

1

Representación del conocimiento: las Ontologías

Las ontologías permiten la generación de vocabularios extendidos para describir clases, propiedades y relaciones entre recursos en la web semántica facilitando la creación de modelos de conocimiento estructurados. Aquí se detallan sus características básicas.

JAIME ALBERTO GUZMAN LUNA, Ph.D

Contenido



Definición de Ontología



Modelo RDF



Ontologías con RDFS



Programación RDF y RDFS

2

1



3

Dato, información y conocimiento

1.5

1.5 m



Dato

- Los datos son brutos
- Simplemente existe y no tiene ningún significado más allá de su existencia (en sí mismo)
- puede existir en cualquier forma, utilizable o no

Información

- son datos a los que se les ha dado significado a través de una conexión relacional
- Está contenida en descripciones
- Responde a preguntas que comienzan con palabras como quién, qué, cuándo, dónde y cuántos

$\text{Guepardo} \sqsubseteq \text{Animal} \sqcap \forall \text{maxLength}.\leq 1.5$

Conocimiento

- Información enriquecida con semánticas
- Para representar el conocimiento, se requiere una representación formal del conocimiento: **Las ontologías**

4

2

Las Ontologías: Definición



VISION GENERAL

Rama de la filosofía que se ocupa de la naturaleza y la organización de la realidad, es decir de lo que “existe”.

Etimológicamente:

Onto = “lo que se es”

Logos = Tratado



LAS ONTOLOGÍAS EN LA COMPUTACIÓN

“Una ontología es una especificación formal y explícita de una conceptualización compartida. El término está tomado de la filosofía, donde una ontología es una cuenta sistemática de existencia. Para los sistemas de IA, lo que ‘existe’ es lo que se puede representar”.

Thomas R. Gruber: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition, 5(2):199-220, 1993.

conceptualización: modelo abstracto

(dominio, conceptos relevantes identificados, relación entre los conceptos y la problemática).

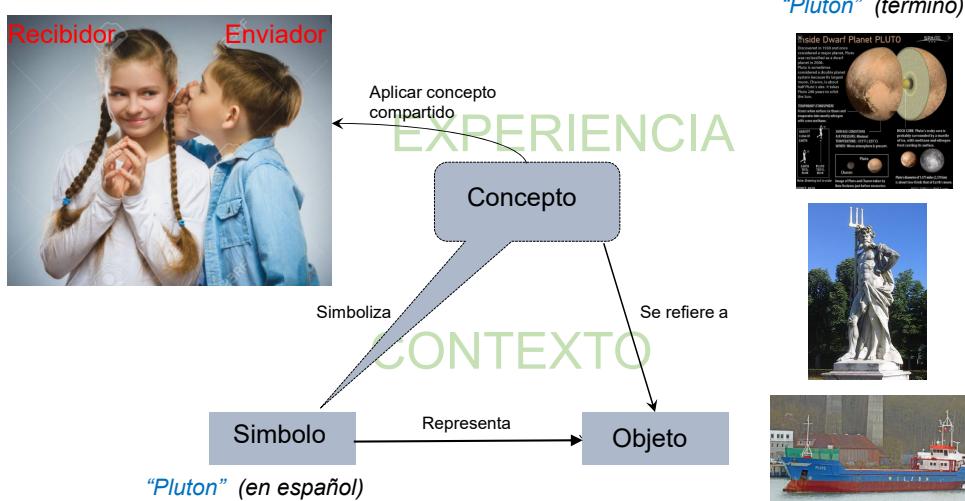
explicativa: se debe definir el significado de todos los conceptos.

formal: entendible por las máquinas

compartida: consenso sobre la ontología

5

Conceptualización

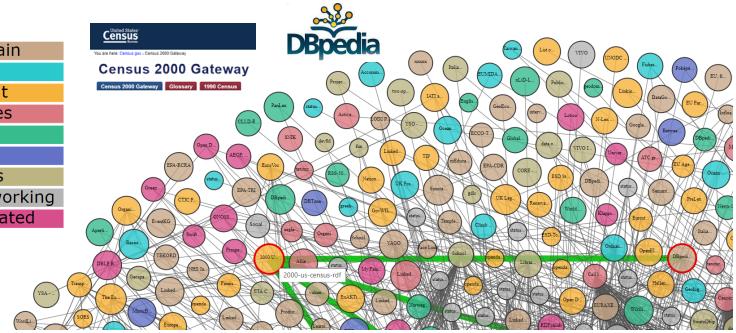


Motivación para el uso de Ontologías

Hendler, Berners-Lee y Miller (2002) ofrecen la siguiente definición de Web semántica:

"La web semántica es una extensión de la actual Web en la que a la información disponible se le otorga un significado bien definido que permita a los computadores y a las personas trabajar en cooperación. Está basada en la idea de proporcionar en la Web datos definidos y enlazados, permitiendo que aplicaciones heterogéneas localicen, integren, razonen y reutilicen la información presente en la Web."

La Web de datos

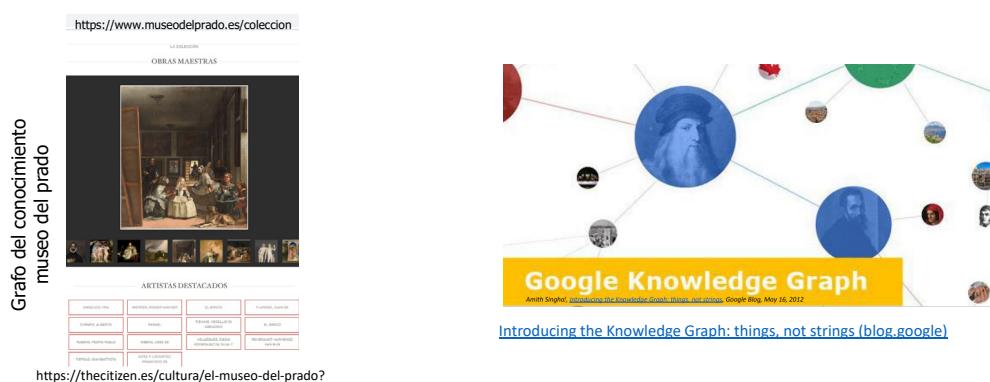


Marzo 2025: 1354 datasets

<https://lod-cloud.net/clouds/lod-cloud.svg>

7

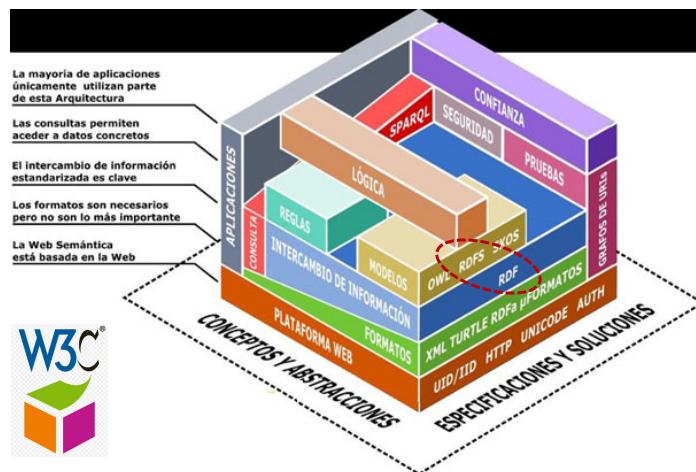
Motivación para el uso de Ontologías



Knowledge Graph	Statements	Entities	
 YAGO Knowledge Graph	120 M	10 M	https://yago-knowledge.org/
 WIKIDATA	610 M	51 M	https://www.wikidata.org/wiki/Wikidata:Main_Page
 DBpedia	1.3 B	6 M	https://es.dbpedia.org/
 GDELT	3.5 B	364 M	https://www.gdeltproject.org/

