UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Centro de Innovación y Desarrollo de Internet

Módulo de procesamiento de contenidos aplicado a la Plataforma de Contenidos Unificados para Búsqueda Avanzada

Trabajo final presentado en opción al título de Máster en Informática Avanzada

Autor: Ing. Leiny Amel Pons Flores, Prof. Instructor

Tutores: Dra.C. Vivian Estrada Sentí, Prof. Titular
Dr.C. José Felipe Ramírez Pérez, Prof. Auxiliar

Ciudad de La Habana, noviembre de 2018

PENSAMIENTO

Nuestra juventud debe procurar adquirir aquellos conocimientos que sean más útiles en cada momento a la nación. Sobre todo, si se tiene en cuenta que estamos entrando en una etapa enteramente nueva.

Fidel Castro Ruz

DECLARACIÓN DE AUTORÍA

Declaro por este medio que yo, Leiny Amel Pons Flores, con carné de identidad 90031438589, soy el autor principal del trabajo final de maestría “Módulo de procesamiento de contenidos aplicado a la Plataforma de Contenidos Unificados para Búsqueda Avanzada”, desarrollado como parte de la Maestría en Informática Avanzada y que autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo. Y para que así conste, firmo la presente declaración jurada de autoría en Ciudad de La Habana a los ____ días del mes de noviembre del año 2018.

Ing. Leiny Amel Pons Flores

_____

Firma del Autor

Dra.C Vivian Estrada Sentí

_____

Firma del Tutor

Dr.C José Felipe Ramírez Pérez

_____

Firma del Tutor

RESUMEN

En la actualidad el desarrollo acelerado y exponencial de Internet genera un gran cúmulo de información y contenido que se almacena en la web y que es accesible a través de los motores de búsqueda. La información almacenada puede ser analizada y utilizada para mejorar la respuesta que los sistemas de recuperación de información brindan. Un análisis de la información basado en la aplicación de técnicas de minería web como la minería de contenido y de utilización, brindaría la posibilidad de utilizar nuevos conocimientos para mejorar la recuperación de información.

De forma conjunta, la categorización de los documentos almacenados, de las consultas de usuarios, el cálculo de la similitud entre consultas y el uso de algoritmos de re-ranking (acto de clasificar algo de nuevo o de manera diferente) son potentes herramientas que aproximan la necesidad de información de los usuarios al conjunto de respuesta de los motores de búsqueda. La presente investigación propone y describe la aplicación de un módulo de procesamiento de contenidos a la Plataforma de Contenidos Unificados para Búsqueda Avanzada, a partir de la implementación de un módulo que perfecciona su funcionamiento. También detalla la arquitectura que debe utilizar la solución y se exponen los resultados de la validación de la propuesta.

Como resultado de la investigación se obtiene un módulo de procesamiento de contenidos aplicado a la Plataforma de Contenidos Unificados para Búsqueda Avanzada que clasifica documentos y consultas haciendo uso del algoritmo de categorización Naive Bayes, realiza cálculo de similitud entre consultas y documentos y utiliza perfiles de búsqueda de usuarios, mejorando los resultados de la plataforma.

Palabras clave: algoritmos de *re-ranking*, categorización de consultas, categorización de documentos, minería web, sistemas de recuperación de información.

ABSTRACT

At present, the accelerated and exponential development of the Internet generates a large amount of information and content that is stored on the web and that is accessible through the search engines. The stored information can be analyzed and used to improve the response that information retrieval systems provide. An analysis of the information based on the application of web mining techniques such as content mining and utilization would offer the possibility of using new knowledge to improve information retrieval.

Together, the categorization of stored documents, of user queries, the calculation of the similarity between queries and the use of re-ranking algorithms are powerful tools that approximate the need for information from users to the response set of users the search engines. This research proposes and describes the application of a content processing module to the Unified Content Platform for Advanced Search, based on the implementation of a module that improves its functioning. It also details the architecture that the solution should use and the results of the validation of the proposal are exposed.

As a result of the research, we obtain a content processing module applied to the Unified Content Platform for Advanced Search that classifies documents and queries using the Naive Bayes categorization algorithm, performs similarity calculation between queries and documents and uses search profiles of users, improving the results of the platform.

Keywords: re-ranking algorithms, query categorization, document categorization, web mining, information retrieval systems.

INDICE

INTRODUCCIÓN

Las tecnologías de la información y la comunicación (TIC) son en la actualidad un pilar elemental de las actividades cotidianas de la sociedad en casi todas sus ramas. Según Romaní (2011), las TIC son aquellos dispositivos tecnológicos tanto de hardware como de software que permiten ejecutar varias acciones como editar, producir, almacenar, intercambiar y transmitir datos entre diferentes sistemas de información que cuentan con protocolos comunes. La revolución científico técnica ha impactado con fuerza en las TIC provocando el desarrollo acelerado y exponencial de Internet, red de redes de ordenadores, generando en su conjunto un gran cúmulo de información y contenido que se almacena en la web.

Para Rodríguez Rueda y Hidalgo Delgado (2012), y a juicio de este autor, la distinción entre información y documentos es importante. Siempre se necesita información, pero a veces se busca más directamente la información misma donde o como aparezca, provocando que se acceda a grandes e inservibles volúmenes de datos. Otras veces buscamos documentos concretos que suponemos la contiene o que nos interesa en particular conocer.

Lograr encontrar la información que se desea con exactitud puede resultar muy complicado teniendo en cuenta que actualmente existen en internet más de 1 100 000 000 sitios web (Internet Live Stats, 2018). Para responder a las necesidades de información de los usuarios surgen los denominados sistemas de recuperación de información (SRI en adelante), conocidos también como motores de búsqueda.

Las definiciones de motores de búsqueda son numerosas. A juicio de Tolosa y Bordignon (2008), estas aplicaciones manejan grandes bases de datos de referencias a páginas o recursos web recopiladas por medio de un proceso automático, sin intervención humana. Uno o varios agentes de búsqueda recorren la web, a partir de una relación de direcciones iniciales y recopilan nuevas direcciones generando una serie de etiquetas que permiten su indexación y almacenamiento en la base de datos.

Los SRI son robustas aplicaciones que poseen grandes bases de datos tan amplias y abarcadoras que provocan elevados costes para la preservación de recursos digitales. Sería inteligente entonces, utilizar todo ese contenido y explotarlo de forma estratégica para generar conocimientos que mejoren los resultados brindados a los usuarios. Las arquitecturas públicas de los SRI más utilizados no explican cómo procesan la información y generan el nuevo conocimiento. Tampoco explican el funcionamiento de sus componentes importantes ni la implementación de sus algoritmos principales. Autores como Ortega Maldonado et al. (2017) afirman que este hecho supone un problema, debido a que la falta de evidencia de este conocimiento se convierte en un obstáculo significativo para el procesamiento y extracción de estos datos en SRI de nueva implementación, lo cual ocasiona que los desarrolladores y administradores de estos sistemas no posean una guía

formal o de estudio para el desarrollo y administración de estos componentes en el SRI del que son responsables.

Este autor considera que un SRI que no analiza los contenidos que posee para generar nuevos conocimientos, deja de contribuir a su propia optimización. La categorización de documentos recuperados y consultas de usuarios es una herramienta indispensable cuando hablamos de analizar y procesar contenidos. Las consecuencias de no categorizar documentos y consultas han sido abordadas por diversos autores, al señalar que no se automatiza este importante proceso ni se obtienen resultados, en términos de exactitud, comparables con los de los expertos humanos (Neyra, 2016; Martínez Albuerne, 2008). Tampoco se logra un mejor entendimiento de las intenciones del usuario a juicio de Valencia et al. (2009), ni se personaliza la respuesta que se le ofrece, lo cual es una desventaja porque no acerca los resultados a las necesidades de información.

Plantea Rodríguez (2016) que el modelado de perfiles de usuario es una técnica para la recomendación de contenido personalizado. Sin la generación de los perfiles de usuarios, los cuales son la representación interna de sus necesidades de búsqueda, se agrava el problema fundamental de la recuperación de información. Esto afecta directamente el conjunto de respuesta que se brinda, debido a que no se adaptan los resultados al usuario ni ponderan aquellos documentos que podrían tener un mayor significado (Rodríguez et al., 2016). Además, al no implementar el cálculo de la similitud entre consultas, se dificulta devolver a los nuevos usuarios respuestas más acercadas a sus necesidades. Esta situación se manifiesta puesto que es imposible comparar la similitud entre la consulta de un nuevo usuario que tiene un perfil relativamente pobre, con la de otros usuarios con mayor experiencia de uso.

La Plataforma de Contenidos Unificados para Búsqueda Avanzada utiliza la tecnología de búsqueda Orión. Esta tecnología carece de elementos tan importantes como la categorización de documentos, categorización de consultas y el cálculo de similitud entre consultas. La situación anterior provoca que al utilizar la Plataforma no se obtengan los mejores resultados y se presenten todos los inconvenientes explicados anteriormente. En la tabla confeccionada por Leyva et al. (2018), se muestran los valores obtenidos de precisión y exhaustividad al realizar un experimento sobre la tecnología Orión. Es necesario señalar que, a mayor aproximación al valor uno, los resultados son mejores.

Tabla 1. Valores de precisión y exhaustividad de la tecnología Orión. Fuente: (Leyva et al., 2018)

Precisión de la tecnología Orión	Exhaustividad de la tecnología Orión
0,2964	0,2468

Por la importancia que la dirección política del país le ha dado a este proyecto, se decidió realizar un diagnóstico preliminar aplicando una entrevista. El objetivo era evaluar, desde la perspectiva de los desarrolladores y especialistas, el funcionamiento del motor de búsqueda. Como resultado de la encuesta aplicada se detectaron insuficiencias relacionadas con los indicadores exhaustividad y precisión. La encuesta fue dirigida a especialistas con más de tres años de experiencia que han

formado parte del desarrollo del motor de búsqueda Orión y de la Plataforma C.U.B.A. Se escogió un grupo de 20 personas conformado por directivos, líderes de proyecto, programadores, arquitectos y especialistas. Los resultados obtenidos de la entrevista realizada se presentan a continuación:

1. El 65% de los encuestados considera que puede mejorarse el ajuste de los resultados a las necesidades de búsqueda del usuario en la Plataforma C.U.BA.
2. El 60% de los encuestados considera que la Plataforma C.U.BA. recupera documentos sin relevancia para la búsqueda.
3. El 85% de los encuestados considera que la Plataforma C.U.BA. no brinda resultados personalizados para cada usuario.
4. El 70% de los encuestados considera que puede mejorarse la ponderación del conjunto de respuesta que brinda la Plataforma C.U.BA.
5. El 90% de los encuestados considera que la Plataforma C.U.BA. necesita incorporar nuevas herramientas para mejorar el proceso de recuperación de información.

La problemática expuesta podría resumirse en que actualmente no existe una solución que procese los contenidos almacenados y genere el conocimiento necesario para mejorar el funcionamiento del motor de búsqueda elevando la eficacia de la recuperación de información.

Se impone la búsqueda de una solución para la problemática descrita anteriormente, que conlleva al planteamiento del siguiente **problema científico** ¿Cómo mejorar los resultados de búsqueda en la Plataforma de Contenidos Unificados para Búsqueda Avanzada?

El problema anterior se enmarca dentro del **objeto de estudio** procesamiento de contenidos en sistemas de recuperación de información y en la solución del mismo se traza como **objetivo general** desarrollar un módulo de procesamiento de contenidos, utilizando técnicas de minería web, que permita mejorar los resultados de búsqueda en la Plataforma de Contenidos Unificados para Búsqueda Avanzada.

Para darle cumplimiento al objetivo general, se definen los siguientes **objetivos específicos**:

1- Realizar el marco teórico referencial de la presente investigación el cual responde al procesamiento de contenidos en SRI.

2- Analizar las herramientas y tecnologías indispensables para realizar el procesamiento de contenidos en SRI.

3- Implementar la solución propuesta para mejorar el proceso de recuperación de información.

4- Validar la solución desarrollada a partir de los métodos científicos definidos.

Mientras el **campo de acción** se centra en el procesamiento de contenidos en la Plataforma de Contenidos Unificados para Búsqueda Avanzada.

Se define como **hipótesis**: si se desarrolla y aplica un módulo de procesamiento de contenidos, utilizando técnicas de minería web, mejorará los resultados de búsqueda en la Plataforma de Contenidos Unificados para Búsqueda Avanzada.

Operacionalización de las variables

La presente investigación trabaja sobre 2 variables. El procesamiento de contenidos en la Plataforma de Contenidos Unificados para Búsqueda Avanzada es la **variable independiente.**
Definición conceptual: Solución informática que se encarga de realizar todas las acciones que conforman el procesamiento y análisis de los contenidos en la Plataforma de Contenidos Unificados para Búsqueda Avanzada.

Tabla 2. Variable independiente. Fuente: elaboración propia.

El procesamiento de contenidos en la Plataforma de Contenidos para Búsqueda Avanzada		
Dimensión	Indicadores	Unidad de medida
Procesamiento de contenidos en la Plataforma C.U.B.A.	Mejorar	Mejora
		No mejora
Aplicabilidad	Aplicación de la propuesta a un SRI	Aplicable
		No aplicable

Como **variable dependiente** se define los resultados de búsqueda de la recuperación de información.
Definición conceptual: Exactitud del conjunto de respuesta del sistema de recuperación de información.

Tabla 3. Variable dependiente. Fuente: elaboración propia.

Resultados de búsqueda de la reuperación de información		
Dimensión	Indicadores	Unidad de medida
Obtener tantos documentos relevantes como sea posible	Exhaustividad	Es exhaustivo
		No es exhaustivo
Utilidad del contenido del documento para el usuario	Precisión	Es preciso
		No es preciso

Indicadores para medir la exactitud:

- Exhaustividad: Obtener tantos documentos relevantes como sea posible (Quiñones Matesanz, 2015; Noya García, 2015; Baeza-Yates y Ribeiro-Neto, 1999; Jaramillo Valbuena y Londoño, 2014).
- Precisión: Este concepto está relacionado con la utilidad del documento para el usuario, de acuerdo a la necesidad de información original que guio su búsqueda, independientemente si es en parte o todo el documento (Tolosa y Bordignon, 2008).

Los valores de precisión y exhaustividad son inversamente proporcionales por lo que se debe medir el valor de estos indicadores por separado. Si el valor que se desea medir es la precisión, entonces el listado de documentos devueltos debe ser lo más pequeño y preciso posible. Si el valor que desea medirse es la exhaustividad, deben devolverse tantos documentos como sean necesarios para que el usuario tenga acceso a todos aquellos que sean relevantes.

Métodos de trabajo científico

Métodos teóricos:

- Analítico-Sintético: Permitió descomponer el problema de investigación en elementos por separados y profundizar en el estudio de cada uno de ellos. Mediante este método se procesó la información y sirvió para arribar a conclusiones en la investigación con respecto a las tendencias actuales de la gestión de perfiles de usuarios. Se empleó además para el análisis de los elementos esenciales referentes a la información relacionada con los Sistemas de Recuperación de Información y el procesamiento de contenidos.
- Histórico - Lógico: Se empleó con el propósito de constatar teóricamente cómo ha evolucionado en el tiempo el procesamiento de los contenidos en sistemas de recuperación de información, así como las herramientas y tecnologías utilizadas en el desarrollo de soluciones de este tipo.
- Hipotético-Deductivo: Este método permitió obtener la hipótesis planteada y a partir de ella derivar conclusiones en el transcurso de la investigación.
- Análisis documental: Permitió realizar un estudio de la bibliografía referente a la temática abordada.

Métodos empíricos:

- Experimentación: mediante experimentos se evaluó la capacidad de la solución propuesta para mejorar la eficacia de los resultados brindados a los usuarios.

Aportes prácticos:

- Solución informática para el procesamiento de contenidos en sistemas de recuperación de información.
- Definición del algoritmo de categorización utilizado para clasificar documentos recuperados y consultas de usuarios.
- Fórmula de similitud coseno modificada para ser utilizada en el cálculo de similitud entre consultas.

El documento que detalla la investigación se encuentra estructurado en resumen, introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas y anexos.

En el primer capítulo se realiza un análisis de los principales conceptos asociados a los sistemas de recuperación de información, el uso de la minería web para la categorización de documentos recuperados y consultas de usuarios. También se abordan otros tópicos como el cálculo de la similitud entre consultas de usuarios y algunos algoritmos de ranking (acto de clasificar algo) más utilizados.

El segundo capítulo describe los elementos esenciales de la propuesta de solución relacionados con el procesamiento de contenidos y mejora del proceso de recuperación de información.

En el tercer capítulo se valida la solución propuesta a través de la realización de un cuasi-experimento y se exponen los beneficios fundamentales cuando se tiene integrada la solución propuesta a la Plataforma de Contenidos Unificados para Búsqueda Avanzada.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

En el presente capítulo se pretende abordar las principales ideas y conceptos relacionados con la recuperación de información y los sistemas que trabajan en este proceso, así como lo relacionado al procesamiento de información y análisis de contenido. Del mismo modo se abordan las técnicas y herramientas utilizadas para categorizar consultas de usuarios y documentos recuperados, el cálculo de la similitud entre consultas y los algoritmos de ranking más utilizados.

1.1 La recuperación de información

La recuperación de información (RI en adelante) es un proceso que ha sido estudiado durante años por numerosos investigadores. Para varios autores, pioneros de dichos estudios, es la representación, el almacenamiento, la organización y el acceso a ítems de información (Baeza-Yates y Ribeiro-Neto, 1999).

El problema de la RI, según Martínez Méndez (2004), se define como "*dada una necesidad de información y un conjunto de documentos, ordenar los documentos de más a menos relevantes para esa necesidad y presentar un subconjunto de aquellos de mayor relevancia*". El mismo autor señala que la RI intenta resolver el problema de encontrar y ordenar documentos relevantes que satisfagan la necesidad de información de un usuario, expresada en un determinado lenguaje de consulta.

Para este autor, la RI se podría definir como la respuesta compuesta de documentos digitales que obtiene un usuario al realizar una consulta debido a una necesidad de conocimiento en cualquiera de los lenguajes soportados por los SRI.

La RI no es un área nueva, sino que se viene desarrollando desde finales de la década de 1950. Sin embargo, en la actualidad adquiere un rol más importante debido al valor que tiene la información para los diversos procesos de la sociedad. Se puede plantear entonces, que disponer o no de la información justa en tiempo y forma puede resultar en el éxito o fracaso de una operación. De aquí, que Tolosa y Bordignon (2008) destaquen la importancia de los SRI que pueden manejar – con ciertas limitaciones – estas situaciones de manera eficaz y eficiente.

1.1.1 Modelos de recuperación de información

Un modelo de RI es la especificación sobre cómo representar documentos y consultas y cómo compararlos. El objetivo de todo modelo es obtener un orden (ranking) de los documentos recuperados por el SRI, que refleje la relevancia de estos con la consulta del usuario (Epifanio Tula y Medeot, 2007). El diseño de un SRI se realiza bajo un modelo, donde según Villena Román (2004) queda definido "*cómo se obtienen las representaciones de los documentos y de la consulta, la estrategia para evaluar la relevancia de un documento respecto a una consulta y los métodos para establecer la importancia (orden) de los documentos de salida*".

Existen muchas alternativas de modelos para determinar la relevancia en el proceso de recuperación de información. Diversos autores han clasificado los modelos de recuperación de información entorno a varios criterios y grupos. Un parte de la bibliografía consultada separa los modelos en dos grupos: modelos clásicos y modelos estructurados.

En el grupo de los clásicos se consideran los modelos booleano, probabilístico y espacio vectorial, siendo este último el más difundido y utilizado por la comunidad. La complejidad de los modelos es muy distinta y variada, al igual que la precisión con la que seleccionan los resultados; mientras el modelo booleano sólo verifica la existencia o no de algún término en el documento, el vectorial basa la similitud en el ángulo comprendido entre los vectores que forman cada documento y la consulta (dichos vectores se forman por el peso de cada término en el documento o consulta) (Baeza-Yates y Ribeiro-Neto,1999).

En el modelo booleano, según Tolosa y Bordignon (2008), la representación de la colección de documentos se realiza sobre una matriz binaria documento–término, donde los términos han sido extraídos manualmente o automáticamente de los documentos y representan el contenido de los mismos. Las consultas se conforman con términos vinculados por operadores lógicos (*AND*, *OR*, *NOT*) y los resultados son referencias a documentos cuya representación satisface las restricciones lógicas de la expresión de búsqueda. En el modelo original no hay ranking de relevancia sobre el conjunto de respuestas a una consulta, todos los documentos poseen la misma relevancia.

Con el modelo vectorial, Epifanio Tula y Medeot (2007) comentan que cada documento es representado mediante un vector de *n* elementos, siendo *n* igual al número de términos indizables que existen en la colección documental. Hay, pues, un vector para cada documento, y, en cada vector, un elemento para cada término o palabra susceptible de aparecer en el documento. Cada uno de esos elementos es cubierto u ocupado con un valor numérico. Si la palabra no está presente en el documento, ese valor es igual a cero. En caso contrario, ese valor es calculado teniendo en cuenta diversos factores, dado que una palabra dada puede ser más o menos significativa. Este valor se conoce con el nombre de peso del término en el documento. En este modelo las consultas son representadas también mediante un vector de las mismas características que las de los documentos (variando los valores numéricos de cada elemento en función de las palabras que forman parte de la consulta, claro está). Esto permite calcular fácilmente una función de similitud dada entre el vector de una consulta y los de cada uno de los documentos. El resultado de dicho cálculo mide la semejanza entre la consulta y cada uno de los documentos, de manera que, aquéllos que, en teoría, se ajustan más a la consulta formulada, producen un índice más alto de similitud.

Las ventajas del modelo vectorial son varias. Para Urbano et al. (2010), entre las más importantes se encuentran:

- Mejores puntuaciones en experimentos, sobre todo con grandes colecciones.
- Existen variaciones que permiten, por ejemplo, relevance feedback (retroalimentación de relevancia).

- Tiene en cuenta tf/idf y longitud del documento.
- Grado de relevancia y matching (correspondencia) parcial.

El autor Comeche (2006) considera que el modelo probabilístico clásico supera al modelo booleano clásico en cuanto que el probabilístico efectúa equiparación parcial mientras que el modelo booleano clásico efectúa equiparación exacta. Sin embargo, ambos siguen presentado una característica negativa: ni el modelo booleano ni el modelo probabilístico tienen en cuenta la frecuencia con la que aparecen los términos de indización dentro de los documentos. Esta característica presente en el modelo vectorial lo convirtió en el utilizado en la tecnología Orión.

1.1.2 Sistemas de recuperación de información

El objetivo principal de la RI es satisfacer la necesidad de información planteada por un usuario a través de una consulta realizada en lenguaje natural, especificada utilizando un conjunto de palabras claves, también llamadas descriptores y así lo reconoce Hallo (2014). También comenta que, en general, este proceso hacia la recuperación de documentos relevantes a la consulta presentada, no es un proceso simple debido a la complejidad semántica del vocabulario.

Los SRI recuperan aquellos documentos cuyos conjuntos de palabras claves contengan las proporcionadas por el usuario a través de su consulta. Entonces, un SRI debe, a criterio de Hallo (2014), interpretar el contenido de los documentos de la colección y ordenarlos de acuerdo con el grado de relevancia para la consulta del usuario. La dificultad no está solo en conocer cómo extraer esta información sino también cómo utilizarla para decidir la relevancia de cada documento y devolver una respuesta.

Reconoce Hallo (2014) que los motores de búsqueda recolectan páginas de la web, las indexan, buscan en los índices las palabras claves que conforman la consulta, utilizan algoritmos de ranking para ordenar los resultados y muestran finalmente al usuario los documentos resultantes. En la recuperación de información tradicional una página web corresponde a un documento. La recuperación de información en la web considera como una colección de documentos la parte de la web que está públicamente indexada, excluyendo las páginas que no puedan ser indexadas por ser muy dinámicas o por ser privadas.

