



# Representación del conocimiento y la web semántica

The Google logo is displayed in its characteristic multi-colored font (blue, red, yellow, blue, green, red) on a white rectangular background. The logo is centered in the upper half of the slide.

Google

Dime: ¿Cuál es el salón del posgrado de la UNAM donde se imparten clases de historia del arte?



¿Cuál es el salón del posgrado de la UNAM donde se imparten clases de r  



 Todos  Imágenes  Noticias  Maps  Videos  Más  Preferencias  Herramientas

Cerca de 155,000 resultados (0.76 segundos)

## Maestría en Dirección de Arte | Matrícula Abierta, 100% Online

Anuncio [www.esdesignbarcelona.com/maestrias/dir\\_arte](http://www.esdesignbarcelona.com/maestrias/dir_arte) ▼

La Escuela Superior de Diseño con Metodología 100% Online y flexible. Apúntate! Profesores 100% en activo y constante evolución con el mercado. Profesores 100% en activo. Online / Distancia. Seguimiento Personalizado. Campus Virtual 24h/7d. Doble Titulación.  
Máster Diseño Interiores · Máster en Diseño Gráfico · Máster en Diseño Web

## Coordinación de Estudios de Posgrado | UNAM |

[www.posgrado.unam.mx](http://www.posgrado.unam.mx) ▼

Conoce nuestros Programas de **Posgrado** ... Eventos Coord. de Estudios de **Posgrado**. Área 1. Área 2. Área 3 ... Estadísticas del **Posgrado UNAM**. Alumnos de ...

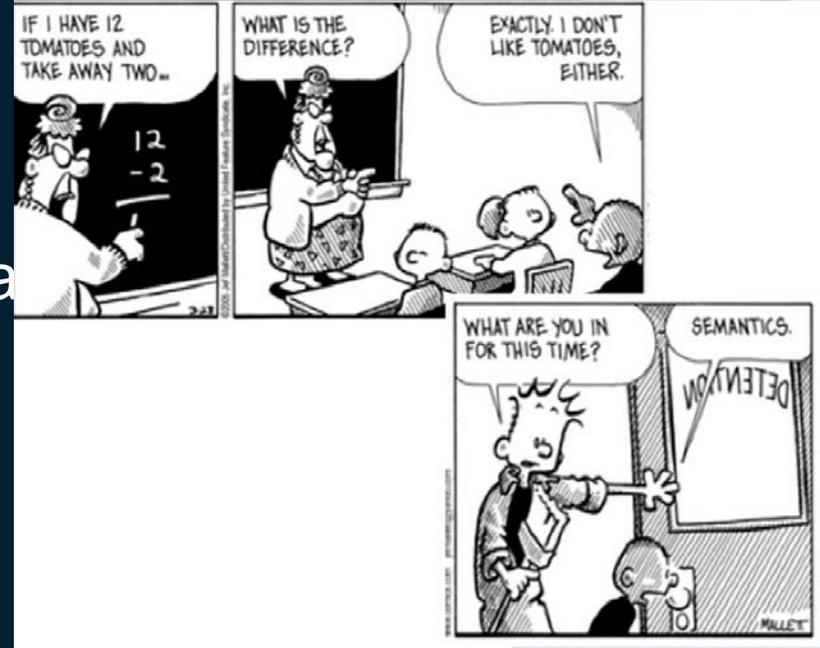
Falta(n): salón

[Ofertas de Posgrado](#) · [Posgrado](#) · [Becas - unam](#) · [Historia](#)

## Posgrado en Historia del Arte – UNAM

# ¿Por qué sucede esto?

- › Información codificada en HTML carente de semántica



The Google logo is displayed in its characteristic multi-colored font (blue, red, yellow, blue, green, red) on a white rectangular background. The logo is centered in the upper half of the slide.

Google

Dime: ¿Cuál es la capital de México?



¿Cuál es la capital de México?



[Todos](#)

[Maps](#)

[Imágenes](#)

[Noticias](#)

[Videos](#)

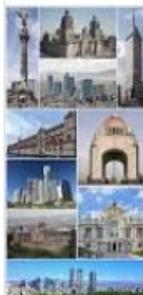
[Más](#)

[Preferencias](#)

[Herramientas](#)

Cerca de 42,800,000 resultados (0.84 segundos)

México / Capital



# Ciudad de México

También se buscó

[Ver 10 más](#)

# Google Knowledge Graph

- › El Knowledge Graph es una **base de conocimiento estructurada** usada por google que recolecta una amplia gama de recursos para mejorar sus búsquedas

# ¿Y qué tiene que ver Google Knowledge Graph con la web semántica?

- › Google Knowledge Graph usa los recursos de la web semántica para ofrecer mejores resultados en sus búsquedas
- › Recursos como DBpedia, Wikipedia y Freebase

