

FONDO EUROPEO DE DESARROLLO REGIONAL (FEDER)

Una manera de hacer Europa

ASIO SPARQL Endpoint

Esteban Sota – Álvaro Palacios

RIAM Intelearning Lab – GNOSS

estebansota@gnoss.com – alvaropalacios@gnoss.com



HĒRCULES

Hércules. ASIO SPARQL Endpoint

- ☐ Reutilización de los datos.
- ☐ Estándares de la Web Semántica.
- ☐ RDF Store.
- ☐ ASIO SPARQL Endpoint.
- ☐ SPARQL.



Hércules ASIO. Reutilización de los datos



Hércules ASIO. Reutilización de los datos

ASIO proporciona dos interfaces de reutilización de los datos del grafo de conocimiento modelados por la Red de Ontologías Hércules ROH:

- ❑ **Linked Data Server.** Es el componente que permite la publicación de los datos cargados en el RDF Store como datos abiertos y enlazados (linked open data).
- ❑ **SPARQL Endpoint.** Permite a usuarios y administradores consultar los datos del grafo de conocimiento almacenados en el RDF Store, mediante el lenguaje y protocolo de consultas SPARQL.

Reutilización: Conocer ROH + SPARQL Queries + Linked Data Server

Hércules ASIO. Arquitectura Semántica. Linked Data Server

Linked Data en el RDF Store las siguientes:

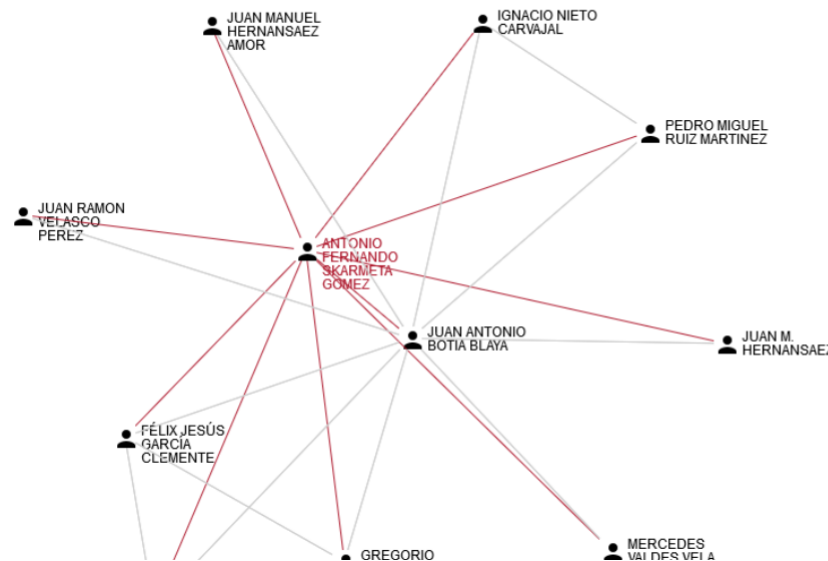
- ☐ Proporciona la presencia
- ☐ Proporciona Cumple
- ☐ Cuenta con

| | | |
|---|---|---|
| http://graph.um.es/res/article/bae9443b-9d51-43c8-843b-ef56c664b79c | Towards semantic web-based management of security services | http://purl.org/ron/mirror/bibo#AcademicArticle |
| http://graph.um.es/res/article/6bcd4117-6a6a-4661-b165-5a7b9858aa50 | Towards semantic-aware management of security services in GT4 | http://purl.org/roh/mirror/bibo#AcademicArticle |
| http://graph.um.es/res/article/6e10fef0-ae84-43f9-bfc8-b500c16e6316 | Toward a Framework for the Specification of Hybrid Fuzzy Modeling | http://purl.org/roh/mirror/bibo#AcademicArticle |

Mostrando página 1 de 3

Anterior 1 2 3

Coautores



se cargados
vicio tiene
e incluyen
máquinas.
nombre.

Hércules ASIO. Arquitectura Semántica. SPARQL Endpoint

Además de Linked Data Server, ASIO cuenta con un punto SPARQL (*endpoint*) que permite a usuarios y administradores consultar los datos del grafo de conocimiento almacenados en el RDF Store.