En la actualidad se encuentran en funcionamiento un gran número de estos motores entre los que podemos señalar Google, Bing, Yahoo y la Plataforma de Contenidos Unificados para Búsqueda Avanzada. Esta última funciona sobre el dominio (.cu) de la web cubana.

1.1.3 Evaluación de la recuperación de información

La evaluación de un SRI no es una tarea sencilla, y es un criterio asumido por Tolosa y Bordignon (2008). Debido a que el conjunto de respuesta no es exacto se requiere ponderar cómo éste se ajusta a la consulta y ésta a la necesidad de información del usuario, lo cual resulta muy complicado.

Para autores como Vogel et al. (2005), un problema importante de la RI es el grado de relevancia de los documentos. Por lo tanto, un objetivo de la RI es recuperar todos los documentos que sean relevantes a una consulta del usuario y recuperar la mínima cantidad de documentos no relevantes. Muy ligado al concepto de relevancia está el de pertinencia; con frecuencia se entremezclan y confunden. En general, relevancia es la medida de cómo una pregunta se ajusta a un documento (relevancia objetiva) y pertinencia es la medida de cómo un documento se ajusta a una necesidad informativa (relevancia subjetiva). Varios autores resaltan la dificultad para determinar la relevancia o no de un documento respecto a una consulta (Méndez y Muñoz, 2004).

La bibliografía consultada proyecta que existen dos indicadores principales aceptados ampliamente por la comunidad de RI, según Tolosa y Bordignon (2008), para medir cuantitativamente y evaluar el proceso de recuperación de información. Estas medidas o criterios son la precisión y la exhaustividad. La exhaustividad se define como la proporción de los documentos relevantes que han sido recuperados y permite evaluar la habilidad del sistema para encontrar todos los documentos relevantes de la colección mientras que la precisión se define como la proporción de los documentos recuperados que son relevantes y permite evaluar la habilidad del sistema para rankear (acción de ordenar elementos) primero la mayoría de los documentos relevantes.

Estas dos medidas se encuentran altamente relacionadas. Empíricamente se ha comprobado que una alta exhaustividad se acompaña de una muy baja precisión y viceversa. Opinan Tolosa y Bordignon (2008) que existe un compromiso entre exhaustividad y precisión, es decir, al aumentar la exhaustividad recuperando mayor cantidad de documentos, veremos disminuir la precisión. Esto se explica en el hecho que la salida de un SRI es un conjunto aproximado (no exacto) y entre ésta se encontrarán documentos no relevantes. Por el contrario, si recuperamos unos pocos documentos y todos son relevantes se tendrá una precisión máxima, pero seguramente se están perdiendo documentos útiles por no ser recuperados, lo cual es una desventaja.

El sistema ideal es aquel que siempre recupera todos los documentos relevantes y solo esos, situación en la que actualmente se trabaja puesto que aún no existe. En el trabajo con el objetivo de acercarse a esta meta se hace muy importante el uso de perfiles de búsquedas de usuarios que pueden ser utilizados en la recuperación de información para hacer el ranking de los documentos que estén fuertemente relacionados con la necesidad de información del usuario.

1.2 Procesamiento de información

A juicio de Dulzaides y Molina (2004), la concepción de una sociedad basada en la información y el conocimiento impone una dinámica constante sobre la práctica y teoría, tanto de la información como de la comunicación. El autor señala además que los nuevos avances han causado una conmoción en los métodos de procesamiento y recuperación de la información, ocupando un lugar especial, el desarrollo de las nuevas tecnologías de la información y la comunicación, que se han abierto, aún más, para la difusión del conocimiento.

Para Giraldo Huertas (2006), el procesamiento de la información tiene como finalidad generar datos agrupados y ordenados que faciliten al investigador el análisis de la información según los objetivos, hipótesis y preguntas de la investigación planteadas.

1.2.1 Minería de uso

La minería de uso de la web (del inglés, Web Usage Mining) trata de extraer patrones de uso de la web por parte de los usuarios. Para ello se utilizan los archivos (log) de los servidores web de forma que aplicando minería de textos sobre ellos se pueda extraer información útil. Este tipo de minería tiene como objetivos principales: identificar patrones generales de uso de un sitio web de manera que se pueda reestructurar para que sea más fácil de utilizar y mejore el acceso por parte de los usuarios, y obtener perfiles de los distintos tipos de usuarios a través de su comportamiento y navegación, para poder atender de forma más personalizada la búsqueda (Lorenzo y Trincado, 2015).

En minería de uso web se pueden convertir las sesiones de navegación en transacciones y estudiar qué páginas relacionan entre sí los usuarios (Lazcorreta, 2018). También Jácome Paneluisa y Meneses Becerra (2017) consideran que permite descubrir lo que los usuarios buscan en Internet.

La minería de uso web extrae la información opcional que se exhibe en los registros y se obtiene de las asociaciones de los clientes con la web (Kaur et al., 2017). Los sistemas de minería están conectados en la pantalla de información en los registros del servidor web, registros de programas, invitaciones, perfiles de clientes, marcadores, clics del mouse, etc. Esta información se ensambla regularmente, por lo tanto, se accede al registro web a través del servidor web (Ladekar et al., 2014). Predominantemente existen cuatro tipos de fuentes de información presentes en el que se registra la información de uso en varios niveles que son:

1. Colección de nivel de cliente: en este nivel, la información se ensambla mediante métodos para scripts (programa usualmente simple, que por lo regular se almacena en un archivo de texto plano) de Java o applets (componente de una aplicación que se ejecuta en el contexto de otro programa) de Java. Esta información demuestra la conducta de un cliente solitario en un solo sitio. La acumulación de información del lado del cliente requiere del interés del mismo para potenciar los scripts de Java o los applets de Java. El lado positivo de la recopilación de información del lado del cliente es que puede detectar todas las instantáneas, incluida la contracción de la parte posterior o la recarga.
2. Colección de nivel del navegador: la segunda técnica para la acumulación de información es cambiando el programa. Demuestra la conducta de un solo cliente en numerosos lugares. Las capacidades de acumulación de información se actualizan ajustando el código fuente del programa existente (Kaur et al., 2017). Brindan una información mucho más adaptable a medida que consideran la conducta de un solo cliente en numerosos sitios (Neelima y Rodda, 2016).

3. Recopilación de nivel Proxy: los servidores proxy (servidor que hace de intermediario en las peticiones de recursos que realiza un cliente a otro servidor) son utilizados por una organización especialista en web para proporcionar acceso a la World Wide Web (red informática mundial, es un sistema de distribución de documentos de hipertexto o hipermedia interconectados y accesibles vía Internet) a los clientes. Este servidor almacena la conducta de varios clientes en varios sitios. Estas capacidades de servidor, como el servidor de la tienda, pueden realizar visitas reservadas en línea. Al analizar el patrón de uso del invitado, la minería de uso web mejora la naturaleza de la administración de empresas basada en web y personaliza la web o, luego, actualiza la ejecución de la estructura web y el servidor web (Anitha y Isakki, 2016; Pratap Singh y Jain, 2014).
4. Información del servidor que se recopila de los servidores web: Incorpora documentos de registro, cookies (pequeña información enviada por un sitio web y almacenada en el navegador del usuario, de manera que el sitio web puede consultar la actividad previa del navegador) e información inequívoca del cliente. Los servidores contienen tipos de registros distintivos, que se consideran el activo de fecha principal para la extracción de utilización web. Los registros más conocidos son: formato de registro normal, dirección IP, nombre de registro remoto, nombre de cliente confirmado, marca de tiempo, solicitud de acceso, URL de referencia, agente de usuario y protocolo.

1.2.2 Minería de contenido

Para autores como Kaur et al. (2017), la minería de contenido es un componente de la minería de datos. Los usos principales de la minería de contenido web son reunir, categorizar, organizar y proporcionar la mejor información posible disponible en la web al usuario que solicita la información.

La minería de contenido web, según Curiel Lorenzo y Pantoja Trincado (2015), trata de extraer información relevante sobre el contenido de la web de manera que pueda ayudar a clasificarlo, aumentando la organización de ese contenido, para posteriormente mejorar el acceso y la recuperación de la información en él contenida (Barriga Mariño, 2017).

Dos enfoques utilizados en la minería de contenido web son el enfoque basado en agentes y el enfoque de bases de datos. Los tres tipos de agentes son agentes de búsqueda inteligente, filtro de información / agente de clasificación, agentes web personalizados. Los agentes de búsqueda inteligente buscan automáticamente información de acuerdo con una consulta particular utilizando las características del dominio y los perfiles de usuario. Los agentes de información utilizan varias técnicas para filtrar los datos de acuerdo con las instrucciones predefinidas. Los agentes web personalizados aprenden las preferencias del usuario y descubren documentos relacionados con esos perfiles de usuario (Leela Mary y Silambarasan, 2017).

1.2.3 Minería de texto

La minería de datos (datamining) y la minería de texto o minería textual (textmining) surgen como tecnologías emergentes que sirven de soporte para el descubrimiento de conocimiento que poseen los datos almacenados y son consideradas técnicas de análisis de información. La minería textual se orienta a la extracción de conocimiento a partir de datos no-estructurados en lenguaje natural almacenados en las bases de datos textuales, se identifica con el descubrimiento de conocimiento en los textos y se le denomina comúnmente Knowledge-Discovery in Text (KDT por sus siglas en inglés) (Gálvez, 2008).

Las aplicaciones prácticas de la minería de textos son muy diversas y han sido abordadas por autores como (Figuerola et al., 2004) donde expone las áreas de aplicación.

Es reconocido por Justicia de la Torre (2017) que una forma particular de Text Mining o Minería de Textos es la Extracción de Metadatos. Los metadatos son datos sobre datos, se refieren generalmente a aspectos, como autor, título, esquema de clasificación, descriptores, etc. Los metadatos pueden considerarse como un registro bibliográfico enriquecido y actualmente estructurado del documento, tal que su objetivo es expandir y acompañar con frases al objeto tomando la noción de extracción de entidades. Pueden incluirse una gran variedad de atributos de estos recursos de información, como e-mail (servicio de correo electrónico), direcciones, resúmenes, tablas de contenido, URL, ISBN, estructuras químicas, ecuaciones matemáticas, etc.

El procesamiento documental se puede acelerar y mejorar significativamente con el uso adecuado de los metadatos. Esto, según De la Puente (2010), permite realizar búsquedas, enlaces y referencias cruzadas entre los documentos representados por los metadatos. Para identificar este tipo de objetos digitales se usan atributos que se completan con piezas individuales de información estructuradas, denominadas información de extracción.

En conclusión, en la Minería de Textos se darían en general, las siguientes etapas (De la Puente 2010):

1. Recuperación de información, es decir, seleccionar los textos pertinentes.
2. Extracción de la información incluida en esos textos: hechos, acontecimientos, datos clave, relaciones entre ellos, etc.
3. Por último se realizaría lo que antes definíamos como minería de datos para encontrar asociaciones entre esos datos claves previamente extraídos de entre los textos.

La categorización de documentos de texto es una aplicación de la minería de texto que asigna a los documentos una o más categorías, etiquetas o clases, basadas en el contenido. Es un componente importante de muchas tareas de organización y gestión de la información (Pérez Abelleira y Cardoso, 2010). La presente investigación abordará cuestiones como la categorización de documentos debido a que este es uno de los procesos que realiza el módulo a desarrollar.

1.2.4 Minería web

La minería web (del inglés Web Mining) se encarga de aplicar las técnicas de minería de datos a documentos y servicios de la web. Consiste en extraer la información, imágenes, textos, audio, video, documentos y multimedia de un sitio web (Orellana Cordero., 2016). La variedad de formatos hace que las técnicas utilizadas en minería web cambien según la tarea particular que haya que llevar a cabo. Por lo tanto, Neptalí Chávez Quispe (2014) asume que la minería web trata básicamente con información de gran tamaño, con hiperenlaces y con las características antes mencionadas.

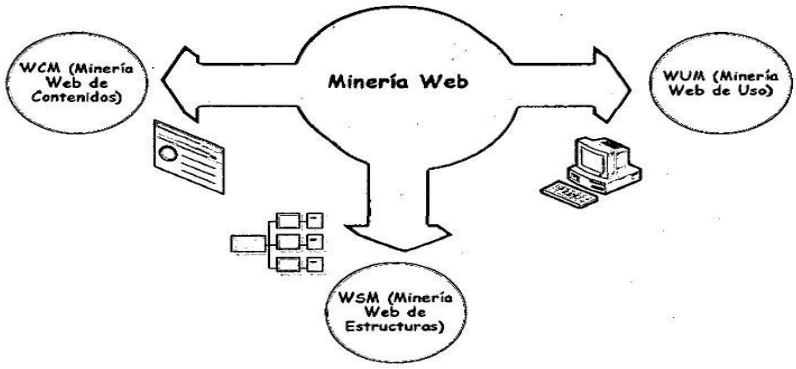

Figura 1. Categorías de la minería web. Fuente: (Ropero Rodríguez, 2010)

Algunos autores la consideran como la exploración de información en la web (Salazar, 2015; Shoaib y Maurya, 2014). Según la bibliografía consultada se conocen tres tipos de Minería Web:

Minería web de uso: es utilizada para monitorear y analizar la actividad de los aprendices, de tal forma que se pueda conocer su patrón de comportamiento (Shoaib y Maurya, 2014; Pandya, 2015; Suthar y Oza, 2015). Además, Aguilar y Mosquera (2015) afirman que el resultado de este análisis permitirá caracterizar (tipificar) al individuo conforme a ese patrón de aprendizaje. Para ello se utilizan los registros de los servidores web de forma que aplicando minería de textos sobre ellos se pueda extraer información útil (Balasubramanian y Uthangarai, 2015).

Minería web de contenido: extrae información relevante sobre el contenido de la web de manera que pueda ayudar a clasificarlo, aumentando la organización del contenido, para posteriormente mejorar el acceso y la recuperación de la información en él contenida (Mele, 2013; Curiel Lorenzo y Pantoja Trincado, 2015; Gök et al., 2015).

Minería web de estructura: intenta descubrir el modelo subyacente de las estructuras de los enlaces de la Web, el cual se basa en la topología de los hiperenlaces. Este modelo puede ser usado para categorizar las páginas web y es útil para generar información, como la calidad de una página web o la relación entre diferentes páginas web (Gupta, 2014; Krishna Murthy, 2015; De Gyves Camacho, 2009).

La naturaleza de esta investigación justifica el uso de técnicas de minería web de uso y contenido para el procesamiento de datos y generación de conocimientos en el módulo, ya que el procesamiento estará enfocado en los registros de los usuarios y el contenido de los documentos web almacenados.

1.2.5 Herramientas para la minería de datos

Para el desarrollo de la funcionalidad de categorización se realizó un estudio sobre tres herramientas para el trabajo enfocado en la Minería de Datos (Knime, RapidMiner, Weka), basándose en sus características, funcionamientos y posibilidades. Son herramientas de software de código abierto para minería de datos que pueden ser obtenidas de forma gratuita.

- **Knime**: Está desarrollado sobre la plataforma Eclipse y programado en Java, su uso se basa en el diseño de un flujo de ejecución que plasme las distintas etapas de un proyecto de minería de datos y predecir posibles resultados. Es una plataforma de código abierto de fácil uso y comprensible para integración de datos, procesamiento, análisis, y exploración. Ofrece a los usuarios la capacidad de crear de forma visual flujos de datos, ejecutar selectivamente algunos o todos los pasos de análisis, y luego estudiar los resultados, modelos y vistas interactivas (Knime, 2017).
- **RapidMiner**: Es un ambiente de experimentos en aprendizaje automático y minería de datos que se utiliza para tareas de minería de datos tanto en investigación como en el mundo real. Permite a los experimentos componerse de un gran número de operadores anidables arbitrariamente, que se detallan en archivos XML y se hacen con la interfaz gráfica de usuario de RapidMiner. Ofrece más de 500 operadores para todos los principales procedimientos de máquina de aprendizaje, y también combina esquemas de aprendizaje y evaluadores de atributos del entorno de aprendizaje Weka. Está disponible como una herramienta autónoma para el análisis de datos y como motor para minería de datos que puede integrarse en sus propios productos (Rapidminer, 2016).
- **Weka:** Entorno Waikato para el Análisis del Conocimiento (Weka) es una conocida suite de software para máquinas de aprendizaje que soporta varias tareas típicas de minería de datos, especialmente pre procesamiento de datos, agrupamiento, clasificación, regresión, visualización y características de selección. Sus técnicas se basan en la hipótesis de que los datos están disponibles en un único archivo plano o relación, donde cada punto marcado es etiquetado por un número fijo de atributos. Weka proporciona acceso a bases de datos SQL utilizando conectividad de bases de datos Java y puede procesar el resultado devuelto como una consulta de base de datos. Su interfaz de usuario principal es el explorer (componente del sistema Windows que presenta la interfaz en el monitor), pero la misma funcionalidad puede ser accedida desde la línea de comandos o a través de la interfaz de flujo de conocimientos basada en componentes (Weka, 2016).

Tabla 4. Comparativa de las herramientas para la minería de datos. Fuente: (Jaramillo Valbuena y Londoño, 2014).

Características	RapidMiner	Weka	Knime
Licencia Libre	Si	Si	Si
Multiplataforma	Si	Si	Si
Modificar modelos	Si	Si	No
Técnicas Descriptivas (agrupación)	Si	Si	Si
Técnicas Predictivas (clasificación)	Si	Si	No
Visualización de datos	Si	Si	Si
Filtros	Si	Si	No
Conversión de datos	Si	Si	No
Procesamiento de datos	Si	Si	Si

Dado que las herramientas seleccionadas en su mayoría comparten características similares se ha optado por realizar la comparativa solamente de los algoritmos de minería de datos que integran, dejando a un lado las opciones de pre-procesamiento, post-procesado y visualización en la siguiente tabla comparativa.

Tabla 5: Comparativa de los algoritmos en las herramientas para la minería de datos. Fuente: (Stratebi, 2010).

Datos	RapidMiner	Weka	Knime
Algoritmos implementados de forma nativa	34	168	9
Algoritmos exportados desde Weka	101	0	102
Total de algoritmos implementados	135	168	111
% de algoritmos nativos	25.2%	100%	8.1%

Con el estudio realizado sobre las herramientas para la minería de datos, se descarta el uso de la herramienta Knime por la ausencia de técnicas predictivas, lo que imposibilita la tarea de categorizar información. Las herramientas RapidMiner y Weka comparten características comunes y son software de código abierto. Se selecciona la herramienta Weka por su vasta colección de algoritmos nativos.

1.3 Procesamiento de documentos

1.3.1 Categorización de documentos

Para comprender las razones del surgimiento de la clasificación de documentos, Buenaño Valencia y Vaca Albán (2017) plantea que se debe conocer con anterioridad la diferencia entre la información estructurada y no estructurada. La información estructurada es aquella que su significado o contenido es exacto y se puede apreciar con claridad en la distribución y disposición de los datos que la componen. La información no estructurada es aquella que podemos encontrar en los

documentos generados en cualquier procesador de texto, correos electrónicos y hasta el texto que se encuentra en páginas web, los cuales carecen de una estructura específica.

Para los autores Buenaño Valencia y Vaca Albán (2017), la expresión "información no estructurada" se refiere típicamente a aquellos datos que no están organizados bajo el Modelo de Datos Relacional, algunos ejemplos comunes de información no estructurada son los archivos de texto, documentos (PDF, Word), imágenes, audio y video, entre otros. Por las características del sistema con el que se trabaja en esta investigación se profundizará en la clasificación de documentos que contienen información no estructurada.

Según Moore (2002), la clasificación de documentos de texto es una aplicación de la minería de textos que pretende extraer información de texto no estructurado disponible en la web. Su interés se justifica porque se estima que entre el 80% y el 90% de los datos de las organizaciones son no estructurados. Esta clasificación se define como la actividad de etiquetar textos en lenguaje natural con categorías temáticas de un conjunto predefinido (Sebastiani, 2005).

La categorización de documentos asigna a los mismos una o más categorías, etiquetas o clases, basadas en el contenido. Es un componente importante de muchas tareas de organización y gestión de la información según Curiel Lorenzo y Pantoja Trincado (2015). Estas etiquetas o clases pueden ser definidas a priori o pueden ser definidas de forma automática haciendo uso de clustering (agrupamiento que consiste en un procedimiento de agrupación de una serie de vectores de acuerdo con un criterio) de textos.

Actualmente son muy utilizados los corpus (conjunto amplio y estructurado de ejemplos reales de uso de la lengua. Estos ejemplos pueden ser textos o muestras orales) lingüísticos en el área de la clasificación. Un corpus lingüístico es definido por Buyse (2018) como un conjunto de textos informatizados producidos en situaciones reales, que se han seleccionado siguiendo una serie de criterios lingüísticos explícitos que garantizan que dicho *corpus* pueda ser usado como muestra representativa de la lengua.

El corpus Sogou es un archivo web chino que se utiliza como recurso de datos (Hu, et al., 2013). Este archivo contiene 135,4 millones de páginas web de 5,3 millones de sitios web chinos recopilados por Sogou.com desde junio de 2006 hasta enero de 2007. Sogou.com es uno de los motores de búsqueda comerciales más grandes en el entorno web chino.

El corpus propuesto por Sogou es utilizado en las investigaciones de varios autores como (Yao et al., 2012; Aliwy and Ameer, 2017). En este corpus, según Yaya y Xueyun (2012), se definen nueve categorías: finanzas, militar, educación, entretenimiento, tecnología, deportes, computación, social y medicina. Se asume en esta investigación las categorías mencionadas anteriormente porque este *corpus* es ampliamente utilizado en la actualidad por la comunidad de investigadores.

El proceso de categorización está compuesto de tres pasos fuertemente relacionados en los cuales se llevan a cabo importantes tareas que dependen entre sí. Estas tareas son el pre-procesamiento

de los datos o información, la construcción y/o entrenamiento del clasificador a utilizar y por último la categorización de los documentos que son sometidos al proceso.

Autores como Herrecalde et al. (2009) señalan que los enfoques para la construcción automática de clasificadores, pueden ser agrupados en dos grandes áreas: la categorización supervisada y la no supervisada. La categorización supervisada utiliza un proceso inductivo general basado en el conocimiento que se tiene de: a) las categorías y b) ejemplos de documentos categorizados por un experto. En la categorización no supervisada (agrupamiento) en cambio, no se conocen a priori las categorías ni asignaciones correctas de categorías (sólo se conoce alguna medida de similitud).

Para otros autores como Vergara (2014) a los enfoques vistos anteriormente se suma el aprendizaje semi-supervisado. En este caso la fase de creación del clasificador utiliza la colección de entrenamiento como base, pero se sigue refinando con documentos sin clasificar. Aquí el número de los documentos sin clasificar suele ser mucho mayor que el número de los documentos ya clasificados. Este tipo de aprendizaje puede ayudar en el caso de tener un número pequeño de documentos preclasificados, pero por lo general es más crítica la creación de un buen clasificador.

Por razones expuestas anteriormente la categorización supervisada es la aplicada a esta investigación al asumir las clases o categorías a utilizar que propone Sogou Corpus.

1.3.2 Algoritmos de categorización

Dentro del área de aprendizaje automático existen una gran variedad de algoritmos. Estos algoritmos se clasifican de acuerdo con su uso en diferentes métodos, ya sean estos supervisados, no supervisados o semi-supervisados, además del tipo de modelo de aprendizaje utilizado. Los modelos, pueden ser probabilísticos, de agrupación, de co-entrenamiento y basados en grafos, entre muchos otros que también han sido estudiados (Di Nunzio, 2009).

Diferentes tipos de algoritmos han sido utilizados para llevar a cabo la tarea de categorización automática de documentos. La función principal de éstos, es realizar el proceso inductivo necesario, basado en el conjunto de entrenamiento, para asignar automáticamente una categoría, de entre varias previamente definidas, a un documento. Entre estos algoritmos se encuentran: árboles de decisión, vecinos más cercanos, el método Naive Bayes y las máquinas de vectores de soporte, por mencionar algunos (Di Nunzio 2009).

