

Este documento es una traducción al castellano de la nota del grupo de trabajo del W3C "SKOS Simple Knowledge Organization System Primer", publicada el 18 de agosto de 2009. La presente traducción se concluyó el 20 de noviembre de 2009.

La versión original en inglés es el único documento válido y se encuentra en:

<http://www.w3.org/TR/2009/NOTE-skos-primer-20090818/>

Puede ver la última versión del documento en inglés en: <http://www.w3.org/TR/skos-primer/>

Se ha tratado de respetar al máximo el contenido del documento original en inglés, adaptando la expresión al español para ayudar a una mejor comprensión del mismo. Por tanto, esta traducción puede contener errores, en ningún caso achacables a sus autores originales.

Cualquier sugerencia de corrección, duda o comentario sobre la misma puede realizarse dirigiéndose a alguno de sus autores: [Juan Antonio Pastor Sánchez](#) y [Francisco Javier Martínez Méndez](#).



Manual de SKOS (Simple Knowledge Organization System, Sistema para la Organización del Conocimiento simple)

Nota del Grupo de Trabajo del W3C de 18 de Agosto de 2009

Versión original (en inglés):

<http://www.w3.org/TR/2009/NOTE-skos-primer-20090818/>

Última versión:

<http://www.w3.org/TR/skos-primer>

Versión anterior:

<http://www.w3.org/TR/2009/WD-skos-primer-20090615/>

Editores:

[Antoine Isaac](#), Vrije Universiteit Amsterdam

[Ed Summers](#), Library Of Congress

Traductores (versión en castellano):

[Juan Antonio Pastor Sánchez](#), Universidad de Murcia

[Francisco Javier Martínez Méndez](#), Universidad de Murcia

Consulte la sección de [erratas](#) en la que se pueden incluir algunas correcciones para este documento.

Vea también la sección de [traducciones](#)

[Copyright](#) © 2009 [W3C](#)® ([MIT](#), [ERCIM](#), [Keio](#)), Todos los derechos reservados. Son aplicables las

Resumen

SKOS—Simple Knowledge Organization System (Sistema para la Organización del Conocimiento simple)— proporciona un modelo para la representación de la estructura básica y el contenido de esquemas de conceptos como tesauros, esquemas de clasificación, listas de encabezamientos de materia, taxonomías, folksonomías y otros vocabularios controlados similares. Al tratarse de una aplicación de [RDF \(Resource Description Framework\)](#), SKOS permite la creación y publicación de conceptos en la Web, así como vincularlos con datos en este mismo medio e incluso integrarlos en otros esquemas de conceptos.

Este documento es una guía para aquellos usuarios que deseen representar un esquema de conceptos mediante SKOS.

Un uso básico de SKOS permite identificar los recursos conceptuales (conceptos) mediante URIs, etiquetarlos con literales de uno o varios idiomas, documentarlos con diversos tipos de notas, relacionarlos entre sí mediante estructuras jerárquicas informales o redes asociativas, y agregarlos a esquemas de conceptos.

La aplicación de los aspectos más avanzados de SKOS permite mapear recursos conceptuales de distintos esquemas de conceptos y agruparlos en colecciones etiquetadas u ordenadas. Es posible definir relaciones entre etiquetas de conceptos. Finalmente, el vocabulario de SKOS puede ser ampliado para adaptarse a las necesidades prácticas de comunidades de usuarios concretas o combinadas con otros vocabularios de modelado.

Este documento es un complemento de la [Guía de Referencia de SKOS](#), que proporciona la referencia normativa sobre SKOS.

Estado de este documento

Esta sección describe el estado de este documento en el momento de su publicación. Otros documentos pueden reemplazar este documento. Una lista de las publicaciones vigentes del W3C y la última revisión de este informe técnico se puede encontrar en el [índice de informes técnicos del W3C](#) en <http://www.w3.org/TR/>.

Este documento es una nota de un grupo de trabajo publicada por el [Grupo de Trabajo para el desarrollo de la Web Semántica](#), formando parte de la [Actividad para la Web Semántica del W3C](#). Esta versión es una actualización del [borrador de trabajo previo, del 15 de junio de 2009](#). Esta versión incluye varios cambios editoriales menores, así como la eliminación de un ejemplo sugerido para referenciar a un sistema de notación (por ejemplo una notación simbólica) en una etiqueta en donde el sistema de notación no se corresponde con un lenguaje natural. Dicha sugerencia se consideró incompatible con la [actual buena práctica 47](#) del IETF, sobre el uso de

etiquetas para la identificación de idiomas. Los usuarios deben considerar el [vocabulario de ampliación de SKOS](#) para apoyar sistemas alternativos de notación.

Este documento complementa la [Recomendación del W3C sobre SKOS](#) de fecha 18 de agosto de 2009.

Los comentarios sobre este documento puede enviarse a public-swd-wg@w3.org, por favor, incluir el texto "SKOS comentario" como asunto del mensaje. Todos los mensajes recibidos en esta dirección están disponibles para su consulta en un [archivo público](#).

Este documento fue elaborado por un grupo que opera en el marco de la [Política de Patentes del W3C del 5 de febrero 2004](#). El W3C mantiene una lista pública de cualquier patente divulgada realizada en coordinación con la difusión de los resultados del grupo de trabajo; dicha página también incluye instrucciones para la divulgación de una patente. Una persona que tenga conocimiento actual de una patente que considere que contiene la reivindicación(es) esencial(es) debe revelar la información de conformidad con el [artículo 6 de la Política de Patentes del W3C](#).

La publicación de una nota de un grupo de trabajo no implica la aprobación por los Miembros del W3C. Se trata de un proyecto de documento y puede ser actualizado, reemplazado o quedar obsoleto por otros documentos en cualquier momento. No es adecuado citar este documento como algo más que un trabajo en curso.

Tabla de contenidos

- [1 Introducción](#)
 - [1.1 Acerca de este manual](#)
- [2 Fundamentos de SKOS](#)
 - [2.1 Conceptos](#)
 - [2.2 Etiquetas](#)
 - [2.2.1 Etiquetas léxicas preferentes](#)
 - [2.2.2 Etiquetas léxicas alternativas](#)
 - [2.2.3 Etiquetas léxicas ocultas](#)
 - [2.3 Relaciones semánticas](#)
 - [2.3.1 Relaciones genéricas/específicas](#)
 - [2.3.2 Relaciones asociativas](#)
 - [2.4 Notas de documentación](#)
 - [2.5 Esquema de conceptos](#)
- [3 Redes de sistemas de organización del conocimiento en la Web Semántica](#)
 - [3.1 Mapeado de esquemas de conceptos](#)
 - [3.2 Reutilización y extensión de esquemas de conceptos](#)
 - [3.3 Indización temática y SKOS](#)
- [4 SKOS Avanzado: cuando los sistemas de organización del conocimiento no son tan simples](#)
 - [4.1 Colecciones de conceptos](#)
 - [4.2 Características avanzadas de documentación](#)

- [4.3 Relaciones entre etiquetas](#)
 - [4.4 Coordinación de conceptos](#)
 - [4.5 Jerarquías transitivas](#)
 - [4.6 Notaciones](#)
 - [4.7 Sobre la especialización del modelo SKOS](#)
 - [5 Combinaciones de SKOS con otras aproximaciones de modelado](#)
 - [5.1 Uso de etiquetas fuera de SKOS](#)
 - [5.2 Conceptos de SKOS y clases de OWL](#)
 - [5.3 SKOS, conjuntos de datos RDF y contención de Información](#)
 - [Referencias](#)
 - [Agradecimientos](#)
 - [Apéndice. Correspondencias entre la norma ISO-2788/5964 y los constructores de SKOS](#)
-

1 Introducción

SKOS (Simple Knowledge Organization System, Sistema de Organización del Conocimiento Simple) es un vocabulario RDF para la representación de sistemas de organización del conocimiento (Knowledge Organization Systems, KOSs) semi-formales, tales como tesauros, taxonomías, esquemas de clasificación y listas de encabezamiento de materias. SKOS está basado en RDF (Resource Description Framework) [[RDF-PRIMER](#)] por lo que dichas representaciones pueden ser legibles por máquinas e intercambiarse entre aplicaciones de software, así como publicarse en la World Wide Web.

SKOS ha sido diseñado para proporcionar un modo de migrar a la Web Semántica sistemas de organización del conocimiento ya existentes con un coste bajo. SKOS también proporciona un lenguaje conceptual de modelado muy sencillo e intuitivo para desarrollar y compartir nuevos sistemas de organización. Puede utilizarse por su cuenta, o en combinación con lenguajes más formales, como el Lenguaje de Ontologías Web (OWL) [[OWL](#)]. SKOS también puede contemplarse como una tecnología de transición que proporciona un nexo de unión entre el formalismo lógico riguroso de los lenguajes de ontologías como OWL y el mundo caótico, informal y débilmente estructurado de las herramientas colaborativas basadas en Web, ejemplificadas por las aplicaciones de etiquetado social.

El objetivo de SKOS no es sustituir vocabularios conceptuales originales en su contexto inicial de uso, sino que puedan implementarse en un espacio compartido, basado en un modelo simplificado, que haga posible su reutilización y una mejor interoperabilidad.

1.1 Acerca de este manual

Este documento está destinado a ayudar a los usuarios que tienen una comprensión básica de RDF para representar y publicar su esquema de conceptos en forma de datos SKOS. El Manual tiene por objeto proporcionar ejemplos de introducción y orientación en el uso del vocabulario de SKOS.

Para una explicación sistemática de todos los elementos del vocabulario SKOS, incluyendo su semántica de referencia, el lector debe consultar la

referencia normativa SKOS [[SKOS-REFERENCE](#)]. Esto puede hacerse a nivel de clases y propiedades, haciendo uso de los correspondientes enlaces del texto (por ejemplo, [skos:Concept](#)). Para una visión general de los casos de uso de SKOS y las causas de los requisitos que guiaron su diseño, el lector debe consultar los *Casos de Uso y requisitos de SKOS* [[SKOS-UCR](#)].

Este Manual, junto con la Guía de referencia de SKOS [[SKOS-REFERENCE](#)], sustituyen a los documentos anteriores de la *Guía de SKOS Core* [[SWBP-SKOS-CORE-GUIDE](#)] y la *Especificación del vocabulario de SKOS Core* [[SWBP-SKOS-CORE-SPEC](#)], que se consideran obsoletas.

Las características esenciales del modelo de SKOS se explican en la Sección 2, donde se presenta al lector el conjunto de elementos utilizados más comúnmente para representar Sistemas de Organización del Conocimiento. En la sección 3, se muestra al lector cómo añadir valor a estas representaciones, ya sea estableciendo relaciones entre ellos o con otros tipos de recursos de la Web Semántica. Se espera que muchas aplicaciones de SKOS empleen algunas características presentadas en la sección 3. La Sección 4 se centra en las necesidades de representación más avanzadas, que puedan requerir determinadas aplicaciones de SKOS. La Sección 5 describe el uso combinado de SKOS con otras aproximaciones de modelización, en particular OWL.

Acerca de los ejemplos incluidos en este manual

La mayoría de los ejemplos de este manual se ofrecen como una serialización de grafos RDF usando la sintaxis Turtle para RDF [[TURTLE](#)]. Los ejemplos serializados con Turtle aparecen en las líneas de código del siguiente modo:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix ex: <http://www.ejemplo.com/>.
```

```
ex:Recurso ex:Propiedad ex:otroRecurso;
ex:otraPropiedad "Literal RDF"@es.
```

Lo anterior es equivalente a la siguiente expresión donde se utiliza la sintaxis de referencia RDF/XML [[RDF/XML-SYNTAX](#)]:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://www.ejemplo.com/">
  <rdf:Description rdf:about="http://www.ejemplo.com/Recurso">
    <ex:Propiedad rdf:resource="http://www.ejemplo.com/otroRecurso"/>
    <ex:otraPropiedad xml:lang="en">Literal RDF</ex:otraPropiedad>
  </rdf:Description>
</rdf:RDF>
```

En aras de la brevedad se omiten una serie de declaraciones de espacios de nombres en los ejemplos. Esto se aplica a los espacios estándares de nombres (SKOS, RDF/RDFS [[RDF-PRIMER](#)], OWL [[OWL](#)] y Dublin Core [[DC](#)]), y a los que se han acuñado para los ejemplos. En general, estos espacios de nombres podrían declararse como se muestra en el siguiente código:

```
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
```

```
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix dct: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix ex: <http://www.ejemplo.com/> .
@prefix ex1: <http://www.ejemplo.com/1/> .
@prefix ex2: <http://www.ejemplo.com/2/> .
```

2 Fundamentos de SKOS

Esta sección presenta el núcleo del modelo de SKOS, es decir, las características necesarias para representar la mayoría de sistemas de organización del conocimiento, tal y como se observa en la mayoría de los casos de uso [[SKOS-UCR](#)].