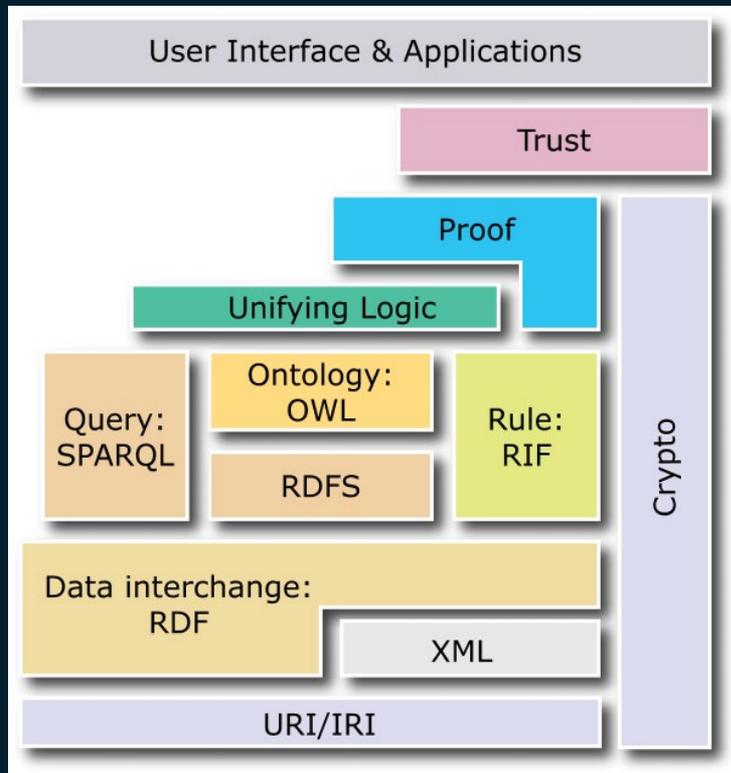


# Web semántica

- › Es una extensión de la web actual para dotarla de herramientas que permitan el uso de **datos en una forma estructurada/enlazada** (linked data).



# Arquitectura del web semántico





# ¿Cómo represento el hecho siguiente?

- Plutón fue descubierto en 1930

```
<discovered>
  <discovery>Pluto</discovery>
  <year>1930</year>
</discovered>
```

```
{
  "discovered": {
    "discovery": "Pluto",
    "year": "1930"
  }
}
```

```
<planet name = "Pluto">
  <yearDiscovered>1930</yearDiscovered>
</planet>
```

```
<planet name = "Pluto" discovered="1930" />
```

# Representación del conocimiento

```
[
class(top,none,[],[],[]),

class(entity, top, [], [], []),

class(human, entity, [], [], [
  [id => charly, [num_id=>11, gender=> men, oldest=>1, youngest=>12, smallest=>1, biggest=>12, highest=>12, tallest=>12, heaviest=>12, lightest=>1],[]],
  [id => second, [num_id=>32, oldest=>120, youngest=>10, smallest=>120, biggest=>120, highest=>120, tallest=>120, heaviest=>120, lightest=>10], []
]),

class(robot, entity, [], [], [
  [id => golem, [in=>exit, attending_to=>nobody, has=>nothing],[]
  ]),

class(place, entity, [], [], []),

%Rooms
class(room, place, [], [], [
  [id => hall, [name=>hall, doors=>1, smallest=>1, biggest=>5, main_point=>bookcase, object_path=>[bookcase], human_path=>[hp1,hp2]],[]],
  [id => bedroom, [name=>bedroom, doors=>1, smallest=>4, biggest=>2, main_point=>bed, object_path=>[bed,bedroomtable], human_path=>[bed,bp1]],[]],
  [id => kitchen, [name=>kitchen, doors=>1, smallest=>2, biggest=>4, main_point=>fridge, object_path=>[fridge,stove], human_path=>[kp1,kp2]],[]
  ]),

%Points
class(points, place, [coordinate=>[0.0,0.0,0.0]], [], [
  [id => point1,[coordinate=>[0.5,0.0,0.0]],[]
  ]),

%Locations
class(location, place, [height=>135.00], [], []),
```

- [Link a la Demostración en la cual Golem-III realiza la estrategia deliberativa del ciclo de inferencia cotidiana](#)

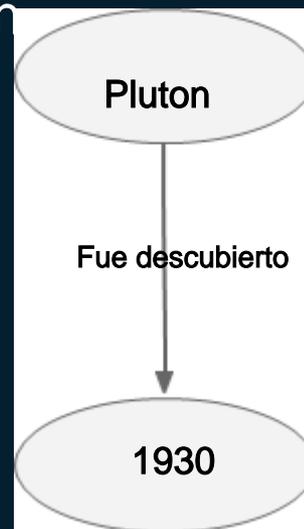
# Resource Definition Framework

## RDF

Representación del conocimiento utilizado por la web semántica para describir los recursos que se encuentran disponibles.

Tripletas:

*Sujeto -> Predicado/Propiedad > Objeto*



SUJETO

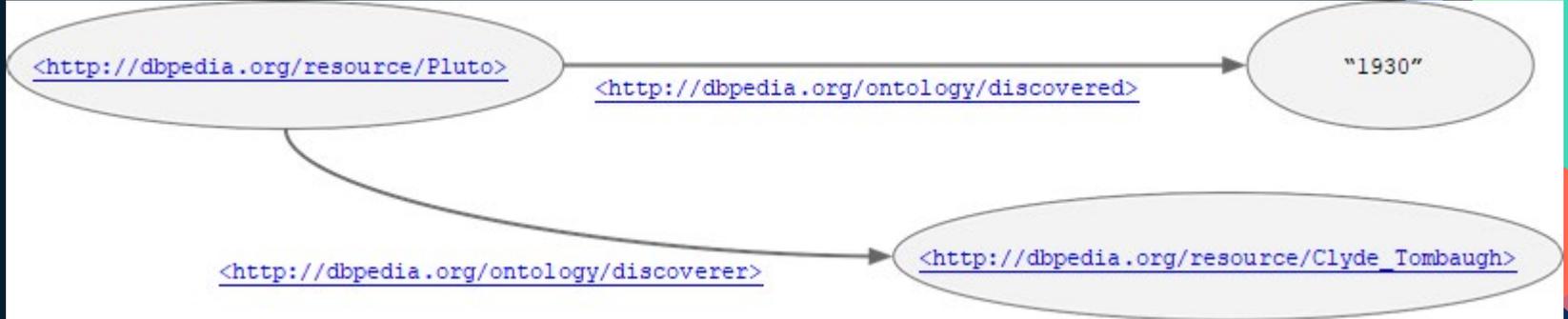
PREDICADO

OBJETO

- Los RFD Statements son tripletas que tienen la siguiente estructura:
  - Sujeto **URI** + Propiedad/Predicado **URI** + Objeto/Valor **URI o Literal**



- URIs y Literales
  - Las URIs referencian recursos únicamente.
  - Las literales describen valores que no tienen una existencia como tal en alguna URI.



- Las literales (typed literals) son expresadas via XMLSchema datatypes
  - El namespace de los literales es:
    - <http://www.w3.org/2001/XMLSchema#>
- Ejemplos:
  - ["Semantics"^^<http://www.w3.org/2001/XMLSchema#string>](http://www.w3.org/2001/XMLSchema#string)  
["1161.00"^^<http://www.w3.org/2001/XMLSchema#float>](http://www.w3.org/2001/XMLSchema#float) ["2015-08-02"^^<http://www.w3.org/2001/XMLSchema#date>](http://www.w3.org/2001/XMLSchema#date)
- También se pueden incluir tags de lenguaje. Ejemplo:
  - "Semantica"@es , "Semantics"@en
- <http://www.w3.org/TR/2013/WD-rdf11-concepts-20130115/#xsd-datatypes>

- La notación de los RDF Statemens en formato de triplas:
  - Los URIs deben ir entre brackets.
  - Las literales deben ir entre comillas.
  - Los enunciados deben terminar con un punto.

```
<http://dbpedia.org/resource/Pluto> <http://dbpedia.org/ontology/discovered> "1930" .
<http://dbpedia.org/resource/Pluto> <http://dbpedia.org/ontology/discoverer>
<http://dbpedia.org/resource/Clyde Tombaugh> .
```

- La notación de los RDF Statemens en formato de xml:

S
P
O

```
<?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ns0="http://dbpedia.org/ontology/">
  <rdf:Description rdf:about="http://dbpedia.org/resource/Pluto">
    <ns0:discovered>1930</ns0:discovered>
    <ns0:discoverer rdf:resource="http://dbpedia.org/resource/Clyde Tombaugh"/>
  </rdf:Description>
</rdf:RDF>
```

- La notación de los RDF Statements en formato de Json:

S

P

O

```
{
  "http://dbpedia.org/resource/Pluto": {
    "http://dbpedia.org/ontology/discovered": [ { "value": "1930",
      "type": "literal" } ]
  } {
    "http://dbpedia.org/ontology/discoverer": [ { "value": "http://dbpedia.org/resource/Clyde_Tombaugh",
      "type": "uri" } ]
  }
}
```

- La notación de los RDF Statemens en formato Turtle (Terse RDF Triple Language):



- starting with N-Triples

```
<http://dbpedia.org/resource/Pluto> <http://dbpedia.org/ontology/discovered> "1930" .
<http://dbpedia.org/resource/Pluto> <http://dbpedia.org/ontology/discovered>
<http://dbpedia.org/resource/Clyde_Tombaugh> .
```

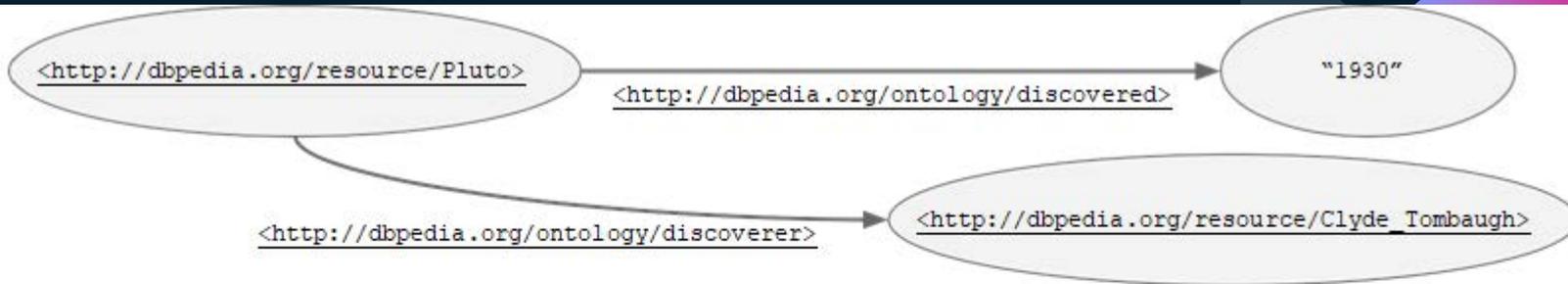
```
@prefix dbo: <http://dbpedia.org/ontology/> .
@base <http://dbpedia.org/resource/> .
```

```
<Pluto> dbo:discovered "1930" .
```

```
<Pluto> dbo:discoverer <Clyde_Tombaugh> .
```

