BUSCADOR INTELIGENTE BASADO EN MINERÍA DE DATOS

GERMAÍN BOLAÑOS VIDAL
ELVIS HERLEY PÉREZ HERNÁNDEZ

Director: MSc. CARLOS ALBERTO COBOS LOZADA

UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE SISTEMAS
POPAYÁN, JUNIO DE 2009

AGRADECIMIENTOS

Nuestros agradecimientos a la Universidad del Cauca institución que nos forjó como personas, brindándonos la oportunidad a través del programa de Ingeniería de Sistemas para realizar nuestros estudios de pregrado.

A nuestro director, MSc. Carlos Alberto Cobos Lozada por su colaboración, consejo, apoyo y paciencia sin lo cual no hubiese sido posible la realización de éste proyecto.

A nuestros compañeros, amigos y profesores por el apoyo brindado en los momentos más oportunos.

Especialmente a nuestras familias por su apoyo incondicional.

TABLA DE CONTENIDO

LISTA DE TABLAS

LISTA DE FIGURAS

INTRODUCCIÓN

Internet es una tecnología que se ha extendido por todo el mundo y la información que abarca es cada vez mayor, existe en este mar de información gran cantidad de recursos útiles, así como gran cantidad de información sin relevancia. El gran problema de Internet es que su información crece de una forma caótica e incontrolable, haciéndose muy difícil encontrar recursos valiosos. Debido a este problema se hizo vital el uso de los buscadores Web para acceder a esta información y es en la actualidad la herramienta con la que la mayoría de los usuarios de Internet inician sus tareas[1].

Actualmente se encuentran disponibles una gran cantidad de buscadores, cada uno de ellos aplica diferentes estrategias de recuperación y filtrado de información, algunos más eficientes que otros pero ninguno satisface completamente las necesidades de los usuarios. Por esa razón se presenta una propuesta de un modelo de meta buscador que utiliza varias estrategias de recuperación y filtrado de información tales como Técnicas de Minería de Datos, Taxonomías del Conocimiento, Perfiles de Usuario y Ontologías.

A lo largo de este documento se presenta el proceso seguido para la realización del proyecto y los conceptos teóricos relevantes necesarios para el desarrollo del mismo. A continuación se hace una descripción general del contenido de este documento y la organización del mismo.

CAPÍTULO I – DEFINICIÓN DEL PROBLEMA Y CONTRIBUCIÓN A LA SOLUCIÓN

En este capítulo se presenta una visión global del proyecto, la problemática que originó su desarrollo, la justificación, los objetivos y los principales resultados del mismo.

CAPÍTULO II – MARCO TEÓRICO

Aquí se describen las bases teóricas que enmarcan el proyecto, se tienen en cuenta los conceptos de ontologías, taxonomías del conocimiento, técnicas de minería de datos, además se realiza una descripción detallada de los algoritmos empleados en la implementación de las tareas de minería necesarias para el desarrollo del meta buscador y se realiza un análisis comparativo entre las soluciones existentes a la problemática y la solución planteada en el presente trabajo.

CAPÍTULO III – MODELO DE BÚSQUEDA INTELIGENTE PROPUESTO

En este capítulo se muestra cómo se define el modelo en cuanto a componentes y sus relaciones. Se hace especial énfasis en la forma como se emplearon las ontologías, taxonomías, perfiles de usuario, las técnicas de minería de datos y los elementos que diferencian el modelo propuesto con buscadores y meta buscadores existentes.

CAPÍTULO IV – DESCRIPCIÓN DEL META BUSCADOR

Este capítulo describe el desarrollo de la aplicación web, la forma como se construyeron los diferentes componentes que conforman el meta buscador y la descripción de los problemas y las respectivas soluciones que se plantearon durante el desarrollo del proyecto.

CAPÍTULO V – VALIDACIÓN

Aquí se describe en qué consistieron y cómo se desarrollaron las diferentes pruebas y los resultados de la evaluación de la exactitud de las respuestas obtenidas con el meta buscador propuesto frente a los otros buscadores.

CAPÍTULO VI – CONCLUSIONES, RECOMENDACIONES Y TRABAJO FUTURO

Aquí se describen las conclusiones que se generaron después de la culminación del proyecto y se establecen posibles mejoras o elementos adicionales que se puedan incluir en un trabajo futuro para la continuidad del proyecto.

CAPÍTULO VII – BIBLIOGRAFÍA Y GLOSARIO

Este capítulo contiene la bibliografía y documentación empleada en la realización del proyecto, incluye además el catálogo de palabras con su respectiva definición.

CAPÍTULO I – DEFINICIÓN DEL PROBLEMA Y CONTRIBUCIÓN A LA SOLUCIÓN

1. DESCRIPCIÓN DEL PROYECTO

1.1 DEFINICIÓN DEL PROBLEMA

La búsqueda de información en Internet es en la actualidad un servicio fundamental que cada vez es más utilizado por los individuos, por esta razón los medios de búsqueda tienen un alto impacto en casi cualquier tarea [2] y se han convertido en el principal medio para acceder de forma directa a la información. Actualmente se estima que los usuarios de Internet comienzan a utilizar los servicios de la red en un 88% de las veces con un buscador [1].Pero aunque los buscadores son muy útiles para los usuarios, cada uno de ellos cuenta con una interfaz de usuario diferente, interpretan las consultas de diferente forma, soportan diferentes tipos de funcionalidades de búsqueda avanzada y emplean diferentes tipos de algoritmos de búsqueda.

Desde el punto de vista de los usuarios finales, tratar con una serie de interfaces diferentes agrega mucha confusión y presenta una sobrecarga cognoscitiva [3]. En este sentido, WebFerret ofrece una posible solución, ya que a través de una única interfaz de usuario (fácil de usar, con menús desplegables y comandos de teclas aceleradoras) consulta y trae la información desde varios buscadores, permitiendo además, guardar las búsquedas en un histórico, filtrar los contenidos y el lenguaje [4].

Como adición a la complejidad de las diferentes interfaces que ofrecen los buscadores, otro problema de la búsqueda en Internet es el enorme crecimiento que se evidencia en la Web tanto superficial[1] (Google, visita y clasifica más de 4.285 millones de páginas [5]) como profunda[2], además de la cantidad de altas, bajas, modificaciones de recursos que hace más difícil localizar los documentos de interés. Según un estudio realizado por la empresa Bright Planet, la Web profunda contiene alrededor de 550 veces más contenido que la Web superficial, además muestra que la forma como trabajan algunos motores de búsqueda no permite acceder a dicho contenido [6].

A menudo la consulta de una simple palabra retorna resultados inconsistentes, con referencias de documentos que reúnen los criterios de búsqueda pero no son de interés para el usuario [7][8]. Además los buscadores no manejan un perfil del usuario para poder identificar sus necesidades reales, ni aprovechan la potencialidad de la máquina (cliente) desde donde se realiza la búsqueda.

Se ha intentado dar una solución a estos problemas exhibiendo en los documentos Web una expresividad que no solo permita la indexación y recuperación, sino varias formas de razonamiento lógico automatizado, pero los desarrollos en este tema normalmente se han hecho en diferentes contextos, puntos de vista y suposiciones acerca de la materia de estudio. Se ha tratado de capturar el significado de los datos (semántica) usando

[1] Web formada por documentos estáticos accesibles.
[2] Web constituida por bases de datos cuyos contenidos, no directamente accesibles por los métodos de búsqueda actuales, se hacen visibles mediante páginas generadas dinámicamente.

jerarquías del conocimiento, pero cada dato toma un solo punto de vista del mundo y describe los objetos de interés bajo una sola interpretación haciendo difícil la reutilización [9][10].

Así mismo, existe una gran cantidad de buscadores en el mercado (los más usados se muestran en la Figura 1[11]), pero solo unos pocos intentan dar respuestas a preguntas reales como "¿Quién fue el primer americano en el espacio?" o ¿Cuál es la segunda montaña más alta en el mundo? Todavía hoy la mayoría de servicios de búsqueda avanzada en la Web (por ejemplo Google, AskJeeves) hacen que sea sorprendentemente tedioso localizar las respuestas a tales consultas [12]. Es por esto que resulta de gran interés seguir planteando la siguiente pregunta ¿Qué páginas desea un usuario recuperar realmente cuando teclea algunas palabras claves en un buscador? Hay miles de páginas que contienen estas palabras, pero el usuario está interesado en un subconjunto mucho más pequeño. En este caso, es procedente preguntarse si el problema se resuelve simplemente realimentando la consulta con preguntas al usuario [13].

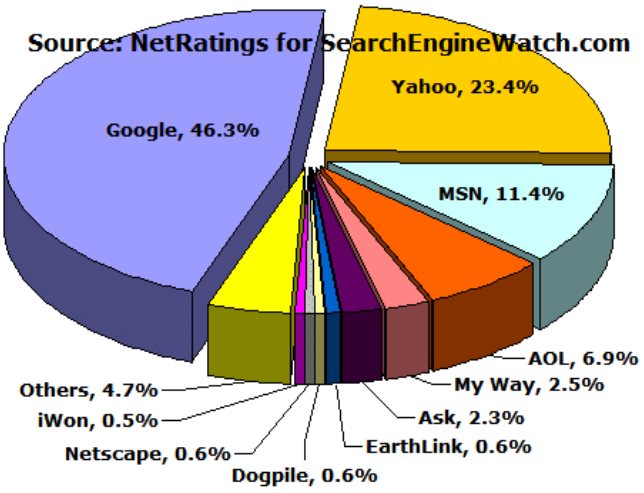


Figura 1 Buscadores más Utilizados en Internet.

Algunos proyectos han buscado dar solución a esta pregunta, por medio de técnicas de Minería de Datos como el agrupamiento de términos "clustering", métodos matemáticos, Minería de Textos, sistemas de representación del conocimiento, clasificación, entre otras, pero aun no ofrecen la solución más apropiada que genere la respuesta que el usuario está esperando.

Teniendo en cuenta los problemas descritos anteriormente, este trabajo presenta la propuesta de un modelo de meta buscador que utiliza una combinación de técnicas de Minería de Datos con ontologías, taxonomías del conocimiento y perfil de usuario, con el propósito de filtrar la información entregada por diferentes buscadores (google, msn,

yahoo) y de esta forma entregar resultados más aproximados a las necesidades más relevantes de los usuarios.

1.2 JUSTIFICACIÓN

El principal interrogante planteado para el desarrollo de este proyecto fue: *"¿Qué páginas desea un usuario recuperar realmente cuando teclea algunas palabras claves en un buscador?",* aunque los buscadores son muy populares [1][2][11] y de gran utilidad cuando se desea recuperar información de la Web, su funcionamiento interno aún presenta fallos en el filtro de la información, presentando resultados que muchas veces nada tienen que ver con la consulta realizada. Este proyecto busca dar una posible solución a través de la inclusión de Minería de Datos (que se ha convertido en una de las herramientas de gran utilidad en la actualidad, como es el caso de su aplicación en diferentes áreas [15][16][17], y para el descubrimiento de información "oculta" en grandes bases de datos) y sus diferentes técnicas que permiten analizar de forma exhaustiva los resultados entregados por diferentes buscadores (google, msn, yahoo). Del mismo modo las taxonomías del conocimiento y las ontologías permiten filtrar la información que necesita el usuario, logrando de este modo respuestas más cercanas a las que desea.

Desde una perspectiva práctica, este proyecto es conveniente ya que buscó disminuir el tiempo gastado por las personas en los procesos de recuperación de información y evitar la lectura y revisión de temas no relacionados con las consultas y que en principio tiene como beneficiarios a los docentes y estudiantes de la Universidad del Cauca.

El proyecto además muestra la viabilidad de aplicar la tecnología de Minería de Datos combinada con ontologías y taxonomías en la recuperación y análisis de la información, tema de gran importancia y relevancia internacional a nivel investigativo, ya que no se encontró ninguna referencia que use la combinación mencionada. Desde esta perspectiva, se obtiene nuevo conocimiento para la comunidad científica internacional, resultado de la combinación apropiada de distintas técnicas de búsqueda de información sobre la Web superficial.

Las herramientas tecnológicas (Microsoft: Microsoft Visual Studio 2005, Microsoft SQL Server 2005, Microsoft Visio 2003 y Microsoft Project 2003) necesarias para la realización de este proyecto fueron seleccionadas con base en la experiencia que el GTI ha obtenido en los últimos cinco (5) años, a las experiencias exitosas en el desarrollo de proyectos relacionados con Minería de Datos en distintos contextos (como por ejemplo: Módulo de Soporte a Toma de Decisiones para Comercio Electrónico B2C[3] e Inteligencia Artificial en la generación de Estrategias de Aprendizaje de un Sistema Tutor Inteligente) y finalmente a la disponibilidad del software, documentación y materiales de aprendizaje que se tiene,

[3] B2C Business To Consumer: Negocio electrónico dirigido al cliente final.

gracias al programa MSDN Academic Alliance[4] con el que la Universidad del Cauca cuenta desde hace más de tres años.

Durante el desarrollo del proyecto se aplicaron varios conceptos aprendidos en el transcurso de la carrera de ingeniería de sistemas, y se ha investigado sobre conceptos nuevos, necesarios para la consecución del producto final. Con lo anterior, los autores han podido demostrar su capacidad para plantear y resolver problemas relacionados con su formación ya como profesionales y como investigadores en formación del alma mater.

2. OBJETIVOS

A continuación se muestran los objetivos del proyecto, conforme fueron aprobados por el Comité de Investigaciones de la Facultad de Ingeniería Electrónica y Telecomunicaciones en el documento de anteproyecto.

2.1 OBJETIVO GENERAL

Modelar y desarrollar un meta buscador que utilice técnicas de minería de datos, taxonomías del conocimiento, perfiles de usuario y ontologías, para manejar las consultas de una manera especializada, logrando así proporcionar respuestas que se aproximen más a los resultados deseados por el usuario en su proceso de recuperación de información.

2.2 OBJETIVOS ESPECÍFICOS

- Elaborar un Modelo para Búsqueda Inteligente en la Web teniendo en cuenta el estudio, la selección y la definición de técnicas de minería de datos, taxonomías del conocimiento, ontologías y perfiles de usuario adecuados para la recuperación de información solicitada por el usuario.
- Desarrollar un meta buscador basado en el modelo propuesto y una arquitectura de aplicación de cliente inteligente, implementándolo con Microsoft Visual Studio 2005 y con el requisito esencial de poder interactuar con las APIs de los siguientes buscadores: Google, Yahoo y MSN.
- Evaluar la exactitud de los resultados entregados por el meta buscador, confrontándolos con los resultados entregados por las APIS de cada buscador en forma independiente a través de pruebas alfa y beta bien automatizadas y documentadas.

3. RESULTADOS OBTENIDOS

- Modelo. El cual presenta los diferentes componentes que integran el meta buscador, la relación de dichos componentes, la arquitectura.

[4] Acuerdo que vincula a Microsoft con entidades educativas universitarias, en la cual se permite tener acceso a software de desarrollo con propósitos académicos.

- Meta Buscador que instancia el modelo: Aplicación para la realización de búsquedas en la web, código fuente e instaladores.
- Prototipo de un editor de Ontologías. Un prototipo de Editor de Ontologías para la edición o creación de ontologías incluido como un módulo del meta buscador
- Artículo: *Búsqueda Inteligente Utilizando Minería de Datos*. Publicado en la Revista de Ciencia y Tecnología Enlace Informático, del Departamento de Sistemas de la Facultad de Ingeniería Electrónica y Telecomunicaciones, Universidad del Cauca. Quinta Edición Diciembre 2006. ISSN: 1692-374X. http://enlaceinformatico.unicauca.edu.co/ y que además fue presentado en el IV Seminario Internacional de Tecnologías en Internet SITI 2006.
- Artículo: *Modelo de búsqueda basada en Minería de Datos.* A presentar en evento internacional denominado International Conference on Knowledge Discovery and Information Retrieval KDIR Madeira Portugal. El plazo de entrega del documento es el 26 de mayo. Sitio Web: http://www.kdir.ic3k.org/cfp.htm.
- Monografía del trabajo de grado. En este documento se describe el proceso seguido en el desarrollo del proyecto, los problemas que se presentaron, las respectivas soluciones, los aportes más sobresalientes, las conclusiones y recomendaciones para desarrollos futuros.

CAPÍTULO II – MARCO TEÓRICO

Para el desarrollo del proyecto fue necesario tener conocimiento de conceptos como la recuperación de información, debido a que el sistema propuesto presenta al usuario información relevante a la consulta elaborada por él. Para ello se debió consultar acerca de los diversos métodos y modelos en la recuperación de información. En el documento se presentará la descripción de los tres modelos clásicos con sus respectivas características. El sistema propuesto trabaja de la mano con las matemáticas, por esta razón se realiza una descripción del modelo vectorial, SVD (Descomposición de Valor Singular) y del LSI (Índice de Semántica Latente). De igual forma se presentará la explicación de algunas herramientas catalogadas como sistemas de recuperación de información, dichas herramientas fueron un referente para el desarrollo del proyecto, debido a que fue necesario profundizar en las características propuestas por dichas herramientas para luego realizar el modelo del sistema propuesto.

Para la mejora en la entrega de resultados a una consulta realizada por el usuario, fue necesario el trabajo con Minería de Datos, por dicha razón se realizará una introducción y explicación de las diferentes técnicas relacionadas con el tema. Por último se realizará la descripción y explicación de los términos taxonomía y ontología, incluidos en el trabajo, de suma importancia para el desarrollo del proyecto, puesto que son los elementos diferenciadores, ya que presentan la clasificación general del conocimiento, permitiéndole al usuario especificar una consulta en un determinado tema.

4. RECUPERACIÓN DE INFORMACIÓN

La recuperación de información (IR), se define como el descubrimiento de material (usualmente son documentos) que satisface una necesidad de información dentro de grandes colecciones de documentos almacenados en servidores locales o en Internet [18].

Aunque en la actualidad la existencia de muchas connotaciones para el término puede causar confusión en su definición, no debe ser confundida con la recuperación de datos. La recuperación de información se preocupa por presentar al usuario la información más relevante a la consulta elaborada por el usuario. Para realizar dicha tarea los sistemas de recuperación de información deben tratar con información no estructurada generalmente presentada en texto (documentos textuales) y semánticamente ambigua e interpretar el contenido de los documentos mediante la extracción sintáctica y semántica. A diferencia con la recuperación de datos que consiste en determinar de una serie de documentos aquellos documentos que contengan palabras claves consultadas por el usuario (en este caso se pueden traer documentos que contengan las palabras claves pero no son relevantes para el usuario), dicho proceso trata con información que tiene una semántica y una estructura bien definida (generalmente observada en bases de datos relacionales) [19].

4.1 MODELOS CLÁSICOS EN LA RECUPERACIÓN DE INFORMACIÓN

Antes de empezar con la descripción de los modelos clásicos (booleano, vectorial y probabilístico), se debe realizar una especificación formal del modelo de recuperación de información.

Definición: Un modelo de recuperación de información es una cuádrupla de la forma; $[D, Q, F, R(q_i, d_j)]$ [20]donde:

- D es la representación del conjunto de documentos
- Q es la representación del conjunto de la información necesitada por el usuario, también llamadas consultas.
- F es el marco de trabajo que modela las representaciones de los documentos, las consultas y sus relaciones.
- $R(q_i, d_j)$ es la función que permite establecer el orden de presentación de los resultados (vínculos a documentos d_j donde $d_j \in D$) con respecto a una consulta $q_i \in Q$

Para comprender mejor la definición, consideremos la construcción de un modelo. Se debe pensar primero en la representación de los documentos y la información que el usuario requiere. Dadas estas representaciones, entonces se concibe el marco de trabajo en el cual ellos pueden ser modelados. Dicho marco de trabajo podría también proveer por intuición la construcción de una función de posicionamiento. Por ejemplo, para el modelo booleano, el marco de trabajo está compuesto de conjuntos de documentos y las operaciones estándar sobre estos conjuntos. Para el modelo vectorial, el marco de trabajo está compuesto de un espacio vectorial y operaciones estándar de algebra lineal sobre vectores. Para el modelo clásico probabilístico, el marco de trabajo está compuesto de conjuntos, operaciones de probabilidad, y el teorema de Bayes[5].

Los modelos clásicos consideran que cada documento sea representado por un conjunto de palabras claves representativas llamadas términos indexados. Un término indexado es simplemente una palabra que conserva una relación semántica con los temas principales del documento.

4.1.1 MODELO BOOLEANO

El modelo de recuperación booleano es uno de los modelos más utilizados para la recuperación de información, basado en teoría de conjuntos y algebra booleana[20]. Dicho modelo funciona agrupando documentos, los cuales están compuestos por conjuntos de términos y preguntas de consulta como expresiones booleanas. La principal característica es la consideración de la relevancia como un carácter puramente binario.

[5] Enunciado por Thomas Bayes, en teoría probabilística proporciona la distribución de probabilidad condicional de un evento A dado otro evento B.

Para el modelo booleano, los documentos se encuentran representados por conjuntos de palabras o términos clave, donde se les asociará un peso binario; uno (1) si el término aparece en el documento por lo menos una sola vez y cero (0) si el término no aparece. Las búsquedas o consultas de los usuarios, son expresiones de palabras claves conectadas a través de operadores lógicos (AND, OR y NOT). El grado de similitud entre la consulta del usuario y un determinado documento también es binario, uno (1) si el documento es relevante para la consulta y cero (0) sino es relevante.

Para comprender el modelo se realizará un ejemplo que ilustrará su funcionamiento. Se supondrá que se desea realizar una consulta $q = t_1 \wedge (t_2 \vee \neg t_3)$, la consulta expresada en palabras quiere decir que debe contener el término t_1 y además el término t_2 o la negación del término t_3. Cada uno de los operadores puede ser representado utilizando un diagrama de Venn (ver Figura 2)

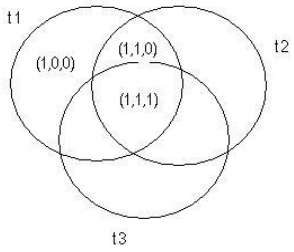


Figura 2 Diagrama de Venn (Tomado de [20])

El modelo booleano representa un documento d_j en forma de tupla (t_1, t_2, t_3), donde por ejemplo la tupla $(1,1,0)$ representa los documentos que contienen los términos t_1 y t_2, además todos los documentos pertenecientes a este conjunto de datos son relevantes para la consulta, en comparación con el conjunto de documentos pertenecientes a la tupla $(0,1,0)$, que no son relevantes para la consulta.

La ventaja del modelo booleano es que es simple de comprender, pero su principal desventaja radica en que el criterio de recuperación no es óptimo, al considerar un documento relevante simplemente por contener los términos que se consultan, además no hay un grado de relevancia específico, ya que un documento es relevante a la consulta o no lo es. Las desventajas hacen que el modelo booleano sea considerado un modelo de recuperación de datos en lugar de un modelo de recuperación de información.

4.1.2 MODELO VECTORIAL

El modelo espacio vector como también es conocido, propone un emparejamiento parcial a diferencia del modelo booleano**,** esto se realiza asignando pesos diferentes de 1 y 0 a los términos claves de las preguntas y de los documentos. Estos pesos se usan para calcular el grado de similitud entre cada documento y la consulta del usuario [20].

El modelo vector o vectorial ha sido usado en operaciones de recuperación de información, filtrado de información, categorización automática de información, etc. En dicho modelo los documentos son representados como vectores en un espacio m-dimensional, donde m es el número de términos en el conjunto de documentos. La descripción formal [21] para un documento puede expresarse como se muestra a continuación:

$$D_i \rightarrow \vec{d} = (T_{i1}, T_{i2}, .., T_{im}) \quad (1)$$

Donde D_i es el documento i-ésimo, de un conjunto N de documentos, con un conjunto de m características consideradas como ocurrencias de palabras o términos T_{ik} en dicho documento. Es necesario tener en cuenta que existen palabras que no significan nada, a estos términos se les conoce como palabras vacías (por ejemplo, artículos, preposiciones, pronombres, entre otros) y se reconocen porque tienen una frecuencia de aparición demasiado alta. La idea general del modelo podría ser definida como la construcción de una matriz de términos y documentos, que representan las columnas y las filas de la matriz respectivamente. De esta forma las filas de la matriz (vectores en términos algebraicos) serían equivalentes a los documentos expresados en función de la frecuencia de aparición de cada término. Para ilustrar la descripción anterior se tomará un ejemplo; un documento podría ser expresado como d1 = (1, 2, 0, 0, 0,.., 1, 3), donde cada uno de los valores es el número de veces que aparece cada término en el documento. La longitud del vector es igual al total de términos de la matriz. De esta manera, un conjunto de m documentos se almacenaría en una matriz de m filas por n columnas, siendo n el total de términos almacenamos en dicho conjunto de documentos.

Teniendo en cuenta el ejemplo anterior es posible que una palabra aparezca más de una vez en un mismo documento, pero para la consulta del usuario dicho término no tenga suficiente relevancia, y también puede ser necesario que algunas palabras deban ser consideradas con más peso o más significativas que otras. De acuerdo con lo anterior se ve la necesidad de dar un peso a cada uno de los componentes del vector, debido a que el simple conteo de términos de un documento no es suficiente garantía para clasificar la relevancia de cada documento frente a una consulta, de igual forma se debe evitar privilegiar documentos muy extensos frente a otros menos extensos, para lograr esto se debe normalizar los vectores de los documentos a través de la ecuación siguiente [21]:

$$\vec{d}_i = \frac{1}{\sqrt{\sum_{j=1}^{m} w_{ij}^2}} (w_{i1}, w_{i2}, ..., w_{im}) \quad (2)$$

La ecuación anterior multiplica cada uno de los elementos del vector, por el inverso de su norma, lo que garantiza un porcentaje de acuerdo a la cantidad de apariciones de un determinado término dentro del conjunto de todas las apariciones de los términos para un documento específico.

La realización del cálculo del peso de cada término en el vector documento se ha estimado con base en el siguiente análisis; si un término posee una frecuencia alta de aparición en un documento, es importante para la relevancia de dicho documento. O si su frecuencia de aparición es alta en muchos otros documentos de la colección, no es beneficioso para distinguir un documento de los demás. Para resolver dicho predicamento se lleva un conteo del número de ocurrencias en dicho documento, dando como resultado la frecuencia del término en el documento (tf). Ahora es necesario consultar la frecuencia de un término en toda la colección de documentos; si la frecuencia es muy elevada, es posible que pertenezca al conjunto de palabras vacías, como solución se opta por eliminarlo del conjunto de términos de la colección. Como conclusión es posible afirmar que la importancia de un término es inversamente proporcional a su frecuencia en la colección de documentos, a esto se lo conoce con el nombre de frecuencia de documento inversa (idf). Con estas bases se calcula el peso de cada elemento del vector documento teniendo en cuenta su frecuencia inversa en la colección y su frecuencia dentro de cada documento, mediante la ecuación 3 [20][22].

$$w_{ij} = tf_i \cdot idf_j \tag{3}$$

Aunque se han desarrollado varias formulas para hallar pesos, una de las más usadas es:

$$w_{ij} = tf_{ij} \cdot \log \frac{N}{df_j} \tag{4}$$

Donde N es el número de documentos de la colección y df_j es la cantidad de documentos donde aparece el término j.

De igual forma el proceso elaborado para los documentos se aplica a la consulta realizada por el usuario, transformando la consulta en un vector m-dimensional, para luego ser procesada buscando entregar los documentos más relevantes para dicha consulta. Dicha relevancia está basada en una medida de similitud, es decir un documento en particular es relevante para la consulta del usuario si su representación vectorial es similar a la representación vectorial de la consulta realizada.

Se dispone de varias formulas para hallar la medida de similitud, pero la más usada es la fórmula del coseno [22][23].

$$\cos\theta = \mathrm{sim}(\mathrm{dj}, \mathrm{q}) = \frac{\overrightarrow{\mathrm{dj}} \cdot \vec{\mathrm{q}}}{\|\mathrm{dj}\| \|\vec{\mathrm{q}}\|} \tag{5}$$

$$\cos\theta = \mathrm{sim}(dj, q) = \frac{\sum_{i=1}^{t} w_{i,j} * w_{i,q}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2}\sqrt{\sum_{i=1}^{n} w_{i,q}^2}} \quad (6)$$

La medida de similitud entre un documento d_j y una consulta q es el coseno del ángulo entre estos dos vectores. Si la similitud entre los vectores da como resultado cero, significa que no hay coincidencia alguna entre los componentes de los vectores, ya que el producto escalar será cero. Si la similitud entre los vectores es 1, se estará hablando de la similitud máxima que sólo se da cuando todos los componentes de los vectores son iguales, en este caso la función del coseno obtiene su máximo valor, la unidad.

