ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

DESARROLLO DE SISTEMA BASADO EN MINERÍA DE DATOS PARA LA BÚSQUEDA Y VISUALIZACIÓN DE REDES DE INVESTIGADORES CON FILIACIÓN EN INSTITUCIONES ECUATORIANAS Y SUS AREAS ACADÉMICAS

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

ARIAS TÚQUERES JOSUÉ NICOLÁS

josue.arias@epn.edu.ec

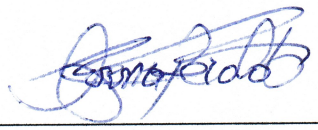
DIRECTOR: PhD. RECALDE CERDA LORENA KATHERINE

lorena.recalde@epn.edu.ec

Quito, junio, 2023

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Josué Nicolás Arias Túqueres, bajo mi supervisión.



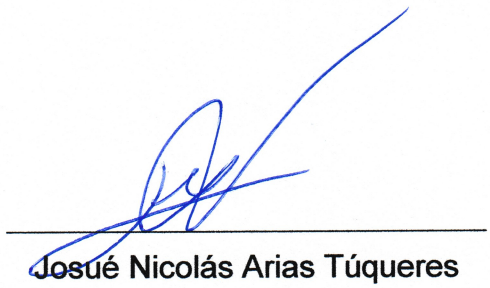
PhD. Lorena Katherine
Recalde Cerda

DIRECTOR DE PROYECTO

DECLARACIÓN

Yo Josué Nicolás Arias Túqueres, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



Josué Nicolás Arias Túqueres

DEDICATORIA

Este proyecto es dedicado a mis queridos padres, cuya dedicación y sacrificio han sido mi mayor fuente de motivación para superar obstáculos. Les brindo este logro con profunda gratitud y amor.

A mis queridos hermanos, les dedico este logro con cariño, y deseo continuar construyendo recuerdos y compartiendo sueños juntos.

AGRADECIMIENTO

En primer lugar, quiero expresar mi profundo agradecimiento a mis padres, quienes han sido mi mayor inspiración y apoyo a lo largo de mi carrera académica. Agradezco todo el sacrificio y esfuerzo que hicieron para poder brindarme una educación de calidad. Su amor y aliento son el motor que me impulsa a alcanzar nuevas metas y superar cualquier desafío.

También me gustaría agradecer de manera especial a PHD. Lorena Recalde, quien ha sido la guía invaluable durante todo el proceso de esta tesis. Su experiencia, conocimiento y paciencia han sido fundamentales alcanzar los objetivos establecidos.

ÍNDICE DE CONTENIDO

INDICE DE TABLAS

INDICE DE FIGURAS

RESUMEN

La calidad de las investigaciones científicas se basa en la colaboración entre expertos. En Ecuador, se ha observado un aumento en la producción científica, lo cual ha generado la necesidad de una herramienta informática eficiente para formar redes, buscar colaboradores y analizar temas de investigación. En respuesta a esto, se ha desarrollado una aplicación web que utiliza modelos de minería de datos para la búsqueda y visualización de redes de investigadores afiliados a instituciones ecuatorianas y sus áreas académicas.

El proyecto se implementó siguiendo las metodologías Scrum y CRISP-DM. Para llevar a cabo todo el proceso de minería de datos se desarrolló una serie de scripts en Python. Los datos fueron extraídos de Scopus a través del conjunto de APIs que ofrece Elsevier. Los modelos generados fueron TF-IDF, los cuales se emplearon para obtener los mejores resultados en las búsquedas realizadas en la aplicación.

La arquitectura de la aplicación consta de tres capas: la base datos orientada a grafos Neo4j; el backend, desarrollado con Flask que contiene la lógica del negocio; y el frontend, que fue desarrollado en Angular, haciendo uso de algunas librerías importantes como Bootstrap para el diseño y D3.js para la creación de gráficos dinámicos e interactivos. Finalmente, como producto final la aplicación ofrece tres tipos de búsqueda: búsqueda de autor, búsqueda de autores relevantes por tópico y búsqueda de artículos relevantes por tópico.

ABSTRACT

The quality of scientific research is based on collaboration among experts. In Ecuador, an increase in scientific production has been observed, which has generated the need for an efficient computer tool to form networks, search for collaborators, and analyze research topics. In response to this, a web application has been developed that utilizes data mining models for searching and visualizing networks of researchers affiliated with Ecuadorian institutions and their academic areas.

The project was implemented following Scrum and CRISP-DM methodologies. To carry out the entire data mining process, a series of Python scripts were developed. The data was extracted from Scopus through the set of APIs offered by Elsevier. The generated models were TF-IDF, which were employed to obtain the best results in the searches performed within the application.

The architecture of the application consists of three layers: the Neo4j graph-oriented database as the data layer, Flask as the backend that houses the business logic, and Angular as the frontend, utilizing important libraries such as Bootstrap for design and D3.js for creating dynamic and interactive graphs. Ultimately, as the final product, the application offers three types of search: author search, relevant authors search by topic, and relevant articles search by topic.

CAPÍTULO 1. INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

La investigación científica y de calidad se fundamenta en la colaboración de expertos en cierta área académica [3]. De hecho, el financiamiento de los gobiernos a proyectos de investigación se otorga cuando son i) multidisciplinarios, ii) inter-universitarios o iii) propuestos por un grupo de investigación formalmente establecido [13]. Entonces, de manera general, se puede deducir que los proyectos de investigación que tienen éxito son llevados a cabo por redes de investigadores cuyo fin es el trabajo colaborativo y por ende tienden a ser más citados [1]. Una muestra de esto son los artículos científicos que hacen públicos los resultados de los trabajos de investigación y muestran la formación de comunidades de investigadores a manera de coautoría. Dicha producción científica y las redes de coautoría han incrementado en todo en el mundo. Brasil es el eje central de las redes de investigación en América Latina y en los últimos cinco años ha reforzado sus colaboraciones con Argentina, México y Chile [1].

En Ecuador sucede algo similar, que de ocupar el 10° lugar en producción cientifica en América Latina en 1998 con 148 artículos, para el 2017 pasó a ocupar el 6° lugar con 3172 artículos [5]. Además, en el 2019 la Secretaría de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT) financió 17 proyectos de investigación científica, donde participaron 427 investigadores en propuestas inter-disciplinarias y/o inter-universitarias [6]. En consecuencia, el propósito de las colaboraciones es realizar un trabajo de investigación enfocado en diferentes áreas de conocimiento [3]. Los investigadores de las universidades de Ecuador emplean diferentes medios para formar parte de dichas comunidades; por ejemplo, grupos de contacto en Facebook y otras redes sociales, búsqueda de información oficial

(nombres de proyectos e investigadores) a través de CEDIA[1], y creación de contactos en conferencias académicas nacionales, etc. Todas estas estrategias son "manuales" porque Ecuador no cuenta con una herramienta informática que permita, de forma eficiente, mostrar los datos académicos de investigadores para formar redes, buscar colaboradores de proyectos que sean expertos en un área determinada, evaluar coautorías, y analizar temas de investigación, sus tendencias y tópicos relacionados; lo cual, en la práctica, se ha vuelto un desafío [2].

El presente proyecto propone el desarrollo de un sistema Web que implemente un modelo basado en la extracción y presentación de información de las redes de investigadores cuya afiliación está en Ecuador. Este modelo tiene como propósito facilitar la comprensión de las tendencias de las investigaciones científicas (*i.e.* relaciones y ocurrencias entre keywords), ayudar a los investigadores(as) en la búsqueda de expertos y sus áreas de investigación y descubrir patrones entre artículos científicos.

1.2. JUSTIFICACIÓN TEÓRICA

Para la creación de un modelo que permita el análisis de patrones de coautoría en redes de investigadores ecuatorianos, primero se deberá recopilar datos académicos de una o varias bases de datos, como por ejemplo: DBPL, APS, MAG, ORC, Scopus, ScienceDirect, etc. [2]. Dependiendo de los conjuntos de datos académicos, estos pueden incluir varias entidades como autor, artículo, lugar, institución, evento y campo de estudio [2].

[1] https://redi.cedia.edu.ec/

Si bien el autor es el dato académico más significativo para la construcción de redes de coautoría, no es el único que se debe tomar en cuenta. Las citas, co-citas y keywords son indicadores que también pueden ser utilizados en las redes de coautoría [2]. Para hallar las keywords en los artículos científicos se utilizará el método de minería de texto TF-IDF [9]. Con este método se pretende determinar las palabras más frecuentes o relevantes dentro de un conjunto de artículos científicos para un autor dado. Esto resulta favorable ya que permitirá asociar una palabra dentro de una consulta con artículos científicos, y por ende con autores [10].

La visualización de datos es primordial para el análisis ya que transforma los datos en múltiples formas de entendimiento [2]. Incluso, la visualización es una de las siete capas de análisis de redes sociales propuesta por Gohar Khan. Esta capa es particularmente útil con datos grandes y complejos porque puede revelar patrones, relaciones y tendencias ocultas [11].

Representar las redes de coautoría como grafos permitirá analizar patrones de coautoría con mayor facilidad [2]. Algunos de los patrones más importantes que son susceptibles a un análisis son: *i)* la asortatividad, el cual es un coeficiente de correlación para el número de colaboradores que tienen los coautores; *ii)* el componente más grande, el cual es un conjunto de nodos de red conectados mediante coautoría, de modo que se puede llegar a cualquier nodo del conjunto desde cualquier otro atravesando un camino adecuado de colaboradores intermedios; *iii)* distancia entre autores utilizando el algoritmo de búsqueda de amplitud en grafos, puesto que puede ser útil para proporcionar vínculos de conocidos que los científicos podrían usar para establecer contacto con otros científicos [4] y *iv)* la fuerza de la colaboración entre autores [8].

Para la implementación de la funcionalidad de búsqueda de redes de investigadores con afiliación en instituciones ecuatorianas se propone recuperar la información a

través de métodos de *Information Retrieval*, que se puede describir como el proceso de buscar y recuperar una colección de datos (bases de datos, documentos de texto, redes, etc.) dada una cadena o "query" proporcionada por el usuario final [10].

Los métodos y estrategias antes presentados y su uso o aplicación serán la base para modelar a manera de grafos los datos de investigadores y artículos recopilados, de tal manera que sean presentados en un sistema de búsqueda y visualización de estas redes.

1.3. JUSTIFICACIÓN METODOLÓGICA

Scrum y Cross Industry Standard Process for Data Mining (CRISP-DM) son las metodologías que se utilizarán para realizar este proyecto. Scrum es un marco de trabajo que facilita el proceso de desarrollo de software iterativo y es utilizado, comúnmente en entornos basados en desarrollo ágil de software [12]; mientras que CRISP-DM provee una descripción normalizada del ciclo de vida de un proyecto estándar de minería de datos [7]. Scrum puede usarse como contenedor para otras técnicas, metodologías y prácticas [12]. Es por ello por lo que Scrum será la metodología contenedora de CRISP-DM. Combinar estas dos metodologías permitirá realizar entregas de software funcional y útil (Sprints), eliminar los posibles ciclos de modelado interminables, y detectar muy temprano los errores en la comprensión de los requisitos durante el ciclo de vida. Además, al igual que en las ciencias de datos, Scrum se basa en el principio de ejecución basado en lo que se conoce como el empirismo [12].

1.4. JUSTIFICACIÓN PRÁCTICA

La selección de investigadores de diferentes áreas de conocimiento para producir un resultado de investigación es complicada, aún más, cuando no se tiene conocimiento sobre los vínculos y sus trabajos de investigación. El presente proyecto ayudará en la toma de decisiones de los investigadores ecuatorianos al momento de buscar colaboradores en un tema de investigación en particular y formar una red de investigación. Además, apoyará a los investigadores a realizar seguimiento de las tendencias y los últimos avances en los temas de investigación, conocer y obtener información de sus investigadores.

1.5. OBJETIVOS

1.5.1. OBJETIVO GENERAL

Desarrollar un sistema basado en minería de datos para la búsqueda y visualización de redes de investigadores con afiliación en instituciones ecuatorianas y sus áreas académicas.

1.5.2. OBJETIVOS ESPECIFICOS

- Diseñar la arquitectura, el modelo de base de datos e interfaces del sistema.
- Extraer los datos académicos a través de una API.
- Desarrollar un método que permita determinar las palabras más relevantes y modelar los documentos científicos y autores mediante minería de texto.
- Implementar una herramienta en el sistema que permita construir redes de coautoría mediante grafos interactivos.
- Implementar un método de information retrieval para la búsqueda y recuperación de información en el sistema web.
- Probar las funcionalidades del sistema mediante pruebas aceptación.

1.6. ALCANCE

El proyecto se enfoca en el desarrollo de una herramienta web que permitirá la extracción y presentación de datos de las redes de coautoría de investigadores ecuatorianos. La herramienta facilitará la visualización de información académica, la búsqueda de expertos en áreas de investigación específicas y el análisis de temas, keywords y tópicos relacionados. Además, se implementarán técnicas de Information Retrieval para mejorar la precisión y relevancia de los resultados en las búsquedas de información académica. La herramienta se diseñará con el objetivo de proporcionar una experiencia óptima y eficiente en la exploración y análisis de información académica.

1.7. MARCO TEÓRICO

1.7.1. REDES DE COAUTORÍA

La coautoría hace referencia a una colaboración entre dos o más autores, y este tipo de relaciones conforman una red de coautoría [4], como la que se muestra en la Figura 1, en donde los nodos representan a los autores unidos por aristas si son coautores en uno o más artículos. Asimismo, el grosor de las aristas varía dependiendo de la frecuencia o fuerza de colaboración entre dos autores.

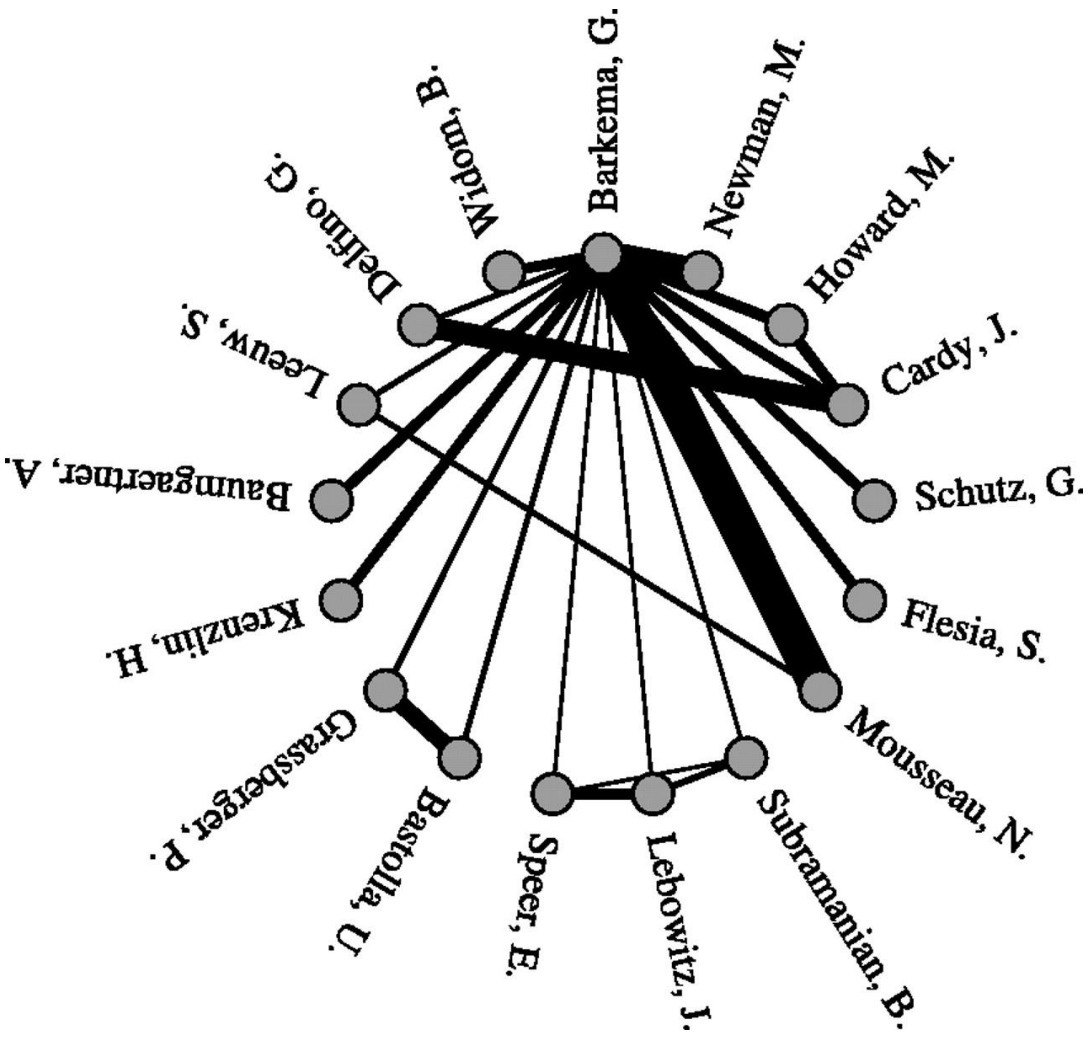


FIGURA 1 Red de coautoría del autor G. Barkema

1.7.2. BASES DE DATOS ORIENTADAS A GRAFOS

Las bases de datos orientadas a grafos son un tipo de base de datos NoSQL y también son una gran alternativa a las bases de datos relacionales. Las redes de telecomunicación, la Web, redes sociales, motores de recomendación y detección de fraude son ejemplos de aplicaciones. Todas estas involucran una gran cantidad de información interconectada [16].

A diferencia de las bases de datos relaciones en donde se utiliza tablas, las bases de datos orientadas a grafos almacenan la información en nodos y aristas. En los nodos se almacenan las entidades y en las artistas las relaciones entre entidades. Además, tanto los nodos como las relaciones tienen propiedades. Las relaciones entre las entidades son tan importantes como las entidades mismas [18].

En la Figura 2, se muestra un ejemplo de un modelo de base de datos orientada a grafos. El modelo está compuesto por las entidades “person” y “car”, y las relaciones “loves”, “lives with”, “drives” y “owns” junto con sus respectivas propiedades [18].

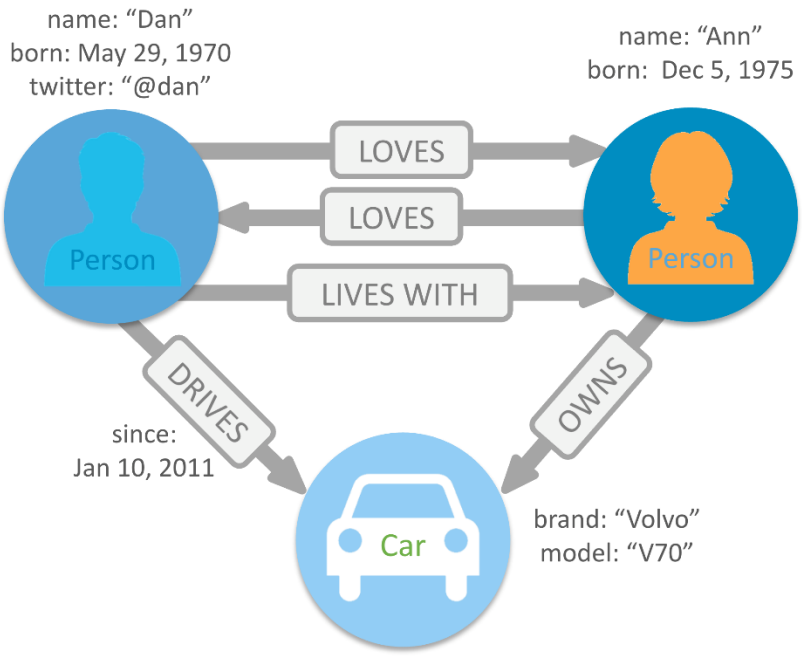


FIGURA 2 Ejemplo de un modelo de base de datos orientada a grafos

Si bien las bases de datos relaciones también pueden almacenar relaciones, estas son navegadas a través de operaciones computacionalmente costosas como JOIN. En cambio, en las bases de datos orientadas a grafos, la navegación de las relaciones es muy rápida porque estas relaciones no se calculan en tiempos de consulta, sino que están almacenadas nativamente en la base de datos [18].

1.7.3. VISUALIZACIÓN DE DATOS ACADÉMICOS

Los datos académicos contienen mucha información como por ejemplo autores, afiliaciones, artículos, citas, etc. Del mismo modo, con el rápido crecimiento de bibliotecas digitales, cómo presentar estos datos de una forma visual para su respectivo análisis se ha vuelto un desafío. Con la visualización de datos se pretende transformar datos en bruto a gráficos fáciles de entender, símbolos, colores, arte, y de igual forma, mejorar la eficiencia del reconocimiento de datos para transmitir información útil [2]. En otras palabras, el objetivo de la visualización es crear una expresión visual en lugar de conceptos o resultados numéricos complejos.

Una de las principales áreas de la visualización de datos es el análisis visual [2]. En términos del presente proyecto, el análisis visual no solo es importante para los científicos o investigadores, sino que también lo es para que los sociólogos analicen las interacciones de los investigadores y la formación de comunidades, para que así, las entidades privadas o gubernamentales evalúen el impacto de los científicos o las afiliaciones y asignen recursos a estos.

1.7.3.1. HERRAMIENTAS DE VISUALIZACIÓN DE DATOS

Actualmente es mucho más fácil tener una comprensión de los datos académicos gracias al desarrollo de las tecnologías de visualización de datos. Estas tecnologías implementan funcionalidades útiles para el procesamiento de datos y análisis de datos, ya sea a través de un lenguaje de programación o utilizando funciones integradas directamente en las herramientas [2]. A continuación, se presenta un listado de herramientas de visualización de datos:

- **Herramientas sin lenguaje de programación:** Tableau**,** ICharts, Infogram, Raw Graphs, Visualize Free, etc.

- **Herramientas basadas en un lenguaje de programación**
    - Javascript: D3.js, Chart.js, FusionCharts, Flot Chart, Zing Chart, etc.
    - Otros lenguajes de programación: Gephi y Processing basados en Java, NodeBox 3 basado en Python, Ggplot2 basado en R, JpGraph basado en PHP, etc.

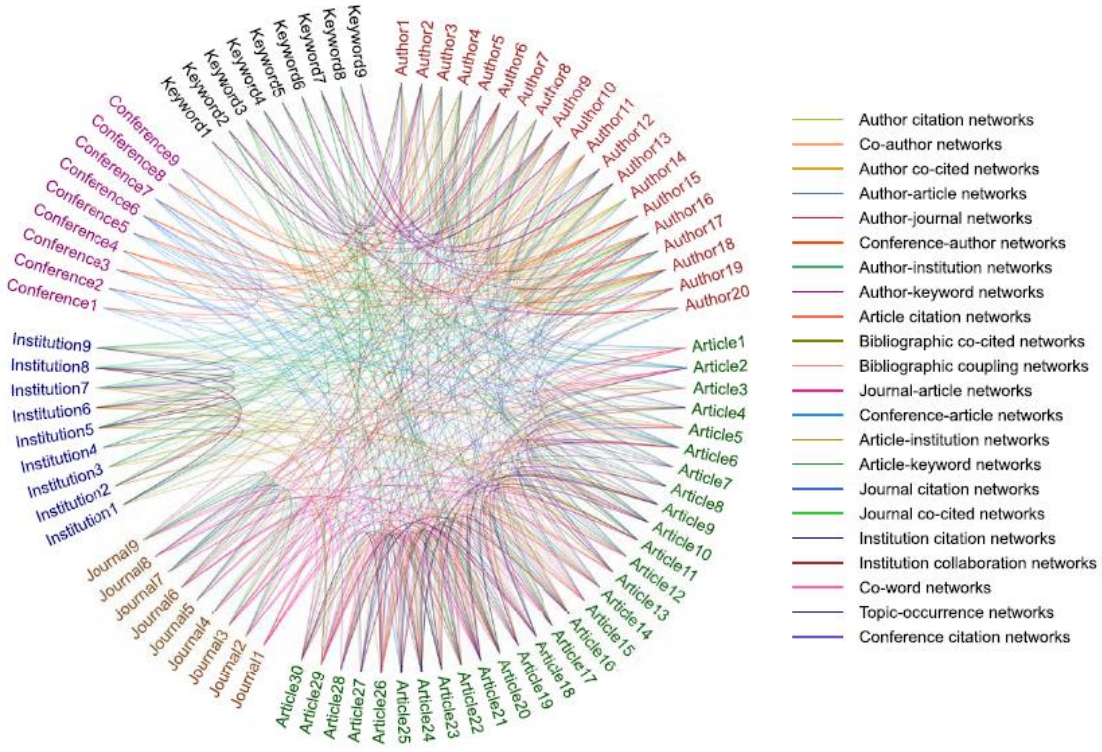


FIGURA 3 Diferentes redes de varias entidades académicas y sus relaciones usando D3.js [2].

