UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA

CARRERA DE INGENIERÍA DE SISTEMAS

Trabajo de titulación previo
a la obtención del título
de Ingeniero de Sistemas

PROYECTO TÉCNICO:
SISTEMA INTELIGENTE BASADO EN ONTOLOGÍAS, PROCESAMIENTO DEL LENGUAJE NATURAL Y TÉCNICAS DE MINERÍA DE DATOS PARA RECOMENDAR CONTENIDOS CIENTÍFICO-METODOLÓGICOS DEL ÁMBITO DE TERAPIA DEL LENGUAJE

AUTORES:

ADRIAN RICARDO MORALES JIMÉNEZ
LUIS ADRIAN TOBAR ALMACHE

TUTOR:

ING. VLADIMIR ESPARTACO ROBLES BYKBAEV

CUENCA - ECUADOR

2020

CESIÓN DE DERECHOS DE AUTOR

Nosotros, Adrian Ricardo Morales Jiménez con documento de identificación Nº 0930007539 y Luis Adrian Tobar Almache, con documento de identificación Nº 0105819833, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana, la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación: **SISTEMA INTELIGENTE BASADO EN ONTOLOGÍAS, PROCESAMIENTO DEL LENGUAJE NATURAL Y TÉCNICAS DE MINERÍA DE DATOS PARA RECOMENDAR CONTENIDOS CIENTÍFICO-METODOLÓGICOS DEL ÁMBITO DE TERAPIA DEL LENGUAJE,** mismo que ha sido desarrollado para optar por el título de: *Ingeniero de Sistemas*, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación de lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores, nos reservamos los derechos morales de la obra ante citada. En concordancia, suscribo este documento en el momento en el hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, febrero del 2020

Adrian Ricardo Morales Jiménez

C.I. 0930007539

Luis Adrian Tobar Almache

C.I. 0105819833

CERTIFICACIÓN

Yo, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **SISTEMA INTELIGENTE BASADO EN ONTOLOGÍAS, PROCESAMIENTO DEL LENGUAJE NATURAL Y TÉCNICAS DE MINERÍA DE DATOS PARA RECOMENDAR CONTENIDOS CIENTÍFICO-METODOLÓGICOS DEL ÁMBITO DE TERAPIA DEL LENGUAJE,** realizado por Adrian Ricardo Morales Jiménez y Luis Adrian Tobar Almache**,** obteniendo el *Proyecto Técnico*, que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana.

Cuenca, febrero del 2020

Ing. Vladimir Robles Bykbaev, PhD.

C.I. 0300991817

DECLARATORIA DE RESPONSABILIDAD

Nosotros, Adrian Ricardo Morales Jiménez con documento de identificación Nº 0930007539 y Luis Adrian Tobar Almache, con documento de identificación Nº 0105819833, autores de trabajo de titulación: **SISTEMA INTELIGENTE BASADO EN ONTOLOGÍAS, PROCESAMIENTO DEL LENGUAJE NATURAL Y TÉCNICAS DE MINERÍA DE DATOS PARA RECOMENDAR CONTENIDOS CIENTÍFICO-METODOLÓGICOS DEL ÁMBITO DE TERAPIA DEL LENGUAJE,** certificamos que el total contenido del *Proyecto Técnico* es de nuestra exclusiva responsabilidad y autoría.

Cuenca, febrero del 2020

Adrian Ricardo Morales Jiménez

C.I. 0930007539

Luis Adrian Tobar Almache

C.I. 0105819833

AGRADECIMIENTOS

Agradezco a la Universidad Politécnica Salesiana por permitirme realizar mis estudios de instrucción superior, profesores y compañeros que hicieron posible este proyecto de tesis en donde impartieron sus conocimientos desinteresadamente.

A mis padres por brindarme su apoyo día a día en mi trayectoria universitaria, quienes fueron mis pilares fundamentales para continuar con mis estudios.

Quiero expresar mis agradecimientos a Dios con todas las bendiciones y fuerza de voluntad para llegar a la meta deseada.

Adrian Ricardo Morales Jimenez

Luis Adrian Tobar Almache

DEDICATORIA

Dedico este proyecto de tesis a mis padres que me enseñaron que con esfuerzo, sacrificio y constancia se puede lograr un objetivo de vida planteado, también quiero dedicar esta tesis a mis abuelos que me han apoyado de un principio en mi carrera profesional, y cada una de las personas que estuvieron presentes en las diferentes adversidades que tuve como estudiante.

Adrian Ricardo Morales Jimenez.

Luis Adrian Tobar Almache.

RESUMEN

Día a día los investigadores de todo el mundo publican útiles descubrimientos en las diferentes bases de datos científicas que se encuentran disponibles al público. Estos descubrimientos pueden incluir una revisión de una investigación ya desarrollada, un estudio comparativo entre diferentes planteamientos, un nuevo planteamiento acerca de algo ya investigado etc., estar al tanto de estos avances científicos puede ser de mucha utilidad para estudiantes y profesionales cuando se realizan sus investigaciones o dentro del campo laboral.

Debido al gran volumen de información que se genera a diario no es lo suficientemente aprovechado, ya que en varias ocasiones es difícil encontrar información puntual y específica acerca del tema que se quiere investigar.

Además, para realizar un trabajo investigativo o encontrar información acerca de algún tema en específico se usa el buscador proporcionado por los sitios web que albergan los artículos científicos. Estos buscadores no siempre proporcionan un resultado adecuado o esperado, y no se cuenta con otra herramienta que ayude al investigador a filtrar correctamente el contenido.

En consecuencia, se desarrolló una herramienta que aplica varias técnicas y tecnologías inteligentes. Por ejemplo, el uso de ontologías, minería de datos y el procesamiento del lenguaje natural (NLP) para generar una recomendación de contenido científico que sea útil al investigador.

Este sistema recomendador está enfocado en un área en específica la terapia de lenguaje, debido a que es un tema poco investigado en el Ecuador pero muy importante para el desarrollo de los niños que presentan problemas relacionados con el habla y lenguaje.

Los resultados que se han obtenido tras realizar este proyecto demuestran que esta herramienta proporciona un mejor resultado comparado con el obtenido en el buscador proporcionado en los sitios web que albergan los artículos científicos. Estos resultados muestran que el uso de las

tecnologías aplicadas mejora y agiliza la búsqueda de contenido científico y puede ser aplicado en otras áreas del conocimiento.

El sistema recomendador tiene una mejora de 35% aproximadamente comparado con el buscador de Elsevier, y de una mejora de 1% aproximadamente contra el buscador de Science Direct.

Palabras clave: Ontología, NLP, Sistema recomendador, Minería de datos, Modelado del conocimiento.

ABSTRACT

Every day researchers from around the world publish useful discoveries in the different scientific databases that are available to the public. These discoveries may include a review of an investigation already developed, a comparative study between different approaches, a new approach about something already investigated etc., being aware of these scientific advances can be very useful for students and professionals when their investigations or within the labor field.

Due to the large volume of information that is generated on a daily basis, it is not sufficiently exploited, since on several occasions it is difficult to find timely and specific information about the topic that is being investigated.

In addition, to perform a research work or find information about a specific topic, use the search engine provided by the websites that host the scientific articles. These search engines do not always provide an adequate or expected result, and there is no other tool that helps the researcher to correctly filter the content.

Consequently, a tool was developed that applies various intelligent techniques and technologies. For example, the use of ontologies, data mining and natural language processing (NLP) to generate a scientific content recommendation that is useful to the researcher.

This recommender system is focused on an area in specific language therapy, because it is a little researched topic in Ecuador but very important for the development of children who have problems related to speech and language.

The results that have been obtained after carrying out this project demonstrate that this tool provides a better result compared to that obtained in the search engine provided on the websites that house the scientific articles. These results show that the use of applied technologies improves and speeds up the search for scientific content and can be applied in other areas of knowledge.

The recommender system has an improvement of approximately 35% compared to the Elsevier search engine, and an approximately 1% improvement against the Science Direct search engine.

ÍNDICE

ÍNDICE DE ILUSTRACIONES

LISTA DE ECUACIONES

INTRODUCCIÓN

Para las personas que sufren de trastornos de la comunicación y habla, existen diferentes tecnologías para la inclusión social, lo que fue el motivo para realizar este proyecto en donde se detallará los objetivos específicos a cumplirse y la arquitectura para su implementación.

A lo largo de este proyecto de investigación, se emplea una serie de técnicas de procesamiento de lenguaje natural y minería de datos que ayudan a optimizar la búsqueda de un tema en específico relacionado en el ámbito de terapia de lenguaje.

En el tema de discapacidad es importante demarcar cifras como indica el Banco Mundial alrededor de “1000 millones de habitantes, equivalente al 15 % de la población mundial” [1] tiene algún tipo de discapacidad, es por ello que un proceso de enseñanza exige una búsqueda de información exhaustiva para encontrar información relacionada con la metodología de educación planteada, dirigida a un grupo de personas en específico.

Puesto que uno de los problemas que mayormente representan un obstáculo es la posibilidad de brindar una educación que resulte inclusiva y de buena calidad para todos, es debido a esta necesidad que se realiza este sistema para fortalecer y complementar los procesos de enseñanza educativos haciendo uso de la tecnología como medio de educación, con el fin de transmitir contenido científico metodológico en el ámbito de la terapia de lenguaje.

Para complementar la metodología de aprendizaje se utilizan herramientas gratuitas que permiten diseñar, gestionar y compartir una serie de actividades que generan un aprendizaje multimedia para personas que

tienen o no discapacidad entre las cuales están: LAMS[1], Malted[2], Squeak[3], Xerte[4] [30].

Los tipos de discapacidades más comunes en la actualidad son: visual, auditiva, cognitiva o intelectual y trastornos de la comunicación y la del habla es por esta razón que el trabajo que realizan los programadores junto con los expertos en el tema de terapia de lenguaje que desarrollan herramientas accesibles en la web que pueda ser utilizado por cualquier tipo de persona.

Considerando las tecnologías de información y comunicación (TIC's) que proveen diversos recursos como librerías y programas que facilitan el desarrollo y procesamiento de algoritmos aplicados en el sistema que se alimenta de documentos en el dominio de terapia de lenguaje para recomendar contenido del mismo.

En donde el sistema consta de 4 fases descritas de la siguiente manera:

La primera es el diseño y creación de una ontología con el tema de terapia de lenguaje, la cual será obtenida usando SPARQL, esto permitirá hacer consultas a la base de datos de tripletas en donde se utilizará todas las propiedades de una clase determinada por su dominio y rango. Sin embargo, la ontología puede ser extendida o modificada para cubrir cualquier otro tipo de concepto.

La segunda fase es el desarrollo del módulo de extracción de conocimiento; aquí se realizará el análisis de los metadatos y en base a ese análisis se realizará una minería de la información. Con esto podremos obtener solo lo necesario y requerido para la realización del siguiente módulo.

La tercera fase consiste en desarrollar un módulo de procesamiento del lenguaje natural con el uso de los modelos LDA y LSA para así analizar su

[1] LAMS: Learning Activity Management System.
[2] Malted: Herramienta para creación y ejecución de unidades didácticas multimedia.
[3] Squeak: Desarrollar contenido multimedia sin tener conocimientos de programación
[4] Xerte: Herramienta de código abierto para el aprendizaje.

contenido y poder hacer una recomendación adecuada según la búsqueda realizada.

La cuarta fase es el diseño y ejecución de un plan de experimentación automatizado, el cual servirá para probar y validar nuestro sistema.

CAPÍTULO 1.- PROBLEMAS Y OBJETIVOS

1. Descripción del problema

La formación educativa inicial, la escuela tiene la designación de favorecer el proceso de formación de los niños partiendo de exigencias culturales y académicas de acuerdo a su capacidad. Sin embargo, no todos aprenden de la misma manera por distintas razones o discapacidades, por lo que merecen una atención y proceso de enseñanza distinto al resto. Uno de los problemas que se registran comúnmente son los relacionados con el desarrollo del lenguaje y en varias ocasiones se presentan asociados a diferentes tipos de trastornos del mismo, motivo por el cual se pone en discusión acerca de cuál será el procedimiento a seguir para tratar a las personas que presentan diferentes trastornos en la comunicación y habla, aquí se detallan las opciones tecnológicas y metodológicas, implementadas en un sistema que usaran los usuarios que pretende optimizar la búsqueda sobre temas científicos-metodológicos en la terapia de lenguaje, que pueden contribuir a formar una perspectiva diferente y mejorada sobre procesos de aprendizaje y enseñanza en la práctica.

1.1. Antecedentes

El avance de la tecnología ha permitido un mejor entendimiento a problemas y dificultades que las personas afrontan día a día en diversas situaciones, es por eso que se ha dado un creciente desarrollo de nuevas Tecnologías de Información (TI) que pretenden solucionar, prevenir o disminuir aquellos problemas. Este constante e imparable desarrollo de TI se ve reflejado en las mayores bases de datos, como Scopus e IEEE Xplore, que almacenan gran cantidad de artículos científicos con nuevas y útiles propuestas de mejora para diferentes temáticas [2].

Debido a la extensa cantidad de valiosos artículos científicos publicados resulta difícil encontrar información específica para un área de interés en concreto. Por otro lado, el análisis del contenido de estos artículos en el área de terapia de lenguaje permitirá una mejor y más personalizada

búsqueda de información que incluye herramientas, técnicas y tecnologías aplicadas; facilitando el proceso y enriqueciendo el conocimiento del investigador.

El sistema a desarrollar será abierto al público y contará con un amplio estudio de artículos científicos, lo cual permitirá hacer una adecuada recomendación de los mismos. Así se pretende que los terapistas de lenguaje y personas interesadas en el tema, estén al tanto de los últimos avances tecnológicos que ayuden en el trato de personas que necesiten la terapia de lenguaje.

A lo largo del tiempo diferentes autores han realizado exhaustivas investigaciones con respecto a los trastornos específicos del lenguaje en niños de edades comprendidas entre 5 y 9 años de edad, mediante pruebas formales e informales de lenguaje como por ejemplo pruebas de vocabulario. Según los resultados basados en esas investigaciones dedujeron que ningún elemento lingüístico es capaz de identificar a los niños con diferentes tipos de trastornos del lenguaje, por lo cual proponen realizar pruebas de carácter específico en terapia de lenguaje que permitan identificar a los niños con algún tipo de trastorno.

También existen características del trastorno de lenguaje específico, que representan un retraso receptivo o expresivo alrededor de 6 meses, dificultades de tiempo de procesamiento y memoria que afectan su desempeño intelectual en sus actividades cotidianas, esta muestra no integra a niños que tengan algún tipo de lesión como son: pérdida auditiva, neurología focal, problemas motores o sensoriales, autismo, déficit de atención, etc.

Este trabajo me permite obtener información estadísticamente sobre los resultados que se obtienen en un grupo de niños con un trastorno de lenguaje específico, en donde indican las pautas para realizar procesos de enseñanza donde se puede mejorar las herramientas de soporte en la terapia de lenguaje.

De acuerdo a un estudio que se realizó en 2017, donde recolectaron información sobre pruebas, resultados en el habla y lenguaje de diferentes

centros de educación especial en la ciudad de Cuenca con niños en el rango de 3 y 7 años de edad, en base a los requerimientos que obtuvieron desarrollaron una herramienta de soporte que ayuda a los expertos al momento de generar un proceso de aprendizaje que ayude a disminuir los diferentes trastornos de la comunicación [2].

Una propuesta desarrollada por el instituto para Niños Ciegos y Sordos del Valle del Cauca, se basó en un prototipo de videojuego el cual se centra en terapia auditivo-verbal, este cuenta con un sistema de interacción que realiza un reconocimiento de voz en conjunto con la retroalimentación de los resultados que obtiene de cada uno de los pacientes, lo que se pretende es que este videojuego funcione sin la presencia de un terapista competente [3].

1.2. Importancia y alcances

El alcance de este proyecto es analizar la efectividad de la recomendación de artículos científicos, que pueden ayudar a la investigación en las distintas intervenciones en el ámbito de terapia de lenguaje, basándose en un sistema recomendador que almacena información relevante sobre dicho problema, que indicará los resultados del meta-análisis propuesto, las técnicas y metodologías necesarias para la efectividad de las intervenciones en la terapia de lenguaje.

Una vez que se obtengan los resultados y evaluaciones con relevancia, los expertos en el tema podrán categorizar que efectos se les dará a las intervenciones que recibirán los pacientes que sufren de trastornos de la comunicación, mediante la recomendación que arroja el sistema de acuerdo a una ponderación prioritaria basada en las palabras claves de búsqueda ingresadas, para causar un efecto positivo en el mismo y reducir las dificultades que se presentan al momento de la terapia de lenguaje.

1.3. Delimitación

Esta herramienta va dirigida principalmente a las personas que trabajan en el área de terapia de lenguaje y también a cualquiera que desee investigar

y mejorar sus conocimientos al obtener información científica-metodológica útil para el tema de terapia de lenguaje.

Al proveer este sistema a los terapistas de lenguaje se pretende mejorar su conocimiento proporcionando una herramienta eficiente que les permita encontrar nueva y valiosa información.

1.4. Objetivo general y específicos

1.4.1. Objetivo general

Mejorar los procesos de investigación de la terapia del lenguaje mediante el desarrollo de una herramienta que analice contenidos científicos empleando ontologías, procesamiento del lenguaje natural y minería de datos.

1.4.2. Objetivos específicos

- Estudiar y conocer los principales fundamentos teóricos de la terapia de lenguaje, los trastornos de la comunicación y el lenguaje con mayor prevalencia y las técnicas inteligentes para procesar textos científicos y ontologías.
- Diseñar y desarrollar una ontología que permita generar un repositorio de metadatos de artículos académicos de una base de datos científica.
- Diseñar y desarrollar un módulo de extracción de conocimiento que permita obtener artículos académicos a partir de la información contenida en el repositorio de metadatos.
- Diseñar y desarrollar un módulo de procesamiento del lenguaje natural que permita analizar los artículos académicos y generar un ranking de recomendaciones y extracción de contenidos científicos.
- Diseñar y ejecutar un plan de experimentación automatizado con al menos 500 artículos académicos.

