Universidad de Matanzas

Facultad de Ciencias Técnicas

Departamento de Informática

SISTEMA DE RECUPERACIÓN DE INFORMACIÓN EN EL REPOSITORIO DE DOCUMENTACIÓN TÉCNICA DE ELECTROMEDICINA, MATANZAS.

TRABAJO PARA OPTAR POR EL TÍTULO DE INGENIERO EN INFORMÁTICA

Autor: Javier Peralta Díaz

Tutor: Josval Díaz Blanco

Yeiniel Alfonso Martínez

Matanzas

2018

DEDICATORIA

A mi familia por apoyarme siempre y alentarme a brindar lo mejor de mí.

AGRADECIMIENTOS

A mis tutores por prestarme tanto de su tiempo.

A mis amigos que estuvieron en todo momento.

A mis compañeros de trabajo por brindarme su ayuda en el desarrollo del proyecto.

Y a todos los que colaboraron de una forma u otra para hacer realidad este trabajo.

DECLARACION DE AUTORIA

Yo, Javier Peralta Díaz, declaro que soy el único autor de este trabajo que lleva como título Sistema de recuperación de información en el repositorio de documentación técnica de Electromedicina, Matanzas, y autorizo a la Universidad de Matanzas, especialmente a la Facultad de Ciencias Técnicas, Departamento de Informática, a que hagan el uso que estimen pertinente de esta investigación.

_____	_____
Firma del Autor	Firma del Tutor
Javier Peralta Díaz	Josval Díaz Blanco

RESUMEN

En la actualidad, las empresas utilizan las Tecnologías de la Información y las Comunicaciones (TIC) para mejorar sus procesos y productos, por ende, la creación, distribución y manipulación de la información es un recurso necesario y cotizado, que acompaña nuestras actividades sociales, culturales y económicas cotidianas. Esta investigación se enfoca en el desarrollo de un sistema de recuperación de información en el repositorio de documentación técnica del Taller de Electromedicina de Matanzas. Esta actividad se realiza de forma manual incurriendo en largas búsquedas y su consecuente pérdida de tiempo. Este problema quedaría resuelto con la creación de un software que facilite y agilice esta tarea, obteniéndose como resultado una mayor cantidad de consultas a la documentación, lo cual tendrá un impacto directo en la calidad de los trabajos de reparación del equipamiento médicos realizados en el taller. Para alcanzar este objetivo se toma como referencia los grandes buscadores de información de internet, los cuales implementan una arquitectura de tres partes, un crawler, un indexador y la aplicación de consultas. Basado en esto, se utilizó las plataformas de código abierto, Norconex HTTP Collector, Apache Solr y la implementación de una herramienta utilizando el framework de Java, Sprint. Este proyecto se desarrolla bajo la guía de la metodología de desarrollo de software ágil SCRUM proporcionándole al software adaptabilidad y posterior actualización.

ABSTRACT

Currently, companies use information and communication technologies (ICT) to improve their processes and products, therefore, the creation, distribution and manipulation of information is a necessary and quoted resource, which accompanies us in our activities social, cultural and economic issues. This research focuses on the development of an information retrieval system in the technical documentation repository of the Matanzas Electromedical. This activity is performed manually incurring long searches and their consequent loss of time. This problem that would be solved with the creation of a software that facilitates and speeds up this task, obtaining as a result a greater amount of consultations to the documentation, which will have a direct impact on the quality of the repair work of the medical equipment made in the workshop. In order to achieve this objective, the great internet information search engines are taken as reference, which implement a three-part architecture, a crawler, an indexer and the application of queries. Based on this, we used the open source platforms, Norconex HTTP Collector, Apache Solr and the implementation of a tool using the Java framework, Sprint. This project is developed under the guidance of the agile software development methodology SCRUM, providing the software with adaptability and subsequent updating.

INDICE

Índice de Tablas

INTRODUCCIÓN

El avance vertiginoso de las tecnologías de la información y las comunicaciones, que se inició en la segunda mitad del pasado siglo, matiza el mundo actual y a una sociedad que ha procurado llamarse sociedad de la información. En ella, una parte importante del esfuerzo del hombre se ha concentrado en la producción, manejo y uso de la información. El surgimiento, desarrollo y expansión de la informática y las telecomunicaciones, ha supuesto una revolución sin precedentes en el mundo. Internet, su mayor exponente, se ha convertido en una gran biblioteca caótica que crece, continua y aceleradamente.

Esto ocurre de igual forma las organizaciones científicas y económicas, las cuales se enfrentan al manejo de una gran cantidad de información estratégica en sus actividades. Un desafío principal es poder encontrar la información correcta rápidamente y con precisión. Para hacerlo, las organizaciones deben dominar el acceso a la información: obtener consultas relevantes, resultados que están organizados y ordenados.

El Centro Provincial de Servicios Técnicos de Electromedicina de Matanzas es una entidad perteneciente al Sistema Nacional de Salud cubano. Tiene como principal objetivo brindar servicios de reparación de forma gratuita a las unidades de salud, a la vez de garantizar, con un personal altamente especializado, de forma sostenible, los servicios técnicos a los equipos médicos, así como la fabricación de piezas de repuesto y la recuperación de muebles clínicos, equipos médicos, piezas de repuesto e instrumental médico, que permitan la vitalidad de las unidades de salud existentes en la provincia.(Electromedicina-Matanzas 2017)

En estos momentos existen funcionando en la provincia de Matanzas una cifra aproximada de 7800 equipos médicos, este número se convierte en un valor cercano a 350 tipos de equipos diferentes. Gran parte de la documentación que poseen cada uno de estos equipos se encuentra en formato digital. Estos datos son

almacenados en uno de los servidores de la institución y distribuidos hacia las computadoras clientes a través de una carpeta compartida.