1.7.4. TF-IDF

TF-IDF se define como una medida numérica de cuan relevante es una palabra en un corpus para un documento. Sus siglas en inglés se refieren a Term Frequency -

Inverse Document Frequency. La relevancia aumenta proporcionalmente a medida que aumenta la frecuencia de la palabra en un documento (TF), pero se compensa con la frecuencia de la palabra en el corpus (IDF) [10].

- **Term Frequency:** Es el número de veces que aparece una palabra en un documento.

$$\boldsymbol{tf_{i,j}} = \boldsymbol{n_{i,j}}$$

- **Inverse Documente Frequency:** Es el logaritmo del número de documentos dividido por el número de documentos que contiene la palabra.

$$\boldsymbol{idf(w)} = \mathbf{log}(\frac{\boldsymbol{N}}{\boldsymbol{df_T}})$$

- **TF-IDF:** Es simplemente el TF multiplicado por el IDF.

$$w_{i,j} = \boldsymbol{tf_{i,j}} \times \mathbf{log}(\frac{\boldsymbol{N}}{\boldsymbol{df_T}})$$

La normalización de los vectores de los documentos en tf-idf es muy importante, ya que ayuda a evitar sesgos en el tf para documentos de longitud variada. La normalización del coseno es la más común. Esta convierte a todos los vectores de los documentos en vectores unitarios, es decir, con una longitud igual a 1.

La ecuación presentada previamente no es la única forma de calcular el tf-idf. En la Figura 4 se presentan las variaciones más comunes de tf-idf. Un esquema para

calcular el tf-idf muy estándar es utilizar la frecuencia de término ponderada logarítmicamente (l), ponderación idf (t) y la normalización del coseno (c) [31].

Term frequency		Document frequency		Normalization	
n (natural)	$\mathrm{tf}_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1+\log(\mathrm{tf}_{t,d})$	t (idf)	$\log \frac{N}{\mathrm{df}_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2+w_2^2+\ldots+w_M^2}}$
a (augmented)	$0.5+\frac{0.5\times \mathrm{tf}_{t,d}}{\max_t(\mathrm{tf}_{t,d})}$	p (prob idf)	$\max\{0, \log \frac{N-\mathrm{df}_t}{\mathrm{df}_t}\}$	u (pivoted unique)	$1/u$ (Section 6.4.4)
b (boolean)	$\begin{cases}1 & \text{if } \mathrm{tf}_{t,d}>0\\ 0 & \text{otherwise}\end{cases}$			b (byte size)	$1/CharLength^{\alpha}, \alpha<1$
L (log ave)	$\frac{1+\log(\mathrm{tf}_{t,d})}{1+\log(\mathrm{ave}_{t\in d}(\mathrm{tf}_{t,d}))}$				

FIGURA 4 Variantes para tf-idf [31]

1.7.5. HERRAMIENTAS UTILIZADAS

A continuación, en la Tabla 1.1 se especifica los frameworks y librerías utilizadas tanto en el backend y frontend, así como también para el data collector.

Tabla 1.1 Frameworks y librerías

Nombre	Descripción
Node.js	Entorno de tiempo de ejecución de JavaScript multiplataforma y de código abierto [15].
Angular	Plataforma de desarrollo, construida en Typescript, para crear aplicaciones web escalables de una sola página [19].
Bootstrap	Bootstrap es un kit de herramientas de código abierto para desarrollos web responsive con HTML, CSS y JavaScript [20].
D3.js	Biblioteca de JavaScript para producir visualizaciones de datos dinámicas e interactivas [21].

Flask	Framework para Python basado en WSGI que permite crear aplicaciones web de todo tipo rápidamente [22].
Pandas	Librería de manipulación y análisis de datos de código abierto rápida, potente, flexible y fácil de usar, construida sobre el lenguaje de programación Python [23].

CAPÍTULO 2. METODOLOGÍA

Las metodologías que se utilizarán en este proyecto son Scrum y CRISP-DM, cuyas siglas en inglés se refiere a Cross Industry Standard Process for Data Mining. En primer lugar, tenemos a Scrum, el cual, más allá de ser una metodología, es un marco de trabajo liviano y simple que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos [12]. Por otro lado, se encuentra CRISP-DM, el cual presenta una descripción general del ciclo de vida de un proyecto de minería de datos [7].

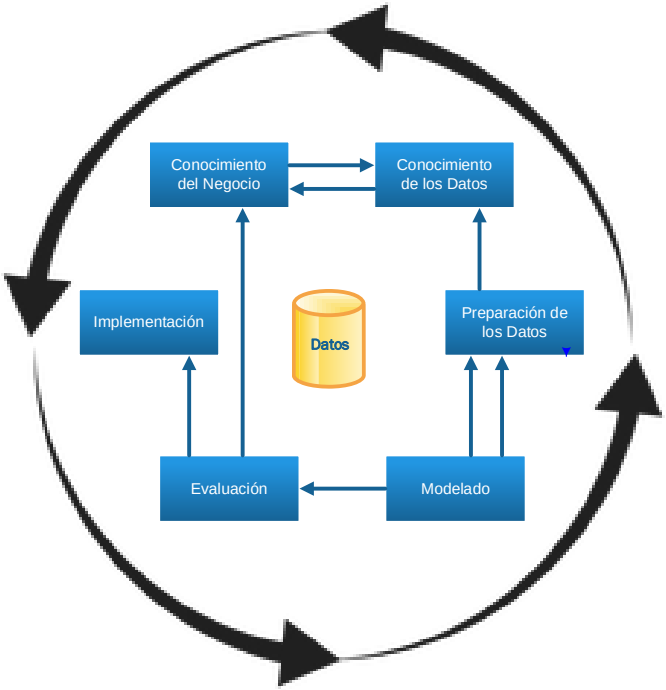


FIGURA 5 Fases de la metodología CRISP DM [7]

Scrum es capaz de envolver varios procesos, técnicas, métodos e incluso metodologías. Por consiguiente, el marco de trabajo Scrum incluirá la metodología CRIPS-DM. Esta última comprende 6 fases (Figura 5), y cada una de estas estará

incluida en diferentes Sprints. En la Tabla 2.1, se especifica cómo se trabajará con estas dos metodologías.

Tabla 2.1 Sprints o iteraciones que incluyen las fases de CRISP-DM

Sprint	Nombre	Descripción
0	Conocimiento del negocio e identificación de los requerimientos del sistema web	En este sprint se encuentra la fase *Conocimiento del negocio* de CRISP-DM, la cual busca entender el objetivo de las metas y los datos con los que se cuenta de un negocio en particular. Entender todo el contexto del negocio servirá como soporte para la definición de los requisitos del sistema web.
1	Conocimiento de los datos para la identificación y extracción de datos académicos	En este sprint se encuentra la fase *Conocimiento de los datos* de CRISP-DM, en donde determinamos qué se espera obtener de los datos recolectados al igual que la calidad de estos. Esta fase de CRISP-DM apoyará a los procesos de identificación de fuentes de datos para su respectiva extracción.
2	Preparación de los datos	En este sprint se realizarán actividades pertenecientes a la fase *Preparación de los datos* de CRISP-DM, la cual busca procesar los datos, realizar limpieza, estandarización, categorizaciones, etc.
3	Modelado de entidades como autor y artículo académico	En este sprint se encuentra la fase Modelado de CRISP-DM, en el cual los resultados deberán satisfacer lo que estamos esperando obtener del estudio de estos datos.
4	Diseño y codificación del sistema web	En este sprint se pretende diseñar las interfaces del sistema web y codificar en base a los requerimientos del sistema.

5	Implementación de una herramienta para construir grafos y recuperar información	En este sprint se pretende utilizar una herramienta para la construcción de redes de coautoría mediante grafos, así como también, la implementación de un método de *Information Retrieval* para la recuperación de información de datos académicos.
6	Realizar pruebas de aceptabilidad, funcionalidad e integración	En este sprint se realizarán pruebas de aceptabilidad, funcionalidad e integración del sistema web.

2.1. ROLES

El Scrum Team es un grupo de personas que representa la unidad fundamental del marco de trabajo Scrum [12]. Este equipo consta de un Product Owner, Scrum Master, y Developers. A continuación, en la Tabla 2.2 se lista las asignaciones de los roles para este proyecto.

Tabla 2.2 Roles del equipo Scrum

Rol	Responsable
Product Owner	Lorena Recalde
Scrum Master	Lorena Recalde
Developers	Josué Arias

2.2. REQUERIMIENTOS

Los requerimientos del sistema fueron obtenidos mediante entrevistas al Product Owner junto con el resto del equipo Scrum. En la Tabla 2.3 se listan los requerimientos levantados.

Tabla 2.3 Requerimientos del sistema

ID	REQUERIMIENTOS	NIVEL
01	Recopilar información de instituciones ecuatorianas.	Muy alta
02	Recopilar información de autores que pertenezcan o hayan pertenecido históricamente a instituciones ecuatorianas.	Muy alta
03	Recopilar información de los documentos de los autores.	Muy alta
04	Seleccionar, limpiar, construir y formatear los datos recopilados.	Muy alta
05	Almacenar la información recopilada en una base de datos orientada a grafos.	Muy alta
06	Generar un modelo para obtener las palabras más relevantes o k*eywords* de los autores.	Muy alta
07	Generar un modelo para obtener las palabras más relevantes o k*eywords* de los artículos.	Muy alta
08	Buscar autores dado su primer nombre y/o apellido.	Alta
09	Visualizar los resultados de búsqueda en una lista.	Alta
10	Mostrar la red de coautoría de un autor en específico.	Muy alta
11	Interactuar con el gráfico de la red de coautoría.	Alta
12	Navegar entre redes de coautoría.	Alta
13	Mostrar un grafo que represente un autor y sus *keyword.*	Media
14	Buscar autores que se encuentren relacionados con una *keyword.*	Media
15	Visualizar los resultados de autores por *keyword* en un grafo.	Alta
16	Buscar artículos (estado del arte) que se encuentren relacionados con una *keyword.*	Media

17	Visualizar los resultados de artículos por *keyword* en un grafo.	Alta

2.3. PRODUCT BACKLOG

El Product Backlog es una lista ordenada de lo que se necesita para desarrollar el producto [12]. La herramienta seleccionada para la gestión de este artefacto es Azure Boards, el cual proporciona un conjunto de funcionalidades para la administración de los procesos Agile, Scrum y Kanban [24]. A continuación, se listan los elementos utilizados en Azure Boards para Scrum.

- **Epic:** Unidad de trabajo que agrupa *features* y trasciende los *release*.
- **Feature:** Unidad de trabajo que agrupa ítems del producto backlog y que no necesariamente pertenece a un único Sprint.
- **Product Backlog Item:** Unidad de trabajo mínima que se entrega en un solo Sprint.

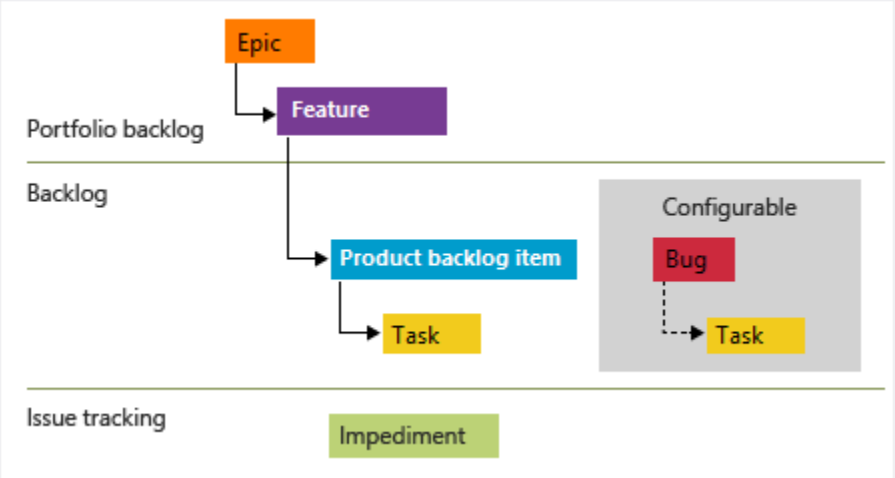


FIGURA 6 Elementos de trabajo de Azure Boards para Scrum [24]

A continuación, se definen las Épicas, Features e Historias de Usuario para el desarrollo del presente proyecto:

- **Epics:** Las fases de la metodología CRISP-DM fueron parametrizadas como épicas. Asimismo, se añadieron épicas que trascienden el trabajo de minería de datos, como la preparación del ambiente de trabajo, el diseño y codificación del sistema web.

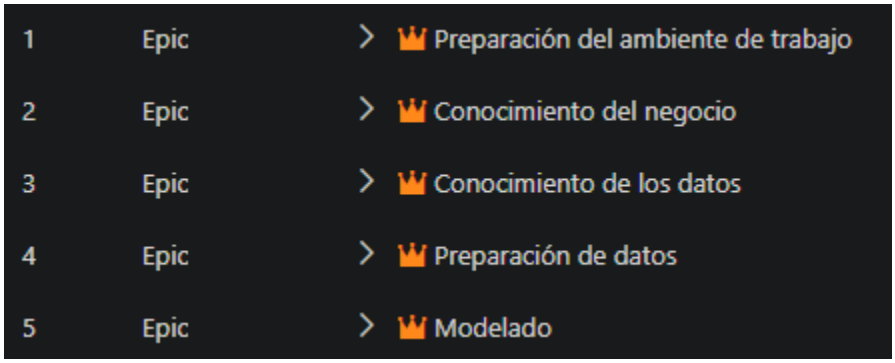


FIGURA 7 Épicas del proyecto en Azure Boards

- **Features:** Cada fase de CRISP-DM contiene tareas genéricas. Éstas fueron parametrizadas como *features.*

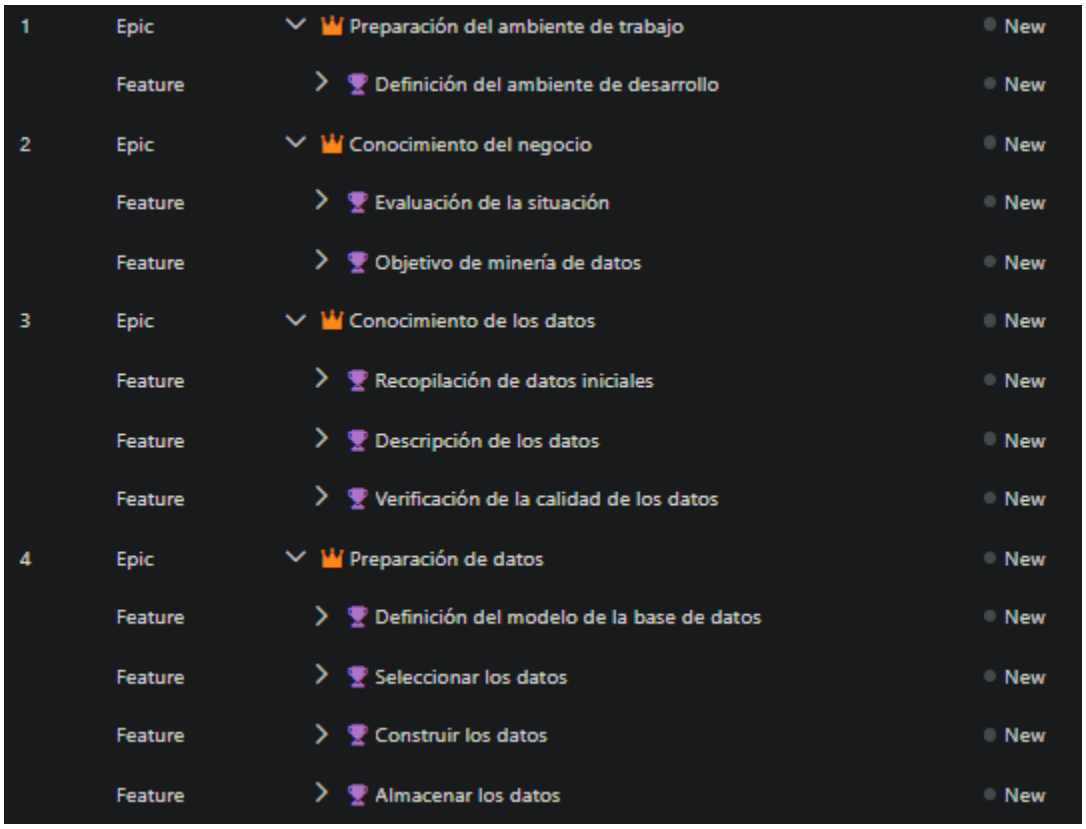


FIGURA 8 Features del proyecto en Azure Boards

- **Product Backlog:** La definición de las historias de usuario se encuentran en la tabla 2.4, la cual consta de las siguientes columnas:
    - **E (ID):** Identificador de épica.
    - **F (ID):** Identificador de feature.
    - **HU (ID):** Identificador de historia de usuario.
    - **Nombre:** Nombre breve y descriptivo de la historia de usuario.
    - **Descripción:** Descripción de la historia de usuario.
    - **Prioridad:** Estimación cuantitativa de la importancia de la historia de usuario.
    - **Esfuerzo:** Estimación cuantitativa del nivel de complejidad de la historia de usuario.

Tabla 2.4 Historias de Usuario

E (ID)	F (ID)	HU (ID)	Nombre	Descripción	Prioridad	Esfuerzo
Preparación del ambiente de trabajo						
E1	E1F1	HU01	Estándares de codificación y documentación	Como equipo Scrum se desea conocer los estándares de codificación y documentación del código con el fin de realizar un desarrollo homogéneo del sistema.	1	3
		HU02	Arquitectura del Sistema	Como equipo Scrum se desea definir la arquitectura final del sistema con la finalidad de construir el sistema de forma organizada.	2	5
Conocimiento del negocio						
E2	E2F1	HU03	Fuentes de datos y conocimientos	Como equipo Scrum se desea listar las fuentes de datos con la finalidad de identificar las fuentes de datos disponibles.	3	3
		HU04	Recursos de hardware y software	Como equipo Scrum se desea listar los recursos de software y hardware con la finalidad de identificar los recursos disponibles.	2	1

		HU05	Restricciones	Como equipo Scrum se desea listar las restricciones de las fuentes de datos con la finalidad de verificar la accesibilidad y las limitaciones de las fuentes de datos.	2	5
	E2F2	HU06	Objetivo de minería de datos	Como equipo Scrum se desea definir el objetivo principal de la minería de datos con la finalidad de identificar el objetivo del proyecto en términos técnicos.	3	1
Conocimiento de los datos						
E3	E3F1	HU07	Planificación de requisitos de datos	Como equipo Scrum se desea planificar qué información se necesita con la finalidad de verificar si la información necesaria se encuentra disponible.	2	2
		HU08	Módulo de peticiones a las APIs	Como equipo Scrum se desea desarrollar un módulo en Python para realizar peticiones con la finalidad de obtener datos de las fuentes de datos (Scopus).	2	8
		HU09	Algoritmo para la extracción de datos	Como equipo Scrum se desea desarrollar un algoritmo para la extracción de datos de las fuentes de datos (Scopus) mediante el uso del módulo de peticiones a las APIs.	4	13

		HU10	Adquisición de datos iniciales	Como equipo Scrum se desea adquirir los datos necesarios iniciales de las diferentes entidades con la finalidad de conseguir una comprensión de los datos.	4	5
	E3F2	HU11	Análisis volumétrico de datos	Como equipo Scrum se desea realizar un análisis del volumen de los datos con la finalidad de determinar el volumen de datos de cada entidad y sus relaciones.	2	3
		HU12	Tipos y valores de atributo	Como equipo Scrum se desea verificar los tipos de los valores de los atributos de los campos con la finalidad de comprender el significado de cada uno.	2	3
	E3F3	HU13	Identificación de valores especiales	Como equipo Scrum se desea identificar valores especiales con la finalidad de catalogar su significado.	2	5
		HU14	Identificación de registros duplicados	Como equipo Scrum se desea identificar registros duplicados con la finalidad de eliminarlos o almacenarlos en base al criterio de duplicación.	2	3
		HU15	Análisis de artículos sin relación	Como equipo Scrum se desea identificar los artículos que fueron extraídos pero que no	3	3

				tienen ninguna relación con autores o afiliaciones ecuatorianas.		
Preparación de los datos						
E4	E4F1	HU16	Diseño de la base de datos	Como equipo Scrum se desea definir el diseño de la base de datos orientada a grafos con la finalidad de almacenar la data.	4	3
	E4F2	HU17	Selección de datos	Como equipo Scrum se desea seleccionar los datos que se utilizarán para el análisis de minería de datos.	4	3
	E4F3	HU18	Construcción de la entidad Afiliación	Como equipo Scrum se desea construir la entidad Afiliación mediante las afiliaciones extraídas de los artículos.	3	3
		HU19	Construcción de la entidad Autor	Como equipo Scrum se desea construir la entidad Autor mediante los autores extraídos de los artículos.	3	3
		HU20	Construcción de la entidad Keyword	Como equipo Scrum se desea construir la entidad Keywords mediante las keywords extraídas de los artículos.	3	3
		HU21	Construcción de las relaciones	Como equipo Scrum se desea construir las relaciones entre las entidades basada en los artículos extraídos.	3	3

	E4F4	HU22	Almacenamiento de los datos en archivos CSV	Como equipo Scrum se desea almacenar los datos en archivos CSV para facilitar la carga de datos en bases de datos relacionales u orientadas a grafos.	3	2
		HU23	Carga de los datos en Neo4j	Como equipo Scrum se desea cargar la data desde archivos CSV a una base de datos en Neo4j.	2	5
Modelado y evaluación						
E5	E5F1	HU24	Seleccionar la técnica de modelado	Como equipo Scrum se desea seleccionar la técnica de modelado apropiada para el objetivo de minería de datos.	2	3
		HU25	Generar diseño de prueba	Como equipo Scrum se desea generar el diseño de pruebas para comprobar la validez y calidad del modelo.	2	5
		HU26	Construir el modelo	Como equipo Scrum se desea ejecutar la herramienta de modelado en el conjunto de datos preparado para crear el modelo.	2	5
	E5F2	HU27	Evaluar el modelo	Como equipo Scrum se desea evaluar el modelo para asegurarse de que cumple con los criterios de éxito de la minería de datos.	2	5
Diseño y codificación						

E6	E6F1	HU28	Diseñar la interfaz de búsqueda	Como equipo Scrum se desea diseñar la interfaz principal del sistema para realizar búsqueda de autores, keywords y/o artículos.	4	3
		HU29	Diseñar la interfaz de resultados de búsqueda por autor	Como equipo Scrum se desea diseñar la interfaz de resultados de búsqueda por autor para poder visualizar la información del autor.	4	5
		HU30	Diseñar la interfaz de resultados de búsqueda de autores por keywords	Como equipo Scrum se desea diseñar la interfaz de resultados de búsqueda de autores por keywords para poder visualizar las interacciones entre autores relacionados con una keyword.	4	3
		HU31	Diseñar la interfaz de resultados de búsqueda de artículos por keywords	Como equipo Scrum se desea diseñar la interfaz de resultados de búsqueda de artículos por keyword para poder visualizar el estado del arte.	4	3
	E6F2	HU32	Codificar la interfaz de búsqueda	Como Usuario deseo realizar los tres tipos de búsqueda en una misma interfaz.	4	5

		HU33	Implementar una capa de abstracción de D3 en Angular	Como equipo Scrum se desea implementar una capa de abstracción de D3 en Angular para facilitar la construcción de grafos.	4	8
		HU34	Codificar la interfaz de resultados de búsqueda de autor	Como Usuario deseo realizar la búsqueda de autor para visualizar la información académica de un autor.	3	5
		HU35	Codificar la interfaz de resultados de la búsqueda de autores relevantes por keyword	Como Usuario deseo realizar la búsqueda de autores relevantes para visualizar los autores más relevantes y sus relaciones en base a una keyword.	3	5
		HU36	Codificar la interfaz de resultados de la búsqueda de artículos relevantes por keyword	Como Usuario deseo realizar la búsqueda de artículos relevantes para visualizar los artículos más relevantes en base a una keyword.	3	5

2.4. PLANIFICACIÓN DE LOS SPRINT

En la planificación que se lleva a cabo para el proyecto, se establece el trabajo que se realizará durante cada uno de los Sprints. Este proceso es llevado a cabo por todo el equipo Scrum. En la tabla que se presenta a continuación, se detallan las historias de usuario que se abordarán en cada sprint, indicando su duración y el esfuerzo total requerido. Cabe destacar que, siguiendo la sugerencia de la Guía de Scrum [12], cada Sprint tiene una duración de un mes o menos para mantener una consistencia en el proceso.

