

Desde ceRo

La Web de Datos



#LDR_MRA15
@marianorico

Supported by:

- **LIDER (EU 610782)**
- **MINECO (JCI-2012-12719 and UNPM13-4E-1814)**

Contenido

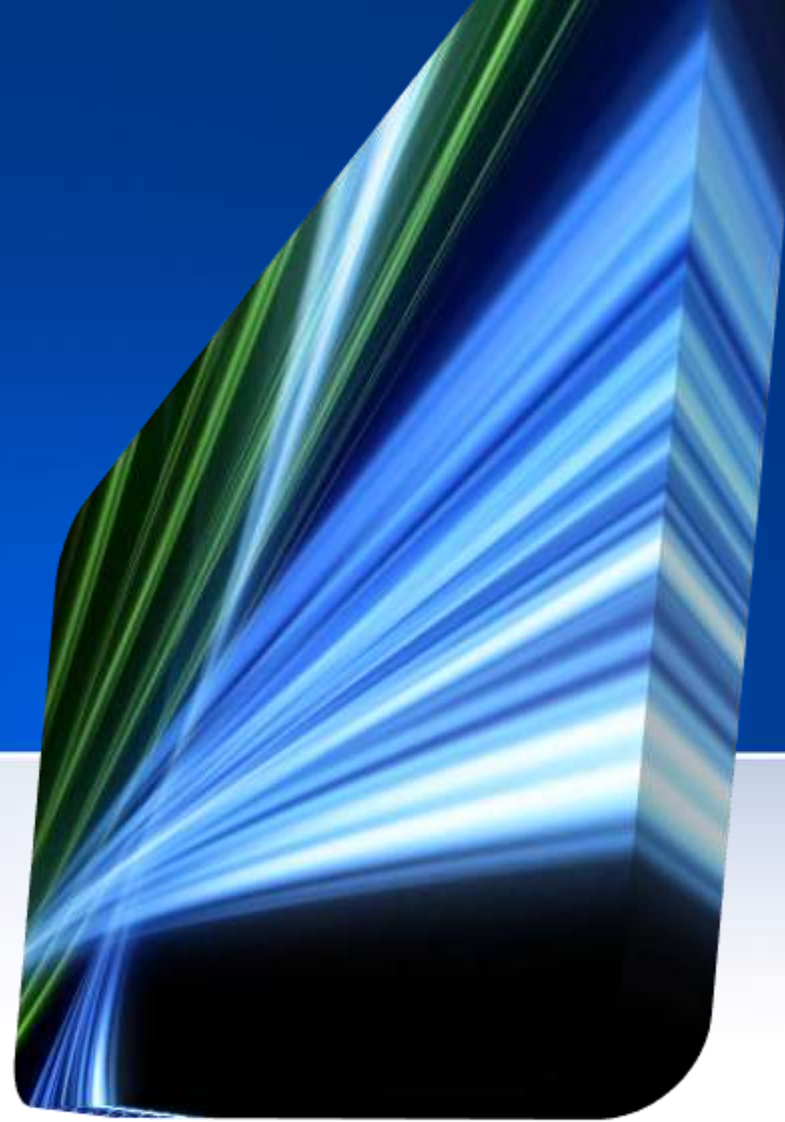


<http://goo.gl/jEd7sJ>

- [Datos enlazados \(linked data\)](#)
- [DBpedia del español](#)
- [Paquetes R](#)

¿Qué son?

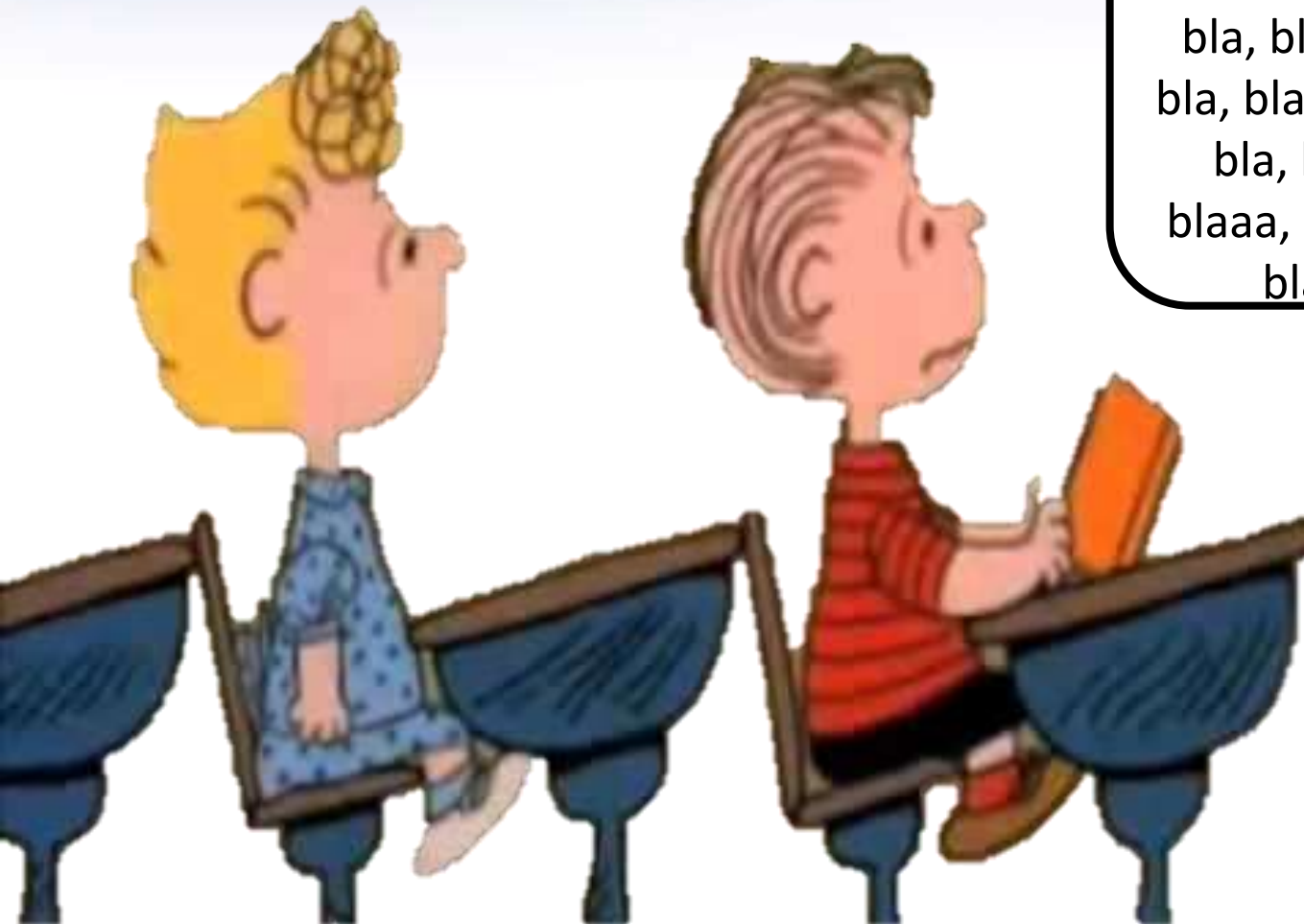
DATOS ENLAZADOS (LINKED DATA)



Terminología



Bla, bla, bla, **RDF**, bla, bla,
blaaa, **SPARQL**, bla, bla, bla,
Semantic Web, bla, bla, bla,
bla, bla, bla, **Linked Data**,
bla, bla **Open Data**, bla, bla,
bla, blaaaaaa, bla, bla,
blaaa, bla, bla, **Ontologías**,
blaaa, blaaa, bla..



Terminología



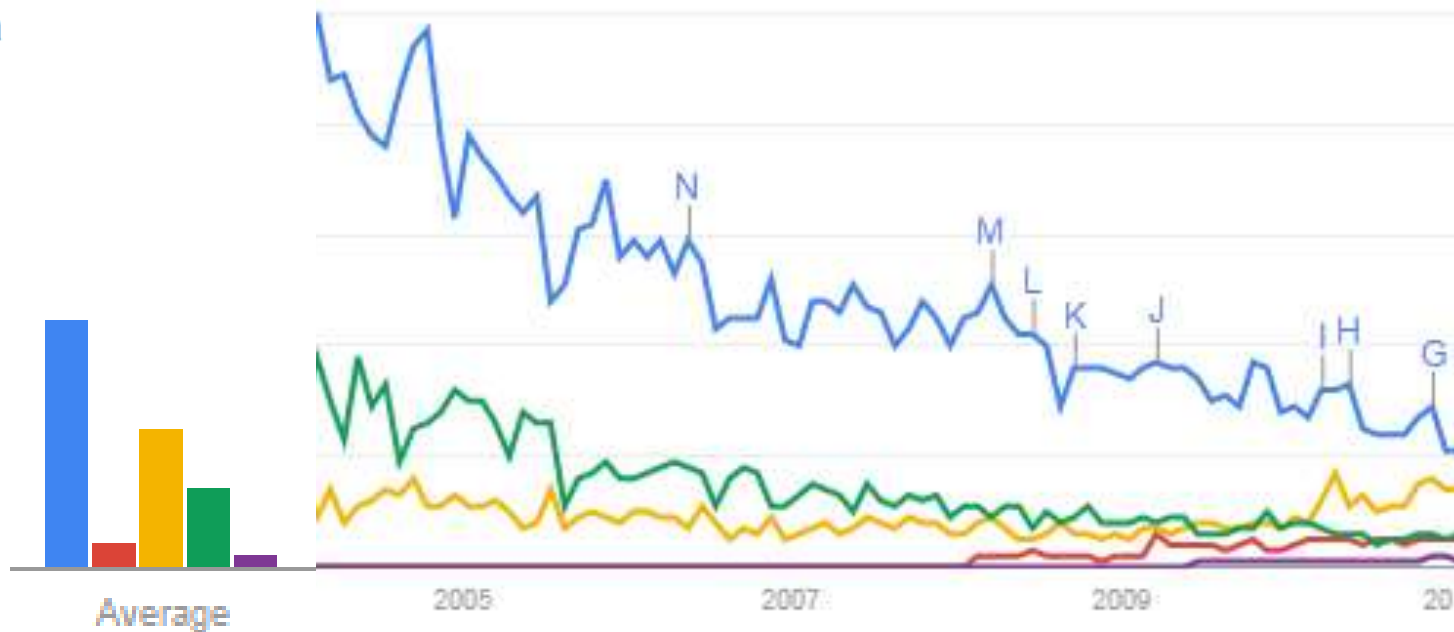
Terminología



Términos de búsqueda



- × "semantic web"
- × "linked data"
- × "open data"
- × eGovernment
- + Añadir término



Datos → Business?



The Economist

Obama the warrior
Misgoverning Argentina
The economic shift from West to East
Genetically modified crops: Blessom
The right to eat cats and dogs

The data deluge

AND HOW TO HANDLE IT: A 14-PAGE SPECIAL REPORT

ISSN 0950-0804

Subscription rates: \$120/yr (US), \$150/yr (UK), \$180/yr (Canada)

MARCH 26, 2011

TIME

A tale of two Libyas
Plus why the U.S. can't win
the war in Libya BY RALPH QUINN

The GOP's reinformation
campaigns BY MICHAEL GLOTT

Could your
baby be
depressed?
BY JENNIFER W. HANSEN

THE GUTPIE
Word of the
year: A dictionary
of slang

Owns a laptop
Age: 38-39
Likes: online news
Likes: Asian cuisine
Dislikes: corn
Likes: green lentils
Favorite celebrities: Pe
ZIP code: 10704
Likes: sci-fi
Likes: Sociology & Ant
Sister is a la
Spent \$130 on int
Dislikes: autos & vehicles
Smart-phone user
Lives: come & go
Has had LASIK surgery
Likes: restaurants

YOUR DATA FOR SALE

Everything about you is being tracked—
get over it

BY PAUL STEIN

DATA - BIG APPLE

g Value, Improving
mance:

PRIME Research Strategic
Communications and Research

SA Public
Relations
Society of
America

Pre-Conference Seminar May 14
May 15-16
New York City

Register Now

PRIME RESEARCH

Datos RDF: Datos con sentido



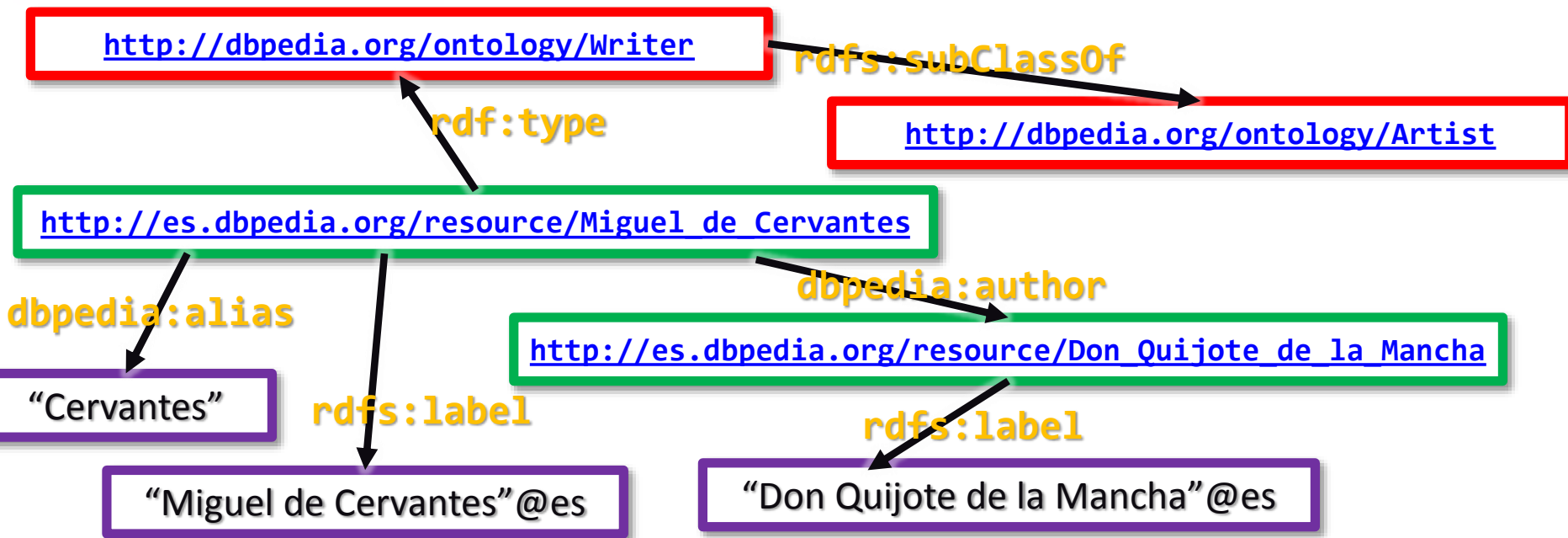


Datos con semántica

- Semántica porque
 - Enlaza con datos de una (o varias) ontologías
 - Una ontología es un modelo matemático que permite
 - Razonar → Crear nuevos datos
 - Preguntar si algo existe o una afirmación es cierta
- Sin enlaces a ontologías no hay semántica

