Universidad de las Ciencias Informáticas

Facultad 1

Subsistema de estandarización de documentos para el buscador Orión.

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autora: Anietsy Perera Pérez

Tutor(es): MsC. Eylín Hernández Luque

Ing. Odisleysi Martínez Furones

Ing. Paúl Rodríguez Leyva

La Habana. Junio, 2017

"Año 59 de la Revolución"

Declaración de autoría

Declaro por este medio que yo: Anietsy Perera Pérez, con número de identidad 94092025735, soy la autora principal del trabajo final de tesis de pregrado que se titula: “Subsistema de estandarización de documentos para el buscador Orión”. El cual ha sido desarrollado como parte del trabajo del Centro de Ideoinformática de la facultad 1 y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Y para que así conste, firmo la presente declaración jurada de autoría en La Habana a los___días del mes de___del año ___.

_____

Anietsy Perera Pérez

Pensamiento

"Somos algo más que nosotros mismos, somos pueblo, somos una idea, somos una esperanza, un ejemplo".

Fidel Castro

Agradecimientos

En especial a mis padres por darme todo ese amor, cariño y educación que me han brindado, sin ellos no hubiese podido ser la mujer que soy hoy. Gracias mami y papi.

A mi hermana por su apoyo incondicional durante la carrera. Gracias tata.

A mi novio por ser mi compañero, mi amigo, mi guía, por estar siempre a mi lado en momentos de alegría y tristeza.

A mi familia en general, mi abuela, mi tía, prima y mi tío, que, aunque estén lejos siempre me están apoyando.

A mis tutores Eylín, Odisleysi y Paúl por ser mi guía en el transcurso de este trabajo, por dedicarme parte de su tiempo, que sé que han estado muy complicado, pero a pesar de eso, han sido los mejores, me han brindado mucha seguridad y confianza. En especial a Paúl por haber tenido tanta paciencia conmigo, por siempre estar cuando lo he necesitado y aunque al principio no era tutor mío oficialmente, siempre lo estaba molestando. Gracias por apoyarme,

En general a mis profesores que contribuyeron en mi preparación para la vida.

A mis compañeros de aula, junto a ellos he pasado una de las etapas más bonita de mi vida.

A mi amiga Anabel por su apoyo, preocupación que siempre ha tenido, gracias por estar en esos momentos importantes en esta universidad.

A mi compañera de cuarto Fany por siempre estar presente cuando la he necesitado, mejor flaca que esa no la quiero.

A mis compañeras de apartamento Zule, Yoli y Rache por haber compartido todos esos momentos alegres en el transcurso de esta carrera. No las voy a olvidar.

A Lexys por brindarme su ayuda en los 1eros años de mi carrera, ha sido como mi profesor.

A Ileana y Duanny que, aunque los conocí tarde son dos grandes personas que me han ayudado muchísimo.

De manera general gracias a todas esas personas que de una forma u otra han permitido que esto fuera posible.

Dedicatoria

Le dedico el trabajo de diploma a mis padres Olidia y Pedro por ser las personas que más quiero en el mundo, por ser la luz de mi vida y mi corazón, por siempre estar a mi lado apoyándome y dándome esas fuerzas que he necesitado para seguir adelante. Ellos son la razón de mi vida. Los quiero.

A mi hermana Yaneisy que la quiero con la vida, ella siempre ha estado a mi lado dándome consejos y protegiéndome.

A mi abuela Margarita que es como si fuera mi segunda mamá, la quiero muchote, ella siempre me transmite su energía, alegría y me da muchas fuerzas para seguir adelante. Te quiero mima.

A mi novio Félix Daniel por siempre estar apoyándome en los buenos y malos momentos, por tener tanta paciencia conmigo y por enseñarme a mirar el lado positivo de las cosas y nunca darme por vencida.

Resumen

Internet constituye un importante medio de información usualmente recurrido por un gran número de usuarios, debido a la masividad y heterogeneidad de los contenidos que enlaza. El crecimiento acelerado de los datos provoca que los usuarios afronten dificultades para encontrar la información que realmente necesitan. Para contribuir a erradicar esta problemática se diseñaron los Sistemas de Recuperación de Información que permiten mostrar al usuario los contenidos más relevantes, mediante técnicas de rastreo, indexado y visualización de información. En la Universidad de las Ciencias Informáticas se desarrolló el buscador Orión, este presenta dificultades en el desempeño óptimo del proceso de estandarización de documentos. La investigación se centró en el desarrollo de un subsistema de estandarización de documentos para el buscador Orión. La propuesta de solución estuvo guiada por la metodología AUP-UCI y la modelación se realizó en la herramienta Visual Paradigm generando los artefactos afines a cada etapa del desarrollo. Durante la implementación se emplearon las tecnologías: Nutch como mecanismo para rastrear la Web, Solr como servidor de indexación, JSON como formato de intercambio de datos, XML como lenguaje para los ficheros de configuración en Solr y Java como lenguaje de programación para la implementación del plugin en Nutch. Para la validación del sistema se realizaron pruebas funcionales, de integración y aceptación, las cuales arrojaron un conjunto de no conformidades que fueron resueltas en iteraciones posteriores. El subsistema implementado presenta funcionalidades que favorecen la calidad de las búsquedas, mediante la estandarización de documentos en la integración de Nutch y Solr.

Palabras clave: estandarización de documentos, búsquedas, buscador Orión, Sistemas de Recuperación de Información.

Índice

Índice de Tablas

Índice de Figuras

Introducción

El desarrollo científico-tecnológico, ha generado un incremento y perfeccionamiento acelerado de las Tecnologías de la Información y la Comunicación (TIC). En la práctica el resultado más importante es Internet, que se ha convertido en un medio imprescindible para el desarrollo del mundo. Cuando se necesita realizar una búsqueda, por trivial que resulte la temática que se desea conocer, se piensa primero en Internet y no en una biblioteca tradicional. De manera que, con el crecimiento de internet se han desarrollado y perfeccionado los motores de búsqueda, dirigidos a facilitar la navegación y el hallazgo de la información.

“Los buscadores, motores de navegación o motores de búsqueda (MB), son aquellos programas o herramientas interactivas que facilitan la búsqueda y recuperación de información. Los motores de búsqueda ofrecen formularios para introducir los datos mediante una interfaz de fácil comprensión para el usuario, el cual teclea una palabra clave o frase y recupera una lista de recursos que se corresponden con el criterio indicado” (Camiño, 2003).

Según estudios realizados por Lluís Codina, máster en Comunicación Social de la Universidad Pompeu Fabra, la mayor parte de las nuevas visitas a un sitio determinado se producen por una consulta realizada a un buscador, lo cual ocasiona que gran parte del tráfico de un sitio web pueda proceder de los motores de búsqueda. El número de visitas que recibe un sitio no depende únicamente de las bases de datos a las cuales se encuentre indexadas, sino que uno de los principales factores que influyen en este sentido es el posicionamiento web que el sitio en cuestión ocupe ya que solo una pequeña parte los usuarios llega a interesarse por la segunda página de los resultados ofrecidos por un buscador (Codina, 2008). Además, un alto porciento de estos usuarios, detienen su búsqueda en la tercera página. A partir de estos datos se puede inferir en alguna medida la importancia que tiene el posicionamiento de un sitio web.

El desarrollo del dominio.cu continúa su ritmo creciente. Según el sitio web oficial del Centro Cubano de Información de Red (CUBANIC) hasta el 26 de febrero del 2017 se contaba aproximadamente con unos 6694 dominios registrados bajo .cu y con ello una amplia gama de información disponible, de ahí la importancia de realizar una búsqueda eficiente de los datos que se desea analizar.

La Universidad de las Ciencias Informáticas (UCI), creada en el 2002 en el marco de la Batalla de Ideas y como parte de uno de los programas de la Revolución para la educación superior y la informatización de la sociedad cubana, cuenta con una infraestructura tecnológica privilegiada entre las demás instituciones nacionales, conectando en su red informática interna una gran cantidad de computadoras. Producto al

desarrollo tecnológico que poseen la Universidad y los centros de investigación y desarrollo de software que radican dentro de la misma, se llevan a cabo diversos trabajos de investigación y eventos de corte docente y científico que generan gran cantidad de documentación.

En el Centro de Ideoinformática (CIDI), perteneciente a la Facultad 1 de la UCI, se desarrolla el buscador Orión, el cual se encuentra en estos momentos desplegado en los servidores de la UCI y en el Ministerio de Educación Superior (MES). Este sistema, brinda servicios de búsqueda y recuperación de información, permitiendo a los usuarios un rápido acceso a los recursos publicados en la intranet de la universidad, bajo el dominio “uci.cu”.

El buscador Orión pone en las manos de todos los cubanos una herramienta de búsqueda de información desarrollada bajo tecnologías del software libre que respeta las buenas prácticas de posicionamiento, garantiza resultados legítimos del país y añade una victoria más en el logro de la soberanía tecnológica de Cuba. Brinda un servicio de alta calidad, los centros educacionales tendrán una herramienta más de apoyo a la formación de los estudiantes, contribuyendo al proceso de enseñanza-aprendizaje que lleva a cabo el Ministerio de Educación para cumplir con el objetivo de dotar al país de una sociedad culta e innovadora (Leyva, 2016).

Orión cuenta con tres mecanismos principales, que permiten acceder a los documentos publicados bajo el dominio “uci.cu”: interfaz web de consulta para realizar las búsquedas, Solr[1] como servidor de indexación de contenidos y Nutch[2] como *spider* o araña, este último constituye el mecanismo de rastreo encargado de analizar todos los documentos publicados.

Durante el proceso de rastreo que realiza Nutch, se recepcionan muchos documentos en los cuales se repiten metadatos como el título, contenido, formato y url, esto ocasiona que se dificulte el proceso de recuperar información válida y sin repeticiones, haciendo recibir a los usuarios más información de la necesaria para responder a sus criterios de búsquedas. En el proceso de estructuración de los documentos los metadatos guardados no tienen prioridad, es decir cuando se rastrea los documentos no se verifica la completitud de los metadatos del mismo y en ocasiones se almacenan documentos incompletos que no aportan ningún valor a la colección o se insertan metadatos en los documentos que no son procesados ni

[1] Solr: Es un servidor de índice empresarial, que actúa como una base de datos no SQL. Una plataforma de búsqueda de código que funciona como un "servidor de búsquedas" (Apache Solr 2015).
[2] Nutch: Es un programa de código abierto libre diseñado especialmente para realizar las tareas de robot de búsqueda, esto es para rastrear la web recuperando las páginas que componen la red a través de la estructura de enlaces.

utilizados en el Sistema de Recuperación de Información (SRI). Todo esto provoca que exista un uso ineficiente del almacenamiento y que se guarden metadatos innecesarios que no aportan ningún conocimiento para el usuario, ni para el buscador. Por otra parte, los usuarios no se sienten satisfechos con los resultados que les brinda la herramienta como respuesta a sus consultas, esto ha traído como consecuencia que muchos internautas dejen de usar el motor de búsqueda Orión. La situación descrita provoca la pérdida de interés de los usuarios del buscador Orión. Por lo tanto, la actual versión de Orión necesita un proceso de estandarización de documentos en la recopilación y almacenamiento de los resultados que indexa.

Partiendo de esta situación se plantea el siguiente **problema científico:** ¿Cómo estandarizar los documentos que se gestionan en el buscador Orión, de manera que mejore la calidad de los documentos almacenados?

Para solucionar el problema planteado, se determinó que el **objeto de estudio** se centra en el proceso de recuperación de la información enmarcado en el **campo de acción** estandarización de documentos en el Sistema de Recuperación de Información Orión.

Se define como **objetivo general**: Desarrollar un subsistema de estandarización de los documentos que se gestionan en el buscador Orión, que mejore la calidad de los documentos almacenados.

Para dar cumplimiento al objetivo general, se han trazado los siguientes **objetivos específicos**:

1. Caracterizar los fundamentos teóricos relacionados con el procesamiento y estandarización de documentos.
2. Definir las tecnologías, las herramientas y la metodología del subsistema para el procesamiento y estandarización de documentos en el buscador Orión.
3. Desarrollar el subsistema para el procesamiento y estandarización de documentos en el buscador Orión.
4. Validar el correcto funcionamiento de la solución propuesta subsistema de estandarización de documentos para el buscador Orión.

Hipótesis: Si se desarrolla un subsistema de estandarización de documentos que se gestionan en el buscador Orión, se mejorará la calidad de los documentos almacenados.

Variable independiente: Componente de estandarización de documentos.
Variable dependiente: Calidad de los documentos almacenados.

Para dar cumplimiento a los objetivos específicos se han trazado las siguientes **tareas de investigación:**

1. Caracterización de subsistemas de estandarización de documentos para la elaboración del marco teórico –conceptual de la investigación.
2. Fundamentación de metodologías, herramientas y tecnologías para el desarrollo de la solución propuesta.
3. Especificación de las funcionalidades que debe poseer el subsistema de estandarización de documentos para el buscador Orión".
4. Definición del modelo conceptual referente a la estandarización de los documentos procesados para el buscador Orión.
5. Elaboración del diseño del subsistema con el empleo de la metodología de desarrollo seleccionada y patrones de diseño.
6. Implementación y realización de pruebas del subsistema de estandarización de documentos para el buscador Orión.

Para el desarrollo de las tareas de investigación se han combinado diferentes métodos teóricos y empíricos en la búsqueda y procesamiento de la información. Estos son:

Métodos Teóricos

Analítico-Sintético: se empleó para explorar los referentes teóricos sobre la estandarización de los documentos, así como para analizar los elementos conceptuales asociados al problema de investigación, que posibilita la extracción de los aspectos más significativos que sustentan la investigación.

Histórico-Lógico: se empleó para constatar cómo surgieron y han evolucionado los motores de búsqueda, así como para determinar las tendencias actuales de los mismos en el contexto internacional. Además, permitió identificar las principales ventajas e inconvenientes de las soluciones homólogas, así como de los avances obtenidos en torno a esta materia, de esta forma viabilizó la definición de las herramientas, técnicas y tecnologías a apropiadas para la implementación de la propuesta de solución.

Modelado: se usó en la elaboración de los artefactos afines a la metodología de desarrollo, tales como: diagramas de componentes, diagrama de despliegue y diagrama de modelo de datos.

Métodos Empíricos

Observación: se utilizó para dar seguimiento a la evolución de la investigación, detectar dilaciones o fallas en el cumplimiento del cronograma de trabajo y reunir la información de los indicadores de las variables que figuran en el problema.

La estructura del presente Trabajo de Diploma se compone por tres capítulos que sintetizan los resultados de cada etapa de la investigación como a continuación se enuncia:

Capítulo 1: Fundamentación teórica del Subsistema de estandarización de documentos para el buscador Orión. Se sistematizan conceptos fundamentales para el entendimiento de la investigación, se incluye el estudio de homólogos, así como la realización de las tendencias actuales y principales autores relacionados con los SRI. Además, se define la metodología de desarrollo de software, los lenguajes de programación y de modelado, así como las herramientas y tecnologías apropiadas para la implementación de la solución.

Capítulo 2: Exploración, planificación y diseño del Subsistema de estandarización de documentos para el buscador Orión. Se exponen las características del subsistema, incluyendo los requisitos funcionales y no funcionales, los patrones de diseño y arquitectura utilizada. Se presentan los artefactos correspondientes a la metodología de desarrollo utilizada.

Capítulo 3: Implementación y pruebas del Subsistema de estandarización de documentos para el buscador Orión. Se exponen algunos aspectos asociados a la implementación de la solución, así como los componentes que la integran. Además, se presentan los diseños de casos de prueba a utilizar en la validación del sistema y se analizan los resultados de las pruebas realizadas que permiten evaluar la calidad de la propuesta de solución.

Capítulo 1: Fundamentación teórica del Subsistema de estandarización de documentos para el buscador Orión.

1.1 Introducción.

El presente capítulo tiene como objetivo analizar los conceptos y aspectos principales relacionados con la estandarización de documentos: estándar, estandarización de documentos, estándares de metadatos, recuperación de información, SRI, buscadores y el estudio de sistemas homólogos. Se definen las tecnologías, lenguajes y herramientas utilizadas en la implementación de la propuesta de solución.

1.2 Fundamentos teóricos asociados al dominio del problema

Para lograr una mejor comprensión de la investigación, se abordan un conjunto de conceptos necesarios que están estrechamente relacionados con el dominio del problema.

1.2.1 La estandarización y sus aspectos generales

Una de las principales herramientas informáticas que han permitido poder interactuar y así proporcionar una experiencia satisfactoria al usuario han sido los **estándares**. Estos son acuerdos (normas) documentados que contienen especificaciones técnicas u otros criterios precisos para ser usados como reglas, o definiciones de características para asegurar que los procesos y servicios se ajusten a su propósito (Hernández, 2011).

La creación de un estándar web requiere un proceso controlado, que consta de varias etapas que aseguran la calidad de la especificación. Este proceso permite la intervención de todos los usuarios de las tecnologías, con el objetivo de que puedan aportar su conocimiento y opiniones para la mejora de los documentos. Los sitios web se implementan atendiendo a estándares abiertos y tecnologías robustas, seguras y muy estables como: Linux, Apache, Mysql y PHP.

Estos estándares han permitido que los sitios web incorporen aspectos para facilitar la navegación sin discriminar a personas con limitaciones visuales, cognitivas y auditivas; además están diseñados de una manera fácil y lógica para navegarlos, de manera uniforme entre todas las instituciones de gobierno, lo que se resume en sitios web rápidos, claros y fáciles de utilizar (Enrique, 2013).

La estandarización de documentos es un proceso de búsqueda de patrones de equilibrio y unificación de las características de un producto o servicio. Consiste en elaborar, aplicar y mejorar los estándares que se aplican a distintas actividades científicas con el objetivo de mejorarlas. Garantizando la calidad del elemento

creado, la seguridad en cuanto a funcionamiento y la responsabilidad social en cuanto a las normas de trabajo.