**@prefix directive** associates prefix-label with URI

**@base directive** provides URI to complement all relative URIs

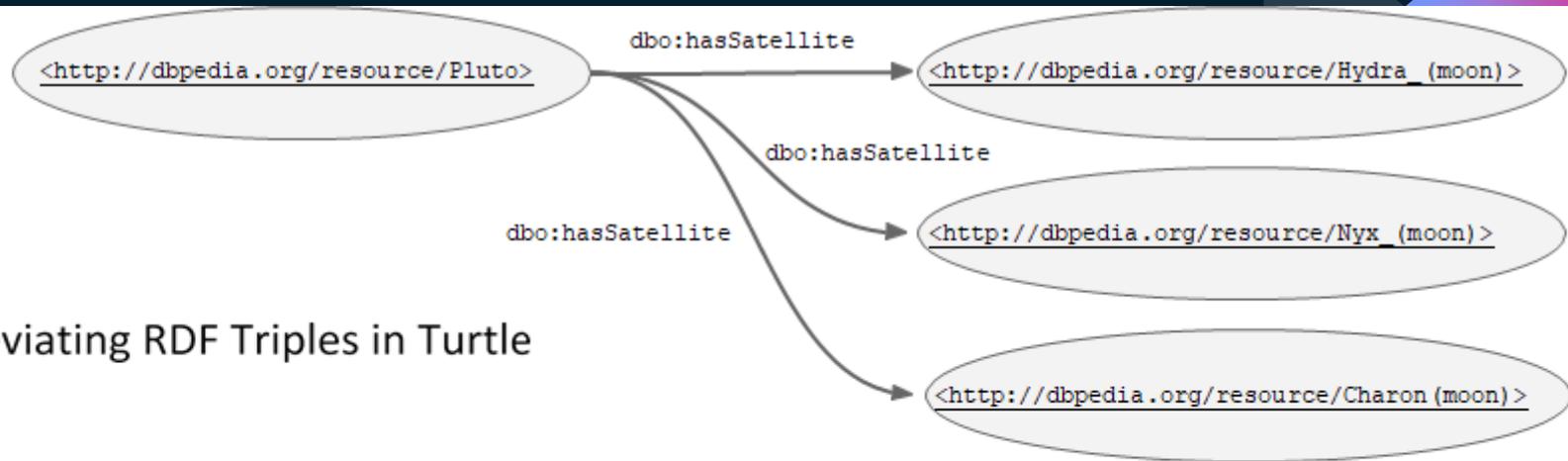


- Abbreviating RDF Triples in Turtle

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

<Pluto> dbo:discovered "1930" ;
        dbo:discoverer <Clyde_Tombaugh> .
```

**semicolon** indicates that subsequent triples have the same subject (**predicate list**)



- Abbreviating RDF Triples in Turtle

```

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

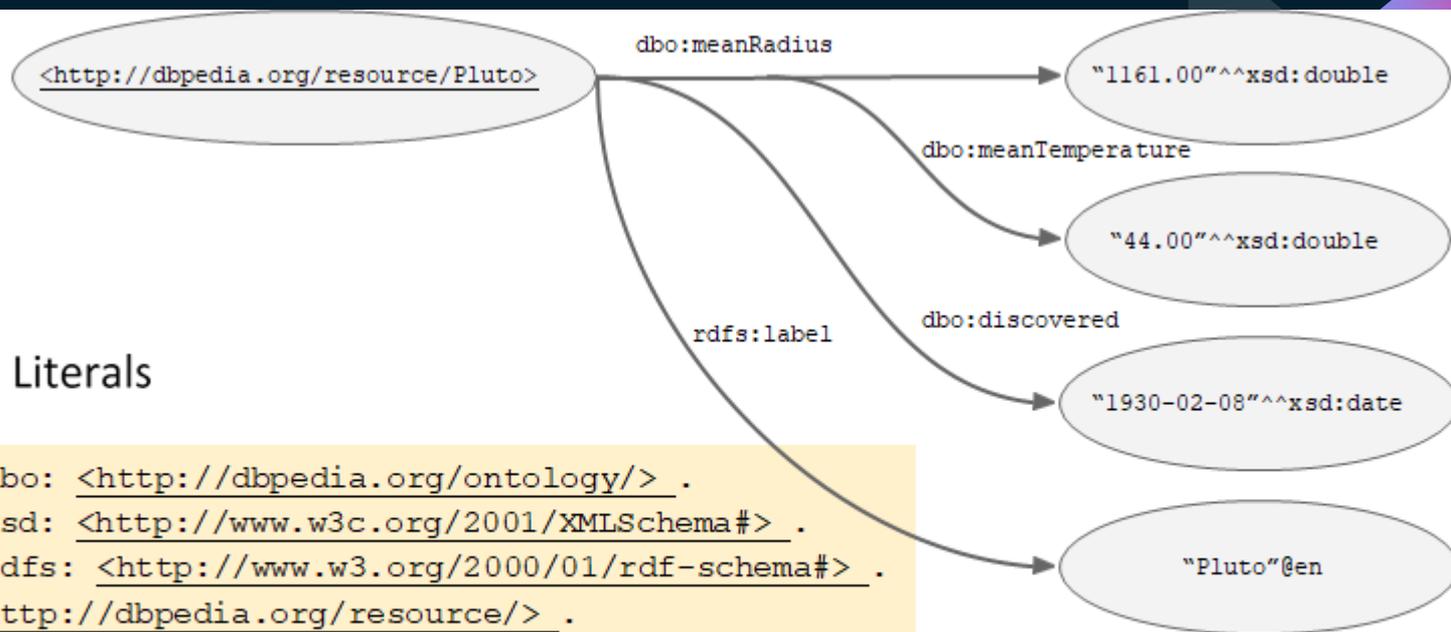
<Pluto> dbo:hasSatellite <Hydra_(moon)> ,
                    <Nyx_(moon)> ,
                    <Charon_(moon)> .
  
```

**comma** indicates that subsequent triples have same subject and property (**object list**)

- Typed Literals