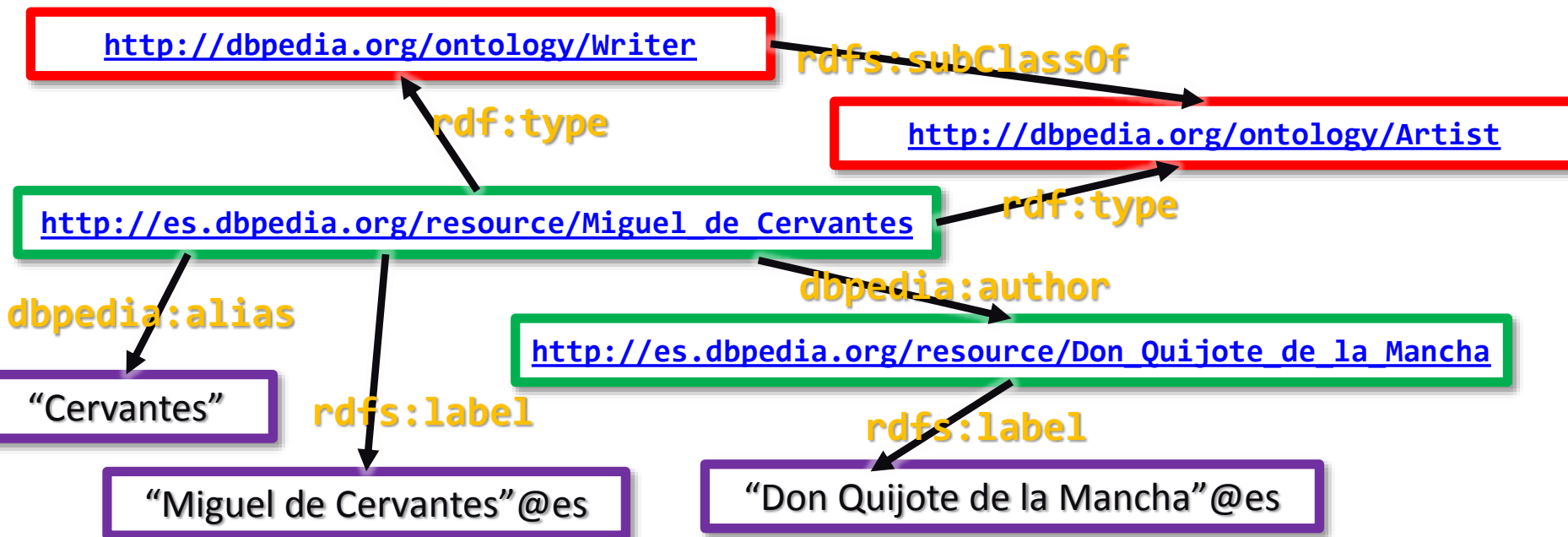
Datos con semántica

- Ejemplo: “Cervantes escribió el Quijote”



Datos con semántica

- Razono: “Cervantes es un artista”



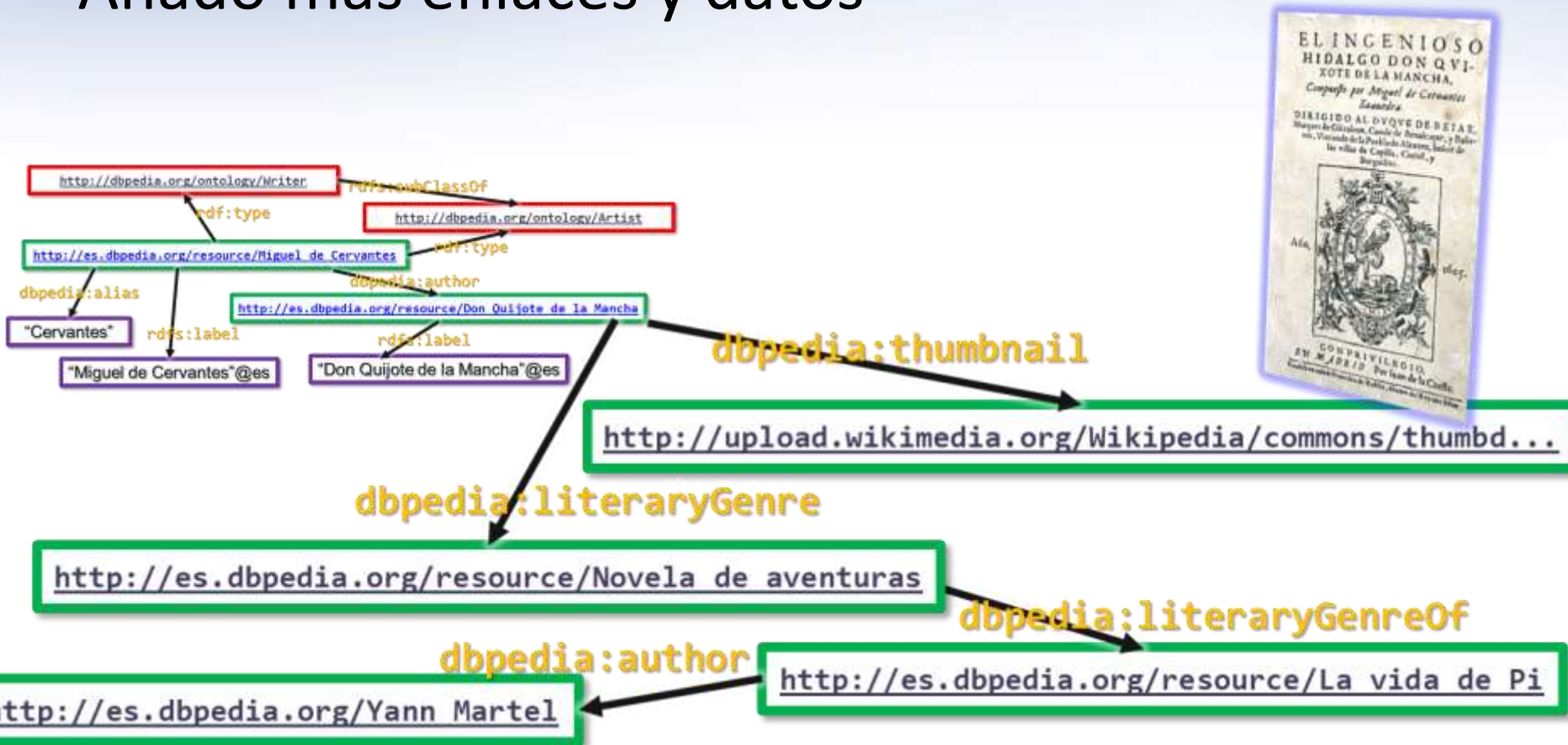
Datos con semántica

- Añado más enlaces y datos



Datos con semántica

- Añado más enlaces y datos



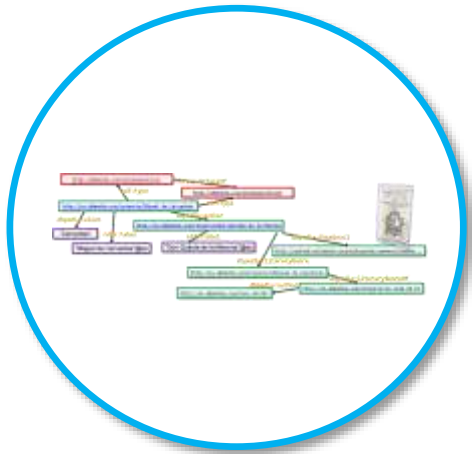
Datos con semántica

- Añado más enlaces y datos



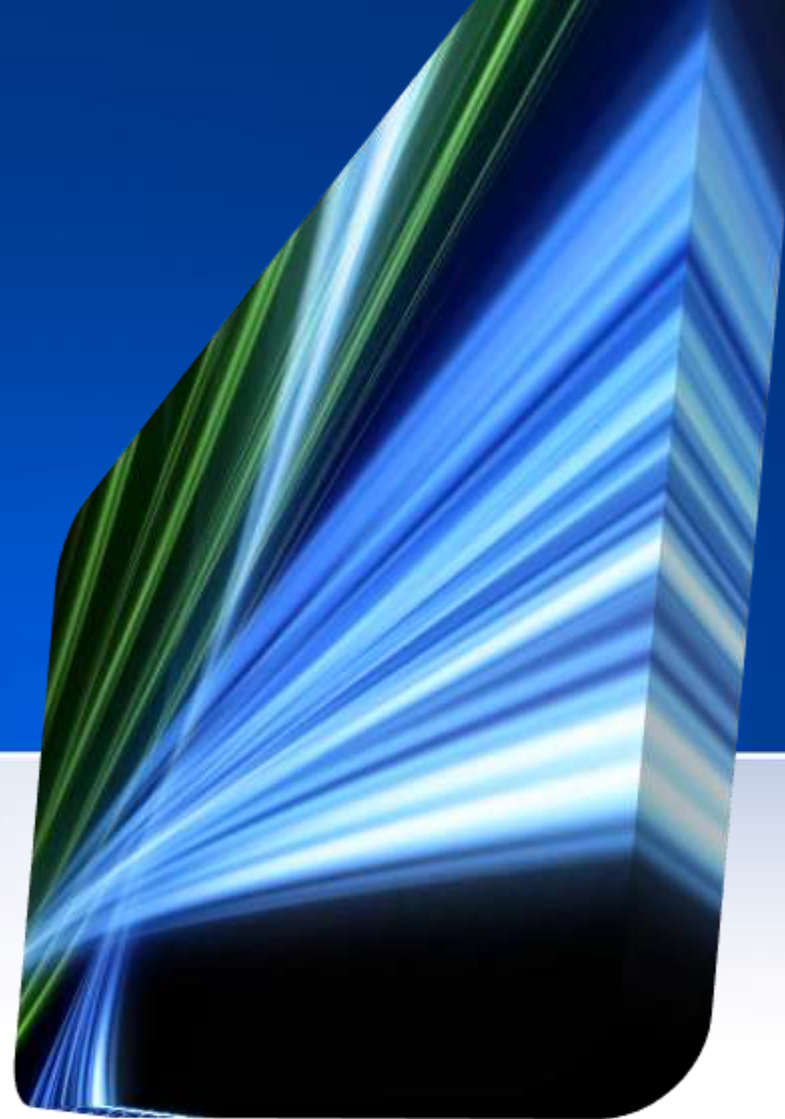
Datos con semántica

- Conjunto de datos (dataset)