Tabla 2.5 Planificación de los Sprint

SPRINT	DURACIÓN	HISTORIAS DE USUARIO	ESFUERZO TOTAL
0	13 días	HU01, HU02, HU03, HU04, HU05, HU06	18
1	23 días	HU07, HU08, HU09, HU10, HU11, HU12, HU13, HU14, HU15	45
2	13 días	HU16, HU17, HU18, HU19, HU20, HU21, HU22, HU23	25
3	15 días	HU24, HU25, HU26, HU27	18
4	12 días	HU28, HU29, HU30, HU31, HU32	19
5	17 días	HU33, HU34, HU35, HU36	23

2.5. SPRINT 0

2.5.1. OBJETIVOS DEL SPRINT

- Definir de los estándares de codificación y documentación del código.
- Evaluar la situación.

    - Inventario de recursos (recursos de hardware y software, fuentes de datos y conocimiento, fuentes de personal, etc.).
    - Restricciones (restricciones de las fuentes de datos, verificación de los derechos de acceso, etc.).
- Determinar el objetivo de minería de datos.
- Definir de la arquitectura del sistema.

2.5.2. HISTORIAS DE USUARIO DEL SPRINT

En la Figura 9 se muestra el listado de las historias de usuario para el Sprint 0, el cual fue extraído de Azure Boards.

Order	Work Item Type	Title	State	Effort	Value Area	Iteration Path
1	Product Backlo...	Estándares de codificación y documentación	Approved	3	Business	ResNet\Sprint 0
2	Product Backlo...	Arquitectura del Sistema	Approved	5	Architectural	ResNet\Sprint 0
3	Product Backlo...	Fuentes de datos y conocimientos	Approved	3	Business	ResNet\Sprint 0
4	Product Backlo...	Restricciones	Approved	5	Business	ResNet\Sprint 0
5	Product Backlo...	Recursos de hardware y software	Approved	3	Business	ResNet\Sprint 0

FIGURA 9 Historias de usuario del Sprint 0

2.5.3. EJECUCIÓN DEL SPRINT

2.5.3.1. ESTÁNDARES DE CODIFICACIÓN Y DOCUMENTACIÓN DEL CÓDIGO

La calidad, la legibilidad y la facilidad de mantenimiento son aspectos clave en el desarrollo de software. Para lograr estos objetivos, se deben seguir directrices y estándares de codificación consistentes. En este proyecto, se han definido los estándares de codificación que se utilizarán para el framework Angular y el lenguaje

de programación Python. La Tabla 2.6 proporciona una descripción detallada de estos estándares.

Para el código generado en Python, se seguirá el estándar de codificación PEP 8, que es la guía de estilo para Python. Este estándar promueve una convención de codificación coherente en todo el proyecto y se centra en la legibilidad del código. Para el código generado en Angular, se utilizará la guía de estilo de codificación de Angular, que tiene como objetivo mejorar la legibilidad y la claridad del código generado y fomentar prácticas de codificación sostenibles y mantenibles. La adopción de estos estándares de codificación es fundamental para garantizar que el código producido sea fácil de leer, entender y mantener, lo que en última instancia contribuirá al éxito del proyecto y mejorará la calidad del software generado.

Tabla 2.6 Estándares de codificación y documentación de código

LENGUAJE DE PROGRAMACIÓN	GUÍA	ESTÁNDARES
Python	PEP 8 – Style Guide for Python Code	• Diseño de código: sangría, tabulaciones, longitudes máximas de línea. • Cotizaciones de datos de tipo string. • Espacio en blanco en expresiones y declaraciones. • Comentarios: bloque de comentarios, comentarios en una línea. • Convenciones de nombres: principio primordial, estilos de

		nomenclatura descriptivo y prescriptivo.
Typescript - Angular	Angular coding style guide	• Convenciones de estructura de archivos. • Principio de responsabilidad única. • Nomenclatura. • Estructura de la aplicación y NgModules. • Componentes como elementos • Usar directivas para mejorar un elemento. • Los servicios son singletons. • Implementación de las interfaces de los hooks.

2.5.3.2. EVALUACIÓN DE LA SITUACIÓN

Esta tarea consiste en la búsqueda detallada sobre todos los recursos, limitaciones, suposiciones y otros factores, que deben tomarse en cuenta al desarrollar el proyecto.

2.5.3.2.1. RECURSOS DE SOFTWARE Y HARDWARE

La Tabla 2.7 muestra una lista completa de todos los recursos disponibles, tanto de hardware como de software, que se utilizarán para llevar a cabo el proyecto en cuestión. Es importante contar con una descripción detallada y precisa de todos los

recursos disponibles, a fin de asegurar que se disponga de todo lo necesario para llevar a cabo el proyecto de manera efectiva.

Tabla 2.7 Recursos de hardware y software disponibles

Tipo de recurso	Nombre	Descripción
Hardware	Laptop	Inspiron 3505
	Procesador	AMD Ryzen 5 3450U with Radeon Vega Mobile Gfx, 2100 Mhz, 4 procesadores principales, 8 procesadores lógicos
	Memoria RAM	Memoria física instalada (RAM) 16,0 GB Memoria física total 13,9 GB
	Unidad de disco duro	AHCI SATA, 6 Gbps, 1TB
	Unidad de estado sólido	M.2 2280, NVMe PCIe, 500GB
Software	Sistema operativo	Microsoft Windows 10 Home Single Language
	IDEs	Anaconda Navigator, Webstorm, Visual Studio Code
	Base de datos	Neo4j
	Lenguajes de programación	Python, Typescript
	Control de versiones	Git, Github, Gitkraken
	Servicios en la nube	Amazon EC2, Heroku

2.5.3.2.2. FUENTES DE DATOS Y CONOCIMIENTO

En la Tabla 2.8 se enumeran las fuentes de datos, en donde se detalla el nombre, tipo y descripción de las fuentes de datos identificadas para este proyecto.

Tabla 2.8 Fuentes de datos y conocimiento

Nombre	Tipo	Descripción
Scopus	Recurso online	Es una base de datos bibliográfica de resúmenes y citas de artículos de revistas científicas. Contiene desde títulos de publicaciones seriadas como revistas, conferencias, series de libros de investigación, así como también revistas revisadas por pares de las áreas de ciencias, tecnología, medicina, ciencias sociales, artes y humanidades. [25].
ScienceDirect	Recurso online	ScienceDirect es un sitio web que proporciona acceso por suscripción a una gran base de datos de investigación científica, técnica y médica [26].

Diferencias entre Scopus y ScienceDirect

ScienceDirect y Scopus utilizan dos bases de datos diferentes. Mientras que Scopus indexa casi toda la base de datos ScienceDirect pero sin el texto completo de los artículos, ScienceDirect contiene artículos de texto completo de revistas y libros, publicados principalmente por Elsevier [14]. Además, en ScienceDirect no se puede realizar búsquedas con identificadores nativos de Scopus como AF-ID, Scopus ID, etc.

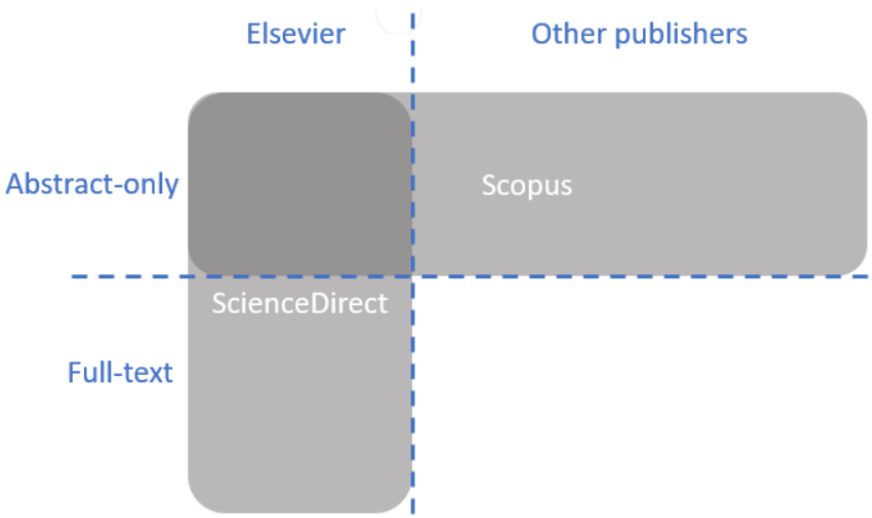


FIGURA 10 Diagrama comparativo entre Scopus y ScienceDirect [14]

2.5.3.2.3. RESTRICCIONES

A continuación, se enumeran las restricciones de las fuentes de datos Scopus y ScienceDirect. En la Tabla 2.9 se detallan las restricciones generales para las fuentes de datos, mientras que en la Tabla 2.10 y Tabla 2.11 se describen las restricciones que tienen las APIs de Scopus y ScienceDirect respectivamente.

Tabla 2.9 Restricciones generales

ID	DESCRIPCIÓN
R01	ScienceDirect contiene el texto completo de artículos de revistas y libros, aunque en su mayoría solo publicados por la editorial Elsevier.
R02	Scopus solo indexa los *abstracts* de artículos de revistas y libros incluyendo a Elsevier y otras editoriales.
R03	Scopus indexa, con algunas excepciones, casi toda la base de datos de ScienceDirect, pero sin el texto completo de los artículos.
R04	No se puede buscar en ScienceDirect utilizando identificadores nativos de Scopus como AF-ID, Scopus ID, etc.

R05	Las API RESTful de Elsevier brindan acceso a contenido de plataformas como ScienceDirect, Scopus e Engineering Village, aunque solo para casos de uso en específico.
R06	Las APIs de Scopus y ScienceDirect admiten varios métodos de autenticación. Para todos los métodos, se pasa una “APIKey” con cada petición.
R07	Autenticación basada en direcciones IP para suscriptores institucionales de Scopus / ScienceDirect: • Este es el valor predeterminado para cualquier APIKey recién registrada. Los clientes que se autentican de esta manera obtienen acceso a todo el contenido asociado con su cuenta institucional.
R08	Autenticación “basada en token”, que incluye: • Uso de tokens de autenticación para resolver conflictos de direcciones IP para suscriptores institucionales de Scopus. • Uso de un token de propiedad (un “Token institucional”) creado para cada usuario por parte del equipo de soporte de integración de Scopus.
R09	Una clave de API tiene recursos, quotas y niveles de servicio específicos de la API habilitados de forma predeterminada.
R10	Las quotas se restablecen cada 7 días.
R11	Los límites de cuota son por API, no una única configuración global para una APIKey determinada.
R12	La API Scopus Search no retorna todos los autores de un artículo si este tiene más de 98 autores.

Tabla 2.10 Restricciones de las APIs de Scopus [14]

ID	Nombre de la API	Vistas	Cuota semanal	Peticiones por segundo

R13	Serial Title	Vistas: STANDARD, COVERIMAGE, ENHANCED. Por defecto 25 resultados / Max 200 resultados	20,000	6
R14	Citations Count Metadata	Vistas: STANDARD. Por defecto 25 resultados / Max 200 resultados.	50,000	10
R15	Subject Classifications	Vistas: STANDARD. Por defecto 25 resultados / Max 200 resultados.	N/A	N/A
R16	Abstract Retrieval	Vistas: todas. Por defecto vista FULL.	10,000	9
R17	Affiliation Retrieval	Vistas: todas. Por defecto vista STANDARD.	5,000	9
R18	Author Retrieval	Vistas: todas. Por defecto vista STANDARD.	5,000	3
R19	Affiliation Search	Por defecto 25 resultados / Max 200 resultados. Límite de resultado de 5000 elementos.	5,000	6
R20	Author Search	Por defecto 25 resultados / Max 200 resultados. Límite de resultado de 5000 elementos.	5,000	2
R21	Scopus Search	Vista STANDARD / Max 200 resultados. Vista COMPLETE / Max 25 resultados. Vista COMPONENT / Max 25 resultados.	20,000	9

		Límite de resultado total de 5000 elementos sin paginación del cursor.		

Tabla 2.11 Restricciones de las APIs de ScienceDirect [14]

ID	Nombre de la API	Vistas	Cuota semanal	Peticiones por segundo
R22	Serial Title	Vistas: STANDARD, COVERIMAGE. Por defecto 25 resultados / Max 200 resultados.	20,000	6
R23	Nonserial Title	Vistas: STANDARD, COVERIMAGE. Por defecto 25 resultados / Max 200 resultados.	20,000	6
R24	Subject Classifications	Sin restricciones.	N/A	N/A
R25	Article Retrieval	Todos los artículos suscritos, OpenAccess y complementarios	50,000. Ilimitado para las API Keys de minería de texto	10
R26	Article Metadata API	Límite de resultados totales de 6000 elementos	N/A	6
R27	ScienceDirect Search v2	Vista STANDARD / Max 200 resultados.	20,000	2

		Límite de resultados totales de 6000 elementos		

2.5.3.3. OBJETIVO DE MINERÍA DE DATOS

El objetivo de la minería de datos en este proyecto es identificar las palabras clave más relevantes o *keywords* para autores ecuatorianos actuales o históricos a partir de sus artículos extraídos de diversas fuentes de datos. Para lograr esto, se analizarán los títulos, abstracts y, si es posible, el texto completo de los artículos de cada autor. El propósito de esta tarea es establecer una relación entre palabras o términos y autores, y el modelo generado se utilizará en el proceso de recuperación de información. De esta manera, se espera facilitar el acceso a la información relacionada con los autores ecuatorianos y mejorar la eficiencia en la búsqueda de documentos relevantes.

2.5.3.4. ARQUITECTURA DEL SISTEMA

La arquitectura principal del sistema ResNet se compone de tres capas fundamentales: una base de datos orientada a grafos, un backend y un frontend. Además, para llevar a cabo el proceso de minería de datos, se desarrollará un conjunto de scripts en Python llamado data collector (Figura 11).

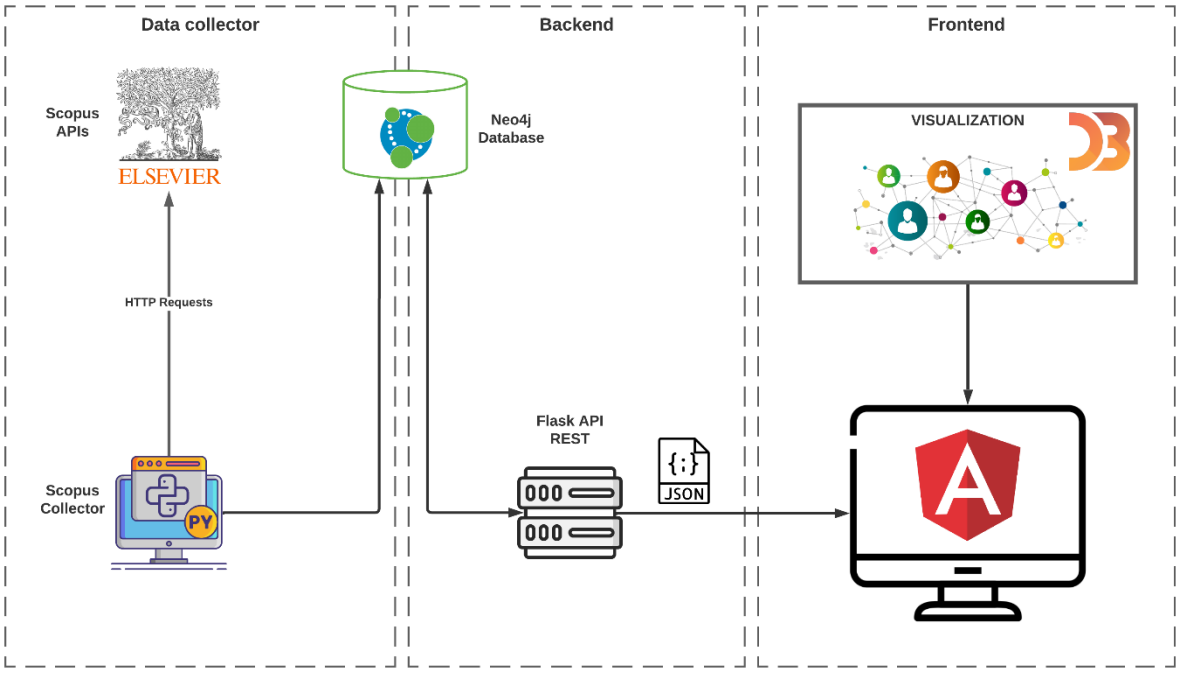


FIGURA 11 Arquitectura del Sistema

El data collector es la parte del sistema que se encarga de la extracción de datos, descripción de datos, verificación de calidad de los datos, selección de datos, limpieza de datos y carga de los datos en la base de datos través de una serie de notebooks desarrollados en Python.

Las fuentes principales para la extracción de datos son las bases de datos bibliográficas Scopus y ScienceDirect, las cuales constan de un conjunto de APIs para la extracción mediante la plataforma para desarrolladores de Elsevier [14].

La base de datos Neo4j es utilizada para almacenar y gestionar los datos de la aplicación. El backend consta de una API REST construida en Flask que proporciona una interfaz de programación para que la aplicación pueda interactuar con la base de datos.

Finalmente, el frontend desarrollado en Angular permite a los usuarios interactuar con la aplicación de forma intuitiva y visualizar los datos almacenados en la base de datos de una manera amigable.

En general, la arquitectura del sistema permite una separación clara de responsabilidades y un alto nivel de modularidad en el diseño de la aplicación. Cada

una de las partes del sistema cumple una función específica y está diseñada para trabajar de manera independiente, lo que facilita la actualización y mantenimiento del sistema a lo largo del tiempo.

2.5.4. REVISIÓN Y RESTROPECTIVA DEL SPRINT

Se realizó la revisión del Sprint 0 en base a los criterios de aceptación de cada una de las historias de usuario. Los objetivos de este sprint culminaron satisfactoriamente.

- Se identificó los estándares de codificación para el framework Angular y el lenguaje de programación Python.
- Se especificó los recursos, fuentes de datos y restricciones con el fin de evaluar la situación.
- Se definió el objetivo de minería de datos.
- Se definió y describió la arquitectura del sistema.

2.6. SPRINT 1

2.6.1. OBJETIVOS DEL SPRINT

- Recopilar datos iniciales
- Describir los datos
- Verificar la calidad de los datos

2.6.2. HISTORIAS DE USUARIO DEL SPRINT

En la Figura 12 se muestra el listado de las historias de usuario para el Sprint 1, el cual fue extraído de Azure Boards.

Order	Work Item Type	Title	State	Effort	Assigned To	Value Area	Iteration Path
1	Product Backlo...	Planificación de requisitos de datos	Approved	2	JOSUE NICOLA...	Business	ResNet\Sprint 1
2	Product Backlo...	Análisis de artículos sin relación	Approved	3	JOSUE NICOLA...	Business	ResNet\Sprint 1
3	Product Backlo...	Identificación de registros duplicados	Approved	3	JOSUE NICOLA...	Business	ResNet\Sprint 1
4	Product Backlo...	Identificación de valores especiales	Approved	5	JOSUE NICOLA...	Business	ResNet\Sprint 1
5	Product Backlo...	Tipos y valores de atributo	Approved	3	JOSUE NICOLA...	Business	ResNet\Sprint 1
6	Product Backlo...	Análisis volumétrico de datos	Approved	3	JOSUE NICOLA...	Business	ResNet\Sprint 1
7	Product Backlo...	Algoritmo para la extracción de datos	Approved	13	JOSUE NICOLA...	Business	ResNet\Sprint 1
8	Product Backlo...	Módulo de peticiones a las APIs	Approved	8	JOSUE NICOLA...	Business	ResNet\Sprint 1
9	Product Backlo...	Adquisición de datos iniciales	Approved	5	JOSUE NICOLA...	Business	ResNet\Sprint 1

FIGURA 12 Historias de usuario del Sprint 1

2.6.3. EJECUCIÓN DEL SPRINT

2.6.3.1. RECOPILACIÓN DE DATOS INICIALES

La recopilación de datos iniciales consiste en la adquisición de los datos (o acceder a los datos) enumerados en los recursos del proyecto.

2.6.3.1.1. PLANIFICACIÓN DE REQUISITOS DE LOS DATOS

Esta tarea consiste en planificar qué información es necesaria, así como también, verificar si toda la información requerida para solucionar los objetivos de minería de datos está disponible [27].

En la Tabla 2.12 se enumeran los campos requeridos y su disponibilidad en las fuentes de datos descritas previamente.

Tabla 2.12 Planificación de requisitos de los datos

Campos	Scopus	ScienceDirect
Artículo		
ID del documento (ISBN, Scopus ID, DOI, etc.)	X	X
Título	X	X
Fecha de publicación	X	X
Autores involucrados	X	X
Afiliaciones involucradas	X	
Abstract	X	X
Texto completo		X
Keywords del autor	X	X
Afiliación		
ID de la afiliación	X	
Nombre	X	
Ciudad	X	
País	X	
Artículos de la afiliación	X	
Autores de la afiliación	X	
Autor		
ID del autor	X	
Nombre	X	
Afiliación actual	X	
Afiliaciones pasadas	X	
Coautores	X	
Artículos del autor	X	

Si bien ScienceDirect tiene disponible el campo “Texto completo”, el cual es uno de los principales campos para el objetivo de minería de datos, esta fuente de datos no dispone de información para las entidades Afiliación y Autor. Por otro lado, la fuente

de datos Scopus tiene a su disposición todos los campos necesarios, a excepción del campo “Texto completo”. Además, Scopus indexa casi toda la base de datos de ScienceDirect [14]. Por lo tanto, Scopus es la única fuente de datos que será utilizada para la adquisición de datos.

2.6.3.1.2. MODELO DE DATOS DE SCOPUS

El modelo de datos de Scopus está diseñado en torno a la noción de que los artículos están escritos por autores afiliados a instituciones [25]. De manera visual y bastante simplista, este modelo relacional se puede representar como muestra la Figura 13:

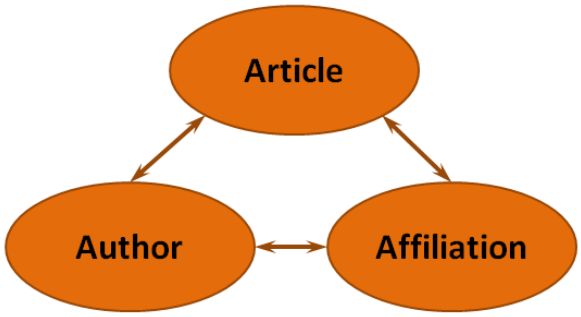


FIGURA 13 Modelo de datos de Scopus

2.6.3.1.3. MÓDULO DE PETICIONES A LAS APIS DE SCOPUS

Este es un módulo desarrollado en Python para consumir datos de las APIs de Scopus. El objetivo del módulo es la automatización del consumo de las APIs de Scopus. Este módulo está basado en el módulo ELSAPY desarrollado por Elsevier. ScopusModule consta de las siguientes clases:

Client: Clase que implementa una interfaz de cliente para api.elsevier.com. El resto de las clases necesitan de un cliente para ejecutar las peticiones a las APIs de Scopus. Para inicializar un cliente se debe especificar una API key y, opcionalmente, insttoken y authtoken.

AffilRetrieval: Clase que representa a la API Affiliation Retrieval. Los parámetros son los siguientes:

- **url:** contiene la URL completa junto con el ID de la afiliación y los parámetros de consulta. Este parámetro se utiliza cuando el resto de los parámetros no se especifican.
- **affil_id:** representa el ID de la afiliación que debe devolverse.
- **view:** representa la lista de elementos que se devolverán en la respuesta.
- **field:** representa el nombre de los campos específicos que deben devolverse.

AuthorRetrieval: Clase que representa a la API Author Retrieval. Los parámetros son los siguientes:

- **url:** contiene la URL completa junto con el ID del autor y los parámetros de consulta. Este parámetro se utiliza cuando el resto de los parámetros no se especifican.
- **author_id:** representa el ID del autor que debe devolverse.
- **view:** representa la lista de elementos que se devolverán en la respuesta.
- **field:** representa el nombre de los campos específicos que deben devolverse.

ArticleRetrieval: Clase que representa a la API Abstract Retrieval. Los parámetros son los siguientes:

- **url:** contiene la URL completa junto con el ID del artículo y los parámetros de consulta. Este parámetro se utiliza cuando el resto de los parámetros no se especifican.
- **scopus_id:** representa el ID del artículo que debe devolverse.
- **view:** representa la lista de elementos que se devolverán en la respuesta.
- **field:** representa el nombre de los campos específicos que deben devolverse.

Search: Clase que representa a las APIs de búsqueda: Affiliation Search, Author Search, Scopus Search. Los parámetros son los siguientes:

- **url:** contiene la URL completa junto con el tipo de búsqueda y los parámetros de consulta. Este parámetro se utiliza cuando el resto de los parámetros no se especifican.
- **searchType:** representa el tipo de búsqueda (artículo, afiliación o autor).
- **query:** representa la búsqueda booleana que se ejecutará en el clúster de artículo, afiliación o autor.
- **facets:** representa el navegador que debe incluirse en los resultados de búsqueda.
- **view:** representa la lista de elementos que se devolverán en la respuesta.
- **field:** representa el nombre de los campos específicos que deben devolverse.
- **count:** número máximo de resultados que se devolverán por petición.