```
@prefix dbo: <http://dbpedia.org/ontology/> .  
@prefix xsd: <http://www.w3c.org/2001/XMLSchema#> .  
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .  
@base <http://dbpedia.org/resource/> .
```

```
<Pluto> dbo:meanRadius "1161.00"^^xsd:double ;  
        dbo:meanTemperature "44.00"^^xsd:double ;  
        dbo:discovered "1930-02-18"^^xsd:date ;  
        rdfs:label "Pluto"@en .
```



- Existe mucho más en RDF por ejemplo qué pasaría si queremos representar que dos transbordadores visitaron plutón en dos fechas distintas?
  - Solución utilizar asociación única mediante blank nodes (bnodes) para representar relaciones multivaluadas.

```
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix xsd: <http://www.w3c.org/2001/XMLSchema#> .
@base <http://dbpedia.org/resource/> .

<Pluto> dbo:SpaceMission [ dbo:spaceShip <New_Horizons> ;
                           dbo:visited "2015-07-14"^^xsd:date .
                           ],
        [ dbo:spaceShip <USS_Enterprise_(NCC-1701)> ;
          dbo:visited "2245-07-14"^^xsd:date .
        ].
```

- También se pueden utilizar listas dentro de RDF (Containers o Collections).

```
@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
).
```

# RDFS (Schema)

Surge debido a la necesidad de una mejor expresividad semántica.



# RFD Schema

- › RFD Schema permite:
  - › Definición de clases via `rdfs:Class`
  - › Instanciación de clases via `rdf:type`
- › Ejemplo:
  - › `:Planet rdf:type rdfs:Class .`
  - › `Earth rdf:type :Planet .`

# RFD Schema

- › Definición de propiedades via `rdf:Property`
- › Definición de restricciones en el dominio y rango via `rdfs:domain` y `rdfs:range`
- › Ejemplo:
- › `:CelestialBody rdf:type rdfs:Class .`
- › `:satelliteOf rdf:type rdf:Property .`
- › `:satelliteOf rdfs:domain :CelestialBody`
- › `:satelliteOf rdfs:range :CelestialBody`

# RDF Schema

- › Definición de relaciones jerárquicas.
- › Subclasses via `rdfs:subClassOf`
  - › Ejemplo:
  - › `:Planet rdfs:subClassOf :CelestialBody .`
- › Subpropiedades via `rdfs:subPropertyOf`
  - › Ejemplo:
  - › `:artificialSatelliteOf rdfs:subPropertyOf :satelliteOf`

```
@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 .
```

Class Definitions

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

Property Definitions