En el uso básico de SKOS, los [recursos conceptuales](#) (conceptos) pueden ser identificados con URIs, [etiquetados](#) con cadenas léxicas en una o más lenguas naturales, [documentados](#) con diversos tipos de nota, [relacionarse semánticamente](#) entre sí mediante estructuras jerárquicas informales y redes de asociación y agruparse en esquemas de conceptos.

2.1 Conceptos

El elemento fundamental del vocabulario de SKOS es el *concepto*. Los conceptos son unidades de pensamiento [[WillpowerGlossary](#)]—ideas, significados, o (categorías de) objetos y sucesos—que subyacen a la mayoría de los sistemas de organización del conocimiento [[SKOS-UCR](#)]. Por tanto, los conceptos existen en la mente como entidades abstractas que son independientes de los términos utilizados para etiquetarlos.

SKOS incluye la clase [skos:Concept](#), que permite implementar declaraciones para afirmar que un recurso dado es un concepto. Esto se realiza en dos pasos:

1. creando o reutilizando un Identificador Uniforme de Recurso (URI [[URI](#)]) para identificar individual e inequívocamente al concepto.
2. con una declaración RDF, usando la propiedad [rdf:type](#), que indique que el recurso identificado mediante con dicho URI es del tipo [skos:Concept](#).

Por ejemplo:

```
<http://www.ejemplo.com/animales> rdf:type skos:Concept.
```

Esto también puede representarse en Turtle de un modo más compacto utilizando el prefijo *ex* para el espacio de nombres definido anteriormente:

```
ex:animales rdf:type skos:Concept.
```

El uso de SKOS para publicar esquemas de conceptos facilita las referencias a conceptos en las descripciones de recursos de la Web Semántica. Se anima a los desarrolladores a usar URIs HTTP para declarar URIs de conceptos, ya que esto permitiría el acceso a dichas representaciones utilizando Tecnologías Web estándares. Para obtener más información

acerca de los URIs en la Web Semántica, véase *URIs Atractivos para la Web Semántica* [[COOLURIS](#)] y las *Fórmulas de buenas prácticas para la publicación de vocabularios RDF* [[RECIPES](#)].

2.2 Etiquetas

Las principales caracterizaciones de los conceptos son las expresiones que se utilizan para referirse a ellos en lenguaje natural: sus etiquetas. SKOS ofrece tres propiedades para asignar etiquetas a los recursos conceptuales: [skos:prefLabel](#), [skos:altLabel](#) y [skos:hiddenLabel](#). Cada propiedad implica un estado específico para la etiqueta que muestra, que van desde una fuerte relación de denotación unívoca, a una cadena para ayudar en la búsqueda. Estas propiedades se definen formalmente como pares disjuntos. Esto significa, por ejemplo, que se cometería un error si a un concepto se le asigna el mismo literal, tanto para su etiqueta preferente y como para una etiqueta alternativa.

Como se especifica en la [sección 5](#) de la *Guía de referencia de SKOS*, [skos:prefLabel](#), [skos:altLabel](#) y [skos:hiddenLabel](#) proporcionan etiquetas simples. Todas son sub-propiedades de [rdfs:label](#), y se utilizan para asociar un [skos:Concept](#), con un [literal RDF simple](#) que es una cadena de caracteres (por ejemplo, "amor"), combinada con una etiqueta de idioma opcional (por ejemplo, "es-ES") [[RDF-CONCEPTS](#)].

2.2.1 Etiquetas léxicas preferentes

La propiedad [skos:prefLabel](#) permite asignar una etiqueta léxica preferente a un recurso. Los términos utilizados como *descriptores* en los sistemas de indización [[WillpowerGlossary](#)] se representarán mediante esta propiedad, como se muestra en el ejemplo siguiente:

```
ex:animales rdf:type skos:Concept;  
            skos:prefLabel "animales".
```

Los literales RDF simples se definen formalmente como cadenas de caracteres con etiquetas opcionales para denotar el idioma. SKOS permite una forma sencilla para el etiquetado multilingüe. Esto se consigue utilizando una etiqueta de idioma en una etiqueta léxica para restringir su alcance a un idioma determinado. El siguiente ejemplo ilustra cómo se asocia a un concepto una denominación preferida en Español y otra en Inglés:

```
ex:animales rdf:type skos:Concept;  
            skos:prefLabel "animales"@es;  
            skos:prefLabel "animals"@en.
```

Hay que tener en cuenta que la noción de etiqueta *preferente* implica que un recurso sólo puede tener una de estas etiquetas por idioma, tal como se explica en la [sección 5](#) de la *Guía de referencia de SKOS* [[SKOS-REFERENCE](#)].

Siguiendo la práctica común en el diseño de sistemas de organización del

conocimiento, la etiqueta preferente de un concepto también puede utilizarse para representar de forma inequívoca al mismo dentro de estos sistemas y sus aplicaciones. Así que, aunque el modelo de datos de SKOS no obligue formalmente a ello, se recomienda que no existan dos conceptos en el mismo sistema de organización que tengan la misma etiqueta léxica preferente para un mismo idioma.

2.2.2 Etiquetas léxicas alternativas

La propiedad [skos:altLabel](#) permite asignar una etiqueta léxica alternativa a un concepto. Esto es especialmente útil cuando se asignan otras etiquetas diferentes de la etiqueta preferente del concepto, por ejemplo, para representar sinónimos:

```
ex:animales rdf:type skos:Concept;  
skos:prefLabel "animales"@es;  
skos:altLabel "criaturas"@es;  
skos:prefLabel "animals"@en;  
skos:altLabel "creatures"@en.
```

Tenga en cuenta que la representación de sinónimos para las etiquetas preferentes no es el único uso para [skos:altLabel](#). Los cuasi-sinónimos, abreviaturas y acrónimos pueden representarse de la misma manera:

```
ex:onu rdf:type skos:Concept;  
skos:prefLabel "Organización de las Naciones Unidas"@es;  
skos:altLabel "ONU"@es.
```

Nota sobre la indicación anterior: También es posible utilizar [skos:altLabel](#) para representar los casos indicados anteriormente [[ISO-2788](#)]. Es decir, cuando un concepto agrupa nociones más especializadas que no están explícitamente incluidas como conceptos en un sistema de organización del conocimiento concreto:

```
ex:rocas rdf:type skos:Concept;  
skos:prefLabel "rocas"@es;  
skos:altLabel "basalto"@es;  
skos:altLabel "granito"@es;  
skos:altLabel "pizarra"@es.
```

Sin embargo, a pesar de que SKOS no pretende sustituir a las directrices existentes para el diseño de los sistemas de organización del conocimiento [[ISO-2788](#), [BS8723-2](#)], el lector debe ser consciente de que la indicación anterior no es recomendable. Un sistema de organización más apropiado para este dominio debería incluir un [skos:Concept](#) para cada tipo de roca (basalto, granito y pizarra) y declararlos como conceptos específicos de `ex:roca`.

2.2.3 Etiquetas léxicas ocultas

Una etiqueta léxica oculta se representa a través de la propiedad [skos:hiddenLabel](#), asociándose a un recurso, cuando el editor de un sistema de organización precisa que determinadas expresiones estén disponibles para aplicaciones que realizan procesos de indexación basados en el texto

completo de documentos y operaciones de búsqueda, pero que **no** desea que dichas etiquetas sean visibles de otra forma. Las etiquetas ocultas pueden ser usadas, por ejemplo, para incluir variantes de errores ortográficos de otras etiquetas léxicas. Por ejemplo:

```
ex:animales rdf:type skos:Concept;  
skos:prefLabel "animales"@es;  
skos:altLabel "bestias"@es;  
skos:hiddenLabel "béstias"@es.
```

2.3 Relaciones semánticas

En los sistemas de organización del conocimiento, las relaciones semánticas desempeñan un papel crucial en la definición de conceptos. El significado de un concepto se define no sólo por las palabras de la lengua natural de sus etiquetas, sino también por sus vínculos con otros conceptos del vocabulario. Es un reflejo de las categorías fundamentales de las relaciones que se utilizan en vocabularios, tales como los tesauros [[ISO2788](#)]. SKOS proporciona tres propiedades estándar:

- [skos:broader](#) y [skos:narrower](#) permiten la representación de los vínculos jerárquicos, como la relación entre un género y sus especies más específicas, o, dependiendo de la interpretación, la relación entre un todo y sus partes
- [skos:related](#) permite la representación de vínculos asociativos (no jerárquicos), como la relación entre un tipo de evento y una categoría de entidades que suelen participar en ella. Otro uso de [skos:related](#) es la asociación de dos categorías entre las que ninguna es más general o más específica que la otra. Tenga en cuenta que [skos:related](#) permite la representación de vínculos asociativos (no jerárquicos), que también pueden usarse para representar vínculos parte-todo cuyo significado no se ajusta al de las relaciones jerárquicas.

2.3.1 Relaciones genéricas/específicas

Para indicar que un concepto tiene un significado más amplio (es decir, más en general) que otro, se utiliza la propiedad [skos:broader](#). La propiedad [skos:narrower](#) se utiliza para realizar la afirmación inversa, es decir, cuando un concepto tiene un significado más concreto (es decir, más específico) que otro. Por ejemplo:

```
ex:animales rdf:type skos:Concept;  
skos:prefLabel "animales"@es;  
skos:narrower ex:mamíferos.  
ex:mamíferos rdf:type skos:Concept;  
skos:prefLabel "mamíferos"@es;  
skos:broader ex:animales.
```

Como sucede a menudo en los sistemas de organización del conocimiento, un concepto SKOS puede conectarse a varios conceptos más amplios al mismo tiempo. Por ejemplo, un concepto `ex:perro` podría tener a `ex:mamíferos` y `ex:animalesDomesticados` como conceptos genéricos.

Nota sobre el sentido de `skos:broader` : por razones históricas, el nombre de la propiedad `skos:broader` (la expresión "broader"/"genérico") no ofrece una indicación explícita de su dirección. La expresión "genérico" debería entenderse como "tiene un concepto más amplio"; el sujeto de una declaración `skos:broader` es el concepto más específico implicado en la declaración y su objeto es el más genérico.

Nota sobre las declaraciones implícitas `skos:broader/skos:narrower` : las propiedades `skos:broader` y `skos:narrower` son mutuamente inversas. Siempre que un concepto X sea genérico de otro concepto Y, entonces Y es un concepto específico de X, de acuerdo con el modelo de datos de SKOS [[SKOS-REFERENCE](#)]. Esto puede ser útil para hacer representaciones de SKOS más eficientes limitando la información que contienen. En el ejemplo anterior, por citar un caso, la declaración `ex:mamíferos skos:broader ex:animales` puede excluirse si, antes de usar los datos del esquema de conceptos, se aplica un razonador OWL para inferir a partir de la declaración `ex:animales skos:narrower ex:mamíferos`.

En muchos casos, las relaciones jerárquicas de un esquema de conceptos pueden considerarse [transitivas](#) [[OWL](#)]. Si, por ejemplo, `ex:animales` es más genérico que `ex:mamíferos`, que a su vez también es más genérico que `ex:gatos`, tiene sentido afirmar que `ex:animales` es más amplio que `ex:gatos`. Sin embargo, hay jerarquías "incorrectas", especialmente en aquellos sistemas de organización que difieren de los tesauros bien diseñados según los estándares, en los que tal función no se considera apropiada. Consideremos, por ejemplo, un caso en el que se afirma que `ex2:vehículos` es más amplio que `ex2:coches`, que a su vez es más amplio que `ex2:ruedas`. Puede resultar problemático deducir automáticamente que "ruedas" es un concepto más específico con respecto a "vehículos". SKOS prevé este tipo de problemas al no haber definido de forma general la propiedad transitiva para `skos:broader` y `skos:narrower`. El lector interesado en la representación de "jerarquías transitivas", debería consultar la [Sección 4.5](#), que muestra cómo definir este tipo de jerarquías conservando la compatibilidad con la semántica de `skos:broader` definida en esta sección.