La estandarización de documentos persigue fundamentalmente tres **objetivos**: simplificación, se trata de reducir los modelos para quedarse únicamente con los más necesarios; la unificación para permitir el intercambio a nivel internacional y la especificación para evitar errores de identificación creando un lenguaje claro y preciso (Diéguez, 2012).

De acuerdo con lo anteriormente expuesto, se asume en este estudio que la estandarización de documentos es el proceso que permite definir de manera óptima y única la estructura que deben mantener los datos almacenados por el servidor de indexación que utiliza el motor de búsqueda.

Cuando internet empezó a crecer y la cantidad de información disponible aumentó, surgió el problema de clasificarla e identificarla. Partiendo de ese problema, se comenzaron a usar los **metadatos**, que se definen como “datos acerca de los datos". Describen el contenido, la calidad, el formato y otras características asociadas al recurso que contiene estos datos, constituyendo un mecanismo para caracterizar datos y servicios de forma que usuarios y aplicaciones puedan localizarlos y acceder a ellos. Dan respuestas a preguntas del tipo (Maganto, 2008):

El qué: nombre y descripción del recurso
El cuándo: fecha de creación de los datos, períodos de actualización, etc.
El quién: creador de los datos
El dónde: extensión geográfica
El cómo: modo de obtención de la información, formato, etc.

Los estándares de metadatos se usan principalmente para la recuperación de información y para describir y catalogar documentos. Otros usos incluyen la definición de derechos de propiedad intelectual, valoración y evaluación de contenidos.

1.2.2 Recuperación de información

A lo largo de los años el concepto de Recuperación de Información (RI), ha sido analizado por varios autores, entre ellos Baeza-Yates y Ribeiro-Neto (1999), Pérez-Carballo y Strzalkowski (2000), Sánchez Jiménez R. (2011); debido a que se trata de un campo que ha involucrado a investigadores y estudiosos de diferentes ciencias como: La Informática, Ciencias de la Información, Documentación y Lingüística.

Baeza-Yates y Ribeiro-Neto (1999), plantean que “la Recuperación de Información trata con la representación, el almacenamiento, la organización y el acceso a ítems de información”. Sin embargo, un año después, Pérez-Carballo y Strzalkowski (2000) indican que “la Recuperación de Información es traer documentos relevantes desde un gran archivo en respuesta a una pregunta formulada”.

Aunque se evidencia con claridad que existe determinada variación en las definiciones de los clásicos, respecto a la evolución del término RI en el tiempo, se puede constatar que existe gran similitud con la definición que plantea años más tarde Sánchez Jiménez (2011), la RI se centra en la representación, almacenamiento, organización y acceso a elementos de información. Estos procesos deberían proporcionar al usuario la capacidad de acceder a la información que necesita.

De acuerdo con la definición anterior, a los efectos de esta investigación en lo adelante se considera la RI como la presentación de información relevante desde una gran colección de documentos, a un usuario que hace una petición sobre la base del lenguaje natural, mediante la utilización de métodos, técnicas y herramientas de búsqueda que logren satisfacer las necesidades de información de los usuarios.

La diversidad de información poco estructurada y dispersa disponible en la web, dificulta el proceso de encontrar información útil que responda a las necesidades de los usuarios. **Los SRI** son los encargados de buscar información dentro de un grupo de documentos que han sido indexados previamente. Estos permiten localizar y procesar cualquier contenido existente, tales como textos, imágenes, videos y archivos de sonido.

De acuerdo con Baeza–Yates (2005) los SRI “deben de alguna manera interpretar el contenido de la información dentro de una colección de documentos y establecer con ellos, un orden de acuerdo al grado de relevancia que estos posean para las consultas de los usuarios”.

Por lo general los SRI comparten una misma arquitectura, la cual se describe a continuación (López, 2006):

Interfaz: Parte de un SRI mediante el cual un usuario introduce los criterios de búsqueda y obtiene un conjunto de resultados. La misma puede ser una interfaz web, una interfaz de escritorio o ambas.

Sistema de formulación de consultas: Realiza un pre-procesamiento de las consultas trasladando las consultas hechas en lenguaje natural a consultas entendibles por los sistemas de información.

Mecanismo de evaluación de consultas: Compara los documentos representados en el sistema de información con la consulta pre-procesada, para obtener un subconjunto de documentos relevantes que

satisfagan la consulta introducida por el usuario, ordenados estos de acuerdo a un criterio de relevancia.

Los SRI, teniendo en cuenta la forma de operar, el alcance que poseen, los tipos de documentos que recuperan, se clasifican principalmente en directorios, metabuscadores y buscadores, constituyendo esta última la más idónea para la implementación de sistemas autónomos dedicados a la recopilación, procesamiento y recuperación de grandes volúmenes de información.

1.2.3 Buscadores

Los buscadores o motores de búsqueda son SRI que permiten obtener aquellos documentos de mayor relevancia, a partir de un criterio de búsqueda introducido por el usuario. Cuentan con un subsistema denominado *spider* que recorre la web de forma automática y almacena la información recopilada indexándola en una base de datos. Estos buscadores se han convertido hoy en día en una herramienta primordial para obtener información en internet.

Un motor de búsqueda está compuesto por: un sistema de exploración o rastreador, un sistema de recuperación de información y una interfaz web que es la parte que el usuario es capaz de ver y con la que puede interactuar.

El rastreador está diseñado para navegar por la WWW[3] a partir de un listado de direcciones de manera sistemática y organizada, su función es analizar el contenido de las páginas web que encuentra. Descarga esos contenidos, en su mayoría páginas web, aunque también otros como presentaciones, archivos de imágenes, audio, etc. en el almacén o repositorio de documentos del motor de búsqueda. A partir de las páginas visitadas, el rastreador obtiene nuevas URL[4] que añade a su lista de recursos a analizar (Cardelas, 2013).

Un **metabuscador** es un *software* que agrega los resultados de varios motores o directorios para encontrar las páginas más relevantes. Son herramientas de búsqueda que actúan sobre algunos de los buscadores de información general más conocidos. Así dirigen las consultas a otros buscadores y presentan la totalidad de respuestas obtenidas. Es por ello que se usan para aquellas búsquedas en las que es interesante obtener el máximo de recursos disponibles en la red.

[3] Siglas del idioma inglés correspondiente a *Word Wide Web*: Es un sistema hipermedia que integra en una interface común a todos los recursos existentes en la red para su acceso en forma organizada y normalizada, cubre todos los recursos del mundo en forma hipermedia o hipertexto. (Andalia, 2004).
[4] Siglas del idioma inglés correspondiente a *Uniform Resource Locator* (Localizador Uniforme de Recursos): Se trata de la secuencia de caracteres que sigue un estándar y que permite denominar recursos dentro del entorno de Internet para que puedan ser localizados. (Castro, 2015).

El objetivo de los metabuscadores es facilitar al máximo y hacer más eficiente la búsqueda de información. Esto lo hacen buscando en múltiples sitios y entregando un resumen de los artículos relevantes acerca del tema que se solicitó. La diferencia entre ellos es la forma de buscar, los lugares en que buscan y como presentan la información (Hernández, 2011).

Características de los metabuscadores:

- No tienen una base de datos propia.
- Su objetivo se basa en la optimización de tiempos de repuesta.
- Existe cierta incertidumbre sobre sus métodos de combinación de buscadores, obtención de pesos, obtención del orden de los resultados, etc…

Tipos de metabuscadores:

- **Metabuscadores:** Agregan los resultados de varios motores o directorios para encontrar las páginas más relevantes.
- **Multibuscadores:** No combinan los resultados, sólo lanzan la consulta en varios buscadores.
- **Agentes de búsqueda:** Son metabuscadores instalados localmente.

1.2.4 Arquitectura de los motores de búsqueda

Los motores de búsqueda deben tener una arquitectura que les permita soportar manejar enormes volúmenes de datos y soportar millones de consultas diarias, manteniendo un tiempo de respuesta que haga aceptable la realización de búsquedas por parte del usuario y, por otro lado, que haga posible la constante actualización de la información que se tiene almacenada. Por estos motivos es que los motores de búsqueda generalmente son sistemas distribuidos (Wainerman, 2001), de los cuales se pueden distinguir varios componentes como se muestra en la siguiente imagen:

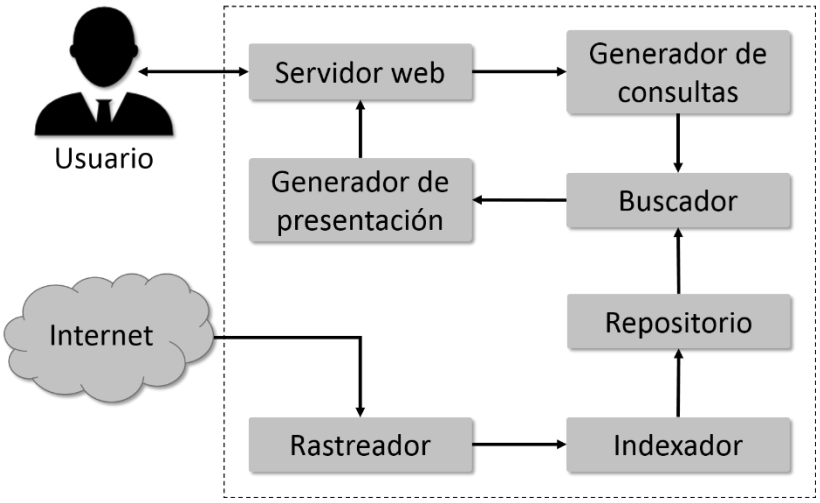

Figura 1. Arquitectura genérica de un motor de búsqueda *(Wainerman, 2001)*.

Rastreador: Es el componente encargado del rastreo.

Indexador: Almacena los datos que recolecta el rastreador dentro de una estructura ordenada. Es el componente encargado del proceso de indexación.

Repositorio: Almacena la información utilizada para generar las respuestas a las solicitudes de los usuarios. La estructura, datos y forma en que se maneja este componente varía en gran medida de acuerdo a la implementación específica del motor de búsqueda.

Servidor web: Este componente es el encargado de establecer la comunicación con el usuario a través del protocolo HTTP[5] , recibiendo consultas de éste y enviándole el resultado de las mismas a través del mismo protocolo.

Generador de consultas: Recibe la consulta realizada por el usuario en forma de una cadena de texto, la interpreta y genera una consulta que pueda ser ejecutada dentro del repositorio.

Buscador: Es el componente encargado de acceder a la estructura interna de los datos (repositorios, índices, etc.), para satisfacer el pedido.

Generador de presentación: Construye la vista a ser incluida en el documento HTML que recibe el usuario a través del servidor web, a partir de los datos que genera el buscador.

[5] Acrónimo de *Hypertext Transfer Protocol*. Según *World Wide Web* Consortium (W3C, principal organización de estándares internacionales de la WWW) e Internet *Engineering Task Force* (IETF, organización dedicada al desarrollo y promoción de normas que conforman el conjunto de protocolos de Internet) es un protocolo utilizado para distribuir información hipermedia.

1.3 Sistemas homólogos

En la actualidad existen numerosos sistemas de recuperación de información. Para identificar ventajas en el uso de estos sistemas y lograr una mejor comprensión de las características y funcionalidades de la propuesta de solución, se hace necesario realizar un estudio de algunos de estos motores de búsqueda. A continuación, se expone el estudio realizado de los sistemas homólogos, tanto en el ámbito nacional como internacional.

1.3.1 Sistemas homólogos a nivel internacional

Google[6]

El motor de búsqueda Google, persigue como objetivo que sus usuarios encuentren la información que necesitan y consigan hacerlo de la forma más sencilla y rápida posible. Ofrece servicios como búsqueda de imágenes, libros, noticias, videos, documentos, entre otros. La búsqueda avanzada de documentos en Google, incluye aspectos comunes con el resto de las búsquedas personalizadas que se ofrecen; como lo son, los criterios para encontrar todas, cualquiera o ninguna de las palabras de una frase o incluso expresiones exactas. Dichos criterios, también se pueden incluir en el cuadro de búsqueda mediante el uso de operadores especiales[7] definidos por el propio buscador (Alcaide, 2014).

Específicamente para la búsqueda avanzada de documentos se ofrecen filtros que permiten obtener resultados específicos por idioma, región, última actualización, sitio o dominio donde se encuentra el contenido, términos en que aparece como por ejemplo título o en su dirección de web, tipo de archivo donde cada página busca el formato que prefiera, derechos de uso. (Miller, 2012).

En la bibliografía estudiada no se encontró el estándar de metadato específico que utiliza Google, sin embargo, en los estudios realizados con la herramienta se logró identificar los campos que se utilizan para visualizar los resultados, ejemplo de estos son: título, formato, url, fecha de rastreo, tamaño del contenido, *host*, *outlinks*, dimensión de las imágenes, resolución y la fecha de la última actualización.

Bing[8]

Bing es un motor de búsqueda de Microsoft. Utilizar Bing es muy parecido a utilizar otros buscadores, en el que puedes ingresar una o más palabras clave en la barra de búsqueda para encontrar páginas web que mencionen las palabras o frases que ingresaste. En Bing, puedes llevar a cabo una búsqueda de artículos,

[6] Accesible en: http//www.google.com/
[7] Palabras y símbolos que pueden ser utilizados en el cuadro de búsqueda para restringir los resultados a mostrar.
[8] http://es.wikihow.com

imágenes, videos, noticias, sitios de compras y más. También puedes ingresar comandos específicos con tus palabras claves para reducir tus resultados de búsqueda.

Para una búsqueda avanzada de documentos en el buscador Bing, se puede contar con filtros que limitan los resultados según el sitio, tipos de archivos, título, idioma, región, búsqueda segura, ubicación, mostrar cantidad de resultados en cada página (Rautenstrauch, 2010).

En la bibliografía estudiada no se encontró el estándar de metadato específico que utiliza Bing, sin embargo, en los estudios realizados con la herramienta se logró identificar los campos que se utilizan para visualizar los resultados que son: título, formato, url, fecha de rastreo, tamaño del contenido, *host*, *outlinks*, dimensión de las imágenes, resolución y la fecha de la última actualización.

Yahoo

Su misión es ser el servicio global de Internet más esencial para consumidores y negocios. Este permite a los usuarios obtener respuestas a sus criterios de búsquedas en un lenguaje natural. Posee un portal de internet, un directorio web y una serie de servicios incluido el popular correo de Yahoo. Para el caso de la búsqueda de documentos, el sistema incluye fecha, dominio, idioma, país, título y tipo (Clarenc, 2011).

Yahoo junto a Google y Bing inició un proyecto para el desarrollo del estándar de metadato LRMI (*Learning Resource Metadata Initiative)* que es un esquema de marcación sencillo destinado a estandarizar la manera de describir o etiquetar las características educativas de los recursos en la web y un paso clave para el desarrollo de una mejor experiencia de búsqueda para los educadores y estudiantes. Además, algunas de sus características permitirán el desarrollo de sistemas de aprendizaje de próxima generación basados en aprendizaje guiado personalizado. Actualmente este estándar no es muy utilizado. (Vargas-Arcila, 2016).

En la bibliografía estudiada no se encontró el estándar de metadato específico que utiliza Yahoo, sin embargo, en los estudios realizados con la herramienta se logró identificar los campos que se utilizan para visualizar los resultados que son: título, formato, url, fecha de rastreo, tamaño del contenido, *host*, *outlinks*, dimensión de las imágenes, resolución y la fecha de la última actualización.

1.3.2 Sistemas homólogos a nivel nacional

Red Cuba

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 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. La plataforma c.u.b.a muestra los datos según el formato configurado. Cuando se analizan los resultados obtenidos para un criterio de búsqueda estándar, los datos más relevantes son: título, palabras clave, host, tipo, fecha, url, contenido, id, descripción, etc. Esta plataforma facilita al usuario la búsqueda de contenidos en varios formatos digitales[9] (páginas web, imágenes y documentos). Analizando 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 (Guevara, 2015).

Orión

Orión es un sistema de recuperación de la información, conocido también como buscador que está diseñado e implementado en la UCI con tecnologías libres que cumple con más de un año de utilización por la comunidad universitaria del país, contiene un gran número de funcionalidades que lo convierten en una herramienta para la búsqueda de información alojada en la web cubana, posee servicios desarrollados y otros en proyección de desarrollo. En Orión se muestran las descripciones de los componentes básicos de una herramienta de recuperación de la información, así como algunos datos de las tecnologías utilizadas para su desarrollo.

Orión cuenta con una búsqueda de documentos, imágenes y páginas web, pero para los criterios de búsqueda introducidos los resultados que se muestran al usuario no son los mejores. Todo esto motiva a desarrollar un subsistema de estandarización de documentos para el buscador Orión.

Con el objetivo de identificar las funcionalidades existentes en el buscador Orión, se elabora una tabla comparativa para analizar los resultados de los metadatos insertados por cada uno de los buscadores.

[9] Se refiere a todo archivo, carpeta o documento generado bajo tecnología computacional (Castellanos, 2012).

Tabla 1. Comparación entre buscadores en cuanto a los metadatos de los documentos.

Buscadores					
Resultados de los metadatos	***Google***	***Yahoo***	***Bing***	**Red Cuba**	**Orión**
Título	Sí	Sí	Sí	Sí	Sí
Formato	Sí	Sí	Sí	Sí	Sí
Url	Sí	Sí	Sí	Sí	Sí
Fecha	Sí	Sí	Sí	No	No
Host	Sí	Sí	Sí	Sí	Sí
Resolución	Sí	Sí	Sí	No	No
Contenido	Sí	Sí	Sí	Sí	Sí
Tipo	Sí	Sí	Sí	Sí	Sí
Dimensión de las imágenes	Sí	Sí	Sí	No	No

1.3.3 Tipos de estándares de metadatos

Dublin Core

La iniciativa *Dublin Core* [10] se creó en 1995 con el propósito de crear estándares que facilitaran la descripción y recuperación de recursos de información. Se creó un conjunto de descriptores que hoy en día es el más extendido en la web. Dublin Core describe material digital como videos, sonidos, imágenes, textos y páginas web. Los metadatos más relevantes para describir, identificar y encontrar un documento en la red son: título, formato, tipo, descripción, fecha, contenido, resumen, url, lenguaje y palabras clave. Si este estándar se usara mundialmente, se conseguiría que todas las aplicaciones automáticas que intentan indexar la información de internet, como los buscadores, tendrían toda la información necesaria para manipular los documentos en su propia cabecera, facilitando su indexación y provocando una mejora de eficiencia de los motores de búsqueda (Lapuente, 2013).

