Buscador Semántico: Para facilitar la detección de los artículos más importantes para el campo de la medicina respecto al COVID

Cynthia Gaviria Castaño

Informe de practica presentado para optar al título de:
Ingeniero de Sistemas

Asesora
Astrid Duque Ramos, Doctora en Ingeniería informática (Interna)

Universidad de Antioquia
Facultad de Ingeniería
Ingeniería de Sistemas
Medellín, Antioquia, Colombia
2023

Cita	(Gaviria Castaño, 2023)
Referencia **Estilo APA 7 (2020)**	Gaviria Castaño, C (2023). *Buscador Semántico: Para facilitar la detección de los artículos más importantes para el campo de la medicina respecto al COVID* [Informe de práctica]. Universidad de Antioquia, Medellín, Colombia.

Grupo de Investigación Ingeniería y Tecnologías de las Organizaciones y de la Sociedad (ITOS).

UNIVERSIDAD DE ANTIOQUIA
Vicerrectoría de Docencia

Sistema de Bibliotecas

Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: http://bibliotecadigital.udea.edu.co

Universidad de Antioquia - www.udea.edu.co

Dedicatoria

Dedico este trabajo a mi madre, quien siempre han sido mi inspiración.

Agradecimientos

Agradezco a la Universidad de Antioquia por proporcionarme el ambiente y las herramientas necesarias para mi formación académica

Tabla de contenido

Lista de tablas

Lista de figuras

Resumen

El COVID-19 y la lucha contra su propagación, ha incentivado un auge en la investigación científica asociada al desarrollo de vacunas y medidas preventivas para combatir esta epidemia. Millones de artículos sé han generado y cada vez es más difícil encontrar la información más pertinente para el investigador y sus objetivos.

Con el fin de dar solución a esta problemática, el grupo de investigación ITOS de la universidad de Antioquia propone el proyecto “Buscador semántico para facilitar la detección de los artículos más importantes para el campo de la medicina respecto al COVID”, que hace uso de la web semántica y métricas de citaciones para mejorar la búsqueda y recuperación de artículos y publicaciones más relevantes dentro de este dominio.

Este trabajo de grado apoya el proyecto mencionado, centrándose en la clasificación de los documentos a partir de un conjunto de ontologías.

Abstract

The COVID-19 and the fight against its spread have incentivized a surge in scientific research associated with the development of vaccines and preventive measures to combat this epidemic. Millions of articles have been generated, and it is increasingly challenging to find the most relevant information for researchers and their objectives.

To address this issue, the ITOS research group at the University of Antioquia proposes the project "Semantic Search Engine to Facilitate the Detection of the Most Important Articles in the Field of Medicine Regarding COVID." This project utilizes semantic web and citation metrics to improve the search and retrieval of more relevant articles and publications within this domain.

This thesis supports the aforementioned project, specifically focusing on the classification of documents based on a set of ontologies.

Introducción

El coronavirus (COVID-19) es una enfermedad infecciosa ocasionada por el virus SARS-CoV-2 que apareció por primera vez en diciembre del 2019, cuya expansión global fue la responsable de una pandemia, la cual ha dejado hasta agosto del 2022 más de 535 millones de casos confirmados y 6 millones de defunciones, aunque sé estima que el total puede estar en el rango de entre 15 a 25 millones de fallecidos.

Ante esta crisis sanitaria hubo un auge investigativo, la búsqueda de una vacuna y/o tratamiento sé convirtió en una prioridad, dando como resultado un incremento significativo en la cantidad de artículos y publicaciones relacionadas al virus; generando así la necesidad de compartir esta información de manera abierta y distribuida entre los investigadores. Diferentes grupos editoriales tales cómo Springer Nature, Science, Taylor & Francis entre otras acordaron publicar en abierto todos los artículos, libros, conferencias y demás relacionadas al COVID-19.

La disponibilidad de toda esta información dificulta a los investigadores y demás personas interesadas, la búsqueda de artículos qué sean relevantes para los propósitos concretos que se tienen. Organizaciones tales como "Cochrane" utilizan un proceso de revisión de la literatura manual, o motores de búsqueda como "COVIDScholar" herramienta que permite la búsqueda de artículos relacionados con el COVID-19 a partir del procesamiento del lenguaje natural. Así mismo, la universidad de Antioquia propone un proyecto, cuyo objetivo es construir un Sistema semántico y automatizado de búsqueda que optimice los resultados de investigación publicados en artículos científicos para responder a la necesidad de información relacionada al COVID-19.

Este proyecto, llamado "Buscador semántico para facilitar la detección de los artículos más importantes para el campo de la medicina respecto al COVID" está siendo ejecutado por el grupo de investigación Ingeniería y Tecnología de las Organizaciones y la Sociedad (ITOS) en conjunto con el Sistema de Bibliotecas y el Colaboratorio de Vinculación para las Ciencias Sociales Computacionales y las Humanidades Digitales (CoLaV), Dentro de este proyecto se enmarca el presente trabajo de grado, que se enfoca, específicamente en la implementación de un Clasificador qué permita la agrupación de los artículos científicos a partir de diferentes etiquetas.

Objetivos.

Objetivo general.

Implementar una herramienta que permita la clasificación y agrupación de artículos relacionados al COVID-19 previamente etiquetados.

Objetivos específicos.

