

Laboratorio 1 – RDF, SPARQL

Objetivos:

1. Entender el modelo RDF.
2. Ser capaz de crear archivos RDF manualmente.
3. Ser capaz de ejecutar y gestionar una Triple Store.
4. Entender el lenguaje y protocolo de consulta SPARQL.

Recursos necesarios:

- Ubuntu (Usuario/password: lsi/lsi) o servidor Google Cloud.

Índice:

1. Introducción.
2. Crear archivos RDF.
3. Triple Store GraphDB.
4. SPARQL.

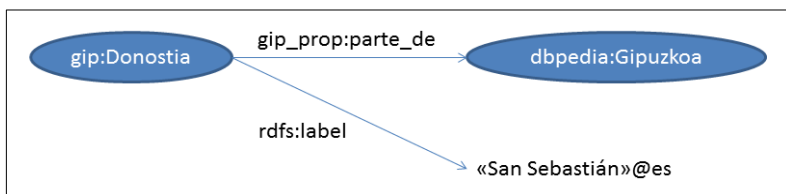
1.- Introducción

RDF (Resource Description Framework) es un estándar para modelar datos en forma de grafos en el contexto de la web. En este laboratorio aprenderéis a crear datos en RDF, cargarlos en una Triple Store, y hacer consultas mediante SPARQL (SPARQL Protocol and RDF Query Language), su lenguaje de consulta.

2.- Crear archivos RDF

Normalmente el origen de los datos RDF son otras fuentes de datos (Bases de datos relacionales, archivos XML, JSON, ...), pero vamos a crear archivos RDF manualmente para entender el modelo RDF.

Crea un archivo RDF que represente el siguiente modelo RDF, usando como guía los ejemplos vistos en clase:



Administración de Bases de Datos 2023/2024

El archivo puede estar escrito en la sintaxis que prefieras (RDF/XML, Turtle, JSON-LD, ...). En Turtle, puedes usar los siguientes Prefix:

- PREFIX gip:<http://gipuzkoa.eus/resource/>
- PREFIX gip_prop:<http://gipuzkoa.eus/prop/>
- PREFIX dbpedia:<http://dbpedia.org/resource/>
- PREFIX rdfs: ??? [Pista: buscar en prefix.cc]

Ahora crea un archivo RDF a partir de los datos que se encuentran en el archivo **Museoak.txt**. Lo puedes crear manualmente o puedes usar una herramienta como Morph-KGC¹ para crear una conversión ejecutable.

Una vez creado el RDF, crea otro archivo diferente con la siguiente informacion en un triple: *Bilbo (nuestro dataset) es igual a (owl:sameAs) "Bilbao" (Dbpedia)*.

Entrega:

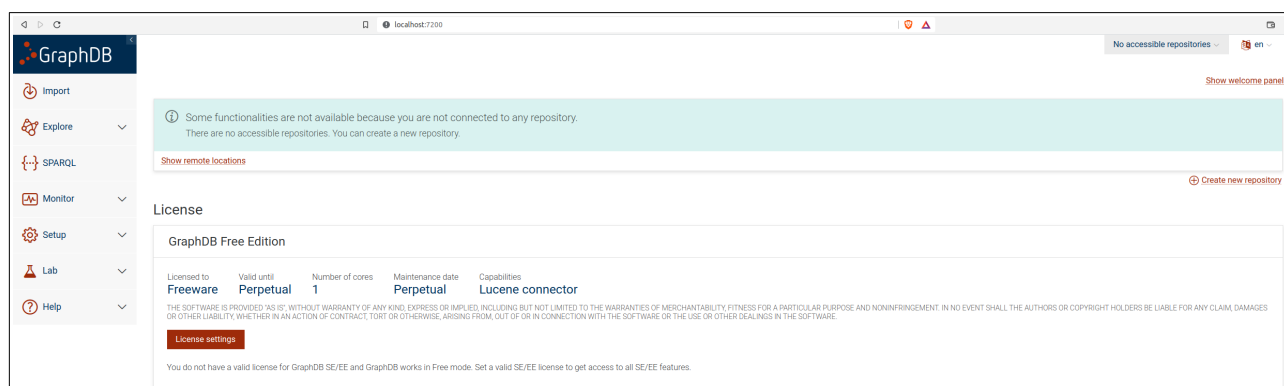
- Archivos RDF.