[10] Accesible en: http://es.dublincore.org.

Learning Object Metadata (*LOM*)

LOM es un modelo de datos, usualmente codificado en XML, usado para describir un objeto de aprendizaje y otros recursos digitales similares. Un LOM se define como una entidad, digital o no digital que puede ser usada, reutilizada o referenciada durante cualquier actividad de aprendizaje basada en la tecnología. Los estándares del LOM se centran en el conjunto mínimo de propiedades que permiten que los objetos educacionales sean gestionados, ubicados y evaluados. Las cualidades relevantes de los objetos de aprendizaje que se describen incluyen: título, idioma, tipo de objeto, autor, propietario, formato, tales como estilo de la enseñanza o de la interacción (Gaitán, 2014).

Text Encoding Initiative (*TEI*)

Están dirigidas a cualquier persona que quiera intercambiar información almacenada en un formato electrónico. En ellas se enfatiza el intercambio de información textual, pero también se hace referencia a otras formas de información como son las imágenes y el sonido. Las normas son del mismo modo aplicables a la creación de nuevos recursos y al intercambio de los ya existentes. El encabezado TEI provee información similar a la de la portada de un texto impreso. Tiene hasta cuatro partes: una descripción bibliográfica del texto electrónico, una descripción de cómo ha sido etiquetado, una descripción no bibliográfica del texto (un perfil del texto), y una revisión de su historia (su creación). (Quero, 2001).

Metadata Object Description Schema (*MODS*)

MODS un esquema de descripción bibliográfica. Los metadatos MODS pueden emplearse junto con MADS **(***Metadata Authority Description Schema*) para la descripción de las autoridades, siendo completamente compatibles por las propiedades de extensibilidad de XML, leguaje que utiliza como base de su sintaxis, igual que el formato MADS. Esto supone una ventaja que le permite una fácil interoperabilidad con cualquier formato de descripción bibliográfica e incluso archivística. Los elementos de MODS presentan generalmente la semántica del MARC. No asume el uso de ningún código específico de catalogación. Presenta varios elementos que tienen un atributo ID opcional para facilitar el enlace a nivel de elemento (Ochando, 2012).

Resource Description Framework (*RDF*)

Provee un mecanismo para integrar múltiples esquemas de metadatos. En RDF un *namespace* se define apuntando a un recurso web que describe el esquema de metadatos usado en la descripción. Se pueden definir múltiples *namespace* lo que permite que en la descripción de un recurso puedan ser combinados elementos de diferentes esquemas. De esta forma pueden enlazarse a otras múltiples descripciones creadas en diferentes momentos y con propósitos diferentes. Por lo general, RDF utiliza el lenguaje XML.

Mientras que XML es un lenguaje para modelar datos, RDF es un lenguaje para especificar metadatos. XML falla en la escalabilidad de los datos puesto que el orden de los elementos es antinatural y su mantenimiento es muy difícil y costoso, por el contrario, RDF permite la interoperabilidad entre aplicaciones que intercambian información comprensible por la página web, para proporcionar una infraestructura que soporte actividades de metadatos (Lapuente, 2013).

OWL

Está diseñado para ser usado en aplicaciones que necesitan procesar el contenido de la información en el lugar de únicamente representar información para los humanos. OWL facilita un mejor mecanismo de interpretabilidad de contenido web que los mecanismos admitidos por XML, RDF, y esquema RDF (RDF-S) proporcionando vocabulario adicional junto con una semántica formal. OWL tiene tres sublenguajes, con un nivel de expresividad creciente: OWL Lite, OWL DL, y OWL Full. Los lenguajes de ontologías deben permitir escribir conceptualizaciones explícitas y formales. Entre sus principales requisitos están que tenga una sintaxis bien definida, que tenga la posibilidad de razonamiento eficiente y que tenga suficiente riqueza semántica (McGuinness, 2004).

Simple Knowledge Organization System (*SKOS*)

SKOS ha sido diseñado para proporcionar un modo de migrar a la web semántica sistemas de organización del conocimiento ya existentes con un coste bajo. SKOS también proporciona un lenguaje conceptual de modelado muy sencillo e intuitivo para desarrollar y compartir nuevos sistemas de organización. Permite la creación y publicación de conceptos en la web, así como vincularlos con datos en este mismo medio e incluso integrarlos en otros esquemas de conceptos. Además, identifica los recursos conceptuales (conceptos) mediante URL, etiquetarlos con literales de uno o varios idiomas, documentarlos con diversos tipos de notas, relacionarlos entre sí mediante estructuras jerárquicas informales o redes asociativas, y agregarlos a esquemas de conceptos. El objetivo de SKOS no es sustituir vocabularios conceptuales originales en su contexto inicial de uso, sino que puedan implementarse en un espacio compartido, basado en un modelo simplificado, que haga posible su reutilización (Sánchez, 2010)

Online Information Exchange (*ONIX*)

ONIX nace de la premisa que mientras más información tengan los clientes de un libro, habrá mayor opción de que lo compren. En el formato impreso, la cubierta sobrepuesta del libro contiene mucha información promocional: diseño de la cubierta, reseña, biografía del autor, etc. Toda esta información empuja al lector a comprarlo. Sin embargo, obtener todos esos datos desde las editoriales para ser entregados a los

vendedores, ha sido un desafío, complicado por el hecho que las más importantes compañías (tal como Ingram, Bowker y Amazon) tienen diferentes formatos para recibir los datos. Esta falta de estandarización ha provocado un consumo de tiempo y ha dificultado a las editoriales, las actividades de exportación de la información de sus libros. Por esto, el propósito de ONIX es estandarizar la transmisión de la información de sus productos, de forma que los mayoristas, minoristas y otros involucrados en la cadena de suministro, sean capaces de aceptar la información transferida electrónicamente en el formato internacional ONIX. Este puede definir muchas cosas en un libro, por ejemplo: título, autor, precio, disponibilidad, opiniones y extractos (García, 2009).

Después de haber realizado un profundo estudio de los metadatos que utilizan los SRI y las interfaces de cada uno de los estándares estudiados se llegó a la conclusión de que Orión posee deficiencia en los metadatos verificados con los estudios de homólogos porque en ocasiones se guardan de forma incorrecta metadatos importantes como: título, formato, url, fecha de rastreo, tamaño del contenido, host, outlinks, dimensión de las imágenes, resolución y la fecha de la última actualización. Por lo tanto, se decide basar la propuesta de estandarización de documentos en el estándar de metadatos Dublin Core que posee características comunes como título, tipo, formato, contenido, URL con los estándares LOM, ONIX y SKOS. (*ver tabla 2*). Este estándar ayuda a los motores de búsqueda en la recuperación de información en la red, convirtiéndose en un formato de propósito general y que a su vez sea de gran utilidad para el usuario.

Tabla 2. Comparación de los metadatos que posee Dublin Core con otros estándares.

Estándares				
Metadatos	**LOM**	**ONIX**	**SKOS**	**Dublin Core**
Título	**Sí**	**Sí**	**Sí**	**Sí**
Tipo	**Sí**	**Sí**	**Sí**	**Sí**
Formato	**Sí**	**Sí**	**Sí**	**Sí**
Contenido	**Sí**	**Sí**	**Sí**	**Sí**
Url	**Sí**	**Sí**	**Sí**	**Sí**

1.3.4 Formatos para el intercambio de información.

JSON

JSON es un formato de intercambio de datos basado en texto. Su nombre corresponde con las siglas en inglés de *JavaScript Object Notation*. Aunque es un formato muy cercano en sintaxis a JavaScript, por ser muy sencillo de tratar para codificar datos de objetos o para obtenerlos, se utiliza también en muchos otros lenguajes de programación (C, C++, Java y Python) como una alternativa, por ejemplo, a XML, que tiene un objetivo similar pero que, por incluir más metainformación, necesita más texto y por tanto ocupa más, consume más ancho de banda y necesita más recursos para codificar y decodificar la información que JSON. Los datos de un objeto codificados en JSON están encerrados entre llaves, las distintas propiedades que se incluyen dentro se separan por comas y los nombres de las propiedades preceden al valor, del que se separan por dos puntos (Ventura, 2016).

XML

XML consiste de una serie de reglas, pautas o convenciones para planificar formatos de texto para tales datos, de manera que produzcan archivos que sean fácilmente generados y leídos por un ordenador, que sean inequívocos y que eviten los problemas más comunes como la falta de extensibilidad, la falta de interoperabilidad entre plataformas o la falta de soporte para universalizar su tratamiento. XML se ha creado para enriquecer la estructura de los documentos que pueden ser usados en la Web, puesto que las otras alternativas viables, HTML y SGML, no eran demasiadas prácticas para este propósito. XML permite gestionar los datos, aunque procedan de diversas fuentes y también permite el intercambio de documentos entre distintas aplicaciones ya estén en un único ordenador o en una red. Sirve para representar e intercambiar datos de forma independiente a su presentación. Los datos se pueden gestionar desde el propio cliente Web, la información en XML está mucho más estructurada, esto facilitará enormemente la labor a los buscadores y los robots y agentes inteligentes, que accederán a los datos de manera más precisa (Lapuente, 2013).

Por lo expuesto con anterioridad se considera que el tipo de formato que se va a utilizar es JSON porque es un objeto válido de JavaScript por lo que es el formato perfecto para ese lenguaje. La mayoría de los navegadores web modernos incluyen funciones nativas para codificar y decodificar JSON, lo que le da un punto de ventaja en lo que se refiere a desempeño y disminuyen los riesgos de seguridad. Además, por su simplicidad y facilidad de implementación le otorgan un gran desempeño y lo convierten en una de las alternativas ideales al momento de reemplazar XML.

1.4 Metodología de desarrollo de software

Una metodología de desarrollo en ingeniería de *software* es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo. Es un proceso para la producción organizada del *software*, empleando para ello una colección de técnicas predefinidas y convencionales en las notaciones. En esta metodología se presenta normalmente como una serie de pasos que deben seguirse para el desarrollo del *software*. Estas especifican (Pressman, 2005):

- Cómo se debe dividir un proyecto en etapas.
- Qué tareas hay que realizar en cada etapa.
- Qué salidas se producen y cuándo.
- Qué herramientas se utilizan.
- Cómo se gestiona y controla un proyecto.

Las que por su parte se adaptan al cambio teniendo un tiempo de respuesta rápido son denominadas Metodologías Ágiles, se centran en la especificación, diseño e implementación del sistema de forma incremental. Implican directamente a los usuarios en el proceso de desarrollo del *software*.

En consecuencia, de lo antes planteado y tomando en cuenta que la propuesta de solución deberá ser integrada a un proyecto real existente el cual reside bajo la tutela y estrategia de trabajo del Centro de Ideo-informática, CIDI a partir de ahora, se decidió elaborar la presente investigación bajo las pautas establecidas por la metodología Proceso Unificado Ágil o AUP por sus siglas, en una adaptación hecha por la UCI. Dicha metodología se seleccionó para que guiara el desarrollo del subsistema de estandarización de documentos, además de estar diseñada para grupos de trabajo pequeños, fomentar el trabajo en equipo integrando al cliente, centrarse en la naturaleza colaborativa de desarrollo de *software* y es la adoptada por el centro y por ende la más acorde a las necesidades del proyecto.

AUP-UCI define tres fases:

1. I**nicio**: El objetivo es realizar un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo, costo y decidir si se ejecuta o no el proyecto.
2. **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.
3. **Cierre:** En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las

actividades formales de cierre del proyecto.

La estrategia de trabajo definida por AUP se rige por cinco principios fundamentales:

1. **Simplicidad**: Todo se describe concisamente utilizando poca documentación
2. **Agilidad**: El ajuste a los valores y principios de La alianza ágil.
3. **Centrarse en actividades de alto valor**: La atención se centra en las actividades que en realidad lo requieren, no en todo el proyecto.
4. **Herramientas independientes**: Las herramientas a utilizar las define el equipo de desarrollo y está a disposición del usuario (Se sugiere utilizar las herramientas más adecuadas, simples y de código abierto).
5. **Adaptar el producto para satisfacer necesidades**: La metodología AUP es un producto de fácil uso utilizando cualquier herramienta. No es necesario comprar una herramienta especial, o tomar un curso, para adaptar esta metodología.

1.4.1 Artefactos que genera la metodología

Modelo de dominio

Un modelo de dominio es un artefacto de la disciplina de análisis, construido con las reglas de UML durante la fase de concepción, que contiene conceptos propios de la realidad física. Los objetos del dominio representan las cosas que existen o los eventos que suceden en el entorno del sistema. (...) El objetivo del modelado del dominio es comprender y describir las clases más importantes dentro del contexto del sistema, por lo cual puede ser tomado como el punto de partida para su diseño (Gónzalez, 2012).

Especificación de los requisitos de software

Los requisitos del *software* permiten establecer lo que el sistema debe hacer, sus características fundamentales, y las restricciones en el funcionamiento del sistema y los procesos de desarrollo del software. De manera general, estos requisitos expresan las necesidades objetivas que presentan los usuarios, ante un sistema que resuelve un problema en particular de un determinado dominio (Sommerville, 2005).

Requisitos Funcionales

Los requisitos funcionales son aquellos que describen qué debe hacer el sistema desde el punto de vista de las necesidades del usuario, son capacidades o condiciones que debe cumplir el sistema y que están fuertemente ligados a las opciones del programa (Sommerville, 2005).

Requisitos no Funcionales

Los requisitos no funcionales, como su nombre sugieren, son aquellos requisitos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento (Sommerville, 2005).

Historias de usuario (HU)

Las HU utilizan los requerimientos de los clientes según el negocio y son utilizadas para poder realizar la estimación de cada una de las iteraciones durante la fase de planificación. Las HU son escritas por el equipo de trabajo en conjunto con los clientes en base a lo que se estima que es necesario para el sistema. Están escritas en un formato de oraciones en la terminología del cliente, sin necesidad de sintaxis técnicas. También son utilizadas para poder crear las pruebas de aceptación. Las HU solo proveen suficiente detalle para poder realizar la estimación de cuánto tardará en ser implementada dicha funcionalidad. Una gran diferencia entre las HU y los documentos tradicionales es que se centran en lo que el cliente necesita (Balarezo, 2013).

En las **HU** se considera:

La prioridad en el negocio:

- **Alta:** Cuando son consideradas por los clientes esenciales para el funcionamiento del negocio.
- **Media:** Cuando el cliente cree que son necesarias, pero estas no intervienen en gran medida en el desarrollo del negocio.
- **Baja:** Cuando constituyen procesos que se deben tener en cuenta, pero su ausencia no perjudica el flujo principal del negocio.

El riesgo en desarrollo:

- **Alto:** Cuando en la implementación de las HU pueden surgir errores que lleven a la inoperatividad del código.
- **Medio:** Cuando en la implementación de las HU pueden existir errores que retrasen la entrega del producto.
- **Bajo:** Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.

1.5 Herramientas, lenguajes y tecnologías de desarrollo

Para desarrollar la propuesta de solución, necesidad surgida del estudio realizado de herramientas

homólogas, se hace necesario estudiar varias tecnologías, lenguajes y herramientas disponibles para llevar a cabo el objetivo general de la investigación.

1.5.1 Lenguaje de modelado

UML (Lenguaje unificado de modelado)

Es el lenguaje de modelado de sistemas de *software* más conocido y utilizado en la actualidad; está respaldado por el OMG (Grupo de Gestión de Objetos). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de *software*. UML ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de *software* reutilizables (Larman, 2013).

Modelado de proceso de negocio

BPMN (Notación de Modelado de Procesos de Negocio) es utilizada como un estándar para la representación de procesos de negocios, independientemente de la metodología empleada para automatizar dichos procesos. Esta notación ha sido diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes en las actividades. Proporciona un lenguaje común para que las partes involucradas en el análisis de negocios (cliente y desarrollador) puedan intercambiar sobre un mismo modelo de forma clara, precisa y eficiente (Baisley, 2006).

Se selecciona UML como lenguaje de modelado para especificar, visualizar, construir y documentar los artefactos del sistema. Es un lenguaje estándar, fácil de aprender, y ofrece una amplia variedad de diagramas para mostrar el sistema desde varias perspectivas. Está especialmente diseñado para apoyar un estilo de desarrollo iterativo e incremental y presenta tecnología orientada a objetos. Tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el diseño con los diagramas de clases, objetos, hasta la implementación y configuración con los diagramas de despliegue.

1.5.2 Herramientas CASE (Ingeniería de software asistida por computadora)

Las herramientas de Ingeniería de *Software* Asistido por Computadora (CASE, en inglés *Computer Aided Software Engineering*) son aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de un *software* reduciendo el costo del mismo. Pueden servir de ayuda durante el ciclo de vida de desarrollo del *software* en tareas como el diseño del proyecto, cálculo de costos, implementación de parte del código

automáticamente a partir del diseño dado, compilación automática, documentación o detección de errores. Existen múltiples herramientas con estos fines, tales como *Enterprise Architect*, Visual Paradigm, ArgoUML, *Rational Rose*, entre otras.

Visual Paradigm

Es una herramienta CASE que propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasa por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Soporta el ciclo de vida completo del proceso de desarrollo de software a través de la representación de todo tipo de diagramas. Fue diseñada para una amplia gama de usuarios interesados en la construcción de sistemas de *software* de forma fiable a través de la utilización de un enfoque orientado a objetos.

Algunas de sus características son:

- Disponibilidad en múltiples plataformas.
- Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Disponibilidad de múltiples versiones, para cada necesidad.
- Licencia: gratuita y comercial.
- Soporta aplicaciones web.

Rational Rose

Es una herramienta CASE, desarrollada por *Rational Corporation* basada en UML, que permite crear los diagramas que se generan durante el proceso de ingeniería en el desarrollo del *software*.