- Analizar los anotadores que se usarán en el proyecto entre las opciones otorgadas.
- Crear un repositorio para el almacenamiento de las ontologías.
- Implementar un servicio que pueda consumir y procesar los resultados obtenidos a partir del anotador.
- Procesar los resultados obtenidos y agruparlos a partir de las etiquetas.
- Ordenar los artículos por el número de apariciones en las diferentes etiquetas.
- Diseñar, Implementar y Exponer una API, que pueda ser consumida por un front-end que retorne los resultados finales.

Marco teórico.

La Web Semántica es la siguiente generación de la Web donde la información además de mostrar datos estáticos ayuda con la automatización de búsquedas, la integración de sistemas y la reutilización de componentes entre aplicaciones. (Vesin et al., 2016, 116). Siempre ha sido un anhelo por parte de los investigadores construirla desde la visión de Tim Berners-Lee, quien expone por primera vez su punto de vista sobre el tema en el año de 1999 (Berners-Lee, 2000):

"*Tengo un sueño para la web en el qué los computadores son capaces de analizar todos los datos de la web -contenido, links, y transacciones entre personas y computadores. La Web Semántica, es la qué lo hace posible, todavía no emerge, pero cuando lo haga, los mecanismos de intercambio del día a día, la burocracia y nuestras vidas diarias serán manejadas por máquinas. Los agentes inteligentes, que la gente ha pregonado durante años, finalmente se materializarán*"

Los principales componentes de la web semántica son los metalenguajes y los estándares de representación. Tales Como XML, XML Schema y OWL, siendo este último un acrónimo para "Web Ontology Language" (OWL Web Ontology Language Guide, 2004), lenguaje que permite la representación explícita de una ontología, vocabulario que define clases, entidades, propiedades, relaciones y demás entre estos componentes, tomando un papel fundamental para la interoperabilidad semántica entre los sistemas.

A partir de su etimología podemos decir que el concepto de ontología se deriva del griego ontos qué significa ser o existir y logos qué significa ciencia o conocimiento, por lo tanto, una ontología es el estudio de las cosas qué existen en un área o dominio particular (Gruber, 1995). La primera definición que se puede encontrar sobre ontología en el ámbito de la Inteligencia Artificial es:

"*Una ontología define los términos básicos y relaciones que conforman el vocabulario de un área específica, así como las reglas para combinar dichos términos y las relaciones para definir extensiones de vocabularios*." (Neches y otros, 1991)

Pero más adelante Tom Gruber diría que “Una ontología es una especificación explícita de una conceptualización [....] Así, en el contexto de IA, podemos describir la ontología de un programa como un conjunto de términos. En tal ontología, las definiciones asocian nombres de entidades del universo del discurso con textos comprensibles por los humanos que describen el significado de los nombres, y axiomas formales que limitan la interpretación y buen uso de dichos términos. Formalmente, una ontología es una teoría lógica.” (Gruber, 1993) convirtiéndose esta en la definición más ampliamente difundida. Existen múltiples formas de clasificación para las ontologías, pero sin llegar a profundizar estas sé pueden clasificar en (Lucas & Rubio, 2001):

• **Ontologías de nivel superior**: Describen conceptos más generales. En relación con los sistemas de información, estas ontologías describirían conceptos básicos.

• **Ontologías De Tareas o de Técnicas básicas**: Describen una tarea, actividad o artefacto.

• **Ontologías de dominio**: Describen un vocabulario relacionado con un dominio genérico.

• **Ontologías de Aplicación**: Describen conceptos que dependen tanto de un dominio específico como de una tarea específica y, generalmente son una especialización de ambas.

Entre las aplicaciones de las ontologías se tiene la indexación, la recuperación de documentos con respecto a sus entidades asociadas y la anotación semántica la cual se define como "la acción y los resultados de describir (parte de) un recurso electrónico mediante metadatos cuyo significado se especifica formalmente en una ontología" (Fernandez, 2010), Siendo este un enfoque para vincular las ontologías a las fuentes de información originales (Lin, 2008). Por ejemplo, BioPortal es un proyecto del National Center for Biomedical Ontology (NCBO), el cual cuenta con el mayor repositorio de ontologías biomédicas.

Y además ofrece el BioPortal Annotator, cuya funcionalidad se encarga de procesar texto enviado por el usuario, reconocer los términos relevantes en el texto y devolver las anotaciones al usuario. En la Ilustración 1 se puede observar el grafo para uno de los conceptos pertenecientes a la ontología SNOMED CT,