Modelado de las Ontologías

TECNOLOGÍAS DE LA WEB SEMÁNTICA



INFORMACION

Formato: XML y XMLS

CONOCIMIENTO

Metadatos: RDF

Ontología: RDFS

9

Aplicaciones



Interoperabilidad entre Dispositivos IoT

Estandariza y estructura la información de dispositivos conectados



Reconocimiento y Extracción de Información

Identificar y clasificar términos clave dentro de textos grandes. Útil en el análisis de documentos legales y científicos



Educación Personalizada

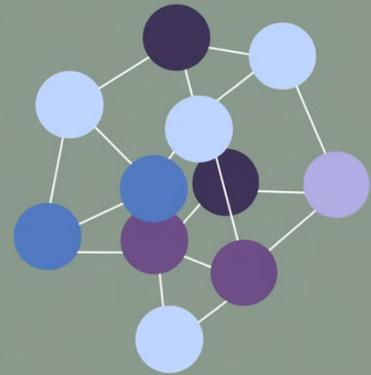
Modelo ontológico del conocimiento que el estudiante necesita adquirir

10

5

Modelado del conocimiento con RDF

Resource Description Framework - RDF



11

¿Qué es RDF?

Resource Description Framework (RDF) es un estándar de la W3C para modelar y conectar datos en la web semántica.



Evolución

De la web tradicional a la web semántica, donde RDF juega un papel fundamental



La Web de Datos

RDF permite describir recursos y sus relaciones de forma que las máquinas puedan entenderlas



Un Modelo de Datos

- Para representar metadatos
- Describe la semántica de manera accesible a las máquinas
- Es independiente del dominio
- Proporciona información inteligentemente
- Permite procesos de computación basados en significado
- Facilita la comunicación entre agentes



Aplicaciones

Estándar de la W3C (<http://www.w3.org/RDF>) para la descripción de recursos en la web.

12

Wikipedia: Concepto Pluto

The screenshot shows the English Wikipedia page for Pluto. On the right side, there is a detailed diagram illustrating the concept of Pluto. The diagram is a 3D cube divided into various colored sections: top face (red) contains 'COMPLIANCE', 'SEGURIDAD', 'PRUEBAS', and 'MANIFESTOS'; left face (blue) contains 'APLICACIONES', 'REGULAS', 'INTERCAMBIO DE INFORMACION', 'FORMATOS', and 'XML TURTLE RDF+RDFa+SPARQL'; bottom face (green) contains 'CONCEPTOS Y ABSTRACCIONES', 'SPECFICACIONES', 'HTTP UNIFORM CODE AUTH', and 'FORMATOS'. A red arrow points from the bottom face of the cube to the 'SPECFICACIONES' section of the Wikipedia page. The page itself has a sidebar with navigation links and a main content area with text and images about Pluto.

<http://en.wikipedia.org/wiki/Pluto>

13

De Wikipedia a Dbpedia: Pluto

The screenshot shows the Dbpedia resource page for Pluto. On the right side, there is a detailed diagram illustrating the concept of Pluto, identical to the one in the Wikipedia screenshot. A red arrow points from the bottom face of the cube to the 'SPECFICACIONES' section of the Dbpedia page. The page includes a summary of Pluto's characteristics, a table of properties and values, and a large amount of detailed text in German and English describing Pluto's discovery, classification, and physical properties.

14

Como nombrar las cosas?



Identificador Universal de Recursos (URI)

Define un esquema extensible para la identificación única a nivel mundial de recursos abstractos o físicos ([RFC 3986](#)).



Un recurso puede ser cualquier objeto con una identidad clara

Ejemplo, páginas web, libros, ubicaciones, personas, relaciones entre objetos, conceptos abstractos, etc.



El concepto de URI ya se ha implementado en varios dominios

La Web (URL), Libros y publicaciones (ISBN, ISSN), Identificador de objeto digital (DOI).



15

Como nombrar las cosas?



16

Aspecto básico del RDF

<http://dbpedia.org/resource/Pluto>

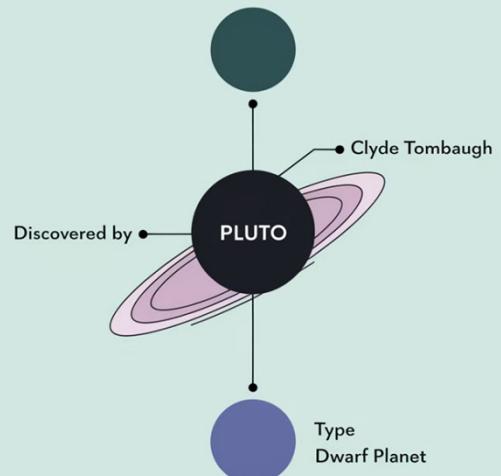
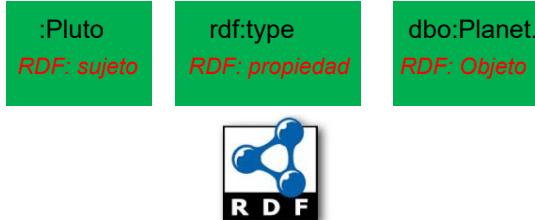
```
:Pluto rdf:type dbo:Planet .  
:Pluto foaf:name "Pluto"@en .  
:Pluto dbo:discoverer :Clyde_Tombaugh .  
:Pluto dbo:discovered "1930-02-18"^^xsd:date .  
:Clyde_Tombaugh rdf:type dbo:Person .  
:Clyde_Tombaugh dbo:birthdate "1906-02-04"^^xsd:date .
```



Plutón

Resource Description Framework

Triplet RDF



17

Resource Description Framework-RDF

- Como puedo representar el hecho: “*Pluto has been discovered in 1930*”?
 - Una manera intuitiva es mediante un **grafo dirigido**



9

18

Terse RDF Triple Language (Turtle)

Turtle es un formato de serialización para representar datos RDF de forma compacta y natural. Su diseño facilita la lectura y escritura de triplets RDF.