CAPÍTULO 2.- FUNDAMENTACIÓN TEÓRICA

2. Conceptos y Fundamentos tecnológicos

2.1. Terapia de lenguaje

Es la especialidad dentro del campo de la rehabilitación que se encarga de la evaluación, diagnóstico y tratamiento de las alteraciones en voz, audición, habla, lenguaje, aprendizaje y los aspectos de la motricidad oral que afectan durante el desarrollo del niño.

Es importante diferenciar que el lenguaje es definido como un sistema de símbolos aprendidos que tienen un significado social y dan la habilidad a un individuo de clasificar experiencias, mientras el habla es la producción y percepción de los símbolos orales [4].

La terapia de lenguaje y habla también es el método para la mayoría de los niños con discapacidades del habla y estudio del lenguaje. Las discapacidades en el habla se refieren a problemas con la elaboración de sonidos, mientras que los problemas con el aprendizaje del lenguaje son las dificultades al juntar las palabras para comunicar ideas [4].

2.1.1. Trastornos del lenguaje en niños

Los trastornos del lenguaje pueden afectar el habla, la escritura, el ritmo, la compresión y en muchas ocasiones varios de ellos combinados [4]. Algunos de los trastornos para los cuales el sistema dará una recomendación de contenido científico metodológico para llevar a cabo la terapia de lenguaje, se describen a continuación.

2.1.1.1. Disartria

Trastorno neuromuscular que afecta a la articulación de la palabra. Los músculos de la boca, la cara y el sistema respiratorio se pueden debilitar, moverse con lentitud o no moverse en lo absoluto después de un derrame cerebral u otra lesión cerebral [4].

2.1.1.2. Dislalia

En general, es transitoria y consiste en la dificultad para pronunciar diferentes fonemas o grupos de fonemas, bien por ausencia o alteración de algunos sonidos concretos [4].

2.1.1.3. Tartamudez

Es la deficiencia o problema de fluidez de la palabra, se acompañan de tensión muscular en cara y cuello, miedo y estrés, ocasionado por factores orgánicos, psicológicos o sociales [4].

2.1.1.4. Dislexia

La dislexia es una condición cerebral que dificulta la lectura, la escritura y algunas veces, el habla esto puede incluir hacer coincidir el sonido de una letra con su símbolo. Los niños disléxicos suelen presentar un retraso de lenguaje que afecta a los procesos fonológicos, semánticos y sintácticos [4].

2.1.1.5. Disfasia

Es el trastorno que obstruye la capacidad del niño para desarrollar las habilidades del lenguaje: errores graves de gramática, vocabulario casi nulo, dificultades fonológicas, etc. [5].

2.1.1.6. Afasia

Es un trastorno causado por lesiones en las zonas del cerebro que controla el lenguaje y que puede afectar la lectura, la escritura y la expresión [5].

La meta de la terapia de lenguaje es que un individuo pueda desarrollar o recuperar habilidades de comunicación efectivas y así mejorar la calidad de vida de las personas que padecen los trastornos del lenguaje y habla.

En este contexto, el avance de la tecnología ha permitido un mejor entendimiento a los problemas que las personas afrontan día a día a diversas

situaciones, las soluciones propuestas para brindar apoyo a la rehabilitación de personas con discapacidades en el ámbito de la terapia de lenguaje son limitadas en relación con el campo de aplicación [6].

De esta manera se mencionan los aspectos más críticos para la construcción y desarrollo de cualquier sistema de soporte a la terapia de lenguaje, puesto que si no se toman en cuenta no podrán realizar las actividades que llevan a cabo los especialistas en el área, cabe destacar que, sin las tecnologías necesarias para las herramientas de soporte, la terapia de lenguaje puede tender a ser ineficientes para su uso o no cumplir con las expectativas de los pacientes.

2.2. Sistema Recomendador

Estos son sistemas de software y bases de datos que permiten a los usuarios reducir el número de alternativas mediante el ordenamiento, conteo o algún otro método de un tema en específico. También se les conoce como herramientas informáticas que poseen un conjunto de técnicas que ayudan a evaluar las sugerencias más relevantes en base a una búsqueda.

Los sistemas recomendadores se pueden clasificar de la siguiente manera [7]:

- Filtrado basado en contenido: Estos realizan recomendaciones únicamente de las preferencias de búsqueda del usuario y los atributos de los ítems a recomendar.

- Filtrado colaborativo: Este usa información de un conjunto de usuarios y su relación con el contenido de búsqueda para realizar una recomendación, identifican usuarios que tengan preferencias similares a un mismo tema de búsqueda de tal manera que recomiende los elementos que hayan satisfecho a otros usuarios.

- Sistemas de recomendación híbridos: combina las técnicas de filtrado colaborativo y filtrado basado en contenido.

Para el desarrollo de este sistema se usó la recomendación basada en contenido donde se emplean técnicas de recuperación de información que utilizan las palabras claves ingresadas por el usuario, de este modo se realizó un algoritmo para asignar una ponderación a las sugerencias más importantes de documentos de contenido científico metodológico en el ámbito de terapia de lenguaje.

2.2.1. Estructura de un sistema recomendador

Los elementos fundamentales que intervienen en el funcionamiento de un sistema recomendador que podemos usar como criterio de clasificación son los siguientes:

- Entradas / salidas del proceso de generación de la recomendación, donde se usan las entradas del usuario activo como parámetros para realizar la búsqueda de esta manera la realimentación por parte del usuario mejoraría los procesos de generación de recomendaciones, y las salidas del sistema están compuestas por recomendaciones generadas de acuerdo a la cantidad y formato de la información proporcionada por el usuario, dentro de las más comunes se tienen las siguientes:
    - Sugerencia o lista de sugerencias al usuario de una serie de ítems.
    - Visualizar al usuario alternativas del grado de satisfacción de un tema en concreto, estas pueden ser personalizadas de acuerdo al tipo de recomendación utilizada para la búsqueda.
- Los métodos usados habitualmente para generar recomendaciones de acuerdo a las necesidades de un tema de interés, se debe tomar en cuenta que no son mutuamente exclusivos entre sí, sino complementarios, es decir, que un mismo sistema recomendador puede usar uno o varios métodos. Se describen a continuación los 3 métodos más simples [7]:

- Medir la similitud de todos los usuarios con respecto al usuario activo.

- Seleccionar un subconjunto de usuarios cuyas valoraciones se van a usar y, por tanto, tendrán influencia en la generación de la predicción para el usuario activo.

- Normalizar las puntuaciones de los distintos usuarios y calcular una predicción a partir de algún tipo de combinación basada de las puntuaciones asignadas al ítem por los usuarios seleccionados al paso anterior.

- Grado de personalización de un sistema recomendador:

  - Los sistemas recomendadores que proporcionan las mismas sugerencias a todos los usuarios, en este ámbito se clasifican como no personalizados. Estas recomendaciones serán basadas en selecciones manejables, datos estadísticos u otras técnicas similares.

  - Los sistemas recomendadores tienen en cuentan la información del usuario que proporciona una personalización pasajera, ya que las recomendaciones tienen una rección al comportamiento y búsqueda del usuario en su sesión actual de navegación.

  - Los sistemas recomendadores que ofrecen una mayor personalización usan información recomendada por distintos usuarios, incluso cuando realizan la misma búsqueda. Estos sistemas están basados en perfil del usuario, por lo que usan métodos de filtrado colaborativo, basado en contenidos.

2.2.2. Métricas de los sistemas recomendadores

Para medir un sistema de recomendación existen 3 pasos fundamentales: Identificar los objetivos del sistema, identificar las tareas o métodos que permiten alcanzar esos objetivos e identificar el nivel de métricas del sistema. La evaluación a nivel de sistema se realizará en caso que los

usuarios identifiquen búsquedas que se puedan medir y evidenciar con la efectividad de la recomendación que arroja el sistema independientemente de la interacción de los usuarios [7].

Según Cleverdon se identifican las siguientes métricas para evaluar el rendimiento del sistema [10]:

- Retardo: intervalo de tiempo transcurrido desde que se hace la búsqueda hasta que se da la recomendación.
- Presentación: el formato físico de los resultados del sistema.
- Esfuerzo del usuario: el esfuerzo intelectual del usuario al realizar la búsqueda.
- Exhaustividad: capacidad del sistema para recomendar todos los documentos relevantes de carácter científico metodológicos al usuario.
- Precisión: capacidad del sistema para mostrar únicamente los documentos que sean relevantes a la búsqueda.

2.3. Base de datos científica del sistema recomendador

Las bases de datos tienen sus inicios en publicaciones a principio del siglo XX, conocidos como resúmenes. Con la llegada de la informática en los años 70 del siglo pasado, estas publicaciones se automatizan facilitando su consulta y distribución.

Las bases de datos científicas son recopilaciones de publicaciones e investigaciones de contenido científico-técnico como por ejemplo libros, tesis, revistas, etc. de un cierto grado de contenido temático que tiene como objetivo principal almacenar toda una producción bibliográfica sobre un área de conocimiento.

Las bases de datos contienen registros y campos y están estructuradas de la siguiente manera:

- Registros: cada uno de estos representa un único documento que tiene una referencia a un artículo de libro, revista, tesis, etc.

- Campos: cada representa un tipo de información distinta sobre un documento o registro como, por ejemplo, palabras claves, titulo, introducción.

- Interfaz de búsqueda: es un sistema en el que se realizan búsquedas en base a un tema de interés seleccionando la base de datos científica que la interfaz permite escoger.

2.3.1. Scopus

Es una base de datos de referencias bibliográficas y citas de la empresa Elsevier que abarca temas como ciencia, tecnología, medicina que facilita la posibilidad de buscar información basada en el número de citas correspondientes a un tema en específico, además permite ver el grado de contenido científico dentro de los artículos.

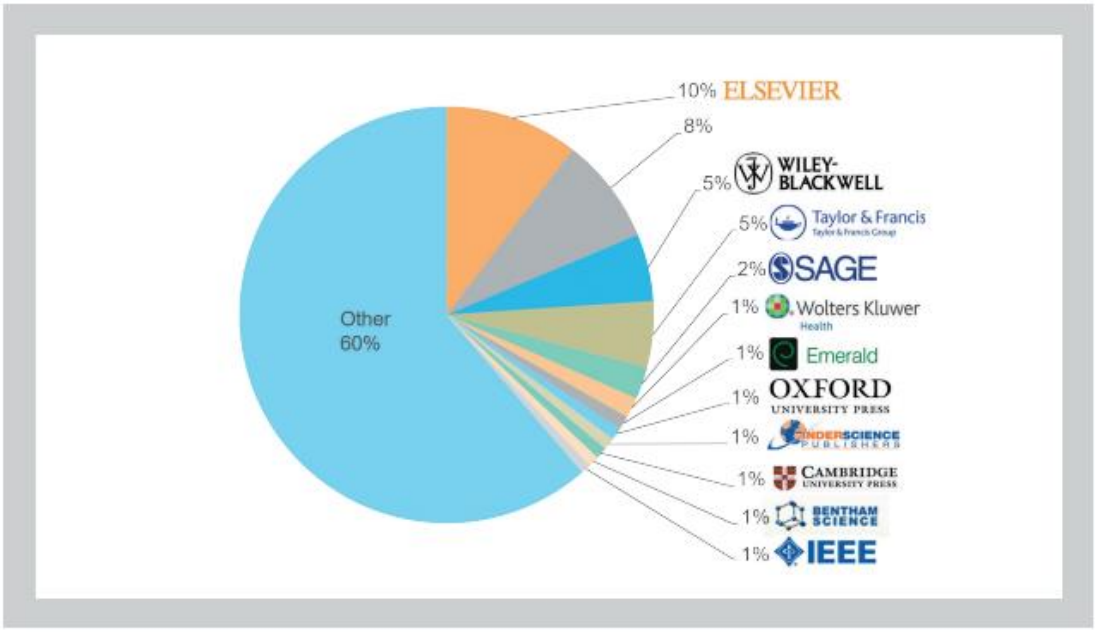


Ilustración 1.- Editores indexados en Scopus [8].

En al año 2017 Scopus contaba con las siguientes cifras [8]:

- Mas de 21950 revistas revisadas, incluidas más de 3600 revistas de acceso abierto completo.
- 280 publicaciones comerciales.
- Más de 560 series de libros.
- Mas de 8 millones de ponencias de más de 100000 eventos mundiales.
- Artículos en prensa de más de 8000 revistas.
- Mas de 150000 libros con 20000 agregados cada año.

Scopus apoya a investigadores y bibliotecarios por medio de 3 áreas claves:

Búsquedas:

- Búsqueda por documentos, autor o por búsqueda avanzada.
- Depurar los resultados por tipo de fuente, año, idioma, autor, etc.
- Enlace artículos de texto completo y otros recursos de la base de datos.
- Usa el documento QUOSA Administrador de descargas para recuperación masiva resultados en formato PDF.
- Actualización de alertas por correo electrónico Fuentes RRS y HTML.

Descubrir:

- Encuentra documentos relacionado por referencias, autores, o palabras clave.
- Identificar y unir una organización con su producción de investigación.

- Beneficios de interoperabilidad con ScienceDirect, Mendeley, SciVal, Reaxys y Engienering Village.

Analizar:

- Rastrear citas a lo largo del tiempo para un conjunto de autores o documentos que contengan citas.
- Evaluar a tendencia en los resultados de búsqueda
- Analizar el resultado y publicaciones por autores

Lo que se logró con Scopus para este proyecto de investigación fue encontrar información relacionada en el ámbito de terapia de lenguaje que contenga: palabras clave, resumen, introducción, cuerpo del documento proporcionado por los autores.

Scopus agrega manualmente términos de índice para el 80 % de los títulos incluidos en el mismo, dichos términos de índice se derivan de tesauros que Elsevier posee o licencia y se añaden para mejorar la recuperación de la búsqueda.

2.3.2. Elsevier Fingerprint Engine

El White paper de Elsevier lo describe como un sistema de software back-end de técnicas de procesamiento de lenguaje natural (NLP) de última generación para extraer información de texto no estructurado [9].

De acuerdo a la indexación Fingerprint se utilizan las herramientas tales como:

Tokenización: Es una de las herramientas de proceso de lenguaje natural que consiste en tomar un texto o un conjunto de textos y dividirlo en palabras individuales, estos tokens se usan como métodos de palabras para otro tipo de análisis como por ejemplo el etiquetado de la relación semántica entre palabras [9].

Normalización: Este proceso consiste en dar un formato uniforme y común para todos los documentos que se van a usar en el sistema, de esta manera se podrá extraer información única de cada documento con el fin de escoger los términos más relevantes del mismo, generando la reducción del texto a indexar lo que facilitara el almacenamiento y búsqueda de información posteriormente [9].

Extracción de frase sustantiva: se basa en la recuperación de frases sustantivas básicas como su fuente principal de identificación de entidad. Ya que una frase sustantiva es una unidad sintáctica de la oración en la que se recopila información sobre el sustantivo [12].

2.4. Ontologías

Es la representación del conocimiento que resulta de la selección de un tema en específico las cuales contienen clases, propiedades y relaciones entre las mismas que tienen un propósito, el cual es capturar el conocimiento sobre algún dominio de interés. Además, el modelo lógico de una ontología permite la verificación si todas las afirmaciones y definiciones que contiene la ontología son o no coherentes entre sí y también puede reconocer cuales son los conceptos que se ajustan a que definiciones [11].

Para el uso de la ontología en el sistema, se usó una API de Elsevier que fue descargada en formato rdf-xml que contiene información semántica, de esta manera se pudo extraer los metadatos necesarios como título, abstract, palabras clave del documento para refinar el mismo en donde las acciones que se ejecutaron se representan en la siguiente imagen.

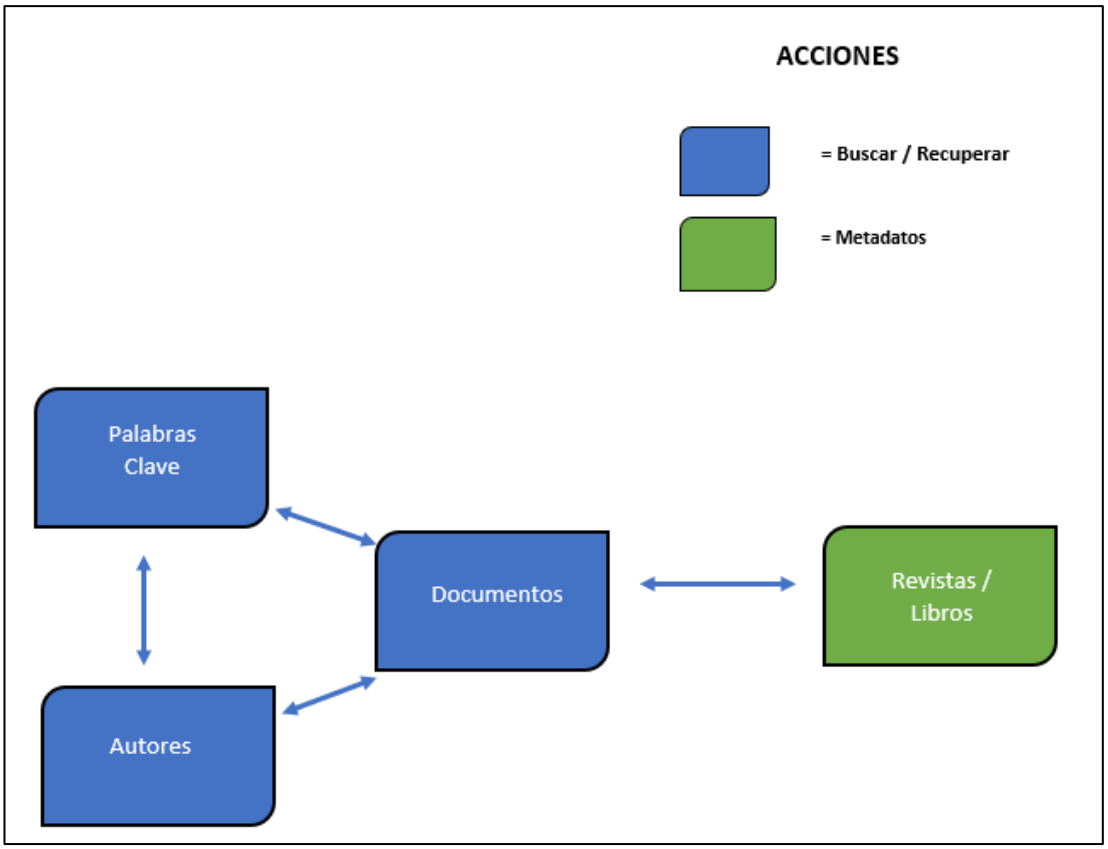


Ilustración 2.-Estructura Api Scopus [13].

