Ministerio de Educación Superior

Universidad Central "Marta Abreu" de Las Villas

Facultad de Matemática, Física y Computación

Licenciatura en Ciencia de la Computación

Trabajo de Diploma

Sistema de Recuperación de Información sobre fuentes de datos heterogéneas. Goomed! Versión 2.0

Autor: Miguel Angel Estela Rodríguez

Tutor: MSc. Darien Rosa Paz

Santa Clara

2011

Dictamen

El que suscribe, Miguel Angel Estela Rodríguez, hago constar que el trabajo titulado Sistema de Recuperación de Información sobre fuentes de datos heterogéneas. Goomed! Versión 2.0 fue realizado en la Universidad Central “Marta Abreu” de Las Villas como parte de la culminación de los estudios de la especialidad de Licenciatura en Ciencia de la Computación , autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

_____

Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

_____ _____

Firma del tutor Firma del jefe del Laboratorio

_____

Fecha

Pensamiento

Todo programa hace algo perfectamente bien, aunque no sea exactamente lo que nosotros queremos que haga.

R.S. Pressmann
PC Users N§ 68.

Dedicatoria

A mi mamá (Eddys Amada Rodríguez Alonso).

A mi papá (Miguel Angel Estela Díaz).

A la memoria de fefa (Ramona Argelia Chaviano).

A mi novia (Isis Nelly Salazar Acosta).

Agradecimientos

Me gustaría expresar mi mayor agradecimiento por haber llegado a este momento tan especial como es la culminación de mi tesis, a mis padres Miguel Angel Estela Díaz y Eddys Amada Rodríguez Alonso por darme su apoyo minuto a minuto y compartir conmigo comprendiendo y analizando cada dificultad y cada problema con amor y dedicación y haciendo hasta lo imposible por ver mis sueños hechos realidad, siempre con la fe puesta en mí de terminar con éxitos, junto a ustedes incluyo también a mis hermanos y a Fefa. Y por todo esto que aunque es poco lo que yo pueda expresar les doy de todo corazón las gracias.

A mi novia Isis Nelly que a lo largo de estos años de estudio compartió conmigo los momentos buenos y difíciles, logrando en mí el placer de admirarla.

A mis suegros Esther y Sergio, quienes representan ejemplos de profesionales y de personas, guías para mí en todo momento.

A Isa, Llilla, Ricardo y a Sergito, personas que tanto estimo, quienes siempre me apoyaron, me enseñaron y creyeron en mí; a ellos muchas gracias.

A Yasnoly y a Rolando quienes tanto me ayudaron y encaminaron durante todos estos años.

A Asley, el Pere, Landy por soportarme todo este tiempo y a Julio por soportarlo yo a él.

A Rubén, Yendry, Deniel, Dairon (El flaco), al Pety (Fidel) y a todos mis amigos, que mi mayor deseo es poderlos mencionar a todos, pero no terminaría ni alcanzaría esta hoja. Para mí, la amistad y la unión en la que pude disfrutar en estos años, me ha servido de gran impulso y estímulo para llegar al final.

A Mary, ella sabe por qué.

A mis vecinos por todo su apoyo y siempre estar pendientes de mis padres y de mí.

A Lissett y a Diana porque ellas querían que las mencionara.

A mi tutor Darien por haber aceptado incondicionalmente mi solicitud para la tutoría de mi tesis de graduación y por toda la ayuda que me brindó.

En general, a toda mi familia y a la de Isis que son una sola, y a todos los que han contribuido de una forma u otra a que este sueño se haga realidad.

RESUMEN

El volumen de información electrónica existente en la actualidad es extremadamente grande y crece rápidamente, por lo que cada vez son más necesarios Sistemas de Recuperación de Información eficientes que permitan obtener en cada momento aquellos documentos que respondan a una necesidad informativa dada.

Teniendo en cuenta la complejidad del manejo de la información estructurada e información en texto libre, cuya integración se convierte en un aspecto esencial en muchos de los Sistemas de Información existentes en la actualidad, y las posibilidades reales de recuperar información a partir de las mismas, se desarrolló en el año 2010 la arquitectura de un Sistema de Recuperación de Información capaz de combinar ambos tipos de información, explotando las capacidades de consulta de las tecnologías de Bases de Datos y Recuperación de Información y se realizó una implementación de este modelo en el hospital de Villa Clara “Arnaldo Milián Castro”. Este centro cuenta con un repositorio de información estructurada y en texto libre referente a historias clínicas de autopsias practicadas en el Departamento de Patología.

En este trabajo se presenta una nueva versión de este sistema con nuevas prestaciones como la inclusión de la búsqueda por términos frecuentes y la creación de perfiles de usuarios. Además se integró al sistema la enciclopedia médica MedlinePlus para buscar en ella artículos relacionados con los términos de búsqueda y de esta forma complementar los resultados obtenidos por una consulta.

ABSTRACT

The volume of electronic information currently available is extremely large and is growing very fast, therefore the construction of efficient Information Retrieval Systems has progressively become more necessary to obtain documents which satisfy given information need.

Taking into account the complexity of managing structured information and free textual information, whose integration becomes an essential aspect in many of the existing Information Systems at present, and also taking into account the real possibilities to retrieve information from the integration of both types of information, is that, on 2010 a architecture of a Information Retrieval System with a capacity to combine both types information types was developed, exploiting the capabilities of querying Database and Information Retrieval technologies. An implement of this model at Arnaldo Milián Castro hospital in Villa Clara was carried out. This center has a repository of structured information and free text information of clinical records concerning clinical histories of autopsies accomplished in the Department of Pathology.

In this research it is presented a new version of this system with new functionalities like an inclusion of the search for frequent terms and the creation of the user profiles. Besides, it was integrated to the system the medical encyclopedia MedlinePlus to look for articles related with terms of search and, in this way, to complement the obtained results by a query.

TABLA DE CONTENIDOS

INTRODUCCIÓN

El problema de crear una historia clínica electrónica es una tarea compleja y en diversas partes del mundo se le ha dado solución parcial al mismo. Uno de los aspectos que complejiza tales esfuerzos es la mezcla de información estructurada y no estructurada presente en las historias clínicas tradicionales. Cuando se trabaja en la búsqueda de la automatización, uno de los enfoques es tratar de estructurar toda la información para que la recuperación de la misma se haga por métodos tradicionales asociados al modelo relacional de datos, lo que trae pérdida de información e incluso no aceptación por parte de los profesionales de la medicina.

En particular, el Departamento de Patología del Hospital “Arnaldo Milián Castro” de la ciudad de Santa Clara, Cuba, cuenta con un repositorio de información relacionada con historias clínicas de autopsias que se realizan en el mismo. Originalmente este repositorio de información se encontraba almacenado en una Base de Datos (BD) que contenía los campos estructurados relativos a los datos personales del paciente cadáver y además presentaba atributos que referían las descripciones de los diferentes elementos que componen una autopsia. Estas descripciones se conforman por texto libre, en su mayoría relativamente extenso.

Dado que los Sistemas de Gestión de Bases de Datos Relacionales (SGBDR) carecen de herramientas potentes de análisis de texto que permitan una eficiente recuperación de la información presente en campos con información textual, y la necesidad de los especialistas médicos de obtener información a partir de los campos que almacenan los elementos descriptivos de las autopsias practicadas, se hace imprescindible la aplicación de técnicas de Recuperación de Información (RI) para acceder a aquellas partes de las autopsias compuestas por texto libre.

Con el objetivo de automatizar la gestión hospitalaria en el citado hospital y que los médicos pudieran consultar de manera más eficiente el repositorio de autopsias, se propuso en (Castellanos et al., 2010) la arquitectura de un Sistema de Recuperación de Información (SRI) capaz de integrar información estructurada y en texto libre, explotando las capacidades de consulta de las tecnologías de BD y RI. Siguiendo este modelo, se implementó en el propio trabajo, la primera versión de un SRI que integra ambos tipos de información mediante la utilización de un SGBDR y una librería de alto rendimiento para la RI.

El proceso de recuperación en esta primera versión del SRI se puede perfeccionar mediante la creación de perfiles de usuarios que permitan en un futuro mejorar la calidad de la recuperación. Para ello se utiliza la información aprendida a partir de consultas previas formuladas al sistema por los usuarios y los juicios de relevancia proporcionados por los mismos.

Otra variante para dotar al sistema de opciones de consultas más flexibles para el usuario es la incorporación de la búsqueda por términos frecuentes, facilitándole así su interacción con el sistema y a su vez simplificándole la acción de buscar.

Se puede complementar los resultados que brinda el sistema dado una consulta integrando el mismo a una enciclopedia médica, de manera tal que una vez mostrados los resultados se le facilite al usuario los artículos de la enciclopedia relevantes a la consulta formulada.

Atendiendo a lo expuesto anteriormente se percibe como **problema de investigación** la necesidad de crear una nueva versión del SRI propuesto en (Castellanos et al., 2010) que mejore sus prestaciones en términos de facilidades de búsqueda y complementación de los resultados de las consultas formuladas e incluya la creación automática de perfiles de usuarios.

De esta forma, el **objetivo general** del presente trabajo es desarrollar una nueva versión de un SRI existente que adicione a éste nuevas prestaciones con el fin de mejorar el proceso de recuperación de información en el hospital “Arnaldo Milián Castro” de Santa Clara.

Para dar cumplimiento a este objetivo se plantean los siguientes **objetivos específicos**:

1. Crear un módulo que amplíe las capacidades de consulta del sistema añadiendo la búsqueda por términos frecuentes e incorporarlo a la nueva versión del SRI.
2. Integrar a la nueva versión del SRI un módulo para la creación automática de perfiles de usuarios que guarde para cada uno de ellos el historial de consultas que ha realizado al sistema y los juicios de relevancia proporcionados.

3. Integrar la nueva versión del SRI con la enciclopedia médica MedlinePlus para complementar los resultados obtenidos a partir de una consulta, con artículos relevantes a los términos de la búsqueda.

Preguntas de investigación

1. ¿Cómo se integró al sistema la enciclopedia médica MedlinePlus?
2. ¿Cómo se realiza la búsqueda por términos frecuentes?
3. ¿Cómo se crearon los perfiles de usuarios?

Justificación de la investigación

En Cuba se hacen ingentes esfuerzos por automatizar la información de salud de hospitales, policlínicos, consultorios, etc. En todos estos casos se presenta la problemática de la creación de historias clínicas electrónicas. De esta manera, es muy necesario para estas instituciones disponer de un sistema de acceso eficiente y eficaz a la información electrónica y especialmente a dichas historias clínicas.

Con el propósito de brindar una adecuada exposición de los resultados obtenidos en esta investigación, el resto de esta memoria queda estructurada de la siguiente manera: en el primer capítulo se repasarán los conceptos básicos de la RI y de los SRI, se tratan además los Sistemas de Recuperación de Información sobre documentos XML y finalmente se aborda el tema de la integración de las tecnologías de BD y RI. En el capítulo dos se describen las herramientas que fueron utilizadas en la implementación del SRI obtenido como resultado de esta investigación, sus potencialidades y capacidades de integración así como las modificaciones realizadas para la obtención de la nueva versión del sistema. Finalmente, en el tercer capítulo se presentan los aspectos fundamentales del diseño del sistema y sus facilidades de uso mediante la descripción de un caso de estudio.

CAPITULO I. INTRODUCCIÓN A LOS SISTEMAS DE RECUPERACIÓN DE INFORMACIÓN

En este capítulo se repasarán los conceptos básicos de la Recuperación de Información a través del estudio de los Sistemas de Información (SI) que se emplean en este campo, los denominados Sistemas de Recuperación de Información. De estos sistemas se analizarán sus componentes principales, los modelos de recuperación que se emplean en su diseño, aspectos relevantes a tener en cuenta para su evaluación y las técnicas más usadas para mejorar la recuperación. Además se aborda el tema de la integración de las áreas de BD y RI.

1.1 Introducción

La RI es una disciplina que desde hace varios años está experimentando un renovado interés debido al aumento de la disponibilidad de documentos en formato electrónico y de la necesidad consiguiente de obtener en cada momento aquellos que respondan a una necesidad informativa dada.

Según (Baeza-Yates and Ribeiro-Neto, 1999) la RI es el conjunto de acciones, métodos y procedimientos para la representación, almacenamiento, organización y recuperación de la información. El objetivo fundamental de la RI es, dada una necesidad de información y un conjunto de documentos, obtener los documentos relevantes para esa necesidad, ordenarlos en función del grado de relevancia, y presentarlos al usuario. Se define relevancia como la medida de cómo un documento se ajusta a una consulta, entendiéndose como consulta una expresión formal de la necesidad informativa del usuario.

Se habla de recuperación de información automática cuando las tareas indicadas anteriormente se lleven a cabo con un ordenador, definiendo un SRI como el software concebido para cumplimentarlas, cuyo objetivo fundamental es brindarle a un usuario que ha articulado una consulta toda la información que la satisfaga.

Seguidamente se describe cuál es el proceso completo de recuperación. Dada una colección de documentos, el primer paso es obtener una representación textual de los mismos en forma de palabras claves o términos de indexación (conjunto de palabras que resume el contenido del documento) y seguidamente, mediante la indexación, conseguir un conjunto de términos por cada documento. En este

momento, la base de datos documental está lista para ser utilizada y es cuando interviene el usuario del SRI, formulándole una consulta que exprese su necesidad de información. El SRI pone en marcha su motor de búsqueda y compara cada uno de los documentos almacenados con dicha consulta, obteniendo en algunos casos el grado con que el documento satisface a la consulta, y en otros simplemente seleccionando sólo los documentos que la satisfagan completamente. El siguiente paso es presentar al usuario la salida del proceso de búsqueda, el mismo evaluará dicha salida y decidirá si ésta es satisfactoria o, por el contrario, no ha satisfecho completamente su necesidad de información. Este hecho originará que el SRI vuelva a recuperar ayudado por la información adicional suministrada por el usuario, proceso que se conoce como realimentación de relevancia. En entornos experimentales, el SRI incorpora un módulo adicional que se encargará de determinar la calidad de la recuperación del sistema, proceso denominado evaluación de la recuperación (Fernández, 2001).

Los SRI implementan estas operaciones de diferentes maneras, lo que provoca una amplia diversidad en lo relacionado con la naturaleza de los mismos. Por tanto, para estudiarlos es necesario profundizar en cuáles son sus principales componentes.

1.2 Principales componentes de un Sistema de Recuperación de Información

Un SRI está compuesto por tres componentes principales: la base de datos documental, el subsistema de consultas y el mecanismo de recuperación. En las secciones siguientes se resumen los principales aspectos de cada uno de ellos.

1.2.1 La base de datos documental

Un documento es un objeto de datos, de naturaleza tradicionalmente textual, aunque la evolución tecnológica ha propiciado la profusión de documentos multimedia, incorporándose al texto fotografías, ilustraciones gráficas, vídeos animados, audio, entre otros. Un aspecto importante es la forma en que se representará el documento en el ordenador. En nuestros días la capacidad de almacenamiento de los ordenadores ha crecido considerablemente, por lo que se podrían almacenar los documentos íntegramente, lo que se conoce como representación a texto completo. Sin embargo, hay casos en que, debido al gran tamaño de las colecciones, no es posible. Por tanto, hay que buscar una manera

alternativa que pasa por el procesamiento del texto para obtener una representación del mismo en forma de palabras claves, descriptores, conceptos, o términos de indexación. Con esta forma de proceder se evitan dos problemas: por un lado, se reduce el espacio físico necesario para almacenar la colección; y por otro, se eliminan palabras que no aportan información alguna a la hora de describir el contenido del documento y que son fuentes de ruido para tareas futuras (Baeza-Yates and Ribeiro-Neto, 1999). Otra ventaja es que se expresan los documentos de una manera mucho más cómoda para que el ordenador los maneje eficientemente (Fernández, 2001). De esta forma, un documento se compondrá de una serie de descriptores. Desde un punto de vista matemático, la base de datos documental se puede representar por una tabla o matriz en la que cada columna indica las asignaciones de un determinado descriptor y cada fila representa un documento. Cada elemento de la matriz contiene cierta información numérica que expresa la asignación de un concepto a un documento y puede tener diferentes significados dependiendo del modelo de recuperación que se trate.

Para obtener estas representaciones se aplica un proceso conocido como *indexación*, que tiene como entrada los documentos en su estado inicial y como salida la base de datos documental, también conocida como *índice* del SRI. Este proceso se representa en la figura 1.1 y sus operaciones se explican brevemente a continuación.

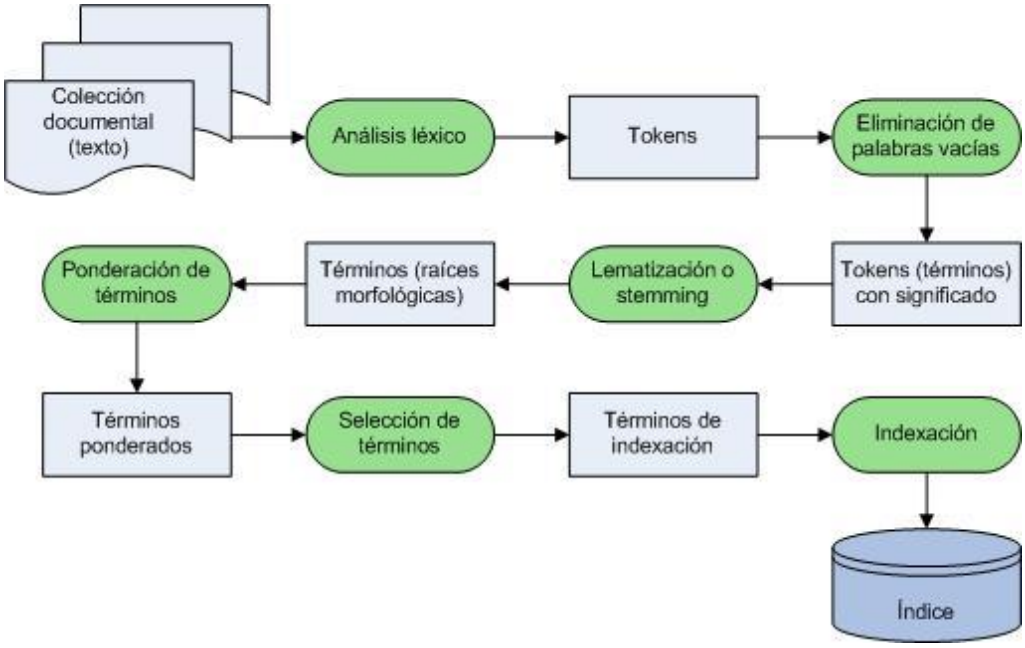

Figura 1.1. Representación del proceso de indexación en un SRI.

Análisis léxico del texto: Tiene como objetivo convertir la cadena de entrada en un conjunto de palabras o tokens, y determinar el tratamiento que se realizará sobre números, signos de puntuación, guiones, mayúsculas y/o minúsculas, nombres propios, entre los principales.

Eliminación de palabras vacías (stopwords en inglés): Este proceso se encarga de eliminar las palabras vacías de significado, como artículos, preposiciones, conjunciones, incluso en algunos casos, se pueden calificar así algunos verbos, adverbios y adjetivos (Baeza-Yates and Ribeiro-Neto, 1999). Por tanto, estas palabras no sirven como términos de indexación, ya que, por un lado son muy frecuentes, y por otro no representan correctamente el contenido del documento.

La acción normal que se lleva a cabo con ellas es su eliminación del texto, proceso que se pone en práctica mediante la búsqueda de cada palabra del texto en un diccionario que contiene la lista de palabras no aptas para la indexación.