4.1.3 MODELO PROBABILÍSTICO

Es un modelo introducido en 1976 por Roberston y Sparck Jones [20]. Para el modelo probabilístico los pesos de los términos, tanto de los documentos como de la consultas son todos binarios. Una consulta q es un subconjunto de términos. Donde R es el conjunto de documentos conocidos como relevantes y $\overline{R}$ es el complemento de R. $P(R \mid \vec{d}_j)$ es la probabilidad de que el documento d_j sea relevante a la consulta q y $P(\overline{R} \mid \vec{d}_j)$ sea la probabilidad de que el documento d_j no sea relevante a la consulta q. La similitud $sim(d_j, q)$ del documento d_j a la consulta q está definida por la siguiente fórmula:

$$sim(d_j, q) = \frac{P(R \mid \vec{d}_j)}{P(\overline{R} \mid \vec{d}_j)} \quad (7)$$

Usando regla de Bayes, se obtiene:

$$sim(d_j, q) = \frac{P(\vec{d}_j \mid R) \times P(R)}{P(\vec{d}_j \mid \overline{R}) \times P(\overline{R})} \quad (8)$$

Donde $P(\vec{d}_j \mid R)$ es la probabilidad de seleccionar al azar el documento d_j del conjunto R de documentos relevantes. Además $P(R)$ es la probabilidad de que un documento seleccionado al azar de la colección completa sea relevante. Los significados para $P(\vec{d}_j \mid \overline{R})$ y $P(\overline{R})$ son análogos y complementarios. Además $P(R)$ y $P(\overline{R})$ son los mismos para todos los documentos en la colección.

4.2 DESCOMPOSICIÓN EN VALORES SINGULARES Y LSI

Uno de los componentes necesarios para el desarrollo de la indexación semántica latente (LSI por sus siglas en inglés) es la descomposición en valores singulares (SVD por sus siglas en inglés), es por dicha razón que se explicará el concepto previamente.

4.2.1 DEFINICIÓN

Dada una matriz A de m x n donde m ≥ n y rango(A)=r, el SVD de A, denotado por SVD(A) se define como [24][25]:

$$A = U \sum V^T \; A = U \sum V^T \tag{9}$$

Donde $U^T U = V^T V = I_n$ $U^T U = V^T V = I_n$ y $\sum = diag(\sigma_1,..., \sigma_n), \sigma_i > 0 \sum = diag(\sigma_1, \dots \dots \dots \dots \sigma_n)$, para 1 ≤ i ≤ r, $\sigma_j = 0 \sigma_j = 0$ para j ≥ r + 1. Las primeras columnas r de las matrices ortogonales ***U*** y ***V*** definen los eigenvectores ortonormales asociados con los eigenvalores r distintos de cero de $A \cdot A^T \; AA^T$y $A^T \cdot A$ respectivamente. Las columnas de **U** y **V** se refieren a los vectores singulares izquierdo y derecho, respectivamente y los valores singulares de **A** se definen como los elementos diagonales de $\sum$ que son las raíces cuadradas no negativas de los eigenvalores n de $A \cdot A^T$.Los siguientes teoremas muestran cómo SVD revela información importante sobre la estructura de una matriz.

TEOREMA 1. Sea el SVD de A dado por la ecuación (9) y

$$\sigma_1 \geq \sigma_2 \dots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_n = 0 \sigma_1 \geq \sigma_2 \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_n = 0$$

Y sea $R(A)$ y $N(A) N(A)$ el rango y espacio nulo de A respectivamente. Entonces

- *Propiedad rango*: $rango(A) = r, N(A) \equiv span\{v_{r+1},..., v_n\}$, y

$R(A) \equiv span\{u_1,..., u_r\} R(A) \equiv span\{u_1, \dots, u_r\}$, donde $U = [u_1 u_2 ... u_m]$ y $V = [v_1 v_2 ... v_n]$

- *Descomposición Dinámica*: $A = \sum u_i \cdot \sigma_i \cdot v_i^T \; A = \sum u_i . \sigma_i . v_i^T$

- *Norma*: $\|A\|_F^2 = \sigma_1^2 + ... + \sigma_r^2$, y $\|A\|_2^2 = \sigma_1 \; \|A\|_F^2 = \sigma_1^2 + \cdots + \sigma_r^2, \; y \; \|A\|_2^2 = \sigma_1$

TEOREMA 2:
Sea el SVD de A dado por la ecuación (1) con $r = rango(A) \leq p = \min(m,n) r = rango(A) \leq p = \min(m,n)$ se define

$$A_k = \sum_{i=1}^{k} u_i \cdot \sigma_i \cdot v_i^T \qquad (10) A_k = \sum_{i=1}^{k} u_i . \sigma_i . v_i^T$$

Entonces

$$\min_{rank(B)=k} \|A-B\|_F^2 = \sigma_{K+1}^2 + ... + \sigma_P^2 \; \mathbf{min}_{rank(B)=k} \|\boldsymbol{A-B}\|_{\boldsymbol{F}}^{\boldsymbol{2}} = \boldsymbol{\sigma_{k+1}^2 + \cdots \sigma_p^2} \tag{11}$$

En otras palabras, A_k, que se construye de las tripletas singulares k más grandes de A, es la matriz de rangos k más cercana de A. De hecho, A_k es la mejor aproximación de A para cualquier norma invariante unilateral. Por lo tanto,

$$\min_{rango(B)=k} \|A-B\|_2 = \|A-A_K\|_2 = \sigma_{K+1} \tag{12}$$

El resultado logrado en la ecuación (10) es una aproximación de la matriz original de términos por documentos, esta nueva matriz representa las relaciones originales, pero como un conjunto de factores ortogonales. Los documentos entonces son expresados nuevamente a través de dichos factores, logrando la combinación de significados de términos y documentos, y eliminando los documentos fuera de la semántica del contexto (documentos que incluyen los términos pero que no tienen sentido con lo que se desea buscar), dichos documentos son considerados como ruido.

En cuanto a la Indexación Semántica Latente (LSI), en [26] se define como un método para la extracción y representación del conocimiento. Se basa en la utilización contextual de las palabras en un corpus[6] de gran tamaño para extraer y representar el significado de las palabras, haciendo uso de cálculos estadísticos. LSI surgió como una herramienta para la indexación y recuperación automática de la información, con el propósito de superar una deficiencia fundamental de algunas técnicas de recuperación de la información: el problema de la semántica. A pesar de que LSI no entiende el significado de las palabras, el reconocimiento de patrones lo hace parecer inteligente.

4.2.2 ALGORITMO PARA LSI

A continuación se muestran los pasos que se deben tener en cuenta al desarrollar el algoritmo para LSI.

- Convertir cada documento en un vector de ocurrencias de palabras. La dimensión del vector será igual al número de palabras únicas en el documento. Se recomienda quitar palabras consideradas como vacías o que no significan nada.
- Escalar el vector de manera que cada término refleje la frecuencia de su ocurrencia en el contexto.
- Se combinan estos vectores con los documentos. Donde las filas representan los términos y las columnas los documentos.

[6] Corpus: Conjunto lo más extenso y ordenado posible de datos o textos científicos, literarios. Etc., que pueden servir de base a una investigación.

- Se realiza SVD en la matriz de términos por documentos, obteniendo la correspondiente $U\sum V$
- Considerar los valores más altos de acuerdo al parámetro k.
- Combinar de nuevo los términos para formar la matriz original.
- Llevar esta matriz de términos por documentos de nuevo a vectores columna.
- Finalmente se tiene un LSI.

Para implementar LSI, se debe construir una matriz de términos por documentos. Los elementos de la matriz términos por documentos son las ocurrencias de cada palabra en un documento particular,

$$A = \lfloor a_{ij} \rfloor$$

Donde a_{ij} denota la frecuencia con que el término i ocurre en el documento j. En la práctica se aplican pesos locales y globales para incrementar o disminuir la importancia de los términos dentro o entre los documentos. Específicamente podemos escribir

$$a_{ij} = L(i, j) \cdot G(i) \quad (13)$$

Donde $L(i, j)$ es el peso local del término i en el documento j, y $G(i)$ es el peso global para el término i.

La matriz A se factoriza con el producto de tres matrices en ecuación (9) usando descomposición de valor singular (SVD). El SVD origina el modelo de la estructura semántica latente de las matrices ortogonales *U* y *V* contenidos en los vectores singulares izquierdo y derecho de A respectivamente y la matriz diagonal ∑ de los valores singulares de *A*. Esas matrices reflejan la descomposición de las relaciones de los vectores linealmente independientes o *valores factor.* El uso de *k* factores o *k* tripletas singulares mas grandes es equivalente a aproximar la matriz términos – documentos de A_k. El uso de *k* factores o *k* tripletas singulares más grandes es equivalente a aproximar la matriz términos – documentos original a $\boldsymbol{A_k}$ en la ecuación (2**).**

El modelo de Indexación Semántica Latente [27] (LSI) es una extensión del modelo vectorial, y presenta una ventaja frente a este, reduce el "ruido" al recuperar documentos; ya que si existen varios documentos que incluyen un término específico solicitado en una consulta; algunos pueden ser considerados como documentos no relevantes debido a que su semántica no es semejante a la que se desea buscar. El modelo de Indexación Semántica Latente busca eliminar dichos documentos, contando con la premisa de que aquellos que tienen la misma semántica se localizan en posiciones cercanas en un espacio multidimensional [28].

LSI y SVD han sido usados en varias publicaciones como aspecto de investigación en la recuperación de información en texto oculto [29], en la evaluación de vínculos de hipertexto [30], en Carrot[2] y Lingo, trabajos que serán descritos a continuación.

Carrot^2 [31] es un marco de trabajo de código abierto desarrollado por David Weiss. Su arquitectura está basada en un conjunto de componentes distribuidos cooperando entre sí a través del intercambio de datos XML. El objetivo principal de Carrot^2 es facilitar la realización de pruebas de procesamiento y la visualización de los resultados entregados en una consulta web. Para lograr su objetivo cuenta con 5 algoritmos de clustering de los cuales uno de ellos es Lingo.

Lingo, es un algoritmo de clustering web, que adopta un enfoque diferente a la hora de etiquetar los clústeres, utilizando para ello como paso inicial encontrar las descripciones significativas de los clústeres, para luego determinar su contenido basado en dichas descripciones. El algoritmo se encarga de asegurar que todas las etiquetas difieran entre sí de la forma más significativa posible. Para lograr tal cometido Lingo hace uso de uno de los modelos clásicos de la recuperación de información; el modelo vector con LSI [31]. Lingo también utiliza SVD para extraer vectores ortogonales de la matriz de términos-documentos, buscando representar los diversos temas como entrada de datos [32].

A diferencia de los sistemas anteriormente mencionados la solución propuesta utiliza el trabajo con sistemas de clasificación del conocimiento, a través de taxonomías y ontologías, que ayudan en la identificación semántica de los términos de un tema específico en los resultados entregados a una consulta web.

Para la recuperación de información se han desarrollado avances que buscan presentar la información más relevante a la consulta realizada por el usuario. Algunos de dichos avances son los buscadores, que a continuación son descritos junto con una explicación de sus diferentes tipos.

5. BÚSQUEDA WEB

Los buscadores son sistemas que permiten recuperar información específica de la web. Ellos presentan al usuario una interfaz para que en ellas se ingrese una petición (la consulta sobre el tema que se necesita), el sistema realiza la búsqueda y devuelve los enlaces a los documentos para que el usuario los analice, acceda a ellos y decida si le sirven o no. Existen tres tipos de buscadores: índices temáticos, motores de búsqueda y meta buscadores [33].

Los **Índices temáticos** son listas de recursos organizadas en jerarquías desde lo más general a lo más específico. El proceso de clasificación se hace de forma manual. **Ventajas**: son fáciles de usar para usuarios no experimentados, la búsqueda se realiza eligiendo la categoría que más se acerca a la consulta y se desciende hasta encontrar los enlaces a los recursos deseados, hay menos ruido en los recursos y existe un proceso de selección de calidad. **Desventajas**: sólo cubre una pequeña parte de los recursos de la Web. No existen criterios homogéneos para la clasificación y selección de recursos. **Ejemplos**: google (www.google.com), yahoo (www.yahoo.com), live (www.live.com), terra (www.terra.es), galaxy (www.galaxy.com).

Los **Motores de búsqueda** recorren la red recolectando e indexando la mayor cantidad de información posible, gracias a programas automáticos conocidos como robots, también llamados spider o crawler. **Ventajas**: Los procesos de recolección e indexación son automáticos, por lo que se recoge gran cantidad de información y además pueden contar con formas de actualización automática. **Desventajas**: Los llamados robots tienen restricciones de navegación sobre la Web profunda ya que sus contenidos son generados dinámicamente mediante consultas, autenticaciones, autorizaciones entre otras y solo recorren la Web superficial. Son más complejos de usar. Se debe conocer la sintaxis de búsqueda del motor. Se debe ser extremadamente cuidadoso cuando se realiza una consulta para obtener un resultado óptimo. No existe un proceso de selección de calidad y fiabilidad de los contenidos. **Ejemplos**: google (www.google.com), altavista (www.altavista.com).

Los **Meta buscadores** no disponen de bases de datos propias, puesto que buscan sobre los resultados de los buscadores. Recogen la petición del usuario y la envían a los buscadores, éstos devuelven los resultados y los meta buscadores la clasifican antes de presentarla al usuario. **Ventajas**: La búsqueda es más extensiva, el usuario accede a una sola página, la consulta se digita una sola vez. **Desventajas**: Al formular la consulta una sola vez, es posible que la sintaxis no sea la más adecuada para cada uno de los buscadores. El proceso de búsqueda es lento. **Ejemplos**: Webferret (www.webferret.com), Copernic (www.copernic.com), metacrawler (www.metacrawler.com).

6. DESCUBRIMIENTO DE CONOCIMIENTO EN BASES DE DATOS

El término KDD (Knowledge Discovery in Database) Descubrimiento de conocimiento en Bases de Datos tiene como objetivo encontrar patrones y similitudes en datos crudos[7][16], KDD considera al conocimiento como producto del proceso de descubrimiento guiado por los datos. El proceso parte inicialmente de una base de datos y busca identificar patrones válidos, que puedan ser útiles y comprensibles [34].

El proceso KDD representado en la Figura 3 tiene varias etapas que son ejecutadas de forma iterativa e interactivamente, tiene cierto nivel de complejidad al incluir acciones que involucran la búsqueda de estructuras, modelos y parámetros en la base de datos. Para lograr dicho proceso se comienza con la definición y comprensión del problema, se continúa con la selección de los datos objetivo, luego con el procesamiento previo de los datos donde se incluye una limpieza de los datos[8], la transformación y aplicación de Minería de Datos y la interpretación de los patrones obtenidos o análisis de resultados.

[7] Datos que aún no han sido procesados, también conocidos como datos fuente.
[8] La limpieza de datos es el proceso de corregir o remover información incorrecta, con formato inapropiado o duplicado en una base de datos.

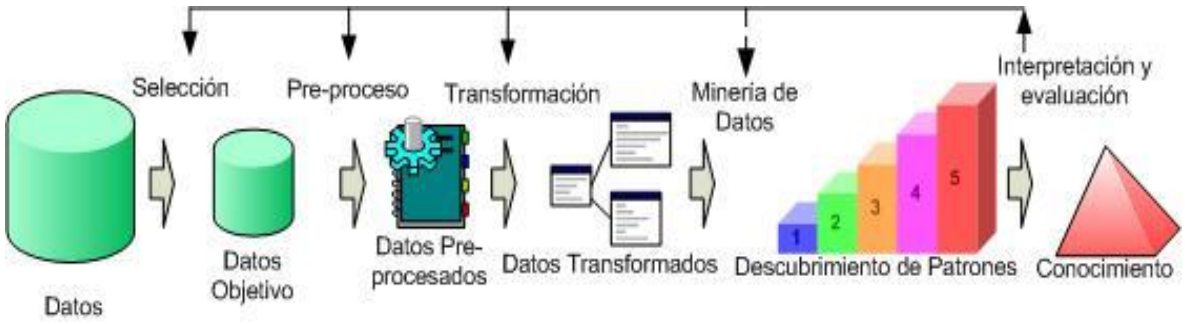


Figura 3 Etapas del Proceso KDD (Tomado de [34])

6.1 MINERÍA DE DATOS

La minería de datos es el proceso de extracción de información desconocida, válida y útil, de grandes bases de datos (o cualquier otra fuente de almacenamiento de datos), usada en la toma de decisiones de negocios [16]. La minería de datos puede ser empleada como un paso dentro del proceso de descubrimiento de conocimiento en bases de datos KDD (Knowledge Discovery in Databases) [35]. Según este punto de vista, en un sistema de minería de datos se deben tener los siguientes componentes:

- **Base de datos, Bodega de datos u otro repositorio de información**: esto es, un conjunto de bases de datos, bodegas de datos o cualquier otro medio de almacenamiento de información, a los cuales se les puede aplicar técnicas de limpieza e integración de datos.
- **Servidores de bases de datos o de bodegas de datos**: responsables de sacar los datos más relevantes, basándose en los requerimientos de minería de datos del usuario.
- **Base de conocimiento:** esto es, el dominio del conocimiento que se usa para guiar la búsqueda o evaluar los patrones resultantes. Tal conocimiento puede incluir el concepto de jerarquías, usado para organizar atributos o valores de atributos en diferentes niveles de abstracción. Se puede denominar dominio del conocimiento a los patrones generados a partir de las creencias de los usuarios, restricciones, umbrales y metadatos (que describen datos de múltiples fuentes heterogéneas).
- **Motor de Minería de Datos:** esto es esencial para un sistema de minería de datos e idealmente consiste de un conjunto de módulos funcionales para tareas tales como caracterización, análisis de asociación, clasificación, análisis de evolución y análisis de desviación.
- **Módulo de evaluación de patrones:** este componente emplea medidas e interactúa con el motor de minería de datos para enfocar las búsquedas hacia patrones de interés. Puede acceder a umbrales almacenados en la base del conocimiento.

Alternativamente el módulo de evaluación de patrones puede ser integrado con el módulo de minería, dependiendo de la implementación del método de minería de datos usado. Para hacer la minería de datos eficiente, es altamente recomendado realizar la evaluación de patrones tan profunda como sea posible, así se confía la búsqueda solamente a patrones realmente interesantes.

- **Interfaz gráfica de usuario:** este módulo comunica al usuario con el sistema de minería de datos, permitiéndole al usuario interactuar con el sistema para especificar una tarea de consulta, proporcionando información para ayudar a enfocar la búsqueda y realizar exploraciones basado en los resultados intermedios de minería de datos. Además este módulo le permite al usuario examinar bases de datos y esquemas de bodegas de datos o estructuras de datos, evaluar patrones y visualizarlos de diferentes formas.

Una aplicación que use tecnologías de Minería de Datos implementará una o más tareas que reflejan la manera de distinguir patrones o tendencias en un conjunto de datos complejo. Las tareas más comunes [36][37], y de las cuales algunas han sido empleadas para desarrollar los sistemas mostrados en [38][39] son:

- **Clasificación**: Es el proceso de subdividir un conjunto de datos con la consideración de un número de resultados específicos, permitiendo así descubrir patrones. Las técnicas más comunes de clasificación son los árboles de decisión y las redes neuronales.
- **Clustering**: Es una tarea no supervisada usada para encontrar grupos de registros con características similares. Para el clustering se usan métodos estadísticos, el algoritmo k-means[9] [40], o una forma especial de red neuronal llamada Kohonen[10] [41], entre otros. Cada registro se compara con un conjunto de clusters existentes para ser asignado al cluster más cercano. El proceso de agrupación es acorde con los procesos humanos de información y una de las motivaciones para usar algoritmos de clustering es proveer herramientas automáticas que ayuden a la construcción de taxonomías. Para el clustering también se suele usar algoritmos genéticos, el algoritmo de Calinski, Harabaz y el muestreo estadístico [36].
- **Análisis de Asociación**: Es una tarea no supervisada que analiza los enlaces entre los registros y un conjunto de datos, usada en grandes bases de datos para encontrar relaciones entre datos.
- **Predicción**: Permite plantear estrategias para afrontar futuras situaciones gracias a clasificaciones que poseen características particulares. Normalmente es elaborada con funciones de regresión para examinar las relaciones entre variables con el fin de predecir valores futuros.

[9] K-means. Es un algoritmo que particiona clusters, reasigna muestras de datos a clusters basadas en la similitud entre estas.
[10] Redes Kohonen. Son mapas auto-organizados que manifiestan aprendizaje Kohonen, el cual consiste en un ajuste de pesos de los nodos vecinos de un nodo ganador.

A diferencia de los sistemas elaborados en [38][39], la propuesta planteada presenta la interacción con taxonomías y las ontologías, los cuales son sistemas de clasificación del conocimiento, tanto general como específico además de interactuar con modelo matemáticos como son el modelo vectorial o vector, SVD.

6.2 ALGORITMO DE CLUSTERING

Los algoritmos de clustering permiten clasificar un conjunto de elementos en un determinado número de grupos basándose en las semejanzas y diferencias existentes entre esos elementos [42]. Una clasificación general divide los algoritmos de clustering en: clustering jerárquico, clustering basado en densidad, clustering particional, clustering basado en grid y basado en modelos [42].

6.2.1 ALGORITMO DE CLUSTERING K-MEANS

El algoritmo de clustering k-means (k-medias) es un algoritmo preciso y eficaz para encontrar clústeres en conjuntos de datos. El procedimiento para el algoritmo se describe a continuación [40]:

- Paso 1: Pregunta al usuario cuántos clusters k en el conjunto de datos deben dividirse.
- Paso 2: Aleatoriamente asigna k registros para ser el centro de los clusters iniciales.
- Paso 3: Para cada registro, encuentra el centro del cluster más cercano. Así, en cierto sentido, cada centro del cluster posee un subconjunto de registros, representando una partición del conjunto de datos. Por consiguiente tenemos k clusters, C1, C2,..., Ck.
- Paso 4: Para cada uno de los k clusters, encuentra el cluster centroide, y modifica la ubicación da cada centro de cluster al nuevo valor del centroide.
- Paso 5: Repita los pasos 3 a 5 hasta que converja o termine.

El criterio más cercano en el paso 3 es usualmente la distancia Euclidiana, aunque pueden ser aplicados otros criterios. El cluster centroide en el paso 4 se encuentra como sigue. Suponga que tenemos n puntos de datos (a1, b1, c1), (a2, b2, c2),..., (an, bn, cn), el centroide de estos puntos es el centro de gravedad de esos puntos y está localizado en el punto (∑ai/n, ∑bi/n, ∑ci/n). Por ejemplo, los puntos (1, 1, 1), (1, 2, 1), (1, 3, 1), y (2, 1, 1) tendrían como centroide;

$$\left(\frac{1+1+1+2}{4}, \frac{1+2+3+1}{4}, \frac{1+1+1+1}{4}\right) = (1.25, 1.75, 1.00)$$

El algoritmo termina cuando los centroides ya no cambian. En otras palabras, el algoritmo termina cuando para todos los clusters C_1, C_2,..., C_k, todos los registros poseídos en cada centro de cluster permanecen en ese cluster. Alternativamente, el algoritmo termina cuando algún criterio de convergencia es encontrado, tal como la reducción insignificante en la suma de errores cuadrados:

$$SSE = \sum_{i=1}^{k} \sum_{p \in C_i} d(p, m_i)^2$$

Donde $p \in C_i$ representa cada punto de datos en el cluster i y m_i representa el centroide del cluster i.

6.2.2 ALGORITMOS DE CLUSTERIZACIÓN JERÁRQUICA

Los algoritmos de clustering son jerárquicos o no jerárquicos. En clustering jerárquico, una estructura clúster en forma de árbol se creada mediante particiones recursivas (método divisivo) o combinación (aglomerativo) de clústeres existentes. Los métodos de clustering aglomerativo inicializan cada observación para que sea un clúster pequeño de sí mismo [40].

Entonces, en los pasos subsiguientes, los dos clústeres más cercanos se agregan en un nuevo clúster combinado. De esta manera, el número de clústeres en conjunto de datos es reducido por uno en cada paso. Luego, todos los archivos se combinan en un solo clúster grande. Los métodos de clustering divisivos empiezan con todos los registros en un clúster grande, con los registros más diferentes que son divididos recursivamente, en un clúster separado, hasta que cada registro represente su propio clúster. La mayoría de programas de computador que aplican métodos de clustering jerárquico usan métodos aglomerativos.

Luego de que la distancia entre registros y la normalización se ha realizado se realizan las siguientes preguntas ¿Cómo determinar la distancia entre clúster de registros? ¿Se debe considerar que dos clústeres son parecidos si sus vecinos más cercanos son parecidos o si sus vecinos más lejanos son parecidos? ¿Con qué criterio se promedia esos extremos? A continuación se examinan varios criterios para determinar la distancia entre los clúster arbitrarios A y B:

- Unión Simple, a veces llamado acercamiento de los vecinos más cercanos, están basados en la distancia mínima entre cualquier registro en el clúster A y cualquier registro en el clúster B. En otras palabras, la similitud del clúster está basada en la similitud de los miembros más similares de cada clúster. La unión simple tiende a formar grandes clústeres delgados que a veces pueden llevar a que registros heterogéneo sean agrupados juntos.
- Unión completa, algunas veces llamado el acercamiento del vecino más lejano, se basa en la máxima distancia entre cualquier registro en el clúster A y cualquier registro en el clúster B. En otras palabras, la similitud del clúster está basada en la similitud del miembro más diferente de cada clúster. La unión completa tiende a formar clústeres más compactos, clústeres como esferas, con todos los registros en un clúster dentro de un diámetro dado de todos los otros registros.

- Unión promedio es diseñada para reducir la dependencia del criterio de unión de clúster en los valores extremos, como los registros más similares o los más diferentes. En la unión promedio, el criterio es la distancia promedio de todos los registros en el clúster A de todos los registros en el clúster B. El clúster resultante tienden a tener aproximadamente igual variabilidad dentro del clúster.

Uno de los de los sistemas encontrados que trabaja con algoritmos de clustering es iBoogie [43], el cual es un sitio de búsqueda desarrollado por CyberTavern, el cual usa un algoritmo de clustering llamado Clusterizer desarrollado por la misma empresa. Las diferencias encontradas con la propuesta presentada resaltan en: la falta de un sistema de filtro de documentos que le permita separar documentos con semántica relacionada de aquellos que no, debido a que agrupa todos los resultados entregados para generar los clústeres de presentación de los resultados, combinando documentos no relevantes y relevantes con semántica dispersa. No permite realizar un feedback sobre los resultados, simplemente agrupa los resultados retornados por otros buscadores.

6.3 DESCUBRIMIENTO DE CONOCIMIENTO EN TEXTOS

El área de KDT (Knowledge Discovery in Text) Descubrimiento de Conocimiento en Textos y TM (Text Mining) Minería de Texto son áreas que se han desarrollado de una forma muy rápida a causa de la necesidad de analizar la gran cantidad de información textual que reside tanto en los sistemas de archivos internos como en la Web. KDT es el proceso de identificación de patrones válidos, nuevos, potencialmente útiles y finalmente entendibles en datos textuales no estructurados [44]. KDT incluye múltiples pasos, dentro de los cuales se incluyen todas las tareas de recolección de documentos con el fin de visualizar la información obtenida y su objetivo principal es obtener patrones entendibles a los seres humanos. TM es considerada un paso en el proceso de KDT consistente de la minería de datos particular y los algoritmos de Procesamiento de Lenguaje Natural (NLP) que producirán patrones sobre un conjunto de datos textuales no estructurados. El objetivo principal de TM es descubrir conocimiento en grandes colecciones de documentos [44]. El proceso KDT tiene tres etapas principales que se describen a continuación (ver Figura 4).

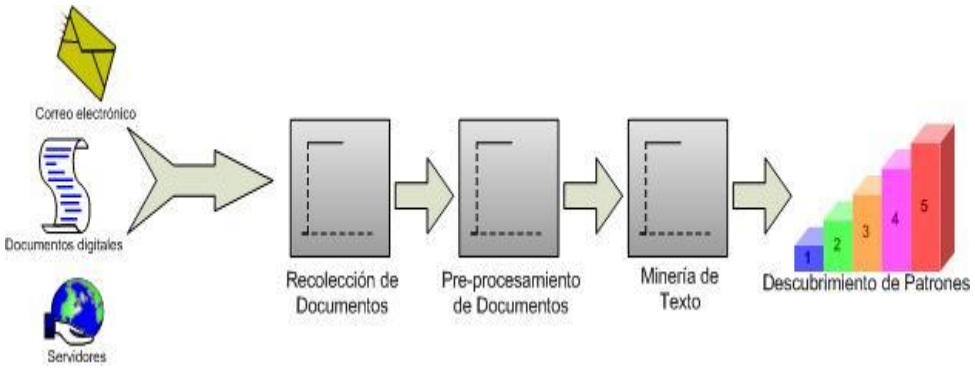


Figura 4 Etapas del Procesamiento KDT (Tomado de [44])

- Recolección de documentos relevantes: etapa en la cual se identifican los documentos relevantes necesarios para el cumplimiento de los objetivos, una vez identificada la fuente se procede con la recuperación ya sea de la Web o de algún sistema de archivos interno.
- Pre-procesamiento de los documentos: Este paso incluye algún tipo de proceso de transformación de los documentos recuperados.
- Operaciones de Minería de Texto: la información de alto nivel es extractada y a partir de dicha información los patrones son descubiertos.