El núcleo de la LOD cloud

DBPEDIA

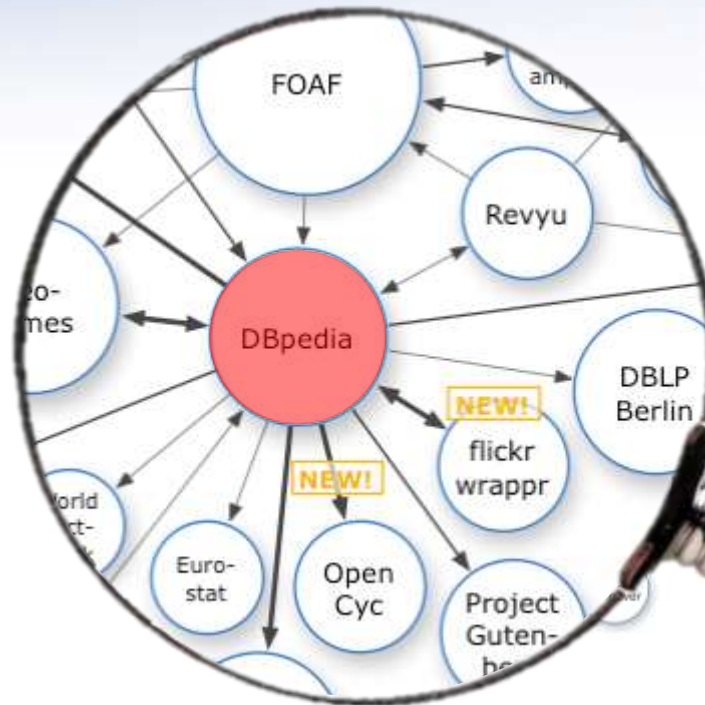


DBpedia, núcleo de Linked Data



2007

25 datasets

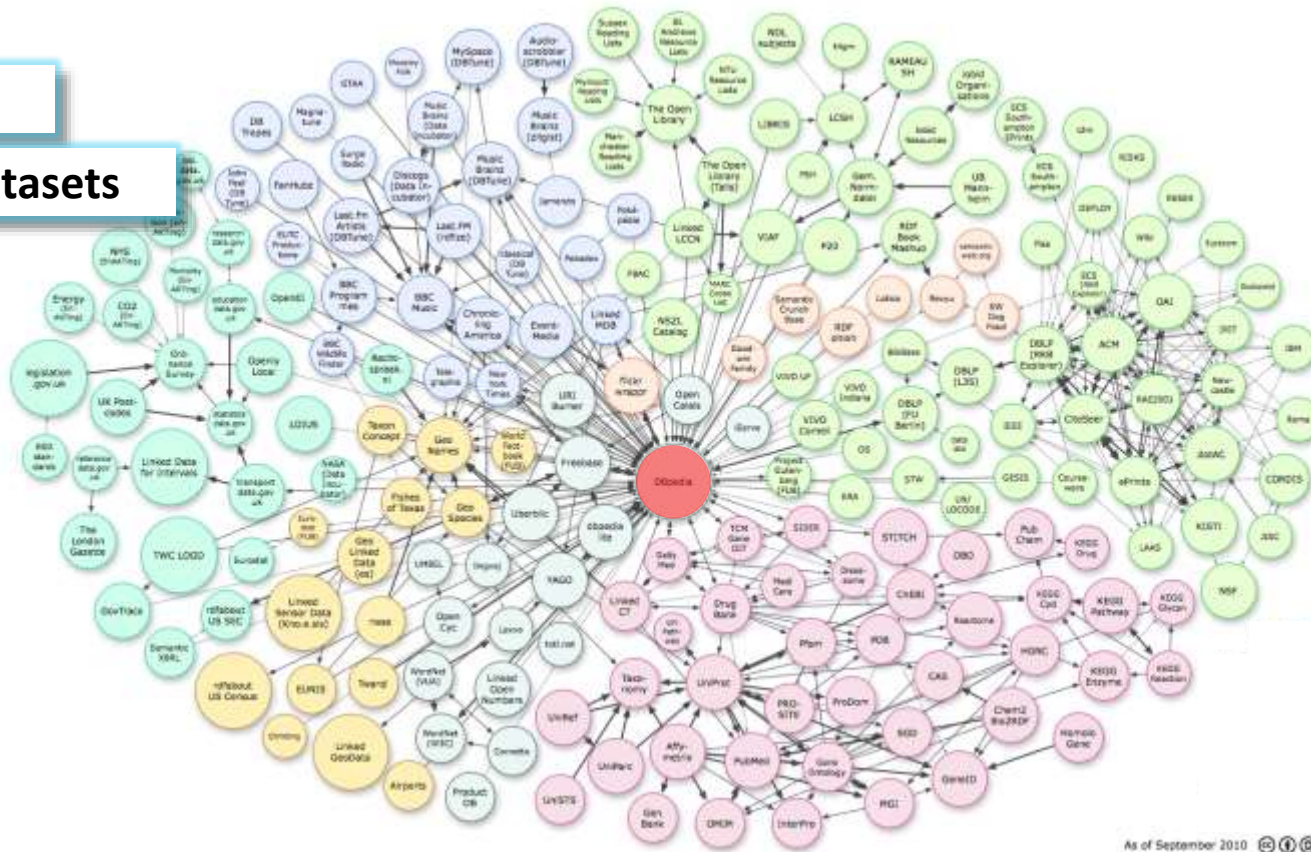


DBpedia, núcleo de Linked Data



2010

203 datasets



- Media (light blue circle)
- Geographic (yellow circle)
- Publications (light green circle)
- User-generated content (orange circle)
- Government (light cyan circle)
- Cross-domain (light grey circle)
- Life sciences (pink circle)

As of September 2010

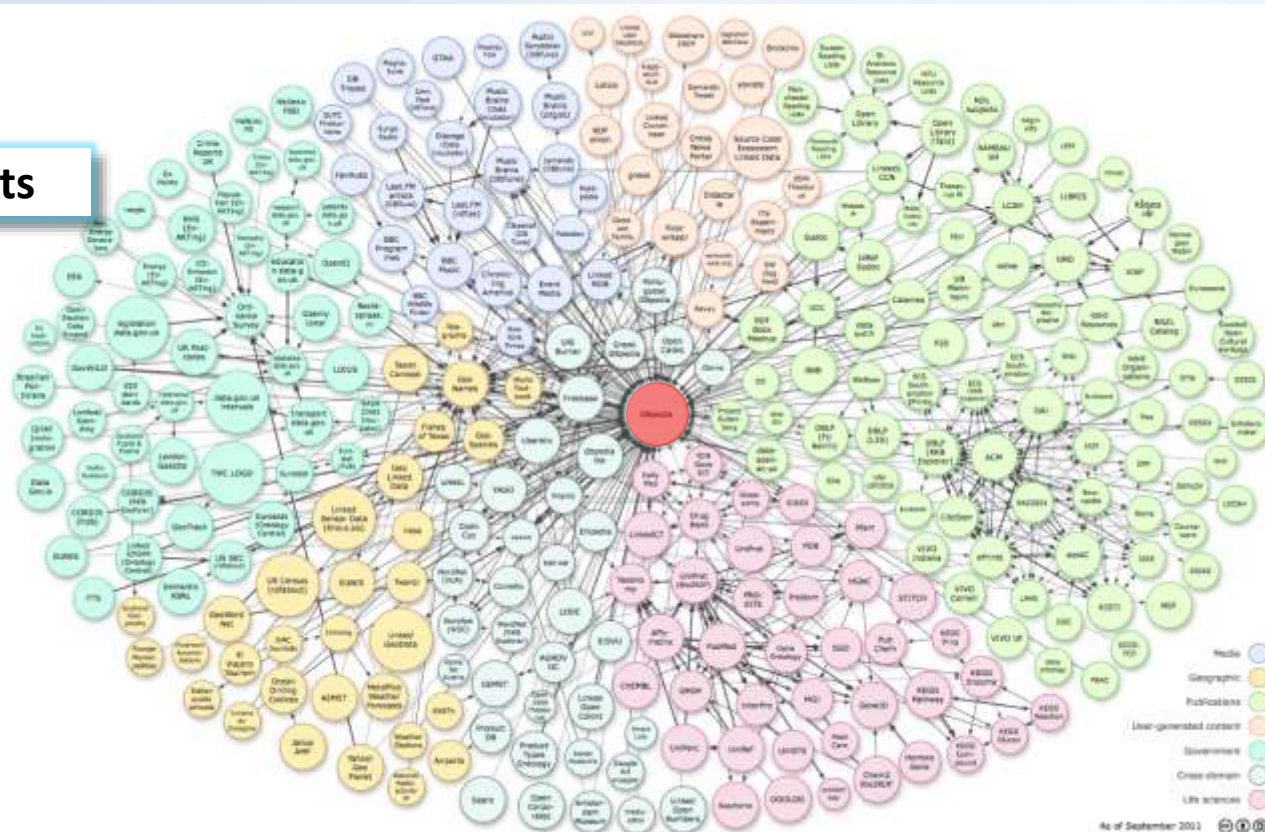
http://lod-cloud.net/versions/2011-09-19/lod-cloud_colored.png

DBpedia, núcleo de Linked Data



2011

295 datasets

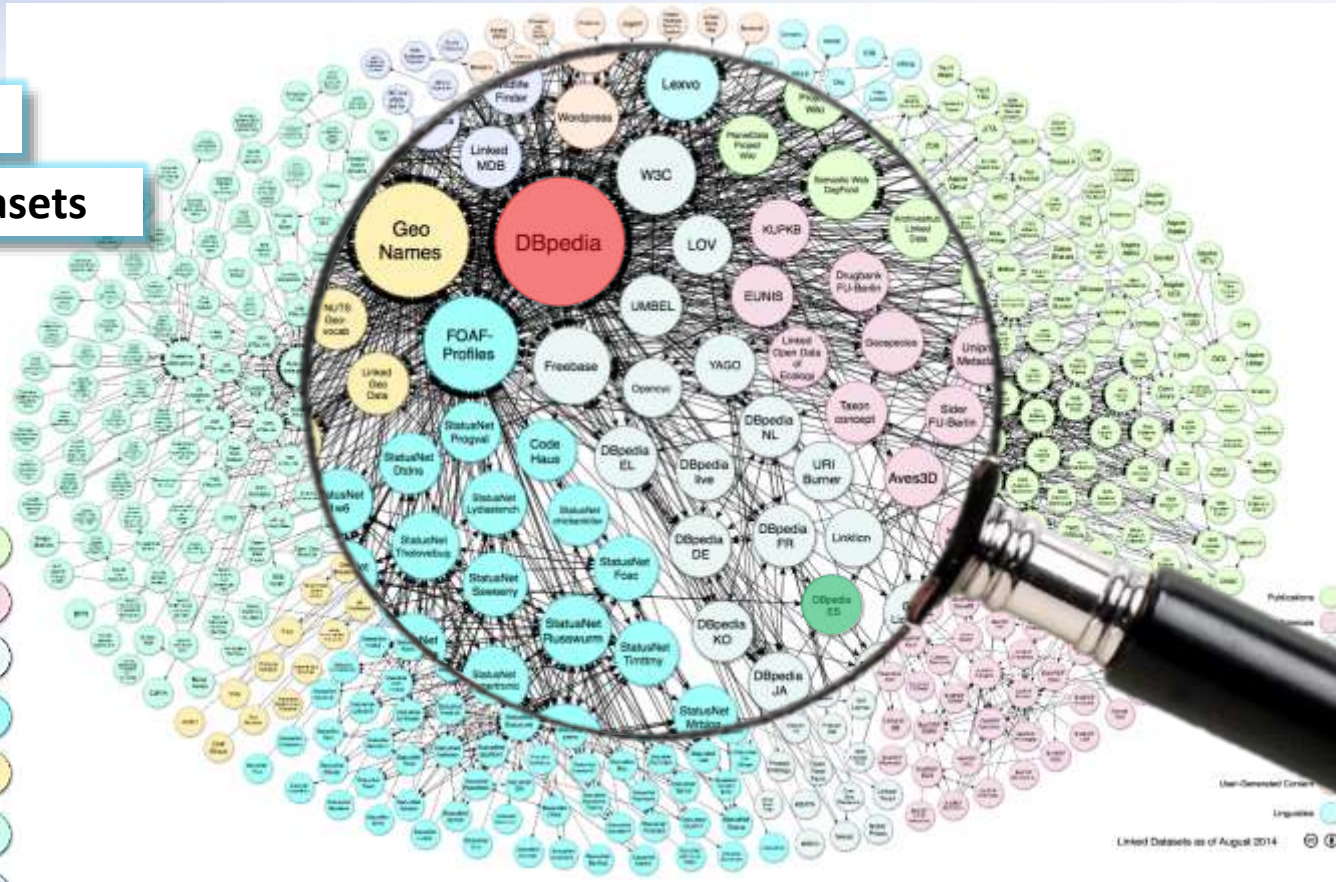











DBpedia, núcleo de Linked Data



2014

570 datasets

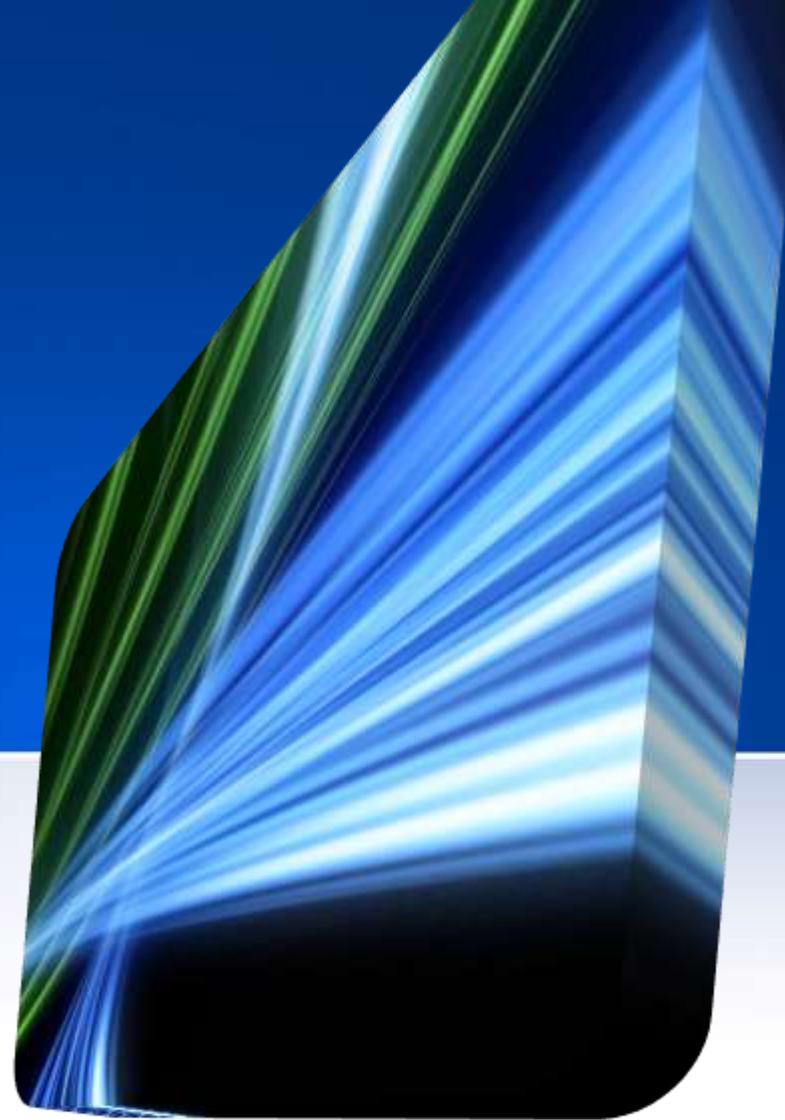


- Publications 
- Life Sciences 
- Cross-Domain 
- Social Networking 
- Geographic 
- Government 
- Media 
- User-Generated Content 
- Linguistics 

