Universidad de las Ciencias Informáticas

Facultad 1

"RANKING DE DOCUMENTOS EN LA ARQUITECTURA OAI-PMH"

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Autor:
Antonio Ramírez Pupo.

Tutores:
Ing. Maikel Manuel Fernández Fernández.
Ing. Luis Domínguez Cruz.

La Habana, junio de 2013

DECLARACIÓN DE AUTORÍA.

Por este medio declaro ser el único autor de la presente tesis y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

_____

Autor

Antonio Ramírez Pupo.

_____

Tutor

Ing. Maikel Manuel Fernández Fernández.

_____

Tutor

Ing. Luis Domínguez Cruz.

La Habana, junio de 2013

DEDICATORIA.

A mis abuelos Antonio y Caridad, por depositar toda su confianza en mí y por ser para ellos la imagen eterna de mi padre. A ustedes va dedicado este trabajo como muestra de mis más sinceros agradecimientos por todos sus esfuerzos dedicados hacia mí durante tantos años.

La Habana, junio de 2013

AGRADECIMIENTOS.

A mis abuelos Antonio y Caridad por su amor infinito, su apoyo y dedicación desde el primer día en que formé parte de este mundo, por haberme inculcado sus principios y haberme guiado por el camino correcto hasta lograr convertirme en quien hoy soy.

A toda mi familia, en especial a mi mamá, mi tía Maricel, mis tíos Tito y Carlos, mi padrastro Daciel que ha sido como un padre para mí, a mi esposa Araceli por haber conocido durante mi etapa universitaria a lo más lindo que me ha pasado en toda mi vida. A todos ellos gracias por confiar siempre en mí y por haberme apoyado a lo largo de todos mis estudios.

A mis vecinos y amigos del barrio por su actitud siempre tan atenta, en especial a Nelson y Aida.

A mis tutores, principalmente al profe Maikel por su dedicación y ayuda en todo el desarrollo de este trabajo.

A todos mis compañeros de estudios y amigos, en especial a Héctor, Ricardo, Daniel y Yadir, por brindarme su amistad incondicional en todo momento. Por ayudarme a cumplir metas que a veces veía inalcanzables, a ellos les estaré agradecido siempre.

A todas esas personas que de una forma u otra han influido en mi vida de forma positiva y aunque no ponga todos los nombres, aquí dejo plasmado mis eternos agradecimientos.

La Habana, junio de 2013

RESUMEN.

En el presente trabajo se desarrolla un método de ranking de documentos que tiene en cuenta las variables autor, actualidad e impacto para mejorar la relevancia de los documentos durante la recuperación de información en el repositorio institucional de la Universidad de las Ciencias Informáticas, el cual utiliza el protocolo OAI-PMH (*Open Archives Initiative Protocol for Metadata Harvesting*) para la recolección de metadatos. Para el desarrollo de este método se hace un estudio de los principales métodos utilizados para realizar ranking, así como del entorno donde se encuentra funcionando el repositorio, aprovechando al máximo las características que ofrece de forma tal que contribuya a la realización de dicho método.

Se estudia también diversos sistemas y artículos donde se refleja la importancia de la aplicación del ranking en los Sistemas de Recuperación de Información. Se hace uso de diversas herramientas y tecnologías para lograr con mayor facilidad la implementación de la propuesta dada y se exponen un conjunto de figuras y diagramas que ayudan a tener un mejor entendimiento acerca del tema tratado. Por último se valida el resultado mediante el análisis de la precisión y el tiempo de respuesta del sistema al cual se le aplica el método de ranking implementado.

Palabras clave:

algoritmo, documentos, OAI-PMH, ranking, recuperación de información.

ÍNDICE

ÍNDICE DE FIGURAS.

ÍNDICE DE TABLAS.

INTRODUCCIÓN.

El acelerado desarrollo de las Tecnologías de la Información y las Comunicaciones ha proporcionado un incremento en el flujo de la documentación académica y científica en formato digital. Debido a este notable aumento de la información surgen ideas para garantizar el acceso libre a la misma sin tener que pagar algún tipo de impuesto monetario por su consumo. Actualmente existe un movimiento llamado Acceso Abierto, el cual va encaminado a aumentar el impacto de la investigación al incrementar el acceso a la misma. Este movimiento tiene como objetivo liberar a los usuarios de cualquier privación ante el acceso a la información digital de carácter científico.

El Acceso Abierto, conocido comúnmente como *Open Access* (OA) es una nueva alternativa muy eficaz para el acceso a la información científica, el mismo ha tenido un rápido proceso de desarrollo causado principalmente por situaciones que impedían el avance del conocimiento científico, una de las causas por la que ha tenido este comportamiento es que los precios abusivos de las publicaciones científicas estaban llegando a un nivel económicamente insostenible para las instituciones consumidoras de esa información.

OA a pesar de ser una tendencia importante a seguir en el desarrollo de las tecnologías de la información para el avance del conocimiento científico se enfrentó a una problemática relacionada con la interoperabilidad que es necesaria para lograr garantizar la distribución y el consumo de la información, debido a esto surgieron varias iniciativas y como resultado más importante surge el protocolo para la transmisión de metadatos OAI-PMH.

La tecnología OAI-PMH se caracteriza por su sencillez. Divide el fenómeno en proveedores de datos y proveedores de servicios y se basa en HTTP y XML, además de abogar por el uso de *Dublin Core* como estándar de metadatos para distribuir los documentos.(1)

Debido al crecimiento de la documentación académica y científica se han implementado sistemas bajo una arquitectura OAI-PMH que faciliten a los usuarios la recuperación de información, algunos ejemplos de estos sistemas son las bibliotecas digitales y repositorios institucionales, estos son los llamados Sistemas de Recuperación de Información (SRI). Para mejorar la precisión durante la recuperación de la información estos sistemas emplean algoritmos matemáticos que posibilitan la realización de un ranking entre los documentos mostrados en la búsqueda que se realiza. Actualmente existen varios algoritmos enfocados a la realización de un ranking en un sistema, entre ellos se encuentran el HITS y el *PageRank*.

Entre los proveedores de documentación científica las universidades juegan un papel protagónico debido al gran volumen de documentación que se genera. Hoy en día existen muchas universidades que exponen su documentación en repositorios con características OA y utilizando el protocolo OAI-PMH como herramienta de interoperabilidad.

La Universidad de Ciencias Informáticas (UCI), entidad dedicada a la formación de profesionales altamente calificados en el área de la informática y enfocada al desarrollo de soluciones informáticas, maneja un gran volumen de documentación científica y posee un repositorio institucional para la recuperación de información con una arquitectura OAI-PMH en el cual se expone una gran parte de dicha documentación para el consumo de toda la comunidad universitaria.

Actualmente para la realización de las búsquedas en este repositorio es utilizado el motor de búsqueda de la herramienta para la gestión de repositorios documentales *DSpace*. Este motor de búsqueda no cuenta con algoritmos matemáticos o estadísticos definidos para contribuir a la realización del ranking entre los documentos expuestos, por lo que el ranking que se hace solo es por similitud y no se tiene en cuenta otros importantes parámetros que realmente afectan el nivel de importancia de los documentos. Estos parámetros son: el autor, la actualidad y el impacto.

Por lo anteriormente visto, el presente trabajo plantea el siguiente **problema de investigación.**

¿Cómo incluir en el ranking de documentos, durante la recuperación de información las variables autor, actualidad e impacto?

De acuerdo al problema planteado se propone como **objetivo general** implementar un método de ranking de documentos mediante la inclusión de las variables autor, actualidad e impacto para aplicarlo en la recuperación de información, en un ambiente OAI-PMH.

Para dar cumplimiento al objetivo general se plantean los siguientes **objetivos específicos**.

- Elaborar el marco teórico conceptual de la investigación.
- Proponer un método de ranking que tenga en cuenta las variables autor, actualidad e impacto.
- Implementar el método de ranking propuesto sobre la recuperación de información en la arquitectura OAI-PMH.
- Evaluar los resultados de la aplicación del modelo propuesto.

El **objeto de estudio** de la investigación se centra en los métodos utilizados para el ranking de documentos y el **campo de acción** se enmarca en el algoritmo *PageRank*.

Preguntas de investigación.

1. ¿Si se tiene en cuenta las variables autor, actualidad e impacto mejorará la precisión en los resultados de las búsquedas?
2. ¿Se verá comprometido el rendimiento del sistema en cuanto a tiempo de respuesta si se emplea un modelo ranking diferente al ranking por similitud?

Métodos de investigación.

Métodos teóricos.

- **Analítico sintético:** se utiliza este método para identificar las distintas partes que intervienen en el problema, analizarlas por separado y luego integrarlas para tener una visión global del problema.
- **Histórico lógico:** este método se usa para analizar el desarrollo histórico de la aplicación de los modelos de ranking de documentos en la recuperación de información con características OAI-PMH, de forma tal que se facilite la obtención de información sobre estos modelos, así como cambios a que han sido sometidos los mismos a través de los años y las nuevas soluciones, mejoras y aportes a los ya existentes.

Métodos empíricos.

- **Observación:** se utiliza para observar el comportamiento de los sistemas que implementen algoritmos de ranking así como de trabajos relacionados con estos algoritmos y para conocer los elementos fundamentales relacionados con los temas abordados en la investigación.
- **Constatación:** mediante este método se hace una comparación entre los resultados obtenidos antes de aplicar el algoritmo y los resultados que se obtienen después de aplicarlo.

Justificación de la investigación.

Esta investigación se realiza para incluir en el ranking de documentos que se hace en el repositorio institucional de la UCI los elementos autor, actualidad e impacto, ya que los métodos que tiene actualmente dicho repositorio para la realización de las búsquedas no tienen en cuenta estos elementos que realmente afectan el nivel de importancia de los documentos.

Para darle una mejor organización al contenido, el documento está estructurado en 3 capítulos los cuales son descritos a continuación.

Capítulo 1. Marco teórico-conceptual.

En este capítulo se realiza un estudio sobre la arquitectura OAI-PMH así como de los algoritmos matemáticos para la realización de modelos de ranking de documentos en los Sistemas de Recuperación de Información con arquitectura OAI-PMH, se ofrece información referente a los principales conceptos tratados y también se dan a conocer las herramientas y el lenguaje de programación a utilizar para la implementación del método.

Capítulo 2. Descripción de la propuesta.

En este capítulo se hace un estudio del algoritmo definido para la solución del problema y se realiza una descripción del mismo, además se procede a identificar los requerimientos para realizar la implementación de la propuesta de solución y por último se exponen los diversos diagramas relacionados con el análisis de la propuesta para llevar a cabo la realización de la solución.

Capítulo 3. Pruebas y resultados.

En este capítulo se da a conocer todo lo relacionado con las pruebas para validar la aplicación del método propuesto. Se realizan mediciones de variables como la precisión antes y después de que se le aplica el método de ranking a un sistema que realiza la recuperación de información por similitud. Estas pruebas se realizan mediante experimentos que permiten arrojar resultados.

CAPÍTULO 1. MARCO TEÓRICO-CONCEPTUAL.

1.1 Introducción.

En este capítulo se hace un estudio de los principales conceptos relacionados con la Recuperación de Información (RI) para consolidar los conocimientos teóricos acerca de esta área. Se realiza también un estudio sobre la problemática en la que se enfoca la RI, se expone cómo se evidencia el ranking y se explican los modelos y técnicas que son usados en la RI. Para un mejor entendimiento del problema se hace un análisis de la arquitectura que presenta el entorno OAI-PMH así como de las aplicaciones del ranking en la actualidad. Por último se describen las herramientas y tecnologías a utilizar para el desarrollo de la propuesta.

1.2 La problemática de la Recuperación de Información.

En las ciencias de la computación existe un área llamada Recuperación de Información en inglés conocida como (*Information Retrieval*) que estudia y propone modelos y algoritmos para solucionar la necesidad de los usuarios relacionada con el acceso y el consumo de importantes volúmenes de información.

¿Qué es la Recuperación de Información?