Con el transcurso de los años este repositorio se ha convertido en un gran almacén de contenido, llegando a alcanzar la cifra de 200Gb de información técnica. Este valor está en un contante ascenso debido al nuevo equipamiento que se instala de forma periódica y los cursos impartidos a los técnicos del taller dentro y fuera del país.

El repositorio cuenta en estos momentos con varios tipos de formatos de datos entre los que se encuentran documentos de la librería de Microsoft Office, documentos PDF, imágenes, páginas web, entre otros, distribuidos de forma poco organizada. Todo esto dificulta grandemente la consulta de la documentación necesaria en el momento de hacer alguna reparación al equipamiento, creando una situación muy peligrosa que puede ocasionar la pérdida del equipo o algún accidente laboral.

A partir de esta situación problémica se toma la decisión de incorporar una herramienta que facilite la revisión documental asociada al proceso de reparación y mantenimiento de los equipos, disminuyendo de esta forma los riesgos asociados a este proceso. Por este motivo se define como **problema de investigación** de este trabajo de diploma, la no existencia de una herramienta de búsqueda de información en el repositorio de documentación técnica del Taller de Electromedicina de Matanzas.

Se introduce de esta manera la **hipótesis** que guiará el proceso de investigación, esta se formula como, si se implementa un sistema de recuperación de información en el repositorio de documentación técnica del Taller de Electromedicina de Matanzas, se garantizará la búsqueda por contenido posibilitando un mejor desempeño en las tareas de reparación de equipos médicos.

Toda esta investigación se acota y se establecen sus límites gracias al **objeto de estudio** que se muestra como, la recuperación de información en grandes repositorios de datos. Pero se centrará más la atención en, la recuperación de

información en repositorios de datos aplicando técnicas de indexación, que es el **campo de acción.**

Todo esto conduce al **objetivo general** de la investigación que se plantea como, implementar un sistema de recuperación de información en el repositorio de documentación técnica del Taller de Electromedicina de Matanzas. Y a sus **objetivos específicos**:

- Analizar los diferentes enfoques para la gestión de grandes volúmenes de información.
- Definir los componentes según un modelo orientado a la recuperación de información aplicando técnicas de indexación.
- Implementar e implantar el modelo propuesto.
- Validar el impacto de la herramienta en el desempeño en las tareas de reparación de equipos médicos.

El proceso de investigación estuvo guiado por los métodos de investigación científica para proporcionar una mayor calidad del estudio. Se tomó como referencia el estudio (Zayas and Lombardía).

Métodos teóricos:

- **Analítico:** Se utilizó para dividir el objeto de estudio en elementos más simple y mejorar el entendimiento de los mismos.
- **Sintético:** Después de analizar los elementos del objeto de estudio por separado se crea una teoría unificadora para entender los elementos como un todo.
- **Hipotético-deductivo:** Para la conformación de la hipótesis en base a los conocimientos teóricos obtenidos.

La investigación contiene una estructura capitular, que se resume a continuación:

Capítulo I: Marco teórico referencial. Se expone el análisis teórico realizado sobre el objeto de estudio de la investigación, la fundamentación teórica de la metodología utilizada y las herramientas que pudieran darle solución a la investigación.

Capítulo II: Propuesta de solución. Descripción de las herramientas seleccionadas para dar una posible solución a la investigación, su integración y configuración. Descripción del proceso de implementación guiado por la metodología SCRUM y la validación de la aplicación obtenida.

CAPÍTULO I: MARCO TEÓRICO REFERENCIAL

1.1 Introducción al Capitulo

En este capítulo, en aras de comprender la problemática planteada para realizar una propuesta de solución, se expone el análisis teórico del objeto de estudio de la investigación. Se realiza el análisis de una serie de herramientas para seleccionar la adecuada a utilizar en la implementación de la solución y se fundamenta la metodología utilizada durante el proceso de desarrollo.

1.2 Marco teórico de la investigación.

Para resolver el problema de la investigación se hace necesario caracterizar el objeto de investigación y su campo de acción en el que se manifiesta el problema. Para ello, el investigador estudia toda la teoría científica que existe acerca del campo de acción, para precisar sus cualidades, propiedades y relaciones. A la vez empieza a observar los fenómenos o procesos que se manifiestan en el objeto, determinando en el mismo ciertas características externas (variables) que le posibilitan diagnosticar la situación del objeto, así como su comportamiento en el tiempo, también llamado tendencias.(Zayas and Lombardía)

1.2.1 Requisitos a tener en cuenta para implantar una herramienta

- Herramienta que permita realizar consulta sobre el repositorio de documentación técnica del Taller de Electromedicina de Matanzas
- Que la consulta se pueda realizar desde cualquier maquina nodo de la red interna de Electromedicina.
- El software utilizado no debe consumir muchos recursos del servidor central de la institución.