1.3.2.1 Árbol de Decisión

El objetivo del árbol de decisión es generar un modelo que a partir de un conjunto de variables sea capaz de predecir un valor (clase) de salida (Carreño, 2017). Para Suca et al. (2016), el Árbol de Decisión C4.5 es una máquina de aprendizaje para predicciones con una variable dependiente que puede llegar al objetivo deseado, basándose en los atributos de los datos disponibles. Los nodos internos son los diferentes atributos, las ramas son los posibles valores y los nodos finales (hojas) son ya la clasificación. Este método al ser iterativo va colocando los posibles valores de las

características (información ganada). Cuando todas caigan en una clasificación y ya no exista ambigüedad entonces se asigna una raíz o un nodo. Además, el C4.5 es una extensión del árbol ya conocido ID3, es decir una mejora al no manejar datos numéricos o continuos, además el C4.5 selecciona un subconjunto en base a una ventana y ya no lo hace de forma aleatoria como lo hacía el ID3.

1.3.2.2 Algoritmo del vecino más próximo

El algoritmo *k*–Vecinos más cercanos (*k*-NN, por sus siglas en inglés) es uno de los métodos de aprendizaje basados en instancias más básicos, pero con resultados aceptables en tareas que involucran el análisis de texto. En resumen, este algoritmo no tiene una fase de entrenamiento fuera de línea, por lo tanto, el principal cálculo se da en línea cuando se localizan los vecinos más cercanos.

La idea en el algoritmo es almacenar el conjunto de entrenamiento, de modo tal que, para clasificar una nueva instancia, se busca en los ejemplos almacenados casos similares y se asigna la clase más probable en éstos (Téllez Valero, 2005). Este autor resume el algoritmo, aquí una manera común de encontrar los k ejemplos más cercanos a la instancia i_q es por medio de la distancia Euclidiana, donde la distancia entre las instancias i_j e i_q es definida por la siguiente ecuación:

$$d(i_j, i_q) = \sqrt{\Sigma k^{|A|} = 1\ (a_{kj} - a_{kq})^2} \quad (1)$$

Ecuación 1. Ecuación de distancia entre instancias. Fuente: (Téllez Valero, 2005)

Según Figuerola et al. (2004) el algoritmo del vecino más próximo es uno de los más sencillos de implementar. La idea básica es como sigue: si calculamos la similitud entre el documento a clasificar y cada uno de los documentos de entrenamiento, el más parecido nos estará indicando a qué clase o categoría debemos asignar el documento que deseamos clasificar. Una vez localizado el documento de entrenamiento más similar, dado que éstos han sido previamente categorizados manualmente, sabemos a qué categoría pertenece y, por ende, a qué categoría debemos asignar el documento que estamos clasificando.

De acuerdo con Figuerola et al. (2004), una de las variantes más conocidas de este algoritmo es la del *k*-nearest neighbour o KNN que consiste en tomar los k documentos más parecidos, en lugar de sólo el primero. KNN parece especialmente eficaz cuando el número de categorías posibles es alto, y cuando los documentos son heterogéneos y difusos. Esto no lo hace idóneo para ser aplicado en esta investigación puesto que las categorías o clases definidas a priori son una cantidad finita y relativamente pequeña.

1.3.2.3 Redes Neuronales

El concepto de Red Neuronal Artificial (RNA) está inspirado en las Redes Neuronales Biológicas (Neyra, 2016). Una Red Neuronal Biológica es un dispositivo no lineal altamente paralelo,

caracterizado por su robustez y su tolerancia a fallos. Sus principales características son las siguientes: aprendizaje mediante adaptación de sus pesos sinápticos a los cambios en el entorno, manejo de imprecisión, ruido e información probabilística y generalización a partir de ejemplos (Paz Arias y Jiménez Ochoa, 2016).

Según Blanco et al. (2015), una red neuronal artificial es un conjunto de modelos matemáticos-computacionales reales e ideales de una red neuronal y se emplea en estadística psicológica e inteligencia artificial. Las RNA no son más que un modelo artificial y simplificado del cerebro humano, es un nuevo sistema para el tratamiento de la información cuya unidad básica de procesamiento está inspirada en la célula fundamental del sistema nervioso humano: la neurona.

Este tipo de red neuronal consiste en un conjunto de elementos computacionales simples unidos por arcos dirigidos. Cada arco tiene asociado un peso numérico *wij* donde (*i*) representa la entrada y (*j*) la salida, este peso indica la significación de la información que llega por este arco (Blanco et al., 2015).

Este algoritmo tiene como desventaja que no existe un algoritmo de entrenamiento óptimo que garantice la convergencia de la red en el mínimo global en un tiempo aceptable. Debido a la importancia del entrenamiento de la colección para este tipo de algoritmo se descarta el uso del mismo en la solución que se propone.

1.3.2.4 Algoritmo Naive Bayes

La clasificación supervisada es una parte del aprendizaje automático con gran número de aplicaciones en distintos campos. Generalmente, en la clasificación supervisada se asume la existencia de dos tipos de variables: las variables predictoras, X = (X_1,..., Xn), y la variable clase, C. Mediante los clasificadores supervisados se trata de aprender las relaciones entre las variables predictoras y la clase, de forma que se pueda asignar un valor de C a un nuevo caso, x = (x1,..., xn), en el que el valor de la clase es desconocido. Uno de los paradigmas ampliamente usados para clasificación supervisada son las redes Bayesianas (Rodrigo et al., 2006).

Existen métodos bayesianos que se encuentran entre los más eficientes y algunos permiten interpretar el funcionamiento de otros métodos en términos probabilísticos, incluso cuando no son aplicables, proporcionan un estándar de toma de decisión óptima, frente al que comparar otros métodos. Naive Bayes (NB) es el modelo de red bayesiana orientada a clasificación más simple.

En el artículo “Análisis de estrategias de clasificación multiclase en Microarrays” se presentan cuatro clasificadores pertenecientes a varias familias entre los que se incluye NB, el cual, según Rish (2001), es un clasificador que emplea la regla de Bayes para determinar la probabilidad de pertenecer a cada clase para una determinada muestra. Este clasificador asume que los atributos son condicionalmente independientes entre sí dada la clase. La palabra naive (ingenuo en inglés) se emplea porque el algoritmo utiliza técnicas bayesianas, pero no tiene en cuenta las dependencias entre variables predictoras, que realmente puedan existir (Vega Vilca y Torres Núñez, 2015).

A consideración de Fernández (2004) y de acuerdo con Hernández et al. (2004), entre las características que poseen las redes bayesianas y que pueden ser consideradas ventajas se puede destacar que permiten aprender sobre relaciones de dependencia y causalidad, permiten combinar conocimiento con datos y pueden manejar bases de datos incompletas. Las ventajas de NB incluyen: menor costo computacional, fácil de entender al ser un clasificador basado en probabilidades, y requiere un solo paso de procesamiento si los datos son discretos (Adnan y Husain, 2012).

A pesar de su diseño 'ingenuo' y su aparente sencillez, los clasificadores NB son eficientes y robustos ante el ruido y los atributos irrelevantes (Rish, 2001). Para Fernández (2004), de manera gradual los investigadores de la comunidad de aprendizaje automático se han dado cuenta de su potencialidad y robustez en problemas de clasificación supervisada. A pesar de la suposición poco realista realizada en el NB, este algoritmo se considera un estándar y sus resultados son competitivos con la mayoría de los clasificadores.

Tabla 6. Comparación de algoritmos de clasificación. Fuente: (Leyva et al., 2018)

Parámetros	**Knn**	**Naive Bayes**	**Redes Neuronales**	**Árboles de clasificación**
Determinista/No determinista	No determinista	Determinista	Determinista	No determinista
Efectividad en colecciones	Pequeñas	Muy grandes	Efectiva en pequeñas o grandes	Grandes
Velocidad	Lento para grandes colecciones	Más veloz que Knn	No existe un algoritmo de entrenamiento óptimo	Veloz
Conjunto de datos	No trabaja con datos difusos	Trabaja con datos difusos	Trabaja con datos difusos	Trabaja con datos difusos
Exactitud	Alta	Mejora la exactitud con un mayor número de datos	No se puede asegurar su exactitud	Alta

En esta investigación se utiliza el algoritmo Naive Bayes porque presenta un costo computacional bajo, siendo esta una característica importante en la RI para poder entregar al usuario los resultados de la búsqueda en el menor tiempo posible. Además, su efectividad aumenta de forma proporcional al volumen de datos de la colección, lo cual es muy importante en SRI porque sistemáticamente los datos se multiplican. De igual forma se comporta la exactitud de la respuesta del algoritmo la cual mejora con un mayor número de datos.

1.4 Categorización de consultas

Según Yu y Xie (2017), la categorización de consultas implica clasificar las consultas web en categorías de destino predefinidas. Se utiliza para mejorar la relevancia de las búsquedas y la publicidad en línea. Las consultas web son generalmente cortas, con una longitud de palabra promedio pequeña, y esto las hace ambiguas. Además, pueden estar en constante cambio y pueden seguir cambiando en función de los eventos actuales.

La categorización de consultas es una técnica que en la actualidad ha cobrado gran importancia por el hecho de que puede ser aplicada en SRI para de forma conjunta con otras técnicas lograr mejorar el proceso de respuesta al usuario. Alcanzar entender la intención de lo que un usuario plantea en una consulta y devolver la información más ajustada a su necesidad es uno de los principales problemas que en la actualidad enfrenta la RI.

Autores como Vogel et al. (2005) mencionan la dependencia entre la eficiencia y eficacia de los motores de búsqueda y su correspondiente habilidad para capturar el significado de una consulta más probablemente deseada por el usuario. En este sentido Valencia (2007) expresa que la clasificación permite dada una consulta asignarle una categoría que permita devolver el mejor resultado a una consulta en un momento dado. Sin embargo, la categorización de consultas web supone un reto desde diferentes puntos de vista.

La longitud de las consultas web junto con el constante cambio en la distribución y el vocabulario de las consultas dificultan su clasificación mediante técnicas convencionales de clasificación de textos, demostrando los retos a los que se enfrenta la categorización de consultas (Vogel et al., 2005).

La categorización de consultas será una técnica a utilizar en la solución que se propone para lograr obtener la categoría de cada una de las consultas de usuarios y documentos almacenados.

1.5 Similitud entre consultas

La similitud de las consultas web juega un papel importante en la captura de preguntas frecuentes, los temas más populares del motor de búsqueda o la expansión automática de consultas. La medición precisa de la similitud entre consultas es crucial (Meng et al., 2013).

El modelo de espacio vectorial emplea el peso de los términos para cada documento. Además, refleja la relevancia de los términos del documento de cara a su representatividad en la colección. Esta idea puede ser aplicada al cálculo de similitud entre consultas, cambiando en su concepción un documento por la consulta del usuario y empleando el peso de los términos para cada consulta. Para realizar el cálculo del coeficiente de similitud entre consultas pueden utilizarse algunas funciones como coseno del ángulo entre los dos vectores, coeficiente de Dice y coeficiente de Jaccard.

El cálculo del coeficiente de similitud de Jaccard, al igual que el de Dice, resultan deudores del coeficiente de similitud del Coseno. Su aplicación, centrada en usos estadísticos, también se aplica a la recuperación de información y mide la similitud entre conjuntos. Se puede definir como el tamaño de la intersección (numerador), dividido por el tamaño de la unión de la muestra (Blázquez, 2013; Torres y Arco, 2016).

El cálculo del coeficiente de Dice, es una adaptación del cálculo del coeficiente del Coseno. La diferencia en la formulación estriba en que la cardinalidad del numerador es 2 veces la información compartida y el denominador, la suma de los pesos al cuadrado del documento y su consulta (Blázquez, 2013; Vargas, 2016; Torres y Arco, 2016).

Según Oliva Arenas (2017), una métrica muy utilizada en el procesamiento del lenguaje es la similitud coseno, que calcula el ángulo entre dos vectores. Comúnmente se utiliza la similitud del coseno cuando el espacio es positivo, donde el resultado de la métrica se limita al intervalo [0,1]. La similitud del coseno está dada por la expresión:

$$Sim_{cos}(x, y) = \frac{x \cdot y}{\|x\|\|y\|} \quad (2)$$

Ecuación 2. Similitud del coseno. Fuente: (Oliva Arenas, 2017)

Donde: $\|x\|$ y $\|y\|$ son vectores bajo la norma euclidiana x $(x_1, x_2, \dots, x_i)$, y $(y_1, y_2, \dots, y_i)$ definido como $\sqrt{x\ x_1^2, x_2^2, \dots, x_i^2}$, $\sqrt{y\ y_1^2, y_2^2, \dots, y_i^2}$

Las palabras que aparecen en una frase se proyectan en un espacio. Una frase puede ser representada mediante la aplicación de una combinación lineal. La función de similitud entre dos frases es entonces el coseno de similitud entre sus correspondientes vectores. Tomando esta idea se representan la consulta del usuario expresada en frases como un vector y se calcula su similitud.

El cálculo de la similitud entre consultas aplicado a la solución que se propone, permitirá agilizar la búsqueda del conjunto de respuesta que se le brindará al usuario al tener en cuenta experiencias de búsquedas anteriores. También contribuirá a resolver el problema que para los nuevos usuarios representa el hecho de contar con perfiles relativamente pobres, porque a partir de sus consultas podrán obtenerse respuestas similares a las ya emitidas a otro usuario que hubiese ejecutado la misma consulta.

1.6 Algoritmos de ranking

A menudo es difícil tomar la decisión binaria sobre si un documento es relevante o no para una consulta determinada. En cambio, los documentos se ordenan de acuerdo a sus grados de relevancia a la consulta, generando un ranking de los documentos (Trotman, 2005). El ranking permite la clasificación de documentos ordenados en base a una consulta, ubicando a los documentos con mayor relevancia en los primeros lugares (Briceño Segovia, 2011).

En la investigación de Sharma y Sharma (2010) se realiza un estudio de varios algoritmos de ranking contemporáneos, exponiendo sus principales características y establece una comparación entre varios de ellos en cuanto a criterios como las técnicas que utilizan, las ventajas y las limitaciones de su utilización. En el caso de Muppidi y Koraganji (2016), realiza un estudio de varios algoritmos de este tipo en cuanto a los criterios de eficiencia y popularidad.

A continuación se presentan algunos de los algoritmos de ranking más utilizados a nivel mundial y se realiza un estudio de sus principales características.

Tabla 7. Varios algoritmos de *ranking*. Fuente: (Muppidi y Koraganji, 2016)

Algorithm Name	Author	Year
HITS(hyperlink-induced Topic search Algorithm)	Jon Kleinberg	1996
Page Ranking Algorithm	Larrypage and Sergey	1996
Improved HITS Algorithm	Longzhuang Li, Yi Shang, Wei Zhang	2002
Weighted Page Rank Algorithm	Wenpu Xing and Ali Ghorbani	2004
Distance Rank Algorithm	Ali Mohammad Zare Bidoki, Nasser Yazdan	2007
Relation Based Algorithm	F Lamberti	2009
New Page Ranking Algorithm	Hema Dubey, Prof. B. N. Roy	2010
Rank Based Finger Algorithm	J. Machaj, R. Piche and P. Brida	2011
Ontology Ranking Algorithm on Semantic Web	K. Sridevi and Dr. R. Umarani	2013

- Page Rank Algorithm: Page Rank es uno de los métodos de clasificación más importantes utilizados en los motores de búsqueda de hoy. No solo es Page Rank una forma sencilla, robusta y confiable de medir la importancia de las páginas web, sino que también es beneficioso desde el punto de vista informático con admiración hacia otros métodos de clasificación, ya que es independiente de consultas y de contenido independiente. De lo contrario, se puede calcular fuera de línea utilizando solo la estructura web y luego se usará más adelante, a medida que los usuarios envíen las consultas al motor de búsqueda, característicamente colectivas con otras clasificaciones que dependen de la consulta. Con el fin de medir la importancia relativa de las páginas web, sugieren Page Rank, un método para calcular una clasificación para cada página web basada en el gráfico de la web. Page Rank tiene aplicación en búsqueda, navegación y estimación de tráfico (Deshmukh y Barve, 2016).
- *HITS* Algorithm (algoritmo): Desarrollado por Jon Kleinberg, hizo uso de los enlaces de la red web para descubrir y clasificar páginas relevantes para la consulta dada por el usuario para recuperar datos (Tian et al., 2014). El algoritmo HITS es un algoritmo de estructura de enlaces. Este algoritmo clasificará las páginas al exceder como enlaces dentro y fuera de las páginas web. Según Garg y Jain (2015), en este algoritmo, una página web se denomina autoridad si la página web está señalada por muchos hipervínculos y una página web es nombrada como *HUB* si la página apunta a varios hipervínculos. Hubs (dispositivo que permite centralizar el cableado de una red de computadoras, para luego poder ampliarla) y autoridades tienen asignadas puntuaciones respectivas. Las puntuaciones se calculan de forma que se refuerzan mutuamente; una autoridad apuntada por varios centros altamente calificados debe ser una autoridad fuerte, mientras que un centro que apunta a varias autoridades altamente calificadas debería ser un centro popular. HITS es un algoritmo

puramente basado en enlaces. Se utiliza para clasificar las páginas que se recuperan de la web, en función de su contenido textual a una consulta determinada. Una vez que estas páginas han sido ensambladas, el algoritmo HITS ignora el contenido textual y se enfoca únicamente en la estructura de la web.

- Distance Algorithm: El objetivo principal de este algoritmo es encontrar la distancia entre las páginas y dar la clasificación basada en la distancia. Este algoritmo se basa en la distancia entre cualquier página. La clasificación se da sobre la base de la distancia logarítmica más corta entre dos páginas (Bidoki y Yazdani, 2008).
- Page ranking based on visit of link: Algoritmo en el que consideraron el comportamiento de navegación del usuario. Este concepto es muy útil para mostrar las páginas más valiosas en la parte superior de la lista de resultados en función del comportamiento de navegación del usuario, lo que reduce el espacio de búsqueda a gran escala; él propuso un algoritmo de Page Rank mejorado. En este algoritmo asignamos más valor de rango a los enlaces salientes que es más visitado por los usuarios. De esta manera, se calcula un valor de rango de página basado en las visitas de los enlaces entrantes (Garg y Jain, 2015; Kumar et al., 2011).
- Apache Lucene™ es una biblioteca de motores de búsqueda de texto de alto rendimiento y con todas las características escritas completamente en Java. Es una tecnología adecuada para casi cualquier aplicación que requiera búsqueda de texto completo, especialmente multiplataforma. La puntuación de Lucene utiliza una combinación del Modelo de Espacio Vectorial (VSM) de recuperación de información y el Modelo Booleano (BM) para determinar qué tan relevante es un documento dado para la consulta de un usuario. Los documentos "aprobados" por BM son calificados por VSM. En VSM, los documentos y las consultas se representan como vectores ponderados en un espacio multidimensional, donde cada término de índice distinto es una dimensión y los pesos son valores de Tf-idf. Se tiene entonces, que para el término t y el documento (o consulta) x, Tf (t, x) varía con el número de ocurrencias del término t en x (cuando uno aumenta también lo hace el otro) e idf (t) varía de manera similar con el inverso de la cantidad de documentos de índice que contienen el término t. La puntuación VSM del documento d para la consulta q es la similitud coseno de los vectores de consulta ponderados V (q) y V (d), donde V (q) × V (d) es el producto escalar de los vectores ponderados, y | V (q) | y | V (d) | son sus normas euclidianas (Apache Lucene Core, 2018).

El módulo que se propone desarrollar se integrará a la Plataforma de Contenidos Unificados para Búsqueda Avanzada. Este SRI cuenta con un componente de indexación desarrollado sobre la tecnología Apache Solr, el cual es un motor de búsqueda de código abierto basado en la biblioteca Java del proyecto Lucene. Al ser necesaria la integración entre el SRI y el módulo se decide asimilar las tecnologías y herramientas del motor de búsqueda. Se utilizará entonces como algoritmo de ranking el que proporciona Lucene.

1.7 Algoritmos de *re-ranking*

En un esfuerzo más por acercar los resultados de búsqueda de un SRI a los intereses del usuario que ejecuta la consulta surgieron los algoritmos de re-ranking. Estos algoritmos, una vez obtenido el ranking de los documentos que conforman el conjunto de respuesta, vuelven a clasificar el resultado obtenido identificando el valor de interés del usuario en los enlaces recuperados. Dicho valor de interés es definido por varios autores en base a numerosos criterios generalmente diferentes.

En el presente epígrafe se realiza un estudio de varios algoritmos utilizados en el proceso de re-ranking de resultados. En la siguiente tabla se encuentran algunos de estos algoritmos.

Tabla 8. Varios algoritmos de re-*ranking*. Fuente: elaboración propia.

Algoritmo propuesto	Año
Wang, M., Li, Q., Lin, Y., & Zhou, B.	2017
Makvana, K., Shah, P., & Shah, P.	2014
Fouad, K. M., Khalifa, A. R., Nagdy, N. M., & Harb, H. M.	2012
Mittal, N., Nayak, R., Govil, M. C., & Jain, K. C.	2010

El algoritmo propuesto por Makvana et al. (2014) propone un enfoque novedoso que personaliza el resultado de búsqueda web mediante la reformulación de consultas y el perfil del usuario. En primer lugar, se propone un marco que identifica el término de búsqueda relevante para el usuario particular del historial de búsqueda anterior analizando el archivo de registro web mantenido en el servidor. Estos términos se anexan a la consulta ambigua del usuario. En segundo lugar, el enfoque propuesto procede al resultado de búsqueda del usuario y vuelve a clasificar el resultado obtenido identificando el valor de interés del usuario en los enlaces recuperados. El nuevo enfoque propuesto también identifica el interés del usuario en los enlaces recuperados combinando el valor de interés del usuario generado a partir de VSM (Vector Space Model, en español Modelo de Espacio Vectorial) y el rango real de ese enlace. En tercer lugar, el marco también sugiere algunas palabras clave que ayudan a incorporar el interés actual del usuario. Este algoritmo no es apto para aplicar en esta investigación porque las expresiones matemáticas que propone contienen elementos de dudosa procedencia que no son explicados ni abordados en el estudio que lo presenta.

En el caso del algoritmo propuesto por Wang et al. (2017) se propone un método personalizado para fusionar los resultados de un metabuscador de acuerdo con una variedad de factores, incluida la distribución del interés del usuario, el número total de motores de búsqueda de componentes explotados, el número de resultados que devuelve cada motor, la posición de clasificación del documento en cada motor de búsqueda y la cantidad de motores de búsqueda de componentes que devolvieron el documento. Este algoritmo está diseñado para metabuscadores, que aún con un funcionamiento que persigue el mismo objetivo que un motor de búsqueda, no se comportan de la

misma forma ni poseen los mismos componentes. Esto descarta la posibilidad de su uso en esta investigación.

El estudio del algoritmo propuesto por Fouad et al. (2012) arrojó que en el mismo se utiliza el modelo semántico IR, orientado a la explotación de las ontologías de dominios y WordNet para soportar capacidades semánticas de IR en documentos web, haciendo hincapié en el uso de ontologías en la perspectiva semántica. El sistema; llamado SPIRS, utiliza web semántica y agente para admitir consultas más expresivas y se proponen resultados más precisos. Por las características del dominio sobre el que rastrea la plataforma C.U.B.A. (dominio.cu), donde prácticamente todo el contenido está indexado en idioma español, no es necesario utilizar la base de datos WordNet. Además, el uso de ontologías no es objetivo de la presente investigación y por tanto se descarta su utilización en el proceso de re-ranking que se propone.

Otros autores proponen un enfoque híbrido de recuperación de información web personalizada que utiliza ontología para recuperar el perfil de usuario del contexto del usuario que se actualiza temporalmente de acuerdo con el comportamiento de navegación de los usuarios y el filtrado colaborativo para considerar la recomendación de usuarios similares. También se utiliza la expansión de consultas para enriquecer las mismas (Mittal et al., 2010).