- ❑ **SPARQL**: SPARQL Protocol and RDF Query Language.
- ❑ [SPARQL Versión 1.1 de 2013](#) (la versión 1 es de 2008).
- ❑ **Protocolo de consulta y lenguaje de interrogación** para grafos RDF, normalizado inicialmente por el RDF Data Access Working Group del World Wide Web Consortium (W3C) y por el SPARQL Working Group en su version 1.1
- ❑ *Endpoint* de **sólo lectura para usuarios públicos**, sin acceso a los datos privados.

Reutilización: Conocer ROH + SPARQL Queries + Linked Data Server

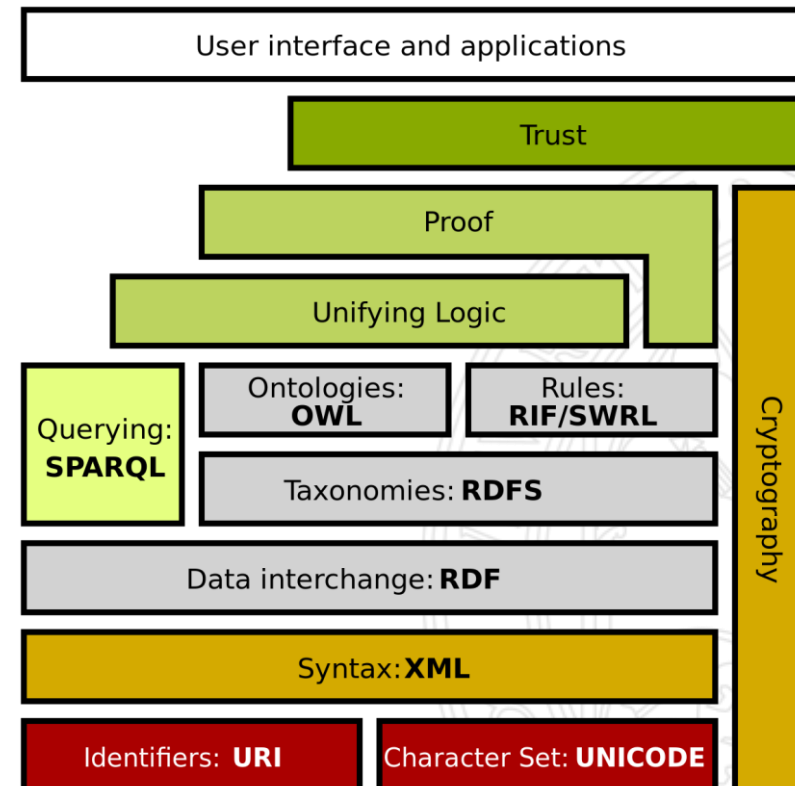
Hércules ASIO. Estándares de la Web Semántica



Pila de la Web Semántica

La Web Semántica está compuesta de:

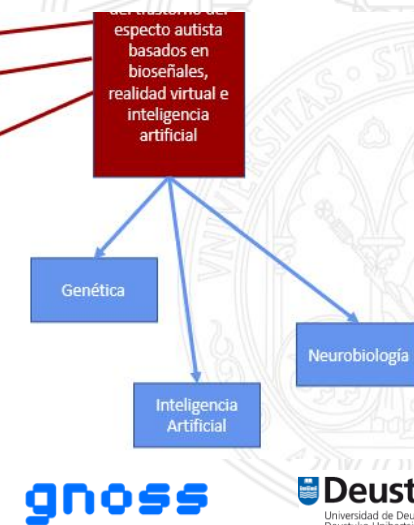
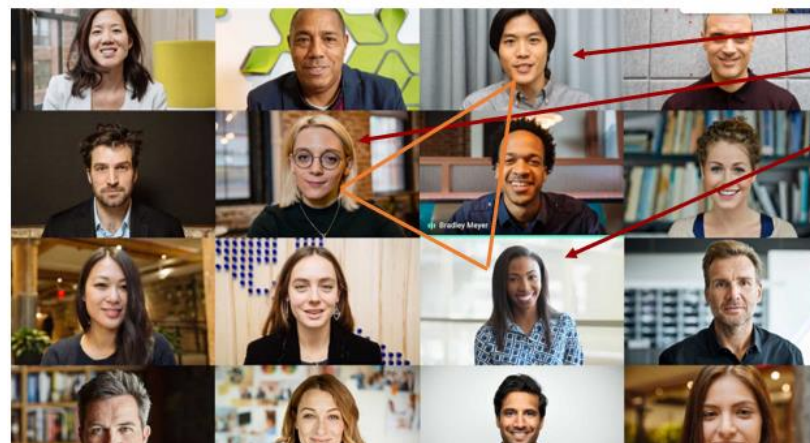
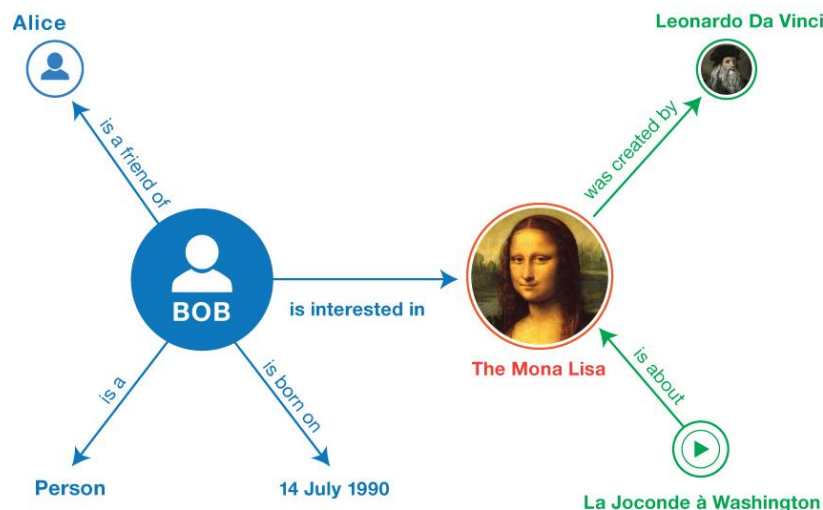
- ☐ Tecnologías web de hipertexto:
 - ☐ XML: sintaxis para documentos estructurados
 - ☐ XML Schema: restringe la estructura de documentos XML
- ☐ Tecnologías web semánticas:
 - ☐ RDF es un modelo de datos que hace referencia a objetos y sus relaciones
 - ☐ RDF Schema: vocabulario para definir propiedades y clases de recursos RDF
 - ☐ OWL: añade más vocabulario que RDFS, permite establecer restricciones adicionales (cardinalidad), restricciones de los valores o las características de las propiedades (transitividad)
 - ☐ Se basa en la lógica de la descripción y, por tanto, aporta capacidad de razonamiento a la web semántica.
 - ☐ **SPARQL**: es un lenguaje de consulta RDF. Necesario para recuperar información para las aplicaciones de la web semántica



RDF. Resource Description Framework

RDF identifica conceptos usando identificadores Web (URIs), y describe recursos con propiedades y valores de las mismas. Definiciones:

- ❑ Un Recurso es cualquier cosa que puede tener un URI, como por ejemplo "http://www.w3schools.com/RDF" o "http://graph.um.es/res/**researcher**/0000-0001-8055-6823"
- ❑ Una Propiedad es un Recurso que tiene un nombre, como "**author**" o "**webpage**"
- ❑ Un Valor de propiedad es el valor de una Propiedad, tal como "**Diego López de Ipiña**" o "http://www.w3schools.com" (un valor de propiedad puede corresponder a un recurso).



RDF. Resource Description Framework

RDF identifica conceptos usando identificadores Web (URIs), y describe recursos con propiedades y valores de las mismas. Definiciones:

- ☐ Un **grafo RDF** crea una web de conceptos distribuidos:
 - ☐ Realiza **aserciones** sobre relaciones lógicas entre entidades
 - ☐ La información en RDF puede **enlazarse con grafos** en otros lugares
 - ☐ Mediante software se pueden **realizar inferencias**
 - ☐ Existen **lenguajes de consulta** sobre triple stores como **SPARQL**
- ☐ Mediante RDF hacemos que la información sea **procesable por máquinas**
 - ☐ **Agentes software** pueden guardar, intercambiar y utilizar metadatos sobre recursos en la web
- ☐ **Ontología**:
 - ☐ jerarquía de términos a utilizar en etiquetado de recursos
 - ☐ formalización de los metadatos de un dominio de conocimiento

FONDO EUROPEO DE DESARROLLO REGIONAL (FEDER)

Una manera de hacer Europa

Hércules ASIO. RDF Store



Hércules ASIO. RDF Store

El RDF Store es el componente de la arquitectura que **almacena el grafo de conocimiento de ASIO en formato RDF**. Dentro de ASIO actúa como un servidor de datos que responde a consultas SPARQL, por lo que también puede ser nombrado como **Servidor RDF o Servidor SPARQL**.

