

SPARQL

Explotación semántica de datos
Máster Universitario en Bioinformática



Nombre	Puerto Blazegraph	Puerto Pubby
ALBALADEJO RIAD, NORA	3030	8170
BELMONTE MARTINEZ, JOSE MARIA	3031	8171
CABANES MADRID, JOSE ALBERTO	3032	8172
CANTON SANDOVAL, JOAQUIN	3033	8173
GARAY MAYOL, BEATRIZ	3034	8174
GONZALEZ-CONDE RUIZ, MIRIAM	3035	8175
GUILLEN MARQUINA, PABLO	3036	8176
HANNAOUI ANAAOUI, ADIL	3037	8177
HERNANDEZ HIDALGO, ALBERTO	3038	8178
LOPEZ CARBONELL, FRANCISCO JAVIER	3040	8180
LOPEZ FERNANDEZ, EVA MARIA	3041	8181
LOPEZ MARIN, JUAN PEDRO	3042	8182
LUCAS RUIZ, FERNANDO	3043	8183
MAS GOMEZ, JORGE	3044	8184
MORENO GONZALEZ, NOELIA	3039	8179
ÑIGUEZ BAEZA, JAIME	3045	8185
PONCE ESTEBAN, ALMUDENA	3046	8186
RAMOS GONZALEZ, SARA	3047	8187
REDONDO ANIORTE, FRANCISCO JAVIER	3048	8188
RIQUELME GARCIA, ANDREA	3049	8189
RODRIGUEZ RODENAS, ALBA	3050	8190
TAHIRI ZAINANE, GHIZLANE	3051	8191
TURPIN GOMEZ, ASUNCION	3052	8192

Instalar Triple Store

```
~# ssh user@dayhoff.inf.um.es
```

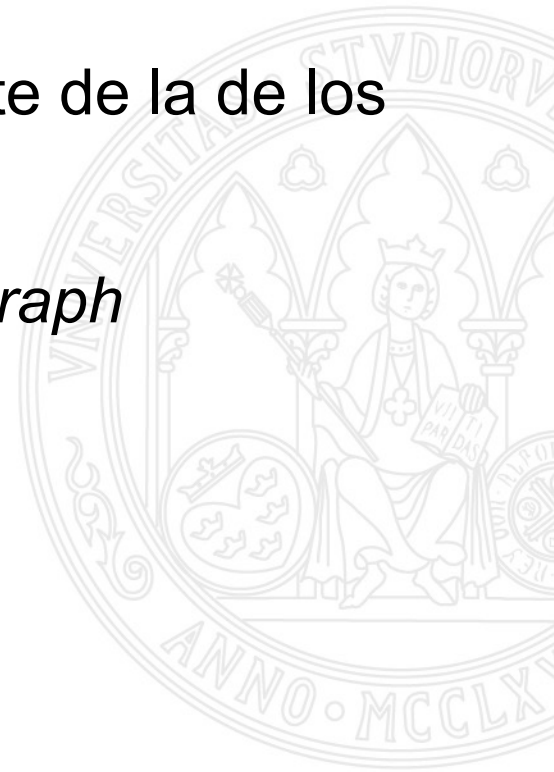
```
~# git clone https://github.com/mikel-egana-aranguren/UM-  
Bioinformatics-MSc-FAIR-data.git
```

Si lo queréis bajar en local podéis descargarlo aquí:
<https://github.com/mikel-egana-aranguren/UM-Bioinformatics-MSc-FAIR-data>

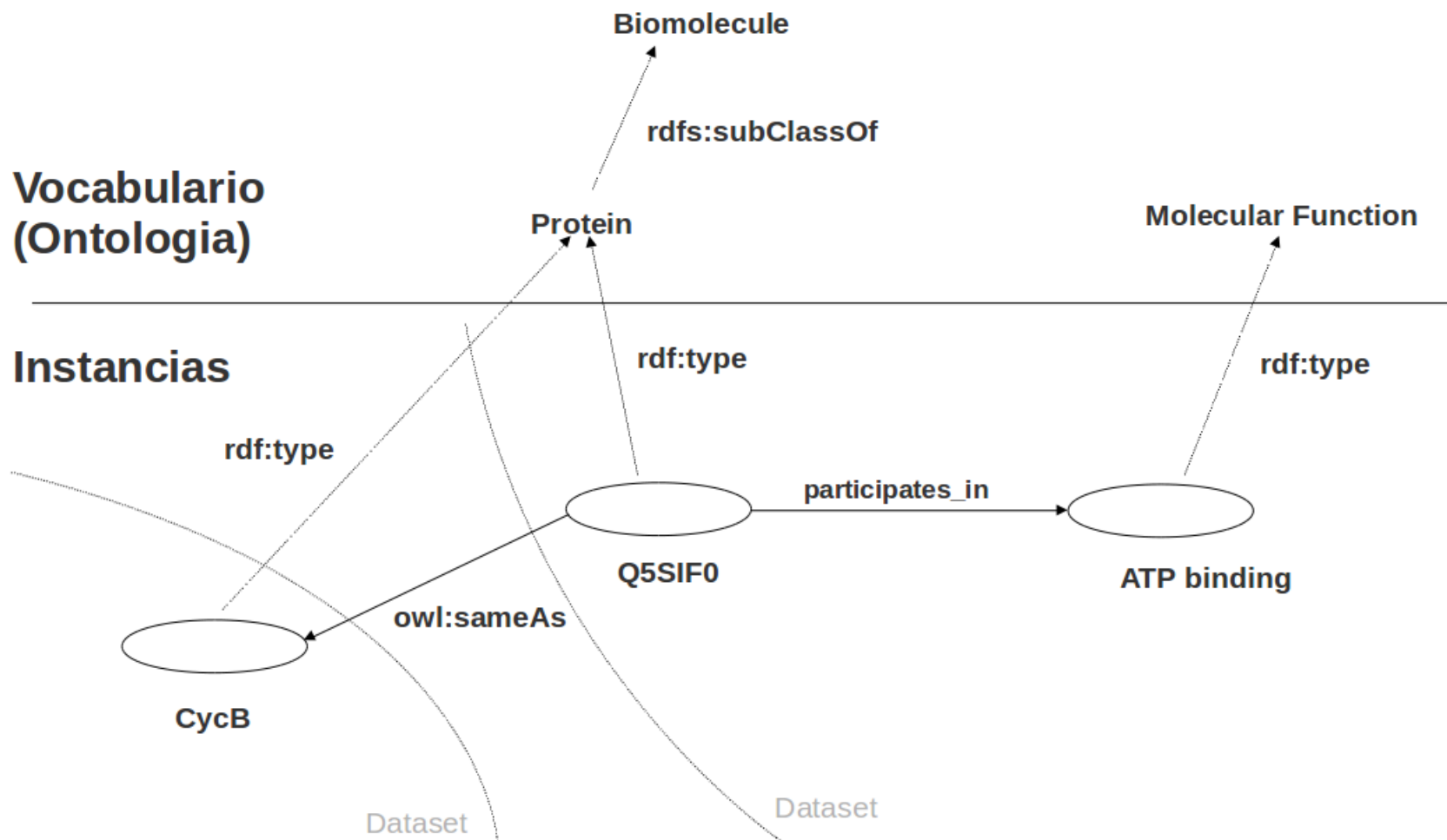
- Lenguaje para hacer consultas sobre grafos RDF (“El SQL para RDF”)
- Especificación actual, SPARQL 1.1
 - <https://www.w3.org/TR/sparql11-query/>
- Se consultan datos RDF mediante SPARQL endpoint: interfaz de consulta a Triple Store