Linked Datasets as of August 2014

El núcleo de los datos semánticos del español

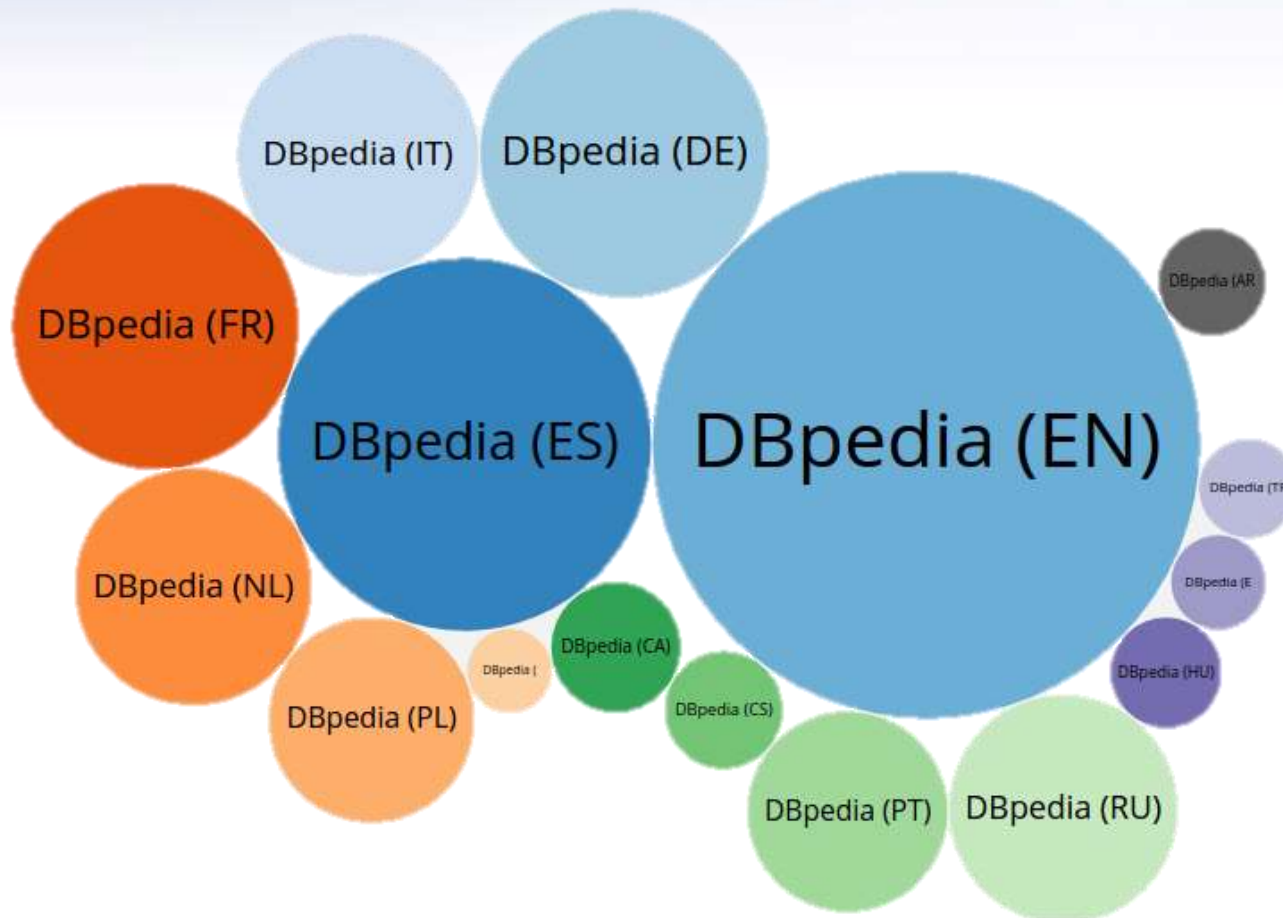
DBPEDIA DEL ESPAÑOL



esDBpedia

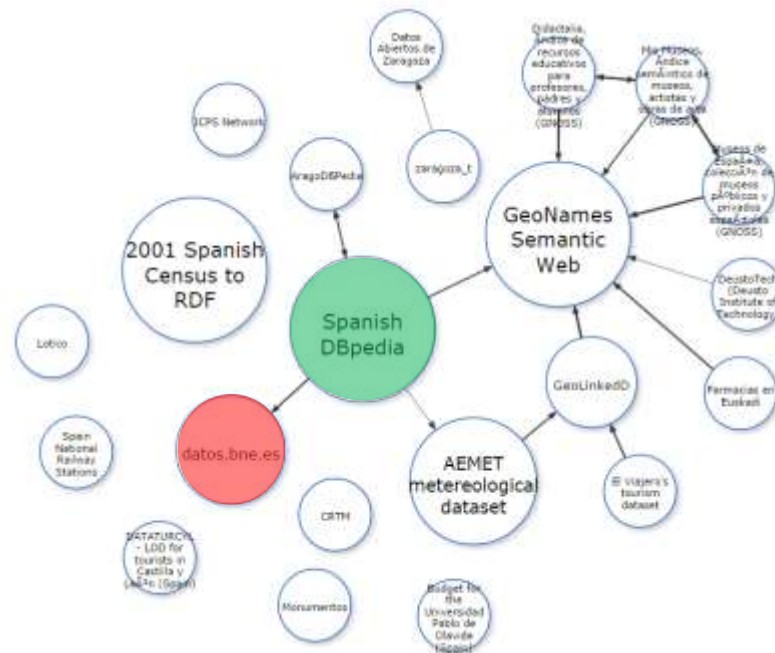


- La no inglesa más grande



esDBpedia

- Centro de la Linked Open Data (LOD) cloud del idioma español



De xx.wikipedia.org a xx.dbpedia.org



WIKIPEDIA **Woody Allen**

Este artículo o sección puede referirse, por esta razón a una **página homónima**.

Para conocer qué artículo o página tiene que ver con esta página, consulte el **listado de homónimos**.

ALAN STEWART KÖNIGSBERG (Brooklyn, 1 de diciembre de 1935), conocido por su nombre artístico **Woody Allen**, es un director, guionista, actor, músico, dramaturgo, humorista y escritor estadounidense. Ha sido ganador del premio Óscar en cuatro ocasiones.

Zu viele de los artículos más importantes, influyentes y prolíficos de la era moderna, ha producido desde 1969, la era de las películas más altas de la "New Wave", incluyendo una gran cantidad de películas "New York" y sus parásitos. Como director, tiene grandes habilidades cinematográficas para ser director, sus películas como "Annie Hall", "Manhattan", "Hannah y sus hermanas" y también comediantes como Groucho Marx.

Woody Allen

Nombre de nacimiento: Alan Stewart Königsberg

Residencia: 88 Madison Avenue Park, Nueva York, 1 de diciembre de 1935, 29 años

Actividad: Director, guionista, actor, músico, dramaturgo, humorista, escritor

100 películas

Web: [www.woodyallen.com](#)

Premios Óscar: Mejor película (1977), Mejor guionista (1977)

Edición de «Woody Allen»

N C [iconos] + Avanzado + Caracteres especiales

[[referencias adicionales]]

[[Pícha de actor]]

[[Pícha de actor]] = Woody Allen [[Archivos:Prince of Belles.jpg|150px|Previsión Príncipe de Asturias]]

[[foto]] = woody allen at the Tribeca Film

[[nombre de nacimiento]] = Allen Stewart Königsberg

[[fecha de nacimiento]] = [[fecha|11|12|1935|ada2]]

[[lugar de nacimiento]] = [[estado|USA]] [[localidad]] [[regionalidad]]

[[ocupación]] = [[Director de cine|Director]], [[Libreturgo]], [[humorista]], [[escritor]]

[[año debut]] = 1959

[[año retiro]] = presente

[[ideología]] = [[Social-Vl Previsión]] [[small]]|1997-

[[hijos]] = 2

[[premios Óscar]] = [[premio Óscar a la mejor dirección]]

[[Categoría:Películas de 1977|1977]] [[Anexo:Hall]] [[original|mejor guion original]] [[de />]] [[Hall]] [[de />]] [[Categoría:Películas de 1977|1977]] [[premios globo de oro]] [[Anexo:Globo de Oro]] [[Categoría:Películas de 1981|1981]] [[premios globo de oro]] [[Anexo:Globo de Oro]] [[Categoría:Películas de 1982|1982]] [[premios globo de oro]] [[Anexo:Globo de Oro]]

About: Woody Allen

An Entity of Type **Actor**, from Named Graph <http://es.dbpedia.org>, within Data Space <es.dbpedia.org>

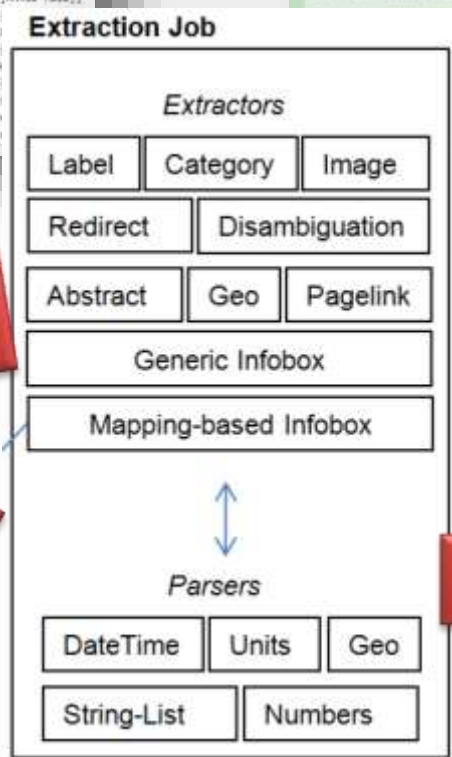
Alan Stewart Königsberg (Brooklyn, 1 de diciembre de 1935), más conocido por su nombre artístico Woody Allen, es un director, guionista, actor, músico, dramaturgo, humorista y escritor estadounidense. Ha sido ganador del premio Óscar en cuatro ocasiones. Es uno de los directores más respetados, influyentes y prolíficos de la era moderna, rodando una película al año desde 1969.

Property

dbpedia-owl:abstract

Value

- dbpedia:Annie_Hall
- dbpedia:Hannah_y_sus_hermanas
- dbpedia:Midnight_in_Paris
- dbpedia:Medianoche_en_Paris
- dbpedia:Hannah_and_Her_Sisters
- dbpedia:Anexo:Óscar_a_la_mejor_película_extranjera
- dbpedia:Anexo:Premio_Gotstadta_del_Festival_de_San_S
- dbpedia:Manhattan_(película)
- dbpedia:Premio_Príncipe_de_Asturias_de las_Artes
- dbpedia:The_Purple_Rose_of_Cairo
- dbpedia:Óscar_a_la_mejor_película_extranjera
- dbpedia:Premio_Dobson
- Alan Stewart Königsberg
- dbpedia:Brooklyn
- dbpedia:Nueva_York



DBpedia

Mapping es Ficha de actor

This is the mapping for the Wikipedia template `!de:actor`. First pages of this Wikipedia template here

Test this mapping @ [de:en:woodyallen](#) (with an example Wikipedia page). Check which properties are not mapped yet!