```
: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" .
```

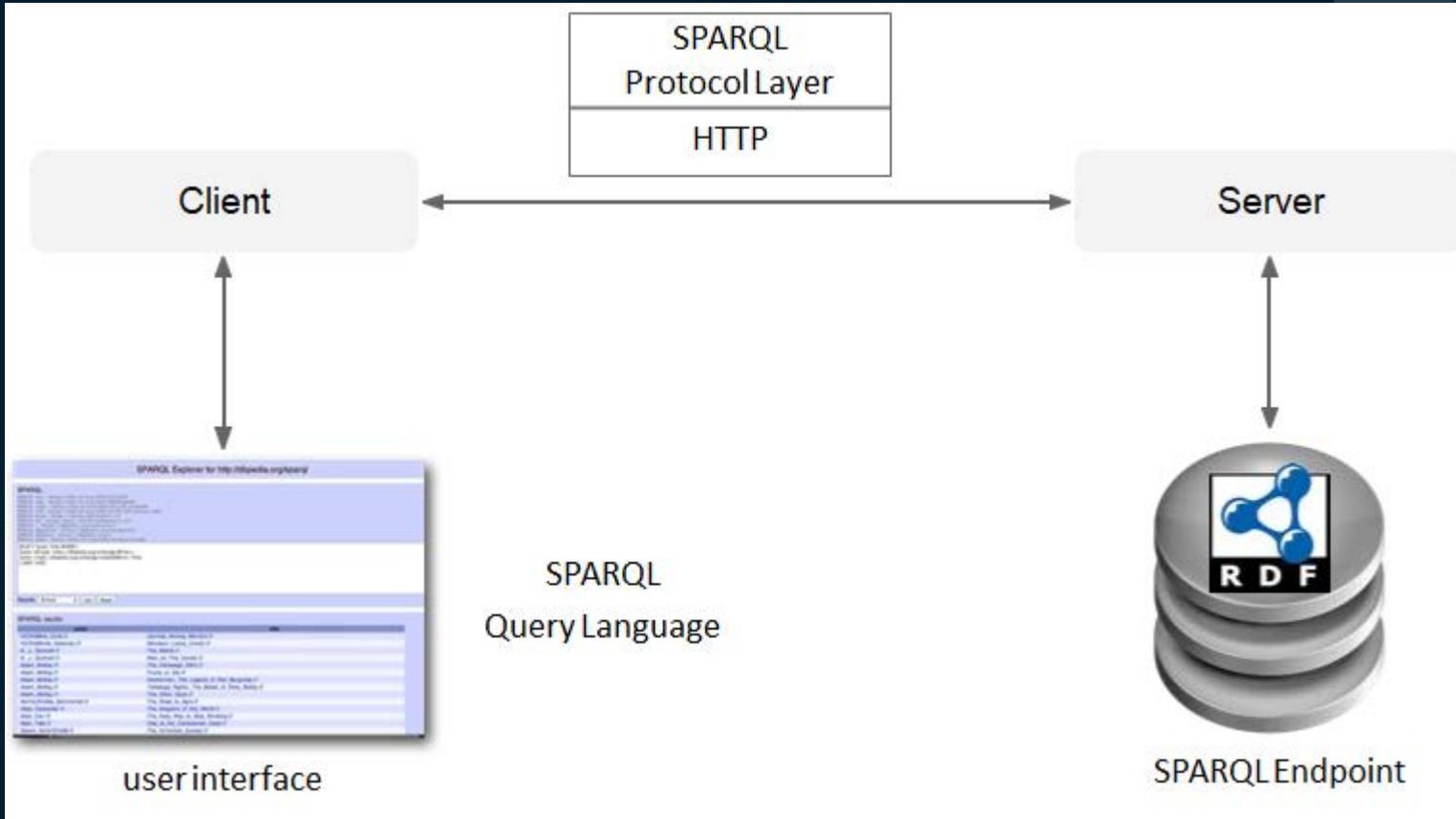
Instance Definitions

# SPARQL

Es un lenguaje de consulta sobre RDF, que permite hacer búsquedas sobre los recursos de la web semántica.

# SPARQL

<https://www.w3.org/wiki/SparqlEndpoints>  
<https://www.w3.org/wiki/OpenLinkSoftware>



# Como funciona?

- › SPARQL utiliza variables ligadas a RDF.
  - › Ejemplo: ?title, ?author, ?address
- › Inspirado en SQL (SELECT, FROM, WHERE).
  - › Ejemplo: SELECT ?title, ?author, ?publish

?title	?author	?published
1984	George Orwell	1948
Brave New World	Aldous Huxley	1932
Fahrenheit 451	Ray Bradbury	1953

SPARQL Result

# SPARQL

- › Funciona mediante pattern matching.

Triple Pattern

```
?country dbo:capital ?capital .
```

RDF Graph

```
dbpedia:Venezuela rdf:type dbo:Country .
```

```
dbpedia:Venezuela dbo:capital dbpedia:Caracas .
```

```
dbpedia:Venezuela dbprop:language "Spanish" .
```

```
dbpedia:Germany rdf:type dbo:Country .
```

```
dbpedia:Germany dbo:capital "Berlin" .
```

```
dbpedia:Germany dbp:language "German" .
```

...

# SPARQL

- › Busca a todos los autores y los títulos de sus obras más notables.

```
PREFIX : <http://dbpedia.org/resource/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
```

```
SELECT ?author_name ?title specifies output variables
```

```
FROM <http://dbpedia.org/> specifies graph to be queried
```

```
WHERE {
    ?author rdf:type dbo:Writer .
    ?author rdfs:label ?author_name .
    ?author dbo:notableWork ?work .
    ?work rdfs:label ?title .
}
```

*specifies graph pattern to be matched*

# SPARQL

[Liga a consulta](#)

- › Busca todos los autores y los títulos de sus obras más notables, ordenados por autor en orden ascendente, solo 100 resultados y empezando en la lista con el offset en la posición 10.

```
PREFIX :      <http://dbpedia.org/resource/>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo:  <http://dbpedia.org/ontology/>

SELECT ?author_name ?title

FROM <http://dbpedia.org/>

WHERE {
    ?author rdf:type dbo:Writer .
    ?author rdfs:label ?author_name .
    ?author dbo:notableWork ?work .
    ?work rdfs:label ?title .
}

ORDER BY ASC (?author_name)
LIMIT 100
OFFSET 10
```

*solution sequence  
modifiers*

# SPARQL

[Liga a consulta](#)

- › Se pueden aplicar FILTROS

```
PREFIX : <http://dbpedia.org/resource/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?author_name ?title ?pages
FROM <http://dbpedia.org/>
WHERE {
    ?author rdf:type dbo:Writer .
    ?author rdfs:label ?author_name .
    ?author dbo:notableWork ?work .
    ?work dbo:numberOfPages ?pages
    FILTER (?pages > 500) .
    ?work rdfs:label ?title .
} LIMIT 100
```

*specifies constraints  
for the result*

# SPARQL

[Liga a consulta](#)

## › Mas filtros

- Example: Filter results only for English labels