2.6.3.1.4. ADQUISICIÓN DE DATOS

La configuración y extracción de los datos fue realizada con el módulo ScopusModule (Figura 14)**.**

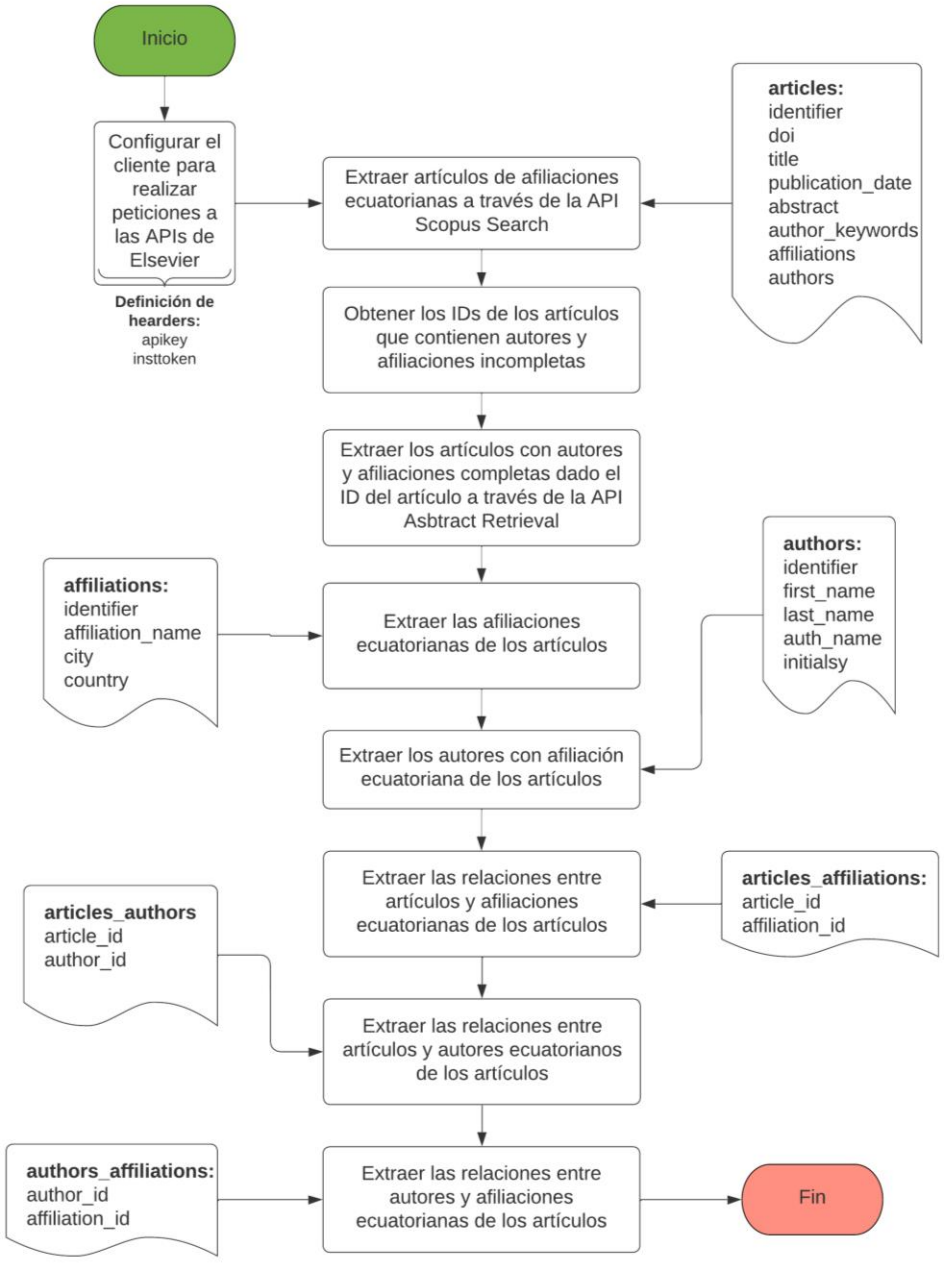


FIGURA 14 Diagrama de flujo de la extracción de datos

1. Configuración del cliente

Antes de realizar peticiones a las APIs de Scopus es necesario configurar un cliente. Para esta tarea se utiliza la clase Client de ScopusModule.

Debido a las restricciones y quotas que tienen las APIs para los suscriptores generales, se solicitó un Token Institucional para poder extraer todos los datos requeridos.

Se ha configurado el cliente con la apikey generada automáticamente en la plataforma para desarrolladores de Elsevier y el insttoken proporcionado por el equipo de soporte de integración de Elsevier junto con la ayuda del equipo de soporte de Bibliotecas de la EPN.

2. Extracción de artículos

La extracción de artículos está dividida en tres partes: extracción de artículos mediante la API Scopus Search, búsqueda de artículos extraídos con autores incompletos y extracción de artículos con autores completos a través de la API Abstract Retrieval.

Con la API Scopus Search se extrajo todos los artículos cuya afiliación o afiliaciones pertenezcan al país Ecuador. Esta especificación se definió a través de la query de búsqueda. Se realizaron 1623 peticiones y en total se extrajeron 40556 artículos.

Tabla 2.13 Parámetros de configuración para la API Scopus Search

Parámetro	Valor
View	COMPLETE
Field	• dc:identifier • doi • dc:title • coverDate • dc:description

	• authkeywords • afid • affilname • affiliation-city • affiliation-country • authid • authname • given-name • surname • initials
Count	25
Query	AFFIL(AFFILCOUNTRY(Ecuador))

Teniendo en cuenta la restricción R12, se determinó qué artículos no fueron extraídos con todos los autores y se almacenó todos los IDs de estos artículos. En total se hallaron 1232 artículos con autores incompletos.

Con la API Abstract Retrieval se recuperó todos los artículos que contenían información faltante a través de los IDs de los artículos que fueron obtenidos en el paso anterior. Dada la naturaleza de la API, en total se realizaron 1232 peticiones, un artículo por petición. Finalmente, el contenido de estos artículos con autores completos fue reemplazado en los artículos que contenían autores faltantes.

Tabla 2.14 Parámetros de configuración para la API Abstract Retrieval

Parámetro	Valor
View	FULL

3. Extracción de afiliaciones

Las afiliaciones fueron obtenidas de los artículos que se extrajeron previamente. En total se encontraron **5372 afiliaciones** ecuatorianas.

4. **Extracción de autores**

Al igual que las afiliaciones, los autores fueron extraídos de los artículos. En total se encontraron **39225 autores** con afiliación ecuatoriana.

2.6.3.2. DESCRIPCIÓN DE LOS DATOS

La descripción de datos consiste en describir detalladamente los datos que se han adquirido, incluyendo el formato de los datos, la cantidad de datos, las identidades de los campos y cualquier otra característica que se haya descubierto [27].

2.6.3.2.1. RESULTADOS DE LA EXTRACCIÓN POR API

- **Scopus Search**
  - Artículos extraídos: 40556
  - Artículos con autores incompletos: 1232
  - Número de peticiones: 1623
  - Duración: 4471.327126026154 segundos.
  - Fecha de inicio de la extracción: Wed Jul 13 20:30:37 2022
  - Fecha de finalización de extracción: Wed Jul 13 21:45:08 2022

- **Abstract Retrieval**
  - Artículos extraídos: 1232
  - Número de peticiones: 1232
  - Duración: 6514.654576063156 segundos.
  - Fecha de inicio de la extracción: Wed Jul 13 21:45:21 2022

- Fecha de finalización de extracción: Wed Jul 13 23:33:56 2022

2.6.3.2.2. ANÁLISIS VOLUMÉTRICO DE LOS DATOS

Todos los artículos extraídos tienen al menos una relación con un autor o afiliación ecuatoriana, pero no todos los autores y afiliaciones obtenidas tiene relación con Ecuador. Esto se debe a que los artículos no fueron desarrollados solo por autores o afiliaciones ecuatorianas. De hecho, solo 15163 (37.38%) artículos contienen únicamente autores o afiliaciones ecuatorianas. En la Tabla 2.15 que se muestra a continuación se puede apreciar el volumen de datos de cada entidad.

Tabla 2.15 Volumen de los datos

Entidad	Relacionado con Ecuador	Otros	Total
Artículos	40556	-	40556
Afiliaciones	5372	23542	28914
Autores	39225	87746	126971
Relación artículos-afiliaciones	55302	287732	343034
Relación artículos-autores	102592	1766748	1869340
Relación autores-afiliaciones	48362	134204	182566

2.6.3.2.3. TIPOS DE ATRIBUTOS Y VALORES

Todos los campos de la entidad artículo que fueron obtenidos son relevantes, a excepción del campo “prism:url”. Este campo, que representa la URI del artículo en la API Abstract Retrieval, no tiene importancia en este estudio debido a que no proporciona información propia del artículo.

El identificador de scopus del artículo contiene únicamente valores numéricos, pero en la Tabla 2.16 se especifica que es de tipo string, ya que los valores de este campo tienen la nomenclatura “SCOPUS_ID:” seguido del identificador, por ejemplo: SCOPUS_ID:85133492759.

El campo “authkeywords” es realmente una lista de strings, sin embargo, esta lista fue retornada como un único string en donde cada keyword está separada por medio del símbolo “|”. Por ejemplo: Audio signals design process | Experimental design processes | Fictional spaces | Sound and changing forms.

Los campos “author” y “affiliation” son posiblemente los campos más importantes del artículo, ya que con estos se obtiene el resto de las entidades y también las relaciones entre sí. En promedio, los artículos tienen 46.092810 autores y 8.458280 afiliaciones. El artículo más extenso cuenta con la participación de 5246 autores y pertenece al CERN. Mientras que el artículo con el mayor número de afiliaciones cuanta con un total de 846, el cual es una investigación desarrollada por el GBD (Global Burden of Disease Study).

Tabla 2.16 Tipos de valores y atributos de la entidad artículo

Campo	Campo renombrado	Tipo	Descripción
prism:url	url	String	URI del artículo en la API Abstract Retrieval

dc:identifier	identifier	String	Identificador de Scopus
dc:title	title	String	Título del artículo
prism:coverDate	publication_date	Date	Fecha de publicación (YYYY-MM-DD)
prism:doi	doi	String	Identificador único y permanente para las publicaciones electrónicas.
dc:description	abstract	String	Abstract del artículo
authkeywords	author_keywords	String	Palabras clave
author	authors	Object list	Autores del artículo
affiliation	affiliations	Object list	Afiliaciones del artículo

Tabla 2.17 Tipos de valores y atributos de la entidad afiliación

Campo	Campo renombrado	Tipo	Descripción
afid	identifier	Number	Identificador de la afiliación
affilname	affiliation_name	String	Nombre
affiliation-city	city	String	Ciudad
affiliation-country	country	String	País

El campo “afid” de la entidad autor es el que permite generar la relación entre autor y afiliación. Este campo es una lista de números ya que un autor pudo haber escrito un artículo mientras se hallaba en dos o más afiliaciones. Por ejemplo: el autor Gerson Ferrari (57208326105) desarrolló el artículo con ID 85133455001 mientras estaba registrado en las afiliaciones 60023383 y 60009462.

Tabla 2.18 Tipos de valores y atributos de la entidad autor

Campo	Campo renombrado	Tipo	Descripción
authid	identifier	Number	Identificador del autor
authname	auth_name	String	Nombre del autor
surname	last_name	String	Apellido
given-name	first_name	String	Nombre de pila
initials	initials		Iniciales
afid	-	Numbers List	Lista de identificadores de las afiliaciones del autor

2.6.3.3. VERIFICACIÓN DE LA CALIDAD DE LOS DATOS

Esta tarea consiste en examinar la calidad de los datos, abordando preguntas como [27]:

- ¿Están completos los datos?
- ¿Es correcto o contiene errores?
- Si hay errores, ¿qué tan comunes son?
- ¿Faltan valores en los datos? Si es así, ¿cómo se representan, ¿dónde ocurren y qué tan comunes son?

2.6.3.3.1. DISCORDIA ENTRE LA CANTIDAD DE LAS AFILIACIONES DE LOS ARTÍCULOS Y LA CANTIDAD DE LAS AFILIACIONES DE LOS AUTORES OBTENIDAS DE LOS ARTÍCULOS

Existen 171 casos donde la cantidad afiliaciones de un artículo no coincide con las afiliaciones de los autores del mismo artículo (campo afid de la entidad autor). Hay dos tipos de casos:

1. Mayor número de afiliaciones de autores del artículo que afiliaciones del artículo.
2. Mayor número de afiliaciones del artículo que afiliaciones de autores del artículo.

A continuación, se presenta un ejemplo del primer caso:

La Figura 15 contiene los autores del artículo con ID 85085698126 y la Figura 16 contiene las afiliaciones del mismo artículo. Aquí existe una discordia, puesto que los autores tienen dos afiliaciones definidas mediante el campo afid (Figura 15). Sin embargo, dentro de la lista de afiliaciones del artículo solo se halla una afiliación (Figura 16). La afiliación faltante es la afiliación con id “114512399”.


```
[
  {'authid': '24484540100',
   'authname': 'Merino P.',
   'surname': 'Merino',
   'given-name': 'Pedro',
   'initials': 'P.',
   'afid': ['114512399', '60072054']},
  {'authid': '57217009320',
   'authname': 'Nenjer A.',
   'surname': 'Nenjer',
   'given-name': 'Alexander',
   'initials': 'A.',
   'afid': ['114512399', '60072054']}
]
```


FIGURA 15 Autores del artículo con ID 85085698126


```
[
  {'afid': '60072054',
   'affilname': 'Escuela Politécnica Nacional',
   'affiliation-city': 'Quito',
   'affiliation-country': 'Ecuador'}
]
```


FIGURA 16 Afiliaciones del artículo con ID 85085698126

Este problema nace debido a que la afiliación no existe. Se realizó una búsqueda en Scopus de esta afiliación basado en el ID, pero no se obtuvo ningún resultado (Figura 17). De igual forma, se buscó dentro del perfil del autor en Scopus para

verificar su historial de afiliaciones, pero, tal y como se muestra en la Figura 18, el autor tiene una afiliación cuyo ID no fue encontrado. Este tipo de caso se encuentra en un total de 6 artículos.

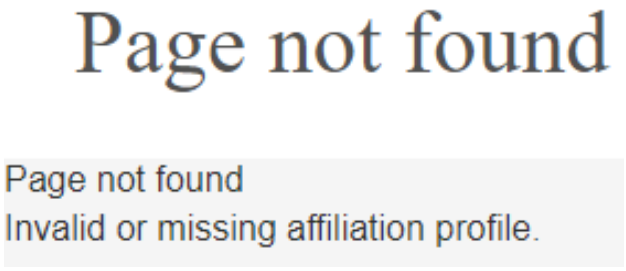


FIGURA 17 Resultado de la afiliación 114512399 en Scopus

FIGURA 18 Afiliaciones del autor Nenjer, Alexander en Scopus

El caso número 2 se encuentra en 165 artículos y, a diferencia del otro caso, las afiliaciones faltantes si existen. Todos estos artículos contienen más de 200 afiliaciones y también una gran cantidad de autores. A continuación, se presenta un ejemplo:

```
ID: 85064570085
Título: Search for nonresonant Higgs boson pair production in the b b ¯ b b ¯ final state at √s = 13 TeV
Número de autores: 2291
Número de afiliaciones de autores del artículo: 247
Número de afiliaciones del artículo: 248
Afiliaciones faltantes: 60015986
```

FIGURA 19 Artículo con ID 85064570085

Tal y como se muestra en la Figura 19, hay más afiliaciones del artículo que afiliaciones de los autores del artículo. En este ejemplo, la afiliación con ID 60015986 es la faltante. Esta afiliación es la faltante también en otros 159 artículos. Aunque parece que esto un problema, en realidad no lo es, ya que esta afiliación no pertenece a Ecuador (Figura 20) y, por lo tanto, no afecta al propósito de este proyecto.

FIGURA 20 Afiliación Università di Trento

2.6.3.3.2. IDENTIFICACIÓN DE ATRIBUTOS FALTANTES Y CAMPOS EN BLANCO.

- **Artículos**

Los únicos atributos que no tienen campos en blanco son: url, identifier, title y publication_date.

En cuanto al atributo doi, se tiene un total de 7089 artículos con campos en blanco. No es muy relevante puesto que el identificador principal de los artículos es el atributo identifier.

Existen 2383 artículos que no tiene abstract. Esto equivale a solamente el 5.8758% del total de artículos. Si bien el porcentaje es relativamente bajo, esto podría afectar al objetivo de minería de datos, ya que tanto el título como el asbtract serán utilizados para este proceso.

El 19.7184% de artículos no tiene keywords. Tampoco es muy importante ya que con el objetivo de la minería de datos se planea obtener las palabras o términos más relevantes de cada artículo.

Dos artículos no tienen autores ni afiliaciones y 179 no tienen afiliaciones, pero sí autores. Los artículos que no contengan autores no serán tomados en cuenta, mientras que los artículos que no contengan afiliaciones, pero si autores, si serán almacenados.

Tabla 2.19 Registros con campos en blanco por atributo de los artículos

Atributo	Registros con campos en blanco
url	0
identifier	0
title	0
publication_date	0
doi	7089
abstract	2383
author_keywords	7997
authors	2

affiliations	181

- **Autores**

Los atributos first_name e initials contienen campos en blanco en ciertos registros de autor. En total existen 74 registros que no contienen first_name. Esto no es un problema ya que todos los autores cuentan con last_name. En algunos casos, como el del autor con ID 57195319225, el first_name se encuentra incluido en el campo last_name (Figura 21).

Tabla 2.20 Registros con campos en blanco por atributo de los autores

Atributo	Registros con campos en blanco
identifier	0
first_name	74
last_name	0
auth_name	0
initials	19

	identifier	first_name	last_name	auth_name	initials
11846	57195319225	None	Renato Mauricio Toasa	Renato Mauricio Toasa	None

FIGURA 21 Información del autor con ID 57195319225

- **Afiliaciones**

Finalmente, las afiliaciones cuentan solo con 1848 registros que tienen en blanco el campo city. No es relevante, puesto que este estudio se centra en el país Ecuador y no en ciudades específicas.

Tabla 2.21 Registros con campos en blanco por atributo de las afiliaciones

Atributo	Registros con campos en blanco
identifier	0
affiliation_name	0
city	1848
country	0

2.6.3.3.3. IDENTIFICACIÓN DE REGISTROS DUPLICADOS

No existen registros duplicados si tomamos en cuenta solo el ID. Sin embargo, si tomamos en cuenta únicamente el campo Title existen 40316 registros con títulos únicos (240 artículos con título repetido).

Hay varios artículos que tienen el mismo título, pero no son registros duplicados. Tal es el caso de los artículos con título “Preface” que en total suman 61.

Por otra parte, si hay registros duplicados con el mismo título. Por ejemplo, existen dos artículos con el título “System Monitoring for bridges structure” (Figura 22). Este es un caso particular puesto que, si se realiza una búsqueda en Scopus de estos dos artículos, el primer artículo será encontrado con éxito y se podrá visualizar toda la información referente al mismo, en cambio, el segundo artículo no será encontrado (Figura 23). Esto es debido a que el segundo artículo fue reemplazado con un nuevo ID (el del primer artículo). En este caso únicamente será tomado en cuenta el artículo que sí exista en Scopus.

dc:identifier	dc:title	prism:coverDate	prism:doi	dc:description
SCOPUS_ID:85082366329	System Monitoring for bridges structure	2017-01-01	NaN	This work describe the design of a monitoring ...
SCOPUS_ID:85039044168	System Monitoring for bridges structure	2017-01-01	NaN	This work describe the design of a monitoring ...

FIGURA 22 Artículos con título System Monitoring for bridges structure

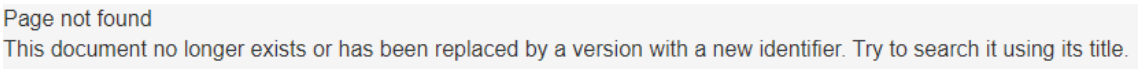


FIGURA 23 Resultado del artículo con ID 85039044168

También existe el caso donde hay artículos con el mismo título pero que fueron indexados en Scopus desde diferentes fuentes o editoriales. Por ejemplo, los artículos con título “Drivers and scorecards to improve hypertension control in primary care practice: Recommendations from the HEARTS in the Americas Innovation Group” fueron indexados desde “The Lancet Regional – Americas” y “Revista Panamericana de Salud Publica/Pan American Journal of Public Health” (Figura 24). En este caso, los artículos con título duplicado si serán almacenados en la base de datos, pero solo formará parte del corpus la primera ocurrencia de cada artículo duplicado.

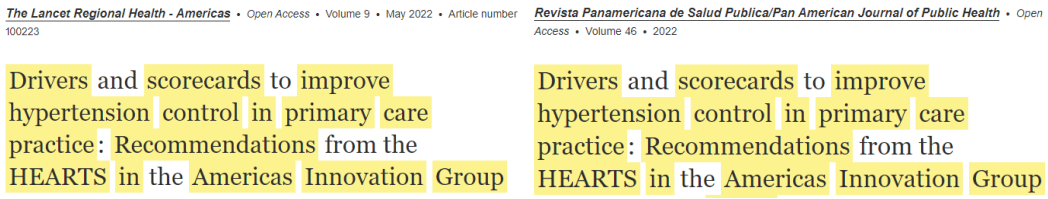


FIGURA 24 Artículos duplicados indexados desde diferentes editoriales

2.6.3.3.4. IDENTIFICACIÓN DE ARTÍCULOS CON CORRECCIONES

Se encontró un total de 108 artículos que son prácticamente actualizaciones con correcciones de otros artículos. Los títulos de estos artículos tienen la particularidad de iniciar con un conjunto de términos (Figura 25) que indican cual es la corrección realizada en el artículo.

Estos artículos sí serán almacenados en la base de datos, pero no formarán parte del corpus para el objetivo de minería de datos.

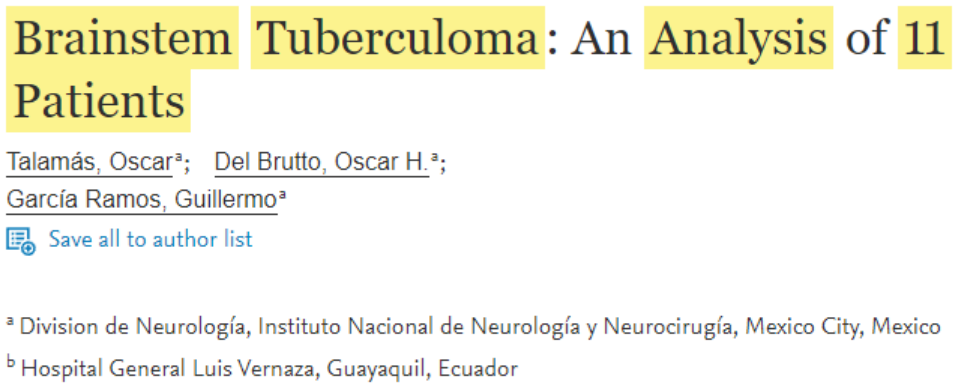
```
keywordsTitle = [
    "Correction to:",
    "Correction:",
    "Erratum to:",
    "Author Correction:",
    "Erratum:",
    "Corrigendum:",
    "Publisher Correction:"
]
```

FIGURA 25 Palabras clave de los títulos de artículos con correcciones

2.6.3.3.5. ANÁLISIS DE ARTÍCULOS SIN RELACIÓN

Todos los artículos extraídos (dejando de lado los 181 artículos que no tienen autores y/o afiliaciones) deberían tener una relación con al menos un autor o afiliación ecuatoriana. Sin embargo, esto no se cumple. Hay 423 artículos que sí tienen autores y afiliaciones, pero ninguno de ellos pertenece a Ecuador. Las razones son variadas. A continuación, se muestra 4 casos diferentes.

En el primer caso, el artículo cuenta con 3 autores, 1 afiliación de México y 1 afiliación de Ecuador. Los 3 autores están relacionados con la afiliación mexicana, pero ninguno con la afiliación ecuatoriana (Figura 26).