Características adicionales:

- Característica de control por separado de componentes modelo que permite una administración más granular y el uso de modelos.
- La generación de código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo –código configurable.
- Capacidad de análisis de calidad de código.
- Publicación web y generación de informes para optimizar la comunicación dentro del equipo (Acosta,

2011) .

Se selecciona la herramienta Visual Paradigm porque propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación. Ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del *software* a través de la representación de todo tipo de diagramas.

1.5.3 Lenguajes de Programación

Según la definición teórica de la RAE[11], se entiende como lenguaje a un sistema de comunicación definido por una determinada estructura y contenido; se entiende además en la programación como el vocabulario propio de la informática, como el procedimiento de escritura del código fuente de un *software*. Con estas nociones en claro, se puede afirmar que un lenguaje de programación es aquella estructura que, con cierta base sintáctica y semántica, imparte distintas instrucciones a un programa de computadora.

Lenguajes de desarrollo del lado del servidor

Java

Java es un lenguaje de programación orientado a objetos que se popularizó a partir del lanzamiento en su primera versión, la JDK 1.0 en 1996. Actualmente es uno de los lenguajes más usados para la programación en todo el mundo. Fue desarrollado por la compañía *Sun Microsystems*, con la idea original de usarlo para la creación de páginas web. La programación en Java, permite el desarrollo de aplicaciones bajo el esquema de clientes servidor, como de aplicaciones distribuidas, lo que hace capaz de conectar dos o más computadoras u ordenadores ejecutando tareas simultáneamente, de esta forma logra distribuir el trabajo a realizar (Fernández, 2005).

Entre las características más importante se encuentran (Laureano, 2004):

- **Simple:** ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de estos. Elimina muchas de las características de otros lenguajes como C++, para mantener reducida la especificación del lenguaje.
- **Orientado a objetos**: implementa la tecnología de C++ y soporta las tres características del paradigma orientado a objetos. Encapsulamiento: Implementa información oculta. Polimorfismo: El mismo mensaje se envía a diferentes objetos, resultando en comportamientos que dependen de la naturaleza del objeto que recibió el mensaje. Herencia: Puede definir nuevas clases y comportamientos (métodos) basados

[11] RAE: Real Academia Española.

en clases existentes.

- **Robusto:** realiza verificaciones en busca de problemas, tanto en tiempo de compilación, como de ejecución. La comprobación de tipos ayuda a detectar errores. Obliga a la declaración explícita de los métodos.
- **Seguro:** la seguridad tiene dos facetas: Se eliminan características como los apuntadores y el casting implícito para prevenir el acceso ilegal a la memoria El código Java pasa por muchas verificaciones antes de ser ejecutado en una máquina mediante el *classloade*[12].
- **Distribuido:** presenta extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como http y ftp. Por si sólo no es distribuido, pero proporciona herramientas para que nuestros programas puedan serlo.

1.5.4 IDE (Entorno integrado de desarrollo)

Un IDE[13] es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica, lo cual proporciona un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Python, Java, PHP, C#, Delphi y Visual Basic. Los IDE pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes (Fernández, 2011).

Netbeans 8.0

NetBeans es un IDE gratuito y de código abierto el cual permite al programador un rápido y fácil desarrollo, apto tanto para aplicaciones de escritorio, móviles y aplicaciones web. Es multiplataforma, encontrándose disponible para Windows, Mac, Linux y Solaris. Posee una amplia plataforma de aplicaciones la cual posibilita desarrollar rápidamente utilizando la plataforma Java, así como Ajax, Groovy, Grails, C /C+ +, así como aplicaciones HTML5, JavaScript y CSS. Ofrece también un gran conjunto de herramientas para PHP (NetBeans, 2015).

El proyecto de NetBeans está apoyado por una amplia comunidad de desarrolladores y ofrece una amplia documentación y recursos de capacitación, así como una gran cantidad de *plugins* de terceros.

Se decide utilizar a Netbeans 8.0 como entorno de desarrollo integrado por ser una herramienta que le brinda facilidades al desarrollador a la hora de programar como es el depurador, además por su soporte a

[12] Classloade: es una clase encargada de cargar otras clases en memoria según la necesidad que tenga la máquina virtual java (Caules, 2013).
[13] Integrated Development Environment por sus siglas en inglés.

tecnologías, estabilidad y grandes posibilidades de ser extendido gradualmente por desarrollos comunitarios, permitiendo agregar continuamente nuevas funcionalidades.

1.5.5 Servidor de indexación

Solr

Es una plataforma de búsquedas basada en Apache Lucene, que funciona como un "servidor de búsquedas". Sus principales características incluyen búsquedas de texto completo, resaltado de resultados y manejo de documentos (como Word y PDF). Solr es escalable, permitiendo realizar búsquedas distribuidas y replicación de índices (Seta, 2010).

Según la página principal de Solr, las características más destacables de este servidor de búsquedas son:

- Capacidades avanzadas de búsqueda de texto completo.
- Optimizado para un tráfico web elevado.
- Estadísticas del servidor expuestas mediante JMX para su monitorización.
- Escalable linealmente, replicación automática del índice, recuperación automática.
- Flexible y adaptable a través de configuraciones en formato XML.
- Arquitectura extensible mediante *plugins*.

Solr está escrito en Java y se ejecuta como un servidor de búsqueda de texto completo independiente dentro de un contenedor de *servlets* como Tomcat. Solr utiliza la biblioteca Java de búsqueda en su base para la indexación de texto completo y de búsqueda. Se utiliza en el SRI Orión y la configuración externa de Solr permite que sea adaptado a casi cualquier tipo de aplicación Java sin codificación, simplemente hay que utilizarlo con peticiones GET para realizar las búsquedas en el índice, y POST para agregar documentos (Apache, 2015).

1.5.6 Mecanismo para la recolección de Información

Nutch

Como *spider* se utiliza Nutch debido a las múltiples ventajas que ofrece a lo largo de todo el proceso de rastreo y almacenamiento de la información. Presenta *crawler* web libre y de código abierto desarrollado en Java bajo la licencia de Apache. Este se utiliza en el SRI Orión, proporciona interfaces extensibles para implementaciones personalizadas. Inicialmente fue implementado sobre la base de Apache Lucene, aunque ya la versión actual es independiente de Lucene. La arquitectura de Nutch es flexible permitiendo realizarle mejoras por parte de los usuarios a través de *plugins*. Este es independiente del servidor de indexación lo

que permite la integración con Solr (NutchWiki, 2015).

Nutch, como mecanismo de rastreo, posee ciertos componentes llamados *parsers*, los cuales se encargan de descomponer las páginas web y analizar cada uno de los recursos que la componen o tienen relación. Uno de estos parsers se denomina Tika, el cual puede descomponer una gran cantidad de documentos, entre ellos HTML, documentos ofimáticos, pdf y muchos más. Actualmente, aunque Tika soporta una variedad de formatos sobre una gran cantidad de tipos de documentos, no es capaz de obtener toda la información que se pudiera obtener de las imágenes que encuentra, constituyendo una de sus debilidades (Leyva, 2016).

Julien Nioche, especialista en Ingeniería de Texto, Procesamiento del Lenguaje Natural y la Recuperación de Información; describe los procesos internos de Nutch de la forma siguiente (Nioche, 2013):

1. Inyección: Obtención de las URLs contenidas en el semillero inicial de Nutch, para la inicialización de la base de datos web y el comienzo del rastreo.

2. Generación: Se genera una nueva lista de selección de URLs para el segmento. En el primer rastreo solo se toman los links del semillero inicial, y en los demás se agregan los nuevos enlaces encontradas en los rastreos anteriores. Además, el generador es el encargado de definir la frecuencia de visitas a las páginas.

3. Selección: Selecciona una de las URLs de la lista generada, trata de establecer una conexión con la página para su descarga y la almacena en el segmento de rastreo.

4. Análisis: Una vez descargada la página, se procede al análisis de su contenido (texto y metadatos). Es aquí donde se descubren los nuevos enlaces en el contenido.

5. Actualización de la Base de Datos: Al terminar el análisis, Nutch actualiza la base de datos web con el contenido encontrado y añade los nuevos enlaces a la base de datos de URLs.

6. Indexación: Al finalizar los procesos anteriores, Nutch envía toda la información a la aplicación que se encargará de indizar todo el contenido, para su posterior consulta.

1.6 Conclusiones del capítulo

El análisis de las técnicas de procesamiento de información, la selección de los estándares de metadatos y los conceptos asociados al objeto de estudio sirvieron de base y guía para el desarrollo de la investigación. Las deficiencias encontradas en los motores de búsqueda nacionales, el limitado acceso a los internacionales y la manipulación de los resultados presente en ellos, hacen necesario la creación de un subsistema de estandarización que permita la búsqueda de documentos en la red cubana. Las funcionalidades que brindan algunos de los buscadores más utilizados dígase: Google, Bing Yahoo, Red Cuba permitió identificar metadatos necesarios que ayudan al usuario a realizar búsquedas más certeras. Con el estudio realizado de las principales herramientas, tecnologías y lenguajes de programación a utilizar, se logró obtener una base tecnológica adecuada que permitirá el desarrollo de la solución.

Capítulo 2: Análisis y diseño del Subsistema de estandarización de documentos para el buscador Orión.

2.1 Introducción

El presente capítulo abordará el análisis y diseño del subsistema de estandarización de documentos para el buscador Orión. Se realiza una descripción de las características del subsistema, se identifican las clases de dominio, y la relación que existe entre ellas es mostrada a través del diagrama de clases del modelo del dominio. También se especifican los requerimientos funcionales y no funcionales, se realizan las historias de usuario y las descripciones textuales de estas. Además, se muestra el diagrama de clases, el modelo de datos y el modelo de despliegue, así como la descripción de los patrones de diseño empleados.

2.2 Modelo de dominio

Atendiendo al concepto del modelo de dominio descrito en el capítulo 1, sección 1.4.1 se presenta la descripción de los conceptos identificados por la autora en la presente investigación.

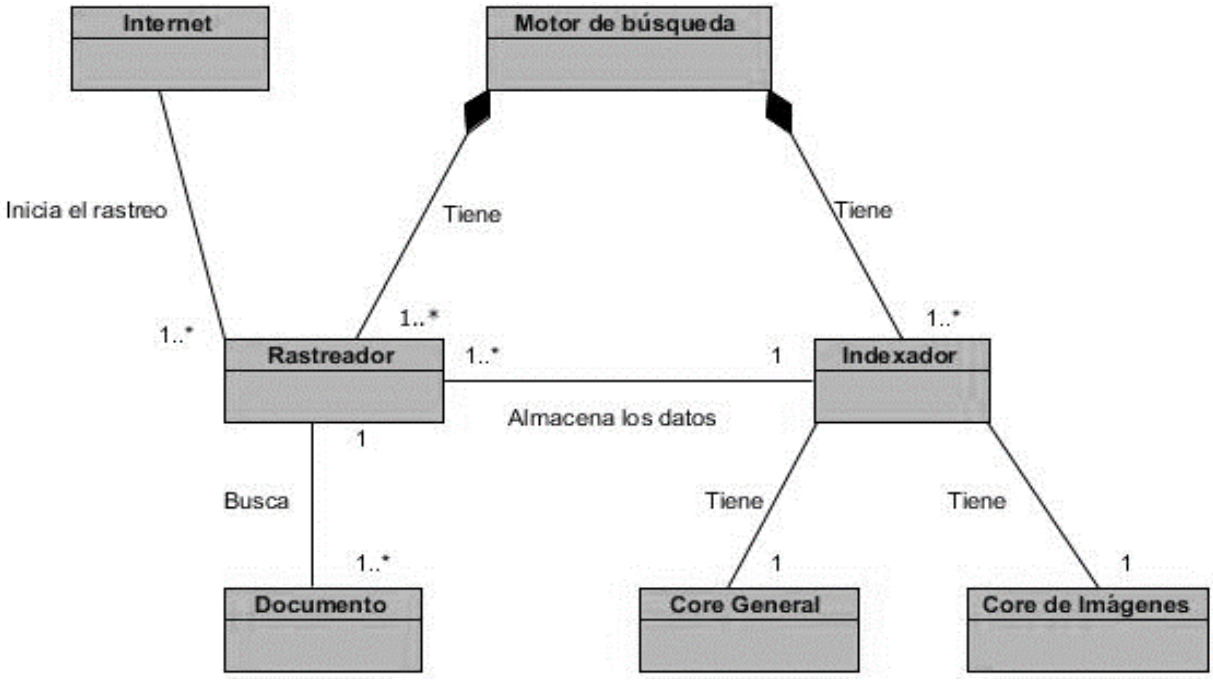

Figura 2. Diagrama del modelo de dominio.

2.2.1. Descripción de las clases del dominio

Motor de búsqueda: Sistema de recuperación de información compuesto por una interfaz, varios indexadores y varios rastreadores.

Indexador: Componente que se encarga de indexar los documentos.

Rastreador: Componente del motor de búsqueda que se encarga de recopilar los documentos en la web.

Documento: Tipo de contenido que puede ser: imágenes, páginas web, documentos.

Core: Es una colección especializada por tipos de documentos (ejemplo: imágenes, elementos generales) que contiene una instancia en ejecución de un índice y los archivos de configuración de Solr necesarios para su uso.

2.3 Descripción de la propuesta de solución

Dadas las necesidades planteadas en la situación problemática del presente trabajo de diploma, la solución propuesta constituye un subsistema de estandarización de documentos para el buscador Orión. Este subsistema debe recopilar información de la web mediante la realización de un *plugin* para la estandarización de documentos en el componente de rastreo (Nutch), siempre y cuando todos los metadatos por los cuales esta información es rastreada no se encuentren vacíos, garantizando así obtener información útil. Después de ser rastreada esa información se le envía al servidor de indexación (Solr).

2.3.1 Arquitectura del sistema propuesto

El proceso de diseño de la arquitectura debe decidir cuál funcionalidad es la más importante a desarrollar. Además, define cuales son los componentes más básicos del sistema y como se relacionan entre ellos para implementar la funcionalidad.

Se determina como arquitectura del subsistema (ver Figura 3), para lograr la consistencia con los componentes del sistema. Se emplea una arquitectura basada en componentes, la cual tiene como objetivo enfocarse en la descomposición del diseño en componentes funcionales o lógicos que expongan interfaces de comunicación bien definidas. Además, es un estilo de diseño para aplicaciones compuestas de componentes individuales (Pelaez, 2009).

Atendiendo a la arquitectura genérica de los motores de búsqueda descrita en la sección 1.2.6 y las particularidades del sistema propuesto, se sugiere la arquitectura que se muestra en la siguiente figura.

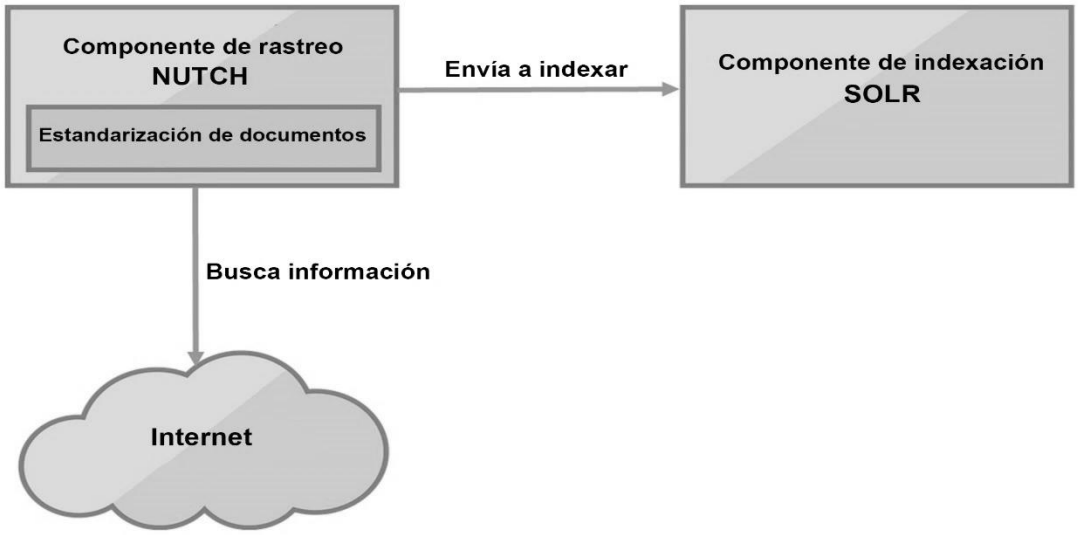

Figura 3. Arquitectura del sistema propuesto.

Como se muestra en la arquitectura del sistema, el componente Nutch a través de sus *plugins* es el encargado de lograr la estandarización de documentos. Este mecanismo se inicia con el rastreo de diferentes recursos localizados en internet. Una vez analizados estos recursos se extraen un conjunto de metadatos que son enviados al componente de indexación Solr.

2.4 Especificación de los requisitos de software

Luego de haber definido y modelado los conceptos asociados al dominio del problema y la relación entre ellos, se presentan los requisitos funcionales y no funcionales de la herramienta a desarrollar.

2.4.1 Requisitos Funcionales

Los requisitos funcionales de un sistema describen lo que el sistema debe hacer (Sommerville, 2005). A continuación, se presentan los requisitos funcionales del subsistema de estandarización de documentos para el buscador Orión.

Tabla 3. Requisitos funcionales del sistema.