1.2.2 Qué son los datos y cómo los vemos

El siguiente análisis sobre los datos toma como referencia el estudio realizado por (Kitchin 2014) donde presenta diferente definiciones y categorías de estos.

Los datos son comúnmente comprendidos por ser la materia prima producida al abstraer el mundo en categorías, medidas y otras formas representacionales como números, caracteres, símbolos, imágenes, bits, que constituyen los bloques de construcción de los cuales la información y el conocimiento es creado.

Son usualmente representativos en la naturaleza, por ejemplo, al medir fenómenos como la edad de una persona, la altura, el peso, el color, pero también pueden ser implicados como a través de una ausencia en lugar de una presencia, o pueden ser derivados como los datos que son producidos a través de otros datos, ejemplo, el cambio del porciento calculado a través del tiempo comparando dos periodos diferentes. Pueden ser grabados y almacenados, o codificados en formato digital.

Los datos de buena calidad son discretos y comprensibles, cada dato es individual, separado y separable, a la vez de estar definido de forma clara. Pueden ser agregados en colecciones y tener asociados metadatos, que son los datos de los datos. Estos pueden unirse a otros dataset para proporcionar un entendimiento no disponible en data set simples.

De forma general los datos proveen una fuerte utilidad y alto valor al ser la llave de entrada a varios modos de análisis que utilizan instituciones, negocios y ciencia para entender y explicar el mundo en que vivimos, a la vez de desarrollar e innovar en nuevos productos y conocimientos.

1.2.3 Tipos de datos

De forma general los datos varían de acuerdo a su forma (cuantitativa o cualitativa), a su estructura (estructurado, semi-estructurado, no estructurado), su procedencia

(capturado, derivado, transitorio), su productor (primario, secundario, terciario) y su tipo (indexical, atributo, metadato)

Idexicalidad no es un término común por lo que se procede a definirlo según el punto de vista de (Reyes 2009)

Se refiere tanto al uso de la situación para crear la independencia del contexto como al uso de elementos específicos de un tiempo y lugar determinados para generar el significado. La afirmación de que el significado se crea y se mantiene mediante el uso de recursos metódicos es fundamental para diferenciar la etnometodología del estructuralismo. El "miembro" no sólo domina series de normas sintácticas y semánticas sino también "características indicativas", supuestos, convenciones e información contextual con el fin de enterarse de lo que "ocurre" en una situación determinada. Las clausulas "ad hoc", "etceteras", "formulaciones" o "glosas" son recursos metódicos que organizan el significado contextualmente.

Datos cuantitativos y cualitativos

Los datos están típicamente divididos en dos grandes categorías, los datos cuantitativos y los datos cualitativos. Los **cuantitativos** son registros numéricos, generalmente relacionados con propiedades físicas o fenómenos como la altura, el ancho, el peso o la distancia, o son representativos o relacionados con características no físicas como las clases sociales, logros educacionales o la calidad de vida.

Este tipo de dato está dividido en cuatro diferentes niveles de medida que limita como estos pueden ser procesados y analizados.

Nivel de Medida	Definición	Ejemplo
Datos Nominales	Categórico en la naturaleza con observaciones registradas en unidades discretas.	Casado, soltero, divorciado.
Datos ordinales	Observaciones que son colocadas en una escala ordenada, donde ciertas observaciones son mayores que otras.	Bajo, medio, alto.
Datos de intervalo	Medidas a lo largo de una escala la cual posee un intervalo fijo pero arbitrario y un origen fijo. La adición o multiplicación por una constante no alterara la naturaleza del intervalo o de las observaciones.	La temperatura a lo largo de una escala de grados Celsius.
Datos de proporción	Similar a los datos de intervalo, exceptuando que la escala procesa un cero original.	Notas de un examen en una escala de 0-100

Tabla 1. Niveles de medida de datos.(Kitchin 2014)

En contraste, los datos **cualitativos** son los no numéricos, como los textos, fotografías, arte, videos, sonidos. Estos pueden ser convertidos a cuantitativo, pero implica una reducción significativa en la riqueza original de los datos y su

consecuente pérdida de información, por lo que el análisis de estos datos es realizado en su material original.

1.2.4 Forma de estructurar los datos

Datos Estructurados

Son aquellos que pueden ser fácilmente organizados, almacenados y transformados en un modelo de datos diferente, es el caso de texto o números, que pueden ser introducidos en una tabla o base de datos relacional que posee un formato consistente. Estos datos pueden ser procesados, consultados, combinados o analizados de forma relativamente sencilla utilizando cálculos o algoritmos, a la vez de poder visualizarlos a través de gráficos o mapas.

Datos Semi-estructurados

Son datos débilmente estructurados, que no poseen un modelo o un esquema predefinido, por tal motivo no pueden estar contenidos en una base de datos relacional. Su estructura es irregular, implícita, flexible y a menudo anidada jerárquicamente, pero poseen un conjunto coherente de campos y los datos son etiquetados, separando el contenido semánticamente y proporcionando metadatos autodefinidos que proporcionan una manera de ordenar y estructurar los datos.

Datos no Estructurados

Estos no poseen una definición del modelo de datos o una estructura común identificable. Cada elemento individual, tal como texto o fotografía pude tener una estructura o formato especifico, pero no todos los datos dentro del conjunto comparten la misma estructura. Estos datos pueden ser buscados o consultados, pero no pueden ser fácilmente combinados o computacionalmente analizados.