Hace varios años Fernando R. A. Bordignon y Gabriel H. Tolosa (Bordignon y Tolosa) autores de un artículo de la revista electrónica de estudios telemáticos Télématique[1] hicieron alusión a algunos conceptos enunciados por varios especialistas en el tema de la RI, dos de estos especialistas son Ricardo Baeza-Yates y Berthier Ribeiro-Neto, ellos plantearon que "*la Recuperación de Información trata con la representación, el almacenamiento, la organización y el acceso a ítems de información*". (2)

Otros de los autores que enunció un concepto para la RI fue Gerard Salton, este autor propuso una definición amplia que plantea que el área de RI "*es un campo relacionado con la estructura, análisis, organización, almacenamiento, búsqueda y recuperación de información*". (3)

Korfhage definió la RI como "*la localización y presentación a un usuario de información relevante a una necesidad de información expresada como una pregunta*" (3). Actualmente la RI toma un importantísimo rol debido al constante crecimiento de la información electrónica, se puede afirmar que encontrar o no la información necesitada en el tiempo previsto puede resultar en el fracaso

[1]Revista electrónica, científica, arbitrada, de periodicidad cuatrimestral, que publica artículos de índole científico y técnico en el área de la telemática (Telecomunicaciones e Informática).

de una operación realizada (3), aquí entra en evidencia la importancia de los Sistemas de Recuperación de Información. Según Jose Luis Castillo Sequera "*los Sistemas de Recuperación de Información (SRI) son una clase de sistemas de información que tratan con bases de datos compuestas por documentos y procesan las consultas de los usuarios permitiéndoles acceder a la información relevante en un intervalo de tiempo apropiado*". (4)

Debido a varios estudios realizados los autores Bordignon y Tolosa plantean que "*la Recuperación de Información intenta resolver el problema de encontrar y rankear documentos relevantes que satisfagan la necesidad de información de un usuario expresada en un determinado lenguaje de consulta*" (3). Sin embargo existe un problema relacionado con poder compatibilizar y comparar el lenguaje en que se encuentra expresada la necesidad de información y el lenguaje de los documentos lo cual hace que sea un poco difícil resolver este problema.

Viendo esto de una manera más general Baeza-Yates plantea que el problema de la Recuperación de Información puede ser analizado partiendo desde dos puntos de vista fundamentales: el computacional y el humano. El primer caso tiene que ver con la construcción de estructuras de datos y algoritmos eficientes que mejoren la calidad de las respuestas. El segundo caso corresponde al estudio del comportamiento y de las necesidades de los usuarios. (3)

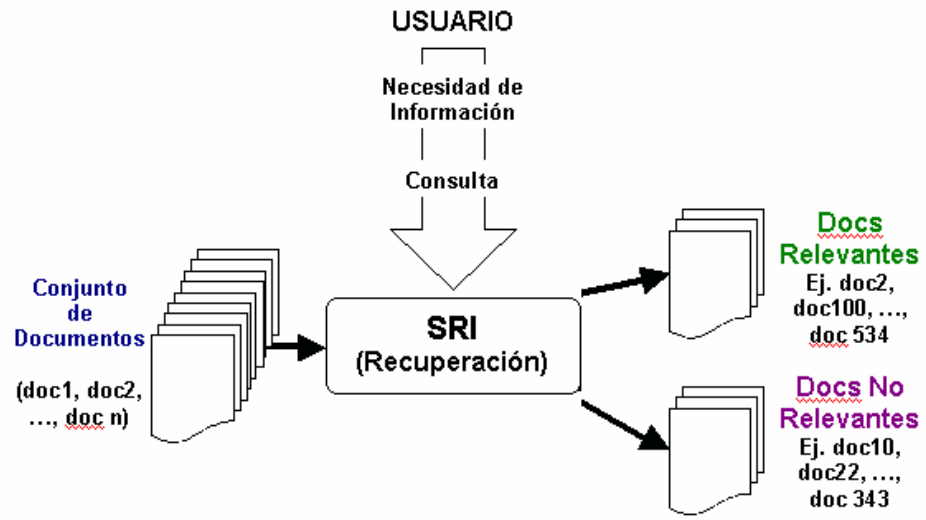

Figura 1 – Problemática de la RI. (3)

Analizando la problemática desde una forma visual se puede apreciar que existe un gran volumen de documentos que contienen información de interés, además existen usuarios con necesidad de consumir esa información y le plantean esa necesidad al SRI en forma de consulta,

el sistema como respuesta “ideal” debería ofrecer los documentos relevantes expresados generalmente en forma de una lista rankeada. Según Bordignon y Tolosa se puede afirmar entonces que la respuesta ideal de un SRI estaría formada solamente por los documentos relevantes a la consulta, pero en la práctica esto no se cumple. En la Figura 1 se puede observar que el sistema responde tanto con documentos relevantes como con no relevantes.

1.3 El ranking dentro de la Recuperación de Información.

Dentro del área de RI se maneja permanentemente el concepto de relevancia ya que este concepto forma parte del funcionamiento de los SRI. Las respuestas de los SRI se encuentran conformadas de acuerdo a uno o varios criterios que evalúe la similitud entre los documentos y las consultas realizadas, por lo tanto de alguna manera los resultados se mostrarán de forma rankeada donde la primera posición corresponde al documento más relevante a la consulta, de esta forma aparece en la RI el término “ranking“.

En el ámbito de la RI, “*un ranking es un ordenamiento de los documentos recuperados que refleja su relevancia para el usuario*“ (5) y su problema está enfocado en la forma de ordenar los resultados de una búsqueda mediante uno o varios criterios definidos. Anteriormente se menciona que un SRI como respuesta ideal ante una búsqueda retorna una colección de documentos relevantes para el usuario, pero para que este logre alcanzar ese objetivo primeramente debe realizar un rankeo de los documentos considerados relevantes para formar el conjunto solución o respuesta, una vez que el sistema haga esto recupera la mayor cantidad posible de documentos relevantes, minimizando la cantidad de documentos no relevantes (ruido) en la respuesta. En términos de eficiencia, se plantea la idea de precisión de la respuesta, es decir, cuando más documentos relevantes contengan el conjunto solución (para una consulta dada), más preciso será. (3)

Existen varias formas de rankear los resultados de una búsqueda, frecuentemente se calcula usando la similitud entre la consulta y el documento pero también se puede calcular el ranking usando otros factores como las citaciones y las referencias. El ranking juega un rol fundamental en el proceso de recuperación de información, debido a su importancia en este proceso es considerado como un componente de la RI formando parte de la arquitectura básica de un SRI. (Figura 2).

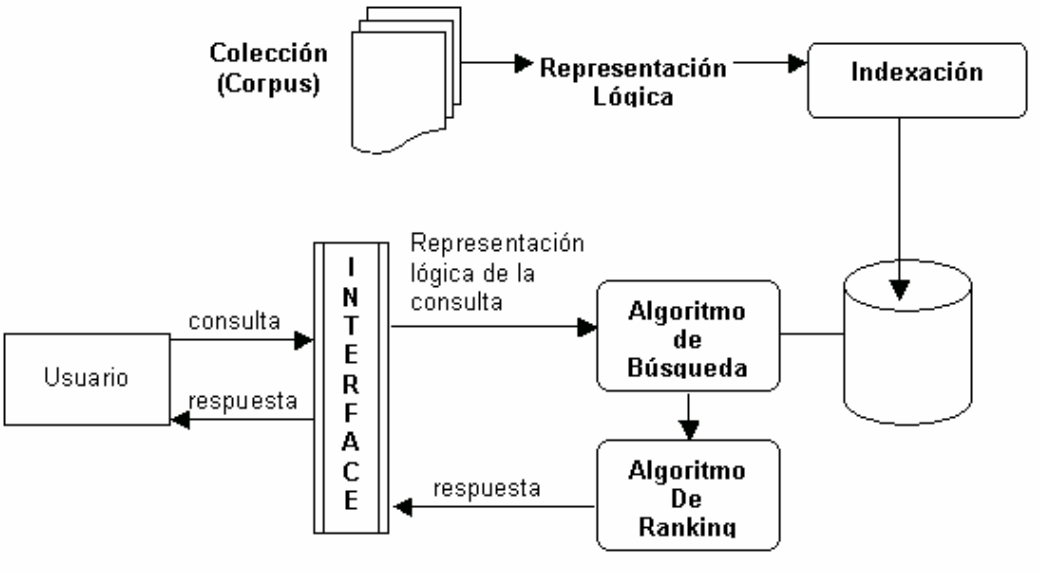

Figura 2 – Arquitectura básica de un SRI. (3)

1.4 Modelos y algoritmos de ranking en la Recuperación de Información.

En la RI existen varios modelos que propician la realización del ranking, entre estos modelos se encuentran el modelo del espacio vectorial y el modelo probabilístico.

1.4.1 Modelo del espacio vectorial.

El modelo de recuperación vectorial o de espacio vectorial propone un marco en el que es posible el emparejamiento parcial asignando pesos no binarios a los términos índice de las preguntas y de los documentos. Estos pesos de los términos se usan para computar el grado de similitud entre cada documento guardado en el sistema y la pregunta del usuario. (6)

En el modelo vectorial tanto los documentos como las consultas se representan mediante conjuntos ordenados de números, pueden tratarse matemáticamente como vectores en un espacio *t* dimensional. Este modelo propone evaluar el grado de similitud entre los documentos de una colección y las consultas mediante algún criterio que muestre la mayor o menor cercanía entre los vectores correspondientes a los documentos y el vector correspondiente a la consulta. (7)

Una vez calculada la similitud entre cada documento de la colección y la consulta, el sistema es capaz de ordenar todos los documentos de la colección en orden descendente de su grado de similitud con la consulta, incorporando de este modo a los resultados aquellos documentos que satisfacen solo parcialmente los términos de la consulta. (7)

1.4.2 Modelo probabilístico.

El modelo probabilístico se basa en el cálculo de la probabilidad de un documento de ser relevante a una consulta dada. Este modelo actúa precisamente sobre los términos que conforman la consulta del usuario imponiéndoles un peso o número a cada uno de ellos, mayor cuanto mejor permita discernir los documentos relevantes de los irrelevantes, y menor en caso contrario. De esta manera el sistema efectúa la recuperación incidiendo sobre todo en los mejores descriptores de entre los términos empleados por el usuario en la consulta, minimizando la importancia de aquellos otros términos que estando en la consulta, son malos descriptores del conjunto respuesta ideal. (7)

En este proceso no se puede definir de entre los términos que conforman la consulta, cuáles son los buenos descriptores y cuáles no lo son, por lo que el modelo considera, para cada uno de los términos empleados en la consulta, la "probabilidad de que el término sea un buen descriptor" (probabilidad de que el término empleado en la consulta esté presente en un documento del conjunto de documentos relevantes en relación a la consulta) y simultáneamente, para ese mismo término, la "probabilidad de que sea un mal descriptor" (probabilidad de que ese mismo término esté presente en un documento del conjunto de documentos irrelevantes en relación a la consulta).

Estas probabilidades para cada uno de los términos empleados en la consulta son desconocidas en el momento de formalizar dicha consulta, para resolver este problema el modelo efectúa inicialmente una hipótesis sobre sus valores. Luego el modelo probabilístico basado en los pesos iniciales asignados a cada término, es capaz de calcular el grado de similitud existente entre cada documento de la colección y la consulta ponderada, ordenando los documentos de la colección en un orden decreciente de probabilidad de relevancia con relación a la consulta.

1.4.3 Algoritmos para realización de ranking.

Los SRI en función de mejorar la precisión en las respuestas emplean diversas técnicas y algoritmos matemáticos que les posibilitan realizar un ranking. Entre los algoritmos que existen para la realización de un ranking se encuentran el *PageRank* y el HITS.

1.4.3.1 Algoritmo *PageRank*.

El algoritmo *PageRank* (PR) fue el método inicial de cálculo que usaron los fundadores de Google[2] para clasificar las páginas web según su nivel de importancia (8). El *PageRank* modela la Web como un grafo dirigido donde los nodos corresponden a las páginas HTML y los enlaces o hipervínculos entre estas son las aristas. Este algoritmo asigna un valor de importancia a todas las páginas del grafo web, independientemente de alguna consulta realizada. Sus autores lo proponen como un modelo del comportamiento de los usuarios, donde estos pueden seleccionar al azar un enlace dentro de una página (navegante aleatorio), cuya probabilidad depende de la cantidad de enlaces salientes de esta página, determinándose que la probabilidad de que un determinado navegante aleatorio alcance una página es la suma de las probabilidades de que siga los vínculos hacia esta. (9)

El algoritmo ha tenido modificaciones para mejorar su precisión, a continuación veremos las ecuaciones correspondientes a su primera y segunda versión.

$$PR_p = (1-d) + d * \sum_{pi \to p} \frac{PR(p_i)}{CT(p_i)} \quad (1)$$

$$PR_p = \frac{(1-d)}{N} + d * \sum_{pi \to p} \frac{PR(p_i)}{CT(p_i)} \quad (2)$$

Donde:

$\boldsymbol{PR_p}$ es el *PageRank* de la página *p*.

$\boldsymbol{PR(P_i)}$ es el *PageRank* de las páginas P_i que poseen enlace a la página *p*.

$\boldsymbol{CT(P_i)}$ es la cantidad de enlaces salientes de P_i

d es un factor de *damping* que representa la probabilidad de que el navegante aleatorio no se detenga, toma entre 0 y 1.

N es el número total de páginas de la Web.

1.4.3.2 Algoritmo HITS.

El algoritmo *Hyperlink Induced Topic Search* (HITS) fue desarrollado por Jon Kleinbergen el año 1999 (10) para determinar la importancia de una página en función de los enlaces que posee y de los que recibe. En el algoritmo HITS se plantean los conceptos de centros y autoridades (*Hubs* y *Authorities*) con el objetivo de clasificar una página web.