Good news about mapping Wikipedia templates

Template Mapping

map to class: actor

Mappings

Property Mapping

template property: nombre

ontology property: foaf:name

Property Mapping

template property: nombre real

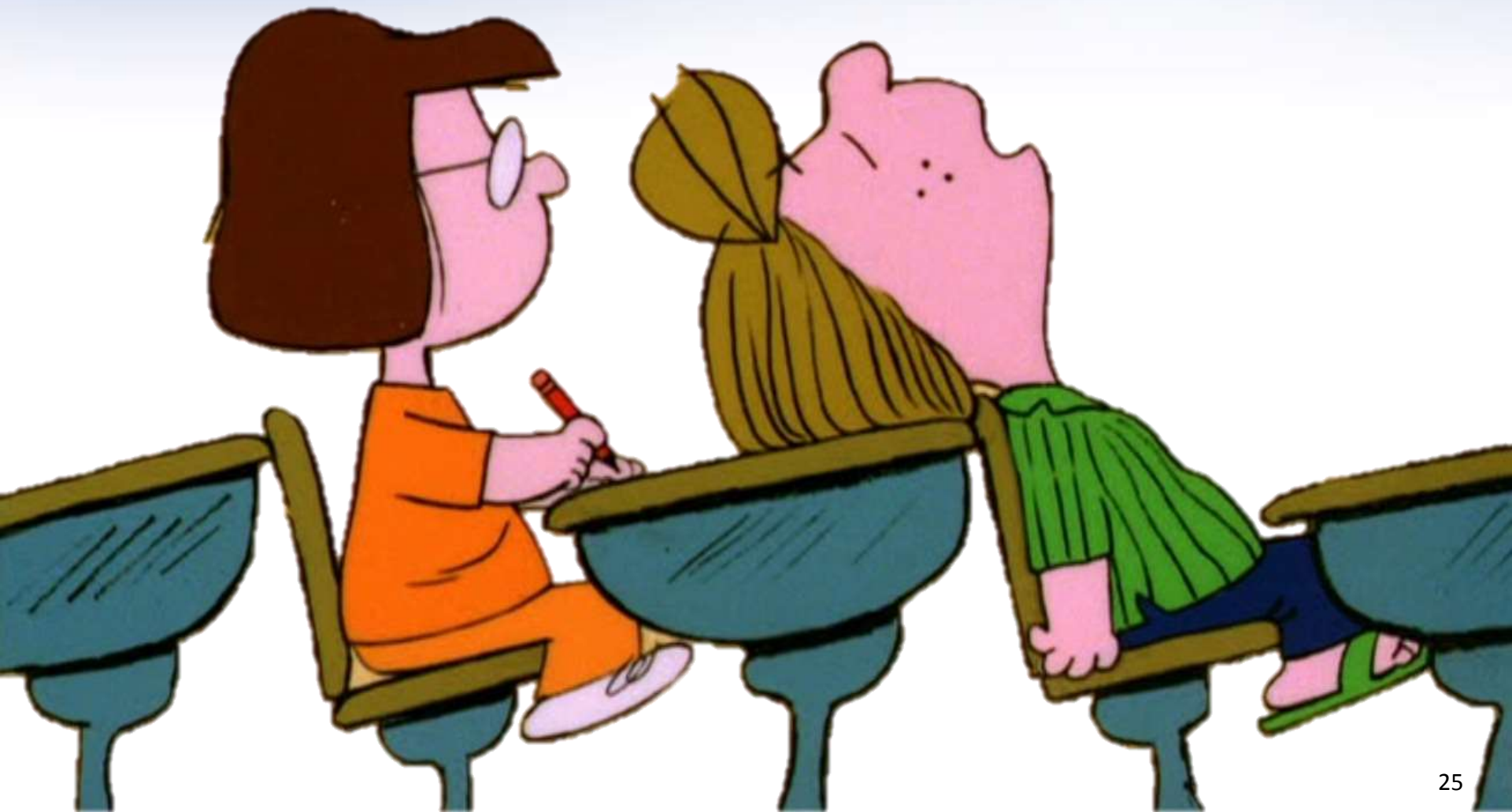
ontology property: foaf:name

Property Mapping

template property: nombre real

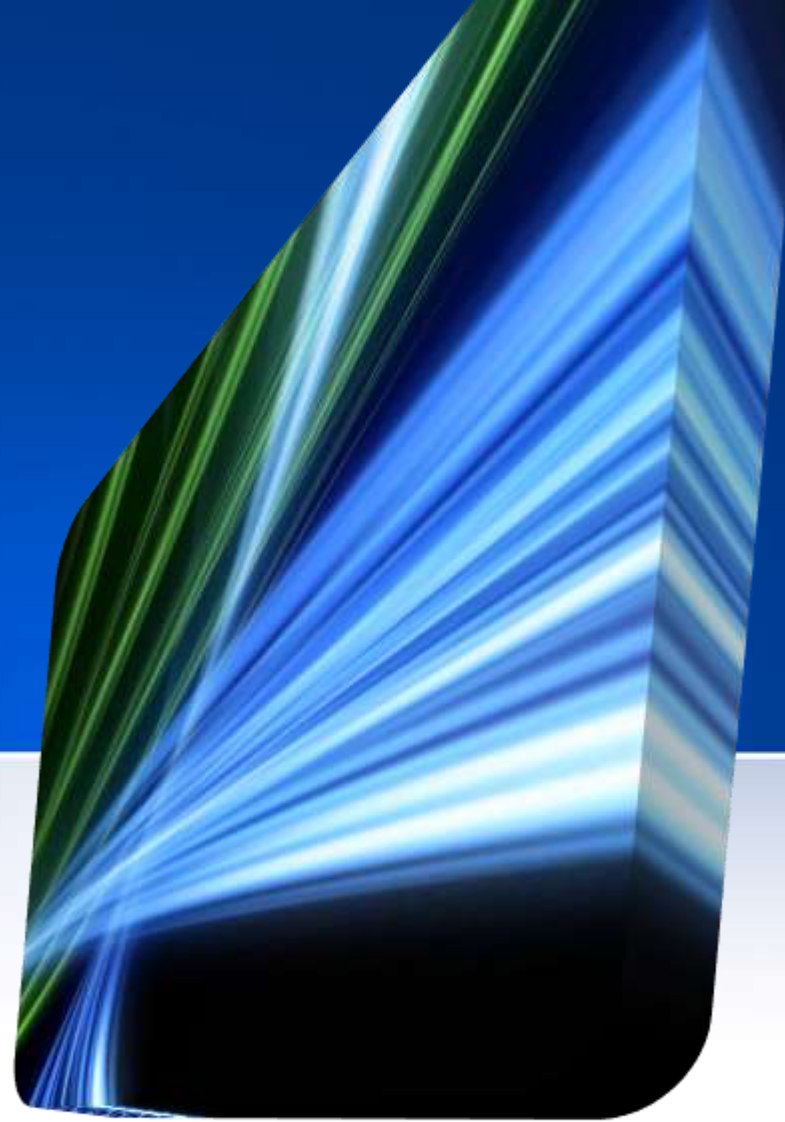
ontology property: foaf:name

Hasta aquí... ¿bien?...



Dame

DATOS, DATOS, DATOS





¿Qué datos tiene DBpedia?

- DBpedia (idioma inglés). Datos septiembre 2014
 - Cosas (entradas Wikipedia): 4.58 millones
 - Clasificadas en la ontología DBpedia: 4.22
 - 1.445.000 personas 735.000 lugares
 - 241.000 organizaciones 123.000 discos 87.000 películas...
- DBpedias de 125 idiomas
 - Enlaces:
 - 25.2 millones a imágenes 29.8 millones a páginas web externas
 - 50.0 millones a otros RDF datasets 80,9 millones a categorías Wikipedia
 - 41.2 millones a categorías YAGO
 - Tripletas (triples)
 - 3000 millones de triples (583 millones de la versión inglesa)
 - (No están todos en el EP)



La ontología DBpedia

- Es una ontología OWL
 - ¡Ojo!, común a todos los idiomas
 - Varias versiones. La última tiene:
 - 685 clases
 - 2679 propiedades
 - 1600 [DBpediaDatatypeProperty](#)
 - 1079 [DBpediaObjectProperty](#)
 - Equivalencias con schema.org
 - 47 owl:equivalentClass
 - 35 owl:equivalentProperty



Ver las clases y propiedades

<http://mappings.dbpedia.org/server/ontology/classes>

- [Astronaut](#) (edit)
- [Athlete](#) (edit)
 - [AustralianRulesFootballPlayer](#) (edit)
 - [BadmintonPlayer](#) (edit)
 - [BaseballPlayer](#) (edit)
 - [BasketballPlayer](#) (edit)
 - [Boxer](#) (edit)
 - [AmateurBoxer](#) (edit)
 - [BullFighter](#) (edit)
 - [ChessPlayer](#) (edit)
 - [Cricketer](#) (edit)
 - [Curler](#) (edit)
 - [DartsPlayer](#) (edit)
 - [FigureSkater](#) (edit)
 - [GaelicGamesPlayer](#) (edit)
 - [GolfPlayer](#) (edit)
 - [GridironFootballPlayer](#) (edit)
 - [AmericanFootballPlayer](#) (edit)
 - [CanadianFootballPlayer](#) (edit)

<http://mappings.dbpedia.org/server/ontology/classes/BullFighter>

BullFighter (Show in class hierarchy)

Label (el): ταυρομάχος
 Label (es): torero
 Label (en): bullfighter
 Label (ja): 闘牛士
 Label (it): torero
 Super classes: [Athlete](#)

<http://mappings.dbpedia.org/index.php/OntologyProperty:DebutTeam>

Properties on *BullFighter*:

Name	Label
club (edit)	club
debutTeam (edit)	debut team
draftTeam (edit)	draft team
espnId (edit)	ESPN id
formerTeam (edit)	former team
nationalTeam (edit)	national team
sportCountry (edit)	sport country
sportSpecialty (edit)	sport specialty

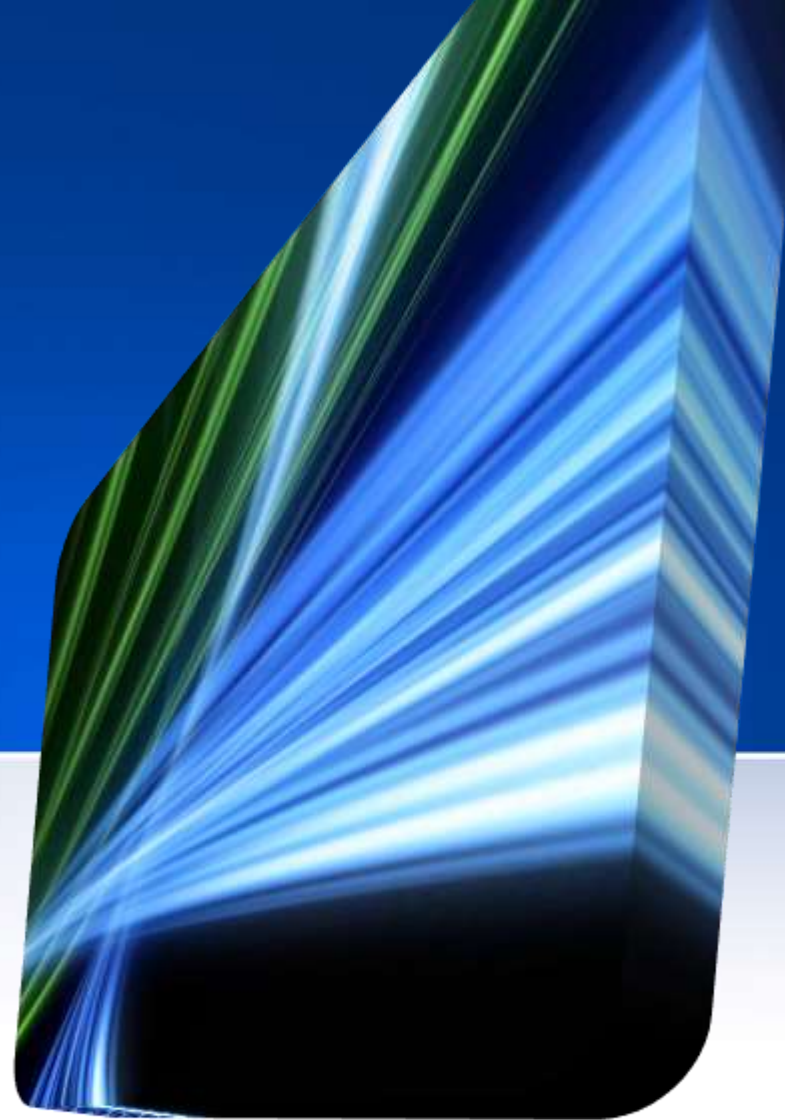
OntologyProperty:debutTeam

This is the definition of an ontology property.
 Read more about editing the ontology schema.
 You can see the result of your edit on [DBpedia Live](#) (t

Ontology object property (help)	
rdfs:label@en	debut team
rdfs:domain	Athlete
rdfs:range	SportsTeam
rdfs:type	
rdfs:subPropertyOf	
owl:equivalentProperty	

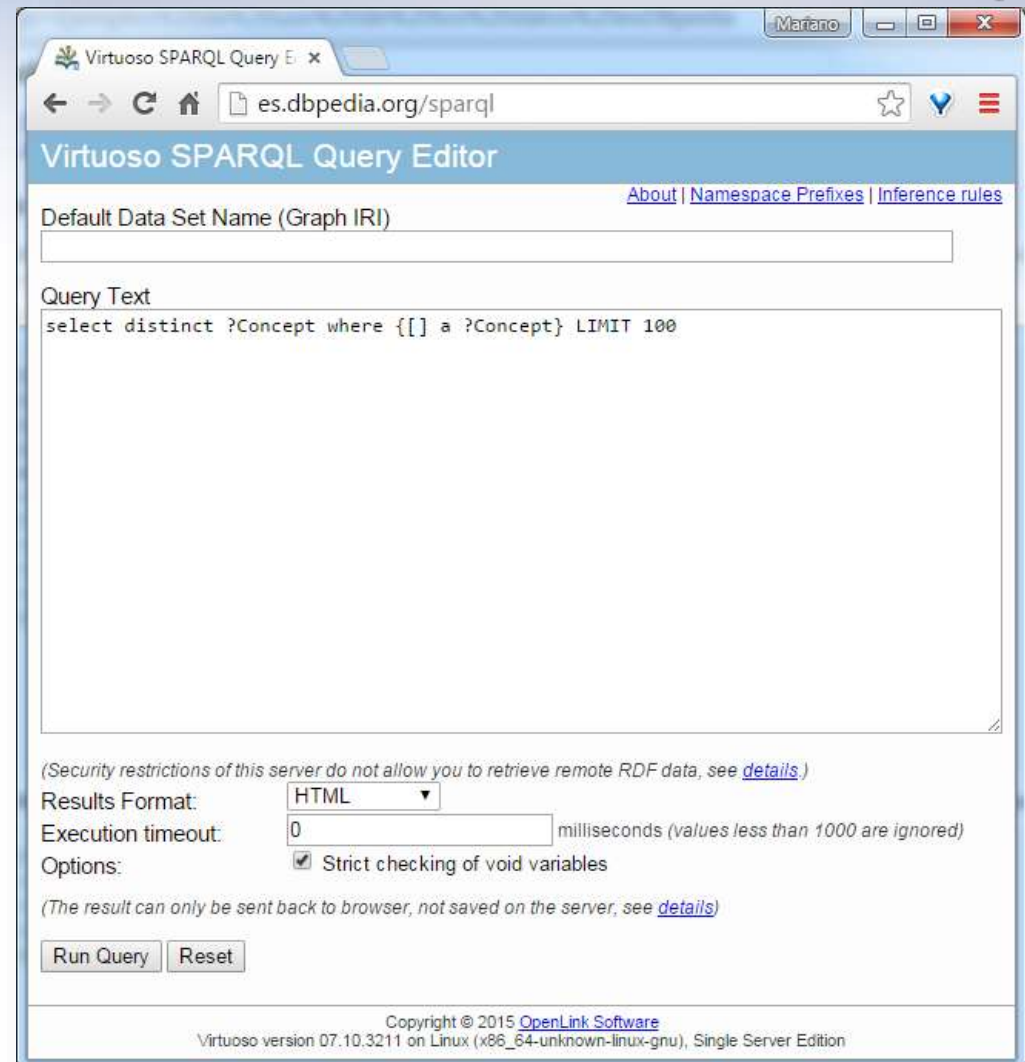
Sacando datos a un dataset de linked data

SPARQL



Sacando datos del EP

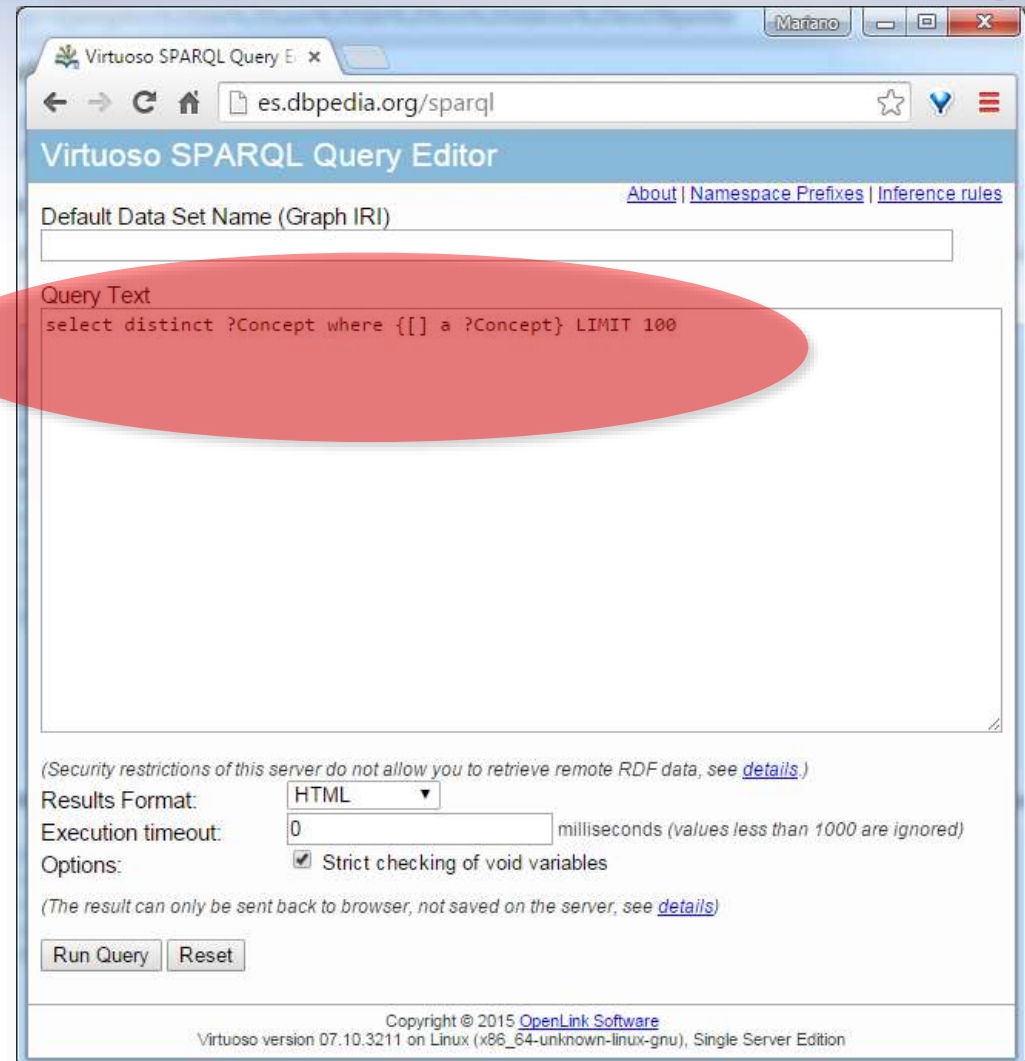
- EP = EndPoint
- “El EP” de esDBpedia
<http://es.dbpedia.org/sparql>



The screenshot shows a web browser window titled "Virtuoso SPARQL Query Editor" with the URL "es.dbpedia.org/sparql". The interface includes a "Default Data Set Name (Graph IRI)" field, a "Query Text" area containing the query `select distinct ?Concept where {[] a ?Concept} LIMIT 100`, and a "Results Format" dropdown set to "HTML". Below the query area, there are settings for "Execution timeout" (0 milliseconds) and "Options" (checked "Strict checking of void variables"). At the bottom, there are "Run Query" and "Reset" buttons. The footer indicates the software is Virtuoso version 07.10.3211 on Linux, Single Server Edition, with a copyright notice for OpenLink Software.