La expansión de consultas, la creación y uso de ontologías unido al filtrado colaborativo son procesos complejos que demoran la respuesta que se ofrece el usuario, por tanto, no se tendrán en cuenta en esta investigación.

Consideraciones finales del Capítulo 1

En el presente capítulo se ha realizado un estudio de algunos elementos teóricos que sustentan la investigación, luego del cual se obtienen las siguientes consideraciones finales:

1. El marco teórico antes expuesto, ha contribuido a un mayor entendimiento del problema planteado.
2. A partir del análisis realizado ha quedado claro la necesidad de contribuir a mejorar la eficacia de las respuestas del SRI Plataforma de Contenidos Unificados para Búsqueda Avanzada.
3. El uso de tecnologías y herramientas como la minería web, el uso de algoritmos de categorización y de perfiles de búsqueda de usuarios son tendencias actuales y necesarias que deben utilizarse en la solución a desarrollar.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

Introducción

En el presente capítulo se describe la solución que se propone para lograr que la Plataforma de Contenidos Unificados para Búsqueda Avanzada mejore los resultados que brinda a los usuarios. El objetivo del capítulo es plantear los componentes que conforman la solución propuesta y explicar su funcionamiento y relación. Se presentan además los requisitos funcionales del módulo, las tecnologías y herramientas que se utilizan y se expone la arquitectura del módulo que se comunica con el SRI.

2.1 Descripción de la solución

La propuesta de solución radica en la integración al SRI Plataforma de Contenidos Unificados para Búsqueda Avanzada del módulo de procesamiento de contenidos. Este módulo está conformado por los elementos:

- Categorización de consultas: Se encarga de categorizar todas las consultas realizadas por los usuarios y que se encuentran almacenadas en la base de datos.
- Categorización de documentos: Se encarga de categorizar todos los documentos indexados por el sistema de recuperación de información.
- Creación y actualización del perfil de usuario: Se encarga de crear el perfil de los nuevos usuarios o de actualizar el perfil de los usuarios ya registrados cuando realizan una consulta.
- Cálculo de similitud entre consultas: Se encarga de obtener el valor correspondiente a la similitud entre cada una de las consultas almacenadas en la base de datos.
- Cálculo de similitud entre documentos: Se encarga de obtener el valor correspondiente a la similitud entre cada uno de los documentos indexados por el SRI.
- Proceso de re-ranking: Recibe dos conjuntos de documentos que se ajustan a la consulta realizada por el usuario y determina en qué orden deben ser mostrados los resultados.

La categorización de consultas, de documentos, el cálculo de la similitud entre consultas y documentos, y el proceso de re-ranking son utilizados para lograr un mejor aprovechamiento de la información y de los contenidos almacenados en el sistema de recuperación de información. Todo esto con el objetivo de acercar los resultados de la búsqueda del usuario a sus necesidades de información.

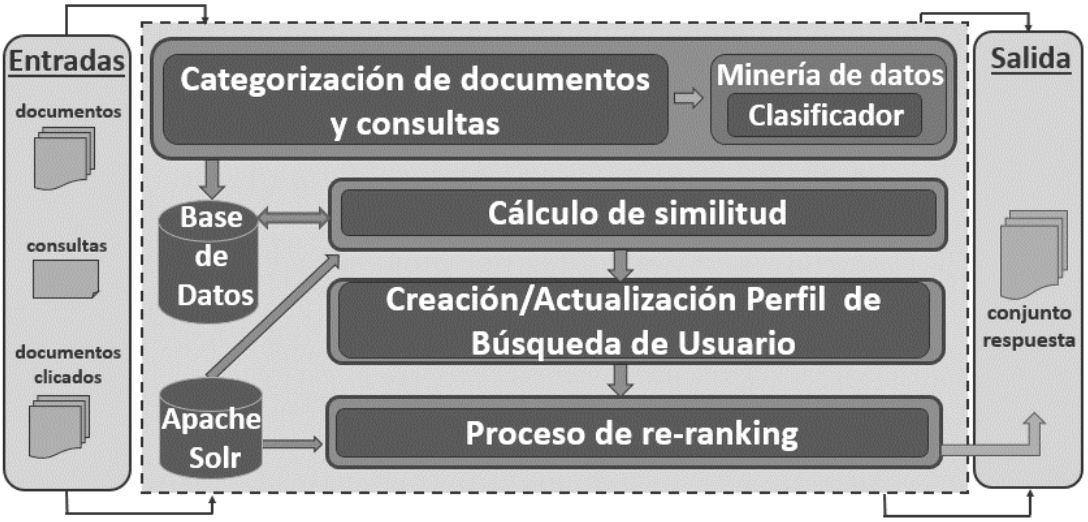

Figura 2. Descripción de la propuesta de solución. Fuente: elaboración propia.

La figura 2 muestra la descripción de la propuesta de solución, consistente en el módulo que debe ser integrado a la Plataforma de Contenidos Unificados para Búsqueda Avanzada.

El componente de rastreo se encargará de visitar las direcciones (url) y enviar al componente de indexación toda la información encontrada. El componente de indexación almacenará los documentos y a cada uno se le aplicará el proceso de categorización. Este proceso asegura que todos los documentos tengan definida una categoría según su contenido. Siempre que un documento es indexado se calcula la similitud entre este y los documentos ya almacenados, y se incorporan los valores obtenidos a una matriz de similitud. Se cuenta con una interfaz de usuario que permite realizar las consultas al motor de búsqueda. Las consultas realizadas se categorizan y almacenan en la base de datos junto al identificador y la categoría de los documentos que del conjunto de respuesta proporcionado por el SRI haya clicado el usuario. Cada vez que se ejecuta una nueva consulta se calcula la similitud entre esta y las almacenadas, registrándose los valores en una matriz de similitud para consultas.

De forma automática y permanente, el SRI se encarga de calcular la similitud entre cada una de las consultas almacenadas y los documentos indexados, registrando los valores de similitud que son consultados durante el proceso de conformación del conjunto de respuesta que se ofrece al usuario.

Si el usuario que trabaja con el sistema de recuperación de información no se encuentra registrado y, por tanto, no cuenta con un perfil, se procede a su creación. En caso de ser un usuario ya registrado se actualiza la información del perfil. Utilizando los documentos clicados y categorizados, se calcula el porciento de representatividad de cada categoría y se obtiene el perfil de búsqueda de usuario.

Siempre que el usuario introduzca un criterio de búsqueda se obtendrá de la matriz de similitud entre consultas aquella que posea el mayor valor de similitud. Al conocer la consulta más similar a la ejecutada se procede a obtener de los documentos indexados por el SRI aquellos que poseen un valor de similitud mayor a (0,9) con respecto a la consulta. Del conjunto de documentos obtenidos se seleccionan aquellos que su categoría coincida con la más representativa del perfil de búsqueda

del usuario que realizó la consulta. Estos documentos son sometidos a un proceso de re-ranking junto a los documentos que se obtengan por el algoritmo de ranking de Apache Solr.

Finalmente, se devuelven como resultado los documentos que se obtienen del módulo y a continuación se completa el conjunto de respuesta con los documentos obtenidos por el ranking de Apache Solr. Estos últimos son reordenados según el valor de similitud que poseen con respecto a cada uno de los documentos que se seleccionan inicialmente por el módulo, asignando mayor relevancia a los que tienen valores más cercanos a uno.

En conclusión, el SRI devuelve como respuesta al usuario un conjunto de documentos que se acerca a las características de su perfil y, por tanto, a sus necesidades de búsqueda.

2.1.1 Categorización de documentos y consultas

La categorización de documentos permite consultar documentos indexados en Solr, adicionándole el campo categoría al cuerpo de los documentos y categorizarlos atendiendo a la información del campo "**contenido**" consumiendo el API de Weka y Solr. Para el caso de la categorización de consultas, en el momento de guardarla en la base de datos, se categoriza y se almacena la consulta con su categoría. En la figura 3 se representa el proceso de categorización para consultas y documentos.

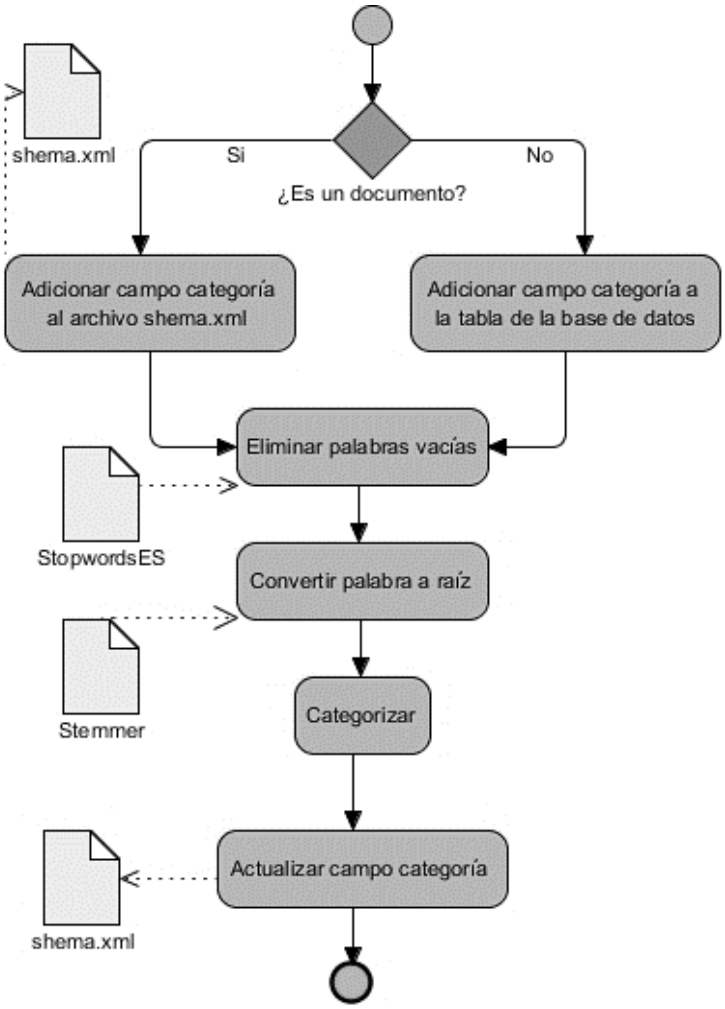

Figura 3. Pasos del proceso de categorización de consultas y documentos. Fuente: elaboración propia.

La funcionalidad de categorización cuenta con 2 etapas:

Etapa 1: Para el caso de los documentos, se plantea una nueva propuesta de diseño para el archivo ***schema.xml*** adicionándole el campo "**my_category**" a la estructura de los documentos a rastrear e indexar ya que ambas herramientas comparten el mismo archivo. La estructura del nuevo campo queda compuesta de la siguiente manera:

<field name= "my_category" type="string" indexed="true" stored= "true" default= "none">
`field name= "my_category":` Nombre del campo ("mi categoría", en español).
`type="string":` Tipo de dato que devuelve el campo creado (de tipo "palabra", en español).
`indexed="true":` Define si se desea que el campo sea indexado.
`stored= "true":` Define si se desea que el campo sea almacenado.
`default= "none":` Valor por defecto que devuelve el campo en los documentos indexados ("nada", en español).

Etapa 2: En el capítulo anterior se referencia la categorización supervisada de documentos, seleccionando este escenario para desarrollar el sistema propuesto. La funcionalidad que se propone permite categorizar consultas y documentos con información solo de naturaleza textual y en español ya que la Plataforma de Contenidos Unificados para Búsqueda Avanzada responde al dominio nacional donde la mayor parte de los sitios indexados son en español.

En caso de categorizar consultas o documentos en otro idioma sería necesario utilizar un archivo de *Stopwords* y un Stemmer (proveniente) de ese idioma en específico. Para ello, en el caso de los documentos, una vez que se modifique el archivo de configuración ***schema.xml*** y los documentos sean indexados con el nuevo campo "**my_category**", el índice generado por Solr será consultado a través de la biblioteca solrj, obteniendo así de los documentos los campos id (identificador), *keyword* (Palabras Claves) y *content* (contenido).

Luego se realizará el procesamiento de los datos, el cual comienza eliminando las palabras vacías de cada documento referenciado en un archivo nombrado StopwordsES, consumido por la funcionalidad de categorización, creando así una cadena de palabras relevantes para la categorización, separada por espacio y en letra minúscula. Consumiendo el API de la herramienta Weka, cada palabra relevante de la cadena es convertida a su raíz a través de un Stemmer para palabras en español. Para categorizar las consultas y los documentos se utiliza el algoritmo Naive Bayes en su versión Multinominal ya que este tiene en cuenta las repeticiones de las palabras; para esto la funcionalidad consume la colección de entrenamiento y las categorías o clases definidas que pueden poseer las consultas y documentos. Esta funcionalidad culminaría, en el caso de las consultas, cuando se actualice el campo categoría en la base de datos que almacena cada consulta realizada. En el caso de los documentos, cuando el campo "**my_category**" es actualizado con los resultados arrojados por el categorizador.

Para la categorización de consultas se procede de forma similar que para la categorización de documentos. Se categoriza la consulta atendiendo a la frase que la compone de forma similar a como se realiza en la etapa dos, explicada anteriormente. La figura 4 muestra el diagrama de actividades que modela el proceso de forma detallada.

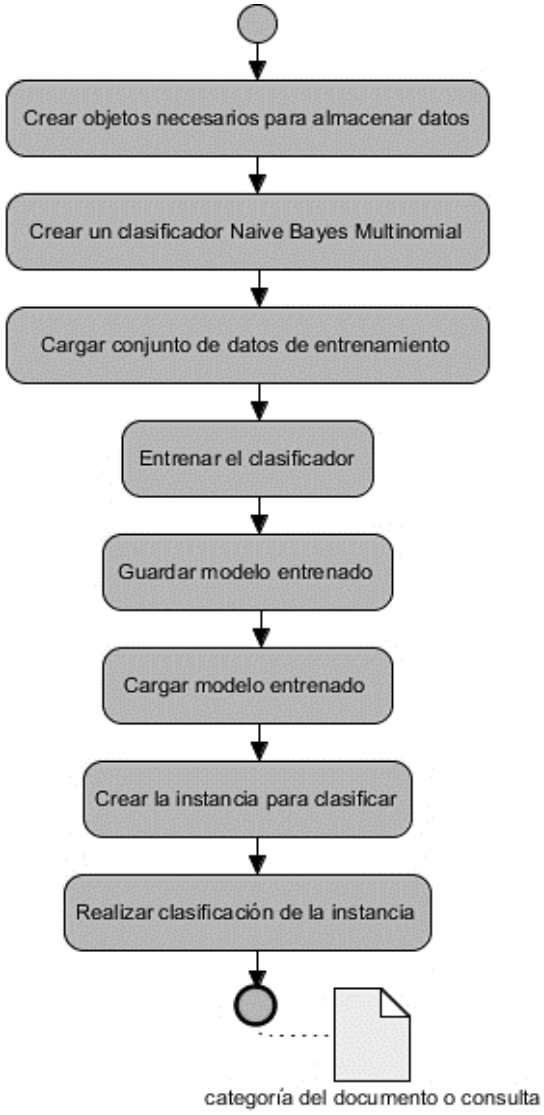

Figura 4. Proceso de categorización de consultas y documentos. Fuente: elaboración propia.

2.1.2 Creación y actualización del perfil de búsqueda de usuario

A efectos de la presente investigación se define el perfil de búsqueda de usuario como la obtención de sus preferencias y datos a partir del uso de la web. El modelado del perfil conlleva un proceso de aprendizaje continuo de la información proporcionada por su actividad en la web. En la figura 5 se muestra el proceso de creación y actualización del perfil de búsqueda de usuario.

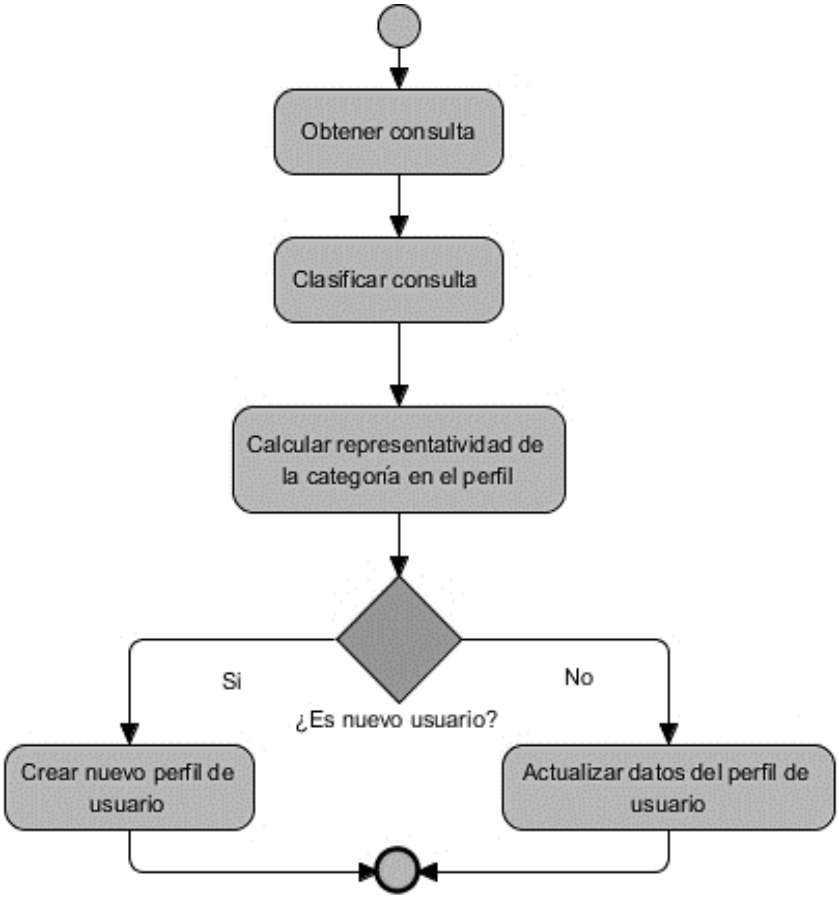

Figura 5. Creación y actualización del perfil de búsqueda de usuarios. Fuente: elaboración propia.

La minería de uso es una técnica clave en la creación y actualización del perfil de búsqueda de usuario, el cual se construye utilizando la información almacenada en la base de datos y las categorías definidas por el corpus Sogus.

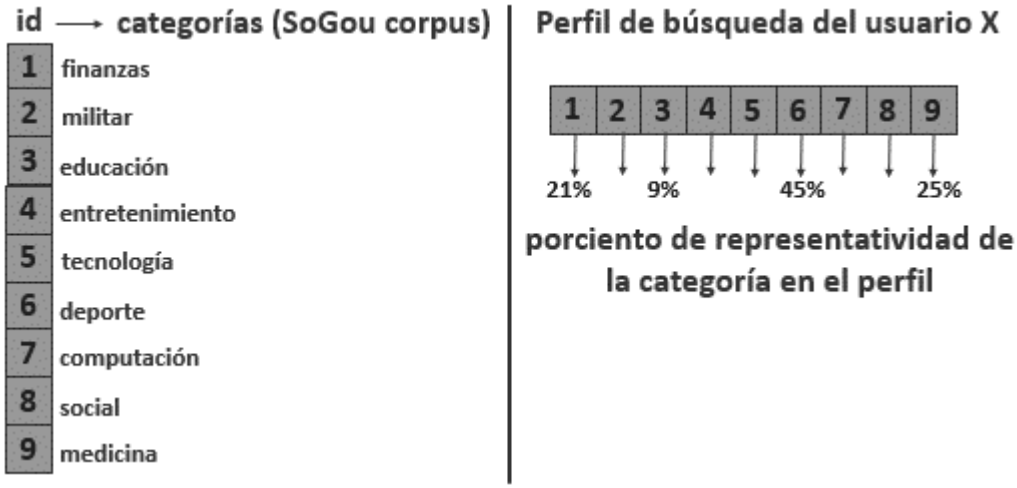

Figura 6. Perfiles de búsqueda de usuarios. Fuente: elaboración propia.

En la base de datos del motor de búsqueda se almacenan todas las consultas realizadas por los usuarios, los documentos clicados en cada conjunto de respuesta para cada consulta y las categorías de los mismos. Utilizando esta información se obtiene la cantidad de documentos clicados para cada una de las categorías definidas por el corpus manejado en la investigación. A

partir de esas cantidades se conoce el porciento que representa cada categoría del total de documentos clicados. Se asume entonces, que las categorías con mayor valor porcentual, representan las áreas sobre las que el usuario tiene mayor necesidad de información.

2.1.3 Cálculo de la similitud

Se pueden utilizar diferentes funciones para realizar el cálculo del coeficiente de similitud, entre las que destacan: producto escalar, coseno del ángulo entre los dos vectores (las más utilizadas), coeficiente de Dice y coeficiente de Jaccard (Monsalve, 2015).

Cuando se habla de modelos vectoriales se aprecia que la distancia más utilizada es la similitud coseno. Al aplicar la similitud coseno en un espacio vectorial de palabras (como una consulta) se aprecia que la misma refleja de buena manera la equivalencia semántica de dichas palabras. Es decir, aquellas palabras cuya similitud coseno sea mayor serán más similares que aquellas palabras cuya similitud coseno sea menor (Claassen y Grill, 2017).

En el contexto de los modelos vectoriales de palabras muchas veces la norma de los vectores está asociada a cuántas veces ocurre la expresión representada en el corpus. La distancia coseno mide el ángulo de separación entre dos vectores y por lo tanto no depende de la norma de los vectores. Es por esto que normalmente se usa la distancia coseno ya que es deseado que palabras o expresiones similares tengan muy poca distancia entre sí sin importar la frecuencia con la que aparecen en el corpus (Claassen y Grill, 2017).

En el artículo *Linguistic Regularities in Sparse and Explicit Word Representations,* se aborda como tema principal la utilización de los modelos vectoriales de palabras para modelar la similitud de palabras (Levy y Goldberg, 2014). Este trabajo se basa fuertemente en una serie de demostraciones y experimentos que demostraban la usabilidad de un modelo vectorial para la temática de similitud de palabras. En el marco teórico del mencionado artículo se explica detalladamente la utilización de modelos vectoriales de palabras para la detección de similitud entre palabras. Sobre todo, se define la distancia coseno como la norma que mejor conserva las propiedades de similitud de palabras.

El proceso de recuperación de información en la tecnología Orión está basado en el funcionamiento del modelo vectorial. Teniendo en cuenta las consideraciones expuestas por varios autores y reflejadas en la presente investigación, se escoge la función de similitud coseno para el desarrollo del módulo que se propone.

La figura 7 muestra el flujo necesario para realizar el cálculo de similitud entre consultas.

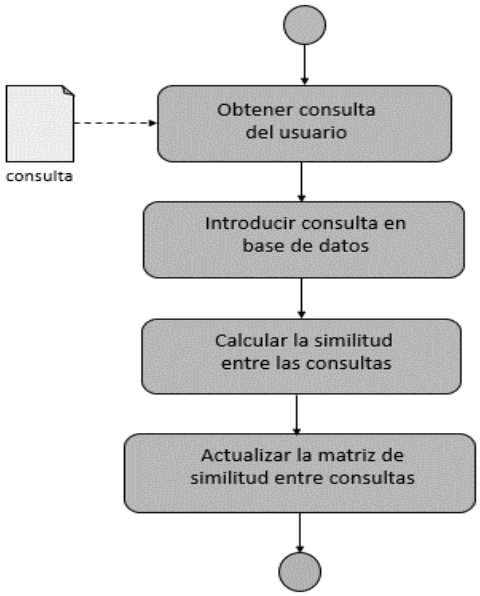

Figura 7. Flujo para el cálculo de similitud coseno entre consultas. Fuente: elaboración propia.

