
Vocabularios y taxonomías

PID_00271454

Blas Torregrosa García

Tiempo mínimo de dedicación recomendado: 1 hora



**Blas Torregrosa García**

Ingeniero en Informática y máster universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones (MISTIC) por la Universitat Oberta de Catalunya (UOC). Especializado en ciberseguridad. Profesor colaborador en el máster de Ciencia de Datos de la UOC y profesor asociado en la Universidad de Valladolid (UVA).

El encargo y la creación de este recurso de aprendizaje UOC han sido coordinados por el profesor: Ferran Prados Carrasco (2020)

Primera edición: febrero 2020
© Blas Torregrosa García
Todos los derechos reservados
© de esta edición, FUOC, 2020
Av. Tibidabo, 39-43, 08035 Barcelona
Realización editorial: FUOC

Ninguna parte de esta publicación, incluido el diseño general y la cubierta, puede ser copiada, reproducida, almacenada o transmitida de ninguna forma, ni por ningún medio, sea este eléctrico, químico, mecánico, óptico, grabación, fotocopia, o cualquier otro, sin la previa autorización escrita de los titulares de los derechos.

Índice

Introducción	5
1. Vocabularios	7
1.1. RDF Schema	8
1.1.1. Definiendo clases y subclases	9
1.1.2. Definiendo propiedades, dominios y rangos	9
1.1.3. Reglas de inferencia en RDFS	10
1.2. Ejemplos de vocabularios	11
1.2.1. El vocabulario schema.org	11
1.2.2. Vocabularios de personas	12
1.2.3. Vocabularios de libros y bibliografía	12
1.2.4. Vocabularios para comunidades	13
2. Taxonomías	14
2.1. SKOS	14
Bibliografía	17

Introducción

Una componente fundamental para alcanzar los objetivos de la web semántica es el uso de vocabularios que posean una semántica bien definida. Un enfoque que facilita enormemente la consecución de este objetivo es el de utilizar vocabularios de referencia o taxonomías.

1. Vocabularios

Los **vocabularios** definen el conjunto de términos utilizados por una aplicación o por un conjunto de aplicaciones. Además califican un conjunto de posibles relaciones y definen restricciones sobre los términos que se pueden utilizar en un cierto contexto. Los vocabularios pueden ser muy complejos (con varios miles de términos) o muy simples (describiendo solo uno o dos conceptos).

El uso principal de vocabularios en la web semántica es ayudar en la integración de datos en los que puede haber ambigüedades entre diferentes conjuntos de datos. Supongamos que hay una biblioteca que quiere integrar datos que llegan de diferentes editores. Algunos editores pueden usar el término «Autor» para describir al escritor del libro, mientras que otros editores pueden usar el término «Creador». Para integrar los datos, un vocabulario tiene que describir el hecho de que autor y creador signifiquen lo mismo.

Los vocabularios controlados recopilan conceptos y términos utilizados para describir un campo o área de interés.

Por ejemplo, para definir a una persona en un formato entendible por máquinas, se necesita un vocabulario que tenga la definición formal de «Persona». Una opción es utilizar el vocabulario *Friend of a Friend* (FOAF, literalmente 'Amigo de un Amigo'), que tiene una clase «Person» que define las propiedades típicas de una persona, entre otras su nombre. Supongamos que tenemos el siguiente fragmento XML:

Figura 1. Definiendo clase y propiedad de un recurso

```
<Person>  
  <name>Neil Armstrong</name>  
</Person>
```

Este fragmento XML proporciona una jerarquía, suponiendo que hay una clase «Person» y que tiene una propiedad «Name». Sin embargo no tiene contexto. Hay que indicar un vocabulario externo que defina esta clase y esta propiedad, utilizando el mecanismo del espacio de nombres (*Namespace*).

En RDF/XML esto se puede hacer usando el atributo `xmlns`; en este caso, `xmlns:foaf = "http://xmlns.com/foaf/0.1/"`, que referencia al espacio de nombres FOAF. Este mecanismo permite abreviar esto mediante el prefijo `foaf`; así `foaf:Person` referencia a `http://xmlns.com/foaf/0.1/Person`.