Espacios de nombres y qnames

Hace uso de los espacios de nombres y permite definir Nombres cualificados resumidos (qnames) para los nodos:

- Prefijos URI que identifican entidades URIs y se declaran al inicio del documento
- Ejemplo:
`@prefix mfg: <http://www.WorkingOntologist.com/Examples/Chapter3/Manufacturing.rdf#>.`
`@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.`

Una vez definido se puede usar en el documento cada qnames así:

`mfg:Product1 rdf:type mfg:Product .`

Sintaxis compacta con punto y coma (;

Es común usar varias triplets que comparten un sujeto común. Turtle permite una representación compacta haciendo uso del (;) así:

```
mfg:Product1 rdf:type mfg:Product;
mfg:Product_Division "Manufacturing support";
mfg:Product_ID "1";
mfg:Product_Manufacture_Location "Sacramento";
mfg:Product_ModelNo "ZX-3"; #esto es un comentario
mfg:Product_Product_Line "Paper Machine";
mfg:Product_SKU "FB3524";
mfg:Product_Available "23".
```

URI: Identificadores de recursos
Literal: Valores como texto o números
Comentario: Anotaciones precedidas por #

El punto es el fin de una tripla sencilla

Sintaxis con coma (,)

Turtle permite una representación compacta haciendo uso de la (,) cuando la tripla varía solo en el objeto así:

```
mfg:Product1 mfg:Product_Division "Manufacturing support",
"supervition section".
```

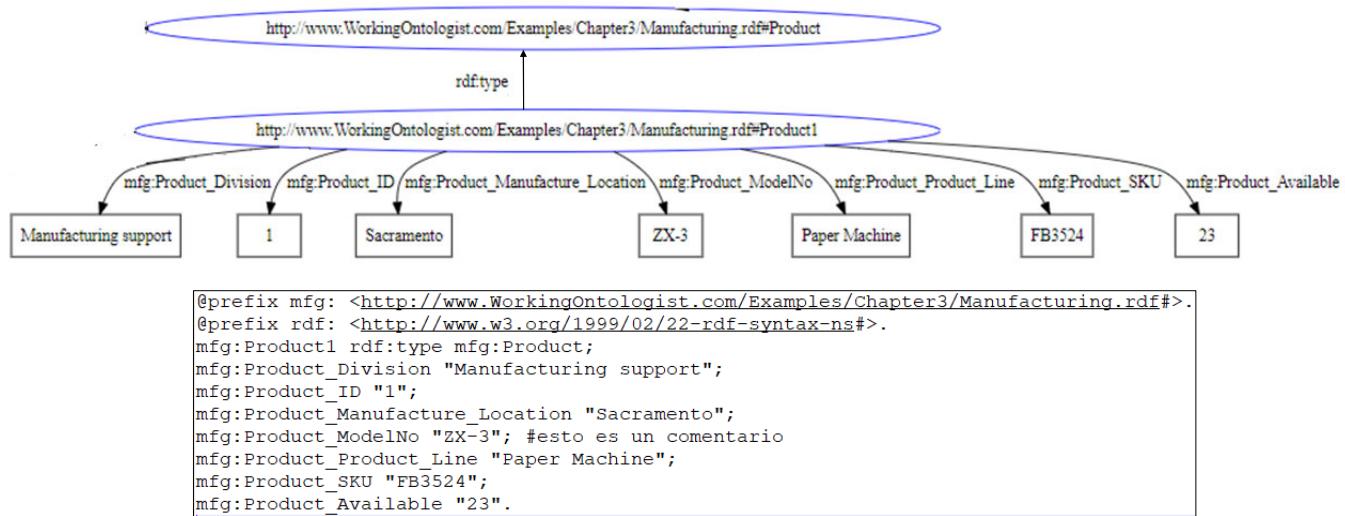
Otros formatos de serialización

- RDF/XML
- JSON-LD
- Otros...

Especificación oficial: <https://www.w3.org/TR/turtle/>

19

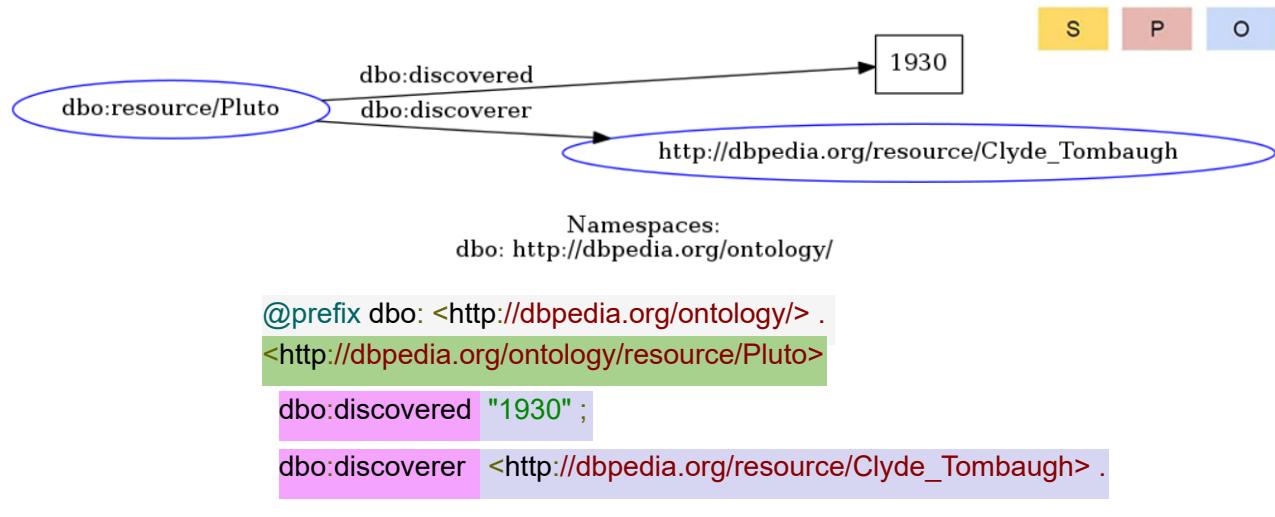
Grafo de ejemplo anterior



20

10

Ejemplo base en Turtle



21

RDF Grapher

RDF Grapher

RDF grapher is a web service for parsing RDF data and visualizing it as a graph.

The service is based on [Redland Raptor](#) and [Graphviz](#).

Supported RDF serialization formats: Turtle, RDF/XML, RDF/JSON, N-Triples, TriG, and N-Quads.

Supported image formats: PNG, SVG, PDF, PS, EPS, GIF, and JPG.

Usage:
http://www.ldf.fi/service/rdf-grapher?rdf=DATA_OR_URI&from=FORMAT&to=FORMAT

GET/POST parameters:

<code>rdf</code>	RDF data or URI
<code>from</code>	input serialization format (ttl, xml, json, nt, trig, ng), default: ttl
<code>to</code>	output image format (png, svg, pdf, ps, eps, gif, jpg), default: png

Examples:
<http://www.ldf.fi/service/rdf-grapher?rdf=http://example.com/s>+<http://example.com/p>+<http://example.com/o>+.&from=ttl&to=png>
<http://www.ldf.fi/service/rdf-grapher?rdf=http://dbpedia.org/resource/Helsinki&from=xml&to=png>

Try the service:

RDF data or URI:
`@prefix dbo: <http://dbpedia.org/ontology/> .`
`<http://dbpedia.org/ontology/resource/Pluto>`
`dbo:discovered "1930" ;`
`dbo:discoverer <http://dbpedia.org/resource/Clyde_Tombaugh> .`
`<http://dbpedia.org/resource/Streator_Illinois>`
`dbo:motto "Quite Surprise in the Prairie".`

From format: To format: Send form as HTTP POST (needed for large RDF data):

Visualize

Namespaces:
dbo: <http://dbpedia.org/ontology/>

<https://www.ldf.fi/service/rdf-grapher>

22

Literales y tipos de datos

Los literales tipados en RDF se definen mediante los tipos de datos establecidos en los estándares.