[2]Google: Compañía estadounidense fundada en septiembre de 1998 cuyo producto principal es un motor de búsqueda

El concepto *Hubs* es asociado con las páginas web que posee una importante cantidad de enlaces a páginas con contenidos relevantes. Por su parte una página *Authority* es aquella que recibe una importante cantidad de enlaces y pocos salen de ella. Esta característica presupone que tales páginas son referentes en un tema específico.(9)

El algoritmo funciona de la siguiente manera, para un conjunto *P* de páginas web, con su correspondiente conjunto *A* de aristas, las cuales determinan el grafo web, HITS realiza el ranking para cada página *p* que pertenece a *P* a partir de su condición como *Authority* (X_p) y *Hub* (Y_p).

Los valores de se calculan de la siguiente forma:

$$Authority_p = x_p = \sum_{q|q\to p} y_q \quad (3)$$

$$Hub_p = y_p = \sum_{q|q\to p} x_q \quad (4)$$

1.5 El ranking en la recuperación de información académica y científica.

Para la recuperación de información académica y científica existen otras técnicas que de alguna manera permiten obtener elementos favorables para realizar un ranking entre estos documentos académicos y científicos. Entre las técnicas que se utilizan para este caso se encuentran el índice *h* y el factor de impacto.

1.5.1 Índice *h*.

El índice *h* de Hirsch es un sistema de medida que permite detectar a los investigadores más destacados dentro de un área de conocimiento, fue propuesto por Jorge Hirsch de la Universidad de California a mediados del 2005. Su cálculo es sencillo, consiste en ordenar los documentos de un investigador en orden descendente de número de citas recibidas, numerarlas e identificar el punto en el que el número de orden coincida con el de citas recibidas por documento (11). Viendo esto de una forma mas sencilla se puede decir que un científico tiene índice *h* si ha publicado *h* trabajos con *h* citas cada uno.

Con relación a este tema del índice *h* como elemento fundamental para el campo de la investigación científica existen muchas tramas, no siempre los artículos son citados por sus aportes a la ciencia, sino que a veces muchas de las citas que reciben son gracias a las redes de

citas, esto se refiere a investigadores que se citan mutuamente ya sea por compañerismo o gratitud.

1.5.2 Factor de impacto.

Es un instrumento para comparar revistas y evaluar la importancia relativa de una revista concreta dentro de un mismo campo científico (12). El factor de impacto proporciona información sobre el número de veces que se cita por término medio un artículo publicado en una revista determinada. Actualmente uno de los criterios que se consideran para juzgar la calidad de una publicación es el índice de impacto de la revista en la que aparece (11). A continuación se muestra un ejemplo de como se calcula el factor de impacto de una revista:

- *A* = Número de veces que se han citado durante el año 2013 artículos publicados por la revista *X* durante el período 2011-2012
- *B* = Número de artículos publicados en la revista *X* durante el período 2011-2012
- *C* = Factor de impacto de la revista *X* en 2013: *C = A/B*

1.6 El entorno OAI-PMH.

El protocolo OAI-PMH "*es una sencilla interfaz que hace posible el acceso a metadatos de contenidos de distintas fuentes*"*,* genera y promueve estándares que facilitan la difusión, el intercambio y la accesibilidad a documentos, para esto se apoya fundamentalmente en la creación de repositorios. (1)

OAI surge a partir de la necesidad de comunicación entre diversos repositorios de documentos electrónicos creados por varias disciplinas. La reunión de Santa Fe, Nuevo México, USA en 1999, marca el punto de partida de la creación del OAI y allí se definen como puntos fundamentales los estándares de metadatos y el protocolo de comunicación. En el 2000 la *Digital Library Federation* (DFL) y *Coalition of Networked Information* (CNI) asumen la organización de la iniciativa, creando un comité técnico y uno de gestión, surge así la versión 1.0. El protocolo ha seguido su avance hasta la versión 2 (1). A partir de la implementación del protocolo las primeras instituciones comenzaron a utilizarlo para poner sus metadatos en Internet.

El protocolo OAI-PMH posee enfoques de interoperabilidad mediante la búsqueda distribuida y la recopilación. El primero consiste en buscar y descubrir información y servicios remotos, por otro lado la recopilación se basa en que los metadatos son transferidos desde la fuente remota hacia el destino en el cual se realizarán los servicios de búsqueda, por ejemplo la unión de catálogos.

Como se mencionó anteriormente este protocolo divide el fenómeno en dos roles, proveedores de datos y proveedores de servicios. Los proveedores de datos albergan un repositorio con los recursos que se quieren publicar y exponen los metadatos de dichos recursos para ser recuperados por los proveedores de servicios. Los proveedores de servicios recuperan metadatos de los proveedores de datos y los utilizan para dar servicios sobredichos datos (interfaz de búsqueda) (13). Para establecer la comunicación entre el proveedor de datos y el proveedor de servicios, OAI establece una lista de verbos los cuales son interpretados por el proveedor de datos para emitir la respuesta. Estos verbos son:

Identify: pregunta por la identidad de una fuente, la respuesta da información del nombre del repositorio, de la granularidad de la fecha, de la fecha de inicio del repositorio y de la identidad en sentido general. (1)

ListMetadataFormats: pregunta por los estándares de metadatos con que están descritos los documentos en la fuente encuestada, la respuesta da una lista de los estándares de metadatos. (1)

ListSet: encuesta sobre la estructura del repositorio, la respuesta contiene la estructura temática del repositorio. (1)

Listrecords: pide el listado de los documentos, debe incluir la variable **metadataPrefix**, el valor asignado a esta variable es uno de los resultados del **ListMetadataFormats**. (1)

ListIdentifier: es una forma abreviada del **ListRecords**, pero la respuesta solo recupera los encabezados. (1)

Getrecord: pide un recurso específico, para ello usa la variable **identifier**, cuyo valor se obtiene del **ListIdentifier**, además la variable **metadataPrefix**. (1)

Para complementar la comunicación existen otras variables como: **from** y **until** para una recopilación basada en fechas, **set** para recopilación basada en materias y **resumptionToken** para el control del flujo. (1)

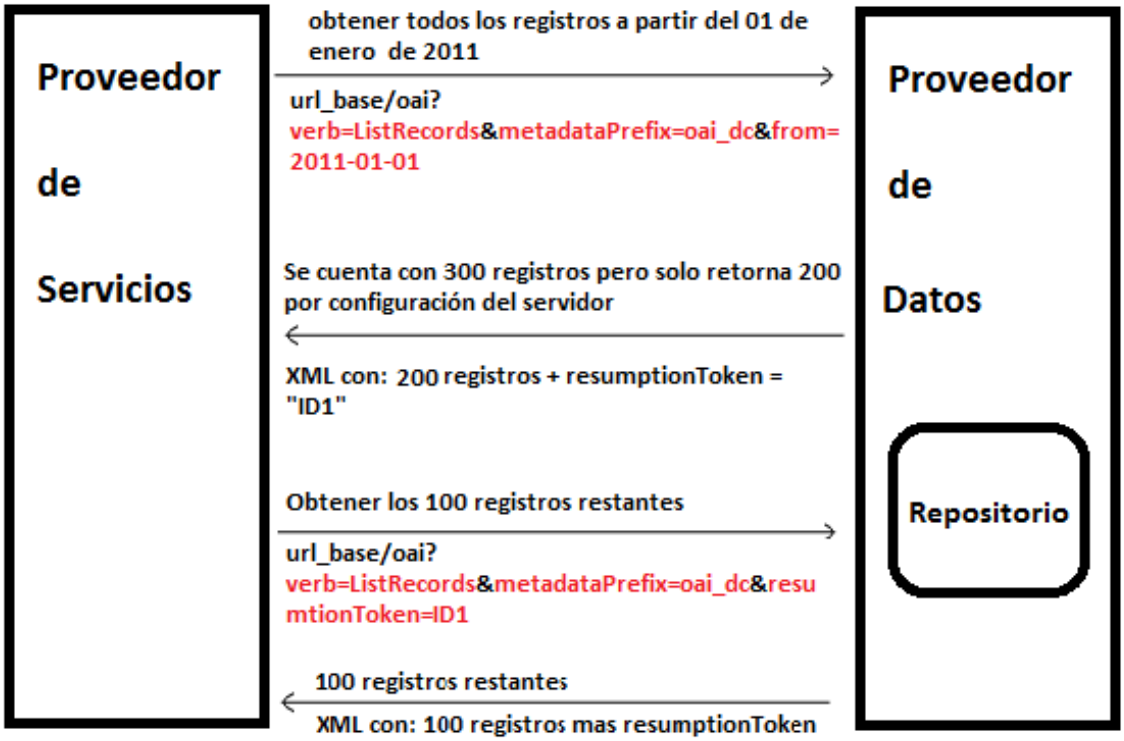

Figura 3 – Ejemplo de una comunicación entre un proveedor de servicios y un proveedor de datos. (1)

OAI-PMH funciona sobre el protocolo HTTP, las peticiones son operaciones HTTP, GET o POST y las respuestas son documentos XML válidos.

El protocolo OAI-PMH hace uso del estándar de metadatos *Dublin Core* para la distribución de los documentos. La iniciativa *Dublin Core* pretende desarrollar estándares de metadatos para la recuperación de información en Internet a través de distintos dominios así como facilitar el desarrollo de conjuntos de metadatos específicos de una disciplina o comunidad que trabaja dentro del marco de la RI. A continuación de describen los metadatos que componen dicho estándar:

Elementos del contenido.

- **Título (<dc:title>).** Título que lleva del documento.
- **Materia (<dc:subject>).** En este campo se hace referencia a los diversos temas que puede contener el material.
- **Descripción (<dc:description>).** Resumen sobre el contenido del objeto digital.
- **Fuente (<dc:source>).** Ficha bibliográfica que se elabora para asentar los datos sobre la procedencia del documento original.

- **Lenguaje (<dc:language>).** En este campo se ponen las siglas correspondientes al idioma en que está la publicación.
- **Relación (<dc:relation>).** Material principal u objetos de su misma referencia, ya sea una colección, una serie, un documento, etc.
- **Cobertura (<dc:coverage>).** Este campo se refiere al proyecto o sitio donde estará resguardada la información. Aquí pueden anotarse fechas, zonas geográficas.

Elementos de propiedad intelectual.

- **Autor (<dc:creator>).** Autor intelectual de la obra o documento original.
- **Editor (<dc:publisher>).** Sitio o colección a la que está adscrito el material.
- **Colaborador (<dc:contributor>).** En este campo se anotan, si es que se da el caso, el nombre u organización que contribuyó a la creación del material, que no se especificó en la parte de Autor.
- **Derechos (<dc:rights>).** Nombre o la institución a la cual pertenece el material.

Elementos de aplicación.

- **Fecha (<dc:date>).** Fecha de elaboración del registro.
- **Tipo (<dc:type>).** Aquí se menciona la presentación que tiene el objeto digital, ya sea como texto, audio, video, etcétera.
- **Formato (<dc:format>).** Tipo de extensión con que se presenta el objeto digital, ya sea HTML, JPG, GIF o PDF.
- **Identificador (<dc:identifier>).** Se refiere a la dirección electrónica de origen a la que está adscrito el material. Para ello se utilizan las siglas URL.

1.7 Elementos que tributan al ranking.

El ranking de documentos que se realiza está basado principalmente en 3 variables las cuales son el autor del documento, esta variable se relaciona directamente con el metadato <dc:creator>; la actualidad, que se corresponde con el metadato <dc:date>; y el impacto que tenga el documento. Según el autor, el ranking se mide mediante la cantidad de publicaciones que ha tenido y la calidad de las mismas; en el caso de la actualidad se tiene en cuenta la fecha en que fue creado el documento. Por último el impacto se mide a través de la cantidad de visitas de los documentos que son similares al documento en cuestión. Mediante estos indicadores se le calcula un valor de ranking final para cada documento y por este valor es que se le establece la prioridad a los mismos.

1.8 Trabajos relacionados.

Actualmente el ranking es un factor indispensable en el proceso de recuperación de información, es por eso que los SRI que se han venido desarrollando en los últimos años incluyen al ranking dentro de su funcionamiento permitiéndoles a los usuarios una recuperación más fácil y precisa de la información.

Una de las aplicaciones que ha surgido a partir de la iniciativa OA es la segunda versión de la Base de Datos Unificada (BDU2), esta a diferencia de su predecesora (BDU) se encarga de cosechar los recursos disponibles en texto completo en los repositorios institucionales o bibliotecas digitales de universidades y otras instituciones de Argentina, y ponerlos a disposición de los usuarios. Para la implementación de esta aplicación se emplearon tecnologías libres y se innovó en lo referente a los motores de búsquedas utilizando una llamada Apache SolR. Esta tecnología le permite a la BDU2 tener muy buenos tiempos de respuestas ante las consultas realizadas por los usuarios, permite también realizar un ranking por similitud textual de los resultados y clasificarlos por año, por tipo de material, por autor y por palabras clave, lo que posibilita que el usuario recorra mucho más dinámicamente los resultados.