Las ontologías también se consideran repositorios de información que están relacionadas a hechos particulares en donde el conocimiento que posee puede ser verdadero, ya que se deriva de alguna forma de razonamiento que permite la clasificación del conocimiento para recuperar la información de manera automatizada [11].

El uso de ontologías en la extracción de conocimiento es muy útil, en especial cuando la ontología va cambiando o es enriquecida en base al conocimiento. Un grupo de investigadores australianos [31] propuso un Sistema de Extracción basado en Ontologías (OBIE) para el dominio de medicina herbal que extiende de otros sistemas de extracción para realizar el enriquecimiento de la ontología. Haciendo uso de palabras clave y patrones, este sistema valida declaraciones contradictorias relacionadas con la instancia dentro de la ontología.

Una publicación hecha por Chenjian Hao [32] muestra que el modelado de conocimiento hecho en forma de texto tiene desventajas a la clasificación

y al intercambio de conocimiento. Esto es resuelto por la tecnología de ontología de conocimiento que es usada en la ingeniería de conocimiento. Es un enfoque ideal para construir un caso de conocimiento basado en ontología para almacenar y representar el conocimiento de dominio.

El proyecto Scientific Knowledge Miner (SKM) [33] desarrollado en España usa un Web Crawler para generar una base de conocimiento con el texto y metadatos de artículos científicos online, prosiguiente hace uso de varias librerías y herramientas Open Source como Dr. Inventor (DRI Framework), Elastic Search y Kibana para realizar la minería de información, el procesamiento de lenguaje natural (NLP), indexación y presentación.

Hua Bolin [34] refiriéndose a la extracción de conocimiento, menciona que esto se puede realizar con el uso de objetos de procesamiento que tienen diferentes formatos, tales como tesauros, ontologías, lexicon o se puede ir más allá y extraer conocimiento de imágenes. En cuanto a los objetos de extracción, menciona la literatura en general y la literatura web; esta contiene información semi estructurada haciendo que el proceso de extracción sea más fácil.

2.4.1. Características representativas de las ontologías

Algunas de las características más representativas de las ontologías se describen [11]:

- Ontologías Múltiples: Se pueden combinar dos o más ontologías con el fin de ser explícito a un punto de vista en donde cada ontología va a tener una representación especifica.

- Varios niveles de abstracción de las ontologías: Niveles de abstracción para obtener una topología de ontologías.

- Multiplicidad de la representación: Un concepto puede ser representado de varias formas.

- Mapeo de ontologías: Establecer relaciones entre los elementos de una o más ontologías generando una esquematización organizada.

2.4.2. Propiedades que deben cumplir las ontologías

Algunas de las propiedades que den cumplir las ontologías son [11]:

- Claridad: Para comunicar el significado de los términos definidos.
- Coherencia: Para sancionar inferencias que son consistentes con las definiciones.
- Extensibilidad: Para anticipar el uso de vocabulario compartido.
- Sesgo de codificación mínimo (Minimal encoding bias): Debe de especificar al nivel de conocimiento sin depender de una codificación particular a nivel de símbolo.
- Mínimo compromiso ontológico: Debe de hacer la menor cantidad de pretensiones acerca del mundo modelado.

2.4.3. Clasificación en cuanto al ámbito del conocimiento

Existen 4 tipos de ontologías que nos ayudaron a precisar el alcance y posibilidad de la aplicación dentro de las más importantes están [11]:

- Ontología de la aplicación: Usadas por la aplicación. Por ejemplo, ontología de procesos de producción, de diagnósticos de fallas, de diseño intermedio de barcos, etc.
- Ontología del dominio: Específicas para un tipo de artefacto, generalizaciones sobre tareas específicas en algún dominio concreto del conocimiento. Por ejemplo, ontología del proceso de producción.
- Ontologías técnicas básicas: Describen características generales de artefactos. Por ejemplo: componentes, procesos y funciones.
- Ontologías genéricas: Describe la categoría de más alto nivel, describiendo conceptos generales como tiempo, espacio, objeto, etc.

Otras posibles clasificaciones de las ontologías son: en función de su punto de vista, por ejemplo: físico, de comportamiento, funcional, estructural, topológico, etc.

Según el grado o nivel de abstracción y razonamiento lógico que permitan, por ejemplo: ontologías descriptivas, que incluyen taxonomías de conceptos, relaciones entre conceptos, pero no permiten inferencias lógicas y ontologías lógicas.

Las que permiten inferencias lógicas mediante la utilización de una serie de componentes como la inclusión de axiomas, etc.

2.4.4. Marco de Descripción de Recursos (RDF)

El Marco de Descripción de Recursos o Resource Description Framework (RDF) es el estándar más potente y fácil de usar para intercambiar datos en la web. Una declaración RDF describe una relación entre dos entidades.

Contiene declaraciones sobre recursos como expresiones de sujeto, predicado y objeto; a estas expresiones se les referencia como una tripleta. RDF es el modelo de datos usado por ontologías de la web semántica y base de datos de conocimiento usado comúnmente para representar metadatos [14].

RDF usa URIs (Uniform Resource Name) para nombrar la conexión entre nodos y los cierres de la conexión.

En la Ilustración 2 se puede observar representada una tripleta con la oración ‘Juan vive en Ecuador’. Donde el sujeto es ‘Juan’, el predicado es ‘vive en’ y el objeto es ‘Ecuador’

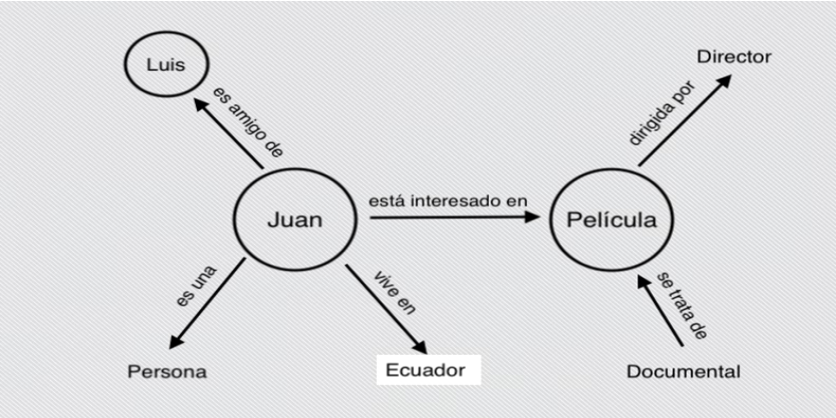


Ilustración 3.-. Ejemplificación de como RDF conecta datos mediante tripletas

2.5. Protégé

Protégé es una plataforma de código abierto que proporciona una creciente comunidad de usuarios con un conjunto de herramientas para la construcción de modelos de dominio y basada en el conocimiento de las aplicaciones de las ontologías [17].

La herramienta viene con una interfaz de usuario que cuenta con una navegación y razonamiento, en donde se pueden explorar clases, propiedades y características de una ontología, el razonamiento sobre la ontología es comúnmente realizado por un built-in llamado HermiT en donde también se puede inferir sobre la jerarquía de clases utilizando el cuadro de lista desplegables respectivamente [17].

Se uso Protégé para la visualización de la ontología, entidades, clases que se usaron en el tratamiento de datos donde se usó el programa de Orange.

2.6. Orange

Orange permite la exploración interactiva de los datos, se reciben los datos de entrada y envía información filtrada o procesado de datos como se muestra en la Ilustración 4.

Herramienta usada en el análisis de datos se realiza por el apilamiento de los componentes en los flujos de trabajo. Cada componente, llamado por

un widget, incorpora algunas funciones de recuperación de datos, procesamiento, visualización, modelado o evaluación de la tarea, la combinación de diferentes widgets en un flujo de trabajo permite construir de manera integral el esquema de análisis de datos y su proceso de desarrollo [16].

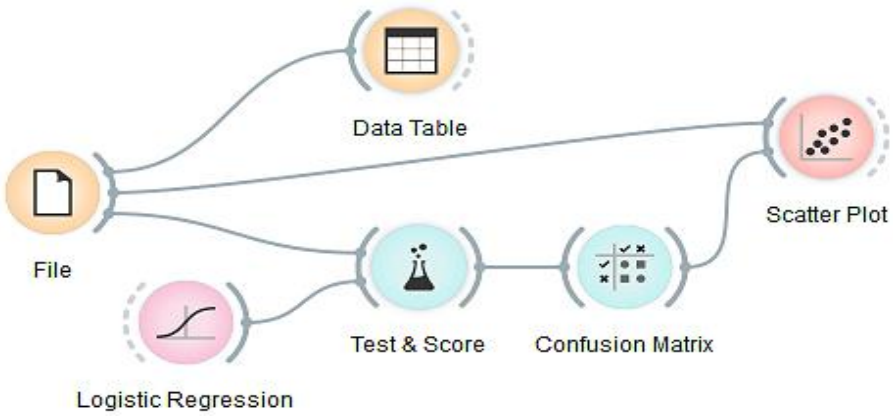


Ilustración 4.- Componentes basados en la minería de datos Orange [14].

Se uso Orange para realizar un tratamiento de datos en donde se juntó un *dataset*[5] que contiene el *IRI*[6] y los pesos de cada término para su integración en Karma.

2.7. Karma

Es una herramienta de integración de información que permite unir de manera rápida y fácil datos que poseen una variedad de fuentes de datos, incluidas bases de datos, hojas de cálculo, archivos de texto delimitados, XML, JSON, KML y API web. Integran la información modelándola de acuerdo con una ontología de su elección utilizando una interfaz gráfica de usuario que automatiza gran parte del proceso [18].

Karma también aprende a reconocer el mapeo de datos a clases de ontologías y luego usa la ontología para proponer un modelo que une estas clases.

[5] *Dataset:* Conjunto de Datos
[6] *IRI:* Identificador de Recursos Internacionales

Durante este proceso se pueden transformar los datos según sea necesario, para normalizar los datos expresados en diferentes formatos y reestructurarlos, una vez que el modelo esta completo, se pueden publicar los datos integrados como RDF o almacenarlos en una base de datos.

2.8. Graph DB

Ontotext GraphDB es una base de datos gráfica altamente eficiente y robusta con soporte RDF y SPARQL. Con características como: [22]

- Administrar un número limitado de declaraciones RDF.
- Soporte completo de SPARQL.
- Implementación usando java.
- Compatible 100% con el marco RDF4J.
- Conjunto de reglas completas y optimizadas para RDFS, OWL, etc.
- Razonamiento personalizado y conjunto de reglas de comprobación de coherencia.
- Complemento de API para la extensión del motor.
- Interfaz de *Workbench* [7] para administrar repositorios, dato, cuentas de usuario y roles de acceso.
- Carga de alto rendimiento, consulta e inferencia simultáneamente.

Esto permitió la creación de gráficos de conocimiento para vincular diversos datos, enriquecerlos mediante análisis de texto e indexarlos en GraphDB para la búsqueda semántica en el ámbito de terapia de lenguaje.

[7] *Workbench:* Interfaz de administración basada en web para GraphDB.

2.8.1. SPARQL

SPARQL (SPARQL Protocol and RDF Query Language) es un estándar propuesto por el World Wide Web Consortium (W3C) en el año 2008 para la lectura de grafos RDF (W3C SPARQL Working Group,2013).

Permite una consulta que consiste en patrones triples, conjunciones, disyunciones y patrones opcionales. Actualmente existen varias implementaciones de SPARQL para múltiples lenguajes de programación [22].

SPARQL nos permite traducir datos en grafo, intensamente enlazados, en datos normalizados en formato tabular, de tal manera que se distribuye en filas y columnas, que se pueden abrir en programas como Excel o importar a programas de visualización [25].

Ejemplo de consulta SPARQL:

```
SELECT ?pintura

WHERE { ?pintura <utiliza la técnica de> <óleo sobre lienzo> .}
```

En esta consulta, `?pintura` representa el nodo (o nodos) que la base de datos retornará. Una vez recibida la consulta de la base de datos buscará todos los valores para `?pintura` que adecuadamente represente la declaración RDF [25].

2.8.2. Índice de Similitud

El módulo de similitud permite explorar y buscar similitudes semánticas en los recursos RDF.

Se realizo una consulta SPARQL lo que permitió hacer una búsqueda de término-documento usando un complemento de similitud en Graph DB que devuelve los documentos más representativos para un término en específico buscado.

Este complemento posee un algoritmo que utiliza un tokenizador para traducir documentos a secuencias de palabras o términos y representarlos en

un modelo de espacio vectorial que representa su significado abstracto en donde se clasifica de 0 hasta 1 siendo 1 la mayor ponderación que se le asigna a la búsqueda en su grado de similaridad para su análisis posterior [22].

2.8.3. URI

Los URIs o Identificadores de Recursos Uniformes (Uniform Resource Identifier) fortalecen la identificación y localización de recursos de información en el contexto de la web semántica [15].

Una URI se define como una cadena de caracteres que posibilita, por un lado, la identificación de recursos de información localizables en la web y, por otro, identifica entidades concretas o abstractas que no se transmiten vía web [15].

En este marco, cada dato se identifica con una URI y es posible establecer relaciones con otros datos, que a su vez estarán identificados con otras URIs. Incluso las relaciones pueden estar identificadas con una URI [19].

2.9. Minería de datos

Es un conjunto de técnicas que permiten explorar grandes bases de datos de manera automática o semiautomática con el objetivo de encontrar patrones repetitivos, tendencias o reglas que expliquen el comportamiento de dichos datos en un contexto.

Un conjunto de clústeres que describe cómo se relacionan los casos de un conjunto de datos. Reglas que describen cómo se agrupan los productos en una transacción, y las probabilidades de que dichos productos se adquieran juntos [21].

Técnicas de minería de datos:

- **Algoritmos de clasificación**, que predicen una o más variables discretas, basándose en los demás atributos del conjunto de datos.
- **Algoritmos de regresión**, que predicen una o más variables numéricas continuas, como pérdidas o ganancias, basándose en otros

atributos del conjunto de datos.

- **Algoritmos de segmentación**, que dividen los datos en grupos, o clústeres, de elementos que tienen propiedades similares.

- **Algoritmos de análisis de secuencias,** este resumen las secuencias frecuentes o episodios en los datos, como una serie de clics en un sitio web o una serie de eventos de registro que preceden al mantenimiento del equipo.

El algoritmo TF-IDF que se implementó para desarrollar el sistema, se encuentra dentro de los algoritmos de segmentación en el ámbito de la minería de datos, ya que la detección de segmentos sería el objetivo principal de la minería de datos a usar en el sistema.

Las técnicas de segmentación de la minería de datos provienen de la inteligencia artificial y de la estadística, dichas técnicas no son más que algoritmos sofisticados que se aplican sobre un conjunto de datos para obtener resultados, por lo tanto, el objetivo principal es clasificar la información que se desea evaluar y analizar, de esta manera se determinan las variables continuas a segmentar o clasificar en grupos, y finalmente determinar que ocurre en una base de datos de gran tamaño.

2.10. TF-IDF

TF-IDF (*Term Frequency - Inverse Document Frequency*) es un método estadístico que refleja cuán importante es una palabra en un documento o corpus. A menudo se usa como un factor de ponderación en la minería de datos [20].

- **Frecuencia de Términos (TF):** mide la frecuencia en la que un término aparece en el documento.

  $TF(t)$ =el número de veces que el término *t* aparece en un documento.

- **Frecuencia Inversa del Documento (IDF):** asigna un peso a una palabra, dado por la siguiente fórmula:

$$\log\left(\frac{N}{NT}\right) \quad (1)$$

Donde *N* es el número de documentos y *NT* es el número que contiene el término *t*.

Entonces:

$$Tf - Idf = tf * \log\left(\frac{N}{NT}\right) \quad (2)$$

2.11. Django

Django es un framework de aplicaciones web gratuito escrito en Python, lo cual lo convierte en un conjunto de componentes que ayudan a desarrollar sitios web de manera más fácil y eficiente en donde se ejecutan una serie de acciones para administrar funciones que nos proporciona este framework entre sus mejoras de modelos y base datos, búsquedas, transacciones, acceso a las optimizaciones de base de datos [23].

2.12. D3

D3 permite vincular datos arbitrarios a un modelo de objetos de documento (DOM) y luego transformarlas basadas en datos al documento. Como, por ejemplo, D3 se puede usar para generar una tabla HTML a partir de una matriz de números [29].

Esta biblioteca de JavaScript combina potentes técnicas de visualización e interacción con un enfoque basado en datos que se necesita representar, brindando la capacidad a los diversos tipos de navegadores la libertad de diseñar la interfaz visual adecuada para sus datos.

2.13. Librerías y herramientas importantes

En el desarrollo del sistema se usaron varias librerías y herramientas que fueron relevantes para el funcionamiento del mismo. En esta sección se listan algunas.

2.13.1. NLTK

NLTK (*Natural Language Toolkit*) es una plataforma que permite desarrollar programas en python que trabajen con datos de lenguaje humano.

Contiene un conjunto de librerías de procesamiento de texto para clasificación, tokenización, tagging, entre otros [26].

2.13.2. TextBlob

Es una librería para python que permite realizar tareas de NLP como etiquetado, extracción de frases nominales, análisis de sentimientos, clasificación, traducción y más [27].

2.13.3. RDFLib

Es una librería Open Source para python que permite trabajar con RDF. Contiene un analizador de texto para diferentes formatos de archivos además de permitir realizar grafos. Implementa soporte para SPARQL [28].

2.13.4. CERMINE

CERMINE (Content Extractor and Miner) es un sistema Open Source para java que permite extraer metadatos estructurados de artículos científicos.