1.2.5 Gestores Documentales y los Gestores de Contenido

Para dar solución al problema de investigación se comenzó la búsqueda de programas que fueran capaces de organizar y agrupar documentos. Se encontraron dos ramas principales de software desarrollados con esta finalidad, los Gestores Documentales y los Gestores de Contenido, cada uno con sus características propias desarrolladas para dar solución a una determinada situación.

Gestores Documentales

Son los encargados de la gestión de forma automatizada de los documentos producidos por una organización. No restringe de forma alguna el formato a usar. Incluye desde el diseño de los procesos de generación de esos documentos, su flujo de trabajo y almacenamiento, hasta su difusión final a quien interese o expurgo final.(Cobos 2002)

Por ejemplo, una empresa genera una serie de documentos, de formato heterogéneo dependiendo de la naturaleza de la actividad de cada departamento. La misión de la gestión documental es administrar todo ese volumen de material de forma que aquellos realmente necesarios sean localizables lo antes posible.

Gestor de Contenido

Entra en el terreno de internet, y tiene más que ver con comunicación, entendida como medio de comunicación, que con documentos. Portales corporativos, horizontales o verticales, redes de comunidades virtuales, periódicos digitales o webzines, todos necesitan herramientas que nos permitan gestionar el flujo de información que hay desde la creación de un contenido diverso (texto, multimedia, interactivos) hasta su publicación y posterior recuperación.(Cobos 2002)

Diferencias entre Gestor Documental y Gestor de Contenido

El Gestor Documental es el encargado de procesar todo archivo que entre dentro de la clasificación de documento, mientras que el Gestor de Contenidos tiene un

rango de acción mucho más amplio, al tener la capacidad de manejar información de varias fuentes como páginas web, ftp, correos, entre otros muchos.

Un buen sistema de gestión de contenidos puede y debe tener un modelo de gestión documental completo, pero su filosofía está encaminada a la presentación de la información, no tan sólo administración.

La gestión documental trabajará desde la empresa para ella misma mientras que la de contenidos lo hará desde ella hacia el usuario.

1.2.6 Biblioteca digital

Es una colección de objetos digitales creados o recopilados y administrados para la creación de colecciones. Se ponen a disposición de los usuarios de forma coherente, perdurable y con el respaldo de los servicios necesarios para que se puedan encontrar y utilizar estos recursos.

El concepto de biblioteca digital es la evolución de las demandas, necesidades y servicios del usuario del siglo XXI, mediante el uso de las TIC. Gestiona materiales multimedia, documentos que pasaron por el proceso de digitalización y de aquellos que se produjeron de manera digital.

Las bibliotecas digitales cuentan con enlaces hipertextuales que facilitan la asociación con otros contenidos relacionados, ampliando las posibilidades de búsqueda de información. Trata los datos teniendo en cuenta el ciclo de la gestión del conocimiento, su organización, comunicación-difusión, almacenamiento, búsqueda, filtrado-selección y reutilización. (Aguilar and Garcia 2013)

1.2.7 Biblioteca virtual

La aplicación de la tecnología de la información posibilita la definición de una nueva estrategia de desarrollo de las organizaciones documentales. Entre ellas el establecimiento de espacios virtuales a través de los cuales los usuarios pueden acceder a las colecciones con independencia de las coordenadas espaciales o

temporales, prestando los servicios desde y hacia cualquier lugar sin necesidad de desplazamiento físico.

Esto supone una reorganización de los procesos técnicos y administrativos que conforman la biblioteca: sus recursos materiales, humanos y servicios para dotarla de una infraestructura cliente-servidor adecuada.

El libre acceso a la información y la posibilidad de contribuir activamente a su organización, búsqueda y recuperación representan ventajas inestimables, superando limitaciones geográficas, culturales, lingüísticas, económicas y sociales. Asimismo, a través de contenidos en formatos auditivos o audiovisuales y aplicaciones especiales para personas con discapacidades, estas bibliotecas promueven el respeto a la diversidad y constituyen herramientas potentes de inclusión y desarrollo social. Contribuyen además a la preservación y difusión de documentos valiosos o únicos y con ello garantizan su consulta y conocimiento por las futuras generaciones.(Fagundo 2016)

Ventaja de las bibliotecas digitales y virtuales

- Ahorro de papel.
- Disminución de la necesidad de espacios en las bibliotecas.
- Crecimiento y mejor organización de los acervos.
- Optimización de los mecanismos de búsqueda de textos, imágenes, videos audio.
- Facultad de acceder a información desde cualquier parte del mundo y compartirla.

Desventajas de las bibliotecas digitales y virtuales

- Introducir nuevos datos es una tarea que se realiza de forma manual incurriendo en elevados costes y gasto de tiempo cuando se necesita introducir grandes volúmenes de información.

- Se corre el riesgo de que la piratería se haga presente y, con ello, que los autores carezcan de los beneficios que, por derecho, les corresponden. Esto puede ser visto desde otro punto de vista, al ser el conocimiento un sector de pago, volviéndose muy privativo para muchos lugares del mundo que no pueden acceder a materiales de calidad.
- Se convierte en una tarea difícil asegurar la veracidad de los materiales almacenados.
- Es importante considerar los costos de los equipos que se requiere para digitalizar y almacenar la información.