Hoy en día también se realizan numerosas investigaciones con el propósito de desarrollar algoritmos que permitan hacer ranking de acuerdo a uno o varios criterios. Uno de los trabajos enfocados a este objetivo lleva por título *"Study on The Method of Ranking Scientific Papers".* En este trabajo los autores realizan una explicación detallada sobre un método llamado PaperRank para el ranking de artículos científicos. Este método se basa en el algoritmo *PageRank* de Google y realiza el cálculo del ranking teniendo en cuenta importantes parámetros como el contenido del artículo, la revista en que está publicado, el autor y el tiempo de publicación. El método ofrece un nuevo enfoque de investigación en el área de la RI ya que puede ser muy eficiente para aplicarlo en sistemas que necesiten establecer un orden en sus resultados de búsquedas.

Otro de los trabajos realizados donde se expone la realización de un ranking es llamado *"Un Sistema Inteligente para Asistir la Búsqueda Personalizada de Objetos de Aprendizaje"*. En este trabajo se describe el desarrollo de un sistema inteligente que ayuda a un usuario a encontrar los recursos educativos electrónicos que le sean más apropiados, de acuerdo con su perfil. Para eso se implementa un prototipo de un Agente Recomendador (Agente-R) utilizando Inteligencia Artificial mediante el lenguaje SWI-Prolog. Este agente se encarga de realizar una recuperación flexible y presentar una lista ordenada con los mejores recursos, de acuerdo con el perfil del usuario. Para ordenar estos recursos el Agente-R tiene varias reglas para calcular el grado de

satisfacción esperado de cada una de las preferencias del usuario a través de cada objeto, las preferencias son: temática, rol, idioma, interacción y estilo de aprendizaje.

El Agente-R luego de calcular los grados de creencia para cada una de las preferencias elegidas por el usuario, calcula el grado de intención para aquellos objetos que forman parte de la lista a recomendar. Finalmente, el Agente-R ordena los objetos según el valor decreciente del grado obtenido, logrando obtener la lista de objetos de aprendizaje que recomendaría al usuario. (14)

1.9 Herramientas y tecnologías a utilizar.

Para desarrollar de una forma más fácil la propuesta dada se hace uso de un conjunto de herramientas y tecnologías las cuales se mencionan a continuación.

1.9.1 HTML v4.0.

Acrónimo de *Hyper Text Markup Language*, el HTML es un lenguaje de marcación especialmente ideado para permitir la creación de contenidos basados en el hipertexto (15). Es una implementación del *standard* SGML (*Standard Generalized Markup Language*), estándar internacional para la definición de texto electrónico independiente de dispositivos, sistemas y aplicaciones. Metalenguaje para definir lenguajes de diseño descriptivos; proporciona un medio de codificar documentos hipertexto cuyo destino sea el intercambio directo entre sistemas o aplicaciones. (16)

1.9.2 Lenguaje de programación.

Un lenguaje de programación es aquel elemento dentro de la informática que nos permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos *hardware* y *software* existentes (17). Para la implementación del método de ranking de documentos propuesto se usa como lenguaje de programación PHP del cual se realiza una descripción a continuación.

1.9.2.1 PHP v5.3.

Es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor, tiene un estilo clásico, con variables, sentencias condicionales, bucles, funciones. No es un lenguaje de marcas como podría ser HTML, XML o WML. El resultado es normalmente una

página HTML pero igualmente podría ser una página WML. La tecnología PHP posee una estructura de programación, la facilidad de llevar a cabo sentencias SQL embebidas, además de permitir la posibilidad de correr en diferentes tipos de servidores, entre ellos Apache. Quizás la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos. PHP también soporta el uso de otros servicios que usen protocolos como IMAP, SNMP, NNTP, POP3, HTTP y derivados. (18)

1.9.3 CSS v2.0.

Las Hojas de Estilo en Cascada o *Cascading Style Sheets* (CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado, escrito en HTML o XML (19).

1.9.4 Entorno de Desarrollo Integrado (IDE).

Un Entorno Integrado de Desarrollo (IDE, *Integrated Development Environment*) es un sistema que facilita el trabajo del desarrollador de *software*, integrando sólidamente la edición orientada al lenguaje, la compilación o interpretación, la depuración, las medidas de rendimiento, la incorporación de los fuentes a un sistema de control de fuentes, etc., normalmente de forma modular. (20)

1.9.4.1 NetBeans v7.2.

NetBeans es un entorno de desarrollo, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extender el NetBeans IDE (21). Se utiliza este IDE ya que soporta entre otros lenguajes, el lenguaje PHP que es en el que se desarrolla la solución. Además es un producto libre, gratuito sin restricciones de uso, fácil y sencillo de usar.

1.9.5 Servidor web.

Los servidores web son aquellos cuya tarea es alojar sitios y/o aplicaciones, las cuales son accedidas por los clientes utilizando un navegador que se comunica con el servidor utilizando el protocolo HTTP. (22)

1.9.5.1 Apache v2.2.22.

Apache 2.2 es un servidor web de *software* libre desarrollado por la *Apache Software Foundation*

cuyo objetivo es servir o suministrar páginas web (en general, hipertextos) a los clientes web o navegadores que las solicitan (23). Apache es el servidor web hecho por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa.

1.9.6 Sistema Gestor de Base de Datos (SGBD).

Conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad. (24)

1.9.6.1 MySQL v5.5.29.

Es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el *copyright* del código fuente del servidor SQL, así como también de la marca. Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del *software* libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. (25)

1.9.7 Lenguaje Unificado de Modelado (UML).

UML de sus siglas en inglés *Unified Modeling Language* en su traducción al español Lenguaje Unificado de Modelado es uno de los más conocidos y utilizados en la actualidad; aún cuando no es un estándar oficial, está respaldado por el Grupo Administrativo de Objetos (OMG) de sus siglas en inglés *Object Management Group*. Ofrece un estándar para describir un plano del sistema, incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema y aspectos concretos como expresiones de lenguajes de programación, esquemas de base de datos y componentes de *software* reutilizables. (19)

1.9.8 Herramientas CASE.

Las herramientas CASE de sus siglas en inglés *Computer Aided Software Engineering* en español Ingeniería de *Software* Asistida por Ordenador, son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de *software* reduciendo el costo de las

mismas en términos de tiempo y dinero. Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del *software* en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores, entre otras. (19)

1.9.8.1 *Visual Paradigm* v8.0.

Visual Paradigm para UML (VP-UML) es una de las herramientas CASE, que proporciona excelentes facilidades de interoperabilidad con otras aplicaciones, considerada como: muy completa y fácil de usar y con soporte multiplataforma. *Visual Paradigm* para UML está diseñado para una amplia gama de usuarios, incluidos los ingenieros de *software*, analistas de sistemas, analistas de negocios, sistema de arquitectos, al igual que para aquellas personas interesadas en la construcción de sistemas de *software* de forma fiable a través de la utilización del enfoque orientado a objetos. Esta herramienta es muy fácil de instalar y actualizar. También proporciona características tales como: generación del código; permite crear una ingeniería directa como inversa; permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación; está diseñada para usuarios interesados en sistemas de *software* de gran escala con el uso del acercamiento orientado a objeto; incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios realizados por sus compañeros. (19)

1.9.9 OHS v2.3.2.

El OHS de sus siglas en inglés *Open Harvester Systems* es un sistema de indexación de metadatos gratuito desarrollado por el *Public Knowledge Project* a través de sus esfuerzos financiados con fondos federales para ampliar y mejorar el acceso a la investigación. OHS permite crear un índice de búsqueda de los metadatos de la *Open Archives Initiative* (OAI) compatibles con archivos, tales como sitios que utilizan *Open Journal Systems* (OJS) u *Open Conference Systems* (OCS).(26)

Para hacerles las pruebas correspondientes al método desarrollado se utiliza el *software* Matlab, del cual se realiza una descripción a continuación.

1.9.10 Matlab v7.6.

Matlab (abreviatura de *Matrix Laboratory*, laboratorio de matrices), es un *software* matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas *Unix*, *Windows* y *Apple Mac OS X*. Es una poderosa herramienta para la resolución numérica de problemas. (27)

Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes. Matlab dispone además de las herramientas: GUIDE (editor de interfaces de usuario – GUI por sus siglas en inglés) y Simulink (plataforma de simulación multidominio). También se pueden ampliar las capacidades de Matlab con las cajas de herramientas (*toolboxes*); y las de *Simulink* con los paquetes de bloques (*blocksets*). Es un *software* muy usado en universidades y centros de investigación y desarrollo. (27)

1.10 Conclusiones del capítulo.

La realización de un estudio sobre los principales conceptos relacionados con la RI, el entorno OAI-PMH y el ranking en diferentes contextos, así como de los algoritmos y técnicas existentes para realizar ranking permitió enriquecer los conocimientos teóricos necesarios para la realización de la investigación. El estudio de sistemas y artículos científicos donde se evidencia la aplicación del ranking trajo consigo la consolidación de estos conocimientos, permitiendo entender de una mejor forma la importancia que toma el ranking dentro del proceso de recuperación de información y su trascendencia en la actualidad. Por último con el estudio y la selección de las herramientas y tecnologías adecuadas para el desarrollo del método de ranking se logrará realizar de una forma más rápida y fácil esta investigación.

CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA.

2.1 Introducción.

En este capítulo se define el algoritmo para la realización del ranking de los documentos, se realiza una breve descripción sobre su funcionamiento y también se definen las principales características del algoritmo mediante los requerimientos que son necesarios para su implementación. Por último se exponen los principales diagramas relacionados con el análisis de la solución.

2.2 Descripción de la propuesta de solución.

La propuesta de solución es un algoritmo de ranking para documentos académicos y científicos que han sido recolectados por un proveedor de servicios OAI-PMH, en este caso mediante el sistema OHS. La información está almacenada en una base de datos que utiliza como gestor MySQL y de forma serializada. La estructura de los datos responde al estándar de metadatos *Dublin Core*, específicamente para este trabajo se tienen en cuenta los metadatos <dc:title>, <dc:creator>, <dc:subject>, <dc:description>, <dc:date>. Luego de la recuperación de la información por similitud se pasa a la etapa de la realización de ranking basado en las ideas del algoritmo *PageRank* que defiende la importancia de un documento a partir de la importancia de aquellos documentos relacionados con él.

Partiendo de que la idea fundamental sería contar con una red de citas entre documentos que por falta de estandarización rara vez el protocolo distribuye, este trabajo parte de construir una red de interacción entre documentos basada en la probabilidad de llegar a un documento *A* estando en un documento *B* siempre y cuando la similitud entre ambos sea mayor que 0.45. Esta similitud se calcula empleando la métrica del coseno. En la literatura que lleva por título "*Web Structure Mining: Exploring Hyperlinks and Algorithms for Information Retrieval*" (28) se define para el algoritmo *PageRank* un valor de esta probabilidad de 0.85, pero para la propuesta de solución se ha reducido este valor teniendo en cuenta que el dominio del trabajo no es tan extendido como la Web. A partir de la red de similitud y de las visitas recibidas por cada documento se construyen los rankings base, impacto, autor y novedad.

El ranking base de un documento *d* se calcula a partir de las visitas recibidas a ese documento entre la cantidad de visitas recibidas por todos los documentos. El ranking de impacto toma en cuenta los ranking base de aquellos documentos *q* similares a *d* y que se consideran posibles

fuentes de enlaces entrantes, su cálculo es el factor de probabilidad 0.45 por la suma de los rankings base de los documentos *q* que por su similitud hace altamente probable que se llegue a *d,* entre la cantidad de documentos *q.* El ranking de autor toma en cuenta la cantidad de publicaciones del autor, así como la importancia de las mismas. Este ranking se calcula mediante la división entre la cantidad de documentos publicados por el autor y el total de documentos publicados, este valor se le multiplica a la sumatoria de la división entre el ranking base de cada uno de los documentos publicados por el autor y la cantidad total de documentos publicados. Para determinar el ranking de novedad del documento *d* se resta la cantidad de documentos que fueron publicados en el año en que se publicó dicho documento menos la menor cantidad de documentos publicados más 1, ese resultado se divide entre la mayor cantidad de documentos publicados menos la menor cantidad de documentos publicados más 1.

Una vez obtenidos estos rankings para un documento *d*, el valor del ranking final es la multiplicación de los valores de los rankings de impacto, autor y novedad respectivamente. El algoritmo itera varias veces por cada uno de los documentos hasta obtener el ranking final de todos. Estos valores para cada documento se almacenan en la base de datos y cada un determinado período de tiempo (de uno a dos meses) se tienen que actualizar.

2.3 Modelo de dominio.

Para un mejor entendimiento de la propuesta de solución a implementar se propone realizar un modelo de dominio permitiendo visualizar los principales conceptos que se manejan dentro del entorno de dicha propuesta. A continuación se expone el modelo de dominio relacionado con la implementación del algoritmo para el ranking de documentos.

