Implementación de un Sistema de Recuperación de Información

Dra. Nora La Serna Palomino[1], Lic. Ulises Román Concha[1], Lic.Norberto Osorio[1]

[1]Facultad de Ingeniería de Sistemas e Informática
Universidad Nacional Mayor de San Marcos

nlasernap@unmsm.edu.pe, uromanc@unmsm.edu.pe , nosorio@unmsm.edu.pe

RESUMEN

El trabajo que se presenta en este artículo se desarrolla en el área de los Sistemas de Recuperación de Información (SRI), y tiene como objetivo principal implementar un sistema de almacenamiento y recuperación de información, y se utilizó como herramientas de desarrollo software libre y aplicaciones Web. Asimismo, se seleccionó tomar como referencia el modelo de espacio vectorial [1,13], el cual es uno de los modelos más utilizados actualmente en estos sistemas. Si bien en esta etapa del trabajo se ha construido el Sistema, el objetivo final es contar con una herramienta eficiente y competitiva que pueda ser utilizada para almacenar y recuperar información de las distintas disciplinas del quehacer humano. Fundamentalmente, se han realizado las siguientes actividades: 1) El desarrollo e implementación de cada uno de los módulos del sistemas; 2) Preparación de los datos de prueba; y 3) Evaluación del sistema y la propuesta de tareas futuras.

Palabras clave: Sistemas de recuperación de información, modelo del espacio vectorial, XML - Extensible Markup Languaje, tecnologías Web.

ABSTRACT

The work that is presented in this article develops in the area of the Information Recovery Systems (IRS), whose main objective is to implement a System of storage and recovery of information, also it considers to use as software development tools free and technologies of information like web technologies and the XML metalanguage. At the same time to take as reference the Vector Model [1, 13] that is one of the most utilized models in these area. The purpose of the research is to build a competitive and efficient tool that can be utilized to store and to recover information in the different disciplines of the human task. Fundamentally, the following activities have been carried out: 1) The development and the implementation of each one of the modules of the systems; 2) Preparation of the data of test; and 3) Evaluation of the system and the proposal of future tasks.

Key words: Information retrieval systems, vector model, XML - Extensible Markup Languaje, Web technologies.

1. INTRODUCCIÓN

El trabajo que se presenta en este artículo describe el diseño e implementación de un prototipo de un sistema de almacenamiento y recuperación de información. En la construcción del prototipo se han utilizado como herramientas de desarrollo software libre, y tecnologías como aplicaciones Web y el metalenguaje XML.

Un sistema de recuperación de información almacena grandes volúmenes de documentos, los cuales pueden venir de procesadores de textos, de páginas Web, de otras bases de datos, documentos electrónicos e inclusive otros archivos. A la vez el sistema debe disponer de interfaces que permitan hacer consultas en lenguaje natural acerca de los documentos que se encuentran almacenados. Para ello utiliza estructuras de datos y operaciones que permiten el almacenamiento y recuperación eficiente de la información. Algunas de las aplicaciones más conocidas son los buscadores Web (de texto e imágenes) y las bibliotecas digitales.

Si bien existen muchos de estos sistemas, todavía hay mucho trabajo por desarrollar en esta área. Las empresas que desarrollan estos sistemas y que destacan en este rubro buscan soluciones adecuadas que permitan mejorar los indicadores de evaluación de estos sistema para: 1) aumentar las tasas de exhaustividad, 2) reducir el espacio de almacenamiento, 3) aumentar la velocidad de proceso, 4) proporcionar interfaces adecuadas, 5) mejorar la precisión para recuperar la información solicitada, entre otros problemas que están pendientes de resolver.

Previo a la realización del presente trabajo, se realizó un estudio y evaluación de las técnicas y modelos que se utilizan en la construcción de estos sistemas [9]. Se encontró que uno de los modelos más utilizados en la actualidad es el modelo de espacio vectorial [13], el cual se ha tomado como referencia para la construcción del sistema planteado.