Número	Requisito Funcional	Prioridad
RF1	Identificar el metadato url del documento para su almacenamiento en el core de los elementos generales.	Alta
RF2	Identificar el metadato contenido del documento para su almacenamiento en el core de los elementos generales.	Alta
RF3	Identificar el metadato título del documento para su almacenamiento en el core de los elementos generales.	Alta
RF4	Identificar el metadato de fecha de rastreo del documento para su almacenamiento en el core de los elementos generales.	Media
RF5	Identificar el metadato tamaño del contenido texto del documento para su almacenamiento en el core de los elementos generales.	Media
RF6	Identificar el metadato tipo del documento para su almacenamiento en el core de core de los elementos generales.	Alta
RF7	Identificar el metadato url de la página del documento para su almacenamiento en el core de los elementos generales.	Alta
RF8	Identificar el metadato título de la página del documento para su almacenamiento en el core de los elementos generales.	Alta
RF9	Identificar el metadato *host* del documento para su almacenamiento en el core de los elementos generales.	Media
RF10	Identificar el metadato *outlinks* del documento para su almacenamiento en el core de los elementos generales.	Media
RF11	Identificar el metadato fecha de última actualización del documento para su almacenamiento en el core de los elementos generales.	Alta
RF12	Identificar el metadato tamaño del contenido original del documento para su almacenamiento en el core de los elementos generales.	Media
RF13	Identificar el metadato tipo del documento para su almacenamiento en el core de las imágenes.	Alta
RF13	Identificar el metadato *host* del documento para su almacenamiento en el core de las imágenes.	Media
RF14	Identificar el metadato url del documento para su almacenamiento en el core	Alta

	de las imágenes.	
RF15	Identificar el metadato de la dimensión del documento para su almacenamiento en el core de las imágenes.	Media
RF16	Identificar el metadato tamaño del contenido del documento para su almacenamiento en el core de las imágenes.	Media
RF17	Identificar el metadato relación de aspecto del documento para su almacenamiento en el core de las imágenes.	Media
RF18	Identificar el metadato resolución del documento para su almacenamiento en el core de las imágenes.	Media
RF19	Identificar el metadato fecha de rastreo del documento para su almacenamiento en el core de imágenes.	Media
RF20	Identificar el metadato título de la página del documento para su almacenamiento en el core de las imágenes.	Alta
RF21	Identificar el metadato url de la página del documento para su almacenamiento en el core de las imágenes.	Alta
RF22	Identificar el metadato fecha de última actualización del documento para su almacenamiento en el core de las imágenes.	Alta
RF23	Identificar el metadato tamaño del contenido original del documento para su almacenamiento en el core de las imágenes.	Media
RF24	Indexar metadatos identificados.	Alta
RF25	Validar metadatos identificados.	Alta

2.4.2 Requisitos no funcionales

A continuación, se presentan los requisitos no funcionales del subsistema de estandarización de documentos para el buscador Orión.

Usabilidad

RnF 1. Se requiere la instalación de la Máquina Virtual de Java (JVM por sus siglas en inglés) para el funcionamiento del rastreador y el indexador.

RnF 2. Se requiere el uso de herramientas y recursos de software libre, las cuales se podrán usar, modificar y distribuir libremente.

Soporte

RnF 3. El soporte del sistema se debe gestionar mediante el Centro de Soporte de la UCI.

Restricciones de diseño

RnF 4. Como lenguaje de programación para el *plugin* del componente de rastreo se deberá utilizar Java.

Hardware

RnF 5. Para el servidor de rastreo se necesita un equipamiento de 4 GB RAM, CPU de 4 núcleos y al menos 80 GB de Disco Duro.

RnF 6. El servidor de indexación debe tener como mínimo: un procesador a 1,60 GHz, 2GB de memoria RAM y 1TB de disco duro.

Historias de usuario (HU)

A continuación, se presentan las historias de usuario del subsistema de estandarización de documentos para el buscador Orión.

Tabla 4. Historia de Usuario # 1. Identificar el metadato url del documento para su almacenamiento en el core de las imágenes.

Historia de usuario	
Número: HU_1	**Nombre Historia de Usuario**: Identificar la url del documento para su almacenamiento en el core las imágenes.
Modificación de Historia de Usuario Número: **0**	
Usuario: **Sistema**	**Iteración Asignada:** 1
Prioridad en el Negocio: **Alta**	**Puntos Estimados:** 1
Riesgo en Desarrollo: **Alto**	**Puntos Reales:**1
Programador responsable: **Anietsy Perera Pérez**	
Descripción: **Luego de rastreada la imagen, identificar la dirección única del recurso informático disponible en la internet.**	

Tabla 5. Historia de Usuario # 2. Identificar el metadato tipo del documento para su almacenamiento en el core de las imágenes.

Historia de usuario	
Número: HU_2	**Nombre Historia de Usuario:** Identificar el metadato tipo del documento para su almacenamiento en el core de las imágenes.
Modificación de Historia de Usuario Número: **0**	
Usuario: **Sistema**	**Iteración Asignada:** 1
Prioridad en el Negocio: **Alta**	**Puntos Estimados:** 1
Riesgo en Desarrollo: **Alto**	**Puntos Reales:**1
Programador responsable: **Anietsy Perera Pérez**	
Descripción: **Luego de ser rastreada la imagen, identificar el tipo que puede ser jpg, png, gif, bmp.**	

Tabla 6. Historia de Usuario # 3. Identificar el metadato de fecha de rastreo del documento para su almacenamiento en el core de los elementos generales.

Historia de usuario	
Número: HU_3	**Nombre Historia de Usuario:** Identificar el metadato de fecha de rastreo para su almacenamiento en el core de los elementos generales.
Modificación de Historia de Usuario Número: **0**	
Usuario: **Sistema**	**Iteración Asignada:** 1
Prioridad en el Negocio: **Media**	**Puntos Estimados:** 1
Riesgo en Desarrollo: **Media**	**Puntos Reales:**1
Programador responsable: **Anietsy Perera Pérez**	
Descripción: **Luego de ser rastreada los elementos generales, identificar la fecha de rastreo en que se sube la información.**	

2.5 Diagramas de clases del diseño

Un diagrama de clases del diseño muestra la especificación para las clases de una aplicación, se centra en mostrar las definiciones entre entidades de un *software*. En su representación incluye datos de interés como: clases, asociaciones y atributos, interfaces, con sus operaciones y constantes, métodos, navegabilidad y dependencias (Thames, 2011).

En las siguientes figuras se muestran ejemplos de Diagramas de Clases de Diseño pertenecientes al Subsistema de estandarización de documentos del buscador Orión.

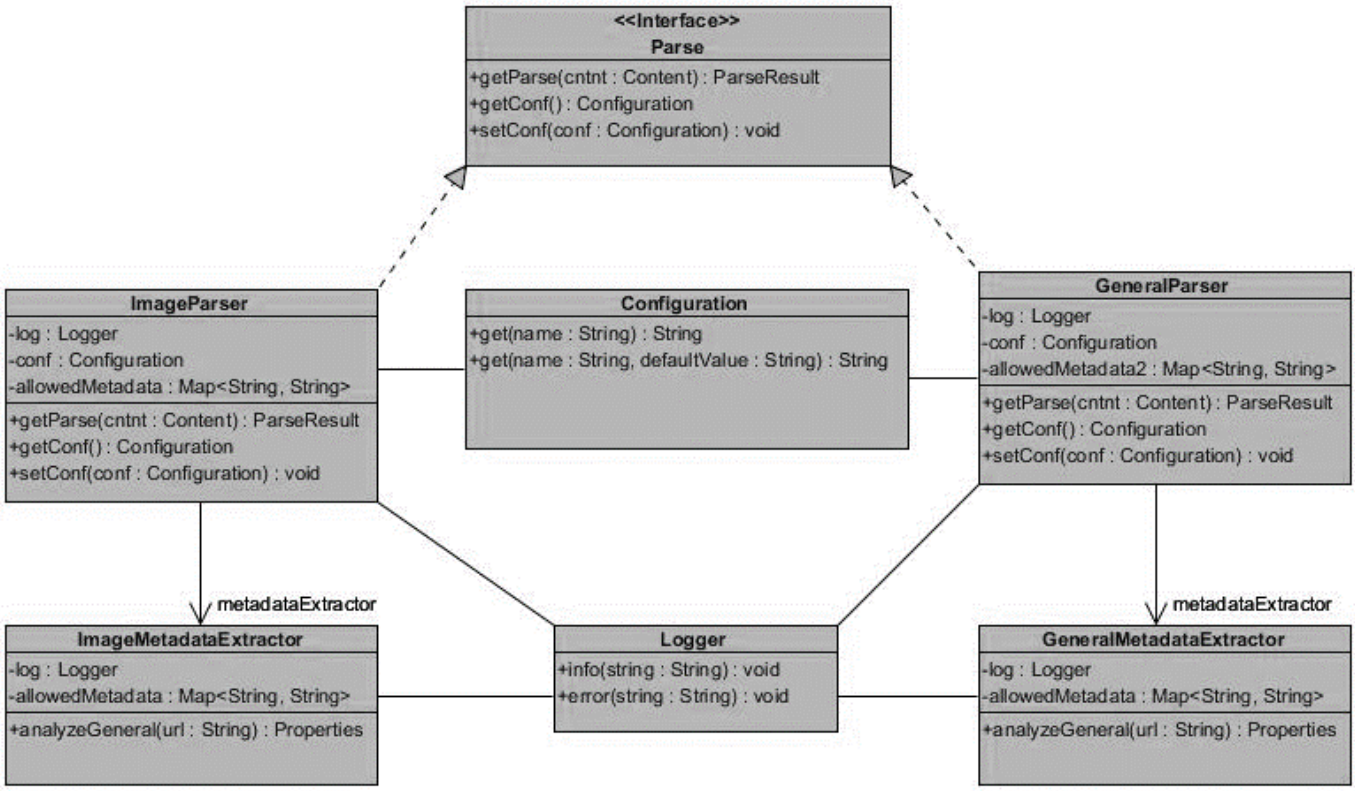

Figura 4. Diagrama de clases del diseño.

En el diagrama se muestra la interfaz ***Parse*** que debe ser implementada por cada uno de los complementos que se encargaran de parsear el contenido de los documentos rastreados. Estas clases son ***ImageParser*** y ***GeneralParser*** las que se auxiliaran de las clases ***ImageMetadataExtractor*** *y* ***GeneralMetadataExtractor*** para poder hacer la extracción de los metadatos, además se hace uso de las clases ***Logger*** y ***Configuration*** para registrar mensajes de diferentes tipos y acceder a las configuraciones respectivamente.

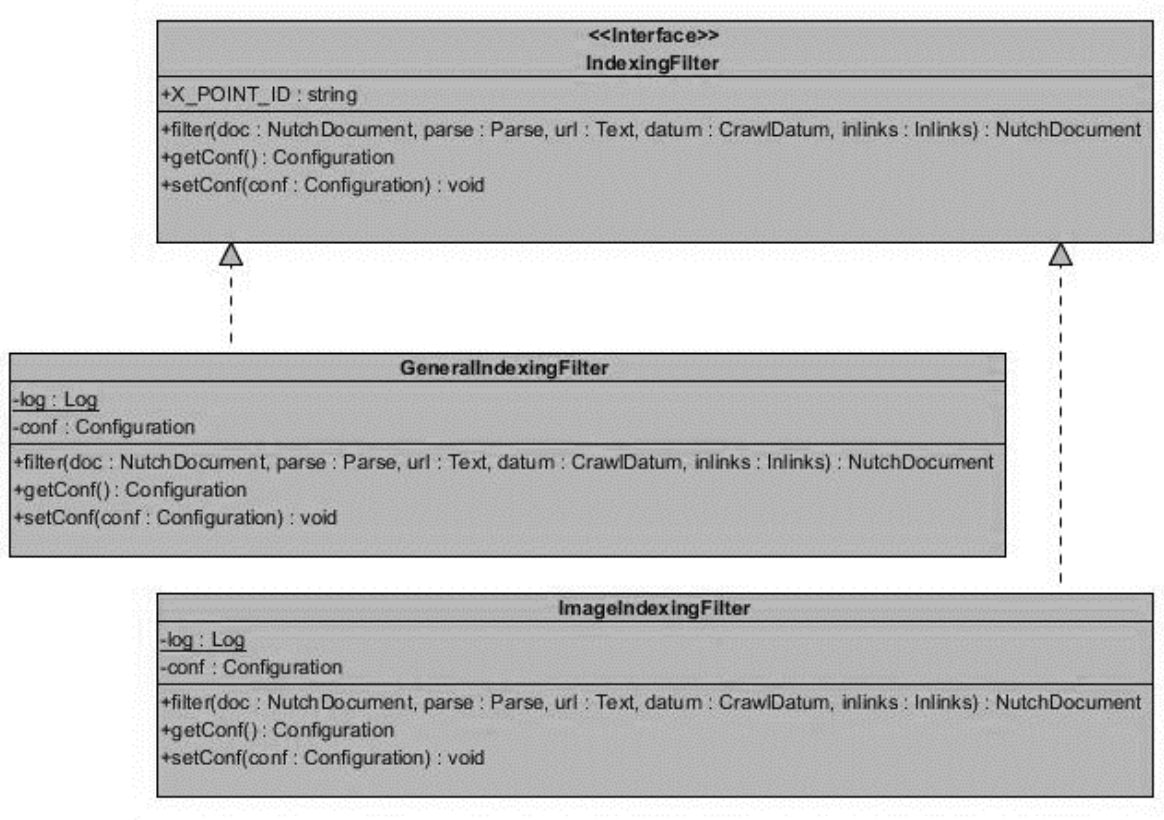

Figura 5. Diagrama de clases del diseño.

En el diagrama se muestra la interface ***IndexingFilter*** que es implementada por las clases ***GeneralIndexingFilter*** y ***ImageIndexingFilter***, las encargadas de asociar los diferentes metadatos a cada uno de los documentos, tanto para el core general como el de las imágenes.

2.6 Patrones de diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de *software* y otros ámbitos referentes al diseño de interacción o interfaces. En el modelo de diseño del subsistema se aplican patrones de asignación de responsabilidades (GRASP[14]) y patrones (GoF[15]), los cuales favorecen un mejor ordenamiento, estructuración y entendimiento del diseño. (Gamma, 2010).

[14] GRASP: Patrones Generales de *Software* para Asignar Responsabilidades, del inglés General *Responsibility Assignment Software Patterns.*

[15] GoF: Patrones de diseño, del inglés *Gang of Four*, nombre con el que se conoce comúnmente a los autores del libro *Desing Patterns*.

Patrones GRASP

Controlador

Este patrón tiene como objetivo asignar la responsabilidad a una clase de recibir o manejar un mensaje de evento del sistema generado por un actor externo, por lo general a través de una interfaz gráfica de usuario a la que accede un usuario para realizar ciertas operaciones en el sistema (Larman, 2013).

El siguiente fragmento indica que el método ***getParse*** responde a una solicitud del rastreador para parsear el contenido, a partir de esta clase coordina un conjunto de objetos para lograr la responsabilidad.

```
34      @Override
35      public ParseResult getParse(Content content) {
36
37        URL base;
38        try {
39          base = new URL(content.getBaseUrl());
40        } catch (MalformedURLException e) {
41          return createParseStatus(e).getEmptyParseResult(content.getUrl(), getConf());
42        }
43        String urlDocument = content.getUrl();
44
45        ParseResult parseResult = createParseStatus(ParseStatus.FAILED).getEmptyParseResult
```

Figura 6. Ejemplo del patrón Controlador

Experto en información

Este patrón plantea que se debe asignar una responsabilidad al experto en información, en otras palabras, a la clase que cuenta con los datos necesarios para cumplir la responsabilidad. De esta forma, se conserva el encapsulamiento de la información, puesto que los objetos ejecutan las tareas que le corresponden de acuerdo a la información que poseen, lo que da lugar a sistemas más robustos y fáciles de mantener (Larman, 2013). El siguiente fragmento indica que el método ***get*** se asigna a la clase ***Configuration*** por experto en información, ya que él es el que conoce cómo se almacena cada uno de los documentos de la configuración.

```
45        this.conf = conf;
46
47        MIME = new MimeUtil(conf);
48
49        documents_allowed = this.conf.get("extentions.general.document.allowed");
50        if(documents_allowed == null){
51          documents_allowed = "(pdf|doc|txt)";
52        }
53    }
```

Figura 7. Ejemplo del patrón Experto en información

Bajo Acoplamiento

El patrón bajo acoplamiento impulsa la asignación de responsabilidades de manera que su localización no incremente el acoplamiento hasta un nivel que lleve a los resultados negativos que puede producir un acoplamiento alto (Larman, 2013). En otras palabras, que una clase no dependa de muchas otras, lo cual potencia la reutilización y disminuye la dependencia entre estas. Este patrón se evidencia en el código que se muestra en el patrón Controlador (*ver figura 6*).

Alta cohesión

En la perspectiva del diseño orientado a objetos, la cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. El reto fundamental del patrón “Alta cohesión” es mantener la complejidad de una clase dentro de los límites manejables para que sea fácil de comprender, reutilizar y conservar (Larman, 2013).Este es el responsable de registrar todos los mensajes de error.

```
65        } catch (Exception ex) {
66          LOG.error("GEN-PARSER-ERROR: [" + ex.toString() + "]");
67          StackTraceElement[] stacks = ex.getStackTrace();
68          for (StackTraceElement stack : stacks) {
69            LOG.error("GEN-PARSER-ERROR: TRACK [" + stack.toString() + "]");
70          }
71        }
```

Figura 8. Ejemplo del patrón Alta cohesión

Creador

La instanciación de una clase es una de las actividades fundamentales en un sistema orientado a objetos. Este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos, con lo que se logra menos dependencia y mayores oportunidades de reutilización de código (Larman, 2004). En el

siguiente fragmento se muestra la clase ***ParseStatus*** con dos parámetros diferentes, por tanto, cada vez que la clase necesite crear una instancia llama a estos métodos.

```
26  private ParseStatus createParseStatus(MalformedURLException e){
27    return new ParseStatus(e);
28  }
29
30  private ParseStatus createParseStatus(int majorCode){
31    return new ParseStatus(majorCode);
32  }
```

Figura 9. Ejemplo del patrón Creador

2.7 Modelo de datos

El modelo físico de la base de datos, permite la identificación de los objetos primarios que va a procesar el sistema, la composición y atributos de los mismos. Además de dónde se encuentran almacenados actualmente dichos objetos, la relación entre ellos y los procesos que los transforman (Pressman, 2005). A continuación, se muestra el modelo de datos para el sistema que se propone desarrollar.