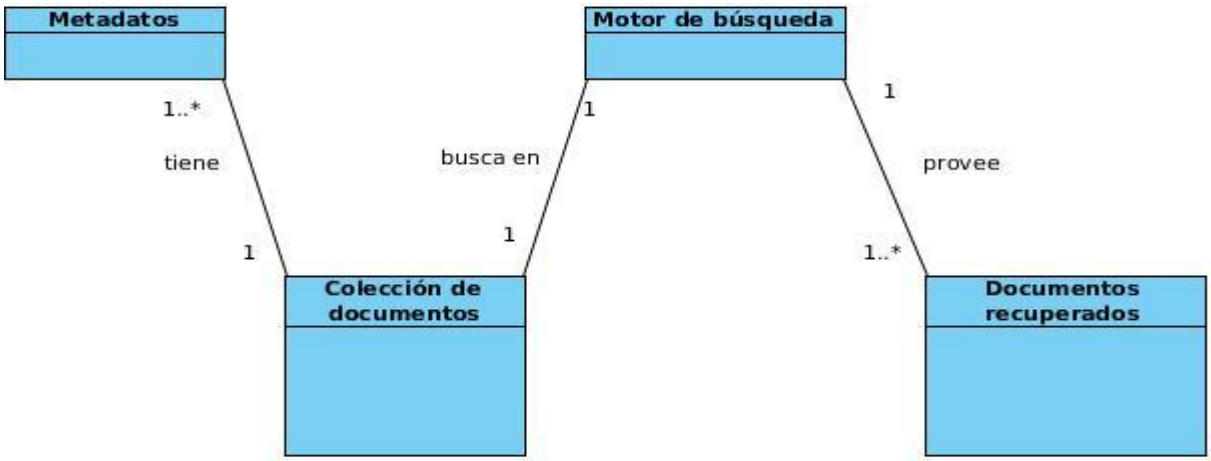

Figura 4 – Modelo de dominio.

2.4 Requerimientos.

A continuación se muestran los requerimientos identificados para la implementación del método propuesto.

2.4.1 Requerimientos funcionales.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir (19). Los requerimientos funcionales necesarios para la implementación del algoritmo son los siguientes:

1. Rankear documentos.
    - Crear índice de autores.
    - Calcular la similitud entre documentos.
    - Calcular cantidad de publicaciones por años.
    - Calcular ranking base.
    - Calcular ranking de impacto.
    - Calcular ranking de autor.
    - Calcular ranking de novedad.
    - Calcular ranking final.

2.5 Diagrama de clases.

Para lograr una implementación más fácil de la propuesta se opta por seguir el paradigma de la Programación Orientada a Objetos (POO), para esto la problemática es modelada mediante clases con sus respectivos atributos y las relaciones existentes entre ellas. El diagrama de clases perteneciente a la solución es el que se muestra a continuación.

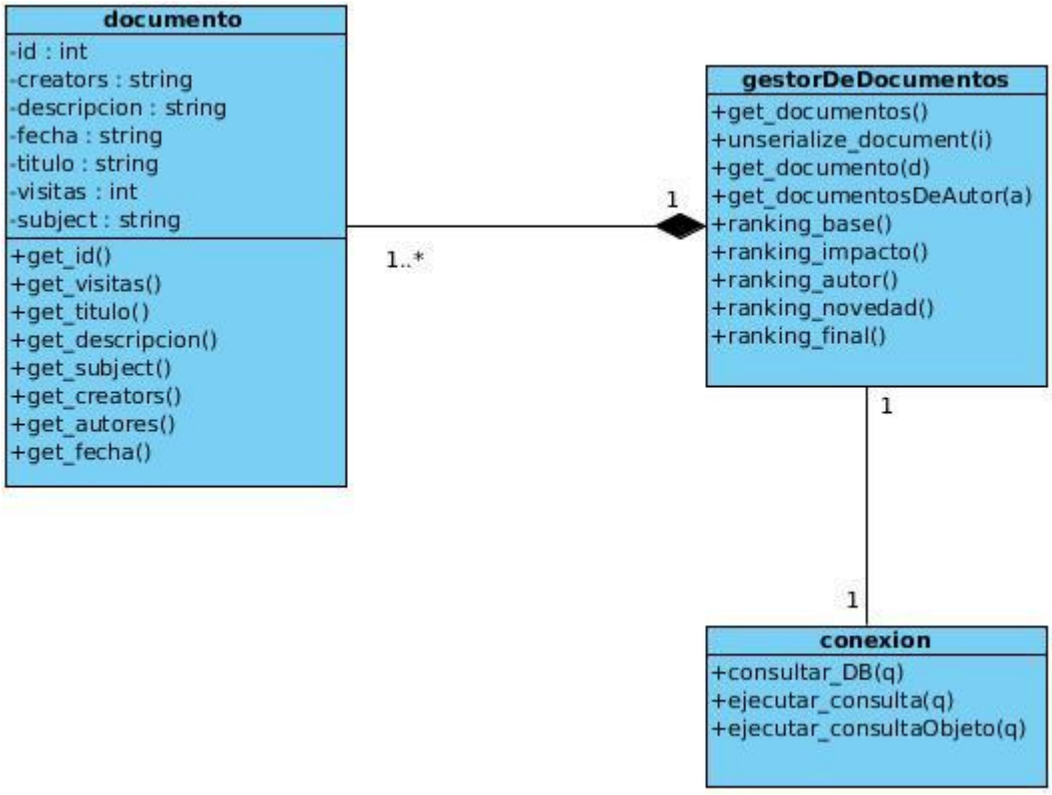

Figura 5 – Diagrama de clases.

2.6 Modelo de datos.

El modelo de datos relacionado con la propuesta de solución cuenta con 2 tablas. En la tabla llamada **records_prueba** se encuentran almacenados los datos referentes a todos los documentos y en la tabla **ranking_documentos** se encuentran los valores de los rankings calculados por el algoritmo para cada uno de los documentos.

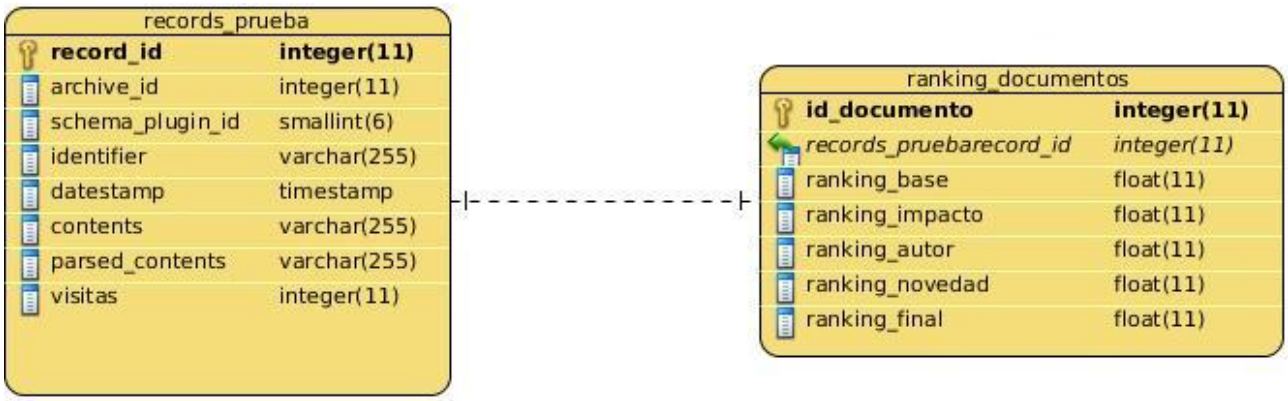

Figura 6 – Modelo de datos.

2.7 Arquitectura.

El patrón arquitectónico que utiliza la solución es el Modelo Vista Controlador (MVC). Este patrón de arquitectura de *software* separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. (29)

La arquitectura de la solución posee la siguiente estructura:

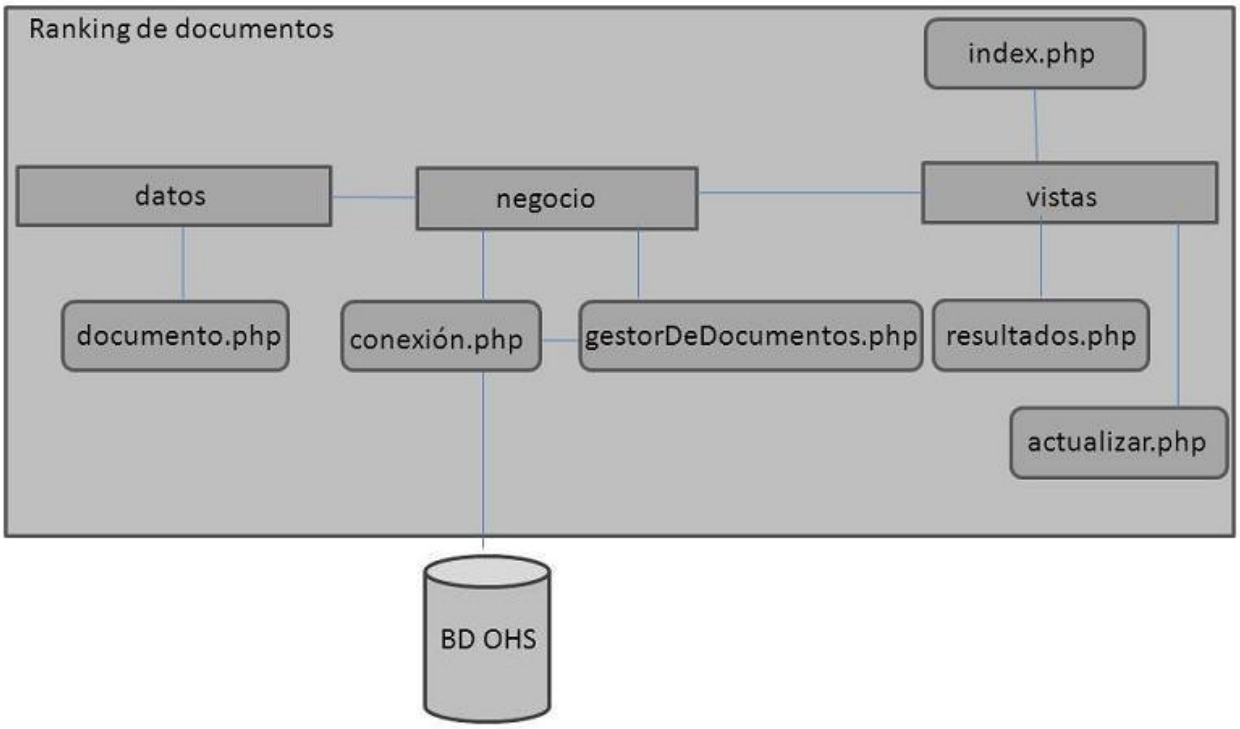

Figura 7– Estructura de la solución.

2.8 Descripción del modelo.

El algoritmo que se propone para el ranking de documentos se basa en realizar el cálculo del ranking para cada documento a partir de otros rankings que se le calculan previamente a dicho documento. A continuación se describe de una forma más detallada las expresiones de cálculo para el desarrollo del algoritmo.

Sean:

d: documento.

Cq: conjunto de documentos *q* que por su similitud hacen altamente probable que se llegue a *d*.

V(d): visitas recibidas en el documento *d*.

V*:* total de visitas recibidas por todos los todos los documentos.

***N**:* total de documentos existentes.

***c**:* factor de probabilidad, 0.45

2.8.1 Cálculo del ranking final de los documentos.

La ecuación general que se define para calcular el ranking a cada documento es la siguiente:

$$R(d) = RI * RA * RN (5)$$

Donde

R(d) es el ranking final del documento *d.*

RI es el ranking de impacto del documento *d.*

RA es el ranking de autor del documento *d.*

RN es el ranking de novedad del documento *d.*

2.8.2 Cálculo del ranking base.

Ecuación para el cálculo del ranking base, toma en cuenta las visitas recibidas de los documentos.

$$I(d) = \frac{V(d)}{V} \ (6)$$

Donde:

I(d) es el ranking base del documento *d*

2.8.3 Función del coeficiente del coseno.

El coeficiente del coseno es una función que se utiliza para el cálculo de la similitud para cada par de documentos, la ecuación correspondiente a dicha función es la siguiente:

$$(d_i, d_j) = \frac{\sum_{h=1}^{k} peso_{ih} * peso_{jh}}{\sqrt{\sum_{h=1}^{k} peso_{ih}^2 * \sum_{h=1}^{k} peso_{jh}^2}} \ (7)$$

Donde ***di*** y ***dj*** son los documentos a comparar, ***k*** es el número de términos que caracterizan los documentos, y peso ***xy*** es el peso del término ***y*** en el documento ***x***, calculado teniendo en cuenta la frecuencia de aparición de ese término en el documento. (30)

2.8.4 Cálculo del ranking de impacto.

En el ranking de impacto influyen los rankings base de los documentos *q* similares al documento

d.

$$RI(d) = c * \sum_{i=1}^{Cq} \frac{I(q)}{Cq} \quad (8)$$

Donde:

RI(d) es el ranking de impacto del documento *d.*

q es un documento que por su similitud se hace altamente probable de que se llegue al documento *d*.