Si bien en esta etapa del trabajo desarrollado se ha construido un prototipo de un sistema de recuperación de información, el objetivo propuesto es contar con una herramienta eficiente y competitiva que pueda ser utilizada para almacenar y recuperar información de diferentes disciplinas del quehacer humano. Para alcanzar el objetivo trazado, en la penúltima sección se plantean tareas inmediatas y a mediano plazo.

La estructura del presente artículo es la siguiente: En la sección 2 se presenta la arquitectura del sistema de recuperación de información desarrollado; en las secciones 3 y 4 se describen cada uno de los módulos del sistema: subsistema de almacenamiento y subsistema de recuperación respectivamente. La sección 5 corresponde a la evaluación del prototipo desarrollado y se proponen tareas futuras para mejorarlo, y finalmente en la sección 6 se presentan las conclusiones del trabajo desarrollado.

2. ARQUITECTURA DEL SISTEMA DE RECUPERACIÓN DE INFORMACIÓN DESARROLLADO

Para el desarrollo del sistema, se seleccionó uno de los modelos más utilizados en estos sistemas, el modelo de espacio vectorial [9,13]. Según el modelo, cada documento es representado mediante un vector de *n* términos, en donde un término es la unidad mínima de información, por ejemplo una palabra o la raíz sintáctica de una palabra. En el modelo, a cada término se le asigna un peso para medir la importancia de un término y de esta manera un término permite distinguir un documento de otro en la colección de documentos.

Siguiendo el modelo de espacio vectorial, las consultas de los usuarios también son representadas mediante un vector de términos, en donde dicho vector debe coincidir con los términos de la matriz que se forma a partir de la colección de documentos. Asimismo, se calcula el peso de los términos de la consulta.

Posteriormente, se seleccionan aquellos documentos que se aproximan más a la pregunta del usuario, mediante un cálculo denominado *similaridad*, que podría ser por ejemplo el producto del vector de la consulta del usuario con cada vector de la matriz de términos. Finalmente, se ordenan los documentos seleccionados de mayor a menor valor de similaridad. En las secciones 3 y 4 se describen en detalle las fórmulas de cálculos utilizados y los detalles de implementación.

A partir del modelo de espacio vectorial se pueden distinguir varias funcionalidades en un sistema de recuperación de información; sin embargo, para alcanzar nuestro objetivo hemos considerado dos componentes principales del sistema: 1) El subsistema de almacenamiento de la colección de documentos, y 2) El subsistema de recuperación de la información a partir de la consulta del usuario. En la Figura N.° 1, se presenta la arquitectura del sistema desarrollado con sus dos componentes principales.

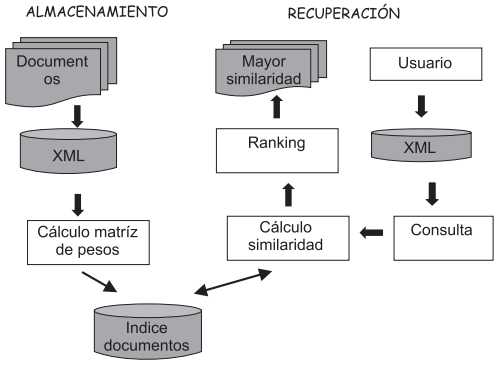


Figura N.º 1. Arquitectura del sistema implementado.

Si bien el motor de búsqueda está diseñado para procesar un tipo de documento, el que se explicará a continuación; sin embargo, se podría actualizar la interfaz de captura para almacenar cualquier tipo de documentos de otras disciplinas del quehacer humano.

Durante la implementación del motor de búsqueda se han utilizado documentos digitales de la Biblioteca de la Facultad de la Universidad a la cual pertenecemos, concretamente hemos utilizado archivos de las tesinas desarrolladas por los bachilleres para optar el Título de Ingeniero. Básicamente para las pruebas del sistema hemos considerado los siguientes datos de las tesinas: 1) título de la tesina, 2) resumen, 3) autor, 4) asesor, y 5) palabras claves. Queda pendiente implementar el resto de información de las tesinas.