La implementación contiene, en su mayoría de pasos, técnicas de machine learning supervisadas y no supervisadas para simplificar el proceso de adaptación a documentos con estructuras y estilos diferentes [25].

CAPÍTULO 3.- METODOLOGÍA

3. Definición de arquitectura y componentes tecnológicos

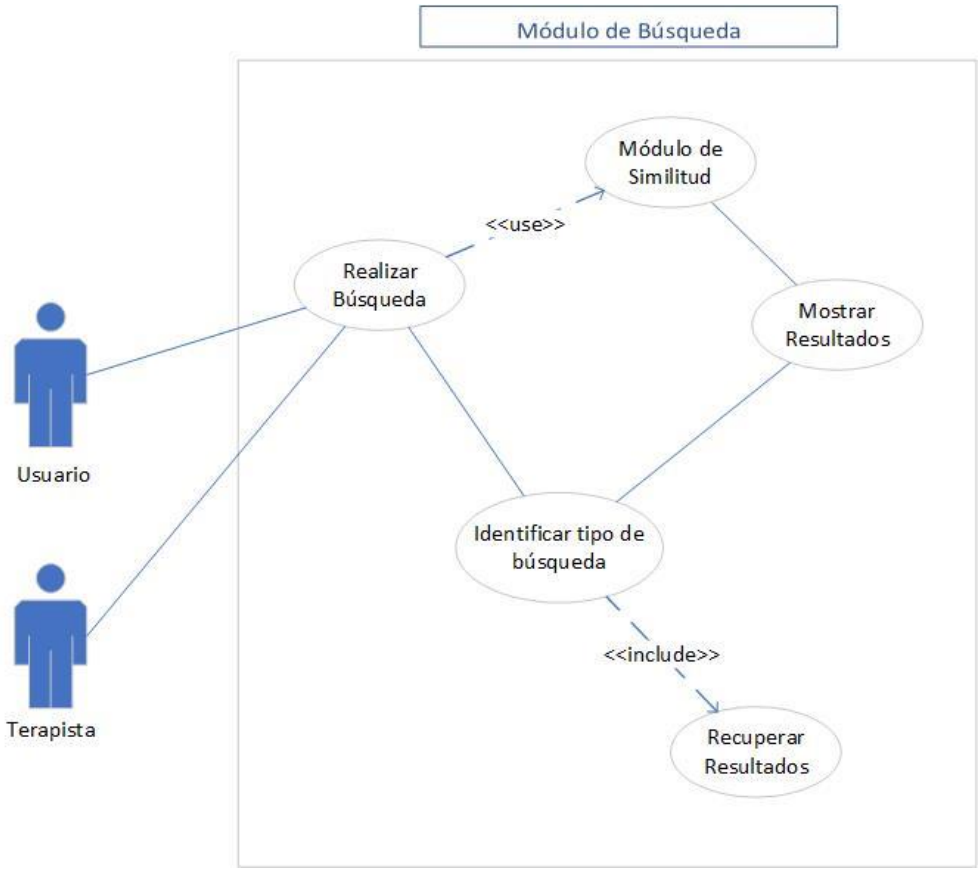
3.1. Requerimientos

La gran ayuda que se obtiene con los artículos científicos publicados en las grandes bases de datos es muy valiosa, sin embargo, en varias ocasiones, se hace difícil encontrar información puntual y específica acerca del tema que se va a investigar.

Por ello, quienes investigan temas relacionados al área de la terapia del lenguaje, no cuentan con una herramienta que les ayude a filtrar adecuadamente el contenido relacionado al tema.

Además, con respecto a esta área de estudio (terapia de lenguaje), es muy importante que el investigador esté al tanto de las nuevas técnicas y tecnologías que se desarrollen.

En donde se demuestra a continuación el diagrama de casos de usos:



3.2. Tecnología

La tecnología que se usó en el sistema como se muestra en la Ilustración 5.

Elsevier FingerPrint Engine: Se describe como un sistema de software back-end de técnicas de procesamiento de lenguaje natural (NLP) de última generación para extraer información de texto no estructurado.

GraphDB: Ontotext GraphDB es una base de datos gráfica altamente eficiente y robusta con soporte RDF y SPARQL.

Document Download Manager Scopus: Es una base de datos de referencias bibliográficas y citas de la empresa Elsevier que abarca temas como ciencia, tecnología, medicina que facilita la posibilidad de buscar información basada en el número de citas correspondientes a un tema en específico, además permite ver el grado de contenido científico dentro de los artículos.

CERMINE: (Content Extractor and Miner) es un sistema Open Source para java que permite extraer metadatos estructurados de artículos científicos.

NLP: El procesamiento del lenguaje natural (NLP, por sus siglas en inglés) es una rama de la inteligencia artificial que ayuda a las computadoras a entender, interpretar y manipular el lenguaje humano. NLP toma elementos prestados de muchas disciplinas, incluyendo la ciencia de la computación y la lingüística computacional, en su afán por cerrar la brecha entre la comunicación humana y el entendimiento de las computadoras.

TF-IDF: (Term Frequency - Inverse Document Frequency) es un método estadístico que refleja cuán importante es una palabra en un documento o corpus.

Django: Django es un framework de aplicaciones web gratuito escrito en Python, lo cual lo convierte en un conjunto de componentes que ayudan a desarrollar sitios web de manera más fácil y eficiente.

D3: Permite vincular datos arbitrarios a un modelo de objetos de documento (DOM) y luego transformarlas basadas en datos al documento.

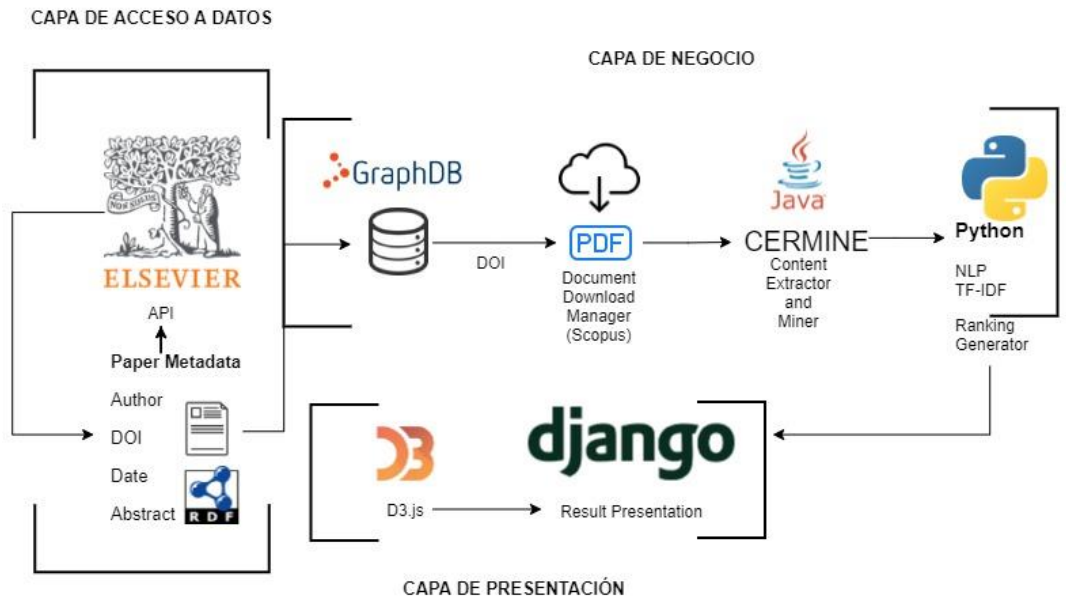


Ilustración 5.- Arquitectura Tecnológica

Está compuesta por 3 capas las cuales se describen a continuación:

- Capa de presentación: Se presenta el sistema al usuario, capturando información ingresada por el usuario en un mínimo proceso en la que se realiza un filtrado para comprobar que la información ingresada sea la correcta, también esta es la única capa que se comunica con la capa de negocio para procesar las diferentes peticiones y posteriormente presentar los resultados.

- Capa de Negocio: Esta capa ocupa un lugar preeminente en la construcción del sistema, ya que consta de procesamiento de lenguaje natural, técnicas de minería de datos y fórmulas que ayudan junto con la base de datos de conocimiento a la inferencia de artículos científicos, usando el lenguaje de programación Python.

- Capa de Acceso a Datos: En esta capa se gestionó el acceso a los datos de la aplicación, que nos provee la base de datos científica Scopus. Se utilizó el gestor de la base de datos de Scopus que realiza la recuperación y almacenamiento físico de datos basado en las solicitudes de la capa de negocio.

3.3. Arquitectura

Las funcionalidades que se detallaron en el buscador, dieron paso a la selección de la arquitectura que se puede apreciar en la Ilustración 6.

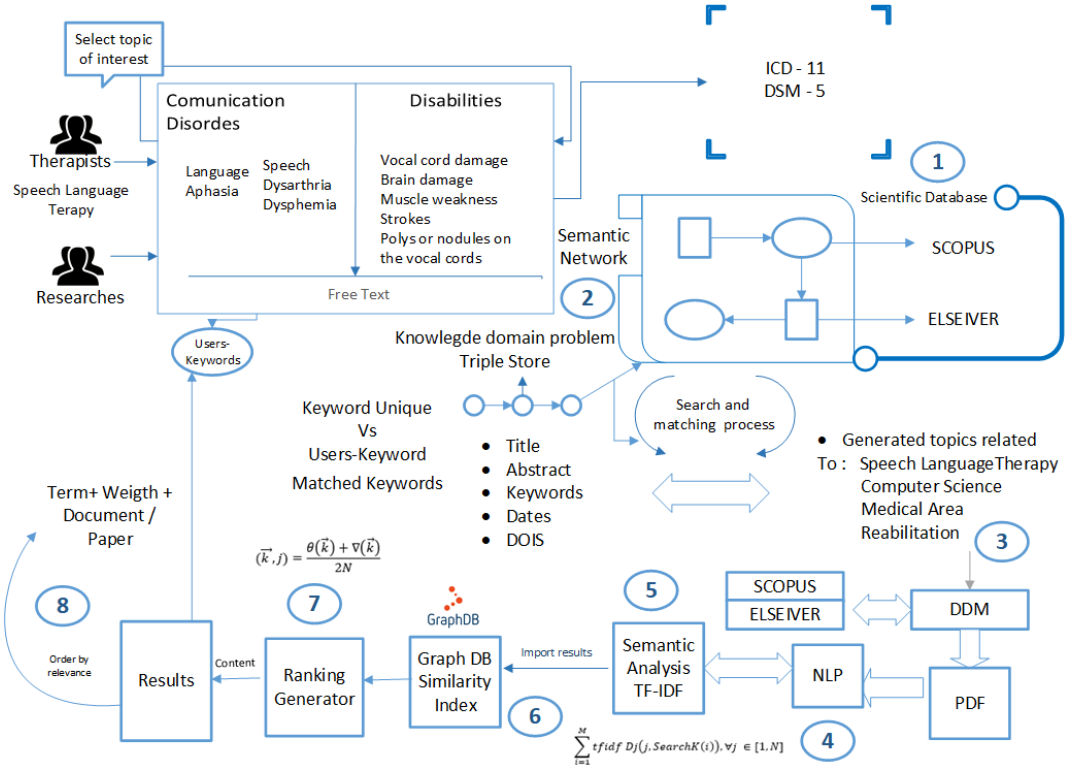


Ilustración 6.- Arquitectura del sistema.

El proceso de obtención de documentos científicos orientadas a la terapia de lenguaje se utilizó el lenguaje de programación Python y sigue el siguiente proceso:

Paso 1:

Ingreso de claves de búsqueda: Esta información puede ser ingresada por parte de los terapistas o los usuarios finales seleccionando un

tema de interés o las palabras claves que desea buscar dentro del corpus de artículos científicos.

Paso 2:

Obtención de artículos científicos: Los artículos científicos usados fueron obtenidos con el uso de la API[8] proporcionada en la página de Elsevier. Esta API permite realizar consultas en la base de datos científica y con ello se obtiene un archivo cuyo contenido es los metadatos de los artículos que arrojó la consulta.

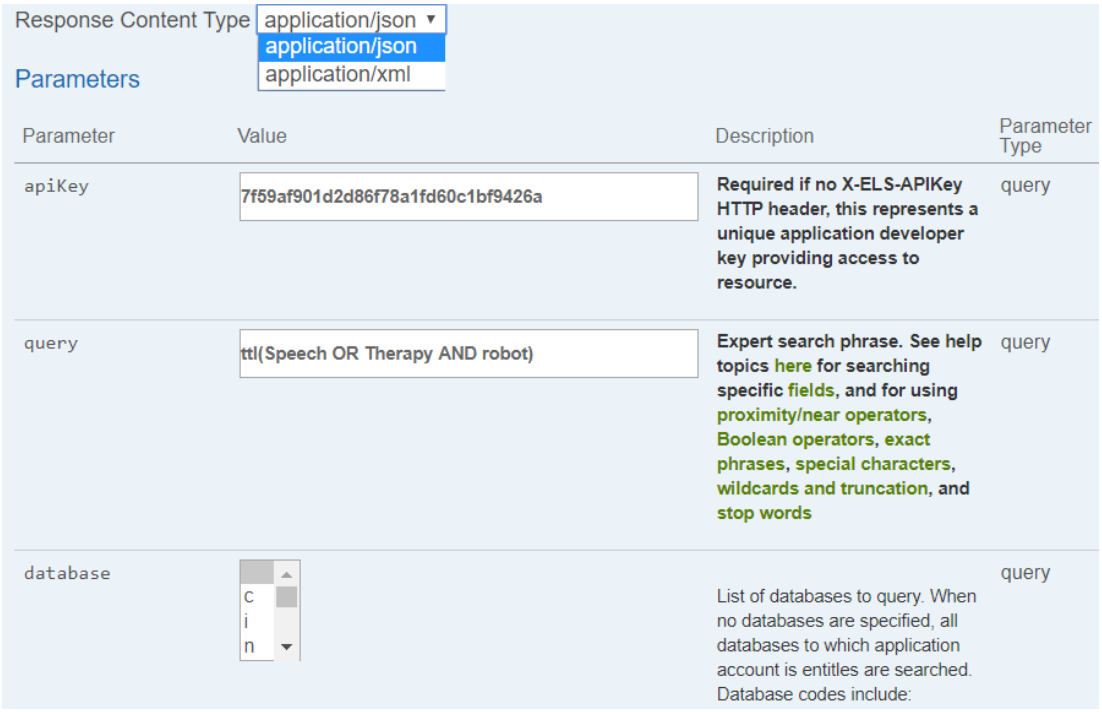


Ilustración 7.- Uso del API interactivo de Elsevier para realizar una consulta.

Adicionalmente, se adhiere a la consulta el dominio de las diferentes discapacidades dentro del ámbito de la terapia de lenguaje. Al finalizar se obtiene un conjunto de tripletas que se almacenan en una base de datos Graph DB.

[8] API: Application Programming Interface o Interfaz de programación de Aplicaciones

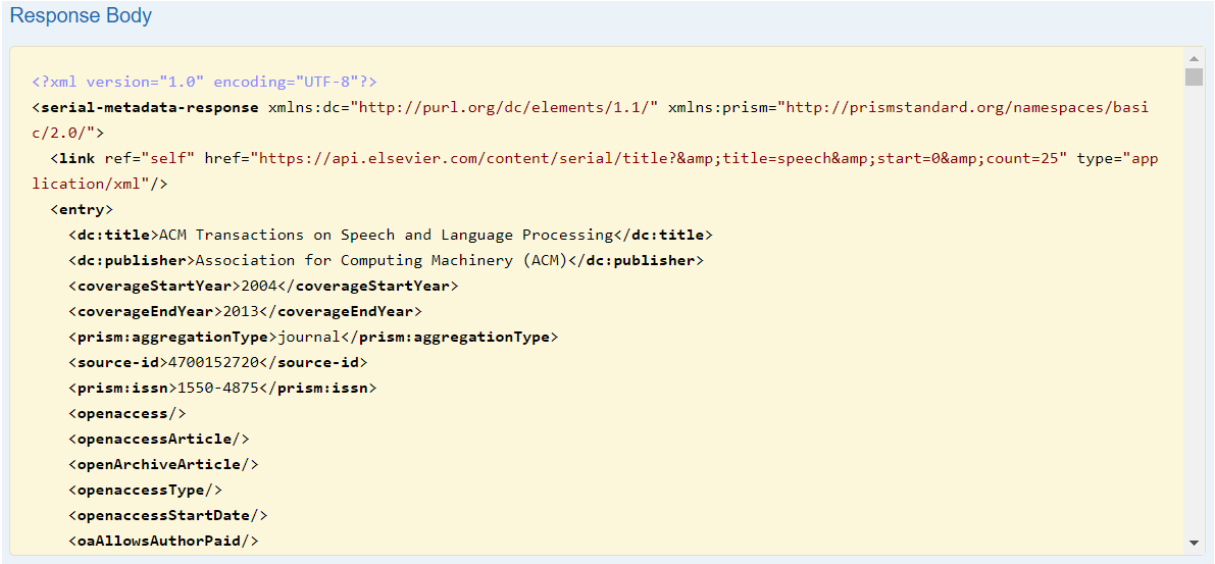


Ilustración 8.- Tripletas obtenidas como resultado de la consulta en formato XML.

Paso 3:

Obtención de documentos en formato PDF: Tras una consulta a la base de datos de tripletas se obtiene el DOI[9] de cada artículo. Con el uso de este identificador único se puede hacer una búsqueda avanzada en el buscador de Scopus y descargar los artículos en formato PDF gracias a la herramienta DDM[10] proporcionada por Scopus.

