Revista Cubana de Ciencias Informáticas
ISSN: 1994-1536
ISSN: 2227-1899
Editorial Ediciones Futuro

García O´reilly, Alexander; Utrera Sust, Eric
Componente para el cálculo de la relevancia en el buscador Orión
Revista Cubana de Ciencias Informáticas, vol. 13, núm. 2, 2019, Abril-Junio, pp. 91-104
Editorial Ediciones Futuro

Revista Cubana de Ciencias Informáticas
Vol. 13, No. 2, Mes Abril-Junio, 2019
ISSN: 2227-1899 | RNPS: 2301
Pág. 91-104
http://rcci.uci.cu

Tipo de artículo: Artículo original
Temática: Inteligéncia Artificial
Recibido: 11/01/2018 | Aceptado: 09/05/2019

"COMPONENTE PARA EL CÁLCULO DE LA RELEVANCIA EN EL BUSCADOR ORIÓN"

COMPONENT FOR THE CALCULATION OF THE RELEVANCE IN THE ORION SEARCHER

Ing. Alexander García O´reilly [1*], Ing. Eric Utrera Sust [2]

[1] Centro de Ideoinformática (CIDI). Facultad 1. Universidad de las Ciencias Informáticas, Carretera a San Antonio delos Baños, km 2 1⁄2, Torrens, Boyeros, La Habana, Cuba. CP.: 19370.
[2] Centro de Ideoinformática (CIDI). Facultad 1. Universidad de las Ciencias Informáticas, Carretera a San Antonio delos Baños, km 2 1⁄2, Torrens, Boyeros, La Habana, Cuba. CP.: 19370.
[*] Autor para correspondencia: aoreilly@uci.cu

Resumen
La presente investigación describe el proceso de desarrollo de un componente para el buscador Orión que incrementará la precisión con que son devueltos los resultados de las consultas. Esta herramienta permitirá mejorar el cálculo de la relevancia de información de dichos resultados, a través del análisis del Perfil de Búsqueda del Usuario y las categorías de los documentos. Se estudiaron motores de búsqueda a nivel nacional e internacional para determinar sus características, así como sus estrategias para el cálculo de la relevancia de información. Para la implementación se utilizó el entorno de desarrollo integrado NetBeans 7.4, como Sistema Gestor de Bases de Datos (SGBD) PostgreSQL 9.1 y Visual Paradigm 8.0 como herramienta de modelado. Para guiar todo el proceso de desarrollo se usó la metodología AUP en su personalización para la UCI. Se emplearon el servidor web Apache 2 y los lenguajes Java, UML, SQL 1.4 y XML. Como herramienta gráfica para la administración del SGBD se utilizó el PgAdminIII 1.14. Las pruebas realizadas permitieron construir un producto con calidad y controlaron el proceso de validación del funcionamiento de la aplicación. Como resultado del trabajo realizado se obtiene un producto con una documentación que sirve de base para futuras investigaciones o modificaciones a la solución.

Palabras clave: cálculo de la relevancia, motores de búsqueda, perfil de búsqueda de usuario.

Revista Cubana de Ciencias Informáticas
Vol. 13, No. 2, Mes Abril-Junio, 2019
ISSN: 2227-1899 | RNPS: 2301
Pág. 91-104
http://rcci.uci.cu

Abstract
This research describes the process of developing a component for the Orion search engine that will increase the precision with which the results of the queries are returned. This tool will make it possible to improve the calculation of the information relevance of said results, through the analysis of the User's Search Profile and the categories of the documents. Search engines were studied at a national and international level to determine their characteristics, as well as their strategies for calculating the relevance of information. For the implementation, the integrated development environment NetBeans 7.4 was used, as the Database Management System (SGBD) PostgreSQL 9.1 and Visual Paradigm 8.0 as a modeling tool. To guide the entire development process, the AUP methodology was used in its customization for the UCI. The Apache 2 web server and the Java, UML, SQL 1.4 and XML languages were used. As a graphic tool for the administration of the DBMS, PgAdminIII 1.14 was used. The tests carried out allowed us to build a product with quality and controlled the validation process of the operation of the application. As a result of the work done, a product is obtained with documentation that serves as the basis for future research or modifications to the solution.

Keywords: *relevance calculation, search engines, user search profile.*

Introducción

El uso de las tecnologías de la informática en casi todas las áreas del conocimiento, provoca que cada día se genere un cúmulo importante de información (M. Coll, 2011). Estos nuevos datos se procesan y luego se almacenan en grandes bases de datos situadas en servidores distribuidos por todo el mundo. Una manera de acceder a dicha información es a través de los Sistemas de Recuperación de Información (SRI). Estos sistemas se corresponden con el área de la ciencia, Recuperación de Información (RI) (Baeza-Yates, 2017). Los SRI se encargan de la adquisición, representación, almacenamiento y organización de la información (Méndez, 2004). El surgimiento de la Internet y el alto grado de consolidación de la web a nivel mundial han propiciado que se publique información en la red de manera acrecentada y acelerada (Álvarez, 2016). Por tal motivo, la evolución y desarrollo de los SRI se ha girado en torno a la web, donde ha encontrado una alta aplicación práctica y un aumento del número de usuarios, especialmente en el campo de los directorios y los buscadores web.