Para el cálculo de la similitud entre consultas, en la investigación realizada por Nikolaevna et al. (2016) y titulada “Comparación de medidas de similitud para desambiguación del sentido de las palabras utilizando *rankeo* de grafos” se define la ecuación siguiente:

$$similaridad = \cos(\theta) = \frac{\overrightarrow{x}}{\overrightarrow{x}} \cdot \frac{\overrightarrow{y}}{\overrightarrow{y}} = \frac{\sum_{i=1}^{|v|} x_i y_i}{\left(\sqrt{\sum_{i=1}^{|v|}(x_i)^2}\right)\left(\sqrt{\sum_{i=1}^{|v|}(y_i)^2}\right)} \quad (3)$$

Ecuación 3. Ecuación para cálculo de similitud coseno. Fuente: (Nikolaevna et al., 2016).

La investigación desarrollada utilizará para el cálculo de la similitud entre consultas la ecuación anterior.

La figura 8 muestra el flujo necesario para realizar el cálculo de similitud entre documentos.

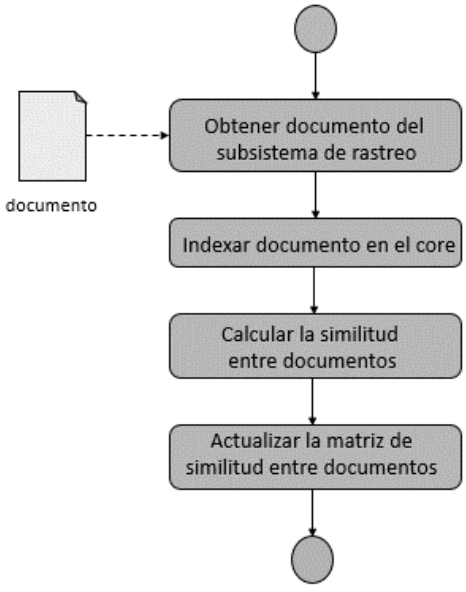

Figura 8. Flujo para el cálculo de similitud coseno entre documentos. Fuente: elaboración propia.

Para el cálculo de la similitud entre documentos se utilizará la ecuación:

$$\cos(d_i, d_j) = \frac{\sum_{k=1}^{n} w_k^i \cdot w_k^j}{\left(\sqrt{\sum_{k=1}^{n} \left(w_k^i\right)^2}\right)\left(\sqrt{\sum_{k=1}^{n} \left(w_k^j\right)^2}\right)} \quad (4)$$

Ecuación 4. Ecuación para cálculo de similitud coseno entre documentos. Fuente: (Artigas Fuentes et al., 2008).

En esta ecuación, d_i representa el nuevo documento indexado y d_j cada uno de los documentos almacenados en el SRI con el que se va a comparar.

File Edit View Tools Help

No limit

	id [PK] integer	identificador1 character varying(255)	identificador2 character varying(255)	similitud double precision
1	1	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/1262	0.454545454545455
2	2	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alfarer%C3%ADa	0.6828
3	3	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alfabeto_Morse	0.6756
4	4	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alfalfa	0.6392
5	5	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alfambra	0.706
6	6	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alfabeto_griego	0.7172
7	7	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alfafetoprote%C3%ADna	0.6456
8	8	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Agapito_Mayor	0.6704
9	9	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alacr%C3%A1n_Rey	0.5916
10	10	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alicia_Alonso	0.6408
11	11	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alacr%C3%A1n_de_corteza_de_Arizona	0.7068
12	12	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Aconitum_napellus	0.71
13	13	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alacr%C3%A1n_colorado	0.7388
14	14	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alicia_Hermida	0.6624
15	15	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alicia_Machado	0.7236
16	16	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Acorralada_(Telenovela)	0.643076923076923
17	17	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alfred_Hitchcock	0.6116
18	18	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alfred_Bester	0.6744
19	19	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alfred_Dreyfus	0.7312
20	20	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alfred_Adler	0.7796
21	21	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alfre_Woodard	0.4904
22	22	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alfred_Fried	0.85
23	23	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alexander_Craig_Aitken	0.7064
24	24	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Aleksandr_Dargomyzhski	0.6532
25	25	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alexander_Koch	0.58
26	26	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alonso_Ercilla_y_Z%C3%BA%C3%B1iga	0.6852
27	27	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alois_Hitler	0.6488
28	28	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alonso_de_Ojeda	0.7268
29	29	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alonso_Ojeda	0.7288
30	30	https://www.ecured.cu/%C3%81lava	https://www.ecured.cu/Alois_Alzheimer	0.6832

Figura 9. Captura del cálculo de similitud coseno entre documentos. Fuente: elaboración propia.

El valor de la similitud, al estar normalizado, implica que todos los valores estarán entre cero y uno, por lo tanto, se considerará que las consultas con valores de similitud más próximos a uno serán las más parecidas a la consulta del usuario. De igual forma, los documentos con valores más próximos a uno serán más similares.

Para el cálculo de la similitud entre cada uno de los vectores que representan los documentos (V_k^t) y el vector consulta (q), se utiliza en la presente investigación la similitud coseno. En este caso el vector consulta es definido como $q = (c_{1q}, c_{2q}, \ldots, c_{kq})$ donde c representa cada una de las coordenadas del vector en el espacio de k dimensiones. El vector del documento es representado como $dj = (v_{1j}, v_{2j}, \ldots, v_{kj})$. El modelo vectorial evalúa el grado de similitud del documento dj con relación a la consulta q como la correlación entre los vectores dj y q (Nuñez González et al., 2018).

$$\mathrm{Sim}(d_j, q) = \frac{\left(\sum_{i=1}^{t} v_{ij} \cdot c_{iq}\right)}{\left(\sqrt{\sum_{i=1}^{t} v_{ij}^2}\right)\left(\sqrt{\sum_{i=1}^{t} c_{ij}^2}\right)} \quad (5)$$

2.1.4 Proceso de *re-ranking*

El proceso de re-ranking se encargará de reordenar los resultados que se mostrarán al usuario. La figura 10 muestra el diagrama de flujo del proceso *de re-ranking* de los documentos.

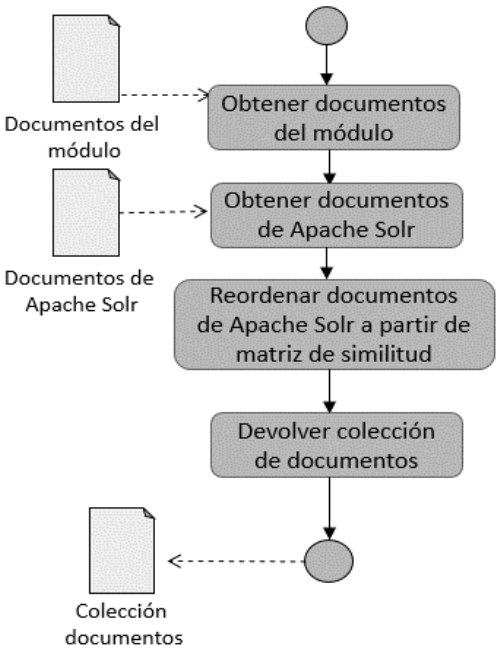

Figura 10. Proceso de *re-ranking* de documentos. Fuente: elaboración propia.

Inicialmente se reciben los documentos que poseen valores de similitud superiores a (0,9) con respecto a la consulta más similar a la ejecutada, y que además su categoría coincide con la más representativa del perfil de búsqueda de usuario. Estos serán los primeros que se devolverán al usuario. Para reordenar el resto de los documentos provenientes del algoritmo de ranking de Apache Solr que conformarán el conjunto de respuesta, se va obteniendo de la matriz de valores de similitud aquellos documentos que son más similares a los del primer conjunto. Este proceso termina cuando son devueltos reordenados todos los documentos y se devuelven como últimos resultados aquellos que poseen los menores valores de similitud. De esta forma se garantiza brindar a los usuarios resultados más precisos y mejor relacionados con sus preferencias de búsquedas.

2.2 Requisitos del sistema

La ingeniería de requisitos del software es un proceso de descubrimiento, refinamiento, modelado y especificación. Se refinan en detalle los requisitos del sistema y el papel asignado al software.

2.2.1 Requisitos funcionales

Una de las disciplinas, fases o etapas que se plantea en la mayoría de las metodologías para el desarrollo de software es la captura de requisitos, fundamental para concebir un sistema pues en ellos se expresa lo que la aplicación debe hacer.

Los requisitos funcionales describen las interacciones entre el sistema y su ambiente, en forma independiente a su implementación. El ambiente incluye al usuario y cualquier otro sistema externo con el cual interactúe el sistema. A continuación se presentan los definidos en esta investigación.

Tabla 9. Requisitos funcionales del sistema. Fuente: elaboración propia.

No.	Nombre	Descripción	Prioridad
1	Autenticar usuario	Premisa para el correcto funcionamiento del módulo. Permite al usuario legitimar su identidad.	Alta
2	Categorizar documentos indexados	Categoriza cada documento asignándole una categoría en dependencia de su contenido.	Alta
2.1	Cargar colección de entrenamiento	Entrenar categorizador y cargar fichero con la colección de entrenamiento.	Alta
2.2	Obtener documentos de Solr	Establecer flujo de comunicación con el servidor Solr.	Alta
2.3	Pre-procesamiento de los documentos	Pre-procesamiento de los documentos obtenidos del servidor de indexación Solr.	Alta
2.4	Categorizar documentos	Proceso de categorizar los documentos utilizando el algoritmo Naive Bayes.	Alta
2.5	Actualizar campo correspondiente a la categoría	Actualiza el campo de la categoría del documento atendiendo al resultado de la aplicación del algoritmo.	Alta
3	Categorizar consultas de usuarios	Categoriza cada consulta asignándole una categoría en dependencia del contenido de la frase que la compone.	Alta
3.1	Cargar colección de entrenamiento	Entrenar categorizador y cargar fichero con la colección de entrenamiento.	Alta
3.2	Obtener consultas de la base de datos	Establecer flujo de comunicación con la base de datos.	Alta
3.3	Pre-procesamiento de las consultas	Pre-procesamiento de las consultas obtenidas de la base de datos.	Alta
3.4	Categorizar consultas	Proceso de categorizar las consultas utilizando el algoritmo Naive Bayes.	Alta
3.5	Actualizar campo correspondiente a la categoría	Actualiza el campo de la categoría de la consulta atendiendo al resultado de la aplicación del algoritmo.	Alta
4	Obtener documentos clicados	Obtiene los documentos que han sido clicados por el usuario para ser utilizados en la creación del perfil de búsqueda.	Alta

5	Obtener consulta de usuario	Obtiene la consulta realizada proveniente de la interfaz de usuario para ser utilizada en la creación del perfil de búsqueda y en el cálculo de la similitud.	Alta
6	Crear/Actualizar el perfil de usuario	Calcula el porciento de representatividad de cada categoría definida a partir de las consultas del usuario.	Alta
7	Calcular similitud entre consultas	Calcula la similitud entre la consulta introducida por el usuario y todas las almacenadas en la base de datos.	Alta
8	Calcular similitud entre documentos	Calcula la similitud entre un nuevo documento indexado por el buscador y todos los documentos ya indexados.	Alta
9	Calcular similitud entre consulta y documentos	Calcula la similitud entre una nueva consulta y todos los documentos ya indexados.	Alta
10	Re-ranking de los resultados	Obtiene los documentos provenientes del proceso de ranking de Apache Solr y los documentos provenientes del módulo de procesamiento de contenidos. Realiza el re-ranking de los resultados.	Alta
11	Mostrar resultados al usuario	Luego del proceso de re-ranking se muestra al usuario el conjunto de documentos que conforman la respuesta del SRI.	Alta

2.2.2 Requisitos no funcionales

Requisitos de hardware:

- Los servidores deben tener prestaciones iguales o superiores a las siguientes: Procesador i5, 8 GB de memoria RAM y 1 TB de espacio en disco.

Requisitos de software:

- Las estaciones de trabajo de los usuarios deben contar con un navegador web que soporte HTML5, CSS3 y JavaScript.
- Los servidores donde se ejecutarán los procesos correspondientes al módulo deben tener una distribución GNU/Linux.
- Los servidores, dependiendo de las funciones que realizan, deben tener instalado las siguientes tecnologías: servidor web Nginx versión 1.9.11, PHP versión 7.0, gestor de base de datos MySQL versión 5.7.21, framework (entorno o marco de trabajo) de aplicaciones Spring Boot versión 5.0, Apache Solr versión 6.3.

Requisitos de rendimiento:

- Los tiempos de velocidad de procesamiento de la información y respuesta serán rápidos para lograr que el SRI sea efectivo.

Requisitos de licencia:

- Uso de la licencia PHP License.
- Uso de la licencia Apache Software.

2.3 Ambiente de desarrollo

El ambiente de desarrollo está conformado por varias herramientas y tecnologías que permiten la implementación de la solución que se propone.

2.3.1 Lenguaje de Modelado

UML es un lenguaje de modelado para visualizar, especificar, construir y documentar partes de un sistema software desde distintos puntos de vista. Puede usarse con cualquier proceso de desarrollo, a lo largo de todo el ciclo de vida y puede aplicarse a todos los dominios de aplicación y plataformas de implementación. También puede usarse en tareas áreas, como la ingeniería de negocio y modelado de procesos gracias a los mecanismos de adaptación/extensión mediante perfiles (García-Peñalvo et al., 2018).

UML tiene como características:

- Tecnología orientada a objetos.
- Viabilidad en la corrección de errores.
- Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos.
- Puede conectarse con lenguajes de programación (Ingeniería directa e inversa).
- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura pruebas, versiones, etc.).
- Cubre las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos.
- Es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas.
- Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.

2.3.2 Herramientas Case

2.3.2.1 Visual Paradigm

Para diseñar las modificaciones que se proponen a la base de datos se utilizó Visual Paradigm, una herramienta CASE de modelado multiplataforma que no se inclina por ninguna metodología específica. Además, ofrece un entorno de creación de diagramas para UML, con soporte para los 13 diagramas de la última versión de UML (UML 2.1).

Usa un lenguaje estándar común para todo el equipo de desarrollo que facilita la comunicación y está capacitado para la ingeniería directa e inversa en Java, C++, PHP, además de la capacidad de generación de código en estos lenguajes. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases. Específicamente presenta dos tipos de diagramas de modelación de bases de datos: entidad-relación (ERD) y mapeo objeto relacional (ORM). Los diagramas ERD modelan la base de datos a nivel físico y los ORM muestran la relación entre las clases (orientado a objeto) y la entidad (de la base de datos). Esta característica permite generar además del script (programas escritos para un entorno de tiempo de ejecución especial) de la base de datos, el código de las clases persistente (o clases de entidad) en el lenguaje escogido.

Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al momento, creando de forma simple toda la documentación. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama.

2.3.3 Programación del lado del cliente

Del lado del cliente se utilizan las siguientes herramientas y tecnologías libres:

- **Mozilla Firefox** es uno de los navegadores web más populares en el mundo, fue lanzado bajo una licencia de código abierto desde la versión 1.0. Con su gran cantidad de código fuente usando más de diez lenguajes de programación diferentes, el proyecto Mozilla Firefox ofrece muchas oportunidades para que los desarrolladores potenciales colaboren en su código fuente, por el cual más de 2,500 desarrolladores voluntarios contribuyeron a la base de código Mozilla Firefox (Frank, 2017).
  Entre sus principales características se encuentran: Es un software libre, con licencia GPL, como su código está abierto para todo el mundo, podemos adaptar las características de este programa a nuestras necesidades, puede ser utilizado libremente, repartir copias y distribuirlo sin incurrir en ningún delito, al tratarse de un programa libre, toda una comunidad de voluntarios, además de los miembros de la fundación Mozilla, trabaja para mejorarlo, que esté disponible su código fuente facilita que pueda ser traducido a cualquier lengua.
- **HTML** (HyperText Markup Language) es un lenguaje de marcación de elementos para la creación de documentos hipertexto y también un lenguaje útil para hacer un sitio web. En realidad, es la columna vertebral de cualquier sitio web. Consiste en diferentes etiquetas utilizadas para fines diferentes. Permite combinarse con otros lenguajes para definir el formato con el que se tienen que presentar las webs, como CSS y JavaScript. Este lenguaje se encarga de indicar a los navegadores cómo deben mostrar el contenido de una página web (Kadam y Onkar, 2015).
- **CSS** (*Cascading Style Sheets*), se refiere a "Hoja de estilos en cascada". Las hojas de estilo en cascada se utilizan para organizar el diseño de las páginas web. Se pueden usar para definir estilos de texto, tamaños de tabla y otros aspectos de páginas web que anteriormente

solo podían definirse en el HTML de una página. La razón básica de CSS es dividir el contenido de un documento web (escrito en cualquier lenguaje de marcado) de su gestión (que se escribe usando hojas de estilo en cascada). Hay muchos beneficios que se pueden eliminar a través de CSS, como la facilidad de acceso a los contenidos mejorados, la flexibilidad recuperada y, además, CSS proporciona un punto de control sobre las diferentes características de disposición del documento. También ayuda a reducir la dificultad y ayuda a ahorrar tiempo de arreglo general. CSS ofrece la opción de seleccionar diferentes esquemas de estilo y reglas de acuerdo con las necesidades y también permite que el documento HTML similar se presente en más de una técnica diferente (Kadam y Onkar, 2015).

- **JavaScript** se considera uno de los lenguajes de *scripting* más famosos de todos los tiempos. Por definición, es un lenguaje de *scripting* de la World Wide Web. El uso principal de JavaScript es agregar varias funcionalidades web, validaciones de formularios web, detecciones de navegador, creación de cookies, etc. Al ser uno de los lenguajes de scripting (lenguaje de programación que admite scripts) más populares y es por eso que es compatible con casi todos los navegadores web disponibles en la actualidad, como Firefox. Usamos el navegador Opera o Internet Explorer. Se considera uno de los lenguajes de scripting más poderosos en uso hoy en día. A menudo se utiliza para el desarrollo web del lado del cliente. También para hacer que la página web sea más interactiva y dinámica. Es un lenguaje de programación ligero y está integrado directamente en el código HTML (Kadam y Onkar, 2015).

2.3.3 Programación del lado del servidor

Para la programación del lado del servidor se emplean las siguientes tecnologías y herramientas.

- **PHP** es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante.

  PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. PHP puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

  Entre sus principales características se encuentran: Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos. Es considerado un lenguaje fácil de aprender, ya que en su desarrollo se simplificaron distintas especificaciones. El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al

navegador. Esto hace que la programación en PHP sea segura y confiable. Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.

Capacidad de expandir su potencial utilizando módulos. Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda. Es libre, por lo que se presenta como una alternativa de fácil acceso para todos. Permite aplicar técnicas de programación orientada a objetos (Chimborazo et al., 2013).

- **Nginx**: La web oficial define a Nginx como "*un servidor HTTP libre, de código abierto, de alto rendimiento y* proxy *inverso, así como un servidor IMAP / POP3*", Nginx es un poderoso servidor diferente a los tradicionales web server (servidor web) (Apache, Internet Information Server de Microsft) debido a que no está basado en subprocesos para gestionar solicitudes sino está basada en una arquitectura de eventos (asincrónica) mucho más escalable.

  La importancia de que una aplicación web sea rápida es vital actualmente tal es el caso que en recientes encuestas los usuarios esperan que la página se cargue en menos de dos segundos, y el 40 por ciento de ellos abandonará el sitio web si tarda más de 3 segundos. Nginx es una solución a este problema ya que es extremamente rápido y brilla incluso bajo carga alta.

  Permite acelerar las herramientas para cualquier aplicación, se podría combinar servidores que estén manejando tareas intensivas de datos en el backend (parte del desarrollo web que se encarga de que toda la lógica de funcione) con Nginx cumpliendo la función de servir los datos al usuario brindándole más rendimiento. Nginx posee un sinnúmero de características que lo convierte en uno de los servidores web más utilizados en el mercado actual.

  Es mucho más que solo un servidor web ya que también sirve como un servidor proxy inverso que es un tipo de servidor proxy que recupera recursos de los servidores en nombre de un cliente. Debido a su arquitectura basada en eventos y asíncrona su rendimiento es notable comparado con Apache e Internet Information Server que usan nuevos hilos por conexión y son bloqueantes por naturaleza.

  El balanceo de carga que provee Nginx puede permitir actuar muy rápido en frontend (parte de un sitio web que interactúa con los usuarios) encargándose solo de esta parte y delegando al backend a otro tipo de servidor, por ejemplo, apache. Soporta múltiples protocolos por ejemplo HTTP(S), Websockets, IMAP, POP, POP3, SMTP. Permite video *streaming* usando MP4/FLV/HDS/HLS (Zuñiga Tinizaray y Sánchez Carrión, 2017).
- **MySQL** es el segundo sistema de gestión de bases de datos relacionales de código abierto (RDBMS) más utilizado. El SQL significa Lenguaje de Consulta Estructurado. Es una opción popular de base de datos para su uso en aplicaciones web, y es un componente central de la pila de *software* de aplicaciones web LAMP de código abierto ampliamente utilizado LAMP significa Linux, Apache, MySQL, Perl / PHP / Python. Se usa a menudo para proyectos de

código abierto que requieren un sistema de gestión de bases de datos con todas las funciones (Gawas et al., 2015).

Gran parte del atractivo de MySQL se origina en su relativa simplicidad y facilidad de uso, que es habilitada por un ecosistema de herramientas de código abierto como phpMyAdmin. En el rango medio, MySQL se puede escalar al implementarlo en un hardware más potente, como un servidor de múltiples procesadores con gigabytes (unidad de almacenamiento de información) de memoria (Katkar, 2015).

- **Spring Boot** es un proyecto que permite crear aplicaciones web independientes que no tienen una dependencia en un servidor de aplicaciones o en un contenedor externo de servlets (clase en el lenguaje de programación Java, utilizada para ampliar las capacidades de un servidor). Las aplicaciones web creadas con Spring Boot incluyen una versión integrada de Tomcat o Jetty y producen un único archivo java ejecutable (JAR). Spring Boot también proporciona opciones de configuración para generar un archivo de aplicaciones web (WAR) que puede implementarse en un contenedor de servlets liviano o en un servidor de aplicaciones.

  Una característica innovadora clave de Spring Boot es la detección automática de dependencias y configuraciones predeterminadas por convención que minimiza la cantidad de configuración requerida para construir e implementar una aplicación web. Cuando las configuraciones predeterminadas proporcionadas no son adecuadas para su uso, la anulación es posible ya sea proporcionando una implementación de una clase base o interfaz y registrándose con el contexto de la aplicación o mediante la especificación en un archivo de configuración (Lundy, 2015).
- **Apache Solr** es un motor de búsqueda extensible para búsqueda de texto completo con código abierto, basado en el proyecto Apache Lucene. Su peculiaridad es que no es solo una solución técnica para la búsqueda, sino una plataforma que puede ampliarse, modificarse y personalizarse fácilmente para diversas necesidades, desde la búsqueda habitual de texto completo en un sitio web hasta un sistema distribuido para almacenar, recibir y analizar texto y otros datos con un poderoso lenguaje de consulta.

  Apache Lucene es el motor de búsqueda más famoso, originalmente se centró específicamente en la inserción en otros programas. Lucene es una biblioteca de búsqueda de texto completo a alta velocidad, escrita en Java. Proporciona capacidades de búsqueda avanzada, un buen sistema de creación y almacenamiento de índices que puede agregar, eliminar documentos y realizar la optimización simultáneamente con la búsqueda, así como la búsqueda paralela en un conjunto de índices que combinan los resultados.

  Apache Solr proporciona una velocidad de búsqueda e indexación muy alta, su tamaño de índice es uno de los más pequeños y tiene una gran capacidad de extensión. También puede actuar como un repositorio (Voit et al., 2017).

2.4 Diseño de la base de datos

A través del modelo de datos se definen los conceptos que se manejan en el sistema y que sirven para describir la estructura de la base de datos diseñada. Es decir, en dicho modelo se representan los datos, sus atributos y tipos, sus relaciones y las restricciones que deben cumplirse sobre ellos.

