Bibliometric Data

Integrantes:
- Diego Pinto
- Bryant Portilla

Introducción

Lo que se busca con este proyecto es obtener información de investigaciones científicas relacionadas a covid-19 y mediante visualizaciones se pueda obtener muchos datos para observar qué áreas o qué intereses tiene la ciencia para ayudar a esta situación que está pasando en el mundo.

Fuente de datos

Los datos son obtenidos desde la base de datos científica SCOPUS y para la esto se usó el modelo ETL (extraer, transformar y cargar).

Cadenas de búsqueda:

TITLE-ABS-KEY ((covid-19) AND (sars-cov-2)) AND (LIMIT-TO (PUBSTAGE , "final"))

TITLE-ABS-KEY ((covid-19) AND (sars-cov-2)) AND (LIMIT-TO (PUBSTAGE , "aip"))

El resultado de la cadena de búsqueda que se estableció nos devolvió 2000 investigaciones relacionadas al covid-19 y con múltiples características relacionadas al artículo obtenido.

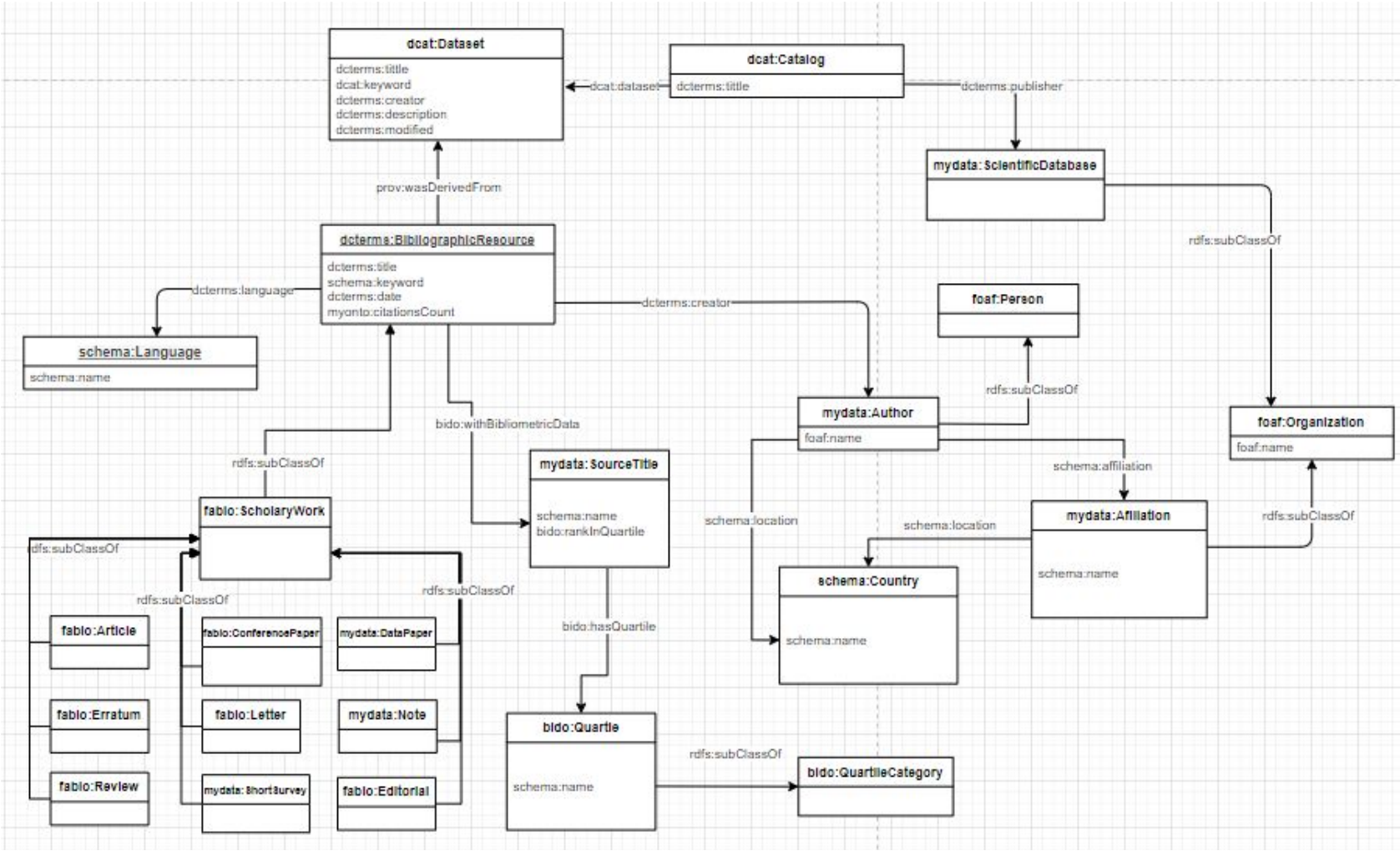
Limpieza de datos

Basándonos en el modelo ETL, dentro de la fase de transformación se realizó una limpieza para mejorar el contenido de la información que se obtuvo.

Los datos de obtuvieron en un csv directo desde la base de datos científica Scopus en donde se ha procedido a realizar:

- Estructurar como un solo atributo todos los autores (nombres, ids)
- Eliminar caracteres especiales (segun la logica de lectura).

Modelo


dcat:Dataset
dcterms:title
dcat:keyword
dcterms:creator
dcterms:description
dcterms:modified
dcat:dataset
dcat:Catalog
dcterms:title
dcterms:publisher
mydata:ScientificDatabase
prov:wasDerivedFrom
dcterms:BibliographicResource
dcterms:title
schema:keyword
dcterms:date
myonto:citationsCount
dcterms:language
dcterms:creator
rdfs:subClassOf
schema:Language
schema:name
foaf:Person
rdfs:subClassOf
mydata:Author
foaf:name
