

---

# Ontologies

---

PID\_00272099

Blas Torregrosa García

---

Temps mínim de dedicació recomanat: 2 hores

---



**Blas Torregrosa García**

Enginyer en Informàtica i màster universitari en Seguretat de les Tecnologies de la Informació i de les Comunicacions (MISTIC) per la Universitat Oberta de Catalunya (UOC). Especialitzat en ciberseguretat. Professor col·laborador del màster de Ciència de Dades de la UOC i professor associat a la Universitat de Valladolid (UVA).

L'encàrrec i la creació d'aquest recurs d'aprenentatge UOC han estat coordinats pel professor: Ferran Prados Carrasco (2020)

Primera edició: febrer 2020  
© Blas Torregrosa García  
Tots els drets reservats  
© d'aquesta edició, FUOC, 2020  
Av. Tibidabo, 39-43, 08035 Barcelona  
Realització editorial: FUOC

*Cap part d'aquesta publicació, incloent-hi el disseny general i la coberta, no pot ser copiada, reproduïda, emmagatzemada o transmesa de cap manera ni per cap mitjà, tant si és elèctric com químic, mecànic, òptic, de gravació, de fotocòpia o per altres mètodes, sense l'autorització prèvia per escrit dels titulars dels drets.*

# Índex

<b>Introducció</b> .....	5
<b>1. Representació del coneixement</b> .....	7
1.1. Lògica descriptiva .....	7
<b>2. Ontologies</b> .....	9
2.1. Tipus d'ontologies .....	10
2.2. OWL .....	11
2.2.1. Classes .....	13
2.2.2. Individus .....	14
2.2.3. Propietats .....	15
2.3. Exemples d'ontologies .....	16
2.3.1. Ontologies de publicació .....	16
2.3.2. Ontologies de música i vídeo .....	17
2.3.3. Ontologia de la DBpedia .....	17
<b>Bibliografia</b> .....	19



## **Introducció**

En primer lloc es considerarà la representació del coneixement i el concepte d'ontologia, i també els diferents tipus d'ontologies. A continuació es presentaran les característiques del llenguatge OWL, com a principal llenguatge ontològic de la web semàntica, i les seves capacitats per modelar les col·leccions i els objectes.



# 1. Representació del coneixement

Per millorar la capacitat de processar automàticament la web calen estàndards formals de representació del coneixement que es puguin usar, no solament per anotar dades simples, sinó també per poder expressar enunciats i relacions complexes de manera processable per a les aplicacions informàtiques.

## 1.1. Lògica descriptiva

En l'àmbit de la intel·ligència artificial, la **lògica** és un formalisme de representació del coneixement que s'utilitza per construir models de la realitat d'un determinat domini i que permet realitzar raonaments sobre ell.

Una lògica està definida per les parts següents:

- 1) **Sintaxi**. Descriu les regles que permet construir expressions o enunciats lícits en un llenguatge.
- 2) **Semàntica**. Descriu la interpretació dels fets als quals es refereixen els enunciats. Cada enunciat conté una afirmació sobre el domini.
- 3) **Teoria**. Descriu les regles per generar nous enunciats a partir d'altres enunciats que se sap que són certs.

Entre les característiques més importants que ha de tenir un **sistema lògic** trobem:

- **Consistència**. Quan cap regla del sistema no contradiu a una altra.
- **Validesa**. Quan el sistema no permet mai inferir enunciats falsos a partir d'enunciats veritables.
- **Completesa**. Quan es pot demostrar qualsevol enunciat cert.
- **Dicibilitat**. Quan per a cada enunciat hi ha un algorisme que determina en un nombre finit de passos si és cert o no.

La lògica descriptiva distingeix tres tipus d'elements: classes (o conceptes), propietats (o rols) i individus. Les **classes** representen conjunts d'individus, les **propietats** representen relacions binàries entre els individus i els **individus** representen elements d'un domini.

Una **teoria** en una lògica descriptiva no conté tot el coneixement del domini que pretén modelar, més aviat consisteix en un conjunt d'enunciats, denominats *axiomes*, que han de ser certs en la realitat descrita.



## 2. Ontologies

Les **ontologies** són definicions complexes i molt formals de termes, individus i les seves propietats, grups d'objectes (classes) i les relacions permeses entre els individus.