Nota acerca de la transitividad frente a la intransitividad: el modelo SKOS no indica que `skos:broader` y `skos:narrower` sean transitivas. Sin embargo, esto no implica que estas propiedades sean intransitivas. Considere el concepto `gatos` que es más específico que el concepto `mamíferos`, que a su vez es más específico que `animales`: se podrían afirmar que `gatos` también es más específico que `animales` y al mismo tiempo mantenerse compatible con el modelo SKOS. La no declaración de `skos:broader` como propiedad transitiva implica que no puede inferirse la propiedad `skos:broader` entre `gatos` y `animales` mediante la aplicación de los axiomas de SKOS. Esto no impide que los editores de un esquema de conceptos SKOS puedan afirmar que las declaraciones de propiedades jerárquicas reflejan un comportamiento transitivo a nivel local.

Del mismo modo, SKOS no asume que las relaciones jerárquicas sean irreflexivas por defecto. En las directrices de muchos tesauros está prohibido declarar un concepto como genérico de sí mismo. Sin embargo, en casos específicos, más allá de los tesauros clásicos, puede darse la reflexividad en algunas declaraciones de `skos:broader`. Considere la posibilidad de la conversión de una ontología RDFS/OWL en un esquema de conceptos SKOS. En tal caso, es válido que cada declaración `rdfs:subClassOf` sea reinterpretada como `skos:broader`. Sin embargo, `rdfs:subClassOf` es una propiedad reflexiva, lo que significa que por cada clase `C`, la declaración `C:subClassOf C` es [cierta \[OWL\]](#). En este caso, todos los conceptos serían, por lo tanto, conceptos genéricos de sí mismos.

En SKOS básico no se contemplan distintos tipos de relación jerárquica, como por ejemplo, las relaciones instancia-clase y parte-todo. El lector interesado puede consultar la [sección 4.7](#), que describe cómo crear especializaciones de relaciones semánticas para hacer frente a este problema.

2.3.2 Relaciones asociativas

Para denotar una relación de asociación entre dos conceptos puede utilizarse [skos:related](#):

```
ex:pájaros rdf:type skos:Concept;  
  skos:prefLabel "pájaros"@es;  
  skos:related ex:ornitología  
ex:ornitología rdf:type skos:Concept;  
  skos:prefLabel "ornitología"@es.
```

Como se describe en la *Guía de referencia de SKOS* [[SKOS-REFERENCE](#)], la propiedad `skos:related` es [simétrica \[OWL\]](#). Del grafo RDF anterior, puede deducirse, por ejemplo, que `ex:ornitología` es el sujeto de una declaración `skos:related` que tiene como objeto a `ex:pájaros`.

Nota sobre la (no-)transitividad de `skos:related`: El lector debe ser consciente que en el modelo de datos de SKOS `skos:related` no se define como una propiedad transitiva. Una relación `skos:related` transitiva podría tener consecuencias no deseadas, como la del ejemplo siguiente:

```
ex:renacimiento skos:related ex:humanismo.  
ex:humanismo skos:related ex:antropologíaFilosófica.  
ex:antropologíaFilosófica skos:related ex:filosofíaMental.  
ex:filosofíaMental skos:related ex:cienciasCognitivas.
```

En caso de que `skos:related` fuera una propiedad transitiva, `ex:renacimiento` estaría directamente relacionado con `ex:cienciasCognitivas`. Si bien cada declaración individual tiene sentido, la declaración inferida no tiene cabida en las intenciones originalmente previstas por el editor del sistema de organización.

Nota sobre la mezcla de estructuras jerárquicas y asociativas: La clausura transitiva `skos:broader` es disyunta de `skos:related`. Si los recursos A y B están relacionados a través de `skos:related`, no debe

establecerse una cadena de relaciones `skos:broader` entre A y B. Esto último también sirve para `skos:narrower`.

2.4 Notas de documentación

Las relaciones semánticas son cruciales para la definición de conceptos, tal y como hacen hincapié muchas directrices sobre sistemas de organización del conocimiento. Sin embargo, junto a estas caracterizaciones estructurales, los conceptos a veces tienen que definirse con mayor precisión utilizando documentación ("informal") que sea legible para las personas, como *notas de alcance* o *definiciones*.

SKOS proporciona la propiedad [skos:note](#) para fines de documentación en general. Inspirado por las actuales directrices de estos sistemas, tales como [\[ISO2788\]](#) o [\[BS8723-2\]](#), esta propiedad se especializa en otras como `skos:scopeNote`, `skos:definition`, `skos:example`, y `skos:historyNote` que permiten ajustar más los tipos específicos de documentación.

[skos:scopeNote](#) suministra información, posiblemente parcial, sobre el significado de un concepto, sobre todo como una indicación de cómo se limita su uso en los procesos de indización. El siguiente ejemplo está adaptado de [\[ISO2788\]](#):

```
ex:frecuenciasMicroondas skos:scopeNote
  "Utilizado para frecuencias entre 1GHz y 300Ghz"@es.
```

[skos:definition](#) proporciona una explicación completa del significado de un concepto. El siguiente ejemplo está adaptado de [\[ISO2788\]](#):

```
ex:documentación skos:definition
  "Proceso de almacenamiento y recuperación de información
  en todos los campos del conocimiento"@es.
```

[skos:example](#) suministra un ejemplo de uso de un concepto:

```
ex:organizacionesCienciaCultura skos:example
  "Académicas científicas, museos generales, ferias mundiales"@es.
```

[skos:historyNote](#) describe cambios significativos en el significado o la forma de un concepto:

```
ex:abusoNiños skos:historyNote
  "establecido en 1975; previamente se usaba: Crueldad con los niños [1952-1975]"@es.
```

Además de estas notas, que están destinadas a los usuarios de un esquema de conceptos, SKOS incluye dos especializaciones de `skos:note` de utilidad para los gestores o editores de sistemas de organización: `skos:editorialNote` y `skos:changeNote`.

[skos:editorialNote](#) suministra información de mantenimiento, tales como recordatorios del trabajo editorial que queda por hacer, o advertencias para el caso de futuros cambios editoriales que pudieran hacerse:

```
ex:dobleclick skos:editorialNote "Revisar este término después de
  completar la fusión de la compañía"@es.
```

```
ex:folksonomía skos:editorialNote "Comprobar la ortografía con Thomas Vander Wal"@es.
```

[skos:changeNote](#) documenta aquellos cambios muy particulares de un concepto, a efectos de administración y mantenimiento:

```
ex:tomate skos:changeNote
  "Clasificado previamente bajo 'frutas' y actualmente bajo 'vegetales' por Horace Gray"@es.
```

Es importante notar que el vínculo jerárquico entre `skos:note` y sus diferentes especializaciones permite que toda la documentación asociada a un concepto sea recuperada de manera directa. Cada `skos:definition` es una `skos:note`, cada `skos:scopeNote` es un `skos:note`, y así sucesivamente.

Como se ejemplifica anteriormente, en las propiedades de documentación de SKOS pueden usarse literales RDF simples. La [sección 4.2](#) muestra que hay otros patrones posibles, puesto que el rango de estas propiedades no puede limitarse a literales. Sin embargo, una característica importante de los literales simples es la capacidad de utilizar etiquetas de idioma, como en el etiquetado de propiedades. Por tanto, la documentación puede estar disponible en varios idiomas:

```
ex:piñas rdf:type skos:Concept;
  skos:prefLabel "piñas"@es;
  skos:prefLabel "pineapples"@en;
  skos:definition "Fruta de un tipo de plantas de la familia de las Bromeliáceas"@es;
  skos:definition "The fruit of plants of the family Bromeliaceae"@en.
```

Antes de concluir esta sección, es importante señalar que otras propiedades no pertenecientes a SKOS podrían utilizarse para documentar los conceptos. La propiedad [dct:creator](#) de Dublin Core [[DC](#)], por ejemplo, puede ser utilizada para indicar la persona que creó el siguiente concepto:

```
ex:águilaPescadoraMadagascar dct:creator [ foaf:name "John Smith" ].
```

2.5 Esquemas de conceptos

Los conceptos pueden crearse y utilizarse como entidades independientes. Sin embargo, especialmente durante la indización, los conceptos generalmente, se asocian a un vocabulario cuidadosamente recopilado, tales como tesauros o esquemas de clasificación. SKOS ofrece los mecanismos para representación de tales sistemas, empleando la clase [skos:ConceptScheme](#).

El siguiente ejemplo muestra la definición de un esquema de conceptos (que representa un tesoro) y la descripción de este recurso mediante [dct:title](#) y [dct:creator](#), ambas propiedades de Dublin Core [[DC](#)]:

```
ex:tesauroAnimales rdf:type skos:ConceptScheme;
  dct:title "Sencillo tesoro sobre animales";
  dct:creator ex:antoineIsaac.
```

Una vez creado el esquema de conceptos, éste puede enlazarse con los conceptos que contiene utilizando la propiedad [skos:inScheme](#):

```
ex:mamíferos rdf:type skos:Concept;
```

```
skos:inScheme ex:tesauroAnimales.  
ex:vacas rdf:type skos:Concept;  
skos:broader ex:mamíferos;  
skos:inScheme ex:tesauroAnimales.  
ex:peces rdf:type skos:Concept;  
skos:inScheme ex:tesauroAnimales.
```

Con el fin de proporcionar un acceso eficaz a los puntos de entrada de las jerarquías de conceptos genéricos/específicos, SKOS define la propiedad [skos:hasTopConcept](#). Esta propiedad permite vincular un esquema de conceptos a uno o varios conceptos cabecera, como en el ejemplo siguiente del tesauro de animales:

```
ex:tesauroAnimales rdf:type skos:ConceptScheme;  
skos:hasTopConcept ex:mamíferos;  
skos:hasTopConcept ex:peces.
```

Los esquemas de conceptos están diseñados para representar vocabularios tradicionales, y se anima a los diseñadores a seguir las directrices existentes de elaboración de sistemas de organización del conocimiento (por ejemplo, [\[ISO2788\]](#) o [\[BS8723-2\]](#)) cuando elaboren un esquema de conceptos basado en SKOS. Por ejemplo, como se describe en la [sección 2.2](#), se recomienda que no haya dos conceptos con la misma etiqueta léxica preferente en un idioma determinado, cuando pertenezcan a un mismo esquema de conceptos.

Sin embargo, el lector debe ser consciente de la existencia de algunas diferencias sutiles entre los esquemas de conceptos de SKOS y los sistemas de organización "tradicionales", principalmente debido al contexto de la Web Semántica en el que se desenvuelve SKOS. La [sección 4.6](#) de la *Guía de referencia de SKOS* [[SKOS-REFERENCE](#)] da cuenta de estas diferencias. Una característica importante de SKOS es posibilitar que el mismo concepto esté asociado a varios esquemas usando la propiedad `skos:inScheme`. Esto se abordará en la próxima sección.

Por último, es importante notar que el vocabulario de SKOS sólo ofrece un apoyo limitado para la representación de la información de un sistema de organización del conocimiento a través de un esquema de conceptos.

`skos:inScheme` y `skos:hasTopConcept` vinculan conceptos y esquemas de conceptos. Sin embargo, aún no existe ningún mecanismo en SKOS para dejar constancia de que una declaración específica sobre estos conceptos, por ejemplo una declaración `skos:broader`, se refiere a un determinado esquema de conceptos, mientras que un sistema de organización del conocimiento generalmente es visto como una estructura que consta tanto de conceptos como de los vínculos que los definen. El lector interesado puede consultar la [sección 5.3](#) donde se debate este tema.

3 Redes de Sistemas de Organización del Conocimiento en la Web Semántica

Representar un Sistema de Organización del Conocimiento mediante SKOS no sólo sirve como un mecanismo de publicación, sino que también le permite participar en una red de esquemas de conceptos. En la Web

Semántica el verdadero potencial de los datos se desencadena cuando éstos se interrelacionan. Igualmente, cuando los conceptos de diferentes esquemas de conceptos se conectan entre sí comienzan a formar un esquema de conceptos distribuido y heterogéneo. Una red de esquemas puede servir de base para nuevas aplicaciones que permitan una navegación eficaz entre diferentes sistemas de organización. Esta sección presenta las características de SKOS que permiten la interconexión de esquemas de conceptos y explica cómo se relacionan los recursos conceptuales con otros recursos en la Web Semántica.