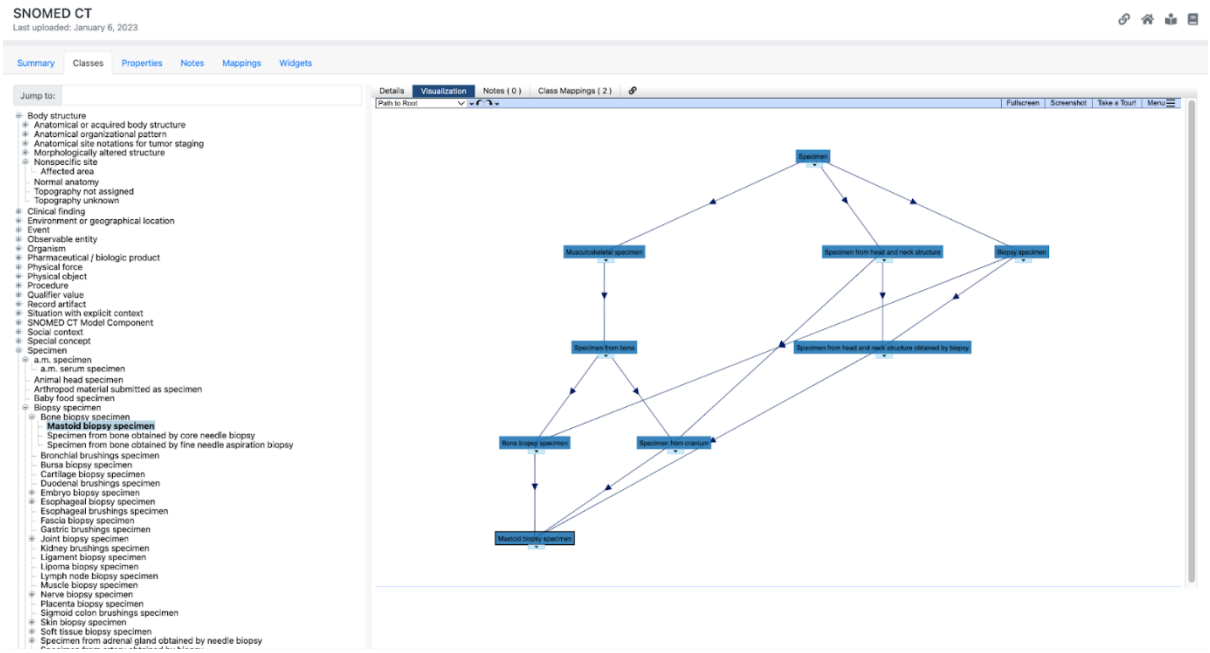

Ilustración 1. SNOMED CT Ontología en Bioportal

Por otra parte, también es importante tener en consideración las tecnologías usadas dentro de la implementación del proyecto. Originalmente Abad-Navarro, Bernabé-Diaz,García-Castro y Fernandez-Breis (2021) ya realizado un avance en la creación de un anotador para artículos científicos pertenecientes a Pubmed, en conjunto con las ontologías seleccionadas por expertos de la biblioteca.

También se tiene en consideración la definición de índice invertido (Witten, Moffat, & Bell, 1999), siendo esta una técnica utilizada para estructurar la información que será recuperada por un motor de búsqueda, con el propósito de facilitar búsquedas de texto completas. En esta metodología, el buscador crea índices o términos de búsqueda a partir de un conjunto de documentos, señalando qué documentos contienen dichos términos. De esta forma, cuando un usuario ingresa un término específico de búsqueda, el motor de búsqueda le muestra los documentos que contienen dicho término, permitiendo una recuperación más eficiente y precisa de la información relevante. Este índice se implementa con MongoDB (Banker, 2011), la cual es una base de datos orientada a documentos, lo que significa que almacena los datos en forma de documentos JSON (Crockford, 2006). Es altamente escalable y permite una recuperación de datos rápida y eficiente.

PostgreSQL (Stonebraker & Rowe, 1986) es una base de datos relacional sólida y eficiente que nos permite obtener los datos necesarios para enriquecer la información presentada en el buscador semántico.

Además, se utiliza Python (Van Rossum & Drake, 2009) en conjunto a Flask (Ronacher, 2010) para crear una API REST (Fielding & Taylor, 2002). Esta API actúa como una interfaz que permite a la interfaz frontal (front-end) realizar consultas y enviar solicitudes al buscador semántico.

La elección de Flask como framework para la creación de la API REST se debe a su simplicidad y facilidad de uso, lo que ha permitido desarrollar rápidamente una interfaz eficiente y funcional para el front-end.

Metodología

Metodología de desarrollo

La metodología para el desarrollo del proyecto sé implemento por etapas definidas a continuación.

Contextualization.

- Estudiar los anotadores
- Entender las ontologias seleccionadas.

Análisis.

- Analizar los datos de los artículos a clasificar.
- Levantar requisitos funcionales y no funcionales.
- Definición del modelo de dominio.

Diseño.

- Definición de la arquitectura del sistema.
- Definición de las tecnologías que se utilizarán en el desarrollo.
- Definir métodos de conexión con los otros servicios y plataformas utilizados por el proyecto.
- Realizar diagramas de componentes y despliegue para definir esquemas de conexión entre componentes.
- Definir los criterios de estandarización para la utilización de las herramientas de monitoreo. Definir criterios de medición que nos permitan evaluar el desempeño y porcentaje de éxito del proyecto.

Implementacion.

- Crear una POC (Proof of concept) o prueba de concepto inicial para realizar acercamientos iniciales a las herramientas a utilizar.
- Desarrollar el clasificador que cumpla con los estándares definidos previamente.

Despliegue.

- Entregar una versión funcional del clasificador y publicar en la plataforma de artefactos del proyecto.
- Desplegar el proyecto funcional para su utilización por parte del grupo de investigación en los procesos internos.

Metodología de seguimiento.

En el desarrollo del proyecto del buscador semántico, se adoptó una metodología ágil para gestionar y avanzar en el proceso de implementación de manera eficiente. Se utilizó la metodología Scrum como marco de trabajo ágil para organizar y coordinar las actividades del equipo, adicionalmente, se realizaron reuniones esporádicas con los demás miembros del grupo de investigación y otros miembros que trabajan en proyectos relacionados, cómo otros estudiantes y la profesora encargada del proyecto, para presentar avances y tomar decisiones sobre aspectos críticos del proyecto.