El RDF Store proporciona el servicio de datos y consultas al Linked Data Server y al SPARQL Endpoint.

En el desarrollo de ASIO se ha realizado un *benchmark* para que cada universidad pueda seleccionar la solución que más le convenga. Uno de los requisitos obligatorios definido por el *benchmark* es que debe **cumplir el estándar SPARQL 1.1**, en particular lo referente a actualizaciones de datos.

Hércules. RDF Store Benchmark

[TripleStore Benchmark deliverable report](#) (Mathieu D'Aquin & Alessandro Adamou):

- ☐ Functionalities.
- ☐ Performance and scalability.
- ☐ Management and maintenance.

Hércules. RDF Store Benchmark

[TripleStore Benchmark deliverable report](#) (Mathieu D'Aquin & Alessandro Adamou):

- ☐ Functionalities.
 - ☐ Base handling of RDF and SPARQL
 - ☐ Extensions
 - ☐ Security
- ☐ Performance and scalability.
- ☐ Management and maintenance.

Hércules. RDF Store Benchmark

[TripleStore Benchmark deliverable report](#) (Mathieu D'Aquin & Alessandro Adamou):

- ☐ Functionalities.
- ☐ Performance and scalability.
 - ☐ Query response time.
 - ☐ Loading and update time.
 - ☐ Memory footprint.
 - ☐ Robustness and Scaling.
- ☐ Management and maintenance.

Hércules. RDF Store Benchmark

[TripleStore Benchmark deliverable report](#) (Mathieu D'Aquin & Alessandro Adamou):

- ☐ Functionalities.
- ☐ Performance and scalability.
- ☐ Management and maintenance.
 - ☐ Costs and rights.
 - ☐ Ease of deployment.
 - ☐ Administration, updates and maintenance.

Hércules ASIO. RDF Store Benchmark

RDF Store analizados:

- ☐ Apache Jena.
- ☐ Virtuoso.
- ☐ AllegroGraph.
- ☐ Stardog.
- ☐ Corese.
- ☐ Blazegraph.
- ☐ Amazon Neptune.
- ☐ Neo4J.



Hércules ASIO. RDF Store Benchmark

Benchmark accesible en: <http://herc-as-front-desata.atica.um.es/benchmark>

HERCULES BENCHMARK

StoresBenchmarkUser GuideAbout ASIO

Hercules triple store ranking

| | |
|---|----------------|
| 1 | Apache Jena |
| 2 | Virtuoso |
| 3 | AllegroGraph |
| 4 | Stardog |
| 5 | Corese |
| 6 | Blazegraph |
| 7 | Amazon Neptune |
| 8 | Neo4J |

Not assessed

Assessment of the following triple stores is underway.

| | |
|---|-----------|
| 1 | 4store |
| 2 | AnzoGraph |

HERCULES BENCHMARK

StoresBenchmarkUser GuideAbout ASIO

Hercules triple store benchmarking framework

Click on a category name to see its details

Functionalities

Criteria concerning the satisfaction of typical functional requirements

Linked Data Platform (LDP)

Linked Data Platform (LDP) defines a set of rules for HTTP architecture for read-write Linked Data on the web. A score of 5 for full compliance with the specification.

| CRITERION | DESCRIPTION |
|--|--|
| Support for the management of LDP RDF resources | The system support the management of LDP RDF resources according to the specification. |
| Support for the management of LDP Non-RDF resources | The system support the management of LDP Non-RDF resources according to the specification. |
| Support for HTTP read methods (GET, HEAD and OPTIONS) | The system support the use of HTTP read methods (GET, HEAD and OPTIONS) gets a score of 5 for full compliance with the specification. |
| Support for HTTP write methods (POST, PUT, DELETE and PATCH) | The system support the use of HTTP write methods (POST, PUT, DELETE and PATCH) gets a score of 5 for full compliance with the specification. |

Base handling of RDF and SPARQL

SPARQL update compliance

Provide SPARQL Update endpoints compliant with the SPARQL Update specification. Other approaches can be used to update the data.