Paso 4:

Procesamiento del lenguaje natural (NLP): Tras la obtención de los artículos en formato PDF se necesita procesar su contenido, para ello se realizó los siguientes pasos:

- **Convertir de PDF a texto y extraer el cuerpo:** por lo general, la estructura de los artículos científicos comienza por el título, autores, la institución de la cual se publica, el cuerpo y las referencias bibliográficas. Al ser el cuerpo del artículo la parte más relevante para el análisis del texto, se usó una aplicación standalone[11] hecha en Java llamada CERMINE cuya entrada es un directorio con documentos PDF

[9] DOI: Digital Object Identifier, forma de identificar un objeto digital
[10] DDM: Document Download Manager, descargar resúmenes y pdf's completos
[11] standalone: Programa que no requiere de una conexión de red para funcionar

y devuelve un archivo de texto por cada artículo PDF analizado. Para correr la aplicación desde línea de comandos se hace de la siguiente manera:

```
java -cp cermine-impl-1.3-jar-with-dependencies.jar
pl.edu.icm.cermine.ContentExtractor -path pathDirecorio
```

- **Tratamiento del texto:** El tratamiento del texto se lo realizó siguiendo varias pautas del White Paper de Elsevier Fingerprint con la ayuda de varias librerías de Python. Los pasos son:

  ✓ **Tokenización:** dividir las oraciones en una lista de palabras

  Entrada: `'La terapia de lenguaje es importante'`

  Salida: `['La', 'terapia', 'de', 'lenguaje', 'es', 'importante']`

Con el uso de la librería NLKT se puede realizar la tokenización de la siguiente manera:

```
words = nltk.word_tokenize(open(texto.txt', "r",encoding='utf-8', er-
rors='ignore').read())
```

  ✓ **Dehyphenation:** es el proceso de eliminar los guiones usados para dividir palabras situadas al final de un a línea.

  Entrada: `'correc-/to'`

  Salida: `'correcto'`

Para ello se requiere recorrer la lista del texto tokenizado y buscar donde exista un guion seguido de un salto de carro[1] para reemplazarlo por un espacio vacío:

```
words = [item.replace('-\n', '') for item in words]
```

  ✓ **Normalización:** se usaron los siguientes pasos para realizar la normalización del texto:

Remover caracteres no ASCII: con el uso de la librería *unicodedata*[12] se eliminaron estos caracteres que no son de utilidad recorriendo cada palabra de la lista de del texto ya tokenizado de la siguiente manera

```
new_word = unicodedata.normalize('NFKD', word).encode('ascii','ignore').decode('utf-8', 'ignore')
```

Texto a minúscula y remover símbolos de puntuación: Para convertir el texto a minúscula se usa una función de Python para manejar cadenas de texto llamada tolower. Y para remover los símbolos de puntuación se usa una librería que permite aplicar expresiones regulares[1]

```
new_word = re.sub(r'[^\w\s]', '', word.tolower())
```

Remover *stopwords*[13]: Debido a que este tipo de palabras no representan alguna utilidad al análisis del texto, se las remueve recorriendo el listado de palabras y excluyendo las stopwords con el uso de la librería NLTK.

Lematización: Por razones gramaticales, el texto en los documentos usa diferentes formas para una palabra por ejemplo lee, leer, leyendo. Es muchas ocasiones, "sería útil para una búsqueda de una de estas palabras devolver documentos que contienen otra palabra en el conjunto" (https://nlp.stanford.edu/). Para encontrar el 'lema' de las palabras se hizo uso de la librería NLTK con la función WordNetLemmatizer.

Paso 5:

Análisis semántico: Para realizar el análisis semántico se utilizó el primero paso de la técnica LSA, este paso es el TF-IDF (Term Frequency – Inverse Document Frequency) en donde, se calcula los términos más frecuentes y el peso que tiene con respecto al cuerpo del documento. Su fórmula se realizó usando la librería sklearn[14] y contiene los siguientes pasos:

[12] *Unicodedata:* Proporciona acceso a los caracteres Unicode de la base de datos que define sus propiedades.
[13] *Stopwords:* Palabras que modifican o acompañan a otras, (pronombres, preposiciones, adverbios e incluso algunos verbos)
[14] Sklearn: Biblioteca de aprendizaje automático de software libre

- **Frecuencia de Términos (TF):** Se usó la función CountVectorizer que devuelve el número de veces que un token aparece en un documento y lo usa como peso

```
CountVectorizer(stop_words='english',min_df=3, max_df=0.5, ngram_ran
ge=(1, 5))
```

Esta función toma como parámetros un umbral para ignorar los términos del documento que tengan una frecuencia más alta o menor que la especificada (min_df y max_df), un conjunto de términos a ignorar (stop_words) ya sea porque ocurrieron muchas veces en el max_df o muy pocas veces en el min_df, y un rango para los n-gramas a extraer (ngram_range).

Luego, se usa la función fit_transfer que toma como parámetro el texto del documento.

```
cvec.fit_transform(document)
```

- **Frecuencia Inversa del Documento (IDF):** Esta parte de la fórmula se la puede encontrar con el uso de la función TfidfTransformer que después de transformarlo en un array[15], se lo almacena en un dataframe[16] junto a las palabras encontradas en el TF y el DOI del artículo.

```
weights_df = pd.DataFrame({

      'term': cvec.get_feataure_names(),

      'weight': weights, 'doi': doi[:-8]})
```

Ahora para obtener la frecuencia de una palabra o una lista de palabras contra los documentos se creó una función que recibe como parámetro el dataframe (weights_df) que se formó en el paso anterior y una lista de palabras (keySearch) de búsqueda.

[15] Array: Estructura de datos java que permite almacenar datos de un mismo tipo
[16] Dataframe: Datos tabulares con distintos tipos de columna

$$raking(Pi) = \sum_{i=1}^{k} i \sum_{j=1}^{h} j\, TFIDFDocument(j, KeySearch(i)) \quad (3)$$

```
peso = weights_df['term'].map(lambda x: x in keySearch)

       return weights_df[peso]['weight'].sum()
```

Donde:

k= Número de palabras claves de búsqueda.

h= Número de artículos que tiene al menos una palabra clave de búsqueda dentro del cuerpo textual.

term	weight	doi	wb
study	0.073956	10.1093_ageing_afn064	0.173956
step	0.044301	10.1161_STROKEAHA.109.563247	0.127331
adverse	0.027526	10.1016_j.brs.2016.06.004	0.124996
study	0.017417	10.1093_ije_dyt282	0.117417
motor	0.032672	10.1016_j.jalz.2014.04.514	0.116717

Ilustración 9.- Resultados aplicando la formula TF-IDF.

Paso 6:

Importar resultados a GraphDB y crear índice de similitud: El resultado obtenido por el algoritmo TF-IDF se exportó a un archivo csv separado por comas, es aquí donde se usó Orange para realizar un preprocesamiento de los datos y obtener un archivo csv con los campos necesarios para ser exportado a Karma.

Con el uso de este software se genera un archivo ttl el cual se cargará en GraphDB. Tas añadir la ontología, el csv y el modelo a Karma podemos apreciar una representación gráfica del archivo ttl que se exportará.

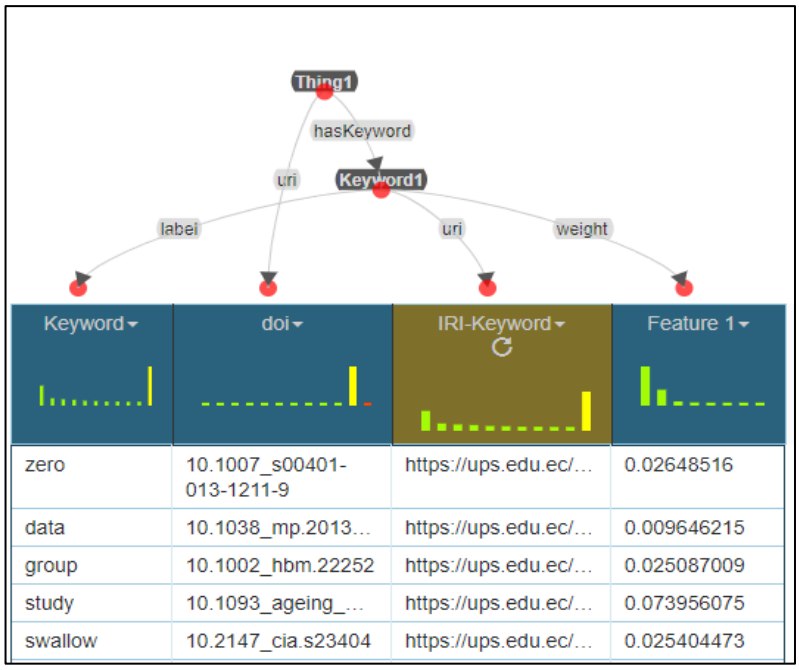


Ilustración 10. Integración de datos con Karma

Una vez exportado el archivo con las tripletas, se los puede importar a GraphDB de la siguiente manera

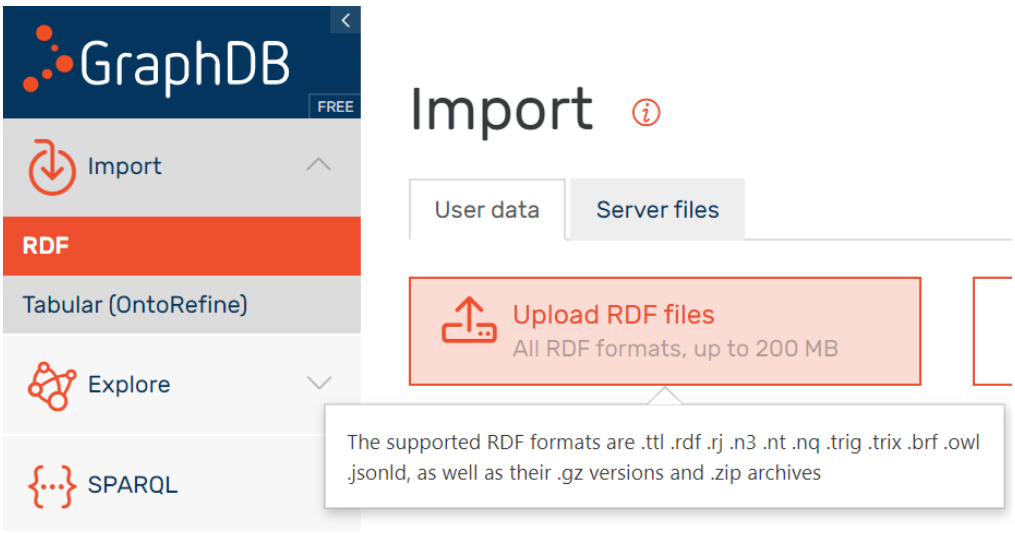


Ilustración 11. Importación de archivo ttl a GraphDB

Y ahora se procede a crear un índice de similitud para realizar búsquedas de la siguiente manera.

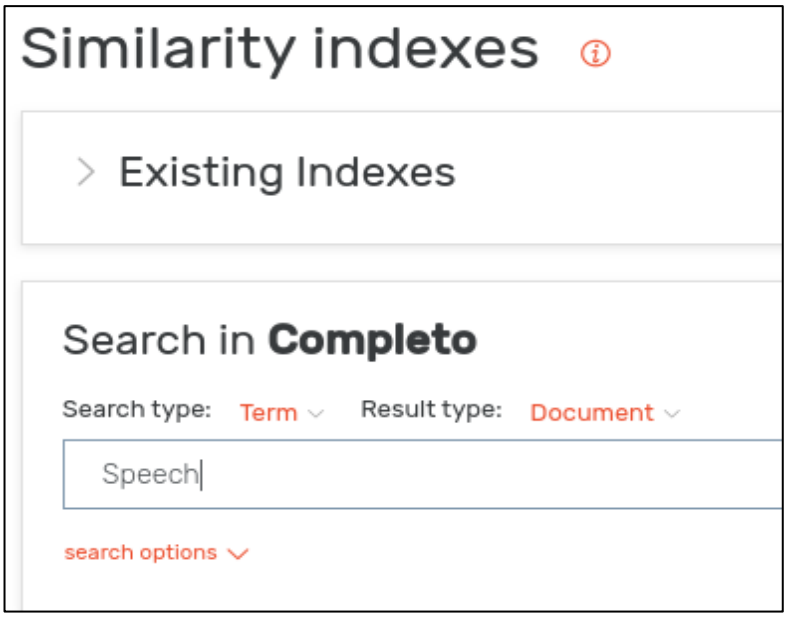


Ilustración 12.- Búsqueda en módulo de similitud.

Al usar el buscador del índice de similitud, este realiza una consulta SPARQL por detrás y se lo puede visualizar de la siguiente manera:

```
PREFIX : <http://www.ontotext.com/graphdb/similarity/>
PREFIX inst: <http://www.ontotext.com/graphdb/similarity/instance/>
PREFIX pred: <http://www.ontotext.com/graphdb/similarity/psi/>
insert {
    inst:Completo :createIndex "-termweight idf" ;
        :documentID ?documentID .
        ?documentID :documentText ?documentText .
} where {
    SELECT ?documentID ?documentText {
            ?documentID ?p ?documentText .
            filter(?p=dc:title || ?p=<http://www.ups.edu.ec/ontology/tesis#keyword-tf-idf>
            || ?p=<http://prismstandard.org/namespaces/basic/2.0/keyword>
            || ?p=<http://purl.org/ontology/bibo/abstract> )
            filter(isLiteral(?documentText))
```

Ilustración 13.- SPARQL para crear índice de similitud.

Para hacer uso del resultado obtenido en el módulo de similitud, se empleó el API REST [17] de GraphDB para obtener el SPARQL generado.

```
PARAMS = {'query': query}
```

[17] API REST: Micro framework con el que se desarrollan aplicaciones web.

```
r1 = requests.get(url=URL, params=PARAMS)
```

Donde:

Url es la dirección donde está desplegado el repositorio y los parámetros (query) es la consulta SPARQL en el índice de similitud.

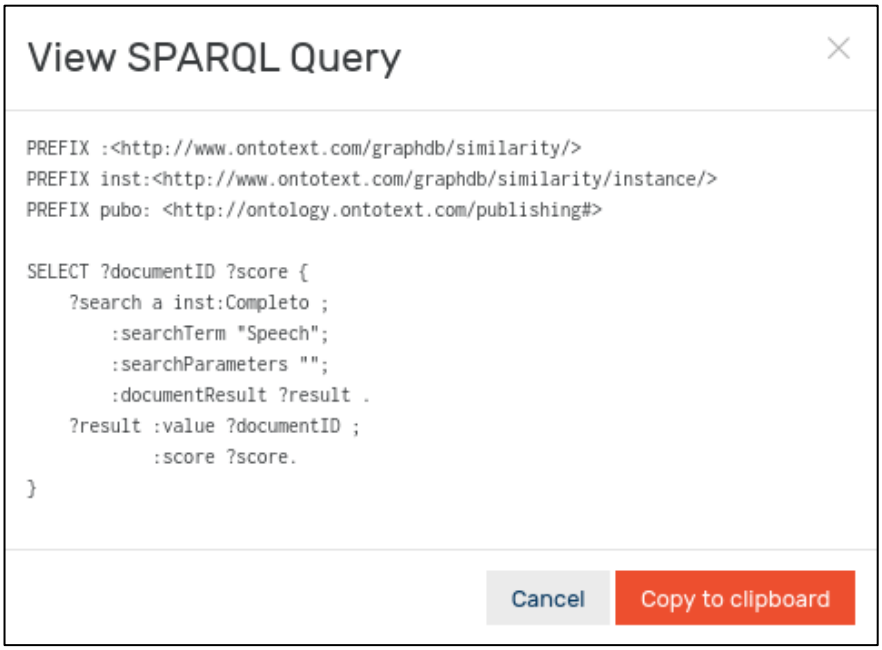


Ilustración 14.- Consulta SPARQL realizada por el índice de similitud.

Paso 7:

Generador de resultados: Para generar el ranking que será presentado al usuario, se combinaron los resultados obtenidos del algoritmo TF-IDF con los resultados obtenidos por el módulo de similitud creado usando la siguiente fórmula:

$$(\vec{k}, j) = \frac{\theta(\vec{k}) + \nabla(\vec{k})}{2N} \quad (4)$$

Donde:

$\vec{k}$ = es el vector de keywords que ingrese el usuario

j = documento analizado

θ = el TF-IDF aplicado al vector de keywords

∇ = el VSM aplicado por el módulo de similitud al vector de keywords

N = número total de keywords de búsqueda

Esto devuelve los cuatro mejores resultados para cada palabra que el usuario busque.

Paso 8:

Presentación de resultados: Para presentar los resultados al usuario se desplegó un simple sitio web usando Django que cuenta con un campo de búsqueda donde se puede ingresar una palabra o un conjunto de palabras para la búsqueda. El resultado se muestra en forma interactiva para el usuario como nodos, cada nodo es un artículo recomendado según el keyword ingresado por el usuario, siendo el nodo más grande, el que mejor puntaje de recomendación tiene. Esta representación gráfica con nodos fue hecha con el uso de la librería *D3js*.