Lematización (stemming en inglés): Durante este paso se extrae la raíz morfológica de cada palabra, eliminando sufijos y prefijos, originando así que el SRI pueda recuperar documentos incluyendo variantes morfológicas de los términos contenidos en la consulta; de esta forma se mejora la recuperación, a la vez que se ahorra espacio al almacenar solo las raíces. Este proceso es dependiente del idioma y puede llegar a ser muy complejo (como en el idioma español). De entre los numerosos métodos para extraer las raíces hay que destacar por su simplicidad y efectividad el diseñado por Porter (Porter, 1980). Adicionalmente, se debe poner en práctica un proceso de reconocimiento de raíces equivalentes con objeto de evitar confusiones con palabras que poseen la misma raíz, pero no están relacionadas en su significado.

Selección de términos: En este momento se tienen todos los términos candidatos a formar parte de la base de datos documental. El siguiente paso consistirá en determinar la importancia de cada término, de forma tal que, si es lo "suficientemente" importante, se escogerá para ser incluido en el conjunto de términos final. El cálculo de la importancia de cada término se conoce como ponderación del término.

Se han propuesto diversas medidas para calcular la importancia de los términos. Un primer enfoque se basa en contar las ocurrencias de cada término en un

documento, medida que se denomina frecuencia del término i-ésimo en el documento j-ésimo (Baeza-Yates and Ribeiro-Neto, 1999). Una segunda medida es la conocida como frecuencia documental inversa de un término en la colección, conocida normalmente por sus siglas en inglés: idf (Inverse Document Frequency) (Salton, 1989, Salton and McGill, 1983). Una propuesta mucho más empleada es la combinación de ambas medidas utilizando un esquema de ponderación que permita identificar a los términos que aparecen bastante en varios documentos individuales, y a la vez, que se hayan observado en contadas ocasiones en la colección completa. Estos son los términos que tendrán una capacidad de discriminación mayor con respecto a los documentos en los que aparecen (Korfhage, 1997).

Una vez que ha finalizado el análisis automático de la base de datos documental, un aspecto importante es su organización para conseguir un acceso eficiente y rápido en las operaciones que se realizarán posteriormente en el proceso de recuperación. De esta forma, se conoce como *fichero invertido* a una estructura de datos que almacena de manera ordenada todos y cada uno de los términos del glosario y, para cada uno de ellos, guarda la lista de documentos donde aparece, junto con su peso asociado (Harman et al., 1992).

La consulta formulada al SRI también es examinada por el módulo de indexación para conseguir su correspondiente representación, dependiendo del modelo de recuperación utilizado.

1.2.2 El subsistema de consultas

Este subsistema está compuesto por la interfaz que permite al usuario formular sus consultas y por un analizador sintáctico que toma la consulta y la desglosa en sus partes integrantes. Para llevar a cabo esta tarea dispone de un lenguaje de consulta que establece todas las reglas para generar consultas apropiadas y la metodología para seleccionar los documentos relevantes.

La interfaz ofrecerá facilidades al usuario a la hora de formular su consulta, ya que este no tiene por qué saber exactamente el funcionamiento interno del sistema. También se ocupará de mostrar al usuario el resultado de su búsqueda, una vez procesada su consulta.

La consulta que facilite el usuario no puede procesarse directamente en su forma original. Ha de recibir un tratamiento previo que consiste en desglosarla en sus componentes básicos, además de comprobar que corresponde con el formato que se espera de ella.

Después de este análisis, se lleva a cabo un preprocesamiento igual al que se hace con los documentos que componen la colección documental y se indexará o vectorizará, para luego ser enviada al mecanismo de recuperación, que determina qué documentos se consideran relevantes para las necesidades de información que representa.

1.2.3 El mecanismo de recuperación

Llegados a este punto, se tiene una representación del contenido de los documentos en la base de datos documental y también una representación de las consultas provenientes del subsistema de consulta. Lo que resta por resolver es la selección de los documentos que se consideran relevantes de acuerdo con los criterios de la consulta formulada.

De esto precisamente se encargará el mecanismo de recuperación, precisa el grado en el que las representaciones de los documentos satisfacen los requisitos expresados en la consulta y recupera aquellos documentos que son relevantes a la misma. Este grado se denomina RSV (Retrieval Status Value) (Baeza-Yates and Ribeiro-Neto, 1999) y la forma de calcularse varía en dependencia del modelo de recuperación que use el SRI.

1.3 Clasificación de los SRI atendiendo a los modelos de recuperación

Existe una gran cantidad de modelos de recuperación basados en tecnologías muy diferentes. A continuación se tratarán brevemente los tres clásicos: el booleano, el del espacio vectorial y el probabilístico.

1.3.1 El modelo booleano

Este modelo (Rijsbergen, 1979) se basa en la teoría del álgebra de Boole. Dentro de un sistema booleano, los documentos se encuentran representados por conjuntos de palabras claves. La indización se realiza asociando un peso binario a cada término del índice: 0 si el término no aparece en el documento y 1 si aparece aunque sea una sola vez. Las búsquedas consisten en expresiones de palabras claves conectadas con algún/os operador/es lógico/s (Y, O y NO).

El grado de similitud entre un documento y una consulta será también binario y un documento será relevante cuando su grado de similitud sea igual a 1, de lo contrario el documento no tendrá ninguna relevancia en cuanto a la consulta.

Cuando se ejecute la consulta, el subsistema de consulta extraerá el RSV de cada documento y decidirá qué conjunto de documentos es el que se considera relevante para dicha consulta. En este modelo esta operación es muy sencilla ya que no existe gradación de relevancia (el documento es totalmente relevante a la consulta o no lo es en absoluto). Por tanto, los valores del RSV serán 0 ó 1 y formarán el conjunto de documentos recuperados aquellos que tengan el RSV igual a 1.

1.3.2 El modelo del espacio vectorial

El marco del modelo vectorial (Salton and Lesk, 1968, Salton et al., 1975) está compuesto por el espacio vectorial de dimensión M (cada dimensión equivale a un término distinto del glosario), representando en él los documentos, las consultas y las operaciones algebraicas sobre los vectores de dicho espacio. Concretamente, la función que obtiene la similitud de un documento con respecto a una consulta se basa en la medida del coseno (Salton and McGill, 1983), la cual devuelve el coseno del ángulo que forman ambos vectores en el espacio vectorial.

Así, cuando ambos vectores son exactamente el mismo, el ángulo que forman es de cero grados y su coseno es uno. Por el contrario, cuando el ángulo es de noventa grados (los vectores no coinciden en ningún término), el coseno será cero. El resto de las posibilidades indicarán una correspondencia parcial entre el vector del documento y el de la consulta, ofreciendo así una gradación en los valores de relevancia, de modo que cuanto más cercanos sean los vectores del espacio, más similares serán éstos (Fernández, 2001).

1.3.3 El modelo probabilístico

El marco del modelo probabilístico está compuesto por conjuntos de variables, operaciones con probabilidades y el teorema de Bayes.

Todos los modelos de recuperación probabilísticos están basados en el *Principio de la ordenación por probabilidad*, conocido originalmente como "*the probability ranking principle*". Este principio, formulado por Robertson (Robertson, 1977), asegura que el rendimiento óptimo de la recuperación se consigue ordenando los

documentos según sus probabilidades de ser juzgados relevantes con respecto a una consulta, siendo estas probabilidades calculadas de la forma más precisa posible a partir de la información disponible. Así, y atendiendo a este principio, el objetivo primordial de cualquier modelo probabilístico pasa por calcular la probabilidad de relevancia dados una consulta y un documento (Fernández, 2001).

1.4 Evaluación de los Sistemas de Recuperación de Información

Un SRI comercial puede evaluarse empleando diversos criterios. La mayoría de los autores coinciden en seleccionar los siguientes como los más importantes: tiempo de ejecución (eficiencia), almacenamiento correcto y calidad de la recuperación efectiva (eficacia). La importancia relativa de estos factores debe decidirla el diseñador del sistema, y la selección de la estructura de datos y los algoritmos apropiados para su implementación dependerán de esa decisión.

La eficiencia en la ejecución se medirá por el tiempo que toma el sistema o una parte del mismo para llevar a cabo una operación. Este parámetro ha sido siempre un factor de peso a la hora de diseñar un SRI, especialmente desde que muchos de ellos son interactivos y un tiempo de recuperación excesivo interfiere con la utilidad del sistema, llegando a alejar a los usuarios del mismo. Los requerimientos no funcionales de un SRI normalmente especifican el tiempo máximo aceptable para una búsqueda y para las operaciones de mantenimiento de una base documental, tales como indexación de la colección, añadir y borrar documentos.
La calidad del almacenamiento se mide usualmente por el tamaño de los índices. Una forma común de medirla es la razón entre el tamaño de la colección original y los ficheros creados tras la indexación.

Se han propuesto múltiples medidas para evaluar la eficacia de un SRI, siendo las más empleadas y conocidas la exhaustividad (recall, en inglés) y la precisión, definidas como sigue en (Salton and Lesk, 1968):

$$Exhaustividad = \frac{Número de documentos relevantes recuperados}{Número de documentos relevantes}$$

$$Precisión = \frac{Número de documentos relevantes recuperados}{Número de documentos recuperados}$$

La *exhaustividad* se define como la proporción de documentos relevantes recuperados en una búsqueda determinada respecto al número de documentos relevantes para esa búsqueda en la base de datos. Por otra parte, la precisión es la proporción de documentos relevantes recuperados respecto al número total de documentos recuperados.

Generalmente, la exhaustividad se incrementa cuando el número de documentos recuperados también lo hace y, al mismo tiempo, la precisión decrece (Salton and McGill, 1983). Habrá usuarios que estén interesados en niveles altos de exhaustividad, por lo que efectuarán consultas muy generales, y otros que lo estén en niveles altos de precisión, diseñando consultas muy específicas. De todas formas, es altamente deseable que la exhaustividad se acerque lo más posible al 100%, al igual que la precisión, aunque ambos objetivos no se pueden alcanzar a la vez porque están inversamente relacionados.

El problema fundamental que poseen estas dos medidas es que la precisión se calcula de manera exacta, mientras que la exhaustividad no, ya que no se tiene un conocimiento claro de cuántos documentos relevantes existen en una colección para una consulta dada, por lo que se tendrá que estimar. Existen ocasiones en las que sí se conocen esos documentos relevantes: cuando la colección es muy pequeña y se pueden revisar uno por uno todos los documentos que la componen, y también en el caso de trabajar con colecciones de prueba en las que hay una batería de consultas con sus correspondientes documentos juzgados como relevantes por quien las efectuó (Korfhage, 1997).

Los criterios mencionados anteriormente son de cierta manera mesurables y se centran principalmente en el punto de vista del diseñador del sistema. Sin embargo, se debe considerar también el punto de vista del usuario ya que los criterios de evaluación de diseñador y usuario no tienen por qué coincidir (Salton and McGill, 1983). Atendiendo a esto, la facilidad de uso (usabilidad) del SRI por parte del usuario es una característica importante a tener en cuenta. Esto se resume en el esfuerzo intelectual o físico, requerido por el usuario en la formulación de las consultas; en el manejo de la búsqueda, en el proceso de examinar los resultados y en la forma de presentación de los resultados de la

búsqueda, pues influye en la potencialidad del usuario para utilizar la información recuperada (Lancaster, 1979).

1.5 Métodos para mejorar la recuperación

Hay ocasiones en que el usuario no es capaz de encontrar una consulta que exprese fielmente su necesidad de información o en las que, por limitaciones del propio SRI, este no consigue recuperar todos los documentos relevantes. Por otro lado, se puede hacer referencia a un mismo concepto usando diferentes palabras, suceso conocido como sinonimia y que tiene un impacto considerable en la exhaustividad de la mayoría de los SRI. Por estas y otras razones se han desarrollado procedimientos que permiten asistir al usuario a la hora de formular la consulta por un lado, y por otro, reformular la consulta de manera iterativa a la luz de los juicios de relevancia expresados por este.

Estos procedimientos se dividen fundamentalmente en dos clases: métodos globales y métodos locales. Los *métodos globales* son técnicas para expandir o reformular los términos de la consulta independientemente de la consulta como tal y los resultados obtenidos a partir de la misma, de manera tal que los cambios en la redacción de la consulta causen que la nueva consulta empareje con otros términos semánticamente similares. Los métodos globales incluyen (Manning et al., 2008):

- Expansión o reformulación de consultas mediante el uso de tesauros.
- Expansión de consultas mediante la generación automática de tesauros.
- Técnicas como corrección ortográfica.

Los *métodos locales* utilizan solo la información obtenida a partir de la lista de documentos devuelta por una primera consulta. Estos métodos incluyen (Manning et al., 2008):

- Realimentación de relevancia.
- Pseudo-realimentación de relevancia (realimentación de relevancia ciega)
- Realimentación global indirecta.

De todas estas técnicas las más difundidas son el uso de tesauros y la realimentación de relevancia. Por su importancia, se tratarán brevemente en las secciones siguientes.

1.5.1 Uso de tesauros

Entre las técnicas que ayudan al usuario a formular una consulta se destaca el uso de los tesauros: un conjunto de palabras y las relaciones que existen entre sí, las cuales van desde sinonimias y antonimias hasta cualquier otro tipo de relación entre ellas.

El tesauro puede usarlo el propio usuario para expresar su necesidad de información, mediante la búsqueda de las palabras adecuadas o alternativamente, se suele utilizar como fuente para añadir nuevos términos a una consulta, proceso que se conoce como *expansión de consultas*.

1.5.2 Realimentación de relevancia

La *realimentación de relevancia* es la más popular de las estrategias de modificación de consultas. Introducida en el campo de la RI por Rocchio (Rocchio, 1971), es un proceso controlado y automático de definición de consultas, sencillo de utilizar y extraordinariamente efectivo. La idea principal viene dada por la elección de términos importantes ligados a ciertos documentos que previamente se han identificado como relevantes por el usuario, para incrementar su importancia en la nueva formulación de la consulta. Análogamente, se puede disminuir la importancia de los términos incluidos en documentos no relevantes previamente recuperados en la futura formulación de la consulta. El efecto de este proceso es el de “mover” la consulta en la dirección de los documentos relevantes y alejarla de los no relevantes, con la esperanza de recuperar así más documentos deseados y menos documentos no deseados en una búsqueda posterior.

El proceso de realimentación de relevancia consiste en formular una consulta inicial, ejecutarla sobre el SRI, especificar la relevancia de los documentos recuperados, aplicar un mecanismo de realimentación para modificar la consulta inicial y ejecutar la nueva consulta obtenida en el sistema. El proceso se efectúa en línea, requiere la interacción con el usuario (que debe proporcionar los juicios de relevancia ante cada ejecución de una nueva consulta), y es cíclico, pues puede repetirse tantas veces como se desee, modificando sucesivamente la última consulta ejecutada con el objetivo de obtener nuevos documentos relevantes.

1.6 Sistemas de Recuperación de Información Estructurada

Los SRI son comparados frecuentemente con BD relacionales. Tradicionalmente, los SRI utilizan como fuentes de datos texto no estructurado, mientras que las BD son diseñadas para consultar datos relacionales: registros que tienen valores para atributos predefinidos como número de empleado, título, salario, entre otros.

Existen diferencias significativas entre los SRI y los SGBDR en términos del modelo de recuperación, estructuras de datos y lenguajes de consulta. Algunos problemas que requieren consultar texto muy estructurado son manejados más eficazmente por una base de datos relacional, donde una consulta en lenguaje SQL puede ser suficiente para satisfacer una necesidad de información con altos niveles de precisión y exhaustividad. Sin embargo, muchas fuentes de datos estructurados que contienen texto son modeladas como documentos estructurados en lugar de datos relacionales. Estos documentos pueden representar obras literarias, artículos científicos, vídeos anotados, historiales médicos, entre otros. (Manning et al., 2008). Reflexionando brevemente sobre el concepto de documento, se pueden citar múltiples ejemplos en los que, a pesar de poder considerarse este como una unidad indivisible, resulta más natural tratarlo como un conjunto de partes:

- Un libro podría dividirse en capítulos, estos a su vez en secciones y las secciones incluso se podrían estructurar en párrafos. En caso de una obra de teatro tendríamos más divisiones: actos, escenas, discursos y líneas.
- Un artículo científico normalmente consta de resumen, una serie de secciones (cada una pudiendo dividirse en varias subsecciones y así sucesivamente), referencias bibliográficas, agradecimientos, entre otras.
- Historias clínicas de pacientes, donde se suele seguir una estructura rígida en cuanto a las partes que la componen y su orden (Bickley et al., 2005).

A la búsqueda que se realiza sobre tales documentos se le denomina *recuperación estructurada* y los sistemas que la ponen en práctica *Sistemas de Recuperación de Información Estructurada (Manning et al., 2008)*.

Las consultas en la recuperación estructurada pueden ser o estructuradas o no estructuradas y pueden combinar criterios basados en el texto del documento o en la estructura.

El proceso clásico de RI no sirve en su totalidad para trabajar con documentos estructurados. En (Chiaramella, 2001) se tratan aspectos relacionados con el proceso de interacción con el usuario que constituyen verdaderos retos a la hora de diseñar un SRI estructurado. Estos problemas están relacionados con el impacto que tiene la estructuración de los documentos en:

- El proceso de consulta y la presentación de los resultados: el sistema puede presentar como respuesta a una consulta una lista en la que se incluyan dos unidades estructurales, donde una de ellas pueda estar lógicamente incluida en la otra y a su vez distanciadas por su valor de relevancia en el orden final presentado al usuario.
- El proceso de obtención de la relevancia: los modelos clásicos no sirven para obtener los valores de relevancia que se necesitan para este tipo de sistemas, pues existen relaciones entre las estructuras que son ignoradas por estos. En (Trotman, 2005) se profundiza en este aspecto.
- La concentración de la información relevante: En un SRI tradicional, si un documento es altamente relevante para una consulta se puede considerar que el documento contiene en casi su totalidad información interesante para el usuario. En el caso estructurado, toda la información puede estar concentrada en una subunidad incluida en la unidad actual devuelta en la consulta. La calidad del proceso de recuperación depende de la habilidad del usuario para tomar decisiones y postprocesar los resultados del sistema, por lo que la calidad de desempeño del sistema baja considerablemente. En general, son dos los criterios que deben seguirse a la hora de determinar los componentes o unidades relevantes: especificidad en el sentido de que si un documento es relevante solo porque una sección del mismo lo es, es preferible recuperar únicamente dicha sección; por el contrario, si el documento es relevante porque todas las secciones tratan el tema en cuestión, probablemente es preferible recuperar el documento completo y multiplicidad, que permite que diferentes partes de un mismo documento puedan ser recuperadas ordenadas por grado de relevancia.

Además de estos desafíos y no menos importantes se encuentran los relacionados con el proceso de indexación expuestos en (Lee et al., 1996, Shah et al., 2004) y con el proceso de búsqueda presentado en (Trotman, 2004).

Un SRI estructurada debe dotarse de mecanismos para representar las interrelaciones entre componentes estructurales de los documentos, así como hacer uso de las mismas para mejorar la efectividad de la recuperación. En este ámbito, el concepto de documento pierde entidad, deja de ser el único elemento recuperable, ya que el sistema no solo podría devolver documentos completos, sino aquellas partes de estos que se acerquen más a la necesidad de información del usuario en respuesta a una consulta.