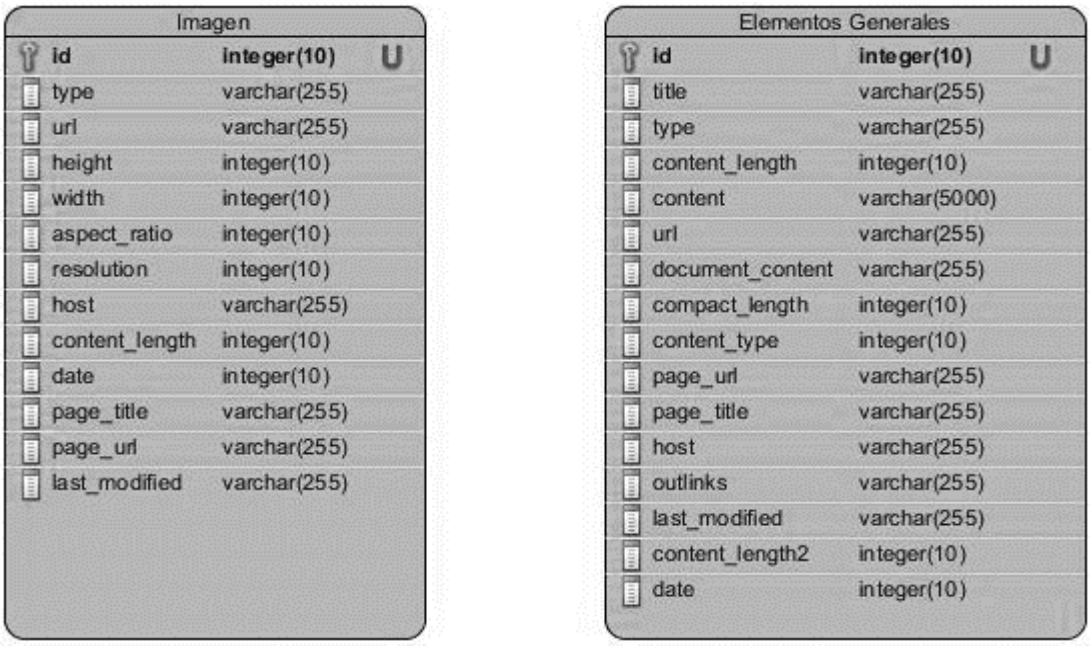

Figura 10. Diagrama del modelo de datos.

En este modelo de datos se representan los metadatos que fueron extraídos de los documentos rastreados e indexados en los core de imágenes y de elementos generales, como es en el caso del identificador (id),

el título (*title*), ancho y alto (*width* y *height* respectivamente), proporción (*aspect_ratio*), nombre que posee el recurso (*file_name*), fecha (*file_date*), tamaño del contenido (*content_length*), resolución (*piexel_resolution*), así como el contenido (*content*).

2.8 Modelo de despliegue

Un diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de *hardware* (nodos) y muestra cómo los elementos y artefactos del *software* se relacionan en esos nodos (SparxSystems, 2014).

El diagrama de despliegue se utiliza para mostrar la estructura física del sistema, incluyendo las relaciones entre el *hardware* y el *software* que se despliega, estas relaciones son representadas por los protocolos de comunicación que se utilizan para acceder a cada uno. En el diagrama se muestra el servidor de Nutch, encargado de hacer uso de los *plugins* responsabilizados de la estandarización de documentos y de enviar los resultados al servidor Solr mediante el protocolo HTTP.

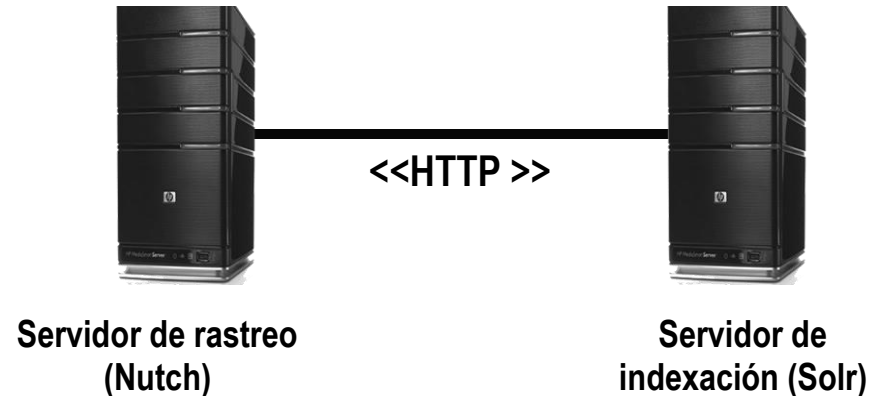

Figura 11. Diagrama del modelo de despliegue.

2.9 Conclusiones del capítulo

Durante el análisis y diseño del subsistema de estandarización de documentos para el buscador Orión, la representación y descripción de los artefactos que se generaron garantizaron la comprensión de los flujos de trabajo presentes en los componentes Nutch y Solr. Además, se listaron los requisitos funcionales y no funcionales que debe cumplir el *software* para satisfacer las necesidades del cliente, lo que es importante para las próximas etapas de desarrollo. La modelación de los diferentes diagramas de clases del diseño, el modelo físico de la base de datos y el diagrama de despliegue, junto con el eficiente levantamiento de los requisitos, ayudaron al equipo de desarrollo a entender con más facilidad el problema planteado. Los patrones de diseño escogidos, propiciaron una arquitectura sólida y robusta al sistema.

Capítulo 3: Implementación y pruebas del Subsistema de estandarización de documentos para el buscador Orión.

3.1 Introducción

En el presente capítulo se detallará el funcionamiento del sistema propuesto para dar solución a la problemática planteada. Durante la implementación se lograrán las tareas de programación definidas para darle cumplimiento al desarrollo de la solución propuesta, además se presentarán aspectos como estándares de codificación utilizados durante la misma.

Durante la etapa de prueba se persigue encontrar errores cometidos al realizar el diseño y construcción de un producto de *software*. En este se detallan los principales resultados de la realización de las pruebas.

3.2 Modelo de componentes que integran el subsistema

El modelo de componentes permite visualizar la estructura de alto nivel del sistema y el comportamiento del servicio que esos componentes proporcionan. Puntualiza que cada componente debe ser tratado como una unidad de composición independiente e indispensable dentro del sistema, y que puede contraer relaciones de dependencia con otros componentes (Hernández, 2013).

3.3 Diagrama de componentes

El diagrama de componentes muestra la relación entre los componentes del *software*, sus dependencias, su comunicación y su ubicación.

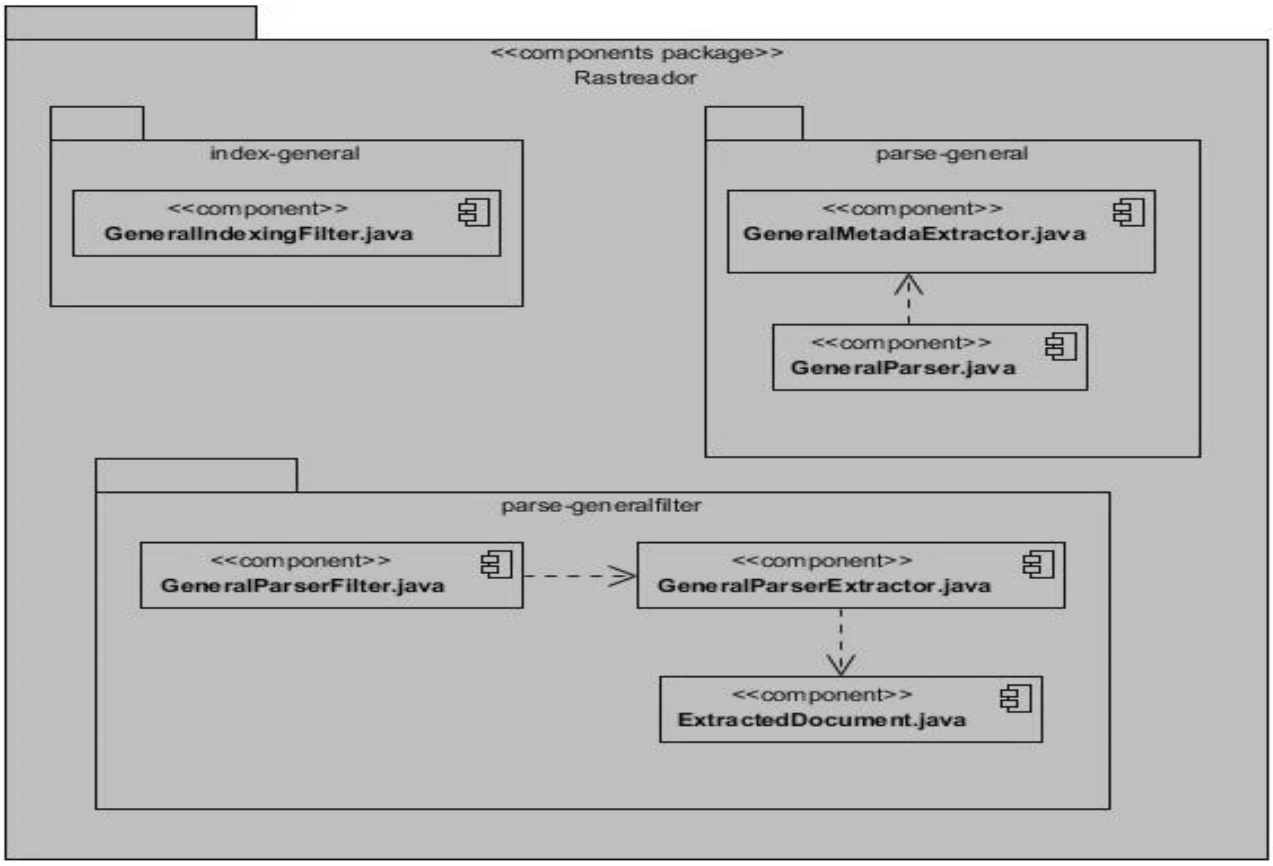

Figura 12. Diagrama de componentes del paquete Rastreador.

El subpaquete index-general incluye un componente:

GeneralIndexingFilter.java: Componente físico que incluye la clase ***GeneralIndexingFilter***, la cual tiene la responsabilidad de decidir cuáles metadatos y cómo serán indexados.

El subpaquete parse-general incluye dos componentes, los cuales son:

GeneralMetadataExtractor.java**:** Es el fichero que contiene la clase ***GeneralMetadataExtractor***, la cual es utilizada para la extracción de los metadatos de los documentos generales.

GeneralParser.java**:** Representa al fichero que contiene la clase ***GeneralParser***, cuya función principal es interactuar con el rastreador y enviarle los metadatos extraídos de un documento.

El subpaquete *parse-generalfilter* está compuesto por tres componentes, los cuales son:

GeneralParserFilter.java**:** Representa al fichero que contiene la clase ***GeneralParserFilter***, la cual

contiene los procedimientos necesarios para gestionar la información de los metadatos de los documentos encontrados en una página web.

GeneralParserExtractor.java: Representa al fichero que se utiliza para identificar los documentos en una página web y obtener la información asociada a ellos.

ExtractedDocument.java: Representa al fichero que contiene la entidad que encapsula un conjunto de informaciones asociadas al documento general y que fueron extraídas por el ***GeneralParserExtractor****.*

El paquete Indexador incluye dos componentes, los cuales son ficheros de configuración que garantizan el correcto almacenamiento de los datos, así como la comunicación con el resto de los componentes de la arquitectura del sistema.

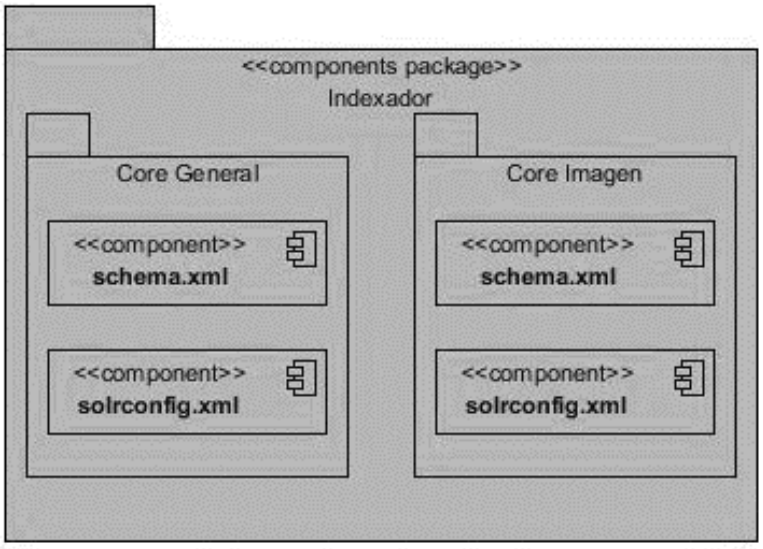

Figura 13. Diagrama de componentes del paquete Indexador.

schema.xml: Contiene la declaración de los campos o atributos de un documento, así como los filtros y tokenizadores[16] que se le aplican a dichos campos, los cuales realizan procesamiento de los datos en el momento de realizar la indexación y la búsqueda.

solrconfig.xml*:* Fichero de configuración principal de Solr.

Core General: Núcleo(core) que Indexa todos los metadatos asociado a los documentos generales.

16 Tokenizadores: Componentes de Solr encargados de dividir los datos en unidades léxicas llamadas tokens.

Core Imagen: Núcleo(core) que Indexa todos los metadatos asociado a las imágenes.

3.4 Estándares de codificación utilizados

Los estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. Seguir un determinado estilo de codificación permite a los programadores revisar, mantener y actualizar el código de una manera sencilla y ordenada, evitando que incurran en errores y malas prácticas que dificultan la comprensión de las líneas de código.

Para facilitar el entendimiento del código y precisar un modelo a seguir, se adoptaron determinados estándares de codificación que a continuación se describen, agrupados por los aspectos en los que fueron utilizados.

Líneas en blanco

Una de las mejores prácticas de codificación que ayuda a garantizar la capacidad de mantenimiento de los sistemas es la inclusión de líneas en blanco entre secciones de código, funciones, clases, sentencias, declaraciones y comentarios. En este trabajo se ha definido el uso de líneas en blanco para separar funciones de una misma clase, así como secciones de código dentro de una misma función. Es prudente señalar que este estilo agrega más líneas de código al programa, pero a su vez se gana en legibilidad y limpieza en el código.

```
50
51      /**
52       * Get the MimeTypes resolver instance.
53       */
54      private MimeUtil MIME;
55      private final Tika tika = new Tika();
56
57      /**
58       * Map for mime-type substitution
59       */
60      private HashMap<String, String> mimeMap = null;
61      private boolean mapMimes = false;
```

Figura 11. Estándar de codificación: Líneas en blanco

Fuente: Elaboración propia

Comentarios

Los comentarios en el código representan la documentación interna más precisa de un *software*. Estos garantizan el entendimiento de lo que realmente realiza un determinado bloque de código, evitando confusiones y agilizando considerablemente las tareas de revisión y mantenimiento.

```
50
51      /**
52       * Get the MimeTypes resolver instance.
53       */
54      private MimeUtil MIME;
55      private final Tika tika = new Tika();
56
57      /**
58       * Map for mime-type substitution
59       */
```

Figura 12. Estándar de codificación: Comentarios

Fuente: Elaboración propia

Llaves

En cuanto a la ubicación de estos operadores de delimitación, existen diversos criterios en los diferentes lenguajes de programación que lo usan para la definición de bloques de código. En este trabajo 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, switch, foreach*. Las llaves de cierre se colocarán solitarias en la línea siguiente a la última línea dentro del bloque e indentadas al nivel de la línea cabecera del bloque.

```
108             }
109           }
110
111           for (String key : str_keys) {
112             String value = metadata.get(key);
113             if (value != null) {
114               doc.add(key, value);
115               LOG.info("ANIETSY-HTM-STR > " + key + " -> " + value);
116             }
117           }
118         }
```

Figura 13. Estándar de codificación: Llaves

Fuente: Elaboración propia

Identificadores

Para la definición del nombre de las clases, funciones, variables y constantes se tuvo en cuenta el estilo *lowerCamelCase*. Este estilo establece que la separación entre palabras internas de los identificadores deberá realizarse escribiendo la letra inicial en minúscula, a excepción de la primera palabra. Además, no deberá colocarse ningún carácter especial entre palabras de los identificadores. En el caso del estilo *UpperCamelCase*, es cuando la primera letra de cada una de las palabras es mayúscula.

Ejemplos de uso:

- Clase: *public class **GeneralParser** implements org.apache.nutch.parse.Parser*
- Función: *public ParseResult **getParse** (Content content)*
- Variable: *ParseResult parseResult = new ParseResult(content.getUrl());*
- Constante: *private static final String **TIKA_CONFIG_FILE** = "tika.config.file*";

Identación

Una de las prácticas más recomendadas para la implementación consiste en la indentación del código. Esta costumbre 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. Gracias al uso de NetBeans como IDE de desarrollo, los espacios de indentación son ajustados automáticamente, permitiendo a los programadores enfocarse en otras funciones de mayor importancia. Por lo tanto, la unidad de indentación de bloques de sentencias son de 2 espacios.

```
234        if (outlinks != null) {
235          Set<String> hosts = new HashSet<>();
236
237          for (Outlink outlink : outlinks) {
238            try {
239              String linkUrl = outlink.getToUrl();
240              String outHost = new URL(linkUrl).getHost().toLowerCase();
241
```

Figura 14. Estándar de codificación: Identación

Fuente: Elaboración propia

3.5 Validación del subsistema

En la ingeniería de *software* es el proceso de revisión que verifica que el sistema producido cumple con las especificaciones y que logra su objetivo, se trata de evaluar el sistema o parte de este durante o al final del desarrollo para determinar si satisface los requisitos iniciales (Hernández, 2016). A continuación, se detallan los tipos de pruebas aplicadas al subsistema de estandarización de documentos. Las mismas persiguen como objetivo fundamental, la detección de las no conformidades respecto a las funcionalidades de la aplicación, así como también la correcta integración entre los componentes de la arquitectura del sistema.