3.- Triple Store GraphDB

GraphDB², de la empresa OntoText³, es una Triple Store de alta calidad que incluye razonamiento automático, conectores para servicios como Lucene y Kafka, e incluso virtualización de bases de datos relacionales (Ontop). Para ejecutar GraphDB, bajar el archivo docker-compose de eGela y ejecutarlo (En la máquina local o en Google Cloud):

\$ docker-compose up

En la dirección <http://localhost:7200> debería aparecer la siguiente pantalla⁴:



1 <https://morph-kgc.readthedocs.io/>

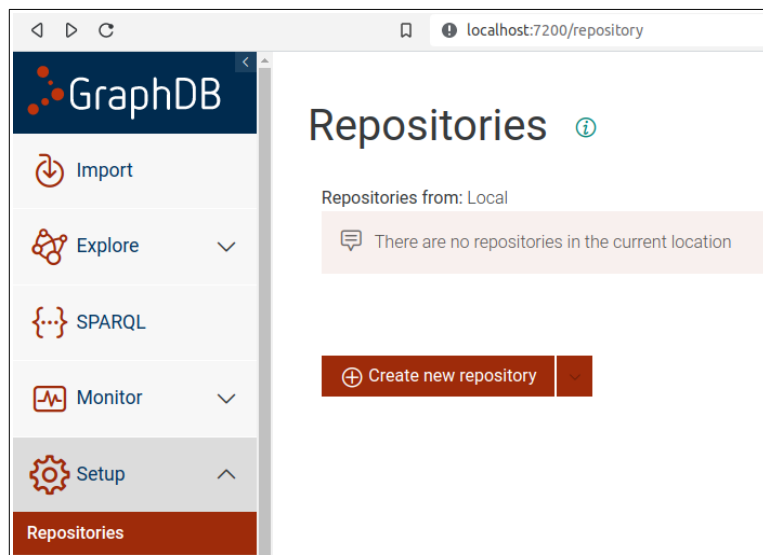
2 <https://www.ontotext.com/products/graphdb/>

3 <https://www.ontotext.com/>

4 Si estáis ejecutando GraphDB en Google Cloud tendréis que abrir el puerto 7200 en el firewall.

Administración de Bases de Datos 2023/2024

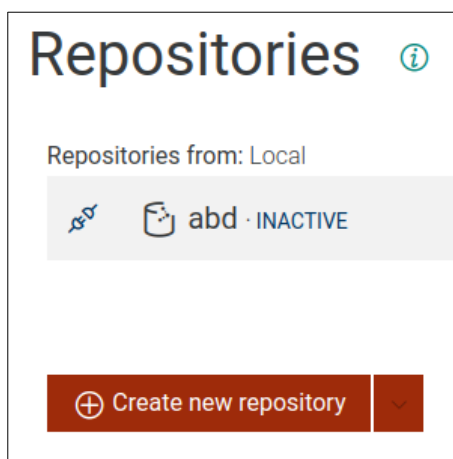
Para poder trabajar con GraphDB hay que crear un repositorio: *Setup > Repositories > Create new repository*:



Hay tres tipos de repositorios en GraphDB:

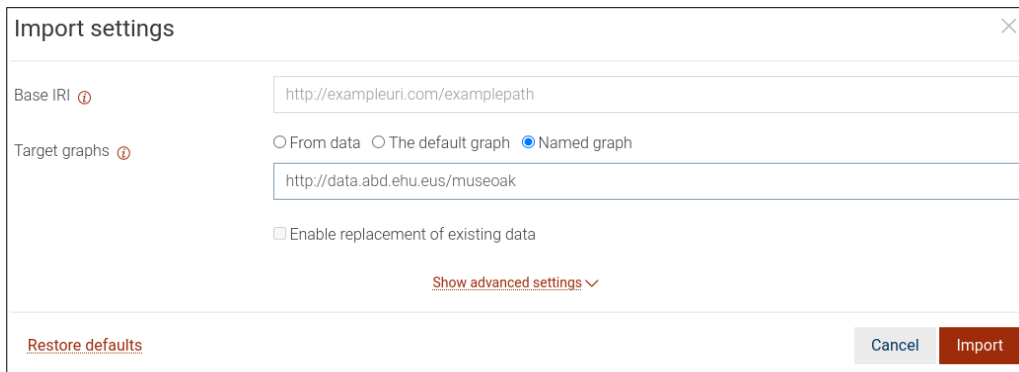
- GraphDB repository: el repositorio local normal.
- Ontop Virtual SPARQL: un endpoint SPARQL virtual (Sin datos) que traduce las consultas SPARQL a consultas SQL, pudiendo conectar así bases de datos relacionales y consultarlas como si fuesen RDF.
- FedX virtual SPARQL: federa diferentes endpoints SPARQL bajo uno solo.