En el funcionamiento general de un buscador una perspectiva importante es: la recuperación de la información. Un buscador web utiliza uno o varios modelos de recuperación de información, en el cual queda definido: cómo se obtienen las representaciones de los documentos y la consulta, la estrategia para validar la relevancia de los documentos respecto a una consulta, y los métodos para establecer el orden de salida (Román, 1997). Según María A. Grado-Caffaro (2000),

el problema fundamental en los buscadores web es determinar si en realidad las páginas que se han recuperado son las más relevantes, y si el ranking obedece a la relevancia real de la información proporcionada.

En la Universidad de las Ciencias Informáticas (UCI) existe un proyecto que tiene a su cargo el desarrollo de un producto llamado el Motor de Búsqueda Orión. El objetivo de este proyecto es disponer de una herramienta propia que realice las búsquedas en la red. En un inicio la universidad reclamaba optimizar la búsqueda y recuperar la información existente en la red interna, en el presente se espera que Orión sea desplegado cubriendo el ámbito cubano. Con tal motivo, el esfuerzo se centra en brindarles a los usuarios nacionales un buscador que satisfaga las necesidades de información de la sociedad. El buscador Orión cuenta con tres mecanismos fundamentales que, al funcionar en conjunto permiten contar con un sistema para acceder a los documentos y sitios publicados en la red. Estos son, una interfaz de tipo web para realizar las consultas de búsqueda, un servidor de indexación de contenidos basado en Solr y como robot de búsqueda o *spider* utiliza Nutch (Delgado, 2010). El éxito de Orión depende de que ofrezca a los usuarios resultados útiles y actualizados, de forma tal que en realidad se satisfagan sus necesidades.

El buscador Orión muchas veces no proporciona información verdaderamente relevante sobre un tema, a pesar de devolver gran cantidad de documentos en un tiempo mínimo y de disponer de una base de datos con millones de documentos indexados. En ese sentido, la relevancia de los documentos constituye un factor clave a la hora de valorar la efectividad de un buscador, ya que se trata del orden en que se presentan los resultados. Como es lógico los usuarios esperan encontrar los documentos más relevantes en los primeros sitios, y en función de ello el buscador es valorado de mejor o peor por los usuarios.

Materiales y métodos

Para el desarrollo de la aplicación se precisaron un conjunto de elementos teóricos, tales como cálculo de relevancia y puntuación de relevancia. Igualmente, se estudiaron características de los motores de búsqueda en función de comprender la base de sus procesos. Dentro de las bibliografías consultadas se encuentra el artículo científico "Posicionamiento en buscadores: una metodología práctica de optimización de sitios web" por Iñigo Arbildi Larreina, donde se puede constatar que la relevancia es una aproximación bibliométrica a la hora de posicionar los resultados. También, se profundizó en el artículo científico *"Algorithm for calculating relevance of documents in information retrieval systems"* por Roberto Passailaigue Baquerizo y otros, en el cual se aprecia un estudio de los modelos clásicos de cálculo de similitud.

Además, se detallaron los lenguajes empleados en el proceso de desarrollo de la propuesta solución, así como las herramientas necesarias para la implementación del componente cálculo de la relevancia de información en el buscador

Orión. Con el objetivo de constatar teóricamente la evolución de los Sistemas de Recuperación de Información (SRI) se aplicaron los métodos científicos histórico-lógico y analítico-sintético.

Igualmente, en el estudio se tomó como referencia los sistemas homólogos y las valoraciones plasmadas en la Guía de Referencia Apache Solr, donde se puede constatar que la relevancia es el grado en que una respuesta de consulta satisface a un usuario que está buscando información. De igual forma, es necesario conocer que la puntuación de relevancia se representa mediante un número de como flotante positivo denominado *score.* En ese caso, a mayor *score* más relevante es el documento (*Elasticsearch of Apache Software Foundation,* 2017).

Trabajos Relacionados

Con el objetivo de encontrar una solución adaptable se analizaron los buscadores existentes en el mundo. En dicho análisis se identificaron en los buscadores internacionales Google, Yahoo! Search y Bing; y en los buscadores nacionales C.U.B.A. y Lupa los siguientes aspectos: motor de búsqueda, robot rastreador y algoritmos de relevancia empleados.