Resultados.

A través del apartado que se presenta a continuación, se proporcionará una visión detallada de los resultados de la implementación del buscador semántico. El objetivo es mostrar el proceso de análisis, diseño, desarrollo e implementación del proyecto.

Definición de estándares.

Los estándares para el proyecto del buscador semántico son guías para facilitar el desarrollo en el proyecto. Estos estándares se aplican en la implementación del buscador y los diferentes componentes del sistema. Para ello, se han definido los siguientes criterios:

Documentación completa:

Sé proporciona documentación detallada que explica cómo utilizar el buscador, las funcionalidades que ofrece y cómo interpretar los resultados. Esta documentación ayuda a evitar malentendidos y garantiza que todos tengan acceso a la misma información.

Versionado y control de cambios:

El proyecto ha implementado un sistema de versionado basado en Git. Este es ampliamente utilizado en el desarrollo de software debido a su capacidad para mantener un historial detallado de las modificaciones realizadas en cada archivo.

También sé hace uso de GitHub como repositorio central, donde los miembros del equipo pueden compartir y colaborar en el código fuente de manera segura y confiable. Esta herramienta proporciona una plataforma en la nube que permite almacenar, gestionar y revisar el código, además de ofrecer herramientas para resolver conflictos y fusionar los cambios sin problemas.

Manejo de errores:

Se incorporó un sistema de manejo de errores que desempeña un papel crucial en el correcto funcionamiento del buscador semántico. A través de la validación de campos y la captura de excepciones, este sistema permite detectar y reconocer posibles problemas y errores que puedan surgir durante la operación del buscador.

La validación de campos asegura que los datos ingresados por los usuarios sean adecuados y coherentes, evitando así posibles errores y malfuncionamientos. Por otro lado, la captura de excepciones permite identificar y registrar situaciones o errores inesperados en el código, lo que facilita la rápida resolución de problemas por parte del equipo de desarrollo.

Mejora continua:

Se requirió una revisión continua y refinamientos iterativos durante el proyecto para acomodar las necesidades emergentes surgidas en el desarrollo. Durante este proceso, es frecuente que nuevas necesidades y desafíos se presenten, requiriendo un grado de flexibilidad y adaptabilidad para incorporar cambios que respondan de manera efectiva a estos escenarios dinámicos.

Posterior a cada revisión, fue necesario evaluar la pertinencia de los resultados obtenidos, generar mejoras, y en caso de que se identificaran áreas de optimización, adaptarse a la nueva situación.

Contextualización.

En la etapa inicial de este estudio, se presentaban dos posibles anotadores para la clasificación de artículos. Inicialmente, se realizaron pruebas con el anotador de Bioportal, no obstante, debido a la voluminosa cantidad de artículos que requerían procesamiento, y el tiempo de respuesta de anotador, se optó por la segunda alternativa. Por lo tanto, se utilizó el anotador proporcionado por la Universidad de Murcia, un método ya adoptado en la investigación de Abad-Navarro, F, Bernabé-Diaz, J. A., García-Castro, A., & Fernandez-Breis, J. T.

Para realizar las anotaciones, se emplearon las ontologías detalladas en la Tabla 1. Lista de Ontologías. Cabe destacar que la selección de estas ontologías fue realizada por expertos en el área y miembros del equipo de la biblioteca de la Universidad de Antioquia, garantizando su relevancia.

Ontology	Description	
	Complete Name	***# of Concepts***
SNOMED CT	*Ontology of Clinical Terms*	365.176
OCHV	*Ontology of Consumer Health Vocabulary*	369.665
PREMEDONTO	*Precision Medicine Ontology*	543
NCIT	*National Cancer Institute Thesaurus*	185.109
IOBC	*Interlinking Ontology for Biological Concepts*	126.847
HL7	*Health Level Seven Reference Implementation Model, Version 3*	9.079
COVID19	*COVID-19 Surveillance Ontology*	32
MESH	*Medical Subject Headings*	349.665
CIDO)	*Coronavirus Infectious Disease Ontology*	31.
CODO	*An Ontology for Collection and Analysis of COviD-19 Data*	90
HPIO	*Host Pathogen Interactions Ontology*	275
IDO-COVID-19	*The COVID-19 Infectious Disease Ontology*	362
VO	*Vaccine Ontology*	486
COVIDCRFRAPID	*WHO COVID-19 Rapid Version CRF semantic data model*	396
BAO	*BioAssay Ontology*	7.508
Total		1’446.853

Tabla 1. Lista de Ontologias

La tabla 1 proporciona una descripción detallada de cada ontología utilizada en este estudio. En ella, se puede apreciar el acrónimo, el nombre completo de la ontología y la cantidad de conceptos que se anotaron con su uso.

Análisis y diseño.

El conjunto de datos para este estudio se compone de 150,000 artículos provenientes de Kaggle, en específico del dataset CORD-19 proporcionado por Allen Institute for AI (2020), y luego recuperados desde PubMed en formato JAR.