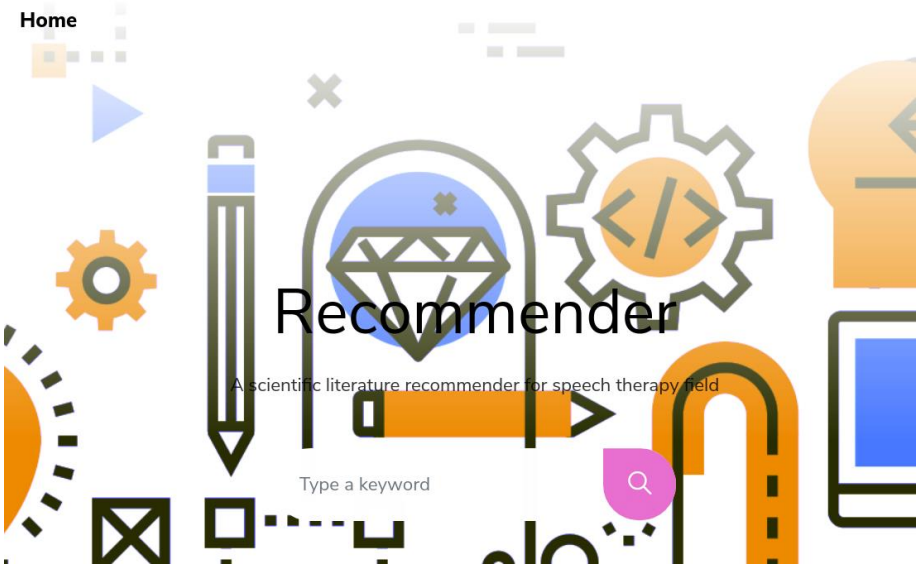


Ilustración 15.- Página principal de búsqueda desplegada con Django

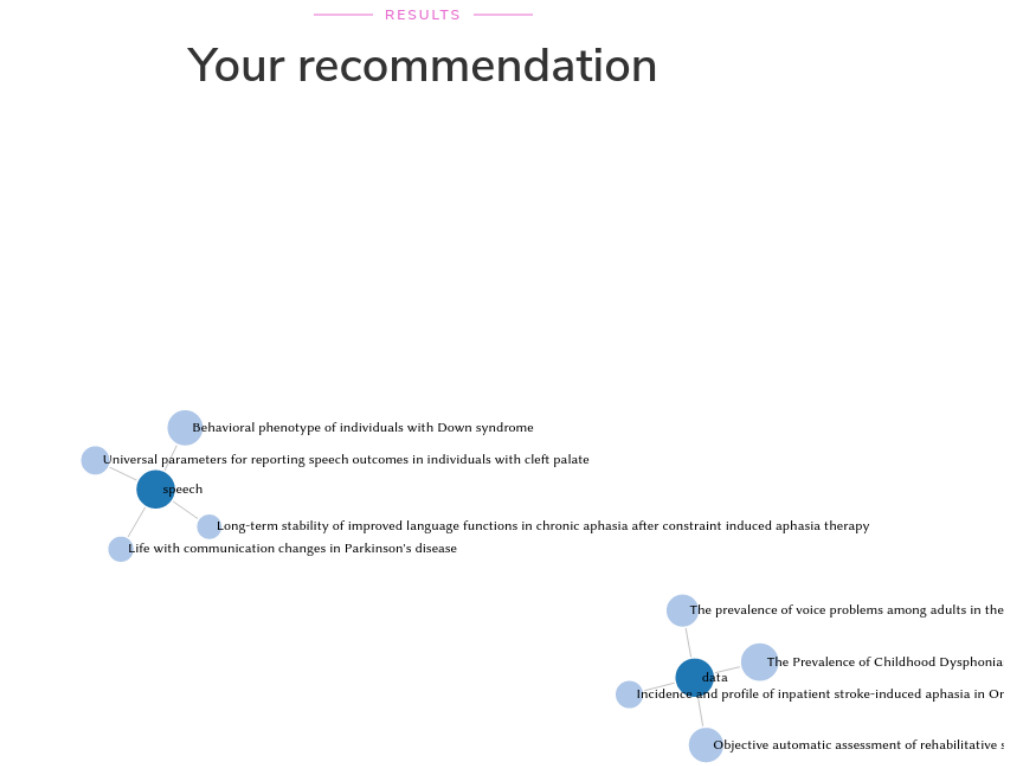


Ilustración 16.- Resultado de búsqueda, nodos creados con D3js

Métricas del sistema:

- Cantidad de artículos encontrados por palabra: 420 aproximados
- Tiempo de búsqueda por palabra: 30 segundos
- Uso del procesador: 83%
- Uso de memoria RAM: 0.9%

Las métricas fueron tomadas en una computadora con las siguientes características:

- Procesador: intel core i7
- RAM: 32 GB

CAPÍTULO 4.- ANÁLISIS DE RESULTADOS

4.1. Análisis técnicos de resultados

La base de datos de tripletas se pobló de manera correcta mediante los datos obtenidos con una consulta sparql a la base de datos científica Scopus.

La herramienta DDM, proporcionada por Scopus, facilitó la obtención de los artículos científicos. Los cuales pasaron por un procesamiento de datos con el uso de varias librerías. Tras este procedimiento el algoritmo implementado TF-IDF pudo mostrar resultados eficientes.

Con el uso del módulo de similitud de la base de datos GraphDB se creó un índice de similitud del cual se obtuvo una recomendación basada en el modelo VSM. Esta recomendación fue complementada con el algoritmo TF-IDF con el uso de una fórmula que permitió obtener resultados más precisos.

Se tuvo de dar un tratamiento de los datos obtenidos para generar un archivo json con el formato deseado y así este pueda ser usado para la visualización de los datos.

Se usó el framework Django para desplegar el sistema en forma de una aplicación web la cual hace la función de un buscador y así hacer uso de la librería d3js que facilita la presentación gráfica de los resultados.

4.1. Validación de los resultados obtenidos por el sistema

Para comparar los resultados obtenidos por el algoritmo TF-IDF y el módulo de similitud creado, se generaron varias búsquedas de los mismos términos en cada algoritmo y se imprimió una gráfica que represente cuan buena es la recomendación en los primeros quince artículos analizados.

Algunos ejemplos de los resultados son los siguientes.

Para la Ilustración 17 se realizó un análisis con la palabra compuesta Speech Therapy aplicando la fórmula de TF-IDF en donde se calcularon los pesos de acuerdo al número de artículos científicos.

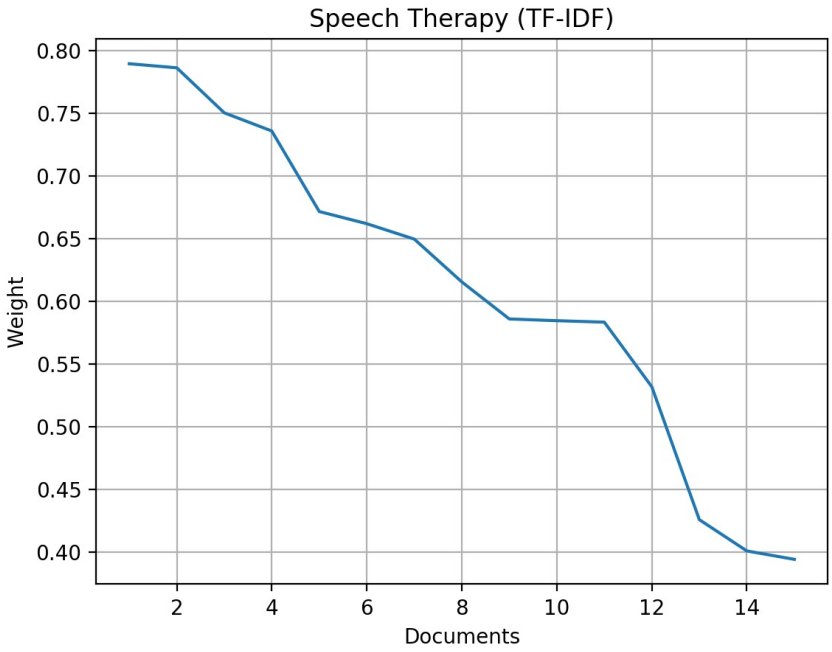


Ilustración 17.- Palabra Speech Therapy (TF-IDF)

Por otro lado, se realizó el proceso de análisis con la misma palabra en la Ilustración 18 haciendo uso del índice de similitud de Graph DB en donde se calcularon los pesos de acuerdo al número de artículos científicos consultados en la base de datos de conocimiento.

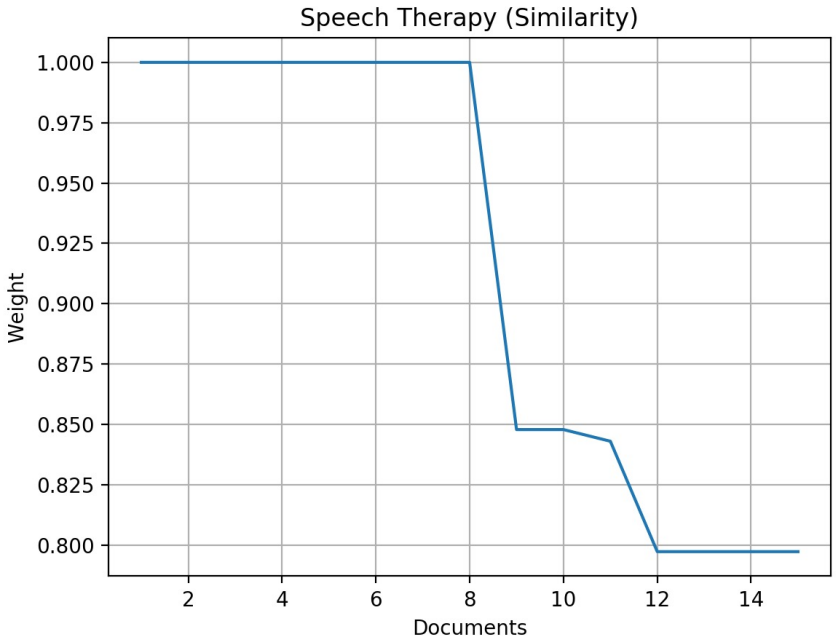


Ilustración 18.- Palabra Speech Therapy (Similarity)

A continuación, se muestra otro ejemplo de análisis usando la palabra Dyslexia usando la fórmula de TF-IDF y el módulo de similaridad respectivamente.

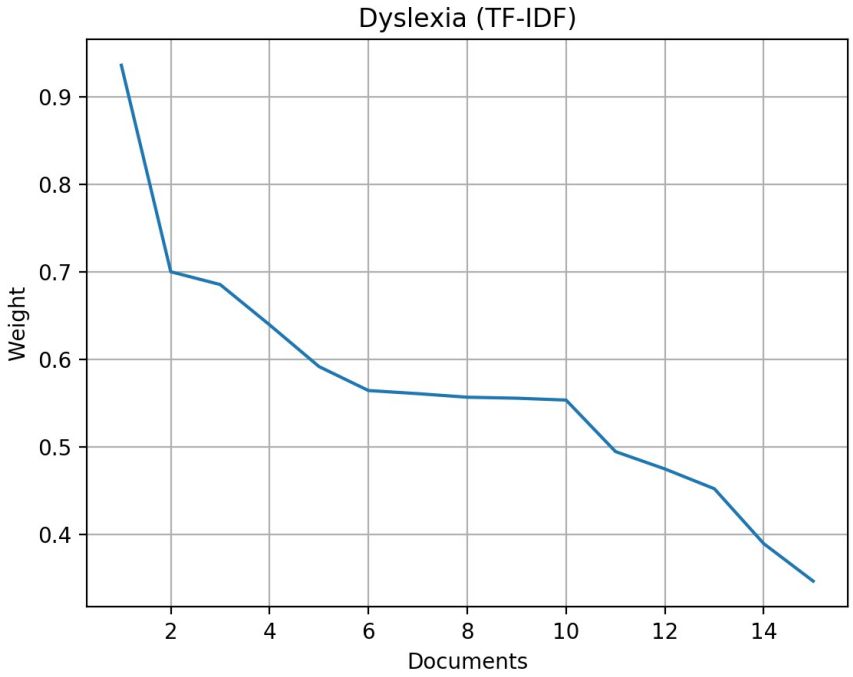


Ilustración 19.- Palabra Dyslexia (TF- IDF)

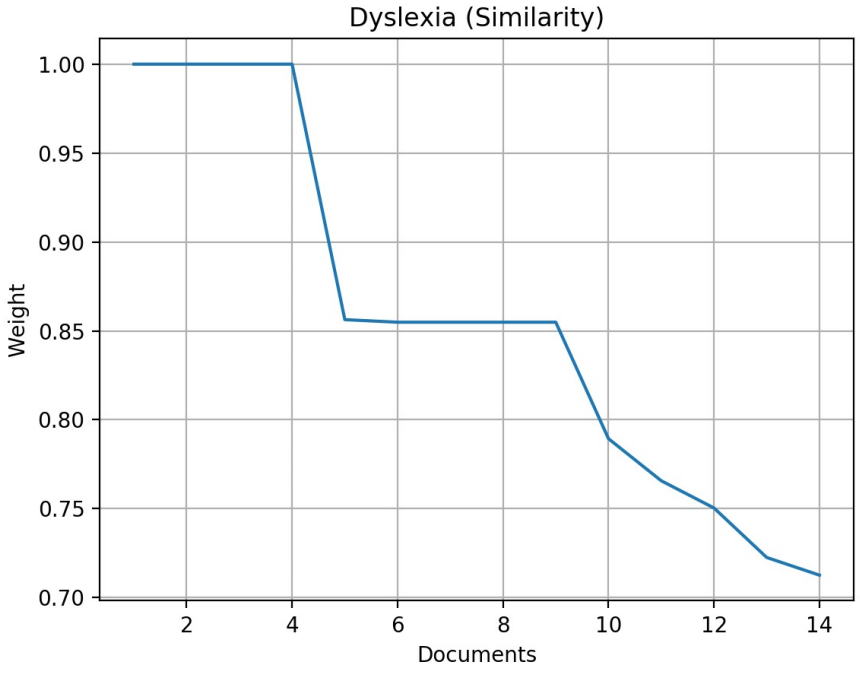


Ilustración 20.- Palabra Dyslexia (Similarity)

Como se observa, el índice de similitud obtiene mejores resultados de manera constante cuando se hace una búsqueda con palabras compuestas, pero al hacer una búsqueda de un solo término, el algoritmo TF-IDF encuentra mejores resultados que el índice de similitud. Es así que se implementó la fórmula de recomendación que une los dos resultados y para obtener la recomendación óptima para el usuario.

A continuación, se visualiza una tabla comparativa de los tres primeros resultados al buscar frases o términos directamente en el buscador de Scopus comparándolos con los resultados del sistema:

Palabra de búsqueda: Speech Therapy

Science Direct	**Puntaje (1-5)**	**Elsevier**	**Puntaje (1-5)**	**Sistema Recomendador**	**Puntaje (1-5)**
Language and Speech Disorders	5	Cleft Palate Speech	2	Change in vocal loudness following intensive voice treatment (LSVT®)	5
A Survivor's Perspective II	1	The Clinician's Guide to Treating Cleft Palate Speech - 2nd Edition	2	A systematic review of clinical outcomes, clinical process, healthcare utilization and costs associated with telerehabilitation	3
Ataxia	5	Speech and Language, Volume 5 - 1st Edition	4	Constraint-induced therapy of chronic aphasia after stroke	5
Total:	11		8		13

Palabra de búsqueda: Speech Disorder

Science Direct	**Puntaje (1-5)**	**Elsevier**	**Puntaje (1-5)**	**Sistema Recomendador**	**Puntaje (1-5)**
Speech and Language Disorders	5	Motor Speech Disorders – 3rd Edition	5	Discrimination of pathological voices using a time-frequency approach	5
Speech and Language Disorders	5	Motor Speech Disorders – 4th Edition	5	Vowel articulation in parkinson's disease	4
What Special Considerations Are Needed for Individuals With Amyotrophic Lateral Sclerosis, Multiple Sclerosis, or Parkinson Disease?	4	Intelligent Speech Signal Processing - 1st Edition	1	Automatic evaluation of articulatory disorders in Parkinson's disease	4
Total:	14		11		13

Palabra de búsqueda: Phonological disorder

Science Direct	**Puntaje (1-5)**	**Elsevier**	**Puntaje (1-5)**	**Sistema Recomendador**	**Puntaje (1-5)**
Phonological Disorders in Children? Design and user experience evaluation of a mobile serious game approach	5	Speech and Language, Volume 8 - 1st Edition	4	Metalinguistic awareness in phonologically disordered children	5

The nature of the phonological disorder in conduction aphasia	5	Speech and Language, Volume 2 - 1st Edition	4	Selective phonological impairment: A case of apraxia of speech	5
Toward Classification of Developmental Phonological Disorders	5	The Concise Encyclopedia of Language Pathology - 1st Edition	4	Consensus paper: Language and the cerebellum: An ongoing enigma	4
Total:	15		12		14

Palabra de búsqueda: Lack of fluency

Science Direct	**Puntaje (1-5)**	**Elsevier**	**Puntaje (1-5)**	**Sistema Recomendador**	**Puntaje (1-5)**
The impact of language co-activation on L1 and L2 speech fluency	4	Neurobiology of Language - 1st Edition	1	The frontal aslant tract underlies speech fluency in persistent developmental stuttering	5
Enhancing Oral Fluency as a Linguodidactic Issue	5	Are you 'information literate'?	1	Perspectives of speech language pathologists regarding success versus abandonment of AAC	5
Instructional Implications from the Woodcock–Johnson IV Tests of Achievement	5	Journal of Fluency Disorders	5	Maintenance of fluency: An experimental program	5
Total:	14		7		15

Palabra de búsqueda: Voice Disorder

Science Direct	**Puntaje (1-5)**	**Elsevier**	**Puntaje (1-5)**	**Sistema Recomendador**	**Puntaje (1-5)**
LANGUAGE AND SPEECH DISORDERS	5	Communication Disorders in Multicultural and International Populations	3	Voice disorders	5
MOTOR SPEECH AND SWALLOWING DISORDERS	5	Palliative Therapy in Otolaryngology - Head and Neck Surgery	1	Does Speech and Language Therapy Work?	5
Developmental-Behavioral Pediatrics	5	Functional Neurologic Disorders, Volume 139 - 1st Edition	2	Lecture notes of the Department of Speech and Hearing Sciences	5
Total:	15		6		15

Finalmente, las pruebas fueron realizadas al buscar una relación del resumen (abstract) del artículo con la palabra buscada.

Las palabras buscadas son los trastornos de habla más comunes en niños.

Resultados:

- Science Direct:69
- Elsevier: 44
- Sistema Recomendador: 70

Tras la prueba realizada se puede observar que el sistema recomendador tiene una mejora de 35% aproximadamente comparado con

el buscador de Elsevier, y de una mejora de 1% aproximadamente contra el buscador de Science Direct.

Las pruebas fueron realizadas al buscar una relación del resumen (abstract) del artículo con la palabra buscada.

CAPÍTULO 5.- CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Para la construcción del sistema se realizó un estudio previo en el ámbito de terapia de lenguaje y sus diferentes tipos de trastornos en las personas que nos proporcionó el conocimiento necesario para plantear, diseñar y desarrollar una herramienta de soporte útil para la investigación de contenido científico metodológico y de esta manera apoyar a las técnicas de inclusión para las personas con discapacidad. Representamos la información del conocimiento sobre los distintos valores de metadatos que tienen formas de aprendizaje, y a través de consultas sparql podamos obtener la información necesaria para el correcto funcionamiento de nuestro sistema recomendador en el ámbito de terapia de lenguaje.