Como resultado del estudio se destacaron algunas características y funciones más relevantes de dichos buscadores web. Basado en ello se concluye y se comprueba que ninguno de los modelos matemáticos de cálculo de relevancia de información presentes en los buscadores web utilizan el perfil de usuario como se expone en el artículo científico *"Algorithm for calculating relevance of documents in information retrieval systems"* (Baquerizo y otros, 2017).

En consecuencia, el motor de búsqueda Orión no cuenta con un algoritmo que elabore las clasificaciones de los resultados según una base de conocimiento del perfil de usuario. Esto pudiera provocar una primera imprecisión en la presentación de los resultados y es que, la información presentada al usuario pudiera no estar relacionada con aquellas materias o aquel ámbito que en realidad le interesa. Esto es válido, sobre todo, cuando se hacen consultas usando términos que se aplican en varios campos o áreas del conocimiento. Igualmente, no devuelve al usuario resultados útiles con un criterio de relevancia personalizada de acuerdo a las categorías de documentos que en realidad interesan al que busca. Esto trae consigo que el usuario pierda el interés y la confianza en la herramienta, si no es capaz de satisfacer sus necesidades de forma óptima. En ese caso, se afecta un factor importante como es la fidelización de los usuarios. Sobre este particular Baeza Yates en una entrevista expresó: "es importante entender lo que el usuario desea hacer con la búsqueda y personalizarla a su tarea" (Marcos, 2008).

Para el autor Baeza Yates la solución a dicho problema se identifica con dos etapas fundamentales: elección de un modelo que permita calcular la relevancia de un documento frente a una consulta y el diseño de algoritmos que implementen este modelo de forma eficiente. En este sentido es necesario integrar esta variable al cálculo de la relevancia de información, debido a la importancia de conocer la intención del usuario después de la búsqueda. Respecto

a ese tema, se evidencia carencia de información en buscadores como Yahoo Search y Bing, debido a la indiscutible competencia con la mundialmente conocida Google.

Se decide utilizar como base del algoritmo en el cálculo de similitud, el Modelo Espacio Vectorial y el Modelo Booleano, teniendo en cuenta que Solr emplea la librería Lucene y a su vez esta tiene implementadas dichas funcionalidades. De igual forma, el modelo vectorial es muy versátil y eficiente a la hora de generar rankings de precisión en colecciones de gran tamaño, lo que le hace idóneo para determinar la puntuación parcial de los documentos (Carrasana, 2014). A los efectos del planteamiento anterior, se añade que según Martínez Méndez (2004) el modelo espacio vectorial es el más utilizado en la web.

Tabla 1. Estudio de los buscadores web en el mundo

Internacionales			
Buscadores Web	**Motor de búsqueda**	**Robot rastreador**	**Algoritmos de relevancia**
Google	Google	Googlebot	PageRank, Modelo Espacio Vectorial
Yahoo	Search Assits	Yahoo Slurp	-
Bing	Bing	Bingbot	-
Nacionales			
C.U.B.A.	Orión	Nutch	Modelo Booleano, Modelo Espacio Vectorial
Lupa	Solr	Lucid Crawler	Modelo Espacio Vectorial

Resultados y discusión

Para dar solución a los objetivos expuestos se propone el desarrollo de un componente de cálculo de relevancia incorporando el perfil de usuario y la categoría de los documentos. Se utilizará el lenguaje Java para generar un compilado (.jar) que se le integrará al servidor de indexación Solr. El componente dependerá de dos variables principales:

1. **Perfil de usuario:** Su función principal es almacenar el perfil de búsqueda del usuario (PBU). Esta variable combina las categorías consultadas por el usuario y su respectivo valor de porcentaje, el cual se interpreta como la cantidad de veces que el usuario realizó una consulta sobre una categoría definida.

   Según el artículo científico *"Algorithm for calculating relevance of documents in information retrieval systems"* estas preferencias de búsqueda del usuario se definen como resultado de la categorización de cada una de las consultas previamente introducidas, éstas se clasifican según el porcentaje de predominio (P) de las categorías más consultadas (Baquerizoy otros, 2017).

Revista Cubana de Ciencias Informáticas
Vol. 13, No. 2, Mes Abril-Junio, 2019
ISSN: 2227-1899 | RNPS: 2301
Pág. 91-104
http://rcci.uci.cu

	id [PK] integer	nombre text	naturaleza double prec	deporte double prec	cultura double prec	ciencias double prec	noticias double prec	tecnologia double prec
1	1	Alexander	0.25	0.35	0.55	0.75	0.68	0.46
2	2	Pedro	0.25	0.55	0.45	0.35	0.78	0.36
3	3	Roberto	0.25	0.35	0.55	0.45	0.68	0.26
4	4	Alvaro	0.25	0.35	0.15	0.45	0.97	0.25
*								

Figura 1. Información de un Perfil de Búsqueda del Usuario


```
"docs": [
  {
    "id": "6",
    "categoria": "ciencias",
    "score": 0.986511
  },
  {
    "id": "5",
    "categoria": "tecnologia",
    "score": 0.9865086
  },
  {
    "id": "1",
    "categoria": "noticias",
    "score": 0.9865067
  },
  {
    "id": "8",
    "categoria": "ciencias",
    "score": 0.9864976
  },
```


Figura 2. Resultados del sistema con el componente

2. **Categorización de los documentos:** Esta variable brindará la categoría de cada documento indexado en Solr.