Actualmente, el estándar más difundido para representar documentos estructurados es el XML. En el contexto de la recuperación de información, los SRI cuya colección documental está formada por este tipo de documentos se denominan SRI XML y con estos, ha surgido un importante campo de investigación dentro de la RI estructurada que es la RI en documentos XML.

1.6.1 Recuperación de información en documentos XML

Debido a la amplia aceptación del lenguaje XML como estándar para el modelado, almacenamiento e intercambio de datos estructurados, su potencialidad para mezclar información estructurada y no estructurada, así como a las ventajas que ofrece para la RI, este formato de datos constituye el más popular para representar documentos estructurados que formarán la colección documental de un SRI estructurada.

En las secciones siguientes se presentarán los desafíos que tienen lugar en la recuperación de información en documentos XML; las diferencias entre los documentos XML orientados a los datos y orientados al contenido, con sus variantes de recuperación; las formas de presentación de las consultas y finalmente cómo se lleva a cabo la evaluación de la recuperación para este tipo de documentos.

1.6.1.1 Desafíos en la recuperación de información sobre documentos XML

En epígrafes anteriores se abordaron los desafíos que trae consigo la recuperación de información cuando la colección documental está formada por documentos estructurados. En esta sección se tratarán estos problemas desde el punto de vista de los documentos XML y se presentan nuevos retos que tienen lugar para este tipo de documento estructurado.

El primer desafío que presenta la RI a partir de documentos XML es que los usuarios requieren que el sistema devuelva como resultado de sus búsquedas partes de documentos (ejemplo: elementos XML) y no documentos completos como es usual en los SRI clásicos.

Un criterio para seleccionar la parte del documento más apropiada es seguir el principio de recuperación de documentos estructurados: *un sistema siempre debe recuperar la parte más específica de un documento en respuesta a una consulta (Chiaramella et al., 1996)*. Este principio motiva la estrategia de recuperación que devuelve la unidad más pequeña que contiene la información buscada, y no retornar nada por debajo de este nivel. Sin embargo, esto puede resultar difícil de implementar algorítmicamente (Manning et al., 2008).

Paralelo al problema de cuáles partes de un documento retornar al usuario, aparece el problema de cuál parte del documento indexar o simplemente encontrar la *unidad de indexación* correcta.

En la recuperación de información no estructurada seleccionar la unidad de indexación adecuada resulta bastante simple. En cambio, en la recuperación estructurada existen diferentes variantes para definirla.

Un enfoque es agrupar los nodos en pseudo-documentos que no se solapen como se muestra en la figura 1.2. En el ejemplo, libros, capítulos y secciones se han diseñado para que constituyan unidades de indexación. La desventaja de este enfoque es que los pseudo-documentos pueden no tener sentido para el usuario porque son unidades semánticamente incoherentes.

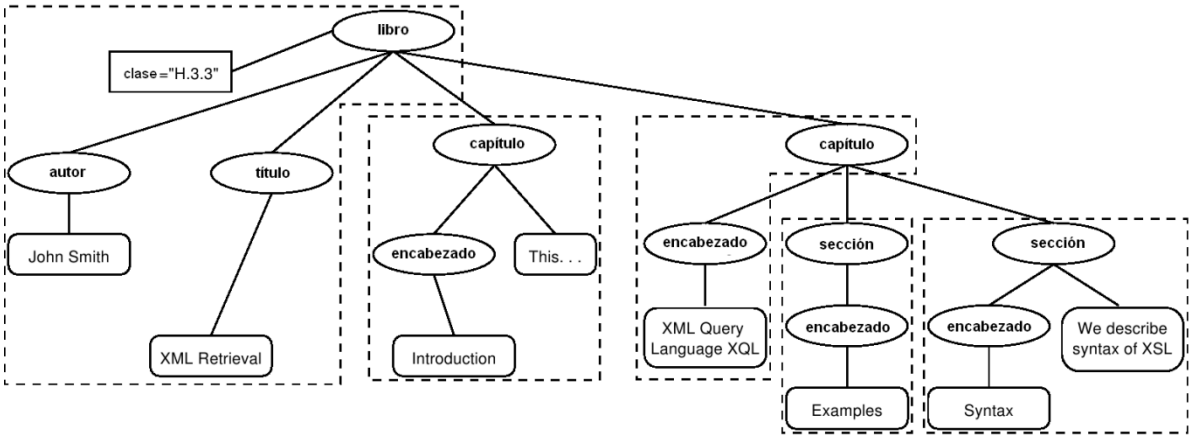

Figura 1.2. Representación de nodos no solapados como unidades de indexación. Figura similar a la presentada en (Fuhr and Großjohann, 2004).

Otro método es usar uno de los elementos más extensos como unidad de indexación; por ejemplo, el elemento libro en una colección de libros. Se pueden postprocesar los resultados de la búsqueda para encontrar para cada libro el subelemento más relevante a la consulta. Desafortunadamente este proceso de recuperación en dos partes falla en retornar el mejor subelemento para muchas consultas, debido a que la relevancia de un elemento en su totalidad no es frecuentemente un buen predictor de la relevancia de los subelementos que lo componen.

En lugar de recuperar unidades extensas e identificar los subelementos relevantes se pueden buscar todos los nodos hojas, seleccionar los más relevantes y propagarlos a elementos más extensos. Este enfoque presenta problemas similares a los del anterior: la relevancia de los nodos hojas no es frecuentemente un buen predictor de la relevancia de los elementos en los que están contenidos.

El método menos restrictivo es indexar todos los elementos. Esto tiene el inconveniente de que muchos elementos XML no tienen importancia para las búsquedas, como elementos tipográficos o números que no tienen significado sin un contexto asociado. Además, indexar todos los elementos podría inducir a que los resultados de las búsquedas sean sumamente redundantes.

El anidamiento de elementos en un documento XML suele igualmente traer problemas de redundancia a la hora de presentar los resultados al usuario y para calcular la relevancia de los términos. En la mayoría de los enfoques, los resultados contienen elementos anidados. Se pueden eliminar algunos elementos en un postprocesamiento para reducir la redundancia o se pueden colapsar los elementos anidados en la lista de resultados y resaltar los términos de la consulta para enfocar la atención del usuario a los fragmentos de texto relevantes. Otro desafío relacionado con el anidamiento es que se necesita distinguir contextos diferentes de un término cuando se calculan las estadísticas del mismo para el ordenamiento, en particular la frecuencia documental inversa. Una posible solución para esto es calcular la frecuencia documental inversa para pares término/contexto (Manning et al., 2008).

En muchas ocasiones, tienen lugar esquemas XML diferentes en una misma colección, pues los documentos XML provienen de fuentes diferentes. Este fenómeno se denomina *heterogeneidad o diversidad de esquemas* y representa otro desafío a tener en cuenta en la RI XML. En (Manning et al., 2008) se presenta una explicación detallada del mismo y posibles soluciones.

1.6.1.2 Visiones de los documentos XML

Atendiendo al contenido del documento XML y a su estructura interna, estos pueden clasificarse de dos formas diferentes: vista del documento centrada en los datos y vista del documento centrada en el contenido.

Los documentos que se orientan siguiendo el primer enfoque se usan fundamentalmente entre aplicaciones de empresa como un formato de intercambio para datos estructurados, donde predominan valores numéricos o datos del tipo atributo-valor y el texto representa usualmente una pequeña fracción del total de datos del documento. Frecuentemente son usados como una nueva representación del esquema relacional.

Por otra parte los documentos orientados al contenido utilizan el lenguaje como el formato para representar su estructura lógica y se identifican por presentar textos extensos, como secciones de una obra literaria.

La recuperación de información en los documentos orientados al contenido se caracteriza por un emparejamiento inexacto y un ordenamiento de los resultados. En cambio, en los documentos orientados a los datos, las consultas suelen imponer condiciones exactas de emparejamiento, enfatizando en las características estructurales de los mismos. Dado que los SGBDR están mejor equipados para manejar restricciones estructurales, muchos de los sistemas de recuperación de documentos XML orientados a los datos son extensiones de sistemas de bases de datos relacionales. Una visión más general de la integración de la RI XML y las BD se presenta en (Amer-Yahia and Lalmas, 2006).

1.6.1.3 Tipos de consultas

Una de las ventajas que presentan los SRI estructurada es la posibilidad de incluir diferentes tipos de consultas. Un documento estructurado, específicamente en formato XML, permite refinar la consulta incluyendo algún tipo de información sobre la estructura. Por ejemplo, si se consulta una colección de artículos científicos se

podría restringir la búsqueda a una cierta unidad estructural, como el título o el resumen. De manera general se distinguen dos tipos de consulta para documentos estructurados:

- Consultas de tipo "solo contenido" (CO, content only): equivalentes a las consultas clásicas de los SRI, se construyen solo con términos o palabras claves.
- Consultas de tipo "contenido y estructura" (CAS, content-and-structure): presentan restricciones estructurales en adición a los términos o palabras claves. Este tipo de consultas se puede tratar siguiendo dos enfoques:
  - SCAS (*Strict content-and-structure*): recuperan elementos relevantes que emparejan exactamente con la estructura especificada en la consulta.
  - VCAS (Vague content-and-structure): recuperan elementos relevantes que pueden no ser los mismos que los elementos de la consulta, pero son similares desde el punto de vista estructural; o recuperan elementos relevantes incluso si no cumplen con las condiciones estructurales, tratan la especificación de la estructura en la consulta como una sugerencia para la búsqueda.

Frecuentemente, asociado a las consultas de tipo CAS, se presenta un lenguaje específico de consulta. Un modelo para obtener un lenguaje de consulta por contenido y estructura puede encontrarse en (Navarro and Baeza-Yates, 1997). Otra propuesta de lenguaje de consulta, ampliamente utilizado, es el NEXI (Narrowed Extended XPath I), propuesto en (Trotman and Sigurbjörnsson, 2005).

1.6.1.4 Evaluación de la recuperación de información en documentos XML

El principal punto de encuentro para la comunidad que investiga en el área de la RI en documentos XML es el programa INEX (INitiative for the Evaluation of XML retrieval) un esfuerzo colaborativo que ha producido colecciones de referencia, conjuntos de consultas y juicios de relevancia para la evaluación de los SRI XML.

En el año 2002, la colección INEX estaba compuesta por 12,000 artículos de las publicaciones de la IEEE y fue expandida en el 2005. Desde el 2006 INEX usa como colección de prueba la Wikipedia en Inglés, la cual es mucho más extensa. Más información sobre esta iniciativa se puede consultar en (Fuhr and Lalmas, 2007, Gövert and Kazai, 2002).

1.7 Integración de las áreas de Bases de Datos y Recuperación de Información.

En las últimas décadas se ha incrementado la percepción de la necesidad de integrar las tecnologías de BD y RI (Amer-Yahia et al., 2005, Chaudhuri et al., 2005). Desde el punto de vista de la RI, bibliotecas digitales de todo tipo se han convertido en repositorios muy ricos en información, con documentos enriquecidos con metadatos y anotaciones, plasmados en formatos de datos semiestructurados como XML; por otro lado, la explotación de la estructura de los documentos constituye un aspecto crítico de la búsqueda en la Web. Desde el punto de vista de las BD, áreas de aplicación tales como soporte al cliente, investigación de mercado y los sistemas encargados de apoyar el mantenimiento y el restablecimiento de la salud, entre otros, exhiben enormes crecimientos de datos en términos de información estructurada y no estructurada, por lo que el uso de bases de datos convencionales resulta insuficiente para su correcta explotación (Weikum, 2007).

Existen muchas posibilidades para la integración tales como extender un modelo de BD para que incorpore eficientemente el uso de las probabilidades, extender un modelo de RI que maneje estructuras más complejas y relaciones múltiples, o desarrollar un modelo y sistema unificados. Las aplicaciones que se basan en la búsqueda en la Web, comercio electrónico y minería de datos, proporcionan las bases de experimentación para que las propuestas que se encaminen en dicho sentido sean evaluadas y comparadas (Bruce Croft and Schek, 2008).

1.8 Consideraciones finales

A manera de resumen se puede afirmar que el funcionamiento de los SRI se sintetiza en: representar el problema de necesidad de información del usuario (consulta); representar y organizar el contenido de la fuente de conocimiento; comparar la consulta con los componentes del contenido y por último, presentar los resultados al usuario para que interactúe o los juzgue.

Investigaciones y debates recientes han mostrado la importancia que tiene para el mundo de hoy contar con sistemas de búsquedas que fusionen las capacidades de consulta de BD y RI y los esfuerzos que se han hecho en función de su integración.

CAPITULO II. MODIFICACIONES REALIZADAS AL SISTEMA DE RECUPERACIÓN DE INFORMACIÓN GOOMED

La implementación de la nueva versión del SRI propuesto en esta investigación requirió un estudio profundo de las tecnologías existentes relacionadas con el desarrollo Web, la RI y los SGBDR. En el presente capítulo se describen y caracterizan dichas tecnologías, analizándose sus potencialidades en su desempeño propio, así como su capacidad de integración para dar solución al problema planteado de forma eficiente.

Se describen las nuevas prestaciones agregadas al sistema tales como la inclusión de la búsqueda por términos frecuentes, a través de los cuales los usuarios pueden interactuar con el sistema de forma simple y eficiente. La creación de perfiles de usuarios que permitan en un futuro mejorar la calidad de la recuperación y la integración con la enciclopedia médica MedlinePlus, la cual constituye uno de los repositorios de artículos médicos de mayor prestigio a nivel internacional.

2.1 Introducción

En la actualidad la creación de aplicaciones web es un proceso complejo desde distintos puntos de vista. Los desarrolladores constantemente buscan métodos para brindar al usuario un mejor servicio, permitiéndole un control cada vez mayor sobre sus datos y libertad de navegación; ofreciéndole a los administradores facilidades para su gestión dentro de la aplicación en lo que se refiere a privacidad, seguridad y control de los datos. La construcción de aplicaciones web que cumplan estos requisitos se ve enriquecida con la retroalimentación de los usuarios, lo que brinda mayor grado de adaptabilidad, cuyo beneficiario final es el propio usuario. El logro de esta variedad de facilidades es alcanzado mediante el uso de diferentes filosofías de creación, las cuales cuentan con un uso intensivo de patrones de diseño y de arquitectura que se traducen en funcionalidad, ejemplo de ello lo son los Sistemas Orientados a la Administración de Contenido (CMS, siglas en inglés) (2010b) los marcos de trabajo (frameworks) para aplicaciones web y los marcos de trabajo para aplicaciones web basados en Desarrollo Rápido de Aplicaciones (RAD, siglas en inglés) (2010k) orientados al almacenamiento de la funcionalidad en bibliotecas para ser utilizadas por los programadores en la construcción de

aplicaciones web y las aplicaciones híbridas que toman ideas de las filosofías anteriores, generalmente de propósito más específico.

La satisfacción del usuario es un aspecto medular a la hora de la construcción de un SRI con las características del problema que se plantea en esta investigación. El sistema debe brindar la posibilidad al usuario de expresar sus necesidades de información y además debe ser capaz de satisfacerlas en la medida de lo posible. Debe ser estable, rápido, simple, eficiente y acoplable a ambientes donde se apliquen técnicas para incrementar la eficiencia.

2.2 Descripción de las herramientas utilizadas

En la nueva versión del SRI se utilizaron herramientas que se caracterizan por su alto grado de rendimiento, escalabilidad, facilidad de uso e integración a diferentes entornos y plataformas de desarrollo.

En la reestructuración de las páginas web dinámicas se mantuvo el uso de Zend Framework (ZF) (2011l) La manipulación de la capa de abstracción de datos (DBAL, siglas en inglés) (2010d), o sea, de la componente estructurada de la información, quedó a cargo del Mapeador Relacional de Objetos (ORM, siglas en inglés) Doctrine (2011b), el cual opera en concreto sobre MySQL (2011i) en el sistema implementado, brindando por sus características, posibilidades de adaptabilidad a otros SGBD. La plataforma de búsqueda de texto avanzada Zend_Search_Lucene (2011m) presente en el Zend Framework, se utilizó para la recuperación de información sobre los artículos de la enciclopedia médica MedlinePlus (2011g). Además se reutilizó el framework para Javascript JQuery (2011e), capaz de crear junto con HyperText Markup Language (HTML) (2010g) interfaces web atractivas y altamente funcionales; requerimiento esencial para la aplicación. A continuación se exponen las características principales de cada una de las herramientas mencionadas anteriormente.

2.2.1 Zend Framework

ZF es un framework para aplicaciones web de código abierto que se utiliza para desarrollar aplicaciones y servicios web con PHP versión 5. Es una implementación que usa código 100% orientado a objetos y la estructura de sus componentes, pues requiere una baja dependencia entre los mismos. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. A

menudo se refiere a este tipo de diseño como “uso a voluntad” (use-at-will, siglas en inglés).

Los componentes de la biblioteca estándar de ZF, aunque se pueden utilizar de forma individual, conforman un potente y extensible framework de aplicaciones web al combinarse. ZF ofrece un gran rendimiento y una robusta implementación del Modelo Vista Controlador (MVC) (2011h). MVC es un patrón de arquitectura de software con el objetivo de separar la lógica de la aplicación de la lógica de visualización. ZF también ofrece una abstracción de base de datos fácil de usar, y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos. Otros componentes como Zend_Auth y Zend_Acl, proveen autenticación y autorización de usuarios sobre los diferentes servicios existentes en la Web. También existen componentes que implementan bibliotecas de tipo cliente para acceder de forma sencilla a los servicios web más populares. Cualesquiera que sean las necesidades, se tienen todas las posibilidades de encontrar un componente de ZF que se pueda utilizar para reducir drásticamente el tiempo de desarrollo, con una base completamente sólida.

ZF debe ejecutarse en el contexto de un servidor web que soporte PHP como por ejemplo el Servidor Web Apache (Foundation, 2011a), el cual fue utilizado en la elaboración y posterior modificación del SRI implementado.

2.2.1.1 Componentes más importantes

A continuación se describe el funcionamiento y las características principales de Zend_Controller, Zend_View, Zend_Registry, Zend_Auth, Zend_Config y Zend_Error, pues constituyen junto con Zend_Db los componentes que proveen a ZF de su principal potencialidad. Zend_Db no es tratado entre los componentes principales pues en su lugar se utilizó Doctrine, el cual cumple en mayor medida los requisitos a tener en cuenta a la hora de la implementación del SRI.

Zend_Controller

Zend_Controller es el componente principal del sistema de MVC de ZF. Zend_Controller_Front implementa el patrón Controlador Frontal (Front Controller Pattern, siglas en inglés) (2011c) en el cual todas las transacciones sobre

HyperText Transference Protocol (HTTP) (2010h) pedidos (requests) son interceptadas por el controlador frontal y enviado a una acción particular de un controlador según el Uniform Resource Locator (URI-URL) (2011k) el pedido, es aquí donde entran en juego las convenciones de nombrado de ZF, brindándole a este componente la inmediata detección de las clases a través del mecanismo de mapeo nombre de clase – camino a la clase dentro de la aplicación.

El componente Zend_Controller brinda una gama de posibilidades en cuanto a extensibilidad se refiere. A través de la herencia o implementación de interfaces expuestas dentro del paquete del componente se puede extender su funcionalidad principal. Otra forma de ampliar la funcionalidad de este componente y de la aplicación en general, es el empleo de plugins y helpers, pilares de la interacción entre componentes de ZF.

La figura 2.1 muestra la organización de directorios de la aplicación Goomed v2.0: SRI obtenido en esta investigación.

