Consultas en la Web Semántica: SPARQL

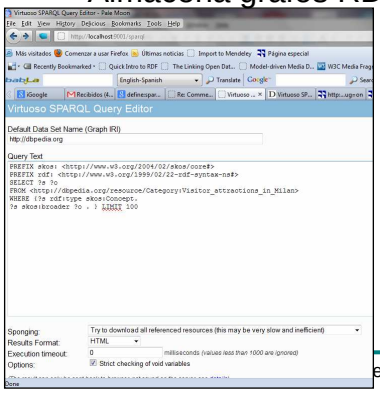
Grupo de Concepción de Sistemas de Información

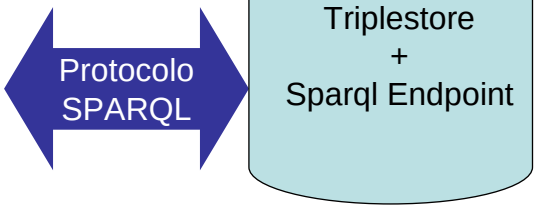
Qué es Sparql?

- SPARQL Protocol And RDF Query Language.
- Simple Protocol And RDF Query Language ([Della Valle & Ceri, 2011])
- Tiene dos partes:
  - Un lenguaje de consulta sobre RDF.
  - Un protocolo que describe cómo hacer las consultas y cómo recuperar los resultados.
- Hay 2 versiones:
  - 1.0 : Bastante implementada y recomendación W3C.
  - 1.1 : Es la recomendación más reciente (marzo 2013), pero bastante implementada también.

Un Poco de Arquitectura

- Triplestore:
  - Base de Datos que utiliza el modelo de datos RDF como nativo.
  - Almacena grafos RDF.





Anatomía de una Consulta

[Della Valle & Ceri, 2011]

Abreviaturas

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

select ?s ?p ?o
from <http://dbpedia.org>
where {
   ?s rdfs:label "Raiders of the Lost Ark"@en .
   ?s ?p ?o
}
order by ?s
limit 100
```

Resultados

Origen

Patrones de grafo

Modificadores

Abreviaturas, Resultados y Origen

- Abreviaturas
  - Pueden ser namespaces
  - Pueden ser cualquier dirección que convenga para abreviar en la consulta.
- Resultados.
  - Es la lista de las variables de las que se desea obtener el resultado
- Origen:
  - Es un Dataset: un grafo por defecto y cero o más grafos con nombre (uri).

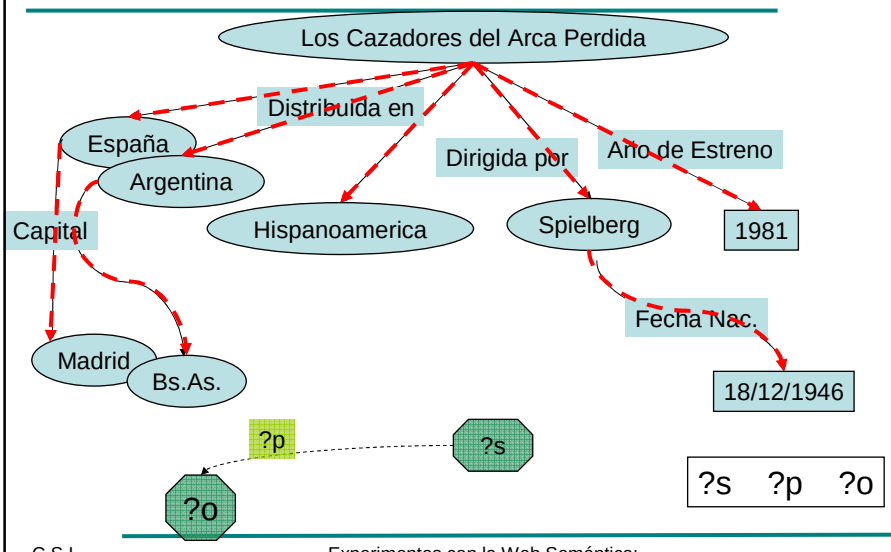
Graph Patterns

- SPARQL se basa en pattern matching contra los grafos.
  - Grafo para ejemplos:
    - http://dbpedia.org/resource/Category:Visitor_attractions_in_Milan
- Los patrones se escriben en Turtle con algunos agregados para construir patrones complejos.

BGP

- Basic Graph Pattern:
  - La forma más simple de patrones.
  - Describen un grafo, usando URIS, literales o variables en donde corresponda.
- El patrón más simple:
  - **?s ?p ?o .**
  - Coincide con cualquier terna.

BGP



Ejemplos de BGP

- **?s skos:broader dbpedia:Visitor_attractions_in_Milan .**
- Coincide con:

  category:Gardens_in_Milan
  category:Churches_in_Milan
  category:Piazzas_in_Milan
  category:Theatres_in_Milan
  category:Palaces_in_Milan
  category:Museums_in_Milan
  category:Parks_in_Milan

Ejemplos de BGP

- **?s ?p dbpedia:Visitor_attractions_in_Milan .**
- Coincide con las ternas anteriores y además con cualquier otra que tenga **dbpedia:Visitor_attractions_in_Milan** como objeto sin importar el predicado.
- Que da el siguiente patron?

  ?s rdf:type skos:Concept .
  ?s skos:broader ?o .

Ejemplos BGP

- Para verlo, se hace la consulta:

```
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?s ?o
FROM <http://dbpedia.org/data/
                Category:Visitor_attractions_in_Milan.rdf>
WHERE {?s rdf:type skos:Concept.
?s skos:broader ?o . }
```


Cómo se calculan los resultados?

- Un BGP devuelve un conjunto de ***soluciones***.
- Cada solución es una ***función parcial*** que asocia una variable con un término rdf del grafo en cuestión.
- De esta forma, luego se combinan o filtran esas funciones de acuerdo los nombres de las variables y la semántica de los operadores.

Cómo se calculan los Resultados?

- Si el patern es:
  ?s rdf:type skos:Concept .
  ?s skos:broader ?o .
- El resultado es un conjunto de funciones:
  μ_1[?s]=category:Visitor_attractions_in_Milan
  μ_1[?o]=category:Milan
  μ_2[?s]=category:Visitor_attractions_in_Milan
  μ_2[?o]=category:Visitor_attractions_in_Lombardy
  μ_3[?s]=category:Visitor_attractions_in_Milan
  μ_3[?o]=category:Visitor_attractions_in_Italy_by_city

Usando Literales

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?s
FROM <http://dbpedia.org/data/Milan_Cathedral.rdf>
WHERE { ?s rdfs:label “Milan Cathedral”. }

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?s
FROM <http://dbpedia.org/data/Milan_Cathedral.rdf>
WHERE { ?s rdfs:label “Milan Cathedral”@en. }
```

Usando Literales

```
PREFIX dbpprop: <http://dbpedia.org/property/>
SELECT ?s
FROM <http://dbpedia.org/data/Milan_Cathedral.rdf>
WHERE { ?s dbpprop:yearCompleted
   “1965”^^xsd:integer }

PREFIX dbpprop: <http://dbpedia.org/property/>
SELECT ?s
FROM <http://dbpedia.org/data/Milan_Cathedral.rdf>
WHERE { ?s dbpprop:yearCompleted 1965. }
```


Filtros

```
PREFIX dbpprop: <http://dbpedia.org/property/>
SELECT ?s
FROM <http://dbpedia.org/data/Milan_Cathedral.rdf>
WHERE {?s dbpprop:yearCompleted ?c.
FILTER(?c>1950)}
```

- El FILTER deja sólo las soluciones que cumplen con la condición.
- Hay muchas funciones para utilizar en un FILTER.

Funciones

- Ver:
  - http://www.w3.org/TR/sparql11-query/
    - #OperatorMapping

Datasets y Named Graphs

- Un **Dataset** tiene:
  - Un grafo por defecto
  - Cero o más Named Graphs.
- Un **Named Graph** es:
  - Un grafo con una uri asociada.

Graph

- Obtener (la URI) la clase de los teatros en milan

```
SELECT ?general ?particular
FROM <http://dbpedia.org/resource/
                Category:Visitor_attractions_in_Milan>
WHERE {?general rdf:type skos:Concept.
    ?particular skos:broader ?general .
   ?particular rdfs:label "Theatres in Milan"@en.
}
```


Graph

- Otro intento

```
SELECT ?general ?particular
FROM
   <http://dbpedia.org/resource/Category:Visitor_attr
   actions_in_Milan>
WHERE {   ?general rdf:type skos:Concept.
                ?particular skos:broader ?general .
GRAPH ?particular {
    ?particular rdfs:label "Theatres in Milan"@en.
    }
}
```

graph

- La primer consulta no devuelve datos, porque la terna con predicado rdfs:label está en un grafo distinto del que se está consultando.
- Para aplicar ese patrón de grafos a otro grafo, se usa la construcción **graph**

Optional

```
SELECT ?s ?label ?lat ?long
FROM
   <http://dbpedia.org/data/Category:Churches_in_Mi
   lan.rdf>
WHERE { ?s dterms:subject
   dbcat:Churches_in_Milan .
    ?s rdfs:label ?label .
    FILTER langMatches( lang(?label), “EN” ) .
    OPTIONAL {
        ?s geo:lat ?lat .
        ?s geo:long ?long
    }
}
```

Optional

- Si encuentran ternas que cumplan con el patrón entonces se recuperan datos.
- Si no se encuentran, el patrón devuelve verdadero.

Not Exists

```
SELECT ?s ?label
FROM
   <http://dbpedia.org/resource/Category:Churches_in_Milan>
WHERE { ?s dcterms:subject
   category:Churches_in_Milan .
   Graph ?s {
      ?s rdfs:label ?label .
      FILTER  NOT EXISTS {?s geo:lat [] } .
   }
}
```

Consultas Federadas

```
SELECT ?s  ?o
FROM
   <http://dbpedia.org/data/Category:Churches_in_Milan.rdf>
WHERE {
?s rdfs:label [].
SERVICE <http://dbpedia.org/sparql> {
  ?o dcterms:subject ?s .
}
}

LIMIT 30
```


Bibliografía

- Della Valle, E., & Ceri, S. (2011). Querying the Semantic Web: SPARQL. In J. Domingue, D. Fensel, & J. A. Hendler (Eds.), *Handbook of Semantic Web Technologies* (pp. 299–363). Springer Berlin Heidelberg. Retrieved from http://dx.doi.org/10.1007/978-3-540-92913-0_8
- W3C. (2012b). *SPARQL 1.1 Query Language* (W3C Working Draft). Retrieved from http://www.w3.org/TR/sparql11-query/
- W3C. (2011). *SPARQL 1.1 Federated Query* (W3C Working Draft). Retrieved from http://www.w3.org/TR/sparql11-federated-query/