Cuando el usuario realiza una consulta, la interfaz del buscador Orión interactúa con el motor de búsqueda y servidor de indexación Solr, el cual posee funciones para el cálculo de relevancia de información.

El componente intervendrá en el cálculo de relevancia de información a través de tres fases:

1. Recopila todos los documentos que tienen en su contenido el token o los tokens que componen la consulta del usuario. Este funcionamiento se corresponde con el Modelo Booleano antes mencionado.
2. Obtiene el resultado de similitud a través del Modelo de Espacio Vectorial, el cual se encuentra implementado en la clase “Similarity” de Solr.
3. Obtiene el valor de por ciento correspondiente a una categoría, la cual se consigue mediante técnicas de minería de datos. Es necesario puntualizar que un documento puede tener más de una categoría, lo cual contribuiría al

criterio de desempate en caso de que el sistema le otorgue a más de un documento el mismo resultado de similitud.

Según los autores Roberto Passailaigue Baquerizo, Paúl Rodríguez Leyva, Juan Pedro Febles, Hubert Viltres Sala y Vivian Estrada Sentí (2017) se plantea que el umbral de similitud inicialmente calculado oscila entre 0 y 1, siendo los documentos más relevantes los más próximos a 1. Cuando se suman la variable de PBU y la similitud inicial, el umbral de puntuación de relevancia aumenta de 0 a 2, por tanto, los documentos más relevantes son aquellos más cercanos a 2. De esta manera se garantiza proporcionar a los usuarios resultados más precisos y mejores relacionados con sus preferencias de búsqueda. La función principal de este componente es mejorar la relevancia de los documentos y posicionarlos en el tope de la búsqueda.

Arquitectura del componente

Se propone el desarrollo de la propuesta de solución sobre la base de una arquitectura basada en componentes. Según Szyperski (1998) el término componente está definido como: "unidad de composición de aplicaciones software, que posee un conjunto de interfaces y un conjunto de requisitos, y que ha de poder ser desarrollado, adquirido, incorporado al sistema y compuesto con otros componentes de forma independiente, en tiempo y espacio".

En dicho caso, las interfaces de un componente determinan tanto las operaciones que implementan como las que necesitan de otros componentes durante su ejecución. En dicha arquitectura es posible proporcionar un entorno compartido de interacción, o sea los componentes son colocados dentro de contenedores (Springer International Publishing AG, 2017).

A continuación, se presenta la arquitectura empleada en la propuesta solución (ver Figura 3). La arquitectura del servidor de indexación Solr dispone de controladores esenciales que procesan las solicitudes. Estas pueden ser solicitudes de consulta o solicitudes de actualización (SolrRequestHandler y UpdateHandler). En el caso del controlador QueryResponseWrite, retorna las respuestas a las consultas realizadas. La solución desarrollada es un componente de búsqueda, el cual se define en el archivo de configuración solrconfig.xml.

Revista Cubana de Ciencias Informáticas
Vol. 13, No. 2, Mes Abril-Junio, 2019
ISSN: 2227-1899 | RNPS: 2301
Pág. 91-104
http://rcci.uci.cu