I(q) es el ranking base del documento *q.*

2.8.5 Cálculo del ranking de autor.

Ecuación por la que se realiza el cálculo del ranking de autor.

$$RA(d) = \frac{M}{N} * \sum_{i=1}^{M} \frac{I(d(A))}{M} \quad (9)$$

Donde:

RA(d) es el ranking de autor del documento *d.*

d(A) es el documento publicado por el autor *A*.

I(d(A)) es el ranking base del documento publicado por el autor *A.*

M es el total de documentos publicados por el autor *A.*

2.8.6 Cálculo del ranking de novedad.

Ecuación para el cálculo del ranking de novedad (actualidad). La misma fue tomada del trabajo *"Study on The Method of Ranking Scientific Papers"* (31) anteriormente mencionado.

$$RN(d) = \frac{(t - \min\{T(k)\} + 1)}{(\max\{T(k)\} - \min\{T(k)\} + 1)} \quad (10)$$

Donde:

RN(d) es el ranking de novedad del documento *d.*

t es la cantidad de documentos que fueron publicados en el año en que se publicó *d.*

T(k) es la cantidad de documentos publicados por años.

2.9 Algoritmo.

El pseudocódigo relacionado con el algoritmo propuesto se describe a continuación.

2.9.1 Ranking base.

INICIO

Entrada: D= $\{d_1; d_2;.....; d_n\}$ – colección de documentos

Salida: RBD – ranking base de los documentos

Rb – ranking base del documento *d*

V – visitas del documento

T – total de visitas existentes

Para Cada d ∈ D Hacer

    Rb = V / T

    RBD(D) = Rb

FIN Para Cada

escribir RBD

FIN

2.9.2 Ranking impacto.

INICIO

Entrada: D= $\{d_1; d_2;.....; d_n\}$ – colección de documentos

Salida: RID – ranking impacto de los documentos

c – factor de probabilidad = 0.45

Ri – ranking de impacto del documento *d*

docs_similares – documentos *q* similares a *d*

t – total de docs_similares

ranking_base – ranking base del documento *q*

suma = 0

Para Cada d ∈ D Hacer

    Para Cada docs_similares Hacer

        suma = suma+ ranking_base / t

    Fin Para Cada

    Ri = c∗suma

    RID(D) = Ri

Fin Para Cada

escribir RID

FIN

2.9.3 Ranking autor.

INICIO

Entrada: D= $\{d_1; d_2;.....; d_n\}$ – colección de documentos

Salida: RAD – ranking de autor de los documentos

autor – autor del documento *d*

DA – documentos publicados por el autor del documento *d*

Ra – ranking de autor del documento *d*

ranking_base – ranking base de los documentos publicados por el autor

M – total de documentos publicados por el autor

N – total de documentos existentes

suma = 0

Para Cada d ∈ D Hacer

    Para Cada autores Hacer

        Para Cada DA Hacer

            suma = suma + ranking_base / M

        Fin Para Cada

        Ra = Ra+ M/N ∗ suma

    Fin Para Cada

    RAD(D) = Ra

Fin Para Cada

escribir RAD

FIN

2.9.4 Ranking novedad.

INICIO

Entrada: D= $\{d_1; d_2;.....; d_n\}$ – colección de documentos

Salida: RND – ranking de novedad de los documentos

Rn – ranking de novedad del documento *d*

t – cantidad de publicaciones realizadas en el año en que se publicó *d*

min – menor cantidad de publicaciones realizadas

max – mayor cantidad de publicaciones realizadas

Para Cada d ∈ D Hacer

    n = t – min + 1

    d = max – min + 1

    Rn = n / d

    RND(D) = Rn

Fin Para Cada

escribir RND

FIN

2.9.5 Ranking final.

INICIO

Entrada: D= $\{d_1; d_2;.....; d_n\}$ – colección de documentos

Salida: RFD – ranking final de los documentos

Rf – ranking final del documento *d*

Ri – ranking impacto del documento *d*

Ra – ranking autor del documento *d*

Rn – ranking novedad del documento *d*

Para Cada d ∈ D Hacer

    Rf = Ri∗Ra∗Rn

    RFD(D) = Rf

Fin Para Cada

escribir RFD

FIN

2.10 Conclusiones del capítulo.

Luego de haber realizado este capítulo el cual su principal objetivo es explicar detalladamente la solución propuesta y los elementos que la conforman, se llega a la conclusión de que con la realización de la descripción de la propuesta, la descripción de cada elemento que conforma el algoritmo y el pseudocódigo del mismo se logró un mejor entendimiento en cuanto al desarrollo de la solución. Se puede afirmar también que la realización de los diferentes diagramas, la identificación de los requerimientos y el uso del patrón arquitectónico MVC permitió una mejor

organización del código y una mejor estructura para la solución ayudando a realizar el trabajo con una alta calidad.

CAPÍTULO 3. PRUEBAS Y RESULTADOS.

3.1 Introducción.

En este capítulo se realizan un conjunto de pruebas para evaluar el funcionamiento del método de ranking propuesto partiendo de que se tiene un estado inicial donde la recuperación de información y el ordenamiento es por similitud y luego se incorpora un método de ranking que tiene en cuenta el autor, la actualidad y el impacto de los documentos. Para medir la precisión y el rendimiento del sistema al que se le aplica el método, se realizan varios experimentos. También se hace una constatación para determinar los cambios que se produjeron en cuanto a los documentos recuperados entre uno y otro estado, permitiendo arrojar conclusiones con respecto al trabajo realizado. Para la realización de estas pruebas fue necesario tomar los datos referentes a los documentos previamente indexados por el OHS almacenados en una base de datos y mostrar en una vista los resultados obtenidos que no son más que los documentos recuperados por un sistema que realiza una recuperación de información por similitud, de cada resultado se muestra el título, la descripción y el año.

3.2 Metodología.

Primeramente se realiza una medición de la precisión, la cual se basa en la efectividad teniendo en cuenta las respuestas que el sistema emite para determinada consulta. Aquí hay que tener presente que al aplicar el método de ranking solo se deberían esperar cambios en el orden, no en los elementos del conjunto solución.

Para esto se parte de realizar un conjunto de 32 consultas y se toman los 10 primeros resultados para cada una de ellas (hay casos en que el sistema retorna menos de 10 resultados debido a la complejidad de la consulta o a que se está trabajando con una muestra de solo 500 documentos). Inicialmente se realizan sin incorporar el ranking y posteriormente incorporando este, permitiendo realizar una constatación de los resultados en términos de precisión.

Para definir el cálculo de la precisión se toma la ecuación de la literatura "*Introducción a la Recuperación de Información*" (32), esta ecuación es la siguiente:

$$P = \frac{DRR}{DR} \quad (11)$$

Donde:

P es la precisión.

DRR son los documentos relevantes recuperados.

DR son los documentos recuperados.

También se mide el rendimiento tomando como referencia los tiempos de respuestas del sistema en el momento en que no se le es aplicado el ranking y posteriormente después que se le aplica. Se realizan mediciones para las 10 primeras consultas y se llega a una expresión analítica mediante la cual se calculan los tiempos de respuestas aproximados dependiendo de la cantidad de documentos recuperados.

Finalmente se realiza un análisis para medir la diferencia que hay entre los documentos mostrados antes y después de aplicar el método de ranking, para esto se toman los 10 primeros elementos de cada consulta.

3.3 Descripción de la fuente de datos.

Los datos provienen del repositorio institucional de la Universidad de las Ciencias Informáticas, por lo que es una muestra de documentos académicos y científicos. Estos documentos están descritos siguiendo el estándar *Dublin Core*, entre los metadatos que contienen están: título, descripción, autores, fecha y palabras clave. La muestra contiene 500 documentos seleccionados de forma aleatoria y donde hay presencia de diversas áreas del conocimiento. En el Anexo 2 se muestran las consultas realizadas.

3.4 Experimentos.

A continuación se realizan diferentes experimentos para observar el comportamiento de varias variables antes y después de aplicar el método de ranking.

3.4.1 Medición de la precisión.

Para medir la precisión de la aplicación del método de ranking se realiza un experimento donde por cada una de las 32 consultas se le calcula el valor de precisión correspondiente según la ecuación de precisión que anteriormente se menciona. Para observar el comportamiento de la precisión se realiza una gráfica donde se visualizan los valores que toma cada una de las consultas realizadas.

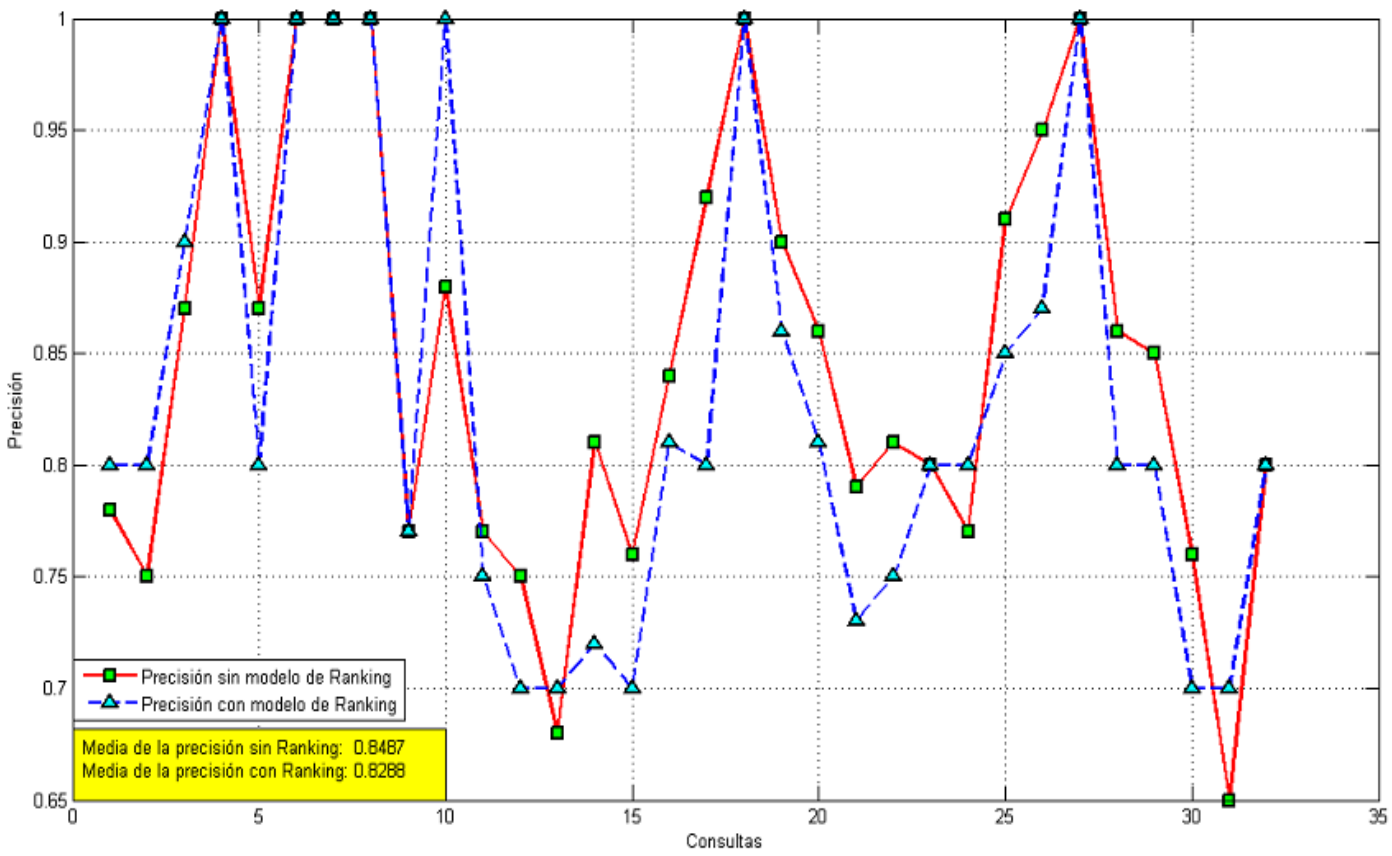

Figura 8 – Gráfica de precisión.

En la Figura 8 se muestra una gráfica de precisión de los resultados de las búsquedas realizadas mediantes las 32 consultas, antes de aplicar el modelo de ranking (línea roja) y después de aplicarlo (línea azul). La precisión media obtenida para uno y otro caso tiene muy poca variación, pero si hay una pequeña disminución al aplicar el modelo. Se hace necesario demostrar que esta diferencia no es significativa, para poder concluir que la aplicación del modelo de ranking no compromete la precisión del SRI.