- Un triple store puede almacenar más de un grafo o dataset
- Grafo: conjunto de tripletas
- Un grafo se identifica con una URI (diferente de la de los datos)
- Todas las Triple Stores tienen un *Default Graph*

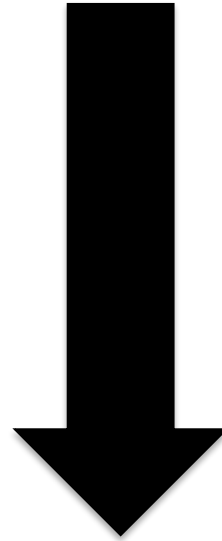


Ontología / Instancias (Recordatorio)



Familiarizarse con ejemplo RDF

Descargar recursos/prácticas/sparql/data.rdf y ontology.owl
en Aula Virtual



<https://www.w3.org/RDF/Validator>

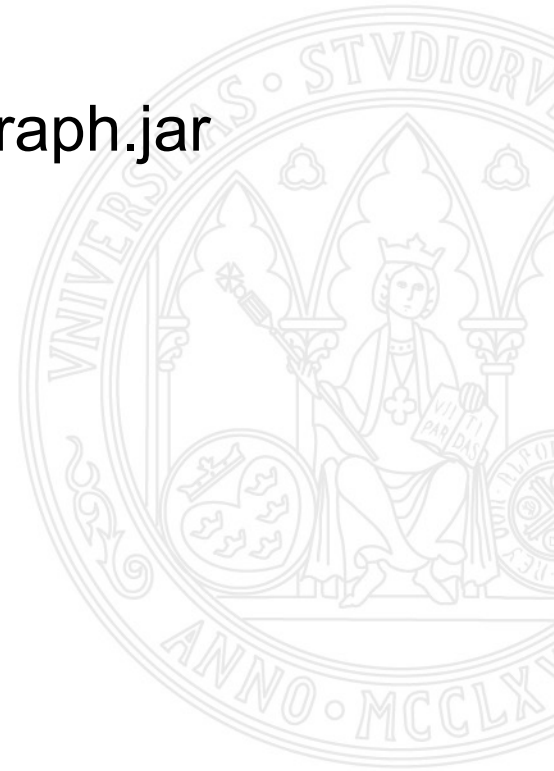
[Subir data.rdf a Blazegraph]



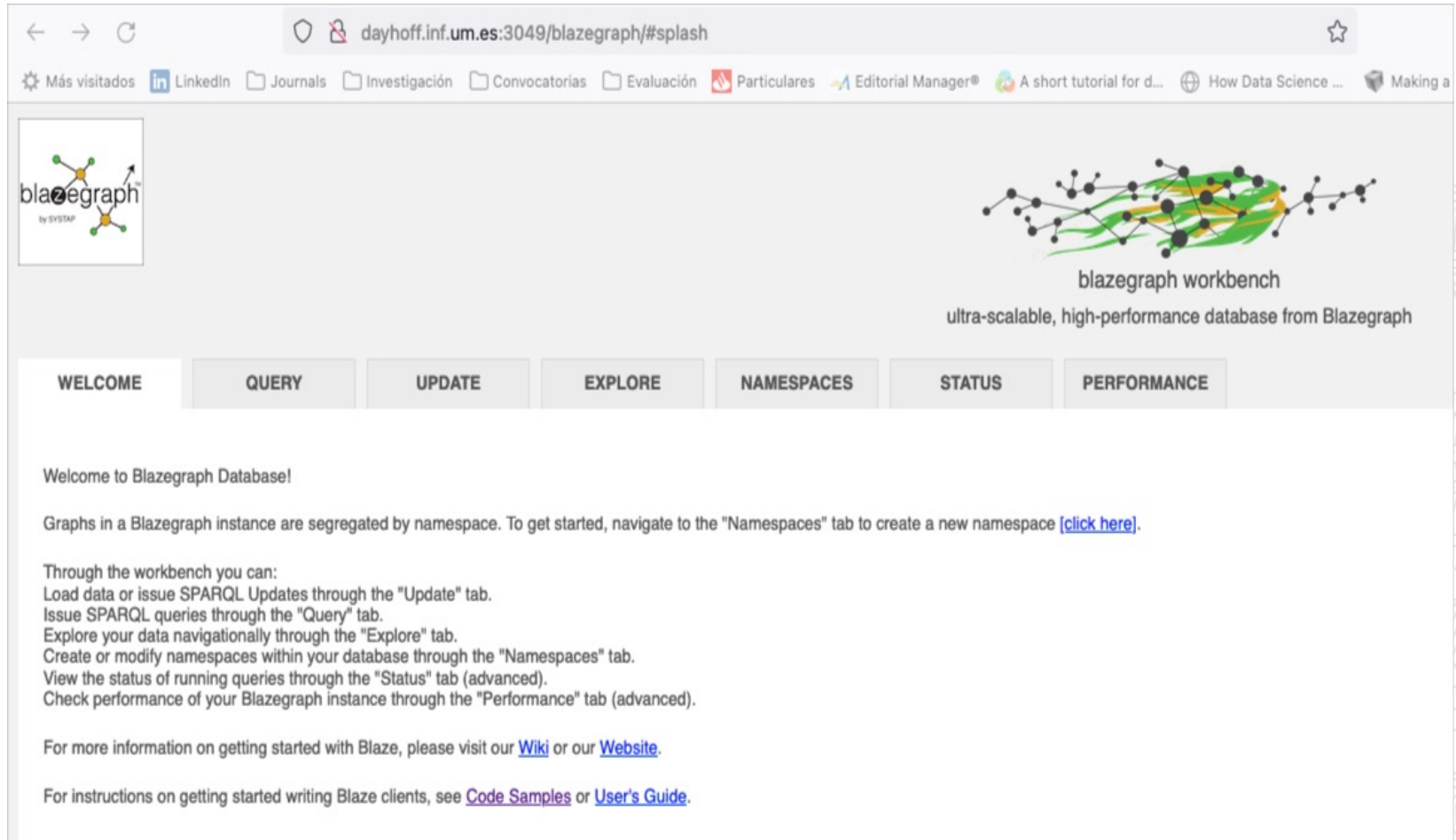
Ejecutar Triple Store

```
~# cd /home/user/UM-Bioinformatics-MSc-FAIR-data  
/LinkedDataServer/blazegraph
```

```
~# java -server -Djetty.port=puerto -jar blazegraph.jar
```



