2004e). World Wide Web Consortium. RDF Semantics. W3C Recommendation 10 February 2004. P. Hayes (ed.). Disponible en:
<http://www.w3.org/TR/2004/REC-rdf-mt-20040210/>
- W3C (2009a). World Wide Web Consortium. SKOS simple knowledge organization system reference. W3C Recommendation 18 August 2009. A. Miles & S. Bechhofer (eds.). Disponible en: <http://www.w3.org/TR/skos-reference>
- W3C (2009b). World Wide Web Consortium. Namespaces in XML 1.0 (Third Edition). W3C Recommendation 8 December 2009. T. Bray, D. Hollander, A. Layman, R. Tobin & H.S. Thompson, (eds.). Disponible en:
<http://www.w3.org/TR/2009/REC-xml-names-20091208/>
- W3C (2011). World Wide Web Consortium. Library Linked Data. W3C Incubator Group Final Report. 25 October 2011. T. Baker et al. (eds.). Disponible en:
<http://www.w3.org/2005/Incubator/lld/XGR-ldd-20111025/>
- W3C (2012a). World Wide Web Consortium. OWL 2 Web Ontology Language Profiles (Second Edition). W3C Recommendation 11 December 2012. B. Motik, B. et al. (eds.). Disponible en: <http://www.w3.org/TR/2012/REC-owl2-profiles-20121211/>
- W3C (2012b). World Wide Web Consortium. W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes. W3C Recommendation 5 April 2012. D. Peterson et al. (eds.). Disponible en: <http://www.w3.org/TR/2012/REC-xmlschema11-2-20120405/>
- W3C (2012c). World Wide Web. OWL 2 Web Ontology Language Primer (Second Edition). W3C Recommendation 11 December 2012. P. Hitzler (eds.). Disponible en:
<http://www.w3.org/TR/2012/REC-owl2-primer-20121211/>
- W3C (2013a). SPARQL 1.1 Overview. W3C Recommendation 21 March 2013. W3C SPARQL Working Group (ed.). Disponible en:
<http://www.w3.org/TR/2013/REC-sparql11-overview-20130321/>
- W3C (2013b) World Wide Web Consortium. SPARQL Query Results XML Format (Second Edition). W3C Recommendation 21 March 2013. D. Beckett, D. & J. Broekstra (eds.). Disponible en: <http://www.w3.org/TR/2013/REC-rdf-sparql-XMLres-20130321/>



8. Sitios web citados

Página del W3C sobre la Web Semántica. <http://www.w3.org/2001/sw>.

Datahub. <http://datahub.io>. Catálogo colaborativo de conjuntos de datos RDF.

Notation3. <http://www.w3.org/DesignIssues/Notation3.html>.

Turtle. <http://www.dajobe.org/2004/01/turtle>.

OWL. <http://www.w3.org/2004/OWL>.

Ontología GND. <http://d-nb.info/standards/elementset/gnd>.

qSKOS. <https://github.com/cmader/qSKOS/wiki>.

Skosify. <http://code.google.com/p/skosify>.

PoolParty. <http://poolparty.biz>.

ISO-THES. <http://www.niso.org/schemas/iso25964/correspondencesSKOS/>.