3.1 Mapeado de esquemas de conceptos

A cada concepto SKOS se le asigna un URI [[COOLURIS](#)], que permite de forma inequívoca referenciar un concepto en cualquier aplicación de SKOS. Esto puede ser especialmente útil para establecer relaciones semánticas entre conceptos preexistentes. Estas asignaciones son cruciales para aplicaciones tales como herramientas de recuperación de información que utilizan varios sistemas de organización al mismo tiempo, donde se solapan los ámbitos de dichos sistemas y deban ser semánticamente compatibles; al respecto pueden encontrarse ejemplos en documento sobre los *Caso de uso y requisitos de SKOS* [[SKOS-UCR](#)].

Una característica fundamental del mapeado es la posibilidad de afirmar que dos conceptos de diferentes esquemas tienen un significado similar, y especificar en qué medida lo son, incluso proviniendo de contextos diferentes y habiendo seguido posiblemente principios de modelado distintos [[BS8723-4](#)]. Se espera que el mapeado conceptual constituya una ventaja clave que haga posible que los sistemas de organización del conocimiento estén disponibles en la Web Semántica mediante SKOS.

SKOS proporciona varias propiedades para mapear conceptos de diferentes esquemas. Esto puede hacerse afirmando que dos conceptos tienen un significado similar utilizando las propiedades [skos:exactMatch](#) y [skos:closeMatch](#) . Dos conceptos de esquemas distintos también pueden ser mapeados utilizando propiedades paralelas a las utilizadas para expresar relaciones semánticas que se presentaron en la [sección 2.3](#): [skos:broadMatch](#), [skos:narrowMatch](#) y [skos:relatedMatch](#).

Considere el siguiente ejemplo, donde dos esquemas de conceptos muestran diferentes puntos de vista sobre los animales:

```
ex1:esquemaReferenciaAnimales rdf:type skos:ConceptScheme;
  dct:title "Extensa lista de animales"@es.
ex1:animal rdf:type skos:Concept;
  skos:prefLabel "animal"@es;
  skos:inScheme ex1:esquemaReferenciaAnimales.
ex1:ornitorrinco rdf:type skos:Concept;
  skos:prefLabel "ornitorrinco"@es;
  skos:inScheme ex1:esquemaReferenciaAnimales.

ex2:esquemaVentaHuevos rdf:type skos:ConceptScheme;
  dct:title "Vocabulario sobre la venta de huevos"@es.
ex2:animalesPonenHuevos rdf:type skos:Concept;
  skos:prefLabel "animales que ponen huevos"@es;
```

```
skos:inScheme ex2:esquemaVentaHuevos.  
ex2:animales rdf:type skos:Concept;  
skos:prefLabel "animales"@es;  
skos:inScheme ex2:esquemaVentaHuevos.  
ex2:huevos rdf:type skos:Concept;  
skos:prefLabel "huevos"@es;  
skos:inScheme ex2:esquemaVentaHuevos.
```

Es posible mapear los conceptos de `ex1:esquemaReferenciaAnimales` con los conceptos de `ex2:esquemaVentaHuevos` usando la siguiente declaración de mapeado:

```
ex1:ornitorrinco skos:broadMatch ex2:animalesPonenHuevos.  
ex1:ornitorrinco skos:relatedMatch ex2:huevos.  
ex1:animal skos:exactMatch ex2:animales.
```

Una declaración `skos:closeMatch` indica que dos conceptos son suficientemente similares, pudiendo usarse indistintamente en aplicaciones que operen con los dos esquemas a los que pertenecen. No obstante, la propiedad `skos:closeMatch` no está definida como transitiva, lo cual previene que dicha similitud se propague más allá de dichos esquemas de conceptos: esto **impide** que si un concepto `ex1:A` coincide exactamente con otro `ex2:B` y éste a su vez se corresponde con `ex3:C`, pueda deducirse a partir de modelo de datos de SKOS que `ex1:A` coincide con `ex3:C`.

Nota sobre `skos:exactMatch` frente a `owl:sameAs`: SKOS ofrece `skos:exactMatch` para mapear conceptos con un significado equivalente, y deliberadamente no utiliza `owl:sameAs` del lenguaje de ontologías OWL [[OWL](#)]. Cuando dos recursos están vinculados con `owl:sameAs` se considera que son el mismo recurso, y las tripletas RDF en las que participan estos recursos se combinan. Esto no se ajusta a lo que se necesita en la mayoría de las aplicaciones de SKOS. En el ejemplo anterior, `ex1:animal` se dice que es equivalente a `ex2:animales`. Si esta relación de equivalencia se representara usando `owl:sameAs`, las siguientes afirmaciones serían válidas para `ex:animal`:

```
ex1:animal rdf:type skos:Concept;  
skos:prefLabel "animal"@es;  
skos:inScheme ex1:esquemaReferenciaAnimales.  
skos:prefLabel "animales"@es;  
skos:inScheme ex2:esquemaVentaHuevos.
```

Esto haría inconsistente a `ex:animal`, ya que un concepto no puede poseer dos etiquetas preferentes distintas en la misma lengua. Los conceptos tienen asociado otro tipo de información, tales como las relaciones semánticas con otros conceptos, o las notas, éstas se fusionarían también, causando que estos conceptos adquirieran nuevos significados.

Por convención, las propiedades de mapeado se utilizan para representar los vínculos que tienen el mismo significado intencional que las propiedades semánticas "estándares", pero con un ámbito de aplicación diferente. Se podría decir que las relaciones de mapeo son menos *inherentes* al significado de los conceptos que implican. Desde el punto de vista del diseñador original del sistema de organización el mapeado podría incluso ser incorrecto.

Se espera que las propiedades de mapeado sean útiles en aplicaciones concretas que utilizan varios sistemas conceptualmente superpuestos. Por convención, se espera que las relaciones de mapeado se apliquen entre conceptos de diferentes esquemas de conceptos.

El lector debe ser consciente de que, según el modelo de datos de SKOS, las propiedades de mapeado que "reflejan" determinadas propiedades relativas a relaciones semánticas también son sub-unidades de la misma en el sentido proporcionado por RDFS. Por ejemplo, `skos:broadMatch` es una sub-propiedad de `skos:broader`. En consecuencia, toda declaración de `skos:broadMatch` entre dos conceptos, conduce a deducir la propiedad `skos:broader` entre estos conceptos.

3.2 Reutilización y extensión de esquemas de conceptos

La vinculación de conceptos por medio de asignaciones no es el único modo de interconectar esquemas de conceptos. El uso de URIs en la Web Semántica permite compartir y reutilizar recursos de manera distribuida. Como resultado, es posible que un concepto SKOS participe en varios esquemas al mismo tiempo. Por ejemplo, un editor de SKOS puede optar por ampliar localmente un esquema de conceptos existente declarando nuevos conceptos que puedan ser necesarios y vincularlos simplemente a los conceptos del esquema ya existente.

La extensión de un sistema de organización puede ser especialmente útil cuando sus diseñadores (o editores externos de otros sistemas de este tipo) quieran conseguir una mejor cobertura de un dominio o subdominio, siguiendo los principios que guiaron el diseño del sistema existente—por ejemplo, reutilizando algunos de sus conceptos. La extensión y reutilización explícita de un sistema puede utilizarse como un mecanismo de modularización, cuando un conjunto coordinado de sistemas de organización (por ejemplo microtesauros que pertenecen a un vocabulario general) se diseña para cubrir varios ámbitos y sus diseñadores quieren permitir que aplicaciones específicas puedan trabajar con un determinado subconjunto de conceptos.

Un nuevo esquema de conceptos puede reutilizar conceptos existentes empleando la propiedad [skos:inScheme](#). Considere el siguiente ejemplo, donde un primer esquema de conceptos sobre animales define un concepto para "gatos":

```
ex1:esquemaReferenciaAnimales rdf:type skos:ConceptScheme;
  dct:title "Lista de referencia de animales"@es.
ex1:gatos rdf:type skos:Concept;
  skos:prefLabel "gatos"@es;
  skos:inScheme ex1:esquemaReferenciaAnimales.
```

El creador de otro esquema de conceptos que contiene descripciones de gato puede libremente referenciar al concepto `ex1:gatos` en su esquema, y a continuación hacer referencia a él de la siguiente forma:

```
ex2:esquemaGatos rdf:type skos:ConceptScheme;
```

```
dct:title "Completo tesoro sobre los gatos"@es.  
  
ex1:gatos skos:inScheme ex2:esquemaGatos.  
  
ex2:abisinio rdf:type skos:Concept;  
  skos:prefLabel "Gatos Abisinios"@es;  
  skos:broader ex1:gatos;  
  skos:inScheme ex2:esquemaGatos.  
  
ex2:siamés rdf:type skos:Concept;  
  skos:prefLabel "Gatos Siameses"@es;  
  skos:broader ex1:gatos;  
  skos:inScheme ex2:esquemaGatos.
```

Tenga en cuenta que la fuente de información que define el nuevo esquema de conceptos no replica información acerca del concepto `ex1:gatos`, como podría ser el caso de su etiqueta preferente. Suponiendo que `ex1:gatos` se publique, una aplicación de la Web Semántica sería capaz de recuperar la información de este concepto simplemente resolviendo la URI del concepto (<http://www.ejemplo.com/1/gatos>).

Nota sobre `owl:imports` y la reutilización de sistemas de organización del conocimiento: La propiedad [`owl:imports`](#) proporciona un mecanismo para la importación de las declaraciones de una ontología OWL en otra. `owl:imports` puede usarse con vocabularios SKOS para proporcionar un caso especial de reutilización/extensión en el que un esquema de conceptos "importe" en su totalidad otro esquema. Continuando con el ejemplo anterior, esto se logra mediante la inclusión de la siguiente declaración en la definición de `ex2:esquemaGatos`:

```
ex2:esquemaGatos owl:imports ex1:esquemaReferenciaAnimales.
```

Usar `owl:imports` de esta forma tiene algunas consecuencias. En primer lugar, el dominio y el rango de `owl:imports` es `owl:Ontology`, mientras que `skos:ConceptScheme` se define como `owl:Class`. Así, afirmar que un esquema de conceptos es importado por otro a través de `owl:imports` conduce a la consecuencia de que las instancias de `skos:conceptScheme` que participan en la importación también son inferidas como instancias de `owl:Ontology`. Esto a su vez conduce a una ontología OWL Full (debido a la doble utilización de un URI como clase y ontología, véase la [sección 4.2](#) del documento sobre la *Semántica de OWL [OWL-SEMANTICS]*).

En segundo lugar, en virtud de la semántica de OWL Full (ver [sección 5.3](#) sobre la *Semántica de OWL [OWL-SEMANTICS]*), la interpretación pretendida de `owl:imports` es que el grafo RDF obtenido del URI a importar se añada al grafo existente desde el que se realiza este proceso. Los usuarios deben ser conscientes de ello, y debe evitarse cualquier otra interpretación alternativa. En particular, no hay dependencia lógica entre `skos:inScheme` y `owl:imports`: el uso de `owl:imports` no dará lugar a la presencia de cualquier declaración `skos:inScheme` distinta a la ya existente en el grafo importado. Si consideramos el ejemplo anterior, `owl:imports` se ha utilizado para afirmar que un esquema de conceptos importa

lógicamente a otro. Pero a pesar de que `ex1:esquemaReferenciaAnimales` contenga la declaración

```
ex1:elefante skos:inScheme ex1:esquemaReferenciaAnimales.
```

de la afirmación

```
ex1:elefante skos:inScheme ex2:esquemaGatos.
```

no debe deducirse que está presente en el grafo de la definición de `ex2:esquemaGatos`.

Si una aplicación se relaciona sobre la procedencia en la práctica o pertenencia de la información, puede que se requieran medidas adicionales para mantener el origen o la autoría de las tripletas importadas, tal y como se menciona en la [sección 5.3](#).