Nos conectamos a través del navegador



The screenshot shows a web browser window with the address bar displaying `dayhoff.inf.um.es:3049/blazegraph/#splash`. The browser's bookmark bar includes links like "Más visitados", "LinkedIn", "Journals", "Investigación", "Convocatorias", "Evaluación", "Particulares", "Editorial Manager", and "A short tutorial for d...". The page features the Blazegraph logo on the left and a large network graph visualization on the right. Below the graph, the text reads "blazegraph workbench" and "ultra-scalable, high-performance database from Blazegraph". A horizontal navigation bar contains tabs: "WELCOME", "QUERY", "UPDATE", "EXPLORE", "NAMESPACES", "STATUS", and "PERFORMANCE". The "WELCOME" tab is active, displaying a message: "Welcome to Blazegraph Database!". Below this, it states: "Graphs in a Blazegraph instance are segregated by namespace. To get started, navigate to the 'Namespaces' tab to create a new namespace [\[click here\]](#)." A list of actions possible through the workbench is provided: "Through the workbench you can: Load data or issue SPARQL Updates through the 'Update' tab. Issue SPARQL queries through the 'Query' tab. Explore your data navigationally through the 'Explore' tab. Create or modify namespaces within your database through the 'Namespaces' tab. View the status of running queries through the 'Status' tab (advanced). Check performance of your Blazegraph instance through the 'Performance' tab (advanced)." At the bottom, it provides links for more information: "For more information on getting started with Blaze, please visit our [Wiki](#) or our [Website](#)." and "For instructions on getting started writing Blaze clients, see [Code Samples](#) or [User's Guide](#)."

blazegraph
by SYSTAP

blazegraph workbench
ultra-scalable, high-performance database from Blazegraph

WELCOME QUERY UPDATE EXPLORE NAMESPACES STATUS PERFORMANCE

Welcome to Blazegraph Database!

Graphs in a Blazegraph instance are segregated by namespace. To get started, navigate to the "Namespaces" tab to create a new namespace [\[click here\]](#).

Through the workbench you can:

- Load data or issue SPARQL Updates through the "Update" tab.
- Issue SPARQL queries through the "Query" tab.
- Explore your data navigationally through the "Explore" tab.
- Create or modify namespaces within your database through the "Namespaces" tab.
- View the status of running queries through the "Status" tab (advanced).
- Check performance of your Blazegraph instance through the "Performance" tab (advanced).

For more information on getting started with Blaze, please visit our [Wiki](#) or our [Website](#).

For instructions on getting started writing Blaze clients, see [Code Samples](#) or [User's Guide](#).

Nos creamos un grafo

WELCOME

QUERY

UPDATE

EXPLORE

NAMESPACES

STATUS

PERFORMANCE

Namespaces

kb (default)	In use	Delete	Properties	Rebuild Full Text Index	Clone	Service Description
--------------	--------	------------------------	----------------------------	---	-----------------------	-------------------------------------

[Download VOID description of all namespaces](#)

Create namespace

There are a number of features to enable. There's full documentation [here](#). You must select "Use" after you have created the KB.

A quick reference is below:

- o PropertyGraph: Select triples.
- o RDF + SPARQL with named graphs: Select quads mode.
- o Support for [Reification Done Right \(RDR\)](#). Select rdr mode.

Name: Mode: Inference: ☒ Isolatable indices: ☐ Full text index: ☒ Enable geospatial: ☐

Seleccionamos nuestro grafo

WELCOME**QUERY****UPDATE****EXPLORE****NAMESPACES**

Namespaces

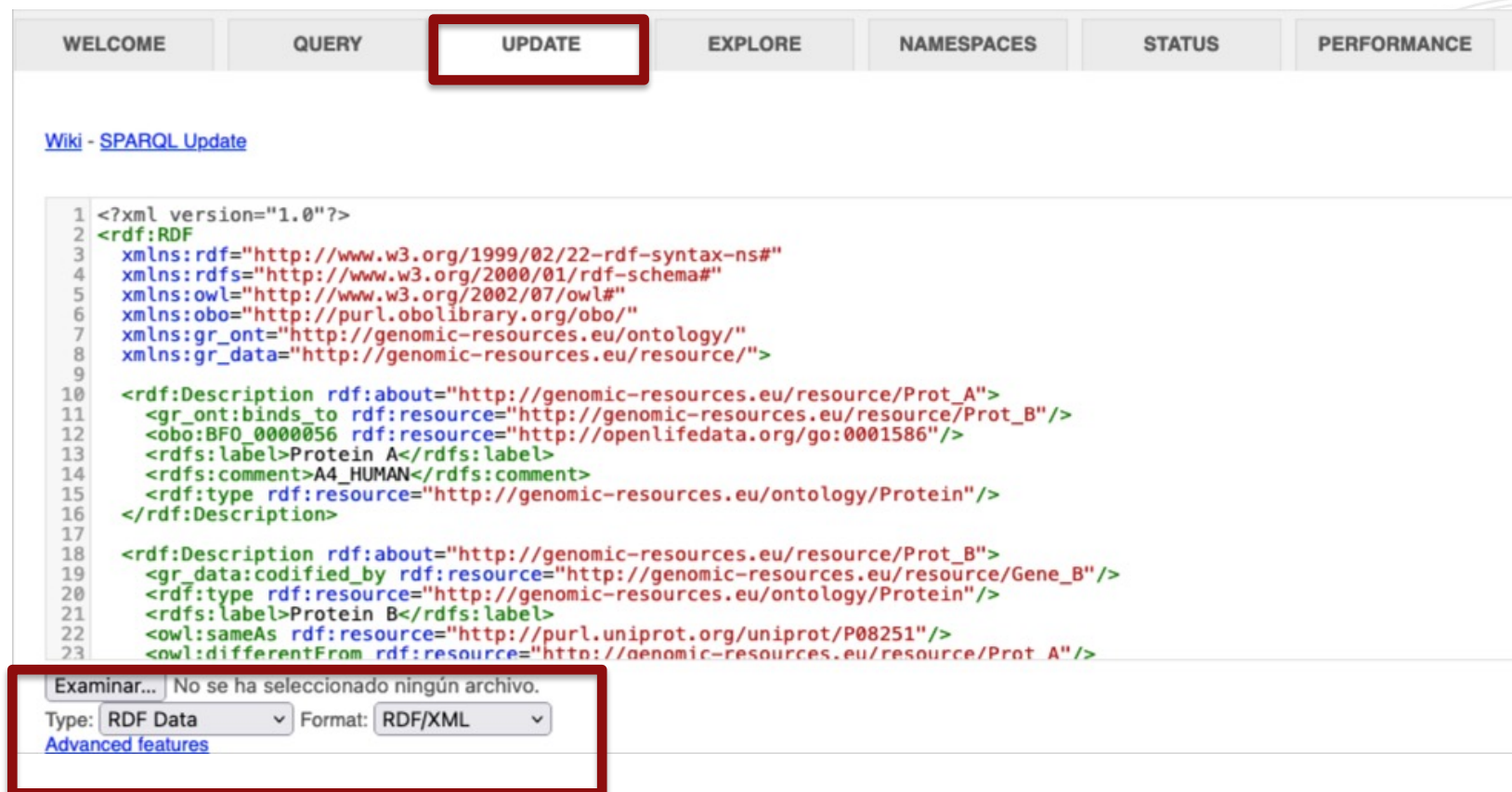
kb (default)	Use	Delete	Properties	Rebuild Full Text Index	Clone	Service Description
mygraph	<input type="checkbox"/>	Delete	Properties	Rebuild Full Text Index	Clone	Service Description

[Download VoID description of all namespaces](#)



Cargamos los datos

- Cargamos data.rdf y ontology.owl



The screenshot shows the SPARQL Update interface. The 'UPDATE' tab is selected and highlighted with a red box. Below the tabs, there is a link to 'Wiki - SPARQL Update'. The main area contains a SPARQL query with line numbers 1 through 23. The query defines namespaces and describes two proteins, Prot_A and Prot_B, with their relationships and properties. At the bottom, there is a file upload section with a red box around it. It includes a button 'Examinar...' and a message 'No se ha seleccionado ningún archivo.' Below this, there are dropdown menus for 'Type: RDF Data' and 'Format: RDF/XML'. A link 'Advanced features' is also present.