Figura 2. Describiendo el nombre de una persona usando un vocabulario

```

...
xmlns:foaf="http://xmlns.com/foaf/0.1/"

<foaf:Person>
  <foaf:name>Neil Armstrong</foaf:name>
</foaf:Person>

```

Las definiciones de los vocabularios no dependen del formato; así, FOAF puede usarse en RDF, RDFa, HTML5 Microdata o JSON-LD. El vocabulario sí que depende del área de interés a la que representa. Sin embargo ciertos dominios usuales como personas o libros pueden describirse con clases y propiedades de más de un vocabulario.

1.1. RDF Schema

El **RDFS** (*RDF Vocabulary Description Language*, originalmente *RDF Schema Language*) es un lenguaje basado en RDF para crear ontología que definen dominios de conocimiento y las relaciones entre ellos.

RDFS es una extensión del vocabulario RDF con elementos básicos de ontología y reutiliza propiedades de RDF. Las ontologías RDFS se representan como grafos RDF. Así, RDFS es adecuado para describir varios tipos de recursos mediante propiedades específicas. Las **clases** y **propiedades** son el vocabulario RDFS, que incluye un conjunto de recursos predefinidos junto con su significado.

Conjunto de recursos

<http://www.w3.org/2000/01/rdfschema#> y el prefijo asociado rdfs.

Las **clases** del vocabulario RDFS se usan para definir:

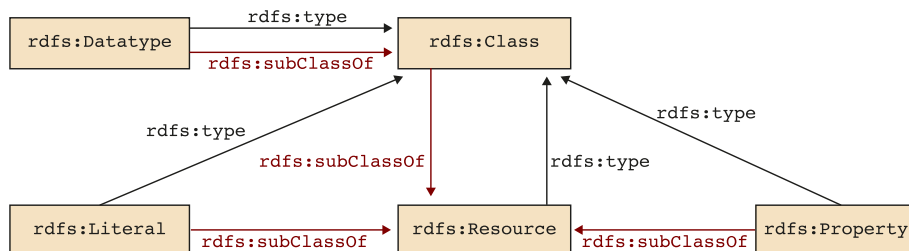
- Recursos (`rdfs:Resource`)
- Literales como enteros o cadenas (`rdfs:Literal`)
- Clases (`rdfs:Class`)
- Tipos de datos RDF (`rdfs:Datatype`)
- Contenedores (`rdfs:Container`)
- Propiedades de miembros de un contenedor (`rdfs:ContainerMembershipProperty`)

Las **propiedades** RDFS pueden expresar:

- El sujeto es una subclase de una clase (`rdfs:subClassOf`).
- El sujeto es una subpropiedad de una propiedad (`rdfs:subPropertyOf`).
- Definir un dominio (`rdfs:domain`).
- Rango de una propiedad (`rdfs:range`).
- Añadir un nombre legible por personas (`rdfs:label`).
- Descripción de un recurso (`rdfs:comment`).
- Identificar un miembro de un recurso (`rdfs:member`).
- Añadir información relativa al recurso (`rdfs:seeAlso`).

- Proporcionar la definición del recurso (`rdfs:isDefinedBy`).

Figura 3. Relaciones entre las clases definidas en el lenguaje RDFS



1.1.1. Definiendo clases y subclases

Una clase RDFS corresponde a un tipo o categoría utilizada para clasificación o jerarquía. En RDFS una clase *C* se define mediante una tripleta:

Figura 4. Definición de clase

```
C rdfs:type rdfs:Class .
```

donde `rdfs:Class` es una clase predefinida y `rdfs:type` una propiedad también predefinida.

De la misma manera, las subclases también se definen mediante una tripleta:

Figura 5. Definición de subclase

```
SC rdfs:subClassOf C .
```

La propiedad `rdfs:subClassOf` es reflexiva, es decir, una vez que una clase RDFS se ha creado, es subclase de sí misma. La propiedad `rdfs:subClassOf` se usa para declarar que una clase es una especialización de otra clase más general.

Para definir una instancia de una clase *C* se utiliza también una tripleta:

Figura 6. Definición de instancia

```
I rdfs:type C .
```

Aquí, la propiedad `rdfs:type` se usa para declarar que el individuo *I* es una instancia de la clase *C*.

1.1.2. Definiendo propiedades, dominios y rangos

Una propiedad en RDF puede utilizarse de una de las siguientes formas:

- 1) Para indicar el valor de un atributo para un recurso.
- 2) Para establecer una relación entre dos recursos.