A continuación, en la Figura 11, se presenta el modelo de datos para la solución propuesta donde se definen las tablas: perfil_busqueda_usuario donde se almacenan los identificadores de las consultas y documentos clicados, usuario_documentos que representa la relación entre los usuarios y los documentos clicados por este y la tabla identificadores_consultas que hace alusión a los identificadores de cada una de las consultas realizadas. Estas son las tablas que se proponen incorporar a la base de datos que utiliza el SRI.

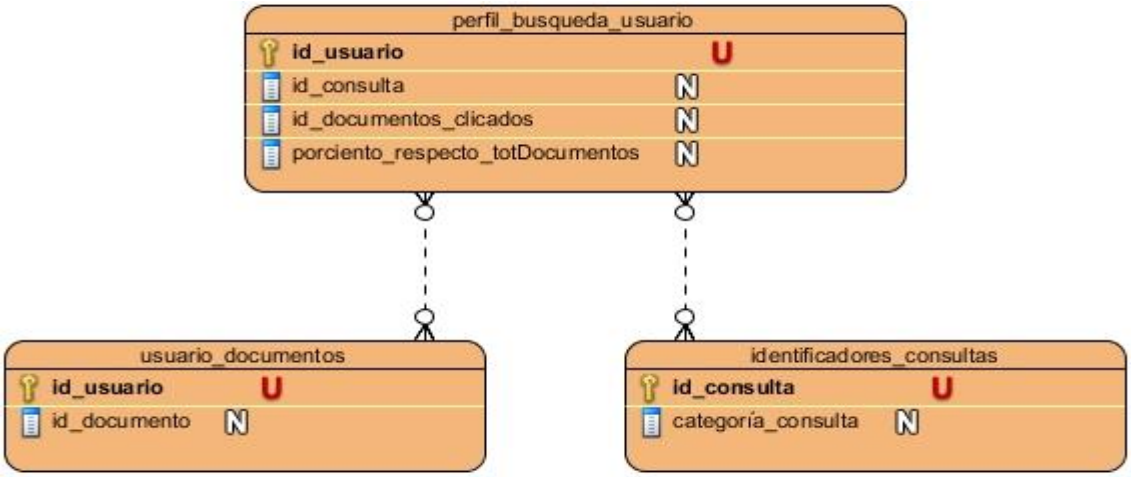

Figura 11. Diagrama del modelo Entidad-Relación de la base de datos. Fuente: elaboración propia.

2.5 Arquitectura del módulo

La Arquitectura del Software representa la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán. El estudio de varios modelos de RI arrojó que existen vacíos teóricos que no permiten definir una guía para diseñar arquitecturas de hardware (conjunto de elementos físicos o materiales que constituyen una computadora o un sistema informático) adaptables a SRI.

De las investigaciones de autores como Palomino et al. (2004) se deriva que las arquitecturas analizadas son muy generales y enfocadas en los tres componentes principales: rastreo, indexación y visualización, lo que limita el proceso de desarrollo y despliegue de otros componentes. No se identifican de forma precisa las características de hardware para cada servidor y el número de servidores destinados al despliegue del SRI no es especificado en relación a un tamaño de la web. Las implementaciones de la mayoría de buscadores comerciales no están disponibles al público, a no ser que sean de carácter experimental. En la figura 12 se muestran las arquitecturas públicas de Google, Altavista y Harvest.

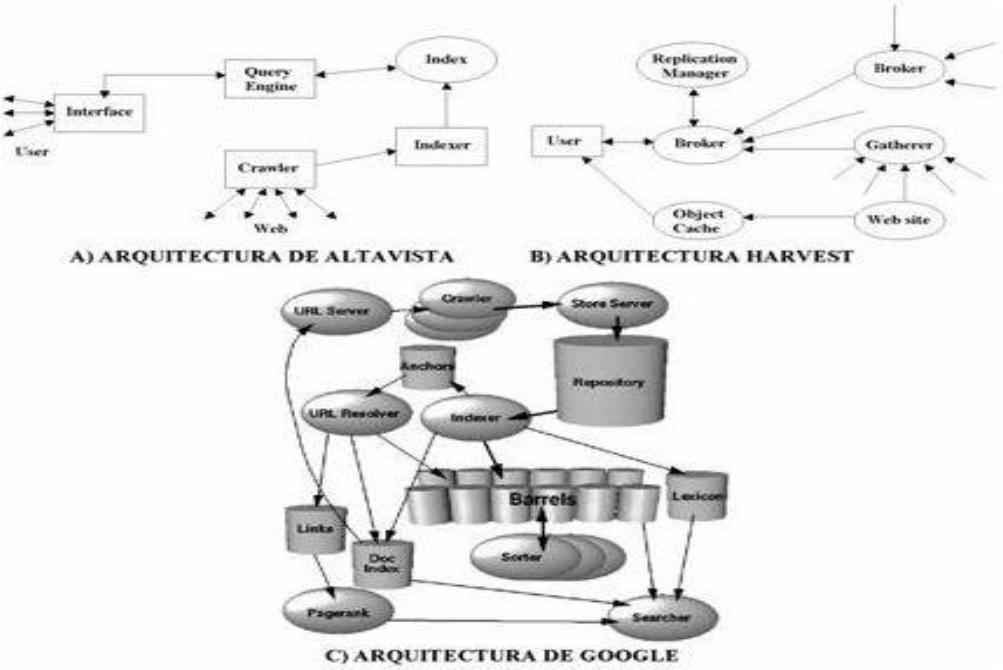

Figura 12. Arquitecturas públicas de Google, Altavista, Harvest. Fuente: (Palomino et al., 2004)

Al no disponer de arquitecturas de referencia, esta investigación propone la modificación de la arquitectura de la Plataforma de Contenidos Unificados para Búsqueda Avanzada basada en la incorporación de nuevos componentes. Esto, debido a que la arquitectura que usa actualmente no cuenta con los elementos necesarios para realizar los procesos de la solución que se propone. La arquitectura de la plataforma está basada en el patrón arquitectónico Modelo-Vista-Controlador (MVC) y se propone incorporar los elementos que se representan en la figura 13.

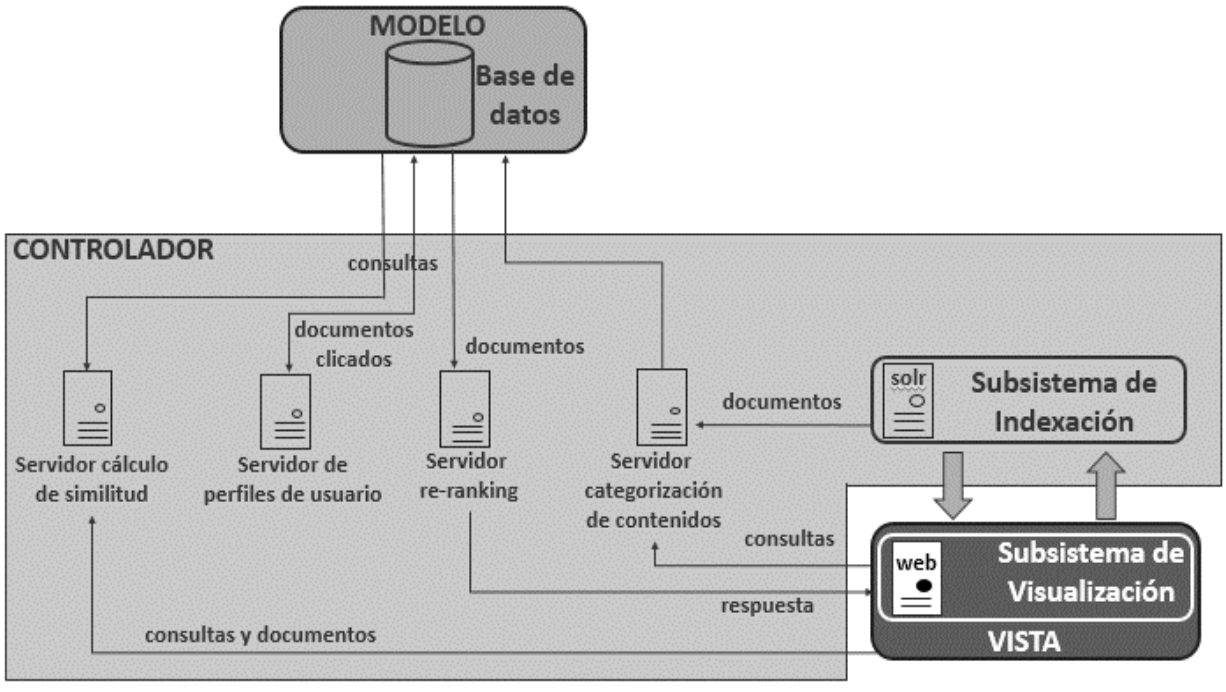

Figura 13. Arquitectura del módulo de procesamiento de contenidos. Fuente: elaboración propia.

Un servidor de categorización de contenidos que se apoya en un balanceador de carga para distribuir el proceso de categorización en tres servidores esclavos, uno para las consultas y dos

para los documentos. Se utilizan dos para documentos porque la cantidad de estos supera en gran medida la cantidad de consultas. El servidor de categorización recibirá los documentos del subsistema de indexación, donde están todos almacenados.

Un servidor para la creación y actualización del perfil de usuario, que recibe del subsistema de visualización todas las consultas realizadas y los documentos clicados. El perfil se almacena en la base de datos y se actualiza siempre que el usuario ejecute una consulta en el SRI.

Un servidor para el cálculo de similitud de consultas y documentos. Se utiliza un balanceador de carga para distribuir el proceso en tres servidores esclavos, uno para el cálculo de similitud entre consultas, otro para el cálculo de similitud entre documentos y un tercero para el cálculo de similitud entre consultas y documentos.

Un servidor para el re-ranking de los resultados. Este recibirá un conjunto de documentos como resultado del proceso de ranking que realiza Apache Solr y otro conjunto de documentos como resultado del procesamiento de contenidos que realiza el módulo propuesto. Los documentos de ambos conjuntos son sometidos a un proceso que se encargará de realizar un nuevo ranking y devolverlos al usuario.

2.6 Patrones de diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas habituales que se presentan durante el desarrollo de software. Es necesario, para que una solución sea considerada un patrón, que cumpla con ciertas características como la comprobación de su efectividad resolviendo problemas similares en ocasiones anteriores. También que sea reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias. En la solución que se propone se utilizan algunos patrones generales de software para asignación de responsabilidades (GRASP, por sus siglas en ingles) y los patrones pandilla de los cuatro (GoF, por sus siglas en ingles).

Los patrones GoF se descubren como una forma indispensable de enfrentarse a la programación a raíz del libro “Design Patterns-Elements of Reusable Software” de Erich Gamma, Richard Helm, Ralph Jonson y John Vlissides (Gamma et al., 1995). A partir de entonces estos patrones son conocidos como los patrones de la pandilla de los cuatro. A continuación se muestran algunos patrones identificados en el desarrollo de la solución propuesta.

- *Iterator* (Iterador): Permite realizar recorridos sobre objetos compuestos independientemente de la implementación de estos. Proporciona un acceso secuencial a una colección de objetos sin que los clientes se preocupen de la implementación de esta colección.
- Memento (Recuerdo): Permite volver a estados anteriores del sistema. Salvaguarda y restaura el estado de un objeto. El categorizador se guarda y restaura en un fichero como un objeto serializado.

Según Larman (2003), los patrones GRASP constituyen un apoyo para la enseñanza que ayuda a uno a entender el diseño de objetos esencial, y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable. Este enfoque para la comprensión y utilización de los principios de diseño se basa en los patrones de asignación de responsabilidades. En la solución que se propone se puede apreciar el uso de los siguientes patrones de diseño GRASP:

- Experto: El patrón Experto en Información se utiliza con frecuencia en la asignación de responsabilidades; es un principio de guía básico que se utiliza continuamente en el diseño de objetos. El Experto no pretende ser una idea oscura o extravagante; expresa la "intuición" común de que los objetos hacen las cosas relacionadas con la información que tienen. Las clases que intervienen en el componente de categorización de documentos cumplen con este patrón ya que todas son expertas en su función (Larman, 2003).
- Creador: El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, una tarea muy común. La intención básica del patrón Creador es encontrar un creador que necesite conectarse al objeto creado en alguna situación. Eligiéndolo como el creador se favorece el bajo acoplamiento (Larman, 2003). En la clase **`Principal`** del componente de categorización de documentos se evidencia este patrón ya que este es el que se encarga de crear los objetos necesarios de las otras clases existentes.
- Bajo Acoplamiento: El patrón Bajo Acoplamiento impulsa la asignación de responsabilidades de manera que su localización no incremente el acoplamiento hasta un nivel que lleve los resultados negativos que pueden producir un acoplamiento alto. El Bajo Acoplamiento soporta el diseño de clases que son más independientes, lo que reduce el impacto del cambio. No se puede considerar de manera aislada a otros patrones como el Experto o el de Alta Cohesión, sino que necesita incluirse como uno de los diferentes principios de diseño que influyen en una elección al asignar una responsabilidad (Larman, 2003). Las clases **`SolrCommunication`**, **`Categorizer`**, **`Cleaner`** no tienen relaciones entre ellas, solo se comunican con la clase **`Principal`**, pueden ser reutilizadas como componentes para otros sistemas sin que ocurran grandes impactos por los cambios.
- Alta Cohesión: La información que almacena una clase debe de ser coherente y debe estar relacionada con la clase (Larman, 2003). Las clases **`SolrCommunication`**, **`Categorizer`**, **`Cleaner`** utilizan dicho patrón ya que cada una de las clases es totalmente coherente con sus funciones.

Consideraciones finales del Capítulo 2

En el presente capítulo se han expuesto los componentes del módulo que se propone como solución para realizar el procesamiento de contenidos. También se presentaron los requisitos de hardware y software, la arquitectura del módulo y el diseño de la base de datos. Se obtienen las siguientes consideraciones finales del Capítulo 2:

1. La implementación de la categorización de documentos y consultas utilizando el algoritmo Naive Bayes, es una de las funcionalidades más importantes y constituye un aporte a la concepción del módulo para la plataforma.
2. La utilización de la minería web facilitó la implementación del módulo y el estudio del comportamiento de los usuarios.
3. La propuesta de modificación de la arquitectura y base de datos es fácil de integrar a la plataforma y no representan un problema para su funcionamiento.

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

Introducción

En el presente capítulo se describe la validación de los resultados de la investigación y se explican los métodos utilizados. Para validar la propuesta se realizó un cuasi-experimento y se aplicaron otras técnicas como la escala de Likert fundamentada en la opinión de los expertos para validar la solución propuesta y la técnica IADOV.

3.1 Diseño de la validación

Se realiza la validación del módulo empleando métodos cuantitativos, cualitativos y experimentales. Para la validación se emplearon las siguientes técnicas y métodos:

- Criterio de expertos empleando el escalamiento de Likert: se utilizó para validar el módulo desarrollado.
- Satisfacción de potenciales usuarios y usuarios desarrolladores mediante la técnica IADOV: fue aplicado para evaluar el nivel de satisfacción con el módulo propuesto, de dos grupos de usuarios diferentes, los usuarios potenciales y los usuarios desarrolladores.
- Experimentación: mediante un cuasi-experimento se evaluó la capacidad del módulo para mejorar la exactitud de las respuestas que se brindan a los usuarios de un sistema de recuperación de información.

Finalmente, se concibió una triangulación metodológica inter-métodos como procedimiento de control para evaluar la confiabilidad de los resultados obtenidos. Permitió constatar la inexistencia de contradicción en los resultados arrojados luego de aplicar las técnicas y métodos de manera independiente.

3.2 Valoración de los expertos

El juicio de expertos, como herramienta de validación, permite obtener valoraciones sobre asuntos afines a una propuesta de solución para un determinado problema. Como método para el procesamiento estadístico de estas valoraciones, se aplica en esta investigación la escala psicométrica creada por Rensis Likert en 1932 (Boone y Boone, 2012). Esto, a través de un cuestionario con el objetivo de conocer el nivel de acuerdo o desacuerdo con el módulo de procesamiento de contenidos para la Plataforma C.U.B.A.

Los indicadores seleccionados para ser evaluados por los expertos pueden observarse en la Sección II, del cuestionario del Anexo 1. Como expertos se seleccionaron 15 personas que, a criterio del autor, cumplen los requisitos de expertos y que están asociados a los temas relacionados con la base teórica y práctica de la recuperación de información.

A los expertos se les aplicó la encuesta para determinar el coeficiente de competencia de forma individual, como se puede ver en el Anexo 2. Se realizó una valoración inicial de los expertos. Se tuvieron en cuenta los siguientes aspectos: título universitario, categoría docente, años de

experiencia en el área, el nivel de dominio sobre el tema que se encuesta y las fuentes de argumentación. Todos cumplen los requisitos de expertos y tienen experiencia en actividades relacionadas con la recuperación de información.

Se determinó el nivel de competencia de cada experto, para asegurar la confiabilidad de las respuestas, mediante el cálculo de su coeficiente de competencia. El procedimiento empleado para determinar el coeficiente de competencia de los candidatos a expertos puede ser consultado en el Anexo 3, así como los resultados arrojados luego de aplicada la encuesta en el Anexo 4.

Como resultado se obtiene que los 15 encuestados tienen un nivel de competencia Alto o Medio. Los resultados de la distribución de los expertos según su nivel de competencia se muestran en la tabla 10.

Tabla 10. Distribución de los expertos según el nivel de competencia. Fuente: elaboración propia.

Nivel de competencia	Cantidad	Porciento
Alta	12	80,00%
Media	3	20,00%
Baja	0	0%
Total	15	100%

Al analizar el comportamiento de los niveles de competencia se determinó escoger los 15 expertos debido a que su nivel de competencia es adecuado para los elementos teóricos a analizar en el módulo desarrollado, siendo una cantidad apropiada para garantizar la confiabilidad de los resultados. La caracterización de los expertos es: 6,67% posee la categoría de Máster en Ciencias, el 20% se preparan para obtener el grado científico de Doctor en Ciencias en el presente año. El 6,67% posee la categoría de Profesor Auxiliar y el 13,33% es Asistente. La media de años de experiencia es de cinco años. El 47% de los expertos está vinculado a la docencia.

Las preguntas del cuestionario diseñado, que se muestra en el Anexo 1, están enfocadas a obtener las valoraciones de los expertos en función de los indicadores definidos. Las preguntas representan ocho (8) aspectos relevantes del módulo desarrollado. El experto expresa su valoración de cada indicador mediante la siguiente escala: 5- muy de acuerdo (MA), 4- de acuerdo (DA), 3- ni de acuerdo ni en desacuerdo (Sí-No), 2- en desacuerdo (ED) y 1 - completamente en desacuerdo (CD). Posteriormente, se procesan los resultados mediante la escala Likert. Con esta técnica son calculados los porcientos de concordancia de los expertos con cada una de las respuestas para los planteamientos formulados, mostradas en el Anexo 5. Luego se calcula un índice porcentual (IP) que integra en un solo valor la aceptación de cada planteamiento por los evaluadores mediante la siguiente fórmula:

La figura 14 muestra que el índice porcentual relacionado con la valoración de los expertos, sobre los aspectos planteados, es superior a 87% en todos los casos.

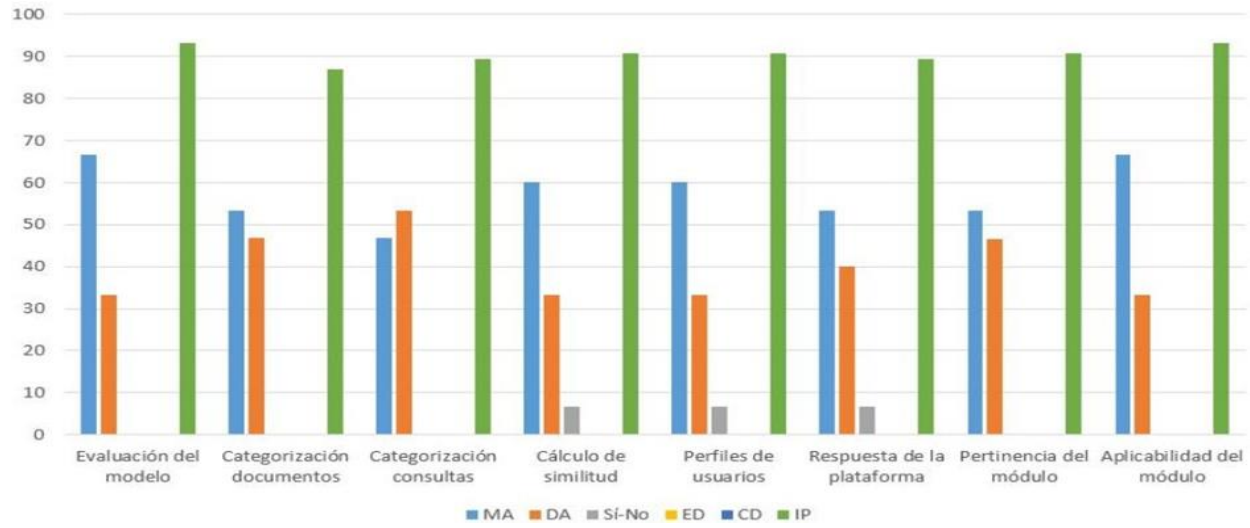

Figura 14. Valoración de los expertos sobre el módulo. Fuente: elaboración propia.

El procesamiento realizado a través del escalamiento de Likert evidencia que tanto los elementos teóricos como las características de los componentes del módulo, tienen una alta valoración por parte de los expertos. Durante el proceso se constataron criterios favorables para el uso y aplicación del módulo para aumentar la exactitud de respuesta en la Plataforma C.U.B.A.

3.3 Satisfacción de usuarios con el módulo

La técnica de Iadov constituye una vía indirecta para el estudio de la satisfacción, ya que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre tres preguntas cerradas y dos abiertas. Estas tres preguntas se relacionan a través de lo que se denomina el Cuadro Lógico de Iadov e indica la posición de cada sujeto en la escala de satisfacción.

En la investigación fue empleada para medir la satisfacción de dos tipos de usuarios, aquellos que participan en el proceso de desarrollo del motor de búsqueda y los usuarios potenciales. Para el desarrollo de esta técnica se aplicaron dos encuestas diferentes para ambos grupos después de interactuar con el módulo incorporado a la plataforma.

Tabla 11. Cuadro Lógico de Iadov para usuarios desarrolladores. Fuente: elaboración propia.

	¿Considera usted que la Plataforma C.U.B.A. deba continuar brindando servicios sin el módulo de procesamiento de contenidos que permita mejorar los resultados que ofrece?								
	No			No sé			Sí		
	¿Utilizaría este módulo para procesar contenidos en la Plataforma C.U.B.A. y mejorar las respuestas que se brindan al usuario?								
¿Le satisface el módulo de procesamiento de contenidos para mejorar las respuestas que se brindan al usuario, a partir de los componentes desarrollados y la salida que provee en la Plataforma C.U.B.A.?	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me satisface mucho	1	2	6	2	2	6	6	6	6
No me satisface tanto	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me insatisface más de lo que me satisface	6	3	6	3	4	4	3	4	4
No me satisface nada	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	6	3	3	3	6	3	4

Tabla 12. Cuadro Lógico de Iadov para usuarios potenciales. Fuente: elaboración propia.

	¿Considera usted que la Plataforma C.U.B.A. cumple las expectativas de los usuarios como motor de búsqueda?								
	No			No sé			Sí		
	¿Recomendaría el uso de la Plataforma C.U.B.A. a otros usuarios?								
¿Le satisface los resultados de búsqueda y el orden en que son mostrados en la Plataforma C.U.B.A.?	Sí	No sé	No	Sí	No sé	No	Sí	No sé	No
Me satisface mucho	1	2	6	2	2	6	6	6	6
No me satisface tanto	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me insatisface más de lo que me satisface	6	3	6	3	4	4	3	4	4
No me satisface nada	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	6	3	3	3	6	3	4

La escala de satisfacción se muestra a continuación:

1 -Clara satisfacción.

