Facultad de Ingeniería
Departamento de Ingeniería de Sistemas y Computación
Maestría en Ingeniería de Sistemas y Computación
ISIS 4514 – Semantic Web

Estudiante	Código	Correo
Juan Esteban Rodríguez	202011171	j.rodriguezo@uniandes.edu.co
Julian Camilo Mora Valbuena	202012747	j.morav@uniandes.edu.co
Andrea Isabel Vega Márquez	202416862	ai.vega@uniandes.edu.co

1. Introducción

Esta etapa del proyecto se centra en el modelado de una ontología y la creación de consultas SPARQL. Tras la recopilación de una base de datos de al menos 5000 artículos y realizar las correcciones sugeridas en la fase anterior, el objetivo de esta entrega es organizar la información recolectada en una ontología coherente, con la capacidad de recibir y manejar consultas. Esto implica identificar tipos de datos, establecer clases y relaciones, definir restricciones y características, y crear instancias para representar cada artículo. De esta manera, para cumplir con los objetivos de esta entrega utilizamos herramientas como Grobid y DBpedia Spotlight, con el fin de extraer metadatos y anotaciones de los artículos, respectivamente.

2. Consideraciones:

Antes de comenzar la nueva fase, revisamos y ajustamos nuestra entrega anterior según las observaciones y sugerencias recibidas durante la sustentación previa, con el objetivo de poder realizar adecuadamente los requisitos funcionales de la segunda entrega. Con esto en mente, para extraer información adicional de nuestros documentos PDF, empleamos Grobid como herramienta de análisis. Utilizamos esta herramienta para analizar todos los PDF almacenados, con el propósito de extraer datos como el abstract, las palabras clave, la introducción y las conclusiones de estos. Este proceso se realizó mediante una consulta al archivo CSV que contiene los metadatos de nuestros artículos. Por cada entrada en el CSV, es decir, por cada artículo en nuestra base de datos, empleamos Grobid para extraer los datos correspondientes y los añadimos como columnas adicionales en nuestro CSV.

Finalmente, con un CSV enriquecido y más completo, nos preparamos para abordar los requisitos específicos de esta entrega.

3. Creación de la ontología

Clases:

Se definen dos clases principales: paper:Paper y persona:Persona. paper:Paper representa los artículos académicos, mientras que persona:Persona representa a las personas asociadas con esos artículos, como autores.

Propiedades:

Se definen varias propiedades para describir las relaciones entre los artículos y las personas, así como otros detalles relacionados con los artículos. Por ejemplo:

paper:autor: Esta propiedad se utiliza para relacionar una persona (instancia de persona:Persona) con un artículo (instancia de paper:Paper), indicando que la persona es un autor del artículo.

id_namespace:authorId: Esta propiedad se utiliza para representar el ID del autor. Se espera que tenga un dominio de persona:Persona y un rango de xsd:string, lo que significa que el ID del autor debe ser una cadena.

url_namespace:semanticUrl: Esta propiedad se utiliza para representar la URL semántica asociada con una persona. Tiene un dominio de persona:Persona y un rango de xsd:string.

Otras propiedades como abstract:tieneAbstract, introduccion:tieneIntroduccion, keywords:tieneKeyWords, etc., se utilizan para describir diferentes aspectos de un artículo, como su resumen, introducción, palabras clave, etc.

Restricciones:

Se establece una restricción de cardinalidad mínima para la propiedad autor de un paper, asegurando que cada paper tenga al menos un autor.

Se impone una restricción de cardinalidad exacta para la propiedad tieneIntroduccion, asegurando que cada paper tenga exactamente una introducción.

También se define una restricción de cardinalidad mínima para la propiedad autor de una persona, asegurando que cada persona esté relacionada con al menos un paper como autor.

4. Estructura del repositorio:

Nuestro repositorio se encuentra estructurado de la siguiente manera:

SemanticWeb/

/Docs/

DocEntrega1

DocEntrega2 ---------- Documento de la entrega

/Entrega1/

/Entrega2/

/Contenedores/---------- Directorio en el que almacenamos los contenedores de Docker

Docker-compose.yml ------- Contenedor con la imagen de Grobid

Spotlight-compose.yml ----- Contenedor con la imagen de DBpedia Spotlight

/CSVs/ ---------- Directorio en el que se almacenamos los CSVs utilizados

/DescargasPDFs/---------- Directorio en el que almacenamos los PDFs descargados

/neo4j/---------- Directorio creado en al MV para utilizar neo4j

/Ontología/---------- Directorio en el que almacenamos los archivos .rdf creados

interference.rdf ------ Ontología creada para utilizar en neo4j (también está en neo4j/data)

ontologia.rdf --------- Ontología creada

esquema.rdf ---------- Esquema creado

analisis_pdf.py ---------- Script utilizado para extraer información con Grobid

busqueda_anotaciones.py - Script utilizado para extraer entidades con Dbpedia Spotlight

crear_ontologia.py ---------- Script utilizado para crear la ontología

.gitignore ---------- Se ignoran los directorios CSVs y DescargasPDFs por su tamaño