1.3 Antecedentes de la Investigación

Después de analizar los resultados obtenidos de la investigación teórica se comienza a seguir una línea de software que pueda dar solución al problema de investigación. Teniendo en cuenta de que no se cuenta con presupuesto para comprar software de pago disponible y su posterior mantenimiento, se desecha la línea del software propietario y se empiezan a considerar las opciones de software libre que se puedan implementar sin ningún tipo de coste.

La búsqueda de herramientas desveló aplicaciones de Gestión de Contenido tales como:

1.3.1 CMS (Content Management System)

1.3.1.1 Nuxeo

Es un software que permite implementar con gran funcionalidad un repositorio documental corporativo. Aporta soluciones a las necesidades primarias de gestión documental de las empresas, permitiendo gestionar cómodamente documentos mediante control de versiones, flujos de trabajo asociados, publicación remota o búsqueda avanzada a texto completo, además de integración con suite ofimáticas habituales como Microsoft Office y Open Office.

El flujo de los documentos a través de los procesos de negocio puede ser gestionado y seguido. Desde la captura inicial y la creación, intercambio y colaboración, a través de la aprobación, análisis y revisión, hasta la publicación y archivo, la gestión de documentos garantiza que los trabajadores puedan encontrar, utilizar, compartir y asegurar el valioso contenido corporativo.(athento.com 2018)

Posee funcionalidades como:

- Captura basada en formularios
- Captura mediante Correo electrónico
- Integración de Aplicaciones de Escritorio
- Espacios de trabajo de colaboración
- Cliente en línea
- Gestión de registros
- Auditoría

Requisitos mínimos de instalación

- RAM: 2 GB es la mínima cantidad de memoria para ejecutar Nuxeo.
- CPU: Intel Core 2, equivalente o superior.
- Disco Duro: Para la instalación de Nuxeo se requieren menos de 300 MB de disco duro.

1.3.1.2 Alfresco

Es una solución versátil compatible con software tanto de la vertiente Microsoft, como de la rama Linux. Posibilita la creación y gestión de contenidos empresariales desde una gran cantidad de CMSs, blogs y paquetes ofimáticos (Office y OpenOffice). Además, ofrece una gran variedad de herramientas colaborativas como calendarios individuales y de equipo, feeds de actividad, tableros de discusión, etc. Alfresco es para las empresas ante todo de colaboración.

Posee funcionalidades como:

- Carpetas inteligentes facilitan el descubrimiento del contenido.
- Modelos de metadatos enriquecidos.
- Flujos de trabajo incorporados simplifican la revisión y aprobación de los documentos.
- La gestión de versiones del documento.

Requisitos mínimos de instalación

- RAM: 4 GB es la mínima cantidad de memoria para ejecutar Alfresco.
- CPU: Intel Core 2, equivalente o superior. Mayor de 2.5Hz. Preferentemente de 64bit

Estas herramientas además de ser gestores documentales, entran dentro de la clasificación de gestores de contenido, por lo cual son aplicaciones muy potentes, eficaces, con gran cantidad de prestaciones pero que consumen muchos recursos de sistema, incumpliendo uno de los requerimientos planteados.

Tras no poder resolver el problema de investigación con este tipo de herramientas se pasa a investigar las **bibliotecas digitales**.

1.3.2 DMS (Document Manager System)

1.3.2.1 Greenstone

Es un conjunto de programas de software diseñado para crear y distribuir colecciones digitales, proporcionando así una nueva forma de organizar y publicar la información a través de Internet o en forma de CD-ROM. Greenstone ha sido producido por el Proyecto Biblioteca Digital de Nueva Zelanda con sede en la Universidad de Waikato y ha sido desarrollado y distribuido en colaboración con la UNESCO y la ONG de Información para el Desarrollo Humano con sede en Amberes, Bélgica. Es un software abierto en varios idiomas distribuido conforme a los términos de la Licencia Pública General GNU.

El objetivo del software Greenstone es dar el potencial de construir sus propias bibliotecas digitales a los usuarios, especialmente en universidades, bibliotecas y otras instituciones de servicio público. Las bibliotecas digitales están cambiando radicalmente la manera en que se adquiere y disemina la información en las comunidades e instituciones que participan en UNESCO, en los campos de educación ciencia y cultura en todo el mundo, y especialmente en los países en desarrollo.(New_Zealand_Digital_Library_Project 2018)

Esta es una muy buena opción para desplegar una biblioteca digital, al brindar la posibilidad de adaptarse a un ambiente online compartiendo su información vía web, o en un entorno completamente sin conexión. Pero no posibilita la búsqueda de información por contenido de documentos el cual es un requisito indispensable para dar solución a las necesidades del problema de investigación.

1.3.2.2 DSpace

Es un software muy utilizado en organizaciones académicas, sin fines de lucro y comerciales para construir sus repositorios digitales. Es gratis y fácil de instalar, su configuración viene lista para utilizar. Es personalizable para adaptarse a las necesidades de cualquier organización.

Esta aplicación posee una gran comunidad que la respalda y ayuda al mejoramiento de esta. Es de código abierto y su código fuente está disponible en GitHub para que cualquier persona pueda descargarlo y adaptarlo.