Literales tipados en RDF

Los **literales tipados** se definen mediante los tipos de datos de [XML Schema](#).
El **espacio de nombre** para los datos tipados: <https://www.w3.org/2001/XMLSchema#>

	Datatype	Value space (informative)
Core types	<code>xsd:string</code>	Character strings (but not all Unicode character strings)
	<code>xsd:boolean</code>	true, false
	<code>xsd:decimal</code>	Arbitrary-precision decimal numbers
	<code>xsd:integer</code>	Arbitrary-size integer numbers
IEEE floating-point numbers	<code>xsd:double</code>	64-bit floating point numbers incl. ±Inf, ±0, NaN
	<code>xsd:float</code>	32-bit floating point numbers incl. ±Inf, ±0, NaN
Time and date	<code>xsd:date</code>	Dates (yyyy-mm-dd) with or without timezone
	<code>xsd:time</code>	Times (hh:mm:ss.sss...) with or without timezone
	<code>xsd:dateTime</code>	Date and time with or without timezone
	<code>xsd:dateTimeStamp</code>	Date and time with required timezone

Tags de lenguaje

Permite definir el lenguaje (natural) del texto
Ejemplo: "Carro"@es, "Car"@en

Fuente: <https://www.w3.org/TR/rdf11-concepts/>

Ejemplo de implementación

Ejemplo en Turtle

```
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@base <http://dbpedia.org/resource/> .

<Pluto> dbo:meanRadius "161.00"^^xsd:double ;
    rdfs:label "Pluto"@en .
```

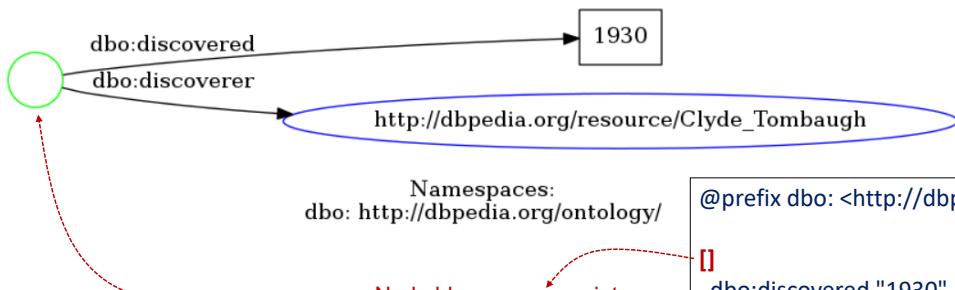
23

Nodos Blancos

Definición

Denotan la existencia de un individuo con atributos específicos, pero sin proporcionar una identificación o referencia

Ejemplo 1 (en DBpedia)



```
@prefix dbo: <http://dbpedia.org/ontology/> .

[] dbo:discovered "1930" ;
    dbo:discoverer
    <http://dbpedia.org/resource/Clyde\_Tombaugh> .
```

24

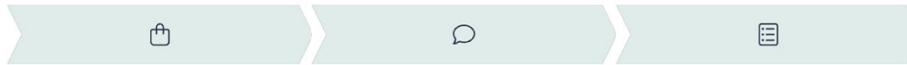
12

Listas en RDF (1)

1. CONTENEDOR

Lista abierta. Es posible extenderla (nuevas entradas)

Existen tres tipos de contenedores en RDF



Bag

Conjunto desordenado

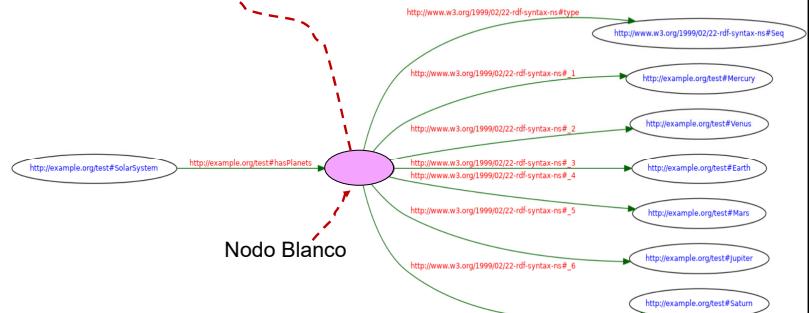
Seq

Conjunto ordenado

Alt

Alternativas

```
@prefix rdf: .
@prefix ex: .
ex:SolarSystem ex:hasPlanets [
  a rdf:Seq;
  rdf:_1 ex:Mercury ;
  rdf:_2 ex:Venus ;
  rdf:_3 ex:Earth ;
  rdf:_4 ex:Mars ;
  rdf:_5 ex:Jupiter ;
  rdf:_6 ex:Saturn
].
```



El ejemplo muestra un contenedor de tipo Seq (secuencia) con un nodo blanco.

25

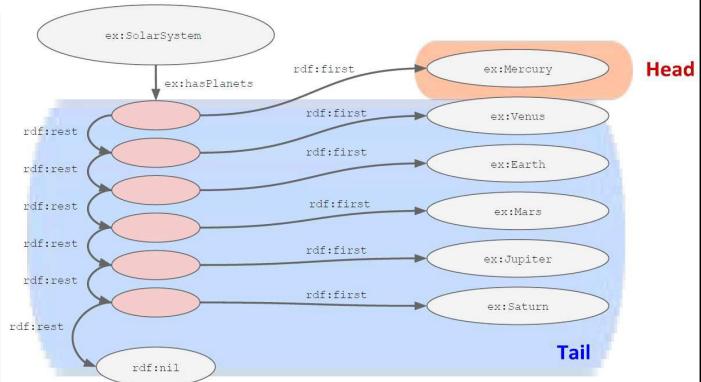
Listas en RDF (2)

2. COLECCIONES

Lista cerrada. No es posible extenderlo

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ex: <http://example.org/test#> .
ex:SolarSystem ex:hasPlanets [
  rdf:first ex:Mercury ; rdf:rest [
    rdf:first ex:Venus ; rdf:rest [
      rdf:first ex:Earth ; rdf:rest [
        rdf:first ex:Mars ; rdf:rest [
          rdf:first ex:Jupiter ; rdf:rest [
            rdf:first ex:Saturn ;
            rdf:rest rdf:nil
          ]
        ]
      ]
    ]
  ]
].
```

Formato extendido



