Ontologías

PID_00272100

Blas Torregrosa García

Tiempo mínimo de dedicación recomendado: 2 horas

Universitat Oberta de Catalunya

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

En primer lugar se considerará la representación del conocimiento y el concepto de ontología, así como los diferentes tipos de ontologías. A continuación se presentarán las características del lenguaje OWL como principal lenguaje ontológico de la web semántica, y sus capacidades para modelar colecciones y objetos.

1. Representación del conocimiento

Para mejorar la capacidad de procesar automáticamente la web se necesitan estándares formales de representación del conocimiento que puedan usarse, no solo para anotar datos simples, sino también para poder expresar enunciados y relaciones complejas de manera procesable por aplicaciones informáticas.

1.1. Lógica descriptiva

En el ámbito de la inteligencia artificial, la **lógica** es un formalismo de representación del conocimiento que se utiliza para construir modelos de la realidad de un determinado dominio y que permite realizar razonamientos sobre él.

Una lógica está definida por las siguientes partes:

1) Sintaxis. Describe las reglas que permite construir expresiones o enunciados lícitos en un lenguaje.

2) Semántica. Describe la interpretación de los hechos a los que se refieren los enunciados. Cada enunciado contiene una afirmación sobre el dominio.

3) Teoría. Describe las reglas para generar nuevos enunciados a partir de otros enunciados que se sabe que son ciertos.

Entre las características más importantes que debe tener un **sistema lógico** se encuentran:

- **Consistencia**. Cuando ninguna regla del sistema contradice a otra.
- **Validez**. Cuando el sistema no permite nunca inferir enunciados falsos a partir de enunciados verdaderos.
- **Completitud**. Cuando se puede demostrar cualquier enunciado cierto.
- **Decibilidad**. Cuando para cada enunciado existe un algoritmo que determina en un número finito de pasos si es cierto o no.

La lógica descriptiva distingue tres tipos de elementos: clases (o conceptos), propiedades (o roles) e individuos. Las **clases** representan conjuntos de individuos, las **propiedades** representan relaciones binarias entre individuos y los **individuos** representan elementos de un dominio.

Una **teoría** en una lógica descriptiva no contiene todo el conocimiento del dominio que pretende modelar, más bien consiste en un conjunto de enunciados, denominados *axiomas*, que deben ser ciertos en la realidad descrita.

2. Ontologías

Las **ontologías** son definiciones complejas y muy formales de términos, individuos y sus propiedades, grupos de objetos (clases) y las relaciones permitidas entre individuos.

Ontología

La Ontología (con O mayúscula) es la disciplina filosófica que trata la naturaleza y la estructura de la realidad.

Las ontologías también se denominan *modelos del dominio*, puesto que posibilitan compartir el conocimiento.

Las **propiedades** principales de una ontología son:

1) Representación explícita, es decir, viene escrita en un lenguaje formal.

2) Conceptualización compartida, es decir, representa la información que un conjunto de personas tiene con respecto a un dominio de conocimiento.