Triple store ranking for this criterion

| TRIPLE STORE | SCORE | JUSTIFICATION |
|----------------|-------|--|
| Amazon Neptune | 5 | SPARQL update provided |
| Apache Jena | 5 | SPARQL update compliance through the SPARQL Update specification |
| Blazegraph | 5 | Compliant with SPARQL Update |
| Stardog | 5 | Support SPARQL update query and endpoints |
| Virtuoso | 5 | Appears compatible, with example queries |
| Corese | 4 | Homepage mentions SPARQL update |
| AllegroGraph | 2 | Update queries available, but no endpoints |
| Neo4J | 0 | No mention of SPARQL update in documentation |

Virtuoso Universal Server

<http://data.datascienceinstitute.ie/software/virtuoso>

Click on a category name to see its details

Functionalities

Criteria concerning the satisfaction of typical functional requirements of triple stores

3.16

Management and maintenance

Criteria concerning the manageability of triple stores over time

3.27

Performance and scalability

Criteria concerning the satisfaction of typical performance requirements of triple stores

3.89

Query response time

| CRITERION | SCORE | JUSTIFICATION | PRIMARY SOURCE |
|----------------------------------|-------|------------------------------------|----------------|
| Tens of thousands of triples | 5.0 | Assumed very good results on 10KT | |
| Hundreds of thousands of triples | 5.0 | Assumed very good results on 100KT | |
| Millions of triples | 5.0 | Very good results on 1MT | |
| Tens of millions of triples | 5.0 | Very good results on 10MT | |
| Hundreds of millions of triples | 5.0 | Very good results on 100MT | |

FONDO EUROPEO DE DESARROLLO REGIONAL (FEDER)

Una manera de hacer Europa

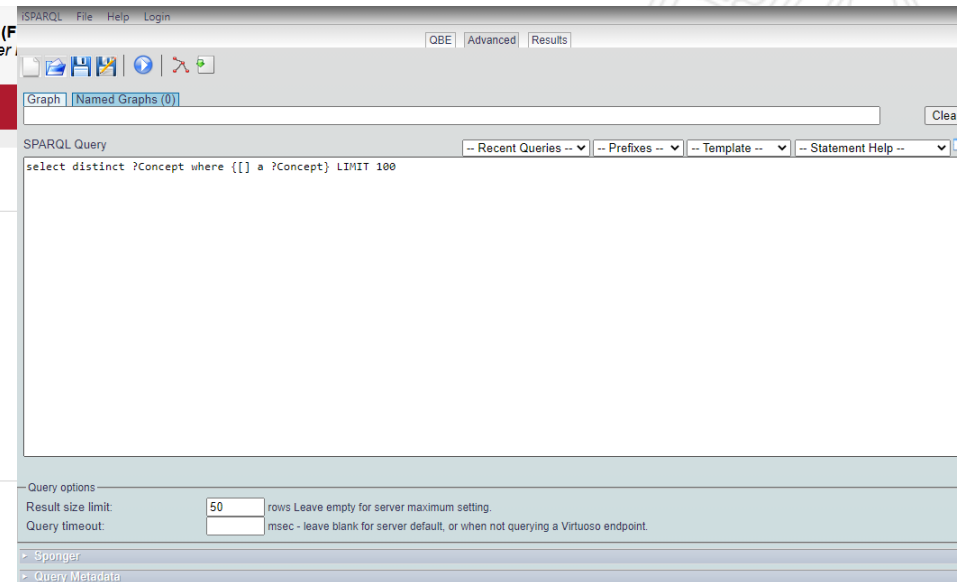
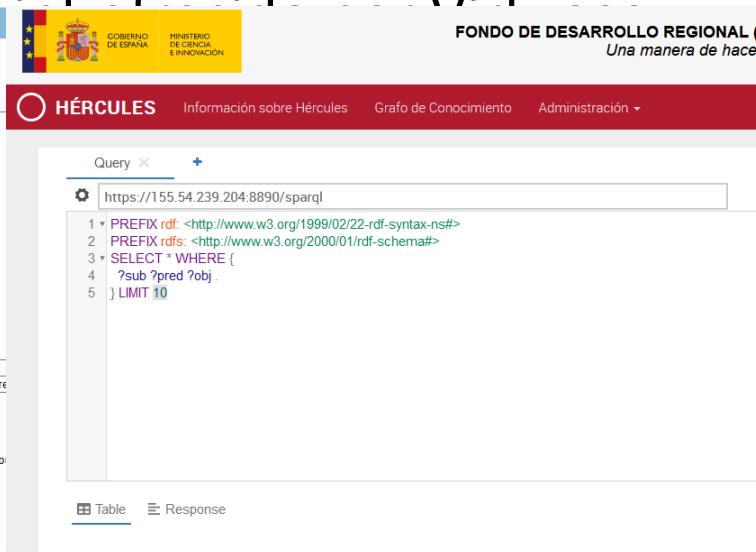
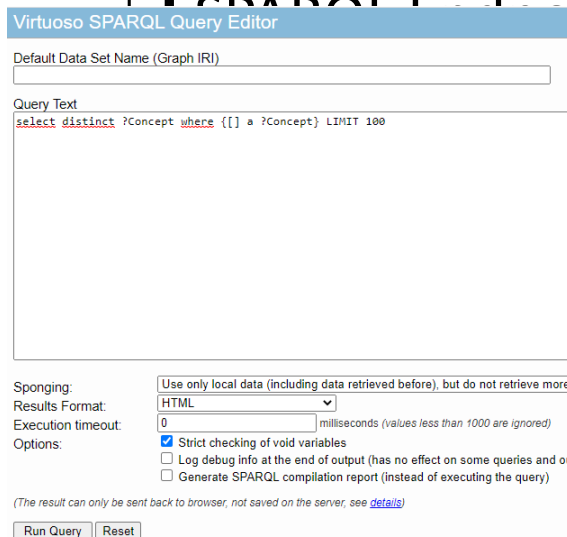
Hércules ASIO. SPARQL Endpoint