Para indicar que una propiedad es de tipo `rdf:Property` utilizamos `rdf:type`.

Figura 7. Definición de una propiedad

```
:p rdf:type rdf:Property .
```

En muchas aplicaciones es útil restringir los recursos a los que una propiedad puede aplicarse. RDFS tiene dos componentes para gestionar el dominio (`rdfs:domain`) y el rango (`rdfs:range`) de una propiedad.

Una restricción de **dominio** permite clasificar los sujetos de una relación.

Figura 8. Definición de propiedad con dominio

```
:p rdf:type rdf:Property .
:p rdfs:domain :C .
```

Una restricción de **rango** permite clasificar los objetos de una relación.

Figura 9. Definición de propiedad con rango

```
:p rdf:type rdf:Property .
:p rdfs:range :D .
```

De esta forma se pueden añadir restricciones sobre el sujeto y sobre el objeto de una propiedad.

1.1.3. Reglas de inferencia en RDFS

El mecanismo de razonamiento en RDFS permite inferir nuevas tripletas a partir de las existentes en un grafo RDF. Este mecanismo utiliza el significado del vocabulario RDFS: `rdfs:Class`, `rdf:type`, `rdf:Property`, `rdfs:domain`, `rdfs:range` o `rdfs:subClassOf`.

Este mecanismo se traduce en una serie de reglas de inferencia. A partir de las tripletas siguientes:

Figura 10. Antecedente

```
:a rdf:type :C .
:C rdfs:subClassOf :D.
```

se infiere la siguiente tripleta (que se denomina una jerarquía de clases):

Figura 11. Consecuente

```
:a rdf:type :D .
```

En RDFS es posible expresar diferentes reglas de inferencia que se resumen en la tabla 1.

Tabla 1. Reglas de inferencia RDFS

Regla de inferencia	Premisa	Inferencia
Jerarquía de clase	:a rdf:type :C . :C rdfs:subClassOf :D .	:a rdf:type :D .
Jerarquía de propiedad	:a :p :b . :p rdfs:subPropertyOf :q .	:a :q :b .
Jerarquías de clases	:A rdfs:subClassOf :B . :B rdfs:subClassOf :C .	:A rdfs:subClassOf :C .
Jerarquías de propiedades	:p rdfs:subPropertyOf :q . :q rdfs:subPropertyOf :r .	:p rdfs:subPropertyOf :r .
Dominio de una propiedad	:a :p :b . :p rdfs:domain :C .	:a rdf:type :C .
Rango de una propiedad	:a :p :b . :p rdfs:range :D .	:b rdf:type :D .

1.2. Ejemplos de vocabularios

En este contexto es necesario establecer un conjunto de vocabularios de referencia, como modo de facilitar la comunicación de los datos.

1.2.1. El vocabulario schema.org

Con aproximadamente 300 definiciones de conceptos, **schema.org** es una de las colecciones más utilizadas como esquema de marcado para datos estructurados. Schema.org fue lanzado por Google, Yahoo y Bing en 2011. Schema.org contiene las definiciones de los conceptos más utilizados, con lo que puede hacer anotaciones de acciones, trabajos creativos, eventos, servicios, conceptos médicos, organizaciones, personas, lugares o productos.

De forma análoga al ejemplo anterior, si queremos describir un libro necesitaríamos un vocabulario para definir libros y sus propiedades típicas. Si quisiéramos describir el título del libro, se podría utilizar la propiedad «Name» de schema.org o utilizar la propiedad «Title» del vocabulario Dublin Core (DC), usando dos declaraciones de espacios de nombres.

Figura 12. Descripción de un libro

```

...
xmlns:schema="http://schema.org/"
xmlns:dc="http://purl.org/dc/terms/"
...
<schema:Book>
  <dc:title>El ingenioso hidalgo Don Quixote de la Mancha</dc:title>
</schema:Book>

```

En este caso, `schema:Book` es una abreviatura de `http://schema.org/Book`, que es una definición de la clase «Book», y `dc:title` referencia `http://purl.org/dc/terms/title`, que es una definición de la propiedad «título».

Clases y propiedades

Los tipos (clases) más frecuentes de schema.org y sus propiedades se pueden encontrar en http://schema.org/docs/gs.html#schemaorg_types, y la lista completa de propiedades está en <http://schema.org/docs/full.html>.