3.3 Indización temática y SKOS

Aunque formalmente no pertenezca a las características que definen un sistema de organización, el vínculo entre un concepto y los recursos que tratan sobre el mismo es fundamental en muchas aplicaciones de estos sistemas, como la indización y recuperación de documentos. Esto es aún más importante en el contexto de la Web Semántica, donde hay una necesidad crucial de anotar los documentos con las unidades conceptuales que definen su objeto.

Mientras que el vocabulario de SKOS en sí mismo no incluye un mecanismo para la asociación de cualquier recurso con un `skos:Concept`, los desarrolladores pueden recurrir a otros vocabularios. Dublin Core, por ejemplo, proporciona la propiedad `dct:subject` [[DC](#)]:

```
ex1:ornitorrinco rdf:type skos:Concept;  
skos:prefLabel "ornitorrinco"@es.
```

```
<http://es.wikipedia.org/wiki/Ornitorrinco> rdf:type foaf:Document;  
dct:subject ex1:ornitorrinco.
```

Nótese que un solo recurso puede tratar diversos temas, y por lo tanto, estar involucrado en varias declaraciones `dct:subject`. Evidentemente, estos temas pueden provenir de esquemas diferentes, resultantes, por ejemplo, de un proceso de anotación distribuido.

4 SKOS Avanzado: cuando los sistemas de organización del conocimiento no son tan simples

Más allá de las características antes mencionadas, SKOS propone una serie de elementos de vocabulario o directrices que se ocupan de las necesidades de representación más avanzadas, haciendo a SKOS compatible con una amplia gama de enfoques de modelos para sistemas de organización del conocimiento. Estos son especialmente diseñados para satisfacer las necesidades que se plantearon en los *Casos de Uso* y *requisitos de SKOS*

[[SKOS-UCR](#)], pero que estaban presentes sólo en un número menor de casos de uso:

- Agrupación de conceptos basados en criterios específicos,
- Documentación avanzada por medio de recursos complejos,
- Establecimiento de relaciones entre las etiquetas de los conceptos,
- Creación de conceptos complejos a partir de otros sencillos (coordinación),
- Acceso transitivo a relaciones jerárquicas,
- Representación de notaciones de conceptos.

Esta sección concluye con una nota general sobre la extensibilidad del modelo de SKOS, allanando el camino a mejoras aún más especializados del vocabulario presentado en este manual.

4.1 Colecciones de conceptos

SKOS permite definir grupos significativos o "colecciones" de conceptos. Tales agrupaciones son normalmente mostradas en los tesauros como en el ejemplo siguiente:

```
leche
  <leche de origen animal>
    leche de vaca
    leche de cabra
    leche de búfalo
```

Estas colecciones pueden utilizarse para representar "conjuntos" en la terminología de desarrollo de tesauros, en la que el término "leche de origen animal" es la "etiqueta de un nodo" [[WillpowerGlossary](#)]. Hay un consenso acerca de que la etiqueta de un nodo **no** representa una etiqueta para un concepto por derecho propio. Por lo tanto, deben introducirse entidades específicas para representarlas.

Colecciones etiquetadas

Para modelar correctamente tales estructuras de colecciones de conceptos, SKOS introduce la clase [skos:Collection](#). Las instancias de esta clase agrupan conceptos específicos mediante la propiedad [skos:member](#), como se muestra en el ejemplo siguiente:

```
ex:leche rdf:type skos:Concept;
  skos:prefLabel "leche"@es.
ex:lecheVaca rdf:type skos:Concept;
  skos:prefLabel "leche de vaca"@es;
  skos:broader ex:leche.
ex:lecheCabra rdf:type skos:Concept;
  skos:prefLabel "leche de cabra"@es;
  skos:broader ex:leche.
ex:lecheBúfalo rdf:type skos:Concept;
  skos:prefLabel "leche de búfalo"@es;
  skos:broader ex:leche.

_:b0 rdf:type skos:Collection;
  skos:prefLabel "leche de origen animal"@es;
```

```
skos:member ex:lecheVaca;  
skos:member ex:lecheCabra;  
skos:member ex:lecheBúfalo.
```

Tenga en cuenta que en el ejemplo anterior, la colección se define como un nodo vacío, es decir, no se define un URI asociada al mismo. Los URIs pueden asignarse a las colecciones, pero generalmente esto no es necesario. Además, se utiliza `skos:prefLabel` para asignar una etiqueta léxica a la colección, ya que esta propiedad (como otras propiedades de etiquetado de SKOS) se puede utilizar con los recursos no conceptuales.

Colecciones ordenadas

A veces es importante captar el orden de los conceptos en una colección, como cuando se enumeran en orden alfabético o cronológico. Para definir una colección ordenada de conceptos se usa la clase [skos:OrderedCollection](#), junto con la propiedad [skos:memberList](#). Esta propiedad vincula una instancia de `skos:OrderedCollection` a un nodo de tipo `rdf:List` (posiblemente vacío), siguiendo el modelo que permite la definición de las [colecciones RDF \[RDF-PRIMER\]](#). Por ejemplo:

```
ex:bebés rdf:type skos:Concept;  
  skos:prefLabel "bebés"@es.  
ex:niños rdf:type skos:Concept;  
  skos:prefLabel "niños"@es.  
ex:adultos rdf:type skos:Concept;  
  skos:prefLabel "adultos"@es.  
  
_:b0 rdf:type skos:OrderedCollection;  
  skos:prefLabel "Personas según la edad"@es;  
  skos:memberList _:b1.  
_:b1 rdf:first ex:bebés;  
  rdf:rest _:b2.  
_:b2 rdf:first ex:niños;  
  rdf:rest _:b3.  
_:b3 rdf:first ex:adultos;  
  rdf:rest rdf:nil.
```

Colecciones SKOS, Relaciones Semánticas y Presentaciones Sistemáticas

Hay que tener en cuenta que, según el modelo de datos de SKOS, las colecciones son disyuntas de los conceptos. Por tanto, es imposible utilizar las relaciones semánticas de SKOS (ver [sección 2.3](#)) para ubicar una colección en una red semántica de SKOS. En otras palabras, las agrupaciones de conceptos en colecciones no sustituyen a las declaraciones sobre la localización de los conceptos en un esquema de conceptos. En el ejemplo anterior sobre la "leche" todos los orígenes definidos de distintos tipos de leche deben vincularse explícitamente a un concepto más genérico `ex:leche` utilizando la propiedad `skos:broader`:

```
ex:lecheVaca skos:broader ex:leche.  
ex:lecheCabra skos:broader ex:leche.  
ex:lecheBúfalo skos:broader ex:leche.
```

Por tanto, es posible generar una presentación sistemática (jerárquica) que

incluya la agrupación de conceptos "leche de origen animal", tal como se presenta en el ejemplo de la introducción de esta subsección. La jerarquía `skos:broader` y la información sobre la pertenencia a una colección puede utilizarse para ello, pero este proceso todavía requiere un algoritmo especializado, cuya implementación se deja a aplicaciones específicas.

Cabría preguntarse si el uso de las colecciones es deseable ya que agregan complejidad a las representaciones que tienen que manejar las aplicaciones. Por ejemplo, en aquellos sistemas de organización del conocimiento destinados principalmente a estructuras jerárquicas de navegación, parece más intuitivo representar "etiquetas de nodo" o "términos guía" como instancias de `skos:Concept`, y usar relaciones semánticas normales para vincularlas a otros conceptos. Veamos la siguiente variación de ejemplo con "leche":

```
ex3:lecheOrigenAnimal rdf:type skos:Concept;
  skos:prefLabel "leche de origen animal"@es;
  skos:broader ex3:leche;
  skos:narrower ex3:lecheVaca;
  skos:narrower ex3:lecheCabra;
  skos:narrower ex3:lecheBúfalo.
```

La elección entre ambas opciones de representación es algo abierto, dependiendo de la aplicación utilizada. Sin embargo, los lectores han de saber que no deben emplearse las colecciones, incluso si eso es más intuitivo, ya que puede suponer una pérdida considerable de precisión semántica. Muchas aplicaciones descriptivas (por ejemplo, "las etiquetas de nodo") resultan de naturaleza muy específica, y no deben usarse en los índices "normales" de manera conjunta con las etiquetas de conceptos. La representación como conceptos no es, por lo tanto, una práctica recomendable.

4.2 Características avanzadas de documentación

Como se muestra en la [sección 2.4](#), SKOS permite documentar conceptos asociándoles múltiples notas. Vale la pena destacar que la [Guía de Referencia de SKOS](#) no restringe el rango de recursos que pueden utilizarse como objeto de las declaraciones. Esto lleva a diferentes modelos de uso, tres de los cuales se explican—y se recomiendan—en este documento.

Documentación como un literal RDF

En este modelo los objetos de las declaraciones son simples literales RDF como muestran todos los ejemplos de la [sección 2.4](#). Este es el modo más sencillo para documentar conceptos y se espera que se ajuste a las aplicaciones más comunes..

Documentación como descripción de recursos relacionados

En este segundo modelo, el objeto de una declaración de documentación es un nodo general RDF no-literal, es decir, un nodo de un recurso (posiblemente vacío) que puede ser el sujeto de otras sentencias RDF [[RDF-PRIMER](#)]. Esto es especialmente útil para representar mediante RDF

información adicional sobre los propios elementos de documentación, como su autor o fecha de creación. Normalmente, esto se hace utilizando la propiedad RDF [rdf:value](#), como en el ejemplo siguiente, que utiliza un nodo vacío:

```
ex:tomate skos:changeNote [  
  rdf:value "Clasificado previamente bajo 'frutas' y actualmente bajo 'vegetales'"@es;  
  dct:creator ex:HoraceGray;  
  dct:date "1999-01-23"  
].  
ex:HoraceGray rdf:type foaf:Person; foaf:name "Horace Gray".
```

Documentación como referencia a un documento

Una tercera opción consiste en introducir, como objeto de una declaración de documentación, la URI de un *documento*, por ejemplo, una página Web. Tenga en cuenta que este modelo, estrechamente relacionado con el anterior, también permite la definición de metadatos adicionales para dicho documento utilizando RDF:

```
ex:zoología skos:definition ex:zoología.txt.  
ex:zoología.txt dct:creator ex:JohnSmith.
```

4.3 Relaciones entre etiquetas

Algunas aplicaciones requieren la creación de vínculos explícitos entre las etiquetas asociadas a los conceptos. Por ejemplo, considerar la relación entre una etiqueta preferente para el concepto "Corporación" y su abreviatura, "Corp." declarada como una etiqueta alternativa, o un vínculo para denotar la traducción entre dos etiquetas en diferentes idiomas: "Vaca"@es y "Cow"@es. El uso de las propiedades SKOS para el etiquetado léxico, como por ejemplo `skos:prefLabel`, se limita a emplear literales RDF. Por lo tanto, estas etiquetas no pueden ser objeto de una declaración RDF, y por tanto no es posible especificar una relación directa entre ellas.