En el Apéndice 1 se observa un ejemplo de un tipo de documento en formato XML utilizado por el sistema.

La base de datos para los subsistemas de almacenamiento y consulta consta de varias tablas. A continuación se indican la descripción, su nombre y los campos definidos para cada tabla.

- Tabla de términos o índices.
  TBDOCTER (termino, frecuencia, id_docume, peso)
- Tabla que registra los documentos.
  TBDOCUME (id_docume, titulo, resume, autor, grado, asesor, keyword, id_archiv)
- Tabla que registra fórmulas para hallar pesos
  TBFORPES (co_forpes, no_forpes, ti_estado)
- Tabla que registra fórmulas para hallar IDF
  TBFORIDF (co_foridf, no_foridf, ti_estado)
- Tabla que registra la consulta del usuario
  TBCONTER (id_consul, termino, nu_peso)
- Tabla que registra términos vacíos
  TBTERVAC (termino)
- Tabla que registra el IDF de cada término en la base de datos
  TBTERIDF (termino, nu_idf, nu_fredoc)

Asimismo, se han utilizado otras tablas igualmente útiles, como el que registra el código de usuarios, la tabla que registra la identificación de archivos, y tablas temporales para el cálculo de pesos intermedios. Es importante mencionar que el motor de búsqueda KARPANTA [6] desarrollado en la Universidad de Salamanca, ha sido una excelente guía en la implementación del sistema. En el Apéndice 2 se muestran un ejemplo de las tablas TBDOCTER.

3. SUBSISTEMA DE ALMACENAMIENTO DE LA COLECCIÓN DE DOCUMENTOS

En este subsistema se pueden distinguir dos módulos que sobresalen: 1) Registro de documentos, y 2) Cálculo de los pesos de los términos.

3.1. Registro de documentos

El objetivo de este módulo es procesar los documentos, para ello se realizan las siguientes tareas: 1) Se leen los documentos, la captura de los documentos en el sistema puede ser a través de una interfaz, o mediante una carga automática de documentos en formato XML, 2) Se hace un análisis léxico del texto, 3) Se eliminan del texto palabras que no son significativas en el proceso de selección de términos, denominadas *palabras vacías*; 4) Se actualiza la tabla de términos TBDOCTER, y finalmente se registra el documento en la tabla de documentos TBDOCUME.

En la tarea del análisis léxico, se seleccionan los términos que pueden ser significativos que ayuden a distinguir un documento de otro en la colección de documentos. En esta tarea además se eliminan signos de puntuación, símbolos separadores como espacios en blanco, tabuladores, tratamiento de mayúsculas, minúsculas, palabras acentuadas, y caracteres extraños.

La eliminación de *palabras vacías* o *stop words*, se realiza con el objetivo de reducir aquellos términos que tienen poca capacidad semántica, o por su alta frecuencia son poco significativos en el proceso de recuperación de la información, por ejemplo los artículos, las preposiciones, conjunciones, etc. Es una forma de delimitar el

número de términos que servirán como términos índice. Las palabras vacías se registran en una tabla TBTERVAC de la base de datos.

Los términos que son seleccionados, también conocidos como *índices*, corresponden a una palabra de la lengua española. Normalmente se caracterizan por su naturaleza sintáctica, pueden ser sustantivos o verbos o derivados de ellos, por ejemplo, casa y casita. Los términos seleccionados de un documento son ingresados en la tabla de términos TBDOCTER de la base de datos, además se actualiza la frecuencia del término en el documento. El proceso de selección de términos se realiza tantas veces como documentos se van a procesar.

Se ha considerado una tarea adicional, que es el registro de usuarios, debido a que en este subsistema solamente podrán acceder personas autorizadas para actualizar la información del sistema. En la Figura N.° 2, se muestra la interfaz de registro de documentos, con la selección de la opción de carga automática. Como se observa en la interfaz, hay tres formas de la captura de la información de los documentos: 1) ingresar el documento a través del teclado, 2) carga automática del documento en formato XML, y 3) carga de un conjunto de documentos mediante un fichero.