foaf:Organization
foaf:name
bido:withBibliometricData
rdfs:subClassOf
mydata:SourceTitle
schema:name
bido:rankInQuartile
schema:affiliation
fabio:ScholaryWork
schema:location
schema:location
mydata:Affiliation
schema:name
rdfs:subClassOf
rdfs:subClassOf
rdfs:subClassOf
rdfs:subClassOf
schema:Country
schema:name
bido:hasQuartile
fabio:Article
fabio:ConferencePaper
mydata:DataPaper
fabio:Erratum
fabio:Letter
mydata:Note
fabio:Review
mydata:ShortSurvey
fabio:Editorial
bido:Quartile
schema:name
rdfs:subClassOf
bido:QuartileCategory

Generacion RDF (Jena)

Estructura de los datos en excel

Dataset	Authors	Authors_ID	Title	EID	Year	Source title
BiblioData	Byass P.	7006527895	Eco-epidemi	2-s2.0-85084	2020	Global Healt
BiblioData	Liu P.; Cai J.;	57200337904	Dynamic sur	2-s2.0-85086	2020	Emerging mi

Volume	Issue	Art. No.	Page start	Page end	Cited by
13	1	1760490			
9	1		1254	1258	

DOI	Link	Document Ty	Publication S	Access Type	Source	Language	keywords
10.1080/165	https://www	Article	Final	Open Access	Scopus	English	China; Coror
10.1080/222	https://www	Letter	Final	Open Access	Scopus	English	virus RNA; B

Publisher	ISSN	Affiliation	Country	Quartile	Rank
Taylor and Fr	16549880	Department	Sweden	Q1	47
NLM (Medlir	22221751	Department	China	Q1	24

Generacion RDF (Jena)

Recurso	URI
Base	http://utpl.edu.ec/COVIDBiblio/ontology/
Author	http://utpl.edu.ec/COVIDBiblio/ontology/Author/ + **ID_AUTOR**
Scientific Database	http://utpl.edu.ec/COVIDBiblio/ontology/ScientificDatabase/ + **NOMBRE FUENTE**
Catalog	http://utpl.edu.ec/COVIDBiblio/ontology/Catalog/ + **ID_CATALOGO**
Dataset	http://utpl.edu.ec/COVIDBiblio/ontology/Dataset/ + **ID_DAT**
Documento (Articulo, Letter, ConferencePaper)	http://utpl.edu.ec/COVIDBiblio/ontology/BibliographicResource/ + **EID DOCUMENTO**