Brainstem Tuberculoma: An Analysis of 11 Patients

Talamás, Oscar[a]; Del Brutto, Oscar H.[a]; García Ramos, Guillermo[a]

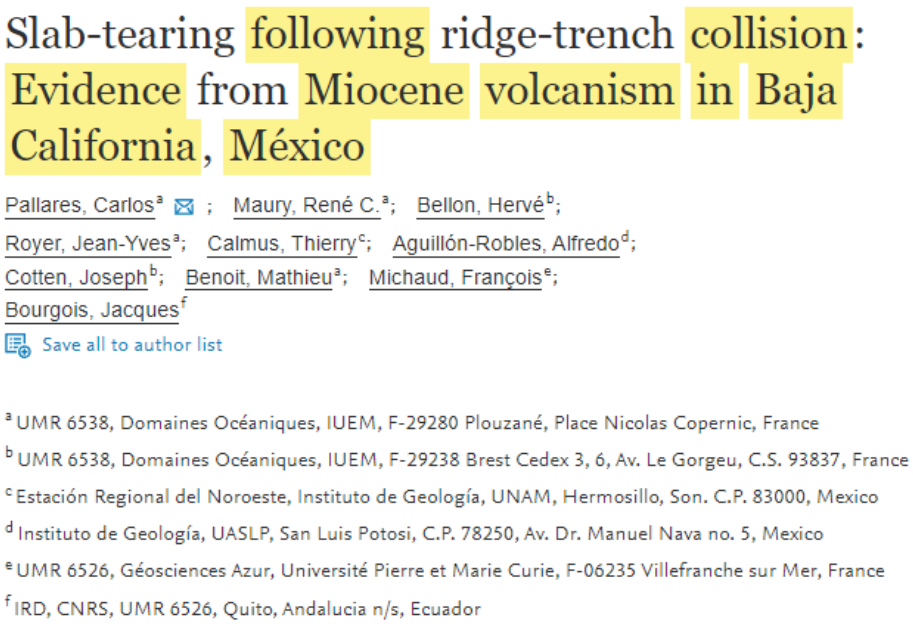
Save all to author list

[a] Division de Neurología, Instituto Nacional de Neurología y Neurocirugía, Mexico City, Mexico

[b] Hospital General Luis Vernaza, Guayaquil, Ecuador

FIGURA 26 Artículo Brainstem Tuberculoma: An Analysis of 11 Patients

En el siguiente artículo (Figura 27), se puede apreciar que el autor Jacques Bourgouis está relacionado con la afiliación con índice f, la cual es ecuatoriana. Sin embargo, esta afiliación se encuentra mal etiquetada.



Slab-tearing following ridge-trench collision: Evidence from Miocene volcanism in Baja California, México

Pallares, Carlos[a] ; Maury, René C.[a]; Bellon, Hervé[b]; Royer, Jean-Yves[a]; Calmus, Thierry[c]; Aguillón-Robles, Alfredo[d]; Cotten, Joseph[b]; Benoit, Mathieu[a]; Michaud, François[e]; Bourgois, Jacques[f]

Save all to author list

[a] UMR 6538, Domaines Océaniques, IUEM, F-29280 Plouzané, Place Nicolas Copernic, France

[b] UMR 6538, Domaines Océaniques, IUEM, F-29238 Brest Cedex 3, 6, Av. Le Gorgeu, C.S. 93837, France

[c] Estación Regional del Noroeste, Instituto de Geología, UNAM, Hermosillo, Son. C.P. 83000, Mexico

[d] Instituto de Geología, UASLP, San Luis Potosi, C.P. 78250, Av. Dr. Manuel Nava no. 5, Mexico

[e] UMR 6526, Géosciences Azur, Université Pierre et Marie Curie, F-06235 Villefranche sur Mer, France

[f] IRD, CNRS, UMR 6526, Quito, Andalucia n/s, Ecuador

FIGURA 27 Artículo Slab-tearing following ridge-trench collision

El problema con el siguiente artículo (Figura 28) es que hay varias afiliaciones definidas en una única afiliación. En esta afiliación se encuentran definidas 3 afiliaciones diferentes, 2 chilenas y 1 ecuatoriana.

Living, belonging and participating: The relationship between neighbourhood and citizen participation in Santiago de Chile

[Vivre, s'appartenir et participer: La relation entre le quartier et la participation citoyenne à Santiago du Chili]

[Viver, pertencer e participar: A relação entre vizinhança e participação cidadã em Santiago do Chile]

[Habitar, pertenecer y participar: La relación entre barrio y participación ciudadana en Santiago de Chile]

Vecchio, Giovanni[a] ; Huerta-Olivares, Consuelo[b] ;

Kanacri, Bernadette Paula Luengo[c]

Save all to author list

[a] Instituto de Estudios Urbanos y Territoriales, Pontificia Universidad Católica de Chile, Santiago Centro de Desarrollo Urbano Sustentable, Pontificia Universidad Católica de Chile, Santiago Facultad de Arquitectura y Urbanismo, Universidad UTE, Quito, Ecuador

[b] Pontificia Universidad Católica de Chile, Chile

[c] Pontificia Universidad Católica de Chile, Santiago, Chile

FIGURA 28 Artículo Living, belonging and participating

El artículo que se muestra a continuación (Figura 29) tiene 1 autor y 1 afiliación, sin embargo, no están relacionados. De igual forma, la afiliación se encuentra mal etiquetada ya que hay 2 afiliaciones definidas en una única afiliación.

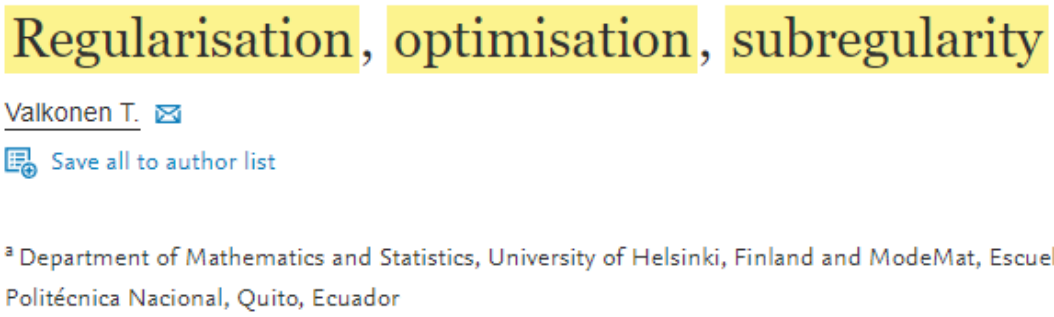


FIGURA 29 Artículo Regularisation, optimisation, subregularity

Si bien las razones son variadas del porqué hay artículos sin relaciones, la mayoría de los casos se debe a que las afiliaciones no fueron indexas correctamente en Scopus, ya sea porque la ciudad o algún otro dato de la afiliación está erróneo o simplemente porque hay varias afiliaciones definidas en una sola afiliación. Otra de las razones comunes es que los autores no están relacionados con las afiliaciones. Esto se puede deber ya no solo a una mala indexación de las afiliaciones, sino que también a una mala indexación de los autores. Todos los artículos que caigan sobre algunos de estos casos no serán tomados en cuenta para el objetivo de minería de datos.

2.6.4. REVISIÓN Y RETROSPECTIVA DEL SPRINT

Se realizó la revisión del Sprint 1 junto con todo el Scrum Team en base a los criterios de aceptación de cada una de las historias de usuario. Los objetivos de este sprint culminaron satisfactoriamente.

- Se planificó los requisitos de los datos por cada fuente de datos.
- Se identificó el modelo de datos de Scopus
- Se desarrolló un módulo en Python para la automatización de peticiones a las APIs de Scopus.
- Se recopiló los datos de las fuentes de datos.
- Se realizó el análisis volumétrico de los datos.

- Se describió los tipos de valores y atributos de cada entidad de los datos.
- Se verificó la calidad de los datos.

2.7. SPRINT 2

2.7.1. OBJETIVOS DEL SPRINT

- Definir el modelo de base de datos.
- Seleccionar los datos que se almacenarán en la base de datos.
- Seleccionar los datos que se utilizarán para el objetivo de minería de datos.
- Construir las entidades derivadas de los artículos y sus relaciones.
- Almacenar los datos en Neo4j.

2.7.2. HISTORIAS DE USUARIO DEL SPRINT

En Figura 30 se muestra el listado de las historias de usuario para el Sprint 2, el cual fue extraído de Azure Boards.

Order	Work Item Type	Title	State	Effort	Assigned To	Value Area	Iteration Path
1	Product Backlo...	Diseño de la base de datos	Approved	3	JOSUE NICOLA...	Business	ResNet\Sprint 2
2	Product Backlo...	Selección de datos	Approved	3	JOSUE NICOLA...	Business	ResNet\Sprint 2
3	Product Backlo...	Construcción de la entidad Afiliación	Approved	3	JOSUE NICOLA...	Business	ResNet\Sprint 2
4	Product Backlo...	Construcción de la entidad Autor	Approved	3	JOSUE NICOLA...	Business	ResNet\Sprint 2
5	Product Backlo...	Contrucción de la entidad Keyword	Approved	3	JOSUE NICOLA...	Business	ResNet\Sprint 2
6	Product Backlo...	Contrucción de las relaciones	Approved	3	JOSUE NICOLA...	Business	ResNet\Sprint 2
7	Product Backlo...	Almacenamiento de los datos en archivos CSV	Approved	2	JOSUE NICOLA...	Business	ResNet\Sprint 2
8	Product Backlo...	Carga de los datos en Neo4j	Approved	5	JOSUE NICOLA...	Business	ResNet\Sprint 2

FIGURA 30 Historias de usuario del Sprint 2

2.7.3. EJECUCIÓN DEL SPRINT

2.7.3.1. DEFINICIÓN DEL MODELO DE BASE DE DATOS

El modelo de la base de datos de este proyecto es un modelo orientado a grafos, el cual consta de nodos, aristas y propiedades (Figura 31). Este modelo está basado en el modelo de datos de Scopus (Figura 13). Las entidades Article, Author y Affiliation del modelo de Scopus han sido parametrizadas como nodos en el modelo orientado a grafos. Adicionalmente, se añade el nodo Topic, que contiene las keywords de los artículos.

Las aristas que unen a los nodos se describen a continuación:

- Un artículo pertenece a una afiliación.
- Un autor está afiliado con una afiliación.
- Un autor escribió un artículo.
- Un artículo usa un tópico.
- Un autor es experto en un tópico.
- Un autor es coautor de otro autor, si escribieron un artículo juntos.

Finalmente, se encuentran las propiedades. Todos los nodos tienen propiedades que describen a estos. Por otro lado, la única arista con propiedades (collab_strength) es la que relaciona un autor con otro autor.

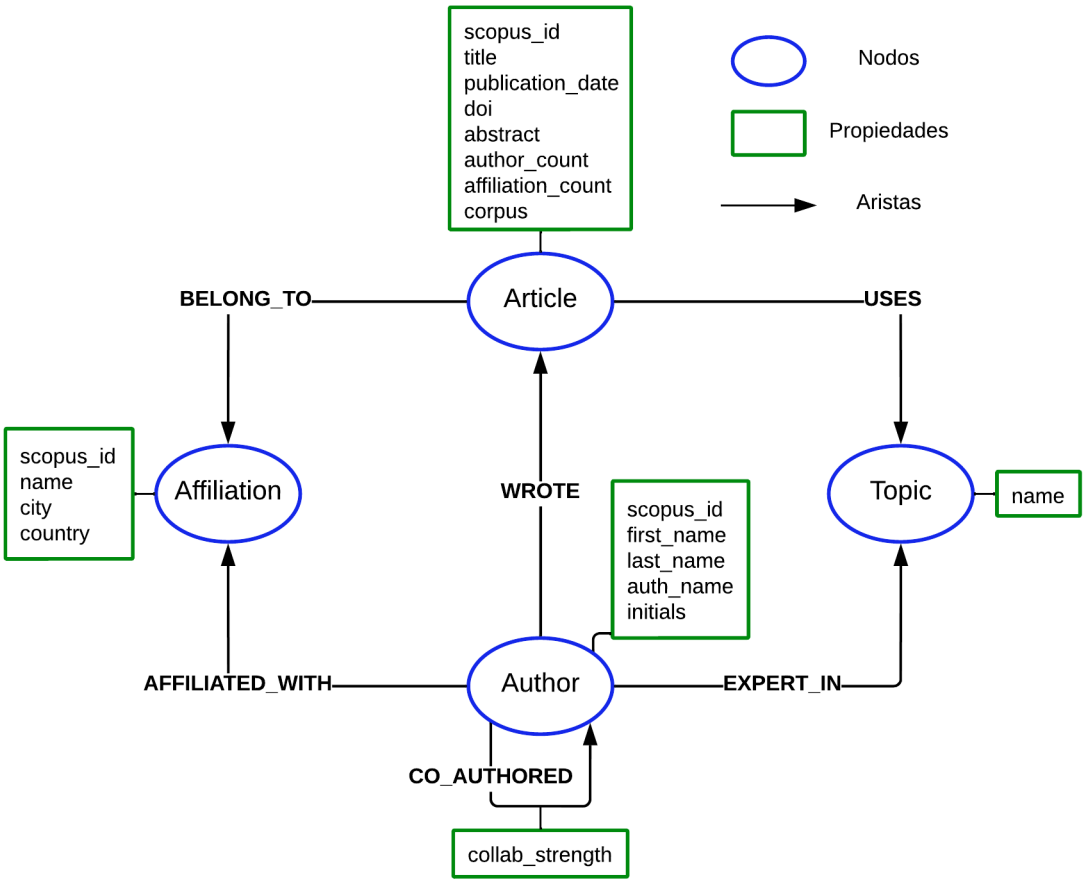


FIGURA 31 Modelo de la base de datos

2.7.3.2. SELECCIONAR LOS DATOS

Esta tarea consiste en decidir qué datos serán almacenados en la base de datos y qué datos se utilizarán en el corpus para el objetivo de minería de datos [27].

2.7.3.2.1. SELECCIÓN DE ARTÍCULOS

Todos los artículos serán almacenados en la base datos a excepción de aquellos que no contengan autores y/o afiliaciones ecuatorianas. En la identificación de los atributos faltantes y campos en blanco se hallaron 181 artículos que no tienen autores y/o afiliaciones. Además, en el análisis de los artículos sin relaciones se

encontraron 423 registros sin relación con autores y/o afiliaciones ecuatorianas. Por lo tanto, el número de artículos que serán incluidos es de 39952.

A continuación, se lista los campos que serán incluidos en los artículos.

- identifier
- title
- publication_date
- doi
- abstract

El atributo URL será excluido. Este atributo representa la URI del artículo en la API Abstract Retrieval y esta información no es relevante. La API Scopus Search retorna por defecto este atributo, sin importar si fue especificado o no en la lista de atributos a ser retornados.

Los atributos authors, affiliations y author_keywords también serán excluidos de los artículos. Estos atributos son utilizados para la creación de las otras entidades y sus relaciones, por lo tanto, incluir estos atributos en los artículos es duplicar información que ya se encuentra en las otras entidades.

Todos los artículos de la base de datos formarán parte del corpus para el objetivo de minería de datos, excepto:

- Los artículos sin abstract.
- Los artículos que son correcciones de otros artículos.
- Los artículos duplicados basado en el título y abstract.

Para identificar a estos artículos se utilizará el campo “corpus” de tipo booleano. En total hay 37526 artículos que serán utilizados en el corpus.

2.7.3.2.2. SELECCIÓN DE AUTORES

Solo los autores que tengan relación con alguna afiliación ecuatoriana serán incluidos. Por lo tanto, el número de autores es de 39225.

A continuación, se lista los atributos incluidos en los autores:

- identifier
- auth_name
- last_name
- first_name
- initials

El atributo afid es el único que será excluido de los autores, ya que este atributo es utilizado para la creación de la relación entre autor y afiliación.

2.7.3.2.3. SELECCIÓN DE AFILIACIONES

Únicamente se incluirán las afiliaciones ecuatorianas. Por lo tanto, el número de afiliaciones es de 5372.

Todos los atributos de la afiliación serán incluidos. A continuación, se lista los atributos:

- identifier
- affiliation_name
- city
- country

2.7.3.3. CONTRUIR LOS DATOS

2.7.3.3.1. CONSTRUCCIÓN DE LAS ENTIDADES DERIVADAS

- **Affiliation:** Esta entidad contiene todas las afiliaciones ecuatorianas extraídas de los artículos a través del atributo “affiliations”, el cual es una lista de objectos (Figura 32). Estas listas son las que generan la entidad Affiliation.


```
[
   {
      "afid":"60106645",
      "affilname":"Universidad Técnica de Ambato",
      "affiliation-city":"Ambato",
      "affiliation-country":"Ecuador"
   },
   {
      "afid":"60001576",
      "affilname":"Universitat de Barcelona",
      "affiliation-city":"Barcelona",
      "affiliation-country":"Spain"
   }
]
```


FIGURA 32 Ejemplo de lista de afiliaciones de un artículo

Las afiliaciones que forman parte de esta entidad son aquellas que tienen como atributo “affiliation-country” a Ecuador. A continuación, en la Figura 33 se muestra el algoritmo:

```
dataAffiliations = []
for index, row in dfArticles.iterrows():
    for affil in row['affiliations']:
        if(affil['affiliation-country'] == "Ecuador"):
            dataAffiliations.append({
                'identifier': affil['afid'],
                'affiliation_name': affil['affilname'] if "affilname" in affil else np.nan,
                'city': affil['affiliation-city'] if "affiliation-city" in affil else np.nan,
                'country': affil['affiliation-country'] if "affiliation-country" in affil else np.nan
            })
```

FIGURA 33 Algoritmo para extraer las afiliaciones ecuatorianas de los artículos

- **Author:** Contiene todos los autores con afiliación ecuatoriana extraídos de los artículos a través del atributo “authors”, el cual es una lista de objetos. Estas listas son las que generan la entidad Author.


```
[
   {
      "authid":"57572669100",
      "authname":"Garcia-Angulo A.C.",
      "surname":"Garcia-Angulo",
      "given-name":"Andrea C.",
      "initials":"A.C.",
      "afid":["60072061"]
   },
   {
      "authid":"35619255200",
      "authname":"Claeskens G.",
      "surname":"Claeskens",
      "given-name":"Gerda",
      "initials":"G.",
      "afid":["60121225"]
   }
]
```


FIGURA 34 Ejemplo de lista de autores de un artículo

Los autores que forman esta entidad son aquellos cuyo atributo “afid” pertenece a algún ID de las afiliaciones ecuatorianas obtenidas previamente. A continuación, en la Figura 35 se muestra el algoritmo:

```
dataAuthors = []
for index, row in dfArticles.iterrows():
    for author in row['authors']:
        if any(item in affiliationsIds for item in author['afid']):
            dataAuthors.append({
                'identifier': author['authid'],
                'first_name': author['given-name'] if "given-name" in author else np.nan,
                'last_name': author['surname'] if "surname" in author else np.nan,
                'auth_name': author['authname'] if "authname" in author else np.nan,
                'initials': author['initials'] if "initials" in author else np.nan,
            })
```

FIGURA 35 Algoritmo para extraer los autores con afiliación ecuatoriana de los artículos

- **Topic:** Esta entidad es un catálogo de todas las keywords que fueron encontradas en los artículos. Las keywords fueron extraídas a través del atributo “author_keywords”. A continuación, en la Figura 36 se muestra el algoritmo:

```
dataAuthorKeywords = []
for index, row in dfArticles.iterrows():
    if type(row['author_keywords']) != float:
        authorKeywords = row['author_keywords'].split(' | ')
        for item in authorKeywords:
            dataAuthorKeywords.append(item)
```

FIGURA 36 Algoritmo para extraer los tópicos de los artículos

2.7.3.3.2. CONSTRUCCIÓN DE LAS RELACIONES

- **Article – Affiliation:** La obtención de las relaciones entre artículos y afiliaciones es muy similar a la obtención de afiliaciones. Mientras que en la entidad Affiliation se almacena todo el objeto referente a la afiliación, en la

relación article – affilation se almacena el ID de la afiliación junto con el ID del artículo. A continuación, en la Figura 37 se muestra el algoritmo:

```
affiliationsIds = dfAffiliations.index.tolist()
dataArticlesAffiliations = []
for index, row in dfArticles.iterrows():
    for affil in row['affiliations']:
        if affil['afid'] in affiliationsIds:
            dataArticlesAffiliations.append({
                'article_id': index,
                'affiliation_id': affil['afid']
            })
```

FIGURA 37 Algoritmo para la extracción de relaciones entre artículos y afiliaciones

- **Article – Author:** Las relaciones entre artículos y autores fueron obtenidas de forma muy similar a la obtención de la entidad Author, solo que aquí únicamente se almacena el ID de autor y el ID del artículo. A continuación, en la Figura 38 se muestra el algoritmo:

```
affiliationsIds = dfAffiliations.index.tolist()
dataArticlesAuthors = []
for index, row in dfArticles.iterrows():
    for author in row['authors']:
        if any(item in affiliationsIds for item in author['afid']):
            dataArticlesAuthors.append({
                'article_id': index,
                'author_id': author['authid']
            })
```

FIGURA 38 Algoritmo para la extracción de relaciones entre artículos y autores

- **Article – Topics:** El proceso para obtener las relaciones entre article y topic es muy similar al proceso de obtención de la entidad Topic. En esta relación se almacena el ID del topic y el ID del artículo. A continuación, en la Figura 39 se muestra el algoritmo.

```
dataArticlesAuthorKeywords = []
for index, row in dfArticles.iterrows():
    if type(row['author_keywords']) != float:
        authorKeywords = row['author_keywords'].split(' | ')
        for item in authorKeywords:
            dataArticlesAuthorKeywords.append({
                'article_id': index,
                'author_keyword_id': dataAuthorKeywords.index(item)
            })
```

FIGURA 39 Algoritmo para la extracción de relaciones entre artículos y tópicos

- **Author – Affiliation:** Para obtener esta relación se realizó un proceso similar a la obtención de autores. En esta relación se almacena el ID del autor y el ID de la afiliación.

```
dataAuthorsAffiliations = []
for index, row in dfArticles.iterrows():
    for author in row['authors']:
        for afid in author['afid']:
            if afid in affiliationsIds:
                dataAuthorsAffiliations.append({
                    'author_id': author['authid'],
                    'affiliation_id': afid
                })
```

FIGURA 40 Algoritmo para la extracción de relaciones entre autores y afiliaciones

- **Author – Author:** Esta es la única relación o arista que tiene propiedades. La propiedad “collab_strength” representa la medida de fuerza de colaboración entre un par de autores. Cada artículo en coautoría de un par de autores dado agrega una cantidad 1/(n-1) a la fuerza de su colaboración, donde n es el número total de autores en el artículo. El fundamento de esta medida es que un autor divide su tiempo entre los n-1 otros autores con los que trabaja en un artículo y, por lo tanto, la fuerza de colaboración con cada uno de ellos varía inversamente con respecto a n-1 [4]. Por ejemplo (Figura 41), los autores A y B son coautores en 3 artículos, en donde cada uno de estos tiene 4, 3 y 3 autores respectivamente. Por lo tanto, aplicando la fórmula, da una fuerza de 1/3, 1/2 y 1/2, con un total de 4/3 de fuerza de colaboración entre los dos autores.

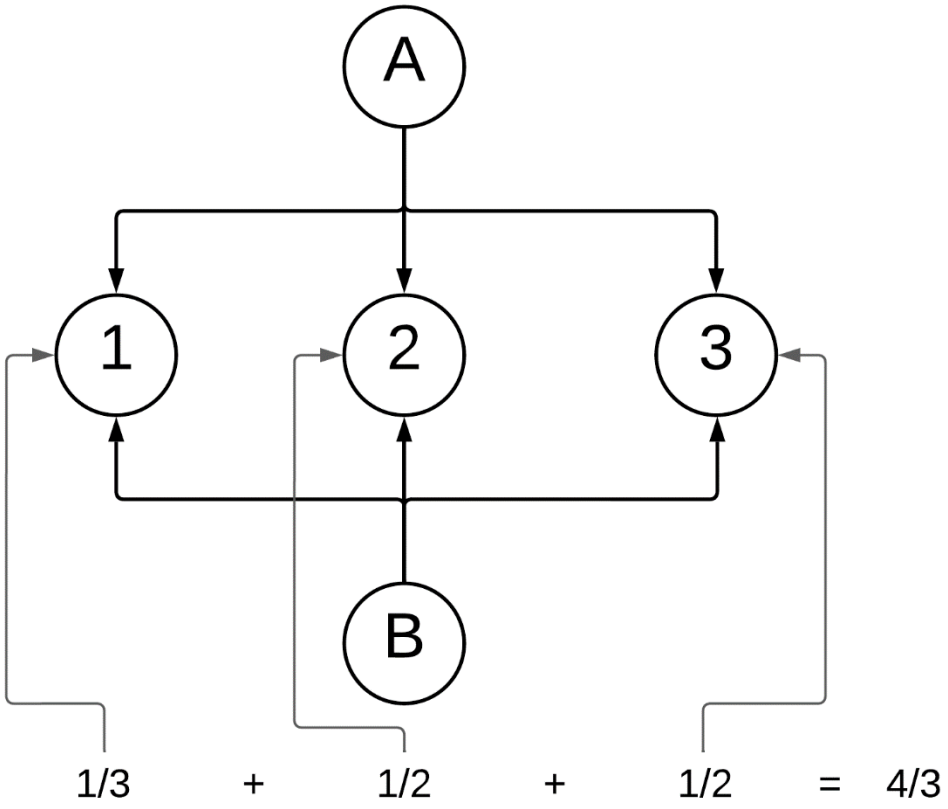


FIGURA 41 Ejemplo de fuerza de colaboración entre dos autores

La creación de esta arista y el cálculo de la medida de fuerza de colaboración entre cada par de autores se encuentra en el notebook “Collaboration Strength Measure Calculation” del **ANEXO 1**, que se encuentra en la sección de ANEXOS del documento

2.7.3.4. ALMACENAR LOS DATOS

Toda la data presentada previamente se encuentra almacenada en archivos de tipo pickle, el cual es un módulo que implementa distintos protocolos para serializar y deserializar un objeto de Python [28]. Se utilizó pickle debido a que era necesario cargar la data en distintos notebooks o scripts y este módulo permite reconstruir un objeto de Python en otro script de una manera rápida. Si bien almacenar la data en archivos pickle resultó muy útil para el procesamiento de los datos, este tipo de archivos solo puede ser manejado por Python. Por lo tanto, es necesario almacenar la data en una fuente que pueda ser utilizada independientemente del lenguaje de programación.

