

Consultas para el *benchmarking*

Con base al documento entregado en proyecto terminal I, los parámetros a medir son:

1. Tiempo de selección del *SPARQL endpoint*
2. Tiempo de ejecución de la consulta
3. Completitud de la respuesta

A continuación, se presenta cómo es que se llevará a cabo dicho *benchmarking*.

Se llevó a cabo el ejercicio de extraer información del sitio LODAtlas[1] el cual es un sitio web que ofrece una API pública con la que se pueden extraer *SPARQL endpoints* y la información asociada a ellos. Se hizo un script en python para extraer la información y guardado de la misma. El script puede ser consultado en el siguiente [enlace](#) e igualmente, puede visitar la lista de *SPARQL endpoints* en el siguiente [Excel](#).

Las consultas por utilizar son las siguientes

Número de teatros en la ciudad de Nueva York

```
PREFIX dbpedia: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX geos: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX : <http://www.semanticweb.org/frubi/ontologies/2017/10/puntsWIFI#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT *
WHERE {
  SERVICE <http://dbpedia.org/sparql/> {
    SELECT ?teatreName ?lat ?long
    WHERE {
      ?teatre rdf:type dbpedia:Theatre .
      ?teatre foaf:name ?teatreName .
      ?teatre geo:lat ?lat .
      ?teatre geo:long ?long .
      ?teatre dbp:city "Paris"^^<http://www.w3.org/1999/02/22-rdf-syntax-ns#langString>
    }
  }
}
```

Fig. 1 consulta #1.

Cantidad de clases en común que tiene el repositorio local respecto a la URL del *SPARQL endpoint* solicitado

```
select * {

  #Tipos de clases extraidos desde dbpedia mediante consulta federada
  service <https://dbpedia.org/sparql> {
    select distinct ?t {
      [] a ?t .
    } limit 1000
  }
  #Recursos locales que pertenecen a tipos de Dbpedia
  ?a a ?t .
} limit 100
```

Fig. 2 Consulta #2.

Información asociada a la provincia de Madrid

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT * WHERE {
  SERVICE<http://geo.linkeddata.es/sparql> {
    <http://geo.linkeddata.es/resource/Provincia/Madrid> owl:sameAs ?resource
  }
}
```

Fig. 3 Consulta #3.

Artistas pertenecientes a la República Federal de Alemania

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX geo: <http://www.geonames.org/ontology#>
SELECT ?name ?location ?news WHERE {
  SERVICE<http://linkededgeodata.org/sparql> {
    ?artist foaf:name ?name .
    ?artist foaf:based_near ?location .
    ?location geo:parentFeature ?germany .
    ?germany geo:name 'Federal Republic of Germany'
  }
}
```

Fig. 4 Consulta #4.

La metodología por usar es la siguiente:

- Usar un script ya sea en C, Java o Python que recorra cada *SPARQL endpoint* y extraiga la url asociada.
- Usar junto a la palabra *service* de la consulta la url del *SPARQL endpoint* extraída por el script.
- Recibir datos y contar el número de datos recibidos.
- Repetir los pasos 1 y 2 tres veces y promediar el tiempo de selección del *SPARQL endpoint* y también el tiempo de ejecución de la consulta.

A continuación, en la figura 1, se muestra el diagrama de flujo que representa el proceso del *benchmarking*

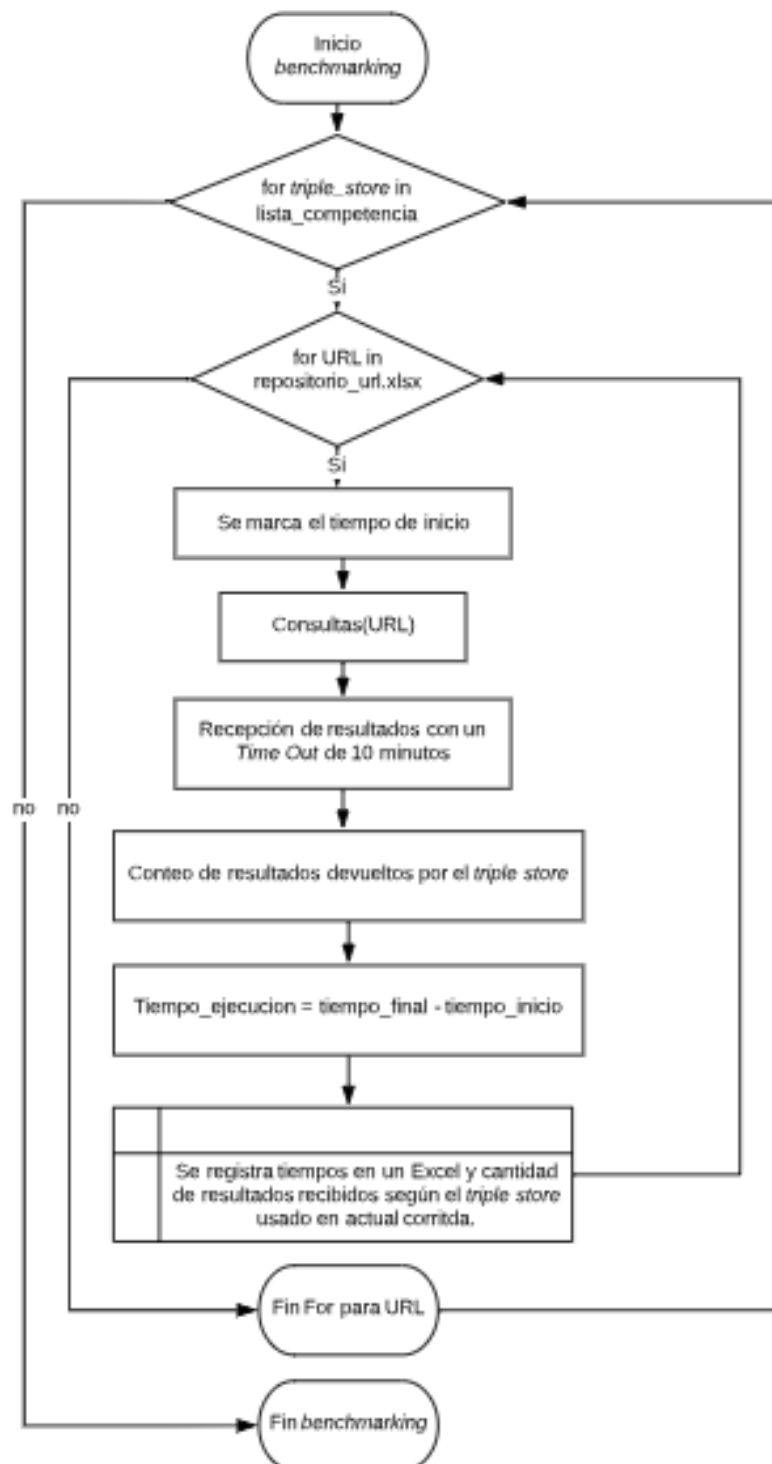


Fig. 5 Diagrama de flujo para la realización del Benchmarking.