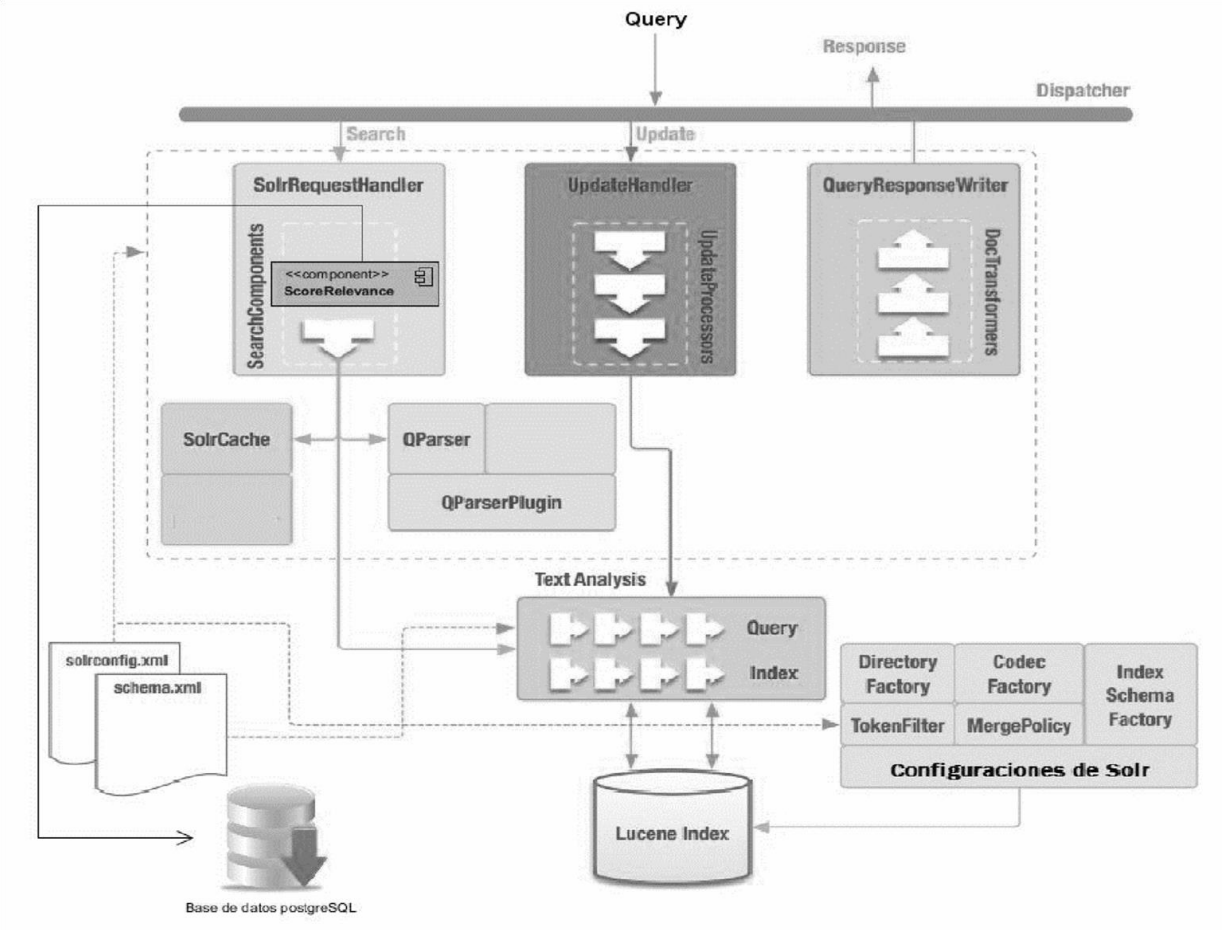

Figura 3. Arquitectura basada en componentes (Ilustración por Jorge Luis Betancourt González, tomada de la conferencia "Extendiendo Solr").

Para el funcionamiento de la solución se diseñaron las siguientes clases (ver Figura 4):

1. **MyQueryParserPlugin:** Contiene instancias personalizadas de procesamiento de consultas de usuarios de Qparser (Apache Lunece, 2017). Se implementa principalmente en el método "createParser" para construir los objetos de consulta apropiados. Luego es registrado en el archivo de configuración solrconfig.xml de esta manera:

<queryparser name="myqueryparser" class="my.package.MyQueryParserPlugin" />

2. **MyParser:** Componente responsable de analizar la consulta textual y convertirla en objetos Lucene Query correspondientes. Dentro de las formas de seleccionar qué analizador de consultas utilizar para una determinada solicitud se encuentran:

   LocalParams: Dentro del parámetro principal q o fq se puede seleccionar el analizador de consultas utilizando la sintaxis localParam. Ejemplo: & q= {! myparser} Cuba
3. **MyQuery:** Encargada de seleccionar la consulta existente y modificar cada puntuación de documentos mediante una devolución de llamada. Esta clase ejecuta la función "getCustomScoreProvider", el cual devuelve un objeto de tipo CustomScoreProvider. En su funcionamiento toma el control de dos cosas:
   - Correspondencia: qué documentos deben incluirse en los resultados de búsqueda.
   - Puntuación: qué puntuación debe asignarse a un documento. El objetivo final esta clase proporciona la capacidad de envolver una consulta de Lucene y redefinir su puntuación.
4. **MyScoreProvider:** En esta clase se utiliza un AtomicReaderContext, el cual es un contenedor en un IndexReader (Apache Software Foundation, 2012). Este objeto trabaja con las estructuras de datos disponibles para anotar un documento como el índice invertido de Lucene. En este preciso momento es cuando se redefine la puntuación del documento.
5. **TFIDF:** Esta clase utiliza el API de programación Solrj y es la encargada de realizar la interacción con Solr (Apache Software Foundation, 2017). En la estructura de Solrj se maneja el javabin que es mucho más rápido y menos pesado que el XML. De esta manera en la interación con el servidor de Solr se evita la carga extra de procesamiento para decodificar el XML.
6. **Categ_Por_ciento:** Es la clase encargada de construir un objeto con las variables categoría y por ciento.
7. **Conection:** Esta clase utiliza el JDBC (Java Data Base Conectividad), el cual es un API para la conexión de bases de datos desde el lenguaje Java. Una vez realizada la conexión se guardará en un arreglo la información referente al identificador del usuario, la categoría y el por ciento de búsqueda por cada categoría (Domínguez-Dorado, 2004).

