Universidad de las Ciencias Informáticas

Facultad 1

“Componente para el cálculo de la relevancia de información en el buscador Orión”

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

Autor:

Alexander García O´reilly

Tutores:

Ing. Paúl Rodríguez Leyva

Ing. Yuneldis Reyes Velázquez

Ing. Maikel Maldonado del Toro

La Habana, 2017

Declaración de autoría

Declaro por este medio que yo Alexander García O´reilly con carnet de identidad 93051832409 soy el autor principal del trabajo titulado “Componente para el cálculo de la relevancia de información en el buscador Orión” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

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

Alexander García O´reilly

_____

Firma del Autor

Ing. Paúl Rodríguez Leyva

_____

Firma del Tutor

Ing. Maikel Maldonado del Toro

_____

Firma del Tutor

Ing. Yuneldis Reyes Velázquez

_____

Firma del Tutor

Agradecimientos

A mis familiares que siempre han estado interesados en mi desarrollo como futuro ingeniero. A la Revolución por darme la oportunidad y las herramientas para estudiar y optar por una carrera universitaria. A la Universidad de las Ciencias informáticas por desarrollar en mis ideas de un ser humano integral, respetuoso y voluntarioso. A la Facultad 1 que ha sido como una familia inmensa donde se aprende algo nuevo cada día. A nuestro eterno Comandante en Jefe que siempre vivirá en la memoria de todo cubano y será siempre un ejemplo a seguir.

Dedicatoria

En especial a mis padres y hermano por apoyarme en la ardua tarea de convertirme en Ingeniero. También a mi abuela Ana María por sus tantos consejos e inculcarme ideas como el sacrificio, la perseverancia y el amor en las cosas que se hacen. Además, a Fidel Castro por fundar la Universidad de las Ciencias Informáticas y darnos las herramientas para crear nuestro futuro.

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.

Índice de Contenido

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. 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). En esta área de la ciencia y la tecnología se trabaja en 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 mucha información en la red. 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 la presente investigación se hará énfasis en los buscadores web debido a que se ajusta a la temática abordada. En el funcionamiento general de un buscador se estudian dos perspectivas: la recopilación y la recuperación de la información. Un buscador compila automáticamente las direcciones que forman parte de su índice tras el proceso de indización. Una vez estén en el registro de la base de datos del buscador, los usuarios consultan su índice a través de una interfaz gráfica de búsqueda. El programa que realiza la recopilación es capaz de rastrear la estructura de hipertexto de la web, recogiendo información sobre las páginas que encuentra. Este rastreador puede recopilar millones de páginas por día, y actualizar la información depositada.

En el diseño de un buscador web se utilizan 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 el diseño de 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 el panorama internacional Google domina el ranking de los buscadores web, debido a que es el de más experiencia acumulada en las búsquedas de los usuarios (Elizalde, 2016). Según análisis estadísticos de uso y tráfico de datos es posible constatar que los usuarios tienen una vivencia positiva y utilitaria en la herramienta. Como resultado este buscador según su desempeño obtiene la fidelización del cliente y el éxito

en el mercado internacional.

En el ámbito nacional las instituciones no cuentan con un buscador web que satisfaga sus necesidades de información. En el caso de tener la posibilidad de acceder a Google, este maneja las prioridades, las búsquedas y la forma en que presenta los resultados de acuerdo a intereses y reglas que pueden no ser del todo útiles para el usuario. Igualmente se limita el acceso a ciertas informaciones por cuestiones políticas. En consecuencia, varias instituciones y países han coincidido en buscar una alternativa a dicho problema.

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. 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.

A través de filtros de consulta y otras funcionalidades de búsqueda, Solr se puede configurar con la flexibilidad necesaria para ayudar a los usuarios a refinar sus búsquedas, con el fin de devolver resultados más relevantes. De igual forma, este mecanismo se puede configurar para satisfacer a una comunidad de usuarios en particular, a través de lo que se denomina cálculo de la relevancia de información.

Algunas investigaciones (Diario Turing, 2013; Saquete, 2014 y Tablado, 2014) expresan que Google utiliza el algoritmo de relevancia *PageRank* para seleccionar las páginas web relacionadas con las palabras clave de búsqueda solicitadas por un usuario. Mediante este algoritmo Google proporciona el contenido cuya información es útil sobre esas palabras clave. De lo analizado anteriormente se infiere que es importante la presentación de los resultados de acuerdo a un orden de relevancia según los criterios de búsqueda empleados.

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.

Actualmente, los modelos clásicos de recuperación de información no manejan variables como las preferencias de búsqueda de usuarios (Rodríguez 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 este autor 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.

Teniendo en cuenta lo anteriormente expuesto se plantea el siguiente **problema de investigación**: ¿Cómo contribuir a ofrecer resultados más relevantes para los usuarios del buscador Orión?

Se define como **objeto de estudio** el proceso de recuperación de información en los motores de búsqueda y el **objetivo general** de la investigación es desarrollar un componente para el cálculo de la relevancia de información en el buscador Orión que permita ofrecer resultados más relevantes para los usuarios.

Por tanto, la investigación se enfoca en el **campo de acción** constituido por los mecanismos de cálculo de relevancia de la información en buscadores.

Como **objetivos específicos** se tienen:

1. Describir el estado del arte referente al cálculo de la relevancia de información en buscadores.

2. Diseñar e implementar las funcionalidades del componente para el cálculo de la relevancia de información en el buscador Orión teniendo en cuenta la categoría de los documentos y el perfil de búsqueda del usuario.
3. Realizar pruebas de software al componente para el cálculo de la relevancia de información en el buscador Orión para validar su correcto funcionamiento.

La investigación presenta la siguiente **hipótesis**: un componente para el cálculo de la relevancia de información en el buscador Orión teniendo en cuenta la categoría de los documentos y el perfil de búsqueda del usuario, permitirá mejorar la calidad de los resultados ofrecidos.

Variable independiente: un componente para el cálculo de la relevancia de información en el buscador Orión teniendo en cuenta la categoría de los documentos y el perfil de búsqueda del usuario.

Variable dependiente: la calidad de los resultados ofrecidos.

En la operacionalización de la variable dependiente se definió utilizar como métricas la precisión y la exhaustividad, debido a que son factores usados para medir el rendimiento de los sistemas de búsqueda y recuperación de información. De esta forma constituyen una forma de medir la relevancia de la información que se recupera (Díaz, 2003).

Para cumplir los objetivos trazados establecieron las siguientes **tareas de investigación**:

1. Realización de un estudio del estado del arte de soluciones para cálculo de la relevancia de información en buscadores.
2. Diseño de la arquitectura y selección de patrones a utilizar en la construcción de la solución.
3. Implementación de la solución.
4. Validación de solución a través de pruebas de software.

Los **métodos de investigación** utilizados en este trabajo son:

Métodos teóricos:

Analítico-Sintético: se utilizó en la búsqueda y análisis de los elementos más importantes asociados a los motores de búsqueda de Internet. Fue de gran importancia porque permitió analizar la bibliografía, así como los sistemas homólogos y estrategias en el ámbito del campo de acción.

Histórico-Lógico: permitió una mayor comprensión del estado actual de los motores de búsqueda de Internet, así como la evolución y desarrollo de sus componentes.

El Inductivo-Deductivo: posibilitó que luego de adquirir una serie de elementos referentes a los motores de búsqueda y el cálculo de relevancia, fuese posible alcanzar razonamientos que pudiesen ser aplicables a la solución del problema.

Modelación: permitió la modelación de diagramas usados en el análisis del problema, así como en el diseño e implementación de la propuesta de solución.

Métodos empíricos:

La Observación: se tuvo en cuenta a la hora de valorar la propuesta más adecuada de las tecnologías y soluciones existentes, además de permitir la observación del proceso de negocio en el proyecto Orión, con el objetivo de conocer su funcionamiento, organización y estructura de la información que contiene.

La Entrevista: se empleó en encuentros con profesores, especialistas y el Jefe de Departamento de Soluciones Informáticas para Internet del centro CIDI con el objetivo de entender mejor el negocio y definir las principales funcionalidades del sistema.

La investigación está estructurada en 3 capítulos:

Capítulo 1: "Fundamentos teóricos del cálculo de la relevancia de información en los buscadores web": en este capítulo se realiza un estudio del arte de las principales herramientas para el cálculo de la relevancia de información existentes en la actualidad en cuanto a su objetivo uso y funcionamiento. También se presentan los principales conceptos referentes a los motores de búsqueda y la función de relevancia utilizada en Orión. También, se realiza un estudio los lenguajes de modelado y las metodologías de desarrollo, seleccionando de ellas la que será utilizada en la realización del presente trabajo.

Capítulo 2: "Diseño del componente para cálculo de la relevancia de información en el buscador Orión": en este capítulo se definen los requerimientos de software mediante el levantamiento de información, las estructuras de datos y los artefactos necesarios para la implementación de la propuesta de solución. De igual forma se detalla el sistema desde el punto de vista ingenieril mediante el modelo de diseño realizado.

Capítulo 3: "Implementación y prueba del componente para cálculo de la relevancia de información en el buscador Orión": dentro del capítulo se exponen algunos aspectos asociados a la implementación de la solución informática, así como los componentes que la integran. Conjuntamente se presentan los casos de pruebas a utilizar en la validación del componente. De tal forma se analizan los resultados

alcanzados en las pruebas y se evalúa la calidad de la propuesta solución.

Capítulo 1: Fundamentos teóricos del cálculo de la relevancia de información en los buscadores web

En el presente capítulo se precisan un conjunto de elementos teóricos y características de los motores de búsqueda que ayudarán a comprender la base de su funcionamiento y uso. Antes de profundizar en estos contenidos es necesario conocer algunos conceptos fundamentales.

Según Ricardo Baeza-Yates (1999), estamos en presencia de SRI cuando: dada una necesidad de información o consulta realizada y un conjunto de documentos, como resultado se presentan un conjunto de documentos ordenados de mayor a menor relevancia. Este autor también hace referencia a que la Recuperación de Información (RI) es la representación, almacenamiento, organización y acceso a los índices de información.

Otros estudiosos del tema destacan que en la RI las principales actividades que se realizan son la indización y la búsqueda; cuando se habla de indización se refiere a la representación y descripción de la información. La RI dispone de componentes esenciales como los documentos estructurados y las bases de datos donde están almacenados. En estos procesos comienza la interacción usuario-intermediario, donde se ejecuta una consulta en lenguaje natural y esta se traduce para emprender una estrategia de búsqueda (Pinto, 2015).

Dentro de los SRI se encuentran los directorios, meta-buscadores y buscadores web, estos últimos son muy populares y de uso constante por los usuarios. Esta aplicación juega un papel importante en la necesidad de encontrar información, debido a que es posible explorar la red en busca de documentación de manera ágil, rápida y sencilla.

Un buscador web es un sistema informático que brinda la posibilidad de consultar una gigantesca base de datos para encontrar documentación en diferentes contenidos y formatos (Buscadores Web, 2016). Esta herramienta requiere operar de manera eficiente y eficaz en su estructura interna. En primer lugar, demanda algoritmos sólidos y bien estructurados capaces de resolver cualquier necesidad de información. Estos algoritmos se conforman a través de una estrategia de recuperación de información, la cual debe abarcar la mayor cantidad de documentación relevante. De igual forma para proporcionarles a los usuarios dichos documentos necesitan realizar un análisis exhaustivo de la información almacenada en índices.

La estructura de almacenamiento en índices permite el acceso rápido a la información, así como analizar y extraer entre toda la información disponible, la verdaderamente relevante. El proceso ejecutado para generar este índice es realizado por el motor de búsqueda, pero el rastreo de la información es llevado a cabo por un *web crawler* o *web spider*. En segundo lugar, es importante que el buscador se retroalimente

de informaciones estadísticas sobre las búsquedas que generan los usuarios. De tal forma se obtiene un mecanismo robot automático capaz de aprender, analizar e interpretar desde el punto de vista informático al menos de manera muy básica las necesidades humanas de información.

Se conoce como cálculo de relevancia al conjunto de operaciones mediante las cuales se decide si el contenido de una página es relevante en relación a la búsqueda que ha hecho el usuario según Fernández (2014). Si se toma en consideración las valoraciones plasmadas en la Guía de Referencia Apache Solr, 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.

Entre los conceptos importantes relacionados con los buscadores se encuentra: optimización para buscadores web (SEO). Este concepto referencia a un conjunto de medidas para mejorar el posicionamiento de una página web, en el ranking o posición de los diferentes motores de búsqueda al realizar una determinada consulta. En la actualidad el SEO es una parte esencial en la modificación y mantenimiento web, ya sea para mejorar su difusión o darse a conocer, un SEO correctamente realizado, facilitará a los diferentes buscadores la obtención de los datos de la página web (R&A Marketing., 2017).

1.1 ESTUDIO DE SISTEMAS HOMÓLOGOS

Internacionales

- Google[1].

Según las valoraciones de Erika Elizalde (2016), experta en buscadores, y del portal TN Relaciones (2016) Google es el mejor y más popular buscador que existe en Internet. Fue creado en el año 1997 y más del 90% de los usuarios de la red utilizan su servicio. Asimismo, es un motor de búsqueda que ha ampliado sus servicios y desarrollado herramientas hasta convertirse en el líder casi absoluto. Al reflexionar en este sentido se considera que son muchas las experiencias que se pudiesen tomar de dicho motor de búsqueda

[1] https://www.google.com.cu/

en pos del beneficio propio. Su robot de búsqueda es *Googlebot* desarrollado en el lenguaje Python[2] (Pinto, 2015). Determinados autores se refieren a este término también como *spider* o araña en español y cumplen un papel importante en la indexación y búsqueda de la información.

Según señala la propia Google Inc., su buscador filtra y muestra los resultados a través de tres fases: una primera fase de *crawling* ("búsqueda de páginas a nivel global"), una segunda en la que se filtran los resultados en base a los más de 200 factores que influyen en los algoritmos de Google y una tercera parte destinada a combatir el SPAM. Durante estas fases, Google trabaja a través de complicadas operaciones, la mayoría de las cuales son desconocidas para el usuario medio. Estas operaciones van cambiando y perfeccionándose con el paso del tiempo, adaptándose a las nuevas necesidades del medio (Fernández, 2014).

En este sentido se refiere al algoritmo *PageRank* el cual es una marca registrada y patentada por Google el 9 de enero de 1998 que ampara una familia de algoritmos utilizados para asignar de forma numérica la relevancia de los documentos ("o páginas web") indexados por un motor de búsqueda (Page, 1998).