Los artículos científicos fueron anotados utilizando las ontologías presentes en la tabla 1 y se encuentran almacenados en la base de datos de Neo4j. Este proceso de etiquetado y almacenamiento fue llevado a cabo con base en un trabajo previo realizado por el grupo

TECNOMOD de la Universidad de Murcia en España. Gracias a este enfoque, los artículos están organizados de manera estructurada y permiten una búsqueda de información relacionada con el campo científico específico. La estructura del grafo resultante sé puede observar en la Ilustración 2 descrita en el artículo “Semantic Publication of Agricultural Scientific Literature Using Property Graphs ”.

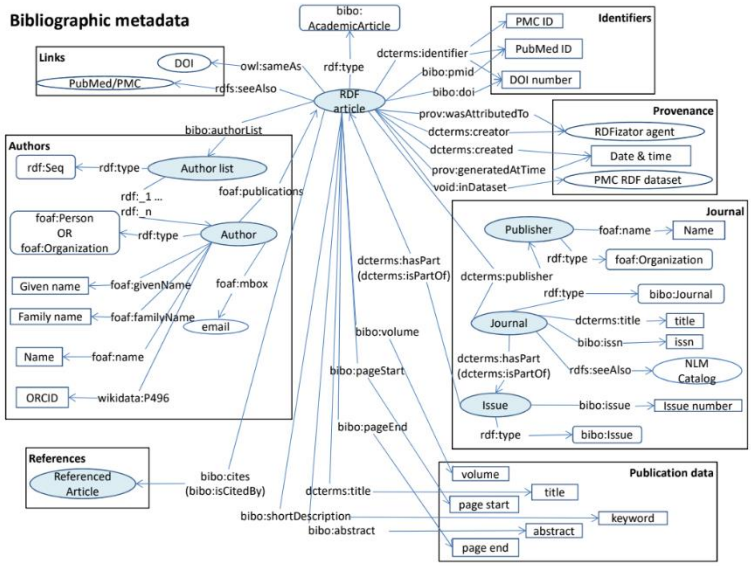

Ilustración 2. Esquema de datos de Neo4j. Imagen de "Semantic Publication of Agricultural Scientific Literature Using Property Graphs."

El grafo resultante consta de los artículos, junto con sus respectivas clasificaciones y conexiones, los cuales están almacenados en una base de datos de grafos en Neo4j como se muestra en la Ilustración 3. En esta base de datos, se registran los ID de los artículos, los metadatos asociados, los conceptos con los que fueron anotados, los sinónimos de estos conceptos, el número de veces qué un concepto fue anotado dentro del artículo, las ontologías y las referencias bibliográficas.

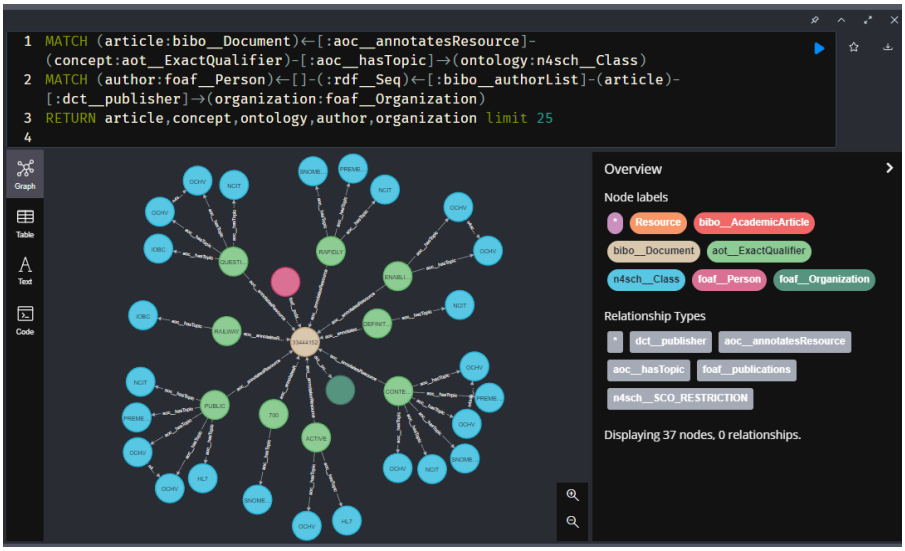

Ilustración 3. Resultado consulta de documento en Neo4j

Dada la cantidad de artículos, presentes en la base de datos sé pudo visualizar qué el tiempo de recuperación de las consultas realizadas a esta base de datos, es sustancial y no es viable para la implementación del buscador semántico. Por lo tanto, se diseñó un índice invertido utilizando MongoDB para mejorar la eficiencia en la recuperación de la información. El índice invertido en MongoDB permite mapear términos o palabras clave a los documentos en los que aparecen, lo que facilita y agiliza las búsquedas. Con esta nueva estructura de datos, los usuarios pueden realizar búsquedas basadas en palabras clave y recuperar rápidamente los artículos que necesitan. Esta implementación en MongoDB complementa la funcionalidad de Neo4j, permitiendo una experiencia de búsqueda más ágil y eficiente para los usuarios en el buscador semántico.

En la **¡Error! No se encuentra el origen de la referencia.** se presenta el esquema del í ndice invertido implementado utilizando MongoDB. A continuación, se explican los campos que componen este esquema:

```
{
      concepto: "String",
      synonyms: ["Array String"],
      ontologies: ["Array String"],
      articles: [{
            pmc_id: "String",
            matches: NumberInt,
            score: NumberInt
      }]
}
```

Ilustración 4. Esquema índice invertido

Concepto: Este campo almacena el término o palabra clave que se ha utilizado como índice para la búsqueda.