### Ontologia

L'Ontologia (amb O majúscula) és la disciplina filosòfica que tracta la naturalesa i l'estructura de la realitat.

Les ontologies també es denominen *models del domini*, ja que possibiliten compartir el coneixement.

Les **propietats** principals d'una ontologia són:

- 1) **Representació explícita**, és a dir, està escrita en un llenguatge formal.
- 2) **Conceptualització compartida**, és a dir, representa la informació que un conjunt de persones té pel que fa a un domini de coneixement.
- 3) **Domini concret**, per tant, representa el domini del coneixement rellevant per a un problema particular.

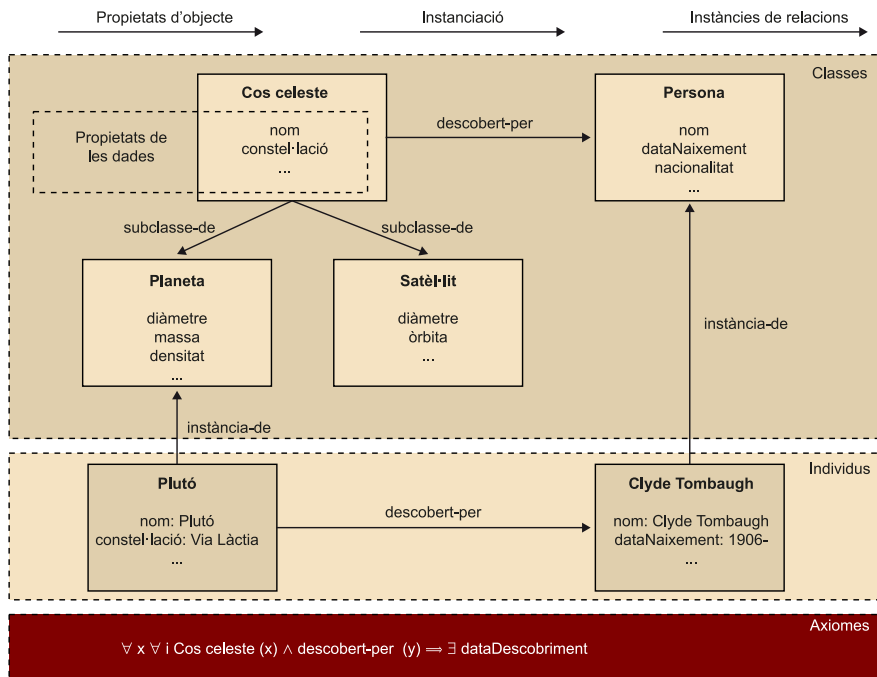
### Conceptualització

Una **conceptualització** és un punt de vista abstracte i simplificat del món que ens interessa representar per algun motiu.

Les ontologies són un mecanisme per aconseguir l'objectiu de la web de dades, atès que facilita la definició formal dels conceptes presents en el domini, la jerarquia que els fonamenta i les diferents relacions que els vinculen entre si.

En la figura 1 es poden apreciar els diferents elements que componen una ontologia.

Figura 1. Fragment d'ontologia sobre cossos celestes



## 2.1. Tipus d'ontologies

Hi ha diferents formes de classificar les ontologies. En aquest apartat les classificarem segons els graus de generalitat (Nicola Guarino, 1998). D'acord amb aquesta classificació, podem distingir els tipus d'ontologies següents:

1) **Ontologies d'alt nivell:** descriuen conceptes generals de la realitat, com ara l'espai, el temps, els objectes, els esdeveniments, les accions, etc. Aquests conceptes són independents de qualsevol problema o domini en particular. Aquestes ontologies s'usen per grans comunitats i, per tant, precisen d'un gran nivell d'acord.

2) **Ontologies de domini:** descriuen un vocabulari relacionat amb un domini genèric (com ara medicina, biologia, química o música) especialitzat i, encara que utilitza termes de les ontologies d'alt nivell, requereixen una terminologia més especialitzada.