Revista Cubana de Ciencias Informáticas
Vol. 13, No. 2, Mes Abril-Junio, 2019
ISSN: 2227-1899 | RNPS: 2301
Pág. 91-104
http://rcci.uci.cu

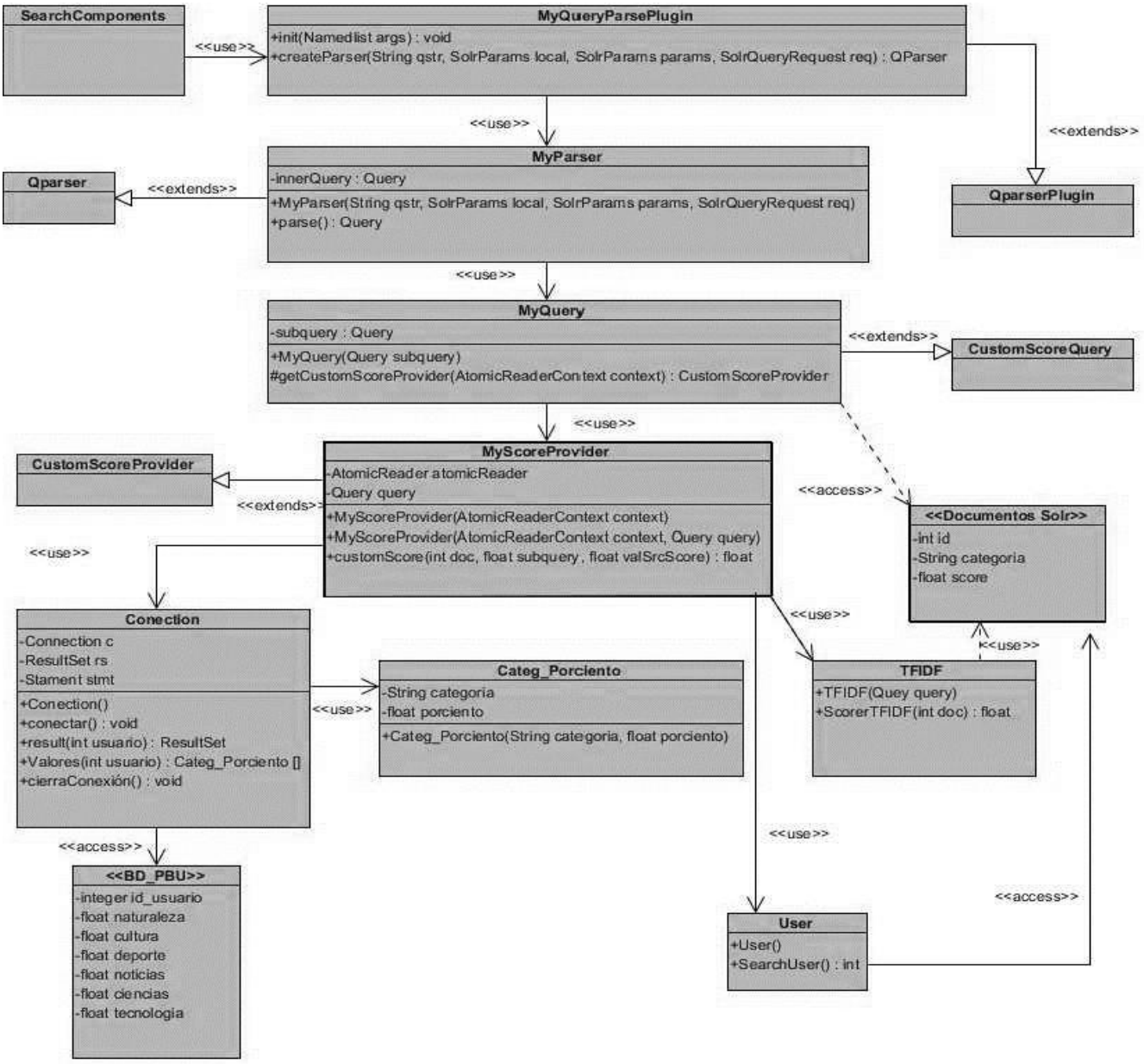

Figura 4. Diagrama de clases de diseño (elaboración propia)

Beneficios de la Aplicación Desarrollada

Con el desarrollo del componente los resultados de búsqueda estarán más ajustados a las preferencias de los usuarios. Para calcular la relevancia, lo habitual es establecer valores binarios: un documento es relevante, es decir, sirve como respuesta a nuestra pregunta, (valor 1) o no sirve (valor 0).