7. ONTOLOGÍAS Y TAXONOMÍAS

Las Ontologías se definen como una especificación explícita y formal sobre una conceptualización compartida [45]. Esto quiere decir que las ontologías definen conceptos y relaciones de algún dominio, de forma compartida y consensuada, y que esta conceptualización debe ser representada de una manera formal, legible y utilizable por los computadores [45]. De esta manera la Ontología proporciona un marco para entender la realidad así como una clasificación de la misma, de la cual se pueden extraer los términos para permitir crear una abstracción de la realidad [46]. Las Ontologías tienen entre otros, los siguientes componentes que servirán para representar el conocimiento de algún dominio:

- **Conceptos:** son las ideas básicas que se intentan formalizar. Los conceptos pueden ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento, etc.
- **Relaciones:** representan la interacción y enlace entre los conceptos del dominio. Suelen formar la taxonomía del dominio. Por ejemplo: subclase-de, parte-de, parte-exhaustiva-de, conectado-a, etc.
- **Funciones**: son un tipo concreto de relación donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la ontología. Por ejemplo, pueden aparecer funciones como categorizar-clase, asignar fecha, etc.
- **Instancias**: se utilizan para representar objetos determinados de un concepto.
- **Axiomas**: son teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología. Por ejemplo: “Si A y B son de la clase C, entonces A no es subclase de B”, “Para todo A que cumpla la condición C1, A es B”, etc.

A diferencia de [47][48], la propuesta presentada realiza el trabajo con taxonomías y ontologías que amplían el rango de cubrimiento de la información recuperada, generando los conceptos pertinentes a cada tema específico y no utiliza agentes en la recuperación de información pues los resultados son entregados por buscadores que realizan dicha tarea [49], además de combinar el trabajo con técnicas y algoritmos de minería de datos.

7.1 TAXONOMÍAS DEL CONOCIMIENTO

Etimológicamente hablando, taxonomía procede de los términos griegos “taxis”, ordenación, y “nomos”, norma. Aristóteles fue uno de los primeros en utilizar este término, en el 300 antes de Cristo, para designar esquemas jerárquicos orientados a la clasificación de objetos científicos. El botánico Karl von Linneo[11] designó con el término taxonomía a la clasificación de los seres vivos en agrupaciones jerárquicamente ordenadas de lo más general a lo más específico (reino, clases, orden, género y especies) [50].

De manera general, se define una taxonomía como una estructura organizada de forma jerárquica que representa algún tipo de conocimiento, en ella se crean categorías para organizar los elementos en mapas simples [51][19]. Una de las ventajas de las taxonomías es que proporcionan una base sistemática y estructurada de un campo del conocimiento a diferentes niveles de abstracción, y han sido usadas desde Aristóteles para cubrir con una compleja y enorme base de información.
Una taxonomía general del conocimiento, es una taxonomía que representa diferentes áreas o disciplinas del conocimiento de manera general (no son específicas a un área de conocimiento). Las partes de una taxonomía son [52].

- **La relación jerárquica:** vincula conceptos desde lo general a lo específico. La relación jerárquica también se llama una relación ES_UN.
- **Nivel:** una jerarquía consiste de varios niveles. El nivel más alto es el más abstracto y los niveles inferiores son más concretos. Todos los elementos en un nivel deberían tener aproximadamente el mismo grado de abstracción.
- **Raíz,** la raíz es el tope de la estructura, usualmente el dominio de la estructura.
- **Nodo,** denota un concepto en la estructura. La mayoría de los nodos son padres (de un nivel inferior) e hijos (del nivel más alto).
- **Nodo superior,** es un concepto en el primer nivel bajo la raíz de la taxonomía.
- **Nodo hoja,** un nodo que no tiene nodos hijos.
- **Hermano,** un nodo que tiene el mismo nodo padre que otro nodo.
- **Camino,** la secuencia de nodos que son recorridos para alcanzar un nodo específico.

Ejemplos de taxonomías incluyen el orden zoológico, botánico, clasificaciones en bibliotecas, etc. [53] .

[11] Carl von Linneo (1707-1778). Naturalista sueco que desarrolló la nomenclatura binómica para clasificar y organizar a los animales y las plantas.

CAPÍTULO III – MODELO DE BÚSQUEDA PROPUESTO

Desde el surgimiento de la Web hasta nuestros días, la cantidad inmensa de información ha crecido vertiginosamente. Sin embargo, la ausencia de una estructura lógica sumada a un crecimiento desmedido está dando lugar a graves problemas en la recuperación de información, uno de ellos es el problema de la precisión en los resultados que puede ser visto como consecuencia de la falta de significado o semántica que para los computadores tienen los documentos Web. Parte de ello se debe a que dichos documentos están desarrollados en Lenguaje de Marcado de Hipertexto (HTML por sus siglas en inglés) que permite determinar la forma de presentación de los contenidos (colores, tipografía, enmarcado, etc.), mas no un significado de los mismos [54].

Los sistemas de recuperación actual (buscadores, meta buscadores, etc.), se han convertido en herramientas incapaces de ofrecer resultados apropiados a la consulta digitada por el usuario, debido a que algunos de estos realizan la recuperación de información a través de la comparación de palabras claves tecleadas en la consulta contra la inclusión de dichas palabras en los documentos, dando como resultado la recuperación de documentos que aunque incluyen las palabras claves no tienen sentido para el usuario que realizó la consulta. Parte del problema se debe a la polisemia[12], debido a que una misma palabra puede tener varios significados en contextos diferentes. Para ilustrar el caso se toma un ejemplo, la consulta puede incluir la palabra masa, pero varios resultados pueden la misma palabra en contextos diferentes como los son: la masa que moldea un panadero o la masa a la que hace referencia un profesor de física. ¿Cuál debe ser la página a recuperar? Según el sistema de recuperación por palabras claves, los dos resultados mostrados anteriormente serán correctos, ya que contienen la palabra masa.

Parte de la solución a este problema es la web semántica, definida como una extensión de la web actual que permitirá encontrar, compartir y combinar información con mayor facilidad, la cual estará dotada de significados bien definidos para solventar las limitaciones que en la actualidad posee la web [55][56]. Una de las formas de lograr la web semántica es a través de las ontologías, que permiten la organización de la información dentro de un dominio específico.

Por otra parte se han considerado las taxonomías como fuente importante de clasificación del conocimiento, ya que sirven para crear un orden y relacionar conceptos entre sí dando apoyo a la comprensión del dominio de un tema específico. Las Taxonomías pueden ser usadas de dos formas; por un lado como vocabulario para la clasificación de recursos y por otro, para facilitar su recuperación. Además, su estructura jerárquica en forma arborescente puede ser utilizada como estructura visual de navegación en la interfaz de usuario [52].

El modelo propuesto en este trabajo incorpora los siguientes componentes semánticos en la recuperación de información: una taxonomía general del conocimiento y una posible

[12] Polisemia: Pluralidad de significados de una palabra o de cualquier signo lingüístico, según el diccionario de la Real Academia de la Lengua Española.

ontología relacionada con cada nodo en la taxonomía. La taxonomía permite estructurar el conocimiento general de un tema específico y las ontologías incorporan una forma semántica asociada a los nodos de la misma, tomando ventaja de toda la información para situar la búsqueda en el mapa conceptual que constituye la ontología; este proceso centra en un contexto específico la búsqueda, facilitando la recuperación y la presentación de los resultados.

8. MODELO PROPUESTO

El modelo propuesto en este trabajo busca definir un dominio preciso en la búsqueda realizada por el usuario mediante el uso de una taxonomía general del conocimiento y ontologías asociadas a las categorías particulares de dicha taxonomía. Además se incluye minería de datos para clasificar los resultados según la similitud que éstos tengan con respecto a la consulta. Con la combinación de las tecnologías mencionadas anteriormente se pretende que las búsquedas realizadas por los usuarios retornen respuestas más precisas con respecto a las necesidades de los usuarios y que el filtro sobre los documentos sea más efectivo. En las secciones posteriores se explican en detalle los componentes y funcionalidades del modelo, en la Figura 5 se detallan los componentes del mismo.

8.1 PREPROCESAMIENTO DE CONSULTA

La consulta digitada por el usuario es previamente procesada y normalizada antes de ser enviada al siguiente módulo. A continuación se detallan las acciones que se realizan sobre la cadena de consulta en este módulo.

8.1.1 NORMALIZACIÓN DE LA CONSULTA

La normalización de la consulta consiste en tomar los términos que forman la cadena de consulta y realizar las siguientes acciones sobre ella: Eliminar caracteres especiales, si los tiene. Pasar todas las palabras que contienen letras mayúsculas a minúscula. Cambiar las palabras con acento por palabras sin acento. Con los resultados retornados se realiza el mismo procedimiento, para poder comparar el contenido de los documentos con la consulta del usuario.

8.1.2 ELIMINACIÓN DE PALABRAS VACÍAS

Teniendo en cuenta que algunas palabras que se repiten demasiado (conocidas como “stop words” o “palabras vacías”), por ejemplo los artículos, pueden afectar la calidad de los resultados retornados, debido a que no tienen un significado importante, se hace necesario definir un mecanismo para eliminarlas de la consulta. Para realizar este procedimiento se almacena en una base de datos una tabla con esas palabras, así, cuando una persona digita una frase, se comparan los términos de la consulta

normalizada con las palabras vacías, si coinciden, entonces se elimina la palabra de la cadena de consulta.

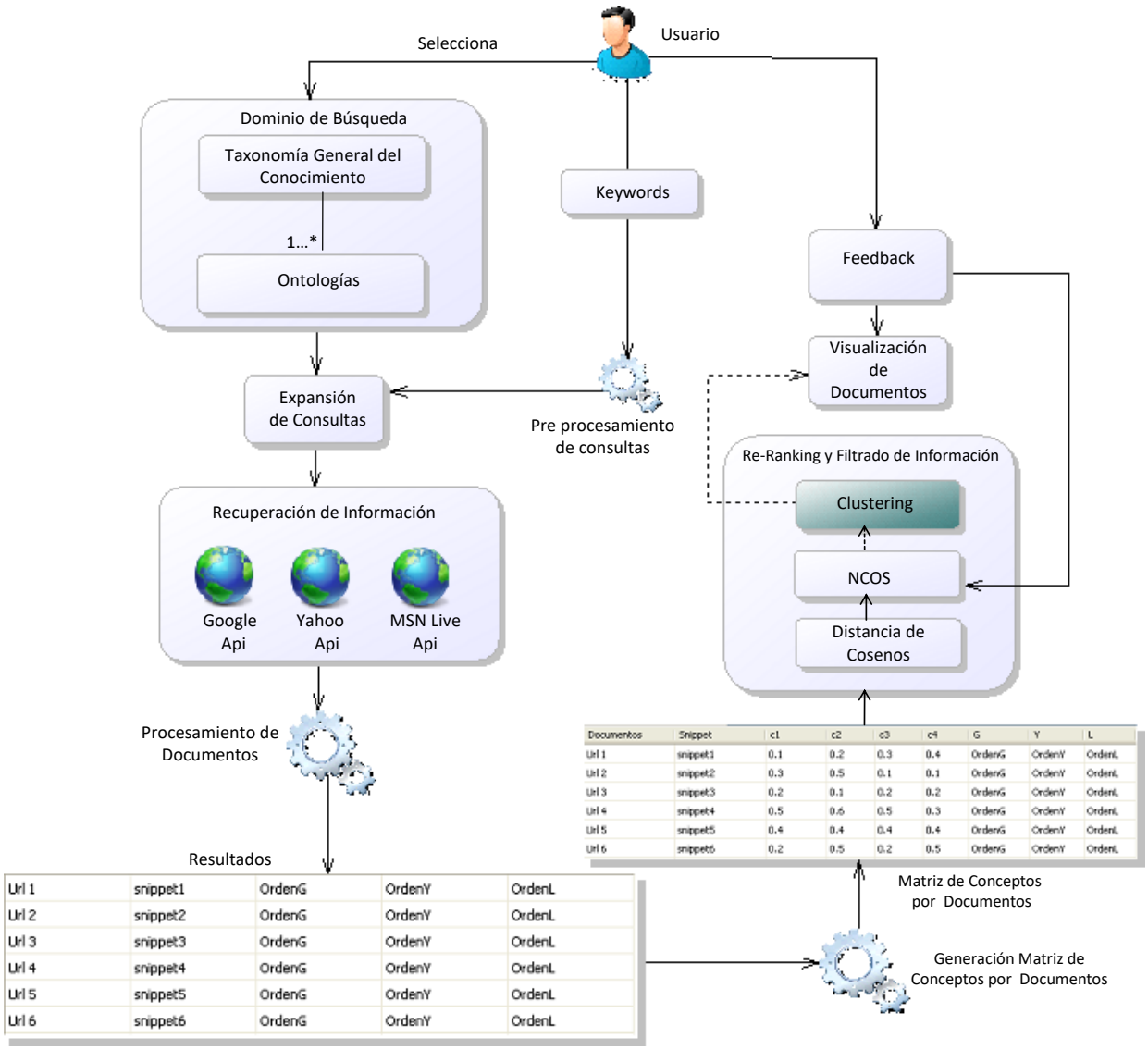


Figura 5 Modelo de búsqueda propuesto.

8.2 DOMINIO DE BÚSQUEDA

En este módulo se tienen dos componentes fundamentales que permiten definir el dominio de la búsqueda, la taxonomía general del conocimiento y las ontologías. Con esto se busca tener mayor precisión en los resultados retornados por los buscadores, a continuación se explican en detalle.

8.2.1 TAXONOMÍA GENERAL DEL CONOCIMIENTO

La Taxonomía General del Conocimiento (TGC) es una taxonomía que representa diferentes áreas o disciplinas del conocimiento de manera general (no son específicas a un área de conocimiento), para este proyecto se hace uso de la edición 22 de la Clasificación Decimal Dewey [57]. La elección para esta propuesta se debe principalmente a que ésta ha sido usada, probada y mejorada por más de 130 años en la mayoría de las bibliotecas del mundo, y en especifico la edición 22 fue concebida en el contexto de la web, frente a otras TGC como lo son la clasificación del conocimiento de DMOZ [58], la Clasificación de la Biblioteca del Congreso[13] de los Estados Unidos [59], la clasificación MERLOT[14] [60], SPAR [61], la clasificación de Yahoo [62] y la de Google [63].

Al realizar una búsqueda, los usuarios pueden navegar a través de la TGC y elegir la categoría del conocimiento sobre la cual desea realizar la consulta. Cada categoría específica de la taxonomía (nodo hoja) tiene asociada una ontología que define el domino para esa temática particular.

Sobre la taxonomía se pueden realizar las siguientes operaciones adicionales:

- Buscar: Se escriben palabras de tal manera que a medida que se van ingresando caracteres se filtran los términos de la taxonomía hasta que quede el tema que se necesita, si existe.

- Selección de varias temáticas. Se pueden seleccionar varios nodos de la taxonomía para realizar una búsqueda, en este caso se toman los conceptos de las ontologías relacionadas con los nodos y los respetivos sinónimos, luego se hace un promedio de la prioridad de los términos similares para expandir la consulta. Cuando un nodo que se selecciona no es hoja (nodo intermedio) se toman las ontologías contenidas por ese nodo (nodos hijos) y se arma una “ontología” en forma dinámica y no permanente.

8.2.2 ONTOLOGÍAS

Para el desarrollo de este proyecto se hace uso de varias ontologías ya creadas y disponibles en la librería de ontologías de DAML [64]. Las ontologías se asocian a temáticas específicas de la taxonomía que al ser elegidas para realizar la búsqueda, éstas permiten que la consulta sea enriquecida, refinando y adicionando términos que tienen igual significado.

Con la utilización de la taxonomía general del conocimiento y las ontologías, se logra que la búsqueda que los usuarios realizan en Internet se defina en un dominio específico del

[13] Biblioteca del Congreso de los Estados Unidos: es una de las bibliotecas ºmás grandes e importantes del mundo. Su colección incluye más de 30 millones de libros catalogados y otros materiales impresos en 470 lenguajes.
[14] MERLOT: Multimedia Educational Resource for Learning and Online Teaching. Repositorio de recursos educacionales para el aprendizaje.

conocimiento, así los resultados obtenidos son consistentes y se acercan más a las necesidades de las personas.

8.3 EXPANSIÓN DE LA CONSULTA

Para expandir la consulta se toman los términos de ésta y se comparan con los conceptos de la ontología asociada a la taxonomía, luego se sacan los sinónimos de cada concepto que coincide con los términos de la búsqueda y se adicionan a la cadena de consulta original haciendo uso del operador lógico OR. Luego se envía la petición de búsqueda a los respectivos buscadores, éstos reciben una cadena de consulta enriquecida por los sinónimos de los conceptos de la ontología. El operador lógico OR permite que se obtengan todos los documentos que contengan al menos uno de los términos de la búsqueda realizada. Si los conceptos no coinciden con los términos de la consulta, entonces se toman todos los conceptos de las ontologías seleccionadas y se obtienen los sinónimos de éstos, teniendo en cuenta la relevancia de los conceptos y se adicionan a la cadena de búsqueda. En este proceso se debe tener en cuenta un límite para la longitud de la cadena de consulta, debido a que las Apis de los buscadores no funcionan con más de un determinado número de palabras por consulta, la longitud máxima permitida por los buscadores es de 120 caracteres [63].

8.4 RECUPERACIÓN DE INFORMACIÓN

Luego de que el usuario ha digitado la consulta y ésta ha sido enriquecida o expandida mediante los sinónimos de los conceptos de la ontología, el módulo de recuperación de información hace uso de tres Api's de los motores de búsqueda más utilizados en la actualidad [1], Google, Yahoo y MSN Live Search respectivamente, para recuperar de Internet la información necesaria.

La información recuperada es almacenada en disco para luego ser procesada y filtrada. De las Url's retornadas por los motores de búsqueda se tiene en cuenta el orden en que apareció para luego compararla con la posición que le asigna el modelo de búsqueda propuesto.

8.5 RESULTADOS DE LOS BUSCADORES

Los resultados retornados por los tres motores de búsqueda se almacenan para posteriormente aplicarles un filtro más riguroso, de éstos se almacena la url, la fecha de consulta, el usuario que realizó la búsqueda, el resumen del contenido de la página web o documento ("snippet") y el orden de aparición de los documentos en cada motor de búsqueda.

8.5.1 PROCESAMIENTO DE DOCUMENTOS

Con el objetivo de facilitar la aplicación de algoritmos matemáticos o vectoriales para realizar el proceso de filtrado, se normaliza el contenido de los documentos, en este caso el contenido de los snippet's. Este proceso consiste en tomar cada snippet y eliminar caracteres especiales, convertir letras mayúsculas a minúsculas y eliminar las palabras vacías, los acentos (palabras con tildes dejarlas sin tildes), entre otras.

Posteriormente se hace el conteo de los términos de la consulta en cada documento, frecuencias, para luego pasar a normalizar los valores y fijarlos entre cero y uno con el fin de crear la matriz definitiva de términos por documentos.

8.6 MATRIZ TÉRMINOS POR DOCUMENTOS

Luego de normalizar el contenido de los snippet's de los documentos se construye la matriz de términos por documentos, en donde los términos incluyen los conceptos de la ontología y sus respectivos sinónimos, cada elemento de la matriz se forma con la frecuencia de cada concepto en el documento multiplicada por la relevancia de ese documento.

La ecuación TF-IDF que se adaptó para el modelo espacio vectorial específico está dada por (14). La matriz tiene N documentos y M Conceptos, $P_{i,j}$ refleja el valor especifico del concepto J en el documento i. Para este caso, FO es la frecuencia observada de un término en el documento que se está procesando, cada T_i es un término asociado al mismo concepto, W es la relevancia del término (este valor se extrae de la ontología que el usuario previamente seleccionó, valor que fue previamente definido por los editores de ontologías). Este cálculo de términos por pesos se acumula para cada concepto y luego de tener las frecuencias observadas de los conceptos, cada valor se divide por el denominador presentado en la ecuación (1), que corresponde a la máxima frecuencia observada de los M conceptos que están en el documento, lo que permite normalizar el valor obtenido al rango [0,1].

$$P_{i,j} = \frac{\sum_{l=1}^{L} FO_{T_l} \times W_{T_l}}{MAX\left(FO_{C_{j=1...M}}\right)} \qquad (14)$$

La matriz además de los conceptos tiene tres columnas relacionadas con la posición (orden) del resultado en Google, Yahoo y MSN Live Search. Esto permite reflejar en el espacio vectorial una dimensión por cada motor que se relacione con la credibilidad o efectividad (basada en una evaluación del algoritmo o de los sitios que han sido indexados por el motor) de los motores de búsqueda tradicional que soportan el modelo. Para dar un ejemplo, y tomando cómo medida la popularidad del motor, se puede decir que hoy los resultados retornados en la primera posición de Google tienen una mayor relevancia que los de Yahoo y MSN Live Search ya que Google es usado por el 63.5% de los usuarios de Internet, frente al 16.7% de Yahoo o el 10,4% de MSN Live Search [1]. En

este caso es posible que el resultado retornado por Google tenga mayor importancia que el retornado por MSN Live Search debido a que el porcentaje de utilización Google es del 69.6%, frente a 12.4% de Msn Live [65].

Los valores $C_{i,m}$ $\mathrm{C_{i,m}}$ para las tres columnas adicionales a los conceptos están dados por la fórmula (15)

$$C_{i,j} = \left(\frac{LSM_i - RankM_i}{LSM_i - LIM_i} \right) * M_i \qquad (15)$$

En donde LSM_i $\mathrm{LSM_i}$es la cantidad de resultados retornados por el buscador i (límite superior), $RankM_i$ $\mathrm{RankM_i}$ es la posición o ranking del documento j en el motor de búsqueda i. LIM_i $\mathrm{LIM_i}$ es la posición uno de los documentos (límite inferior) y M_i $\mathrm{M_i}$ es el porcentaje de utilización del motor de búsqueda i. Luego de creada la matriz de términos por documentos se pasa al siguiente módulo.

8.7 RE RANKING Y FILTRADO DE INFORMACIÓN

En el módulo de **Re Ranking** se clasifican los documentos que tienen el contenido más apropiado para la búsqueda realizada por una persona. Para lograr esto, se representa en un espacio n-dimensional (Figura 6), tanto la consulta, punto negro, como los documentos retornados por los buscadores, puntos rojos, en donde las n dimensiones están dadas por los n términos de la consulta. Previamente, se ha construido la matriz de términos por documentos.

En la Figura 6 se representa una consulta formada por tres términos y n documentos retornados por los buscadores. Ahora se debe calcular cuáles son los vecinos d (documentos) más cercanos al punto q (consulta), para ordenarlos y presentarlos al usuario. Se usa el concepto de los vecinos más cercanos (similar al algoritmo k-nn) aplicando la distancia de cosenos, debido a que con éste método se obtiene de manera más sencilla los documentos similares en un determinado orden.

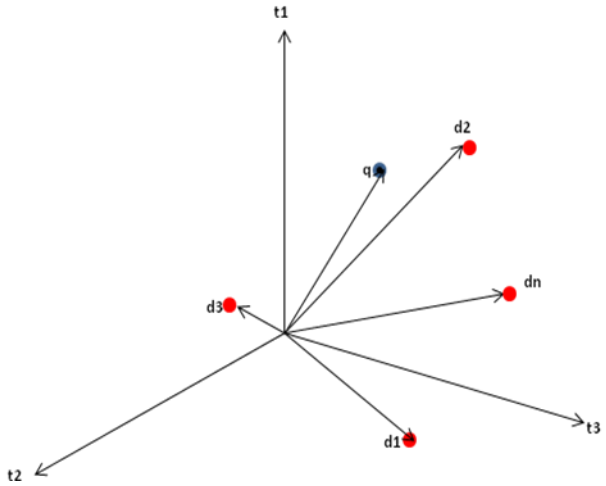


Figura 6 Representación de una consulta y los documentos retornados.

Luego se busca la similitud entre los documentos (vecinos más cercanos) y la consulta, utilizando la función de similitud dada por la ecuación (5) y (6). Cuando el coseno se acerca a cero el documento se aleja de la consulta, es decir, no es similar. Por el contrario cuando el coseno se acerca a uno el documento es similar a la consulta. De forma análoga, cuando $\theta \to 0$ se tiene que el documento es similar a la consulta y si $\theta \to 1$ significa que el contenido del documento retornado no tiene parecido con la búsqueda del usuario. Finalmente, y como se muestra en la figura anterior, sólo se tiene en cuenta una parte del primer cuadrante del espacio n-dimensional, debido a que los valores que se toman están entre cero y uno.

En el módulo de **Filtrado,** teniendo en cuenta estudios que analizan el comportamiento que adoptan los usuarios con respecto a los resultados de los buscadores web tradicionales, se ha encontrado que sólo el 82% de los usuarios hacen click en el primer resultado y la mayoría de ellos solo revisan la primera página de resultados. En nuestro modelo se ha definido un parámetro φ que sirve como filtro a los documentos que realmente se presentan al usuario. Para entender esto es preciso definir la utilidad del documento con base en la similitud y la retroalimentación del usuario previamente calculadas. La utilidad (o relevancia) del documento se calcula con la ecuación (17).

Si la utilidad del documento supera el valor definido en el parámetro φ el documento ingresa a la lista de resultados que se presentan al usuario, de lo contrario no se presenta. Finalmente se visualizan los resultados en orden descendiente según el valor de su utilidad.

Igualmente, para desarrollos futuros se puede emplear a partir de dicha estructura, Singular Value Decomposition SVD, descomponiendo la matriz de términos por documentos en factores ortogonales, para luego emplear LSI, a continuación los datos entregados por LSI se agrupan utilizando algoritmos de clustering: k-means, clúster

jerárquico, frases frecuentes o reglas de asociación, luego se etiquetan para identificarlos claramente y finalmente se muestran los resultados obtenidos.

8.8 VISUALIZACIÓN DE DOCUMENTOS

En este módulo se listan los documentos para que el usuario los visualice, después de que han pasado por el proceso de filtrado y se han escogido los que más se adaptan a las necesidades de búsqueda del usuario. Los resultados se listan en orden de acuerdo a la similitud que presentan con respecto a la consulta realizada.

8.9 FEEDBACK

Luego de realizar el filtro se visualizan los resultados, el usuario tiene la posibilidad de calificar el documento, de esta forma se realiza un feedback sobre los resultados, porque se recalcula el ranking (Nuevo Coseno - $N\cos\theta$ Ncosθ) o ubicación de los documentos según la utilidad que representó para el usuario. El feedback está dado por la ecuación (16).

$$N\cos\theta = \begin{cases} ret < 0, & N\cos\theta = \cos\theta / 2 \\ ret > 0, & N\cos\theta = (1+\cos\theta)/2 \\ ret = 0, & N\cos\theta = \cos\theta \end{cases} \qquad (16)$$

En donde ret es la calificación o retroalimentación que da el usuario, esto significa que el usuario ha determinado que el documento le fue inútil, indiferente o útil mediante los valores de retroalimentación -1, 0 o 1 respectivamente. Finalmente se calcula el porcentaje de utilidad con la fórmula (17).

$$utilidad = N\cos\theta * 100 \qquad (17)$$

9. USO DE CACHÉ LOCAL DE DATOS

La propuesta presentada contiene varios procesos fundamentales que articulan su funcionamiento. El primero de ellos consiste en la selección manual de las categorías específicas de la taxonomía, puede seleccionar sólo una categoría si así lo prefiere, para definir el contexto o área del conocimiento en el que se va a realizar la búsqueda. Debido a que las categorías más específicas de la taxonomía tienen asociadas ontologías, el usuario está seleccionando implícitamente los conceptos que forman esa ontología. Como segundo proceso, se realiza la expansión de la consulta para luego realizar la búsqueda mediante la utilización de las Api´s de los motores de búsqueda Google, Yahoo y MSN Live Search. Se verifica primero si la consulta ya se ha hecho con anterioridad, de ser así, se le da la opción al usuario de realizar nuevamente la búsqueda en Internet o recuperar los resultados anteriores que se encuentran en ***caché.*** En caché se almacenan las url's de búsquedas anteriores y el respectivo feedback realizado por el usuario, de esa forma

se logra mayor velocidad al realizar las búsquedas. Luego de que los buscadores han retornado los resultados a la búsqueda, se procede a realizar el filtrado de los mismos para luego presentarlos al usuario. Los resultados son almacenados con la información de la similitud que éstos han tenido con respecto a la consulta del usuario y éste es el criterio de ranking del modelo propuesto en este trabajo. Finalmente en la visualización de los resultados, el usuario puede interactuar con los documentos para proveer la retroalimentación necesaria que permite determinar si un resultado le ha sido más útil que otro, para realizar esta acción, el usuario tiene la posibilidad de calificar los documentos, de esa manera se re calcula la similitud pero ahora con respecto a la utilidad que el documento le presentó al usuario.

Durante el proceso de búsqueda, el modelo permite conocer las preferencias de búsqueda, en este caso se puede saber sobre qué temas consultó un determinado usuario, las ontologías que utilizó para realizar su búsqueda, los resultados que obtuvo en dicha consulta y la retroalimentación que dio, además se puede determinar cuáles documentos le fueron útiles y cuáles no, de esta manera se puede crear un perfil sobre las preferencias de búsqueda de los usuarios. En la Figura 7 se muestra el funcionamiento del modelo en un diagrama de flujo.