WELCOME QUERY **UPDATE** EXPLORE NAMESPACES STATUS PERFORMANCE

[Wiki - SPARQL Update](#)

```
1 <?xml version="1.0"?>
2 <rdf:RDF
3   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
5   xmlns:owl="http://www.w3.org/2002/07/owl#"
6   xmlns:obo="http://purl.obolibrary.org/obo/"
7   xmlns:gr_ont="http://genomic-resources.eu/ontology/"
8   xmlns:gr_data="http://genomic-resources.eu/resource/">
9
10  <rdf:Description rdf:about="http://genomic-resources.eu/resource/Prot_A">
11    <gr_ont:binds_to rdf:resource="http://genomic-resources.eu/resource/Prot_B"/>
12    <obo:BF0_0000056 rdf:resource="http://openlifedata.org/go:0001586"/>
13    <rdfs:label>Protein A</rdfs:label>
14    <rdfs:comment>A4_HUMAN</rdfs:comment>
15    <rdf:type rdf:resource="http://genomic-resources.eu/ontology/Protein"/>
16  </rdf:Description>
17
18  <rdf:Description rdf:about="http://genomic-resources.eu/resource/Prot_B">
19    <gr_data:codified_by rdf:resource="http://genomic-resources.eu/resource/Gene_B"/>
20    <rdf:type rdf:resource="http://genomic-resources.eu/ontology/Protein"/>
21    <rdfs:label>Protein B</rdfs:label>
22    <owl:sameAs rdf:resource="http://purl.uniprot.org/uniprot/P08251"/>
23    <owl:differentFrom rdf:resource="http://genomic-resources.eu/resource/Prot_A"/>

```

Examinar... No se ha seleccionado ningún archivo.

Type: **RDF Data** Format: **RDF/XML**

[Advanced features](#)

- Sintaxis similar a Turtle y SQL
- Las variables comienzan por el signo de interrogación
 - ?nombre
- URIs entre < >
 - up: <<http://purl.uniprot.org/core/>>
- Comentarios empiezan por #
 - # esto es un comentario
- Literales entre “ ”
 - “GeneB”, “24”^^xsd:integer
- El resultado es una tabla donde cada columna se corresponde con una propiedad especificada en SELECT



Estructura de consulta SPARQL

PREFIX uniprot: <http://uniprot/ontology/prop/>

...

Definición de prefijos
a utilizar en la consulta

SELECT ?variable Variables a consultar

FROM <> Grafo a consultar (opcional)

WHERE {
 ?s uniprot:participates-in ?o
}

Condición que debe cumplirse.
Patrón del grafo que queremos extraer
del grafo mayor

GROUP BY ..

ORDER BY ...

VALUES ...

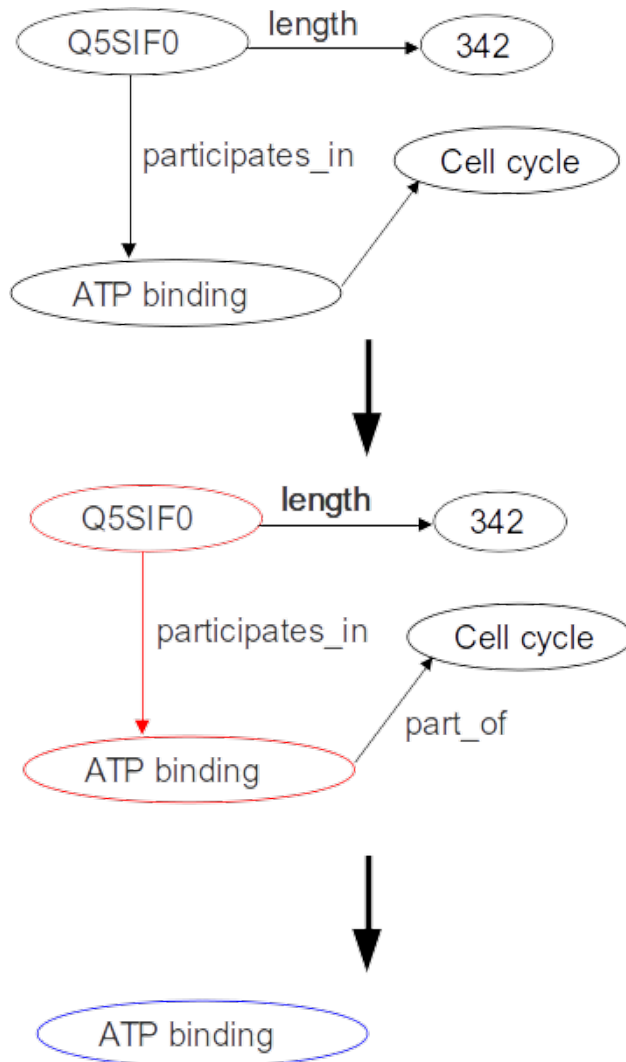
LIMIT ...

....

Modificadores de consulta (opcional)

FROM en SPARQL es opcional. Indica el o los grafos sobre los
que se debe buscar.

Por defecto existe un grafo activo

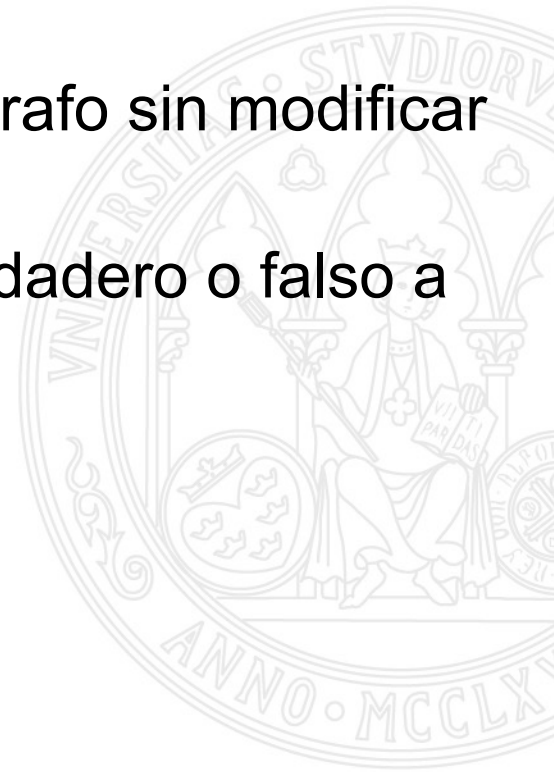


```

SELECT ?pro
WHERE {
  Q5SIF0 participates_in ?pro
}
  