Conceptualización

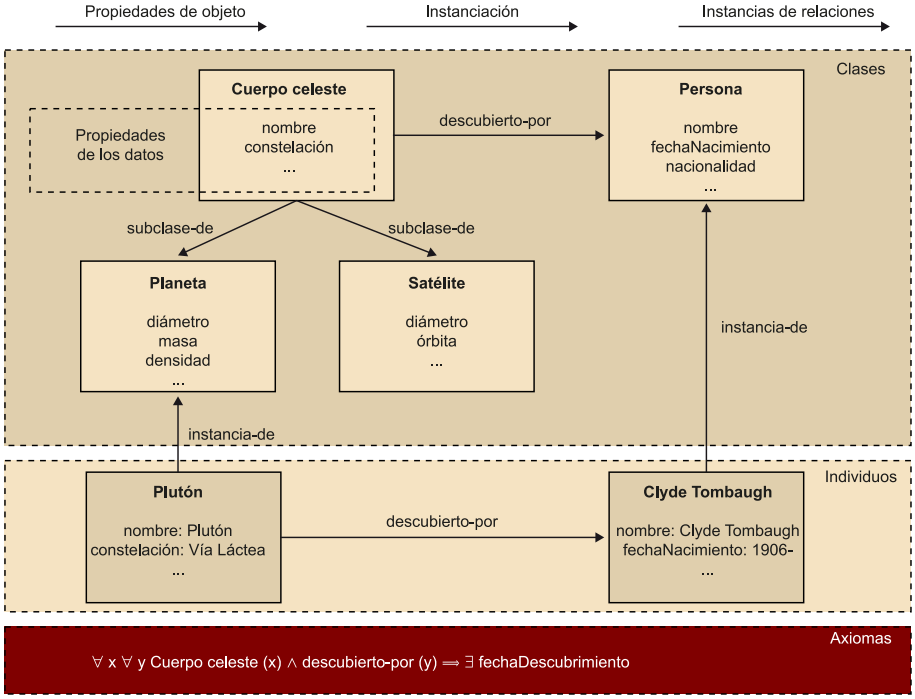
Una **conceptualización** es un punto de vista abstracto y simplificado del mundo que nos interesa representar por algún motivo.

3) Dominio concreto, por lo tanto representa el dominio del conocimiento relevante para un problema particular.

Las ontologías son un mecanismo para conseguir el objetivo de la web de datos, al facilitar la definición formal de los conceptos presentes en el dominio, la jerarquía que los fundamenta y las diferentes relaciones que los vinculan entre sí.

En la figura 1 se pueden apreciar los diferentes elementos que componen una ontología.

Figura 1. Fragmento de ontología sobre cuerpos celestes



2.1. Tipos de ontologías

Hay diferentes formas de clasificar las ontologías. En este apartado vamos a clasificarlas según los grados de generalidad (Nicola Guarino, 1998). De acuerdo con esta clasificación, podemos distinguir los siguientes tipos de ontologías:

Referencia bibliográfica

Nicola Guarino (1998). «Formal Ontology and Information Systems». *Proceedings of FOIS'98*.

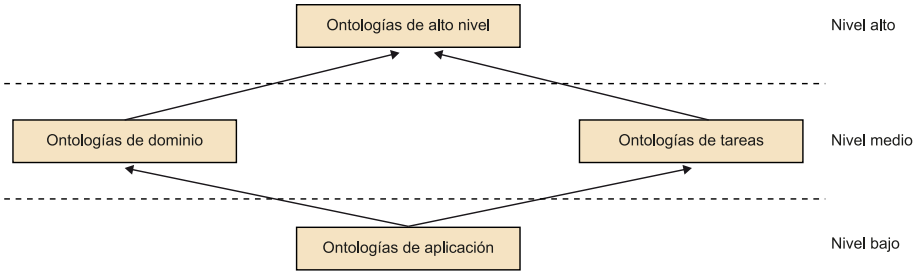
1) Ontologías de alto nivel: describen conceptos generales de la realidad, como el espacio, el tiempo, los objetos, los eventos, las acciones, etc. Estos conceptos son independientes de cualquier problema o dominio en particular. Estas ontologías se usan por grandes comunidades y, por tanto, precisan de un gran nivel de acuerdo.

2) Ontologías de dominio: describen un vocabulario relacionado con un dominio genérico (como medicina, biología, química o música) especializado y, aunque utiliza términos de las ontologías de alto nivel, requieren de una terminología más especializada.

3) Ontologías de tareas o actividades: describen el vocabulario especializado de una tarea o actividad genérica (como diagnosticar, vender o planificar) por medio de la particularización de las ontologías de alto nivel.

4) Ontologías de aplicación: son las ontologías más específicas de todas y describen conceptos que dependen de dominios o tareas concretas. Tienen una reutilización limitada, puesto que dependen del ámbito concreto y de los requisitos de una aplicación en particular. En ocasiones existen distintas ontologías que modelan el mismo dominio o tarea.