Puede reconocer y administrar una gran cantidad de formatos de archivos y mime types. Algunos de los formatos más comunes actualmente administrados dentro del entorno DSpace son archivos PDF, Word, JPEG, MPEG, TIFF. Aunque el DSpace listo para usar solo reconoce automáticamente los formatos de archivo comunes. También proporciona un registro de formato de archivo simple donde puede registrar cualquier formato no reconocido, para que pueda ser identificado en el futuro.(DuraSpace 2018)

Esta aplicación es muy moderna y eficaz pero no puede realizar consultas al contenido de los documentos guardados en su base de datos, por lo que no se puede tener en cuenta para su utilización.

1.3.2.3 Fedora

Fedora es un sistema de repositorio robusto, modular y de código abierto para la gestión y difusión de contenido digital. Es especialmente adecuado para bibliotecas digitales y archivos, tanto para el acceso como para la preservación. También se utiliza para proporcionar acceso especializado a colecciones digitales muy grandes y complejas de materiales históricos y culturales, así como a datos científicos. Fedora tiene una base de usuarios instalados en todo el mundo que incluye organizaciones de patrimonio académico y cultural, universidades, instituciones de investigación, bibliotecas universitarias, bibliotecas nacionales y agencias gubernamentales.

El proyecto Fedora está liderado por Fedora Leadership Group y está bajo la administración de la organización sin fines de lucro DuraSpace que brinda liderazgo e innovación para proyectos y soluciones de tecnología de fuente abierta que se enfocan en el acceso duradero y persistente a datos digitales.

1.3.2.4 Calibre

Hoy calibre es una comunidad de código abierto vibrante con varios desarrolladores y muchos probadores y reporteros de errores. Se utiliza en más de 200 países y ha sido traducido a una docena de idiomas diferentes por voluntarios. Calibre se ha convertido en una herramienta integral para la gestión de textos digitales, que le permite hacer todo lo que pueda imaginar con su biblioteca de libros electrónicos.

La lectura es esencial y muy importante y uno de sus objetivos siempre ha sido evitar la fragmentación o la monopolización del mercado de libros electrónicos por parte de entidades que solo se preocupan por objetivos a corto plazo. A medida que la comunidad de calibre continúa creciendo, impulsada por los amantes de los libros,

para los amantes de los libros, es de esperar que siempre presente una alternativa para las personas que aman leer libros electrónicos y desean tener el control de sus propias bibliotecas digitales.(GOYAL 2018)

Las bibliotecas virtuales son de gran ayuda en la organización de grandes volúmenes de documentos y su posterior búsqueda y revisado. Esta clasificación depende de dos métodos de introducción de datos, el primero es la búsqueda automática en el repositorio contenedor de documentos configurado en la aplicación, este método realiza la clasificación de documentos basado en los metadatos encontrados en cada uno de estos. La desventaja principal con este método se encuentra en la veracidad de los metadatos contenidos en cada uno de los archivos, que en la mayoría de los casos no son correctos o están puestos en los campos equivocados, este problema introduce la segunda forma de introducción de datos.

Al no ser capaz el sistema de clasificar de forma automática los archivos se hace necesario la clasificación manual de cada uno de estos, esto trae consigo la utilización de gran cantidad de personas para esta tarea con el objetivo de acelerar el proceso o la consecuente demora si no se dispone de una cantidad razonable en concordancia con el volumen de datos a procesar.

Además de los recursos humanos para completar la tarea se hacen necesarios recursos materiales, los cuales son proporcionales a la cantidad de personas utilizadas. Otro requisito a tener en cuenta es la introducción de nuevos materiales al repositorio, los cuales afectarían el tiempo estimado para el trabajo, alargando de forma indefinida este proceso.

Los materiales a ser procesados en Electromedicina Provincial responden a distintas ramas dentro de la institución, por lo que habría de desviar del trabajo a los especialistas para que se dedicaran a organizar los datos en la biblioteca digital, esta no es una solución viable para dar solución al problema, por lo que se comienza a investigar otra posible solución como son los recuperadores de información.

1.3.3 Recuperadores de Información

Para definir bien que es recuperación de información utilizaremos dos fuentes bibliográficas la primera (Gray and Belew 2018) y la segunda (Manning, Raghavan et al. 2008)

Es la actividad o proceso de obtener recursos de información relevante, usualmente documentos, de naturaleza no estructurada, usualmente texto, para responder a una necesidad de información desde una colección de recursos de información, usualmente almacenada en una computadora.

1.3.3.1 Open Semantic Framework

Es un software integrado que utiliza tecnología semántica para manejar contenido. Contiene una arquitectura de capas que combina software de código abierto existente con componentes adicionales de código abierto. Está diseñado como una plataforma de contenidos integrada accesible vía web a la vez de responder a la licencia de Apache 2.(Structure_Dynamics_LLC 2018)

Requerimientos para la instalación

- Distribución de Linux Soportada:
  - CentOS 7
  - CentOS 6
  - Ubuntu 14.04
- PHP 5.4 o mayor.
- 64 Bit Sistema operativo de 64bit.
- 5 GB de espacio en la partición donde se instala OSF.
- 2 GB de RAM.

Factores por los que no se utiliza

- El requerimiento de RAM mínimo es muy alto para mantenerlo en los servidores de la institución.
- Es una aplicación que ya no tiene soporte por parte de sus desarrolladores.