Sacando datos del EP

- La consulta (“la query”)
 - Sintaxis similar a SQL
(bases de datos tradicionales)



Virtuoso SPARQL Query Editor

Default Data Set Name (Graph IRI)

Query Text

```
select distinct ?Concept where {[ ] a ?Concept} LIMIT 100
```

(Security restrictions of this server do not allow you to retrieve remote RDF data, see [details](#))

Results Format:

Execution timeout: milliseconds (values less than 1000 are ignored)

Options: Strict checking of void variables

(The result can only be sent back to browser, not saved on the server, see [details](#))

Copyright © 2015 [OpenLink Software](#)
Virtuoso version 07.10.3211 on Linux (x86_64-unknown-linux-gnu), Single Server Edition

Sacando datos del EP

- Formatos de salida
 - Por omisión HTML
 - Otros formatos
 - JSON
 - CSV
 - RDF
 - ...

Virtuoso SPARQL Query Editor

Default Data Set Name (Graph IRI)

Query Text

```
select distinct ?Concept where {[] a ?Concept} LIMIT 100
```

(Security restrictions of this server do not allow you to retrieve remote RDF data, see [details](#).)

Results Format: HTML

Execution timeout: milliseconds (values less than 1000 are ignored)

Options: of void variables

(The result can only be sent by saved on the server, see [details](#))

Run Query Reset

Virtuoso ver © 2015 [OpenLink Software](#)
 linux (x86_64-unknown-linux-gnu), Single Server Edition

Sacando datos del EP

- Ejemplo “zero”
 - Número de instancias de cada clase de la ontología DBpedia

```
SELECT ?class
      (COUNT(?s) AS ?count)
WHERE {
  ?s a ?class .
  filter (strstarts(str(?class),
    "http://dbpedia.org/ontology"
  ))
}
GROUP BY ?class
ORDER BY DESC(?count)
```

The screenshot shows the Virtuoso SPARQL Query Editor interface. The browser address bar is at `es.dbpedia.org/sparql`. The page title is "Virtuoso SPARQL Query Editor". Below the title, there are links for "About", "Namespace Prefixes", and "Inference rules". A text input field for "Default Data Set Name (Graph IRI)" is empty. The "Query Text" area contains the following SPARQL query:

```
SELECT ?class (COUNT(?s) AS ?count) WHERE {
  ?s a ?class .
  filter (strstarts(str(?class), "http://dbpedia.org/ontology"))
} GROUP BY ?class ORDER BY DESC(?count)
```

Below the query text, there are settings for "Results Format" (set to HTML), "Execution timeout" (set to 0 milliseconds), and "Options" (with "Strict checking of void variables" checked). There are "Run Query" and "Reset" buttons. Below the interface, a table displays the results of the query.

class	count
http://dbpedia.org/ontology/Agent	391362
http://dbpedia.org/ontology/Place	365479
http://dbpedia.org/ontology/Wikidata:Q532	365479
http://dbpedia.org/ontology/Person	351860
http://dbpedia.org/ontology/PopulatedPlace	301112
http://dbpedia.org/ontology/Region	294327
http://dbpedia.org/ontology/AdministrativeRegion	294327

¡¡Alto ahí!! SPARQuééeeel?????



Aprende SPARQL



Aprende SPARQL

- A base de ejemplos
 - En el wiki de esDBpedia



The screenshot shows a web browser window with the URL `es.dbpedia.org`. The page features a navigation bar with 'Home', 'Wiki', and 'Apoyos' links. The 'Wiki' link is highlighted with a red oval. Below the navigation bar, the page title is 'DBpedia del español' and the subtitle is 'Este es el capítulo del idioma español de DBpedia (esDBpedia)'. There are social media icons for Twitter, Google+, and Facebook. The main content area starts with a language selector 'Español English' and a heading '¿Qué es esDBpedia?'. Below this, there is a paragraph of text: 'El proyecto DBpedia ha generado durante mucho tiempo información semántica a partir de la wikipedia inglesa. Desde junio de 2011 el proceso de generación de información extrae información de wikipedia en 15 de sus versiones'.

Aprende SPARQL

- A base de ejemplos
 - En el wiki de esDBpedia

- [Navega por los datos](#)
- [Browse the data](#)
- [Recent Changes](#)
- [WikiEtiquette](#)
- [Unused pages](#)
- [Undefined pages](#)
- [Page Index](#)
- [SystemInfo](#)

iBienvenido!

Esto es la sección del idioma español de la DBpedia.

¿Tienes prisa?. Esta es una consulta sobre la dbpedia del español: "[¿Qué toreros se casaron con cantantes de copla?](#)".

¿Lo quieres ver más despacio?

1) Selecciona y copia el texto de la consulta que aparece en el siguiente recuadro (expresada en lenguaje SPARQL):

```

PREFIX dcterms: <http://purl.org/dc/terms/>
SELECT ?torero ?cantante WHERE{
  ?torero rdf:type dbpedia-owl:BullFighter .
  ?torero dbpedia-owl:spouse ?cantante .
  ?cantante dcterms:subject <http://es.dbpedia.org/resource/Categoría:Cantantes_de_coplas>
}
```

2) Pégalo en el [SPARQL endpoint](#) y pulsa sobre el botón "Run Query".

Como has visto, esta es la respuesta:


torero	cantante
http://es.dbpedia.org/resource/José_Ortega_Cano	http://es.dbpedia.org/resource/Rocío_Jurado
http://es.dbpedia.org/resource/Curro_Romero	http://es.dbpedia.org/resource/Concha_Márquez_Piquer
http://es.dbpedia.org/resource/Francisco_Rivera	http://es.dbpedia.org/resource/Isabel_Pantoja

Esta consulta aprovecha los datos proporcionados por la [wikipedia del idioma español](#). Por ejemplo, la categoría '[Cantantes de copla](#)' sólo existe en la wikipedia del español.


[Más ejemplos y preguntas abiertas](#) #

⚠ Para evitar spammers, sólo los usuarios registrados pueden editar las páginas

ℹ Con tu ayuda podemos mejorar este sitio. Por favor, deja tus comentarios en [Mejoras](#).
¿Buscas [Eventos pasados](#)?




POLITÉCNICA
"Ingeniamos el futuro"

Queries “duras”



- Hay queries que necesitan mucho tiempo de cálculo
- Son rechazadas por el EP
- Ejemplo
 - Número de instancias de cada propiedad de la ontología DBpedia

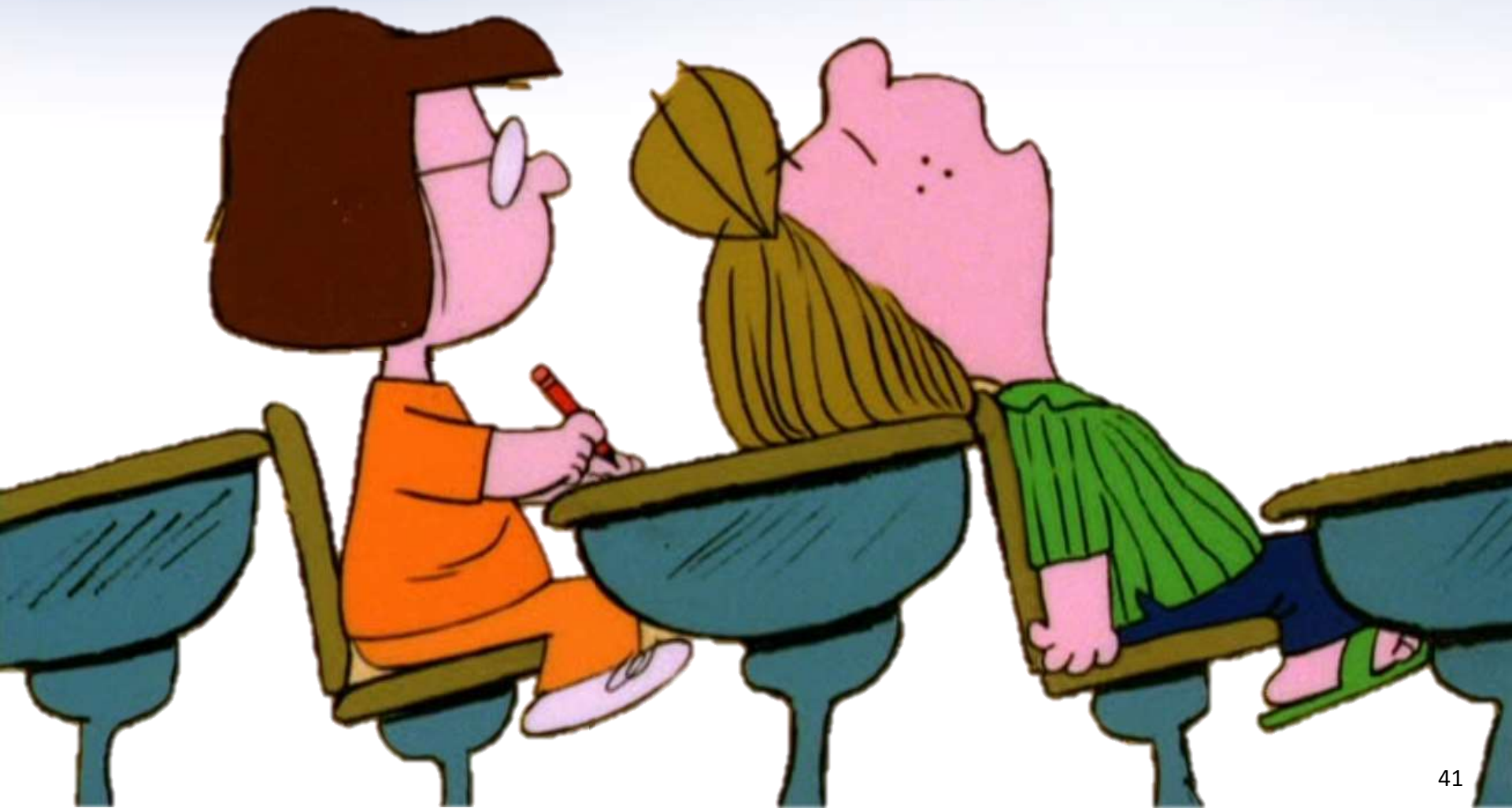
```
SELECT ?p (COUNT(?s) AS ?count) WHERE {  
  ?s ?p ?o .  
  filter (strstarts(str(?p), "http://dbpedia.org/ontology"))  
} GROUP BY ?p ORDER BY DESC(?count)
```

Más datasets

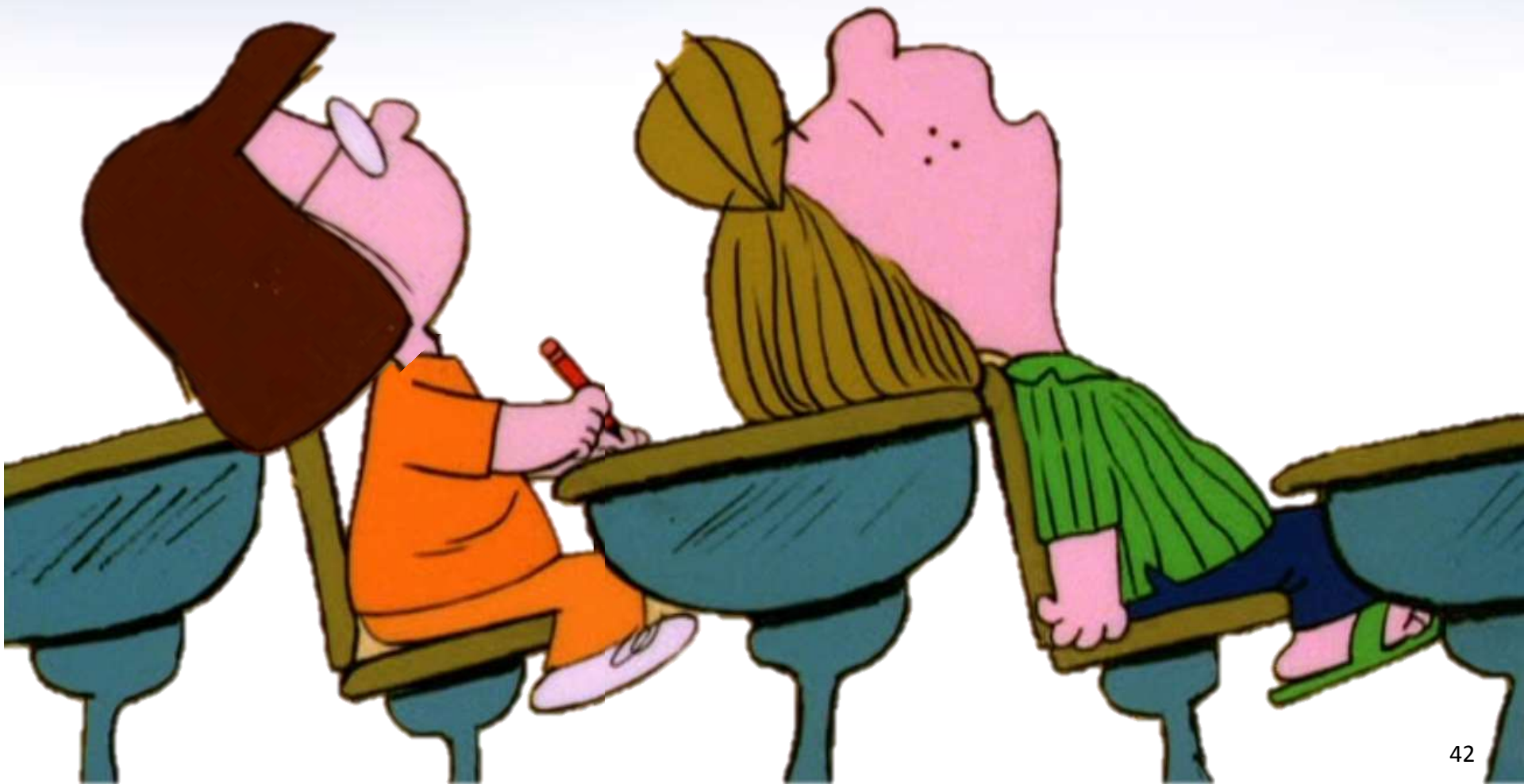
- The Data hub
 - Filtra por
 - Tags
 - lod
 - format-rdf
 - Formats
 - api/sparql