La mayoría de los especialistas insisten que para realizar esta tarea el motor de búsqueda utiliza los *keywords* o palabras clave de la consulta, que son las palabras clave por las cuales un documento se posiciona en Internet. También hacen alusión al análisis matemático para calcular la densidad de las palabras clave, mediante el cual se puede conocer si la página está relacionada con el tema de búsqueda. En esa línea Google ofrece la herramienta Adwords para calcular la densidad de una palabra clave de forma rápida y cómoda.

Muchos usuarios muestran satisfacción y confianza cuando utilizan Google. Esto se debe a que "Google utiliza la estructura hipertextual de la web de dos maneras: primero para establecer el posicionamiento de los documentos recuperados a través del algoritmo y segundo, para extender las búsquedas textuales. También emplea esta estructura para extender la búsqueda a documentos que no han sido o no pueden ser indexados" (Galindo, 2000).

[2] Lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional.

- Yahoo! *Search*[3]

Este motor de búsqueda fue fundado en enero de 1994 por Jerry Yang y David Filo, ambos estudiantes graduados de la Universidad de Stanford. Yahoo comenzó siendo una lista de sitios realizada por los autores. A medida que navegaban por la web, recopilaban las direcciones de los sitios y los clasificaban a mano, sin la intervención de un sistema automático. Poco a poco fue adquiriendo notoriedad entre los internautas de la época, hasta convertirse en lo que hoy se conoce como Yahoo, una empresa que ha tenido muy buena aceptación con sus productos y servicios.

Utiliza el motor de búsqueda "*Search Assist*" ("asistente de búsqueda"), incluyendo una función que permite ver sugerencias mientras el usuario se decide entre las opciones encontradas por el buscador. Yahoo Slurp es el robot rastreador o *spider* de Yahoo, el mismo recopila documentos de la red de sitios web para construir un índice rastreable para servicios de búsqueda que usan el motor de búsqueda de Yahoo. Yahoo Slurp almacena toda la información que recoge durante el proceso de rastreo y luego almacena la información en la base de datos (Marrero, 2012).

De acuerdo con las valoraciones de Esteban (2011) este buscador utiliza un algoritmo similar al de Google, el algoritmo de posicionamiento *WebRank*. Sin embargo, este buscador tiene en cuenta la popularidad del sitio medida en las personas que utilizan la barra Yahoo! para acceder a él. Conjuntamente este autor hace mención a la importancia de la densidad de las palabras clave como aspecto fundamental para obtener un buen resultado. También hace énfasis en que las palabras clave de la URL tienen un mayor peso, sobre todo cuando más a la izquierda se encuentren.

- Bing[4]

Bing (anteriormente Live Search, Windows Live Search y MSN Search) es un buscador web de Microsoft[5]. Presentado por el antiguo director ejecutivo de Microsoft, Steve Ballmer, en la conferencia *All Things Digital* en San Diego, fue puesto en línea el 3 de junio de 2009. Está basado en la tecnología semántica de

[3] https://search.yahoo.com/

[4] https://www.bing.com/

[5] Microsoft: Empresa multinacional de origen estadounidense, fundada el 4 de abril de 1975 por Bill Gates y Paul Allen.

Powerset[6], la cual introduce notables cambios en la lista de sugerencias en tiempo real y la lista de búsqueda relacionada (Microsoft, 2015).

Además de buscar páginas web, Bing ofrece búsquedas como: traducir textos, buscar imágenes, noticias, ver videos en línea, seguir tendencias o clasificaciones de popularidad e indexar semánticamente referencias de Wikipedia[7] mejoradas visualmente (Microsoft, 2017). En julio del 2009, Microsoft y Yahoo! anuncian que Bing reemplazaría a Yahoo! *Search*, mediante este acuerdo con Yahoo, se crearía más innovación en la búsqueda, el mejor valor para anunciantes y elección del consumidor real en un mercado dominado por Google (BBC News, 2009). Msnbot era el robot de rastreo web desplegado por Microsoft para recopilar documentos de la web para construir un índice de búsqueda para el motor de búsqueda de MSN. En el mes de octubre de 2010 se sustituye Msnbot de la mayoría de los deberes activos de rastreo web por Bingbot (Tullis, 2010)

Hoy día, este buscador pone más atención a palabras clave directas. En ese sentido, para un mejor posicionamiento en Bing, es obligado construir una estrategia SEO[8] sobre palabras clave determinadas y concretas. Dentro de los factores de clasificación de este buscador se encuentra el uso de la búsqueda de las personas. Para dicha tarea emplean la barra de herramientas de Bing, la cual ve el flujo de datos del usuario, esencialmente todas las URL que se visitan (Itani, 2012).

A diferencia de Google, Bing no revela las actualizaciones realizadas a los algoritmos de búsqueda y posicionamiento. Solo especifica factores a tener en cuenta en la optimización de los sitios web para SEO. Un factor relevante para SEO es el contenido de calidad. Este es un factor prioritario en el algoritmo de clasificación de Bing, junto con la relevancia del tópico y el contexto. El contenido de calidad se compone de tres factores principales, fundamentales para la experiencia del usuario y para la búsqueda de información en la web (Corbalán, 2017):

---

[6] Powerset: Buscador semántico basado en el procesamiento del lenguaje natural, desarrollado socialmente.

7 Wikipedia: Autodefinida como un esfuerzo colaborativo por crear una enciclopedia gratis, libre y accesible por todos. Permite revisar, escribir y solicitar artículos.

[8] SEO (*Search Engine Optimization*): Proceso técnico mediante el cual se realizan cambios en la estructura e información de una página web, con el objetivo de mejorar la visibilidad de un sitio web en los resultados orgánicos de los diferentes buscadores.

- Autoridad: Confianza en el contenido
- Utilidad: Texto útil y suficientemente detallado.
- Presentación: Fácil de encontrar y de leer.

Buscadores nacionales

- C.U.B.A.[9]

Según se señala Contenidos Unificados para Búsqueda Avanzada (C.U.B.A.) es una plataforma que integra los servicios web disponibles en la red cubana. Está basada en la tecnología del motor de búsqueda Orión, desarrollado por la Universidad de las Ciencias Informáticas (UCI). Surge como respuesta a la necesidad de mostrar a los usuarios la información existente en el dominio cubano y brinda la posibilidad de acceder a sitios de interés cultural, informativos e investigativos. Esta plataforma facilita al usuario la búsqueda de contenidos en varios formatos digitales (“páginas web, imágenes y documentos”), proporciona referencias concretas de un tema alojado en varias fuentes. Una vez analizada la información proporcionada por el motor de búsqueda que utiliza la plataforma, el usuario puede tener una visión más amplia acerca de un mismo tema al contar con varias fuentes de información y diferentes materiales de consulta.

La relevancia de resultados es la medida predeterminada que se utiliza para determinar el orden en que se muestran los resultados de búsqueda. Como modelo matemático en la recuperación de la información para otorgar cierta relevancia a los documentos este buscador utiliza el Modelo Espacio Vectorial (VSM). En este modelo cada término se expresa como un vector t-dimensional, donde t representa la cantidad de términos asociados al documento. Cada elemento del vector tomará un valor de relevancia (peso) dependiendo de cuan representativo es en el documento (Salton y otros, 1975).

Para calcular el peso se emplea la ponderación TF.IDF que consiste en multiplicar los factores TF (abreviatura de *Term Frecuency*) y IDF (abreviatura de *Inverse Document Frecuency*). El primer factor refleja la importancia de los términos en el documento, concediendo mayor importancia a los términos que aparecen con mayor frecuencia en los documentos. El segundo factor dará mayor importancia a un término cuanto menor sea la colección de documentos donde aparece. Es decir, un IDF de un término es

[9] https://www.redcuba.cu/

inversamente proporcional al número de documentos en que aparece dicho término (Salton y otros, 1975).

- Lupa[10]

LUPA es un buscador de contenidos en la REDUNIV, ha sido desarrollado en la Universidad de Pinar del Río (UPR) por el Grupo de Desarrollo y Diseño Web de la Dirección de Informatización con el propósito de localizar contenidos dispersos en sitios web y repositorios públicos de la red MES y en sus enlaces transversales de alta velocidad (MINED, INFOMED, Cultura, Joven Club, UCI) (UPR, 2013).

Su motor de búsqueda Apache Solr presenta complejos procesos internos que aseguran el rendimiento de la aplicación. Para mejorar la interacción del usuario con la interfaz web, el equipo de desarrollo de la UPR ha realizado esfuerzos en lograr un diseño limpio y amigable, que intenta mostrar de forma intuitiva las principales funcionalidades del sistema. Según el propio equipo de desarrollo esta interfaz web ha sido optimizada para disminuir el tiempo de carga, lo cual favorece a usuarios en redes telefónicas y demás líneas de baja velocidad (UPR, 2013).

Algunas de las funcionalidades notables son (UPR, 2013):

Documentos Similares: Los documentos similares a otro documento son aquellos que comparten el mismo tema. El buen funcionamiento de esta característica depende de lo bien definido que esté el tema en el documento seleccionado como punto de partida.

Filtrado de Documentos por tipo de archivo: Lupa es capaz de organizar por tipo de archivo los resultados obtenidos en una consulta.

Búsquedas relacionadas: Las búsquedas relacionadas están formadas por una nube de frases o términos representativos del conjunto de documentos devueltos en una consulta. En Lupa se utilizan para ayudar al usuario a direccionar su búsqueda hacia los resultados que desea.

Esta herramienta ha sido implementada usando una combinación de programas libres encabezado por Apache Lucene, el robot rastreador Lucid Crawler y la herramienta de extracción de texto Tika, con el objetivo de lograr un poderoso y escalable buscador. Aunque la bibliografía consultada sobre este buscador, no brindó información asociada al modelo matemático de recuperación de información empleado en el

[10] http://lupa.upr.edu.cu/

cálculo de la similitud, se considera la posibilidad de que en este proceso se utilice el modelo espacio vectorial, debido a que utiliza como núcleo la librería de Lucene y el motor de búsqueda Solr.

Resultados obtenidos del estudio

Como resultado se tiene el estudio de los buscadores web más importantes, de los cuales se destacan algunas de sus características y funciones más relevantes. Basado en dicho estudio se concluye que ninguno de los modelos matemáticos de cálculo de relevancia de información presentes en los buscadores web utilizan el perfil de usuario (Rodríguez y otros, 2017). 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 la 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. A los efectos del planteamiento anterior, se añade que según Martínez Méndez (2004) el modelo espacio vectorial es el más utilizado en la web.

1.2 ANÁLISIS DE LAS TECNOLOGÍAS, HERRAMIENTAS Y LENGUAJES A UTILIZAR

Lenguajes utilizados

A continuación, se detallan los lenguajes empleados en el proceso de solución de la investigación correspondiente al desarrollo del cálculo de la relevancia de información en el buscador Orión.

Java

En la presente investigación se utilizará este lenguaje en la implementación sin el estudio previo de otros lenguajes de desarrollo. El planteamiento anterior se encuentra condicionado a que el servidor de indexación de contenidos Solr está implementado en este lenguaje. Por tal motivo, es necesario utilizar java para la correcta integración de la propuesta solución con Solr.

Java es un lenguaje de programación orientado a objeto desarrollado por Sun Microsystem a principio de

los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C[11] y C++[12], pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. Hasta la fecha, la plataforma Java ha atraído a más de 6.5 millones de desarrolladores alrededor del mundo y está presente en un elevado número de dispositivos, equipos y redes. Se caracteriza por su portabilidad y seguridad, lo cual la han convertido en la tecnología ideal para su aplicación a redes (Java Team, 2009).

XML

Se requerirá del empleo del lenguaje XML debido a ya que las configuraciones y los esquemas de Solr están condicionados por este lenguaje. En ese sentido XML es un subconjunto de SGML (Estándar *Generalised Mark-up Language*), simplificado y adaptado a Internet. Se define como: la especificación para diseñar lenguajes de marcado, que permite definir etiquetas personalizadas para descripción y organización de datos. Dentro de sus aplicaciones se utiliza para representar información estructurada en la web (todos documentos), de modo que esta información pueda ser almacenada, transmitida, procesada, visualizada e impresa, por muy diversos tipos de aplicaciones y dispositivos.

Ventajas de XML (Empresa Expertos en Servicios de Consultoría Exes, 2016):

- Fácilmente procesable.
- Separa radicalmente el contenido y el formato de presentación.
- Diseñado para cualquier lenguaje y alfabeto.

Características (Empresa Expertos en Servicios de Consultoría Exes, 2016):

- XML es un subconjunto de SGML que incorpora las tres características más importantes de este:
  - ✓ Extensibilidad
  - ✓ Estructura

[11] Lenguaje de programación originalmente desarrollado por Dennis Ritchie entre 1969 y 1972 en los Laboratorios Bell, como evolución del anterior lenguaje B, a su vez basado en BCPL.

[12] Lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al lenguaje de programación C mecanismos que permiten la manipulación de objetos.

- ✓ Validación

- Basado en texto.
- Orientado a los contenidos no presentación.
- Las etiquetas se definen para crear los documentos, no tienen un significado preestablecido.
- No es sustituto de HTML.
- No existe un visor genérico de XML.

SQL

Se hará uso del lenguaje SQL en las peticiones de consulta a la base de datos de PostgreSQL. Dentro de las características del mismo se considera que es estándar, y participa en la obtención de información importante para el funcionamiento del sistema que se implementará. Aunque SQL es a la vez un ANSI y una norma ISO, muchos productos de bases de datos soportan SQL con extensiones propietarias al lenguaje estándar. Las consultas toman la forma de un lenguaje de comandos que permite seleccionar, insertar, actualizar, averiguar la ubicación de los datos, y la creación y modificación de esquemas y el control de acceso a los datos. También se describe como un lenguaje declarativo e incluye elementos procesales (Escofet, 2007).

UML

El lenguaje Unificado de Modelado (UML) es el lenguaje estándar para visualizar, especificar, construir y documentar los artefactos del sistema, en el cual se incluye la estructura y el diseño. Utiliza símbolos y notaciones para representar los componentes que conforman la arquitectura de software. Permite el modelado de los procesos de negocio y el modelado de los requisitos apoyado en el análisis orientado a objetos.

Sistemas gestores de base de datos (SGBD)

Actualmente son muchas las aplicaciones que necesitan acceder a informaciones sin importan la magnitud de los datos. Por tal motivo las aplicaciones necesitan de un medio para acceder a ellos. Es aquí donde aparecen los SGBD, los cuales proporcionan una interfaz entre la aplicación y los datos. A continuación, se presentarán algunos de los SGBD más importantes y se procederá a la selección de uno de ellos para el uso en la investigación.