Hércules. SPARQL Endpoint

ASIO cuenta con un SPARQL Endpoint de acceso público, que:

- ☐ Permite consultas de lectura sobre los datos públicos del SGI.
- ☐ Se pueden configurar limitaciones de uso para evitar saturaciones y ataques.
- ☐ Disponible en varias URLs (pendiente SSL):



Hércules. SPARQL Endpoint

ASIO cuenta con un SPARQL Endpoint de acceso público, que es accesible:

- ☐ Por personas, en un interfaz web (como hemos visto en la diapo anterior)
- ☐ Por máquinas, mediante peticiones HTTP de tipo GET o POST. Por ejemplo, con una petición GET con un comando cURL:

```
curl
```

```
"http://server/sparql?query=SELECT%20*%20WHERE%20%7B%3Fs%20%3Fp%20%3Fo%7D%20LIMIT%208"
```

La consulta enviada es:

```
SELECT * WHERE {?s ?p ?o} LIMIT 8
```

FONDO EUROPEO DE DESARROLLO REGIONAL (FEDER)

Una manera de hacer Europa

Hércules ASIO. SPARQL



SPARQL. SPARQL Protocol and RDF Query Language

Es un Protocolo de consulta y lenguaje de interrogación para grafos RDF, normalizado inicialmente por el RDF Data Access Working Group del World Wide Web Consortium (W3C) y por el SPARQL Working Group en su version 1.1

- ❑ SPARQL (<https://www.w3.org/TR/sparql11-overview/>) permite la consulta de grafos RDF a través de un lenguaje sencillo
- ❑ SPARQL es idóneo para extraer y consultar información mantenida por aplicaciones, servicios o repositorios ad-hoc de terceras partes expresados en RDF
- ❑ Elementos destacados:
 - ❑ **Lenguaje de consultas** para seleccionar o actualizar datos.
 - ❑ Mecanismo para **transmitir una consulta** a un servicio de procesamiento de consultas remoto
 - ❑ Formato **XML, CSV, TSV y JSON** para obtención de los resultados

SPARQL. Ejemplo de consulta SPARQL

Obtención de los triples de una investigadora concreta:

```
select * from <http://graph.um.es/graph/um cvn>  
where { <http://graph.um.es/res/person/f6544839-0e1e-4e4b-bc5d-70199198a50f> ?p ?o }
```