```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix ex: <http://example.org/test#> .
ex:SolarSystem ex:hasPlanets (
  ex:Mercury ex:Venus ex:Earth ex:Mars ex:Jupiter ex:Saturn
).
```

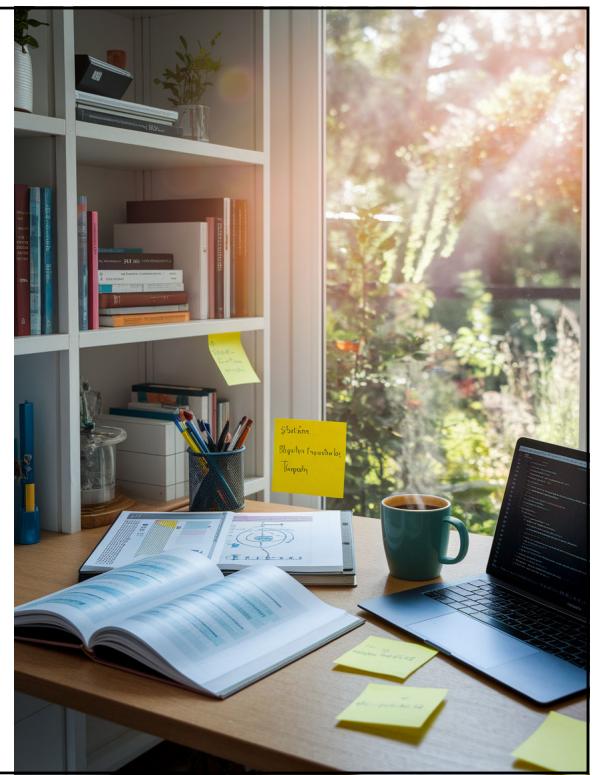
Formato compacto

26

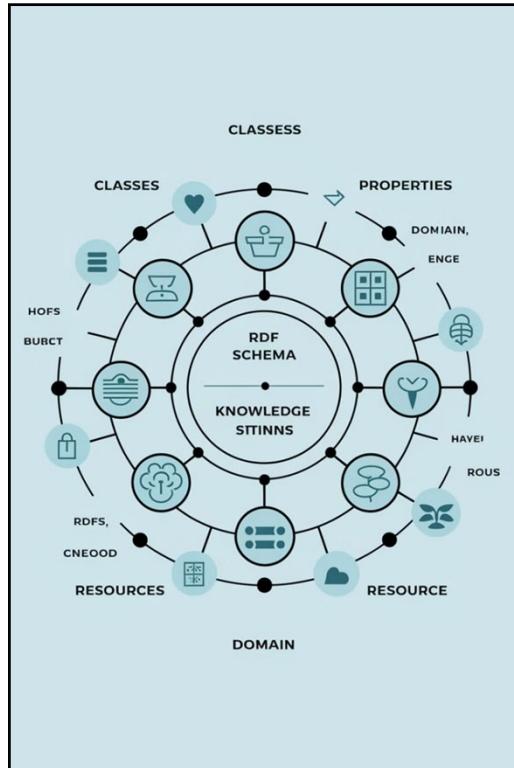
13

Referencias

- Presentación de la sección basada en:
 - RDF 1.1 Turtle
 - Terse RDF Triple Language
 - W3C Recommendation 25 February 2014
 - <https://www.w3.org/TR/turtle/>
 - Material del curso Knowledge Engineering with semantic Web Technologies. OpenHPI. Dr. Harald Sack.



27



Ontologías con RDFS

RDF Schema

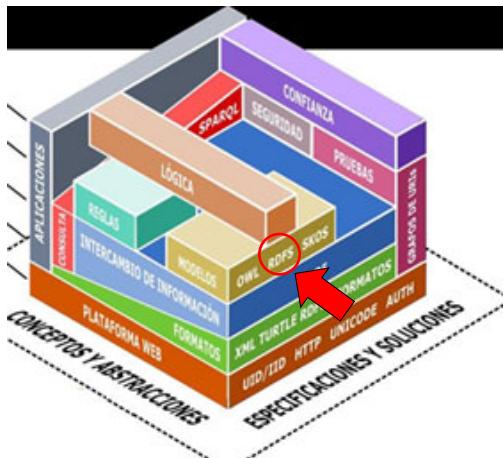
JAIME ALBERTO GUZMAN LUNA, Ph.D

14

28

Modelado de ontologías con RDFS

<http://dbpedia.org/resource/Pluto>



```

dbo:Planet rdf:type rdfs:Class .
dbo:Planet rdfs:subClassOf dbo:CelestialBody .
dbo:discovered rdf:type rdf:Property .
dbo:discovered rdfs:domain rdfs:Resource .
dbo:discovered rdfs:range xsd:date .
dbo:discover rdf:type rdf:Property .
dbo:discover rdfs:domain dbo:Planet .
dbo:discover rdfs:range dbo:Person .
  
```



RDF Schema-RDFS:

Llamado “RDF Vocabulary Description Language”



<https://www.w3.org/TR/rdf-schema/>

29

RDF Schema permite....

Definición e instanciación de Clases

```

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

```

:Planet rdf:type rdfs:Class . Definición de clases
:Earth rdf:type :Planet . Instanciación de clases en RDF
  
```

Definición y restricción de Propiedades

Definición de propiedades
Restricción de Propiedades en domain y range

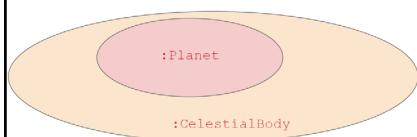
```

:CelestialBody rdf:type rdfs:Class .
:satelliteOf rdf:type rdf:Property .
:satelliteOf rdfs:domain :CelestialBody .
:satelliteOf rdfs:range :CelestialBody .
  
```

Relaciones de Jerarquía

```

:Planet rdfs:subClassOf :CelestialBody . Subclases y Superclases
:artificialSatelliteOf rdfs:subPropertyOf Subpropiedades y Superpropiedades
:satelliteOf .
  
```



30

15

RDF schema

- Todo en el modelo RDF es un **Resource**
 - `rdfs:Class rdf:type rdfs:Resource .`
 - `rdf:Property rdf:type rdfs:Resource .`
 - `rdfs:Literal rdf:type rdfs:Resource .` Clase de valores literales como cadenas y números enteros
 - `rdf:XMLLiteral rdf:type rdfs:Resource .`
 - `rdfs:Datatype rdf:type rdfs:Resource .` Corresponde a la clase datatype del modelo RDF: Cada instancia de `rdfs:Datatype` es una subclase de `rdfs:Literal`
 - `rdfs:Container rdf:type rdfs:Resource .` Es una superclase de las clases RDF Container: `rdf:Bag`, `rdf:Seq`, `rdf:Alt`.
 - `rdfs:ContainerMembershipProperty rdf:type rdfs:Resource .` Tiene como instancias las propiedades `rdf:_1`, `rdf:_2`, `rdf:_3...` que se utilizan para indicar que un recurso es miembro de un contenedor.

31

Representación de otras propiedades

- `rdfs:seeAlso`
 - Define una relación de un recurso a otro, el cual lo explica
- `rdfs:isDefinedBy`
 - subpropiedad de `rdfs:seeAlso`, define la relación de un recurso a su definición
- `rdfs:comment`
 - Comentario, usualmente como texto
- `rdfs:label`
 - Etiqueta de lectura de un recurso
- `rdfs:member`
 - Super-propiedad de las propiedades de los miembros del contenedor (e.g. `rdf:_1`, ...)