La fuente de almacenamiento seleccionada es Neo4j, el cual es una base de datos orientada a grafos.

2.7.3.4.1. ALMACENAMIENTO DE LA DATA EN ARCHIVOS CSV

No se puede cargar la data directamente desde un archivo pickle a Neo4j. La forma más fácil para realizar esta carga es primero almacenar la data en archivos CSV.

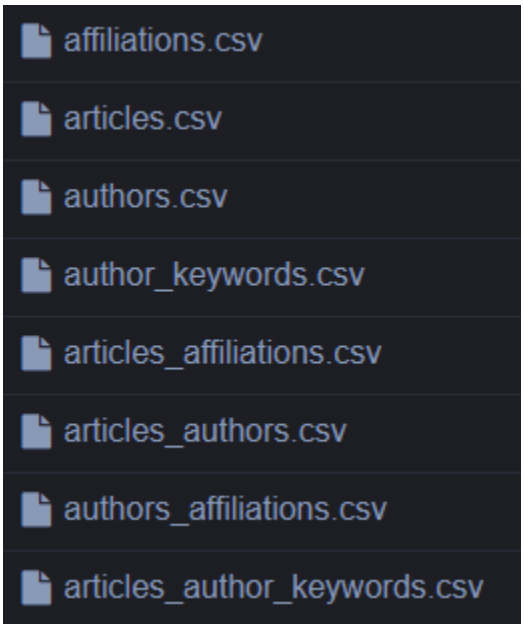


FIGURA 42 Data almacenada en archivos CSV

2.7.3.4.2. CARGA DE LOS DATOS EN NEO4J

En esta sección se describe los pasos para crear el proyecto en Neo4j y cargar la data desde archivos CSV.

1. Crear e iniciar un proyecto en Neo4j.

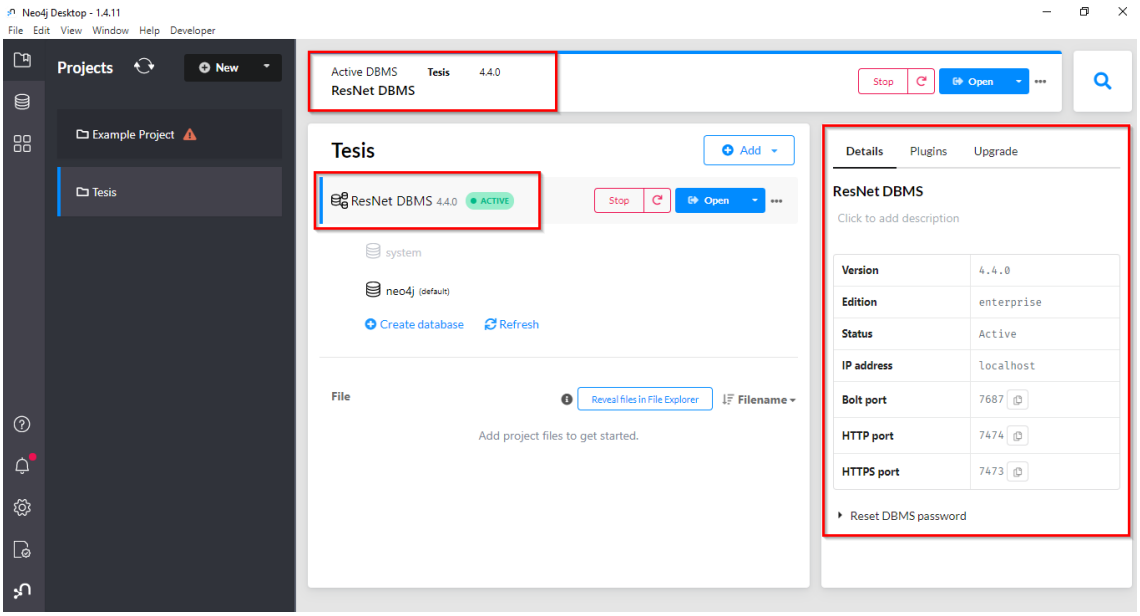


FIGURA 43 Proyecto nuevo en Neo4j

2. Configurar el proyecto para que permita cargar información desde archivos externos.

Esta configuración restringe todos los archivos de importación `LOAD CSV` para que necesariamente estén en el directorio 'import'. Quitarlo o comentarlo permite que los archivos se carguen desde cualquier parte del sistema de archivos.

```
#dbms.directories.import=import
```

FIGURA 44 Configuración en Neo4j para que se pueda cargar archivos

El siguiente comando determina si Cypher permitirá importar archivos CSV desde la URL de estos.

```
dbms.security.allow_csv_import_from_file_urls=true
```

FIGURA 45 Configuración en Neo4j para que se pueda cargar archivos CSV

3. Crear los constrains y cargar la data.

En este paso se trabaja principalmente con la librería Py2neo, la cual es una biblioteca y un conjunto de herramientas para trabajar con Neo4j desde aplicaciones de Python.

Primero, es necesario configurar la conexión a la base de datos Neo4j.

```
graph = Graph("bolt://localhost:7687", auth=("neo4j", "narias"))
```

FIGURA 46 Conexión a Neo4j desde un notebook de Python

A continuación, se crean los 'constraints'. Los identificadores de los nodos Affiliation, Article y Author son únicos, al igual que el campo name del nodo Topic.

```
#Constraint del id de los autores
graph.run("CREATE CONSTRAINT authorScopusIdConstraint ON (au:Author) ASSERT au.scopus_id IS UNIQUE")

(No data)

#Constraint del name de las author_keywords
graph.run("CREATE CONSTRAINT topicsNameConstraint ON (t:Topic) ASSERT t.name IS UNIQUE")

(No data)
```

FIGURA 47 Constraints de la base de datos

Finalmente, se carga la data de los nodos y sus aristas.

```
#articles
query = """
LOAD CSV WITH HEADERS
FROM """+'"'+articles_path+'"'+""" AS csvLine
CREATE (ar:Article {scopus_id: csvLine.identifier,
title: csvLine.title,
publication_date: csvLine.publication_date,
doi: csvLine.doi,
abstract: csvLine.abstract,
author_count: csvLine.author_count,
affiliation_count: csvLine.affiliation_count,
corpus: csvLine.corpus
})
RETURN count(ar)
"""
graph.run(query)
```

count(ar)
39952

FIGURA 48 Carga de los artículos a Neo4j desde archivo CSV

```
#articles_affiliations
query = """
USING PERIODIC COMMIT 500
LOAD CSV WITH HEADERS
FROM """+'"'+articles_affiliations_path+'"'+""" AS csvLine
MATCH (ar:Article {scopus_id: csvLine.article_id}),
(af:Affiliation {scopus_id: csvLine.affiliation_id})
CREATE (ar)-[r:BELONGS_TO]->(af)
RETURN count(r)
"""
graph.run(query)
```

count(r)
55302

FIGURA 49 Carga de las aristas entre los nodos articles y affilitions a Neo4j desde archivo CSV

2.7.3.4.3. RESULTADOS DE LA CARGA DE DATOS

A continuación, se muestra el tiempo de duración y fecha de carga de la data a la base de datos Neo4j desde archivos CSV.

- Duración: 30.461089611053467 segundos.
- Fecha de inicio de la carga: Mon Oct 17 22:44:27 2022
- Fecha finalización de la carga: Mon Oct 17 22:44:54 2022

2.7.4. REVISIÓN Y RETROSPECTIVA DEL SPRINT

Se realizó la revisión del Sprint 2 junto con todo el Scrum Team en base a los criterios de aceptación de cada una de las historias de usuario. Los objetivos de este sprint culminaron satisfactoriamente.

- Se definió el modelo de base de datos orientada a grafos.
- Se seleccionó los artículos para el objetivo de minería de datos.
- Se seleccionó los autores para el objetivo de minería de datos.
- Se seleccionó las afiliaciones para el objetivo de minería de datos.
- Se construyó las entidades derivadas a partir de la entidad article.
- Se construyó las relaciones de la entidad article.
- Se almacenó la data en Neo4j.

2.8. SPRINT 3

2.8.1. OBJETIVOS DEL SPRINT

- Seleccionar la técnica de modelado.
- Generar el diseño de pruebas.

- Construir el modelo.
- Evaluar el modelo.

2.8.2. HISTORIAS DE USUARIO DEL SPRINT

En la Figura 50 se muestra el listado de las historias de usuario para el Sprint 3, extraído de Azure Boards.

Order	Work Item Type	Title	State	Effort	Assigned To	Value Area	Iteration Path
1	Product Backlo...	Seleccionar la técnica de modelado	Approved	3	JOSUE NICOLA...	Business	ResNet\Sprint 3
2	Product Backlo...	Generar diseño de prueba	Approved	5	JOSUE NICOLA...	Business	ResNet\Sprint 3
3	Product Backlo...	Construir el modelo	Approved	5	JOSUE NICOLA...	Business	ResNet\Sprint 3
4	Product Backlo...	Evaluar el modelo	Approved	5	JOSUE NICOLA...	Business	ResNet\Sprint 3

FIGURA 50 Historias de usuario del Sprint 3

2.8.3. EJECUCIÓN DEL SPRINT

2.8.3.1. SELECCIONAR LA TÉCNICA DE MODELADO

En el modelado de datos se utilizó tf-idf para determinar cuan relevante es una palabra o palabras en el corpus. Los pasos para el modelamiento fueron los siguientes:

1. Generar dos corpus con el texto procesado. El primer corpus está formado por artículos (title, abstract y topics del artículo), mientras que el segundo corpus está formado por autores (listado de artículos del autor). En el corpus por artículo se pretende determinar la relevancia de una palabra en un artículo. En el corpus por autor se pretende determinar la relevancia de una palabra para un autor basado en sus artículos.

2. Obtener la matriz dispersa de tf-idf para cada corpus que contiene los pesos para cada palabra. Las columnas de la matriz dispersa corresponden a las palabras del corpus y las filas corresponden a los artículos o autores dependiendo del corpus.

2.8.3.2. GENERAR EL DISEÑO DE PRUEBA

El procedimiento que se utilizará para evaluar el modelo es el siguiente:

1. Obtener aleatoriamente un topic de la base de datos.
2. En el caso de que el topic esté formando por varias palabras se sumarán los pesos de cada palabra.
3. Extraer los documentos (artículos o autores dependiendo del corpus) que tengan el mayor peso para el topic.
4. Determinar si los documentos obtenidos están relacionados con el topic.

2.8.3.3. CONSTRUIR EL MODELO

Para la generación de las matrices dispersas se utilizará la clase TfidfVectorizer de la librería de Python scikit-learn. Esta clase convierte un conjunto de documentos procesados o sin procesar en una matriz tf-idf [29].

La clase TfidfVectorizer recibe varios parámetros para el procesamiento del corpus y la aplicación de las variantes más comunes para calcular tf-idf. En la Tabla 2.22, describe los parámetros utilizados para este caso.

Tabla 2.22 Parámetros de la clase TfidfVectorizer

Parámetro	Descripción

norm	Cada fila de la matriz tendrá una norma de unidad, ya sea [29]: • 'l2': La suma de los cuadrados de los elementos vectoriales es 1. La similitud del coseno entre dos vectores es su producto escalar cuando se ha aplicado la norma l2. • 'l1': La suma de los valores absolutos de los elementos vectoriales es 1. • Ninguno: Sin normalización.
smooth_idf	Suaviza los pesos de idf agregando uno a las frecuencias de los documentos, como si se viera un documento adicional que contiene todos los términos de la colección exactamente una vez. Esto evita las divisiones por cero [29].
sublinear_tf	Aplica la escala sublineal tf, es decir, reemplaza el tf con 1 + log(tf) [29].

El método “fit_transform” es el que retorna la matriz dispersa. Dado que los documentos de los corpus ya se encuentran procesados, el único parámetro que recibe este método es el corpus mapeado en una lista de strings.

```
tfidf = TfidfVectorizer(norm='l2', smooth_idf=True, sublinear_tf=True)

Obtener la matrix tf-idf del método fit_transform()

matrix = tfidf.fit_transform(corpus['preprocessed_doc'].to_list())
```

FIGURA 51 Generación de la matriz tf-idf

La matriz que retorna no es un array de dos dimensiones, sino que es una matriz de filas dispersas comprimida. Este tipo de matriz pertenece al paquete de matriz dispersa SciPy 2-D para datos numéricos [30].

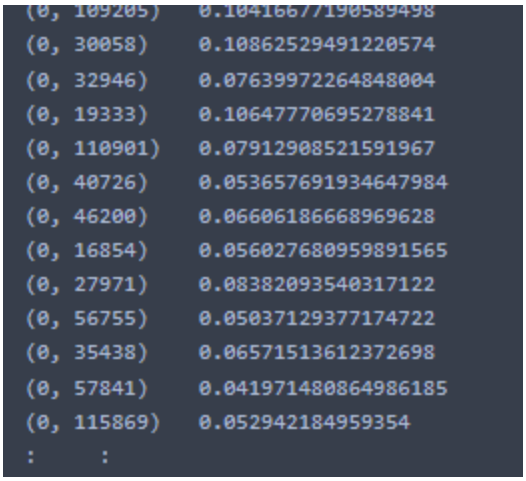


FIGURA 52 Matriz dispersa tf-idf por autor

Para acceder a los pesos de la matriz se debe utilizar índices en lugar de palabras. Para ello, la clase TfidfVectorizer tiene el atributo “vocabulary_”, el cual es un diccionario que mapea las palabras a índices.

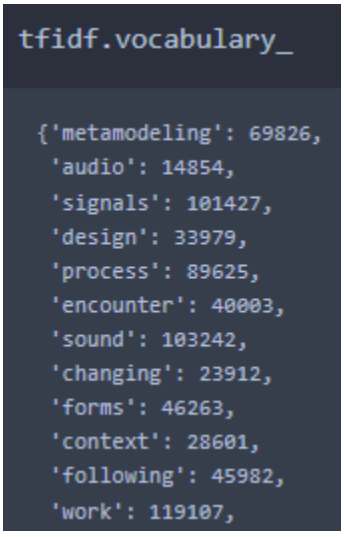


FIGURA 53 Vocabulario del corpus por autor

2.8.3.4. EVALUAR EL MODELO

La evaluación será realizada para las dos matrices tf-idf generadas. Esta evaluación se encuentra en el notebook “3. Evaluación del modelo” del **ANEXO 1**, que se encuentra en la sección de ANEXOS del documento.

En la Tabla 2.23 se muestra los resultados de la evaluación de la matriz tf-idf por autor. A continuación, se describe el contenido de la tabla:

- **Topic:** Topics obtenidos aleatoriamente de la base de datos neo4j.
- **Topic procesado y tokenizado:** Topics procesados y tokenizados mediante el mismo método con el que fue procesado el corpus.
- **Índice del topic:** Índice de los tokens de cada topic.
- **Autores más relevantes:** Identificadores de los 10 autores más relevantes.
- **Autores relacionados con el topic:** Número de autores relacionados con el topic.

Tabla 2.23 Evaluación de la matriz tf-idf por Autor

Topic	Topic procesado y tokenizado	Índice del topic	Autores más relevantes	Autores relacionados con el topic
Text mining	['text', 'mining']	[109419, 71296]	['57290915500', '57202512236', '57207911849', '57200571551', '57204965246', '57218271614', '57202686886', '57223280018', '57211492684', '57219593332']	10/10

Spanish-speaking populations	['spanish', 'speaking', 'populations']	[103392, 103503, 87853]	['57196245340', '57196234048', '57196244505', '57196244568', '55957692600', '57196237493', '6507659036', '7005457720', '55962551700', '57219112431']	10/10
Cyberbullying	['cyberbullying']	[31418]	['57219837893', '57222126064', '57218554945', '57218551081', '57659803500', '57217131613', '57219372651', '57207570239', '57219841379', '57219837818']	10/10
Political communication	['political', 'communication']	[87361, 27236]	['57202946139', '57200917428', '57200918489', '57214916312', '57214910972', '57274029500', '57211394703', '57192641923', '57211049014', '57219655945']	10/10

Internet of Things	['internet', 'things']	[58971, 109973]	['57216967387', '57465239000', '57465676200', '57466119100', '57217228711', '57210091916', '57218209207', '57218212445', '57218209389', '57194717717']	10/10

En la Tabla 2.24 se muestra los resultados de la evaluación de la matriz tf-idf por artículo. A continuación, se describe el contenida de la tabla:

- **Topic:** Topics obtenidos aleatoriamente de la base de datos neo4j.
- **Topic procesado y tokenizado:** Topics procesados y tokenizados mediante el mismo método con el que fue procesado el corpus.
- **Índice del topic:** Índice de los tokens de cada topic.
- **Artículos más relevantes:** Identificadores de los 10 artículos más relevantes.
- **Artículos relacionados con el topic:** Número de artículos relacionados con el topic.

Tabla 2.24 Evaluación de la matriz tf-idf por Artículo

Topic	Topic procesado y tokenizado	Índice del topic	Artículos más relevantes	Artículos relacionados con el topic
Public health	['public', 'health']	[91140, 52067]	['84857382685', '85049415498', '85052649511',	10/10

			'85071164002', '85048535587', '85019115967', '40949107228', '85006977973', '41949137310', '85087140287']	
Rural development	['rural', 'development']	[97620, 34268]	['85016269034', '85029577297', '55949113939', '85090042928', '85104838301', '85100894019', '85050337004', '85085701452', '85104891427', '85100910068']	10/10
Primary education	['primary', 'education']	[89356, 38550]	['85078408666', '85106020540', '85119093697', '85101790287', '85118627123', '85097394722', '85090621418', '85091246747', '85075281267', '85133359838']	10/10
Business Intelligence	['business', 'intelligence']	[20494, 58610]	['84942517860', '84982098362', '85027235734', '85048600288',	10/10

			'85066957387', '84961864457', '85105517043', '85108692604', '85105512565', '85026312087']	
Network analysis	['network', 'analysis']	[75980, 11230]	['85102897015', '85094572726', '85046267475', '85018980182', '85091396092', '85026820061', '85077490882', '85061186593', '85086586279', '85032038011']	10/10

2.8.4. REVISIÓN Y RETROSPECTIVA DEL SPRINT

Se realizó la revisión del Sprint 3 junto con todo el Scrum Team en base a los criterios de aceptación de cada una de las historias de usuario. Los objetivos de este sprint culminaron satisfactoriamente.

- Se seleccionó a tf-idf como la técnica de modelado.
- Se generó un diseño de pruebas.
- Se construyó el modelo en base al corpus de artículos.
- Se construyó el modelo en base al corpus de autores.
- Se evaluó los modelos y sus resultados.

2.9. SPRINT 4

2.9.1. OBJETIVOS DEL SPRINT

- Diseñar las interfaces de usuario
    - Diseñar la interfaz de búsqueda
    - Diseñar la interfaz de resultados de búsqueda de autor
    - Diseñar la interfaz de resultados de búsqueda de autores más relevantes
    - Diseñar la interfaz de resultados de búsqueda de artículos más relevantes
- Desarrollar y codificar la interfaz de búsqueda

2.9.2. HISTORIAS DE USUARIO DEL SPRINT

En la Figura 54 se muestra el listado de las historias de usuario para el Sprint 4, extraído de Azure Boards.

Order	Work Item Type	Title	State	Effort	Assigned To	Value Area	Iteration Path
1	Product Backlo...	Diseñar la interfaz de búsqueda	Approved	3	JOSUE NICOLA...	Business	ResNet\Sprint 4
2	Product Backlo...	Diseñar la interfaz de resultados de búsqueda por autor	Approved	5	JOSUE NICOLA...	Business	ResNet\Sprint 4
3	Product Backlo...	Diseñar la interfaz de resultados de búsqueda de autores m...	Approved	3	JOSUE NICOLA...	Business	ResNet\Sprint 4
4	Product Backlo...	Diseñar la interfaz de resultados de búsqueda de artículos ...	Approved	3	JOSUE NICOLA...	Business	ResNet\Sprint 4
5	Product Backlo...	Codificar la interfaz de búsqueda	Approved	5	JOSUE NICOLA...	Business	ResNet\Sprint 4

FIGURA 54 Historias de usuario del Sprint 4

2.9.3. EJECUCIÓN DEL SPRINT

2.9.3.1. DISEÑAR LAS INTERFACES DE USUARIO

El diseño de páginas web es fundamental para generar un producto de calidad. Esta fase cubre todos los aspectos relacionados a la estructuración, organización y distribución de todos los componentes visuales del sistema. Los prototipos de las interfaces de usuario del sistema fueron creados a través de la herramienta Figma.

2.9.3.1.1. INTERFAZ DE BÚSQUEDA

Esta es la interfaz principal del sistema. En esta interfaz se puede realizar búsquedas de información académica, es decir, búsqueda de autor, búsqueda de autores relevantes por tópico y búsqueda del estado de arte.

La Figura 55 muestra el prototipo de la interfaz de búsqueda que consta de los siguientes componentes:

- **Navbar:** Es la barra de navegación del sistema. Esta se encuentra ubicada en la parte superior y se repite en todas las interfaces.
- **Carousel:** Componente que muestra información acerca de los tipos de búsqueda que se puede realizar en el sistema.
- **Search:** Componente que realiza las búsquedas.
- **Footer:** Es el pie de página. Esta se repite en todas las interfaces.

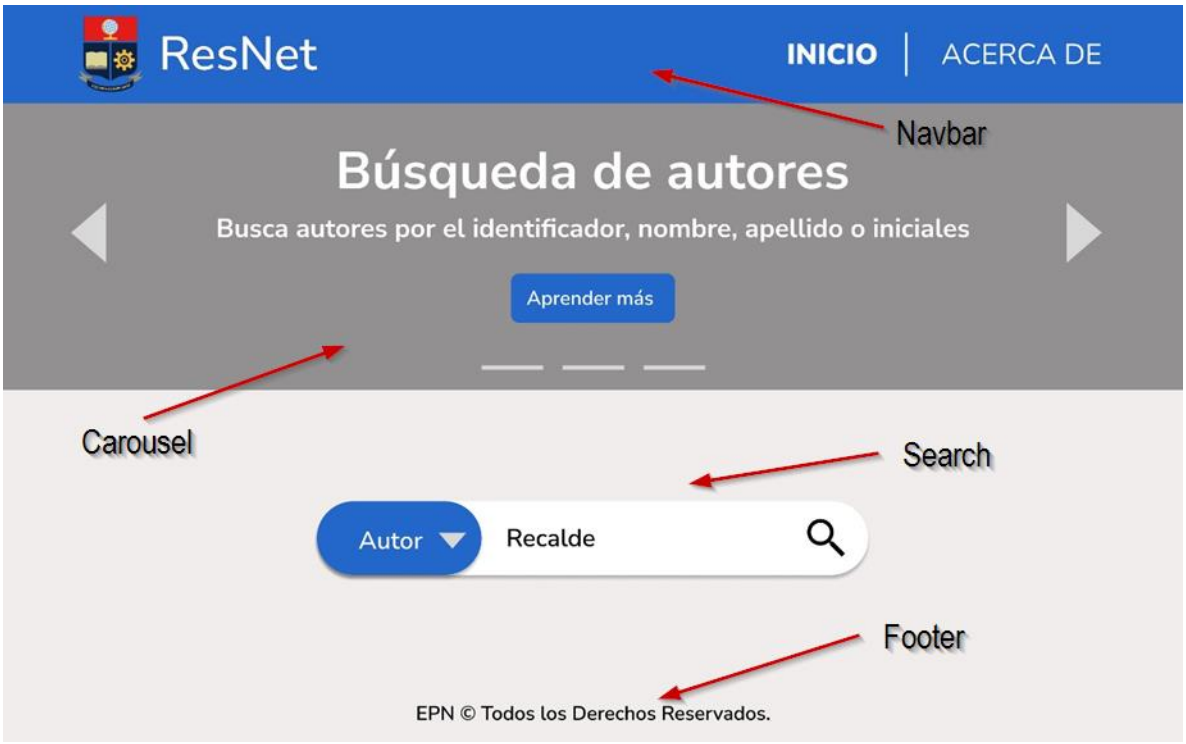


FIGURA 55 Prototipo de la interfaz de búsqueda

2.9.3.1.2. INTERFAZ ACERCA DE

Esta interfaz es meramente informativa. Su propósito es informar detalladamente a los usuarios acerca de las funcionalidades del sistema, es decir, los tipos de búsqueda que se puede realizar y otros detalles del proyecto desarrollado. En la Figura 56 se muestra el prototipo de la interfaz.