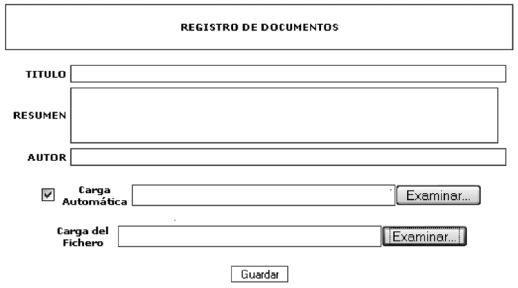


Figura N.° 2. Interfaz de registro de documentos.

3.2. Cálculo de los pesos de los términos

Una vez seleccionado el conjunto de términos de la colección de documentos, se deben calcular los pesos de los términos, para ello se tiene en cuenta dos factores importantes:

1. Hallar la frecuencia de un término (*tf*), es decir, el número de veces que aparece el término en un documento. Así, si un término aparece mucho en un documento, se supone que es importante en ese documento.
2. Cálculo del factor *idf (inverse document frequency),* es una función inversamente proporcional a la frecuencia de un término en la colección de documentos, concretamente el número de documentos en que aparece el término; este cálculo se debe a que si un término aparece en muchos documentos, entonces ese término no es útil para distinguir un documento de los otros de la colección, el factor *idf* se aplica en estos casos para elevar el poder discriminatorio del término.

Existen varias técnicas para asignar pesos a los términos, dos de ellas las que presentamos a continuación son las más representativas; las expresamos mediante las ecuaciones 1 y 2. En la ecuación 1, una de las más simples, para calcular el peso de cada elemento del vector, se tiene en cuenta la frecuencia del término dentro de cada documento *tf*, combinándola con la frecuencia inversa del término en la colección *idf* [1, 7].

$$W_{ij} = tf_i * idf_j, \qquad \text{(ecuación 1)}$$

Donde, W_{ij} es el peso del término j en el documento i.

En la ecuación 2, por cierto una de las técnicas más utilizadas, el peso del término se calcula como el producto de la frecuencia del término j en el documento i, multiplicado por el logaritmo de N / df_j.

$$W_{ij} = tf_{ij} * \log N / df_j, \qquad \text{(ecuación 2)}$$

Donde, N es el número de documentos de la colección, y df_j es el número de documentos en que aparece el término j.

Adicionalmente, suele aplicarse algún factor de normalización que permita equilibrar las diferencias en tamaño de los documentos, evitando la posibilidad de que las frecuencias sean mayores en documentos más grandes.

Para hallar los pesos de los términos, básicamente se realizan las siguientes operaciones:

1. Calcular las frecuencias de cada término en la colección.
2. Cálculo del IDF (Inverse Document frecuency)
3. Cálculo de la frecuencia en cada documento
4. Hallar el peso del término en cada documento
5. Aplicar un factor de normalización

En el sistema, una vez que el proceso de selección de términos de un documento o de una colección de documentos se ha realizado, se llaman a las funciones *CalcularIDF*, *CalcularPesos* y *Normalizar*. En donde se utilizan las fórmulas seleccionadas a partir de las tablas

TBFORIDF y TBFORPES, que registran las fórmulas para hallar IDF y pesos respectivamente.

El IDF es el mismo para cada término, su cálculo se registra en la tabla TBTERIDF de la base de datos, asimismo se registra en la tabla la frecuencia del término hallado. Luego se multiplica el IDF por la frecuencia y se determina el peso final de un término al dividir el producto de ambos entre el factor de normalización. Finalmente, el peso hallado de cada término se actualiza en la tabla de términos o índices TBDOCTER.