MySQL

MySQL es el Sistema de Gestión de Base de Datos de código abierto más popular del mundo, con más de 100 millones de copias a lo largo de su historia. Con su velocidad, fiabilidad y facilidad de uso, se ha convertido en la elección predilecta por los desarrolladores web. MySQL es parte importante de LAMP (Linux, Apache, MySQL y PHP), toda una suite o compendio de aplicaciones de código abierto con amplia aceptación por las empresas (Oracle Corp., 2016).

PostgreSQL

PostgreSQL es un Sistema de Gestión de Base de Datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones están tan competente como otras bases de datos comerciales. PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando (Martínez, 2010).

MongoDB

MongoDB es un Sistema de Gestión de Base de Datos escalable y de alto rendimiento de almacenes de datos No SQL. Es un sistema de código abierto escrito en C++ y está orientado al almacenamiento de datos en documentos al estilo JSON con esquemas dinámicos, que ofrecen potencia y simplicidad. Se destaca por conservar los índices de todos los atributos y hacer mucho más flexible la agregación y procesamiento de datos.

MongoDB se basa en colecciones, o sea, los datos se agrupan en conjuntos llamados "colecciones". Cada colección tiene un nombre único en la base de datos y puede contener un número ilimitado de documentos. Una colección es análoga a una tabla, excepto que no tienen un esquema definido (Hawkins y otros, 2010).

Selección del Sistema gestor de base de datos

El Sistema de gestor de base de datos a emplear es PostgreSQL por ser uno de los gestores más populares según destaca Rodrigo (2016). Su sintaxis SQL es estándar y fácil de entender, se puede administrar con facilidad y es multiplataforma. PostgreSQL ha ido incorporando gradualmente características NoSQL y que se amplían en la versión 9.4 e incorpora el tipo de dato JSONB.

Es posible afirmar que el gestor PostgreSQL está avanzando sólidamente en el soporte de características

no relacionales en acuerdo con la referencia de los desarrolladores en el proyecto Orión. Se podrá hacer uso de una sola herramienta para satisfacer sus requerimientos tanto relacionales (transaccionales y otros) como no relacionales (Fonticoba, 2013).

Servidor web

Un servidor web constituye un programa que permite alojar aplicaciones que comúnmente funcionan bajo la filosofía Cliente-Servidor. Estos sistemas son los encargados de procesar las peticiones de los clientes y devolver una respuesta mediante el uso del lenguaje HTML.

Nginx

Es un servidor HTTP de código abierto, distribuido bajo la licencia BSD[13] simplificada o licencia FreeBSD y de proxy inverso de alto rendimiento, igualmente funciona como servidor proxy para IMAP/POP3/SMTP. Fue desarrollado por Igor Sysoev para uno de los sitios más visitados en Rusia, Rambler. NGINX es muy conocido por su estabilidad, sus características, configuración simple y su bajo consumo de recursos, lo último como consecuencia de manejar requerimientos basados en eventos, a diferencia de Apache que lo hace basado en procesos y que permite asegurar un funcionamiento óptimo bajo mucha carga (Lucas, 2011).

Algunas de las características de Nginx son:

- Capacidad de manejar más de 10 000 conexiones simultáneas con bajo uso de memoria.
- Balanceo de carga.
- Tolerancia a fallos.
- Soporte TLS/SSL.
- Autenticación de acceso.
- Compresión y descompresión gzip.
- Reescritura de URL.

[13] Licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*), un tipo del sistema operativo *Unix-like*. Es una licencia de software libre permisiva como la licencia de *OpenSSL* o la *MIT License*.

- Limitaciones de conexiones concurrentes y respuestas.
- Manejo de ancho de banda.
- Proxy IMAP, POP3, SMTP.
- Procesamiento de datos XSLT.

Apache 2

Apache 2 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. La licencia Apache es descendiente de la licencia BSD. Esta licencia te permite hacer lo que quieras con el código fuente. Actualmente más del 60 por 100 de los administradores de toda la Web utilizan Apache. Se trata de la plataforma de servidores Web de código fuente abierto más poderosa del mundo. Día a día aumenta el número de corporaciones que aceptan este maravilloso código fuente abierto en su infraestructura IT (Kabir, 2003).

Es un servidor de red para el protocolo HTTP, elegido para poder funcionar como un proceso independiente, sin que eso solicite el apoyo de otras aplicaciones o directamente del usuario. Se distribuye como software modular multiplataforma, extensible, popular (fácil de conseguir ayuda/soporte) y gratuito (Márquez y otros, 2012).

Características de Apache:

- Apache es un servidor altamente configurable. Actualmente existen módulos para Apache que son adaptables a este, y están ahí para ser instalados de acuerdo a una necesidad. Otro elemento importante es que cualquiera que posea una experiencia decente en la programación de C o Perl puede escribir un módulo para realizar una función determinada.
- Apache permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.
- Permite la creación de ficheros de logs a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en el servidor.

Herramientas de desarrollo

A continuación, se describen las herramientas seleccionadas para la implementación del componente de cálculo de la relevancia de información según el perfil de usuario y la categorización de los documentos. También se enuncian otras tecnologías necesarias para cumplimentar determinadas tareas dentro del

desarrollo de la solución.

NetBeans IDE

"Esta es una herramienta de código abierto bajo GPL[14] que permite el desarrollo de aplicaciones con características modulares. Entre sus funcionalidades permite escribir, depurar, compilar y ejecutar programas, aparte de, que soporta otros lenguajes de programación. Se destacan funcionalidades como el auto-completamiento de código, capacidades para el diseño de GUI y la integración de Frameworks de desarrollo. Es un software multiplataforma y extensible que conjuntamente posee abundante documentación referente a las clases y APIs que lo conforman.

La herramienta NetBeans IDE desarrollada sobre la Plataforma NetBeans posee las ventajas de un cliente enriquecido, donde la mayoría de las aplicaciones tienen características semejantes visibles a través de menús, barras de herramientas, visualizaciones de progreso, representaciones de datos y otros componentes, potenciando el desarrollo de las interfaces de usuario. La plataforma permite hacer uso de varios elementos que facilitan la construcción de extensiones y aplicaciones con arquitectura modular. Entre estos se encuentra la internacionalización, el editor de datos para agregar funcionalidades, la personalización de los elementos de la pantalla y un generador de ayuda para el proyecto que se desarrolla en conjunto con las APIs y componentes reutilizables" (García y otros, 2013).

Visual Paradigm para UML

Herramienta de Ingeniería de Software Asistida por Computadora (CASE) que permite la creación de modelos visuales con Lenguaje de Modelado Unificado (UML). Es una herramienta profesional implementada en lenguaje Java que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Presenta una licencia gratuita y comercial, es de tipo multiplataforma y brinda soporte al ciclo de vida del software. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación (Visual Paradigm, 2016).

[14] Fue creada por Richard Stallman en 1989 para proteger los programas liberados como parte del proyecto GNU. La GPL original se basó en la unificación de licencias similares utilizadas en versiones anteriores de GNU Emacs, GNU Debugger y de GNU C Compiler.

PgAdminIII

Es una plataforma de desarrollo para el gestor de base de datos PostgreSQL. Este software fue diseñado para responder a las necesidades de todos los usuarios, desde la escritura de simples consultas SQL a la elaboración de bases de datos complejas. La interfaz gráfica es compatible con todas las características de PostgreSQL y facilita la administración. La conexión al servidor puede hacerse mediante TCP/IP y puede encriptarse mediante SSL para mayor seguridad. Algunas de sus principales características son (PostgreSQL tools, 2016):

- Multiplataforma.
- Diseñado para múltiples versiones de PostgreSQL y derivados.
- Amplia documentación.
- Acceso a todos los objetos de PostgreSQL.
- Interfaz multilingüe.

Apache Nutch

Nutch es un *spider* o *web crawler* libre de código abierto, desarrollado en Java por *The Apache Software Foundation.* Este ha sido implementado sobre la base de Apache Lucene, una librería de alto rendimiento para la búsqueda basada en texto. Utiliza una modificación del algoritmo "*Vector Space Model*" (Modelo de Espacio Vectorial en español), con un enfoque booleano que restringe las estimaciones de los resultados obtenidos. Es considerada la solución de código abierto más usada en motores de búsqueda. Posee una amplia comunidad de desarrolladores y usuarios (Apache Nutch, 2016; I.A.N. Mahecha, 2016).

"Este *spider* constituye una alternativa transparente a los buscadores comerciales más difundidos; los cuales poseen fórmulas propietarias de ranking, evitando la explicación de por qué una página determinada ocupa un lugar relevante en los resultados" (Aleman y otros, 2014).

Apache Solr

Apache Solr es un servidor de indexación desarrollado en Java por Yonik Seeley en 2004 para la compañía *CNET Networks*. Este servidor se comunica a través del protocolo HTTP (REST). Su mecanismo interno soporta formatos de XML y JSON; y su configuración es vía ficheros. Está basado en la librería *open-source* de búsqueda *full-text* Lucene y es orientado a documentos (NoSQL). Sus características principales incluyen

resaltado de búsqueda de facetas, la integración de las bases de datos y documentos. Posee caché a diferentes niveles, es extensible, altamente escalable y proporciona búsquedas distribuidas y réplica de índices. También tiene APIs[15] que hacen que sea fácil de utilizar desde cualquier lenguaje de programación (Marrero, 2012).

Metodología de desarrollo de software

Proceso de Desarrollo Ágil (AUP)

Según las valoraciones de los autores Roca Díaz y Pérez González (2012) AUP es una versión simplificada del Proceso Unificado Racional (RUP). Se describe como una metodología simple y fácil de entender para el desarrollo del software, a su vez utiliza las tendencias y técnicas de RUP. AUP abarca siete flujos de trabajos, cuatro ingenieriles y tres de apoyo: Modelado, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyectos y Ambiente. El modelado agrupa los tres primeros flujos de RUP (Modelado del negocio, Requerimientos y Análisis y Diseño). Dispone de cuatros fases igual que RUP: Creación, Elaboración, Construcción y Transición. Surge por la necesidad de acelerar los proyectos que sean pequeños.

La metodología es flexible, está orientada a pequeños equipos y presenta una significativa simplificación. Solo se utilizan los artefactos imprescindible y realmente necesarios.

Ciclo de vida AUP (Díaz y otros, 2012):

Se establecen cuatro fases que transcurren de manera consecutiva y que acaban con hitos claros alcanzados.

- Creación: El objetivo de esta fase es obtener una comprensión común entre cliente y equipo de desarrollo sobre el alcance del nuevo sistema, para definir una o varias arquitecturas candidatas.
- Elaboración: El objetivo es que el equipo de desarrollo profundice en la comprensión de los requisitos del sistema y en validar la arquitectura.
- Construcción: Durante la fase de construcción el sistema es desarrollado y probado al completo en

[15] Interfaces de Programación de Aplicaciones abreviada como API del inglés: *Application Programming Interface*, es un conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

el ambiente de desarrollo.

- Transición: El sistema se lleva a los entornos de preproducción donde se somete a pruebas de validación y aceptación y finalmente se despliega en los sistemas de producción.

Las disciplinas se llevan a cabo de manera sistemática a fin de desarrollar, validar y entregar el software de trabajo que responda a las necesidades. Las disciplinas son:

- Modelado: Comprender el negocio de la organización, comprender el dominio del problema abordado por el proyecto, e identificar una solución al mismo que sea viable.
- Implementación: Transformar el modelo realizado en código ejecutable y realizar pruebas de nivel básico, en particular pruebas unitarias.
- Prueba: Realizar una evaluación objetiva para asegurar la calidad. Esto incluye buscar defectos, validar que el sistema funcione como debería, y verificar que se cumplen los requerimientos.
- Despliegue: Planificar la liberación del sistema.
- Gestión de configuración: Administrar el acceso a los artefactos del proyecto.
- Gestión de proyectos: Dirigir las actividades que forman parte del proyecto.
- Ambiente: Facilitar todo el entorno que permita el normal desarrollo del proyecto.

Principios de AUP (Díaz y otros, 2012):

- El personal necesita saber lo que está haciendo: El trabajador no va a leer la documentación de los procesos en detalle, sino que quieren una orientación de alto nivel o formación de vez en cuando.
- Simplicidad: Todo se describe brevemente usando unas páginas, no miles de estas.
- Agilidad: El AUP se ajusta a los valores y principios de la Alianza Ágil.
- Centrarse en las actividades importantes: La atención se centra en las actividades que realmente cuentan.
- Herramienta de la independencia: Usar cualquier herramienta que se desee con AUP. Es recomendable utilizar herramientas simples o incluso herramientas de código abierto.

Metodología de desarrollo para la Actividad productiva de la UCI.

La variación AUP-UCI es una metodología que se puede adaptar a las características de cualquier proyecto de modo que el proceso sea configurable. Entre sus objetivos tiene aumentar la calidad del software que se produce, por ello se apoya en el Modelo CMMI-DEV v1.3.

Descripción de las fases AUP-UCI:

- Inicio: Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.
- Ejecución: En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.
- Cierre: En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Descripción de las disciplinas:

- Modelado de negocio: Se comprende cómo funciona el negocio que se desea informatizar para tener garantías de que el software desarrollado va a cumplir su propósito. Para modelar el negocio se proponen las siguientes variantes: Casos de Uso del Negocio (CUN), Descripción de Proceso de Negocio (DPN) y Modelo Conceptual (MC).
- Requisitos: Comprende la administración y gestión de los requisitos funcionales y no funcionales del producto. Existen tres formas de encapsular los requisitos: Casos de Uso del Sistema (CUS), Historias de usuario (HU) y Descripción de requisitos por proceso (DRP).
- Análisis y diseño: En esta disciplina los requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa de estos. Además, se modela el sistema y su forma (incluida su arquitectura) para que soporte todos los requisitos, incluyendo los requisitos no funcionales.
- Implementación: Se construye el sistema a partir de los resultados del análisis y diseño.
- Pruebas internas: Se verifica el resultado de la implementación probando cada construcción,

incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas.

- Pruebas de liberación: Pruebas diseñadas y ejecutadas por una entidad certificadora de la calidad externa, a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación.
- Pruebas de aceptación: Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido.

Escenarios para la disciplina Requisitos:

- Escenario No 1: Proyectos que modelen el negocio con CUN solo pueden modelar el sistema con CUS.
- Escenario No 2: Proyectos que modelen el negocio con MC solo pueden modelar el sistema con CUS.
- Escenario No 3: Proyectos que modelen el negocio con DPN solo pueden modelar el sistema con DRP.
- Escenario No 4: Proyectos que no modelen negocio solo pueden modelar el sistema con HU.

El escenario utilizado en el desarrollo de la solución es el No. 4, el cual aborda: “Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos, ya que una HU no debe poseer demasiada información” (Sánchez, 2015).