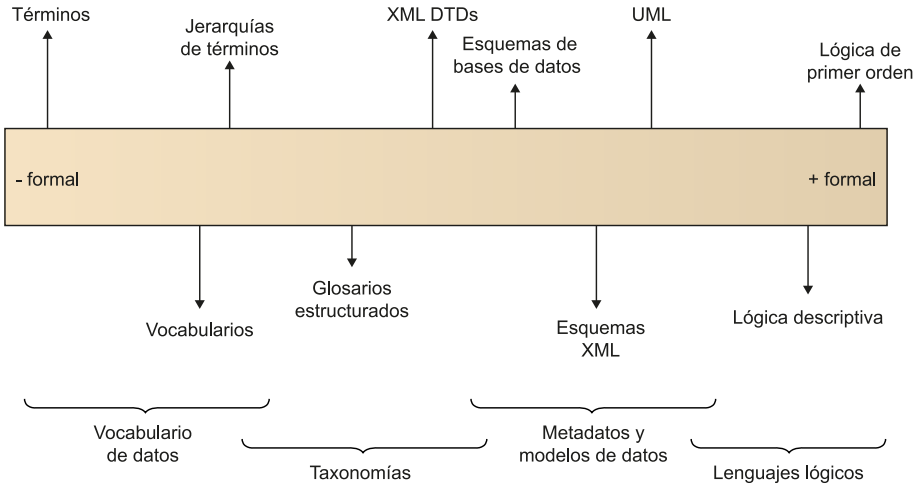
Figura 2. Tipos de ontologías según el nivel de especialización



Clasificando de esta forma las ontologías se potencia la reutilización del conocimiento de las ontologías de una forma modular, pues las ontologías de alto nivel describen el conocimiento general, mientras que las ontologías de tarea describen el conocimiento de una determinada tarea.

Por otro lado se pueden distinguir ontologías con mayor o menor grado de formalismo. Así, hay desde ontologías informales expresadas en lenguajes que solo permiten la definición de los términos usados hasta ontologías rigurosamente formales.

Figura 3. Rango de formalismos para expresar ontologías



2.2. OWL

Mientras que algunas ontologías sencillas se pueden crear usando RDFS, ciertos dominios de conocimiento exigen más capacidades, como las mencionadas a continuación:

- Relaciones entre clases (unión, intersección, desunión, equivalencia).

- Restricciones de cardinalidad de las propiedades (mínimo, máximo, número exacto).
- Tipos de las propiedades mejorados (objetos como tipos de datos, tipos de datos específicos).
- Características de propiedades especiales (transitiva, simétrica, funcional).
- Definición de una propiedad como clave única para instancias de una clase.
- Igualdad entre clases, que expresa que dos clases con diferente URI representan a la misma clase.
- Igualdad entre individuos, que expresa que dos instancias con diferentes URI representan al mismo individuo.

Propiedad funcional

Hacer una propiedad funcional significa que solo habrá un objeto para un sujeto con esa propiedad.

OWL[1] (**Web Ontology Language**) es un lenguaje que extiende RDF y RDFS, que ofrece un conjunto mucho más amplio de posibilidades para crear ontologías web, y que supera las limitaciones de RDFS.

(1)El acrónimo de *Web Ontology Language* no es WOL, sino OWL de manera no casual.

El desarrollo de la primera versión de OWL empezó en 2002 y la segunda versión OWL2 en 2008. OWL es recomendación de W3C desde 2004. Desde 2009 OWL2 está estandarizado.

OWL es un lenguaje muy extenso con partes complejas. De hecho, el propio OWL está dividido en tres sublenguajes de creciente capacidad de complejidad y expresividad, denominados *OWL-Lite* (el más sencillo), *OWL DL* y *OWL Full*. En este apartado veremos someramente OWL-Lite y su capacidad para definir tipos y propiedades (que son las aplicaciones más útiles de OWL).