Generacion RDF (Jena)

https://github.com/diepinto30/GenerateBibliometricDataCovid

Incluir los prefijos que se utilizaran en nuestro modelo

```
// CREACION DE SCIENTIFIC DATABASE
Resource fuenteDocumento = model.createResource(URI_FUENTE)
        .addProperty(RDF.type, myOntoModel.getResource(dataPrefix + "/ScientificDatabase"))
        .addProperty(FOAF.name, fuente)
        .addProperty(RDFS.subClassOf, FOAF.Organization);

//CREACION DEL DATASET
//String URIDataset = dataPrefix + "Dataset/BiblioDataCovid/";
Resource datasetInfo = model.createResource(URI_DATASET)
        .addProperty(RDF.type, dcatModel.getResource(dcat + "Dataset/")
                .addProperty(DCTerms.title, "BiblioDataCovid")
                .addProperty(dcatModel.getProperty(dcat + "keyword/"),("covidl9; sars-cov-2"))
                .addProperty(DCTerms.modified, "10-06-2020"));

//CREACION DEL CATALOG
//String URICatalog = dataPrefix+"Catalog/CatalogScopusCOVID/";
Resource catalog = model.createResource(URI_CATALOG)
        .addProperty(DCTerms.title, "CatalogScopusCOVID")
        .addProperty(RDF.type, dcatModel.getResource(dcat + "Catalog/")
                .addProperty(DCTerms.publisher, fuenteDocumento)
                .addProperty(dcatModel.getProperty(dcat +"dataset"), datasetInfo ));
```

Generacion RDF (Jena)

Creacion de SCIENTIFIC DATABASE -CATALOG - DATASET

```
// CREACION DE SCIENTIFIC DATABASE
Resource fuenteDocumento = model.createResource(URI_FUENTE)
        .addProperty(FOAF.name, fuente)
        .addProperty(RDFS.subClassOf, FOAF.Organization);

//CREACION DEL DATASET
String URIDataset = dataPrefix + "Dataset/BiblioDataCovid/";
Resource datasetInfo = model.createResource(URIDataset)
        .addProperty(RDF.type, dcatModel.getResource(dcat + "Dataset/")
                .addProperty(DCTerms.title, "BiblioDataCovid")
                .addProperty(dcatModel.getProperty(dcat + "keyword/"),("covidl9; sars-cov-2"))
                .addProperty(DCTerms.modified, "10-06-2020"));

//CREACION DEL CATALOG
String URICatalog = dataPrefix+"Catalog/CatalogScopusCOVID/";
Resource catalog = model.createResource(URICatalog)
        .addProperty(DCTerms.title, "CatalogScopusCOVID")
        .addProperty(RDF.type, dcatModel.getResource(dcat + "Catalog/")
                .addProperty(DCTerms.publisher, fuenteDocumento)
                .addProperty(dcatModel.getProperty(dcat +"dataset"), datasetInfo ));
```

Generacion RDF (Jena)

Creación de Recurso bibliográfico (Article-Letter-ConferencePaper...)