1.3.3.2 Open Semantic Search

Es un servidor de búsqueda integrado que posee un framework ETL (Extract, Transform, Load) que le permite realizar crawling, estación de texto, análisis de texto, OCR para imágenes, entre otras funciones. Este software es de código abierto y su código fuente se encuentra en GitHub sin ningún tipo de restricción.

Requerimientos para la instalación

- 4GB de RAM mínimos, se recomienda asignarle más.
- Virtual Box v5.2.6 o superior si se escoge la versión de máquina virtual suministrada por el fabricante.

Factores por los que no se utiliza

- El requerimiento de RAM mínimo es muy alto para mantenerlo en los servidores de la institución.
- Resulta complejo modificar la aplicación para adaptarla a las necesidades o eliminar los elementos no necesarios.

Al no encontrar una herramienta que satisfaga las necesidades de la institución, se decide desarrollar una aplicación que pueda dar solución a los problemas planteados y esté en consonancia de los requerimientos expuestos.

Para esto se toma como ejemplo la aplicación Open SemanticSearch la cual cuenta con un crawler que se encarga de gestionar el directorio configurado donde se encuentran los contenidos a recuperar, seguido de una herramienta indexadora que tiene como objetivo guardar el contenido de los documentos y por ultimo un software que se encargue de controlar estas dos herramientas a la vez de recuperar la información deseada por el usuario final en el momento de realizar la consulta.

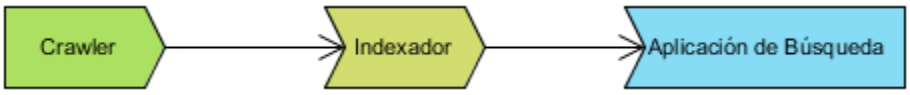


Diagrama 1: Diagrama guía para desarrollo de aplicación.

Para alcanzar este objetivo se analizaron los siguientes crawler buscando el más adecuado para la implementación.

1.3.4 Crawlers

Es un sistema de recuperación de información en Internet, basado en páginas previamente catalogadas, y cuyos resultados son enlaces a las páginas reales que contengan ciertos parámetros o criterios. El buscador habitualmente toma como referencia meta-etiquetas de marcado como títulos, descripción o palabras clave dentro de los documentos, y con base en ello clasifican o ponderan los documentos. De esta forma, un documento con sus meta-etiquetas completas y con contenido relevante será mejor ponderado por un buscador web que un documento que no contenga tales etiquetas. A esta optimización se le conoce como “Optimización para Motores de Búsqueda”, o SEO por sus siglas en inglés.

El objetivo principal de un Web crawler es proporcionar datos actualizados a un motor de búsqueda. Son utilizados principalmente para crear una copia de todas las páginas rastreadas para su posterior procesamiento por un motor de búsqueda luego de ser indexadas para proporcionar resultados de una forma rápida. Las metas de un crawler óptimo son su fácil escalabilidad, su habilidad de determinar qué contenido es susceptible de descarga y cuál se debe desechar y mantener su responsabilidad social y ética.

1.3.4.1 WebSPHINX

Website-Specific Processors for HTML INformatione Xtraction es una librería de clases de Java y un ambiente de desarrollo interactivo para realizar web crawler. Posee funcionalidades como la recuperación de páginas web en procesos multihilos, la clasificación de contenidos de página y la exclusión estándar del robot. Es de código abierto, disponible para la descarga en cualquier momento. Este crawler ya no posee un seguimiento de los creadores, dejando de ser actualizado desde el año 2002. No posee características relevantes ni comodidad de uso para tenerlo en cuenta en el desarrollo de esta investigación.

1.3.4.2 WebLech

Es un software araña web desarrollado en java con características que le permiten descargar un sitio web completo o hacer un espejo del mismo. Es una herramienta multihilo que posee una consola GUI para interactuar con las opciones brindadas. Esta aplicación dejo de ser actualizada desde junio del 2004. Las características de este crawler no son las necesarias para la implementación de la herramienta necesaria.

1.3.4.3 Arale

Es una araña web desarrollada para descargar sitios web enteros o archivos específicos. Posee funcionalidades como la conversión de sitios dinámicos a paginas estáticas para la posterior revisión de las mismas de forma offline. Posee configuraciones para el trabajo tales como, el número de conexiones simultaneas, la profundidad máxima a recorrer y el tamaño mínimo y máximo de los archivos a descargar. Se le dejo de dar soporte a esta herramienta desde el 2001. Este software no posee características relevantes para tomarlo en consideración.

1.3.4.4 Apache Nutch

Es un web crawler utilizado para extraer información bajo el protocolo HTTP. Es de código abierto y responde a la licencia de Apache Software Foundation. Posee funcionalidades de organización y análisis de contenido proporcionada por la herramienta de gran prestigio en la comunidad, Apache Hadoop, desarrollada para el análisis de Big Data. Es común su integración con otras herramientas como Apache Solr, la cual actúa como repositorio de los datos recolectados.

1.3.4.5 Norconex HTTP Collector

Es un colector de contenidos de sitios web diseñado para proveer información a motores de búsqueda o cualquier otro repositorio de datos. Puede trabajar de forma independiente o ligado a una aplicación que consuma la salida de sus datos. Está

diseñado para trabajar bajo cualquier sistema operativo y posee una documentación muy completa. Se distribuye en un paquete fácil manejo con una pre configuración de ejemplo para su uso inmediato. La compañía Norconex distribuye otro crawler Norconex Filesystem Collector, especialmente de destinado para extracción de información de forma local. Esta es su característica diferenciadora en comparación al HTTP Collector. Ambas son de código abierto y libre distribución. Por todas estas características y por contar con una comunidad que respalda esta herramienta, se escoge la implementación de la aplicación del presente trabajo.