The screenshot shows the DataHub website interface. The top navigation bar includes the DataHub logo, a search bar, and links for 'Datasets', 'Organizations', 'About', 'Blog', and 'Help'. The 'Datasets' link is highlighted with a red box. Below the navigation bar, the page title is 'DataHub / Datasets'. On the left sidebar, there is a 'Tags' section with a search bar and a list of tags. The 'lod (1085)' tag is highlighted with a red box. The main content area shows '1,085 datasets found' and a list of datasets, including 'WordNet 3.0 (VU Amsterdam)' and 'provenanceweb'.

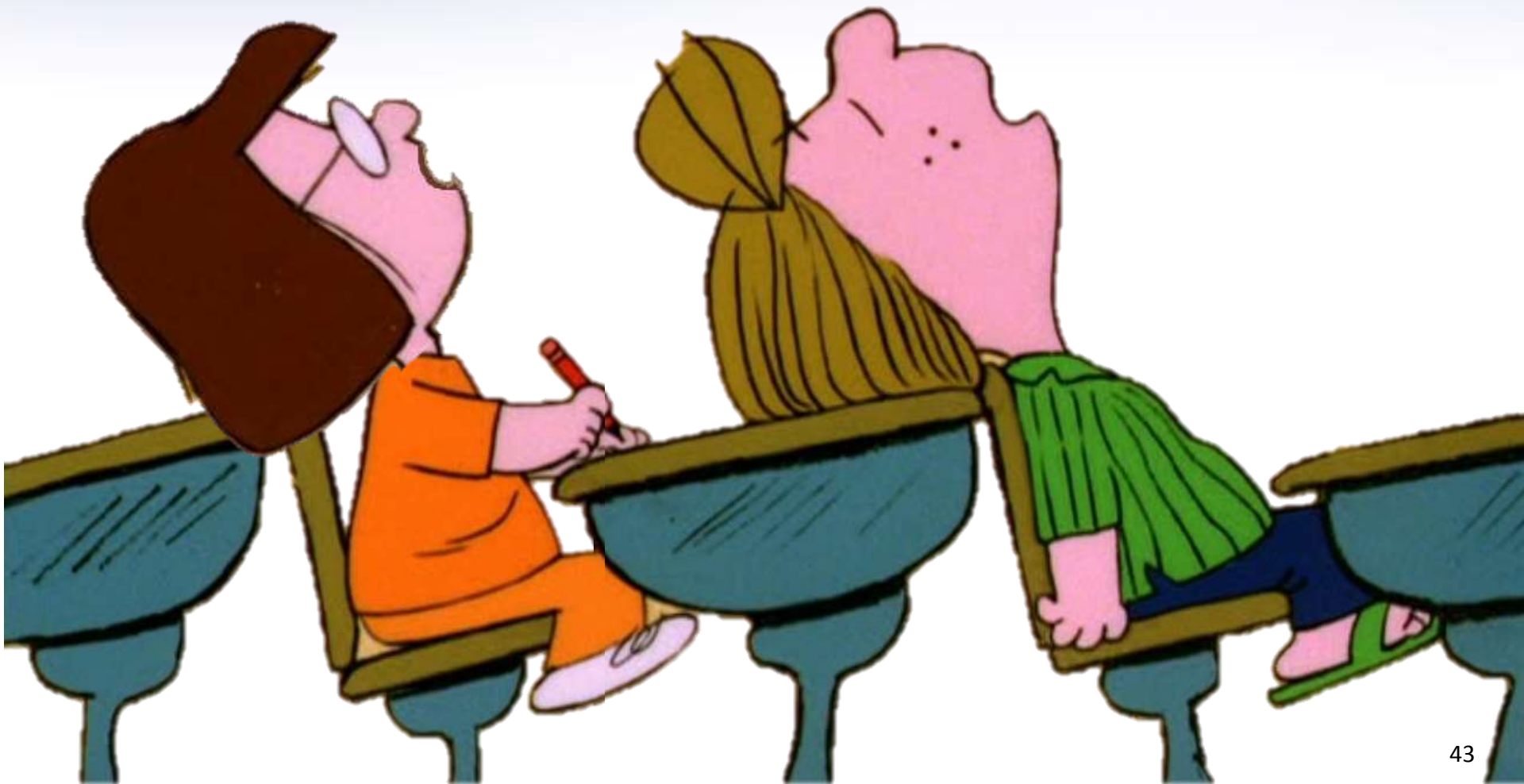
Hasta aquí... ¿bien?...



Hasta aquí... ¿bien?...

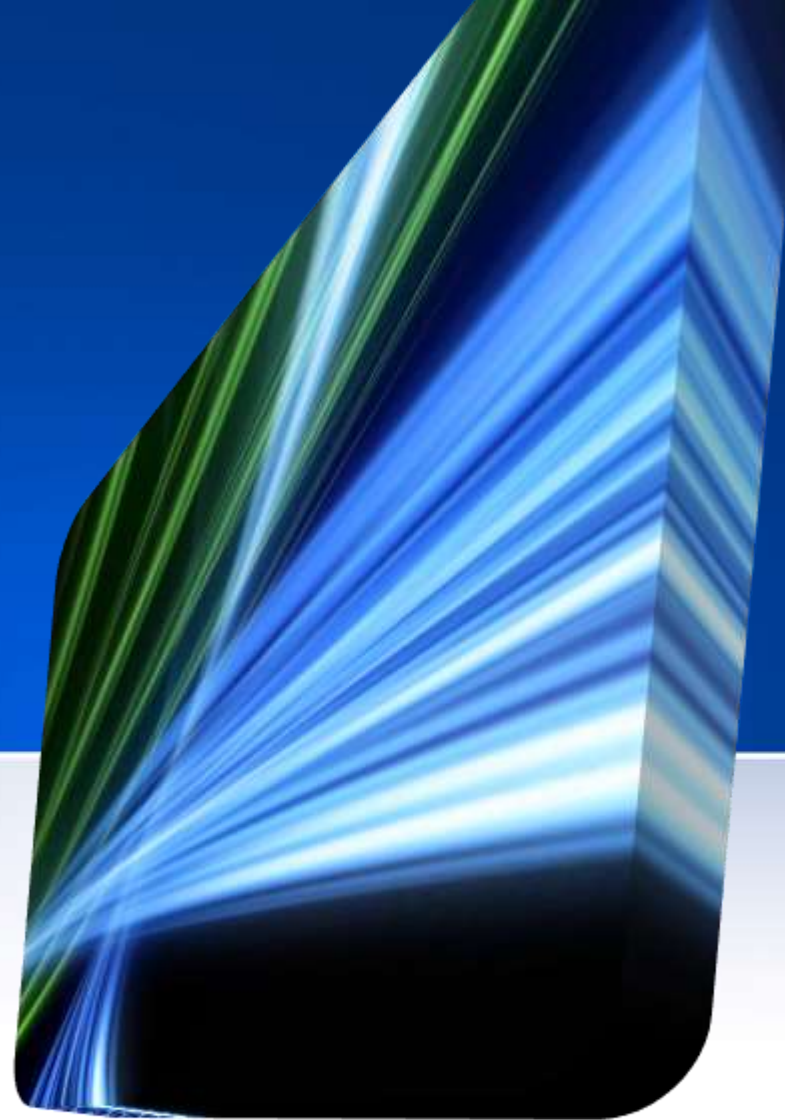


Hasta aquí... ¿bien?...



¡¡Por fin!!

USANDO R



Package SPARQL

```
library(SPARQL)
endpoint <- "http://es.dbpedia.org/sparql"
query <-
'
PREFIX dcterms: <http://purl.org/dc/terms/>
SELECT ?torero ?cantante WHERE{
  ?torero rdf:type dbpedia-owl:BullFighter .
  ?torero dbpedia-owl:spouse ?cantante .
  ?cantante dcterms:subject
<http://es.dbpedia.org/resource/Categoría:Cantantes_de_coplas>
}
'

reslist <- SPARQL(endpoint,query)
df <- reslist$results
df
```

	torero	cantante
1	<http://es.dbpedia.org/resource/Curro_Romero>	<http://es.dbpedia.org/resource/Concha_Márquez_Piquer>
2	<http://es.dbpedia.org/resource/Francisco_Rivera>	<http://es.dbpedia.org/resource/Isabel_Pantoja>
3	<http://es.dbpedia.org/resource/José_Ortega_Cano>	<http://es.dbpedia.org/resource/Rocío_Jurado>

Package SPARQL



- Uso de argumento **ns**
 - Para reducir tamaño de resultado

```
#continuación...
nss <- c("esdbres", "http://es.dbpedia.org/resource/")
reslist <- SPARQL(endpoint, query, ns = nss)
df <- reslist$results
df
```

```
           torero           cantante
1  esdbres:Curro_Romero esdbres:Concha_Márquez_Piquer
2 esdbres:Francisco_Rivera esdbres:Isabel_Pantoja
3 esdbres:José_Ortega_Cano esdbres:Rocío_Jurado
```

Múltiples maneras de preguntar



- Científicos españoles

Forma1

resultado: 66 científicos

```
SELECT ?person WHERE{
  ?person dcterms:subject
<http://es.dbpedia.org/resource/Categoría:Científicos_de_España>
}
```

Forma2

resultado: 143 científicos

```
PREFIX esdbpr: <http://es.dbpedia.org/resource/>
SELECT ?person WHERE{
  ?person rdf:type          dbpedia-owl:Scientist .
  ?person dbpedia-owl:country esdbpr:España .
}
```

Múltiples maneras de preguntar



- Desde SPARQL EP: Científicos españoles
 - UNION

resultado: 209 = 66 + 143

```
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX esdbpr: <http://es.dbpedia.org/resource/>
SELECT ?person WHERE {
  {
    ?person dcterms:subject
    <http://es.dbpedia.org/resource/Categoría:Científicos_de_España>
  }
  UNION
  {
    ?person rdf:type dbpedia-owl:Scientist .
    ?person dbpedia-owl:country esdbpr:España .
  }
}
```

Múltiples maneras de preguntar



- Desde SPARQL EP: Científicos españoles
 - UNION

resultado: 200
(había 9 duplicados)

```
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX esdbpr: <http://es.dbpedia.org/resource/>
SELECT DISTINCT ?person WHERE{
  {
    ?person dcterms:subject
    <http://es.dbpedia.org/resource/Categoría:Científicos_de_España>
  }
  UNION
  {
    ?person rdf:type dbpedia-owl:Scientist .
    ?person dbpedia-owl:country esdbpr:España .
  }
}
```


Múltiples maneras de preguntar



- Desde R: Científicos españoles

```
library(SPARQL)
endpoint <- "http://es.dbpedia.org/sparql"
nss <- c("esdbres", "http://es.dbpedia.org/resource/")
queryF1 <-
'
SELECT ?person WHERE{
  ?person dcterms:subject
    <http://es.dbpedia.org/resource/Categoría:Científicos_de_España>
}
'
```

```
queryF2 <-
'
PREFIX esdbpr: <http://es.dbpedia.org/resource/>
SELECT ?person WHERE{
  ?person rdf:type          dbpedia-owl:Scientist .
  ?person dbpedia-owl:country esdbpr:España .
}
'
```

```
reslistF1 <- SPARQL(endpoint,queryF1, ns = nss)
reslistF2 <- SPARQL(endpoint,queryF2, ns = nss)
```

#Devuelve los 9 comunes a las dos queries

```
intersect (t(reslistF1$results), #Si devuelve una única columna "se lía" y necesita t()
          t(reslistF2$results))
```

```
[1] "esdbres:Severo_Ochoa"
[2] "esdbres:David_Vázquez_Martínez"
[3] "esdbres:Simón_Méndez_Ferrer"
[4] "esdbres:Ginés_Morata"
[5] "esdbres:Carlos_López_Otín"
[6] "esdbres:Juan_Modolell_Mainou"
[7] "esdbres:Cayetano_López"
[8] "esdbres:Julio_Rodríguez_Villanueva"
[9] "esdbres:Miguel_Ángel_Mayer"
```



Siguiendo la pista

- Ejemplo: Políticos
- Veo la página de Rajoy en Wikipedia
- Veo qué nombre de recurso tiene
 - “Mariano_Rajoy”
- Busco sus datos en esDBpedia
 - http://es.dbpedia.org/resource/Mariano_Rajoy
 - Veo properties
 - occupation
 - party
 - ...