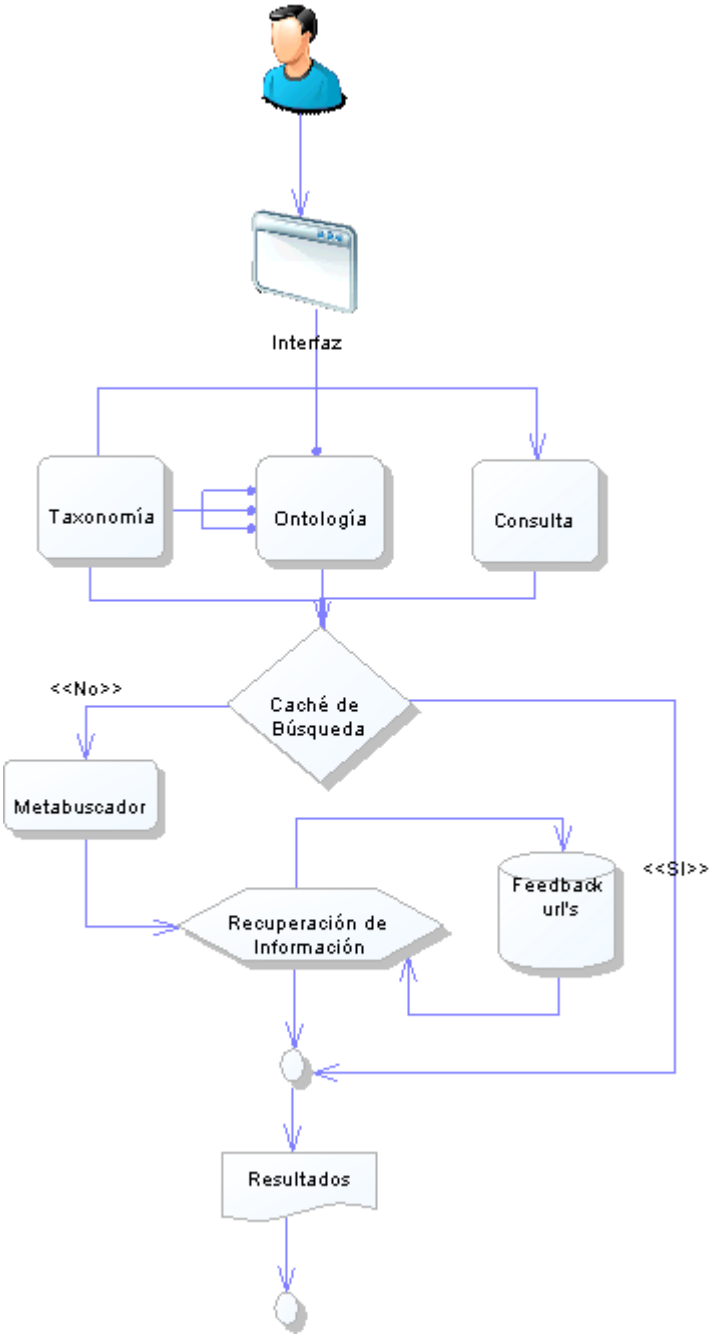


Figura 7 Diagrama de flujo del modelo propuesto

CAPÍTULO IV – DESCRIPCIÓN DEL META BUSCADOR

10. METODOLOGÍA DE DESARROLLO DEL META BUSCADOR

10.1 DESCRIPCIÓN GENERAL DE LA METODOLOGÍA

Para el desarrollo del proyecto se utilizó una instanciación del Proceso Unificado, para ello se tuvieron en cuenta las siguientes fases:

10.1.1 PLANEACIÓN Y ELABORACIÓN

En esta fase se definieron los requerimientos necesarios para el desarrollo del sistema, así como el estudio de las diversas alternativas de conseguir su desarrollo. En esta fase se obtuvieron los siguientes artefactos de manera preliminar. Casos de uso de alto nivel, Modelo conceptual, Diagrama de secuencia, Glosario, Arquitectura.

10.1.2 CONSTRUCCIÓN

Con los resultados obtenidos en la fase de Planeación y Elaboración se dio inicio a esta fase, con ella se logró obtener una versión operativa del sistema en las siguientes etapas.

- **Análisis**: En esta etapa se obtuvo una concepción clara de los requisitos a desarrollar en cada ciclo, se desarrolla la descripción de los casos de uso de alto nivel y el modelo conceptual específico para cada ciclo
- **Diseño**: Teniendo en cuenta el análisis del sistema se procedió a generar una solución lógica del prototipo software que finalmente fue implementado, para lo cual se diseñaron los casos de uso reales, los diagramas de clases de diseño, de secuencia, de paquetes y de despliegue.
- **Implementación**: Se implementaron los componentes lógicos obtenidos en la etapa de diseño.
- **Pruebas**: Se realizaron las validaciones pertinentes para verificar el resultado de la implementación generada en esta fase.

10.1.2.1 CICLOS DE DESARROLLO

Los ciclos de desarrollo permitieron dividir la funcionalidad completa del sistema en tareas más pequeñas que facilitaron la labor de construcción del sistema cumpliendo con cada una de las etapas mencionadas anteriormente.

- Ciclo 1 – Base de datos: En este ciclo se modeló e implementó sobre Microsoft SQL Server Express 2005, la base de datos relacional.
- Ciclo 2 – Acceso a datos: En este ciclo se integró SubSonic como capa de acceso a datos.
- Ciclo 3 – Selección del lenguaje de Ontologías: Se estudiaron las diferentes opciones existentes de lenguajes de ontologías para integrarlo al meta buscador, optando finalmente por OWL. Además se escogió la Api de JENA que permite crear las ontologías en el lenguaje seleccionado.

- Ciclo 4 – Editor de Ontologías: En este ciclo se desarrollaron las funcionalidades básicas de edición, inserción y actualización de ontologías para el prototipo de la herramienta de edición de ontologías.
- Ciclo 5: Meta Buscador. En este ciclo se desarrolló la funcionalidad que permitió la integración de las API's de Google, MsnSearch y Yahoo, para realizar la recuperación de la información.
- Ciclo 6 – Elección de la Taxonomía. En este ciclo se procedió a realizar el estudio de las diferentes taxonomías del conocimiento existentes y se realizó la integración al Meta Buscador de la más adecuada, en este caso la Clasificación Decimal Dewey.
- Ciclo 7 – Expansión de consultas. Se creó la funcionalidad del sistema que permite expandir las consultas. Previamente se adicionaron al sistema varias ontologías en el dominio de la informática para poder tomar sus conceptos y sinónimos con el fin de enriquecer las consultas del usuario.
- Ciclo 8 – Filtrado de información. En este ciclo se integró al sistema la funcionalidad que permite determinar cuáles son los documentos retornados por los buscadores que más parecido tienen con las necesidades de consulta del usuario.
- Ciclo 9 – Visualización de resultados: Se creó el módulo que permite visualizar los resultados ordenados por el meta buscador con varias opciones que muestran la utilidad de los resultados, el enlace a los documentos web recuperados, el snippet y una gráfica que visualiza la utilidad que ha tenido para el usuario en las ocasiones que ha sido visitado.
- Ciclo 10 – Feedback. En este ciclo se desarrolló la funcionalidad que le permite al usuario realizar la retroalimentación sobre los documentos filtrados, de esta manera se puede determinar cuáles son los documentos que más le han servido a un usuario según su propio criterio.

10.1.3 TRANSICIÓN

Luego de culminar la fase de construcción del sistema, se realizaron las respectivas comparaciones entre los resultados que retornan los buscadores de manera independiente y los resultados filtrados por el meta buscador, con el fin de realizar su adecuación final e implantación. Posteriormente se llevaron a cabo pruebas beta con estudiantes con el objetivo de comparar el desempeño del meta buscador.

10.1.4 DOCUMENTACIÓN Y DIVULGACIÓN DE RESULTADOS

A lo largo de cada fase se desarrolló la documentación respectiva. Finalmente la divulgación se termina con la realización de la monografía y la sustentación de la tesis ante los jurados definidos por la Facultad de Ingeniería Electrónica y Telecomunicaciones, FIET. La documentación presentó los resultados obtenidos del uso del meta buscador y las recomendaciones sugeridas.

10.2 ARQUITECTURA DEL SISTEMA

El sistema está desarrollado bajo una arquitectura general de Cliente Inteligente, se puede acceder a ella mediante una aplicación Windows. En la Figura 8 se muestra la arquitectura del sistema y en la Figura 9 se muestran los módulos que representan la funcionalidad general del meta buscador, a continuación se hace una descripción de los diferentes módulos que componen el sistema.

El Módulo de **Administración de Ontologías** ofrece la funcionalidad necesaria para gestionar las ontologías, por ejemplo, importar ontologías existentes, personalizarlas o crearlas. En caso de que un usuario edite una ontología, podrá adicionar algunas características particulares, por ejemplo, puede adicionar a cada concepto o clase de la ontología los sinónimos que considere necesarios, igualmente tiene la opción de adicionar a cada concepto o sinónimo una relevancia o prioridad que varía de uno a cinco, según la importancia del término o concepto. Para este proyecto no se crean ontologías, únicamente se hace uso de ontologías ya creadas y disponibles en la librería de ontologías de DAML [64]. A medida que un usuario edita las ontologías, el sistema permite almacenar un historial de las ontologías.

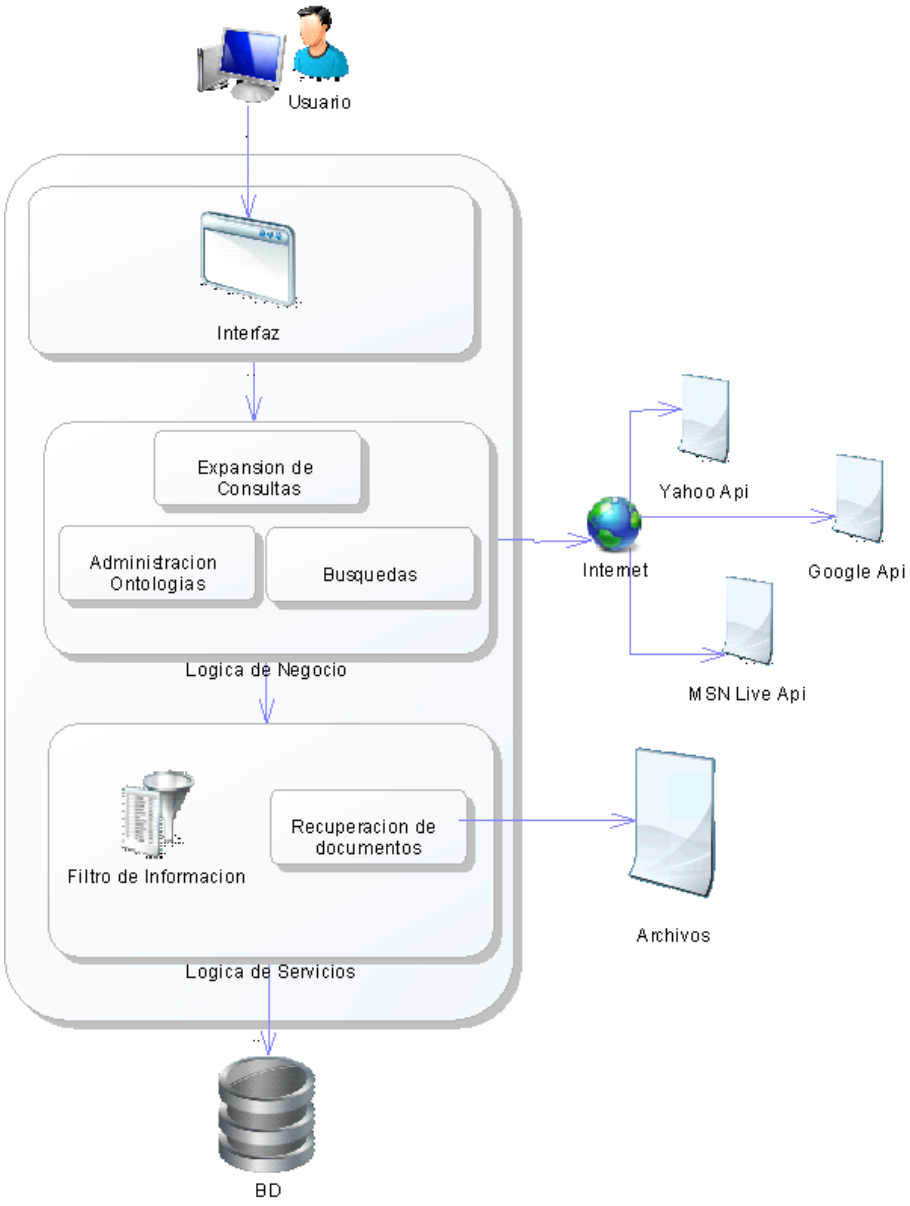


Figura 8 Arquitectura del Sistema.

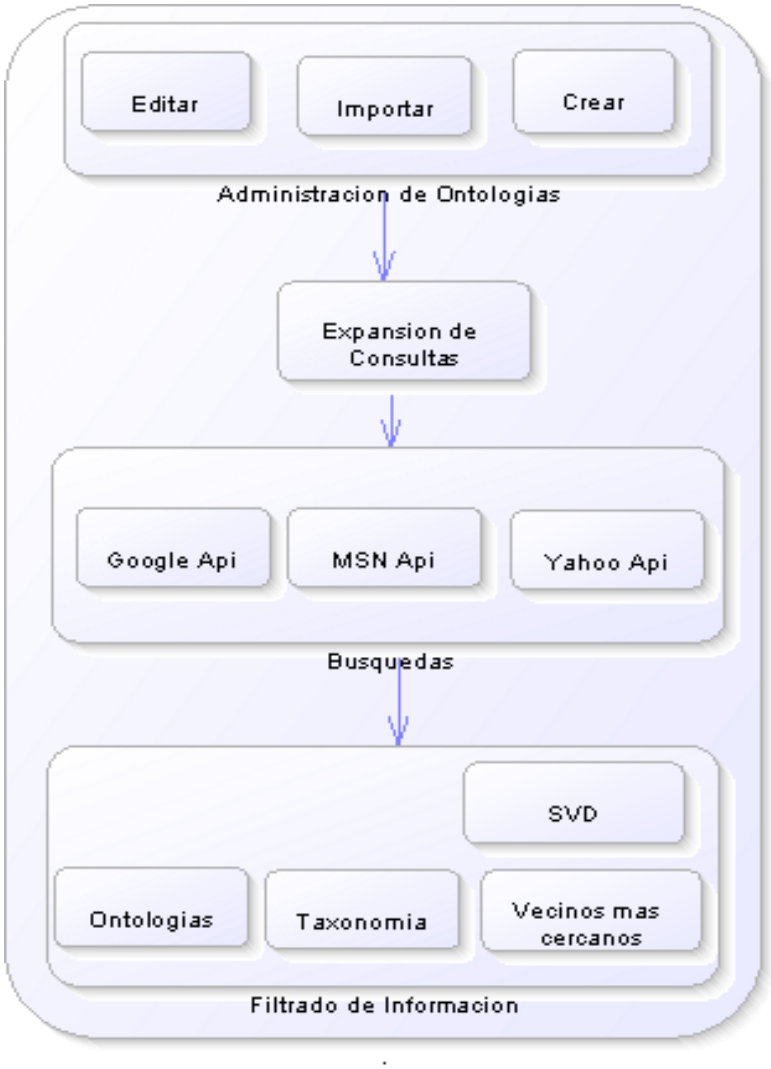


Figura 9. Diagrama de Componentes

En el módulo de **Expansión de Consultas** se toma la consulta del usuario y se enriquece con nuevos términos. Para expandir la consulta, se toma la (s) ontología (s) que el usuario ha seleccionado para realizar la búsqueda, se comparan los términos de la consulta con los conceptos de las ontologías, para cada concepto igual al término de consulta se obtienen los sinónimos y se crea una nueva consulta enriquecida, si no coinciden los conceptos, entonces se obtienen todos los sinónimos de la ontología teniendo en cuenta la relevancia de cada uno y se expande la consulta con esos términos. Finalmente se envía la consulta enriquecida a los buscadores Google, Yahoo y MSN Live.

El módulo de **Búsqueda** integra las Api's que ofrecen Msn Search, Google y Yahoo para hacer el proceso de recuperación de información. Para realizar una búsqueda en Internet, el usuario navega a través de la taxonomía del conocimiento y selecciona la categoría que le interesa. Cada nodo hoja de la taxonomía tiene asociado una ontología, cuando se

hace la búsqueda se toman los conceptos y sinónimos de los conceptos para enriquecer la búsqueda y definir un dominio preciso de búsqueda, así las Api's de los buscadores utilizados, reciben en realidad una consulta expandida.

El módulo de **Filtrado de Información** se encarga de tomar los términos de la consulta enriquecida y los documentos retornados por las Api's de los buscadores, luego forma una matriz de n términos por m documentos. Para determinar cuáles son los documentos que tienen mayor similitud con la consulta del usuario se representa en un espacio n dimensional los documentos y la consulta en forma de vectores, la dimensión del espacio está dada por la cantidad de términos de la consulta. Luego, se calculan cuáles son los documentos más cercanos a la consulta mediante la función de similitud dada por la ecuación (5) y (6) y se muestran en orden, desde el más cercano hasta el más lejano.

En el módulo de **Feedback** se realiza una retroalimentación sobre los documentos presentados por el meta buscador. El proceso consiste en presentarle los documentos al usuario junto con la opción de calificarlos mediante tres variables: no sirvió, sirvió e indiferente, si el usuario abre un documento y se olvida calificarlo, por defecto se toma una calificación indiferente, dejando el mismo valor de similitud con respecto a la consulta. Cuando el usuario decide calificar, se recalcula la similitud mediante las fórmula (16) en donde la nueva similitud es $N\cos\theta$, y ret es la calificación que da el usuario.

Si en ocasiones futuras el usuario realiza la misma búsqueda, puede elegir mirar los resultados retornados la última vez que consultó, de esa manera podrá mirar cuáles resultados le fueron útiles y cuáles no, en esta sección se le presenta al usuario una gráfica que le indica las calificaciones que tuvo el documento cuando fue visitado.

Finalmente el sistema guarda las tendencias de búsqueda del usuario, almacenando las consultas realizadas con anterioridad (historial de búsquedas) y los dominios de búsqueda (sobre qué ontologías realizó determinada búsqueda), con esto se puede determinar qué temas son de interés y los enlaces que ha visitado y cuáles le han sido útiles.

10.3 ANÁLISIS Y DISEÑO

Luego del análisis llevado a cabo para el desarrollo del meta buscador se muestran a continuación algunos resultados generales del sistema.

10.3.1 CASOS DE USO DE ALTO NIVEL

En la Figura 10 se muestran las operaciones que todos los usuarios del sistema pueden realizar, a saber: ingresar a la aplicación, cambiar la contraseña y editar el perfil (preferenias de búsqueda, e información).

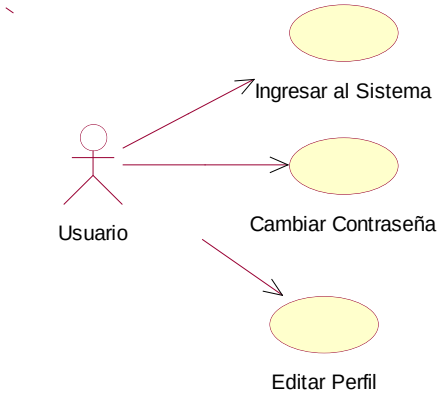


Figura 10. Casos de Uso comunes para los usuarios.

En la Figura 11 se aprecian los casos de uso para el usuario experto, quien se encarga de las tareas de administración de ontologías, como edición y eliminación. También puede crear ontologías y ponerlas a disposición de los usuarios.

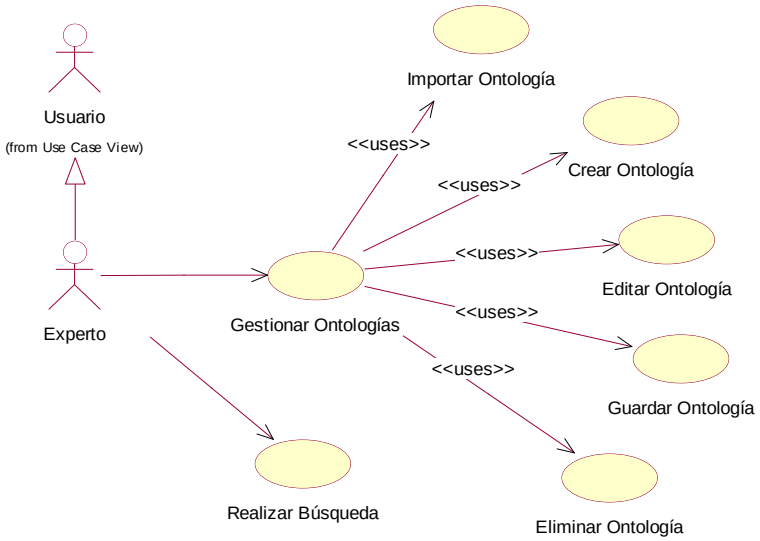


Figura 11. Casos de Uso para el usuario Experto.

En la Figura 12 se muestra el actor Cliente, quien realiza consultas de temas específicos en Internet, para ello debe escribir las palabras clave que describen su consulta y seleccionar uno o varios tópicos de la taxonomía general del conocimiento. El usuario puede gestionar su historial de consultas, borrar alguna búsqueda en particular o todas, además gestiona sus ontologías, las personaliza según sus necesidades, las edita o simplemente las consulta. Los cambios que un cliente realiza en las ontologías, sólo los

observa él. Además, las ontologías originales sólo las puede cambiar el usuario experto y los cambios se distribuyen para todos los clientes.

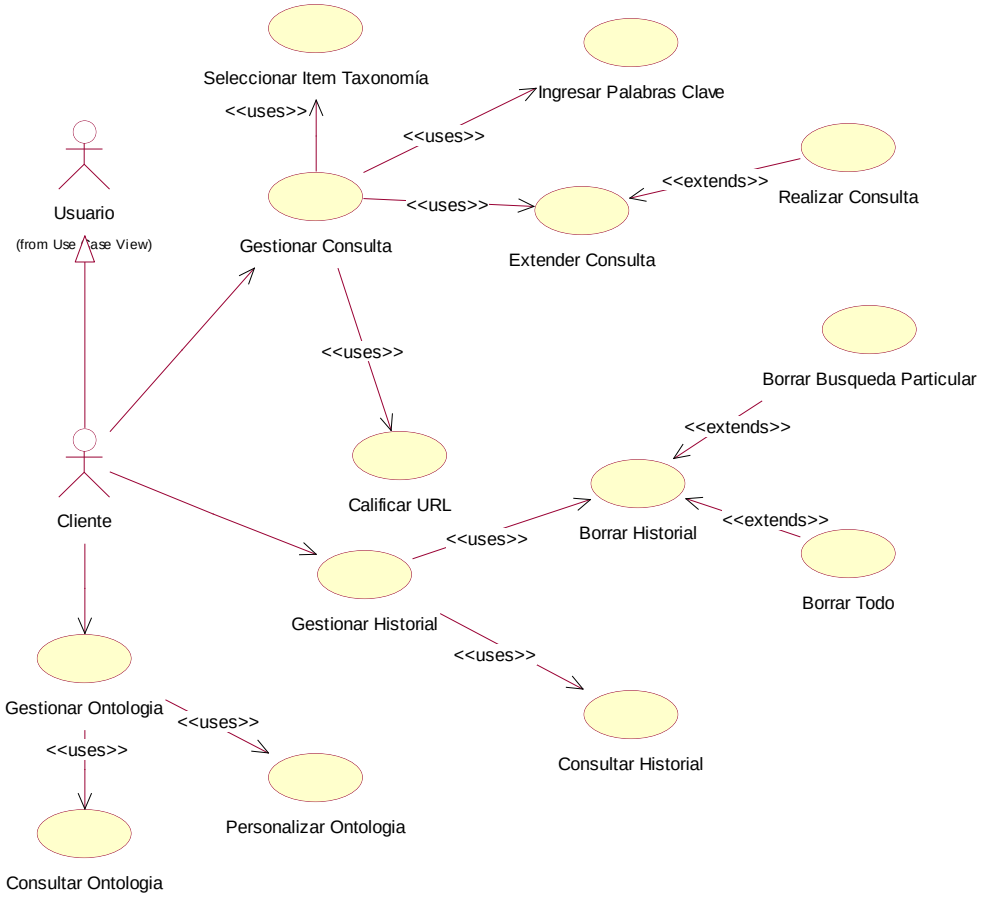


Figura 12 Casos de Uso para el Cliente del sistema.

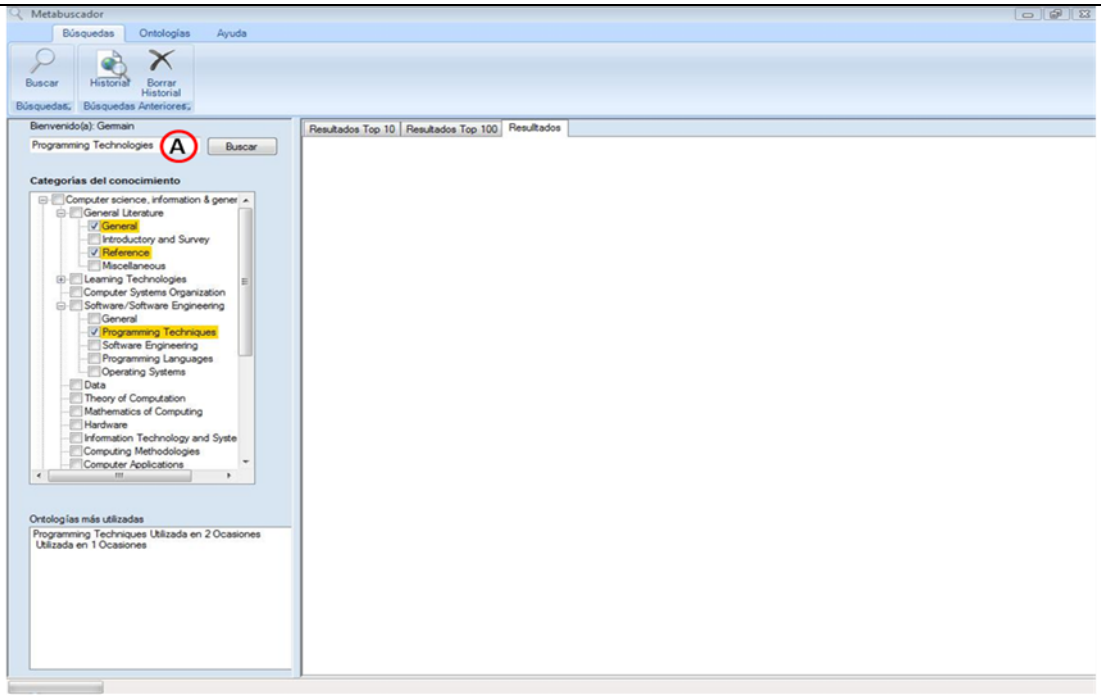
10.3.2 CASOS DE USO REALES

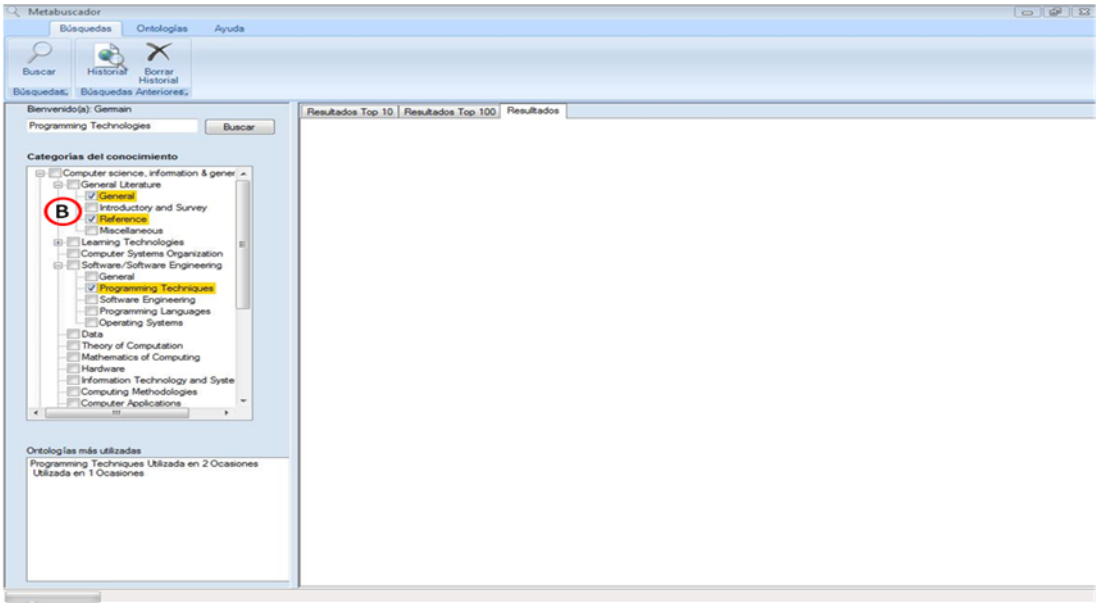
A continuación se muestran tres de los casos de uso reales más importantes del sistema, los demás casos de uso se encuentran en el anexo A.