```
PREFIX : <http://dbpedia.org/resource/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?author_name ?title ?pages
FROM <http://dbpedia.org/>
WHERE {
    ?author rdf:type dbo:Writer .
    ?author rdfs:label ?author_name
    FILTER (LANG(?author_name)="en").
    ?author dbo:notableWork ?work .
    ?work dbo:numberOfPages ?pages
    FILTER (?pages > 500) .
    ?work rdfs:label ?title .
    FILTER (LANG(?title)="en").
} LIMIT 100
```

# Mas de SPARQL

- › Se pueden utilizar expresiones regulares en los filtros.
  - › Ejemplo: `FILTER REGEX (?title, "love", "i")`.
- › Se pueden hacer consultas sobre datos opcionales, por ejemplo podemos agregar a la consulta títulos que tengan idioma en ingles:
  - › Ejemplo: `OPTIONAL {?work rdfs:label ?en_title  
FILTER (LANG(?en_title)="en")`
- › Se pueden realizar consultas Federadas.

Wikidata Query Service

Ejemplos Query Builder Ayuda Más herramientas

```

1 #Personas nacidas en México
2 select distinct ?item ?itemLabel ?itemDescription ?sitelinks where {
3   ?item wdt:P31 wd:Q5; # Any instance of a human.
4     wdt:P19 wd:Q96; # Who was born in México.
5     wikibase:sitelinks ?sitelinks.
6
7   SERVICE wikibase:label { bd:serviceParam wikibase:language "en,es" }
8 }
9 ORDER BY DESC(?sitelinks)

```

item itemLabel itemDescription

wd:Q972927	Felipe Olivares	Mexican footballer
wd:Q926221	Francisco Garza Gutiérrez	Mexican footballer (1904-1965)

[https://www.wikidata.org/wiki/Wikidata:SPARQL\\_query\\_service/Wikidata\\_Query\\_Help/Result\\_VIEWS/es](https://www.wikidata.org/wiki/Wikidata:SPARQL_query_service/Wikidata_Query_Help/Result_VIEWS/es)

[https://www.wikidata.org/wiki/Wikidata:SPARQL\\_tutorial](https://www.wikidata.org/wiki/Wikidata:SPARQL_tutorial)

<http://histropedia.com/showcase/wikidata-viewer.html>

<http://query.wikidata.org/>



# 4,220,000

Tripletas en dbpedia.org



95,437,902

Elementos en wikidata

# Limitaciones de RFD(S) y SPARQL

- › Con RDF tenemos relaciones de taxonomía y clases.
- › Con OWL tenemos adicionalmente opciones de inferencia como:
  - › Cardinalidad
  - › Union
  - › Interseccion

# Web Ontology Language OWL

Es un vocabulario para describir ontologías a partir de tripletas RDF.

Viene en diferentes versiones:

- › OWL EL, OWL RL, OWL QL  $\subseteq$  OWL2 DL  $\subseteq$  OWL2 Full

# ¿Qué cosas tiene OWL que no tiene RDFS?

- › Class Constructors
  - › Conjunción, disyunción y negación.
- › Restricciones en propiedades (universales y existenciales).
- › Restricciones de cardinalidad y en rango mayor o menor que.
- › Tipos de datos.

# OWL

- › Al igual que RDF se puede representar en diversos formatos.
  - › Funcional.
  - › RDF/XML
  - › OWL/XML
  - › Manchester-Syntax
  - › Turtle.

# OWL Classes

- › Hay dos clases predefinidas
  - › owl:Thing
  - › owl:Nothing
- › Una clase se define como:
  - › :Book a owl:Class

# OWL Individuos

- › Se definen individuos via la pertenencia a una clase
  - › `:NineteenEightyfour` a `:Book` .
- › Se pueden definir individuos sin pertenecer a una clase y se realiza mediante los `named individuals`
  - › `:Lupita` a `owl:NamedIndividual`

# OWL Propiedades

- › Existen dos tipos de propiedades:
  - › object properties
  - › datatype properties
- › Object Properties
  - › `:author a owl:ObjectProperty;`
  - › `rdfs:domain :Book;`
  - › `rdfs:range :Writer;`

```
∃ author.τ ⊆ Book
```

```
τ ⊆ ∀ author.Writer
```

# OWL Propiedades

- › Datatype Properties
  - › `:publicationYear a owl:DatatypeProperty;`
  - › `rdfs:domain :Book;`
  - › `Rdfs:range xsd:integer;`

```
∃publicationYear.τ ⊆ Book
```

```
τ ⊆ ∀publicationYear.Integer
```

```
:Book a owl:Class .
:Writer a owl:Class .

:author a owl:ObjectProperty ;
        rdfs:domain :Book ;
        rdfs:range :Writer .

:publicationYear a owl:DatatypeProperty ;
        rdfs:domain :Book ;
        rdfs:range xsd:integer .

:GeorgeOrwell a Writer .
:NineteenEightyFour a :Book ;
        :author :GeorgeOrwell ;
        :publicationYear 1948 .
```

# Owl Jerarquias