2-Más satisfecho que insatisfecho.

3-No definida.

4-Más insatisfecho que satisfecho.

5-Clara insatisfacción.

6-Contradictoria.

Para medir el grado de satisfacción de ambos grupos se tomaron muestras de 15 usuarios. En el caso de los usuarios desarrolladores se tuvo en cuenta los años de experiencia, la experiencia en el trabajo en SRI, conocimientos en los aspectos relacionados con la recuperación de información, entre otros elementos. El resultado de la satisfacción individual fue el siguiente:

Tabla 13. Satisfacción individual de usuarios desarrolladores. Fuente: elaboración propia.

Resultado	Cantidad	%
Máximo de satisfacción	11	73,33%
Más satisfecho que insatisfecho	3	20,00%
No definida	1	6,67%
Más insatisfecho que satisfecho	0	0%
Clara insatisfacción	0	0%
Contradictoria	0	0%

Tabla 14. Satisfacción individual de usuarios potenciales. Fuente: elaboración propia.

Resultado	Cantidad	%
Máximo de satisfacción	12	80,00%
Más satisfecho que insatisfecho	2	13,33%
No definida	1	6,67%
Más insatisfecho que satisfecho	0	0%
Clara insatisfacción	0	0%
Contradictoria	0	0%

Con el objetivo de obtener el índice de satisfacción grupal (ISG) se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y - 1 de la siguiente forma:

Tabla 15. Escala numérica para el ISG. Fuente: elaboración propia.

Escala	Nivel de satisfacción
1	Máximo de satisfacción
0,5	Más satisfecho que insatisfecho
0	No definido y contradictorio
-0,5	Más insatisfecho que satisfecho
-1	Máxima insatisfacción

Luego es posible calcular el ISG a partir de la siguiente ecuación:

$$ISG = \frac{A(+1) + B(+0{,}5) + C(0) + D(-0{,}5) + E(-1)}{N} \quad (5)$$

Donde:

- A, B, C, D, E, representan el número de sujetos con su índice individual.
- N representa el número total de sujetos del grupo.

El índice grupal arroja valores entre + 1 y - 1. Los valores que se encuentran comprendidos entre - 1 y - 0,5 indican insatisfacción; los comprendidos entre - 0,49 y + 0,49 evidencian contradicción y los que caen entre 0,5 y 1 indican que existe satisfacción. Iadov contempla además preguntas complementarias de carácter abierto.

El ISG obtenido para los usuarios desarrolladores fue el siguiente:

$$ISG = \frac{11(+1) + 3(+0{,}5) + 1(0) + 0(-0{,}5) + 0(-1)}{15} = 0{,}833 \quad (6)$$

El ISG obtenido para los usuarios potenciales fue el siguiente:

$$ISG = \frac{12(+1) + 2(+0{,}5) + 1(0) + 0(-0{,}5) + 0(-1)}{15} = 0{,}867 \quad (7)$$

Como se aprecia, los índices de satisfacción grupal son de 0,833 y 0,867 respectivamente, lo que significa una clara satisfacción con la propuesta y reconocimiento de su utilidad en el mejoramiento de la exactitud de los resultados de la Plataforma C.U.B.A. Sobre las dos preguntas complementarias de carácter abierto realizadas a los usuarios desarrolladores encuestados respondieron lo siguiente:

Pregunta 4:

- Incorporar la utilización de ontologías en el proceso de categorización.

Pregunta 5:

- Permite agrupar los contenidos por categorías reduciendo el tiempo de búsqueda sobre el cúmulo de información.
- Posibilita el reconocimiento de las categorías sobre las que descansa todo el contenido indexado.

En las figuras 15 y 16 se representan los porcientos de satisfacción obtenidos luego de calculado el ISG, no encontrándose los dos últimos colores por estar en 0%.

Figura 15. Satisfacción de usuarios desarrolladores con el módulo. Fuente: elaboración propia.

Figura 16. Satisfacción de usuarios potenciales con el módulo. Fuente: elaboración propia.

3.4 Resultados experimentales en la aplicación del módulo

La satisfacción del usuario con respecto a los resultados de búsqueda que se le ofrecen depende totalmente del funcionamiento del SRI. El funcionamiento de un SRI se evalúa a través de las métricas precisión y exhaustividad por tanto se decide realizar un experimento para tasar el comportamiento de las mismas. Se trabajó con 15 usuarios que hacen uso de este buscador.

Para el experimento se utilizaron 150 documentos que fueron categorizados previamente y dos grupos (grupo de control y grupo experimental) de 20 consultas. Se realiza la primera fase del experimento con el conjunto de documentos y los usuarios, registrándose las cantidades de documentos relevantes (DocRelev), documentos relevantes recuperados (DocRelevRec) y documentos recuperados (DocRec). Un grupo de expertos decide cuáles son los documentos relevantes a partir de la consulta introducida por el usuario. Las cantidades de DocRec y DocRelevRec son tomadas a partir de la salida que produce el motor de búsqueda.

En la segunda fase del experimento se trabaja con el conjunto de documentos categorizados y se le pide al usuario que establezca previamente su perfil de búsqueda. Del conjunto de documentos se seleccionaron por parte de los expertos aquellos que se consideraron relevantes con respecto a una consulta que introduce el usuario y al perfil de búsqueda del mismo. Se procede a registrar las cantidades de DocRec, DocRelev y DocRelevRec. Al contar con las cantidades de documentos relevantes, documentos relevantes recuperados y documentos recuperados se procedió al cálculo de las métricas utilizando las siguientes ecuaciones:

$$Precisión = \frac{Documentos\ relevantes\ recuperados}{Documentos\ recuperados} \quad (8)$$

Ecuación 8. Ecuación para cálculo de precisión. Fuente: elaboración propia.

$$Exhaustividad = \frac{Documentos\ relevantes\ recuperados}{Documentos\ relevantes} \quad (9)$$

Ecuación 9. Ecuación para cálculo de exhaustividad. Fuente: elaboración propia.

En la tabla 16 se pueden observar los resultados obtenidos en el cálculo de las métricas antes y después de la utilización del módulo.

Tabla 16. Valores de precisión y exhaustividad. Fuente: elaboración propia.

Precisión promedio sin la utilización del módulo	Precisión promedio con la utilización del módulo	Exhaustividad promedio sin la utilización del módulo	Exhaustividad promedio con la utilización del módulo
0,327	0,808	0,386	0,503

3.5 Análisis estadístico de los resultados experimentales.

Se aplicó la prueba de normalidad Shapiro-Wilk para comprobar que los datos obtenidos de precisión y exhaustividad se ajustan a una distribución Normal. El valor $p > 0,05$ de la prueba indica la normalidad de los datos. También se aplicó la prueba estadística paramétrica t-student, con el objetivo de comparar dos muestras, a partir de sus medias, para comprobar si los resultados son estadísticamente diferentes. Las muestras en este caso fueron los resultados de precisión y exhaustividad antes y después de la utilización del módulo. Se aplica la prueba a los resultados de exhaustividad, donde se obtiene que $p < 0,5$ y por tanto se rechaza la hipótesis nula, existiendo diferencias estadísticamente significativas en las muestras comparadas. A continuación, se aplicó a la precisión, donde el valor de p fue $< 0,5$ indicando diferencia significativa entre las medias. El análisis estadístico de los datos evidencia la mejora de la calidad de los resultados de búsqueda de acuerdo a la precisión y en menor medida a la exhaustividad.

3.6 Resultados de la triangulación metodológica de los métodos aplicados

A criterio de Ramírez (2016), una técnica usada para tomar múltiples puntos de referencia y localizar una posición desconocida es la triangulación metodológica. Esta permite disminuir el sesgo que se produce al comparar resultados obtenidos en la cuantificación de variables mediante un método cuantitativo, las tendencias y dimensiones que surgen de la aplicación de métodos cualitativos (Valencia, 2013). Esta técnica es también definida como la combinación de múltiples métodos en un estudio del mismo objeto o evento para abordar mejor el fenómeno que se investiga (Hussein, 2015).

En la presente investigación se utiliza la triangulación de métodos, con la intención de validar los datos recolectados tanto cualitativos como cuantitativos a partir de la aplicación de los métodos experimento, escalamiento de Likert y la técnica Iadov.

Tabla 17. Resultados de la triangulación metodológica. Fuente: elaboración propia.

Objetivo a evaluar	Métodos cuantitativos	Métodos cualitativos	Conclusión
Desarrollar un módulo de procesamiento de contenidos	**Pre-experimento:** los valores de exhaustividad y precisión obtenidos	**Criterio de expertos**: alta valoración de los expertos con el módulo desarrollado. En todos los indicadores se	No existe contradicción en los resultados

para mejorar los resultados de búsqueda en la Plataforma C.U.B.A.	validan el correcto funcionamiento del módulo.	obtuvo un IP superior al 87,00%. **Iadov**: alto grado de satisfacción de los usuarios potenciales y desarrolladores. Se obtuvo un ISG de 0,867 y 0,833 respectivamente.	arrojados por los métodos aplicados. Se comprueba la capacidad del módulo de procesamiento de contenidos para mejorar los resultados de búsqueda en la Plataforma C.U.B.A. de acuerdo a la precisión y la exhaustividad.

Consideraciones finales del Capítulo 3

Luego de aplicar los métodos científicos con el objetivo de validar la propuesta de solución al problema planteado se concluye que:

1. El juicio de los expertos fue positivo basado en criterios como el funcionamiento de los componentes del módulo, las respuestas brindadas al usuario y la pertinencia y aplicabilidad del mismo.
2. La aplicación de la técnica de Iadov proporcionó el nivel de satisfacción de los usuarios con el módulo evidenciando la posibilidad de su implantación en la Plataforma C.U.B.A.
3. El experimento arrojó valores satisfactorios para la métrica precisión y en menor medida para la métrica exhaustividad debido a que son inversamente proporcionales, comprobando de esta forma que se contribuye a mejorar los resultados de búsqueda en la Plataforma C.U.B.A.

CONCLUSIONES

Una vez concluida la investigación se arribó a las siguientes conclusiones:

- Los sistemas de recuperación de información estudiados realizan procesos importantes como la categorización de consultas y documentos, el cálculo de similitud entre consultas y el uso del perfil de búsqueda de usuario para mejorar los resultados que se brindan.

- La utilización de técnicas de minería web agilizó los procesos de categorización y facilitó la implementación del módulo.

- La propuesta de la nueva arquitectura y base de datos es fácil de integrar a la plataforma, permitiendo su correcto funcionamiento.

- Los métodos científicos empleados permitieron validar el módulo de procesamiento de contenidos aplicado a la Plataforma de Contenidos Unificados para Búsqueda Avanzada.

RECOMENDACIONES

Se recomienda:

- Incorporar herramientas de web semántica como el uso de ontologías en el proceso de recuperación de información con el objetivo de mejorar el motor de búsqueda.

- Hacer uso de la técnica de expansión de consultas para obtener consultas más completas que permitan delimitar mejor el campo sobre el que se debe recuperar la información.

REFERENCIAS BIBLIOGRÁFICAS

1. Adnan, M. H. M., & Husain, W. (2012). Hybrid Approaches Using Decision Tree, Naive Bayes, Means and Euclidean Distances for Childhood Obesity Prediction. International Journal of Software Engineering and Its Applications, 6(3), 99–106. Recuperado de http://www.earticle.net/Article.aspx?sn=208385
2. Aguilar, J., & Mosquera, D. (2015). Middleware Reflexivo para la gestión de Aprendizajes Conectivistas en Ecologías de Conocimientos (eco-conectivismo). Latin American Journal of Computing Faculty of Systems Engineering Escuela Politécnica Nacional Quito-Ecuador, 2(2). ISNN: 1390-9134.
3. Aliwy, A. H., & Ameer, E. H. A. (2017). Comparative Study of Five Text Classification Algorithms with their Improvements. International Journal of Applied Engineering Research, 12(14), 4309-4319. Recuperado de https://pdfs.semanticscholar.org/893e/9821c86d6338491f56f92dee6907213b76ac.pdf
4. Anitha, V., & Isakki, P. (2016). A survey on predicting user behavior based on web server log files in a web usage mining. In Computing Technologies and Intelligent Data Engineering (ICCTIDE), International Conference on (pp. 1-4). IEEE. Recuperado de https://ieeexplore.ieee.org/abstract/document/7725340
5. Apache Lucene Core. (2018). Recuperado 5 marzo, 2018, de https://lucene.apache.org/core/.
6. Artigas Fuentes, F., Gil García, R., Badía Contelles, J.M., Pons Porrata, A. (2008). Cálculo de la vecindad mediante grafos en minería de textos. *Universidad, Ciencia y tecnología*, 12(48), 163-170.
7. Baeza-Yates, R., & Ribeiro-Neto, B. (1999). Modern information retrieval. Recuperado de www.dcc.ufmg.br/irbook/print/chap10.ps.gz
8. Balasubramanian, T., & Uthangarai, K. D. (2015). Web mining using research issues of data mining techniques. *Asian Journal of Science and Applied Technology* (AJSAT), 3(1), 2–3. Recuperado de www.goniv.com/pdf/ajsatvol3no12015-17.pdf
9. Baquerizo, R. P., Leyva, P. R., Febles, J. P., Viltres, H., & Sala, V. E. S. (2017). Algorithm for calculating relevance of documents in information retrieval systems. *International Research Journal of Engineering and Technology*, 4(3), 1-5. Recuperado de https://www.irjet.net/archives/V4/i3/IRJET-V4I350.pdf
10. Barriga Mariño, J. C. (2017). Desarrollo y aplicación de una herramienta de extracción y almacenamiento de datos de Twitter a un contexto social de violencia política. Recuperado 8 febrero, 2018, de https://repository.ucatolica.edu.co/handle/10983/14597
11. Bidoki, A. M. Z., & Yazdani, N. (2008). DistanceRank: An intelligent ranking algorithm for web pages. *Information Processing & Management*, 44(2), 877-892. Recuperado de http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.494.2614&rep=rep1&type=pdf

12. Blanco, L. C., Casas, L., González, O. L. P., & Mota, Y. C. (2015). Redes neuronales artificiales en la producción de tecnología. *Revista Academia y Virtualidad*, 8(1), 12-20. Recuperado de https://dialnet.unirioja.es/servlet/articulo?codigo=5104742
13. Blázquez Ochando, M. (2015). Técnicas avanzadas de recuperación de información: Procesos, técnicas y métodos. Recuperado 16 marzo, 2018, de https://eprints.ucm.es/31307/
14. Boone, H. N. y Boone, D. A. (2012). Analyzing likert data. *Journal of extension*, 50(2), 1 -5. Recuperado de https://www.researchgate.net/profile/Mahdi_Safarpour/post/what_is_a_logistic_regression_analysis/attachment/59d622fb79197b8077981513/AS:304626539139073@1449640034657/download/Likert+Scale+vs+Likert+Item+%28Good+Source%29.pdf
15. Briceño Segovia, F. S. (2011). Clasificación automática de textos basado en ranking. Recuperado 29 febrero, 2018, de opac.pucv.cl/pucv_txt/txt-0000/UCF0311_01.pdf
16. Buenaño Valencia, E. H., & Vaca Albán, R. V. (2017). El uso de big data y su incidencia en la calidad de los Servicios académicos de la Universidad Técnica de Ambato. Recuperado 5 marzo, 2018, de http://repositorio.uta.edu.ec/handle/123456789/25796
17. Buyse, K. (2018). Qué tipo de corpus para qué tipo de texto: de la teoría a la práctica. Recuperado 19 febrero, 2018, de https://limo.libis.be/primo-explore/fulldisplay?docid=LIRIAS1899793&context=L&vid=Lirias&search_scope=Lirias&tab=default_tab&lang=en_US&fromSitemap=1
18. Carreño López, A. (2017). Detección de sucesos raros con machine learning. Recuperado 7 marzo, 2018, de http://oa.upm.es/47931/
19. Chimborazo Chacha, J. G., & Taco Quispe, L. A. (2013). Desarrollo de sistema de control biométrico de docentes del Departamento de Eléctrica y Electrónica utilizando Frameworks de PHP. Recuperado 7 marzo, 2018, de http://repositorio.espe.edu.ec/handle/21000/6768
20. Claassen, M., & Grill, P. (2017). Aprendizaje semisupervisado de rasgos de temporalidad en el léxico del español. Recuperado 16 abril, 2018, de https://www.fing.edu.uy/inco/grupos/pln/prygrado/InformeTemporalidadEmbeddings.pdf
21. Comeche, J. A. M. (2006). Los modelos clásicos de Recuperación de información y su vigencia. Memoria del Tercer Seminario Hispano-Mexicano de investigación en bibliotecología y documentación, 187.
22. Curiel Lorenzo, S., & Pantoja Trincado, A. (2015). Estudio webmétrico de la revista electrónica Avanzada Científica. *Revista de Arquitectura e Ingeniería*, 9(1), 1–3. Recuperado de http://www.redalyc.org/html/1939/193948443003/
23. De Gyves Camacho, F. M. (2009). Web Mining: Fundamentos básicos.
24. De la Puente, M. (2010). Gestión del conocimiento y minería de datos. Recuperado 21 enero, 2018, de http://eprints.rclis.org/14884/
25. Deshmukh, V., & Barve, S. S. (2016). A Technique for Web Page Ranking by Applying Reinforcement Learning. *International Journal of Computer Applications*, 155(7).

Recuperado de https://pdfs.semanticscholar.org/a11e/544d3fee2a42f02e496256fc7d1b05fc992e.pdf

26. Di Nunzio, G. M. (2009). Using scatterplots to understand and improve probabilistic models for text categorization and retrieval. *International Journal of Approximate Reasoning*,2–8. Recuperado de https://core.ac.uk/download/pdf/82280542.pdf
27. Dulzaides, I. ME; Molina G. AM. 2004. Análisis documental y de información: dos componentes de un mismo proceso. ACIMED, 2(12), 4.
28. Epifanio Tula, L. G., & Medeot, M. D. (2007). Sistema de Recuperación de Información Sistema de Recuperación de Información Motor de Búsqueda: Innuendo. Recuperado 9 febrero, 2018, de http://www.jidis.frc.utn.edu.ar/viewabstract.asp?id=17
29. Fernández, E. (2004). Análisis de clasificadores bayesianos. Trabajo Final de Especialidad en Ingeniería de Sistemas Expertos. Escuela de Postgrado. Instituto Tecnológico de Buenos Aires. Recuperado 15 marzo, 2018, de materias.fi.uba.ar/7550/clasificadores-bayesianos.pdf
30. Figuerola, C. G., Alonso Berrocal, J. L., Zazo Rodríguez, A. F., & Rodríguez, E. (2004). Algunas Técnicas de Clasificación Automática de Documentos. Recuperado 9 marzo, 2018, de https://core.ac.uk/download/pdf/153334293.pdf
31. Fouad, K. M., Khalifa, A. R., Nagdy, N. M., & Harb, H. M. (2012). Web- Web - based Semantic and Personalized Information Retrieval. *International Journal of Computer Science Issues*, 9(3), 266–276. Recuperado de citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.401.7813&rep=rep1&type=pdf
32. Frank, F. (2017). Activity of Core and Peripheral Developers: A Case Study on Mozilla Firefox. Recuperado 12 marzo, 2018, de https://www.infosun.fim.uni-passau.de/spl/theses/FerdinandFrankBA.pdf
33. Gamma, E., Helm, R., Johonson, R., & Vlisside, J. (1995). Design Patterns Elements of Reusable Object-Oriented Software. Recuperado 15 abril, 2018, de https://sophia.javeriana.edu.co/~cbustaca/docencia/DSBP-2018-01/recursos/Erich%20Gamma,%20Richard%20Helm,%20Ralph%20Johnson,%20John%20M.%20Vlissides-Design%20Patterns_%20Elements%20of%20Reusable%20Object-Oriented%20Software%20%20-Addison-Wesley%20Professional%20(1994).pdf
34. García-Peñalvo, F. J., Moreno García, M. N., & García-Holgado, A. (2018). UML. Unified Modeling Language. Recuperado 12 mayo, 2018, de https://repositorio.grial.eu/handle/grial/1147
35. Garg, Y., & Jain, M. V. (2015). A brief survey of various ranking algorithms for web page retrieval in web structure mining. *International Journal of Engineering Trends and Technology*, 21(3), 168-172. Recuperado de www.ijettjournal.org/2015/volume-21/number-3/IJETT-V21P230.pdf
36. Gawas, D., Ramakrishnan, J., Joshi, D., & Khochare, N. (2015). Centralized E-Book Storage using Cloud. *International Journal of Current Engineering and Technology*, 5(2), 840-842. Recuperado de inpressco.com/wp-content/uploads/2015/03/Paper45840-842.pdf

37. Giraldo Huertas, J. J. (2006). Manual para los seminarios de investigación en psicología: profundización conceptual y textual. Recuperado de https://books.google.es/books?hl=es&lr=&id=9QxB0XqtFuUC&oi=fnd&pg=PA13&dq=Manual+para+los+seminarios+de+investigaci%C3%B3n+en+psicolog%C3%ADa:+profundizaci%C3%B3n+conceptual+y+textual&ots=9a5Oh2D62N&sig=kOjE_2ujUEoAZU1fz77uh1AC2PY#v=onepage&q=Manual%20para%20los%20seminarios%20de%20investigaci%C3%B3n%20en%20psicolog%C3%ADa%3A%20profundizaci%C3%B3n%20conceptual%20y%20textual&f=false.
38. Gupta, R. (2014). Journey from Data Mining to Web Mining to Big Data. Recuperado 11 marzo, 2018, de https://arxiv.org/abs/1404.4140
39. Gálvez, C. (2008). Minería de textos: La nueva generación de análisis de literatura científica en biología molecular y genómica. *Revista Electrónica de Biblioteconomía y Ciencias de la Información*, 13(25), 3–4. Recuperado de https://periodicos.ufsc.br/index.php/eb/article/download/1518-2924.2008v13n25p1/1251
40. Gök, A., Waterworth, A., & Shapira, P. (2015). Use of web mining in studying innovation. Scientometrics, 102(1). Recuperado 22 de febrero, 2018, de https://link.springer.com/article/10.1007/s11192-014-1434-0
41. Hallo, M. (2014). Bases de datos NoSQL. Recuperado 13 febrero, 2018, de https://www.researchgate.net/publication/304216502_Bases_de_datos_NoSQL
42. Hernández Orallo, J., Ferri Ramirez, C. E. S. A. R., & Ramirez Quintana, M. J. (2004). Introducción a la Minería de Datos. Pearson Prentice Hall.
43. Herrecalde, M. L., Ingaramo, D. A., Rosas, M. V., & Asencio, A. (2009). Tópicos avanzados en categorización de textos. Recuperado 7 marzo, 2018, de http://sedici.unlp.edu.ar/handle/10915/19673
44. Hu, X., Matsuda, S., Hori, C., & Kashioka, H. (2013). Collecting Colloquial and Spontaneous-like Sentences from Web Resources for Constructing Chinese Language Models of Speech Recognition. *Journal of information processing*, 21(2), 168-175. Recuperado de https://www.jstage.jst.go.jp/article/ipsjjip/21/2/21_168/_pdf/-char/ja
45. Hussein, A. (2015). The use of triangulation in social sciences research: Can qualitative and quantitative methods be combined?. *Journal of comparative Social Work*, 4(1). Recuperado de http://www.bnemid.byethost14.com/NURSING%20RESEARCH%20METHODOLOGY%205.pdf
46. Internet Live Stats.Total number of Websites. (2018). Recuperado 25 marzo, 2018, de http://www.internetlivestats.com/watch/websites/
47. Jaramillo Valbuena, S., & Londoño, J. M. (2014). Document search supported on an ontologial indexing system created with mapreduce. *Ciencia e Ingeniería Neogranadina*, 24(2), 57-75. Recuperado de www.scielo.org.co/pdf/cein/v24n2/v24n2a04.pdf
48. Justicia de la Torre, M. C. (2017). Nuevas técnicas de minería de textos: Aplicaciones. Recuperado 15 enero, 2018, de http://digibug.ugr.es/handle/10481/46975