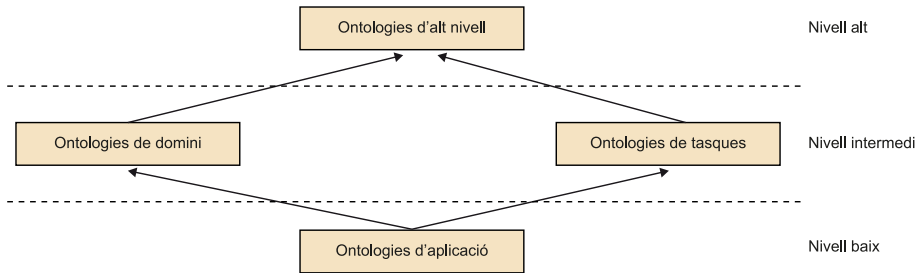
3) **Ontologies de tasques o activitats:** descriuen el vocabulari especialitzat d'una tasca o activitat genèrica (com ara diagnosticar, vendre o planificar) per mitjà de la particularització de les ontologies d'alt nivell.

4) **Ontologies d'aplicació:** són les ontologies més específiques de totes i descriuen conceptes que depenen de dominis o tasques concretes. Tenen una reutilització limitada, ja que depenen de l'àmbit concret i dels requisits d'una aplicació en particular. De vegades hi ha diferents ontologies que modelen el mateix domini o tasca.

### Referència bibliogràfica

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

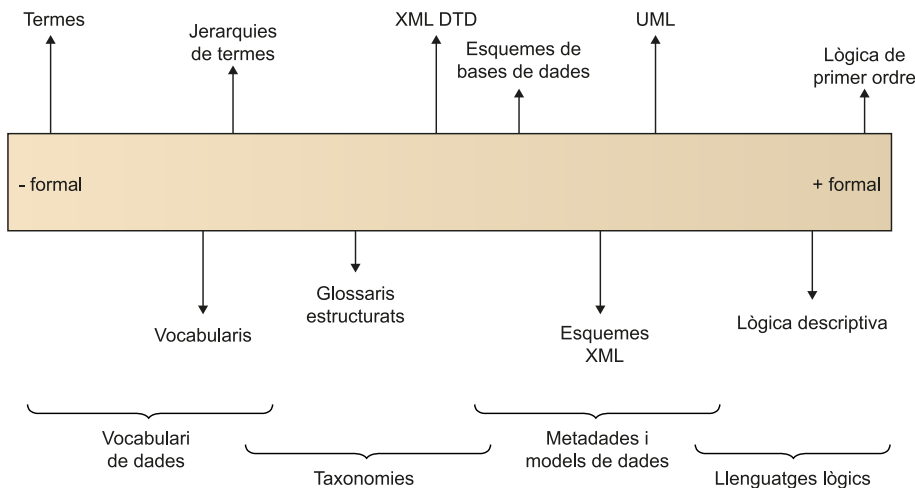
Figura 2. Tipus d'ontologies segons el nivell d'especialització



Classificant d'aquesta forma les ontologies es potencia la reutilització del coneixement de les ontologies d'una forma modular, ja que les ontologies d'alt nivell descriuen el coneixement general, mentre que les ontologies de tasca descriuen el coneixement d'una determinada tasca.

D'altra banda, es poden distingir les ontologies amb major o menor grau de formalisme. Així, hi ha des d'ontologies informals expressades en llenguatges que solament permeten la definició dels termes usats fins a ontologies rigorosament formals.

Figura 3. Rang de formalismes per expressar les ontologies



## 2.2. OWL

Mentre que algunes ontologies senzilles es poden crear usant RDFS, certs dominis de coneixement exigeixen més capacitats, com les esmentades a continuació:

- Relacions entre classes (unió, intersecció, desunió, equivalència).
- Restriccions de cardinalitat de les propietats (mínim, màxim, nombre exacte).
- Tipus de propietats millorades (objectes com a tipus de dades, tipus de dades específiques).

- Característiques de propietats especials (transitiva, simètrica, funcional).
- Definició d'una propietat com a clau única per a instàncies d'una classe.
- Igualtat entre classes, que expressa que dues classes amb diferent URI representen la mateixa classe.
- Igualtat entre individus, que expressa que dues instàncies amb diferents URI representen el mateix individu.

#### Propietat funcional

Fer una propietat funcional significa que solament hi haurà un objecte per a un subjecte amb aquesta propietat.

**OWL<sup>1</sup> (Web Ontology Language)** és un llenguatge que estén RDF i RDFS, que ofereix un conjunt molt més ampli de possibilitats per crear ontologies web i que supera les limitacions d'RDFS.