FIGURA 56 Prototipo de la interfaz acerca de

2.9.3.1.3. INTERFAZ DE AUTOR

En esta interfaz se presenta la información de un autor divida por las siguientes secciones: información general, artículos del autor, coautores y tópicos del autor. En la Figura 57 se muestra el prototipo de esta interfaz.

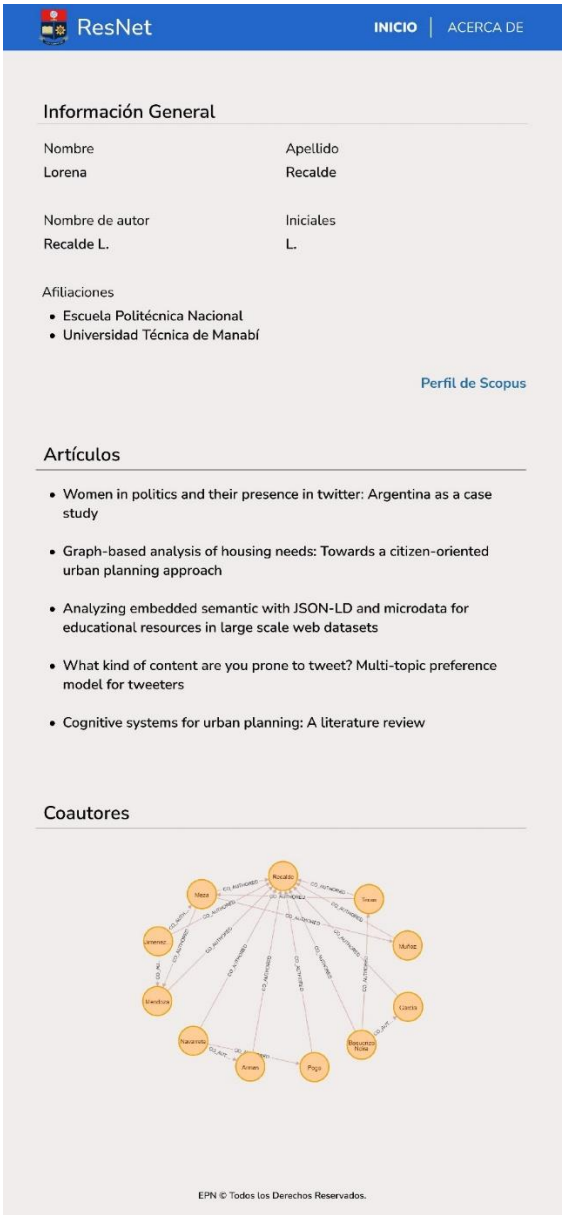


FIGURA 57 Prototipo de la interfaz de autor

2.9.3.1.4. INTERFAZ DE RESULTADOS DE BÚSQUEDA DE AUTORES RELEVANTES

En esta interfaz se muestra los resultados de la búsqueda de autores relevantes por tópico. La Figura 58 muestra el prototipo de esta interfaz que consta de dos compontes:

- **Filtros:** Cuadro que permite filtrar los resultados por número de autores y afiliaciones.
- **Grafo:** Grafo que contiene los autores más relevantes y cómo estos están relacionados.

FIGURA 58 Prototipo de la interfaz de resultados de búsqueda de autores relevantes

2.9.3.2. CODIFICAR LA INTERFAZ DE BÚSQUEDA

Antes de iniciar con la codificación, se creó un proyecto en Angular y se instaló las dependencias necesarias. Estas tareas fueron realizadas desde la consola con la ayuda de Angular CLI, la cual es una herramienta de interfaz de línea de comandos para agilizar el desarrollo de proyectos en Angular.

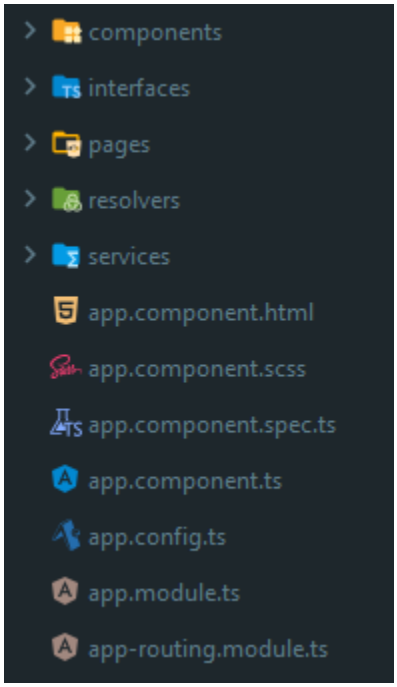


FIGURA 59 Estructura de carpetas del frontend

El proyecto cuenta con la siguiente estructura de carpetas (Figura 59):

- **Components:** En esta carpeta se almacenan todos los componentes que conforman el sistema (navbar, search, footer, etc).
- **Interfaces:** Aquí se encuentran definidas todas las interfaces o modelos de los objetos utilizados en el sistema (article, author, etc).
- **Pages:** Esta carpeta contiene todas las plantillas de las páginas que son renderizadas mediante la ruta.

- **Resolvers:** En esta carpeta se encuentran los servicios que se utilizan para precargar datos para una ruta antes de que se cargue la vista.
- **Services:** Aquí se encuentran todos servicios que realizan peticiones HTTP a la API de Flask.

La interfaz de búsqueda cuenta con varios elementos visuales (carousel, search). Cada uno de estos elementos fueron desarrollados como componentes dentro de angular y añadidos a la plantilla “home” (Figura 60).

Los componentes navbar y footer no pertenecen a ninguna plantilla. Estos dos son renderizados en el componente raíz, App Component y podrán visualizarse en todas las plantillas.

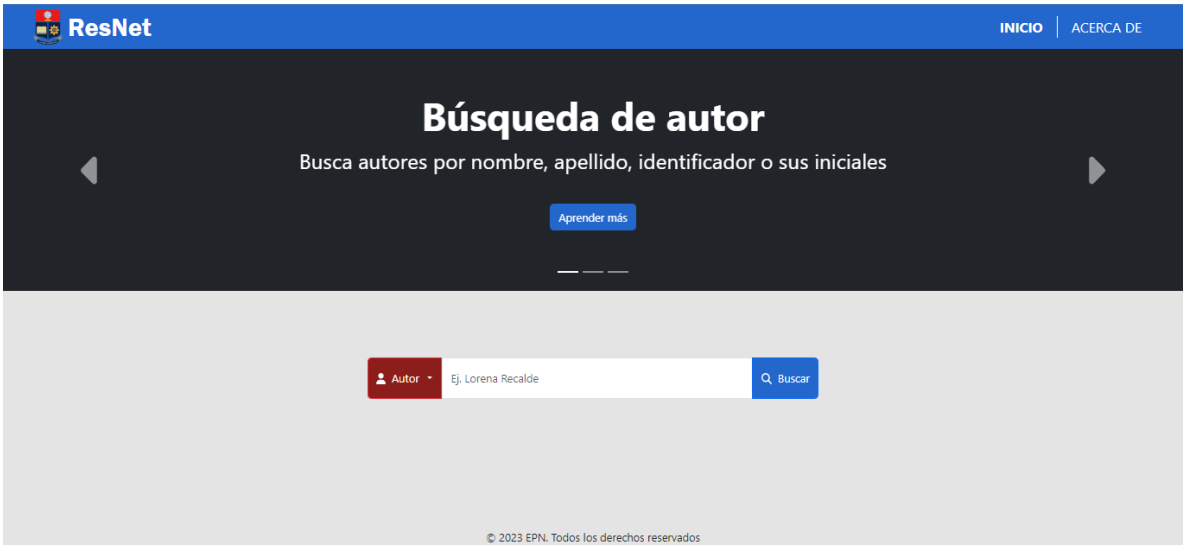


FIGURA 60 Interfaz de búsqueda

2.9.4. REVISIÓN Y RETROSPECTIVA DEL SPRINT

Se realizó la revisión del Sprint 4 junto con todo el Scrum Team en base a los criterios de aceptación de cada una de las historias de usuario. Los objetivos de este sprint culminaron satisfactoriamente.

- Se diseñó la interfaz de búsqueda.
- Se diseñó la interfaz de resultados de búsqueda de autor.
- Se diseñó la interfaz de resultados de búsqueda de autores más relevantes.
- Se diseñó la interfaz de resultados de búsqueda de artículos más relevantes.
- Se desarrolló la interfaz de búsqueda.

2.10. SPRINT 5

2.10.1. OBJETIVOS DEL SPRINT

- Implementar una capa de abstracción de D3 en Angular.
- Codificar la interfaz de resultados de búsqueda por autor.
- Codificar la interfaz de resultados de búsqueda de autores más relevantes.
- Codificar la interfaz de resultados de búsqueda de artículos más relevantes.

2.10.2. HISTORIAS DE USUARIO DEL SPRINT

En la Figura 61 se muestra el listado de las historias de usuario para el Sprint 5, extraído de Azure Boards.

Order	Work Item Type	Title	State	Effort	Assigned To	Value Area	Iteration Path
1	Product Backlo...	Codificar la interfaz de resultados de búsqueda por autor	Approved	5	JOSUE NICOLA...	Business	ResNet\Sprint 5
2	Product Backlo...	Implementar una capa de abstracción de D3 en Angular	Approved	8	JOSUE NICOLA...	Business	ResNet\Sprint 5
3	Product Backlo...	Codificar la interfaz de resultados de la búsqueda de autore...	Approved	5	JOSUE NICOLA...	Business	ResNet\Sprint 5
4	Product Backlo...	Codificar la interfaz de resultados de la búsqueda de artícul...	Approved	5	JOSUE NICOLA...	Business	ResNet\Sprint 5

FIGURA 61 Historias de usuario del Sprint 5

2.10.3. EJECUCIÓN DEL SPRINT

2.10.3.1. INTERFAZ DE RESULTADOS DE BÚSQUEDA DE AUTOR

La búsqueda de autor se halla en la interfaz principal. Está compuesto por el componente Search para realizar la búsqueda y los resultados que se presentan en una tabla paginada con todos los autores que coincidan con la query de búsqueda (Figura 61). La tabla es parte de los componentes que ofrece Bootstrap. Todos estos componentes se renderizan dentro de la plantilla “home”.

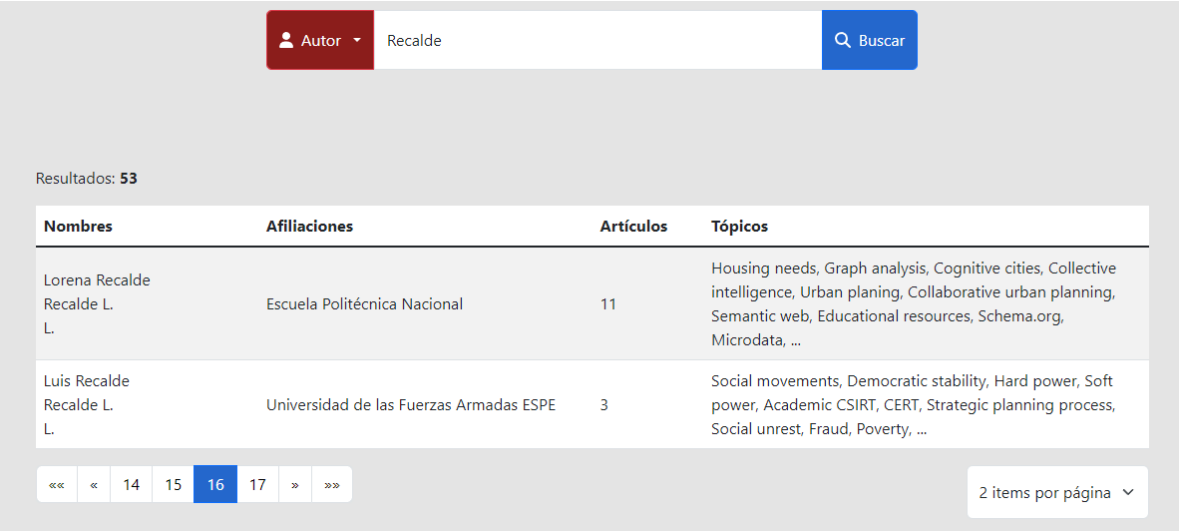


FIGURA 62 Interfaz de resultados de búsqueda de autor

Una vez hallado al autor, es posible hacer clic sobre éste para poder visualizar su perfil (Figura 57). Antes de que cargue la vista que renderiza el perfil del autor, se realiza una petición al backend para obtener toda la información académica del autor. Esto se consigue gracias al resolver author-resolver.ts, que es el encargado de hacer la petición y cargar la data en la vista. El grafo de coautoría es la única información académica que no proporciona el resolver. En cambio, esta información es obtenida por demanda del usuario al hacer clic en el botón “Visualizar grafo” (Figura 62).

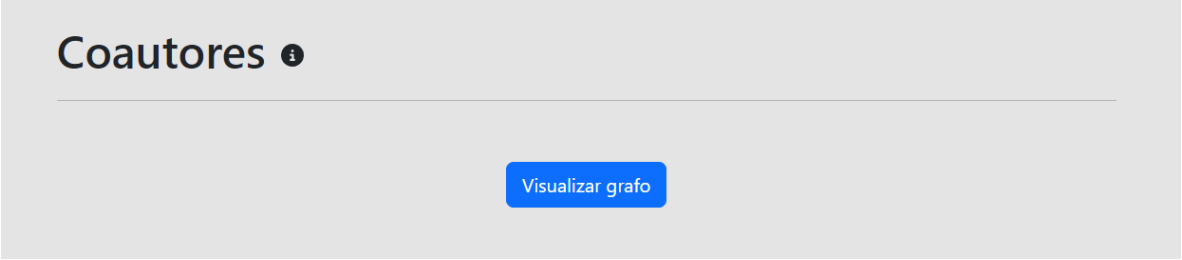


FIGURA 63 Sección de coautores del perfil de autor

2.10.3.2. IMPLEMENTAR UNA CAPA DE ABSTRACCIÓN DE D3 EN ANGULAR

Para utilizar D3 o cualquier otra librería dentro de un framework tan robusto como Angular de manera correcta, se debe interactuar con ella mediante una capa de abstracción o interfaz a través de clases, servicios y directivas. De esta manera, se separan las funciones principales que se están introduciendo en los componentes que las implementan, lo que proporciona una estructura de aplicación más flexible y escalable, y permite aislar los errores en su entorno adecuado [32]. A continuación, se presentan los elementos creados para dicha interacción:

- **Directives:** guiarán a los elementos sobre cómo implementar los comportamientos de D3 como ampliar o arrastrar.
- **Models:** son clases que representan los datos (grafo, nodo y arista) y proporcionan seguridad en la escritura.

- **d3.service.ts:** servicio que brindará funcionalidades a los modelos y directivas de D3.
- **Visuals:** componentes que implementan las directivas y modelos para mostrar el grafo dirigido por fuerza de D3.

2.10.3.3. INTERFAZ DE RESULTADOS DE LA BÚSQUEDA DE AUTORES RELEVANTES

La búsqueda de autores relevantes por keyword(s) se halla en la interfaz principal. Está compuesta por el componente Search, el grafo que contiene los autores más relevantes y los filtros para autores y afiliaciones (Figura 63). Todos los componentes de los resultados de esta búsqueda son renderizados en la plantilla "Home".

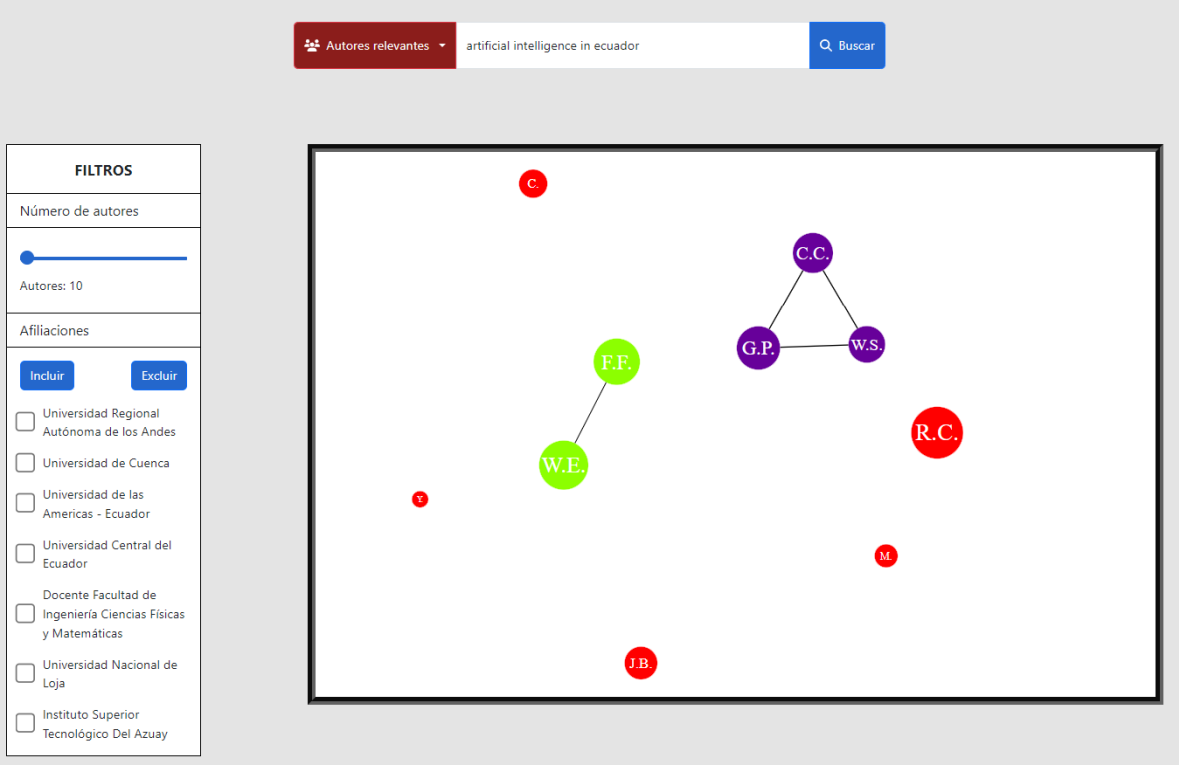


FIGURA 64 Interfaz de resultados de búsqueda de autores relevantes

2.10.3.4. LA INTERFAZ DE RESULTADOS DE LA BÚSQUEDA DE ARTÍCULOS RELEVANTES

Al igual que las otras dos búsquedas, esta también se halla en la plantilla “home”. La interfaz está compuesta por el componente Search, el filtro de año de publicación y la tabla paginada donde se presentan los artículos más relevantes (Figura 64). En esta interfaz también se encuentra el modal para visualizar a detalle la información de un artículo.

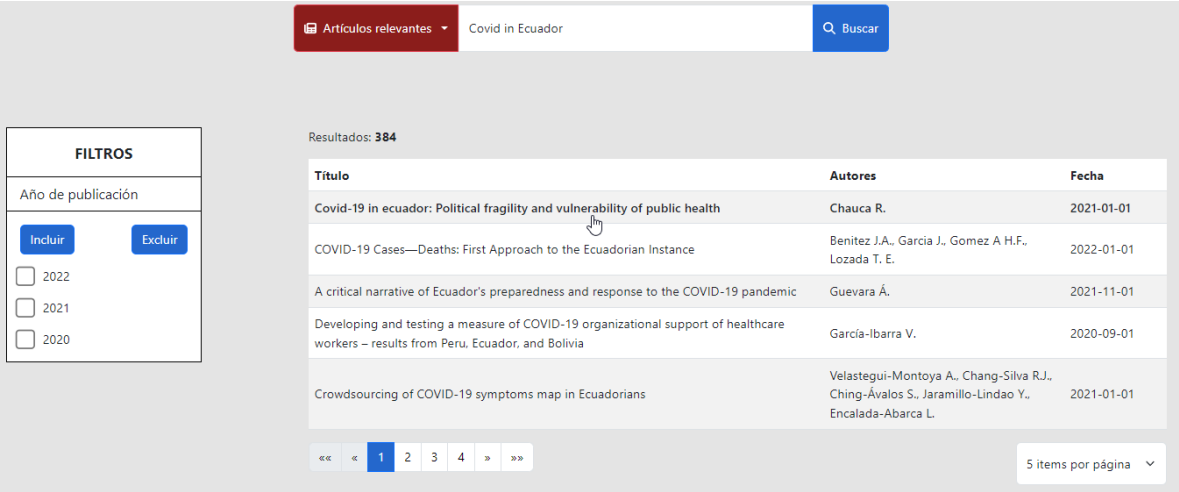


FIGURA 65 Interfaz de resultados de búsqueda de artículos relevantes

2.10.4. REVISIÓN Y RESTROSPECTIVA DEL SPRINT

Se realizó la revisión del Sprint 5 junto con todo el Scrum Team en base a los criterios de aceptación de cada una de las historias de usuario. Los objetivos de este sprint culminaron satisfactoriamente.

- Se desarrolló la interfaz de resultados de búsqueda de autor.
- Se implementó una capa de abstracción de D3 en Angular.

- Se desarrolló la interfaz de resultados de la búsqueda de autores relevantes por keyword(s).
- Se desarrolló la interfaz de resultados de la búsqueda de artículos relevantes por keyword(s).

CAPÍTULO 3. RESULTADOS Y DISCUSIÓN

3.1. PRODUCTO FINAL

La aplicación está compuesta por 3 pantallas. La primera pantalla es la interfaz principal donde se puede realizar los 3 tipos de búsqueda: búsqueda de autor, búsqueda de autores relevantes por keywords y búsqueda de artículos relevantes por keywords. En la segunda pantalla se muestra la información de un autor. Esta pantalla está relacionada con la búsqueda de autor. Finalmente, se encuentra la pantalla “acerca de”, que muestra información sobre la aplicación y su propósito.

3.1.1. BÚSQUEDA DE AUTOR

La búsqueda de un autor inicia en la pantalla principal. Únicamente es necesario seleccionar la opción “autor” en el componente de búsqueda (la opción autor está seleccionada por defecto) y escribir el nombre, apellido, nombre autor, iniciales y/o el identificador de scopus del autor, como se observa en la Figura 66.

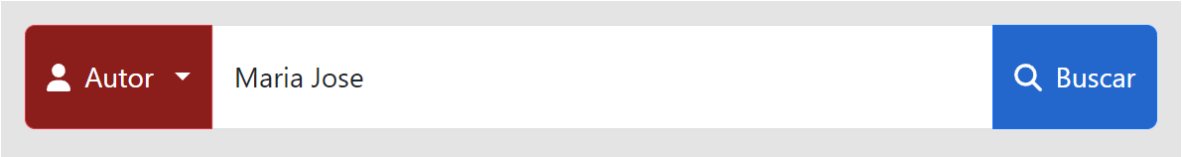


FIGURA 66 Ejemplo de búsqueda de autor

Cuando la búsqueda finalice, los resultados se presentarán en una tabla con paginación y con la opción de cambiar el número de resultados por página (Figura 67). La tabla está compuesta por las siguientes columnas:

- **Nombres:** Nombre y apellido, nombre de autor e iniciales.
- **Afiliaciones:** Afiliación actual e históricas del autor.

- **Artículos:** Número de artículos del autor.
- **Tópicos:** Tópicos relacionados con el autor ordenados por relevancia.

FIGURA 67 Ejemplo de resultados de búsqueda de autor

Una vez encontrado al autor requerido, se puede hacer clic en él para acceder a la pantalla de perfil de autor, donde se muestra la información detallada correspondiente a dicho autor, como se observa en la Figura 68. La pantalla de perfil de autor está divida en las siguientes secciones:

- **Información general:** Nombres, afiliaciones y enlace al perfil del autor en la plataforma de Scopus.
- **Artículos:** Artículos del autor. Al hacer clic sobre cualquier artículo se desplegará el modal que muestra información detallada de este.
- **Tópicos:** Tópicos relacionados con el autor ordenados por relevancia. Al hacer clic sobre cualquier tópico será redireccionado a la pantalla de búsqueda y se realizará la búsqueda de autores relevantes del tópico seleccionado.
- **Coautores:** Grafo interactivo de coautoría.

FIGURA 68 Ejemplo de perfil de autor

3.1.2. BÚSQUEDA DE AUTORES RELEVANTES POR KEYWORDS

Para realizar esta búsqueda primero se debe acceder a la pantalla principal. A continuación, se selecciona la opción “autores relevantes” en el componente de búsqueda y se escribe el tópico en el recuadro, como se observa en la Figura 69.