32

16

Ejemplo de RDFS

Definición de clases

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

```
:Planet rdf:type rdfs:Class ;
         rdfs:subClassOf :CelestialBody .
:Satellite rdf:type rdfs:Class ;
         rdfs:subClassOf :CelestialBody .
:ArtificialSatellite rdf:type rdfs:Class ;
         rdfs:subClassOf :Satellite .
```

Definición de propiedades

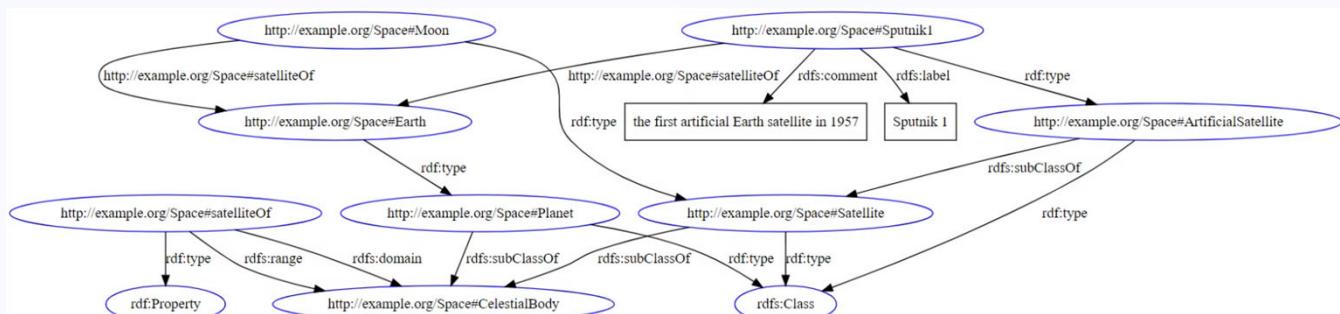
```
:satelliteOf rdf:type rdf:Property .
:satelliteOf rdfs:domain :CelestialBody .
:satelliteOf rdfs:range :CelestialBody .
```

Definición de instancias

```
:Earth rdf:type :Planet .
:Moon rdf:type :Satellite ;
       :satelliteOf :Earth .
:Sputnik1 rdf:type :ArtificialSatellite ;
           :satelliteOf :Earth ;
           rdfs:label "Sputnik 1"@en ;
           rdfs:comment "the first artificial Earth satellite in 1957" .
```

33

Ejemplo de RDFS

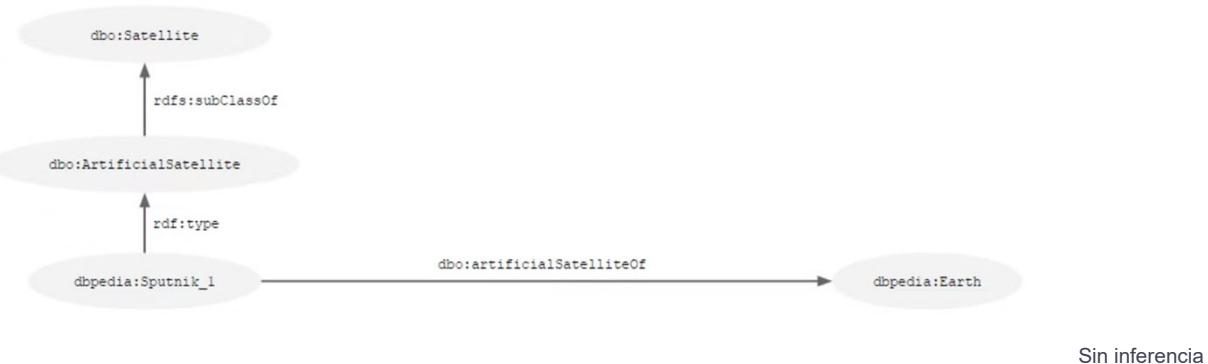


34

17

Inferencias lógicas con RDFS

CASO 1: Nuevos hechos desde una jerarquía de clases



35

Inferencias lógicas con RDFS

CASO 1: Nuevos hechos desde una jerarquía de clases



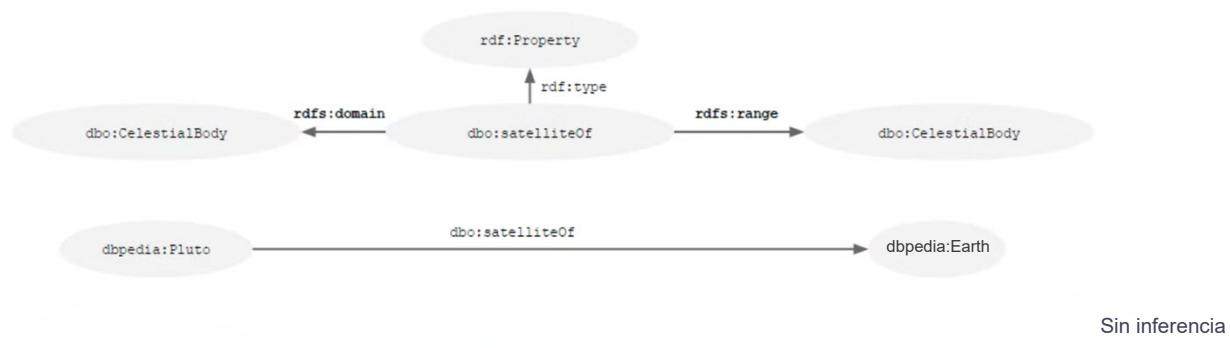
Con inferencia

36

18

Inferencias lógicas con RDFS

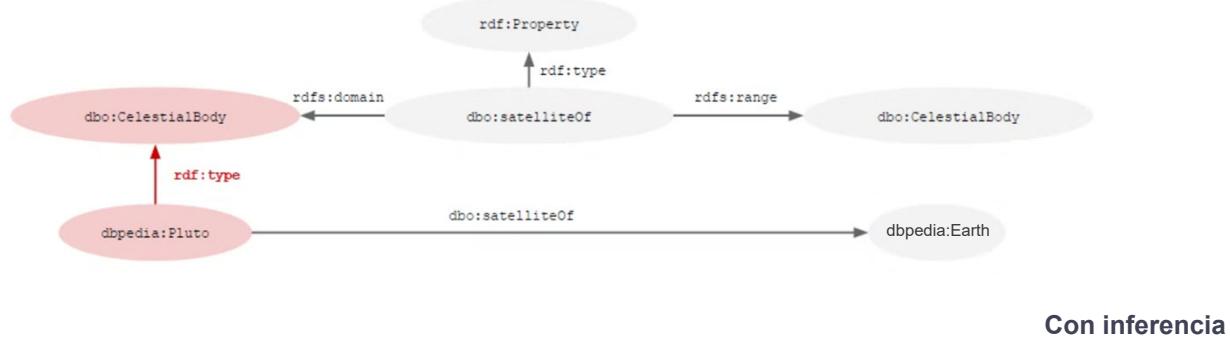
CASO 2: Miembros de una clase desde el dominio de sus propiedades