1.3 CONCLUSIONES PARCIALES DEL CAPÍTULO 1

El estudio de los buscadores web homólogos permitió conocer sus principales características y las diferentes valoraciones acerca del cálculo de la relevancia. Esas valoraciones proporcionaron un conjunto conocimientos sobre los rasgos y funciones de los spiders e indexadores en las búsquedas de la red. Basado en dicho estudio se concluye que es necesario desarrollar un componente que pueda ser integrado al servidor de indexación Solr, completando así su eficiencia en las búsquedas. Para argumentar dicho planteamiento fue posible constatar que los modelos clásicos de recuperación y posicionamiento de

información no cuentan con variables asociadas a las preferencias de los usuarios.

La descripción de las herramientas, metodologías y lenguaje definidos por los autores permitió la familiarización de los elementos del ambiente de desarrollo de la solución, al mismo tiempo de adquirir los conocimientos para su utilización enmarcado en dicha tarea. Para este proceso se ha seleccionado AUP en su variación UCI como la metodología indicada debido a que está diseñada para grupos pequeños, donde los proyectos no contengan volúmenes muy altos y necesiten una rápida implementación. Además, es una metodología que estandariza el proceso de desarrollo de software en los proyectos y da cumplimiento a las buenas prácticas que define CMMI-DEV v1.3.

También con la variación AUP-UCI solo se generan los artefactos importantes para el desarrollo de la solución. Se recurrió al Visual Paradigm para modelar con la cual se podrá realizar el análisis y diseño de la solución a través del lenguaje UML. Para la implementación se utilizará el lenguaje de programación Java, la herramienta NetBeans IDE en su versión 8, el servidor web Apache en su versión 2 y el servlest Tomcat 7 para el correcto funcionamiento de Solr.

.

Capítulo 2. Diseño del componente para el cálculo de la relevancia de información en el buscador Orión

En este capítulo se abordarán aspectos esenciales relacionados con el diseño de la propuesta de solución. Entre los elementos a destacar se encuentran el diagrama de modelo de dominio, mediante el cual se representan los objetos que intervienen en el sistema. Como vía para definir funcionalidades se generaron artefactos relacionados con los requisitos funcionales y no funcionales. Como parte del diseño se definieron estilos y patrones, de modo que estuviesen presentes las buenas prácticas del diseño y la programación.

2.1 CARACTERÍSTICAS DE LA PROPUESTA DE SOLUCIÓN

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:

- 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 (Rodríguez Leyva y otros, 2017).

- 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 Paúl Rodríguez Leyva y Hubert Viltres Sala (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 (Rodríguez y otros, 2017). La función principal de este componente es mejorar la relevancia de los documentos y posicionarlos en el tope de la búsqueda.

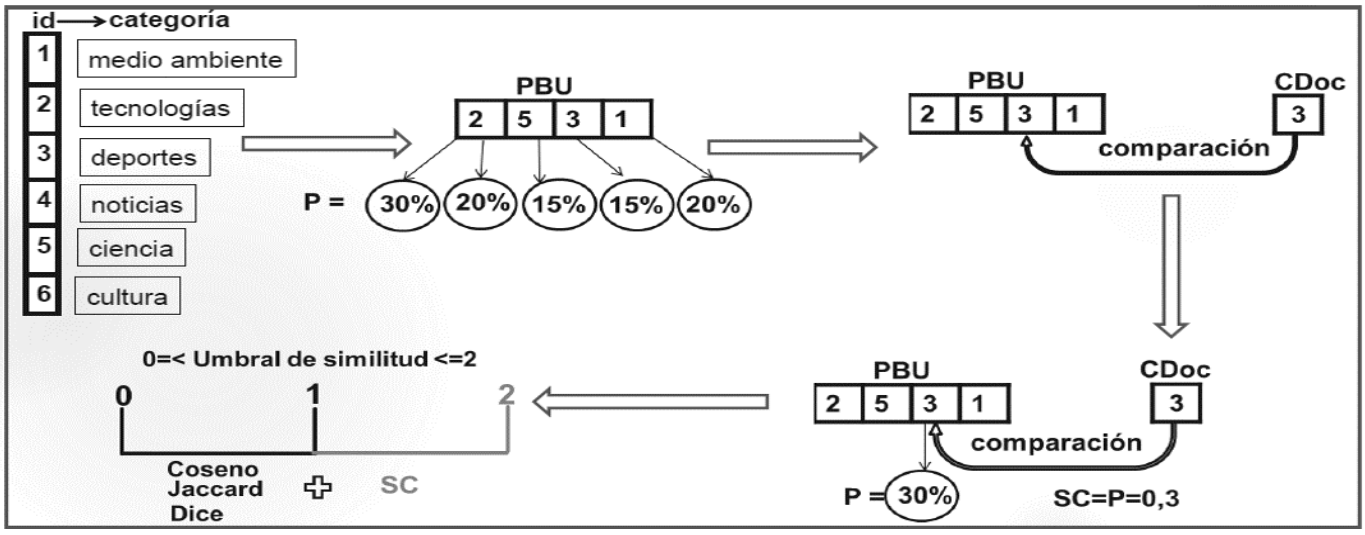

Figura 1 Tomada del artículo científico “Algorithm for calculating relevance of documents in information retrieval systems” (2017).

2.2 MODELO DE DOMINIO

Para la modelación del entorno y los procesos que intervienen en el sistema a informatizar se utiliza el modelo de dominio, en el cual se representan los conceptos más importantes y significativos en el desarrollo de la propuesta de solución. Este modelo permite identificar las relaciones entre entidades, clases y conceptos comprendidos en el ámbito del problema, al igual que sus atributos y restricciones. Lo anterior no implica que se modelan clases o componentes de software, si no conceptos propiamente dichos.

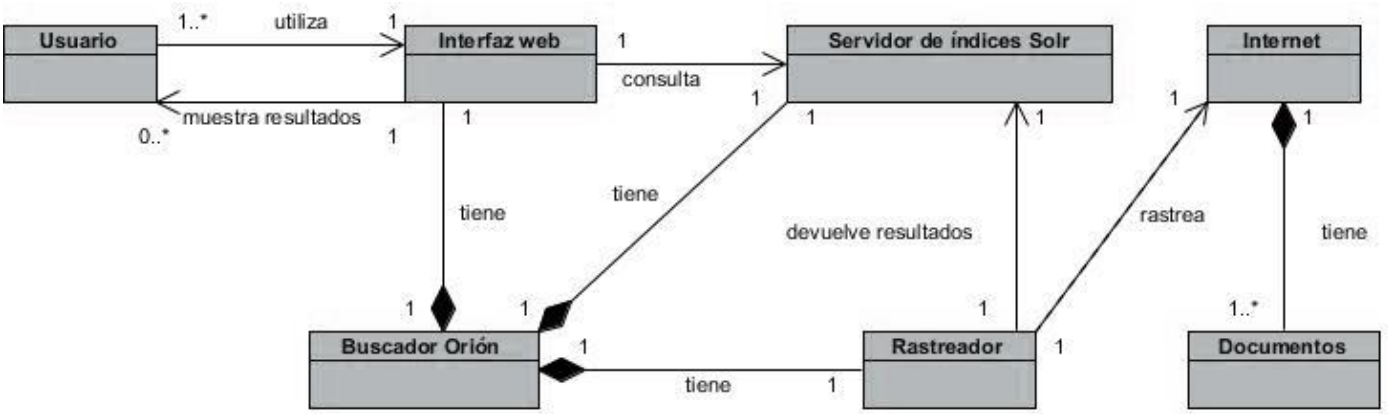

Figura 2 Diagrama de Clases del Modelo de Dominio (elaboración propia).

El modelo de dominio representado anteriormente muestra el proceso de consulta de un usuario al buscador Orión. El funcionamiento comienza cuando un usuario tiene una necesidad de información y decide utilizar la herramienta. El sistema te muestra una interfaz correspondiente con la que los usuarios interactúan. Como primer paso para ejecutar una consulta el usuario puede autenticarse o registrarse en el sistema y luego introduce los datos de la búsqueda. Posteriormente según la relevancia se devuelven los documentos indexados en Solr, los cuales son rastreados por Nutch desde la Internet.

Descripción de las clases del Modelo de Dominio

A continuación, se explica en qué consiste cada una de las clases que conforman el modelo de dominio.

Usuario: Representa a todas las personas registradas en el sistema y que acceden al sistema con el fin de realizar consultas.

Interfaz Web de Orión: Se refiere a la interfaz que brinda el buscador Orión una vez que el usuario decide realizar una consulta.

Buscador Orión: Sistema que se encarga de los procesos necesarios para devolver a los usuarios los

resultados de las consultas.

Servidor de indexación Solr: Componente del sistema que se encarga de indexar las páginas web y realiza el cálculo de relevancia de información.

Documentos indexados: Representa todos los documentos almacenados a los cuales se les realizará la consulta.

Usuarios registrados: Representa la base de datos de los usuarios registrados con la información de su perfil.

Araña Nutch: Componente del sistema encargado del rastreo y extracción de los contenidos verdaderamente relevantes.

Internet: Conjunto descentralizado de redes de comunicación de alcance mundial.

2.3 LEVANTAMIENTO DE INFORMACIÓN

En la etapa de levantamiento de información es necesario hacer uso de una identificación exhaustiva y correcta de los requisitos funcionales y no funcionales de la propuesta de solución. Con tal grado de compromiso se obtiene el éxito del proyecto en desarrollo. La base de esos requisitos estará en analizar qué debe cumplir la propuesta de solución para darle respuesta a los objetivos planteados en el trabajo.

Requisitos Funcionales identificados

Tabla 1. Requisitos Funcionales.

Código	Nombre de Requisito Descripción (Requisitos Funcionales)	Prioridad
RF1	• Obtener el perfil de búsqueda del usuario. Obtener del perfil del usuario el por ciento de búsqueda respecto a una categoría almacenada en la base de datos.	Alta
RF2	• Obtener categoría de los documentos. Obtener de la estructura de documentos de Solr el campo	Alta

	categoría con su respectivo valor.	
RF3	• Comparar el perfil de búsqueda del usuario con las categorías de los documentos indexados en Solr.	Alta
	Compara cada documento en Solr con las categorías registradas en el perfil del usuario.	
RF4	• Calcular valor de relevancia del documento.	Alta
	Adiciona el por ciento del perfil de búsqueda del usuario al cálculo de relevancia realizado por Solr.	

Requisitos no Funcionales identificados

Diseño e implementación:

- RNF1: Como lenguaje de programación para la implementación se utilizará Java.
- RNF2: Se requiere PostgreSQL como Sistema Gestor de Base de Datos.
- RNF3: Se requiere la instalación del servidor web Apache y del servlest Tomcat 7 para el correcto funcionamiento del servidor de Solr.
- RNF4: Se requiere de la máquina virtual de Java para el correcto funcionamiento del rastreador.

Licencia:

- RNF5: Se requiere el uso de herramientas y recursos de software libre, las cuales se podrán utilizar y modificar libremente.

Soporte:

- RNF6: El Sistema debe brindar la posibilidad de incorporarle nuevas funcionalidades en caso de ser necesarias.

Seguridad:

- RNF7: Se hace uso del lenguaje Java y del servidor web Apache en los procesos de envío de datos

en la tarea de cálculo de relevancia de información.

2.4 HISTORIAS DE USUARIO

Tabla 2. Descripción de HU "Calcular el perfil de búsqueda del usuario".

De	
Número: 1	Nombre del requisito: Obtener el perfil de búsqueda del usuario.
Programador: Alexander García O´reilly	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 25 horas
Riesgo en Desarrollo: Alta	Tiempo Real: 30 horas
Descripción: El usuario del sistema estará autenticado. Para esto se requiere una base de datos con las búsquedas realizadas por el usuario.	
Observaciones:	
Prototipo de interfaz:	

Tabla 3. Descripción de HU "Obtener categorización de los documentos".

Número: 2	Nombre del requisito: Obtener categoría de los documentos.
Programador: Alexander García O´reilly	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 25 horas
Riesgo en Desarrollo: Alta	Tiempo Real: 30 horas
Descripción: Se obtendrá la categoría de los documentos indexados en Solr, el cual será almacenado en el campo categoría.	
Observaciones:	
Prototipo de interfaz:	

Tabla 4. Descripción de HU "Comparar el perfil de búsqueda del usuario con las categorías de los documentos indexados en Solr".

Número: 3	Nombre del requisito: Comparar el perfil de búsqueda del usuario con las categorías de los documentos indexados en Solr.
Programador: Alexander García O´reilly	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 25 horas
Riesgo en Desarrollo: Alta	Tiempo Real: 30 horas
Descripción: El componente deberá comparar el PBU y la categoría de los documentos indexados en Solr. De tal forma se conocerá qué documentos serán más relevantes para determinado usuario.	
Observaciones:	
Prototipo de interfaz:	

Tabla 5. Descripción de HU "Asignar el nuevo valor de relevancia al documento".

Número: 4	Nombre del requisito: Calcular valor de relevancia del documento.
Programador: Alexander García O´reilly	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 25 horas
Riesgo en Desarrollo: Alta	Tiempo Real: 30 horas

Descripción: El componente adiciona el valor de PBU al cálculo de similitud original de Solr.
Observaciones:
Prototipo de interfaz:

2.5 ARQUITECTURA

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).
Algunos de los principales beneficios de esta arquitectura son (Springer International Publishing AG, 2017):

- Facilidad de Instalación: Cuando una nueva versión esté disponible será posible reemplazar la versión vieja por una más actualizada.
- Costos reducidos: El uso de componentes de terceros permitirá distribuir el costo del desarrollo y del mantenimiento.
- Facilidad de desarrollo: Los componentes implementan una interfaz bien definida para proveer u obtener una funcionalidad sin impactar otras partes del sistema.
- Reusable. El uso de componentes reutilizables significa que ellos pueden ser usados para distribuir el desarrollo y el mantenimiento entre múltiples aplicaciones y sistemas.