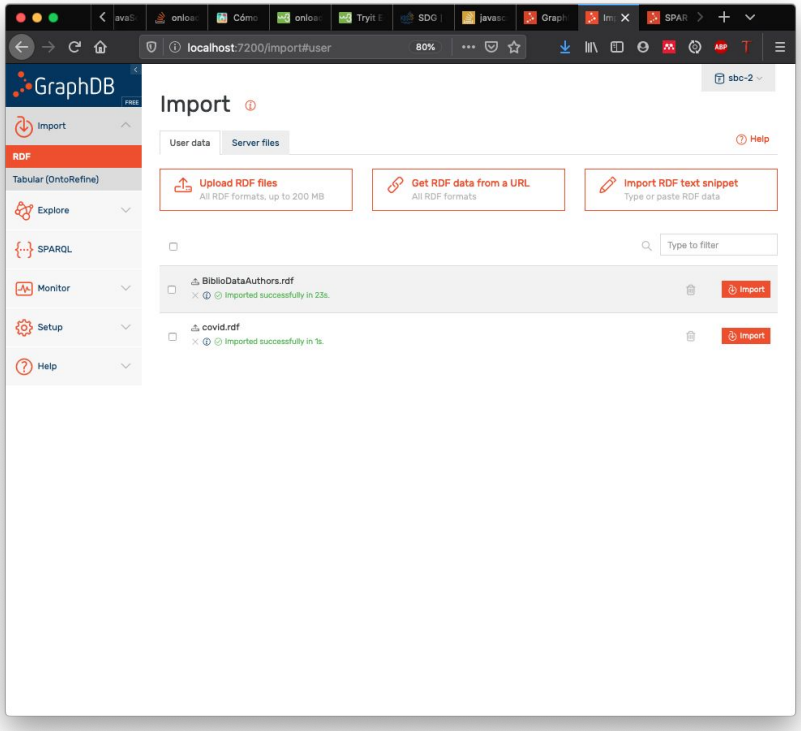
```
// CREACION DE DOCUMENTO BIBLIOGRAFICO
Resource documento = model.createResource(URI_DOCUMENTO)
        .addProperty(DCTerms.title, titulo)
        .addProperty(DCTerms.date, anio)
        .addProperty(myOntoModel.getProperty(dataPrefix +"citationsCount"), num_citas)
        .addProperty(DCTerms.language, dboModel.getResource(dbr+language))
        .addProperty(RDFS.subClassOf, fabioModel.getResource(fabio+ "ScholaryWork/")
        .addProperty(prismModel.getProperty(prism +"doi/"), doi)
        .addProperty(prismModel.getProperty(prism +"volume/"), vol)
        .addProperty(provModel.getProperty(prov + "wasDerivedFrom"), datasetInfo));
// Se crea el tipo de documento
// Se compara el tipo con los de fabio y si es igual toma la uri de fabio
for (String nombre : tipos_Fabio) {
    if(document_type.replace(" ","").equals(nombre)){
        documento.addProperty(RDF.type, fabioModel.getResource(fabio + document_type.replace(" ","")))
    }else{
        documento.addProperty(RDF.type, myOntoModel.getResource(dataPrefix + document_type.replace(" "
    }
}
```

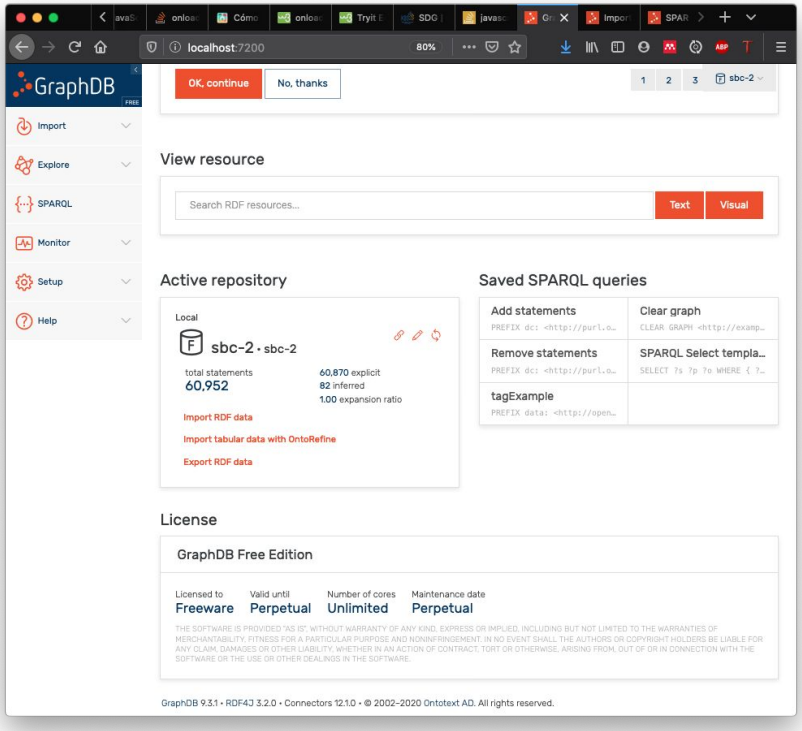
Generacion RDF (Jena)