```



- Tipos de consulta SPARQL:
 - **SELECT**, para mostrar datos
 - **CONSTRUCT**, permite crear un nuevo grafo sin modificar el original
 - **ASK**, sirve para preguntar, devuelve verdadero o falso a la consulta
 - **DESCRIBE** extrae la estructura RDF



Ejemplo de consulta SPARQL

Ejecutar ejemplos en Triple Store

Variables que queremos recibir

```
SELECT ?s ?p ?o
```

Patrón del grafo que queremos extraer del grafo mayor

```
WHERE {  
    ?s ?p ?o  
}
```



Ejemplo de consulta SPARQL

CONSULTA:

>> Dame todas las relaciones que tiene Gene_B

Prefijos

PREFIX gr_ont: <http://genomic-resources.eu/ontology/>

PREFIX gr_data: <http://genomic-resources.eu/resource/>

Variables que queremos recibir

SELECT ?p ?o

Patrón del grafo que queremos extraer del grafo mayor

WHERE {

gr_data:Gene_B ?p ?o

}

Ejemplo de consulta SPARQL

CONSULTA:

>> Dame todas las relaciones que tienen los genes

Prefijos

PREFIX gr_ont: <http://genomic-resources.eu/ontology/>

PREFIX gr_data: <http://genomic-resources.eu/resource/>

Variables que queremos recibir

SELECT ?s ?p ?o

Patrón del grafo que queremos extraer del grafo mayor

WHERE {

 ?s rdf:type gr_ont:Gene .

 ?s ?p ?o

}

Ejemplo de consulta SPARQL

CONSULTA:

>> Tiene que ser parte del Nucleolo

Prefijos

PREFIX gr_ont: <http://genomic-resources.eu/ontology/>

PREFIX gr_data: <http://genomic-resources.eu/resource/>

Variables que queremos recibir

SELECT ?p

Patrón del grafo que queremos extraer del grafo mayor

WHERE {

?p gr_ont:part_of gr_data:Nucleolus

}



Ejemplo de consulta SPARQL (rdf:type)

CONSULTA:

>> Tiene que ser parte del Nucleolo y clase a la que pertenecen en la ontología

```
PREFIX gr_ont: <http://genomic-resources.eu/ontology/>  
PREFIX gr_data: <http://genomic-resources.eu/resource/>
```

```
SELECT ?p ?clase  
WHERE {  
  ?p gr_ont:part_of gr_data:Nucleolus .  
  ?p rdf:type ?clase  
}
```

¿Por qué salen dos clases como resultado?



Ejemplo de consulta SPARQL (rdfs:label)

CONSULTA:

>> Tiene que ser parte de algo y tener un rdfs:label

PREFIX gr_ont: <http://genomic-resources.eu/ontology/>

PREFIX gr_data: <http://genomic-resources.eu/resource/>

SELECT ?p ?label ?part

WHERE {

Tiene que ser parte de algo

?p gr_ont:part_of ?part .

Tiene que tener un rdfs:label

?p rdfs:label ?label

}



FILTER

- Permite añadir restricciones sobre la condición de la cláusula WHERE

- Ejemplos:

```
FILTER ( datatype(?var) = xsd:integer )
```

```
FILTER ( lang(?var) = "ES" )
```

```
FILTER ( ?var >30 || ?var <18 )
```

PREFIX <...>

SELECT ...

WHERE {

...

```
FILTER regex(?var, 'texto.*')
```

}



FILTRAR RESULTADOS

Category	Functions / Operators	Example
LOGICA	!, &&, ,	?age < 25 ?age > 50
COMPARACIONES	=, !=, <, <=, >, >=	?age < 25
CÁLCULOS	+, -, *, /	?decimal * 10 > ?minPercent
TESTS SPARQL	isURI, isBlank, isLiteral, bound	isURI(?person) !bound(?person)
ACCESO A DATOS	str, lang, datatype	lang(?title) = "en"
OTROS	sameTerm, langMatches, regex	regex(?ssn, "\\d{3}-\\d{2}-\\d{4}")

CONSULTA:

>> Dame el nombre expresado como valor de la propiedad `gr_ont:name` ó `rdfs:label` y cuyo valor cumple la expresión regular `'Protein.*'`

```
PREFIX gr_ont: <http://genomic-resources.eu/ontology/>  
PREFIX gr_data: <http://genomic-resources.eu/resource/>
```

```
SELECT ?name  
WHERE {  
  {?p gr_ont:name ?name }  
  UNION  
  {?p rdfs:label ?name}  
  FILTER regex(?name, 'Protein.*')  
}
```



VALORES DE VARIABLE PREDEFINIDOS

CONSULTA:

>> Dame el nombre expresado como valor de la propiedad
gr_ont:name y cuyo valor es “Gene B”

```
PREFIX gr_ont: <http://genomic-resources.eu/ontology/>  
PREFIX gr_data: <http://genomic-resources.eu/resource/>
```

```
SELECT ?name  
WHERE {  
    ?p gr_ont:name ?name .  
    VALUES ?name {"Gene B"}  
}
```



EJERCICIO 1 UNIPROT

(<https://sparql.uniprot.org>)

UNIVERSIDAD DE
MURCIA

CONSULTA:

>> ¿Cuál es la descripción de la actividad enzimática de la proteína Q9SZZ8?

Usar uniprot.xlsx para identificar qué clases y propiedades de Uniprot debemos usar



EJERCICIO 2 UNIPROT

(<https://sparql.uniprot.org>)

UNIVERSIDAD DE
MURCIA

CONSULTA:

>> Conseguir todas las proteínas que se añadieron a uniprot a partir del año pasado



FUNCIONES DE AGREGACIÓN: AVG, SUM, COUNT, SAMPLE, MAX, MIN

```
@prefix e: <http://ejemplo.org#>.
```

```
e:Pepe e:nombre "Jose" .
```

```
e:Pepe e:edad "31"^^xsd:integer .
```

```
e:Juan e:nombre "Juan" .
```

```
e:Juan e:edad "12"^^xsd:integer .
```

```
e:Ana e:nombre "Ana" .
```

```
e:Ana e:edad "25"^^xsd:integer .
```

```
PREFIX e <http://ejemplo.org#>
```

```
SELECT
```

```
  (AVG(?edad) AS ?media)
```

```
  (SUM(?edad) AS ?suma)
```

```
  (COUNT(?edad) AS ?cuenta)
```

```
  (SAMPLE(?edad) AS ?muestra)
```

```
  (MAX(?edad) AS ?mayor)
```

```
  (MIN(?edad) AS ?menor)
```

```
WHERE
```

```
{
```

```
  ?n e:edad ?edad .
```

```
}
```

media	suma	cuenta	muestra	mayor	menor
22.66	68	3	25	31	12

CONSULTA:

>> Dime cuántas instancias tienen como nombre una cadena de texto que empieza por “Protein”. El nombre puede indicarse a través de la propiedad `gr_ont:name` o `rdfs:label`.

```
PREFIX gr_ont: <http://genomic-resources.eu/ontology/>
PREFIX gr_data: <http://genomic-resources.eu/resource/>
```

```
SELECT (count(distinct ?p) as ?count)
WHERE {
  {?p gr_ont:name ?name }
  UNION
  {?p rdfs:label ?name}
  FILTER regex(?name, 'Protein.*')
}
```



CONTROL DE LOS RESULTADOS

- **DISTINCT** elimina valores duplicados
- **ORDER BY** permite especificar el orden de los resultados (puede especificarse ASC, DESC ...)
- **LIMIT n** indica el número de resultados
- **OFFSET m** indica a partir de qué resultado empezar a contar

PREFIX <...>

```
SELECT DISTINCT ?var
WHERE {
    ...
}
ORDER BY ?var
LIMIT 5
OFFSET 10
```



EJERCICIO 3 UNIPROT

(<https://sparql.uniprot.org>)

UNIVERSIDAD DE
MURCIA

CONSULTA:

>> ¿Cuántas entradas de proteínas hay en uniprot?