49. Jácome Paneluisa, H., & Meneses Becerra, E. R. (2017). Análisis de tendencias y hallazgos de patrones de comportamiento de las remuneraciones para el personal que labora en el Ejército Ecuatoriano utilizando algoritmos y técnicas de minería de datos aplicado a la industria de Defensa Nacional. Recuperado 12 febrero, 2018, de http://repositorio.espe.edu.ec/handle/21000/13003
50. Kadam, K. S., & Chandure, O. V. (2015). A Review paper on Student Information Supervision System. *International Journal of Research In Science & Engineering*, 1(1), 6–7. Recuperado de http://www.ijrise.org/asset/archive/15SANKALP11.pdf
51. Katkar, M. (2015). Performance Analysis for NoSQL and SQL. *International Journal of Innovative and Emerging Research in Engineering*, 2(3), 12–13. Recuperado de https://pdfs.semanticscholar.org/97e6/92548e55dfc9c40de114d98d177623013f0a.pdf
52. Kaur, A., Maini, R., & Singh Ahuja, A. (2017). Analysis of Web Usage Mining techniques to predict the user behavior from Web Server Log Files. *International Journal of Advanced Research in Computer Science*, 8(5), 1–2. Recuperado de https://search.proquest.com/openview/5be9b2f42d0417ec1bb2cd98a21b60d7/1?pq-origsite=gscholar&cbl=1606379
53. Knime. (2017). Recuperado de http://kanime.org/.
54. Krishna Murthy, A. (2015). XML URL Classification based on their semantic structure orientation for Web Mining Applications. Recuperado 16 abril, 2018, de https://www.sciencedirect.com/science/article/pii/S1877050915000691/pdf?md5=bedd68ea844deb2d767fd38a037a73af&pid=1-s2.0-S1877050915000691-main.pdf
55. Kumar, G., Duhan, N., & Sharma, A. K. (2011). Page ranking based on number of visits of links of Web page. In Computer and Communication Technology (ICCCT), 2011 2nd International Conference on (pp. 11-14). IEEE.
56. Ladekar, A., Raikar, D., & Pawar, P. (2014). Web Log Based Analysis of User's Browsing Behavior. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 3(11), 1–2. Recuperado de ijarcet.org/wp-content/uploads/IJARCET-VOL-3-ISSUE-11-3895-3899.pdf
57. Larman, C. (2003). UML y Patrones. Una introducción al análisis y diseño orientado a objetos y al proceso unificado (2ª ed.). Recuperado de http://www.fmonje.com/UTN/ADES%20-%20208/UML%20y%20Patrones%20%202da%20Edicion.pdf
58. Lazcorreta Puigmartí, E. (2018). Análisis de catálogos robustos desde la perspectiva de la minería de reglas de asociación. Recuperado 27 mayo, 2018, de dspace.umh.es/bitstream/11000/4487/1/TD%20Lazcorreta%20Puigmart%C3%AD%2C%20Enrique.pdf.
59. Leela Mary, X., & Silambarasan, G. (2017). Web Content Mining: Tool, Technique & Concepts. *International Journal of Engineering Science*, 7(5), 2–4. Recuperado de ijesc.org/upload/fdd53c087b0e38051e61003b486418f2.Web%20Content%20Mining%20Tool%20Technique%20Concepts.pdf

60. Levy, O., & Goldberg, Y. (2014). Linguistic regularities in sparse and explicit word representations. In Proceedings of the eighteenth conference on computational natural language learning (pp. 171-180). Recuperado de www.aclweb.org/anthology/W14-1618
61. Leyva, P. R., Mesa, Y. D., Sala, H. V., Sentí, V. E., & Rodríguez, J. P. F. (2018). Modelo computacional para el desarrollo de sistemas de recuperación de información. *Revista Cubana de Ciencias Informáticas*, 12(1), 173-188. Recuperado de http://scielo.sld.cu/pdf/rcci/v12n1/rcci13118.pdf
62. Lundy, D. (2015). A Teaching Tool for the IEC 61850 Substation Configuration Language. Moodle Integration for Energy Technology ICT. Recuperado 20 marzo, 2018, de http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.829.3959&rep=rep1&type=pdf
63. Makvana, K., Shah, P., & Shah, P. (2014, September). A novel approach to personalize web search through user profiling and query reformulation. In Data Mining and Intelligent Computing (ICDMIC), 2014 International Conference on (pp. 1-10). IEEE.
64. Martínez Albuerne, J. L. (2008). Diseño del catálogo de metadatos para la automatización del proceso de carga de un data webhouse. Recuperado 4 abril, 2018, de http://oa.upm.es/1024/
65. Martínez Méndez, J. F. (2004). Recuperación de información: Modelos, sistemas y evaluación. Recuperado 15 enero, 2018, de http://libros.metabiblioteca.org/handle/001/227
66. Mele, I. (2013, February). Web usage mining for enhancing search-result delivery and helping users to find interesting web content. In Proceedings of the sixth ACM international conference on Web search and data mining (pp. 765-770). ACM.
67. Meng, L., Huang, R., & Gu, J. (2013). A new model for measuring similarity of web queries and its application in query expansion. Int J Grid Distrib Comput, 6(4), 51-62. Recuperado 01 abril, 2018, de https://pdfs.semanticscholar.org/3391/5809f0c0331ae70ece27851966b70e844766.pdf
68. Mittal, N., Nayak, R., Govil, M. C., & Jain, K. C. (2010). A hybrid approach of personalized web information retrieval. In Web Intelligence and Intelligent Agent Technology (WI-IAT), (Vol. 1, pp. 308-313). IEEE.
69. Monsalve Toledo, R. (2015). Estudio de gestión de sistemas basados en grafos. Recuperado 15 marzo, 2018, de https://e-archivo.uc3m.es/bitstream/handle/10016/23656/TFG_Roberto_Monsalve_Toledo_2015.pdf
70. Moore, C. (2002). Diving into data. InfoWorld, 25.
71. Muppidi, S., & Koraganji, V. N. (2016). Survey of contemporary ranking algorithms. *International Journal of Applied Engineering Research*, 11(1), 322-325. Recuperado de https://www.ripublication.com/ijaer16/ijaerv11n1_48.pdf
72. Méndez, F. J. M., & Muñoz, J. V. R. (2004). Reflexiones sobre la evaluación de los sistemas de recuperación de información: necesidad, utilidad y viabilidad. In Anales de documentación (Vol. 7, pp. 153-170).

73. Neelima, G., & Rhoda, S. (2016). Predicting user behavior through sessions using the web log mining. In Advances in Human Machine Interaction (HMI), 2016 International Conference on (pp. 1-5). IEEE. Recuperado de https://ieeexplore.ieee.org/abstract/document/7449167
74. Neptalí Chávez Quispe, L. M. (2014). Aplicación de bases de datos no relacionales NOSQL para la mejora del acceso a la información en el proceso de segmentación de clientes en el centro de actualización profesional para ingenierías CAPI. Recuperado 30 enero, 2018, de http://repositorio.unc.edu.pe/handle/UNC/530
75. Neyra Romero, L. A. (2016). Categorización automática de respuestas aplicando algoritmos de clasificación supervisada al análisis de las contestaciones de estudiantes a una serie de preguntas tipo test. Recuperado 27 marzo, 2018, de https://addi.ehu.es/handle/10810/19240
76. Nikolaevna Ledeneva, Y., García Hernández, R. A., Sidorov, G., & Vargas Flores, S. (2016). Comparación de medidas de similitud para desambiguación del sentido de las palabras utilizando rankeo de grafos. Recuperado 28 mayo, 2018, de ri.uaemex.mx/oca/view/20.500.11799/49398/1/Tesis%20Selene.pdf
77. Noya García, E. (2015). Optimización de un sistema de indexación y búsqueda de palabras clave en grandes colecciones de imágenes de texto manuscrito. Recuperado 8 febrero, 2018, de https://riunet.upv.es/handle/10251/55625
78. Nuñez González, A., Sánchez Díaz, A., & Barcenas Mompeller, Y. (2018). La indexación semántica latente como alternativa para recuperar información relevante. Recuperado 26 julio, 2018, de http://www.informaticahabana.cu/sites/default/files/ponencias2018/MUL36.pdf
79. Oliva Arenas, S. N. (2017). Combinación de métricas y rasgos léxico-semánticos para el análisis de similitud textual entre dos frases. Recuperado 3 marzo, 2018, de http://repositoriodigital.ucsc.cl/handle/25022009/1212
80. Orellana Cordero, M., & Arias Zhañay, M. B. (2016). Minería de texto en medios sociales. Caso de estudio del proyecto Tranvía de Cuenca. Recuperado 6 febrero, 2018, de http://dspace.uazuay.edu.ec/handle/datos/5454
81. Ortega Maldonado, C., Rodríguez Leyva, P., Febles, J. P., Viltres Sala, H., & Delgado Mesa, Y. (2017). Computational model for the processing of documents and support to the decision making in systems of information retrieval. *International Research Journal of Engineering and Technology (IRJET)*, 14, 4–6. Recuperado de www.irjet.net
82. Palomino, N. L. S., Concha, U. R., Beltrán, N. A. O., Pacheco, O. B., Chalco, J. J. E., & Huerta, H. V. (2004). Estudio y evaluación de los sistemas de recuperación de información. *Revista de investigación de Sistemas e Informática*, 1(1), 49-58. Recuperado de https://www.researchgate.net/profile/Oscar_Benito/publication/237479135_ESTUDIO_Y_EVALUACION_DE_LOS_SISTEMAS_DE_RECUPERACION_DE_INFORMACION/links/57b6c06708aede8a665cf3ba/ESTUDIO-Y-EVALUACION-DE-LOS-SISTEMAS-DE-RECUPERACION-DE-INFORMACION.pdf

83. Pandya, R. (2015). Web Usage Mining with Personalization on Social Web. *International Journal of Engineering Trends and Technology (IJETT)*, 29(6), 1–3. Recuperado de www.ijettjournal.org/2015/volume-29/number-6/IJETT-V29P260.pdf
84. Paz Arias, H. P., & Jiménez Ochoa, D. Y. (2016). Desarrollo de un sistema inteligente para la clasificación de documentos ya digitalizados aplicando redes neuronales supervisadas. Recuperado 17 marzo, 2018, de http://dspace.unl.edu.ec/handle/123456789/11395
85. Pratap Singh, A., & Jain, R. C. (2014). A Survey on Different Phases of Web Usage Mining for Anomaly User Behavior Investigation. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, 3(3), 1–3. Recuperado de ijettcs.org/Volume3Issue3/IJETTCS-2014-06-03-066.pdf
86. Pérez Abelleira, M. A., & Cardoso, C. A. (2010). Minería de texto para la categorización automática de documentos. Recuperado 7 marzo, 2018, de www.ucasal.edu.ar/htm/ingenieria/cuadernos/archivos/5-p11-alicia-articulo-cuadernos-formateado.pdf
87. Quiñones Matesanz, E. (2015). Recuperación cruzada de información pública clínico-genómica a partir de consultas sobre repositorios clínicos privados. Recuperado 17 enero, 2018, de http://oa.upm.es/38319/
88. Ramírez Pérez, J. F. (2016). Modelo para la selección de equipos de trabajo quirúrgico en sistemas de información en salud aplicando técnicas de inteligencia organizacional. (Tesis doctoral). Universidad de las Ciencias Informáticas, La Habana, Cuba.
89. RapidMiner: Data Science Platform. (2016). Recuperado de https://rapidminer.comhttps://rapidminer.com.
90. Rish, I. (2001). An empirical study of the naive Bayes classifier. Recuperado 8 febrero, 2018, de https://www.cc.gatech.edu/~isbell/reading/papers/Rish.pdf
91. Rodrigo, G. S., Alonso, J. A. L., & Múgica, P. L. (2006). Aprendizaje discriminativo de clasificadores Bayesianos. Inteligencia artificial: *Revista Iberoamericana de Inteligencia Artificial*, 10(29), 39-47. Recuperado de http://www.redalyc.org/html/925/92502905/
92. Rodríguez Rueda, E., & Hidalgo Delgado, Y. (2012). Los spiders y su función en los motores de búsqueda. Recuperado 18 febrero, 2018, de https://www.researchgate.net/publication/233425516_LOS_SPIDERS_Y_SU_FUNCION_EN_LOS_MOTORES_DE_BUSQUEDA
93. Rodríguez, M. F. (2016). Modelado de perfiles de usuario para la recomendación de contenido en Twitter. Recuperado 27 enero, 2018, de http://ridaa.unicen.edu.ar/xmlui/handle/123456789/557
94. Rodríguez, P. A., Duque, N. D., & Ovalle, D. A. (2016). Método Híbrido de Recomendación Adaptativa de Objetos de Aprendizaje basado en Perfiles de Usuario. Recuperado 19 marzo, 2018, de https://scielo.conicyt.cl/pdf/formuniv/v9n4/art10.pdf
95. Romaní, J. C. C. (2011). El concepto de tecnologías de la información. Benchmarking sobre las definiciones de las TIC en la sociedad del conocimiento. *Zer-Revista de Estudios de*

Comunicación, 14(27). Recuperado de https://www.ehu.eus/ojs/index.php/Zer/article/download/2636/2184

96. Ropero Rodríguez, J. (2010). Método general de extracción de información basado en el uso de lógica borrosa. Aplicación en portales web. Recuperado 9 marzo, 2018, de https://idus.us.es/xmlui/bitstream/handle/11441/15542/2010ropermetod.pdf?sequence=1
97. Salazar García, A. (2015). Contribución de la inteligencia competitiva en el proceso de I+D+i del sector farmacéutico: el caso de España. Recuperado 29 enero, 2018, de https://upcommons.upc.edu/handle/2117/95813
98. Sebastiani, F. (2005). Text Categorization Text Mining and Its Applications. Text Mining and its Applications, A, 109-129.
99. Sharma, D. K., & Sharma, A. K. (2010). A comparative analysis of Web page ranking algorithms. *International Journal on Computer Science and Engineering*, 2(8), 2670-2676. Recuperado de citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.301.8489&rep=rep1&type=pdf
100. Shoaib, M., & Maurya, A. K. (2014). Comparative Study of Different Web Mining Algorithms to Discover Knowledge on the Web. Recuperado 24 marzo, 2018, de https://s3.amazonaws.com/academia.edu.documents/36727901/9789351072638_ERCICA_2014_VolIII_102.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1541114233&Signature=mHhDmJeBjlfpXh8VjnaowqV8bDQ%3D&response-content-disposition=inline%3B%20filename%3DComparative_Study_of_Different_Web_Minin.pdf
101. Stratebi. (2010). Comparativa de Algoritmos de Herramientas de Data mining. Recuperado 27 mayo, 2018, de http://www.dataprix.com/files/Comparativa-Algoritmos-Herramientas_Data_Mining.pdf
102. Suca, C., Córdova, A., Condori, A., Cayra, J., & Sulla, J. (2016). Comparación de algoritmos de clasificación para la predicción de casos de obesidad infantil. Recuperado 10 marzo, 2018, de https://www.researchgate.net/profile/Abel_Condori_Castro/publication/301567339_COMPARACION_DE_ALGORITMOS_DE_CLASIFICACION_PARA_LA_PREDICCION_DE_CASOS_DE_OBESIDAD_INFANTIL/links/571a985c08aee3ddc568f97d.pdf
103. Suthar, P., & Oza, B. (2015). A survey of web usage mining techniques. *Int. J. Comput. Sci. Inf. Technol*. (IJCSIT), 6(6). Recuperado de citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.734.9741&rep=rep1&type=pdf
104. Tian, X., Du, Y., Song, W., Liu, W., & Xie, Y. (2014). Improvements of HITS Algorithm Based on Triadic Closure. *Journal of information & computational science*, 11(6), 1861-1868. Recuperado de http://manu35.magtech.com.cn/Jwk_ics/CN/article/downloadArticleFile.do?attachType=PDF&id=2271
105. Tolosa, G. H., & Bordignon, F. R. A. (2008). Introducción a la Recuperación de Información. Recuperado 17 enero, 2018, de http://eprints.rclis.org/12243/

106. Torres López, C., & Arco García, L. (2016). Representación textual en espacios vectoriales semánticos. *Revista Cubana de Ciencias Informáticas*, 10(2), 148-180. Recuperado de https://scielo.sld.cu/pdf/rcci/v10n2/rcci11216.pdf

107. Trotman A., Learning to rank. Information Retrievel, Vol.8, pp. 359-381, 2005. Recuperado de https://link.springer.com/article/10.1007/s10791-005-6991-7

108. Téllez Valero, A. (2005). Extracción de Información con Algoritmos de Clasificación. Recuperado 4 marzo, 2018, de https://ccc.inaoep.mx/~villasen/tesis/TesisMaestria-AlbertoTellez.pdf

109. Urbano, J., Morato, J., Marrero, M., & Sánchez-Cuadrado, S. (2010). Recuperación y Acceso a la Información. OpenCourseWare, University Carlos III of Madrid. Recuperado 14 abril, 2018, de http://ocw.uc3m.es/ingenieria-informatica/recuperacion-acceso-informacion/material-de-clase-1/MC-F-006.2.pdf

110. Valencia, M. (2007). Categorización de consultas enviadas a un motor de búsqueda web (Doctoral dissertation, Tesis Doctoral. Valencia, España: Universidad Politécnica de Madrid).

111. Valencia, M. M. A. (2013). La triangulación metodológica: sus principios, alcances y limitaciones. *Investigación y educación en enfermería*, 18(1). Recuperado de www.redalyc.org/pdf/1052/105218294001.pdf

112. Valencia, M., Eibe, S., & Menasalvas, E. (2009). Proceso de Categorización de Consultas Basado en Visibilidad en un Dispositivo Móvil. *Revista Colombiana de Computación-RCC*, 10(2).

113. Vargas, A. M. (2016). [Reseña de libro] Cibermetría. Midiendo el espacio red. Comunicación y Medios, (34), 118-119. Recuperado de https://revistas.uchile.cl/index.php/RCM/article/download/42556/46815

114. Vega Vilca, J. C., & Torres Núñez, D. A. (2015). Una metodología para encontrar el mejor clasificador en decisión empresarial. *Revista de Ciencias Económicas*, 33(1), 63–73. Recuperado de https://revistas.ucr.ac.cr/index.php/economicas/article/view/19971/21613

115. Vergara García, P. M. (2014). Selección y evaluación de algoritmos para clasificación de documentos. Recuperado 16 marzo, 2018, de http://digibuo.uniovi.es/dspace/handle/10651/27539

116. Villena Román, J. (2004). Sistemas de Recuperación de información. Valladolid: Departamento Ingeniería Sistemas Telemáticos, Universidad. Citado por: Martínez Mendez, F. J. Recuperación de Información: Modelos, sistemas y evaluación. Murcia: Kiosko JMC 2004, p. 8

117. Vogel, D., Bickel, S., Haider, P., Schimpfky, R., Siemen, P., Bridges, S., & Scheffer, T. (2005). Classifying search engine queries using the web as background knowledge. ACM SIGKDD Explorations Newsletter, 7(2), 117-122.

118. Voit, A., Stankus, A., Magomedov, S., & Ivanova, I. (2017). Big Data Processing for Full-Text Search and Visualization with Elasticsearch. *International journal of advanced*

computer science and applications, 8 (12), 76-83. Recuperado de https://pdfs.semanticscholar.org/6ed3/8eb7d147eb665168ae0bd748d35d2583e583.pdf

119. Wang, M., Li, Q., Lin, Y., & Zhou, B. (2017). A personalized result merging method for metasearch engine. In Proceedings of the 6th International Conference on Software and Computer Applications (pp. 203-207).ACM.

120. Weka: Weka. (2016). Recuperado de http://se-weka.blogspot.comhttp://se-weka.blogspot.com/.

121. Yao, M., Pi, D., & Cong, X. (2012). Chinese text clustering algorithm based k-means. *Physics Procedia*, 33, 301-307. Recuperado de https://core.ac.uk/download/pdf/82824110.pdf

122. Yaya, W., & Xueyun, J. (2012). Chinese Text Classification Based on VC-Dimension and BP Neural Network. *In Advances in Control and Communication* (pp. 497-502). Recuperado de https://link.springer.com/content/pdf/10.1007%2F978-3-642-26007-0.pdf

123. Yu, X., & Xie, J. Q. (2017). U.S. Patent Application No. 15/325,060.

124. Zuñiga Tinizaray, A. V., & Sánchez Carrión, M. F. (2017). Sistema web para la gestión de la producción académica-científica de docentes y estudiantes de la universidad nacional de Loja. Recuperado 7 marzo, 2018, de http://dspace.unl.edu.ec/handle/123456789/19412

GLOSARIO

1. Stopwords: Palabras no relevantes o superfluas que no aportan significado para la categorización (artículos, preposiciones, conjunciones, entre otras)
2. Stemmer: Es el proceso de reducir las palabras inflexas (o algunas veces derivadas) a su forma de palabra, base o raíz.
3. SQL (Structured Query Language): Lenguaje de consulta estructurada.
4. Ranking: Acto de clasificar algo.
5. Re-ranking: El acto de clasificar algo de nuevo o de manera diferente.
6. Rankear: En informática se refiere a la acción de ordenar elementos.
7. Relevance feedback: Retroalimentación de relevancia.
8. Matching: Correspondencia.
9. Web usage mining: Minería de uso web.
10. Scripts: Programa usualmente simple, que por lo regular se almacena en un archivo de texto plano.
11. Applets: *C*omponente de una aplicación que se ejecuta en el contexto de otro programa.
12. World wide web: En informática, la World Wide Web (WWW) o red informática mundial es un sistema de distribución de documentos de hipertexto o hipermedia interconectados y accesibles vía Internet.
13. Proxy: Servidor —programa o dispositivo—, que hace de intermediario en las peticiones de recursos que realiza un cliente a otro servidor.
14. Cookies: Pequeña información enviada por un sitio web y almacenada en el navegador del usuario, de manera que el sitio web puede consultar la actividad previa del navegador.
15. Data Mining: Minería de datos.
16. Text Mining: Minería de textos.
17. E-mail: Servicio de correo electrónico.
18. Explorer: Componente del sistema Windows que presenta la interfaz en el monitor.
19. Clustering: Algoritmo de agrupamiento que consiste en un procedimiento de agrupación de una serie de vectores de acuerdo con un criterio.
20. Corpus: Conjunto amplio y estructurado de ejemplos reales de uso de la lengua. Estos ejemplos pueden ser textos o muestras orales.
21. Hubs: Dispositivo que permite centralizar el cableado de una red de computadoras, para luego poder ampliarla.
22. Page: Página.
23. Stemming: Proveniente.
24. Framework: Entorno de trabajo o marco de trabajo. Conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.
25. Scripting: Lenguaje de programación que admite scripts, programas escritos para un entorno de tiempo de ejecución especial.

26. Frontend: Parte de un sitio web que interactúa con los usuarios.
27. Backend: Parte del desarrollo web que se encarga de que toda la lógica de una página web funcione.
28. Gigabyte*:* Unidad de almacenamiento de información cuyo símbolo es el GB, equivalente a 109 (1 000 000 000 -mil millones-) de bytes.
29. Servlet: Clase en el lenguaje de programación Java, utilizada para ampliar las capacidades de un servidor.
30. Hardware: Conjunto de elementos físicos o materiales que constituyen una computadora o un sistema informático.
31. Software: Término general para los diversos tipos de programas utilizados para operar computadoras y dispositivos relacionados
32. Web server: Servidor web.

SIGLARIO

1. DocRelev: Documentos relevantes.
2. DocRec: Documentos recuperados.
3. DocRelevRec: Documentos relevantes recuperados.
4. API: Es un conjunto de subrutinas, funciones y procedimientos en la programación orientada a objetos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.