1. La raíz web de la aplicación está ubicada en la carpeta public, en esta carpeta se encuentran los elementos que se desean compartir; o sea, archivos CSS, Javascript, imágenes, entre otros. Además se encuentra el punto de entrada de la aplicación web junto con el archivo de directivas de configuración de Apache htaccess para la definición de los permisos usando las posibilidades de configuración que brinda el módulo rewrite del servidor mencionado, este último junto con el motor de enrutamiento de ZF constituyen la lógica de funcionamiento del procesamiento de pedidos con URLs limpios, así como la administración de seguridad.
2. La implementación de la aplicación como tal se encuentra en la carpeta application, dentro de la cual se encuentran directorios para la ubicación de módulos y configuración principalmente.

Cada módulo tiene como organización de directorios la estructuración típica para el patrón MVC, o sea, contiene directorios para los controladores, modelos y vistas.

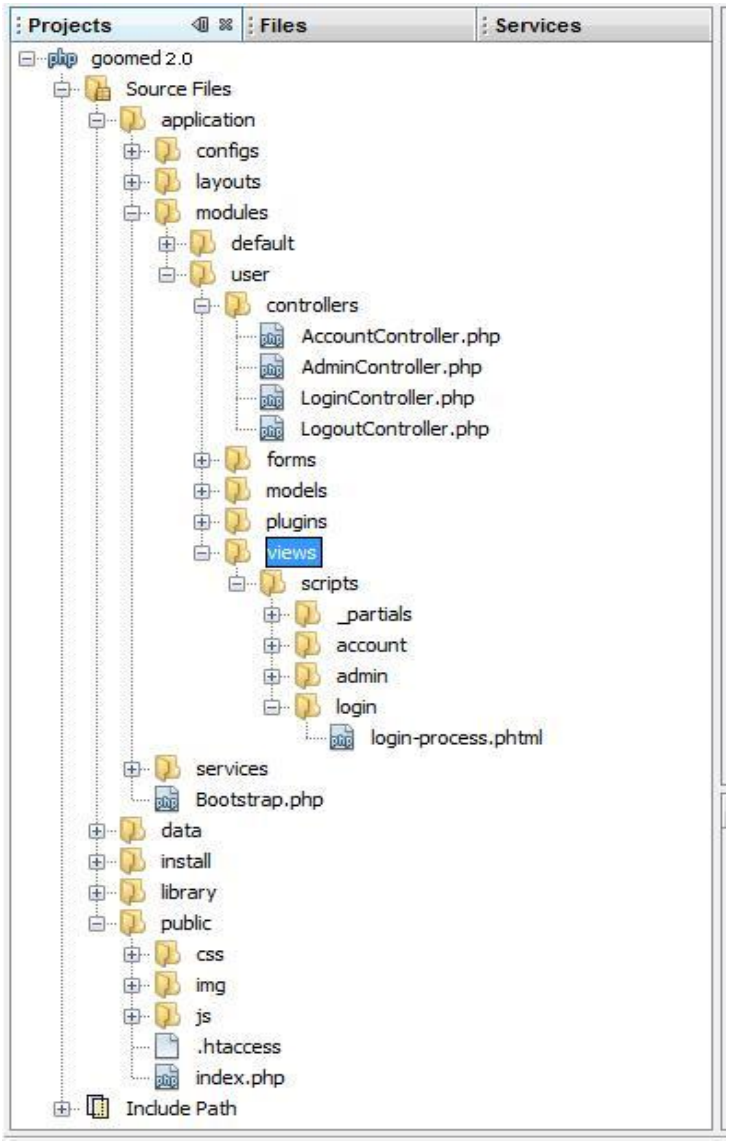

Figura 2.1. Sistema de archivos de la aplicación Goomed 2.0.

- El despachador del controlador frontal

Por defecto el despachador de ZF interpreta el camino pedido de la forma "/:módulo/:controlador/:acción/:parámetro1:/valor1/parámetro2/:valor2…", de manera que si se omiten elementos a la derecha se toman los valores por defecto.

Los valores por defecto son:

- para los parámetros: el valor vacío
- para la acción: el valor index (que mapea a la acción *indexAction*)
- para el controlador: el valor index (que mapea al controlador *IndexController*)
- para el módulo: el valor default

De manera tal que si el URI es "/" entonces ZF interpreta el pedido como "/default/index/index/".

ZF establece para el enrutador por defecto una relación entre el URI del pedido y el camino al archivo en el sistema de archivos dentro de la raíz de la aplicación, de manera que un pedido hace referencia a un módulo, controlador y acción específicos. Sin embargo, si se quieren agregar restricciones al proceso de enrutamiento se pueden utilizar una serie de enrutadores avanzados de ZF que permiten el uso de expresiones regulares y reglas de direcciones de IP entre otros.

Zend_View

Zend_View es la clase que permite el trabajo con la Vista en el patrón MVC. Existe con el objetivo de mantener la lógica de visualización separada del modelo y los controladores. Provee un amplio sistema de ayudantes (helpers) y filtros para formateo de salida, que en un final ayudan a la hora de la creación de las distintas vistas dinámicas de la aplicación.

El sistema de helpers cuenta con ayudantes de formularios los cuales permiten crear componentes de formularios de una forma más sencilla, con directivas de inclusión (partials) que permiten la inserción de fragmentos de vistas dentro del ámbito de otras vistas, con fijadores de contenido (placeholders) que permiten la persistencia de contenido entre las vistas, entre otros; todos funcionan en armonía utilizando Zend_Layout para envolver las vistas de toda la aplicación con capas (layouts) que funcionan como plantillas globales.

Zend_View es un sistema de plantillas adaptable a cualquier lenguaje. Se puede usar PHP como lenguaje de plantillas o crear instancias de cualquier otro sistema de plantillas a través de una serie de interfaces y clases abstractas que brinda este componente y utilizarlo indistintamente en la aplicación sin cambiar el código.

El proceso por el que pasa la vista consiste en dos pasos fundamentales:

1. El controlador crea una instancia de Zend_View y le asigna valores a las variables de dicha instancia.
2. El controlador manda a mostrar una vista en particular y le pasa el control para que genere la salida en HTML.

Zend_Registry

Esta clase representa un contenedor de registros para almacenar objetos y valores en el ámbito de la aplicación. Una vez guardado un valor en el contenedor de registros es persistente a lo largo del proceso de manipulación del pedido HTTP, además es accesible desde cualquier parte de la aplicación. Usualmente es utilizado para guardar los objetos de configuración u otros datos que pueden ser leídos o escritos desde distintos ámbitos.

Zend_Config

Este componente está diseñado para simplificar el acceso y el uso de datos de configuración dentro de las aplicaciones. Provee una interfaz de usuario basada en propiedades de objetos anidadas para acceder a datos de configuración dentro del código de la aplicación. Los datos de configuración pueden venir de multitud de medios que soporten almacenamiento de datos de forma jerárquica. Actualmente Zend_Config provee adaptadores para datos de configuración que están almacenados en archivos con formato INI (Windows Initialization file) (2010i), con Zend_Config_Ini y archivos con formato XML con Zend_Config_Xml.

Zend_Auth

Zend_Auth provee una Interfaz de Programación de Aplicaciones (API, siglas en inglés) para autenticación e incluye adaptadores concretos de autenticación para escenarios de casos de uso común.

Zend_Auth se hace cargo de lo concerniente sólo con la *autenticación* y no con *autorización* de los usuarios. *Autenticación* es vagamente definido como: determinar si una entidad realmente es lo que pretende ser (o sea, identificación), basándose en un grupo de credenciales. *Autorización* es el proceso de decidir si se permite a una entidad acceso a realizar operaciones en otras entidades, esta funcionalidad está fuera del alcance de Zend_Auth.

Zend_Exception y Error_Controller

ZF tiene un mecanismo especial para el control de los errores. Las excepciones no tratadas explícitamente en el código que son lanzadas mientras se procesa el pedido HTTP son automáticamente capturadas por el Controlador Frontal de ZF, éste a su vez redirecciona el pedido al controlador de errores (Error_Controller) con los detalles de la excepción para ser mostrados al usuario. De esta manera no

queda en manos del programador la manipulación de excepciones estándares de tratamiento común. Las excepciones que no se quieran dejar a cargo del controlador de errores pueden ser tratadas internamente para dar un tratamiento especial al error.

Zend_Search_Lucene

Zend_Search_Lucene (2011m) es un motor de búsqueda de texto de propósito general escrito completamente en PHP 5. Almacena sus índices en el sistema de archivos y no requiere un servidor de base de datos. Puede añadir capacidades de búsqueda a casi cualquier sitio web PHP. Usa la biblioteca de RI Lucene como su componente principal para la indexación y búsqueda de textos.

Zend_Search_Lucene soporta las siguientes características:

- Búsqueda por Ranking:
  - Mostrará al principio los mejores resultados.
- Muchos tipos de consulta de gran alcance:
  - Las consultas de frases, las consultas booleanas, comodines en una consulta, las consultas por proximidad, rangos y muchas otras.
- Búsqueda por un campo específico:
  - Por ejemplo, título, autor, contenidos.

Zend_Search_Lucene se derivó del proyecto Apache Lucene. En la actualidad (comenzando por ZF 1.6) soportan el formato del índice de las versiones de lucene de 1.4 - 2.3.

Opera con documentos como objetos atómicos para la indexación. El documento se divide en campos con nombre, y los campos tienen un contenido que se puede buscar.

Un documento es representado por la clase Zend_Search_Lucene_Document y los objetos de esta clase contienen instancias de Zend_Search_Lucene_Field que representan los campos en el documento.

Es importante tener en cuenta que cualquier información se puede añadir al índice. Información específica de la aplicación o metadatos pueden ser almacenados en los campos del documento, y más tarde recuperados con el documento durante la búsqueda.

Es responsabilidad de la aplicación controlar el indexado. Esto significa que los datos puedan ser indexados de cualquier fuente que sea accesible por la aplicación. Por ejemplo, este podría ser el sistema de archivos, una base de datos, un formulario HTML, etc.

No requiere ninguna extensión de PHP o software adicional para ser instalado, y puede ser utilizado inmediatamente después de ser instalado Zend Framework. Zend_Search_Lucene es PHP puro, puerto del popular motor de búsqueda de código abierto conocido como Apache Lucene.

La información debe ser indexada para estar disponible para la búsqueda. Zend_Search_Lucene y Java Lucene utilizan el concepto de documento conocido como "elemento de indexación atómica".

Cada documento es un conjunto de campos: <nombre, valor>pares donde nombre y valor son cadenas UTF-8. Cualquier subconjunto de los campos del documento puede ser marcado como "indexado" para incluir los datos del campo en el proceso de indexación de texto.

Los valores del campo pueden o no ser tokenizados durante la indexación. Si un campo no se tokeniza, entonces el valor del campo se almacena como un término, de lo contrario, el analizador actual se utiliza para tokenización. Varios analizadores se proporcionan en el paquete Zend_Search_Lucene. El analizador por defecto funciona con texto ASCII (ya que el UTF-8 analizador necesita la extensión *mbstring* para que esté activada). Utilice otros analizadores o cree su analizador propio si necesita cambiar este comportamiento.

Las consultas de búsqueda también son tokenizadas con el "analizador actual", por lo que el mismo analizador se debe establecer como predeterminado, tanto durante el proceso de indexación y búsqueda. Esto garantizará que la fuente y el texto buscado se transformen en términos de la misma manera.

Los valores del campo son opcionalmente almacenados en un índice. Esto permite que los datos originales del campo se recuperen del índice durante la búsqueda. Esta es la única manera de asociar los resultados de búsqueda con los datos originales (los identificadores internos del documento pueden ser cambiados después de la optimización del índice o la optimización automática).

Lo que se debe recordar es que un índice de Lucene no es una base de datos. No proporciona mecanismos de copia de seguridad para el índice con la excepción de la copia de seguridad del directorio del sistema de archivos. No proporciona mecanismos transaccionales aunque la actualización de índices concurrentes, así como la actualización simultánea y la lectura son compatibles. No se puede comparar con bases de datos en cuanto a velocidad de recuperación de datos.

Se recomienda no utilizar el índice Lucene como dispositivo de almacenamiento, ya que podría reducir drásticamente el éxito de la búsqueda disminuyendo el rendimiento. Conservar sólo los identificadores únicos del documento (camino a los documentos, las direcciones URL, identificadores únicos de la base de datos) y los datos asociados en un índice. Por ejemplo título, comentario, categoría, la información del idioma.

Los documentos individuales en el índice puede tener sistemas totalmente diversos de campos. Los mismos campos en diferentes documentos no deben tener los mismos atributos. Por ejemplo un campo puede ser indexado para un documento y saltado de la indexación por otro. Lo mismo se aplica para el almacenamiento, tokenización, o el tratamiento del valor del campo como una cadena binaria.

Utilizando esta librería de Zend Framework fue posible la recuperación de información sobre los artículos de la enciclopedia médica MedlinePLus.

2.2.1.2 Aplicaciones basadas en ZF

Una aplicación basada en ZF consta de 3 partes fundamentales:

1. La configuración, que es donde se definen los datos del SGBDR, directorios, opciones por defecto, entre otras. Queda a conveniencia del desarrollador qué sección de su aplicación desea hacer configurable.
2. El iniciador (bootstrapper) es la clase que se encarga de inicializar los diferentes componentes de la aplicación como el motor del ORM, el motor de vistas, el control de errores y trazas, el registro y el enrutador principal, teniendo en cuenta el orden de dependencias lógico.
3. Los módulos, que se encargan de realizar la funcionalidad fundamental del sitio. Son ellos los encargados de recibir y analizar los pedidos, construir la respuesta y enviarla al cliente.

2.2.2 Doctrine

Doctrine es un ORM para PHP versión 5.2.3+ orientado a objetos se basa en el patrón de diseño Registro Activo (Active Record Pattern) (2010a), situado en la cima de una potente capa de abstracción de datos que permite el acceso de casi todos los SGBD conocidos a través de un lenguaje de consultas llamado Lenguaje de Consultas de Doctrine (DQL, siglas en inglés) (2010e), inspirado en el famoso Lenguaje de Consultas de Hibernate (HQL, siglas en inglés) (2010f), que internamente es la representación de la consulta en el lenguaje de consultas SQL.

Escribir consultas explícitamente en la mayoría de los casos no es necesario pues Doctrine con su potente implementación orientada a objetos manipula las relaciones de entidades a través de las relaciones definidas entre objetos de forma automática, así como todo tipo de consultas.

2.2.3 JQuery

jQuery (2011e) es una biblioteca o framework de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC, es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. Una de las principales características de JQuery es que casi todos los efectos que podemos realizar tienen efecto deslizante o efecto opacidad, lo cual hace que los sitios sean más elegantes y les dé un toque de profesionalidad, según como se esté utilizando.

Se centra principalmente en la presentación de la información y el atractivo visual mejorando así la usabilidad y la capacidad de respuesta del sitio. Después de todo, esto es exactamente lo que la mayoría de sitios web quieren desesperadamente: presentar su contenido para el usuario de una manera fácil y visualmente agradable. Con uso de esta biblioteca no se está ligado a ninguna de las

estructuras predefinidas de HTML (2010g) o CSS (2010c), es muy útil para mejorar un sitio web existente con las grandes posibilidades que las técnicas modernas de JavaScript tienen para ofrecer. Esta es esencialmente la idea de "mejora progresiva", que es un patrón de diseño común en la actualidad. Se puede diseñar libremente la apariencia del sitio o simplemente se puede utilizar cualquiera de los diseños presentados en sus demostraciones como una plantilla.

Las posibilidades que ofrece esta librería son infinitas y los efectos que se pueden lograr son impresionantes. Con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

2.2.4 SOLR

SOLR es una plataforma empresarial de búsqueda. Entre las principales ventajas que presenta esta plataforma de código abierto se incluyen su alta escalabilidad, potencialidades de búsqueda de texto y las facilidades que ofrece para la búsqueda por facetas, el resaltado de los términos de la consulta en los textos de los resultados, la clusterización dinámica, y replicación de índices, la integración con las bases de datos y el manejo de documentos en formato de texto enriquecido como documentos Word y PDF. Está implementada en el lenguaje de programación Java (2010j), y debe ejecutarse en un Servidor de Aplicaciones (Servlet Container) como el Apache Tomcat (Foundation, 2011b). SOLR contiene APIs de interacción estilo REST (2011j) de HTTP/XML y JSON (2011f) , esto facilita su utilización desde otros lenguajes de programación. La potente configuración externa del SOLR permite ser manipulada y adecuada por cualquier aplicación sin la necesidad de escribir código Java, además SOLR contiene una plataforma extensa de complementos (plugins) que permite añadir funcionalidad avanzada o personalizada si es necesario.

2.2.4.1 Conceptos básicos de SOLR

Documento (Document): Un documento es una colección de campos. Se puede considerar como una sección de datos almacenados virtualmente, dígase una página web, un correo electrónico o un archivo de texto; o sea, se representa mediante un documento cualquier información que quiera ser recuperada. Los campos de los documentos almacenan los datos y los metadatos del origen de información. Un documento Word o XML, es irrelevante para el SOLR. Los datos y

metadatos de los mismos son indexados y almacenados en campos separados que conforman un documento dentro del índice el sistema.

Campo (Field): Es la unidad componente de un documento. Cada documento almacenado en el índice contiene uno o varios *campos*, los cuales constituyen una sección de datos que puede ser encuestada o devuelta desde el índice como parte del mecanismo de recuperación durante la ejecución de una consulta.

2.2.4.2 Consultas en SOLR

El mecanismo de interacción con las APIs estilo REST del SOLR requiere se haga un pedido HTTP/XML POST o GET a una URL determinada. Dicho pedido debe seguir la estructuración que se muestra a continuación para lograr el propósito deseado:

- Para agregar documentos al índice: Se debe hacer un HTTP POST al URL "http://solrserver:port/update", la sintaxis del pedido debe ser como se muestra en la figura 2.3.

```
<add>
    <doc boost="2">
      <field name="article">05991</field>
      <field name="title">Apache Solr</field>
      <field name="subject">An intro...</field>
      <field name="category">search</field>
      <field name="category">lucene</field>
      <field name="body">Solr is a full...</field>
    </doc>
</add>
```

Figura 2.2. Sintaxis XML de una consulta de agregado.

- Para eliminar documentos del índice: Se debe hacer un HTTP POST al URL "http://solrserver:port/update", si se desea eliminar un único documento por id se puede lograr como en muestra el ejemplo de la figura 2.4.

```
<delete><id>05591</id></delete>
```

Figura 2.3. Sintaxis XML de una consulta de eliminación.

- Para eliminar varios documentos: Se debe hacer un HTTP POST al URL "http://solrserver:port/update" y se debe especificar la dupla de

<NombreCampo>:<ValorCampo>que se desea borrar dentro de una consulta de eliminación como se muestra en la figura 2.5.

```
<delete>
  <query>manufacturer:microsoft</query>
</delete>
```

Figura 2.4. Sintaxis XML de una consulta de eliminación de varios documentos.

- Para aplicar cambios al índice: Igualmente se debe hacer un POST al URL "http://solrserver:port/update". El proceso de hacer visible cualquier cambio al índice se hace a través de la acción commit como se muestra en la figura 2.6, esto convierte en obligatorio la ejecución de pedidos de este tipo para el correcto funcionamiento de cualquier consulta que modifique dicho índice.

```
<commit />
```

Figura 2.5. Sintaxis XML de una consulta commit.

Al igual que commit existe la acción optimize, la cual realiza las mismas operaciones pero además hace un proceso complejo de optimización del índice, el cual, entre otras cosas, fusiona todos los archivos dispersos del índice en uno solo. La sintaxis del pedido debe ser como se muestra en la figura 2.7.

```
<optimize />
```

Figura 2.6. Sintaxis XML de una consulta optimize

Sintaxis de las consultas

En la tabla 2.1 se muestra a través de ejemplos la sintaxis de consultas que soporta SOLR.