37

Inferencias lógicas con RDFS

CASO 2: Miembros de una clase desde el dominio de sus propiedades



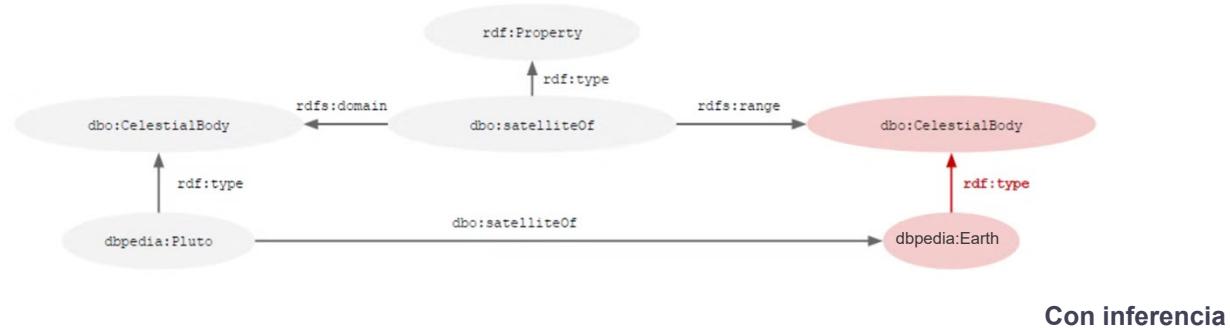
38

19

Inferencias lógicas con RDFS



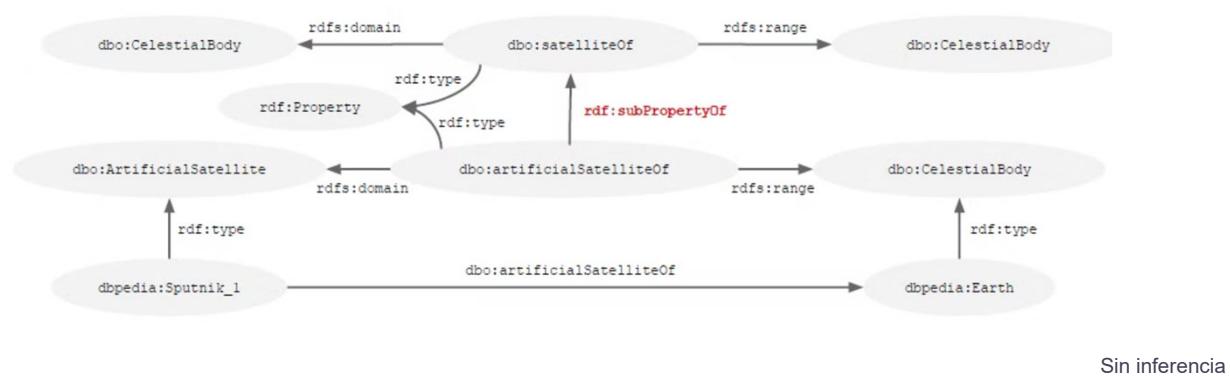
CASO 2: Miembros de una clase desde el dominio de sus propiedades



39

Inferencias lógicas con RDFS

CASO 3: Nuevos hechos desde la relación subproperty



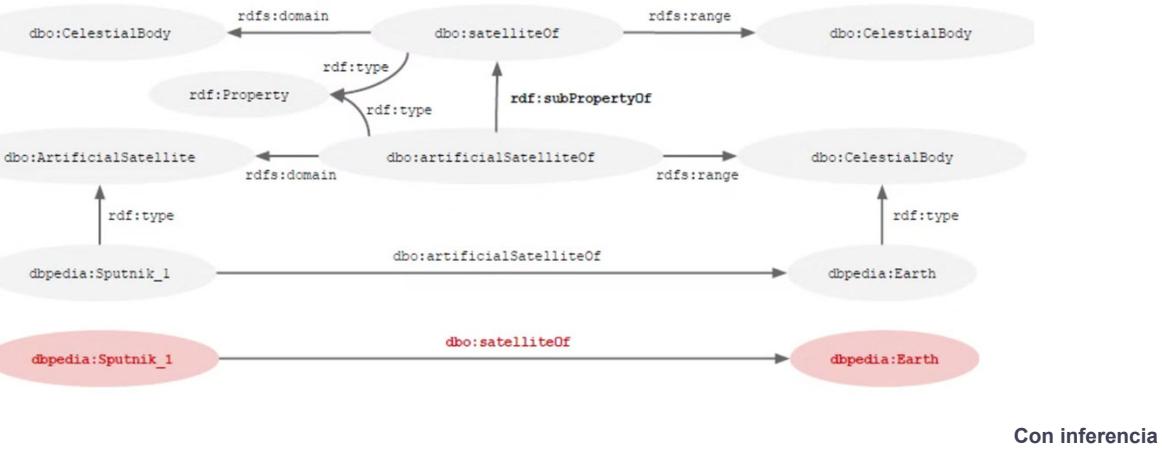
40

20

Inferencias lógicas con RDFS



CASO 3: Nuevos hechos desde la relación subproperty



41

Vocabularios



Vocabularios

- Dublin Core <http://dublincore.org/>
- FOAF <http://www.foaf-project.org/> <http://xmlns.com/foaf/spec/>
- SKOS <https://www.w3.org/TR/skos-primer/>

• FOAF - Friend of a Friend

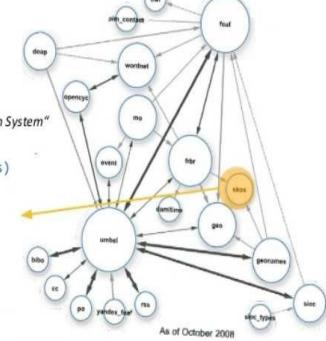
```

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
<#me>
  a foaf:Person ;
  foaf:name "Harald Sack" ;
  foaf:mailbox <mailto:harald.sack@hpi.de> ;
  foaf:homepage <https://hpi.de/meinei/lehrstuhl/team-fotos/senior-researcher/sack.html> ;
  foaf:depiction <https://hpi.de/fileadmin/_migrated/pics/harald_min.jpg> ;
  foaf:interest <http://linkeddata.org> ;
  foaf:knows [
    a foaf:Person ;
    foaf:name "Magnus Knuth"
  ].

```

• SKOS

- „Simple Knowledge Organization System“
- based on RDF and RDFS
- skos:Concept (classes)
- skos:narrower
- skos:broader
- skos:related
- skos:exactMatch, skos:narrowMatch, skos:broadMatch, skos:relatedMatch



42



Algunas limitaciones de RDF Schema

No se puede decir que dos clases son disjuntas

'Hombre' y 'Mujer' son clases disjuntas

No podemos definir nuevas clases como unión, diferencia, ... de otras existentes

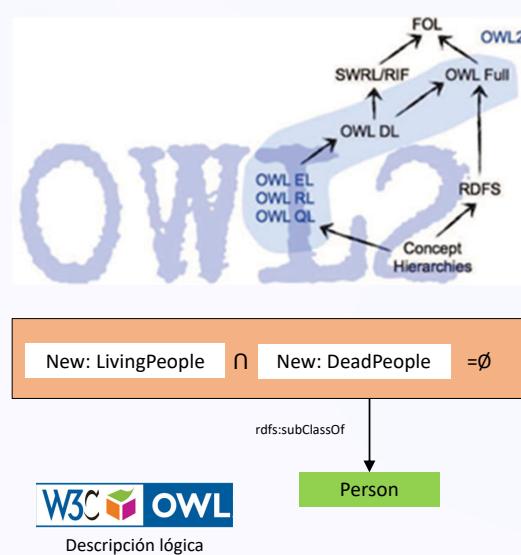
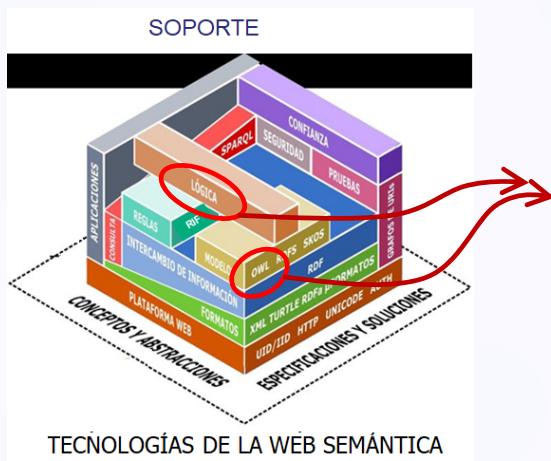
'Persona' es la clase resultante de hacer la unión de 'Hombre' y 'Mujer'