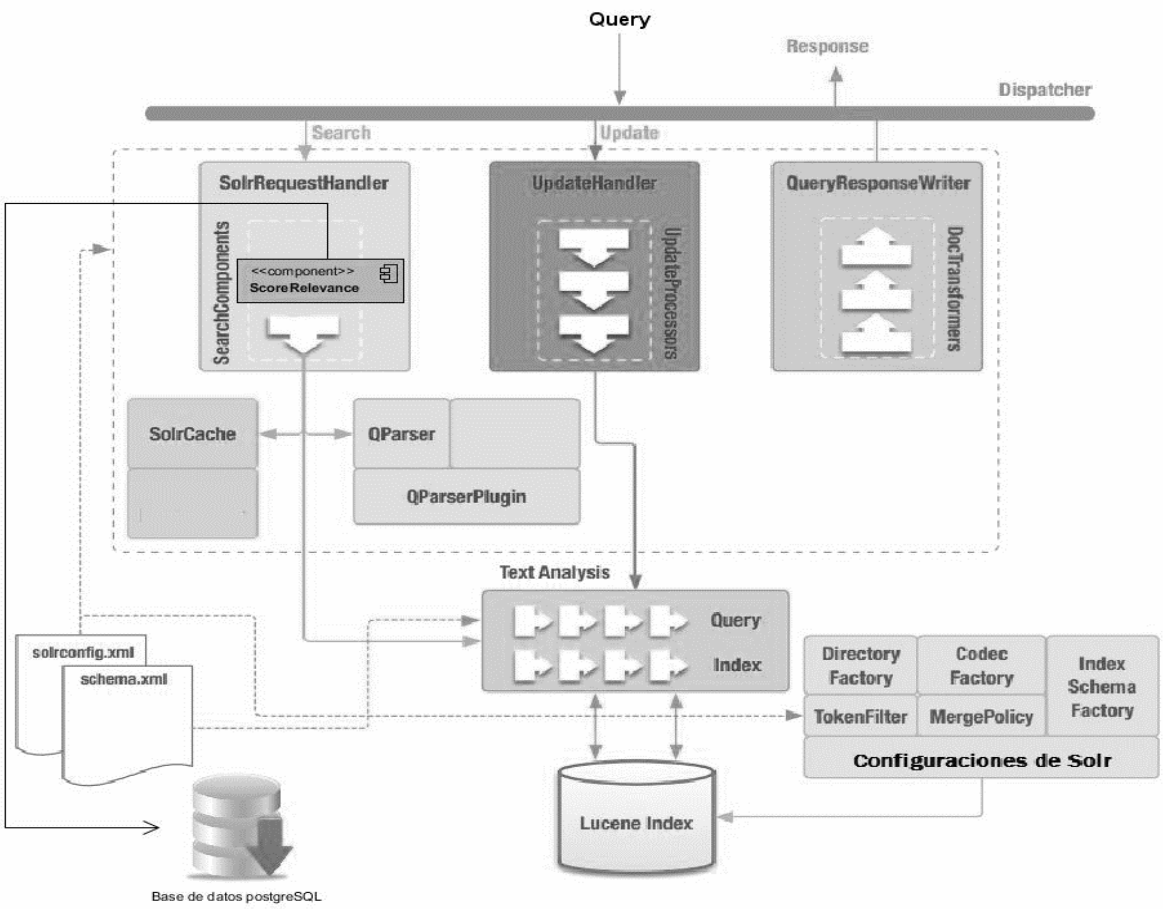

Figura 3 Arquitectura basada en componentes (Ilustración por Jorge Luis Betancourt González, tomada de la conferencia "Extendiendo Solr").

2.6 DISEÑO

Patrones de Diseño

"Un patrón se define como una solución probada con éxito que aparece una y otra vez ante determinado tipo de problema en un contexto dado. Los patrones se definen por un nombre, un problema, una solución y las consecuencias de su aplicación. Este define una posible solución correcta para un problema de diseño dentro de un contexto dado, describiendo las cualidades invariantes de todas las soluciones" (Abreu y otros, 2013).

Patrones GOF

Estos patrones están definidos por *The Gang of Four* (GOF) en procesos de desarrollo de software

orientados a la Web. El grupo GOF se dedicó a analizar los problemas recurrentes en el desarrollo de software y realizaron una clasificación y agrupación a partir de dos criterios, su propósito y alcance. Las categorías utilizadas son las siguientes (Suárez y otros, 2013):

- De creación:

Singleton (instancia única): Este patrón está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Su intención consiste en garantizar que cada clase tenga una instancia única y un punto de acceso a la misma. Se hace uso de este en la clase “Conection”, la cual es un objeto que utiliza una base de datos que es una instancia única para el proyecto en uso.

- Estructural:

Facade (Fachada): Este patrón está motivado por la necesidad de estructurar un entorno de programación y reducir su complejidad con la división en subsistemas, minimizando las comunicaciones y dependencias entre estos. Se hace uso de este en la clase “MyScoreProvider” en el método “customScore”, debido a que se conocen cuáles son los subsistemas responsables de determinada petición y delega esas peticiones a los objetos involucrados del subsistema.

Patrones Generales de Software para Asignación de Responsabilidades (GRASP)

GRASP son patrones generales de software para asignación de responsabilidades, es el acrónimo de "GRASP (*object-oriented design General Responsibility Assignment Software Patterns*)", se utiliza en el diseño orientado a objetos. Se considera que más que patrones propiamente dichos, son una serie de "buenas prácticas" de aplicación recomendable en el diseño de software (Departamento de Lenguajes y Sistemas Informáticos, 2017).

- Experto:

Este patrón plantea que se debe asigna una responsabilidad al experto en información, o sea a la clase que cuenta con los datos necesarios para cumplir la responsabilidad. De esta forma los objetos ejecutan las tareas de acuerdo a la información que poseen, lo cual trae como beneficio que se conserve el encapsulamiento. Este patrón se utiliza en todas las clases del sistema debido a que cada una de ellas es responsable de una actividad en específico, de modo que se distribuye el comportamiento. Un ejemplo es la clase “MyScoreProvider”, la cual se encarga de personalizar la puntuación de los documentos.

- Creador:

Garantiza una guía a la asignación de responsabilidades cuando se crea un objeto, con lo que se logra menos dependencias cuando se quiera reutilizar el código. La instanciación de una clase es una actividad fundamental del sistema por lo que está presente en todas las clases. Un ejemplo de ello es en la clase MyScoreProvider, la cual es la encargada de crear un objeto de la clase TFIDF para obtener la puntuación original de los documentos otorgada por Solr.

- Alta cohesión:

Este patrón tiene como objetivo asignar responsabilidades de tal forma que la cohesión siga siendo alta. En la programación orientada a objeto, la cohesión es la medida de la fuerza con que se relacionan las clases de un sistema. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. La aplicación de este patrón se manifiesta en las clases “MyQueryParserPlugin”, “MyQuery” y “MyScoreProvider”, creadas con el objetivo de generar el flujo de la información en dicho orden y donde se focaliza la responsabilidad de cada elemento.

Diagrama de Clases de Diseño

El Diagrama de Clases es el diagrama principal de diseño y análisis para un sistema. En él, la estructura de clases del sistema se especifica, con relaciones entre clases y estructuras de herencia. Durante el análisis del sistema, el diagrama se desarrolla buscando una solución ideal. Durante el diseño, se usa el mismo diagrama, y se modifica para satisfacer los detalles de las implementaciones (IBIBLIO, 2002).

Con el objetivo de alcanzar un mayor grado de comprensión de la solución a desarrollar, es indispensable tener en cuenta el funcionamiento del motor de búsqueda Solr, específicamente los paquetes que este genera. Para el funcionamiento de la solución se diseñaron las siguientes clases:

- MyQueryParserPlugin: Contiene instancias personalizadas de procesamiento de consultas de usuarios de QParser. 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" />
```

- 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

- 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.
- MyScoreProvider: En esta clase se utiliza un AtomicReaderContext, el cual es un contenedor en un IndexReader. 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.
- TFIDF: Esta clase utiliza el API de programación Solrj y es la encargada de realizar la interacción con Solr. En la estructura de Solrj se maneja el *javabin*[16], 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.
- Categ_Por_ciento: Es la clase encargada de construir un objeto con las variables categoría y por ciento.
- 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.

[16] Formato binario personalizado que se utiliza para escribir la respuesta de Solr de una manera rápida y eficiente.

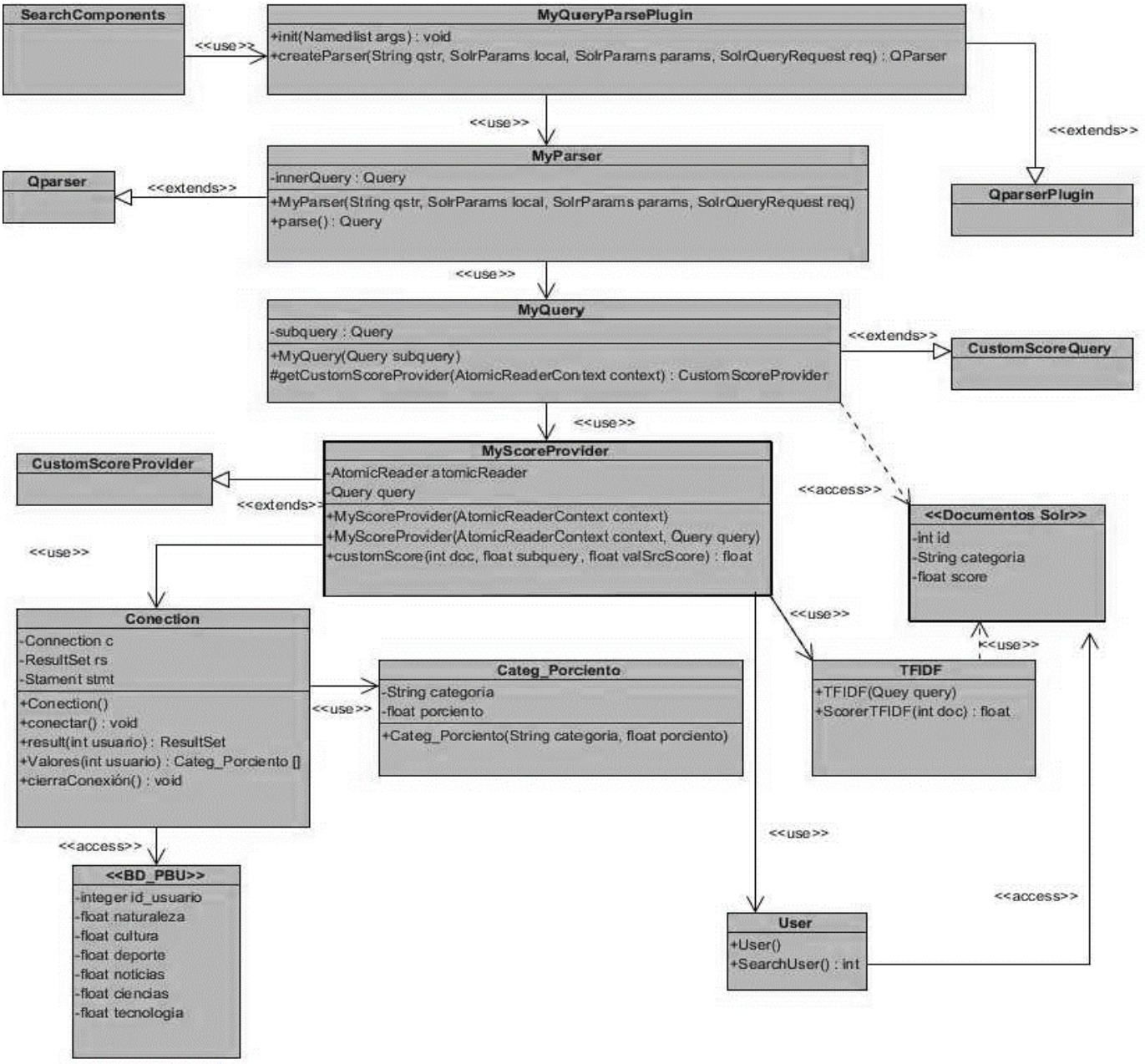

Figura 4 Diagrama de clases de diseño (elaboración propia).

2.7 MODELO DE DESPLIEGUE

El diagrama de despliegue contiene los nodos que conforman la topología hardware sobre la que se ejecuta el sistema, el software necesario para su funcionamiento y los protocolos de comunicación. A continuación, se describe y muestra el diagrama de despliegue correspondiente a la integración del componente para el

cálculo de la relevancia de información y el buscador Orión.

Como se puede apreciar en la Figura 5, el nodo “Dispositivo Cliente” representa un dispositivo utilizado por el usuario desde el cual se puede acceder a la interfaz del buscador Orión, a través de un navegador web y el protocolo de comunicación HTTP. El nodo “Postgresql” constituye la base de datos encargada de almacenar la información relacionada con el perfil del usuario. Este se comunica con la interfaz del buscador mediante el protocolo TCP. Además se observa el nodo “Servidor de indexación Solr”, el cual se encarga de atender todas las peticiones de consulta y se conecta a la interfaz mediante el protocolo HTTPS. Para el correcto funcionamiento de dicho nodo es necesario la instalación del servlest Tomcat 7 y del servidor de aplicaciones web Apache. También se aprecia el nodo “Nutch” como robot rastreador, el cual se conecta a Solr mediante protocolo HTTPS.

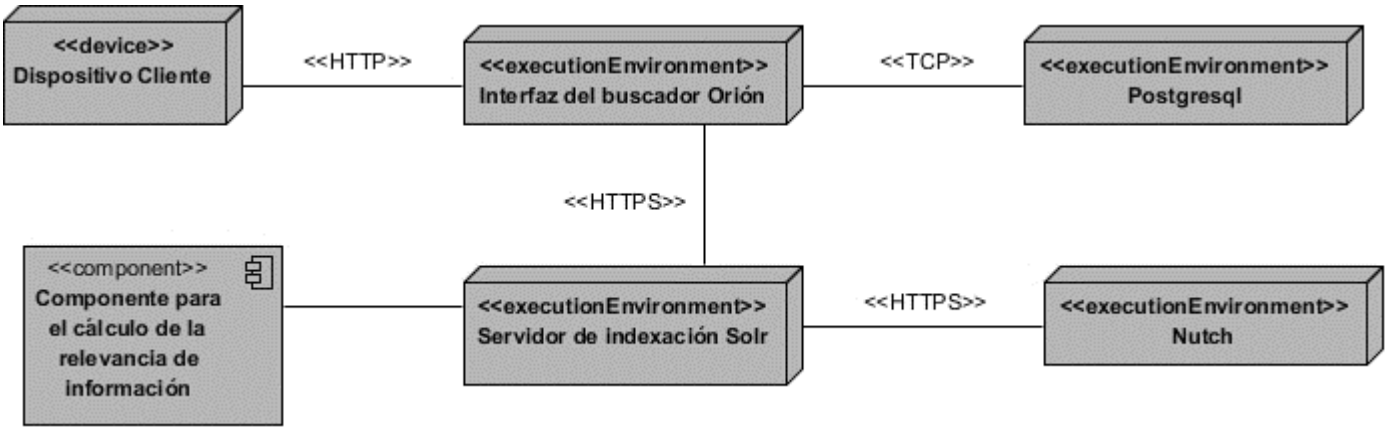

Figura 5 Diagrama de despliegue (elaboración propia).

2.8 MODELO DE DATOS

La permanencia de la información es una de las exigencias clave en el diseño de la arquitectura de una aplicación informática. Se considera que es de vital importancia la realización de un diseño de base de datos que cumpla con las buenas prácticas en la representación de las clases entidades y sus relaciones. Estas permitirán almacenar la información con un mínimo de redundancia, pero que a la vez faciliten su recuperación.

El componente se propone hacer uso de una base de datos, puesto que necesita la información referente al usuario y su perfil de búsqueda. Como ya se ha definido anteriormente la información será almacenada con PostgreSQL, un sistema gestor popular en la comunidad de código abierto. Para dicha tarea se

describen las colecciones de la base de datos:

- BD_PBU: Almacena la información del usuario registrado con su respectivo perfil de búsqueda (PBU).
- BD_user_conection: Almacena y provee información sobre qué usuario se encuentra conectado y autenticado.
- BD_Lucene_Index: Almacena los documentos en Solr a través de un conjunto de campos.

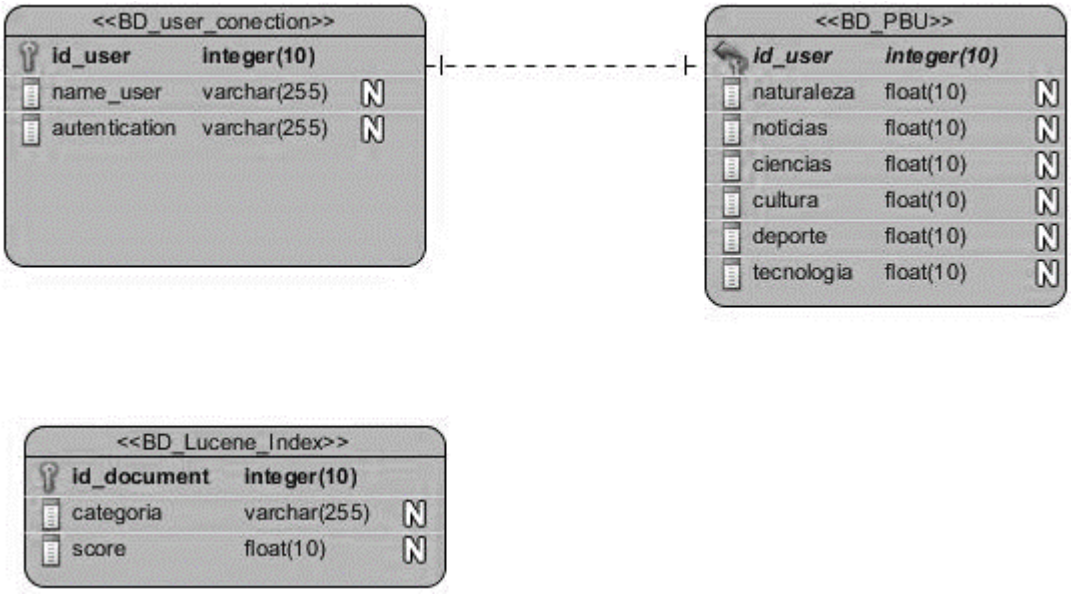

Figura 6 Diagrama de Modelo de Datos (elaboración propia).

2.9 CONCLUSIONES PARCIALES DEL CAPÍTULO 2

En este capítulo se realizó el análisis, el diseño y la arquitectura en general del componente para el cálculo de la relevancia a partir del perfil de usuario y la categoría de los documentos. Para cumplir dicha tarea se representaron y describieron los artefactos definidos por la metodología de manera que se entendiera el flujo de trabajo presente. Se representó el entorno en el que se ubica el problema a través del modelo de dominio, que constituye un punto de partida para el diseño del componente. La especificación de los requisitos funcionales y no funcionales sirvió de guía para desarrollar las funcionalidades y satisfacer las necesidades detectadas. Además, se realizó una breve descripción de los patrones de diseño utilizados en el desarrollo de la solución con el fin de fomentar la reutilización y las buenas prácticas de programación.

En conclusión, lo antes dicho garantizará la obtención de un producto flexible y modular capaz de adaptarse al cambio futuro de su código fuente.

Capítulo 3. Implementación y prueba del componente para cálculo de la relevancia de información en el buscador Orión

En este capítulo se realiza la fase implementación y posteriormente las pruebas al componente desarrollado. Dentro de la fase de implementación corresponde la materialización, en forma de código, de todos los artefactos, descripciones y arquitectura propuesta en la etapa de diseño.

Junto al proceso de implementación, el componente debe ser expuesto a diferentes pruebas que validen la correspondencia entre el producto y los requisitos definidos. En esta fase pueden realizarse diferentes pruebas en función de los objetivos de las mismas.

3.1 DIAGRAMA DE COMPONENTES

En este diagrama se ilustra la relación existente entre los componentes de software, así como la ubicación de cada uno de ellos en el sistema. Además, se describe cómo se implementan las clases en términos de componentes, como pueden ser ficheros de código fuente, ejecutables, entre otros. Igualmente detallan cómo se organizan los componentes de acuerdo a los mecanismos de estructuración disponible en el entorno de implementación y en el lenguaje de programación utilizado. Al mismo tiempo muestra las dependencias entre componentes (Pressman, 2010).

A continuación, se muestran las descripciones de los componentes y el diagrama de componentes, este último acorde a los requerimientos del sistema a implementar y la arquitectura definida.

Descripción del diagrama de componentes

La distribución de los componentes está conforme a la arquitectura propuesta en el capítulo anterior, de modo que se entienda la estructura del sistema. Los componentes son los siguientes:

- SolrRequestHandler: Maneja las peticiones (*request*) con *URL´s REST*[17] *style* como por ejemplo *"/select"* e incluye otros componentes de búsqueda (SearchComponents) que manejan las peticiones. Dentro de los SearchComponents se encuentran: Query, Facet, Highlight, Debug y Stats.

[17] La Transferencia de Estado Representacional (en inglés *Representational State Transfer*) o *REST* es un estilo de arquitectura software para sistemas hipermedia distribuidos como la *World Wide Web*.

- MyQueryParserPlugin: Ofrece un control de cómo las consultas son analizadas. Este analizador extiende sus funcionalidades de las subclases de “QParserPlugin” como necesidad de personalizar el análisis.
- QParser: Encargada de instanciar la consulta y convertirla en objeto de Lucene.
- MyQuery: Permite conformar otra consulta y personalizar la puntuación de la misma derivada de cálculos previamente definidos.
- IndexSearcher: Este componente actúa como el núcleo que lee y obtiene los índices durante el proceso de búsqueda.
- CustomScoreProvider: Provee la puntuación personalizada el método “customScore” sobrescrito. De esta forma se accede individualmente a cada documento y se le modifica su puntuación.
- TFIDF: Este componente establece conexión con el servidor de contenidos Solr y devuelve la puntuación de similitud original otorgada por el mecanismo a cada documento.
- Conection: Establece la conexión a la base de datos mediante el “*driver JDBC”.* Para cargar el *driver* es necesario utilizar la función “Class.forName”, de igual forma es necesario el método “DriverManager.getConnection” para hacer conexión a la base de datos. Una vez concretada la conexión se utiliza el objeto de tipo “ResultSet” y la función “executeQuery” para mostrar los resultados almacenados en la base de datos.
- IndexReader: Este componente proporciona una interfaz para acceder a un índice almacenado por Solr. La búsqueda de un índice se hace completamente a través de esta interfaz abstracta. Las subclases concretas de IndexReader se construyen generalmente con una llamada al método estático “open ()”. Un IndexReader se puede abrir en un directorio, pero no se puede utilizar para borrar documentos del índice (Apache Lucene, 2017).
- QueryResponseWriter: Componente que serializa y transmite la respuesta final al cliente. Utiliza una interfaz la cual detecta el tipo de contenido y lo escribe en el cuerpo de respuesta. Dentro de los formatos de respuesta se encuentran “JSON Response Writer” y “XML Response Writer”.
- User: Componente que establece conexión con el servidor de contenidos Solr mediante el API Solrj, la cual permite consultar los índices de contenidos almacenados. Su función principal es conocer qué usuario se encuentra conectado y cuál es su identificador.

- Documentos: Representa la estructura de almacenamiento de los documentos en índices determinada por Solr.

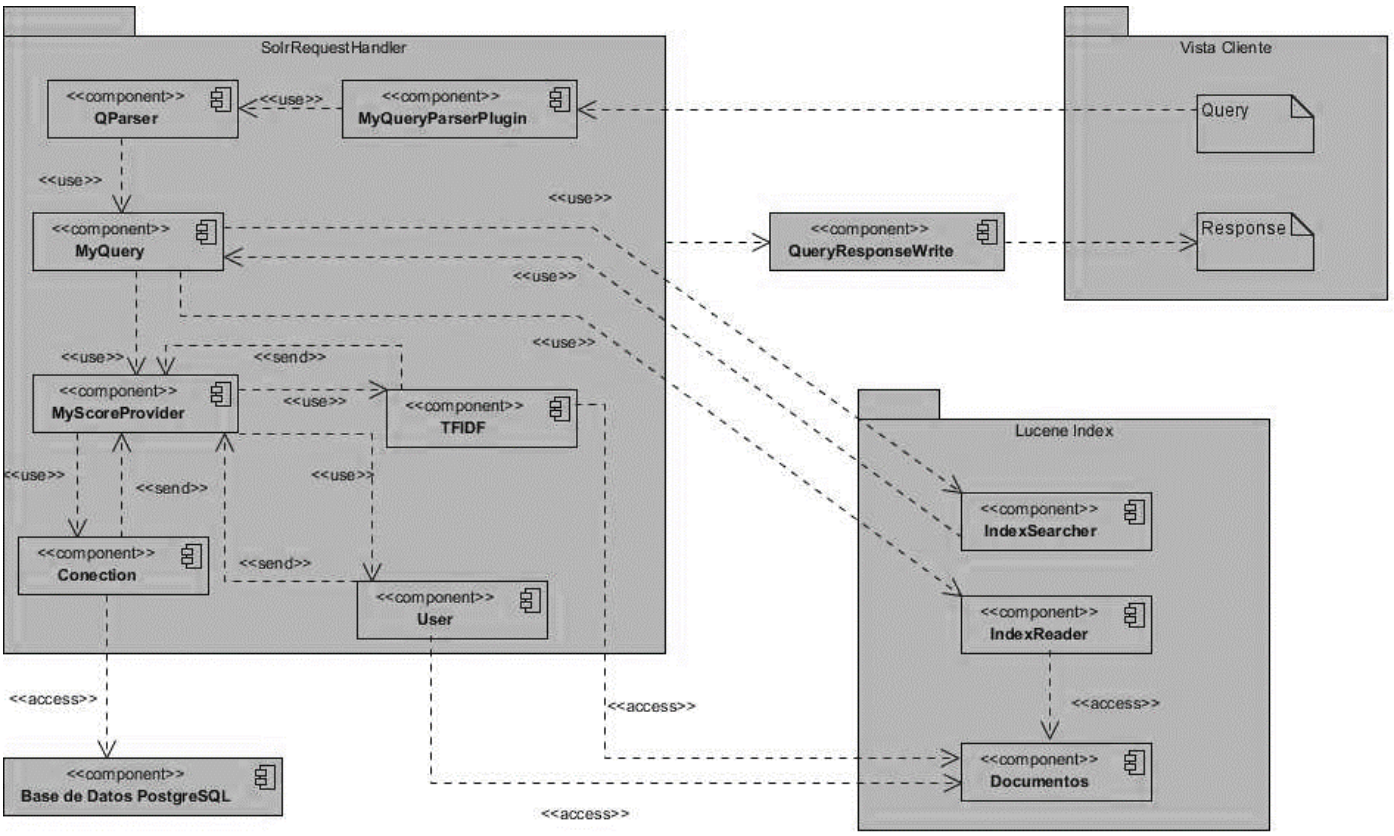

Figura 7 Diagrama de componentes (elaboración propia).

3.2 ESTÁNDARES DE CODIFICACIÓN UTILIZADOS

Los estándares de codificación son especificaciones que establecen la forma de generar el código funcional de las soluciones informáticas. La adopción de un estilo único de codificación constituye uno de los factores de mayor peso en el rendimiento, legibilidad, calidad y capacidad de mantenimiento del producto final.

Para proporcionar el entendimiento del código y establecer un modelo a seguir, se crearon estándares de codificación. Los aspectos para los que generalmente se establecen estándares son los siguientes:

Identificadores

En el caso de los identificadores existen estilos como el *lowerCamelCase* y el *UpperCamelCase*. Para ambos estilos cada palabra interna en identificadores compuestos comienza con mayúsculas y no se

colocan caracteres de separación entre las palabras que conforman un identificador compuesto en ninguno de los dos casos. Para el primer estilo los identificadores comienzan con minúscula y para el según comienzan con mayúscula.

Ejemplo de uso:

- Clase: public class MyQueryParsePlugin {}

  Estilo *UpperCamelCase.*

- Función: Public float TFIDF (int doc) {}

  Estilo *UpperCamelCase.*

- Variable: float porciento;

  Estilo *lowerCamelCase*

Indentación

En esta práctica se enfatiza en comenzar a escribir cada línea de código a diferentes distancias desde el borde izquierdo del área de edición. La distancia deberá regirse por la jerarquía que se forma al introducir sentencias dentro de bloques de estructuras. Con el uso de NetBeans IDE los espacios de indentación son ajustados automáticamente, permitiendo a los programadores enfocarse en otras funciones de mayor importancia. Para construir cierta homogeneidad y legibilidad, se escribirá cada sentencia en una línea de código, y en caso de ser necesario cortar las líneas, se hará luego de una coma o antes de un operador.

Llaves