Ejemplo de expresión	Descripción
Insuficiencia	Busca los documentos que contengan

	insuficiencia.
insuficiencia–renal	Busca los documentos que contengan insuficiencia y que no contengan renal.
“Insuficiencia respiratoria aguda”	Busca los documentos que contengan la frase Insuficiencia respiratoria aguda
“insuficiencia aguda” ~1	Busca los documentos que contengan los términos insuficiencia y aguda separados por 1 palabra
respi*	Busca los documentos que contengan palabras que comienzan con “respi”, como respiratorio o respiración
sepsis~	Busca los documentos que contengan palabras parecidas a “sepsis”, como por ejemplo “sépticos”.

Tabla 2.1. Ejemplos de expresiones de consulta.

2.2.4.3 SOLRPhpClient

SOLRPhpClient (2010l) es un cliente de SOLR orientado a objetos para PHP versión 5.2.x, el cual permite la creación y ejecución de consultas de selección, actualización y eliminación de documentos pertenecientes al índice de la base documental administrado por SOLR. Brinda también funcionalidades de crear y optimizar un índice, traduciendo la consulta realizada a formato XML o JSON para de esta forma interactuar con SOLR utilizando sus APIs al estilo REST.

2.3 Aportes de la nueva versión

A continuación se describen las nuevas prestaciones agregadas al sistema.

2.3.1 Perfiles de usuario

Un perfil es el modelado de un objeto en forma compacta mediante sus características primordiales. En el caso de un perfil de usuario de un sistema de software, este puede comprender tanto datos personales y características del sistema computacional, como también patrones de comportamiento, intereses personales y preferencias. Este modelo de usuario está representado por una estructura de datos adecuada para su análisis, recuperación y utilización. En

términos computacionales: *un perfil de usuario es la representación de un conjunto de características que describen a una persona, en su rol de usuario de algún sistema adaptativo.*

Estos perfiles guardan la información personalizada de cada uno de los usuarios que utilizan el sistema. Los aspectos que se deben tener en cuenta para el desarrollo de perfiles de usuario son:

1. ¿Cuál es la información relevante?

   La información relevante en Goomed v2.0 relacionada con los perfiles de usuario son las búsquedas realizadas por los mismos en su uso diario, de manera tal que cada usuario tiene su historial de consultas almacenado en la base de datos.

   También son importantes los documentos devueltos por el sistema tras efectuar una y los juicios de relevancia que proporciona el usuario.

2. ¿Cómo obtenerla?

   La información relevante es obtenida a través de los usuarios al acceder a los documentos de interés devueltos por una consulta, realizando un click sobre los mismos. Estos documentos son guardados en la base de datos como relevantes.

3. ¿Cómo representarla?
   La información obtenida anteriormente se representa tanto en la búsqueda simple como en la avanzada a través de los términos frecuentes, los cuales utilizan información referente a las consultas realizadas por los usuarios para así poder obtener los términos más buscados en el sistema.

4. ¿Cómo mantenerla actualizada?
   La información de los usuarios se mantiene actualizada a través de su perfil, donde cada uno al acceder a su sesión tiene registrado en la base de datos todas las búsquedas realizadas así como los documentos devueltos por la misma a los cuales el usuario haya accedido.

5. ¿Qué métodos de recuperación implementar?

Se emplean varios métodos para recuperar información. Uno de ellos se utiliza para recuperar información de los documentos referentes a las autopsias en formato xml y otro es utilizando las funcionalidades del SGBDR, del cual se obtienen los datos del paciente cadáver conformando así la historia clínica de la autopsia combinado ambos tipos de información.

De esta forma el usuario puede acceder a estos documentos quedando así registrados los relevantes a la búsqueda realizada que hayan accedido.

6. ¿Cómo utilizar esa información para adaptar el sistema en forma automática?

   Esa información es utilizada en estos momentos para la representación de los términos frecuentes. Pero el uso fundamental a la que está destinada es para su integración con otras herramientas propuestas para el desarrollo posterior del sistema que permitan en un futuro mejorar la calidad de la recuperación.

A partir de la evaluación de esta primera versión del sistema se planteó la importancia de su extensión mediante una base de casos de consultas realizadas por los usuarios y los juicios de relevancia brindados por los mismos. Para esta nueva implementación se consideró fundamental tener en cuenta la actividad de cada usuario en particular, manteniendo así actualizada su historia en cuanto a consultas realizadas, de modo que permitan en un futuro mejorar la calidad de la recuperación.

Cada usuario tiene registrado en una tabla en la base de datos un historial de consultas formuladas por ellos en su uso diario, así como los juicios de relevancia que ofrecen para cada una de ellas. Esto contribuye además a medir el grado de satisfacción de estos usuarios y de esta forma contar con otra variante de evaluación para el SRI obtenido.

2.3.2 Búsqueda por términos frecuentes

Una alternativa para la búsqueda simple es la búsqueda por términos frecuentes. Cuando la información se utiliza correctamente se puede interactuar con los usuarios, optimizando la web y orientándola para satisfacer todas sus necesidades y de esta forma alcanzar los objetivos propuestos. Además, es fundamental ir

evolucionando con la web y testear diferentes opciones, analizando cada uno de los resultados, para ir perfilando la mejor forma de hacer llegar los mensajes.

Es importante, en este sentido, que el administrador revise los contenidos de las búsquedas realizadas por los usuarios referidas a los temas abordados en el sitio web, para determinar las nuevas palabras que podrían llevar a estos usuarios a buscar con dichos términos. Gracias a esto, se podrá modificar o mejorar sus contenidos para que los nuevos términos también permitan que más usuarios lleguen a la misma conclusión tras una búsqueda.

Asimismo es posible tener información acerca de las palabras ingresadas en el sistema, lo que ayuda a entender cuáles son los términos más buscados y para los cuales el sitio web constituye una fuente de información.

Utilizando la información almacenada en la base de datos sobre las consultas realizadas al sistema por los usuarios, se pueden acceder a los términos de búsqueda los cuales son guardados con cada uno de los campos seleccionados del documento XML en una tabla en la base de datos de manera que se incremente un contador cuando estos coincidan nuevamente para así obtener los términos más buscados realizando una consulta sencilla a la base de datos.

Los términos frecuentes dentro de la búsqueda simple y avanzada ofrecen comodidades para búsquedas sencillas ya que ponemos al alcance de un click facilidades para la ejecución de consultas al sistema desde las más simples hasta las más complejas, haciéndole así el trabajo más fácil al usuario lo cual es la meta fundamental para este proyecto.

2.3.3 Integración con la enciclopedia médica MedlinePlus

Cuando se le muestra al usuario los resultados de la consulta formulada aparecen los artículos relacionados con los términos de búsqueda, los cuales forman parte de la enciclopedia médica MedlinePlus. Esto es posible mediante el uso del ZF y en particular de la librería Zend_Search_Lucene que es la encargada de recuperar la información en este repositorio de artículos médicos.

Este servicio amplía el horizonte de búsqueda en el sistema luego de una **búsqueda simple o avanzada** permitiendo no solo buscar la información en el repositorio de autopsias sino en la enciclopedia médica MedlinePlus, perteneciente a la Biblioteca Nacional de Medicina de los Estados Unidos, y que constituye uno

de los repositorios médicos más reconocidos internacionalmente, donde los profesionales de la salud pueden consultar su contenido, fiable y actualizado. A continuación se describen las características principales de Medline y MedlinePlus.

2.3.3.1 Medline

Medline es posiblemente la base de datos de bibliografía médica más amplia que existe, producida por la Biblioteca Nacional de Medicina de los Estados Unidos. En realidad es una versión automatizada de tres índices impresos: Index Medicus, Index to Dental Literature e International Nursing Index, recoge referencias bibliográficas de los artículos publicados en unas 4.800 revistas médicas desde 1966. Actualmente reúne más de 15.000.000 citas y está en marcha un proceso para la carga paulatina de citas anteriores a 1966.

Cada registro de MEDLINE es la referencia bibliográfica de un artículo científico publicado en una revista médica, con los datos bibliográficos básicos de un artículo (Título, autores, nombre de la revista, año de publicación) que permiten la recuperación de estas referencias posteriormente en una biblioteca o a través de software específico de recuperación.

La base de datos contiene alrededor de 15 millones de artículos de aproximadamente 5.000 publicaciones seleccionadas cubriendo las áreas de biomedicina y salud desde 1950. Además de contar con gran intercambio con otros hospitales para poder adquirir más conocimiento de muchas enfermedades o estadísticas de muchos países acerca de sus niveles de incidencia.

2.3.3.2 MedlinePlus

En MedlinePlus en español (2011g) se pueden encontrar respuestas sobre cualquier tema de salud, brinda información de calidad de la Biblioteca Nacional de Medicina de los Estados Unidos, (2011a), los Institutos Nacionales de la Salud (2011d) y otras organizaciones gubernamentales y organizaciones de salud. Los profesionales de la salud pueden consultar su contenido, fiable y actualizado.

La Biblioteca Nacional de Medicina de los Estados Unidos produce y mantiene MedlinePlus en español. Sus páginas contienen enlaces, cuidadosamente seleccionados, a portales de Internet con información de alrededor de 700 temas de salud. Además, las páginas de temas de salud incluyen enlaces a noticias de salud actualizadas diariamente. MedlinePlus en español se lanzó en 2002,

brindando acceso a una colección de información de salud confiable, selectiva y en español. Los enlaces a búsquedas pre-formuladas en la base de datos de MEDLINE, ayudan a encontrar referencias a los artículos más actualizados de los profesionales de la salud.

MedlinePlus contiene:

- ✓ Información de salud y una enciclopedia médica ilustrada que cubren cientos de enfermedades, condiciones y temas sobre el bienestar general.
- ✓ Información sobre medicinas de receta y sin receta médica.
- ✓ Información sobre hierbas medicinales y suplementos dietarios.
- ✓ Ofrece a los consumidores de servicios de salud una extensa biblioteca de imágenes médicas, así como también más de 4.000 artículos con información sobre enfermedades, exámenes, síntomas, lesiones y procedimientos quirúrgicos.
- ✓ También puede encontrar enlaces a investigaciones sobre su tema de interés así como también ensayos clínicos sobre una enfermedad o condición.
- ✓ AHFS® Consumer Medication Information provee amplia información sobre cerca de 1.000 medicamentos de receta y venta libre, sus posibles efectos secundarios, precauciones y almacenamiento.
- ✓ Natural Medicines Comprehensive Database Consumer Version (Versión para el Consumidor de la Base exhaustiva de datos de medicamentos naturales) provee una colección de reseñas sobre tratamientos alternativos basada en evidencia científica y experiencia clínica. MedlinePlus cuenta con 100 monografías sobre hierbas y suplementos.
- ✓ Los tutoriales interactivos creados por el Patient Education Institute (Instituto de Educación al Paciente), ofrecen información sobre más de 165 enfermedades y procedimientos médicos. En cada tutorial se presentan gráficos y se utiliza vocabulario fácil de entender.

2.4 Consideraciones finales

Las herramientas expuestas anteriormente se destacan por su alto rendimiento, su aceptación por gran parte de la comunidad de programadores de aplicaciones web y en concreto de SRI, así como por las facilidades que brindan a los desarrolladores para manipular las partes componentes de la aplicación.

Se describe la integración con la enciclopedia médica MedlinePlus, uno de los mayores y prestigiosos repositorios de artículos relacionados con la medicina. Esta integración fue posible a través de la librería de recuperación de información Zend_Search_Lucene presente en el Zend framework que utiliza este sistema.

Otro aporte fundamental es la búsqueda por términos frecuentes. Este tipo de búsqueda goza de gran popularidad en el mundo de la web por su sencillez y fácil manejo.

Por último y no menos importante es la creación de perfiles de usuarios a través de los cuales se mantiene actualizado el historial de consultas realizadas al sistema para cada uno de ellos permitiendo así en un futuro mejorar la calidad de la recuperación de información.

Estas importantes prestaciones fueron el basamento fundamental a la hora de evaluar la utilización de las mismas en la implementación del SRI obtenido en esta investigación.

CAPÍTULO III. DISEÑO DEL SISTEMA

En este capítulo se describen los aspectos fundamentales del diseño y la implementación del SRI obtenido como resultado de esta investigación. Este sistema se utiliza para automatizar la recuperación de la información relativa a las historias clínicas de autopsias que se practican en el Hospital “Arnaldo Milián Castro” de la ciudad de Santa Clara, Cuba. El problema que se pretende resolver constituye un ejemplo de recuperación de información sobre fuentes de datos heterogéneas ya que los datos de las autopsias se encuentran almacenados en una base de datos y en una colección de documentos XML. Se muestra como queda conformada la arquitectura del sistema Goomed v2.0 con la inclusión del módulo de búsqueda por términos frecuentes y del módulo de búsqueda en la enciclopedia médica MedlinePlus.

3.1 Descripción del problema

El Departamento de Patología del Hospital “Arnaldo Milián Castro” cuenta con una base de datos que almacena la información relacionada con los reportes de autopsias que se realizan en el mismo. Esta base de datos contiene los campos tradicionales relativos a los datos personales del paciente cadáver y además presenta atributos que refieren las descripciones de los diferentes elementos que componen una autopsia, estas descripciones se conforman por texto libre, en su mayoría relativamente extensas. La información en dos de estos campos de descripción guarda a su vez cierta estructura que se mantiene uniforme para todas las autopsias. En el Anexo 1 se muestran todos los atributos de esta base de datos inicial, ejemplos de valores para los mismos y una clasificación de acuerdo al nivel de estructuración que presentan.

Los especialistas médicos necesitan usualmente obtener información de esta base de datos y de manera especial consultar los campos que almacenan los elementos descriptivos de las autopsias practicadas.

Los SGBDR se han encargado tradicionalmente de manejar los datos estructurados o datos del tipo atributo-valor y han perfeccionado las técnicas de almacenamiento y de acceso a los mismos. Sin embargo, carecen de herramientas potentes de análisis de texto que permitan una eficiente recuperación de la información presente en campos con información textual y de un ordenamiento de

los resultados obtenidos a partir de las consultas que se le formulan de acuerdo a algún criterio de relevancia.

Teniendo en cuenta esto, se hace imprescindible la aplicación de técnicas de RI para acceder a aquellas partes de las autopsias compuestas por texto libre. De esta forma se explotan, a través de su integración, las potencialidades de las tecnologías de BD y RI para satisfacer en mayor medida las necesidades de información de los usuarios.

3.2 Arquitectura del sistema

El primer paso al diseñar el SRI es procesar la información relativa a las autopsias, dejándola en un formato cuya manipulación sea fácil y rápida. De esta forma se transforma la información que se tiene de cada autopsia usando dos tecnologías: la componente estructurada se almacena en una base de datos y la componente semiestructurada se almacena en archivos XML, los cuales son objeto de un preprocesamiento que generará una representación de los mismos, formada por una secuencia de términos de indexación, los cuales mantendrán lo más fielmente posible su contenido original. Estos términos, la lista de documentos donde aparecen, su posición en el documento o componente estructurado, así como el peso asociado, pasan a formar parte del índice invertido del SRI. Una vez concluido este proceso el SRI está listo para ser consultado por el usuario.

Los módulos que componen el SRI se muestran en la figura 3.1.

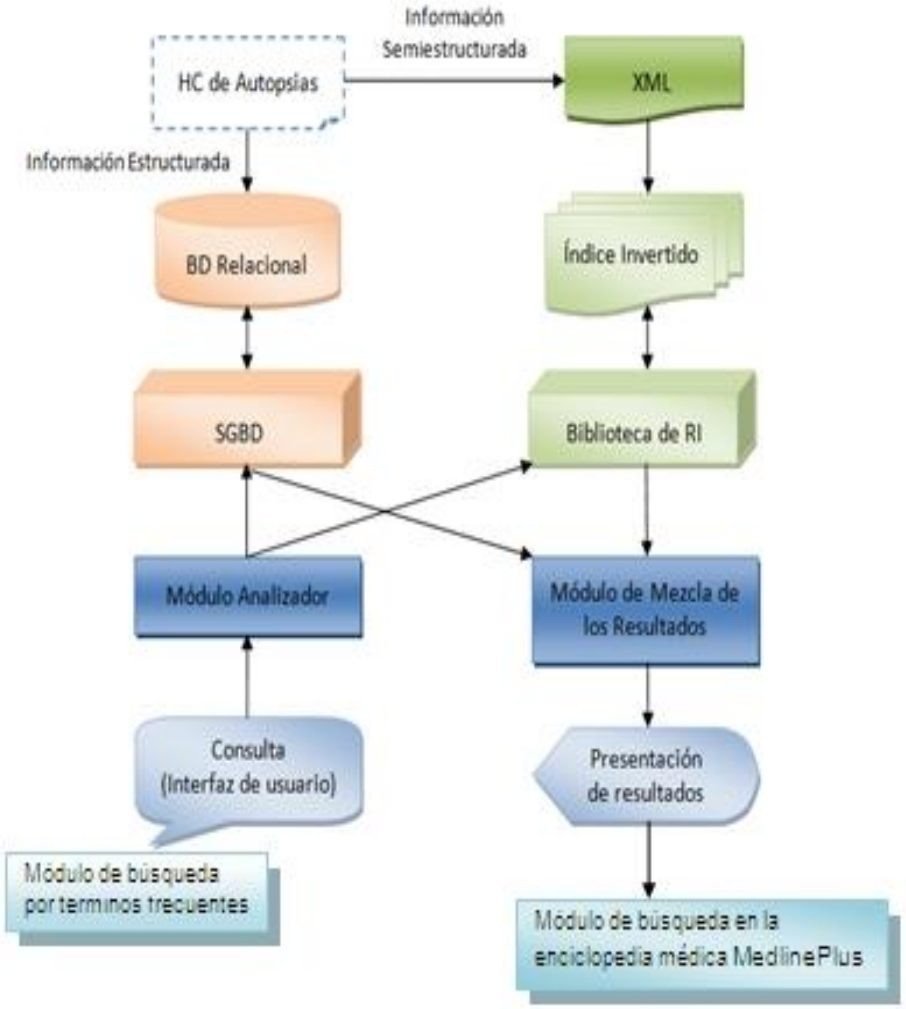

Figura 3.1. Arquitectura del SRI.

3.2.1 Interfaz de usuario

La interfaz de usuario del SRI permite capturar la necesidad de información de los usuarios mediante una consulta que haga referencia a la información estructurada y a la información en texto libre. Para ello presenta dos opciones de búsqueda: una búsqueda simple y una búsqueda avanzada.

Ambos tipos de búsqueda permiten consultar la información estructurada almacenada en la BD y la información semiestructurada presente en los archivos XML. La forma de consultar la información estructurada es común para las dos opciones y se presenta al usuario en forma de filtros, por edad, sexo y fechas de ingreso, egreso y conclusión de la autopsia. Por defecto no se le aplica ningún filtro a la consulta. A la hora de consultar la información semiestructurada el usuario

tiene la posibilidad de seleccionar los campos XML por los cuales desea buscar, de esta forma dirige la consulta a una parte del documento que le resulte de interés.