La prueba de que la diferencia de los valores de la media para cada caso (sin aplicar el modelo de ranking y después de aplicado el modelo) no es estadísticamente significativa es una muestra de que cualquier resultado atípico sería un resultado casual y no refleja el comportamiento habitual del sistema, además permitiría extrapolar los resultados. Para realizar el análisis de la diferencia entre las medias se parte de contrastar la normalidad de los conjuntos de datos. Como solamente se ejecutaron 32 consultas se empleó el *test* de Shapiro-Wilk (33). Este test se ejecutó a partir de la función *swtest*, desarrollada por Ahmed Ben Saida (34), distribuida bajo licencia BSD. El resultado retornó que se está en presencia de una distribución normal.

Sabiendo que la muestra se distribuye normalmente se ejecuta el estadístico *t* de *Student* (35) para dos muestras independientes. El *test* se ejecutó mediante la función *ttest* que provee el asistente Matlab. El resultado fue cero, lo que demuestra que la diferencia entre las medias poblacionales está dentro de lo aceptable. Los dos *test* se realizaron con un intervalo de confianza de 5%. En el Anexo 3 se muestran las consultas y los valores de precisión correspondientes antes y después de aplicar el modelo de ranking, estos constituyen los vectores que se pasan como parámetros a los *test*.

3.4.2 Medición del rendimiento.

En la medición del rendimiento se toman los valores correspondientes a los tiempos de respuestas del sistema (en segundos) por cada una de las 10 primeras consultas, teniendo en cuenta la cantidad de elementos recuperados para cada una de ellas. En la siguiente gráfica se muestran estos valores.

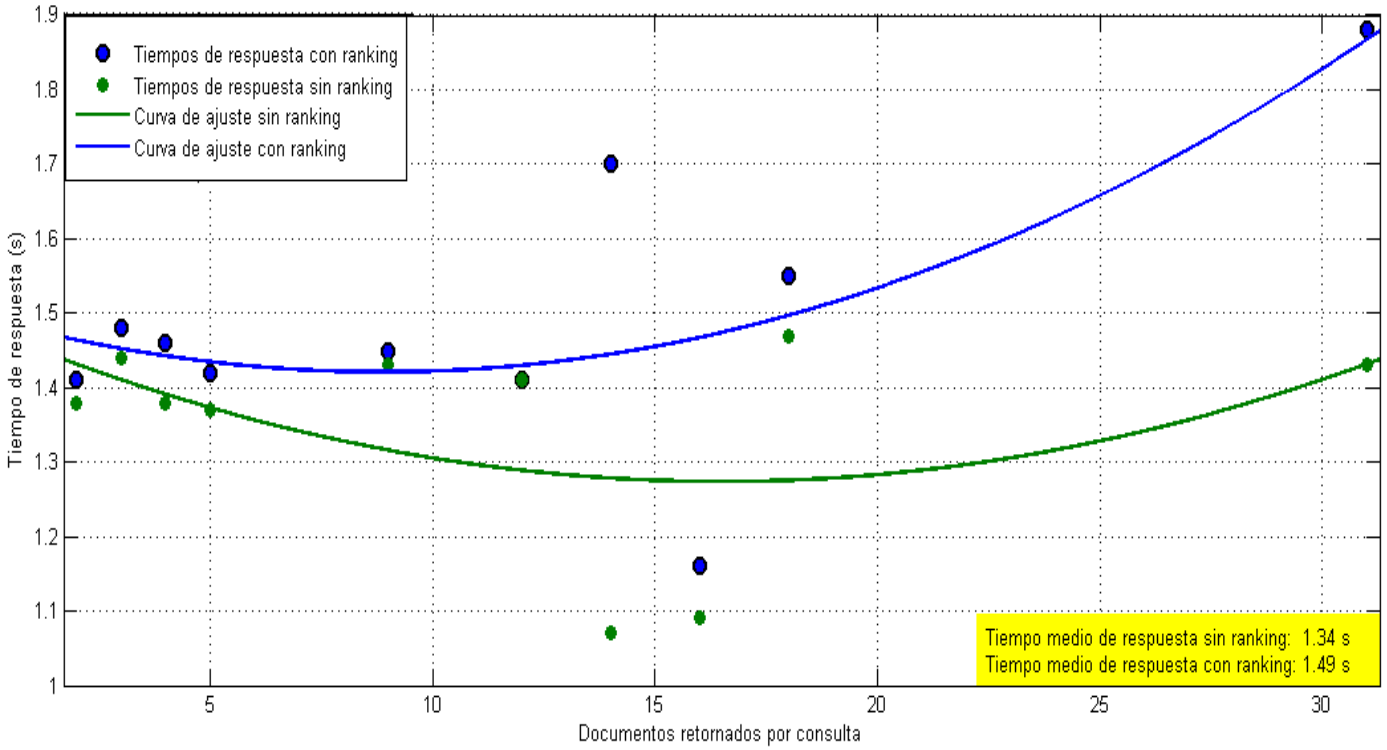

Figura 9 – Gráfica de tiempos de respuestas del sistema.

En la Figura 9 se muestra la gráfica correspondiente a los tiempos de respuestas del sistema antes (curva verde) y después del ranking (curva azul), en ambos casos muy similares. Como promedio al incorporar el método de ranking la demora es de 0.15 segundos, por lo que de igual forma que en la precisión, el rendimiento no se ve afectado al aplicar el método. En la gráfica

se muestran las curvas de ajustes para ambos casos.

Mediante el experimento anterior se llega a una expresión analítica para calcular los valores aproximados de los tiempos de respuestas del sistema dependiendo de la cantidad de documentos recuperados. La expresión es la siguiente:

$y = 0.000913x^2 - 0.01621x + 1.493\ (12)$

Donde:

y es el tiempo de respuesta del sistema.

x la cantidad de documentos recuperados.

3.4.3 Medición de la diferencia entre los 10 primeros resultados.

En este experimento se mide la diferencia en cuanto a los documentos recuperados antes y después de aplicar el método, tomando los 10 primeros elementos para el caso de las consultas que recuperen más de 10 documentos (en caso que sean menos de 10 no habrá diferencias porque el ranking se realiza siempre sobre los documentos similares).

Esta medida arrojó que como promedio el cambio es de 5.8. Entre 5 y 6 elementos cambian de posición al aplicar el ranking. Los valores de ranking en promedio, en el caso de aplicar el modelo duplican a los que se obtendrían sin aplicarlo, en los 10 primeros elementos. No se muestran todos los resultados experimentales ya que estos son muy similares entre sí. En el Anexo 1 pueden verse los resultados para la consulta *"Sistemas informáticos de apoyo a la salud"*.

3.5 Resultados.

Luego de realizar los experimentos anteriores se tiene como resultado que las variables precisión y rendimiento no se ven afectadas en gran medida con la incorporación del método de ranking que se desarrolla, esto sucede porque existe muy poca variación de los valores entre un estado y otro (antes y después de aplicar el método). En el caso de la precisión influyen mucho las visitas que tengan los documentos ya que el método de ranking parte desde esa base para realizar el cálculo. Sin embargo, si se demuestra que el método aplicado realiza cambios en el orden de los elementos, así como en la aparición de los elementos entre uno y otro estado, según el promedio calculado de cada 10 elementos, 5 o 6 cambian de posición.

3.6 Conclusiones del capítulo.

Con la realización de este capítulo se logró tener un conocimiento sobre el comportamiento del

método de ranking implementado durante la recuperación de información. Las mediciones y experimentos realizados demostraron que la incorporación del método no realiza cambios importantes en cuanto a la precisión y el rendimiento del sistema, se demostró además que se realizan cambios en el orden entre los elementos mostrados antes y después de aplicar el método de ranking.

CONCLUSIONES.

Después de haber realizado esta investigación se concluye que:

1. El estudio realizado sobre la Recuperación de Información, la trascendencia del ranking dentro de esta área, así como de la arquitectura OAI-PMH, permitió crear las bases teóricas necesarias para proponer un método capaz de calcular un valor de importancia a cada documento tomando en cuenta otros importantes parámetros como: el autor, la actualidad y el impacto.
2. En el entorno OAI-PMH es muy complicado realizar un método de ranking basado en las citas ya que el protocolo OAI-PMH no distribuye esa información por falta de metadatos destinados para tal propósito. Una forma de resolver este problema es tomando las visitas que recibe cada documento.
3. La aplicación del método de ranking no implica cambios sustanciales en cuanto a tiempo de respuesta del sistema.
4. Los valores de precisión antes y después de aplicar el método de ranking varían muy poco, por lo que se puede afirmar que la precisión no se ve afectada a gran escala con la incorporación del método.

RECOMENDACIONES.

- Valorar la posibilidad de que se pueda mediante el protocolo distribuir las citas, esto puede mejorar la efectividad del ranking.
- Incluir para el cálculo del ranking una variable que describa el tipo de material que es el documento (tesis, revista científica, etc.).

BIBLIOGRAFÍA CONSULTADA.

1. BRONCANO, R. G. *MODELOS DE RECUPERACION .Recuperación y organización de la información.* Universidad Carlos III de Madrid, 2006, Disponible en: http://modelosrecuperacion.tripod.com

2. FERNÁNDEZ, M. M. F.; BARTOMEO, Y. A.*, et al. EL PROTOCOLO OAI-PMH, COMPONENTE TECNOLÓGICO PARA EL ACCESO ABIERTO.* La Habana: 2012, Disponible en: http://repositorio_institucional.uci.cu/jspui/handle/ident/4113 ISBN 978-959-286-019-3.

3. GERLING, V. B. *Un Sistema Inteligente para Asistir la Búsqueda Personalizada de Objetos de Aprendizaje.* UNIVERSIDAD NACIONAL DE ROSARIO, 2009.

4. GUANGQIAN, Z. y XIN, L. *Study on The Method of Ranking Scientific Papers.*2010, ISBN 978-0-7695-3997-3.

5. KUMAR, P. R. y SINGH, A. K. *Web Structure Mining: Exploring Hyperlinks and Algorithms for Information Retrieval.*Curtin University of Technology, 2010, ISBN 1546-9239.

6. MA, N.; GUAN, J.*, et al. Bringing PageRank to the citation analysis.* 2007, Disponible en: www.sciencedirect.com

7. *METADATOS PARA DESCRIBIR E IDENTIFICAR UN DOCUMENTO EN LA RED.* Madrid: Universidad Carlos III de Madrid, 2010, Disponible en: http://www.metadatos-xmlrdf.com/metadatos/dublin-core.

8. PEDROCHE, F. *Métodos de cálculo del vector PageRank.* Institut de Matemática Multidisciplinar Universitat Politécnica de Valéncia., 2009,

9. TOLOSA, G. H.; BORDIGNON, F. R. A.*, et al.Búsqueda de Sitios Web con Autoridad en un Tema.* Universidad Nacional de Luján, 2006, Disponible en:

http://sedici.unlp.edu.ar/bitstream/handle/10915/20862/Documento_completo.pdf?sequence=1

10. VOLDER, C. D. *El proyecto SIU-Bibliotecas del Consorcio SIU puso en marcha la BDU2* 2010, Disponible en: http://a-abierto.blogspot.com/2010/04/el-proyecto-siu-bibliotecas-del.html

REFERENCIAS BIBLIOGRÁFICAS.

1. FERNÁNDEZ, M. M. F.; BARTOMEO, Y. A.*, et al. EL PROTOCOLO OAI-PMH, COMPONENTE TECNOLÓGICO PARA EL ACCESO ABIERTO.* La Habana: 2012, Disponible en: http://repositorio_institucional.uci.cu/jspui/handle/ident/4113 ISBN 978-959-286-019-3.

2. BAEZA-YATES, R. y RIBEIRO-NETO, B. *Modern Information Retrieval.* Addison Wesley ed. 1999, ISBN 0-201-39829-X.

3. BORDIGNON, F. R. A. y TOLOSA, G. H. RECUPERACIÓN DE INFORMACIÓN:UN ÁREA DE INVESTIGACIÓN EN CRECIMIENTO. *Télématique*, 2007, vol. 6, nº 1, Disponible en: http://www.urbe.edu/publicaciones/telematica/indice/pdf-vol6-1/4-recuperacion-de-informacion.pdf. ISSN 1856-4194.

4. SEQUERA, J. L. C. *NUEVA PROPUESTA EVOLUTIVA PARA EL AGRUPAMIENTO DE DOCUMENTOS EN SISTEMAS DE RECUPERACIÓN DE INFORMACIÓN.* Tesis Doctoral, Universidad de Alcalá, 2010.

5. GODOY, D. *Análisis y Recuperación de Información.* 2012, Disponible en: www.exa.unicen.edu.ar/catedras/ayrdatos/slides/intro-1p.pdf.

6. BRONCANO, R. G. *MODELOS DE RECUPERACION .Recuperacion y organizacion de la informacion.* Universidad Carlos III de Madrid, 2006, Disponible en: http://modelosrecuperacion.tripod.com/.