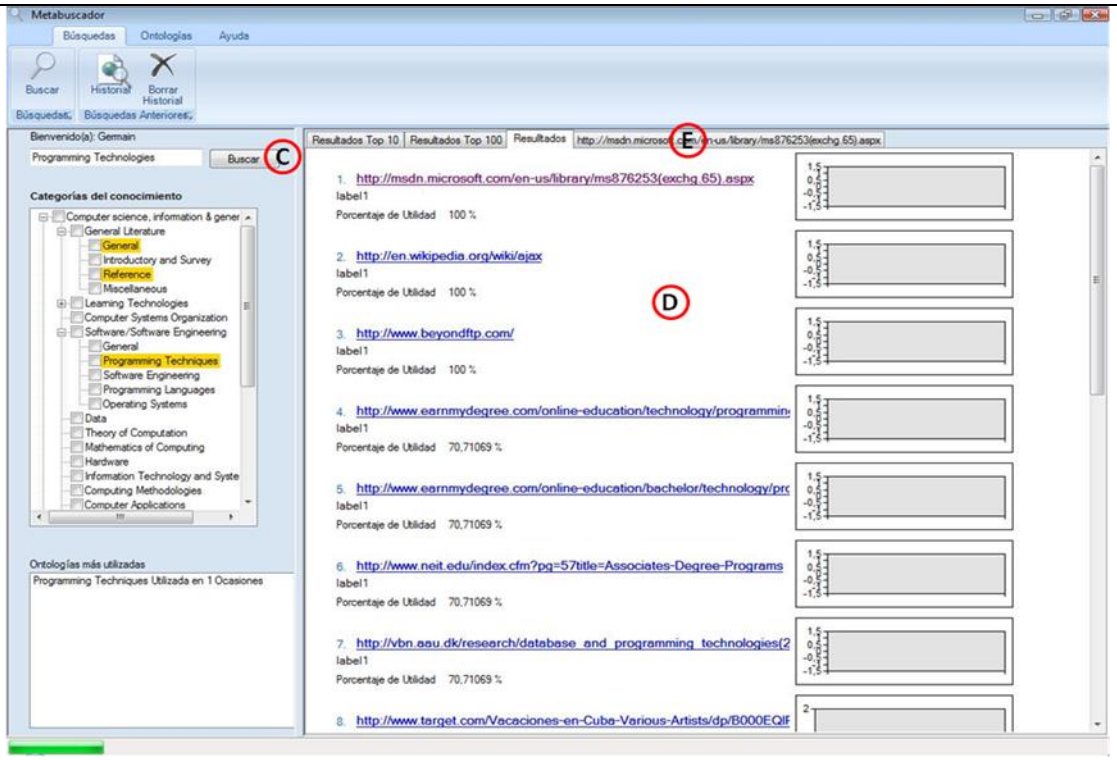
CASO DE USO REAL: GESTIONAR CONSULTA
Actores: Cliente.
Propósito: Gestionar las consultas con respecto a una temática específica.
Resumen: El cliente realiza una consulta digitando las palabras clave, navegando a

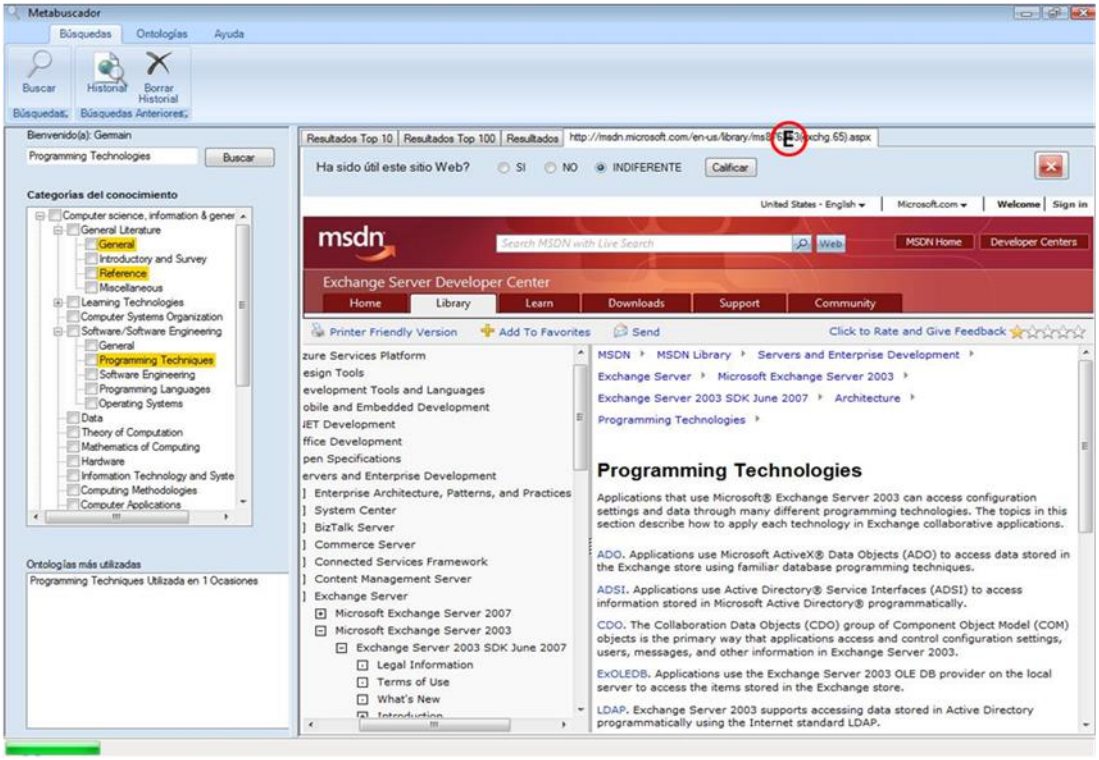
través de la taxonomía general del conocimiento y seleccionando una o varias ontologías.

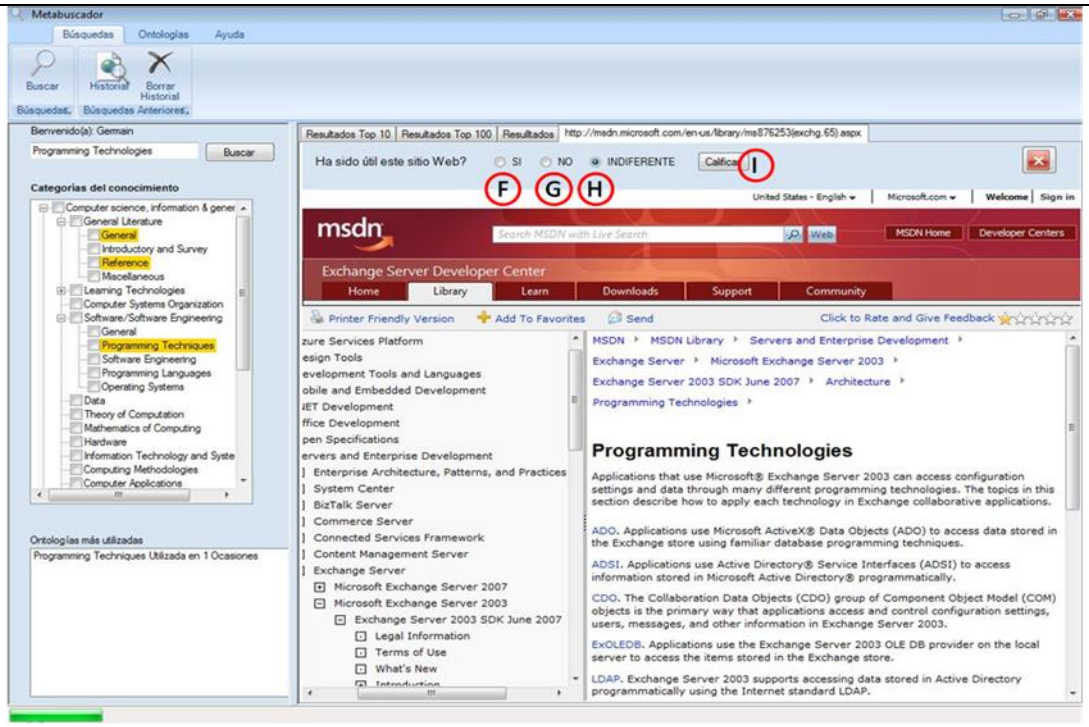
Tipo: Primario

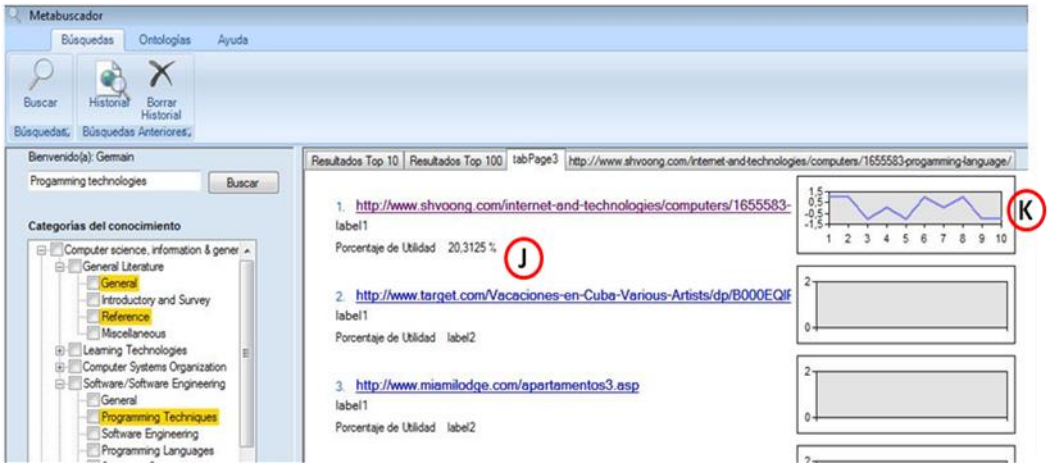





Metabuscador
Búsquedas
Ontologías
Ayuda
Buscar
Historial
Borrar Historial
Búsquedas:
Búsquedas Anteriores:
Bienvenido(a): Germain
Programming Technologies
Buscar
C
Categorías del conocimiento
Computer science, information & gener
General Literature
General
Introductory and Survey
Reference
Miscellaneous
Learning Technologies
Computer Systems Organization
Software/Software Engineering
General
Programming Techniques
Software Engineering
Programming Languages
Operating Systems
Data
Theory of Computation
Mathematics of Computing
Hardware
Information Technology and Syste
Computing Methodologies
Computer Applications
Ontologías más utilizadas
Programming Techniques Utilizada en 1 Ocasiones
Resultados Top 10
Resultados Top 100
Resultados
E
1. http://msdn.microsoft.com/en-us/library/ms876253(exchg.65).aspx
label1
Porcentaje de Utilidad 100 %
2. http://en.wikipedia.org/wiki/ajax
label1
Porcentaje de Utilidad 100 %
D
3. http://www.beyondftp.com/
label1
Porcentaje de Utilidad 100 %
4. http://www.earnmydegree.com/online-education/technology/programmin
label1
Porcentaje de Utilidad 70,71069 %
5. http://www.earnmydegree.com/online-education/bachelor/technology/prc
label1
Porcentaje de Utilidad 70,71069 %
6. http://www.neit.edu/index.cfm?pg=57title=Associates-Degree-Programs
label1
Porcentaje de Utilidad 70,71069 %
7. http://vbn.aau.dk/research/database_and_programming_technologies(2
label1
Porcentaje de Utilidad 70,71069 %
8. http://www.target.com/Vacaciones-en-Cuba-Various-Artists/dp/B000EQIF


Metabuscador
Búsquedas
Ontologías
Ayuda
Buscar
Historial
Borrar Historial
Búsquedas:
Búsquedas Anteriores:
Bienvenido(a): Germain
Programming Technologies
Buscar
Categorías del conocimiento
Ontologías más utilizadas
Programming Techniques Utilizada en 1 Ocasiones
Resultados Top 10
Resultados Top 100
Resultados
E
Ha sido útil este sitio Web?
SI
NO
INDIFERENTE
Calificar
msdn
Search MSDN with Live Search
Web
MSDN Home
Developer Centers
Exchange Server Developer Center
Home
Library
Learn
Downloads
Support
Community
Printer Friendly Version
Add To Favorites
Send
Click to Rate and Give Feedback
MSDN
MSDN Library
Servers and Enterprise Development
Exchange Server
Microsoft Exchange Server 2003
Exchange Server 2003 SDK June 2007
Architecture
Programming Technologies
Programming Technologies
Applications that use Microsoft® Exchange Server 2003 can access configuration settings and data through many different programming technologies. The topics in this section describe how to apply each technology in Exchange collaborative applications.
ADO. Applications use Microsoft ActiveX® Data Objects (ADO) to access data stored in the Exchange store using familiar database programming techniques.
ADSI. Applications use Active Directory® Service Interfaces (ADSI) to access information stored in Microsoft Active Directory® programmatically.
CDO. The Collaboration Data Objects (CDO) group of Component Object Model (COM) objects is the primary way that applications access and control configuration settings, users, messages, and other information in Exchange Server 2003.
ExOLEDB. Applications use the Exchange Server 2003 OLE DB provider on the local server to access the items stored in the Exchange store.
LDAP. Exchange Server 2003 supports accessing data stored in Active Directory programmatically using the Internet standard LDAP.
Microsoft Exchange Server 2007
Microsoft Exchange Server 2003
Exchange Server 2003 SDK June 2007
Legal Information
Terms of Use
What's New



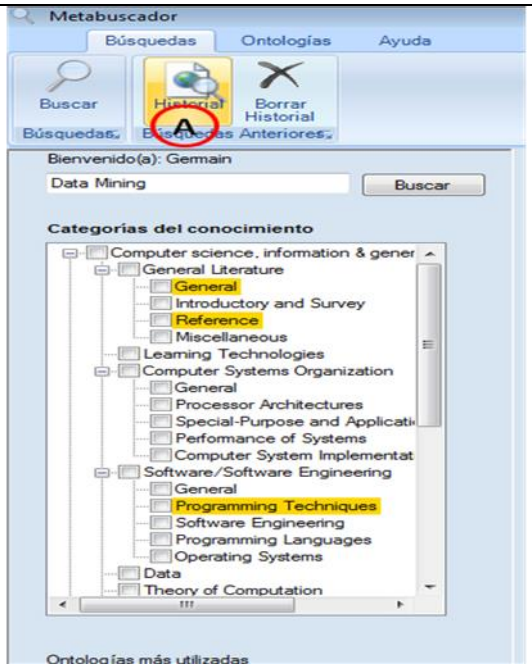


CURSO NORMAL DE LOS EVENTOS	
Acción del actor	**Respuesta del sistema**
1. El usuario (Cliente) digita las palabras clave en el cuadro de texto [A].	
2. El usuario da clic (selecciona) en uno o varios nodos del árbol (Taxonomía del Conocimiento, preferiblemente los que están subrayados) [B].	

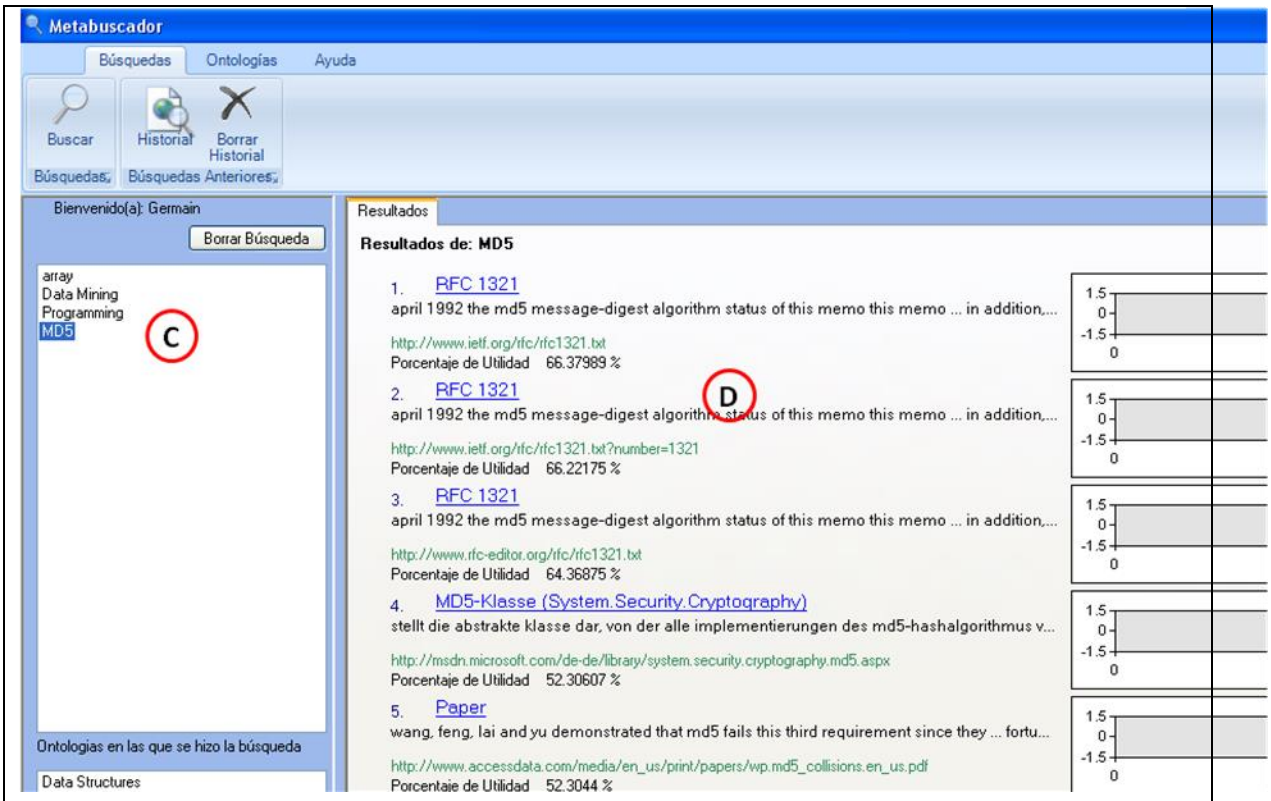
3. El usuario da clic en el botón buscar [C].	4. El sistema realiza la búsqueda en Internet en segundo plano.
	5. El sistema realiza el filtrado (utilizando los términos de la ontología asociada a la rama de la taxonomía y sus sinónimos) de los resultados retornados por los buscadores consultados.
	6. El sistema muestra los enlaces a los documentos filtrados [D].
7. El usuario da clic en los enlaces de su interés [D].	8. El sistema visualiza las páginas que el usuario ha elegido [E].
9. Después de revisar la página el usuario califica el sitio [F] [G] [H] [I].	10. El sistema guarda la calificación, actualiza el nivel de utilidad y muestra en un gráfico la calificación del sito. [J] [K]

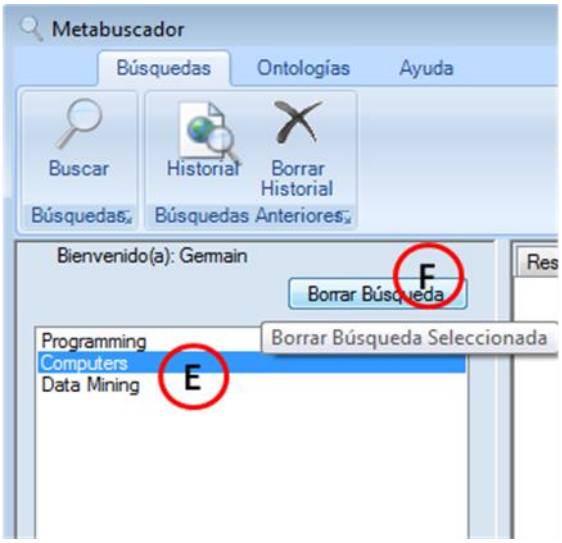
Tabla 1 Caso de Uso Real Gestionar Consulta.

CASO DE USO REAL: GESTIONAR HISTORIAL
Actores: Cliente.
Propósito: Gestionar el historial de búsquedas, en este caso el cliente puede consultar sus búsquedas anteriores o borrarlas.
Resumen: El Cliente tiene la opción de revisar cuáles han sido las búsquedas que ha realizado con anterioridad, igualmente puede manipular su historial, borrando una búsqueda en particular o borrando todo su historial de consultas.
Tipo: Primario

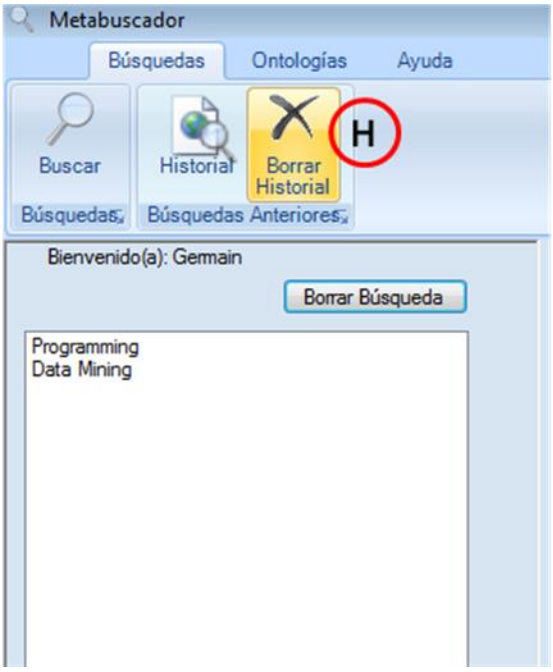

Metabuscador
Búsquedas
Ontologías
Ayuda
Buscar
Historial
Borrar Historial
A
Búsquedas
Búsquedas Anteriores
Bienvenido(a): Germain
Data Mining
Buscar
Categorías del conocimiento
Computer science, information & gener
General Literature
General
Introductory and Survey
Reference
Miscellaneous
Learning Technologies
Computer Systems Organization
General
Processor Architectures
Special-Purpose and Applicati
Performance of Systems
Computer System Implementat
Software/Software Engineering
General
Programming Techniques
Software Engineering
Programming Languages
Operating Systems
Data
Theory of Computation
Ontologías más utilizadas

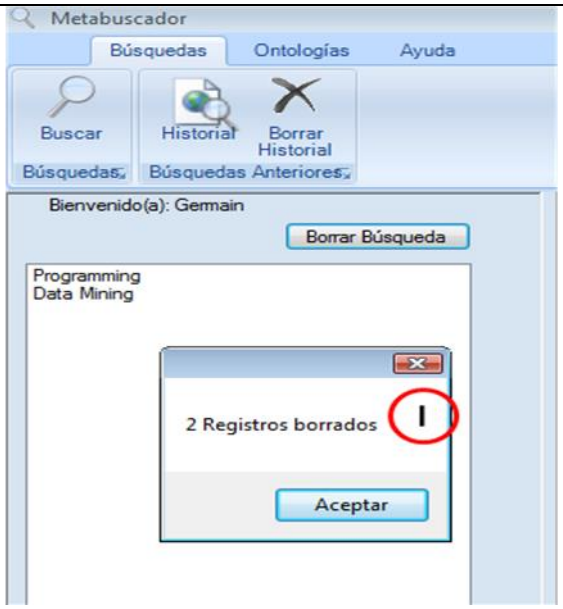

Metabuscador
Búsquedas
Ontologías
Ayu
Buscar
Historial
Borrar Historial
Búsquedas
Búsquedas Anteriores
Bienvenido(a): Germain
Borrar Búsqueda
array
Data Mining
Programming
MD5
B
Ontologias en las que se hizo la búsqueda
Data Structures

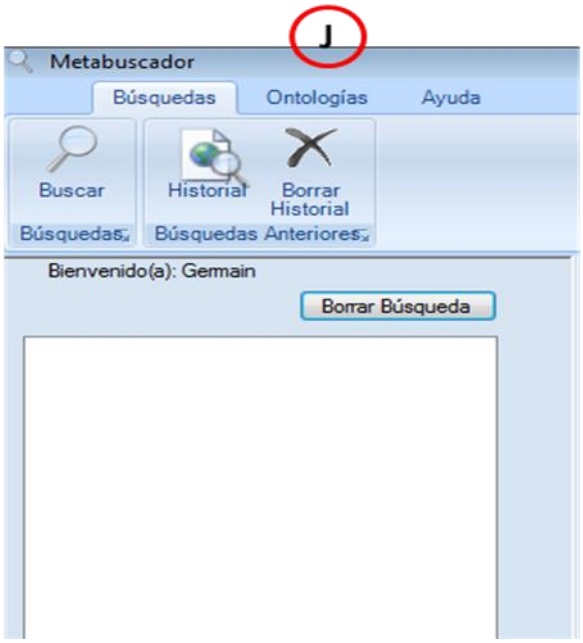

Metabuscador
Búsquedas
Ontologías
Ayuda
Buscar
Historial
Borrar Historial
Búsquedas
Búsquedas Anteriores
Bienvenido(a): Germain
Borrar Búsqueda
array
Data Mining
Programming
MD5
C
Ontologías en las que se hizo la búsqueda
Data Structures
Resultados
Resultados de: MD5
1. RFC 1321
april 1992 the md5 message-digest algorithm status of this memo this memo ... in addition,...
http://www.ietf.org/rfc/rfc1321.txt
Porcentaje de Utilidad 66.37989 %
2. RFC 1321
D
april 1992 the md5 message-digest algorithm status of this memo this memo ... in addition,...
http://www.ietf.org/rfc/rfc1321.txt?number=1321
Porcentaje de Utilidad 66.22175 %
3. RFC 1321
april 1992 the md5 message-digest algorithm status of this memo this memo ... in addition,...
http://www.rfc-editor.org/rfc/rfc1321.txt
Porcentaje de Utilidad 64.36875 %
4. MD5-Klasse (System.Security.Cryptography)
stellt die abstrakte klasse dar, von der alle implementierungen des md5-hashalgorithmus v...
http://msdn.microsoft.com/de-de/library/system.security.cryptography.md5.aspx
Porcentaje de Utilidad 52.30607 %
5. Paper
wang, feng, lai and yu demonstrated that md5 fails this third requirement since they ... fortu...
http://www.accessdata.com/media/en_us/print/papers/wp.md5_collisions.en_us.pdf
Porcentaje de Utilidad 52.3044 %
1.5
0
-1.5
0


Metabuscador
Búsquedas
Ontologías
Ayuda
Buscar
Historial
Borrar Historial
Búsquedas
Búsquedas Anteriores
Bienvenido(a): Germain
F
Borrar Búsqueda
Borrar Búsqueda Seleccionada
Programming
Computers
Data Mining
E


Metabuscador
Búsquedas
Ontologías
Ayuda
Buscar
Historial
Borrar Historial
Búsquedas
Búsquedas Anteriores
Bienvenido(a): Germain
Borrar Búsqueda
Programming
Data Mining
G


Metabuscador
Búsquedas
Ontologías
Ayuda
Buscar
Historial
Borrar Historial
H
Búsquedas
Búsquedas Anteriores
Bienvenido(a): Germain
Borrar Búsqueda
Programming
Data Mining