No podemos expresar restricciones de cardinalidad

Una persona sólo puede tener dos progenitores

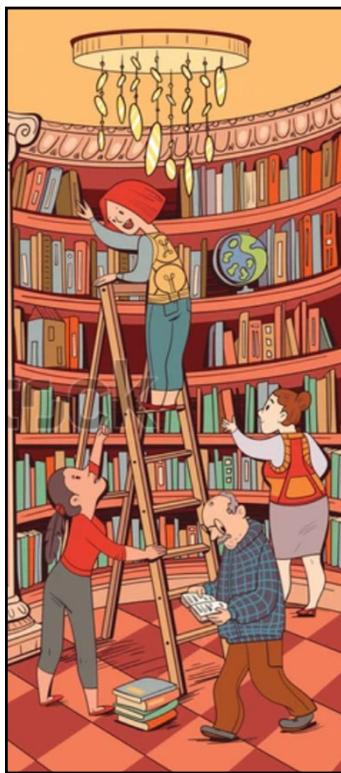
43

Temas complementarios



OWL Web Ontology Language Guide. <http://www.w3.org/TR/owl-guide>

44



Referencias

- Presentación basada en:
 - RDF Schema 1.1
 - W3C Recommendation 25 February 2014
 - <https://www.w3.org/TR/rdf-schema/>
 - Material del curso Knowledge Engineering with semantic Web Technologies. OpenHPI. Dr. HaraldSack.

45

Programación de RDF y RDFS

RDFLib y OWL-RL

```
from .unit import Unit
from .enemy import Enemy
from .item import Item

class Core:
    def __init__(self):
        self.items = []
        self.enemies = []
        self.units = []

    def add_item(self, item):
        self.items.append(item)

    def add_enemy(self, enemy):
        self.enemies.append(enemy)

    def add_unit(self, unit):
        self.units.append(unit)

    def remove_item(self, item):
        self.items.remove(item)

    def remove_enemy(self, enemy):
        self.enemies.remove(enemy)

    def remove_unit(self, unit):
        self.units.remove(unit)

    def get_items(self):
        return self.items

    def get_enemies(self):
        return self.enemies

    def get_units(self):
        return self.units
```

23

46

APIS de programación para RDF y RDFS

Lo importante es hacer uso de una librería adecuada

JAVA

- [Jena](#) – Maneja RDF, SPARQL, Ontologías OWL y permite el razonamiento
- [RDF4J](#)
- [Commons-rdf](#)

Python

- [RDFLib](#) - Pythonic RDF API.

C

- [Librdf](#) – Es una API de RDF y repositorio de tripletas.
- [Raptor](#) – Librería llamada Redland Raptor RDF

JavaScript

- [RDFJS](#) – Organización Github que mantiene librerías de código abierto actualizadas de JavaScript para el estándar RDF

C#

- [dotNetRDF](#)
- [RDFSharp](#)

Para una lista mas detallada ver Awesome Semantic Web

47

RDFLib



- RDFLib fue creado por Daniel Krech.
- Todos los componentes únicos de un modelo RDF se han definido como objetos Python en RDFLib:

RDFLib.URIRef

Un recurso con una URI

RDFLib.Literal

Un literal

TripleStore

Almacenamiento de tripleta en memoria

RDFLib.Graph

Un conjunto de tripletas

RDFLib.BNode

Un recurso sin URI

RDFLib.Namespace

Administrar un espacio de nombres

RDFLib.constants

contiene definiciones para las propiedades RDF como tipo y valor

- Ejemplos de RDFLib: Clase_6_RDFLIB.ipynb

48

24



Librería OWL-RL

- **OWL-RL (Web Ontology Language - Rule Language)**
 - Implementa razonamiento con ontologías mediante sistemas basados en reglas. Incluye todas las características de RDFS.
- **Reglas de Inferencia Claves**
 - **Subsunción de clases:** Aplicación transitiva de `rdfs:subClassOf` para inferir jerarquías completas.
 - **Inferencia de tipos:** Deducir tipos implícitos a través de `rdf:type` y jerarquías de clases.
 - **Propagación de propiedades:** Maneja la herencia de propiedades mediante `rdfs:subPropertyOf`.
- **owlrl.DeductiveClosure()**
 - Su propósito principal es aplicar un conjunto de reglas de inferencia sobre un grafo RDF para derivar todas
 - Funcionalidad principal
 - **Aplicar reglas de inferencia:** Ejecuta sistemáticamente un conjunto de reglas de inferencia sobre un grafo RDF hasta alcanzar un punto fijo (donde no se pueden derivar más tripletes nuevas).
 - **Materializar inferencias:** Añade explícitamente al grafo todas las triplets que pueden deducirse mediante
- **Ejemplo**

Aplicar razonamiento RDFS para expandir el grafo con inferencias

```
owlrl.DeductiveClosure(owlrl.RDFS_Semantics, axiomatic_triples=True, datatype_axioms=False).expand(g)
```

<https://pypi.org/project/owlrl/>

<https://owl-rl.readthedocs.io/en/latest/stubs/owlrl.DeductiveClosure.html>

49



Razonando RDFS con OWL-RL

- **closure_class**
 - Tipo: Clase
 - Obligatorio: Sí
 - Este parámetro determina qué conjunto de reglas de inferencia se aplicarán al grafo RDF. Representa la semántica o el perfil de razonamiento a utilizar.
 - Valores comunes:
 - **RDFS_Semantics:** Aplica las reglas de inferencia del estándar RDFS.
 - **OWLRL_Semantics:** Implementa el perfil OWL 2 RL completo.
 - **OWLRL_Extension:** Extiende OWL 2 RL con reglas adicionales.
- **axiomatic_triples=False**
 - Tipo: Booleano
 - Por defecto: False
 - Determina si se deben añadir triplets axiomáticas al grafo como parte del proceso de razonamiento.
 - Las triplets axiomáticas son afirmaciones básicas que forman parte de la definición de RDFS/OWL, como:
 - Definiciones de clases básicas (`rdfs:Class`, `rdfs:Resource`)
 - Definiciones de propiedades del sistema (`rdfs:subClassOf`, `rdf:type`)
 - Relaciones entre conceptos fundamentales
 - **Cuando se establece en True, estas triplets se añaden al grafo,** permitiendo inferencias más completas, pero también aumentando considerablemente el tamaño del grafo resultante.
- **datatype_axioms=False**
 - Tipo: Booleano
 - **Por defecto: False**
 - Controla si se deben incluir axiomas relacionados con los tipos de datos XSD estándar.
 - Cuando está activo, añade:
 - Definiciones de los tipos de datos básicos (`xsd:string`, `xsd:integer`, etc.)
 - Relaciones jerárquicas entre tipos de datos
 - Propiedades específicas de los tipos de datos
 - Este parámetro es particularmente útil cuando se trabaja con ontologías que hacen un uso intensivo de tipos de datos y restricciones basadas en ellos.

<https://owl-rl.readthedocs.io/en/latest/stubs/owlrl.DeductiveClosure.html>

50

25