7. COMECHE, J. A. M. *Los modelos clásicos de Recuperación de información y su vigencia.* Universidad Complutense de Madrid, 2006, Disponible en: http://eprints.ucm.es/5979/1/Modelos_RI_preprint.pdf.

8. PEDROCHE, F. *Métodos de cálculo del vector PageRank.* Institut de Matemática Multidisciplinar Universitat Politécnica de Valéncia., 2009,

9. TOLOSA, G. H.; BORDIGNON, F. R. A.*, et al. Búsqueda de Sitios Web con Autoridad en un Tema.* Universidad Nacional de Luján, 2006, Disponible en: http://sedici.unlp.edu.ar/bitstream/handle/10915/20862/Documento_completo.pdf?sequence=1.

10. VALVERDE, G. *Algoritmo HITS: Hubs y Authorities. Alternativa y/o dependecia del PageRank.* 2010, Disponible en: http://blog.lineasdemarketing.com/algoritmo-hits-hubs-authorities-alternativa-dependecia-pagerank/2010/09/25/.

11. *Indicadores para la evaluación de la investigación.* València: Universitat de València, 2011, Disponible en: http://www.uv.es/bibsoc/GM/dosieres/citas.html.

12. *Factor de impacto de una revista.* Universitat Oberta de Catalunya, 2008, Disponible en: http://www.bib.utfsm.cl/nuevositio/attachments/Factorimpacto.pdf.

13. CONTRERAS, G. E. *OAI-PMH, protocolo para la transmisión de metadatos.* 2004,

14. GERLING, V. B. *Un Sistema Inteligente para Asistir la Búsqueda Personalizada de Objetos de Aprendizaje.* UNIVERSIDAD NACIONAL DE ROSARIO, 2009.

15. MACEIRAS, M. *¿Qué es el Lenguaje HTML?* 2007, Disponible en: http://www.tecnocosas.es/que-es-el-lenguaje-html/.

16. MARTÍN, P. R. *HTML.* 2009, Disponible en: http://www.asptutor.com/zip/cbhtml.pdf.

17. MARIN, M. D. A. *Definición de lenguaje de programación. Tipos. Ejemplos.* 2008, Disponible en: http://catedraprogramacion.foroactivos.net/t83-definicion-de-lenguaje-de-programacion-tipos-ejemplos.

18. *Lenguaje PHP (Hypertext Preprocessor).* Bogotá D.C: Universidad Nacional de Colombia, 2010, Disponible en: http://www.virtual.unal.edu.co/cursos/sedes/manizales/4060029/lecciones/cap11-2.html.

19. FAURE, D. G. y RONDÓN, A. V. *Aplicación web para la gestión de eventos científicos en la Universidad de las Ciencias Informáticas.* Universidad de las Ciencias Informáticas, 2010.

20. GONZÁLEZ, J. M.; BARAHONA, J.*, et al. Introducción al software libre* 2007, Disponible en: http://curso-sobre.berlios.de/introsobre/2.0.1/sobre.pdf.

21. SANTA, C. *DEFINICION DE DE NETBEANS* 2011, Disponible en: http://mogoyita.blogspot.com/2011/02/definicion-de-de-netbeans.html.

22. MORALES, P. A. A.; TORRES, M. I. H.*, et al. Servidores Web.* 2009, Disponible en: http://www.monografias.com/trabajos75/servidores-web/servidores-web.shtml.

23. *Apache 2.2: servidor web.* Madrid: Instituto Nacional de Tecnologías Educativas y Formación del Profesorado 2008, Disponible en: http://recursostic.educacion.es/observatorio/web/es/software/servidores/580-elvira-mifsud.

24. REBOLLEDO, J. y SANTACOLOMA, A. Disponible en: http://www.angelfire.com/ultra2/pecanpie/Bimestral/Glosario.htm.

25. PECOS, D. *PostGreSQL vs. MySQL.* 2007, Disponible en: http://danielpecos.com/docs/mysql_postgres/x57.html.

26. *Open Harvester Systems.* 2010, Disponible en: http://pkp.sfu.ca/?q=harvester.

27. DUQUE, M. L. y RAVELO, R. O. *Herramienta en Matlab para la obtención de información de la base de datos y ficheros electroencefalograma del proyecto Mapeo Cerebral Humano Cubano.* Universidad de las Ciencias Informáticas, 2010.

28. KUMAR, P. R. y SINGH, A. K. *Web Structure Mining: Exploring Hyperlinks and Algorithms for Information Retrieval.* Curtin University of Technology, 2010, ISBN 1546-9239.

29. PÉREZ, M. T. *Sistema para el Control de Asistencia integrado al CMS Drupal.* Universidad de las Ciencias Informáticas, 2010.

30. AMÓN, I. y JIMÉNEZ, C. *Funciones de Similitud sobre Cadenas de Texto: Una Comparación Basada en la Naturaleza de los Datos.* 2010,

31. GUANGQIAN, Z. y XIN, L. *Study on The Method of Ranking Scientific Papers.* 2010,

32. TOLOSA, G. H. y BORDIGNON, F. R. A. *Introducción a la Recuperación de Información.* Buenos Aires: Universidad Nacional de Luján, 2007,

33. JIMÉNEZ, A. *Contraste de Shapiro-Wilk.* 2006, Disponible en: http://www.xatakaciencia.com/matematicas/contraste-de-shapiro-wilk.

34. SAIDA, A. B. *Shapiro-Wilk and Shapiro-Francia normality tests. - File Exchange - MATLAB Central.* 2007, Disponible en: http://www.mathworks.com/matlabcentral/fileexchange/13964-shapiro-wilk-and-shapiro-francia-normality-tests.

35. GARCIA. *T-student estadistica.* 2012, Disponible en: http://www.buenastareas.com/ensayos/t-Student-Estadistica/3490345.html.

GLOSARIO DE TÉRMINOS.

GPL (*General Public License*): Licencia Pública General que permite el uso y modificación del código para desarrollar *software* libre, pero no propietario.

Hardware: se refiere a todas las partes tangibles de un sistema informático, sus componentes son: eléctricos, electrónicos, electromecánicos y mecánicos.

HTTP (*HyperText Transfer Protocol*): es un protocolo de transferencia de hipertexto, es el usado en cada transacción de la web.

IMAP (*Internet Message Access Protocol):* es un protocolo de aplicación de acceso a mensajes electrónicos almacenados en un servidor. Mediante IMAP se puede tener acceso al correo electrónico desde cualquier equipo que tenga una conexión a Internet.

Internet: conjunto descentralizado de redes de comunicación interconectadas.

NNTP (*Network News Transport Protocol*): es un protocolo inicialmente creado para la lectura y publicación de artículos de noticias en Usenet (*Users Network*, red de usuarios consistente en un sistema global de discusión en Internet).

PHP: acrónimo recursivo que significa *Hypertext Pre-processor*, es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.

POO: paradigma de programación que utiliza objetos, estructuras de datos compuestos por datos y métodos, además de sus interacciones para diseñar aplicaciones.

POP3 (*Post Office Protocol*): en español llamado Protocolo de Oficina Postal en clientes locales de correo para obtener los mensajes de correo electrónico almacenados en un servidor remoto.

Protocolo: conjunto de reglas y normas que permiten que dos o más entidades de un sistema de comunicación se comuniquen entre ellos para transmitir información.

SNMP (*Simple Network Management Protocol*): es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red. Permite a los administradores supervisar el funcionamiento de la red, buscar y resolver sus problemas.

***Software*:** equipamiento lógico o soporte lógico de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas.

SQL (S*tructured Query Language*): es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.

Unix: es un sistema operativo portable, multitarea y multiusuario.

Web: sistema para presentar información en internet basado en hipertexto.

WML (*Wireless Markup Language):* es un lenguaje cuyo origen es el XML. Este lenguaje se utiliza para construir las páginas que aparecen en las pantallas de los teléfonos móviles y los asistentes personales digitales.

XML (*Extensible Markup Language*): siglas en español, Lenguaje de Marcas Extensible. Es un formato estándar para el intercambio de datos.

ANEXOS.

Anexo 1. Valores de ranking antes y después de aplicar el modelo.

En esta tabla solo se muestran los 10 primeros resultados para la consulta "*Sistemas informáticos de apoyo a la salud".*

Antes de aplicar el modelo de ranking.

ID	R-Base	R-Impacto	R-Autor	R-Novedad	R-Final
6	0,002774	0,000802	0,000006	1	4,8E-09
28	0,003329	0,000695	0,000007	1	4,9E-09
31	0,002774	0	0,000006	0,854015	0
41	0,002219	0,001391	0,000004	0,934307	5,2E-09
78	0,000159	0,000906	0	1	0
89	0,00103	0,000678	0,000004	0,729927	2E-09
107	0,001189	0,000633	0,000005	0,729927	2,3E-09
139	0,000317	0,000908	0,000002	0,729927	1,3E-09
150	0,003012	0,0006	0,000012	1	7,2E-09
Promedio	0,001867	0,00073478	5,1111E-06	0,88645589	3,0778E-09

Tabla 1 – Valores de rankings de los primeros 10 documentos antes de aplicar el modelo.

Después de aplicar el modelo de ranking.

ID	R-Base	R-Impacto	R-Autor	R-Novedad	R-Final
203	0,003884	0,000642	0,000016	1	1,03E-08
364	0,002933	0,000963	0,000012	1	1,16E-08
359	0,00317	0,000611	0,000019	0,729927	8,5E-09
389	0,002378	0,000808	0,00001	1	8,1E-09
150	0,003012	0,0006	0,000012	1	7,2E-09
483	0,000872	0,001712	0,000004	1	6,8E-09
237	0,001902	0,000796	0,000008	0,934307	5,9E-09
298	0,003804	0,000865	0,000008	0,854015	5,9E-09
41	0,002219	0,001391	0,000004	0,934307	5,2E-09
Promedio	0,002686	0,000932	1,0333E-05	0,93917289	7,7222E-09

Tabla 2 – Valores de rankings de los primeros 10 documentos después de aplicar el modelo.

Las filas marcadas en verdes fueron los elementos que no cambiaron (fueron recuperados entre los 10 primeros aplicando y sin aplicar el modelo de ranking).

Anexo 2. Consultas realizadas.

1. Uso de técnicas de Inteligencia Artificial en el desarrollo de sistemas informáticos.

2. Aplicación de la informática en la educación.
3. Sistemas informáticos de apoyo a la salud.
4. Uso de metodologías ágiles en el desarrollo de *software*.
5. Componentes de redes informáticas.
6. Desarrollo de redes telemáticas.
7. Aplicaciones para telefonía móvil.
8. Aplicación de la informática en las FAR.
9. Patrón de arquitectura Modelo Vista Controlador.
10. Arquitectura en capas como patrón en el desarrollo de *software*.
11. El protocolo OAI-PMH como herramienta de interoperabilidad.
12. Aplicaciones desarrolladas usando el lenguaje de programación Java.
13. Aplicaciones desarrolladas usando el lenguaje de programación PHP.
14. Técnicas para el agrupamiento de documentos.
15. Técnicas para la clasificación supervisada de documentos.
16. Modelos fundamentales utilizados para la recuperación de información.
17. Algoritmos de encriptación simétricos.
18. Algoritmos de recorridos de grafos.
19. Aplicaciones web para la gestión empresarial.
20. Sistemas de archivos del sistema operativo Linux.
21. Uso de la metodología RUP en el desarrollo de *software*.
22. Flujos de trabajo de la metodología de desarrollo de *software* RUP.
23. Sistemas informáticos desarrollados para la gestión documental.
24. Aplicaciones basadas en servicios web.
25. Aplicación de la informática en el turismo.
26. Protocolos de enrutamientos.
27. Capas que conforman el modelo OSI.
28. Dispositivos de redes inalámbricas.
29. Algoritmos para el reconocimiento facial.
30. Algoritmos de reconocimiento de voz.
31. Sistemas informáticos para la toma de decisiones.
32. Herramientas de gestión de proyectos.

Anexo 3. Valores de precisión antes y después de aplicar el modelo de ranking.

Consultas	Precisión sin ranking	Precisión con ranking
1	0.78	0.8
2	0.75	0.8
3	0.87	0.9
4	1	1
5	0.87	0.8
6	1	1
7	1	1
8	1	1
9	0.77	0.77
10	0.88	1
11	0.77	0.75
12	0.75	0.7
13	0.68	0.7
14	0.81	0.72
15	0.76	0.7
16	0.84	0.81
17	0.92	0.8
18	1	1
19	0.9	0.86
20	0.86	0.81
21	0.79	0.73
22	0.81	0.75
23	0.8	0.8
24	0.77	0.8
25	0.91	0.85
26	0.95	0.87
27	1	1
28	0.86	0.8
29	0.85	0.8
30	0.76	0.7
31	0.65	0.7
32	0.8	0.8

Tabla 3 – Valores de precisión antes y después de aplicar el modelo de ranking.