Synonyms: En este campo, se almacenan sinónimos que están relacionados con el concepto principal. Esto permite que diferentes términos relacionados con el mismo concepto sean considerados en las búsquedas.

Ontologies: En este campo se registran las ontologías o categorías específicas a las cuales pertenece el concepto

Articles: Este es un campo de tipo array que contiene la información sobre los documentos o artículos que contienen el concepto o término. Cada elemento del array corresponde a un artículo específico y puede contener información adicional como su ID, la cantidad de veces que el concepto aparece en el artículo (matches) y una puntuación o score que indica la relevancia del artículo para el término buscado.

Para la implementación de la arquitectura se definió un repositorio, el cual se ejecuta en dos diferentes pasos.

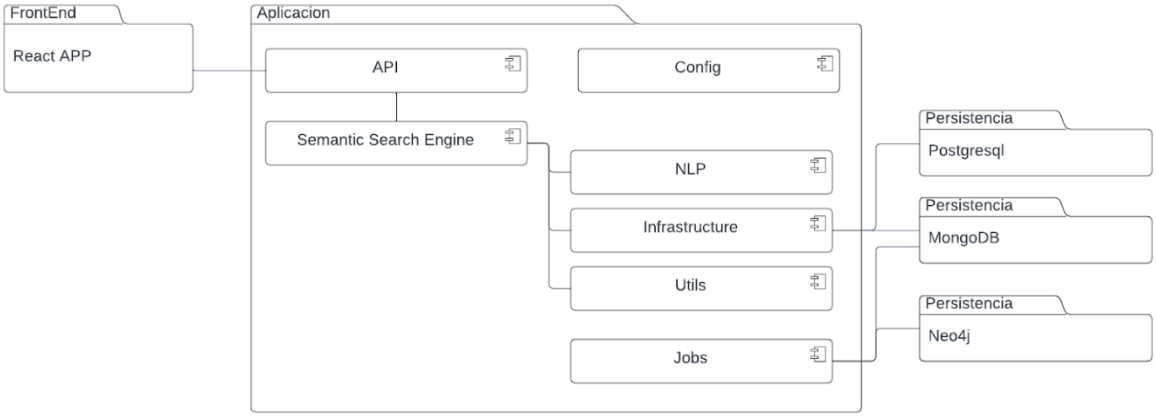

Ilustración 5. Diagrama de componentes

En primer lugar, se lleva a cabo una etapa inicial en la que se ejecutan los scripts almacenados en el módulo de Jobs que se puede ver en la ilustración 5, para procesar la población de la base de datos de MongoDB. Durante esta fase, los datos de Neo4j son analizados y posteriormente almacenados en MongoDB. Además, se generan los puntajes o scores asociados a

los artículos que serán utilizados en el proceso. Este puntaje se calcula haciendo uso de la Ecuación 1. frecuencia inversa de documento (IDF) llamada TF-IDF. Esta medida muestra la importancia de una palabra dentro de un documento en una colección o corpus.

$$tf(c,d) = \frac{Numero\ de\ apariciones\ del\ concepto\ en\ el\ documento}{Numero\ de\ conceptos\ anotados\ en\ el\ documento}$$

Ecuación 1. Formula IDF

El buscador semántico es expuesto a través de una API REST. Cuando una solicitud llega a la API, la cual contines una consulta generada por el usuario, para después iniciar el módulo del motor de búsqueda semántica. Aquí, se lleva a cabo un proceso de procesamiento de lenguaje natural (NLP) utilizando las bibliotecas SciPy y NLTK. Durante este paso, se eliminan las palabras irrelevantes (stop words), se realiza la lematización del texto y se estandariza, dando como resultado los conceptos relacionados con la consulta.

Una vez obtenido este resultado se procede a comunicarse con el módulo de infraestructura, donde se buscan estos conceptos en la base de datos de MongoDB. Si se encuentran resultados, se obtienen los artículos asociados a dichos conceptos. Posteriormente, se recuperan los metadatos de estos artículos desde PostgreSQL utilizando el módulo de Utils.

Finalmente, con la disponibilidad de los metadatos, se procede a ejecutar un mapeo de estos con el objetivo de generar la respuesta que se anticipa en la interfaz del usuario. Esta parte del procedimiento se realiza con la implementación de técnicas de paralelismo y concurrencia, lo que permite una ejecución más ágil y eficiente. Gracias a la simultaneidad de las operaciones y el manejo eficaz de recursos, se logra optimizar el tiempo de procesamiento y se mejora la capacidad de respuesta del sistema.

Despliegue.

En el desarrollo de este proyecto, debido a las demandas computacionales y de memoria implicadas en el procesamiento de las anotaciones, la base de datos de Neo4j, que alberga todos los artículos anotados, se ubicó en el servidor de la Universidad de Murcia.

Durante la etapa inicial del proyecto, se efectuó un procesamiento preliminar de la base de datos MongoDB en un entorno local, mediante la ejecución de los Jobs previamente descritos. Tras la finalización de estos jobs, se realizó una copia de seguridad, la cual se utilizaría en fases posteriores del proyecto.