4. SUBSISTEMA DE RECUPERACIÓN

La consulta o solicitud del usuario hecha en lenguaje natural, también se expresa mediante un vector de términos, los términos deben ser los mismos que el de la colección de documentos. El mecanismo de obtención de pesos descritos en la sección 3, también se aplica a la consulta del usuario. De esta manera, la consulta y cada uno de los documentos están expresados en vector y matriz de pesos de términos respectivamente, que posteriormente, mediante un cálculo de similaridad permiten hallar aquellos documentos que se aproximan más a la pregunta del usuario.

La resolución de la consulta Baeza-Yates y Ribeiro-Neto [1, 7] consiste en un proceso de establecer el grado de semejanza entre el vector consulta y el vector de cada uno de los documentos; aquellos cuyo grado de similitud sea más elevado se ajustarán mejor a las necesidades expresadas en la consulta. Sin embargo, es el usuario el que debe decidir la relevancia de los documentos recuperados, siendo ésta una característica totalmente subjetiva del mismo.

Hay varios métodos para calcular la similitud que existe entre un documento y una consulta, una de las más utilizadas es aquella que calcula la distancia que existen entre los vectores que los representan, y realiza el producto escalar de esos vectores, dicho producto a su vez corresponde al coseno del ángulo entre los dos vectores (ecuación 3).

Sim (vector di, vector dj) = vector di * vector dj
(ecuación 3)

$$= \cos(\text{vector di}, \text{vector dj}) = \frac{\text{vector di} * \text{vector dj}}{|\text{vector di}| * |\text{vector dj}|}$$

La similaridad es un valor entre cero y uno. Dos vectores iguales tienen similaridad uno, dos vectores que no comparten ningún término tienen similaridad cero. Se seleccionan los documentos que tienen mayor similaridad con la consulta del usuario.

Para hallar los documentos que un usuario solicita, básicamente se realizan las siguientes operaciones:

1. Selección de términos de la consulta del usuario.
2. Cálculo de los pesos de la consulta del usuario.
3. Cálculo de similaridad entre la consulta del usuario y cada documento de la colección.
4. Se ordenan los documentos que tienen mayor valor de similaridad, y se muestran los resultados al usuario.

En el subsistema de recuperación del sistema, el proceso que selecciona los términos de la consulta del usuario es el mismo que cuando se extraen los términos de un documento en el subsistema de almacenamiento descrito en la sección 3, pero en este caso es más sencillo debido a que el número de términos en la consulta del usuario no es grande. Los términos seleccionados se registran en la tabla TBCONTER junto con el código de identificación de la consulta.

El cálculo de los pesos de los términos de la consulta del usuario, también se realizan como en el subsistema de almacenamiento, pero en este caso igualmente el número de términos en una consulta es pequeño, por lo tanto la frecuencia de aparición de cada término es generalmente igual a la unidad. En este caso también se aplica el IDF obtenido en el susbsistema de almacenamiento. Luego, el peso hallado de cada término se registra en la tabla TBCONTER.

El proceso para hallar el cálculo de similaridad entre la consulta del usuario y cada documento de la colección, y ordenar los documentos que tienen mayor valor de similaridad es simple, en el sistema se resuelve con la instrucción que se muestra en la Figura N.° 3.

```
" select doc.id_docume,doc.no_titulo,doc.no_resume," +
" sum(dt.nu_pesnor*con.nu_peso) as nu_simila " +
" from tbconter con,tbdocter dt,tbdocume doc where " +
" doc.id_docume=dt.id_docume and " +
" dt.no_termin=con.no_termin and " +
" con.id_consul=? " +
" group by doc.id_docume,doc.no_titulo,doc.no_resume " +
" order by nu_simila desc ";
```

Figura N.° 3. Cálculo de la similaridad en el sistema.

En la Figura N.° 4, la interfaz muestra el módulo de consulta, primero aparece el formulario en donde se ingresa el texto de la consulta, y luego en la parte inferior de la misma página se muestran los documentos ordenados por el valor de similaridad, una vez activado el botón "Buscar". Aquellos documentos que tienen mayor valor se acercan más a la consulta del usuario. Para ver el detalle de cada documento se debe pulsar en el ícono de la columna "Ver Documento".