FIGURA 69 Ejemplo de búsqueda de autores relevantes

Al finalizar la búsqueda se presentarán los resultados en un grafo interactivo que tendrá a los autores más relevantes y las relaciones entre sí. Además de poder filtrar por número de autores y afiliaciones (Figura 70).

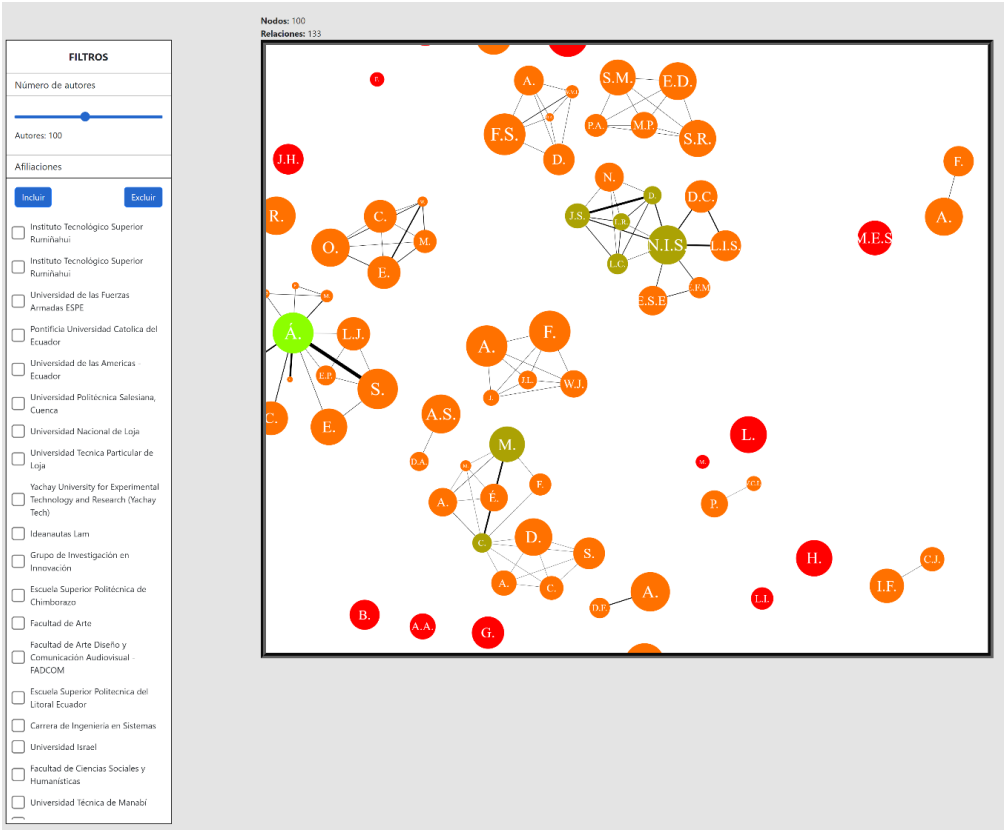


FIGURA 70 Ejemplo de resultados de búsqueda de autores relevantes

3.1.3. BÚSQUEDA DE ARTÍCULOS RELEVANTES POR KEYWORDS

En la pantalla principal, es necesario elegir la opción "Artículos relevantes" que se encuentra en el componente de búsqueda y luego ingresar el tema del cual se desea obtener los artículos más importantes, tal como se muestra en la Figura 71.

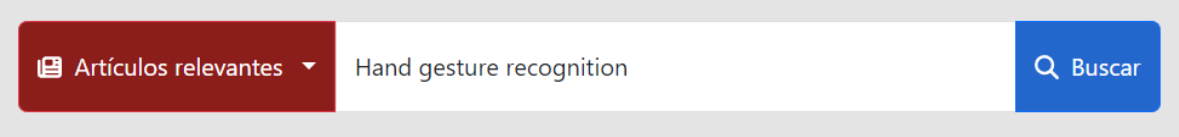


FIGURA 71 Ejemplo de búsqueda de artículos relevantes

Después de completar la búsqueda, los resultados se presentarán en una tabla que tiene la opción de paginación, así como también la capacidad de filtrar los resultados según la fecha de publicación del artículo (Figura 72). La tabla está compuesta por las siguientes columnas: título del artículo, autores y fecha de publicación. Además, se puede hacer clic sobre cualquier artículo para visualizar a detalle la información de este en un modal, como se observa en la Figura 73.

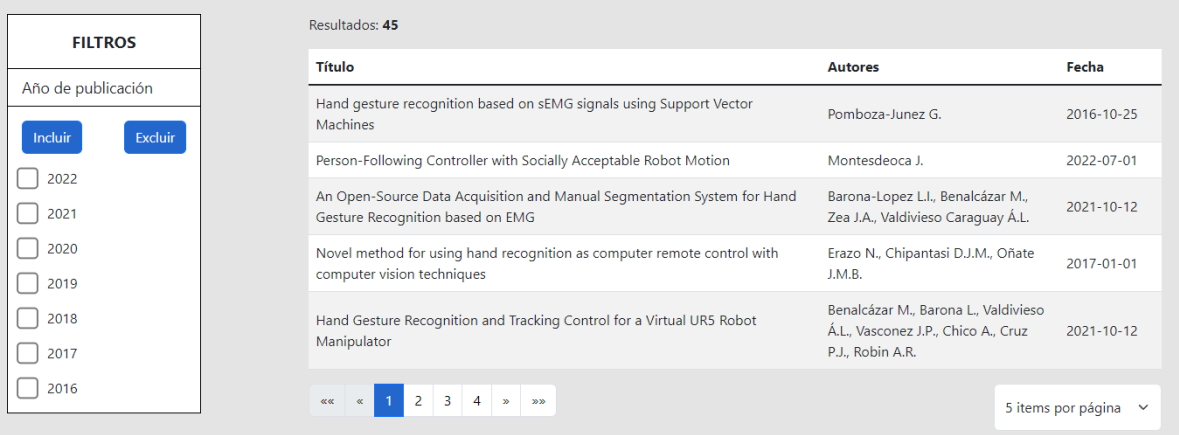


FIGURA 72 Ejemplo de resultados de búsqueda de artículos relevantes

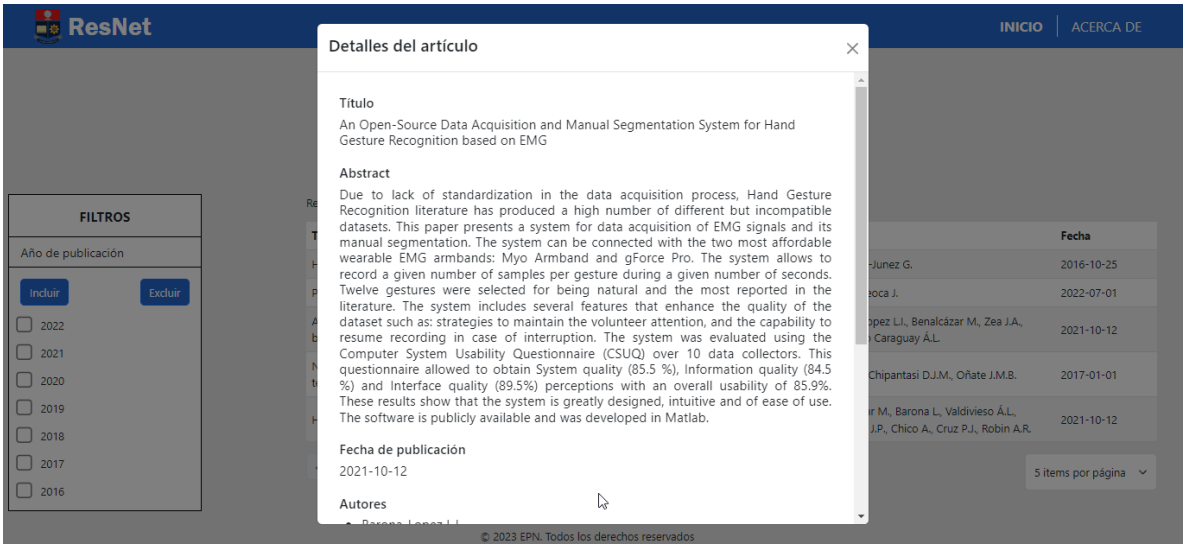


FIGURA 73 Ejemplo del modal de detalles de artículo

3.2. PRUEBAS CON USUARIOS FINALES

Para evaluar la aceptación del sistema ResNet, se desarrolló una encuesta basada en el modelo TAM (Technology Acceptance Model, por sus siglas en inglés). El modelo TAM proporciona una metodología efectiva para medir la aceptación de un sistema al enfocarse en dos aspectos clave: la facilidad de uso y la utilidad percibida. La facilidad de uso se refiere a la comodidad y simplicidad que los usuarios perciben al interactuar con el sistema ResNet. Por otro lado, la utilidad percibida se centra en cómo los usuarios valoran los beneficios y ventajas que el sistema ResNet ofrece.

La encuesta fue alojada en Google Forms y constó de 6 preguntas distribuidas en los 3 tipos de búsqueda proporcionados por ResNet. Cada tipo de búsqueda incluyó una pregunta para evaluar la facilidad de uso y otra para medir la utilidad percibida. En la Tabla 3.1 se detallan las preguntas realizadas.

Tabla 3.1 Preguntas de la encuesta de aceptación

Nº	Preguntas
Búsqueda de Autor	
1	¿Fue fácil realizar la búsqueda de un autor?
2	¿Fue útil la información presentada en el perfil del autor?
Búsqueda de Autores Relevantes	
3	¿El grafo de autores relevantes fue claro y fácil de entender?
4	¿Fue útil la información presentada en la búsqueda de autores relevantes?
Búsqueda de Artículos Relevantes	
5	¿La tabla de resultados de la búsqueda de artículos relevantes fue clara y fácil de entender?
6	¿Fue útil la información presentada en la búsqueda de artículos relevantes?

Para analizar los resultados de la evaluación de la aceptación del sistema ResNet, se consideraron las respuestas de un total de 9 participantes, entre estudiantes de maestría y profesores de la Facultad de Ingeniería en Sistemas. Las preguntas se calificaron en una escala del 1 al 5, donde una puntuación más alta indica una mayor facilidad de uso o utilidad percibida.

3.2.1. FACILIDAD DE USO

En la Figura 75, se presentan los resultados promedio de las preguntas que miden la facilidad de uso. La pregunta 1 obtuvo una calificación promedio de 4.67, 4.11 en la Pregunta 3 y 4.56 en la Pregunta 5. Estos resultados refuerzan la percepción general de los usuarios sobre la comodidad y simplicidad que experimentan al utilizar el sistema ResNet.

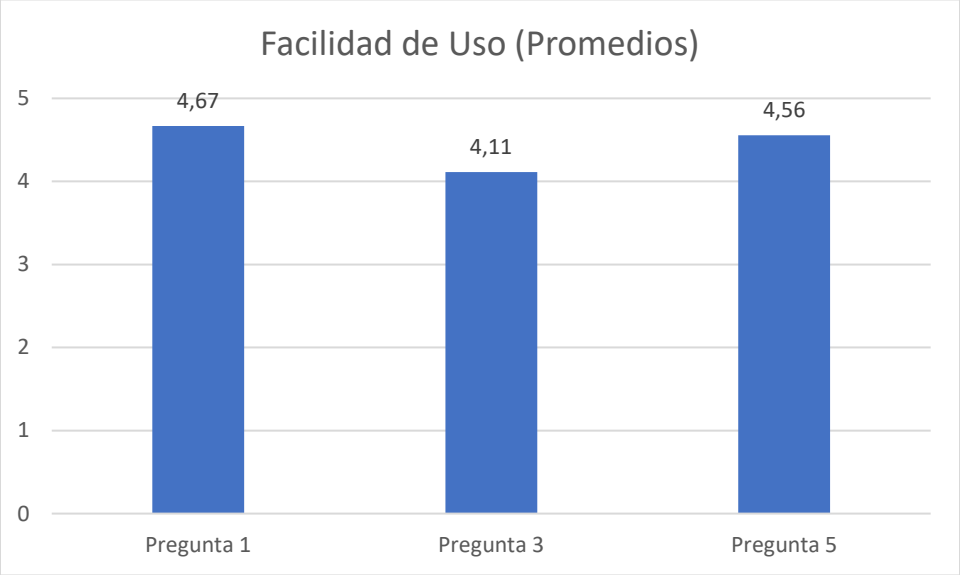


FIGURA 74 Resultados promedio de Facilidad de Uso

Es importante destacar que uno de los comentarios recibidos resaltó expresamente la amigabilidad de la herramienta. Este comentario refuerza aún más la idea de que los usuarios consideran que la aplicación ResNet es fácil de usar.

3.2.2. UTILIDAD PERCIBIDA

En la Figura 76, se muestran los promedios de las preguntas que miden la utilidad percibida. Las puntuaciones registradas fueron de 4.33 para la Pregunta 2, 4.89 para la Pregunta 4 y 4.78 para la Pregunta 6. Estas puntuaciones indican que los usuarios reconocen y valoran la utilidad que ofrece ResNet en diversas áreas, como la búsqueda de colaboradores, la formación de redes de investigadores y el análisis de temas de investigación.

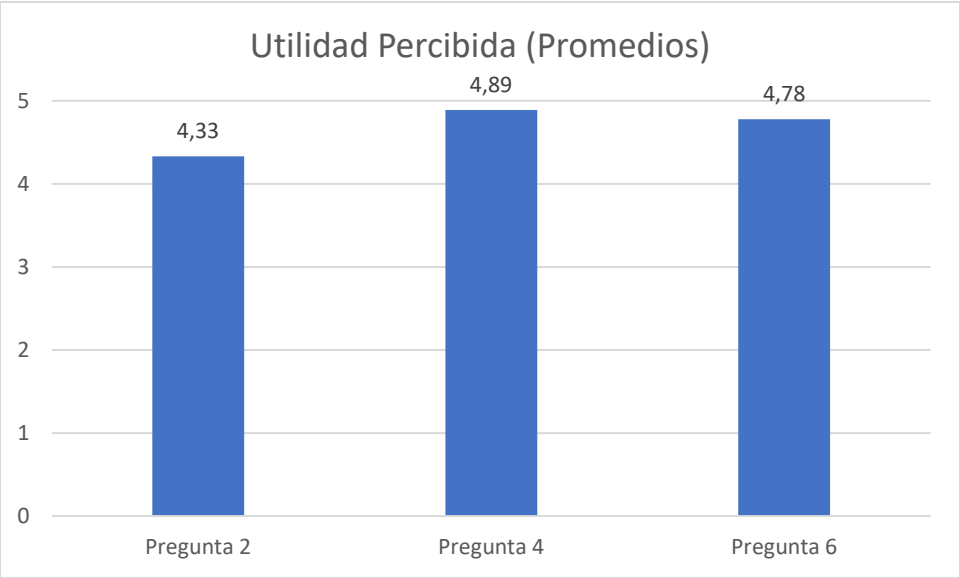


FIGURA 75 Resultados promedio de la Utilidad Percibida

Un participante expresó el deseo de incluir el DOI o la URL del artículo como una mejora potencial, resaltando una posible funcionalidad adicional que podría mejorar aún más la utilidad percibida de la aplicación. Además, otro comentario enfatizó que la información expuesta en la aplicación es considerada adecuada.

CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- Se desarrolló una aplicación web altamente eficiente en Ecuador, que facilita la formación de redes de investigadores, búsqueda de colaboradores y análisis de temas de investigación.
- La metodología Scrum se implementó eficientemente, permitiendo una organización y planificación efectivas. Brindó una estructura ágil que facilitó la comunicación y adaptabilidad. Su enfoque iterativo y colaborativo mejoró la productividad y el cumplimiento de objetivos en el proyecto.
- La metodología CRISP-DM garantizó una estructura sólida en el proceso de minería de datos, seleccionando y aplicando los modelos TF-IDF para la búsqueda y análisis de información. Su enfoque riguroso y sistemático contribuyó al éxito y la calidad de los resultados obtenidos en el proyecto.
- La aplicación de la técnica TF-IDF en el modelamiento de texto desempeña un papel fundamental en las aplicaciones de Recuperación de Información. En ResNet, el uso de TF-IDF demostró ser altamente efectivo para mejorar la precisión y relevancia de los resultados obtenidos en la búsqueda de autores y artículos relevantes.
- El uso de Notebooks de Python en el proceso de minería de datos proporcionó beneficios significativos. La versatilidad y facilidad de uso de Python permitieron una implementación eficiente de algoritmos y análisis de datos. Los Notebooks ofrecieron un entorno interactivo para explorar y visualizar resultados, mejorando la comprensión y la toma de decisiones.
- La arquitectura de tres capas, compuesta por la base de datos orientada a grafos Neo4j, el backend desarrollado en Flask y el frontend en Angular, demostró ser altamente efectiva. Estas tecnologías proporcionaron un almacenamiento eficiente de datos en grafo, una lógica de negocio robusta,

una interfaz de usuario moderna e interactiva, y la capacidad de generar gráficos dinámicos.

- Los resultados de la encuesta basada en el modelo TAM demuestran que el sistema ResNet ha sido ampliamente aceptado por los usuarios, quienes perciben un alto nivel de facilidad de uso y utilidad en la aplicación. Con un resultado promedio general de 4.56 sobre 5, los participantes expresaron su satisfacción con la experiencia proporcionada por el sistema ResNet.

4.2. RECOMENDACIONES

- Se sugiere realizar evaluaciones periódicas de rendimiento y usabilidad de la aplicación, con el objetivo de identificar posibles áreas de mejora y optimizar continuamente su funcionamiento. Esto ayudará a garantizar que la aplicación se mantenga eficiente y alineada con las necesidades de los usuarios.
- Se propone desarrollar tutoriales en formato de video para brindar a los usuarios una guía visual práctica y comprensible sobre el uso de las diferentes funcionalidades de la aplicación. Estos recursos audiovisuales serán de gran ayuda para facilitar la familiarización y el máximo aprovechamiento de las características y herramientas disponibles de la aplicación.
- La información de la aplicación requiere actualizaciones periódicas, ya que los datos extraídos tienen como fecha límite el 13 de julio de 2022. Para abordar este desafío, se recomienda desarrollar una serie de scripts basados en los notebooks utilizados en el proceso de minería de datos. Estos scripts permitirán extraer nuevos datos de Scopus utilizando las APIs proporcionadas por Elsevier, asegurando así que la información se mantenga actualizada y refleje los avances más recientes en la investigación científica.
- Se recomienda mantener actualizados los modelos TF-IDF generados. En particular, para el modelo de tópicos-autores, es conveniente emplear una

nueva forma de normalización para abordar el sesgo aún existente en los documentos extensos y variados del corpus. Esto garantizará que los modelos estén alineados con los cambios en la base de datos y proporcionen resultados más precisos y actualizados.

- Al inicio del proyecto, no existía ninguna herramienta similar a ResNet en Ecuador, pero actualmente CEDIA ha desarrollado la herramienta REDI, que ofrece características similares. Sin embargo, ResNet ofrece ventajas distintivas, como una mayor eficiencia en los resultados y una interfaz más intuitiva. Se sugiere evaluar cuidadosamente las ventajas y funcionalidades de ambas herramientas, considerando posibles colaboraciones o integraciones para maximizar los beneficios y ofrecer una experiencia aún más completa a los usuarios e investigadores.
- Se sugiere implementar una función de búsqueda de afiliaciones por tópico en la aplicación. Esta funcionalidad permitirá a los usuarios ingresar un tema de interés y visualizar un grafo que representa las conexiones entre las diferentes afiliaciones relacionadas con ese tema. Esto facilitará la identificación de colaboradores potenciales y fortalecerá la capacidad de establecer vínculos académicos en áreas específicas de investigación.

REFERENCIAS BIBLIOGRÁFICAS

1. J. Adams, "The rise of research networks", Nature, vol. 490 No. 1, pp. 335–336, 2012.
2. J. Liu, T. Tang, W. Wang, B. Xu, X. Kong and F. Xia, "A Survey of Scholarly Data Visualization," in IEEE Access, vol. 6, pp. 19205-19221, 2018.
3. S. Kumar, "Co-authorship networks: a review of the literature", Aslib Journal of Information Management, vol. 67, no. 1, pp. 55-73, 2015.
4. M. Newman, "Coauthorship networks and patterns of scientific collaboration", PNAS, vol. 101, pp. 5200-5205, 2004.
5. A. Rodríguez, K. Bonilla and A. Rodríguez, "Publicar Desde América Latina. ¿En dónde estamos?", Revista Ecuatoriana de Neurología, vol. 27, no. 3, 2018.
6. SENESCYT, "Rendición de cuentas 2019", 2020.
7. R. Wirth y J. Hipp, “CRISP-DM: Towards a Standard Process Model for Data Mining”, p. 11.
8. M. Newman, “Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality”, Physical review E, vol. 64, no. 1, 2001.
9. S. Qaiser y R. Ali, “Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents”, International Journal of Computer Applications, vol. 181, no. 1, 2018.
10. J. Ramos, “Using TF-IDF to Determine Word Relevance in Document Queries”, Proceedings of the first instructional conference on machine learning, vol. 242, no. 1, 2003.
11. G.F. Kahn, “Seven Layers of Social Media Analytics: Mining Business Insights from Social Media Text, Actions, Networks, Hyperlinks, Apps, Search Engine, and Location Data”, pp. 21-23, 2015
12. K. Schwaber and J. Sutherland, “La Guía de Scrum”, Scrumguides.org, 2020.
13. CEDIA, "SOBRE NOSOTROS", Cedia.edu.ec, 2021. [En línea]. Disponible: https://www.cedia.edu.ec/es/sobre-nosotros.

14. Elsevier, “Elsevier Developer Portal”, 2021. [En línea]. Disponible: https://dev.elsevier.com/.
15. “Node.js”, 2021. [En línea]. Disponible: https://nodejs.org/en/.
16. J. Miller, “Graph Database Applications and Concepts with Neo4j”, Proceedings of the southern association for information systems conference, vol. 2324, no. 36, 2013.
17. Amazon Web Services, “What Is a Graph Database?”, 2021. [En línea]. Disponible: https://aws.amazon.com/es/nosql/graph/.
18. Neo4j, “What is a Graph Database?”, 2021. [En línea]. Disponible: https://neo4j.com/developer/graph-database/.
19. Angular, “What is Angular?”, 2022. [En línea]. Disponible: https://angular.io/guide/what-is-angular.
20. Bootstrap, “Get started with Bootstrap”, 2022. [En línea]. Disponible: https://getbootstrap.com/docs/5.2/getting-started/introduction/.
21. D3.js, “D3 Home”, 2021. [En línea]. Disponible: https://github.com/d3/d3/wiki.
22. Pallets Projects, “Flask”, 2022. [En línea]. Disponible: https://palletsprojects.com/p/flask/.
23. PyData, “pandas”, 2022. [En línea]. Disponible: https://pandas.pydata.org/.
24. Microsoft, “What is Azure Boards?”, 2022. [En línea]. Disponible: https://learn.microsoft.com/en-us/azure/devops/boards/get-started/what-is-azure-boards.
25. Elsevier, “About Scopus”, 2021. [En línea]. Disponible: https://www.elsevier.com/solutions/scopus.
26. Elsevier, “About ScienceDirect”, 2021. [En línea]. Disponible: https://www.elsevier.com/solutions/sciencedirect.
27. C. J. Chapman, et al, “CRISP-DM 1.0: Step-by-step data mining guide”, SPSS inc, 2000, vol. 9, no 13.
28. Python Software Foundation, “pickle - Python object serialization”, 2022. [En línea]. Disponible: https://docs.python.org/3/library/pickle.html.

29. scikit-learn, “TfidfVectorizer”, 2022. [En línea]. Disponible: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
30. SciPy, “Sparse matrices (scipy.sparse)”, 2022. [En línea]. Disponible: https://docs.scipy.org/doc/scipy/reference/sparse.html.
31. C. D. Manning, P. Raghavan, H. Schütze, “Introduction to Information Retrieval”, Cambridge University Press, 2008. [En línea]. Disponible: https://nlp.stanford.edu/IR-book/.
32. L. Sharir, “Visualizing Data with Angular and D3”, 2023. [En línea]. Disponible: https://medium.com/netscape/visualizing-data-with-angular-and-d3-209dde784aeb

ANEXOS

ANEXO 1: NOTEBOOKS PARA MINERÍA DE DATOS

Enlace al repositorio: https://github.com/jozuenikolas/resnet-data-mining

ANEXO 2: BACKEND DE RESNET

Enlace al repositorio: https://github.com/jozuenikolas/resnet-backend

ANEXO 3: FRONTEND DE RESNET

Enlace al repositorio: https://github.com/jozuenikolas/resnet-frontend

ANEXO 4: PROTOTIPOS DE LA INTERFAZ GRÁFICA

Enlace a Figma: https://shorturl.at/kxDOS

ANEXO 5: RESULTADOS DE LA EVALUACIÓN TAM

Enlace al documento: TAM Results.xlsx