Para resolver este problema de representación, se ha ampliado el vocabulario de SKOS con una extensión opcional para etiquetas, [SKOS-XL](#) [[SKOS-REFERENCE](#)]. Esta extensión presenta la clase `skosxl:Label` que permite tratar a las etiquetas como recursos RDF de primer orden. Cada instancia de esta clase debe adjuntarse en primer lugar a un literal RDF simple a través de la propiedad `skosxl:literalForm`. Consideremos el ejemplo del concepto "Organización de las Naciones Unidas" que está etiquetado por su nombre, tanto el oficial como el acrónimo de la organización. Ambas etiquetas pueden representarse del siguiente modo:

```
ex:ONUetiqueta1 rdf:type skosxl:Label;  
  skosxl:literalForm "Organización de las Naciones Unidas"@es.  
ex:ONUetiqueta2 rdf:type skosxl:Label;  
  skosxl:literalForm "ONU"@es.
```

Las instancias de `skosxl:Label` pueden estar relacionadas con conceptos utilizando ciertas propiedades ([skosxl:prefLabel](#), [skosxl:altLabel](#), [skosxl:hiddenLabel](#)) que reflejen el [tipo de literal de etiquetado utilizado](#). Por último, estas instancias pueden vincularse entre sí mediante declaraciones

[skosxl:labelRelation](#):

```
ex:ONU rdf:type skos:Concept;  
  skosxl:prefLabel ex:ONUetiqueta1;  
  skosxl:altLabel ex:ONUetiqueta2.  
ex:ONUetiqueta2 skosxl:labelRelation ex:ONUetiqueta1.
```

La solución anterior no es totalmente satisfactoria, ya que una aplicación que sea "sensible" a los acrónimos perdería la información real de que, entre ambas etiquetas, hay una relación de *acronimia*. Igualmente carecería de información acerca de la dirección hacia la que apunta el vínculo. Consiguientemente, se recomienda que en caso de utilizar SKOS-XL se especialice la propiedad `skosxl:labelRelation` a fin de satisfacer las necesidades de ciertas aplicaciones, tal y como se muestra a continuación:

```
ex:esAcrónimoDe rdfs:subPropertyOf skosxl:labelRelation.  
ex:ONUlabel2 ex:esAcrónimoDe ex:ONUetiqueta1.
```

Es necesario tener en cuenta que el modelo de datos de SKOS-XL garantiza que su uso siga siendo compatible con el estándar de etiquetado de SKOS. Si una instancia de `skosxl:Label` se adjunta a un concepto mediante una declaración con `skosxl:altLabel`, se desprende del modelo de datos de SKOS-XL que la forma literal de la instancia `skosxl:Label` se relaciona con este concepto a través de una declaración estándar con `skos:altLabel`. En el ejemplo anterior, `ex:ONU` por lo tanto tiene asociada la etiqueta alternativa (en su forma literal) `"ONU"@es`.

4.4 Coordinación de conceptos

Las prácticas de indización en las que participan tesauros y otros sistemas de organización a menudo incluyen la noción de coordinación. La coordinación es la actividad en la que se combinan los conceptos de este tipo de sistemas. En general hay dos tipos de coordinación: [pre-coordinación](#) y [post-coordination](#) [[WillpowerGlossary](#)]. La diferencia fundamental entre ambas depende de cuando se produce la coordinación en relación al momento en el que se realiza la recuperación de información.

La Pre-coordinación se lleva a cabo antes de la recuperación de información, por el gestor del sistema de organización o por un indizador que está utilizándolo, por ejemplo, si un indizador toma dos conceptos de un esquema de conceptos, como "Bicicletas" y "Reparación", y explícitamente los combina con una sintaxis determinada como "Bicicletas--Reparación" para asignar una materia a un documento concreto.

Por otro lado, la Post-coordinación se realiza durante la recuperación de información, por ejemplo, si se indiza un determinado documento con dos conceptos distintos "Bicicletas" y "Reparación" y un usuario decide realizar una búsqueda de todos los documentos que se han indexado con dichos conceptos.

La Post-coordinación como actividad de recuperación de información se presta a la representación *indirecta* como una consulta SPARQL para acceder a datos RDF [[SPARQL](#)]. Por ejemplo, dados dos conceptos distintos:


```
ex:bicicletas skos:prefLabel "Bicicletas"@es.  
ex:reparación skos:prefLabel "Reparación"@es.
```

se podría construir una consulta SPARQL que devuelva únicamente los documentos que se indexan con ambos conceptos

```
SELECT ?documento  
WHERE {  
  ?document dct:subject ex:bicicletas.  
  ?document dct:subject ex:reparación.  
}
```

Sin embargo, el vocabulario de SKOS no proporciona por sí mismo ningún mecanismo para expresar que un determinado concepto es en realidad una coordinación previa de otros. Por supuesto, es perfectamente factible [extender SKOS](#) para establecer un modelo de representación de conceptos coordinados. Por ejemplo, [se ha sugerido](#) que pudiera definirse una nueva propiedad `ex:coordinationOf`:

```
ex:coordinationOf a rdf:Property;  
  rdfs:domain skos:Concept;  
  rdfs:range rdf:List.
```

que podría utilizarse en afirmaciones tales como:

```
ex:reparaciónBicicletas a skos:Concept;  
ex:coordinationOf (ex:bicicletas ex:reparación);  
skos:prefLabel "Bicicletas--Reparación"@es.
```

También se ha sugerido que OWL podría utilizarse para coordinar estos conceptos:

```
ex:reparaciónBicicletas a skos:Concept;  
owl:intersectionOf (ex:bicicletas ex:reparación);  
skos:prefLabel "Bicicletas--Reparación"@es.
```

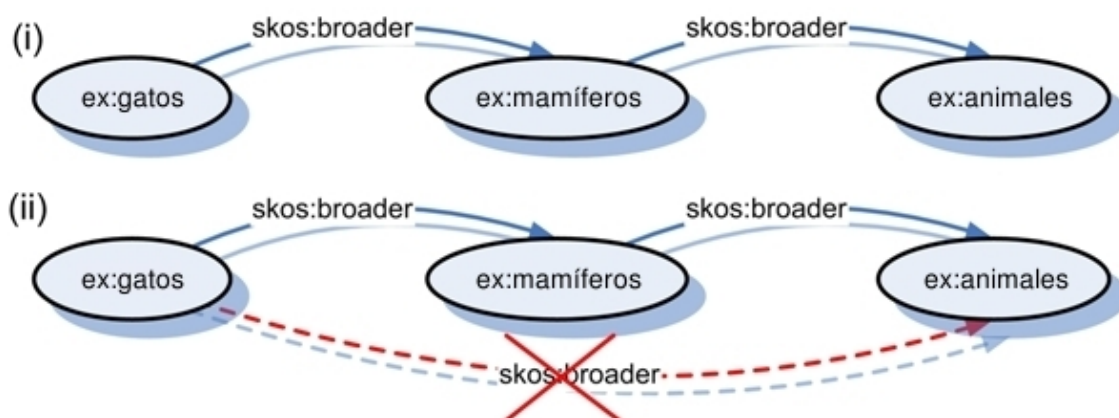
Sin embargo, los modelos establecidos para la pre-coordinación de este tipo todavía no han surgido en la comunidad SKOS. `ex:coordinationOf` (o alguna extensión equivalente), y las posibilidades de [uso de SKOS con OWL](#) no se han explorado suficientemente aún para justificar su inclusión en el vocabulario de SKOS. En lugar de comprometerse a un modelo de diseño que no se ha demostrado útil, el Grupo de trabajo para la Implementación de la Web Semántica decidió aplazar la cuestión de la coordinación, para permitir a los modelos de extensión que surgieran de forma natural conforme se vaya utilizando SKOS. Se espera que se establezcan modelos satisfactorios que puedan publicarse en la Web como una [extensión del vocabulario de SKOS](#) y documentarse como una Nota del W3C o equivalente.

4.5 Jerarquías transitivas

Como se describe en la [sección 2.3.1](#), las propiedades utilizadas para representar las jerarquías de un sistema de organización, `skos:broader` y `skos:narrower`, no se definen como transitivas. Como se muestra en la figura 4.5.1 (i) y (ii), esto significa que su semántica no es compatible con

inferencias del tipo: si "animales" es más genérico que "mamíferos" y "mamíferos" es más genérico que "gatos", luego "animales" es más amplio que "gatos".

Figura 4.5.1: `skos:broader` no es transitiva



Las flechas discontinuas representan declaraciones inferidas a partir del modelo de datos de SKOS.

Las flechas continuas representan declaraciones explícitas.

Para aplicaciones que precisan de tal semántica—por ejemplo, para realizar la expansión de una consulta—SKOS dispone de dos propiedades específicas, [skos:broaderTransitive](#) y [skos:narrowerTransitive](#). Están definidas como super-propiedades transitivas de `skos:broader` y `skos:narrower` [[SKOS-REFERENCE](#)]. Este modelo permite, usando herramientas de inferencia para la Web Semántica, acceder a la "clausura transitiva" de una jerarquía expresada con `skos:broader` y `skos:narrower`.

Considere el ejemplo de la figura 4.5.1 (i):

```
ex:animales skos:prefLabel "animales"@es.  
ex:mamíferos skos:prefLabel "mamíferos"@es;  
skos:broader ex:animales.  
ex:gatos skos:prefLabel "gatos"@es;  
skos:broader ex:mamíferos.
```

Al leer las declaraciones anteriores, un razonador podría hacer uso de la definición de `skos:broaderTransitive` como una super-propiedad de `skos:broader` para inferir las siguientes sentencias:

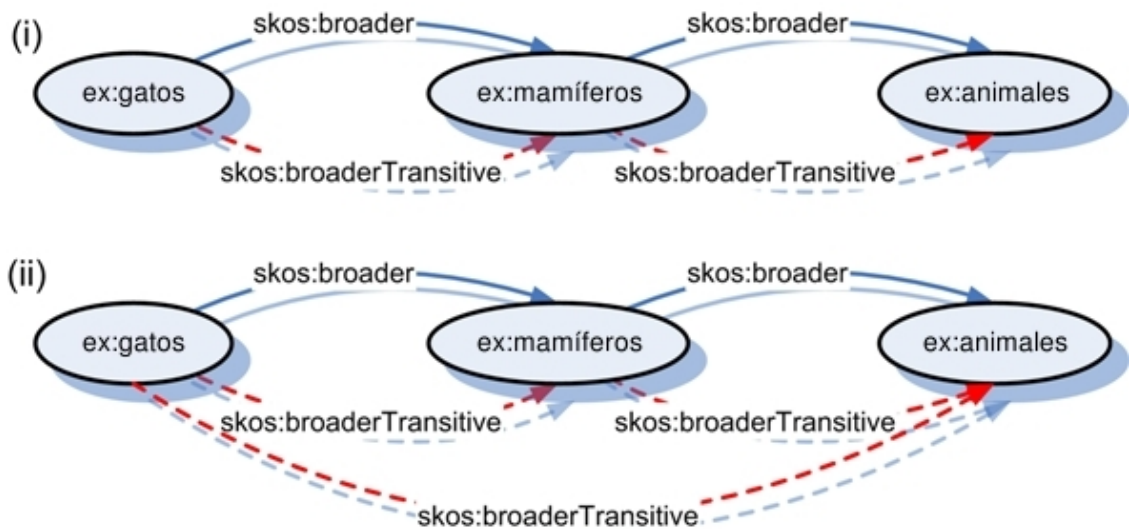
```
ex:gatos skos:broaderTransitive ex:mamíferos.  
ex:mamíferos skos:broaderTransitive ex:animales.
```

La transitividad de `skos:broaderTransitive` hace que pueda deducirse la declaración deseada:

```
ex:gatos skos:broaderTransitive ex:animales.
```

Estos dos pasos se muestran en la siguiente figura:

Figura 4.5.2: inferencia de una jerarquía transitiva a partir de una declaración explícita con `skos:broader`



Las flechas discontinuas representan declaraciones inferidas a partir del modelo de datos de SKOS.

Las flechas continuas representan declaraciones explícitas.

El uso de la super-propiedad `skos:broaderTransitive` permite a los grupos de desarrollo explotar las interpretaciones transitivas de las redes jerárquicas como mejor les parezca, mientras no interfieran con la semántica de `skos:broader`, que no cumple dicha transitividad. Intuitivamente, podría interpretarse que las declaraciones con `skos:broader` son lo suficientemente explícitas para asegurar la existencia de *vínculos directos con los conceptos superiores*, mientras que `skos:broaderTransitive` se usa para reflejar de forma más general (y posiblemente indirecta) las relaciones con aquellos ubicados en una *posición más ancestral*.

Nota sobre la supuesta "herencia de la transitividad": a primera vista, la super-propiedad existente entre `skos:broader` y `skos:broaderTransitive` puede parecer contraria a la intuición. Una propiedad no-transitiva se deriva a partir de otra que si lo es, por lo que no hereda su transitividad. Sin embargo, esto es totalmente compatible con la [semántica de RDFS/OWL](#) para `rdfs:subPropertyOf` [OWL]: una propiedad P es un sub-propiedad de Q si y sólo si cada P se sitúa entre dos recursos, entonces Q también se da entre ellos. Esto no garantiza la herencia de la transitividad: por el contrario, el conjunto de todas las parejas de recursos relacionados con P (su *grafo*), como un subconjunto de Q, puede que pierda algunas de las parejas que hacen transitiva a Q.