Creación de Autores y vinculación al Recurso Bibliográfico

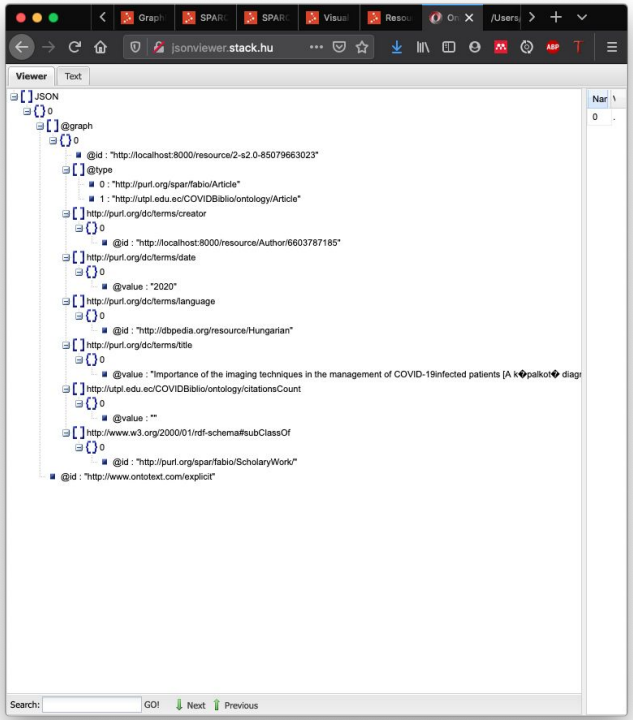
```
for (int j = 0; j < parts_id_autores.length; j++) {
    // CREANDO INSTANCIAS DE LOS AUTORES
    int long_parts_nombres = parts_nombres.length;
    String URI_AUTOR = dataPrefix+ "Author/" + parts_id_autores[j];
    Resource autor = model.createResource(URI_AUTOR)
            .addProperty(RDF.type, myOntoModel.getResource(dataPrefix+"Author"))
            .addProperty(RDFS.subClassOf, FOAF.Person);
    if (j<long_parts_nombres) {
        autor.addProperty(FOAF.name, parts_nombres[j]);
    }
    // Vinculando el autor al documento
    documento.addProperty(DCTerms.creator, autor);

}
```

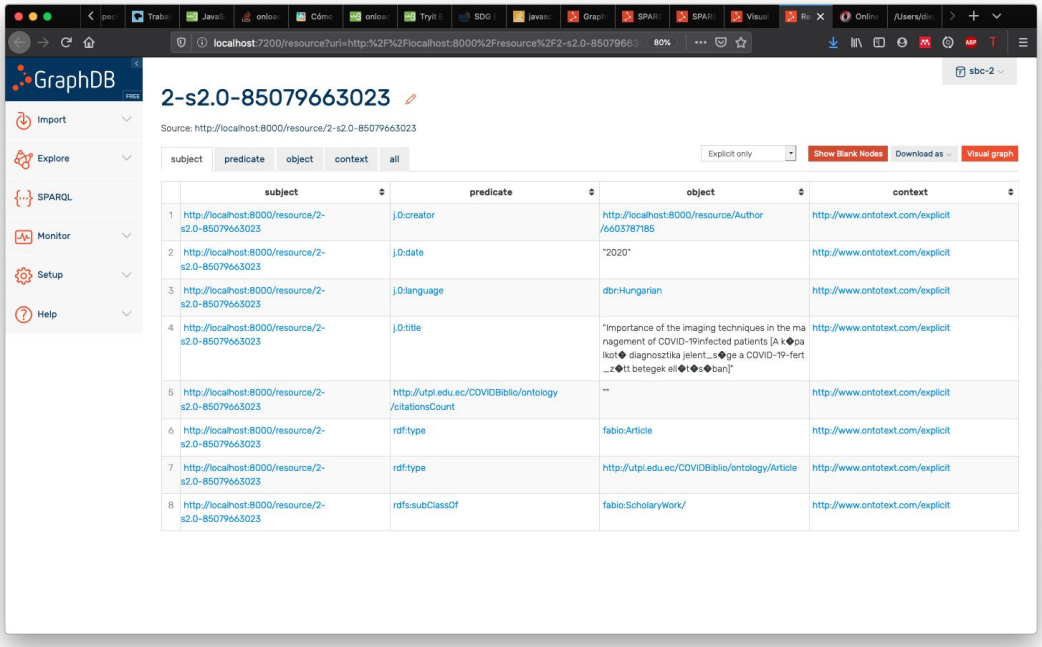
Almacenador de datos semánticos - GraphDB