En paralelo, se desplegó el buscador semántico, MongoDB y PostgreSQL en el servidor del Instituto de Física de la Universidad de Antioquia. Este despliegue se realizó utilizando Docker, como se ilustra en la Ilustración 6.

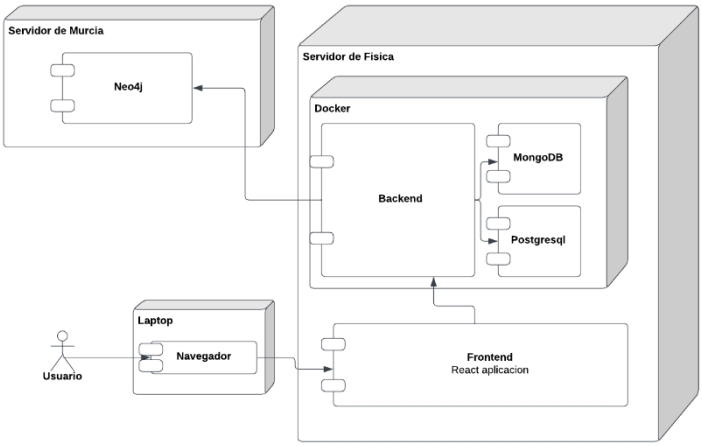

Ilustración 6. Diagrama de despliegue

En este mismo servidor, y como parte de otros proyectos emprendidos por el grupo de investigación, se desplegó una interfaz de usuario (frontend) desarrollada en React. Esta interfaz permite a los usuarios interactuar con el sistema a través de un navegador web, facilitando la ejecución de consultas y la navegación de manera.

En la fase inicial de despliegue de la aplicación, se llevaron a cabo procesos como la adquisición de las imágenes de MongoDB y PostgreSQL, tal como se evidencia en la Ilustración 7. Imagenes en Docker. Posterior a la obtención de estas imágenes, se emprendió la restauración de las bases de datos, un paso determinante en la configuración del sistema. A continuación, se procedió al arranque de los contenedores como se ve en la Ilustración 8. Contenedores en ejecución.

```
colav@gfif:~/semantics_covid$ sudo docker images
REPOSITORY                TAG       IMAGE ID       CREATED          SIZE
semantic_covid_backend    latest    c59765e210d6   8 days ago       929MB
mongo                     latest    1f3d6ec739d8   2 weeks ago      654MB
postgres                  latest    f0ff6ef79497   2 weeks ago      412MB
```

Ilustración 7. Imagenes en Docker

```
colav@gfif:~/semantics_covid$ sudo docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED      STATUS      PORTS                      NAMES
27261bfef0b0   semantic_covid_backend   "python3 src/main/ap…"   8 days ago   Up 8 days   0.0.0.0:8000->8000/tcp     semantic_covid_backend
4aafe206ffe7   mongo                    "docker-entrypoint.s…"   8 days ago   Up 8 days   0.0.0.0:27018->27017/tcp   semantic_covid_mongodb
ba5f50d23aae   postgres                 "docker-entrypoint.s…"   8 days ago   Up 8 days   0.0.0.0:5432->5432/tcp     semantic_covid_postgres
```

Ilustración 8. Contenedores en ejecución

Como resultado, ahora es posible realizar solicitudes al buscador semántico a través de su API REST, lo que permite a los usuarios enviar consultas de búsqueda y obtener resultados relevantes basados en la consulta realizada por los usuarios.

En la Ilustración 9, se muestra el resultado de una petición realizada al backend del buscador semántico para la consulta: "*COVID-19 in Europe: the Italian lesson*". En esta respuesta generada, se pueden observar diferentes detalles clave del artículo relacionado con la consulta realizada., la respuesta proporcionada se presenta en un arreglo de objetos JSON, cada uno de los elementos del arreglo representa un artículo pertinente. Dentro de este objeto JSON, se pueden identificar metadatos que incluyen el abstract, los autores, el identificador de objeto digital (DOI), una lista de identificadores, la revista de publicación, las métricas que indican la frecuencia de anotación de un concepto. También se incluye la puntuación que se calculó previamente utilizando la fórmula de la frecuencia inversa de documento (IDF, por sus siglas en inglés), especificada en la Ecuación 1. Además, se pueden identificar las ontologías con las que se anotó el artículo, la fecha de publicación, la fuente y el título.


```
// http://localhost:8000/api/search?query=COVID-19%20in%20Europe:%20the%20Italian%20lesson&ontology=OCHV&num_results=4&page=1

[
  {
    "abstract": "",
    "authors": [↔],
    "doi": "10.1016/s0140-6736(20)30690-5",
    "external_ids": [↔],
    "journal": "Lancet",
    "metrics": {↔},
    "ontologies": [↔],
    "pmid": "PMC7118630",
    "published_date": "2020-03-24",
    "source": "https://api.elsevier.com/content/article/pii/S0140673620306905; https://www.sciencedirect.com/science/article/pii/S01406736
    "title": "COVID-19 in Europe: the Italian lesson"
  },
  {↔},
  {↔},
  {↔}
]
```


Ilustración 9. Respuesta de la aplicación para el query: "COVID-19 in Europe: the Italian lesson "

Conclusiones

El uso de ontologías para la clasificación de documentos ha resultado ser un método eficaz para organizar y filtrar la creciente cantidad de literatura científica generada durante la pandemia, facilitando a los investigadores el acceso a la información más pertinente a sus objetivos de investigación.