Existen diversos criterios de la utilización de los operadores de delimitación desarrollados por los diferentes lenguajes de programación que lo usan para la definición de bloques de código. En el desarrollo de la implementación las llaves de apertura se colocarán inmediatamente al final de la línea de cabecera del bloque, así como en las estructuras if, for, while, else, foreach. Las llaves de cierre se colocarán solitarias en la línea siguiente a la última línea dentro del bloque y al nivel de la línea cabecera del bloque.

Líneas en blanco

En función de mejorar la legibilidad y organización del código se utilizan líneas en blanco para separar segmentos de código que pueden corresponder a clases, funciones, declaraciones, implementaciones, comentarios, bloques o sencillamente secciones críticas que se deseen despejar. En la implementación se

ha definido emplear líneas en blanco entre funciones de una misma clase y secciones de código dentro de una misma función.

Comentarios

Los comentarios en el código personifican la documentación interna más precisa de un software. Estos responden al entendimiento de lo que realmente realiza un determinado bloque de código, evitando confusiones y agilizando considerablemente las tareas de revisión y mantenimiento. Para inclusión de comentarios es necesario seguir algunas reglas básicas, entre las que se encuentran:

- Abreviar el contenido de los comentarios.
- Usar lenguaje técnico, entendible y no usar vocablos rebuscados.
- Emplear un estilo uniforme de comentarios estándares.
- Esquivar la descripción paso a paso de un bloque de código.

3.3 VALIDACIÓN DEL COMPONENTE IMPLEMENTADO

Terminada la implementación del producto es necesario realizarle pruebas al componente con el objetivo de detectar errores en la aplicación y la documentación; este proceso resulta de gran importancia, ya que proporciona una medida de la calidad del mismo siempre que se lleve a cabo de la forma correcta.

A continuación, se presentan los tipos de pruebas de software aplicados al componente implementado. Las mismas persiguen como objetivo fundamental, la detección de las no conformidades respecto a los requisitos funcionales de la aplicación, la correcta integración entre los componentes de la arquitectura, la aceptación del usuario y el rendimiento del sistema al utilizar el componente.

Pruebas funcionales

Las pruebas funcionales son aquellas que se aplican a un software determinado, con el propósito de validar que las funcionalidades implementadas funcionen de acuerdo a los requisitos registrados con anterioridad. Para la realización de dichas pruebas suelen emplearse dos métodos principales: el método de Caja Negra y el método de Caja Blanca. El primero centra su atención en el funcionamiento de la interfaz, mientras que el segundo permite a los probadores utilizar el código en las pruebas a las aplicaciones.

En este epígrafe se exponen los aspectos concernientes a las pruebas de Caja Negra a través de la entrada y salida de los datos. Para ello fueron creados los casos de prueba basados en las historias de usuario.

Caso de Pruebas para la HU "Asignar el nuevo valor de relevancia al documento".

- Descripción General

El usuario se registra en el sistema y es generado un PBU, de igual forma los documentos son almacenados en los índices de Solr con sus respectivas categorías. En este punto el componente se encarga de calcular el valor de relevancia del documento si la categoría del documento coincide con la categoría del PBU.

- Condiciones de ejecución

Los documentos que se corresponden con la consulta del usuario registrado han llegado a la función "customScore".

Tabla 6. Caso de prueba basado en la HU "Asignar nuevo valor de relevancia".

Escenario	**Descripción**	**Variable 1** **Categoría del documento**	**Variable 2** **Categoría del PBU**	**Respuesta del sistema**	**Flujo central**
EC 1.1 Asignar nuevo valor de relevancia.	Si la categoría del documento se corresponde con la categoría del PBU se valida el resto del procedimiento y el documento obtiene la puntuación original más la adición del PBU.	Categoría=deporte V	deporte=0,35 V	El sistema compara las variables, se asigna el nuevo valor de relevancia y se muestran los documentos ordenados.	1-El usuario realiza la consulta. 2-El sistema selecciona los documentos que se corresponden con la consulta. 3-El método "customScore" del componente, compara las variables 1 y 2. 4-El sistema asigna el nuevo valor de relevancia. 5-El sistema muestra en la interfaz los documentos ordenados por el nuevo valor de relevancia.
		Categoría=cultura V	deporte=0,35 V	El sistema compara las variables, no se asigna el nuevo valor de relevancia y se muestran los documentos ordenados.	
		Categoría=desconocida	cultura=0,55	El sistema	

		I	V	compara las variables, se mantiene el valor de relevancia por defecto y se muestran los documentos ordenados.	
		N/A	N/A	El sistema no compara las variables, asigna el valor de relevancia por defecto y muestra los resultados ordenados.	

Tabla 7. Variables empleadas en el caso de prueba basado en la HU "Asignar el nuevo valor de relevancia".

No	Nombre de campo	Clasificación	Valor nulo	Descripción
1	Categoría del documento	Campo de texto	si	Solo debe contener la categoría definida en el índice de Solr.
2	Categoría del PBU	Campo de texto	si	Solo debe contener la categoría definida en PostgreSQL.

Resultados de las pruebas funcionales

Para la detección de las no conformidades presentes en el componente desarrollado, se realizaron 4 iteraciones de pruebas funcionales. En la siguiente tabla se exponen los resultados obtenidos en cada iteración de prueba al componente de cálculo de relevancia a partir del perfil de usuario y la categoría de los documentos en el buscador Orión. De igual forma se muestra la corrección de cada uno de los errores.

Tabla 8. Cantidad de no conformidades por iteración de las pruebas funcionales.

No conformidades	1era iteración	2da iteración	3era iteración	4ta iteración
Detectadas	15	10	8	0
Resueltas	13	12	8	0
Pendientes	2	0	0	0

En la figura 8 se muestra de manera ilustrativa el comportamiento de las no conformidades por cada iteración de las pruebas funcionales ejecutadas.

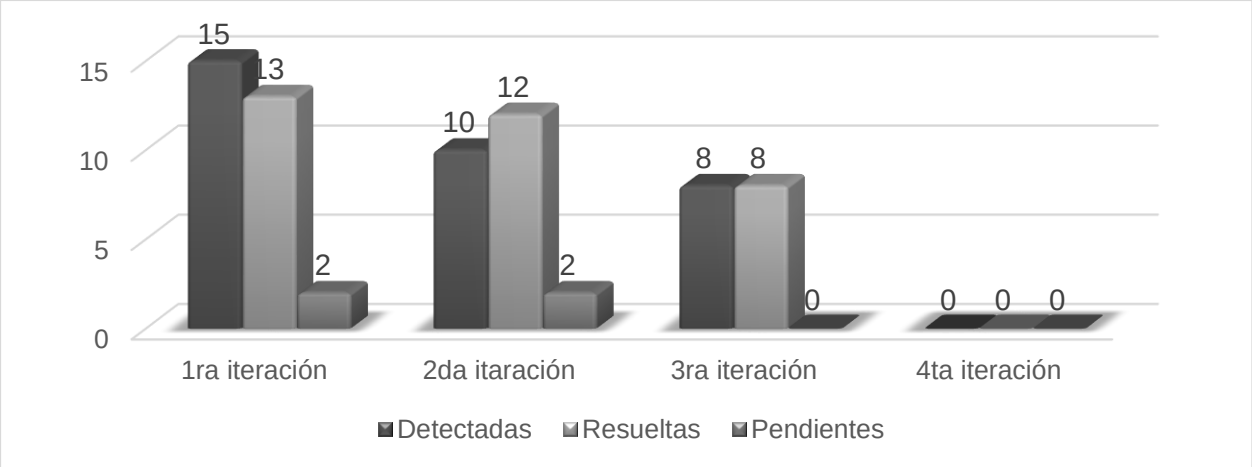

Figura 8 Comportamiento de las no conformidades por iteración (elaboración propia).

Las no conformidades fueron agrupadas en diferentes tipologías. En la figura 9 se representan las no conformidades detectadas por cada tipo.

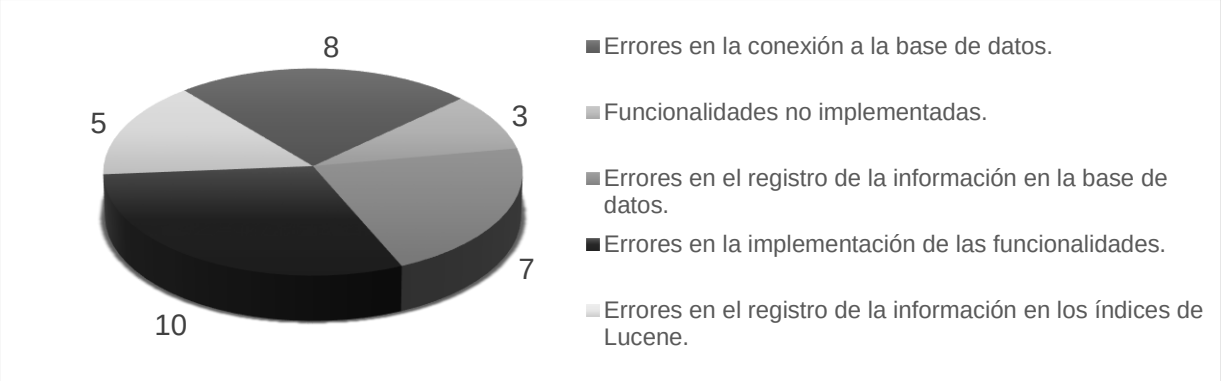

Figura 9 Cantidad de no conformidades detectadas por tipo (elaboración propia).

Pruebas de integración

Las pruebas de integración son ejecutadas para comprobar el correcto ensamblaje de los distintos componentes que conforman el sistema informático. Estas validan si estos componentes realmente funcionan juntos, son llamados correctamente y si transfieren los datos precisos por las vías de comunicación establecidas (Sommerville, 2005).

A partir de las pruebas funcionales realizadas al componente de cálculo de relevancia, fue posible determinar su correcto funcionamiento a través del flujo de datos. Mediante las pruebas de integración, se logró verificar la operación en conjunto de todos los componentes, junto a los componentes que conforman el modelo de implementación del buscador Orión.

Para la ejecución de dichas pruebas se llevaron a cabo diferentes acciones fundamentales:

- Comprobación del funcionamiento del enlace entre el motor de búsqueda Orión y el componente.
- Verificación de la conexión entre el sistema y la base de datos PostgreSQL.
- Correspondencia entre las instancias de Solr y la información almacenada en la base de datos.

De dicha prueba fue posible la detección de deficiencias en el manejo del objeto IndexSearcher en la clase TFIDF del componente, ya que no resultaba adecuado crear dos instancias de dicho tipo. Dicha deficiencia fue erradicada para la correcta integración del componente con el motor de búsqueda Solr.

3.4 PRUEBAS DE ACEPTACIÓN DE USUARIO

El objetivo de estas pruebas es validar mediante la Técnica de IADOV el grado de satisfacción de los usuarios con la implementación del sistema de indicadores para medir el impacto en los usuarios al utilizar el buscador Orión. El procedimiento para la selección de la muestra respecto a la población fue de tipo no probabilístico e intencional, o sea que permitió seleccionar directa y explícitamente los sujetos. En ese sentido fueron seleccionados los estudiantes de la brigada 1504 y algunos profesores del Centro de Ideoinformática.

El cuestionario empleado para determinar el grado de satisfacción cuenta con una pregunta cerrada. Esta pregunta cerrada utiliza el “Cuadro lógico de Iadov”, el cual se presenta adaptado a la actual investigación.

Tabla 9. Cuadro lógico de Iadov.

Pregunta:	Si	No sé	No
¿Se siente satisfecho con el resultado obtenido en la aplicación utilizando el componente de cálculo de relevancia?	12	0	0

La siguiente tabla indica un resumen de la posición de cada individuo en la escala de satisfacción. La escala de satisfacción es la siguiente:

Tabla 10. Relación de satisfacción individual en la escala.

Escala de satisfacción	Satisfacción individual	%
1- Clara satisfacción	10	83,00
2- Más satisfecho que insatisfecho	2	17,00
3- No definida	0	0
4- Más insatisfecho que satisfecho	0	0
5- Clara insatisfacción	0	0
6- Contradictoria	0	0

La satisfacción grupal se calcula mediante la siguiente fórmula**:**

$$\textbf{ISG} = A\,(+1) + B\,(+0{,}5) + C\,(0) + D\,(-0{,}5) + E\,(-1)/\,N$$

“En esta fórmula A, B, C, D, E, representan el número de sujetos con índice individual 1; 2; 3 ó 6; 4; 5 y donde N representa el número total de sujetos del grupo. El índice grupal arroja valores entre + 1 y - 1. Los valores que se encuentran comprendidos entre - 1 y - 0,5 indican insatisfacción; los comprendidos entre - 0,49 y + 0,49 evidencian contradicción y los que caen entre 0,5 y 1 indican que existe satisfacción” (Fabrel y otros, 2014).

Figura 10 Satisfacción individual de los estudiantes y profesores (elaboración propia).

El proceso de validación mediante la Técnica de IADOV de la consulta a los usuarios confirmó su factibilidad de uso, expresado cuantitativamente en el alto Índice de Satisfacción Grupal (**ISG**=0,92), lo que refleja aceptación de la propuesta de solución.

3.5 EVALUACIÓN DE LA CALIDAD DE LA RELEVANCIA

Con el propósito de validar el indicador referente a la calidad de la relevancia en el proceso de cálculo del componente, correspondiente a la variable dependiente definida en la hipótesis de investigación, se realizó un pre experimento para medir la exhaustividad y la precisión. En este se comparan los resultados obtenidos entre una primera iteración con la ejecución del sistema sin el componente y una segunda iteración con la incorporación del componente.

En la aplicación del pre experimento es necesario conocer las principales medias en los modelos clásicos de recuperación de información, los cuales se refieren a que los documentos pueden ser recuperados o rechazados al establecer la comparación entre la pregunta y la base de datos. En caso disponer de un conjunto de documentos recuperados se dividen en dos grupos: documentos relevantes recuperados, es decir aquellos que se han recuperados correctamente y los no relevantes, recuperados erróneamente que provocan ruido en la salida. De igual forma los documentos no recuperados se dividen en los relevantes, rechazados por el sistema de manera errónea y los no relevantes, rechazados de manera correcta por el sistema (Díaz, 2003).