3.5.1 Pruebas funcionales

Las pruebas funcionales son aquellas que se aplican a un *software* determinado, con el objetivo de validar que las funcionalidades implementadas estén de acuerdo a las especificaciones de los requisitos definidos con anterioridad. A continuación, se muestran fragmentos de los casos de pruebas para algunos requisitos con prioridad alta del subsistema de estandarización de documentos para el buscador Orión.

Casos de pruebas

Los casos de prueba son artefactos diseñados para validar la funcionalidad de cada uno de las historias de usuario definidos para la aplicación. Para lograr este propósito se realizan diferentes entradas de datos, y luego se lleva a cabo una evaluación de los resultados obtenidos. De acuerdo a su estructura las entradas de los datos de la tabla contienen los valores V, I, o N/A. V indica válido, I indica inválido, y N/A que no aplica a un valor del dato.

Se realizaron todos los casos de pruebas por cada una de las historias de usuario. A continuación, se muestra un fragmento de la historia de usuario “Identificar el metadato contenido del documento para su almacenamiento en el core de los elementos generales”. El resto de los diseños de casos de prueba se pueden encontrar en los anexos.

Caso de Prueba 1: SC RF1_ Identificar el metadato contenido del documento para su almacenamiento en el core de los elementos generales.

Descripción general: Luego de rastreado el documento, identificar el contenido texto para el core de elementos generales.

Escenario	Descripción	Contenido	Respuesta del sistema	Flujo central
EC 1.1 Identificar el metadato contenido del documento existente para su almacenamiento en el core de los elementos generales.	El sistema identifica el contenido de forma correcta.	V Contenido existente	El sistema a partir de los documentos rastreados identifica el contenido correspondiente.	A partir de los documentos rastreados se extraen los metadatos dentro de los que se incluye el contenido.
EC 1.2 Identificar el metadato contenido del documento no existente para su almacenamiento el core de los elementos generales.	El sistema no identifica el contenido de forma correcta.	I Contenido no existente.	El sistema a partir de los documentos rastreados no identifica el contenido correspondiente.	Se excluye el documento.

Descripción de las variables

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Contenido	campo de texto	No	Identifica la información que presenta el documento que se analiza.

Caso de Prueba 2: SC RF 2_Identificar el metadato título del documento para su almacenamiento en el core de los elementos generales.

Descripción: Luego de rastreado el documento, identificar el título para el core de elementos generales.

Escenario	Descripción	Título	Respuesta del sistema	Flujo central
EC 1.1 Identificar el metadato título del documento existente para su almacenamiento en el core de los elementos generales.	El sistema identifica los títulos de forma correcta.	V Título existente	El sistema a partir de los documentos rastreados identifica los títulos correspondientes.	A partir del contenido del documento se extraen de el mismo todos los metadatos dentro de los que se incluye el título.
EC 1.2 Identificar el metadato título no existente para su almacenamiento en el core de los elementos generales.	El sistema no identifica los títulos de forma correcta.	I Título no existente.	El sistema a partir de los documentos rastreados no identifica los títulos correspondientes.	Se asigna la URL al título.

Descripción de las variables

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
2	Título	campo de texto	No	Identifica el título del documento que se analiza.

Caso de Prueba 3: SC RF3_ Identificar el metadato fecha de última actualización del documento para su almacenamiento en el core de las imágenes.

Descripción general: Luego de ser rastreada la imagen, identificar la fecha última de modificación de la imagen.

Escenario	Descripción	Fecha	Respuesta del sistema	Flujo central
EC 1.1	El sistema	V	El sistema a partir de los	A partir del contenido

Identificar el metadato de fecha última de actualización del documento existente para el core de elementos generales.	identifica la fecha de forma correcta.	Fecha existente	documentos rastreados identifica las fechas correspondientes.	del documento se extraen de el mismo todos los metadatos dentro de los que se incluyen la fecha última de actualización del documento.
EC 1.2 Identificar el metadato de fecha última de actualización del documento no existente para el core de elementos generales.	El sistema no identifica la fecha de forma correcta.	I Fecha no existente.	El sistema a partir de los documentos rastreados no identifica las fechas correspondientes.	Se asume como valor de este metadato el de la fecha de rastreo.

Descripción de las variables

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
3	Fecha de última de actualización del documento	campo de texto	Si	Identifica la fecha última en que fue modificado el documento analizado.

Resultado de las pruebas funcionales

Las pruebas funcionales se realizaron en tres iteraciones a medida que avanzaba el desarrollo de la solución, guiando la calidad y determinando en cada momento las condiciones para continuar avanzando con el ciclo de desarrollo. A continuación, se muestran los resultados de las pruebas realizadas en cada iteración, las que arrojaron un total de 10 no conformidades que fueron corregidas satisfactoriamente, en

los que se detectaron errores ortográficos, la identificación incorrecta del tipo de documento (html, pdf, word, etc), la captura de los metadatos y la identificación de las URL.

No conformidades	Primera iteración	Segunda iteración	Tercera iteración
Detectadas	6	4	0
Resueltas	6	4	0
Pendientes	0	0	0

Tabla 7. Cantidad de no conformidades por cada iteración de las pruebas funcionales.

En la figura 14 se puede apreciar de forma más ilustrativa el comportamiento de las no conformidades por cada iteración de las pruebas funcionales realizadas.

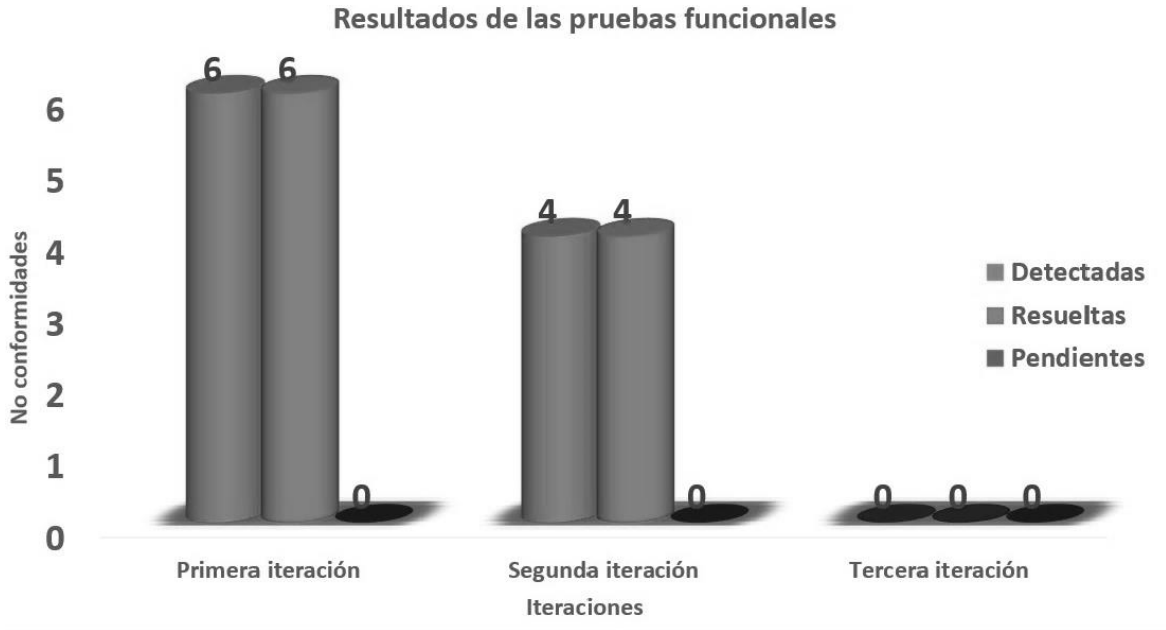

Figura 14. Resultados de las pruebas funcionales

La gráfica muestra las iteraciones que se llevaron a cabo en el desarrollo de las pruebas. En una primera iteración se identificaron 6 no conformidades, las que se resolvieron en su totalidad; en la segunda iteración se detectaron 4 no conformidades y de igual forma fueron resueltas y para una tercera iteración no se detectaron no conformidades.

3.5.2 Pruebas de integración

El proceso de integración del sistema implica construirlo a partir de sus componentes y probar el sistema

resultante para encontrar problemas que pueden surgir debido a la integración. La integración del sistema implica identificar grupos de componentes que proporcionan alguna funcionalidad del sistema e integrar estos añadiendo código para hacer que funcione conjuntamente (Sommerville, 2005).

El motor de búsqueda cubano Orión, cuenta con una estructura que facilita la integración de nuevos subsistemas para incrementar sus funcionalidades. Por tal motivo, se decidió realizar pruebas de integración descendentes. Las cuales, consisten en desarrollar la infraestructura del sistema en su totalidad y luego añadirle los componentes funcionales.

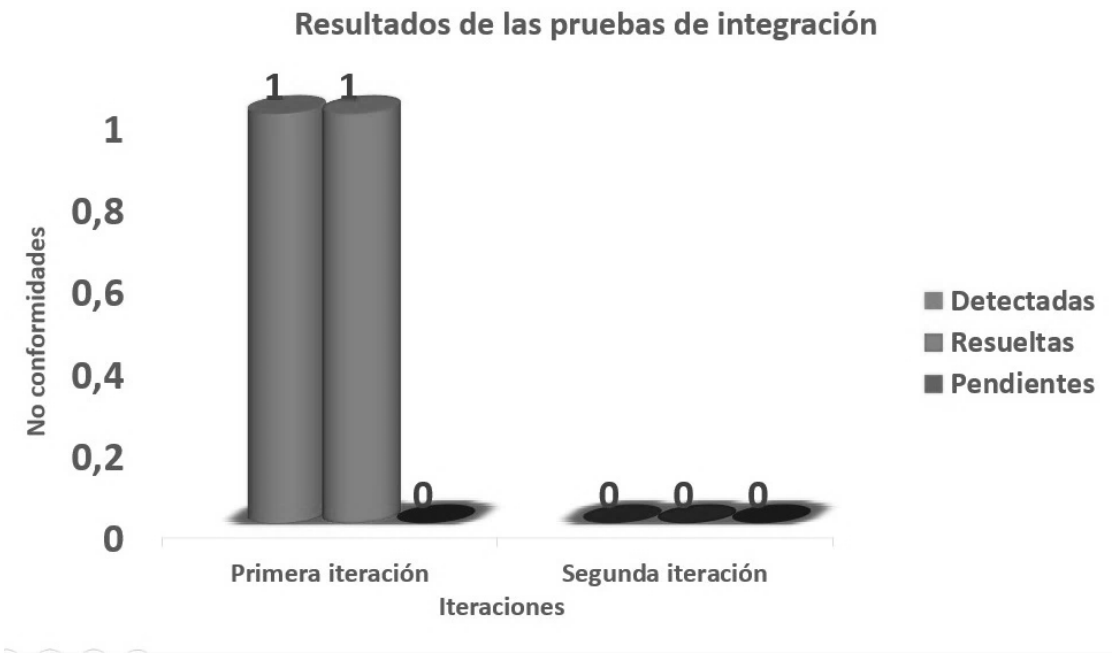

Figura 15: Resultados de las pruebas de integración.

Las pruebas de integración permitieron identificar en la primera iteración que en los esquemas de configuración de Nutch y Solr existían campos con un mismo nombre, pero diferentes tipos de datos. En una segunda iteración no se detectaron no conformidades. Luego de solucionar la no conformidad, el subsistema de estandarización de documentos se integró correctamente con el componente de Solr.

3.6 Validación de la hipótesis de investigación

El objetivo principal de la verificación y la validación es mejorar la calidad del producto de *software*. En el proceso de verificación se ejecutan pruebas al sistema a fin de detectar la mayor cantidad de errores, mientras que en la validación se pretende corregir dichos fallos a con el objetivo de lograr el correcto funcionamiento del *software*.

3.6.1 Pruebas de aceptación

El uso de cualquier producto de *software* tiene que estar justificado por las ventajas que ofrece. Sin embargo, antes de empezar a usarlo es muy difícil determinar si sus ventajas realmente justifican su uso. El mejor instrumento para esta determinación es la llamada “prueba de aceptación”. En esta prueba se evalúa el grado de calidad del *software* con relación a todos los aspectos relevantes para que el uso del producto se justifique.

Técnica de V.A Iadov

Para evaluar la satisfacción con el proceso de evaluación y caracterización del desempeño efectuado, se recogió el Criterio de usuarios, utilizando la Técnica de Iadov. Mediante dicha técnica se determina el nivel de satisfacción individual y grupal a partir de una encuesta elaborada según las exigencias pertinentes, que, en la presente investigación, se aplicó a 10 especialistas del centro que trabajan en el área de recuperación de información.

Encuesta inicial de la investigación

Especialista, le invito a responder el siguiente cuestionario con el fin de conseguir su colaboración en la presente investigación, solicito que exprese en sus respuestas criterios verídicos que guíen a la autora del trabajo. Marque con una x en una sola opción y en el caso de la 5 responda brevemente. Por el tiempo brindado, muchas gracias.

1. ¿Considera importante la estandarización de documento para mejorar la calidad de los resultados de búsqueda?

Sí ___ No ___ No sé___

2. ¿Los resultados que se muestran satisfacen a las necesidades existentes de acuerdo al criterio de búsqueda?

Sí ___ No ___ No sé___

3. ¿Considera usted factible estandarizar los criterios que permitan aumentar la calidad de los resultados de búsqueda?

Sí ___ No ___ No sé___

4. Luego de haber mostrado los resultados de la estandarización de documentos refleje en qué medida le gusta la solución desarrollada.

__Me gusta mucho __Me disgusta más de lo que me gusta

__Me gusta más de lo que me disgusta __No me gusta nada

__Me da lo mismo __No sé decir

5. ¿Qué opina usted acerca de los beneficios que traería para la universidad disponer del motor de búsqueda que implementa la estandarización de documento con el objetivo de aumentar la calidad de los resultados?

La técnica de Iadov constituye una vía indirecta para el estudio de la satisfacción, ya que los criterios que se utilizan se fundamentan en las relaciones que se establecen entre las tres preguntas cerradas, que se intercalan dentro de un cuestionario y cuya relación el encuestado desconoce. En este caso, de la encuesta, son las preguntas 2, 3 y 4 se relacionan a través de lo que se denomina el "Cuadro Lógico de Iadov" que se muestra en la tabla 8.

4. Luego de haber mostrado los resultados de la estandarización de documentos refleje en qué medida le gusta la solución desarrollada.	2. ¿Los resultados que se muestran satisfacen a las necesidades existente de acuerdo al criterio de búsqueda?								
	No			**No sé**			**Sí**		
	3. ¿Considera usted factible estandarizar los criterios que permitan aumentar la calidad de los resultados de búsqueda?								
	Sí	**No sé**	**No**	**Sí**	**No sé**	**No**	**Sí**	**No sé**	**No**
Me gusta mucho	1	2	6	2	2	6	6	6	6
Me gusta más de lo que me disgusta	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	3	4
No me gusta nada	6	6	6	6	4	4	6	6	5
No sé decir	2	3	6	3	3	3	6	6	4

Tabla 8. Cuadro Lógico de Iadov.

La forma de utilizar la tabla es la siguiente: Cada encuestado recibe una evaluación individual en dependencia de las respuestas que dé a las preguntas cerradas. Para facilitar el procesamiento posterior, en el diseño de la encuesta se debe tener en cuenta que a estas preguntas sólo se responda de la forma prevista en el cuadro lógico de Iadov. En nuestro caso, las respuestas a las preguntas 2 y 3 pueden ser SI, NO, NO SÉ, y a las preguntas 4, “Me gusta mucho”, “Me gusta más de lo que me disgusta”, “Me da lo mismo”, “Me disgusta más de lo que me gusta”, “No me gusta nada”, o “No sé qué decir”.

Para obtener el índice de satisfacción grupal (ISG) se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y – 1. El número resultante de la interrelación de las tres preguntas nos indica la posición de cada encuestado en la siguiente escala de satisfacción:

1. Clara satisfacción +1
2. Más satisfecho que insatisfecho 0.5
3. No definido y contradictorio 0
4. Más insatisfecho que satisfecho -0.5
5. Clara insatisfacción -1

La satisfacción grupal se calcula por la siguiente fórmula:

$$ISG = \frac{\mathrm{A}\,(+1) + \mathrm{B}\,(+0.5) + \mathrm{C}\,(0) + \mathrm{D}\,(-0.5) + \mathrm{E}\,(-1)}{\mathrm{N}}$$

En esta fórmula A, B, C, D, E, representan la cantidad de encuestados colocados respectivamente en las posiciones de satisfacción 1; 2; 3 ó 6; 4; 5 y donde N representa la cantidad total de encuestados

$$ISG = \frac{9(+1) + 1(+0.5) + 0\,(0) + 0\,(-0.5) + 0\,(-1)}{10}$$

ISG= 0.9

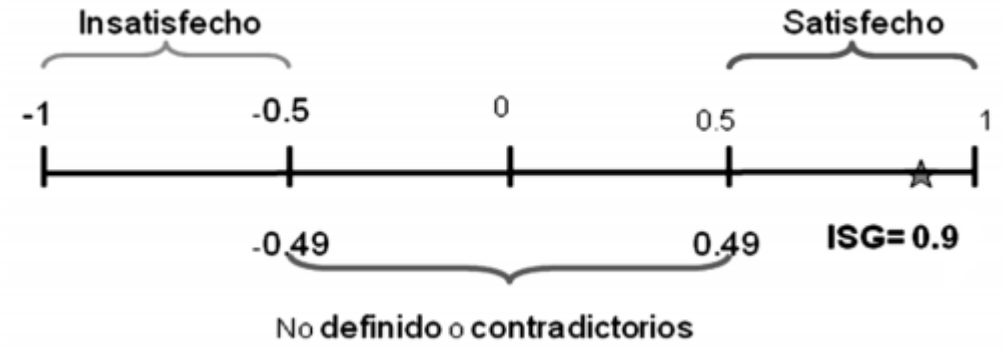

Figura 16. Ubicación del Índice de Satisfacción Grupal con el Sistema de Indicadores propuesto.