Property	Value
dbpedia-owl:abstract	<ul style="list-style-type: none"> ▪ Mariano Rajoy Brey (Sant... en el 17.º Congreso celeb... partido por mayoría absol... María Aznar en diversas c...
dbpedia-owl:occupation	<ul style="list-style-type: none"> ▪ dbpedia:Político
dbpedia-owl:party	<ul style="list-style-type: none"> ▪ dbpedia:Partido_Popular
dbpedia-owl:profession	<ul style="list-style-type: none"> ▪ dbpedia:Registro_de_la_p...
dbpedia-owl:religion	<ul style="list-style-type: none"> ▪ dbpedia:Iglesia_católica
dbpedia-owl:successor	<ul style="list-style-type: none"> ▪ dbpedia:Rodrigo_Rato ▪ dbpedia:Ángel_Acebes ▪ dbpedia:Juan_José_Lucas



Siguiendo la pista

- Ejemplo: Políticos
- Obteniendo los datos de Rajoy en esDBpedia

```
library(SPARQL)
endpoint <- "http://es.dbpedia.org/sparql"
nss <- c("esdbres", "http://es.dbpedia.org/resource/")
query <-
  '
  PREFIX dbo: <http://dbpedia.org/ontology/>
  PREFIX esdbp: <http://es.dbpedia.org/property/>
  PREFIX esdbr: <http://es.dbpedia.org/resource/>
  select * where{
    esdbr:Mariano_Rajoy ?p ?v
  }
  '
reslist <- SPARQL(endpoint,query, ns = nss) #Salen 550 datos
head(reslist$results)
```

```

1 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> p <http://www.w3.org/2002/07/owl#Thing> v
2 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://wikidata.dbpedia.org/resource/Q215627>
3 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://schema.org/Person>
4 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person>
5 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://wikidata.dbpedia.org/resource/Q5>
6 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#Agent>
```



Siguiendo la pista

- Ejemplo: Políticos
- Políticos y su partido político

```
library(SPARQL)
endpoint <- "http://es.dbpedia.org/sparql"
nss <- c("esdbres", "http://es.dbpedia.org/resource/")
query <-
  '
  PREFIX dbo: <http://dbpedia.org/ontology/>
  PREFIX esdbp: <http://es.dbpedia.org/property/>
  PREFIX esdbr: <http://es.dbpedia.org/resource/>
  select ?uri ?par where{
    ?uri dbo:occupation esdbr:Político .
    ?uri dbo:party ?par
  }
  ORDER BY ?par
  '
reslist <- SPARQL(endpoint,query, ns = nss)
nrow(reslist$results)
#Salen 1514 de todas partes del mundo
```




Siguiendo la pista

- Ejemplo: Políticos
- ¿Cómo saco los políticos españoles?

???

- Ver [preguntas abierta 3](#) en wiki esDBpedia



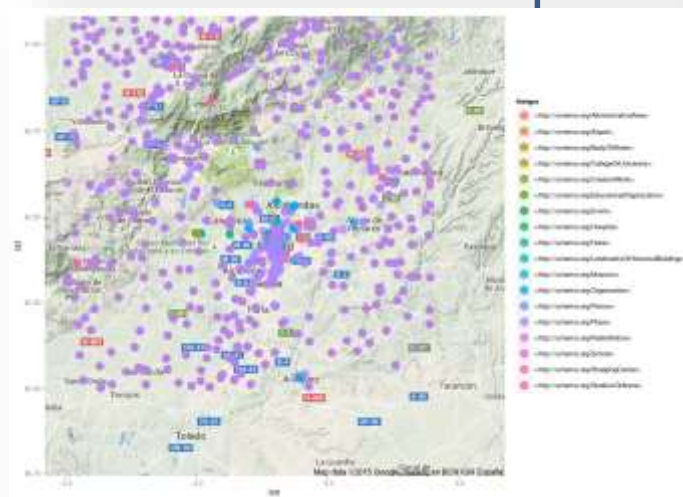
Coordenadas de recursos

```

library(ggmap) #Para geocode(), get_map(). Carga ggplot2
library(SPARQL)
endpoint <- "http://es.dbpedia.org/sparql"
query <-
'SELECT * WHERE {
  ?uri geo:lat ?lat .
  ?uri geo:long ?lon .
  ?uri rdf:type ?thetype .
  FILTER ( (?lat > 40.0 && ?lat < 41.15) &&
           (?lon > -4.5 && ?lon < -3.1)
           && regex(?thetype, "^http://schema.org")
        )
}
'

reslist <- SPARQL(endpoint,query)
df <- reslist$results
df$thetype <- factor(df$thetype)
map.center <- geocode("Madrid, Spain")
map <- get_map(c(lon=map.center$lon, lat=map.center$lat),
               source="google", zoom=9)
ggmap(map) + geom_point(data=df,
                        aes(x=lon, y=lat, colour=thetype, position="dodge"),
                        size=6, alpha=0.8
                        )

```



Packages rrdf y rrdflibs



```
library("rrdf")
#Descarga de la ontología DBpedia
db2014URL <- "http://data.dws.informatik.uni-mannheim.de/dbpedia/2014/dbpedia_2014.owl.bz2"
tempBZ2 <- tempfile()
download.file(db2014URL, tempBZ2) #Lo guarda en tempBZ2
con <- bzfile(tempBZ2) #descomprimo
lines <- readLines(con) #Tengo las líneas en memoria
unlink(tempBZ2)
#Guardo en un fichero (requerido por load.rdf())
tempOWL <- tempfile()
write(lines, file=tempOWL )
model <- load.rdf(tempOWL, format="RDF/XML")
unlink(tempOWL)
#Ya tengo el model y todos los ficheros aux borrados
summarize.rdf(model)
#[1] "Number of triples: 29402"
```

Packages rrdf y rrdflibs

```
#Continuación...
#Clases con label
query <-
  'PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
  SELECT ?class ?classLabel WHERE {
    ?class rdf:type rdfs:Class.
    ?class rdfs:label ?classLabel.
  }'
#OJO! matrix con variables ?class y ?classLabel
res <- sparql.rdf(model, query)
#Lo paso a data.frame
df <- as.data.frame(res, stringsAsFactors = TRUE)
#Pongo TRUE para que sea mono el summary
summary(df)
```

	class	classLabel
http://dbpedia.org/ontology/Actor	: 13	Imdb : 5
http://dbpedia.org/ontology/Airport	: 12	album : 5
http://dbpedia.org/ontology/Person	: 12	ginkgo : 5
http://dbpedia.org/ontology/Planet	: 12	Gemeinde: 4
http://dbpedia.org/ontology/Automobile:	11	manga : 4
http://dbpedia.org/ontology/Book	: 11	single : 4
(Other)	:3449	(Other) :3493

Packages rrdf y rrdflibs



#Continuación...

nrow(df)

#En total hay 3520, pero muc

#Elimino duplicados

length(unique(df\$class))

#Ahora salen 683

#¿Cómo selecciono las labels en inglés?

????

	class	classLabel
http://dbpedia.org/ontology/Actor	: 13	Imdb : 5
http://dbpedia.org/ontology/Airport	: 12	album : 5
http://dbpedia.org/ontology/Person	: 12	ginkgo : 5
http://dbpedia.org/ontology/Planet	: 12	Gemeinde: 4
http://dbpedia.org/ontology/Automobile:	11	manga : 4
http://dbpedia.org/ontology/Book	: 11	single : 4
(Other)	:3449	(Other) :3493

Packages rrdf y rrdflibs



#Continuación...

#¿Cómo selecciono las labels en inglés?

```
query <-
```

```
'PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
SELECT ?class ?classLabel WHERE {
```

```
  ?class rdf:type rdfs:Class.
```

```
  ?class rdfs:label ?classLabel.
```

```
  FILTER(lang(?classLabel) = "en")
```

```
}'
```

```
res <- sparql.rdf(model, query)
```

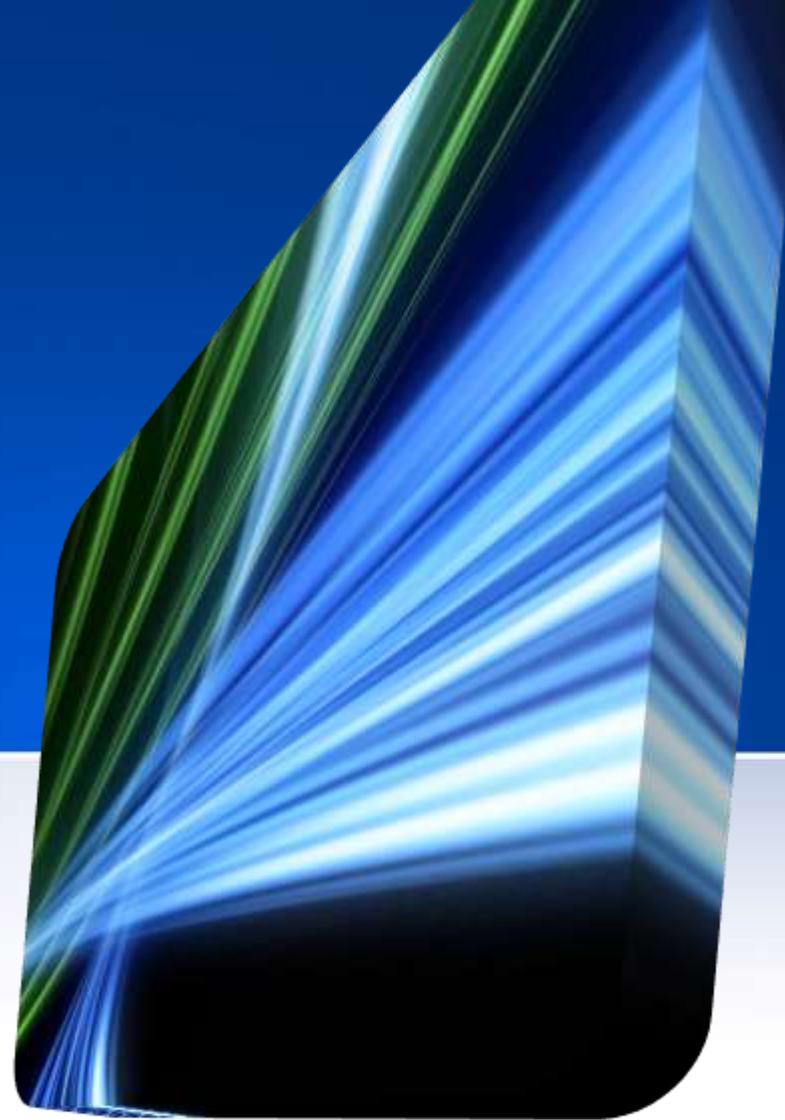
```
df <- as.data.frame(res, stringsAsFactors = TRUE)
```

```
summary(df) #Para ver que no hay multiplicidad
```

```
nrow(df) #Salen 683 classes
```

¿Nos echas una mano?

IDEAS PARA DESARROLLAR



Visualización



- De ontologías
 - Jerarquía de clases
 - Jerarquía de propiedades
- De datos
 - Grafos (enormes)



Más información

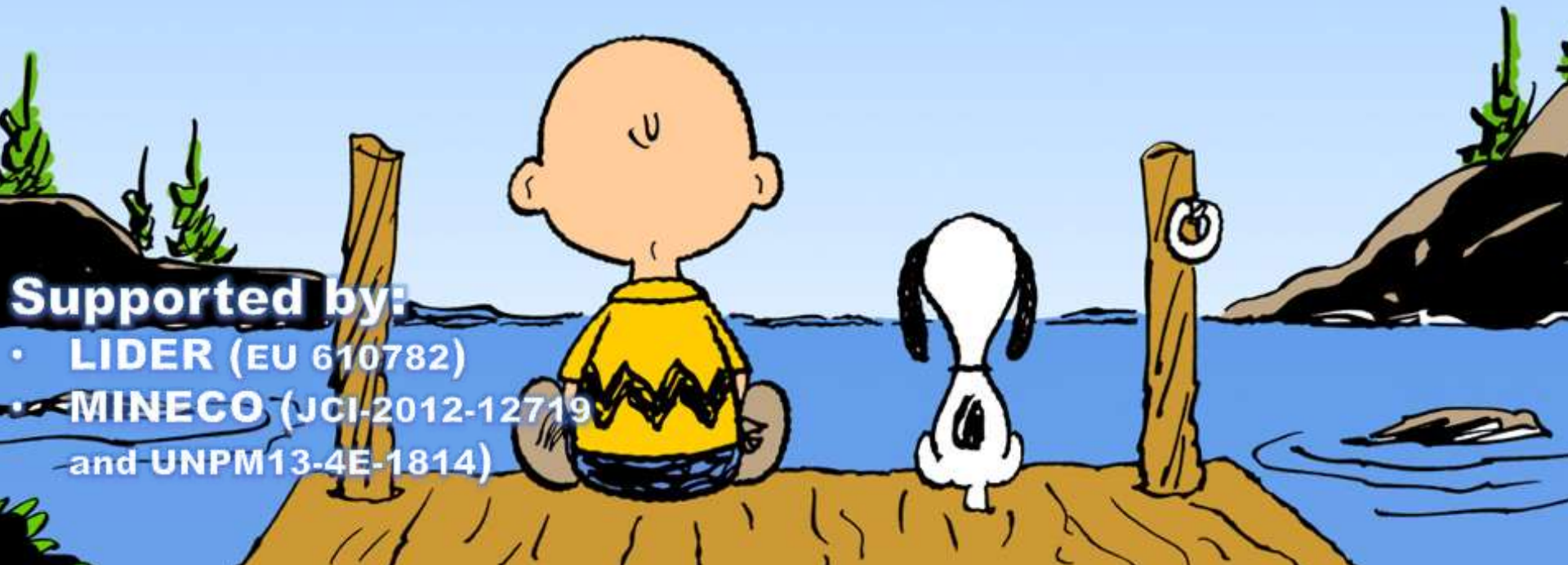
<http://es.dbpedia.org>

<http://linkeddata.es>

<http://www.oeg-upm.net>

Thanks a LOD!!

Mariano.Rico@upm.es
@marianorico



Supported by:

- LIDER (EU 610782)
- MINECO (JCI-2012-12719)
- and UNPM13-4E-1814)



Muchas gracias por vuestra atención

Mariano.Rico@upm.es

Supported by:

- **LIDER (EU 610782)**
- **MINECO (JCI-2012-12719
and UNPM13-4E-1814)**