EJERCICIO 4 UNIPROT

(<https://sparql.uniprot.org>)

UNIVERSIDAD DE
MURCIA

CONSULTA:

>> ¿Cuántas entradas de proteínas hay en uniprot que sean de *Arabidopsis thaliana*?



- Se utiliza en la cláusula WHERE para indicar que dichas tripletas no tienen porque existir en el grafo resultado.
- Para obtener datos que pueden existir o no

```
SELECT # propiedades a consultar
WHERE {
  # tripletas que se deben cumplir
  ...
  OPTIONAL{
    # tripletas que no necesariamente deben cumplirse
    ...
  }
}
```



CONSULTA:

>> Tiene que ser parte de algo y puede tener opcionalmente un `rdfs:label`

```
PREFIX gr_ont: <http://genomic-resources.eu/ontology/>  
PREFIX gr_data: <http://genomic-resources.eu/resource/>
```

```
SELECT ?p ?label ?part  
WHERE {  
  # Tiene que ser parte de algo  
  ?p gr_ont:part_of ?part .  
  # Tiene que tener un rdfs:label, o no  
  OPTIONAL {  
    ?p rdfs:label ?label.  
  }  
}
```



- Cuando queremos especificar varias condiciones y todas son válidas
- Útil para consultar varios grafos parecidos a la vez

```
SELECT # propiedades a consultar
WHERE {
  {... # tripletas }
  UNION
  {... # tripletas }
}
```



CONSULTA:

>> Dame el nombre expresado como valor de la propiedad
gr_ont:name ó rdfs:label

```
PREFIX gr_ont: <http://genomic-resources.eu/ontology/>  
PREFIX gr_data: <http://genomic-resources.eu/resource/>
```

```
SELECT ?name  
WHERE {  
  {?p gr_ont:name ?name}  
  UNION  
  {?p rdfs:label ?name}  
}
```



ORDENAR Y LIMITAR RESULTADOS

CONSULTA:

>> Dame el nombre expresado como valor de la propiedad `gr_ont:name` ó `rdfs:label` y cuyo valor cumple la expresión regular `'Protein.*'`. Ordena el resultado alfabéticamente y límitalo a 1

```
PREFIX gr_ont: <http://genomic-resources.eu/ontology/>
PREFIX gr_data: <http://genomic-resources.eu/resource/>
```

```
SELECT ?name
WHERE {
  {?p gr_ont:name ?name }
  UNION
  {?p rdfs:label ?name}
  FILTER regex(?name, 'Protein.*')
}
ORDER BY ?name
LIMIT 1
```

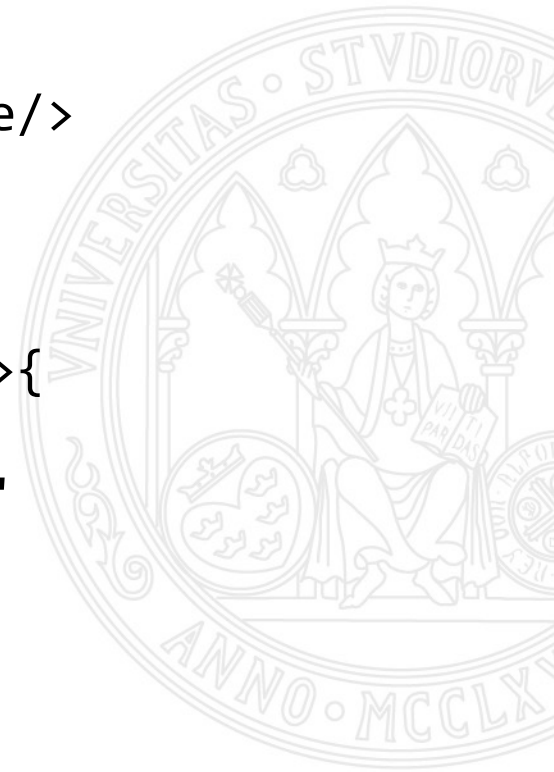


CONSULTAS FEDERADAS

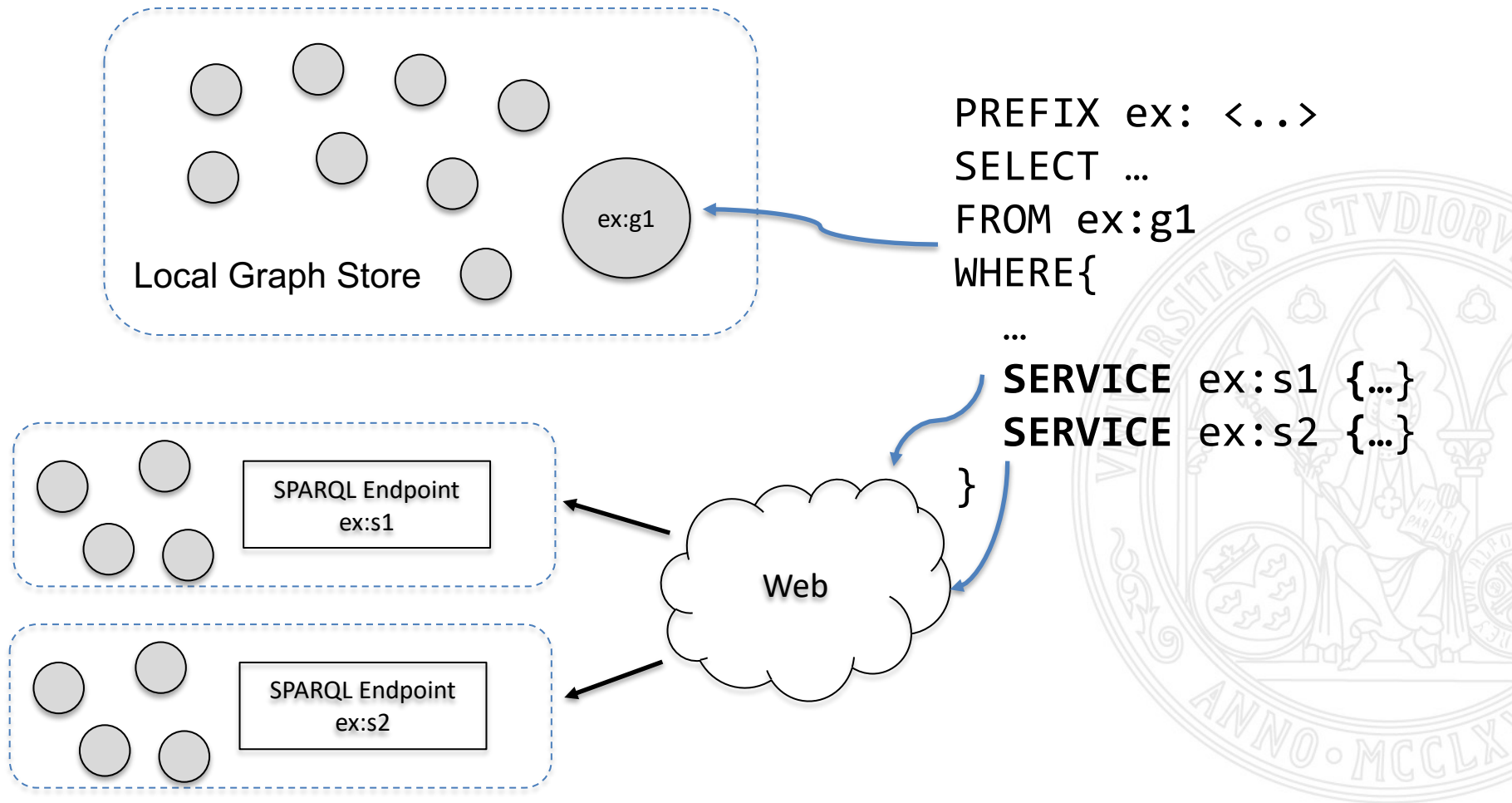
- SERVICE uri = indica un endpoint SPARQL

```
PREFIX up:<http://purl.uniprot.org/core/>
SELECT ?protein
WHERE{

    SERVICE <https://sparql.uniprot.org/>{
        ?protein a up:Protein .
        ?protein up:mnemonic 'A4_HUMAN'
    }
}
```