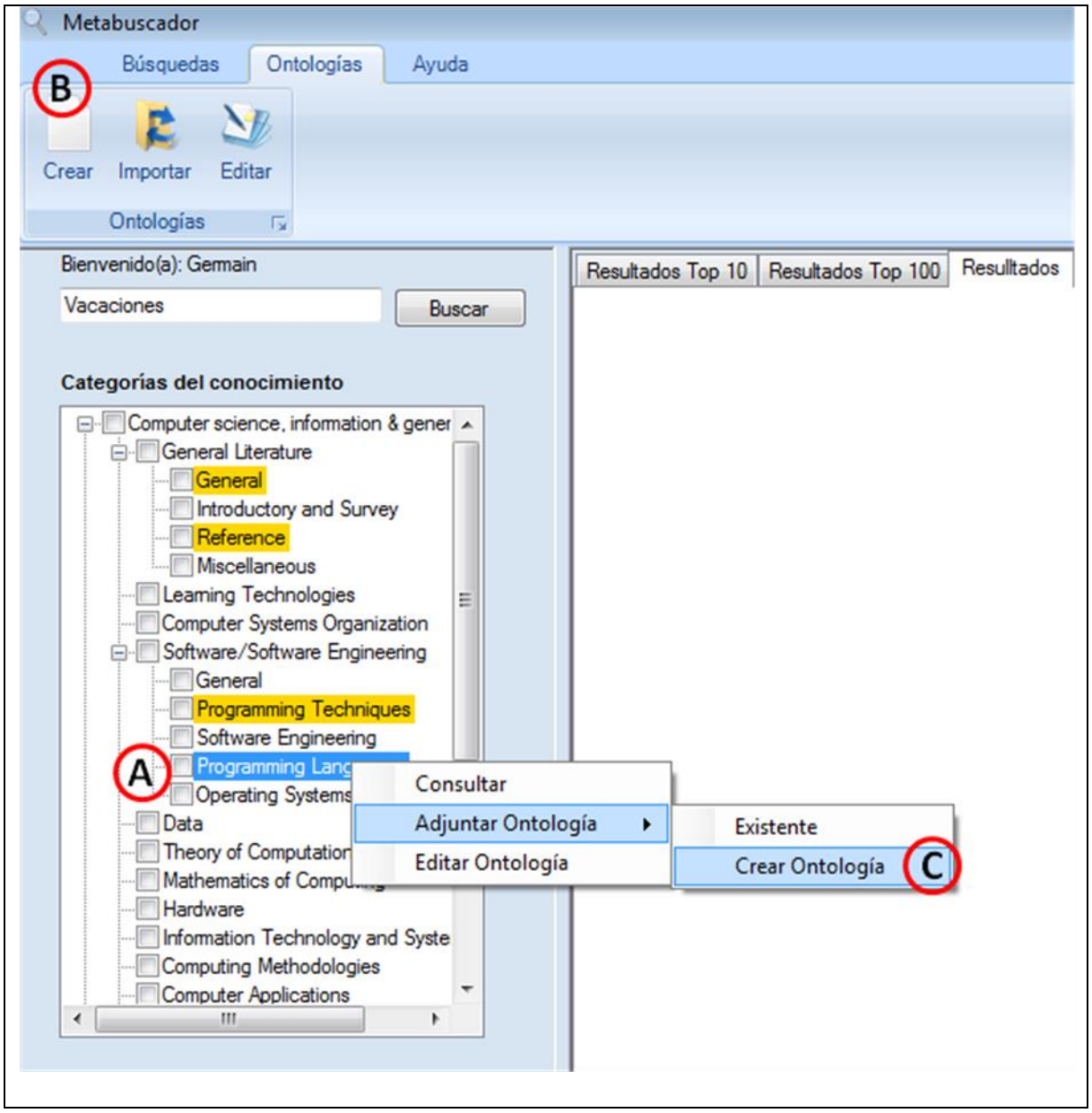


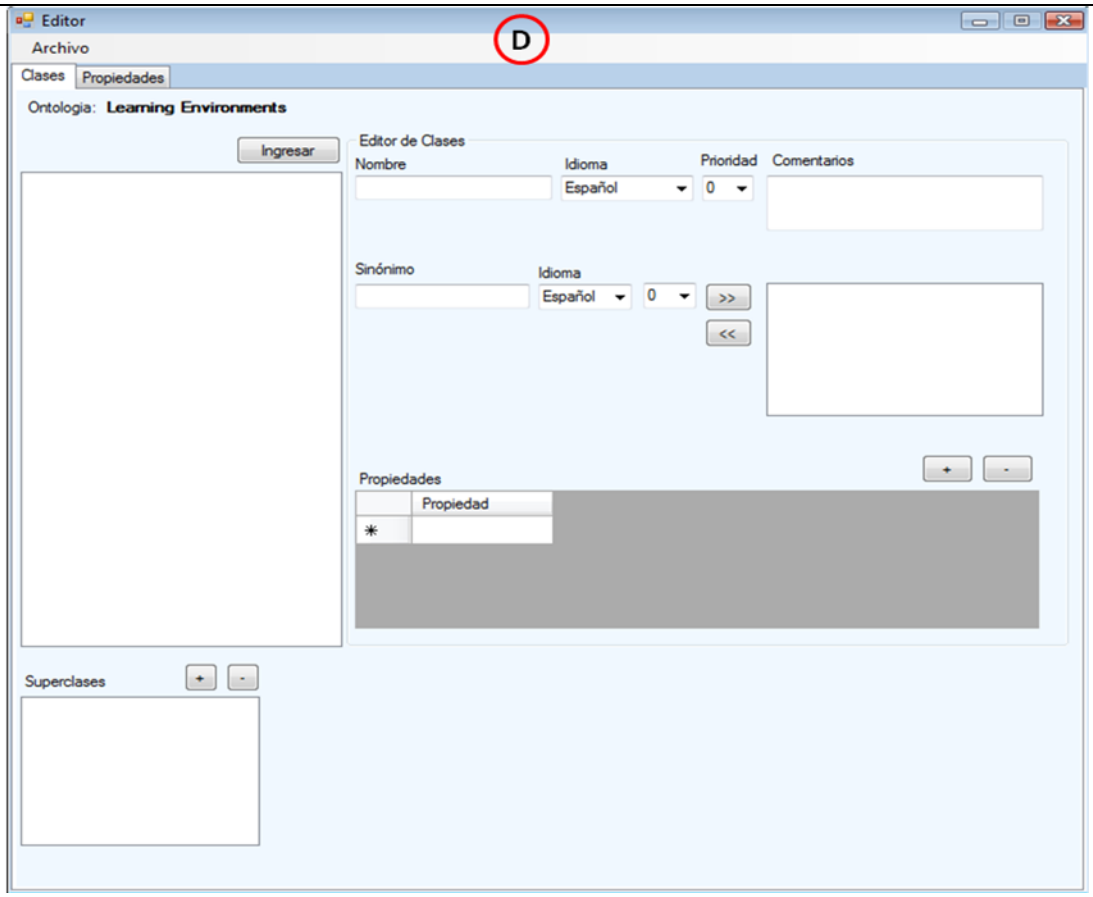
CURSO NORMAL DE LOS EVENTOS	
Acción del actor	**Respuesta del sistema**
1. El usuario hace click en el botón historial [A].	2. El sistema muestra una lista con las búsquedas realizadas por el usuario en ocasiones anteriores [B].
3. El usuario selecciona una de las	4. El sistema despliega la ventana con los

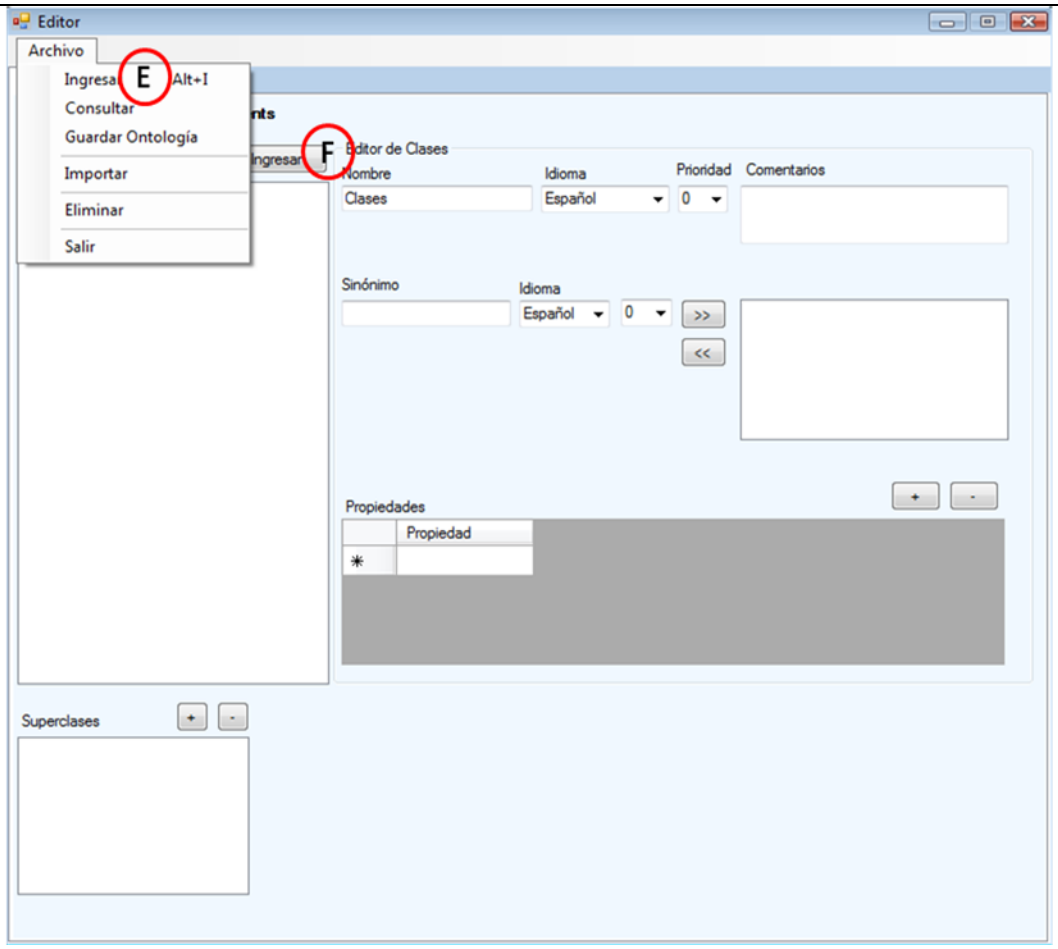
búsquedas [C].	enlaces que retornó la búsqueda anterior [D].
5. El usuario selecciona una de las búsquedas [E] y luego hace click en el botón borrar búsqueda [F].	6. El sistema elimina la búsqueda y sus resultados y los quita de la lista [G].
7. El usuario da click en el botón borrar historial [H].	8. El sistema elimina todo el historial de búsquedas y muestra en un cuadro de diálogo la cantidad de búsquedas que eliminó [I].
	9. El sistema elimina de la lista todas las búsquedas [J].

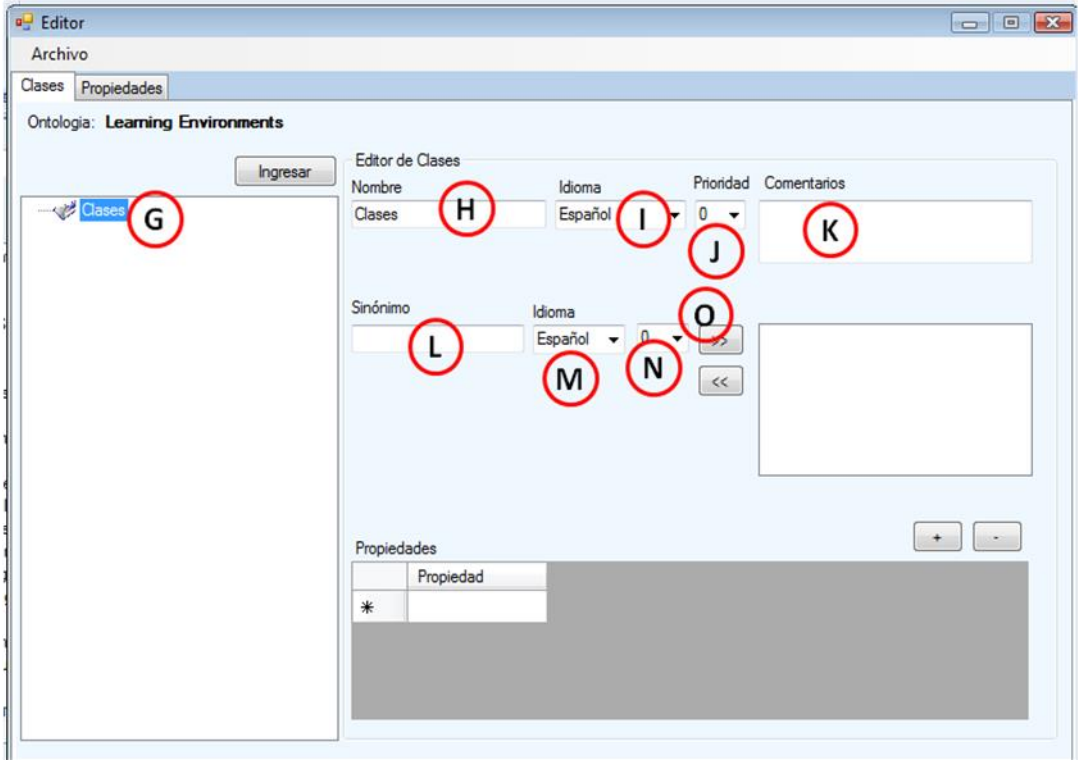
Tabla 2 Caso de Uso Real Gestionar Historial.

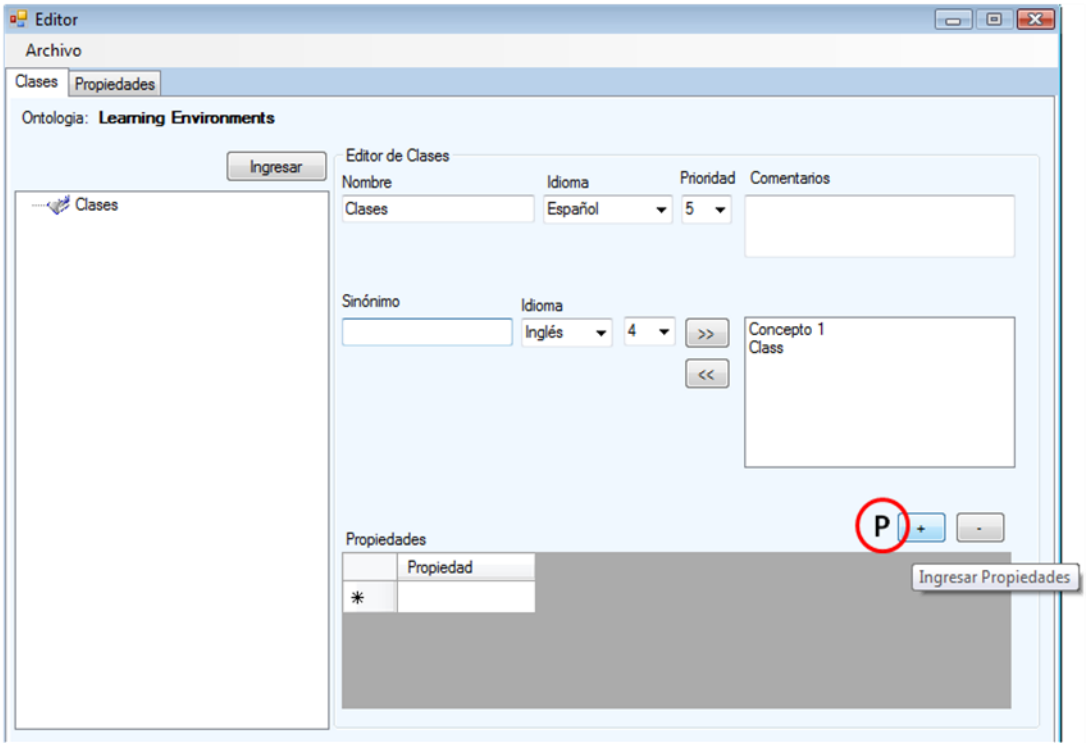
CASO DE USO REAL: CREAR ONTOLOGÍAS
Actores: Usuario Experto.
Propósito: Crear ontologías.
Resumen: El usuario experto puede crear los conceptos que forman la ontología, las propiedades de cada concepto y finalmente guardar la ontología.
Tipo: Primario

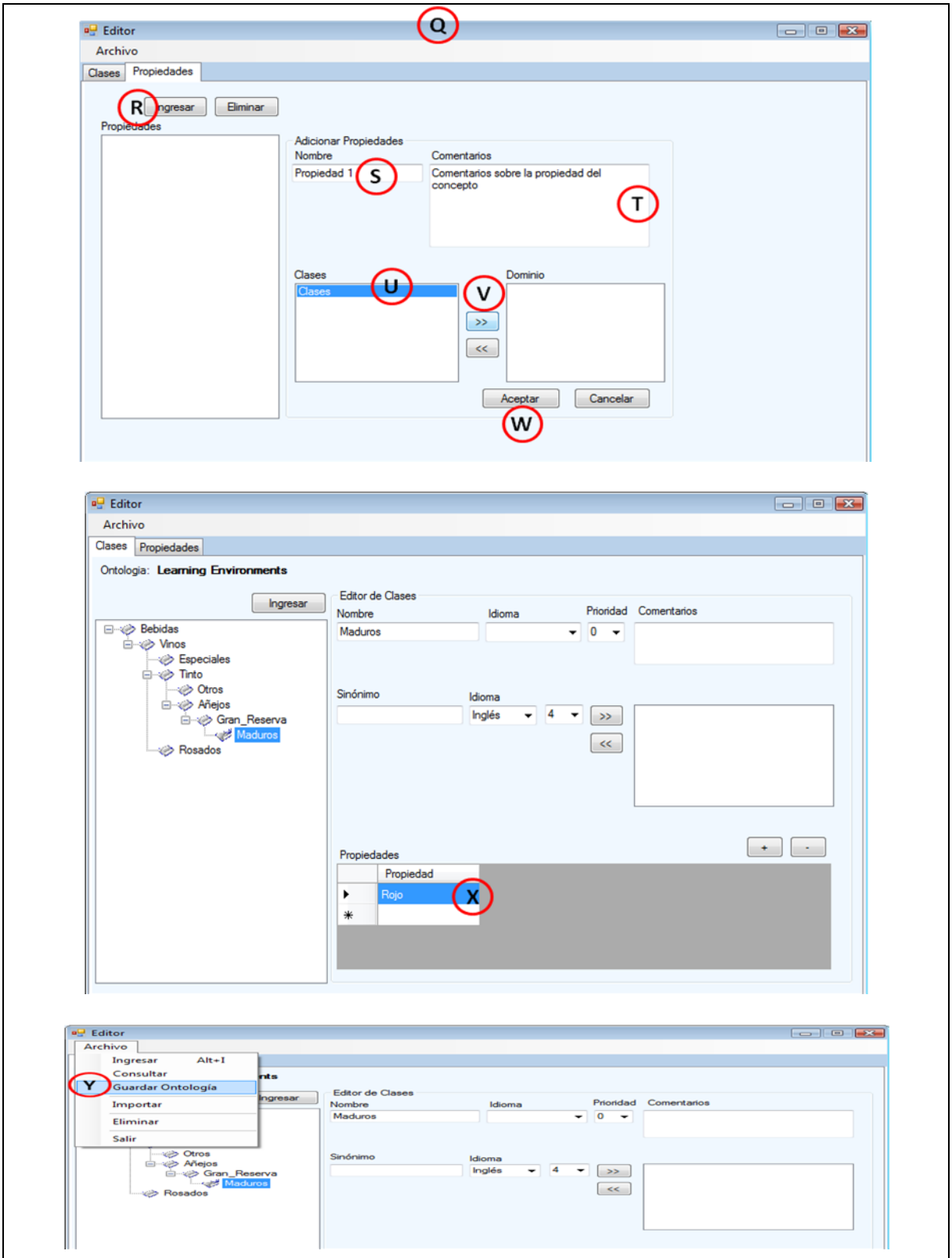

Metabuscador
Búsquedas
Ontologías
Ayuda
B
Crear
Importar
Editar
Ontologías
Bienvenido(a): Germain
Vacaciones
Buscar
Categorías del conocimiento
Computer science, information & gener
General Literature
General
Introductory and Survey
Reference
Miscellaneous
Learning Technologies
Computer Systems Organization
Software/Software Engineering
General
Programming Techniques
Software Engineering
A
Programming Lang
Operating Systems
Data
Theory of Computation
Mathematics of Compu
Hardware
Information Technology and Syste
Computing Methodologies
Computer Applications
Consultar
Adjuntar Ontología
Editar Ontología
Existente
Crear Ontología
C
Resultados Top 10
Resultados Top 100
Resultados


Editor
D
Archivo
Clases
Propiedades
Ontologia: Learning Environments
Ingresar
Editor de Clases
Nombre
Idioma
Español
Prioridad
0
Comentarios
Sinónimo
Idioma
Español
0
>>
<<
Propiedades
+
-
Propiedad
Superclases
+
-


Editor
Archivo
Ingresar Alt+I
Consultar
Guardar Ontología
Importar
Eliminar
Salir
E
F
Editor de Clases
Nombre
Clases
Idioma
Español
Prioridad
0
Comentarios
Sinónimo
Idioma
Español
0
>>
<<
Propiedades
Propiedad
Superclases


Editor
Archivo
Clases
Propiedades
Ontologia: Learning Environments
Ingresar
Editor de Clases
Nombre
Idioma
Prioridad
Comentarios
Clases
Español
0
Sinónimo
Idioma
Español
0
>>
<<
Propiedades
Propiedad
+
-
G
H
I
J
K
L
M
N
O


Editor
Archivo
Clases
Propiedades
Ontologia: Learning Environments
Ingresar
Editor de Clases
Nombre
Idioma
Prioridad
Comentarios
Clases
Español
5
Sinónimo
Idioma
Inglés
4
>>
<<
Concepto 1
Class
Propiedades
Propiedad
+
-
P
Ingresar Propiedades


Editor
Archivo
Clases
Propiedades
R
Ingresar
Eliminar
Propiedades
Adicionar Propiedades
Nombre
Propiedad 1
S
Comentarios
Comentarios sobre la propiedad del concepto
T
Clases
U
Clases
V
>>
<<
Dominio
Aceptar
Cancelar
W
Q
Editor
Archivo
Clases
Propiedades
Ontologia: Learning Environments
Ingresar
Editor de Clases
Nombre
Maduros
Idioma
Prioridad
0
Comentarios
Bebidas
Vinos
Especiales
Tinto
Otros
Añejos
Gran_Reserva
Maduros
Rosados
Sinónimo
Idioma
Inglés
4
>>
<<
Propiedades
Propiedad
Rojo
X
+
-
Editor
Archivo
Ingresar Alt+I
Consultar
Y
Guardar Ontología
Importar
Eliminar
Salir
Editor de Clases
Nombre
Maduros
Idioma
Prioridad
0
Comentarios
Sinónimo
Idioma
Inglés
4
>>
<<
Otros
Añejos
Gran_Reserva
Maduros
Rosados

CURSO NORMAL DE LOS EVENTOS	
Acción del actor	**Respuesta del sistema**
1. El usuario hace clic en uno de los nodos de la taxonomía del conocimiento en el cual desea adicionar la Ontología [A].	
2. El usuario hace click en el menú principal, pestaña Ontologías, opción Crear o en el submenú (haciendo click derecho sobre el nodo de la taxonomía seleccionado) [B] [C].	3. El sistema muestra la ventana que le permite crear la ontología al usuario [D].
4. El usuario Hace click en Ingresar, para adicionar un concepto o clase [E] [F].	5. El sistema adiciona un nodo concepto para que el usuario edite su contenido [G].
6. El usuario escribe el nombre del concepto [H], elige su idioma y prioridad [I] [J] y adiciona comentarios [K].	
7. El usuario agrega uno o varios sinónimos del concepto en un idioma específico y con la prioridad conveniente [L] [M] [N].	
8. El usuario da click en el botón agregar sinónimo [O].	9. El sistema enlaza los sinónimos al concepto.
10. El usuario da clic en el botón agregar propiedades [P].	11. El sistema muestra una ventana para adicionar las propiedades al concepto [Q].
12. El Usuario da click en Ingresar Propiedad [R].	13. El sistema crea la propiedad y da opciones para que el usuario la edite.
14. El usuario edita el nombre de la propiedad [S], el comentario sobre la misma [T] y el dominio al cual pertenece [U] y luego da click en el botón agregar Dominio [V].	
15. El usuario da clic en el botón Aceptar [W].	16. El sistema agrega al concepto la propiedad [X].
17. El Usuario da click en la opción	18. El sistema guarda la ontología en la rama

guardar ontología [Y].	de la taxonomía seleccionada.

Tabla 3. Caso de Uso Real Crear Ontología.

10.3.3 DIAGRAMA DE CLASES

En la Figura 13 se muestra de manera general el diagrama de clases del sistema, en el Anexo B se ilustran con más detalle cada una de las Clases. Más adelante, en la Tabla 4, se describe la funcionalidad de cada Clase.

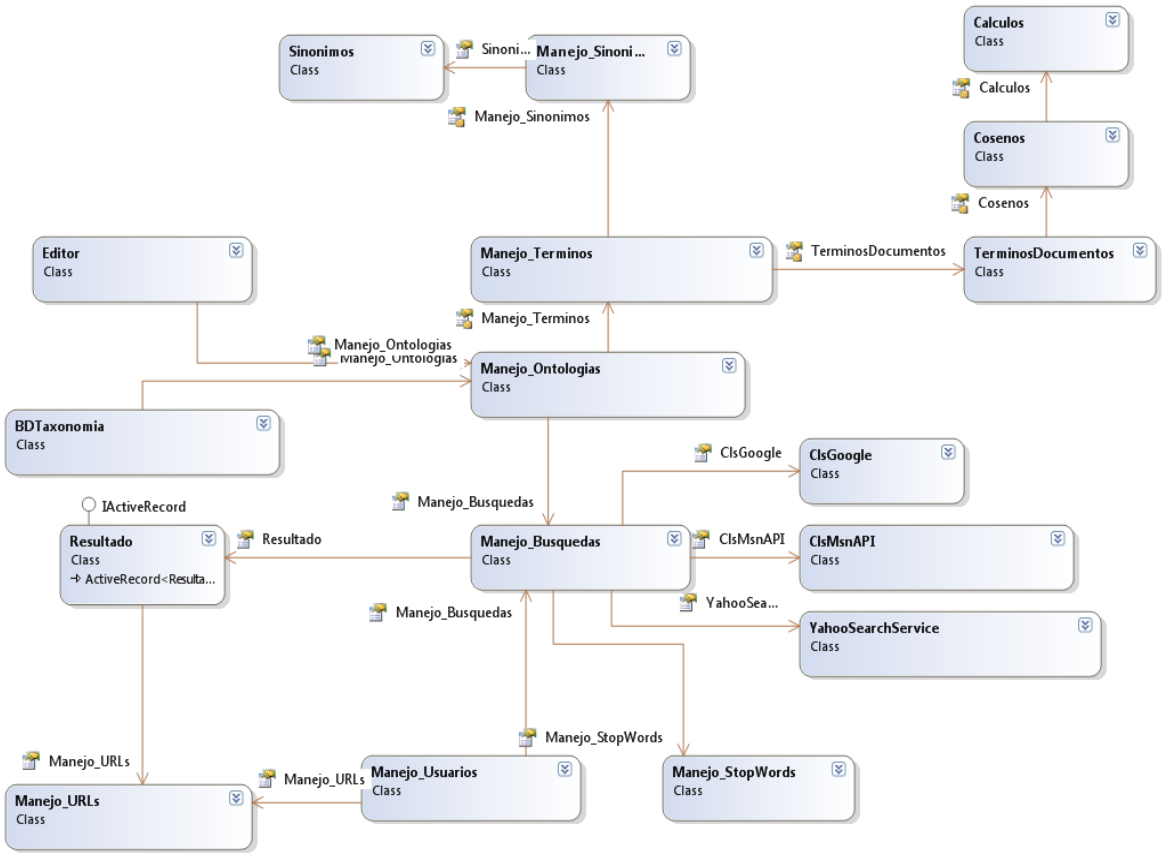


Figura 13 Diagrama general de Clases.

CLASE	FUNCIÓN
BDTaxonomia	En esta Clase se encuentran las funciones que proveen acceso a la taxonomía general del conocimiento.
Editor	Provee las interfaces para la gestión de las ontologías.
Manejo_Ontologias	Permite realizar las diferentes tareas para la gestión de las ontologías.
Manejo_Terminos	Permite la administración de los términos de las ontologías.

CLASE	FUNCIÓN
Manejo_Sinonimos	Provee las funciones para gestionar los sinónimos de los términos de las ontologías.
Idioma	Permite gestionar los diferentes idiomas en los que se pueden escribir los términos de las ontologías.
Manejo_Usuarios	Permite gestionar los actores del sistema.
Manejo_Busquedas	Permite gestionar las búsquedas que realizan los usuarios.
Resultados	Provee las funciones para gestionar los resultados retornados por el sistema.
TerminosDocumentos	Provee las funciones para gestionar los términos presentes en un documento en particular.
Cosenos	Provee el algoritmo que se aplica sobe los documentos para el cálculo de la similitud de los documentos con respecto a la consulta.
Manejo_URLs	Permite la gestión de las URL's retornadas por las búsquedas realizadas.
ClsGoogle	Provee las funciones para acceder a los servicios de búsqueda de Google.
ClsMsnApi	Provee las funciones para acceder a los servicios de búsqueda de Msn Search.
YahooSearchService	Provee las funciones para acceder a los servicios de búsqueda de Yahoo Search.
Manejo_StopWords	Permite realizar las operaciones de preprocesamiento de la consulta y procesamiento de los documentos para eliminar las palabras vacías o stop words.

Tabla 4 Descripción de las Clases.

10.3.4 MODELO DE LA BASE DE DATOS

A continuación en la gráfica se muestra el modelo de la base de datos del sistema y enseguida se explica la finalidad de cada tabla. Debido a la limitación para visualizar el diagrama de la base de datos, en esta sección sólo se muestran los atributos correspondientes a las llaves foráneas y primarias (ver Figura 14), en el Anexo B se ilustran de manera clara y detallada todos los atributos de las tablas.

TABLA	FUNCIÓN
Taxonomia	Contiene la Información de la taxonomía del conocimiento.

TABLA	FUNCIÓN
Ontologias	Contiene información referente a las ontologías. El archivo que contiene la ontología en formato owl, es almacenado en un campo texto (campo ONT_OWL)
Personalizacion de Ontologias	Almacena la información de las ontologías editadas por los usuarios. En el campo PER_XML se guarda la ontología personalizada por los usuarios.
Terminos	Contiene los términos o conceptos de las ontologías.
Sinonimos	Contiene los sinónimos de los términos de las ontologías. Esto es, contiene palabras con el mismo significado de los términos.
Idiomas	Contiene información referente al idioma de la ontología, los términos y los sinónimos.
Historia de las Ontologias	Contiene los históricos de las ontologías, es decir, lleva un registro en los que se va guardando las versiones que han sido modificadas, por ejemplo, un usuario modifica una ontología tres veces, esta tabla contiene un campo en el que se almacenan los cambios a los que ha sido sometida la ontología, en este caso guardaría tres versiones de la ontología.
Permisos en Ontologias	Contiene información referente a los permisos que tienen los usuarios para manipular las ontologías. Cuando alguien quiere editar una ontología, primero se valida el rol del usuario y luego, según los permisos que tenga puede modificarla y esos cambios se reflejan en los demás usuarios o solo los visualiza él.
Usuarios	Contiene información de los usuarios del sistema.
Busquedas	Contiene información referente a las búsquedas que realizan los usuarios, entre otras guarda información sobre la ontología que el usuario utilizó para su consulta.
Resultados	Contiene la información de los resultados arrojados por el meta buscador, en este caso se almacenan las direcciones de los documentos o páginas retornados.
URLs	Contiene información de las URLs de los resultados arrojados por el sistema.
URLs Visitadas	Contiene información sobre las URLs que el usuario visita.
Frecuencias	Contiene la frecuencia con la que aparecen de los términos de las ontologías en un resultado arrojado por la búsqueda.
StopWords	Almacena palabras vacías o que no tienen significado en una consulta.