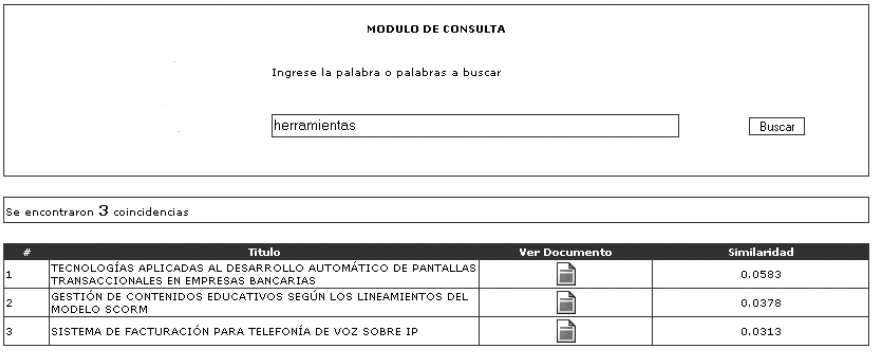
5. EVALUACIÓN DEL PROTOTIPO Y TRABAJOS FUTUROS

El sistema desaarrollado es una aplicación web, codificado en lenguaje Java, y utiliza una base de datos MySQL. Se han utilizado las siguientes herramientas para su implantación: 1 PC Pentium 4, el compilador Java 2 Standard Edition Runtime Environment, XML 1.0, el servidor de aplicaciones Tomcat, el SGBD MySQL 5.0.

El trabajo desarrollado nos ha permitido diseñar e implementar un prototipo de un sistema de recuperación de información utilizando software libre y tecnologías de información emergentes. Otra ventaja del sistema es el hecho de utilizar el metalenguaje Extensible Markup Languaje – XML para la definición de la estructura y el contenido de los documentos electrónicos, que facilitará el intercambio de información y la cooperación con otros sistemas de la institución. En el Apéndice 2 se muestra ejemplos de la tabla TBDOCTER, con valores de los cálculos realizados durante la implementación del sistema.

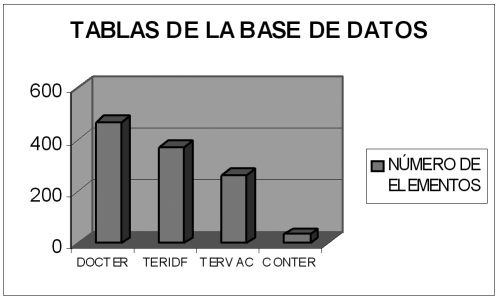
En la Figura N.° 5 se observa lo siguiente: 1) Tablas de la base de datos, con el número de elementos procesados en cada tabla, y 2) El gráfico correspondiente de la tabla.

El objetivo a mediano plazo es que el sistema desarrollado pueda ser competitivo y utilizado como una herramienta que registre y recupere documentos de otras disciplinas. Para llegar al objetivo trazado, tenemos planeado desarrollar actividades inmediatas y a mediano plazo. Una tarea inmediata es procesar al menos 400 documentos digitales que disponemos, y con ello evaluar y mejorar la performance del sistema implementado. A la fecha se han ingresado cuarenta documentos.



#	Titulo	Ver Documento	Similaridad
1	TECNOLOGÍAS APLICADAS AL DESARROLLO AUTOMÁTICO DE PANTALLAS TRANSACCIONALES EN EMPRESAS BANCARIAS		0.0583
2	GESTIÓN DE CONTENIDOS EDUCATIVOS SEGÚN LOS LINEAMIENTOS DEL MODELO SCORM		0.0378
3	SISTEMA DE FACTURACIÓN PARA TELEFONÍA DE VOZ SOBRE IP		0.0313

Figura N.° 4. Interfaz del módulo de consulta.



TABLAS	NÚMERO DE ELEMENTOS
TBDOCTER	467
TBTERIDF	370
TBTERVAC	265
TBCONTER	33

Figura N.° 5. Tabla y gráfico de la base de datos con el número de elementos procesados.