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). En este sentido, existen dos métodos para calcular la relevancia, uno manual y otro conocido como *polling*:

- Manual: consiste en la exploración de los documentos uno a uno para saber si se adecúan o no como respuesta a una pregunta.
- *Polling*: Analiza de manera manual un número determinado de documentos recuperados con distintos sistemas y se corresponde con los primeros documentos recuperados con cada sistema. Este conjunto de documentos es el que de manera manual analizan los expertos, que son los encargados de decir en último término si son relevantes o no.

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:

Precisión = Documentos_relevantes_recuperados **/** Documentos_recuperados

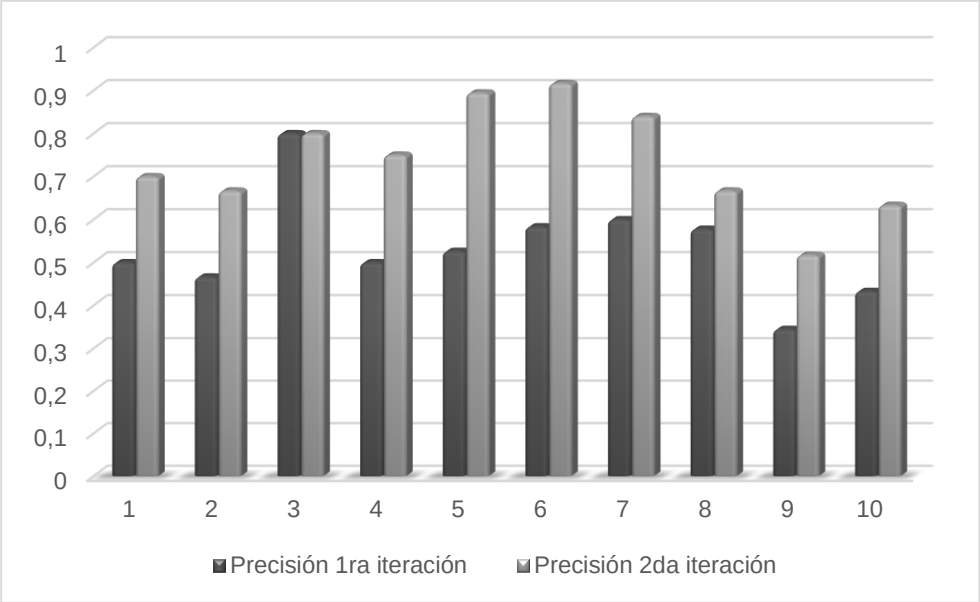

Figura 11. Precisión en la recuperación de información (elaboración propia).

Exhaustividad = Documentos_relevantes_recuperados **/** Documentos_relevantes

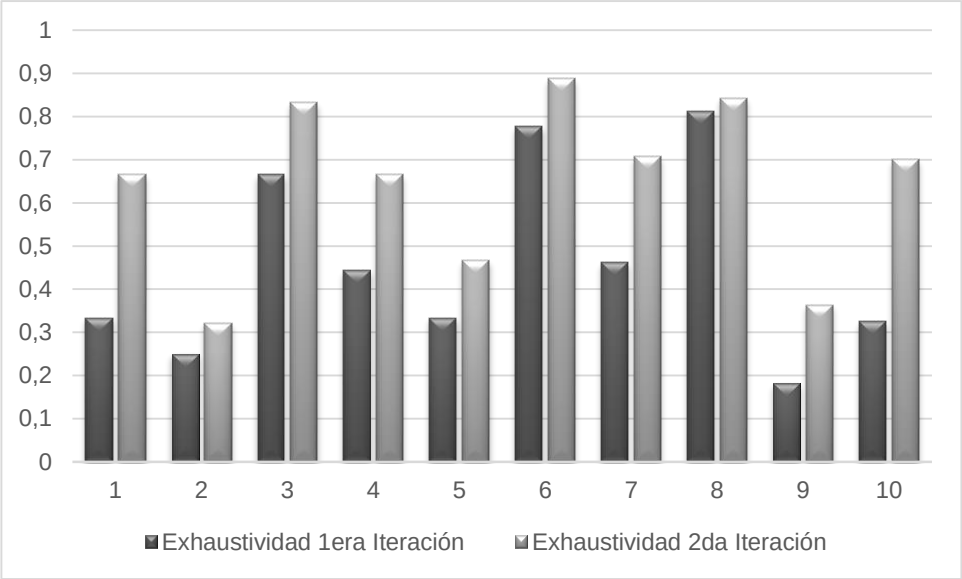

Figura 12 Exhaustividad en la recuperación de la información (elaboración propia).

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.

3.6 CONCLUSIONES PARCIALES DEL CAPÍTULO 3

Las pruebas realizadas al componente contribuyeron al impacto positivo del producto. En ese sentido, las deficiencias encontradas en el funcionamiento del sistema y las correcciones realizadas de las mismas, favorecieron a su calidad.

Para validar la hipótesis de la investigación se procedió a la aplicación un método pre experimental. En conclusión, se emplearon métricas que tributaron al conocimiento de elementos esenciales para aprobar la hipótesis, y con ello la factibilidad de uso del componente implementado.

Conclusiones

En el presente trabajo se ha llevado a cabo un proceso de desarrollo de software completo, separado en fases y flujo de trabajo, con el objetivo de lograr un producto de calidad en el tiempo establecido. 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.

Recomendaciones

Una vez concluida la investigación y el desarrollo de la propuesta solución, el autor del presente trabajo recomienda:

- Implementar funcionalidades para el cálculo de similitud semántica del documento.
- Añadir funcionalidades al componente con la incorporación de variables tales como: la usabilidad del contenido en el documento, la fiabilidad de la información y la importancia de la documentación devuelta en los resultados de una consulta.
- Unificar los estudios de inteligencia artificial y los procesos de recuperación de información en función de disponer de un componente adaptable a cualquier contexto.

Bibliografía

BBC News. Microsoft and Yahoo seal web deal. [En línea] 29 de julio de 2009. http://news.bbc.co.uk/2/hi/business/8174763.stm.

Abreu, A. P.; Pérez, H. A. A. *Editor de Consultas para el Sistema de Análisis y Modelado de Yacimientos Minerales (SYAM).* 2013. Trabajo de Diploma.

Afable, J. D. *A Beginner's Guide to Understanding Technical Support.* 2002. ISBN: 0595225748.

Aleman, Y.; Thomas, Y. J. *Módulo de configuración para el mecanismo de rastreo del buscador Orión.* La Habana : s.n., 2014. Trabajo de Diploma.

Anon. Definición de soporte técnico - Qué es, Significado y Concepto. Definiciones en linea. [En línea] 2015. http://definicion.de/soporte-tecnico/.

Apache Lucene. Class IndexReader. [En línea] 2017. https://lucene.apache.org/core/3_6_1/api/core/org/apache/lucene/index/IndexReader.html.

Apache Software Foundation. 2017. *Apache Solr Reference Guide. Covering Apache Solr 4.10.* 2017.

Baeza-Yates, R. *Modern Information Retrieval.* 1999.

Bon, J. V.; de Jong, A.; Kolthof, A.; Pieper, M.; Tjassing, R.; Van der Veen, A.; Verheijen, T. Operación del Servicio basada en ITIL® V3 – Guía de Gestión(spanish version). [En línea] 2008. https://books.google.com.cu/books?id=yUO5nvlJMFsC.. ISBN: 9789087531522.

Buscadores Web. Buscadores en Internet - Encuentra la información en Internet. [En línea] 2016. http://www.buscadores-web.com.

Corbalán, J. 2017. SEO en Bing. Guía de optimización SEO para Bing. [En línea] 2017. https://www.corbax.com/.

Delgado, Y. H. *Orion, un motor de búsqueda para la web de la UCI.* La Habana : s.n., 2010. Trabajo de Diploma.

Departamento de Lenguajes y Sistemas Informáticos. *Patrones de Asignación de Responsabilidades (GRASP).* Unisersidad de Sevilla. Sevilla : s.n., 2017. PDF.

Díaz, R. G. *La evaluación en recuperación de la información en línea.* Barcelona, Cataluña : Hipertext.net, 2003.

Díaz, Y. R.; González, M. Y. P. *Sistema de Información Geográfica para el transporte obrero de la Universidad de las Ciencias Informáticas. Versión 2.0.* 2012. Trabajo de Diploma.

Elizalde, E. Los buscadores más populares de Internet. [En línea] 25 de julio de 2016. http://buscadores.about.com.

Empresa Expertos en Servicios de Consultoría Exes, S.L. Área de programación y desarrollo. Manual de XML. [En línea] 2016. http://www.mundolinux.info/validacion-de-documentos-xml.htm.

Escofet, C. M. *El lenguaje SQL.* 2007. PDF.

Esteban, T. Visibilidad, tráfico y conversión. [En línea] 29 de enero de 2011. https://visibilidad-trafico-conversion.com/2011/01/29/webrank-algoritmoposicionamiento-yahoo/.

Fabrel, A. F.; Padrón, A. L. *Validación mediante criterio de usuarios del sistema de indicadores para prever, diseñar y medir el impacto en los proyectos de investigación del sector agropecuario.* La Habana : Revista Ciencias Técnicas Agropecuarias,, 2014. ISSN -1010-2760.

Fernández, A. B.; Herrera, Y. B. y Martínez, E. Y. V. *Propuesta de proceso de seleccion para el rol de programador.* 2009. Trabajo de Diploma.

Fernández, F. T. Educación Online. Instituto de Marketing Online. [En línea] 9 de abril de 2014. www.educaciononline.com/Instituto-de-marketingonline/palabras-clave-relevancia-y-algoritmos-de-google/.

Fonticoba, H. M. M. *Idoniedad de PostgreSQL en comparación con MongoDB. Caso Estudio: Módulo Publicidad del buscador cubano Orion.* CIDI-UCI. La Habana : Revista Cubana de Ciencias Informáticas, 2013. Artículo original.

Galindo, M. A. *Nuevas técnicas de búsqueda en Internet.* Universidad Complutense. Madrid : s.n., 2000.

García, S.; Martínez, F. G. *Extensión del NetBeans IDE para el diseño de Interfaces Gráficas de Usuario con Ext JS.* La Habana : s.n., 2013. Trabajo de Diploma.

Hawkins, T.; Plugge, E.; Membrey, P. *The Definitive Guide to MongoDB: The NoSQL Database for Cloud and Desktop Computing.* 2010. ISBN 9781430230519.

Itani, S. What algorithms does Bing use to rank search results? [En línea] 2 de enero de 2012. https://www.quora.com/.

Java Team. 2009. Conozca más sobre la tecnología Java. [En línea] 2009. http://www.java.com/es/about.

Kabir, M. J. *La Biblia del Servidor Apache 2 (Anaya Multimedia).* 2003. ISBN: 8441514682.

Lawrence, P. *Method for node ranking in a linked database. US6285999 B1* Estados Unidos de América, 9 de enero de 1998. Algoritmo PageRank.

León, M. V.; Serrano, R. L. *Portal Web de la Subsede Cubana de la Final Caribeña del ACM-ICPC.* La Habana : s.n., 2013. Trabajo de Diploma.

Leyva, P. R.; Sala, H. V.; Febles, J. P. *Algorithm for calculating relevance of documents in information retrieval systems.* s.l. : International Research Journal of Engineering and Technology (IRJET), 2017. Artículo científico.

Lucas, J. *Tutorial: Como instalar NGINX EN uBUNTU 14.04.* 2011. Tutorial.

Mahecha, I. A. N. Buscador web open-source: Nutch. [En línea] 2016. http://dis.unal.edu.co/profesores/eleon/cursos/tamd/presentaciones/nutch.pdf.

Marcos, M. C. Entrevista a Ricardo Baeza-Yates de Yahoo! [En línea] 2008. http://www.hipertext.net.

Márquez, J.; Sampedro, L.; Vargas, F. *Instalación y configuración de Apache, un servidor web gratis.* 2012. págs. 10-23. Vol. 012.

Marrero, Y. P. *Buscador de la Red Social de la Universidad de las Ciencias Informática.* La Habana : s.n., 2012. Trabajo de Diploma.

Martínez, R. Página Principal de Postgresql. [En línea] 2010. http://www.postgresql-es.org/principal.

Méndez, F. J. M. *Recuperación de información: modelos, sistemas y evaluación.* Universidad de Murcia. 2004.

Microsoft. Bing Product Guide. [En línea] 2017. http://www.bing.com.

Microsoft. *Microsoft's New Search at Bing.com Helps People Make Better Decisions: Decision Engine goes beyond search to help customers deal with information overload.* 2015.

Moleiro, L. J. P.; Gutierrez, A. A. S. *Herramienta informática de generación de reportes dinámicos basados en Apache Solr para el Sistema para repositorios digitales REPXOS 3.0.* La Habana : s.n., 2015. Trabajo de Diploma.

Nutch, Apache. Sitio Oficial de Apache Nutch. *The Apache Software Foundation.* [En línea] 2016. http://nutch.apache.org/index.html.

Oracle Corp. About MySQL. . [En línea] 2016. http://www.mysql.com/about/.

Ortiz, S. S. *Estrategia de soporte técnico para el proceso de migración a código abierto en los Organismos de la Administración Central del Estado.* Facultad 1. Centro de Software Libre. 2015. Trabajo final presentado en opción al título de Máster en Informática Aplicada.

Osiatis. ITIL® Foundation: Curso online gratuito. Gestión de servicios TI. [En línea] 2015. http://itilv3.osiatis.es/.

Pinto, M. Electronic Content Management Skills. . [En línea] 13 de diciembre de 2015. http://mariapinto.es/e-coms.

PostgreSQL tools. PgAdmin. [En línea] 2016. http://www.pgadmin.org.

Pressman, R. S. *Ingeniería de software.* s.l. : McGraw-Hill Interamericana de España, 2010. ISBN:9786071503145.

Pressman, R. *Ingeniería de Software.* 2008.

R&A Marketing. SEO y Análisis. [En línea] Plaza de Castilla, Madrid, 2017. http://www.ra-marketing.com/que-es-seo.aspx.

Rios, S. Manual de ITIL v3. [En línea] 2014. www.biable.es.

Rodríguez, Y. C. O.; Guerrero, S. P. *Módulo de traducción automática para el motor de búsqueda Orión.* La Habana : s.n., 2013. Trabajo de Diploma.

Román, J. V. *Sistemas de Recuperación de Información.* Ingeniería Sistemas Telemáticos, Universidad de Valladolid: Departamento. 1997.

Salton, G.; Wong, A.; Yang, C. S. *A Vector Space Model for Automatic Indexing. Communications of the ACM.* 1975.

Sánchez, T. R. *Metodología de desarrollo para la Actividad productiva de la UCI.* La Habana : s.n., 2015.

Sommerville, I. *Ingeniería de software.* s.l. : Pearson Educación, 2005. ISBN 84-7829-074-5.

Springer International Publishing AG. *Ingeniería de software basada en componentes.* Berlín, Alemania : s.n., 2017.

Suárez, J. M.; Gutiérrez, L. E.; Guerrero, C. A. *Patrones de Diseño GOF (The Gang of Four) en el contexto de Procesos de Desarrollo de Aplicaciones Orientadas a la Web.* Información Tecnológica, Grupo de

Investigación en Ingeniería del Software-GRIIS. Colombia : s.n., 2013. Artículo.

Tullis, S. Bingbot, the Sequel. [En línea] 29 de septiembre de 2010. https://blogs.bing.com/webmaster/2010/09/29/bingbot-the-sequel.

UPR. Lupa Buscador de la RedUniv. [En línea] Grupo de desarrollo de aplicaciones web y sistemas de la UPR, 2013. http://lupa.upr.edu.cu/acerca-de.html.

Visual Paradigm. Visual Paradigm for UML. [En línea] 2016. https://www.visualparadigm.com/.