En la búsqueda simple se forma la expresión de consulta formulada en los campos seleccionados, haciendo un OR lógico entre ellos. Por ejemplo, si se seleccionan los campos “CBM” y “Historia de la Enfermedad” para la consulta “Cardiopatía isquémica” el sistema establecerá la relevancia de los documentos para dicha consulta teniendo en cuenta el texto que aparece en al menos uno de los dos campos. La búsqueda se realiza sobre una serie de campos definidos por defecto que resumen los hallazgos de la autopsia, aquellos que el usuario seleccione durante una consulta se guardan en su perfil de usuario, tal que cada vez que vaya a realizar una búsqueda de este tipo sobre los mismos campos, no tenga que seleccionarlos nuevamente.

Los campos definidos por defecto son: CDM, CIM1, CIM2, CIM3, CBM, CC1, CC2, Otros Diagnósticos, Historia de la Enfermedad, Necropsia, Hábito Externo y Estudio de las Cavidades

Por su parte, en la búsqueda avanzada el usuario puede formular una consulta construyendo un predicado lógico formado por una expresión de consulta para cada uno de los campos XML. Estas expresiones pueden ser ponderadas por el usuario en dependencia de la importancia que tengan dentro de la consulta.

3.2.2 Módulo de búsqueda en la enciclopedia médica MedlinePlus.

Este módulo se encarga de buscar artículos relacionados con el término de búsqueda en la enciclopedia médica MedlinePlus. Estos serán mostrados en los resultados de manera que el usuario pueda visualizar dichos artículos haciendo click sobre ellos e incluso se le brinda la posibilidad de navegar por la enciclopedia en busca de otros artículos de su interés.

3.2.3 Módulo de búsqueda por términos frecuentes.

En este módulo se realiza la búsqueda por los términos más buscados. Se representan por el par campo: término y se busca en el repositorio de autopsias por el término de búsqueda en su campo correspondiente.

Los restantes módulos se describen en (Castellanos et al., 2010).

3.3 Diseño del sistema

En este trabajo se presenta el diseño de un SRI que integra información estructurada y en texto libre, apoyándose en un SGBDR y una biblioteca escalable de alto rendimiento para la RI; y se realiza una implementación particular de este modelo para la recuperación de información en la colección de autopsias mencionada anteriormente, utilizándose las herramientas expuestas en el capítulo anterior.

Inicialmente se debe definir la manera de representar toda la información que se recoge en la práctica de autopsias y determinar, sobre la base del criterio de los especialistas en esta área de la medicina, aquellos componentes que resultan relevantes y a través de los cuales se desea recuperar algún tipo de información en el futuro. A continuación se relacionan dichos componentes:

- Campos estructurados: edad, sexo, fecha ingreso y fecha de egreso del cadáver; y fecha de conclusión de la autopsia.
- Campos de texto libre: causa directa de muerte (CDM); primera causa intermedia de muerte (CIM1); segunda causa intermedia de muerte (CIM2); causa básica de muerte (CBM); tercera causa intermedia de muerte (CIM3); primera causa contribuyente a la muerte (CC1); segunda causa contribuyente a la muerte (CC2); otros diagnósticos; epicrisis, que se divide en historia de la enfermedad y necropsia; y descripción macroscópica, que se divide en hábito externo, estudio de las cavidades y estudio por aparatos o sistemas (respiratorio, cardiovascular, digestivo, genitourinario, hemolinfopoyético y endocrino).

Teniendo en cuenta las diferencias en la naturaleza de los valores que van a tomar los campos mencionados anteriormente, se escogió un SGBDR para almacenar los valores estructurados; y el lenguaje XML para representar los campos de texto libre, que en su conjunto responden igualmente a una estructura predefinida para una autopsia. En lo adelante se llamará a este tipo de información *semiestructurada* para diferenciarla de la información estrictamente estructurada que se almacenará en el SGBDR. Los SGBDR manejan con mucha eficiencia los datos estructurados, enfatizando en su consistencia; y han desarrollado técnicas

sofisticadas para el procesamiento eficiente de consultas complejas y precisas sobre los mismos.

Por otro lado, el lenguaje XML hace gala de una creciente popularidad y cuenta con una amplia aceptación como estándar para el modelado, almacenamiento e intercambio de datos semiestructurados. Además existe toda una línea de investigación consolidada sobre la RI en este tipo de documentos y son conocidas las ventajas que ofrece para la recuperación. Para este problema en particular, la estructura autodescriptiva de los documentos XML mejora notablemente la precisión de la recuperación, toda vez que la información estructural puede dar valiosas pistas para consultar, ordenar y recuperar información. Esto comprende dar la posibilidad al usuario de especificar restricciones estructurales sobre lo que va a buscar y lo que desea obtener, así como establecer diferentes pesos a diversas partes de los documentos o recuperar solo sus partes más relevantes. En el Anexo 2 se muestra la definición del tipo de documento ó DTD (Document Type Definition) que valida los documentos XML obtenidos.

La principal ventaja de la utilización de estas dos fuentes de datos viene dada por la posibilidad de ejecutar consultas que seleccionen determinados registros a partir de los datos estructurados y busquen información en el texto libre presente en los datos semiestructurados.

Esta nueva versión de Goomed mantiene el mismo diseño de la anterior incluyéndole la búsqueda por términos frecuentes, haciéndoles así más fácil a los usuarios su interacción con el sistema a la hora de realizar consultas. Se crearon perfiles de usuarios atendiendo al historial de consultas que se le hayan realizado al sistema. Cada usuario tiene registrado en una tabla en la base de datos un historial de consultas formuladas por ellos mismos en su uso diario, así como los juicios de relevancia que ofrecen para cada una de ellas. Esto contribuye además a medir el grado de satisfacción de estos usuarios y de esta forma contar con otra variante de evaluación para el SRI obtenido. Además se integró al sistema la enciclopedia médica MedlinePlus para buscar en ella artículos relacionados con los términos de búsqueda.

3.3.1 Diagrama de la base de datos

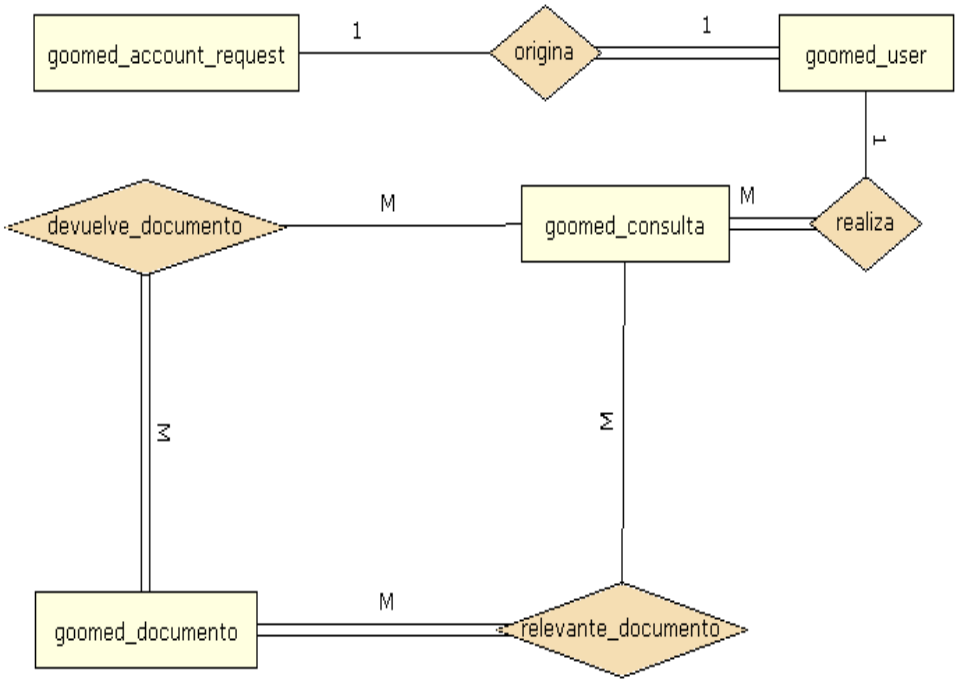

Figura 3.2. Diagrama de la base de datos.

3.3.2 Casos de uso del sistema

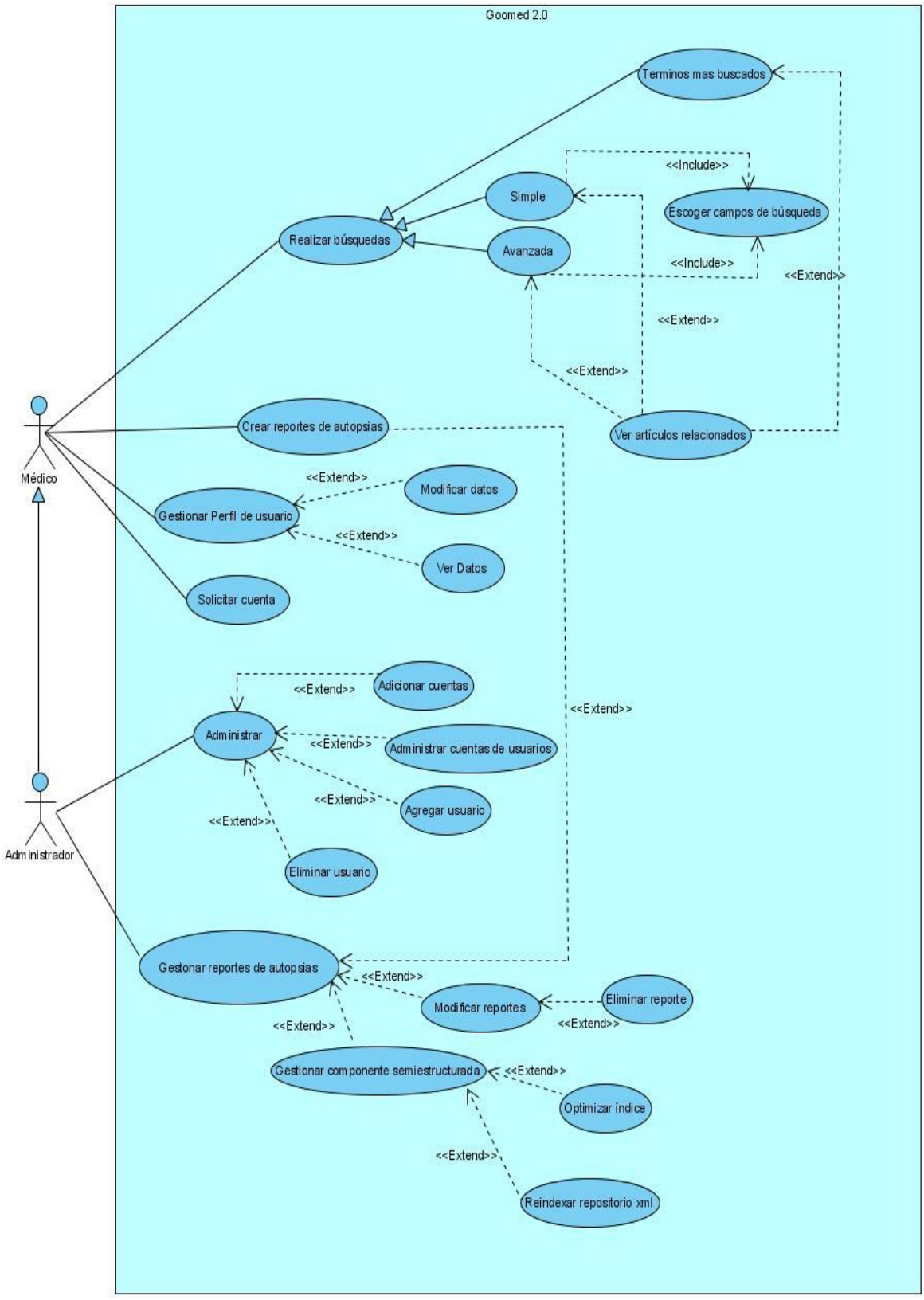

Figura 3.3. Casos de uso de sistema.

3.3.3 Diagrama de actividades para el caso de uso *realizar búsquedas*

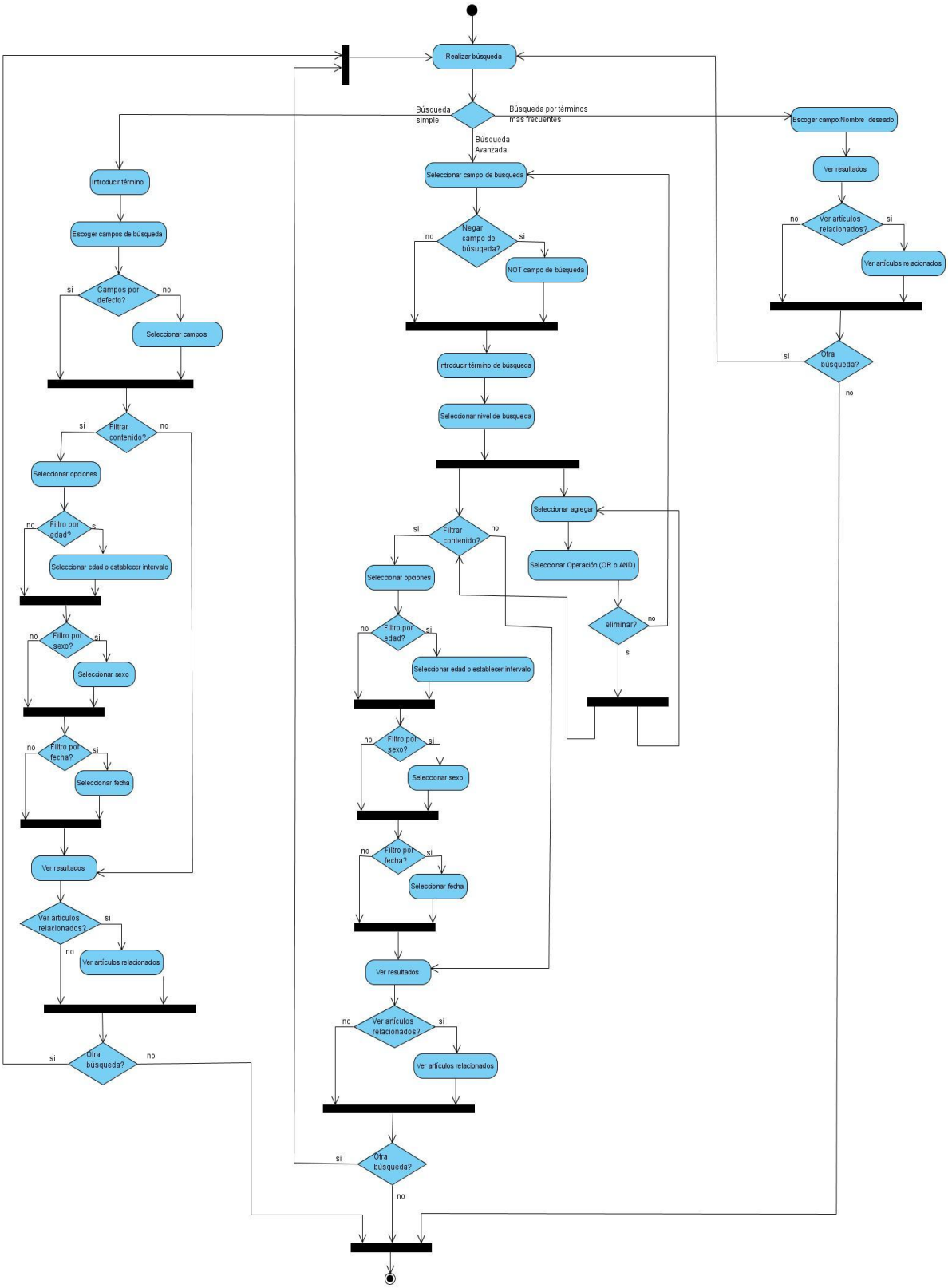

Figura 3.4. Diagrama de actividades para el caso de uso *realizar búsquedas*.

3.3.4 Solución para el caso de uso anterior

Utilizando la alternativa de búsqueda por términos frecuentes se le da solución al caso de uso *realizar búsqueda* visto anteriormente. Por ejemplo, en la figura se buscan los pacientes cuya autopsia muestre como causa básica de muerte (CBM) cualquier índole de infarto. Debajo de las opciones de filtro aparecen los términos más buscados por los usuarios unidos al campo correspondiente.

Haciendo click sobre la etiqueta que representa el par campo: término se realiza una búsqueda del término en cuestión en ese campo.

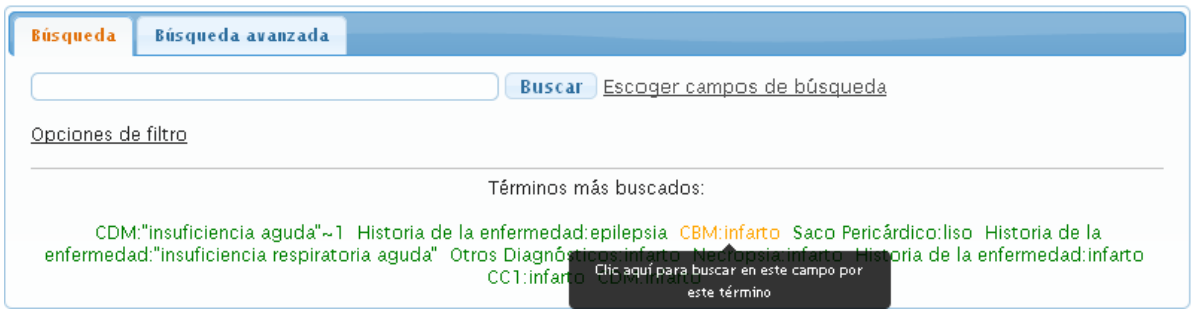

Figura 3.5. Búsqueda por términos frecuentes.

Cuando se le muestra al usuario los resultados de la consulta formulada se le indica en el panel de la derecha accesos directos a los artículos relacionados con los términos de su consulta que forman parte de la enciclopedia médica MedlinePlus.

En el ejemplo, al hacer click sobre “Artículos relacionados con infarto” nos aparece un panel con artículos relacionados con este término.

Figura 3.6. Artículos relacionados con infarto.

Al seleccionar uno de los artículos se muestra el contenido del mismo:

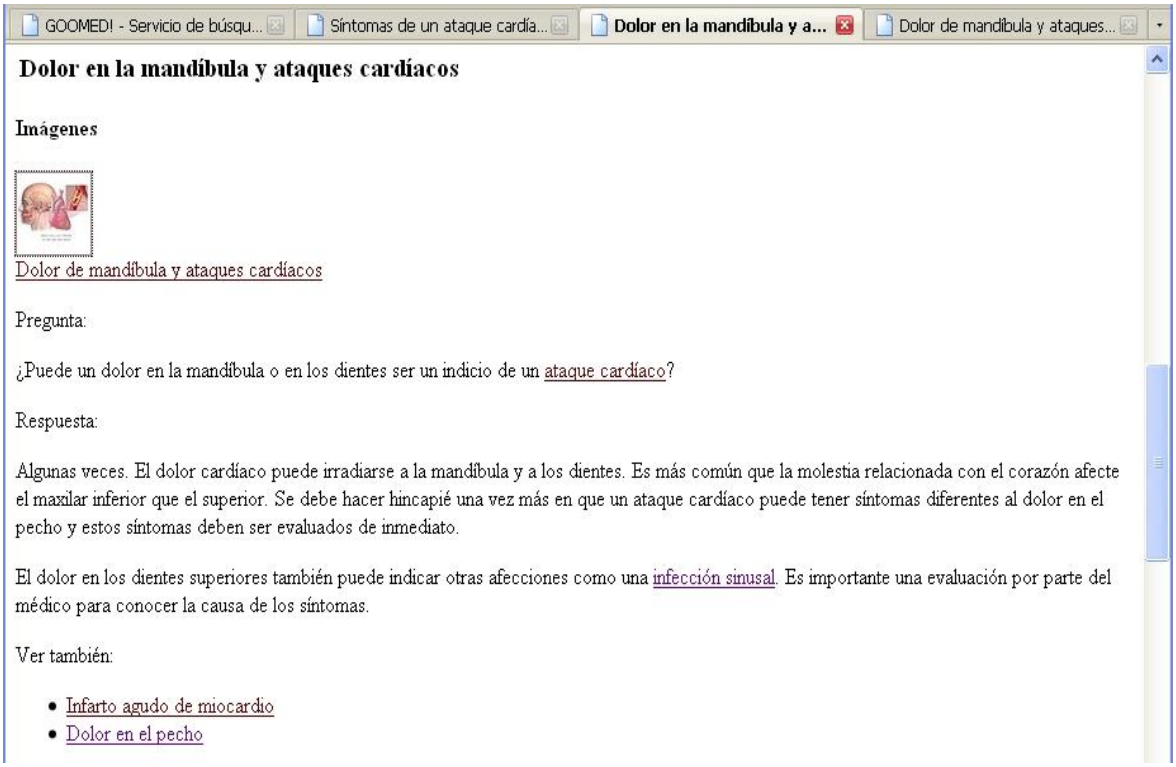

Figura 3.7. Contenido de uno de los artículos.