```
:Novel a owl:Class ;  
    rdfs:subClassOf :Book .  
:Book a owl:Class ;  
    rdfs:subClassOf :Work .  
:Work a owl:Class .
```

- › Se puede inferir que una novela también es una subclase de Work.

# Otras cosas en OWL

- › Se pueden declarar clases disjuntas.
  - › owl:disjointWith
- › Clases equivalentes.
  - › owl:equivalentClass
- › Clases que se intersectan con otras, que se unen o sean el complemento de otras.  
(AND, OR, negación)
  - › intersectionOf, unionOf, complementOf

# Sistemas Manejadores de Bases de Datos para el web semántico



Apache Jena

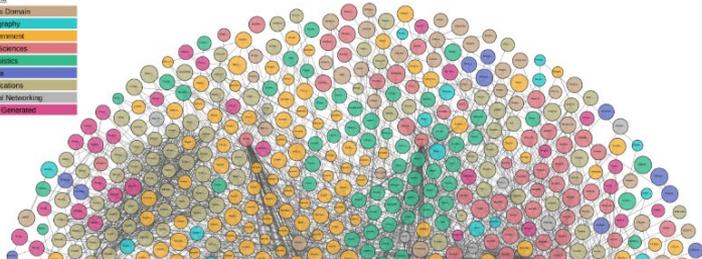


ontotext



# The Linked Open Data Cloud

- Legend
- Local Domain
- Geography
- Government
- Life Sciences
- Expressions
- Media
- Publications
- Social Networking
- User Generated



The GeoNames geographical database covers all countries and contains over eleven million placenames that are available for download free of charge.

all countries

search [\[advanced search\]](#)

enter a location name, ex: "Paris", "Mount Everest", "New York"

## Browse the names

- [Countries](#)
- [Postal codes](#)
- [Country statistics](#)
- [Recent modifications](#)

## Information

- [About GeoNames](#)
- [Data Sources](#)
- [User manual](#)
- [Ambassadors and Team](#)
- [Forum](#)
- [Blog](#)
- [Mailing list](#)
- [Commercial Support and Consulting](#)

## Download

- [Info](#)
- [Free Gazetteer Data](#)
- [Free Postal Code Data](#)
- [Premium Data](#)

## Web Services

- [Overview](#)
- [Documentation](#)
- [Client Libraries](#)
- [Premium Web Services](#)



THE YUMMLY EXPERIENCE

## Yummly is your smart cooking sidekick

From recipe recommendations just for you to handy tools and helpful videos, Yummly has everything you need to improve life in the kitchen.



DATOS ENLAZADOS NOVEDADES LABS

## Corpus de Sonetos del Siglo de Oro

25/07/2019 admin

### CORPUS DE SONETOS

SIGLO DE ORO

El Corpus de Sonetos del Siglo de Oro está formado por sonetos escritos en castellano entre los siglos XVI y XVII del catálogo de obras de la Biblioteca Virtual Miguel de Cervantes.

Cada soneto se ha anotado en XML de acuerdo al standard TEI. Aparte de la cabecera y la información de

## CATEGORÍAS

Elegir categoría

## ÚLTIMOS TWEETS

Tweets por @FBVMC

FBVMC rebotó

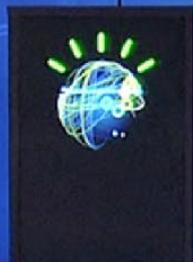
Hispanoamericano de Escritores @HE\_escritores  
Gisela Belli ingresa a la Academia Nicaragüense de la Lengua ehispano.com.mn/cultura/letras... via @El\_Universal\_MX

ΠΙΕΝΣΕ

ΣΜΑΘΜΙΣ

THINK

सोचिए



ΣΚΕΨΟΥ

DENKE

PENSER

\$300,000

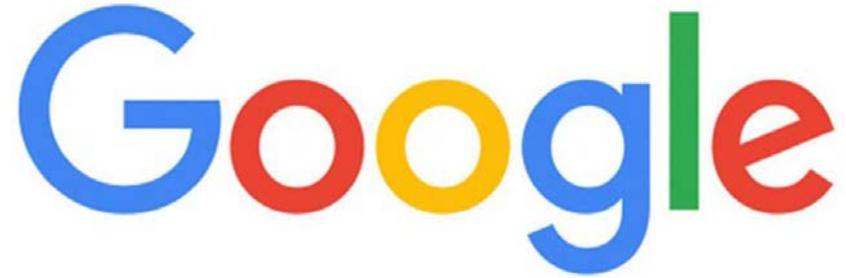
Who is Stoker?  
(FOR ONE WELCOME OUR  
NEW LOANER OVERLODS)  
\$ 1,000

\$1,000,000

Who is Bram  
Stoker?  
\$ 17,973

\$200,000

WHO IS  
BRAM STOKER?  
\$5600

The Google logo is displayed in its characteristic multi-colored font (blue, red, yellow, blue, green, red) on a white rectangular background. The logo is centered in the upper half of the slide.

Google

Dime: ¿Cuál es la capital de México?



¿Cuál es la capital de México?



[Todos](#)

[Maps](#)

[Imágenes](#)

[Noticias](#)

[Videos](#)

[Más](#)

[Preferencias](#)

[Herramientas](#)

Cerca de 42,800,000 resultados (0.84 segundos)

México / Capital



# Ciudad de México

También se buscó

[Ver 10 más](#)

## Default Data Set Name (Graph IRI)

## Query Text

```
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT ?capital
WHERE {
  ?country dbo:capital ?capital .
  FILTER ( ?country = dbr:Mexico)
}
```

Results Format

SPARQL | HTML5 table

**capital**[http://dbpedia.org/resource/Mexico\\_City](http://dbpedia.org/resource/Mexico_City)

# Bibliografia

Foundations of Semantic Web Technologies, Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph, CRC Press/Chapman and Hall (2009)

<https://www.w3.org/2001/sw/>

[https://www.w3.org/2001/sw/wiki/Books#Books\\_on\\_Semantic\\_Web:\\_Intro](https://www.w3.org/2001/sw/wiki/Books#Books_on_Semantic_Web:_Intro)