<sup>(1)</sup>L'acrònim de *Web Ontology Language* no és WOL, sinó OWL de manera no casual.

El desenvolupament de la primera versió d'OWL va començar el 2002 i la segona versió OWL2 el 2008. OWL és la recomanació de W3C des de 2004. Des de 2009 OWL2 està estandarditzat.

OWL és un llenguatge molt extens amb parts complexes. De fet, el propi OWL està dividit en tres subllenguatges de creixent capacitat de complexitat i expressivitat, denominats *OWL-Lite* (el més senzill), *OWL DL* i *OWL Full*. En aquest apartat, veurem succintament OWL-Lite i la seva capacitat per definir tipus i propietats (que són les aplicacions més útils d'OWL).

OWL divideix l'univers de recursos en tres: individus, classes i propietats, que es corresponen amb les extensions de les classes `rdfs:Resource` (individus), `rdfs:Class` (classes) i `rdfs:Property` (propietats). Les principals **característiques** d'OWL són:

**1) Classes.** Una classe permet agrupar un conjunt d'instàncies (individus) que comparteixen propietats. Relacionat amb les classes:

- `owl:Thing`. La classe de totes les coses amb OWL. Totes les classes són subclases d'aquesta classe. Per tant, totes les instàncies seran implícitament instàncies d'`owl:Thing`. L'ús d'aquesta classe és molt similar a la classe *Object* en els llenguatges de programació orientats als objectes. Es poden utilitzar propietats amb rang i domini d'`rdfs:Resource` amb qualsevol instància.
- `owl:Class`. La classe dels recursos RDF que són classes. Totes les classes són instàncies de `rdfs:Class`.
- `owl:Nothing`. Denota la classe buida, és a dir, que no conté cap individu.

2) **Individus.** Són instàncies de les classes.

3) **Propietats.** Són relacions binàries entre dues instàncies:

- `owl:ObjectProperty`. Denota la classe de les propietats relacionals, és a dir, aquelles el rang de les quals són individus i no dades bàsiques.
- `owl:DatatypeProperty`. Denota la classe de les propietats de tipus de dades, és a dir, aquelles el rang de les quals són els tipus de dades bàsiques.
- `owl:FunctionalProperty`. Denota la classe de les propietats funcionals.
- `owl:ReflexiveProperty`. Denota la classe de les propietats reflexives.
- `owl:SymmetricProperty`. Denota la classe de les propietats simètriques.
- `owl:TransitiveProperty`. Denota la classe de les propietats transitives.
- `owl:Restriction`. Denota la classe de les restriccions sobre propietats.

4) **Recursos.** És qualsevol element que té un URI.

Una ontologia codificada amb OWL consisteix en un conjunt d'enunciats (axiomes), separats en tres grups:

1) **Tbox** (*Terminological*). Tbox descriu relacions entre les classes. Per exemple, l'enunciat que la classe «Persona» és equivalent a la classe «Ser\_Humà» significa que ambdues classes posseeixen el mateix conjunt d'individus.

2) **Abox** (*Assertion*). Abox captura coneixement sobre els individus, és a dir, les classes a les quals pertanyen o com es relacionen entre si. Per exemple, podem construir un enunciat sobre l'individu «Alícia» pertanyent a la classe «Persona», és a dir, que «Alícia» és una instància de la classe «Persona». Si s'uneix amb l'exemple de Tbox, es pot inferir que «Alícia» també és una instància de la classe «Ser\_Humà». Amb Tbox també es poden definir relacions entre els individus.

3) **Rbox** (*Role*). Rbox conté enunciats sobre les propietats, és a dir, les metapropietats, com ara la transitivitat, la simetria, etc. Aquests enunciats permeten realitzar inferències a partir de tripletes definides de manera explícita.

### 2.2.1. Classes

Atès que no totes les classes definides en RDFS són vàlides amb OWL, hi ha la classe `owl:Class` que és una subclasse d'`rdfs:Class`. Igual que amb RDFS, unes classes poden estar incloses en altres mitjançant la propietat

`rdfs:subClassOf`. Totes les classes OWL són subclasses de la classe universal `owl:Thing`, que conté tots els individus de l'univers. El seu complement és la classe `owl:Nothing` que representa la classe buida.