4.6 Notaciones

Algunos sistemas de organización del conocimiento, como por ejemplo la Clasificación Decimal Universal [CDU], usan notaciones (o leyendas) como medio principal de acceso a los conceptos que contienen. Las notaciones son símbolos que normalmente no son reconocibles como palabras o secuencias de palabras en cualquier lenguaje natural y por lo tanto utilizables independientemente de dicho contexto. Son generalmente

dígitos, complementados con signos de puntuación y otros caracteres, como en el siguiente ejemplo de la CDU:

```
512 Álgebra
512.6 Ramas especiales del Álgebra
```

SKOS permite representar las notaciones de dos formas, dependiendo de las prioridades del editor del esquema de conceptos. La primera técnica, que es la preferida, consiste en utilizar la propiedad `skos:notation`. Esta propiedad permite asociar un concepto a un [literal tipado RDF](#)—un literal con un tipo de datos explícito [[RDF-PRIMER](#)]. El tipo de datos del literal especifica la sintaxis de un esquema de codificación, que se inscribe en el uso de notaciones en el sistema de organización en particular. El valor del literal es la notación en sí mismo (en este caso el propio código de clasificación):

```
ex:cdu512 skos:prefLabel "Álgebra"@es ;
skos:notation "512"^^ex:NotaciónCDU.
```

La [sección 6.5.1](#) de la *Guía de Referencia SKOS* ofrece más detalles sobre cómo manejar tipos de datos [[SKOS-REFERENCE](#)]. Este enfoque puede ser especialmente útil si un editor de un sistema de organización quiere proporcionar a los usuarios reglas de procesamiento específicas para la notación del sistema. Por ejemplo, muchos sistemas de clasificación tienen reglas sintácticas específicas que permiten descomponer notaciones complejas, lo que lleva a la vinculación del concepto correspondiente con otros conceptos más simples. Además, este modelo puede ayudar a los creadores de herramientas de SKOS y a los editores de sistemas de organización que deseen visualizar las notaciones de un modo especial.

Sin embargo, la gestión de tipos de datos puede ser engorroso. Además, el modelo anterior no es realmente necesario cuando los editores consideran a las notaciones en sí mismas como un lenguaje sencillo de etiquetas independientes. En tales casos es posible utilizar, por ejemplo, la propiedad `skos:prefLabel` sin ningún indicador de idioma, como en:

```
ex:cdu512 skos:prefLabel "Álgebra"@es ;
skos:notation "512"^^ex:NotaciónCDU ;
skos:prefLabel "512" .
```

Hay que tener en cuenta que es poco probable que las notaciones representadas de tal manera se beneficien de los mecanismos específicos para notaciones (como procedimientos de visualización) en las herramientas que hagan uso de SKOS. De forma predeterminada, los usuarios deben esperar que estas notaciones sean tratadas, conforme al modelo de SKOS, como simples etiquetas.

4.7 Sobre la especialización del modelo de SKOS

SKOS está pensado como un denominador común entre los diferentes enfoques de modelado. Como tal, la especificación actual del vocabulario permitirá que muchos sistemas de organización del conocimiento existentes puedan ser transferidos a la Web Semántica. Sin embargo, la gran variedad de modelos de este tipo de sistemas hace imposible que se puedan captar

todos los detalles de los mismos al tiempo que "SKOS" conserve la primera "S" ("Simple").

Las aplicaciones que requieren un nivel de detalle mayor se beneficiarán enormemente de SKOS al tratarse de un vocabulario para la Web Semántica. SKOS de hecho puede ser ampliado sin problemas para adaptarse a las necesidades específicas de un grupo de usuarios de un sistema de organización del conocimiento en particular, manteniendo la compatibilidad con las aplicaciones que se fundamentan en las características básicas de SKOS.

Esto se puede conseguir principalmente mediante la especialización de los constructores de SKOS en otros más específicos. Los usuarios pueden crear sus propias características y clases, adjuntándolas a los elementos estándar del vocabulario de SKOS, usando las propiedades `rdfs:subPropertyOf` y `rdfs:subClassOf` del vocabulario de Esquema RDF [[RDF-PRIMER](#)].

El ejemplo de la [sección 4.3](#) muestra el modo en el que `skosxl:labelRelation` puede especializarse en una propiedad semánticamente más rica dedicada a la representación de la vinculación de acrónimos. Son posibles otros usos, como la creación de diferentes "variantes" de las propiedades `skos:broader` y `skos:narrower`. Las normas sobre Tesoros identifican un reducido conjunto de tipos de relaciones jerárquicas, como genérico, parte-todo, o instancia-clase [[ISO2788](#)]. El enfoque de SKOS permite al diseñador de aplicaciones crear nuevas propiedades plasmando esta distinción mediante la declaración de sub-propiedades de `skos:broader`:

```
ex:broaderGeneric rdfs:subPropertyOf skos:broader.  
ex:broaderPartitive rdfs:subPropertyOf skos:broader.  
ex:broaderInstantive rdfs:subPropertyOf skos:broader.
```

Cada declaración con `ex:broaderPartitive` entre dos conceptos, por ejemplo, puede ser formalmente interpretada por un motor de inferencia adecuado para la Web Semántica. Esta interpretación proporcionará la posibilidad de deducir setencias `skos:broader` entre dichos conceptos—un elemento de información que puede aprovecharse posteriormente por herramientas que operan con el modelo básico de SKOS.

Nota sobre la manipulación del vocabulario propio de SKOS:

En general, es mejor evitar declaraciones en las que el *sujeto* sea un URI del vocabulario de SKOS. Haciendo esto puede alterarse el modelo de datos de SKOS y presentar efectos secundarios no deseados. Ello puede poner en peligro la interoperabilidad de los vocabularios. Si se quiere adaptar el comportamiento de vocabularios "incorporados" a casos específicos, primero debe considerarse la introducción de constructores propios en forma de sub-clases o sub-propiedades.

Por supuesto, se anima a los creadores de extensiones para SKOS que las publiquen, por ejemplo, utilizando la [lista de correo pública de SKOS \(public-esw-thes@w3.org\)](mailto:public-esw-thes@w3.org). Estas extensiones pueden corresponder a intereses comunes y por lo tanto ser reutilizados en diferentes aplicaciones. Además, esta reutilización puede proporcionar una retroalimentación para la

comunidad, contribuyendo a mejorar la calidad de las extensiones publicadas.

5 Combinaciones de SKOS con otras aproximaciones de modelado

Como se ha visto anteriormente, SKOS es un vocabulario RDF/OWL que puede ser ampliado fácilmente para adaptarse a necesidades concretas. Asimismo, las características de SKOS también pueden utilizarse en la Web Semántica como complemento de otros vocabularios de modelado. Esta sección ofrece ejemplos de reutilización de las propiedades de etiquetado de SKOS para describir recursos que no son necesariamente conceptos de SKOS. A continuación, se aborda la problemática específica de la articulación de conceptos de SKOS con *clases* definidas por el lenguaje de ontologías OWL.

Nota: esta sección trata cuestiones que surgen cuando una aplicación precisa que las características de SKOS se utilicen coordinadamente con otros enfoques de modelado. Los usuarios que no precisen de ello pueden omitir esta sección.

5.1 Uso of etiquetas fuera de SKOS

Es posible utilizar las propiedades de etiquetado de SKOS para etiquetar recursos que no sean del tipo `skos:Concept`. Considere estas sentencias que describen a Tim Berners-Lee:

```
<http://www.w3.org/People/Berners-Lee/card#i> rdf:type foaf:Person;  
foaf:name "Timothy Berners-Lee";  
rdfs:label "TBL";  
skos:prefLabel "Tim Berners-Lee"@en.
```

Una aplicación que desee mostrar una etiqueta para este recurso es capaz de identificar a "Tim Berners-Lee", como la etiqueta preferente en lugar de tener que elegir entre ambas etiquetas compatibles, [rdfs:label](#) "TBL" o [foaf:name](#) "Timothy Berners-Lee"—estas etiquetas son compatibles porque `foaf:name` es una sub-propiedad de `rdfs:label`.

Otro ejemplo es el de las etiquetas legibles por personas asociadas a clases, propiedades e individuos en ontologías OWL, que normalmente se expresan utilizando únicamente `rdfs:label`. Considere las siguientes declaraciones describen a los seres humanos:

```
ex:Human rdf:type owl:Class;  
rdfs:label "humano"@es;  
rdfs:label "hombre"@es.
```

Una aplicación podría tener dificultades para determinar la etiqueta correcta que ha de mostrar a los usuarios ya que ambas tienen el mismo peso. La semántica de `skos:prefLabel` permite a los desarrolladores definir explícitamente la denominación preferente para un recurso determinado. En general, la capacidad de reutilizar los elementos del vocabulario de SKOS

y otros vocabularios RDF según se precise es lo que da RDF gran parte de su poder expresivo.

5.2 Conceptos SKOS y clases OWL

La [Guía de referencia de SKOS](#) define a `skos:Concept` como una clase OWL [[SKOS-REFERENCE](#)]:

```
skos:Concept rdf:type owl:Class.
```

Así pues, las instancias de `skos:Concept` (por ejemplo `ex:Pintura` en un vocabulario sobre Arte) son, en términos de OWL, individuos.

```
ex:Pintura rdf:type skos:Concept.
```

Esto plantea la cuestión acerca de si una instancia de un concepto SKOS, como `ex:Painting` puede tratarse como una clase por sí misma. Los usuarios, podrían definir para `ex:painting` propiedades como `ex:support`:

```
ex:soporte rdf:type owl:DatatypeProperty.  
ex:soporte rdfs:domain ex:Pintura.
```

Cabría preguntarse acerca del motivo por el que alguien querría hacer esto. Pues bien, conceptualmente, una clase como `skos:Concept` puede ser vista como una metaclass: sus instancias serían los conceptos que aparecen en un vocabulario. Por lo tanto, es concebible que los usuarios de SKOS quieran especificar las características a nivel de clase de los conceptos, por ejemplo, que las pinturas tienen un soporte o que el queso tiene un país de origen.

Cabe destacar que SKOS no toma una postura con respecto a qué variante de OWL—OWL Full u OWL-DL [[OWL-REFERENCE](#)—debería usarse junto con SKOS. Los usuarios de OWL Full serán capaces de manejar el escenario anterior mediante el tratamiento de los conceptos SKOS explícitamente como clases, por ejemplo, mediante la adición de las declaraciones del tipo:

```
ex:Pintura rdf:type owl:Class.
```

Esto es posible porque OWL Full no requiere que el conjunto de clases e individuos sea disyunto. Las personas que deseen utilizar la variante DL de OWL no pueden utilizar este mecanismo de metamodelado, puesto que es imprescindible que se cumpla la condición de disyunción entre clases e individuos en toda ontología OWL-DL. Los usuarios de OWL-DL que estén interesados en la vinculación de clases OWL a conceptos SKOS han de mantener esta distinción formal. Sin embargo, pueden utilizar las *propiedades de anotación OWL* para salvar esta limitación, siempre que creen y utilicen sus propias extensiones para SKOS, como en:

```
ex:ClasePintura rdf:type owl:Class.  
ex:ConceptoPintura rdf:type skos:Concept.  
ex:ClassPintura ex:correspondenciaConcepto ex:ConceptoPintura.
```

Tenga en cuenta que en el momento de redactar este documento, el *Grupo de Trabajo sobre OWL* [[OWL-WG](#)] había sido creado recientemente para manejar (algunas formas de) metamodelado de un marco de descripción

lógica. Esto podría permitir a los usuarios de OWL-DL optar por modelos que son más fáciles de explotar.

En resumen, la relación entre los conceptos de SKOS y las clases/individuos de OWL es la siguiente:

- Los conceptos de SKOS son individuos de OWL;
- SKOS no toma una postura sobre si también debería ser posible tratar los conceptos de SKOS como clases de OWL;
- Las restricciones de OWL-DL evitan que los conceptos de SKOS se traten como clases de OWL;
- Se espera que la revisión en curso de OWL solviera este problema, ofreciendo alguna forma de metamodelado.

5.3 SKOS, conjuntos de datos RDF y contención de información

En un contexto de sistemas de organización del conocimiento en red, algunas aplicaciones pueden exigir el seguimiento de la procedencia o propietario de las declaraciones de SKOS para, por ejemplo, determinar su veracidad. Un aspecto específico es la forma de establecer vínculos explícitos entre un esquema de conceptos y cada elemento de información del sistema de organización original que representa, como por ejemplo las relaciones semánticas entre conceptos.