README.md

requirements.txt ---------- Requerimientos necesarios para ejecutar los scripts

En caso de tener dudas sobre la ejecución de nuestro proyecto, toda la documentación respectiva se encuentra en el Readme de nuestro repositorio: https://github.com/julian27m/SemanticWeb

En la imagen a continuación se puede apreciar la estructura completa de la entrega:

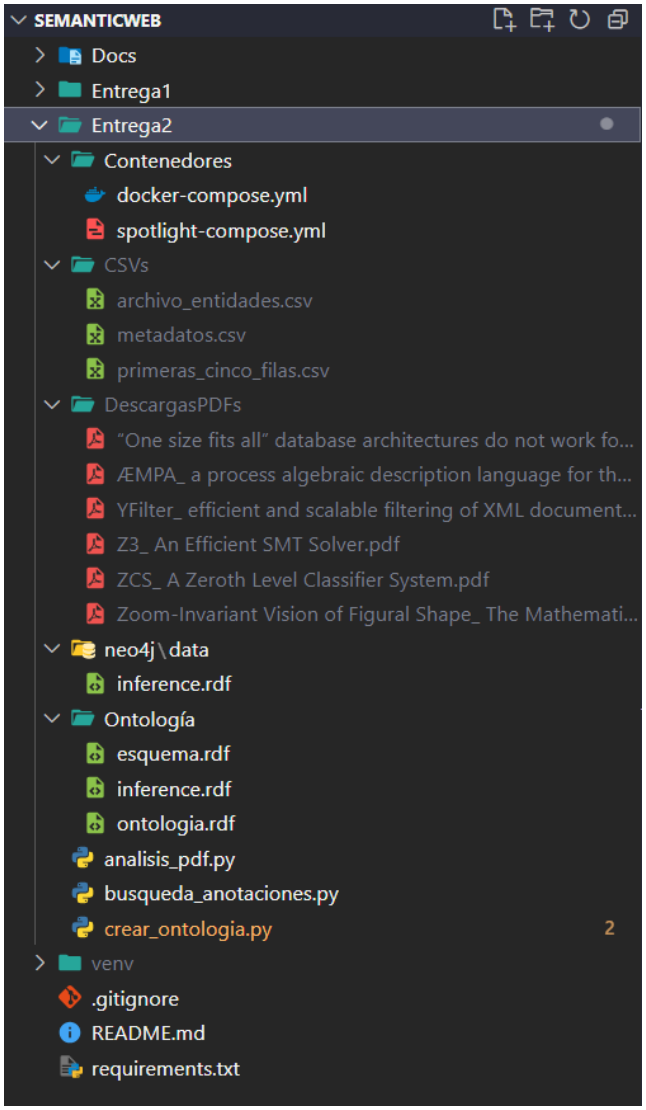

Ilustración 1: estructura entrega 2

5. Sentencias SPARQL:

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX paper: <http://example.org/paper#>

PREFIX persona: <http://example.org/persona#>

```
SELECT ?paper1 ?paper2
WHERE {
  ?paper1 paper:autor ?author .
  ?paper2 paper:autor ?author .
  FILTER (?paper1 != ?paper2)
}

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX paper: <http://example.org/paper#>
PREFIX revista: <http://example.org/revista#>

SELECT ?paper1 ?paper2
WHERE {
  ?paper1 paper:venue ?journal .
  ?paper2 paper:venue ?journal .
  FILTER (?paper1 != ?paper2)
}

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX paper: <http://example.org/paper#>

SELECT ?paper1 ?paper2
WHERE {
  ?paper1 paper:tieneKeyWords ?keywords .
  ?paper2 paper:tieneKeyWords ?keywords .
  FILTER (?paper1 != ?paper2)
}
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX paper: <http://example.org/paper#>

SELECT ?paper
WHERE {
  ?paper paper:perteneceA ?campoDeEstudio .
  FILTER (?campoDeEstudio = "campo de estudio específico")
}
```

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX paper: <http://example.org/paper#>

SELECT ?paper
WHERE {
  ?paper paper:perteneceA <http://example.org/campoDeEstudioEspecifico> .
}
```

6. Notas:

Se han establecido relaciones entre las entidades para capturar la estructura y la semántica de un paper y su contexto.

Se han definido propiedades tanto de objeto como de datos para capturar tanto las relaciones entre entidades como los atributos específicos de cada una.

Se han aplicado restricciones para garantizar la consistencia y la integridad de los datos, como la presencia mínima de autores en un paper.

La ontología proporciona una estructura flexible que puede extenderse y adaptarse según las necesidades específicas del dominio de los papers.