Elegir *GraphDB repository* y crear un repositorio con todas las opciones por defecto. Una vez creado el repositorio, conectarlo haciendo click en el enchufe:

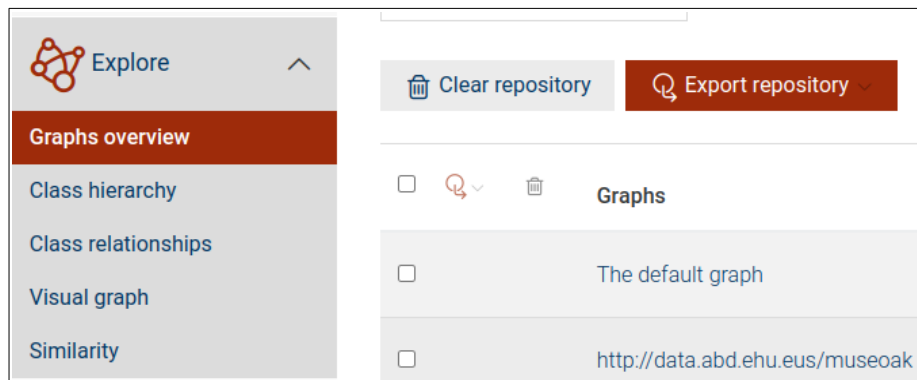


Administración de Bases de Datos 2023/2024

Para cargar el RDF creado en la sección anterior, ir a la pestaña *Import* y ahí se puede cargar RDF eligiendo un archivo, pegándolo en un formulario, desde una URL, o desde archivos del servidor. Es importante cargar los datos en un Named Graph para organizar la información adecuadamente. La URI del Named Graph puede ser cualquiera:



Si los datos han sido cargados adecuadamente, en *Explore > Graphs overview* debería aparecer la URI del Named Graph:



Haciendo click en la URI del Named Graph deberían aparecer los triples RDF recién cargados. A través de la opción *Visual Graph* se pueden ver los triples en forma de grafo.

Entrega:

- URL GraphDB en Google Cloud con RDF cargado.

4.- SPARQL

Cargar archivo RDF Museoak.rdf en GraphDB, en el Named Graph <http://data.abd.ehu.eus/museoak>. Para ejecutar consultas en GraphDB, ir a pestaña SPARQL:



Crea un archivo .rq para cada una de las siguientes consultas SPARQL:

- ¿Cuál sería la consulta SPARQL para obtener todos los triples de un grafo?
- ¿Cuál sería la consulta SPARQL para saber qué museo tiene trabajadores cuyo nombre empieza por "Mi" y más de 800 trabajadores?
- ¿Cuál sería la consulta SPARQL para saber dónde se sitúa el museo en el que trabaja Aitor Labajo?
- ¿Cuál sería la consulta SPARQL para saber quién es el alcalde de la ciudad en la que se sitúa el museo Guggenheim, aunque esa información no esté en Museoak.rdf? **[Pista: INSERT owl:sameAs, SERVICE dbpedia]**

RDF permite codificar los datos y los metadatos sobre esos datos usando el mismo lenguaje. Para añadir metadatos sobre el Named Graph de museos, ejecuta una consulta INSERT que añada el siguiente triple: <http://data.abd.ehu.eus/museoak> dcat:theme "museos".

- ¿Cuál sería la consulta SPARQL para saber qué Named Graphs almacena GraphDB?
- ¿Cuál sería la consulta SPARQL para saber qué dataset trata sobre museos y que información contiene sobre Aitor Labajo?

SPARQL, además de un lenguaje de consulta, es un protocolo para ejecutar consultas sobre HTTP. Crea un Shell Script (.sh) que contenga un línea de cURL que, mediante el método POST, ejecute una consulta SPARQL. Tendrás que buscar en la documentación de GraphDB para saber cuál es la URL del endpoint SPARQL de GraphDB.

Entrega:

- Archivos .rq con consultas SPARQL.
- Archivo .sh con ejecución CURL POST.