Dicha funcionalidad, aunque identificada como un [requisito candidato \[SKOS-UCR\]](#), está actualmente fuera del ámbito de aplicación de SKOS. En RDF, las declaraciones se dan en forma de declaraciones-libres de triplas, lo que hace difícil representar su contención y procedencia.

Sin embargo, se han propuesto soluciones para estos problemas, tales como los grafos asociados a un URI [[NAMED-GRAPHS](#)], y el uso de [Conjuntos de datos RDF](#) en SPARQL [[SPARQL](#)]. Un esquema de conceptos SKOS puede estar relacionado con un conjunto de datos RDF, o incluso definido por dicho conjunto, lo que permite la creación de consultas SPARQL encargadas de proporcionar alguna forma de procedencia o de contención. Continuando con el ejemplo de la [sección 3.2](#), y suponiendo que `ex1:esquemaReferenciaAnimales` y `ex2:esquemaGato` se hayan gestionado como conjuntos de datos RDF apropiados (en este caso, grafos asociados a un URI), la consulta

```
SELECT ?x ?y
WHERE {
  GRAPH ex2:esquemaGato { ?x skos:broader ?y }
}
```

puede devolver (`ex2:abisinio`, `ex1:gato`) como resultado, mientras que esta tupla no aparecería entre los resultados de

```
SELECT ?x ?
WHERE {
  GRAPH ex1:esquemaReferenciaAnimales { ?x skos:broader ?y }
}
```

Sin embargo, el lector debe ser consciente de que estos mecanismos no

han sido ampliamente utilizados en el momento de la elaboración de este documento, y que en el futuro podrían darse prácticas estándar distintas.

Referencias

[BS8723-2]

BS 8723-2:2005 Structured vocabularies for information retrieval. Guide. Thesauri, British Standards Institution, London, 2005.

[BS8723-4]

BS 8723-4:2007 Structured vocabularies for information retrieval. Guide. Interoperability between vocabularies, British Standards Institution, London, 2007.

[COOLURIS]

[Cool URIs for the Semantic Web](#), Leo Sauermann, Richard Cyganiak, Editores, W3C Interest Group Note, 3 December 2008. [Última versión](#) disponible en <http://www.w3.org/TR/cooluris/> .

[DC]

[DCMI Metadata Terms](#), 14 January 2008. [Última versión](#) disponible en <http://dublincore.org/documents/dcmi-terms/> .

[ISO2788]

[ISO 2788:1986](#) Documentation - Guidelines for the establishment and development of monolingual thesauri. Segunda edición. ISO TC 46/SC 9, 1986.

[ISO5964]

[ISO 5964:1985](#) Documentation - Guidelines for the establishment and development of multilingual thesauri. Primera edición. ISO TC 46/SC 9, 1985.

[NAMED-GRAPHS]

Named graphs, provenance and trust, Jeremy Carroll, Christian Bizer, Patrick Hayes, Patrick Stickler, WWW 2005.

[RDF/XML-SYNTAX]

[RDF/XML Syntax Specification \(Revised\)](#), Dave Beckett, Editor. W3C Recommendation, 10 February 2004. [Última versión](#) disponible en <http://www.w3.org/TR/rdf-syntax-grammar/> .

[RECIPES]

[Best Practice Recipes for Publishing RDF Vocabularies](#). Diego Berrueta, Jon Phipps. W3C Working Draft, 23 January 2008. [Última versión](#) disponible en <http://www.w3.org/TR/swbp-vocab-pub/> .

[OWL-WG]

[OWL Working Group](#), <http://www.w3.org/2007/OWL/>.

[OWL]

[OWL Web Ontology Language Reference](#), Mike Dean, Guus Schreiber, Editors, W3C Recommendation, 10 February 2004. [Última versión](#) disponible en <http://www.w3.org/TR/owl-ref/> .

[OWL-SEMANTICS]

[OWL Web Ontology Language Semantics and Abstract Syntax](#), Peter F. Patel-Schneider, Patrick Hayes, Ian Horrocks, Editors, W3C Recommendation, 10 February 2004. [Última versión](#) disponible en <http://www.w3.org/TR/owl-semantics/> .

[RDF-PRIMER]

[RDF Primer](#), Frank Manola, Eric Miller, Editors, W3C Recommendation, 10 February 2004. [Última versión](#) disponible en <http://www.w3.org/TR/rdf-primer/> .

[RDF-CONCEPTS]

[Resource Description Framework \(RDF\): Concepts and Abstract Syntax](#), Graham Klyne, Jeremy Carroll, Editors, W3C Recommendation, 10 February 2004. [Última versión](#) disponible en <http://www.w3.org/TR/rdf-concepts/> .

[RFC4646]

[Tags for Identifying Languages](#), A. Phillips, M. Davis, Editors, September 2006. Disponible en <http://www.ietf.org/rfc/rfc4646.txt> .

[SWBP-SKOS-CORE-GUIDE]

[SKOS Core Guide](#), Alistair Miles, Dan Brickley, Editors, W3C Working Draft, 2 November 2005. [Última versión](#) disponible en <http://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20051102/> .

[SKOS-REFERENCE]

[SKOS Reference](#), Alistair Miles, Sean Bechhofer, Editors, W3C Recommendation, 18 August 2009. [Última versión](#) disponible en <http://www.w3.org/TR/skos-reference> .

[SKOS-UCR]

[SKOS Use Cases and Requirements](#), Antoine Isaac, Jon Phipps, Daniel Rubin, Editors, W3C Working Group Note, 18 August 2009. [Última versión](#) disponible en <http://www.w3.org/TR/skos-ucr> .

[SPARQL]

[SPARQL Query Language for RDF](#), Eric Prud'hommeaux, Andy Seaborne, Editors, W3C Working Draft, 15 January 2008. [Última versión](#) disponible en <http://www.w3.org/TR/rdf-sparql-query/> .

[SWBP-SKOS-CORE-SPEC]

[SKOS Core Vocabulary Specification](#), Alistair Miles, Dan Brickley, Editors, W3C Working Draft, 2 November 2005. Disponible en <http://www.w3.org/TR/2005/WD-swbp-skos-core-spec-20051102/> .

[SWD]

[The Semantic Web Deployment Working Group](#), <http://www.w3.org/2006/07/SWD/> .

[TURTLE]

[Turtle - Terse RDF Triple Language](#), David Beckett, Tim Berners-Lee. W3C Team Submission, 14 January 2008. [Última versión](#) disponible en <http://www.w3.org/TeamSubmission/turtle/> .

[UDC]

[UDC - Universal Decimal Classification](#), UDC Consortium, <http://www.udcc.org/> .

[URI]

[RFC 3986 - Uniform Resource Identifiers \(URI\): Generic Syntax](#), Tim Berners-Lee, Roy Fielding, Larry Masinter, IETF, January 2005. Disponible en <http://tools.ietf.org/html/rfc3986> .

[WillpowerGlossary]

[Glossary of terms relating to thesauri and other forms of structured vocabulary for information retrieval](#), Stella Dextre Clarke, Alan Gilchrist, Ron Davies and Leonard Will, Willpower Information. Disponible en <http://www.willpowerinfo.co.uk/glossary.htm> .

Agradecimientos

Los autores quieren agradecer a Alistair Miles and Dan Brickley quienes editaron la Guía sobre SKOS Core (en la cual se ha basado principalmente este manual); así como a Tom Baker, Guus Schreiber y Sean Bechhofer quienes han participado en partes significativas de este texto. Los miembros del Grupo para el Desarrollo de la Web Semántica, Tom Baker, Margherita Sini, Quentin Reul también han realizado extensas revisiones durante el proceso de revisión.

Este documento es el resultado de amplios debates en el seno del Grupo para el Desarrollo de la Web Semántica en su totalidad. Los miembros del Grupo de Trabajo no mencionados anteriormente son, en orden alfabético: Ben Adida, Diego Berrueta, Jeremy Carroll, Michael Hausenblas, Elisa Kendall, Vit Novacek, Jon Phipps, Clay Redding, Daniel Rubin, Manu Sporny, y Ralph Swick.

Los comentarios públicos (especialmente a través de la lista pública de correo public-esw-thes@w3.org) de las siguientes personas han aportado consejos, sugerencias y correcciones de gran valor: Mark van Assem, Stephen Bounds, Dan Brickley, Johan De Smedt, Stella Dextre-Clarke, Alasdair Gray, Andrew Houghton, Simon Jupp, Carl Mattocks, Emma McCulloch, Mikael Nilsson, Alan Ruttenberg, Aida Slavic, Simon Spero, Doug Tudhope, Bernard Vatant, Jakob Voss, Leonard Will, Sue Ellen Wright.


Apéndice. Correspondencias entre la norma ISO-2788/5964 y los constructores de SKOS

SKOS debe mucho a décadas de esfuerzos en la comunidad de Sistemas de Organización del Conocimiento, en forma de aplicaciones, directrices y formatos estándar. La compatibilidad entre el modelo de SKOS y dos de esos esfuerzos, las especificaciones ISO 2788 para tesauros monolingües [[ISO-2788](#)] y las especificaciones ISO 5964 para tesauros multilingües [[ISO-5964](#)], fueron específicamente planteadas como un requisito candidato en los *Casos de uso y Requisitos de SKOS* [[SKOS-UCR](#)].

SKOS no especifica en sí mismo reglas sobre cómo crear esquemas de conceptos, sin embargo, su modelo de datos refleja algunos principios de la construcción de Sistemas de Organización del Conocimiento. En el diseño de su vocabulario ha influido especialmente la orientación de los tesauros estándares, ya que se tratan de algunas de las propuestas más maduras en el campo de este tipo de sistemas. En particular, hay muchos puntos comunes entre SKOS y la norma ISO 2788/5964. La siguiente tabla resume los paralelismos y destaca el modo en que el diseño de SKOS varía con respecto a las recomendaciones de la norma ISO. Es de esperar que esto ayudará a los esfuerzos futuros de migración a SKOS tesauros que sigan las directrices ISO.

El lector debe ser consciente de que esta comparación no debe de ninguna

manera interpretarse como una limitación del alcance de SKOS a los tesauros estándares. Como ya se ha dicho en este documento, se puede utilizar SKOS—posiblemente con las extensiones adecuadas para otros tipos de sistemas de organización, o tesauros que no sigan las directrices ISO.

Aspectos de diseño	ISO 2788/5964	SKOS
conceptos frente a términos	<p>En las normas ISO, los tesauros son lenguajes de indización que contienen términos.</p> <p>ISO 2788 aborda extensamente la elaboración de los términos, centrándose en su forma. Por ejemplo, los calificadores explícitos se utilizan para distinguir homógrafos: Mercurio (planeta) frente a Mercurio (elemento).</p>	<p>Los conceptos constituyen el núcleo primordial de SKOS. Los términos en las normas ISO corresponden a las etiquetas de los conceptos de SKOS.</p> <p>SKOS, como un sencillo medio de publicación, no proporciona normas para el diseño de la etiqueta. Además, SKOS utiliza literales simples para representar las etiquetas, por lo que no es posible expresar mecanismos para la formación de términos como la cualificación formal y explícita. Para ello, y para otros casos que requieran adjuntar información a las etiquetas y no al concepto que expresan, debe utilizarse la extensión SKOS-XL (ver sección 4.3).</p>
<p>Relaciones</p>  <p>sistema de organización—equivalencia</p>	<p>Los términos pueden ser idénticamente equivalentes. Se distinguen entre preferentes y no-preferentes, utilizando las relaciones USE (usar o véase) y UF (used for, usado por).</p> <p>Se supone que un término no-preferente sólo puede apuntar a un término equivalente preferente, siendo este último el principal punto de entrada para el concepto que ambos expresan.</p>	<p>Los términos equivalentes son representados como etiquetas de un solo concepto. De forma predeterminada, no hay relación directa entre estas etiquetas. Al igual que en la norma ISO 2788, las etiquetas preferentes se distinguen de las no-preferentes (alternativas). Además, SKOS incorpora las etiquetas ocultas.</p> <p>Un concepto puede tener sólo una etiqueta preferente (por idioma). Sin embargo, dentro de un mismo esquema los conceptos pueden tener idénticas etiquetas preferentes, aunque esto no es recomendable.</p>