Se puede ampliar las imágenes de los artículos:

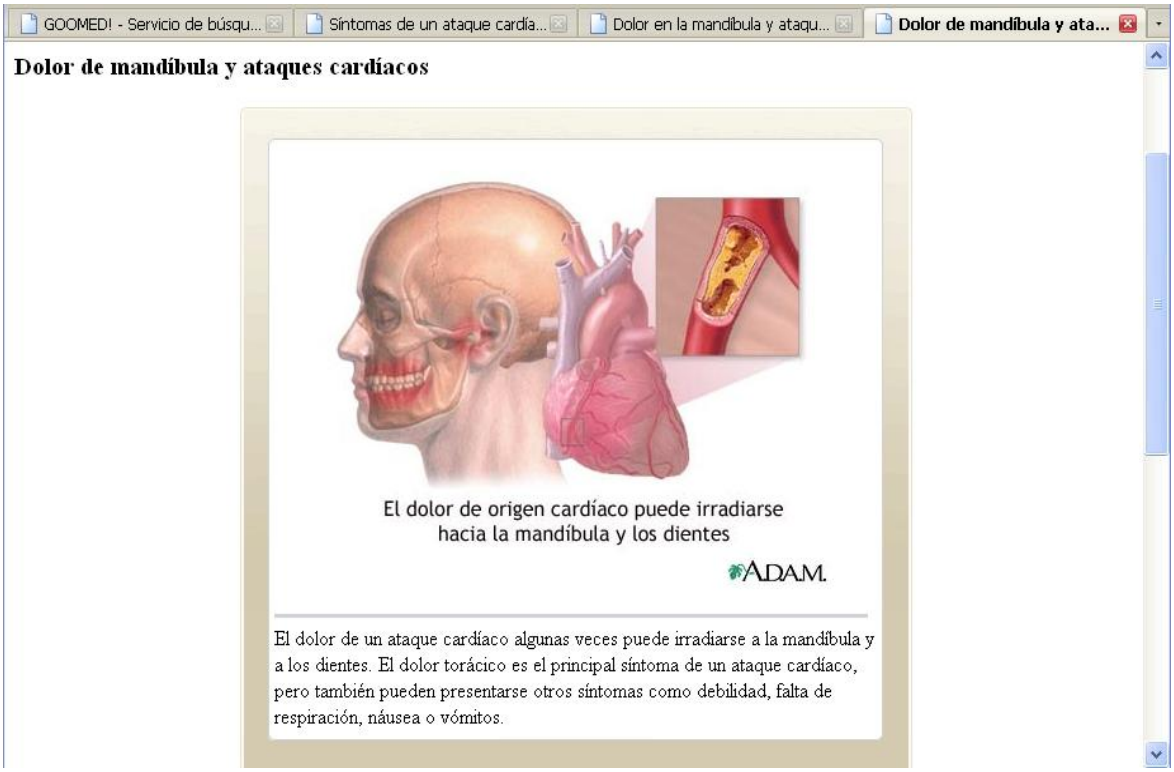

Figura 3.8. Imagen de uno de los artículos.

3.4 Características del sistema

El Anexo 3 muestra detalladamente las indicaciones a seguir a la hora de instalar el sistema. Además en el Anexo 4 se exponen los requerimientos mínimos exigidos para su correcta instalación y puesta a punto.

3.5 Consideraciones finales

La información relativa a las historias clínicas de autopsias se encuentra distribuida en bases de datos y archivos XML, por lo que el problema a tratar constituye un caso particular de RI sobre fuentes de datos heterogéneas.

Dada la complejidad del manejo de la información estructurada y semiestructurada presente en estas historias clínicas y las posibilidades reales de recuperar información a partir de las mismas, se le añadieron nuevas prestaciones a un SRI capaz de combinar ambos tipos de información integrando las tecnologías de BD y RI.

CONCLUSIONES

Se desarrolló una nueva versión de Goomed manteniendo la arquitectura de la versión anterior añadiéndole nuevas prestaciones al sistema, tales como la inclusión términos frecuentes, los cuales le facilitan y optimizan la búsqueda al usuario. Se creó un perfil para cada usuario atendiendo al historial de consultas que ha realizado al sistema y los juicios de relevancia proporcionados. Además se crearon accesos directos a los artículos relacionados con los términos de consulta que forman parte de la enciclopedia médica MedlinePlus. Este servicio amplía el horizonte de búsqueda en el sistema luego de una búsqueda simple o avanzada permitiendo no solo buscar la información en el repositorio de autopsias sino en la enciclopedia médica MedlinePlus, perteneciente a la Biblioteca Nacional de Medicina de los Estados Unidos, y que constituye uno de los repositorios médicos más reconocidos internacionalmente, donde los profesionales de la salud pueden consultar su contenido, fiable y actualizado.

RECOMENDACIONES

1. Incorporar al sistema un mecanismo de realimentación de relevancia mediante la identificación de resultados (pacientes) similares a uno o varios obtenidos en una búsqueda inicial. Para ello se tendrán en cuenta información estructural a la hora de medir la similitud.
2. Implementar un mecanismo de agrupamiento de resultados mediante la aplicación de técnicas de visualización.
3. Finalmente se necesita un mecanismo de evaluación para este SRI atendiendo a medidas establecidas en la literatura como la exhaustividad (recall, en inglés) y la precisión.

BIBLIOGRAFÍA

***Active Record Pattern**. (2010a).* [Online]. Disponible en: http://www.devshed.com/c/a/PHP/The-Active-Record-Pattern/.

***CMS**. (2010b).* [Online]. Disponible en: http://en.wikipedia.org/wiki/Content_management_system.

***CSS**. (2010c).* [Online]. Disponible en: http://www.w3.org/Style/CSS/.

***DBAL**. (2010d).* [Online]. Disponible en: http://en.wikipedia.org/wiki/Database_abstraction_layer.

***DQL**. (2010e).* [Online]. Disponible en: http://www.doctrine-project.org/documentation/manual/1_2/en/dql-doctrine-query-language.

***HQL**. (2010f).* [Online]. Disponible en: http://docs.jboss.org/hibernate/core/3.3/reference/en/html/queryhql.html.

***HTML**. (2010g).* [Online]. Disponible en: http://www.w3.org/MarkUp/.

***HTTP**. (2010h).* [Online]. Disponible en: http://www.w3.org/Protocols/.

***INI**. (2010i).* [Online]. Disponible en: http://es.wikipedia.org/wiki/INI_(extensi%C3%B3n_de_archivo.

***Java**. (2010j).* [Online]. Disponible en: http://www.java.com/.

***Rapid Application Development**. (2010k).* [Online]. Disponible en: http://www.cs.bgsu.edu/maner/domains/RAD.htm.

***SOLRPhpClient**. (2010l).* [Online]. Disponible en: http://code.google.com/p/solr-php-client/.

***Biblioteca Nacional de Medicina de los Estados Unidos**. (2011a).* [Online]. Disponible en: http://es.wikipedia.org/wiki/Biblioteca_Nacional_de_Medicina_de_los_Estados_Unidos.

***Doctrine**. (2011b).* [Online]. Disponible en: http://www.doctrine-project.org/.

***Front Controller Pattern**. (2011c).* [Online]. Disponible en: http://java.sun.com/blueprints/corej2eepatterns/Patterns/FrontController.html .

***Institutos Nacionales de la Salud**. (2011d).* [Online]. Disponible en: http://es.wikipedia.org/wiki/Institutos_Nacionales_de_la_Salud.

***JQuery**. (2011e).* [Online]. Disponible en: http://jquery.com/.

***JSON**. (2011f).* [Online]. Disponible en: http://www.json.org/.

***MedlinePlus**. (2011g).* [Online]. Disponible en: http://medlineplus.gov/spanish/.

***MVC**. (2011h).* [Online]. Disponible en: http://java.sun.com/blueprints/patterns/MVC.html.

***MySQL**. (2011i).* [Online]. Disponible en: http://www.mysql.com/.

***REST**. (2011j).* [Online]. Disponible en: http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm.

***URI-URL**. (2011k).* [Online]. Disponible en: http://www.w3.org/Addressing/.

***Zend Framework**. (2011l).* [Online]. Disponible en: http://framework.zend.com/.

***Zend Lucene**. (2011m).* [Online]. Disponible en: http://framework.zend.com/manual/en/learning.lucene.html.

AMER-YAHIA, S., CASE, P., RÖLLEKE, T., SHANMUGASUNDARAM, J. & WEIKUM, G. 2005. Report on the DB/IR panel at SIGMOD 2005. *SIGMOD Rec.,* 34, 71-74.

AMER-YAHIA, S. & LALMAS, M. 2006. XML search: languages, INEX and scoring. *SIGMOD Rec.,* 35, 16-23.

BAEZA-YATES, R. A. & RIBEIRO-NETO, B. 1999. *Modern Information Retrieval*, Addison-Wesley Longman Publishing Co., Inc.

BICKLEY, L. S., BATES, B. & SZILAGYI, P. G. 2005. *Bates' Guide to Physical Examination And History Taking,* Philadelphia, USA, Lippincott Williams & Wilkins.

BRUCE CROFT, W. & SCHEK, H.-J. 2008. Introduction to the special issue on database and information retrieval integration. *The VLDB Journal,* 17, 1-3.

CASTELLANOS, C. I., BORGES, L. & ROSA, D. 2010. *Diseño e implementación de un Sistema de Recuperación de Información a partir de fuentes de datos heterogéneas.* Bachelor of Science, Universidad de Las Villas.

CUTTING, D. *Apache Lucene. (2010).* [Online]. Disponible en: http://lucene.apache.org/nutch/. Consultado: Diciembre 10, 2010.

CHAUDHURI, S., RAMAKRISHNAN, R. & WEIKUM, G. 2005. *Integrating DB and IR Technologies: What is the Sound of One Hand Clapping?*

CHIARAMELLA, Y. 2001. Information retrieval and structured documents. *Lectures on information retrieval.* Springer-Verlag New York, Inc.

CHIARAMELLA, Y., MULHEM, P. & FOUREL, F. 1996. A model for multimedia information retrieval. Technical Report, University of Glasgow.

FERNÁNDEZ, J. M. 2001. *Modelos de Recuperación de Información basados en Redes de Creencia.* Tesis Doctoral Tesis Doctoral, Universidad de Granada.

FOUNDATION, A. S. *Apache. (2011a).* [Online]. Disponible en: http://www.apache.org/. Consultado: Enero 16, 2011.

FOUNDATION, T. A. S. *Apache Tomcat. (2011b).* [Online]. Disponible en: http://tomcat.apache.org/. Consultado: January 16, 2011.

FUHR, N. & GROßJOHANN, K. 2004. XIRQL: An XML query language based on information retrieval concepts. *ACM Trans. Inf. Syst.,* 22, 313-356.

FUHR, N. & LALMAS, M. 2007. Advances in XML retrieval: the INEX initiative. *Proceedings of the 2006 international workshop on Research issues in digital libraries.* Kolkata, India: ACM.

GOSPODNETIC, O. & HATCHER, E. 2005. *Lucene in Action*, Manning Publications Co.

GÖVERT, N. & KAZAI, G. 2002. Overview of the INitiative for the Evaluation of XML retrieval (INEX) 2002. *In:* FUHR, N., GÖVERT, N., KAZAI, G. & LALMAS, M., eds. Proceedings of the First Workshop of the INitiative for the Evaluation of XML Retrieval (INEX), December 9-11 2002 Schloss Dagstuhl, Germany. 1-17.

HARMAN, D., BAEZA-YATES, R., FOX, E. & LEE, W. 1992. Inverted files. *Information retrieval: data structures and algorithms.* Prentice-Hall, Inc.

KORFHAGE, R. R. 1997. *Information storage and retrieval*, John Wiley \& Sons, Inc.

LANCASTER, F. W. 1979. *Criteria by Which Information Retrieval Sytstems May Be Evaluated*.

LEE, Y. K., YOO, S.-J., YOON, K. & BERRA, P. B. 1996. Index structures for structured documents. *Proceedings of the first ACM international conference on Digital libraries.* Bethesda, Maryland, United States: ACM.

MANNING, C. D., RAGHAVAN., P. & SCHÜTZE, H. 2008. *An Introduction to Information Retrieval,* New York, Cambridge Univ. Press.

NAVARRO, G. & BAEZA-YATES, R. 1997. Proximal nodes: a model to query document databases by content and structure. *ACM Trans. Inf. Syst.,* 15, 400-435.

PORTER, M. F. 1980. An algorithm for suffix stripping. *Program,* 14, 130-137.

RIJSBERGEN, C. J. V. 1979. *Information Retrieval*, Butterworth-Heinemann.

ROBERTSON, S. E. 1977. The probability ranking principle in IR. *Journal of Documentation,* 33, 294 - 304.

ROCCHIO, J. J. 1971. *Relevance Feedback in Information Retrieval*.

SALTON, G. 1989. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*, Addison-Wesley Longman Publishing Co., Inc.

SALTON, G. & LESK, M. E. 1968. Computer Evaluation of Indexing and Text Processing. *J. ACM,* 15, 8-36.

SALTON, G. & MCGILL, M. J. 1983. *Introduction to Modern Information Retrieval*, McGraw-Hill, Inc.

SALTON, G., WONG, A. & YANG, C. S. 1975. A vector space model for automatic indexing. *Commun. ACM,* 18, 613-620.

SHAH, B., GUMMADI, A., YOON, J. P. & RAGHAVAN, V. 2004. Efficient Dynamic Indexing and Retrieval of XML Documents using Three-Dimensional Quasi-BitCube. *In:* Proc. of the First International Workshop on High Performance XML Processing, 2004.

TROTMAN, A. 2004. Searching structured documents. *Inf. Process. Manage.,* 40, 619-632.

TROTMAN, A. 2005. Choosing document structure weights. *Inf. Process. Manage.,* 41, 243-264.

TROTMAN, A. & SIGURBJÖRNSSON, B. 2005. Narrowed Extended XPath I (NEXI).

WEIKUM, G. 2007. *DB&IR: both sides now,* Beijing, China, ACM.

ANEXOS

Anexo 1.

Descripción de la tabla de la base de datos que almacena la información de las autopsias en el hospital "Arnaldo Milián Castro".

Tabla: Autopsias		
Atributo	**Clasificación**	**Ejemplo**
Nºde autopsia	Estructurado	1
HC	Estructurado	145444
Nombre	Estructurado	Simón
Primer Apellido	Estructurado	Machado
Segundo Apellido	Estructurado	González
Edad	Estructurado	91
Sexo	Estructurado	M
Piel	Estructurado	Blanca
Fecha de ingreso	Estructurado	08/12/2008
Fecha de egreso	Estructurado	01/01/2009
Fecha de concluido	Estructurado	10/01/2009
Completa	Estructurado	Si
Médico especialista	Estructurado	1
Residente	Estructurado	4
Técnico	Estructurado	8
CDM (Causa directa de muerte)	Texto libre	Insuficiencia respiratoria aguda
1CIM (Primera causa inmediata de muerte)	Texto libre	Bronconeumonía severa bilateral
2CIM (Segunda causa inmediata de muerte)	Texto libre	Broncoaspiración
3CIM (Tercera causa inmediata de muerte)	Texto libre	Hemorragia cerebral intraparenquimatosa hemisférica derecha
CBM (Causa básica de muerte	Texto libre	Hipertensión arterial esencial. Crisis hipertensiva
1CC (Primera causa complementaria de muerte)	Texto libre	Encefalopatía hipóxica

2CC (Segunda causa complementaria de muerte)	Texto libre	-
Otros diagnósticos	Texto libre	Enfisema pulmonar apical senil Cardiopatía arterioesclerótica Ateromatosis severa de aorta y ramas principales Éstasis pasivo crónico hepático Nefroangioesclerosisarteriolar benigna Cistitis aguda
Epicrisis	Texto libre	Paciente masculino, blanco de 80 años con antecedentes de cardiopatía isquémica crónica e HTA esencial. Anamnesis reciente de cuadro neurológico de instalación brusca dado por pérdida de la fuerza muscular con hemiplejia izquierda directa, total y proporcional acompañada de cifras elevadas de TA (200/120) por lo que fue valorado en CG de Hospital Arnaldo Milián donde se indicó TAC de cráneo de urgencia que constató imagen hiperdensa en territorio profundo de ACM derecha, y se decidió ingreso en servicio de Neurología donde se sobreañadió precozmente sepsis respiratoria interpretada como de etiología broncoaspirativa y se inició terapia antimicrobiana de amplio espectro. En dicha sala evolucionó tórpidamente por espacio de 8 días con deterioro de la función neurológica (estupor profundo) y ventilatoria y finalmente presentó PCR no recuperado y falleció con los diagnósticos clínicos a egreso de insuficiencia respiratoria aguda, BNB aspirativa, hemorragia cerebral intraparenquimatosa del territorio profundo de ACM derecha, cardiopatía isquémica crónica e HTA esencial. En la necropsia se comprobó una hemorragia cerebral intraparenquimatosa hemisférica derecha del hipertenso, con sepsis respiratoria sobreañadida de etiología broncoaspirativa que lo condujo a la muerte en insuficiencia respiratoria aguda
Descripción macroscópica	Texto libre	Hábito externo: Cadáver masculino, de raza blanca, normolíneo. Estudio de las cavidades: Sin contenido patológico Por aparatos. Respiratorio: Laringe: de forma normal, cubierta por los músculos perilaríngeos y el tiroides, al abrirla observamos mucosa rosada sin alteraciones