Todos los datos

Grafo origen de los triples

URI de la investigadora de la que se quieren recuperar los datos

SPARQL. Ejemplo de consulta SPARQL

Obtención de las entidades de las que un investigador concreto es objeto:

```
select ?s ?rdfType from <http://graph.um.es/graph/um_cvn>
where {
  ?s ?p <http://graph.um.es/res/person/f6544839-0e1e-4e4b-bc5d-70199198a50f>.
  ?s rdf:type ?rdfType.
}
```

Devuelve los URIs y tipos de las entidades de las que un investigador es objeto. Por ejemplo, publicaciones de las que es “author”

Grafo origen de los triples

URI del investigador como objeto de otras entidades

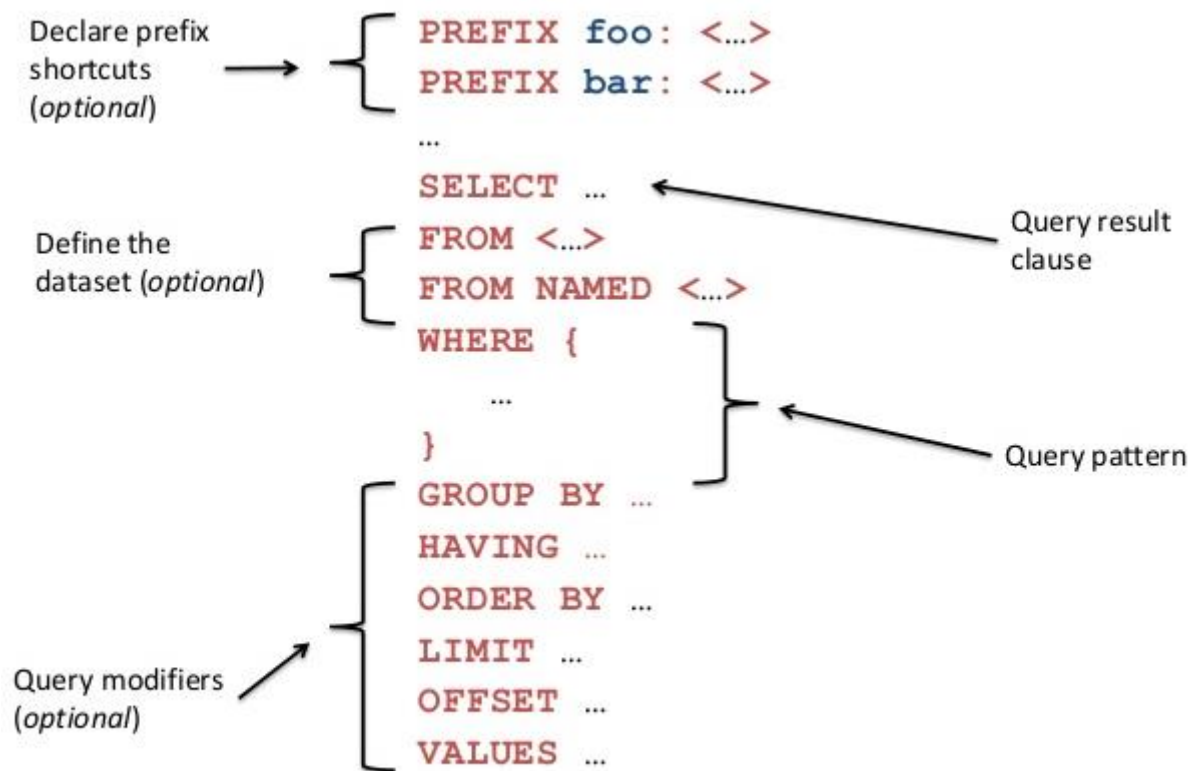
Petición del tipo de la entidad de la que el investigador es objeto

SPARQL. Prefijos comunes

| Prefijo | Representa |
|---------|---|
| rdf: | http://www.w3.org/1999/02/22-rdf-syntax-ns# |
| rdfs: | http://www.w3.org/2000/01/rdf-schema# |
| owl: | http://www.w3.org/2002/07/owl# |
| xsd: | http://www.w3.org/2001/XMLSchema# |
| dc: | http://purl.org/dc/elements/1.1/ |
| foaf: | http://xmlns.com/foaf/0.1/ |

Referencia: [SPARQL Cheat Sheet. Lee Feigenbaum](#)

SPARQL. Anatomía de una query



Referencia: [SPARQL Cheat Sheet. Lee Feigenbaum](#)

SPARQL. Combinando Graph Patterns

Considerando que A y B son *graph patterns*:

- ❑ *Graph Pattern* básico, con uno o más *triple patterns*

A . B

- ❑ Conjunción. Unir los resultados de resolver A y B mediante la igualdad de los valores de variables en común.