1.2.2. Vocabularios de personas

Las características de una persona y las relaciones entre personas pueden describirse mediante una variedad de vocabularios controlados, como se resume en la tabla 2.

Tabla 2. Vocabularios de persona

Vocabulario	Abreviatura	Espacio de nombres	Uso
Person de schema.org	schema:Person	https://schema.org/Person	Nombre, apellidos, género, nacionalidad, profesión, prefijo (o sufijo) honorífico, etc.
Friend Of A Friend	foaf	http://xmlns.com/foaf/0.1/	Persona, nombre, género, conocidos, etc.
vcard	vcard	http://www.w3.org/2006/vcard/ns#	Tarjetas personales electrónicas (<i>electronic business cards</i>)
Contact	contact	http://www.w3.org/2000/10/swap/pim/contact	Contactos personales, lengua materna, etc.
Bio	bio	http://purl.org/vocab/bio/0.1/	Información biográfica
Relationship	relationship	http://purl.org/vocab/relationship	Relaciones entre personas (friendOf, parentOf, etc.)

1.2.3. Vocabularios de libros y bibliografía

Los libros se pueden describir con bastante precisión utilizando las propiedades del vocabulario <http://schema.org/Book>, que define el formato, el número de páginas, el titular de los derechos de autor, el género literario y otras características de los libros. Dublin Core también se usa en muchas ocasiones para definir metadatos sobre libros.

El ISBN¹ es un número internacional para dotar a cada libro de un código numérico que lo identifique. También es una propiedad de la clase «Book» de schema.org (definida en <http://schema.org/isbn>).

⁽¹⁾Acrónimo de sus siglas en inglés, *International Standard Book Number*.

Los libros que ya se han leído o libros favoritos se pueden describir utilizando el esquema de lista de lectura a través del espacio de nombres <http://purl.org/net/schemas/book/>.

PRISM (*Publishing Requirements for Industry Standard Metadata*) describe muchos componentes sobre contenido impreso, en línea, móvil o multimedia, incluidos los siguientes:

- Creador, colaborador, propietario de los derechos de autor.

- Lugares, organizaciones, temas, personas, eventos, condiciones de reproducción.
- Fecha de publicación, incluyendo fecha de portada, fecha de edición, volumen, número, etc.
- Restricciones de publicación y reutilización.

PRISM se usa para describir agregación de contenidos, reutilización de contenidos, descubrimiento de recursos, captura de información sobre el uso de derechos y anotaciones en la web. El espacio de nombres para PRISM 2.1 Basic lleva el prefijo prism y para PRISM 3.0 el prefijo es prism-ad.

1.2.4. Vocabularios para comunidades

SIOC (*Semantically-Interlinked Online Communities*) permite representar información sobre redes sociales y comunidades en línea. Este vocabulario también ofrece URI para indicar que un recurso es de un determinado tipo en el contexto de redes sociales (un foro, un post, un sitio web, etc.). Por ejemplo, con el predicado «Name» se indica un nombre de usuario de una persona, o `member_of` para indicar si un usuario es miembro de un grupo o aplicación.

Facebook utiliza el vocabulario de OpenGraph (og) que consiste básicamente en una plataforma centrada en aspectos sociales de los usuarios. El funcionamiento de **OpenGraph** permite que las acciones específicas realizadas por los usuarios sean publicadas y compartidas a través de su cronología, noticias o barra lateral.

SIOC

El espacio de nombres de SIOC Core es <http://rdfs.org/sioc/ns#>. Su especificación está en <http://rdfs.org/sioc/spec/>.

OpenGraph

El espacio de nombres de OpenGraph es <http://ogp.me/ns#>.

2. Taxonomías

Las taxonomías proporcionan los términos o categorías por medio de las que se puede describir una entidad. El objetivo de la taxonomía es determinar qué clase es la más adecuada para una entidad dada. Es similar al trabajo de un bibliotecario que tiene que clasificar los nuevos libros según una taxonomía.

Una **taxonomía** es un vocabulario controlado organizado en forma jerárquica.