Tabla 5 Descripción de las Tablas de la Base de Datos.

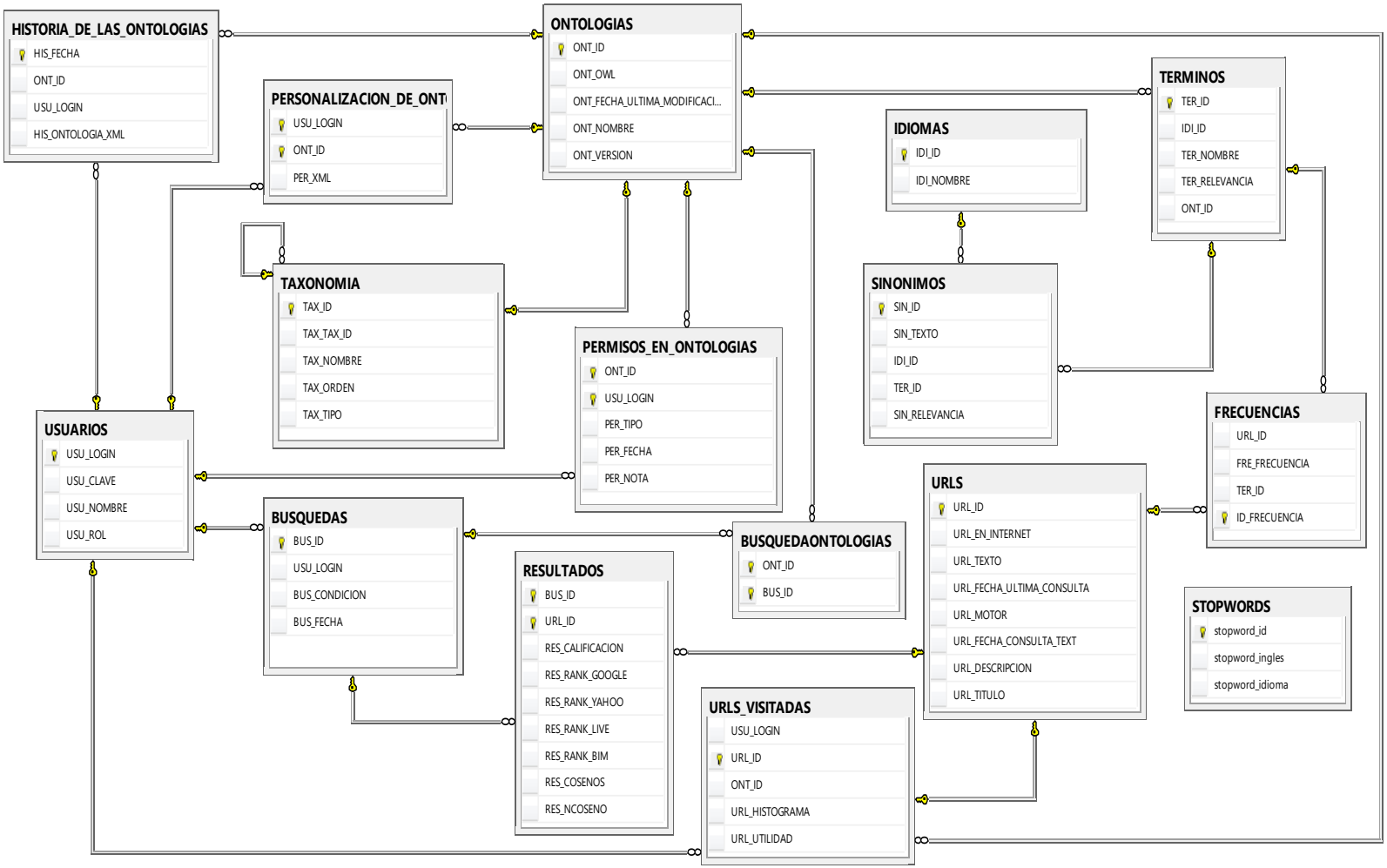


Figura 14 Modelo de la Base de Datos del Sistema.

10.4 IMPLEMENTACIÓN

La aplicación está dividida en varios módulos que describen la funcionalidad del sistema, a continuación se explican en detalle.

10.4.1 ADMINISTRACIÓN DE ONTOLOGÍAS

En este módulo se desarrollan las funcionalidades que permiten interactuar con las ontologías. En este proyecto se hace uso de ontologías desarrolladas en el lenguaje OWL, principalmente porque es el lenguaje más avanzado y completo que actualmente existe (ver Anexo E). Las funcionalidades que se encuentran en este módulo son las siguientes:

- Importar Ontologías. Permite adicionar a la base de datos una ontología existente. Por ahora el sistema sólo está pensado para aceptar ontologías en OWL. Las ontologías importadas son agregadas a una rama específica de la taxonomía del conocimiento.
- Personalizar Ontologías. Da la opción a los usuarios de editar las ontologías según las necesidades de cada uno de ellos. Cuando un cliente personaliza la ontología original queda intacta y se crea una ontología personalizada o modificada que solo la visualiza quien la cambia. Adicionalmente el usuario experto puede modificar la ontología y los cambios se reflejan para todos los usuarios del sistema.
- Crear Ontologías. Permite a los usuarios crear ontologías, para ello se hace uso de Jena (ver Anexo A) que provee una API de ontologías con soporte para OWL, DAML y RDF Schema[15].

10.4.2 BÚSQUEDA

En este módulo se tiene la funcionalidad básica para realizar el proceso de recuperación de información, para ello se hace uso de las API's de Google, Yahoo y Msn Search. Aquí el usuario digita las palabras clave que desea buscar en Internet, después de enriquecer la consulta, las API's de los buscadores mencionados se encargan de recuperar la información que posteriormente es tratada por el meta buscador.

10.4.3 EXPANSIÓN DE CONSULTAS

Para que las consultas del usuario sean delimitadas en un dominio del conocimiento específico, se enriquece la consulta tomando las palabras que ha digitado en el sistema y se comparan con los conceptos de la ontología que seleccionó. Para cada término que coincide con los conceptos de la ontología se obtienen todos los sinónimos y se agregan a la cadena de consulta, así se amplía el rango de consulta para un tema específico. De esta manera, si el usuario desea buscar en el dominio de la "Informática", en la ontología de "Programación Estructurada" el tema de "Stack", el sistema saca los sinónimos del concepto correspondiente a "Stack" y los adiciona a la consulta original. La consulta

[15] http://jena.sourceforge.net/

expandida se envía a los buscadores con el operador lógico OR para que trate de recuperar todos los resultados que coincidan con alguno de los términos de la cadena de búsqueda, así los buscadores reciben en este caso la cadena: “stack OR pila OR lifo”.

10.4.4 FILTRADO DE INFORMACIÓN

Este es el módulo más importante, debido a que en él se combinan las tecnologías que hacen que el meta buscador se diferencie de los buscadores y meta buscadores existentes y convierten la solución planteada en una solución novedosa.

En esta sección son obtenidos los documentos que los buscadores devuelven como respuesta a la consulta del usuario, se construye la matriz de términos (términos de la ontología) por documentos con el fin de calcular la similitud de los resultados con la consulta realizada, para lo cual se utiliza la fórmula 14, detallada en el ítem sobre la arquitectura del sistema.

10.4.5 FEEDBACK

Una vez se visualizan los resultados, el usuario puede calificarlos ver Figura 15 en el visualizador de las páginas, al hacer esto, se re calcula la similitud, acercando el documento a la consulta si la calificación fue si, alejándola si fue no y dejándola igual si no la calificó, lo cual significa que el resultado le fue indiferente. El usuario puede hacer seguimiento de la retroalimentación realizada sobre los resultados observando la gráfica al lado derecho del enlace en la sección de resultados, ver Figura 16. Igualmente el usuario puede observar el porcentaje de utilidad que tiene el resultado.

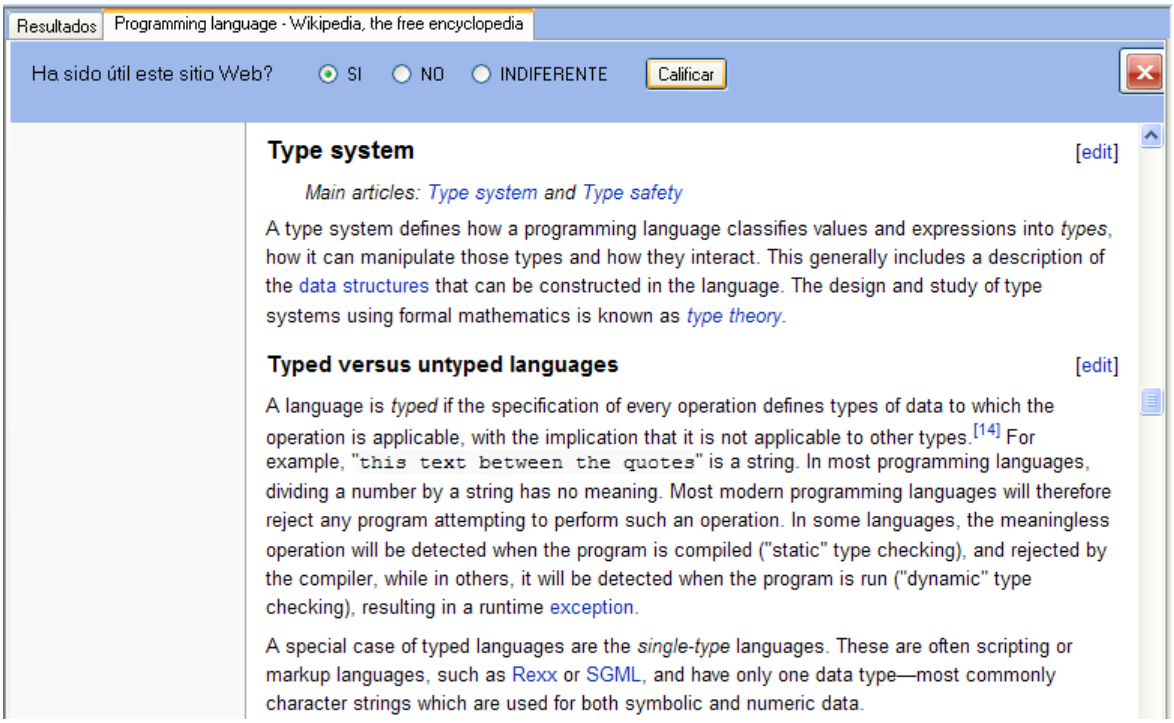


Figura 15 Visualizador de las páginas.

Figura 16 Visualización de la retroalimentación de los resultados.

11. PROBLEMAS Y SOLUCIONES

Durante el desarrollo del proyecto se presentaron varios inconvenientes, a continuación se mencionan los más relevantes.

- Integración de un editor de ontologías. En uno de los objetivos específicos del proyecto se planteó dar la posibilidad de usar ontologías, para ello era necesario integrar un editor de ontologías que permitiera gestionarlas. Debido a la complejidad de realizar esta tarea, no sólo en cuestiones tecnológicas, sino también porque muchos de los editores son privados y no es permitido su uso sin una previa licencia, se tomó la decisión de generar un prototipo que permitiera realizar la gestión de ontologías e incluirlo como un módulo del proyecto, para ello se utilizó un framework para Java que provee una Api para construir Ontologías OWL (JENA) y que puede usarse desde Visual Studio .Net a través de una máquina virtual de Java que se incorpora en las aplicaciones de VS.NET denominada IKVM.NET.

- Algoritmo para crear clusters (grupos temáticos). El algoritmo que se tomó inicialmente para realizar la agrupación de temas *K-means,* aunque es uno de los más usados y referenciados en la literatura científica tiene varias desventajas: 1) Requiere prefijar el número inicial de clusters (*k*), 2) La inicialización de los centroides es aleatoria y por lo tanto no son repetibles de una a otra corrida a pesar de que los datos no cambien, 3) Normalmente busca óptimos locales y 4) Es sensible al ruido (valores anómalos). En pruebas iniciales se tomo SVD para definir el valor de K, y ejecutar varias veces el k-means seleccionado el resultado (centroides) que minimizaran el error cuadrado. Después de estas pruebas se decidió realizar la primera implementación de la herramienta utilizando los vecinos más cercanos (Nearest Neighbor), en el que básicamente se clasifican los resultados que tienen más se parecen a la consulta del usuario. Para calcular los vecinos más cercanos se representan los documentos en forma de vectores de n dimensiones, en donde las dimensiones del espacio vectorial las determina la cantidad de términos de la consulta del usuario. Para formar los vectores de documentos se calculan los términos (con sus respectivas relevancias) de la consulta presentes en los documentos y la relevancia de los mismos. Finalmente se utiliza la distancia de cosenos para calcular y ordenar los documentos que tienen más similitud con respecto a la consulta del usuario. Este proceso se describe en detalle en el capítulo tres.
- La realización de las pruebas automatizadas no se logró debido a que se debía interactuar con el usuario en la calificación de los resultados retornados por el meta buscador BIM, para determinar cuáles de ellos le eran relevantes o no. En su lugar, se realizó la captura y almacenamiento de los resultados retornados por los tres buscadores tradicionales (Google, Yahoo y Live) y nuestro meta buscador BIM. Además, se captura la calificación del usuario con respecto a dichos resultados para lograr analizar la relevancia en el procesamiento de la información lograda por el meta buscador propuesto en comparación con los resultados de los buscadores tradicionales.

CAPÍTULO V – COMPARACIÓN DE RESULTADOS

Para el proceso de comparación de resultados se realizaron pruebas alfa y beta con estudiantes del Servicio Nacional de Aprendizaje (SENA), que permitieron verificar que tan significativos fueron los resultados para los usuarios y también se logró obtener una retroalimentación general con respecto al meta buscador.

Para la realización de estas actividades se procuró que los estudiantes realizaran las búsquedas en temáticas específicas haciendo uso de las ontologías disponibles en el sistema hasta el momento. Las actividades realizadas se detallan a continuación.

12. COMPARACIÓN DE RESULTADOS

12.1 PRUEBAS ALFA

Para realizar la comparación entre los resultados del meta buscador y los buscadores tradicionales se realizó un taller con estudiantes del SENA seccional Cauca. El taller consistió en una serie de preguntas sobre temáticas relacionadas con el curso que estudian, en este caso *Tecnología en Análisis y Desarrollo de Sistemas de Información.* Esta prueba se llevó a cabo en las instalaciones del SENA y se contó con 14 alumnos, 14 computadores y el tiempo de duración fue de 2 horas.

A cada alumno se le asignó el respectivo taller y fue ubicado en un computador. Antes de iniciar la prueba se les explicó en qué consistía y cómo funcionaba la herramienta. En esta introducción se utilizó aproximadamente 10 minutos y a lo largo de la actividad se resolvieron dudas respecto al uso del meta buscador. Los alumnos debieron realizar la revisión de los primeros 10 resultados, esta decisión fue tomada teniendo en cuenta un estudio sobre el comportamiento de los usuarios que consultan información en internet utilizando un buscador realizado por la empresa iProspect [66], donde se muestra que cerca del 27% de los usuarios consultan solo algunos resultados de la primera página y el 41% revisan la primera página de resultados, dando un total de 68% de usuarios consultantes que solo revisan la primera página. Igualmente se les solicitó a los estudiantes que después de visualizar los resultados calificaran el documento, con el fin de tener la posibilidad (en el momento de comparar resultados) de determinar cuáles le fueron útiles y en qué posiciones los retornó cada buscador de manera independiente. Igualmente la herramienta no tuvo la opción de Borrar Historial habilitada para poder obtener todos los resultados disponibles para realizar la comparación respectiva.

Los datos de los estudiantes y las preguntas del taller realizadas para esta prueba se pueden ver en el Anexo I.

Como resultado de esta prueba, se obtuvieron las base de datos del meta buscador en donde se encuentran las consultas, los resultados sin expansión de consulta, los resultados con expansión de consulta, el orden de cada url en cada motor, la posición de

la url en BIM y la posición de la URL en BIM después de usar retroalimentación (Figura 17). Esta sistematización de las pruebas permitió verificar las consultas que realizaron los estudiantes (ver Anexo I) y finalmente, poder afirmar que los resultados devueltos por el meta buscador propuesto obtienen en la mayoría de ocasiones mejores o iguales resultados que los buscadores tradicionales, en este caso Google, Yahoo y MSN Live Search.

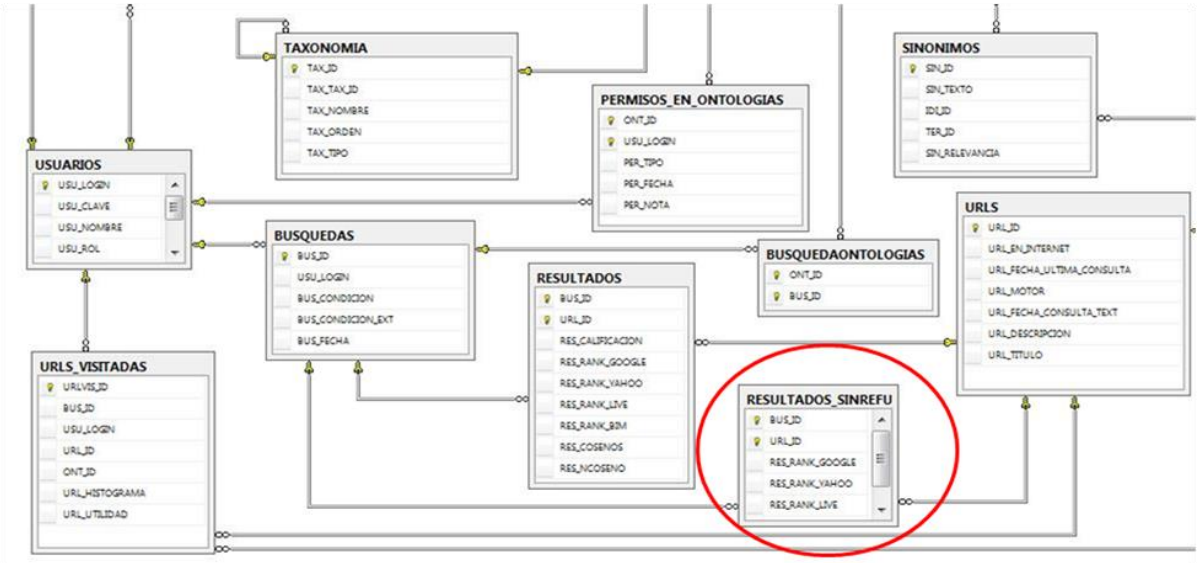


Figura 17 Tabla adicional para las consultas sin refuerzo

Como conclusiones para esta actividad se tiene que:

La ontología le permite al usuario contextualizar bien el ámbito de la consulta, por ejemplo, para el caso de la búsqueda del tema relacionado con estructura de datos, con la palabra clave "***Stack***", los buscadores tradicionales retornan en los primeros lugares temas que nada tienen que ver con el concepto de *pila* en informática, como por ejemplo pilas voltaicas. Para casos como estos, la ontología permite tener en cuenta más conceptos relacionados a *stack*, en el contexto de la computación, haciendo que se le presenta al usuario los resultados que realmente está buscando.

Algunos resultados que realmente eran útiles para los usuarios fueron retornados en posiciones intermedias, de ahí que cuando los buscadores tradicionales retornan resultados útiles en posiciones intermedias o últimas, no es posible reubicar ese resultado en una posición más cercana a los primeros resultados o viceversa. Con la propuesta presentada, se les permite a los usuarios hacer una retroalimentación de tal forma que el ranking de documentos o el orden es re-calculado en futuras consultas, teniendo en cuenta la utilidad que este le presentó, de esta manera el resultado es reubicado, dependiendo de la calificación: en las primeras posiciones, intermedias o últimas. Este comportamiento fue comprobado en la prueba, obteniendo una clara ventaja de la

propuesta presentada en este proyecto con respecto a los buscadores más utilizados en la actualidad.

A continuación se muestran los resultados que han sido entregados por el metabuscador BIM y los buscadores (Google, Yahoo y Live) a dos de las preguntas planteadas para la prueba. El informe completo se puede consultar en el Anexo I

Nombre del aprendiz: Julián Mosquera

¿Cómo se recorre un arreglo bidimensional?

Cadena de consulta digitada por el usuario: Recorrer un arreglo bidimensional

Para el desarrollo de esta actividad se han tomado los primeros 10 (diez) resultados del metabuscador BIM y de cada uno de los motores de búsqueda (Google, Yahoo y Live) y se califican cuales son acertados para resolver la pregunta y cuáles no, se divide el número de resultados acertados entre el total mostrado en la página (10 resultados), obteniendo de esta manera un factor de exactitud dado en porcentaje.

En el metabuscador BIM se muestran los siguientes resultados:

Dirección en Internet	Posición	Relevante
http://sophia.javeriana.edu.co/~acarrillo/poo/material/2clasepoojavaarreglos.pdf	1	NO
http://www.frbb.utn.edu.ar/electronica/gfried/info-ii/guia%20tp%201.pdf	2	NO
http://www.mailxmail.com/curso-aprende-programar/estructuras-datos-arreglos	3	SI
http://hwfiestasb1.iespana.es/estructuras/sesion8_arrays_bi_dimensionales.pdf	4	SI
http://tio-alberto.galeon.com/inv/estructuras.doc	5	SI
http://www.scribd.com/doc/2892761/unidad-6-estructuras-estaticas	6	SI
http://www.forosdelweb.com/f18/recorrer-arreglo-bidimensional-529631/	7	SI
http://html.rincondelvago.com/estructura-de-datos_7.html	8	SI
http://www.quinqui.cl/qminiportales/qmp2_items.php?p=903&s=310&id=737	9	NO
http://ozarate.utj.edu.mx/academia/logica/pract/practica13.pdf	10	SI

Porcentaje: resultados relevantes primera página = resultados relevantes / total de resultados primera página. X 100%

El factor de exactitud dado por BIM fue del 70%

En el motor de búsqueda de Google

Dirección en Internet	Posición	Relevante
http://www.forosdelweb.com/f18/recorrer-arreglo-bidimensional-529631/	1	SI
http://www.forosdelweb.com/f69/arreglo-bidimensional-dinamico-como-crear-recorrer-501990/	2	NO
http://ozarate.utj.edu.mx/academia/logica/pract/practica13.pdf	3	NO
http://ozarate.utj.edu.mx/academia/logica/pract/practica14.pdf	4	NO
http://casidiablo.net/matrices-en-c-sharp/	5	SI
http://www.elcamajan.com/programacion/11991-arreglo-bidimensional-dinamico.html	6	NO
http://hwfiestasb1.iespana.es/estructuras/sesion8_arrays_bi_dimensionales.pdf	7	SI
http://www.todoexpertos.com/categorias/tecnologia-e-internet/programacion/c-sharp/respuestas/1922458/matrices-bidimensionales	8	SI
http://antares.itmorelia.edu.mx/~jcolivar/courses/c208a/c2_u2d.ppt	9	SI
http://exa.unne.edu.ar/depar/areas/informatica/programacion1/public_html/archivos/estructuras_arreglos.pdf	10	SI

Porcentaje: resultados relevantes primera página = resultados relevantes / total de resultados primera página. X 100%

El factor de exactitud dado por Google fue del 60%

En el motor de búsqueda de Yahoo

Dirección en Internet	Posición	Relevante
http://www.ccperalta.com/blog/2009/04/05/arreglos-en-php.html	1	NO
http://matematicas.udea.edu.co/~hugo/cursophp/arreglos.doc	2	NO
http://www.slideshare.net/videoconferencias/lenguaje-de-alto-nivel-ii-bimestre	3	SI
http://www.slideshare.net/videoconferencias/tutoria-ii-bim20082	4	SI
http://html.rincondelvago.com/estructura-de-datos_8.html	5	NO
http://html.rincondelvago.com/estructura-de-datos_7.html	6	NO
http://www.reloco.com.ar/prog/java/collections.html	7	NO
http://www.ldc.usb.ve/~ruckhaus/materias/ci2615/proyectoalgv1.doc	8	NO
http://www.scribd.com/doc/289166/sesion-04-arrays-y-colecciones	9	NO
http://www.mitecnologico.com/main/programacionorientadaobjetos	10	NO

Porcentaje: resultados relevantes primera página = resultados relevantes / total de resultados primera página. X 100%

El factor de exactitud dado por Yahoo fue del 20%

En el motor de búsqueda de Live

Dirección en Internet	Posición	Relevante
http://www.forosdelweb.com/f18/recorrer-arreglo-bidimensional-529631/	1	SI
http://www.scribd.com/doc/547400/us-internal-revenue-service-voneuw	2	NO
http://www.scribd.com/doc/2873579/antologia-estructura-datos-i	3	NO
http://hwfiestasb1.iespana.es/estructuras/sesion8_arrays_bi_dimensionales.pdf	4	SI
http://brevesnotanbreves.blogspot.com/	5	NO
http://www.todoexpertos.com/categorias/tecnologia-e-internet/programacion/pascal/respuestas?i=3	6	NO
http://radiovisioncasasgrandes.com/cotorrandoconelgallito.htm	7	NO
http://metaciencia.com	8	NO
http://anjaviermx888.webcindario.com/files/sub_paginas/proyectos/arreglo matrices y archivos.pdf	9	NO
http://www.talentoenvivo.net/noticias.htm	10	NO

Porcentaje: resultados relevantes primera página = resultados relevantes / total de resultados primera página. X 100%

El factor de exactitud dado por MSN Live Search fue del 20%

Para la cadena de consulta: ***Recorrer un arreglo bidimensional*** se logró una exactitud del 70% para el metabuscador BIM, 60% para el buscador Google, 20% para Yahoo y 20% para MSN Live Search. Se observa una clara ventaja del metabuscador propuesto con respecto a los buscadores más utilizados en la actualidad.

¿Cómo acceder a los elementos de un arreglo bidimensional?

Cadena de consulta digitada por el aprendiz: elementos de un arreglo bidimensional

En el metabuscador BIM se muestran los siguientes resultados:

Dirección en Internet	Posición	Relevante
http://www.monografias.com/trabajos14/estruct-datos/estruct-datos.shtml	1	SI
http://tio-alberto.galeon.com/inv/estructuras.doc	2	SI
http://markmail.org/download.xqy?id=xyneay4a2rq3ggb4&number=1	3	SI
http://www.fismat.umich.mx/mn1/tutor_fort/arrays.html	4	SI
http://www.geocities.com/inf135/tutc/tema07.htm	5	NO
http://www.mailxmail.com/curso-aprende-programar/estructuras-datos-arreglos	6	SI
http://sistemas.itlp.edu.mx/tutoriales/progorientobjetos/t12.htm	7	SI
http://www.mitecnologico.com/main/arreglobidimensionalconceptosbasicos	8	SI
http://www.recursosvisualbasic.com.ar/htm/tutoriales/tutorial-	9	SI

basico6.htm		
http://www.forosdelweb.com/f18/recorrer-arreglo-bidimensional-529631/	10	NO

Porcentaje: resultados relevantes primera página = resultados relevantes / total de resultados primera página. X 100%

El porcentaje de exactitud dado por BIM fue del 80%

En el motor de búsqueda de Google

Dirección en Internet	Posición	Relevante
http://sistemas.itlp.edu.mx/tutoriales/progorientobjetos/t11.htm	1	NO
http://www.mitecnologico.com/main/arreglobidimensionalconceptosbasicos	2	NO
http://www.geocities.com/inf135/tutc/tema08.htm	3	NO
http://www.forosdelweb.com/f18/problema-para-eliminar-elemento-arreglo-bidimensional-693615/	4	NO
http://www.monografias.com/trabajos14/estruct-datos/estruct-datos.shtml	5	SI
http://www.fismat.umich.mx/mn1/tutor_fort/arrays.html	6	SI
http://html.rincondelvago.com/arreglos.html	7	SI
http://foros.hackerss.com/index.php?showtopic=212	8	NO
http://sapiens.ya.com/electrotext/logprog/cap9a.htm	9	NO
http://www.udb.edu.sv/academia/laboratorios/informatica/ip/guia10ip.pdf	10	NO

Porcentaje: resultados relevantes primera página = resultados relevantes / total de resultados primera página. X 100%

El porcentaje de exactitud dado por Google fue del 30%

En el motor de búsqueda de Yahoo

Dirección en Internet	Posición	Relevante
http://html.rincondelvago.com/arreglos.html	1	SI
http://www.geocities.com/inf135/tutc/tema08.htm	2	NO
http://riosur.net/modules.php?name=news&file=article&sid=21	3	NO
http://www.mitecnologico.com/main/arreglosunidimensionales	4	NO
http://www.mitecnologico.com/main/resoluci%f3ndeproblemasconarreglos	5	NO
http://www.mastermagazine.info/termino/3920.php	6	NO
http://fismat.umich.mx/mn1/tutor_fort/arrays.html	7	SI
http://www.geocities.com/inf135/tutc/tema07.htm	8	NO
http://fismat.umich.mx/mn1/manual/node9.html	9	NO
http://matematicas.udea.edu.co/~hugo/cursophp/arreglos.doc	10	NO

Porcentaje: resultados relevantes primera página = resultados relevantes / total de resultados primera página. X 100%

El porcentaje de exactitud dado por Yahoo fue del 20%

En el motor de búsqueda de Live

Dirección en Internet	Posición	Relevante
http://www.scribd.com/doc/2275141/elaborar-un-arreglo-floral	1	NO
http://www.forosdelweb.com/f18/recorrer-arreglo-bidimensional-529631/	2	NO
http://sistemas.itlp.edu.mx/tutoriales/progorientobjetos/t12.htm	3	SI
http://html.rincondelvago.com/arreglos.html	4	SI
http://wiki.lidsol.org/index.php?title=representando_matrices_con_un_doble_apuntador	5	NO
http://www.educarchile.cl/creasitio/itfuser/home.aspx?siteid=907102&sitename=razonamientografico&sitetypename=personas&pageid=934471&pagename=rangos%20de%20propor.html	6	NO
http://yaqui.mxl.uabc.mx/%7eaflores/nueva_pagina/apuntes/unidad2	7	SI
http://anjaviermx888.webcindario.com/files/sub_paginas/proyectos/arreglo matrices y archivos.pdf	8	NO
http://www.mastermagazine.info/termino/5518.php	9	NO
http://fismat.umich.mx/mn1/tutor_fort/arrays.html	10	SI

Porcentaje: resultados relevantes primera página = resultados relevantes / total de resultados primera página. X 100%

El porcentaje de exactitud dado por MSN Live Search fue del 40%

En este caso se observa que la exactitud de los resultados retornados para la consulta: ¿Cómo acceder a los elementos de un arreglo bidimensional? fueron del 80% para el metabuscador BIM, 30% para Google, 20% para Yahoo, 40% para MSN Live Search observándose claramente una ventaja sobre los otros buscadores con los que se comparan los resultados.