OWL divide el universo de recursos en tres: individuos, clases y propiedades, que se corresponden con las extensiones de las clases `rdfs:Resource` (individuos), `rdfs:Class` (clases) y `rdfs:Property` (propiedades). Las principales **características** de OWL son:

1) Clases. Una clase permite agrupar un conjunto de instancias (individuos) que comparten propiedades. Relacionado con las clases:

- `owl:Thing`. La clase de todas las cosas en OWL. Todas las clases son subclases de esta clase; por tanto, todas las instancias serán implícitamente instancias de `owl:Thing`. El uso de esta clase es muy similar a la clase *Object* en los lenguajes de programación orientados a objetos. Se pueden uti-

lizar propiedades con rango y dominio de `rdfs:Resource` con cualquier instancia.

- `owl:Class`. La clase de los recursos RDF que son clases. Todas las clases son instancias de `rdfs:Class`.

- `owl:Nothing`. Denota la clase vacía, es decir, que no contiene ningún individuo.

2) Individuos. Son instancias de las clases.

3) Propiedades. Son relaciones binarias entre dos instancias:

- `owl:ObjectProperty`. Denota la clase de las propiedades relacionales, es decir, aquellas cuyo rango son individuos y no datos básicos.

- `owl:DatatypeProperty`. Denota la clase de las propiedades de tipos de datos, es decir, aquellas cuyo rango son los tipos de datos básicos.

- `owl:FunctionalProperty`. Denota la clase de las propiedades funcionales.

- `owl:ReflexiveProperty`. Denota la clase de las propiedades reflexivas.

- `owl:SymmetricProperty`. Denota la clase de las propiedades simétricas.

- `owl:TransitiveProperty`. Denota la clase de las propiedades transitivas.

- `owl:Restriction`. Denota la clase de las restricciones sobre propiedades.

4) Recursos. Es cualquier elemento que tiene un URI.

Una ontología codificada en OWL consiste en un conjunto de enunciados (axiomas), separadas en tres grupos:

1) Tbox (*Terminological*). Tbox describe relaciones entre clases. Por ejemplo, el enunciado de que la clase «Persona» es equivalente a la clase «Ser_Humano» significa que ambas clases poseen el mismo conjunto de individuos.

2) Abox (*Assertion*). Abox captura conocimiento sobre los individuos, es decir, las clases a las cuales pertenecen o cómo se relacionan entre sí. Por ejemplo, podemos construir un enunciado sobre el individuo «Alicia» perteneciente a la clase «Persona», es decir, que «Alicia» es una instancia de la clase «Persona».

Si se une con el ejemplo de Tbox, puede inferirse que «Alicia» también es una instancia de la clase «Ser_Humano». En Tbox también pueden definirse relaciones entre los individuos.

3) Rbox (*Role*). Rbox contiene enunciados sobre las propiedades, es decir, las metapropiedades, como la transitividad, la simetría, etc. Estos enunciados permiten realizar inferencias a partir de tripletas definidas de manera explícita.

2.2.1. Clases

Puesto que no todas las clases definidas en RDFS son válidas en OWL, existe las clase `owl:Class` que es subclase de `rdfs:Class`. Al igual que en RDFS, unas clases pueden estar incluidas en otras mediante la propiedad `rdfs:subClassOf`. Todas las clases OWL son subclases de la clase universal `owl:Thing`, que contiene a todos los individuos del universo. Su complemento es la clase `owl:Nothing` que representa la clase vacía.

OWL aporta nuevos constructores para definir clases o establecer características de estas:

1) Clases equivalentes. Una clase puede definirse como equivalente a otra mediante el constructor `owl:equivalentClass`, lo que indica que ambas clases tienen los mismos individuos.

