**Input files**

datasets/geo/data.nq

dataset/geo/void.ttl

**Services**

**- geo/service/getOffers**Obtiene la lista de ofertas para un dataset  
{recurso1:  label, uri, comment} {license1: label, uri, precio, imgsrc}  
{recurso2:  label, uri, comment} {license2: label, uri, precio, imgsrc}

¿dónde se guardan las ofertas?

- cada política puede ser una oferta si tiene un tag que así lo indique  
  
**- geo/service/getResources?page=1&size=100**Obtiene la lista de provincias, (Almeria, Albacete...)  
{label1, uri, numeroTriples, licenses: [{uri:'ffaf', 'label', 'color'''}, {uri:'ffaf', label, color''}]}  },   
{label1, uri, openOrclosed}....

@prefix ldp: <http://www.w3.org/ns/ldp#>.

**- geo/service/getOpenResource?uri=**[**http://geo/Almeria**](http://geo/Almeria)Obtiene la lista de triples de una provincia como JSON-LD

**- geo/service/getClosedResource?uri=**[**http://geo/Almeria**](http://geo/Almeria) **como json-ld**

**- geo/service/getIndexedResources**

indica cuáles están en el front page.

**- geo/service/getRandomResource**Obtiene un recurso aleatoriamente  
{label1, uri, openOrclosed, numeroTriples?},

**Workflow**

1. Decidir la partición de los datos. Escoger un nombre para cada partición

2. Convertir los datos a un RDF dump como NQUADS data.nq. Asignar un named graph a cada triple según la partición

3. Asignar una política a cada partición de datos (named graph)

4. Hacer algunas políticas como "special offers"

5. Listar Recursos de un grafo:

- o bien enumerados sistematicamente con ldp:contains

- o bien que se adhieran a un patron. <?s ?p ?o> Por ejemplo:

Esto lo podemos poner de m <?s rdf:type schema:Provincia> directamente hardcodeado

ldp:contains "<?s rdf:type schema:Provincia>"