Los resultados restantes se pueden consultar en el Anexo I.

12.2 PRUEBAS BETA

En una segunda actividad se instaló la aplicación en los computadores de la sala de sistemas del SENA (ver Anexo I), luego se le pidió al mismo grupo de estudiantes con los que se hizo la prueba alfa que utilizara la aplicación sin presencia de los desarrolladores. Después de que utilizaron la herramienta se les pidió que llenaran una encuesta que permitió encontrar las dificultades que tuvieron en el tema de usabilidad y qué tan satisfechos quedaron con los resultados arrojados por el meta buscador. En la encuesta, manifestaron los problemas que enfrentaron al usar el meta buscador, los errores que arrojó y sus puntos de vista acerca de la aplicación y lo que se debería mejorar. En el Anexo I se encuentran los resultados de esta prueba.

En general, los comentarios de los usuarios fueron muy ***positivos***, ya que manifestaron que el meta buscador arrojó los resultados que estaban esperando y que no tuvieron que escudriñar todos los resultados para encontrar lo que necesitaban, es decir, las respuestas estuvieron en la gran mayoría de casos, en la primera página de resultados. En cuanto a la facilidad de utilización del metabuscador, el 71.42% de los usuarios respondió que le pareció fácil de usar, el 78% manifestó que encontró fácilmente lo que estaba buscando en Internet y otro 78% encontró los resultados que necesitaba en menos de 2 minutos, resultados que cumplieron con las expectativas iniciales.

Sin embargo también manifestaron la necesidad de hacerlo más veloz específicamente al momento de recuperar la información. Como en el meta buscador es necesario seleccionar una taxonomía y luego la visualización de los resultados es un poco diferentes a los buscadores tradicionales, algunos usuarios al principio pensaron que era un poco difícil de utilizar, pero a medida que lo utilizaron se fueron adaptando rápidamente. El uso de la taxonomía definitivamente implica un cambio en el modelo de búsqueda al que la gente está acostumbrada a usar.

Esta prueba igualmente permitió encontrar algunas falencias que se mejoraron, pero que se deben seguir mejorando, por ejemplo: los usuarios consideran que el tiempo para mostrar los resultados de las búsquedas es un poco alto. Esta situación se debe a dos circunstancias principales, a saber:

- Las imágenes que muestran la calificación que han tenido los documentos a lo largo del tiempo se genera con el Office Web Component (OWC) el cual es un poco demorado, y que debe ser cambiado por un componente que demore menos tiempo en la generación de las gráfica. En nuestro caso se utilizó OWC porque es gratis, esta incluido en VS.NET y para efectos del proyecto, muestra la retroalimentación que necesitábamos.

- El tiempo que se demora invocando las tres APS's de los buscadores es alto, comparado con la respuesta que da cada motor por separado. Es preciso analizar la calidad de los resultados frente al tiempo de respuesta cuando se usa una sola API, seleccionar la mejor fuente (cuál de las tres APIS's) y tambien analizar si la mejor solución es tomar menos resultados de las tres APIS's (por ejemplo, sólo los 20 primeros resultados de cada buscador).

Finalmente, la mitad de los usuarios coinciden en que los nuevos elementos de la interfaz (taxonomía general del conocimiento y feedback) hacen un poco complejo de usar el meta buscador, por ejemplo, consideraron tedioso tener que seleccionar manualmente la o las categorías del conocimiento en la taxonomía.

La encuesta planteada para la actividad, las respuestas obtenidas y algunos comentarios a dichas respuestas se presentan a continuación:

1. ¿Le pareció fácil utilizar el meta buscador?

   El 71.42% de los usuarios respondió que SI, mientras que el restante 28.58% respondió que no.

2. Las diferencias en interfaz del meta buscador con respecto a los buscadores que habitualmente utiliza fueron:

   El 21.43% de los usuarios encontró que el meta buscador tenía muchas diferencias con respecto a los buscadores que habitualmente utilizan, mientras que el 78.57% lo encontró totalmente diferente.

3. ¿Qué elementos le cambiaría al meta buscador para mejorar la apariencia en la interfaz?

   a. El explorador de los documentos (visualización de resultados) debe ser más amplio (46%)
   b. No les gusta seleccionar los nodos de la taxonomía (37,5%)
   c. Historial de búsquedas (17%)

   Teniendo en cuenta que la visualización debe hacerse muy similar a como se hace en Google, Yahoo y MSN Live para no cambiar el paradigma de los usuarios (“No me has pensar”), hecho que no modifica en absoluto el modelo interno del motor.

   En cuanto a la selección de la taxonomía, se considera apropiado que el usuario en un paso inicial seleccione las taxonomías que acostumbra a usar y después esta parte de la interfaz se oculte hasta que el usuario decida cambiar a otra taxonomía, visualizando en todo momento la rama taxonómica por la que está consultando. Con lo anterior se logra que el modelo de búsqueda por palabras claves que se usa en Google, Yahoo y MSN Live no cambie sustancialmente.

4. En cuanto a los procesos de búsqueda realizados ¿Encontró fácilmente lo que estaba buscando?

   En cuanto a las búsquedas, el 78.57% de los usuarios manifestó que encontró de manera fácil los resultados que necesitaba, mientras que el 21.43% manifestó lo contrario

5. ¿Los resultados arrojados por el meta buscador cumplieron las expectativas de su búsqueda?

   En cuanto a las expectativas de búsqueda, el 92.85% se mostró satisfecho con los resultados mientras que el 7.15% restante manifestó lo contrario.

6. ¿Cuánto tiempo necesitó para encontrar algún documento después de que el meta buscador visualizara los resultados?

Según los usuarios que utilizaron la aplicación, un total de 78.57% manifestaron haber gastado entre 1 y 2 minutos mientras que el 21.43% gastó entre 2 y 4 minutos y no hubo personas que necesitaran más de 4 minutos.

7. ¿Qué sugerencias daría para mejorar el meta buscador?

    a. Aumentar la velocidad (64%)
    b. Mejorar la interfaz (29%)
    c. Hacer el sistema más intuitivo (7)

    Debido a que se están utilizando algunos componentes que permiten construir gráficas, se ve reducido el tiempo en visualizar los resultados, pero es necesaria esta acción puesto que le permite al usuario saber si el resultado le ha sido útil.

    En cuanto a la interfaz, se han agregado varios elementos que permiten interactuar más con los procesos de búsqueda, permitiéndole al usuario indagar sobre búsquedas anteriores y calificar resultados.

CAPÍTULO VI – CONCLUSIONES RECOMENDACIONES Y TRABAJO FUTURO

13. CONCLUSIONES

En este trabajo se diseñó un modelo de búsqueda web que utiliza minería de datos (probado con distancia de cosenos similar a k-nn y con pruebas preliminares con SVD y k-means) en conjunto con taxonomías y ontologías. Con la combinación de estos conceptos se logra que la información recuperada por los motores de búsqueda Google, MSN Search y Yahoo sea filtrada y organizada de acuerdo a la similitud que éstos tienen con respecto a una consulta ampliada por una ontología de un dominio especifico. Este modelo puede ser más prometedor que el establecido por DMOZ, que pretende clasificar con expertos las páginas de Internet.

Se desarrolló una aplicación basada en el modelo propuesto, que hace uso e integra tres de los buscadores más reconocidos de la actualidad (Google, Yahoo y MSN Live Search), tomando en cuenta esto, se considera que el resultado del proyecto fué el desarrollo de un meta buscador que aplica distancia de cosenos con base en los términos de la ontología previamente seleccionada y modifica dicha distancia de acuerdo al feedback del usuario. El proceso de cálculo de distancias cuenta con una mejora al incluir la ontología, ya que con este recurso se hace un proceso de reducción de la dimensionalidad (selección de características) que es muy importante en el proceso de búsqueda web.

Igualmente se desarrolló un prototipo de un editor de ontologías que permite adicionar características particulares a las ontologías, por ejemplo, fijar el idioma y relevancias/pesos/prioridades a los conceptos, y adicionar sinónimos a cada concepto en múltiples idiomas. Para el desarrollo del prototipo de editor de ontologías fue necesario utilizar una API de código abierto desarrollado para java, denominado JENA, que permite crear ontologías OWL, la cual interoperó sin mayores problemas con el entorno de desarrollo seleccionado para el presente proyecto, Visual Studio .NET 2005. Dicha interoperabilidad facilitó el trabajo de creación y manejo de ontologías, lo que muestra además la importancia de utilizar las dos plataformas sin tener que desarrollar aplicaciones por separado y en el futuro cercano esta interoperabilidad se muestra como una poderosa herramienta de uso habitual en el futuro.

La comparación entre los resultados del meta buscador y los motores de búsqueda mostraron que el modelo propuesto en la mayoría de las ocasiones obtiene mejores resultados que los retornados por los motores de búsqueda tradicionales en forma independiente. Se puede decir que el modelo es apropiado para mejorar las búsquedas a pesar de que involucre un paso más en la búsqueda, la selección del nodo (s) en la taxonomía.

14. RECOMENDACIONES Y TRABAJO FUTURO

Teniendo en cuenta que en el presente trabajo no se crean ontologías, únicamente se hace uso de algunas que existen en la biblioteca de DAML, se recomienda tener en cuenta alguna metodología o proponer una nueva, para la creación de ontologías en forma distribuida a través de Internet y colaborativamente por un grupo de usuarios expertos para lo cual se podría usar el feedback que el sistema proporciona con su uso.

Es importante que el prototipo del editor de ontologías sea mejorado, hacerlo más robusto para que acepte cualquier lenguaje de ontologías. Es necesario que se agreguen funcionalidades que permitan trabajar con axiomas, instancias y razonadores que permitan hacer inferencias, y posteriormente analizar cómo incorporar estas funcionalidades para mejorar el modelo de búsqueda.

En el corto plazo es necesario integrar y probar más técnicas de minería de datos, por ejemplo, agrupar los resultados por temáticas utilizando k-means, clúster jerárquico, frases frecuentes o reglas de asociación, luego etiquetarlas para identificar cada grupo claramente y finalmente mostrar los resultados obtenidos.

CAPÍTULO VII – GLOSARIO Y BIBLIOGRAFÍA

15. GLOSARIO

- **SVD**: Acrónimo de Singular Value Decomposition. Técnica del álgebra lineal para factorizar matrices.
- **Minería de Datos**: DM o Data Mining, es el proceso de extracción de información y patrones de comportamiento que permanecen ocultos en grandes cantidades de información.
- **Ontología**: Es la formulación de un exhaustivo y riguroso esquema conceptual dentro de un domino dado, con la finalidad de facilitar la comunicación y compartir la información entre diferentes sistemas. Para que la ontología sea aceptada debe ser formal y aceptada por una comunidad de expertos.
- **Taxonomía del conocimiento**: es un sistema formal de clasificación del conocimiento.
- **Motor de Búsqueda**. Sistemas que recorren la red recolectando e indexando la mayor cantidad de información posible, gracias a programas automáticos conocidos como robots, también llamados spider.
- **Meta Buscador**: Sistemas que no disponen de bases de datos propias, puesto que buscan en otros buscadores. Recogen la petición del usuario y la envían a los buscadores, éstos la devuelven y los meta buscadores la clasifican antes de presentarla al usuario.
- **Índices temáticos:** Son listas de recursos organizadas en jerarquías desde lo más general a lo más específico. El proceso de clasificación se hace de forma manual.
- **K-means:** Algoritmo que tiene como objetivo minimizar las diferencias de los elementos en cada clúster al mismo tiempo que maximiza la diferencia de los elementos que caen en diferentes clústeres.
- **Clúster:** Es un punto usado para representar un conjunto de valores que tienen algo en común y se pueden agrupar en función de una característica determinada.

16. BIBLIOGRAFÍA

[1] NIELSEN, Jakob. *When search engines become answer engines*. *Jakob Nielsen's Alertbox* [en línea]. Publicado el 16 de Agosto de 2004. Disponible en Web: http://www.useit.com/alertbox/20040816.html

[2] O'HARA, Kieron; SHABDOLT, Nigel. *Knowledge Technologies and the semantic web* [en línea]. Publicado Junio 2004. Disponible en Web: http://eprints.ecs.soton.ac.uk/12469/01/Knowledge%2520technologies%2520and%2520the%2520semantic%2520web.pdf

[3] CHAU, Michael; FANG, Xiao; SHENG, Olivia R. Liu. "Analyzing the Query Logs of a Website Search Engine" [en línea]. *Journal of the American Society for Information Science and Technology.* John Wiley & Sons, Inc, 2005, volumen 56, edición 13, p 1363-1376. Disponible en Web: http://portal.acm.org/citation.cfm?id=1127134&dl=ACM&coll=&CFID=15151515&CFTOKEN=6184618#abstract

[4] Ferretsoft. *Sitio Web Webferret* [en línea]. Disponible en Web: http://www.webferret.com/learn.htm

[5] GIMENO, José Manuel. *Otras formas de buscar en Internet* [en línea]. Publicado el 14 de Junio de 2004. Disponible en Web: http://laflecha.net/articulos/blackhats/internet_oculta/

[6] BrightPlanet. *The Deep Web: Surfacing Hidden Value* [en línea]. Publicado en Julio de 2000. Disponible en Web: http://citeseer.ist.psu.edu/cache/papers/cs/19136/http:zSzzSzmaya.cs.depaul.eduzSz~mobasherzSzclasseszSzcs589zSzpaperszSzdeepweb.pdf/llc00deep.pdf

[7] MADRID, Juan M; GAUCH, Susan. *Incorporating Conceptual Matching in Search* [en línea]. Disponible en Web: http://www.ittc.ku.edu/~sgauch/papers/CIKM02.pdf

[8] MOORE, Jerome; HAN, Eui-Hong; BOLEY, Daniel; GINI, Maria; GROSS, Robert; HASTINGS, Kyle; KARYPIS, George; KUMAR, Vipin y MOBASHER, Bamshad. *Web Page Categorization and Feature Selection Using Association Rule and Principal Component Clustering* [en línea]. Universidad de Minnesota, Minneapolis, USA. Disponible en Web: http://citeseer.ist.psu.edu/cache/papers/cs/3856/http:zSzzSzwww-users.cs.umn.eduzSz~grosszSzpaperszSzwits97.pdf/moore97web.pdf

[9] RAMAKRISHNAN, Naren. *Frontiers of Search* [en línea]. IEEE Computer Society, Octubre 2005. Disponible en Web: http://csdl2.computer.org/comp/mags/co/2005/10/rx026.pdf

[10] DACONTA, Michael C.; OBRST, Leo J.; SMITH, Kevin T. *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*. John Wiley & Sons, 2003. "Capítulo 8: Understanding Ontologies". ISBN: 0471732571

[11] SULLIVAN, Danny. *Nielsen NetRatings Search Engine Ratings* [en línea]. Publicado el 22 de Agosto de 2006. Disponible: http://searchenginewatch.com/showPage.html?page=2156451

[12] KWOK, Cody C. T.; ETZIONI, Oren, WELD, Daniel S. *Scaling Question Answering to the web* [en línea]. Universidad de Washington, Seattle, WA, USA. Publicado el 13 de Noviembre de 2006. Disponible en Web: http://www.cs.washington.edu/homes/weld/papers/mulder-www10.pdf

[13] JOACHIMS, Thorsten. *Optimizing Search Engines using Clickthrough Data* [en línea]. Universidad de Cornell, New York, USA. Disponible en Web: http://www.cs.cornell.edu/people/tj/publications/joachims_02c.pdf

[14] BOULTER, Mark. *Smart Client Architecture and Design Guide.* Microsoft Corporation. 2004.

[15] CHAPMAN, Peter; CLINTON, Julian; KERBER, Randy; KHABAZA, Thomas; REINART, Thomas; SHEARER, Colin y WIRTH, Rudiger. *CRISP— DM Step-by-Step Data Mining Guide* [en línea]. Publicado en Agosto de 2000. Disponible en Web: http://www.crisp-dm.org/CRISPWP-0800.pdf.

[16] CABENA, Peter; HADJINJAN, Pablo; STADLER, Rolf. *Discovering Data mining From Concept To Implementation.* Prentice Hall PTR, Upper Saddle River, New Jersey 1998

[17] CHEN, Hsinchun; CHUNG, Wingyan; QIN, Yi; CHAU, Michael; XU, Jennifer Jie; WANG, Gang; ZHENG, Rong y ATABAKHSH, Homa. *Crime Data Mining: An Overview and Case Studies* [en línea]. Publicado en Mayo 2003. Disponible en Web: http://delivery.acm.org/10.1145/1130000/1123231/p34-chen.pdf

[18] CHEN, Bernard, TAI, Phang C, HARRISON, R. y PAN, Yi. *Novel Hybrid Hierarchical – K – means Clustering Method (H-K-means) for Microarray Analysis.*

[19] GAUCH, Susan. *BDEI: Biodiversity Information Organization using Taxonomy (BIOT)* Disponible en Web: http://www.digitalgovernment.org/library/library/pdf/gauch.pdf

[20] BAEZA-Yates, R., A., & RIBEIRO-Neto, B. (1999). Modern Information Retrieval: Addison-Wesley Longman Publishing Co., Inc.

[21] SALTON, G. Automatic Information Organization and Retrieval New York: Mc Graw-Hill Computer Series, 1968.

[22] HARMAN, D. BAEZA −Yates, R. Information retrieval: Data Structures and Algorithms. Prentice−Hall, Englewood Cliffs, 1992. p. 363−392.

[23] SALTON, G. and McGill, M.J. Introduction to Modern Information Retrieval. New York: Mc Graw-Hill Computer Series, 1983

[24] LENTILUCCI, Emmett. Using the Singular value decomposition. Rochester Institute of Tecnology. Mayo 29 de 2003

[25] M. W. Berry, S.T. Dumais y G. W. O'Brien Using Linear Algebra for Intelligent Information Retrieval

[26] JAUREGI, A. Zelaia. Fundamentos de Latent Semantic Indexing (LSI) y su aplicación a la categorización de textos periodísticos en euskera. Dpto. Ciencias de la Computación e Inteligencia Artificial, Universidad del País Vasco UPV-EHU. Publicado en Procesamiento del Lenguaje Natural, núm. 32 (2004)

[27] FURNAS, W. G., DEERWESTER, S., DUMAIS, S., LANDAUER, T. K., HARSHMAN, R., STREETER L. A., LOCHBAUM, K. E., 1988 Information retrieval using a singular value decomposition model of latent semantic structure. Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 465-480.

[28] GROSSMAN, D. A. y FRIEDER O.1998. Information Retrieval: Algorithms and Heuristics. Kluwer Academic Publishers, USA

[29] ZELIKOVITZ, Sarah. HIRSH, Haym. Using LSI for Text Classification in the Presence of Background Text. Rutgers University. Disponible en web: http://portal.acm.org/citation.cfm?id=502605

[30] BLUSTEIN, James. WEBBER, E. Robert. Using LSI to evaluate the quality of hypertext links. University of Western Ontario. Disponible en web: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.46.4051

[31] OSINSKI, Stanislaw. An Algorithm for clustering of web search results. Disponible en web: http://project.carrot2.org/publications/osinski-2003-lingo.pdf

[32] OSINSKI, Stanislaw. STEFANOWSKI, Jerzy. WEISS Dawid. Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition. Disponible en web: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.9.5370

[33] BLANCO GÓMEZ, Manuel. *Estudio de buscadores* [en línea]. Publicado en Abril 2003 Disponible en Web: http://trevinca.ei.uvigo.es/~pcuesta/sm/practicas/Estudio.pdf

[34] FAYYAD, Usama; PIATETSKY SHAPIRO, Gregory; SMYTH, Padharaic. "The KDD process for extracting useful knowledge from volumes of data" [en línea]. *Communications of the ACM*. Volumen 39, edición 11, noviembre de 1996. Disponible en Web: http://delivery.acm.org/10.1145/250000/240464/p27-fayyad.pdf

[35] HAN, Jiawei; KAMBER, Micheline. *Data Mining: Concepts and techniques*. Morgan Kaufman Publishers.

[36] KANTARZDZIC, Mehmed. Data *Mining: Concepts, Models, Methods and Algorithms*. John Wiley & Sons 2003.

[37] MICHAEL, Berry; GORDON, Linoff, *Data Mining Techniques for Marketing, Sales and Customer Support*, Wiley, Hoboken, NJ, 1997.

[38] Helsinki Institute For Information Technology. *Open Source Search: A Data Mining Platform* [en línea]. Association for Computing Machinery, Junio 2005. Disponible en Web: http://www.acm.org/sigs/sigir/forum/2005J/buntine_sigirforum_2005j.pdf

[39] HAN, Eui-Hong; KARYPIS, George; MEWHORT, Doug. HATCHARD, Keith. "Intelligent Metasearch Engine for Knowledge Management" [en línea]. *Conference on Information and Knowledge Management. Proceedings of the twelfth international conference on Information and knowledge management.* [New Orleans, USA]: Association for Computing Machinery, 2003, p. 492 – 495. Disponible en Web: http://delivery.acm.org/10.1145/960000/956955/p492-han.pdf

[40] LAROSE, Daniel T. *Discovering Knowledge in Data. An Introduction to Data Mining* [en línea]. Universidad del Cauca, Biblioteca Virtual Unicauca, 2005. John Wiley & Sons, Inc. "Capítulo 8. Hierarchical and k-means clustering". Páginas 147 – 162. Disponible en Web: http://site.ebrary.com/lib/bibliocauca/Top?channelName=bibliocauca&cpage=1&docID=10114096&f00=text&frm=smp.x&hitsPerPage=10&layout=document&p00=Data+Mining+Introduction&sortBy=score&sortOrder=desc

[41] LAROSE, Daniel T. *Discovering Knowledge in Data. An Introduction to Data Mining* [en línea]. Universidad del Cauca, Biblioteca Virtual Unicauca, 2005. John Wiley & Sons, Inc. "Capítulo 9. Kohonen networks". Páginas 163 – 179. Disponible en Web: http://site.ebrary.com/lib/bibliocauca/Top?channelName=bibliocauca&cpage=1&docID=10114096&f00=text&frm=smp.x&hitsPerPage=10&layout=document&p00=Data+Mining+Introduction&sortBy=score&sortOrder=desc

[42] Jain, A. K., Murty, M.N. y Flynn P. J. *Data Clustering a review. ACM Computing surveys. 1999.*

[43] CyberTavernTV LLC. *About iBoogie* [en línea]. Disponible en Web http://www.iboogie.com/Text/about.asp

[44] KARANIKAS, Haralampos y THEODOULIDIS, Babis. *Knowledge Discovery in Text and Text Mining Software* [en línea]. Disponible en Web: http://www.crim.co.umist.ac.uk/parmenides/internal/docs/Karanikas_NLDB2002%20.pdf

[45] LOZANO, Tello Adolfo. *Ontologías en la Web Semántica* [en línea]. Disponible en Web: http://www.informandote.com/jornadasIngWEB/articulos/jiw02.pdf

[46] SÁNCHEZ, Diana Marcela; CAVERO, José María; MARCOS, Esperanza. *Ontologías y MDA: una revisión de la literatura* [en línea]. Disponible en Web: http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-157/paper03.pdf

[47] RODRIGUEZ MENESES, Olga, LOAIZA SALAZAR, Carol. *MAESTRA-Sistema multiagente basado en ontologías para optimizar la recuperación de información* [en línea]. *IV Iberoamericano Telematicn Conference CITA 2006.* [Monterrey, Mexico]. Mayo 3 – 5 de 2006. Disponible en Web: http://cita2006.mty.itesm.mx/ponencias/MAESTRA-Sistema%20multiagente%20basado%20en%20ontologias%20para%20optimizar%20la%20recuperacion%20de%20informacion.pdf

[48] FUENTES FERNANDEZ, Rubén; PAVON, Juan. *Agentes para la recuperación de información especializada en Internet*. Universidad Complutense de Madrid. Disponible en Web: http://ma.ei.uvigo.es/desma2005/articulos/1084_Fuentes.pdf

[49] NOY, Natalya F.; MCGUINNES, Deborah L. *Ontology Development 101: A Guide to Creating Your First Ontology* [en línea]. Universidad de Stanford, Stanford, CA. Publicado el 19 de septiembre de 2005. Disponible en Web: http://protege.stanford.edu/publications/ontology_development/ontology101-es.pdf

[50] CENTELLES, Miquel. *Taxonomías para la categorización y la organización de la información en sitios Web* [en línea]. "Hipertext.net", núm. 3, 2005. Disponible en Web: http://eprints.rclis.org/archive/00008748/01/Taxonom%C3%ADas_para_la_categorizaci%C3%B3n_y_la_organizaci%C3%B3n_de_la_informaci%C3%B3n_en_sitios_web.pdf

[51] WELTY, Christopher A. *The Ontological Nature of Subject Taxonomies* [en línea]. Disponible en Web: http://www.cs.vassar.edu/faculty/welty/papers/fois-98/fois-98-1.html

[52] SCHWARZ, Katharina. Domain model enhanced search A comparison of taxonomy, thesaurus and ontology. University of Utrecht. Disponible en web: http://homepages.cwi.nl/~media/publications/masterthesis_kat_domainmodel_2005.pdf

[53] SACCO, Giovanni M. “Dynamic Taxonomies: A model for large information bases” [en línea]. *IEEE Transactions on Knowledge and Data Engineering*. Mayo 2006, volumen 12, edición 3. Disponible en Web: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=846296

[54] LOZANO, A. Ontologías en la Web Semántica, España: Universidad de Extremadura, Área de Lenguajes y Sistemas Informáticos, Departamento de Informática. 2003.

[55] MILLER, Eric. The Semantic Web. Disponible en web: http://www.w3.org/2002/Talks/www2002-w3ct-swintro-em/slide3-0.html

[56] BERNERS-LEE, T., HENDLER, J. y LASSILA, O. The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities.

[57] OCLC Online Computer Library Center. Sitio Web [en línea]. Disponible en Web: http://www.oclc.org/about/default.htm

[58] Dmoz. Open Directory Project Web Site (Dmoz). from http://www.dmoz.org/about.html

[59] The Library of Congress. Sitio Web de la Biblioteca de los Estados Unidos [en línea]. Disponible en Web: http://www.loc.gov/index.html

[60] MERLOT Multimedia Educational Resource for Learning and Online Teaching. Sitio Web [en línea]. Disponible en Web: http://www.merlot.org/merlot/index.htm

[61] Universidad del Cauca. Repositorio de acceso público basado en SCORM [en línea]. Disponible en Web: http://spar.unicauca.edu.co/spar/default.aspx

[62] Yahoo. Sitio Web del directorio de Yahoo [en línea]. Disponible en Web: http://search.yahoo.com/dir?fr=yfp-t-501

[63] Google. Sitio Web del directorio de Google [en línea]. Disponible en Web: http://www.google.com/dirhp?hl=es

[64] *DARPA Agent Markup Language DAML* [en línea]. Disponible en Web: http://www.daml.org/

[65] TOP U.S. SEARCH ENGINES February 2009 – Nielsen .[en línea]. Obtenida el 6 de Abril de 2009. Disponible en Web: http://r-rwebdesign.com/blog/?p=458.

[66] IPROSPECT BLENDED SEARCH RESULTS STUDY Abril 2008 – iProspect [en línea]. Disponible en Web: http://www.iprospect.com/premiumPDFs/researchstudy_apr2008_blendedsearchresults.pdf