- ❑ *Optional Graph Patterns*

A OPTIONAL { B }

- ❑ Equivalente a *left join*. Unir los resultados de resolver A y B mediante la igualdad de los valores de variables en común, si es posible. Conservar los resultados de A aunque no haya variables en común con B.

Referencia: [SPARQL Cheat Sheet. Lee Feigenbaum](#)

SPARQL. Subconsultas

Considerando que A, B y C son *graph patterns*:

```
A .  
{  
  SELECT ...  
  WHERE {  
    B  
  }  
}
```

C .

☐ Une los resultados de la subconsulta con los resultados de resolver A y C

Referencia: [SPARQL Cheat Sheet. Lee Feigenbaum](#)

SPARQL. Filtros

- ❑ Los filtros SPARQL eliminan soluciones que no causan la evaluación de una expresión como verdadera.
- ❑ Los filtros se colocan en una consulta junto a un *graph pattern* básico

A . B . FILTER (...expr...)

Referencia: [SPARQL Cheat Sheet. Lee Feigenbaum](#)

SPARQL. Agregados

1. Particionar los resultados en grupos según las expresiones de la cláusula GROUP BY
2. Evaluar proyecciones y funciones agregadas en la cláusula SELECT para obtener un resultado por grupo.
3. Filtrar resultados agregados con la cláusula HAVING.

SPARQL 1.1 incluye: COUNT, SUM, AVG, MIN, MAX, SAMPLE, GROUP_CONCAT

Referencia: [SPARQL Cheat Sheet. Lee Feigenbaum](#)

| ?key | ?val | ?other1 |
|------|------|---------|
| 1 | 4 | ... |
| 1 | 4 | ... |
| 2 | 5 | ... |
| 2 | 4 | ... |
| 2 | 10 | ... |
| 2 | 2 | ... |
| 2 | 1 | ... |
| 3 | 3 | ... |

| ?key | ?sum_of_val |
|------|-------------|
| 1 | 8 |
| 2 | 22 |
| 3 | 3 |

| ?key | ?sum_of_val |
|------|-------------|
| 1 | 8 |
| 3 | 3 |

SPARQL. SPARQL sobre HTTP.

SPARQL define un protocolo de ejecución de consultas sobre HTTP.

☐ Petición al SPARQL Endpoint:

<http://host.domain.com/sparql/endpoint?<parameters>>

☐ Los parámetros pueden incluir:

☐ query=<encoded query string>

☐ default-graph-uri=<encoded graph URI>

☐ named-graph-uri=<encoded graph URI>

☐ El protocolo admite peticiones HTTP GET o POST.

☐ Los grafos indicados en el protocolo sobrescriben a los indicados en la query.

☐ A veces la consulta excede la longitud máxima de un GET. En ese caso sería obligatorio hacer la petición con un POST.

Referencia: [SPARQL Cheat Sheet. Lee Feigenbaum](#)

SPARQL. Actualizaciones

Comandos de actualización del lenguaje SPARQL

INSERT DATA { triples }

DELETE DATA {triples}

[DELETE { template }] [INSERT { template }] WHERE { pattern }

LOAD <uri> [INTO GRAPH <uri>]

CLEAR GRAPH <uri>

CREATE GRAPH <uri>

DROP GRAPH <uri>

Referencia: [SPARQL Cheat Sheet. Lee Feigenbaum](#)

SPARQL. Tutorial de SPARQL

Tutorial de SPARQL recomendado:

<https://docs.data.world/tutorials/sparql/introduction.html>



FONDO EUROPEO DE DESARROLLO REGIONAL (FEDER)

Una manera de hacer Europa

Hércules ASIO. Ejemplos SPARQL



FONDO EUROPEO DE DESARROLLO REGIONAL (FEDER)

Una manera de hacer Europa

GRACIAS