Adicionalmente, se espera incorporar nuevas funciones al sistema que permitan el registro en el subsistema de almacenamiento de diferentes tipos de documentos como por ejemplo archivos de procesadores de textos, de páginas Web, de otras bases de datos, documentos electrónicos e inclusive otros tipos de archivos.

Otra tarea es realizar el proceso con otras fórmulas para el cálculo de pesos principalmente en la sección de almacenamiento. Así como utilizar otros criterios de similaridad que permitan optimizar el sistema desarrollado.

Con el desarrollo de las tareas propuestas se espera robustecer el sistema para mejorar los indicadores de evaluación del sistema, es decir: 1) aumentar las tasas de exhaustividad, 2) Optimizar el espacio de almacenamiento, 3) aumentar la velocidad de proceso, 4) proporcionar interfaces adecuadas, y 5) mejorar la precisión para recuperar la información solicitada.

Adicionalmente, es importante mencionar que el motor de búsqueda KARPANTA [6], desarrollado en la Universidad de Salamanca, ha sido una excelente guía en el trabajo desarrollado.

6. CONCLUSIONES

El trabajo desarrollado ha dado lugar a una propuesta de diseño e implementación de un sistema de almacenamiento y recuperación de información, que inicialmente utiliza para el registro de la información, documentos digitales de la Biblioteca de la Universidad a la que pertenecemos, pero el objetivo a mediano plazo es optimizar el sistema para que sirva como una herramienta de almacenamiento y recuperación de diferentes tipos de documentos. En el prototipo se han incorporado herramientas de desarrollo de software libre y tecnologías de información emergentes como aplicaciones Web y el metalenguaje XML.

Anteriormente a la realización del presente trabajo, se realizó un estudio y evaluación de las técnicas y modelos que se utilizan en la construcción de los SRI [9], y se encontró que uno de los modelos más utilizados en la actualidad es el modelo de espacio vectorial [13]. Modelo que se ha tomado como referencia para la construcción del sistema planteado.

La arquitectura del sistema desarrollado presenta dos subsistemas: 1) El subsistema de almacenamiento de la colección de documentos, y 2) El subsistema de recuperación de la información a partir de la consulta del usuario. El primero realiza el registro de documentos y el cálculo de pesos de los términos. Mientras que el segundo selecciona los términos de la consulta del usuario, realiza el cálculo de los pesos de los términos de la consulta, y mediante un factor de similaridad determina los documentos que más se acercan a la pregunta del usuario.

Finalmente, se ha propuesto un conjunto de tareas que se deben realizar con la finalidad de robustecer el sistema desarrollado para que sea una herramienta eficiente y competitiva.

AGRADECIMIENTOS

El presente trabajo se desarrolla en el marco del proyecto “Construcción de un motor de recuperación de información para un Sistema de Bibliotecas”, financiado parcialmente por el Consejo Superior de Investigación de la Universidad Nacional Mayor de San Marcos.

BIBLIOGRAFÍA

[1] Baeza-Yates R, Ribeiro-Neto B. (1999). *Modern Information Retrieval*. Maryland: Addison-Wesley-Longman Publishing co.

[2] Baeza-Yates R, Davis E. (2004). Ranking global de páginas Web basado en atributos de los enlaces; CLEI 2004.

[3] Brin S, Page L. (1998). The anatomy of a large-scale hypertextual Web search engine. Computer Netsorks and ISDN Systems, 30. pp. 107-117

[4] Chu H, Rosenthal M. “Search engines for the WWW: A comparative study and evaluation methodology” En http://www.asis.org/annual-96/ElectronicProceedings/chu.html

[5] Delgado Domínguez. “Mecanismos de recuperación de Información en la www”, Universidad de las Islas Baleares, España. 1998. http://dmi.uib.es/people/adelaida/tice/modul6/memfin.pdf

[6] Figuerola C, Alonso J, Zazo A. (2000). Diseño de un motor de recuperación de la información para uso experimental y educativo. BID N.º 4.