Una gran cantidad de personas sufren de trastornos del habla y lenguaje, esto dificulta la manera en que se pueden relacionar con otras personas. Sin embargo, el debido tratamiento puede hacer una gran diferencia, en especial si se comienza desde temprana edad. Es por eso que el estar al tanto de los avances que ha dado la comunidad científica es de sumo interés para todas las personas que estén interesadas en este tema.

Tras el estudio de lo que implica el área de la terapia de lenguaje, se utilizó una ontología que permitió la creación de una base de datos científica de la cual, mediante consultas SPARQL, se pudo obtener información relevante de los datos almacenados en la misma. Esta información permitió la obtención de una gran cantidad de artículos científicos. Con el uso de varias técnicas de procesamiento de lenguaje natural y diferentes librerías, se logró la obtención de los datos relevantes de cada documento analizado para así poder llegar a una correcta recomendación.

El desarrollo de este proyecto presenta una ayuda a personas que desean obtener información científica de una de las mayores bases de datos que alojan artículos científicos. Esto permite que los usuarios del sistema puedan estar actualizados en temas referentes al ámbito de terapia de lenguaje. Debido a que el buscador permite ingresar varias palabras en una

sola búsqueda, se podrán obtener diferentes resultados los cuales se representan en un gráfico a manera de nodos. La representación gráfica ayuda al usuario a apreciar si existen documentos que están relacionados entre los términos de búsqueda y además se puede visualizar los enlaces a los mismos.

En la fase de análisis de resultados con al menos 500 artículos científicos, se hace una comparación de los datos que arroja la formula usando TF-IDF y de los datos que muestra usando el módulo de similaridad de GraphDB, demostrando así que las dos recomendaciones se unieron para crear una nueva fórmula, que ayuda a entregar al usuario los resultados de una mejor recomendación.

5.2. Recomendaciones

Se recomienda que las palabras que el usuario ingrese al buscador sean relacionadas al área de terapia de lenguaje, debido a que los documentos que el sistema analiza son exclusivos de esa área. Además, para obtener mejores resultados el usuario deberá insertar solo palabras claves como Therapy, Speech.

No ingresar una cantidad excesiva de palabras para la búsqueda, debido a que el sistema tarda un tiempo considerable entre 30 segundos en devolver los resultados ya que realiza la búsqueda en alrededor de 500 artículos científicos por cada palabra ingresada.

Para obtener un resultado efectivo el usuario debe ingresar palabras únicamente en inglés debido que la base de datos de Scopus nos proporciona artículos científicos únicamente en Ingles.

Se puede mejorar el rendimiento en tiempos de búsqueda de la arquitectura tecnológica, usando una librería de software para procesamiento de lenguaje natural avanzado llamada Spacy.

Usar la base de datos orientado a grafos (BDOG) Neo4j para mejorar el rendimiento de la misma en su agilidad de la gestión de datos, en donde si se requiere superar el límite de su capacidad se necesitaría un volumen mayor

de 34000 millones de nodos (datos), 34000 millones de relaciones entre esos datos, 68000 millones de propiedades y 32000 tipos de relaciones.

5.3. Trabajos Futuros

Durante el desarrollo de sistema recomendador se identificaron varios aspectos importantes que se pueden tener en cuenta para realizar trabajos futuros y mejorar el mismo:

- Agregar al buscador un auditor de las búsquedas planteadas en el ámbito de terapia de lenguaje y las recomendaciones que muestra el sistema.
- Añadir un historial de búsquedas por usuario en el tema a investigar.
- Dar la facilidad al buscador de cambiar la base de conocimiento para investigar temas de otras áreas.
- Proporcionar un mayor detalle en las búsquedas de cada palabra obteniendo no solo un orden jerárquico, si no una pequeña introducción de cada artículo científico que recomienda el sistema.
- Desarrollar una aplicación móvil para el uso del buscador.

REFERENCIAS BIBLIOGRÁFICAS

[1] B. Mundial, discapacidad: panorama general, vol. 2016. 2015

[2] Chuchuca, F. M. (2017). desarrollo de un sistema de información con soporte inteligente para brindar apoyo en el estudio de casos clínicos para estudiantes de fonoaudiología. Cuenca, Azuay, Ecuador.

[3] Cristian Timbi-Sisalima, v. R.-b.-z.-a.-a. (07 de julio de 2015). Adacof: una aproximación educativa basada en tic para el aprendizaje digital de la articulación del código fonético en niños con discapacidad. México: scielo.

[4] Moreno-Flagge, n. (2013). Trastornos del lenguaje. Diagnóstico y tratamiento. Rev neurol, 57(supl 1), s85-94.

[5] Llorente, A. (30 de 08 de 2016). BBC NEWS. Obtenido de BBC NEWS: https://www.bbc.com/mundo/noticias-36983267

[6] Bykbaev, V. E. (06 de septiembre de 2016). Contribución a los modelos de soporte pedagógico basados en TICs y sistemas inteligentes como herramientas de apoyo a la educación especial y terapia de lenguaje. España.

[7] Enrique Herrera-Viedma, C. P. (15 de 02 de 2004). Hipertext.net. Obtenido de Hipertext.net: https://www.upf.edu/hipertextnet/numero-2/recomendacion.html?fbclid=IwAR3nmZoDacI8zC2vL6VK-ljkXUfIg5-RMtaWhIfy-bZn4oEuwDpJy6iUrvs

[8] Scopus. (2017). Scopus Content Guide. Empowering Knowledge.

[9] Engine, E. F. (2019). Research Intelligence Elsevier. Retrieved from Research Intelligence Elsevier: https://www.elsevier.com/__data/assets/pdf_file/0010/175249/ACAD_FP_FS_FPEFactSheet2019_WEB.pdf

[10] C. Cleverdon, M. Keen. Factors Determining the Performance of Indexing Systems, Vol. 2--Test Results. ASLIB Cranfield Res. Proj., Cranfield, Bedford, England, 1966.

[11] P. V. León, «¿Ontología u Ontologías? » Disputatio. Philosophical Research Bulletin, vol. 4, nº 5, pp. 299-339, 2015.

[12] V. J. S. K. R. Subhashini, «Shallow NLP Techniques for Noun Phrase Extraction, » IEEE, Chennai, India, 2010

[13] A. Zigoni, «Sci Val,» Scopus, [En línea]. Available: https://www.recursoscientificos.fecyt.es/sites/default/files/fecyt_scopus_apis_curso_20130903_es_def.pdf. [Último acceso: 5 12 2019].

[14] M. &. H. I. &. D. Y. &. M. G. &. N. S. &. J. S. &. S. T. Dayarathna, «Acacia-RDF: An X10-Based Scalable Distributed RDF Graph Database Engine, » Moratuwa, Sri Lanka, 2016.

[15] Reyes-lillo, d. (octubre de 2018). Integración semiautomática de tecnologías de la web semántica en bases de datos de patentes. La plata, buenos aires, argentina.

[16] Ljubljana, u. O. (24 de 08 de 2019). Orange.biolab.si. Obtenido de orange-visual programming data mining fruitful and fun: https://orange.biolab.si/home/visual-_programming/

[17] Protegé wiki. (06 de 08 de 2019). Obtenido de protegé wiki: https://protegewiki.stanford.edu/wiki/main_page

[18] C. Knoblock, «Karma, » University Southern California, 2016. [En línea]. Available: https://usc-isi-i2.github.io/karma/#home.

[19] Labra gayo, j. E. (2011). Web semántica: comprendiendo el cambio hacia la web 3.0. La coruña: netbiblo.

[20] Mishra, a., & vishwakarma, s. (2015). Analysis of tf-idf model and its variant for document retrieval. 2015 international conference on computational intelligence and communication networks (cicn).

[21] D. S. G. César Perez López, Mineria de Datos Técnicas y Herramientas, Madrid, España: Clara M de la Fuente Rojo, 2010.

[22] A. Kiryakov, «Ontotext, » Graph DB 9.0, 7 11 2019. [En línea]. Available: http://graphdb.ontotext.com/documentation/standard/.

[23] Django documentation. (1 de 9 de 2019). Obtenido de django documentation: https://docs.djangoproject.com/en/2.2/topics/

[24] Dominika tkaczyk, paweł szostek, mateusz fedoryszak, piotr jan dendek, łukasz bolikowski (2015). Cermine: automatic extraction of structured metadata from scientific literature.

[25] M. Lincoln, «Uso de SPARQL para acceder a datos abiertos enlazados,» 24 11 2015. [En línea]. Available: https://programminghistorian.org/es/lecciones/retirada/sparql-datos-abiertos-enlazados?fbclid=IwAR0wZZ2ETFtzefQn8aswBp1D016CGo8tJY3ZygFAoaxeFBZBbMeiDd5znfg#buscando-rdf-con-sparql.

[26] NLTK, «Documentación de NLTK 3.4.5,» NLTK, 20 08 2019. [En línea]. Available: https://www.nltk.org/.

[27] «TextBlob: Simplified Text Processing, » TextBlob, [En línea]. Available: https://textblob.readthedocs.io/en/dev/index.html#. [Último acceso: 08 12 2019]

[28] D. R. 2.2.2, «RDFLib,» [En línea]. Available: https://rdflib.readthedocs.io/en/stable/. [Último acceso: 08 12 2019].

[29] M. Bostock, «Documentos Basados en Datos,» dj3js, 2019. [En línea]. Available: https://d3js.org/.

[30] I. Crea, «Inevery Crea,» 20 04 2015. [En línea]. Available: https://ineverycrea.net/comunidad/ineverycrea/recurso/12-herramientas-para-generar-en-el-aula-recursos/82443c12-f775-429c-95b2-fd8d5f9c9f5f?rdf. [Último acceso: 03 01 2020].

[31] W. R. E. P. Dhomas Hatta Fudholi, «Ontology-Based Information Extraction for Knowledge Enrichment and Validation,» IEEE XPLORE, nº 10.1109/AINA.2016.70, pp. 1-5, 2016.

[32] C. Hao, «Research on Knowledge Model for Ontology-Based,» IEEE XPLORE, pp. 1-6, 2016.

[33] A. F. D. S.-T. a. H. S. Francesco Ronzano, «Making Sense of Massive Amounts of Scientific Publications: the Scientific Knowledge Miner Project,» BIRNDL , pp. 1-6, 2016.

[34] H. Bolin, «Knowledge Extraction Based on Sentence Matching and Analyzing,» International Symposium on Knowledge Acquisition and Modeling, pp. 1-5, 2018.

MANUAL DE USUARIO

SISTEMA INTELIGENTE BASADO EN ONTOLOGÍAS, PROCESAMIENTO DEL LENGUAJE NATURAL Y TECNICAS DE MINERIA DE DATOS PARA RECOMENDAR CONTENIDO CIENTÍFICO-METODOLÓGICO DEL ÁMBITO DE TERAPIA DE LENGUAJE

Luis Adrian Tobar Almache – Adrian Ricardo Morales Jimenez
ltobara@est.ups.edu.ec – amoralesj1@est.ups.edu.ec

Tabla de Contenido

1. Introducción

A lo largo de este proyecto de investigación, se emplea una serie de técnicas de procesamiento de lenguaje natural y minería de datos que ayudan a optimizar la búsqueda de temas relacionados a el ámbito de terapia de lenguaje

En el tema de discapacidad es importante demarcar cifras como indica el Banco Mundial alrededor de “1000 millones de habitantes, equivalente al 15 % de la población mundial” tiene algún tipo de discapacidad, es por ello que un proceso de enseñanza exige una búsqueda de información exhaustiva para encontrar información relacionada con la metodología de educación planteada, dirigida a un grupo de personas en específico.

Puesto que uno de los problemas que mayormente representan un obstáculo es la posibilidad de brindar una educación que resulte inclusiva y de buena calidad para todos, es debido a esta necesidad que se realiza este sistema para fortalecer y complementar los procesos de enseñanza educativos haciendo uso de la tecnología como medio de educación, con el fin de transmitir contenido científico metodológico en el ámbito de la terapia de lenguaje.

1.1. Propósito

El propósito de este documento es brindar un manual técnico en el cual se da a conocer la manera en la que fue implementado el sistema.

1.2. Alcance

Este documento va dirigido a cualquier persona que esté interesada en entender la manera en la que está implementado el sistema.

2. Manual de ejecución

2.1. Requerimientos

2.1.1. Hardware

- RAM >= 8 GB
- Procesador intel i3 o mejor
- Espacio en disco 2 GB

2.1.2. Software

- Windows 7 o mayor, Linux 4.4 o mayor, mac os Sierra o mayor
- Python >= 3.0.0
- Java JDK 1.8
- API Elsevier
- Django >= 2.5.0
- GraphDB >= 2.0.0
- D3js

2.1.3. Configuraciones, instalaciones de paquetes, etc.

Para la puesta en marcha se tendrán que instalar todos los requerimientos de software especificados en el punto 2.1.2 de la siguiente manera:

- Python: dirigirse a la página oficial (python.org) y descargar la última versión disponible para el sistema operativo en el cual se va a implementar. Ejecutar instalador.
- Java JDK: dirigirse a la página oficial (oracle.com) y descargar la última versión disponible para el sistema operativo en el cual se va a implementar. Ejecutar instalador.
- En caso de que se deseen hacer más consultas a Elsevier, se deberá conseguir una llave en la página de Elsevier Developers (dev.elsevier.com)
- Django: pip install Django

- GraphDB: descargar la versión gratuita de GraphDB y ejecutar como administrador el archivo con extensión bat o como super usuario el archivo sh (graphdb.ontotext.com).
- D3js: descargar el archivo zip desde la página ofical y descomprimirlo (d3js.org).

Luego se procederá a instalar todas las librerías de python con el comando pip (pip install –r requirements.txt), estas librerías están detalladas en el archivo *requirements.txt,* dentro de un entorno virtual si así se desea.

2.1.4. Creación de entornos de desarrollo / Configuración de servidores

Para poder desplegar la aplicación se deberá iniciar primero GraphDB de la siguiente manera:

- Windows: Abrir una ventada de CMD o PowerShell con permisos de administrador y escribir el PATH al archivo “graph.bat”.
- Linux/Mac: abrir una terminal y ejecutar como super usuario el archivo “graphdb”.

Ahora se procede a desplegar Django en el puerto 8000 de la siguiente manera:

Dentro de una terminal o CMD dirigirse a la ruta del proyecto Django y ejecutar el comando

```
python manage.py runserver
```

Por defecto va a ser desplegado en localhost, es por ello que para visualizar la aplicación se deberá dirigir a un explorador web y escribir la siguiente URL: localhost:8000

2.1.4.1. Realizar una búsqueda

Para realizar una búsqueda se deberá escribir una o más palabras – frases en el idioma inglés separadas por una coma (ej: speech therapy,kid,robot) en el buscador que se muestra en la Figura 2.

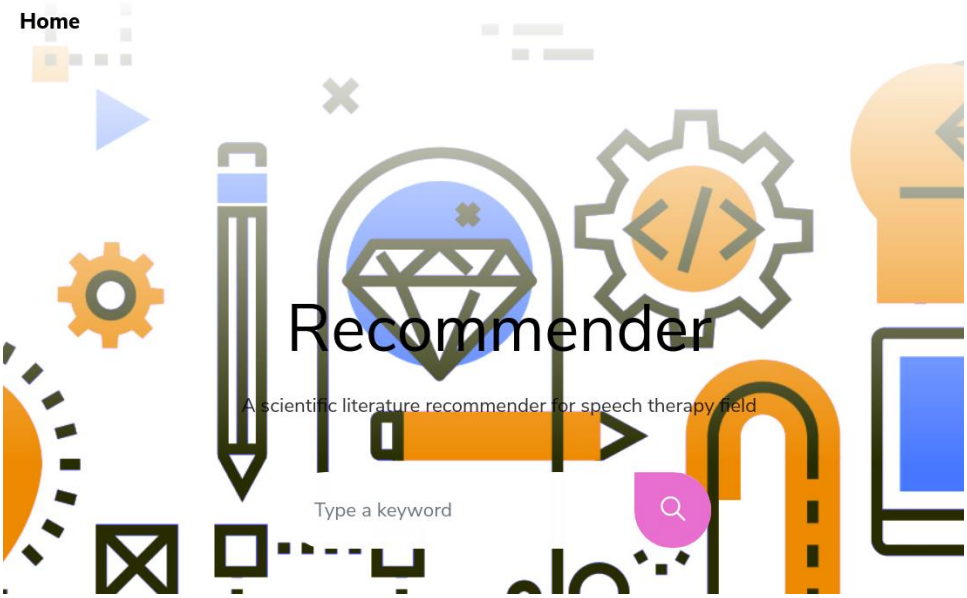


Ilustración 21.- Pantalla principal del sistema

Cuando se termine de ingresar las palabras para la búsqueda, se deberá hacer clic el ícono de la lupa o en su defecto presionar la tecla Enter. Tras este paso se deberá esperar a que la página se recargue y así el sistema arroje una recomendación.

Una vez que la página termine de cargar, se deberá desplegar hacia abajo en la página o hacer clic en el ícono de flecha (como se puede ver en la Figura3) para visualizar los resultados en forma gráfica representados por nodos.