Sin embargo antes de que un bibliotecario pueda realizar esa tarea tiene que existir una taxonomía. Melvil Dewey fue bibliotecario en la Amherst Library y desarrolló el Sistema de Clasificación Decimal Dewey en la década de 1870. Dewey ideó un sistema de clasificación que asignaba los temas en un rango entre 0 y 1.000. Una de las principales aspiraciones en un sistema de clasificación es minimizar la ambigüedad. Las jerarquías son herramientas de clasificación útiles porque reducen la ambigüedad.

El problema de estos sistemas es que una misma entidad tenga más de una categoría y que estas categorizaciones estén en ramas diferentes.

Finalmente, un **tesauro** es una taxonomía con información adicional sobre cada término que incluye términos preferidos y alternativos. Los términos guardan entre sí relaciones semánticas y genéricas: de equivalencia, jerárquicas y asociativas.

2.1. SKOS

SKOS (*Simple Knowledge Organization System*) es una recomendación del W3C para representar esquemas de clasificación como vocabularios controlados, taxonomías y tesauros. Muchos de estos sistemas comparten una estructura similar y se utilizan en aplicaciones parecidas.

SKOS permite compartir datos entre aplicaciones. En SKOS los conceptos se identifican mediante URI, con etiquetas en uno o más idiomas y documentados con diferentes tipos de notas. Los conceptos pueden relacionarse semánticamente entre sí, en jerarquías y redes de asociación informales, así como también se los puede agrupar en esquemas conceptuales.

A continuación se presentan los principales elementos de SKOS:

- `skos:Concept`. Constituye el elemento fundamental de SKOS. Es una clase que define que un determinado recurso es un concepto.
- `skos:prefLabel`, `skos:altLabel`. Son etiquetas que permiten hacer referencia a conceptos en lenguaje natural. Aquí `prefLabel` es la etiqueta para la definición principal y `altLabel` es una definición alternativa; por ejemplo, para sinónimos.
- `skos:broader`, `skos:narrower`. El significado de un concepto está vinculado con otros conceptos del vocabulario. Así, `broader` indica que un concepto es más amplio que otro (un concepto engloba o incluye a otro concepto) y `narrower` indica lo contrario, que un concepto es más concreto que otro.
- `skos:related`. Expresa una relación asociativa entre dos conceptos.
- `skos:note`. Se refiere a una observación genérica respecto al concepto. Existe la posibilidad de calificar diferentes tipos de observaciones utilizando `skos:scopeNote`, `skos:historyNote`, `skos:editorialNote` o `skos:changeNote` relativas al ámbito, histórico, cuestiones editoriales o cambios realizados, respectivamente.
- `skos:definition`. Indica una definición del concepto.
- `skos:ConceptScheme`. Los conceptos se pueden utilizar como entidades independientes, aunque normalmente están organizados como esquemas de clasificación o tesauros. Estos esquemas pueden representarse con esta clase.

El modelo de datos de SKOS está basado en RDF y estructura los datos en forma de tripletas que pueden representarse en cualquier formato válido de RDF.

Figura 13. Taxonomía con SKOS

```
@prefix ex: <.> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix skos: <http://www.w3.org/2008/05/skos#> .

ex:Planet    rdf:type      skos:Concept ;
              skos:prefLabel "Planeta"@es ;
              skos:related  ex:Satellite .

ex:Dwarf_Planet rdf:type      skos:Concept ;
                skos:prefLabel "Planeta Enano"@es ;
                skos:narrower ex:Planet .

ex:Satellite  rdf:type      skos:Concept ;
              skos:prefLabel "Satélite"@es ;
              skos:altLabel  "Luna"@es;
              skos:related  ex:Planet .
```


Bibliografía

DuCharme, R. (2013). *Learning SPARQL* (2.^a ed.). O'Reilly Media.

Guarino, N.; Oberle, D.; Staab, S. (2009). *What Is an Ontology?* [en línea]. [Fecha consulta: enero 2020]. Disponible en: <https://iaoa.org/isc2012/docs/Guarino2009_What_is_an_Ontology.pdf>

Kumar, A. (2018). *Architecting Data-Intensive Applications*. Packt Pub.

Noy, N. F.; McGuinness, D. L. (2001). *Ontology development 101: A guide to creating your first ontology*. Stanford knowledge systems laboratory technical report (Informe SMI-2001-0880).

Powers, S. (2003). *Practical RDF*. O'Reilly Media.