CONSULTAS FEDERADAS



```
PREFIX up:<http://purl.uniprot.org/core/>

SELECT ?prot ?protein

WHERE{
    ?prot rdfs:comment 'A4_HUMAN'

    SERVICE <https://sparql.uniprot.org/>{
        ?protein a up:Protein .
        ?protein up:mnemonic 'A4_HUMAN'
    }
}
```



OMA (<https://sparql.omabrowser.org/>)

CONSULTA:

>> Conseguir todas las proteínas en OMA codificadas por el gen INS, así como su mnemonic en Uniprot



OMA (<https://sparql.omabrowser.org/>)

```
PREFIX orth: <http://purl.org/net/orth#>
PREFIX obo: <http://purl.obolibrary.org/obo/>
PREFIX up: <http://purl.uniprot.org/core/>
PREFIX taxon: <http://purl.uniprot.org/taxonomy/>
PREFIX lscr: <http://purl.org/lscr#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
select distinct ?proteinOMA ?species ?mnemonic ?UniProt_URI
where {
  ?proteinOMA a orth:Protein .
  ?proteinOMA orth:organism ?organism .
  ?organism obo:RO_0002162 ?taxon . #in taxon
  ?taxon up:scientificName ?species .
  ?proteinOMA rdfs:label ?label .
  FILTER (?label="INS")
  ?proteinOMA lscr:xrefUniprot ?UniProt_URI.
  #Search the INS gene mnemonics from Uniprot database.
  service <http://sparql.uniprot.org/sparql> {
    ?UniProt_URI up:mnemonic ?mnemonic .
  }
}
```



OMA (<https://sparql.omabrowser.org/>)

proteinOMA	species	mnemonic	UniProt_URI
https://omabrowser.org/oma/info/SALSA66702	Salmo salar	B5X6U5_SALSA	http://purl.uniprot.org/uniprot/B5X6U5
https://omabrowser.org/oma/info/SALSA66702	Salmo salar	B5XDT4_SALSA	http://purl.uniprot.org/uniprot/B5XDT4
https://omabrowser.org/oma/info/OTOGA04835	Otolemur garnettii	B5FWC2_OTOGA	http://purl.uniprot.org/uniprot/B5FWC2
https://omabrowser.org/oma/info/CALJA04280	Callithrix jacchus	F6YAF3_CALJA	http://purl.uniprot.org/uniprot/F6YAF3
https://omabrowser.org/oma/info/PANTR02017	Pan troglodytes	INS_PANTR	http://purl.uniprot.org/uniprot/P30410
https://omabrowser.org/oma/info/MYOLU19828	Myotis lucifugus	G1P146_MYOLU	http://purl.uniprot.org/uniprot/G1P146
https://omabrowser.org/oma/info/FICAL04829	Ficedula albicollis	U3KAN7_FICAL	http://purl.uniprot.org/uniprot/U3KAN7
https://omabrowser.org/oma/info/CHICK21856	Gallus gallus	INS_CHICK	http://purl.uniprot.org/uniprot/P67970
https://omabrowser.org/oma/info/MELGA10632	Meleagris gallopavo	G1N7C1_MELGA	http://purl.uniprot.org/uniprot/G1N7C1
https://omabrowser.org/oma/info/HORSE04360	Equus caballus	F6QQU6_HORSE	http://purl.uniprot.org/uniprot/F6QQU6
https://omabrowser.org/oma/info/ICTTR07348	Ictidomys tridecemlineatus	INS ICTTR	http://purl.uniprot.org/uniprot/Q91XI3
https://omabrowser.org/oma/info/CHLSB00063	Chlorocebus sabaeus	A0A0D9RBQ0_CHLSB	http://purl.uniprot.org/uniprot/A0A0D9RBQ0
https://omabrowser.org/oma/info/PAPAN05288	Papio anubis	A0A096MTW9_PAPAN	http://purl.uniprot.org/uniprot/A0A096MTW9
https://omabrowser.org/oma/info/GORGO03059	Gorilla gorilla gorilla	INS_GORGO	http://purl.uniprot.org/uniprot/Q6YK33
https://omabrowser.org/oma/info/HUMAN03911	Homo sapiens	I3WAC9_HUMAN	http://purl.uniprot.org/uniprot/I3WAC9
https://omabrowser.org/oma/info/HUMAN03911	Homo sapiens	INS_HUMAN	http://purl.uniprot.org/uniprot/P01308
https://omabrowser.org/oma/info/NOTFU18540	Nothobranchius furzeri	A0A1A8ACH4_NOTFU	http://purl.uniprot.org/uniprot/A0A1A8ACH4
https://omabrowser.org/oma/info/PETMA08583	Petromyzon marinus	INS_PETMA	http://purl.uniprot.org/uniprot/P68987
https://omabrowser.org/oma/info/ORYLA19513	Oryzias latipes	A0A3P9KCJ8_ORYLA	http://purl.uniprot.org/uniprot/A0A3P9KCJ8
https://omabrowser.org/oma/info/ORYLA19513	Oryzias latipes	Q05K39_ORYLA	http://purl.uniprot.org/uniprot/Q05K39
https://omabrowser.org/oma/info/DANRE34450	Danio rerio	B2GSI0_DANRE	http://purl.uniprot.org/uniprot/B2GSI0
https://omabrowser.org/oma/info/DANRE34450	Danio rerio	INS_DANRE	http://purl.uniprot.org/uniprot/O73727
https://omabrowser.org/oma/info/DANRE34450	Danio rerio	Q9DDE5_DANRE	http://purl.uniprot.org/uniprot/Q9DDE5
https://omabrowser.org/oma/info/ORENI00270	Oreochromis niloticus	I3IUZ1_ORENI	http://purl.uniprot.org/uniprot/I3IUZ1
https://omabrowser.org/oma/info/LOXAF01659	Loxodonta africana	G3TLG9_LOXAF	http://purl.uniprot.org/uniprot/G3TLG9