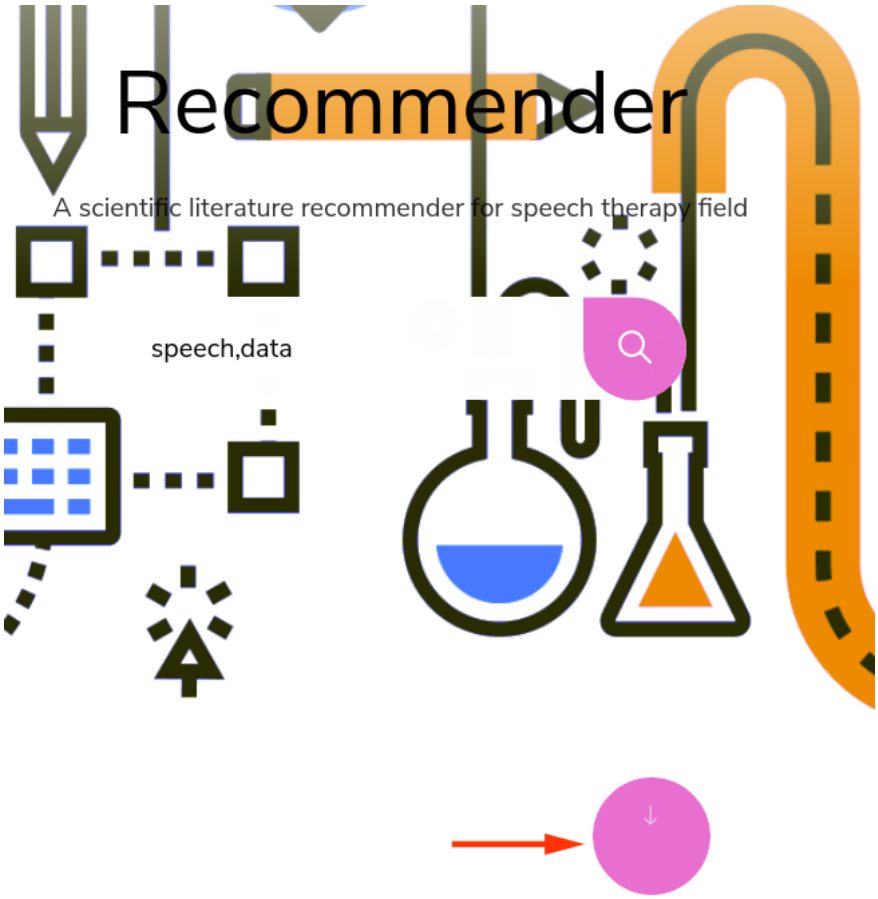


Ilustración 22.- Botón para desplegar resultados

2.1.4.2. Representación de resultados

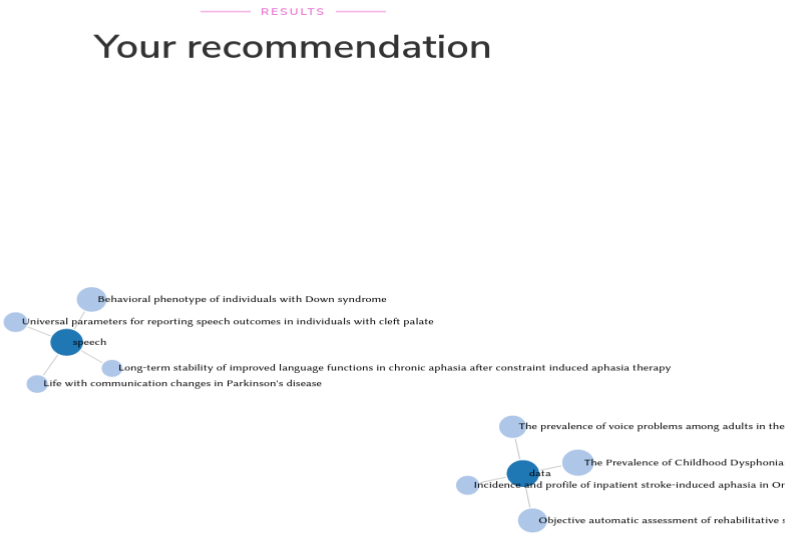


Ilustración 23.- Representación gráfica de resultados

Los nodos mostrados en la Figura 4 se los representa de la siguiente manera:

Por cada palabra se creará un nodo de color azul (nodo de búsqueda)

Cada recomendación (nodo recomendación) que se genere estará enlazada al nodo de búsqueda

Mientras más grande sea el nodo de recomendación significa que esa recomendación tiene una mayor relación con la palabra de búsqueda

2.1.4.3. Enlaces a artículos de la recomendación

Ilustración 24.- Enlaces de los Artículos

Como se puede observar en la Figura 5, se creará un apartado para cada palabra que se ingrese en el campo de búsqueda y se presentarán enlaces a los artículos recomendados.

2.2. Consideraciones / Recomendaciones

El tiempo de búsqueda por cada palabra es alrededor de treinta segundos.

Todas las palabras de búsqueda deberán estar en inglés, debido a que los artículos analizados están en dicho idioma.

MANUAL TÉCNICO

SISTEMA INTELIGENTE BASADO EN ONTOLOGÍAS, PROCESAMIENTO DEL LENGUAJE NATURAL Y TECNICAS DE MINERIA DE DATOS PARA RECOMENDAR CONTENIDO CIENTÍFICO-METODOLÓGICO DEL ÁMBITO DE TERAPIA DE LENGUAJE

Luis Adrian Tobar Almache – Adrian Ricardo Morales Jimenez
ltobara@est.ups.edu.ec – amoralesj1@est.ups.edu.ec

Tabla de contenido

2. Introducción

A lo largo de este proyecto de investigación, se emplea una serie de técnicas de procesamiento de lenguaje natural y minería de datos que ayudan a optimizar la búsqueda de temas relacionados a el ámbito de terapia de lenguaje

En el tema de discapacidad es importante demarcar cifras como indica el Banco Mundial alrededor de “1000 millones de habitantes, equivalente al 15 % de la población mundial” tiene algún tipo de discapacidad, es por ello que un proceso de enseñanza exige una búsqueda de información exhaustiva para encontrar información relacionada con la metodología de educación planteada, dirigida a un grupo de personas en específico.

Puesto que uno de los problemas que mayormente representan un obstáculo es la posibilidad de brindar una educación que resulte inclusiva y de buena calidad para todos, es debido a esta necesidad que se realiza este sistema para fortalecer y complementar los procesos de enseñanza educativos haciendo uso de la tecnología como medio de educación, con el fin de transmitir contenido científico metodológico en el ámbito de la terapia de lenguaje.

2.3. Propósito

El propósito de este documento es brindar un manual técnico en el cual se da a conocer la manera en la que fue implementado el sistema.

2.4. Alcance

Este documento va dirigido a cualquier persona que esté interesada en entender la manera en la que está implementado el sistema.

3. Manual de ejecución

3.1. Requerimientos

3.1.1. Hardware

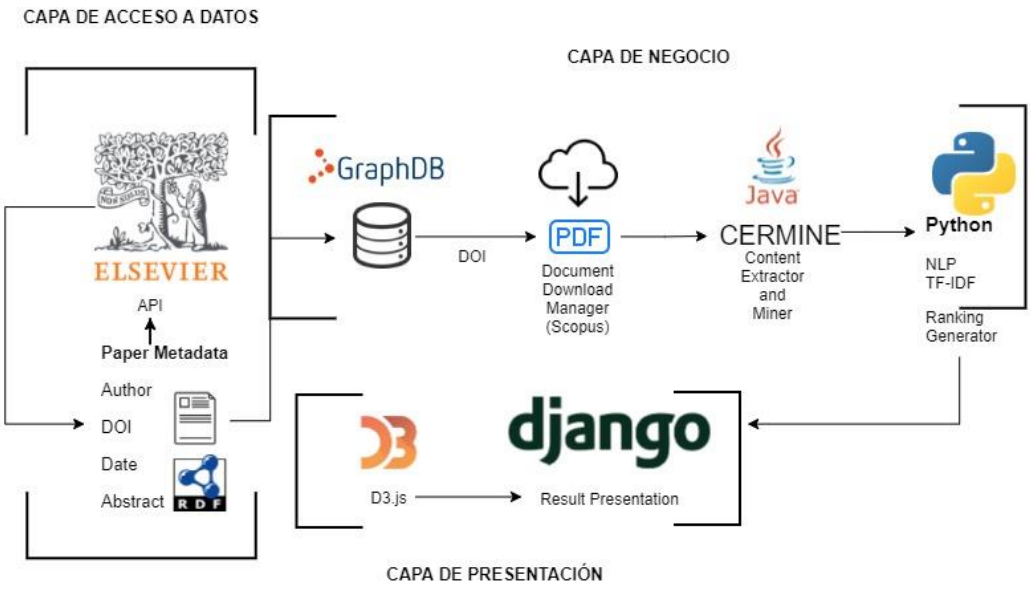
- RAM >= 8 GB
- Procesador intel i3 o mejor
- Espacio en disco 2 GB

3.1.2. Software

- Windows 7 o mayor, Linux 4.4 o mayor, mac os Sierra o mayor
- Python >= 3.0.0
- Java JDK 1.8
- API Elsevier
- Django >= 2.5.0
- GraphDB >= 2.0.0
- D3js

3.2. Diseño / Construcción

El sistema está compuesto por tres capas visualizadas en la siguiente figura:



3.2.1. Configuraciones, instalaciones de paquetes, etc.

Para la puesta en marcha se tendrán que instalar todos los requerimientos de software especificados en el punto 2.1.2 de la siguiente manera:

- Python: dirigirse a la página oficial (python.org) y descargar la última versión disponible para el sistema operativo en el cual se va a implementar. Ejecutar instalador.
- Java JDK: dirigirse a la página oficial (oracle.com) y descargar la última versión disponible para el sistema operativo en el cual se va a implementar. Ejecutar instalador.
- En caso de que se deseen hacer más consultas a Elsevier, se deberá conseguir una llave en la página de Elsevier Developers (dev.elsevier.com)
- Django: *pip install Django*
- GraphDB: descargar la versión gratuita de GraphDB y ejecutar como administrador el archivo con extensión bat o como super usuario el archivo sh (graphdb.ontotext.com).
- D3js: descargar el archivo zip desde la página oficial y descomprimirlo (d3js.org).

Luego se procederá a instalar todas las librerías de python con el comando pip (pip install –r requirements.txt), estas librerías están detalladas en el archivo requirements.txt, dentro de un entorno virtual si así se desea.

3.2.2. Creación de entornos de desarrollo / Configuración de servidores

Para poder desplegar la aplicación se deberá iniciar primero GraphDB de la siguiente manera:

- Windows: Abrir una ventada de CMD o PowerShell con permisos de administrador y escribir el PATH al archivo “graph.bat”.
- Linux/Mac: abrir una terminal y ejecutar como super usuario el archivo “graphdb”.

Ahora se procede a desplegar Django en el puerto 8000 de la siguiente manera:

Dentro de una terminal o CMD dirigirse a la ruta del proyecto Django y ejecutar el comando

```
python manage.py runserver
```

Por defecto va a ser desplegado en localhost, es por ello que para visualizar la aplicación se deberá dirigir a un explorador web y escribir la siguiente URL: localhost:8000

3.2.2.1. Realizar una búsqueda

Para realizar una búsqueda se deberá escribir una o más palabras – frases en el idioma inglés separadas por una coma (ej: speech therapy,kid,robot) en el buscador que se muestra en la Figura 2.

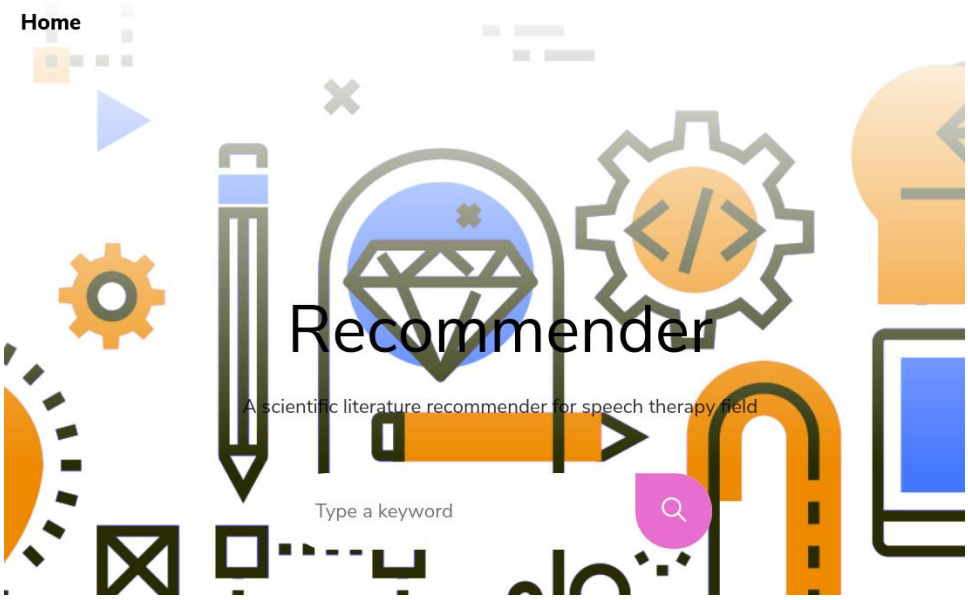


Figura 2. Buscador

Cuando se termine de ingresar las palabras para la búsqueda, se deberá hacer clic el ícono de la lupa o en su defecto presionar la tecla Enter. Tras este paso se deberá esperar a que la página se recargue y así el sistema arroje una recomendación.

Una vez que la página termine de cargar, se deberá desplegar hacia abajo en la página o hacer clic en el ícono de flecha (como se puede ver en la Figura3) para visualizar los resultados en forma gráfica representados por nodos.

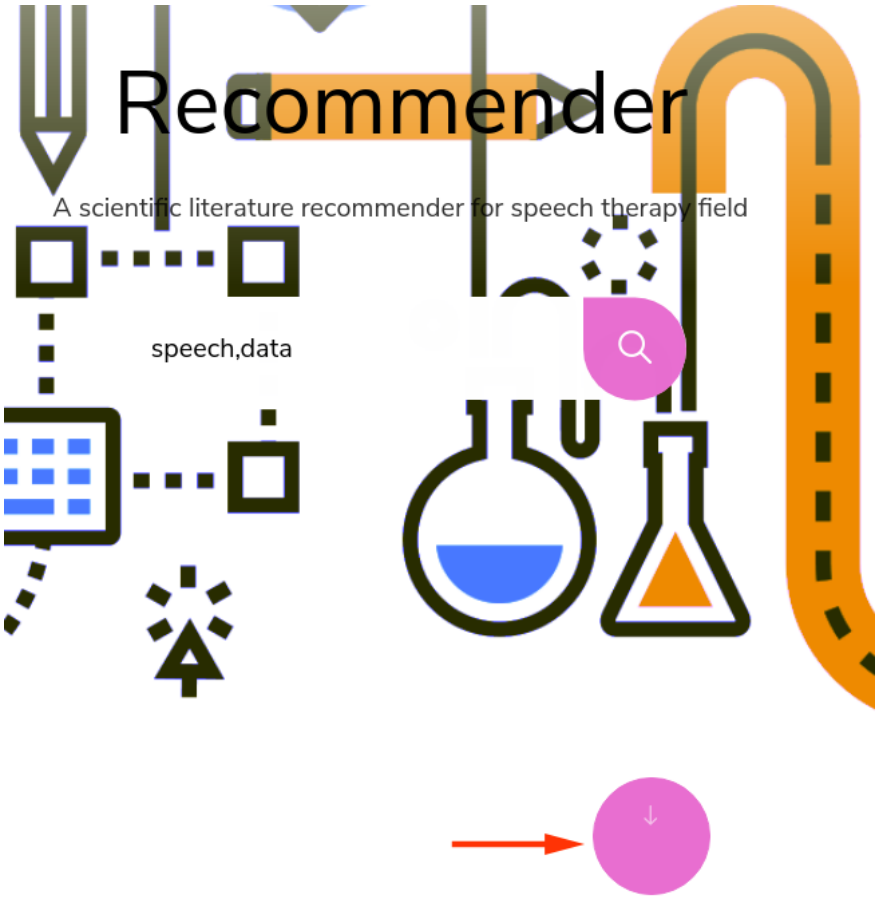


Figura 3. Botón para desplegar resultados

3.2.2.2. Representación de resultados

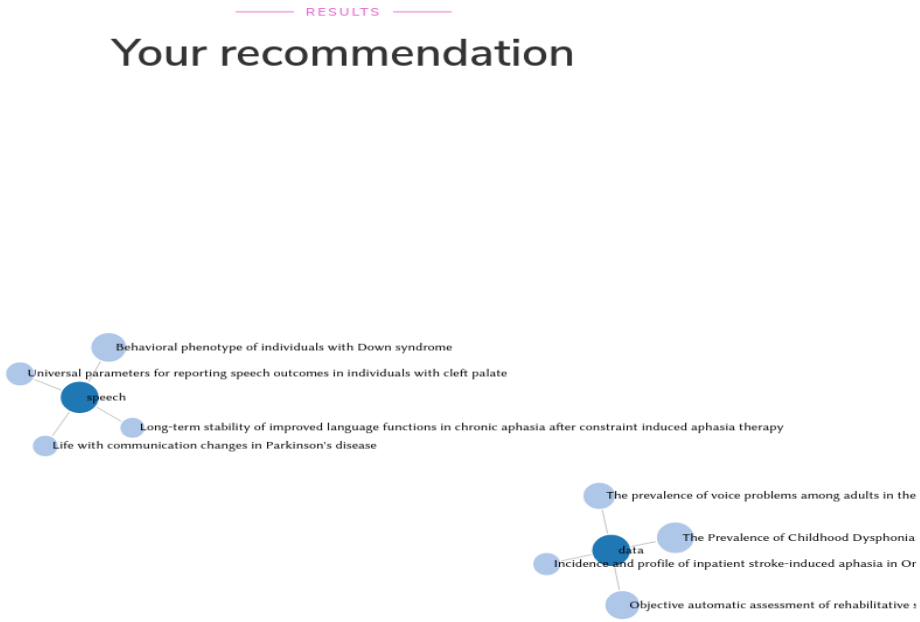


Figura 4. Representación gráfica de resultados

Los nodos mostrados en la Figura 4 se los representa de la siguiente manera:

Por cada palabra se creará un nodo de color azul (nodo de búsqueda)

Cada recomendación (nodo recomendación) que se genere estará enlazada al nodo de búsqueda.

Mientras más grande sea el nodo de recomendación significa que esa recomendación tiene una mayor relación con la palabra de búsqueda

3.2.2.3. Enlaces a artículos de la recomendación

Figura 5. Enlaces a los artículos

Como se puede observar en la Figura 5, se creará un apartado para cada palabra que se ingrese en el campo de búsqueda y se presentarán enlaces a los artículos recomendados.

3.3. Consideraciones / Recomendaciones

El tiempo de búsqueda por cada palabra es alrededor de treinta segundos.

Todas las palabras de búsqueda deberán estar en inglés, debido a que los artículos analizados están en dicho idioma.