Revista Cubana de Ciencias Informáticas
Vol. 13, No. 2, Mes Abril-Junio, 2019
ISSN: 2227-1899 | RNPS: 2301
Pág. 91-104
http://rcci.uci.cu

Para esta tarea se tomó una muestra de 10 usuarios familiarizados con el uso del motor de búsqueda Orión. En la ejecución de las iteraciones fue necesario conocer las ecuaciones de la precisión y la exhaustividad en la recuperación de información. A continuación, se muestran las respectivas ecuaciones con sus resultados asociados (Kent A. Et al., 1955):

Precisión = Documentos_relevantes_recuperados / Documentos_recuperados,

Exhaustividad = Documentos_relevantes_recuperados / Documentos_relevantes.

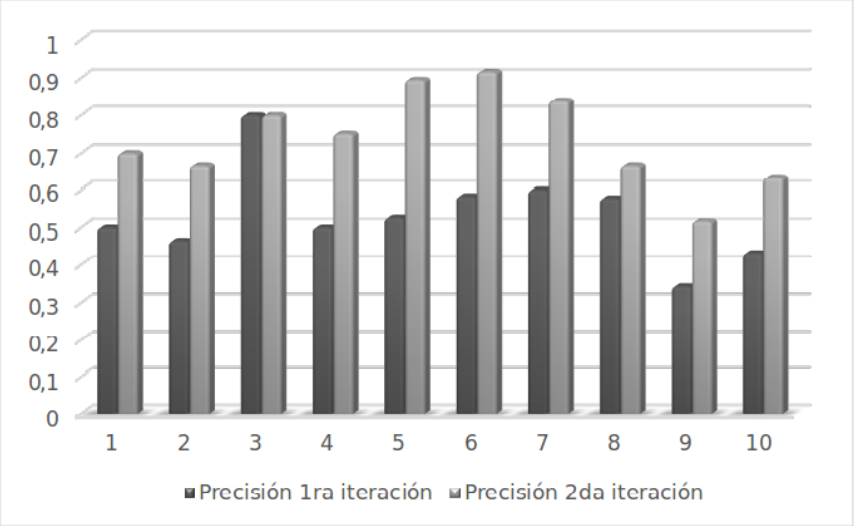

Figura 5. Precisión en la recuperación de la información

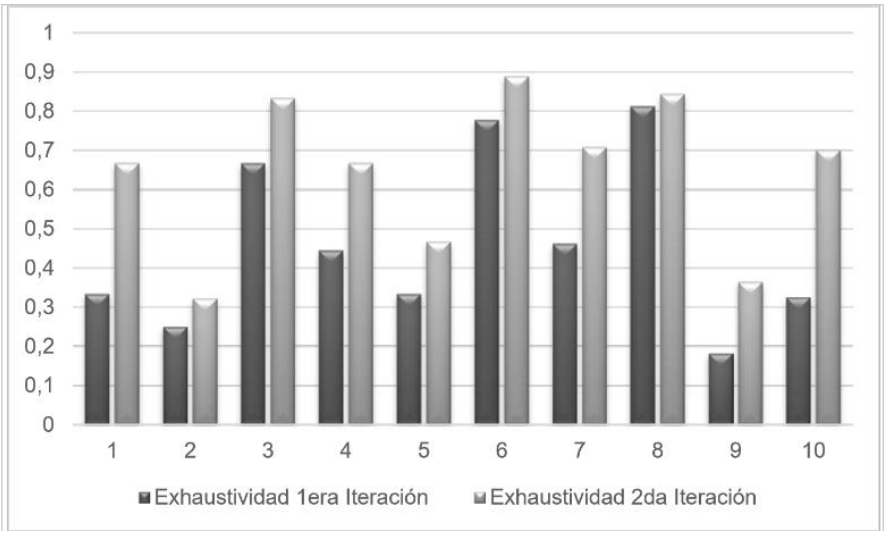

Figura 6. Exhaustividad en la recuperación de la información

A partir del análisis anterior en las iteraciones se evidencia un aumento significativo en los índices de precisión y exhaustividad en proceso de cálculo de relevancia en el componente. Apoyado en lo antes dicho, es posible afirmar que el sistema es más preciso y exhaustivo con la utilización del componente, debido a que los valores obtenidos en la segunda iteración son más cercanos a 1. En ese sentido, queda verificada la hipótesis de investigación anteriormente planteada.