El papel fundamental de los algoritmos de Procesamiento de Lenguaje Natural en la construcción del buscador semántico. Estos algoritmos han permitido identificar conceptos y sinónimos, para su respectiva búsqueda y recuperación.

La búsqueda de información en grandes volúmenes de datos ha emergido como uno de los desafíos en el desarrollo del buscador semántico. Este proyecto ha resaltado la necesidad de gestionar eficazmente estos conjuntos de datos para extraer información pertinente, de manera eficiente teniendo en consideración los recursos existentes.

Por último, la implementación y el despliegue de este proyecto, abre la puerta a futuras investigaciones y mejoras en el buscador semántico, con la esperanza de continuar mejorando la calidad de los resultados y expandiendo su funcionalidad. Quizás expandiendo el buscador a otras áreas de conocimiento y generando modelos de lenguaje natural propio qué permita realizar mejores búsquedas en el futuro.

Referencias Bibliográficas

Baig, N. I., & Bhatia, G. (2013). *WSD Tool for Ontology-based Text Document Classification. International Journal of Applied Information Systems.* https://research.ijais.org/icwac/number3/icwac1328.pdf

Berners-Lee, T. (2000, 12 06). *Semantic Web - XML2000 - Slide list. W3C*. Retrieved August 9, 2022, from https://www.w3.org/2000/Talks/1206-xml2k-tbl/

Contreras, J. (n.d.). *TUTORIAL ONTOLOGÍAS*. Retrieved 8 22, 2022, from https://core.ac.uk/download/11883351.pdf

Fernandez, N. (2010) - Wiktionary. Retrieved Julio 22, 2022, from http://www.it.uc3m.es/labgimi/teoria/Module2/SAIntro.pdf

Garcia, O. C., Gomez-Perez, A., Gómez-Pérez, A., Corcho, O., & Fernandez-Lopez, M. (2004). *Ontological Engineering: With Examples from the Areas of Knowledge Management, E-Commerce, and the Semantic Web*. First Edition. Springer.

Gruber, T. R. (1995). *Toward principles for the design of ontologies used for knowledge sharing? International Journal of Human-Computer Studies*, Volume 43(Issues 5–6), 907-928. https://doi.org/10.1006/ijhc.1995.1081.

Lin, Y. (2008). *Semantic Annotation for Process Models: Facilitating Process Knowledge Management via Semantic Interoperability.*

Lucas, E. S., & Rubio, R. M. (2001). *LAS ONTOLOGÍAS Y SU APLICACIÓN EN EL ÁMBITO DE LA DOCUMENTACIÓN*. UPV. Retrieved September 08, 2022, from http://personales.upv.es/ccarrasc/doc/20002001/Ontolog%C3%ADas/Index.htm

Nisbet, R., Miner, G., & Yale, K. (2017*). Handbook of Statistical Analysis and Data Mining Applications.* Elsevier Science. https://doi.org/10.1016/B978-0-12- 416632-5.00009-8.

OWL Web Ontology Language Guide. (2004, February 10). W3C. Retrieved September 15, 2022, from https://www.w3.org/TR/owl-guide/

Vesin, B., Budimac, Z., Jain, L. C., Ivanović, M., & Klašnja-Milićević, A. (2016). *ELearning Systems: Intelligent Techniques for Personalization*. Springer International Publishing.

Banker, K. (2011). *MongoDB in Action*. Manning Publications Co.

Crockford, D. (2006*). The application/json Media Type for JavaScript Object Notation (JSON).* Network Working Group. Retrieved from https://www.ietf.org/rfc/rfc4627.txt

Fielding, R. T., & Taylor, R. N. (2002). *Principled design of modern Web architecture.* ACM Transactions on Internet Technology, 2(2), 115-150.

Ronacher, A. (2010). *Flask (A Python Microframework).* Flask Documentation. Retrieved from https://flask.palletsprojects.com/en/1.1.x/

Stonebraker, M., & Rowe, L. (1986). *The design of POSTGRES.* In Proceedings of the 1986 ACM SIGMOD international conference on Management of data (pp. 340-355). ACM.

Van Rossum, G., & Drake, F. L. (2009*). Python 3 Reference Manual*. CreateSpace.

Witten, I. H., Moffat, A., & Bell, T. C. (1999*). Managing Gigabytes: Compressing and Indexing Documents and Images.* Morgan Kaufmann Publishers.

Abad-Navarro, F., Bernabé-Diaz, J. A., García-Castro, A., & Fernandez-Breis, J. T. (2021). *Semantic Publication of Agricultural Scientific Literature Using Property Graphs.* Departamento de Informática y Sistemas, Universidad de Murcia, IMIB-Arrixaca, CP 30100 Murcia, Spain; francisco.abad@um.es (F.A.-N.); joseantonio.bernabe1@um.es (J.A.B.-D.). BASF SE, G-FDR/BI-G200, Carl-Bosch-Strasse 38, 67056 Ludwigshafen am Rhein, Germany; alexgarciac@gmail.com. Correspondence: jfernand@um.es; Tel.: +34-868884613.

Allen Institute for AI. (2022). *COVID-19 Open Research Dataset (CORD-19).* Kaggle. Retrieved from https://www.kaggle.com/datasets/allen-institute-for-ai/COR