Figura 4. El axioma establece que las clases «Satélite» y «Luna» son equivalentes

```
@prefix ex: <.> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .

ex:Satellite a owl:Class ;
   owl:equivalentClass ex:Moon .
```

2) Clases disjuntas. Una clase puede definirse como disjunta de otra mediante el constructor `owl:disjointWith`, lo que significa que ambas clases no tendrán ningún elemento en común.

Figura 5. El axioma establece que la clase «Planeta» es disjunta con «Estrella» y con «Satélite»

```
@prefix ex: <.> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .

ex:Planet a owl:Class ;
   owl:disjointWith ex:Star ;
   owl:disjointWith ex:Satellite .
```

3) Enumeraciones. Permite definir clases enumerando todos y cada uno de sus individuos.

Figura 6. El axioma establece la clase «Sistema solar» compuesto por una serie de planetas

```
@prefix ex: <.> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .

ex:Solar_System a owl:Class ;
   owl:equivalentClass [
      owl.OneOf (ex:Mercury, ex:Venus, ex:Earth, ex:Mars,
                 ex:Jupiter ex:Saturn ex:Neptune )] .
```

2.2.2. Individuos

En OWL la asignación de un individuo a una clase se realiza como en RDF mediante `rdf:type` o simplemente con `a`. En OWL cuando en una ontología se definen los individuos pertenecientes a una clase, la clase no está cerrada y puede haber otra ontología en otra parte de la web que defina nuevos individuos de esa clase. Por lo tanto puede no haber nombres únicos para individuos. OWL proporciona el constructor `owl:sameAs` para indicar que dos URI se refieren al mismo individuo, mientras que el constructor `owl:differentFrom` indica que se refiere a individuos distintos.

2.2.3. Propiedades

OWL dispone de los constructores de RDFS `rdfs:domain` y `rdfs:range` para indicar el dominio y el rango de una propiedad. OWL aporta nuevas posibilidades para definir las propiedades y las características de estas:

1) Subpropiedades. Una propiedad puede definirse como subpropiedad de otra mediante `rdfs:subPropertyOf`, de forma que cualquier par de individuos relacionados por esta propiedad también lo estarán mediante la superpropiedad.

2) Propiedades equivalentes. Una propiedad puede definirse como equivalente a otra mediante `owl:equivalentProperty`, lo que significa que ambas propiedades relacionan el mismo conjunto de entidades relacionadas. De la misma forma, si dos propiedades son equivalentes, significa que la primera es subpropiedad de la segunda y la segunda subpropiedad de la primera.

3) Propiedades inversas. Una propiedad puede definirse inversa de otra mediante `owl:inverseOf`, lo que significa que si un individuo *a* está relacionado con otro *b* mediante la propiedad, entonces *b* está relacionado con *a* mediante la inversa.

Figura 7. El axioma establece que la propiedad «Ser satélite» es la inversa de «Tiene satélite»

```
@prefix ex: <.> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .

ex:satelliteOf a owl:ObjectProperty ;
   owl:inverseOf ex:hasSatellite    .
```

4) Propiedades funcionales. Una propiedad puede definirse como funcional asignándola a la clase `owl:FunctionalProperty`, lo que implica que un individuo no puede estar relacionado mediante esta propiedad con más de un individuo o literal. Si un individuo estuviera relacionado mediante una propiedad funcional con dos individuos, se concluye que ambos individuos son el mismo.

Figura 8. El axioma establece que la propiedad «Descubrir» relaciona un «Planeta» con una «Persona»

```
@prefix ex: <.> .
@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#> .

ex:discoverer a owl:FunctionalProperty ;
   rdfs:domain ex:Planet  ;
   rdfs:range  ex:Person  .
```

5) Propiedades simétricas. Una propiedad puede definirse como simétrica mediante la clase `owl:SymmetricProperty`, lo que significa que si un individuo está relacionado con otro mediante esta propiedad, entonces el segundo también está relacionado con el primero con la misma propiedad.