Fuente: Elaboración propia

Los valores de ISG 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.

Categorías grupales de satisfacción	N=10	Escala
Clara satisfacción	9	A
Más insatisfecho que satisfecho	1	B
No definido	0	C
Más satisfecho que insatisfecho	0	D
Máximo de satisfacción	0	E
Contradictorio	0	C

Tabla 9. Resultados de satisfacción.

El proceso de validación mediante la Técnica de Iadov de la consulta a los usuarios, donde se ha implementado el sistema propuesto, confirmó su factibilidad de uso, expresado cuantitativamente en el alto Índice de Satisfacción Grupal (ISG = 0,9) y cualitativamente en los criterios emitidos donde evidencia la satisfacción de prever, diseñar y medir el impacto en el centro de investigación CIDI, lo que refleja aceptación de la propuesta y un reconocimiento a su utilidad.

3.7 Conclusiones del capítulo

En el presente capítulo se identificó la estructura y las relaciones que existen entre los diferentes componentes empleados en la implementación del subsistema de estandarización de documentos a partir de una representación y descripción del diagrama de componentes. La utilización de los estándares de codificación para la implementación de la solución posibilitó adoptar una estructura que facilita la comunicación mediante el código legible, fácil de comprender y reutilizable. La realización de pruebas funcionales y no funcionales condujo a la detección de errores en el sistema, lo cual permitió su solución oportuna, posibilitando un mejor resultado en términos calidad avalado por el cumplimiento de los requisitos propuestos. La calidad de los documentos almacenados para el componente de estandarización de documentos fue avalada por la técnica Iadov, que mostró un nivel alto de satisfacción.

Conclusiones generales

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

- La sistematización teórico-metodológico de los principales conceptos asociados a la investigación científica, permitió la conceptualización de la estandarización de documentos, los tipos de estándares, los buscadores y los SRI.
- La fundamentación del marco teórico de la investigación científica, posibilitó la selección de la metodología, herramientas y tecnologías a utilizar para la implementación de la solución propuesta, las que se enmarcan en: Nutch como mecanismo para rastrear la Web, Solr como servidor de indexación, JSON con formato de intercambio de datos, XML como lenguaje para los ficheros de configuración en Solr, Java para la implementación del plugin en Nutch y Visual Paradigm como herramienta para el modelado.
- En el análisis documental se evidencia que cinco de los metadatos más representativos de un documento web son: título, tipo, fecha de actualización, url y contenido.
- La gestión de la información y el conocimiento integrada a diversas áreas del saber cómo son: la ingeniería y gestión de *software*, base de datos, programación, entre otras, permitió el análisis, diseño, elaboración e implementación del subsistema de estandarización de documentos para el buscador Orión.
- El objetivo de la investigación fue cumplido, porque el subsistema de estandarización de los documentos que se gestionan en el buscador Orión, mejoró la calidad de los documentos almacenados.
- La validación del correcto funcionamiento de la solución propuesta se realizó a través de las pruebas de *software* y la técnica de Iadov, logrando así un producto que almacena documentos de mejor calidad conforme a las necesidades del sistema.

Recomendaciones

Una vez concluida la investigación y el desarrollo de la propuesta de solución, se recomienda:

- Incorporar otro componente que permita la identificación correcta de metadatos relevantes como: título y resumen en documentos ofimáticos.

Referencias Bibliográficas

Ambysoft. 2014. *agilemodeling.* [En línea] 2014. [Citado el: 3 de diciembre de 2015.] http://www.agilemodeling.com/essays/simpleTools.htm#SelectingCASE.

Acosta, David. 2011. Rational. *Rational.* [En línea] 3 de Marzo de 2011. [Citado el: 4 de Diciembre de 2016.] http://www.rational.com.ar/herramienta/roseenterprise.html.

Alcaide, Andrea Osorio. 2014. Recursos Humanos y responsabilida SOCIAL CORPORATIVA. [En línea] 4 de Junio de 2014. [Citado el: 15 de Mayo de 2017.] https://recursoshumanosyresponsabilidadsocialcorporativa.wordpress.com.

Anny Flores, Desiree Andérico. 2012. *Desarrollo de Sistema Transaccional para Manejo de Reportes del Proceso de Inyección de Químicos en la Superintendencia de tratamiento de calidad de fluidos PDVSA distrito Morichal,Edo Monogas.Universidad de Orient .* Venezuela : s.n., 2012.

Apache, Solr. 2015. Apache Software Fundation. [En línea] 2015. [Citado el: 4 de Diciembre de 2016.] http://httpd.apache.org/..

Baisley, B. 2006. *Unified Modeling Language Infraestructure.* s.l. : Pearson, 2006.

Balarezo. 2013. *Metodologías Ágiles,Programación Extrema XP.* s.l. : Facultad de Ciencias Físicas y Matemáticas., 2013.

Brito Acuña, Karenny. 2009. Selección de metodologías de desarrollo para aplicaciones web en la facultad de infórmatica de la Universidad de Cienfuegos. *Selección de metodologías de desarrollo para aplicaciones web en la facultad de infórmatica de la Universidad de Cienfuegos.* [En línea] 2009. [Citado el: 29 de Noviembre de 2016.] http://www.eumed.net.

Camiño, R.R. 2003. Motores de búsqueda sobre salud en internet. *Motores de búsqueda sobre salud en internet.* [En línea] 2003. [Citado el: 7 de Diciembre de 2016.] http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352003000500002..

Cardelas, Claudia Maricela Martinez. 2013. ¿Que es un motor de búsqueda? *¿Que es un motor de*

búsqueda? [En línea] 7 de Enero de 2013. [Citado el: 29 de Noviembre de 2016.] http://mcclaudiamari.blogspot.com.

Castellanos, Alejandro Munguía. 2012. Formatos digitales. 2012.

Caules, Cecilio Álvarez. 2013. El concepto de ClassLoader - Arquitectura Java. *El concepto de ClassLoader - Arquitectura Java.* [En línea] Octubre de 16 de 2013. [Citado el: 2 de diciembre de 2016.] http://www.arquitecturajava.com/el-concepto-de-classloader.

Clarenc, Claudio Ariel. 2011. Deseo aprender. [En línea] 28 de Mayo de 2011. [Citado el: 30 de Mayo de 2017.] https://books.google.com.cu.

Codina, L. 2008. Posicionamiento Web para comunicadores: Análisisy Métodos. *Posicionamiento Web para comunicadores: Análisisy Métodos.* [En línea] 2008. [Citado el: 7 de Diciembre de 2016.] http://documents.mx/documents/lluis-codina-upfidec-abril-2008-posicionamiento-web-tercera-parte-analisis-y-metodos.html.

Diéguez, Rodolfo A. 2012. Estandarizacion y Normalización. *Estandarizacion y Normalización.* [En línea] 2012. [Citado el: 7 de Diciembre de 2016.] http://www.monografias.com.

Enrique, Alameda Dr. Manuael. 2013. Estandarización de sitios web. *Estandarización de sitios web.* [En línea] 15 de mayo de 2013. [Citado el: 29 de Noviembre de 2016.] http://herramientas.presidencia.gob.sv/itiges/estandarizacion.

Fernández, Juan. 2011. El codigok. *El codigok.* [En línea] 14 de Mayo de 2011. [Citado el: 4 de Diciembre de 2016.] http://www.elcodigok.com.ar/2010/09/caracteristicas-de-un-excelente-entorno-de-desarrollo-integrado...

Fernández, Oscar Belmonte. 2005. *Introducción al lenguaje de.* 2005.

Gaitán, Jessica Daniela. 2014. LOM. *LOM.* [En línea] 2014. [Citado el: 9 de Mayo de 2017.] http://www.portalhuarpe.com.ar.

Gamma, Eric. 2010. Elements of Reusable Object Oriented Software. 2010.

García, Manuel. 2009. Manual para la creación de metadatos usando Dublin Core . [En línea] Diciembre de

2009. [Citado el: 10 de Mayo de 2017.] http://biblioestandares.bn.cl/sites/...bn.cl/.../MANUAL_METADATOS_BN%20_Oficial.pdf.

Giginio, Ruben. 2010. El Desarrollo Científico Tecnológico. *El Desarrollo Científico Tecnológico.* [En línea] Ruben Higinio Rivera Aguilera, julio de 2010. htp://www.monografias.com/.../desarrollo-cientifico-tecnologico.

Gónzalez, Lianet Cabrera. 2012. Extensión de Visual Paradigm for UML para el desarrollo dirigido por modelos de apliaciones de gestión de información. [En línea] 2012. [Citado el: 23 de Febrero de 2017.] https://www.redib.org/recursos/Record/oai_articulo983403-extensión-visual-paradigm-uml-desarrollo-dirigido-modelos-aplicaciónes-gestión-información.

Guevara, Yurisander. 2015. Diario de la juventud cubana. *Diario de la juventud cubana.* [En línea] 13 de Julio de 2015. [Citado el: 29 de Enero de 2017.] digital@juventudrebelde.cu.

Hernández, Carmen. 2011. Buscadores y Metabuscadores. [En línea] 18 de Noviembre de 2011. [Citado el: 25 de Febrero de 2017.]

Hernández, Javier. 2011. ¿Qué es Estándar en informática? . *¿Qué es Estándar en informática? .* [En línea] 2011. [Citado el: 7 de Diciembre de 2016.] https://es.answers.yahoo.com.

Hernández, Leovigilda. 2013. Modelo de implementación. [En línea] 1 de junio de 2013. [Citado el: 27 de Junio de 2017.] http://es.scribd.com/doc/7884665/Arquitectura-de-Software-II-Diagrama-de-Componentes.

Hernández, Lic.Rodrigo Santiago. 2016. Pruebas de software. 2016.

Krall, César. [En línea]

Lapuente, María Jesús Lamarca. 2013. Metadatos Dublin Core. *Metadatos Dublin Core.* [En línea] 8 de diciembre de 2013. [Citado el: 9 de Mayo de 2017.] www.hipertexto.info/documentos/dublin_core.html.

Larman. 2013. *Introducción al análisis y diseño orientado a objetos.* México : Prentice Hal.ISBN:970-1 7-0261-1, 2013.

Laureano, Prof. Dra. Ana Lilia. 2004. El lenguaje de programación Java. *El lenguaje de programación Java.* [En línea] Noviembre de 2004. [Citado el: 2 de Diciembre de 2016.] http://slideplayer.es/slide/3613519.

Leyva, Paúl Rodríguez. 2016. *Componentes y funcionalidades de un sistema de recuperación de la información.* 2016.

Maganto, Alejandra Sánchez. 2008. *Normas sobre metadatos.* 2008.

2009. Manual para la creacion de metadatos utilizando Dublin Core. [En línea] Diciembre de 2009. [Citado el: 10 de Mayo de 2017.] http://biblioestandares.bn.cl/sites/...bn.cl/.../MANUAL_METADATOS_BN%20_Oficial.pdf.

McGuinness, Deborah L. 2004. Lenguaje de Ontologías Web (OWL). [En línea] 10 de Febrero de 2004. [Citado el: 10 de Mayo de 2017.] https://www.w3.org/2007/09/OWL-Overview-es.html.

Miller, Michael. 2012. 2012. ISBN 978-0-7897-4365-7.

NetBeans. 2015. NetBeans IDE. *NetBeans IDE.* [En línea] 2015. [Citado el: 4 de Diciembre de 2016.] http://netbeans.org..

Nioche, Julien. 2013. Large Scale Crawling with Apache Nutch and Friends. [En línea] 2013. [Citado el: 2 de Junio de 2017.]

NutchWiki. 2015. Nutch Tutorial- Nutch Wiki. [En línea] 2015. [Citado el: 2017 de Febrero de 26.] http://wiki.apache.org/nutch/NutchTutorial..

Ochando, Prof. Dr. Manuel Blázquez. 2012. Sistemas de Recuperación e Internet . [En línea] 5 de Noviembre de 2012. [Citado el: 10 de Mayo de 2017.] http://ccdoc-sistemasrecuperacioninternet.blogspot.com.

Pelaez, Juan Carlos. 2009. Arquitectura basada en Componentes. [En línea] 18 de abril de 2009. [Citado el: 28 de Febrero de 2017.] http://geeks.ms/jkpelaez/2009/04/18/arquitectura-basada-en-componentes/.

Pressman. 2005. *Ingeniería de Software,un enfonque práctico.* New York : s.n., 2005.

Quero, Manuel Sánchez. 2001. Una introducción al Text Encoding for Interchange. [En línea] 2001. [Citado el: 10 de Mayo de 2017.] www.tei-c.org/Vault/P4/Lite/teiu5_sp.html.

Rautenstrauch, Ramón. 2010. Opciones avanzadas en las búsquedas de Google y Bing. *Opciones avanzadas en las búsquedas de Google y Bing.* [En línea] 28 de Octubre de 2010. [Citado el: 6 de Diciembre

de 2016.] http://www.apasionadosdelmarketing.es.

Sánchez, Pastor. 2010. Manual de Skos . *Manual de Skos .* [En línea] 2 de Junio de 2010. [Citado el: 3 de Diciembre de 2016.] http://hdl.handle.net/10760/14614.

Seta. 2010. Apache Solr:una introducción. [En línea] 2010. [Citado el: 2017 de Febrero de 26.] http://wwwdosideas.com/noticias/java/913-apache-solr-unaintroducción.html..

Sommerville, Ian. 2005. *Ingenieria de Software.* Madrid : Pearson Eduación S.A , 2005. 84-7829-074-5.

SparxSystems. 2014. Diagrama de Despliegue UML. [En línea] 2014. [Citado el: 27 de Febrero de 2017.] http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html.

Thames, Juan Pablo Bustos. 2011. Diagramas de clases . 2011.

Vargas-Arcila, Angela M. 2016. Análisis de Esquemas de Metadatos para la Marcación de Contenidos Educativos. Colombia : s.n., 2016.

Ventura, Victor. 2016. Formato JSON. *Formato JSON.* [En línea] 16 de Marzo de 2016. [Citado el: 3 de Diciembre de 2016.]

Wainerman, Efraim. 2001. *Motores de búsqueda en Internet.* 2001.

Zorrilla Castro, César. 2010. *Guía de Arquitectura N-Capas orientada al Dominio.* 2010. ISBN 978-936696-3-8.

Anexos

Tabla 10. Historia de Usuario # 4. Identificar el metadato url del documento para su almacenamiento en el core de los elementos generales.

Historia de usuario	
Número: HU_4	**Nombre Historia de Usuario:** Identificar el metadato url **< url >** del documento para su almacenamiento en el core de los elementos generales.
Modificación de Historia de Usuario Número: **0**	
Usuario: **Sistema**	**Iteración Asignada:** 1
Prioridad en el Negocio: **Alta**	**Puntos Estimados:** 1
Riesgo en Desarrollo: **Alto**	**Puntos Reales:**1
Programador responsable: **Anietsy Perera Pérez**	
Descripción: **Luego de ser rastreada los elementos generales, identificar la dirección única del recurso informático disponible en la internet.**	

Tabla 11. Historia de Usuario # 5. Identificar el metadato contenido del documento para su almacenamiento en el core de los elementos generales.

Historia de usuario	
Número: HU_5	**Nombre Historia de Usuario:** Identificar el metadato contenido **< document_content >** del documento para su almacenamiento en el core de los elementos generales.
Modificación de Historia de Usuario Número: **0**	
Usuario: **Sistema**	**Iteración Asignada:** 1
Prioridad en el Negocio: **Alta**	**Puntos Estimados:** 1
Riesgo en Desarrollo: **Alto**	**Puntos Reales:**1
Programador responsable: **Anietsy Perera Pérez**	

Descripción: **Luego de ser rastreada los elementos generales, identificar el texto plano que tiene el documento (contenido).**

Tabla 12. Historia de Usuario # 6. Identificar el metadato fecha de última actualización del documento para su almacenamiento en el core de elementos generales.

Historia de usuario	
Número: HU_6	**Nombre Historia de Usuario:** Identificar el metadato fecha **< last_modified >** de última actualización del documento para su almacenamiento en el core de los elementos generales.
Modificación de Historia de Usuario Número: **0**	
Usuario: **Sistema**	**Iteración Asignada:** 1
Prioridad en el Negocio: **Alta**	**Puntos Estimados:** 1
Riesgo en Desarrollo: **Alto**	**Puntos Reales:**1
Programador responsable: **Anietsy Perera Pérez**	
Descripción: **Luego de ser rastreada los elementos generales, identificar la última fecha en que fue modificado el contenido de los documentos que se detectaron.**	

Tabla 13. Historia de Usuario # 7. Identificar el metadato host del documento para su almacenamiento en el core de los elementos generales.

Historia de usuario	
Número: HU_7	**Nombre Historia de Usuario:** Identificar el metadato host **< host >** del documento para su almacenamiento en el core de los elementos generales.
Modificación de Historia de Usuario Número: **0**	
Usuario: **Sistema**	**Iteración Asignada:** 1
Prioridad en el Negocio: **Media**	**Puntos Estimados:** 1

Riesgo en Desarrollo: **Media**	**Puntos Reales:**1
Programador responsable: **Anietsy Perera Pérez**	
Descripción: **Luego de ser rastreada los elementos generales, identificar el dominio a la que pertenece la dirección donde se encuentra esa información.**	

Tabla 14. Historia de Usuario # 8. Identificar el metadato de la dimensión del documento para su almacenamiento en el core de las imágenes.

Historia de usuario	
Número: HU_8	**Nombre Historia de Usuario:** Identificar el metadato de la dimensión **< width >** y **< height >** del documento para su almacenamiento en el core de las imágenes.
Modificación de Historia de Usuario Número: **0**	
Usuario: **Sistema**	**Iteración Asignada:** 1
Prioridad en el Negocio: **Media**	**Puntos Estimados:** 1
Riesgo en Desarrollo: **Media**	**Puntos Reales:**1
Programador responsable: **Anietsy Perera Pérez**	
Descripción: **Luego de ser rastreada las imágenes, identificar el ancho y el largo de la imagen.**	