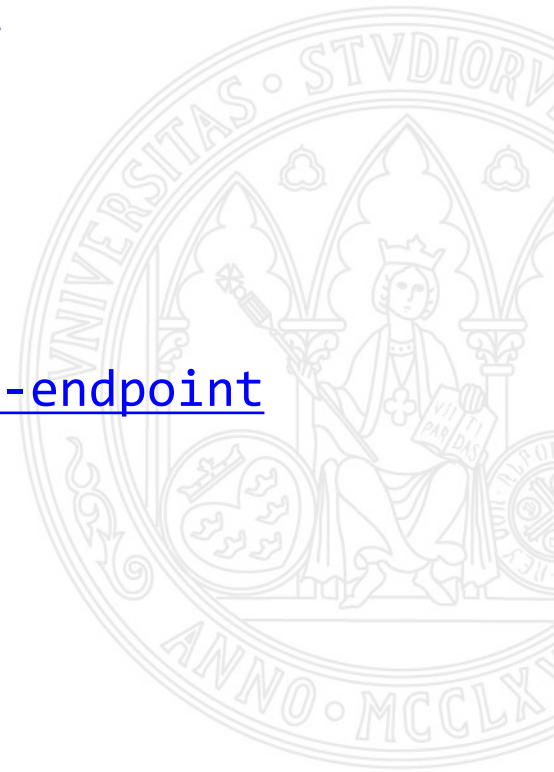
OTROS SPARQL ENDPOINTS

DISGENET <http://rdf.disgenet.org/sparql/>

BIO2RDF <https://bio2rdf.org/sparql>

OMA <https://sparql.omabrowser.org/>

Biogateway: <https://biogateway.eu/sparql-endpoint>



SPARQL desde R

- Acceder a Rstudio: <https://dayhoff.inf.um.es:8787/>
- Descargar desde Aula Virtual → Recursos → Prácticas → sparql el archivo sparql.R
- En la misma carpeta se encuentra el package SPARQL para instalarlo en vuestro ordenador, ya no está en CRAN



SPARQL – ASPECTOS AVANZADOS

Explotación semántica de datos
Máster Universitario en Bioinformática



```
SELECT *  
FROM <http://genomic-resources.eus/graph/um>  
WHERE {  
    ?s ?p ?o  
}
```

```
SELECT *  
FROM <http://genomic-resources.eus/graph/um>  
WHERE {  
    ?s ?p ?o  
}
```



- Consultar a qué grafo pertenecen las tripletas del repositorio

```
SELECT ?graph ?s ?p ?o
WHERE{
  GRAPH ?graph{
    ?s ?p ?o
  }
}
```



INSERT DATA

```
PREFIX gr_ont: <http://genomic-resources.eu/ontology/>
PREFIX gr_data: <http://genomic-resources.eu/resource/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
INSERT DATA {
  GRAPH <http://genomic-resources.eus/graph/um>{
    gr_data:NucleolusPart rdfs:comment "Comment inserted"
  }
}
```



DELETE

```
PREFIX gr_ont: <http://genomic-resources.eu/ontology/>
PREFIX gr_data: <http://genomic-resources.eu/resource/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
DELETE {
  GRAPH <http://genomic-resources.eus/graph/um>{
    gr_data:NucleolusPart rdfs:label ?label
  }
}
WHERE{
  gr_data:NucleolusPart rdfs:label ?label
}
```

DELETE DATA

```
PREFIX gr_ont: <http://genomic-resources.eu/ontology/>
PREFIX gr_data: <http://genomic-resources.eu/resource/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
DELETE DATA {
  GRAPH <http://genomic-resources.eus/graph/um>{
    gr_data:NucleolusPart rdfs:comment "Comment inserted"
  }
}
```



- ASK devuelve verdadero o falso
- Puede ser útil para chequeo de errores

```
@prefix e: <http://ejemplo.org#>.
```

```
e:Pepe e:nombre "Jose" .
```

```
e:Pepe e:edad 31 .
```

```
e:Juan e:nombre "Juan" .
```

```
e:Juan e:edad "Doce" .
```

```
e:Ana e:nombre "Ana" .
```

```
e:Ana e:edad 25 .
```

```
PREFIX e:<http://ejemplo.org#>
```

ASK

```
WHERE{
```

```
  ?x e:edad ?n
```

```
  FILTER(!isNumeric(?n))
```

```
}
```

YES

CONSULTA:

>> Evalúa si el recurso Gene_B tiene como valor de la propiedad gr_ont:name “Gene B”

```
PREFIX gr_ont: <http://genomic-resources.eu/ontology/>
PREFIX gr_data: <http://genomic-resources.eu/resource/>
ASK
WHERE{
  gr_data:Gene_B gr_ont:name "Gene B"
}
```



- Devuelve una descripción RDF de uno o varios nodos

```
DESCRIBE ?var  
WHERE{  
    ...  
}
```



CONSULTA:

>> Describe la estructura del recurso Prot_B

```
PREFIX gr_data: <http://genomic-resources.eu/resource/>  
DESCRIBE gr_data:Prot_B
```

```
DESCRIBE <http://genomic-resources.eu/resource/Prot_B>
```



CONSTRUCT A RESULT GRAPH

- Permite crear un grafo de salida

```
CONSTRUCT{  
    ...  
}  
WHERE{  
    ...  
}
```



CONSTRUCT A RESULT GRAPH

CONSULTA:

>> Añade las tripletas que indiquen que Cualquier cosa parte del nucléolo pertenece a la clase `gr_data:NucleolusPart` y esta última tiene `rdfs:label` “Part of the nucleolus”

```
PREFIX gr_ont: <http://genomic-resources.eu/ontology/>
PREFIX gr_data: <http://genomic-resources.eu/resource/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
CONSTRUCT{
?p rdf:type gr_data:NucleolusPart .
gr_data:NucleolusPart rdfs:label "Part of the nucleolus"
}
WHERE {
?p gr_ont:part_of gr_data:Nucleolus
}
```



TRATAMIENTO DE GRAFOS

INSERT DATA = insertar tripletas

DELETE DATA = borrar tripletas

DELETE/INSERT = borrar / insertar tripletas condicionalmente

CLEAR = borrar todas las tripletas de un grafo

CREATE = crear grafo

DROP = eliminar grafo

Ejemplos:

borra todas las tripletas del grafo por defecto

CLEAR DEFAULT

#borra todas las tripletas del grafo GRAPH

CLEAR <GRAPH>

borra todas las tripletas de todos los grafos

CLEAR GRAPHS

Elimina todos los grafos con sus tripletas

DROP GRAPHS



INSERT EN UN GRAFO CONCRETO

```
PREFIX gr_ont: <http://genomic-resources.eu/ontology/>
PREFIX gr_data: <http://genomic-resources.eu/resource/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
INSERT {
  GRAPH <http://genomic-resources.eus/graph/um>{
    ?p rdf:type gr_data:NucleolusPart .
    gr_data:NucleolusPart rdfs:label "Part of nucleolus"
  }
}

WHERE{
  ?p gr_ont:part_of gr_data:Nucleolus
}
```