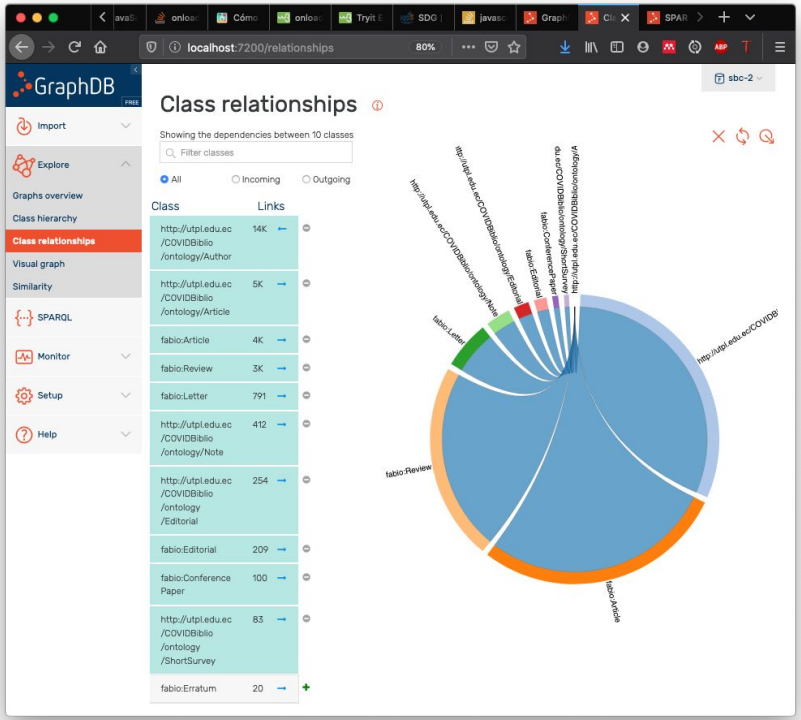


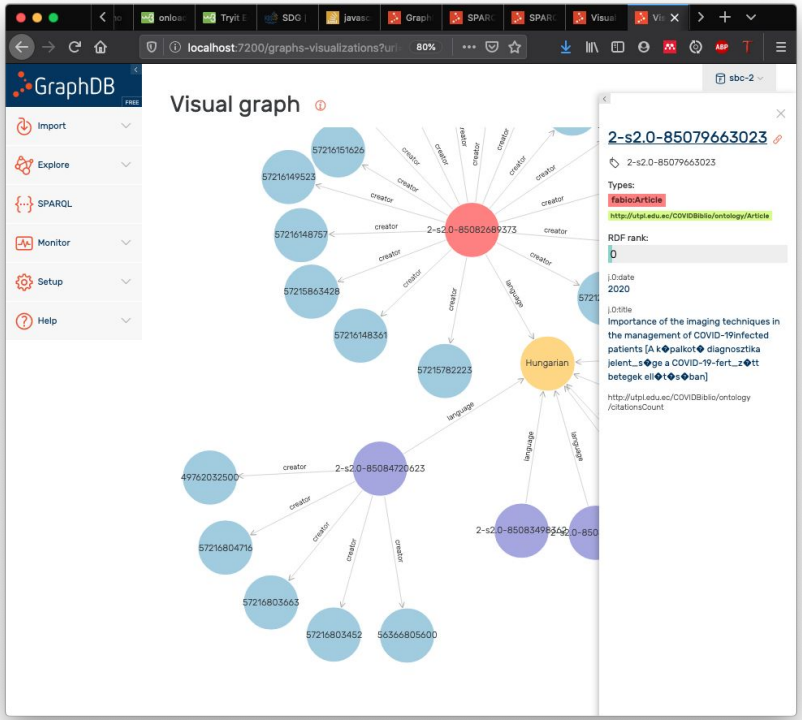
Almacenador de datos semánticos - GraphDB





Almacenador de datos semánticos - GraphDB





Gracias