		Pulmones: Marcada consolidación del parénquima pulmonar bilateral, más marcada hacia base pulmonar derecha, friable, cubierto por pleura rojo-azulada. Contenido mucopurulento abundante en luz traqueobronquial. Áreas hiperareadas de localización apical. PI: 400gr, PD: 480gr. Arterias pulmonares de distribución normal, con aisladas placas de ateroma. No trombos. CVC: Saco pericárdico: liso y brillante. Corazón: Peso: 400gr, de forma cónica, hipertrofia VI Epicardio: Liso y brillante Miocardio: Color pardo rojizo, con estrías blanquecino-amarillentas. VI: 2cm VD: 0,5cm. Endocardio mural: color blanquecino sin alteraciones. Endocardio valvular: de color amarillento Válvula Mitral: 10cm. Válvula Aórtica: 7cm. Válvula Tricúspide: 12cm. Válvula Pulmonar: 7cm Músculos papilares y cuerdas tendinosas: músculos papilares y cuerdas tendinosas que guardan continuidad con el aparato valvular. Aorta y grandes vasos arteriales: de configuración normal con numerosas placas fibrocalcificadas, distribuidas asimétricamente en segmento abdominal-ilíaco, más prominentes alrrededor de grandes vasos Arterias coronarias: los tres vasos principales de distribución normal con numerosas placas fibrocalcificadas en los tres troncos coronarios epicárdicos, oclusivas entre el 25-50% de la luz Vena cava: de configuración normal con superficie interna lisa y brillante, permeable en su totalidad. Aparato Digestivo: Cavidad oral: Lengua saburral, ausencia de piezas dentarias Esófago: de configuración normal, permeable en todo su trayecto, adventicia sin depósitos anómalos, mucosa de color rosado grisáceo, sin alteraciones

		Estómago: Forma y tamaño normal, superficie serosa lisa y brillante, al abrirlo vemos escasos restos alimentarios, mucosa pardo- rojiza, lisada Grasa mesentérica: Sin alteraciones Peritoneo: Liso y brillante Duodeno: Marco duodenal de configuración normal serosa lisa y brillante, mucosa rosada sin alteraciones. Ampolla de Vater sin alteraciones Yeyuno-ileon: de configuración normal, serosa lisa y brillante, al abrirla mucosa de color rosado y conserva los pliegues, contenido alimentario en luz intestinal Colon: Serosa lisa y brillante, al abrirla mucosa de rosada que conserva los pliegues, así como abundante contenido fecaloideo en luz intestinal Apéndice cecal: serosa lisa y brillante, al corte luz con escaso material fecaloideo. Recto y ano: Permeable sin alteraciones Hígado: Peso: 1400gr con superficie lisa, color pardo-amarillento, moscado, consistencia firme Páncreas: Aspecto racimoso, sin alteraciones. Vesícula y vías biliares: Forma y tamaño normal. No formaciones litiásicas, mucosa aterciopelada. Resto de VB sin alteraciones Aparato genitourinario: Riñones: RI: 140gr RD:140gr, simétricos. Decapsulan con facilidad. Superficie granular fina, con buena relación córtico-medular. Cicatrices de base estrecha en forma de V. Sistema pielocalicial permeable Uréteres: de forma y longitud normal, permeables. Vejiga: De tamaño normal, mucosa rosada, edematosa y con punteado eritematoso Próstata: De forma y tamaño normal, no nódulos Sistema hemolinfopoyético: Bazo: Peso: 60gr, superficie lisa, color rojo vino, firme. Cadenas linfáticas: No adenomegalias Sistema endocrino:

		Glándula tiroides: De forma y tamaño normal y color pardo homogéneo. Glándulas suprarrenales: De forma y tamaño normal y color amarillo ocre. Encéfalo: Peso 1300gr. Focos de HSA de la convexidad hemisférico parietal izquierdo y hemisferio cerebeloso derecho. Vasos del polígono S/A. No hernias. Dislaceración del hemisferio derecho por encima del cuerpo calloso. Al corte extenso coágulo hemático de 8x3cm hemisférico derecho (parietal). Edema y punteado congestivo de la sustancia blanca

Anexo 2.

Definición del tipo de documento ó DTD (Document Type Definition) que valida los documentos XML que forman la base documental del SRI.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENTAortaGrandesVasosArteriales (#PCDATA)>
<!ELEMENTApendiceCecal (#PCDATA)>
<!ELEMENTArteriasCoronarias (#PCDATA)>
<!ELEMENT Autopsia ((Nro, CDM, CIM1, CIM2, CIM3, CBM, CC1, CC2,
OtrosDiagnosticos, Epicrisis, DescripcionMacroscopica))>
<!ELEMENT Bazo (#PCDATA)>
<!ELEMENTCadenasLinfaticas (#PCDATA)>
<!ELEMENTCavidadOral (#PCDATA)>
<!ELEMENT CBM (#PCDATA)>
<!ELEMENT CC1 #PCDATA>
<!ELEMENT CC2 #PCDATA>
<!ELEMENT CDM (#PCDATA)>
<!ELEMENT CIM1 (#PCDATA)>
<!ELEMENT CIM2 (#PCDATA)>
<!ELEMENT CIM3 (#PCDATA)>
<!ELEMENT Colon (#PCDATA)>
<!ELEMENTCorazon (#PCDATA)>
<!ELEMENT CVC ((SacoPericardico, Corazon, Epicardio, Miocardio, VI, VD,
EndocardioMural, EndocardioValvular, ValvulaMitral, ValvulaAortica,
ValvulaTricuspide, ValvulaPulmonar, MusculosPapCuerdasTendinosas,
AortaGrandesVasosArteriales, ArteriasCoronarias, VenaCava))>
<!ELEMENTDescripcionMacroscopica ((HabitoExterno, EstudioCavidades,
Sistemas))>
<!ELEMENT Digestivo ((CavidadOral, Esofago, Estomago, GrasaMesenterica,
Peritoneo, Duodeno, Yeyuno-ileon, Colon, ApendiceCecal, RectoAno, Higado,
Pancreas, VesiculaViasBiliares))>
<!ELEMENT Duodeno (#PCDATA)>
<!ELEMENTEncefalo (#PCDATA)>
<!ELEMENTEndocardioMural (#PCDATA)>
<!ELEMENTEndocardioValvular (#PCDATA)>
<!ELEMENT Endocrino ((GlandulaTiroides, GlandulasSuprarrenales, Encefalo))>
<!ELEMENT Epicardio (#PCDATA)>
<!ELEMENT Epicrisis ((HistoriaEnfermedad, Necropsia))>
<!ELEMENTEsofago (#PCDATA)>
<!ELEMENT Estomago (#PCDATA)>
<!ELEMENTEstudioCavidades (#PCDATA)>
```

<!ELEMENT Genitourinario ((Rinnones, Ureteres, Vejiga, UteroAnejos, Prostata))>
<!ELEMENTGlandulasSuprarrenales (#PCDATA)>
<!ELEMENTGlandulaTiroides (#PCDATA)>
<!ELEMENTGrasaMesenterica (#PCDATA)>
<!ELEMENTHabitoExterno (#PCDATA)>
<!ELEMENTHemolinfopoyetico ((Bazo, CadenasLinfaticas))>
<!ELEMENTHigado (#PCDATA)>
<!ELEMENTHistoriaEnfermedad (#PCDATA)>
<!ELEMENT Laringe (#PCDATA)>
<!ELEMENT Miocardio (#PCDATA)>
<!ELEMENTMusculosPapCuerdasTendinosas (#PCDATA)>
<!ELEMENT Necropsia (#PCDATA)>
<!ELEMENTNro (#PCDATA)>
<!ELEMENTOtrosDiagnosticos (#PCDATA)>
<!ELEMENTPancreas (#PCDATA)>
<!ELEMENTPeritoneo (#PCDATA)>
<!ELEMENTProstata (#PCDATA)>
<!ELEMENT Pulmones (#PCDATA)>
<!ELEMENTRectoAno (#PCDATA)>
<!ELEMENT Respiratorio ((Laringe, Pulmones))>
<!ELEMENTRinnones (#PCDATA)>
<!ELEMENTSacoPericardico (#PCDATA)>
<!ELEMENT Sistemas ((Respiratorio, CVC, Digestivo, Genitourinario, Hemolinfopoyetico, Endocrino))>
<!ELEMENTUreteres (#PCDATA)>
<!ELEMENTUteroAnejos #PCDATA>
<!ELEMENTValvulaAortica (#PCDATA)>
<!ELEMENTValvulaMitral (#PCDATA)>
<!ELEMENTValvulaPulmonar (#PCDATA)>
<!ELEMENTValvulaTricuspide (#PCDATA)>
<!ELEMENT VD (#PCDATA)>
<!ELEMENT Vejiga (#PCDATA)>
<!ELEMENTVenaCava (#PCDATA)>
<!ELEMENTVesiculaViasBiliares #PCDATA>
<!ELEMENT VI (#PCDATA)>
<!ELEMENT Yeyuno-ileon (#PCDATA)>

Anexo 3.

Pasos para la instalación del SRI.

Nota: En esta descripción se usa “Archivos de Programa” para referenciar el directorio de los programas instalados en una versión en español de Windows. Si se tiene una versión en inglés se debe reemplazar “Archivos de Programa” por “Program Files”.

1. Instalar el JRE 1.5 o superior (http://java.sun.com/javase/downloads/index.jsp). En este directorio: *jdk-6u12-windows-i586-p.exe*.
2. Instalar el software xampp en Archivos de Programa. (http://www.apachefriends.org/es/xampp.html). En este directorio: *xampp-win32-1.7.3.exe*.
3. Instalar xampp-tomcat-addon dentro de xampp. En este directorio: *xampp-tomcat-addon-win32-6.0.20.exe*.

   Nota: Si se tiene Windows 7 64 bits como Sistema Operativo se debe además copiar el archivo **msvcr71.dll** de la carpeta “JRE” de Java (“C:\Archivos de Programa\Java\jre6\bin”) a la carpeta “bin” de Apache Tomcat (“C:\Archivos de programa\xampp\tomcat\bin”).

   Para probar que esté instalado correctamente se inicia el servidor y en un navegador se teclea http://localhost:8080. Si todo está bien instalado debe aparecer en el navegador la página de bienvenida de Apache Tomcat.
4. Iniciar los servicios de Apache,MySQL y Tomcat en el panel de control del xampp.
5. Editar el archivo *http.conf* de Apache, que estará en “C:\Archivos de programa\xampp\apache\conf\”.

   En la parte de carga de módulos añadir:

   LoadModule jk_module modules/mod_jk.so

   Además se debe descomentar la línea:

   LoadModule rewrite_module modules/mod_rewrite.so
6. Reiniciar el servidor Apache. Para comprobar que se han integrado correctamente los servidores, se puede visitar el URL http://localhost el cual

devolvería la página de inicio del Apache en este caso del XAMPP y escribiendo por ejemplo, http://localhost/docs se obtendría la página la documentación de Tomcat.

7. Instalación del SOLR

7.1 Editar el archivo *server.xml* en "C:\Archivos de programa\xampp\tomcat\conf\" añadiendo el conector URIEncoding como sigue:

```
<Server ...>
  <Service ...>
    <Connector ... URIEncoding="UTF-8"/>
      ...
    </Connector>
  </Service>
</Server>
```

O simplemente reemplazarlo por el que se adjunta en este directorio.

7.2 Instalar en el disco C: *Solr config 1 for Goomed 2.0.exe*

7.3 Instalar en el disco C: *Solr config 2 for Goomed 2.0.exe*

7.4 Detener el servicio del Tomcat.

7.5 Copiar el archivo *apache-solr-1.3.0.war* desde "C:\temp\solrZip\dist\" al directorio *webapps* de tomcat, "C:\Archivos de programa\xampp\tomcat\ webapps".

7.6 Renombrar el archivo *apache-solr-1.3.0.war* a *solr.war*.

7.7 Ejecutar *tomcat6w.exe* en "C:\Archivos de programa\xampp\tomcat\bin" y añadir la siguiente línea para que el servidor Tomcat se inicie con esta opción de Java (véase la figura 1).

-Dsolr.solr.home=C:\web\solr

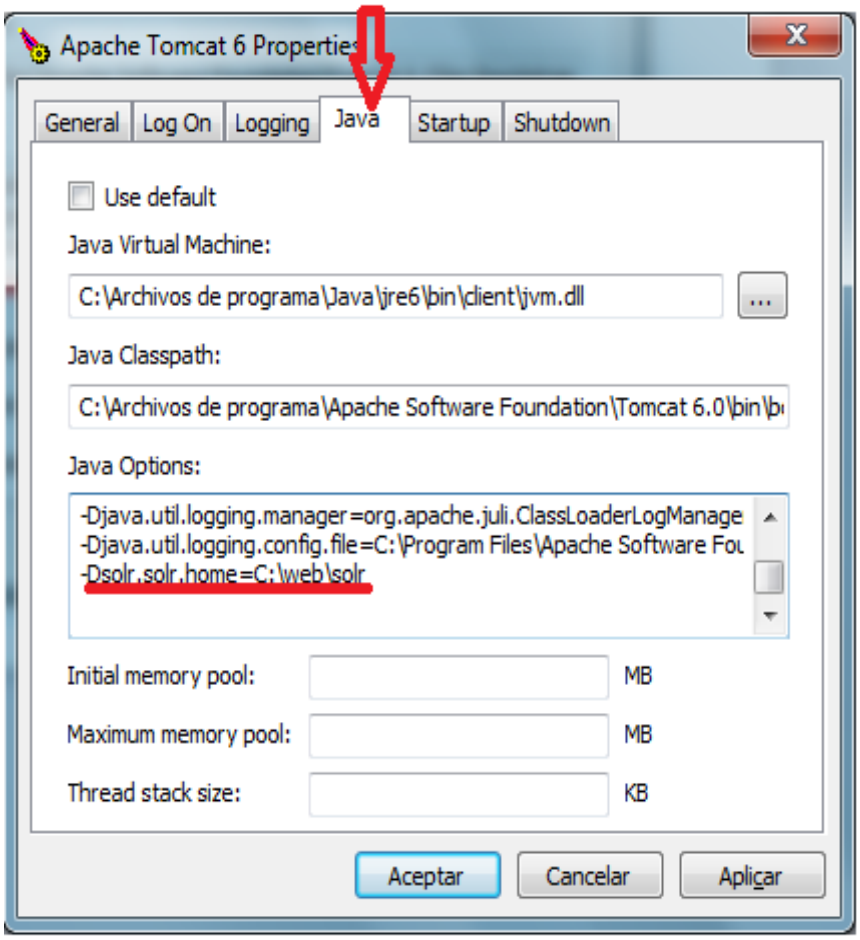

Figura 1. Configuración para que el Tomcat inicie con la opción de Java especificada.

7.8 Editar el archivo “mod_jk.conf” "C:\Archivos de programa\xampp\tomcat\conf\auto\" añadiendo:

```
<VirtualHost localhost>
        (...)
        JkMount /solr ajp13
        JkMount /solr/* ajp13
</VirtualHost>
```

7.9 Iniciar el servicio de Tomcat.

7.10 Reiniciar el servidor de Apache.

7.11 Ir a la página de administración de SOLR y verificar que la instalación está funcionado: http://localhost:8080/solr/admin y para verificar que funciona desde Apache: http://localhost/solr/admin.

8. Copiar el proyecto y configurar apache para la ejecución

8.1 Instalar Goomed 2.0 Install.exe en la carpeta *htdocs* del xampp.

8.2 Editar el archivo *application.xml* en "C:\Archivos de programa \xampp\htdocs\goomed\application\configs\" para configurar:

- ✓ Los datos generales de sitio en la sección <site> y la configuración del servidor SMTP para el envío de correo electrónico a los usuarios en la sección <email> dentro de <site>

  Ejemplo:

```
<site>
    <name>GOOMED! - Servicio de búsqueda médica online</name>
     <email>
            <admin>mestela@uclv.edu.cu</admin>
            <server>localhost</server>
            <username>drosa</username>
            <password>darien</password>
    </email>
 </site>
```

- ✓ En la sección <application> se configura el directorio donde se ubica el repositorio XML, además la dirección y puerto donde está accesible el SOLR.

  Ejemplo:

```
<application>
    <repPath>C:\ Archivos de programa \xampp\htdocs\goomed\data\xml-rep</repPath>
    <solrServer>localhost</solrServer>
    <solrPort>8080</solrPort>
    <solrPath>/solr</solrPath>
</application>
```

- ✓ En la sección <modules> se puede configurar los módulos a usar por el sistema, también se permite añadir nuevos módulos a través de esta sección de la configuración siguiendo la sintaxis del módulo "user" activado por defecto.
- ✓ La configuración del SGBDR se especifica en la sección <database> donde se deben configurar las credenciales de autenticación con el gestor de BD escogido y el nombre de la base de datos que se emplea en el sistema.

Ejemplo:

```
<database>
    <adapter>mysql</adapter>
    <params>
        <host>localhost</host>
        <username>goomed</username>
        <password>goomed</password>
        <dbname>goomed</dbname>
        <proto-opts></proto-opts>
    </params>
</database>
```

8.3 Crear una base de datos en el servidor de MySQL con la opción utf8 -- UTF-8 Unicode en "Character set". Se le debe asignar el mismo nombre que se especificó en el paso anterior.

8.4 Restituir (restore) la base de datos que se encuentra en el directorio del proyecto usando el *backup o el* script *goomed.sql*.

8.5 Configurar un Servidor Virtual (*Virtual Host*) para el sitio, añadiendo al archivo *httpd-vhosts.conf* en "C:\Archivos de programa\xampp\apache\conf\extra\" las siguientes líneas o simplemente sustituyéndolo por el que se adjunta en este directorio:

```
<VirtualHost *:80>
        ServerAdmin <CORREO DEL ADMINISTRADOR DEL SISTEMA>
        DocumentRoot <CAMINO_A_GOOMED>/public/
        ServerName <NOMBRE DEL SERVIDOR>
        <Directory <CAMINO_A_GOOMED>/public/>
                DirectoryIndex index.php
                AllowOverride All
                Order allow,deny
                Allow from all
        </Directory>
</VirtualHost>
```

Donde:

<CORREO DEL ADMINISTRADOR DEL SISTEMA>: Dirección de correo a la cual se enviarán notificaciones en caso de algún problema con el funcionamiento del servidor.

<NOMBRE DEL SERVIDOR>: Nombre del host y puerto que el servidor usa para identificarse a él mismo. Sintaxis: ServerName [scheme://]fully-qualified-domain-name[:port]. Ejemplo: goomed.com

<CAMINO_A_GOOMED> es el directorio donde se encuentra GOOMED.

En Windows 7:

Sustituirlo por goomed/public/ y editar el DocumentRoot en el archivo httpd.conf en “C:\Archivos de programa\xampp\apache\conf\”.

Ejemplo: Poner en el DocumentRoot:

"C:/Archivos de Programa/xampp/htdocs/goomed/public"

En Windows XP:

Sustituirlo directamente por el directorio donde se encuentra goomed.

Ejemplo: “C:\Archivos de programa\xampp\htdocs”.

Para completar la creación del Servidor Virtual debe asegurarse que el valor de la directiva ServerName y el IP asociado se encuentran en “C:\WINDOWS\system32\drivers\etc\hosts” o en el servidor de DNS de su subred. A continuación se muestra como debe quedar configurado el archivo *hosts* para el ejemplo anterior:

```
127.0.0.1  goomed
```

O simplemente puede sustituirlo por “hosts” archivo que le adjuntamos en este directorio.

Más información sobre la configuración de un Servidor Virtual en:
http://httpd.apache.org/docs/2.2/vhosts/

9. Reiniciar los servicios de Apache y Tomcat. Para comprobar la correcta instalación visitar el URL configurado en el Host Virtual. Para el ejemplo anterior sería: http://goomed.

Anexo 4.

Para el correcto funcionamiento del sistema se necesita los siguientes requerimientos de software:

- PHP 5.2.6 o superior.
- Apache 2.2.9 o superior.
- MySQL 5.x.x.

Nota: Podría instalarse cualquier SGBD, solamente requiere la correcta configuración dentro de la sección de <database> en los archivos de configuración de Goomed.

- JRE 1.5 o superior.
- Apache Tomcat 6.0.20 o superior
- SOLR 1.3 o superior