[7] Frakes WB, Baeza Yates R. (1998). “Information Retrieval: data structures and algorithms”. Prentice Hall.

[8] Lancaster FW, Warner AJ. (1973). Information retrieval today. Arlington, VA: Information Resources.

[9] La Serna PN y grupo. (2005). Diseño del sistema de recuperación de información para la biblioteca FISI. Vol 2. Revista RISI.

[10] Martínez MF, Rodríguez MJ. (2003). Síntesis y crítica de las evaluaciones de la efectividad de los motores de búsqueda en la Web. http://InformationR.net/ir/8-2/paper148.html

[11] Martínez Méndez FJ. (2000). Sistemas de almacenamiento y recuperación de información, http://www.um.es/gtiweb/fjmm/sari2000.htm

[12] Salton G. (1980). The SMART system. Encyclopedia of Library and Information Science.

[13] Salton G, McGill M. (1983). Introduction to Modern Information Retrieval. Mc. Graw-Hill.

[14] Van Rijsbergen CJ. (1979). Information Retrieval. London: Butterworths.

APÉNDICES

1. Ejemplo de un tipo de documento utilizado en formato XML.

<?xml version="1.0" encoding="ISO-8859-1" ?>

- <!--

<!DOCTYPE web-app

PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"

"http://java.sun.com/j2ee/dtds/web-app_2.2.dtd">

-->

- <documento>

<titulo>GESTIÓN DE CONTENIDOS EDUCATIVOS SEGÚN LOS LINEAMIENTOS DEL MODELO SCORM</titulo>

<alumno>Aissa Medallit Tasayco Pozo</alumno>

<asesor>Román Concha, Ulises</asesor>

<grado>Licenciado en Computación</grado>

<abstract> El presente trabajo tiene como objetivo proponer un entorno para la gestión de contenidos educativos e-learning con una adecuada metodología, con la finalidad de optimizar y hacer más eficiente los procesos de recolección, creación, transformación y distribución de los contenidos. El modelo educativo SCORM combina lo mejor de la educación impartida por un instructor con herramientas que mejoran el aprendizaje de los estudiantes y la efectividad de los profesores, logrando que los contenidos educativos sean accesibles, interoperables, durables, y reutilizables. Se espera que la gestión de contenidos educativos bajo el modelo SCORM permita el desarrollo de componentes y sistemas de educación adecuados, para poder ejecutar cualquier acción posterior de portabilidad de los mismos.</abstract>

<keyword>SCORM, Agregación, Objeto de Aprendizaje, Metadata, Contenidos, Asset, Objeto de Contenido Compartible, Metadatos de los Objetos de Aprendizaje, Empaquetamiento de Contenido, Sistema de Gestión de Contenidos (CMS), Sistema de Gestión del Aprendizaje (LMS)</keyword>

</documento>

2. Ejemplo de la tabla TBDOCTER con cálculos realizados

INSERT INTO `tbdocter` (`id_docume`,`no_termin`,`nu_frecue`,`nu_peso`,`nu_pesnor`) VALUES

(5,'CARTERA',1,2.6094379124341,0.0715972795406304),

(5,'CRECER',1,2.6094379124341,0.0715972795406304),

(5,'ORGANIZACIÓN',1,2.6094379124341,0.0715972795406304),

(5,'IMPORTANCIA',1,2.6094379124341,0.0715972795406304),

(5,'RADICA',1,2.6094379124341,0.0715972795406304),

(5,'PROPORCIONARÁ',1,2.6094379124341,0.0715972795406304),

(5,'PERMITIRÁ',1,2.6094379124341,0.0715972795406304),

(5,'CLIENTE',1,1.91629073187416,0.0525788341455999),

(5,'COMODIDAD',1,2.6094379124341,0.0715972795406304),

(5,'SEGURIDAD',1,1.91629073187416,0.0525788341455999),

(5,'PROCESO',1,1.91629073187416,0.0525788341455999),

(5,'COMPRAS',1,2.6094379124341,0.0715972795406304);