OWL aporta nous constructors per definir classes o establir característiques d'aquestes:

1) **Classes equivalents.** Una classe es pot definir com a equivalent a una altra mitjançant el constructor `owl:equivalentClass`, la qual cosa indica que ambdues classes tenen els mateixos individus.

Figura 4. L'axioma estableix que les classes «Satèl·lit» i «Lluna» són equivalents

```
@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) **Classes disjunctes.** Una classe es pot definir com a disjunta d'una altra mitjançant el constructor `owl:disjointWith`, la qual cosa significa que ambdues classes no tindran cap element en comú.

Figura 5. L'axioma estableix que la classe «Planeta» és disjunta amb «Estrella» i amb «Satèl·lit»

```
@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) **Enumeracions.** Permet definir classes enumerant tots i cadascun dels seus individus.

Figura 6. L'axioma estableix la classe «Sistema solar» compost per una sèrie de planetes

```
@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. Individus

Amb OWL l'assignació d'un individu a una classe es realitza, com amb RDF, mitjançant `rdf:type` o simplement amb `a`. Amb OWL quan en una ontologia es defineixen els individus que pertanyen a una classe, la classe no està tancada i hi pot haver una altra ontologia en una altra part de la web que defineixi nous individus d'aquesta classe. Per tant, pot no haver-hi noms

únics per a individus. OWL proporciona el constructor `owl:sameAs` per indicar que dos URI es refereixen al mateix individu, mentre que el constructor `owl:differentFrom` indica que es refereix a individus diferents.

### 2.2.3. Propietats

OWL disposa dels constructors d'RDFS `rdfs:domain` i `rdfs:range` per indicar el domini i el rang d'una propietat. OWL aporta noves possibilitats per definir les propietats i les característiques d'aquestes:

1) **Subpropietats.** Una propietat es pot definir com a subpropietat d'una altra mitjançant `rdfs:subPropertyOf`, de manera que qualsevol parell d'individus relacionats per aquesta propietat també ho estaran mitjançant la subpropietat.

2) **Propietats equivalents.** Una propietat es pot definir com a equivalent a una altra mitjançant `owl:equivalentProperty`, la qual cosa significa que ambdues propietats relacionen el mateix conjunt d'entitats relacionades. De la mateixa forma, si dues propietats són equivalents, significa que la primera és una subpropietat de la segona i la segona una subpropietat de la primera.

3) **Propietats inverses.** Una propietat es pot definir inversa d'una altra mitjançant `owl:inverseOf`, la qual cosa significa que si un individu *a* està relacionat amb un altre *b* mitjançant la propietat, llavors *b* està relacionat amb *a* mitjançant la inversa.

Figura 7. L'axioma estableix que la propietat «Ser satèl·lit» és la inversa de «Té satèl·lit»

```
@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) **Propietats funcionals.** Una propietat es pot definir com a funcional assignant-la a la classe `owl:FunctionalProperty`, la qual cosa implica que un individu no pot estar relacionat mitjançant aquesta propietat amb més d'un individu o literal. Si un individu estigués relacionat mitjançant una propietat funcional amb dos individus, es conclou que tots dos individus són el mateix.

Figura 8. L'axioma estableix que la propietat «Descobrir» relaciona un «Planeta» amb 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) **Propietats simètriques.** Una propietat es pot definir com a simètrica mitjançant la classe `owl:SymmetricProperty`, la qual cosa significa que si un individu està relacionat amb un altre mitjançant aquesta propietat, llavors el segon també està relacionat amb el primer amb la mateixa propietat.

Figura 9. L'axioma estableix que la relació estrella binària és simètrica

```
@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:SymmetricProperty ;
  rdfs:domain ex:Star ;
  rdfs:range ex:Star .
```

6) **Propietats transitives.** Una propietat es pot definir com a transitiva mitjançant la classe `owl:TransitiveProperty`, la qual cosa significa que si un individu està relacionat amb un segon mitjançant la propietat i aquest, al seu torn, està relacionat amb un tercer mitjançant la mateixa propietat, llavors el primer també ho està amb el tercer.

En principi, qualsevol propietat d'objecte pot ser qualsevol de les descrites, encara que no totes les combinacions d'aquestes estan permeses. En concret, en el cas de propietats compostes, la seqüència de les propietats equivalent i transitiva pot causar problemes de dicibilitat.

## 2.3. Exemples d'ontologies

A continuació s'exposen alguns exemples d'ontologies per a diversos àmbits.