Figura 9. El axioma establece que la relación estrella binaria es simétrica

```
@prefix ex: <.> .
@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#> .

ex:binaryStar a owl:SimmetricProperty ;
   rdfs:domain ex:Star  ;
   rdfs:range  ex:Star  .
```

6) Propiedades transitivas. Una propiedad puede definirse como transitiva mediante la clase `owl:TransitiveProperty`, lo que significa que si un individuo está relacionado con un segundo mediante la propiedad y este a su vez está relacionado con un tercero mediante la misma propiedad, entonces el primero también lo está con el tercero.

En principio cualquier propiedad de objeto puede ser cualquiera de las descritas, aunque no todas las combinaciones de las mismas están permitidas. En concreto, en el caso de propiedades compuestas la secuencia de las propiedades equivalente y transitiva puede causar problemas de decibilidad.

2.3. Ejemplos de ontologías

A continuación se exponen algunos ejemplos de ontologías para diversos ámbitos.

2.3.1. Ontologías de publicación

Si bien los metadatos de Dublin Core sirven para describir publicaciones, existen ontologías especialmente diseñadas para describir publicaciones y citas.

Tabla 1. Ontologías de publicación y citación

Ontología	Abrev.	Espacio de nombres	Uso
Bibliographic Ontology	bibo	http://purl.org/ontology/bibo	Citas bibliográficas, descripción de documentos, etc.
Bibliographic Reference Ontology	biro	http://purl.org/spar/biro	Registros bibliográficos, referencias, colecciones y listas
Citation Counting and Context Characterization Ontology	c4o	http://purl.org/spar/c4o	Número de citas
Citation Typing Ontology	cito	http://purl.org/spar/cito	Caracterización de la naturaleza o tipo de cita
Document Components Ontology	doco	http://purl.org/spar/doco	Capítulo, sección, párrafo, tabla, glosario, etc.
FRBR-aligned Bibliographic Ontology	fabio	http://purl.org/spar/fabio	Resúmenes, artículos, tesis, comunicaciones de congresos, blogs, etc.
Publishing Roles Ontology	pro	http://purl.org/spar/pro/	Roles: autor, editor, revisor, etc.
Publishing Status Ontology	pso	http://purl.org/spar/pso	Estados de una publicación: enviada, aceptada, revisada, etc.
Publishing Workflow Ontology	pwo	http://purl.org/spar/pwo	Etapas del flujo de publicación: en revisión, etc.

Las cuatro ontologías de publicación más implementadas (FaBiO, PRO, PSO y PWO) y las cuatro ontologías de referencias (CiTO, BiRO, C4O y DoCO) conocidas conjuntamente como **SPAR** (*Semantic Publishing and Referencing Ontologies*) constituyen un ecosistema integrado de ontologías genéricas. Estas ontologías se pueden usar individualmente o en conjunto, según lo requiera el caso.

2.3.2. Ontologías de música y vídeo

Existen ontologías dedicadas a los recursos multimedia como música o vídeo, como las que se pueden ver en la tabla 2.

Tabla 2. Ontologías de música y vídeo

Ontología	Abrev.	Espacio de nombres	Uso
The Music Ontology	mo	http://purl.org/ontology/mo/	Artistas, compositores, discografía, imdb, grabaciones, remezclas, cantantes, tempo, etc.
VidOnt: The Video Ontology	vidont	http://vidont.org/	Propiedades de películas (remake, secuela, etc.), propiedades del archivo de vídeo (códec, tasa de bits, etc.).

2.3.3. Ontología de la DBpedia

La ontología de la *DBpedia Ontology* es una ontología general que actualmente incluye 685 clases que forman una jerarquía y tiene descritas 2.795 propiedades diferentes. La ontología es un gráfico acíclico dirigido, no un árbol. Las clases pueden tener múltiples superclases y esto es importante para las correspondencias con schema.org.

Enlace de interés

La versión actual de *DBpedia Ontology* puede encontrarse en mappings.dbpedia.org/server/ontology/classes/.

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.