Conclusiones

Una vez completada la presente investigación, se puede concluir que:

- A partir del marco teórico analizado en la presente investigación, se determinó que los buscadores web constituyen sistemas informáticos complejos y con gran número de operaciones de configuración. En esta ardua tarea intervienen los algoritmos de cálculo de relevancia de información, los cuales tienen en cuenta diferentes criterios de posicionamiento correspondientes al resultado de una consulta.
- Fue viable la modelación de los artefactos que mediaron en el diseño de la propuesta solución. En este sentido, se garantizó la estructura para la organización lógica del código fuente y la disminución del impacto ante posibles cambios.
- Se construyó un sistema modular a partir de la implementación de las clases en el lenguaje java, es decir se definieron funcionalidades independientes y menos susceptibles a futuros cambios.
- Se obtuvo un componente capaz de modificar la puntuación de relevancia de un documento a partir del perfil de búsqueda del usuario y las categorías de los documentos.
- La evaluación de las pruebas de software realizadas permitió erradicar las insuficiencias detectadas en el componente desarrollado, que comprometían la satisfacción del cliente y la facilidad de uso de las funcionalidades presentes.

Referencias

- **Apache Software Foundation.** *Using SolrJ.* [en línea] Apache Solr Reference Guide 6.6. 2017. [consultado: junio de 2017]. [Disponible en: https://www.apache.org/].
- **Apache Lucene.** *SolrPlugins: Solr Wiki.* [en línea] Apache Wiki. 3 de enero 2017. [consultado: junio de 2017]. [Disponible en: https://wiki.apache.org/solr/SolrPlugins].

- **Apache Software Foundation.** *Class AtomicReaderContext.* [en línea]. Lucene. 2012. [consultado: mayo 2017]. [Disponible en: https://lucene.apache.org/lucene/AtomicReaderContext.html].
- **Álvarez, I.** *Internet facturará más por publicidad que la televisión en 2017.* [en línea]. Forbes. 2016. [consultado: septiembre de 2017]. [Disponible en: http://forbes.es/up-down/.../internet-facturara-mas-por-publicidad-que-la-television-en-2017]
- **Baeza-Yates, R.** *Página Web de Ricardo Baeza-Yates.* 2017. [Disponible en: http://www.dcc.uchile.cl/~rbaeza/spanish.html]
- **Carrazana, E.** *Modelo Espacio Vectorial.* Universidad de las Ciencias Informáticas. 2014. [Disponible en: https://www.researchgate.net/Modelo_Espacio_Vectorial/]
- **Delgado, Y. H.** *Orión, un motor de búsquedas para la web de la UCI.* Trabajo de Diploma, Universidad de las Ciencias Informáticas, La Habana, junio del 2010.
- **Domínguez-Dorado, M.** *Todo Programación.* Nº 6. Págs. 35-38. Editorial Iberprensa (Madrid). DL M-13679-2004. Diciembre, 2004. Acceso a bases de datos desde aplicaciones Java: JDBC 1.0.
- **Kent A. Et al.,** (1955), *Machine literature searching. VIII.* Operational Criteria for Designing Information Retrieval Systems American Documentation Abril, 6 (2) p. 93-101.
- **Baquerizo,R. P.; Leyva, P. R.; Febles, J. P.; Sala, H. V.; Sentí, V. E.** *Algorithm for calculating relevance of documents in information retrieval systems.* s.l.: International Research Journal of Engineering and Technology (IRJET), 2017. Artículo científico.
- **Larreina, I. A.** *Posicionamiento en buscadores: una metodología práctica de optimización de sitios web.* El profesional de la información, v. 14, n. 2, marzo-abril 2005. Artículo científico
- **Méndez, F. J. M.** *Recuperación de información: modelos, sistemas y evaluación.* Universidad de Murcia. 2004.
- **Marcos, M. C.** *Entrevista a Ricardo Baeza-Yates de Yahoo!* [En línea] 2008. [Disponible en: http://www.hipertext.net.]
- **M. Coll, J. C.** *El uso de las tecnologías de la informática y las comunicaciones (TIC) en la formación de una cultura de estilos de vida sanos de los adolescentes***.** [En línea] 2011 ISSN: 1988-7833.
- **Román, J. V.** *Sistemas de Recuperación de Información. Ingeniería Sistemas Telemáticos,* Universidad de Valladolid: Departamento. 1997.

- ***Springer International Publishing AG.*** *Ingeniería de software basada en componentes.* Berlín, Alemania: s.n., 2017.
- **Szyperski, C.** *Component software: beyond object-oriented programming. 1*998. p: 411. ISBN: 0-201-17888-5.