### 2.3.1. Ontologies de publicació

Si bé les metadades de Dublin Core serveixen per descriure publicacions, hi ha ontologies especialment dissenyades per descriure publicacions i cites.

Taula 1. Ontologies de publicació i citació

Ontologia	Abrev.	Espai de noms	Ús
<i>Bibliographic Ontology</i>	bibo	<a href="http://purl.org/ontology/bibo">http://purl.org/ontology/bibo</a>	Cites bibliogràfiques, descripció de documents, etc.
<i>Bibliographic Reference Ontology</i>	biro	<a href="http://purl.org/spar/biro">http://purl.org/spar/biro</a>	Registres bibliogràfics, referències, col·leccions i llistes.
<i>Citation Counting and Context Characterization Ontology</i>	c4o	<a href="http://purl.org/spar/c4o">http://purl.org/spar/c4o</a>	Nombre de cites.
<i>Citation Typing Ontology</i>	cito	<a href="http://purl.org/spar/cito">http://purl.org/spar/cito</a>	Caracterització de la naturalesa o tipus de cita.
<i>Document Components Ontology</i>	doco	<a href="http://purl.org/spar/doco">http://purl.org/spar/doco</a>	Capítol, secció, paràgraf, taula, glossari, etc.



Ontologia	Abrev.	Espai de noms	Ús
<i>FRBR-aligned Bibliographic Ontology</i>	fabio	<a href="http://purl.org/spar/fabio">http://purl.org/spar/fabio</a>	Resums, articles, tesis, comunicacions de congressos, blogs, etc.
<i>Publishing Rols Ontology</i>	pro	<a href="http://purl.org/spar/pro/">http://purl.org/spar/pro/</a>	Rols: autor, editor, revisor, etc.
<i>Publishing Status Ontology</i>	pso	<a href="http://purl.org/spar/pso">http://purl.org/spar/pso</a>	Estats d'una publicació: enviada, acceptada, revisada, etc.
<i>Publishing Workflow Ontology</i>	pwo	<a href="http://purl.org/spar/pwo">http://purl.org/spar/pwo</a>	Etales del flux de publicació: en revisió, etc.

Les quatre ontologies de publicació més implementades (FABIO, PRO, PSO i PWO) i les quatre ontologies de referències (CITO, BiRO, C4O i DoCO) conegudes conjuntament com a **SPAR** (*Semantic Publishing and Referencing Ontologies*) constitueixen un ecosistema integrat d'ontologies genèriques. Aquestes ontologies es poden usar individualment o en conjunt, segons ho requereixi el cas.

### 2.3.2. Ontologies de música i vídeo

Hi ha ontologies dedicades als recursos multimèdia com ara música o vídeo, com les que es poden veure en la taula 2.

Taula 2. Ontologies de música i vídeo

Ontologia	Abrev.	Espai de noms	Ús
<i>The Music Ontology</i>	mo	<a href="http://purl.org/ontology/mo/">http://purl.org/ontology/mo/</a>	Artistes, compositors, discografia, imdb, enregistraments, remescles, cantants, tempo, etc.
<i>VidOnt: The Video Ontology</i>	vidont	<a href="http://vidont.org/">http://vidont.org/</a>	Propietats de pel·lícules (remake, seqüela, etc.), propietats de l'arxiu de vídeo (códec, taxa de bits, etc.).

### 2.3.3. Ontologia de la DBpedia

L'ontologia de la *DBpedia Ontology* és una ontologia general que actualment inclou 685 classes que formen una jerarquia i té descrites 2.795 propietats diferents. L'ontologia és un gràfic acíclic dirigit, no un arbre. Les classes poden tenir múltiples superclasses i això és important per a les correspondències amb [schema.org](http://schema.org).

#### Enllaç d'interès

La versió actual de *DBpedia Ontology* es pot trobar a [mappings.dbpedia.org/server/ontology/classes/](http://mappings.dbpedia.org/server/ontology/classes/).



## Bibliografia

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

**Guarino, N.; Oberle, D.; Staab, S.** (2009). *What Is an Ontology?* [en línia]. [Data de consulta: gener 2020]. Disponible a: <[https://iaoa.org/isc2012/docs/Guarino2009\\_What\\_is\\_an\\_Ontology.pdf](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.