1.3.5 Indexadores

Para seleccionar el indexador que se ajustara a las necesidades de la implementación se tomó como referencia un artículo escrito por Hiberus, una compañía experta en negocios digitales, la cual brinda servicios de SEO, User Experience (UX), análisis de datos en otros. (Hiberus 2018)

Dentro de las tecnologías NoSQL están los motores de búsqueda y es que ya sea para poder hacer frente al big data, construir servicios basados en la nube o desarrollar aplicaciones web con alto tráfico, es vital tener un buscador rápido, fiable y optimizado.

Lucene (1999) se ha convertido en la base de dos de los mejores motores de búsqueda de código abierto de nuestro tiempo: **Apache Solr** (2004) y **Elasticsearch** (2010).

Estos motores posen característica como:

- **Escalables:** capaces de distribuir el trabajo (indexación y procesado de consultas) a múltiples servidores en un cluster.
- **Listo para ser desplegado:** ambas soluciones vienen con ejemplos prácticos para levantar un servicio con el mínimo esfuerzo.
- **Optimizados para búsquedas:** muy rápidos, capaces de ejecutar consultas complejas en decenas de milisegundos.

- **Grandes volúmenes de datos:** están diseñados para lidiar con índices de billones de documentos.
- **Centrados en texto:** aunque soportan búsquedas sobre fechas y números, su base y principal fuerza es manejar textos naturales, extrayendo la estructura implícita del mismo al índice del motor para mejorar la búsqueda.
- **Resultados ordenados por relevancia:** dependiendo de la consulta del usuario se le devuelven documentos clasificados en base a dicha consulta.

Comparativa respecto a varios aspectos:

- **Popularidad:** En estos momentos Elasticsearch cuenta con mucha mayor popularidad que Apache Solr, en cuanto a utilización de los desarrolladores.
- **Comunidad y código abierto:** Ambos tienen comunidades muy activas con gran cantidad de aportes y trabajando ambas bajo la licencia Apache 2.0 pero difieren en un punto: Solr es realmente código abierto, cualquiera puede ayudar y contribuir. Sin embargo, en Elasticsearch solo los empleados de ElasticStack pueden aceptar dichas contribuciones.
- **Documentación:** Ambos tienen una documentación exquisita y muy detallada. Solr la distribuye a través de Atlassian Confluence y Elasticsearch a través de Github.
- **Java APIs y REST:** Elasticsearch al ser más reciente ha basado su modelo en la API REST Web 2.0 mientras que la REST de Solr es menos flexible. Sin embargo, Solr tiene una mejor API Java con SolrJ (SolrNET para sistemas Microsoft), en este aspecto Elasticsearch cuenta con Nest y elastisearch.NET respectivamente. Solr soporta JSON, aunque inicialmente fue construido para XML, por lo que es más reciente esta adaptación mientras que en Elasticsearch tiene JSON de base.
- **Procesamiento de contenido:** Solr puede extraer información de archivos binarios utilizando Apache Tika gracias al ExtractRequestHandler. Elastic puede realizar la misma funcionalidad con Logstash que puede leer de cualquier fuente e indexarla.

- **Escalabilidad:** En este punto es donde Solr pierde posiciones y es parte del gran motivo que lleva a la creación de Elasticsearch. El problema inicial en la escalabilidad de Solr fue no renovar el sistema master-slave ya que es un sistema obsoleto y Elasticsearch ha sabido aprovechar este nicho. Sin embargo, la creación de SolrCloud y la integración con Zookeeper ha hecho posible que Solr escale de manera mucho más rápida y sencilla.
- **Rendimiento:** Desde la experiencia de Hiberus se puede apreciar que no hay diferencias en rendimiento entre uno u otro sistema tanto para aplicaciones de búsqueda internas como externas si es que se diseñan, realizan y utilizan correctamente.

De forma general se puede apreciar que ambos buscadores son sistemas excepcionales gracias a Lucene, con pequeñas variaciones entre ellos. Por lo que se escoge Apache Solr para utilizar en la aplicación a desarrollar por tener un conocimiento y uso previo de este software, lo cual agiliza el proceso de desarrollo.

Teniendo seleccionado el crawler y el indexador a utilizar, se toma como herramienta para la implementación de la aplicación el framework de Java, SpringBoot, el cual cuenta con librerías específicamente diseñada para trabajar con Apache Solr a la vez de ser una herramienta auto-configurada que permite enfocarse en directamente en la implementación.

1.4 Metodología de Desarrollo Utilizada

Una metodología es un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. Es un proceso de software detallado y completo, se basan en una combinación de los modelos de proceso genéricos. Definen artefactos, roles y actividades, junto con prácticas y técnicas recomendadas. La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Comprende los procesos a seguir sistemáticamente para idear, implementar

y mantener un producto software desde que surge la necesidad del producto hasta que cumplimos el objetivo por el cual fue creado.(Maida and Pacienzia 2015)

Ventajas de utilizar una metodología en el proceso de ingeniería de software

- Optimiza el proceso y el producto software.
- Métodos que guían en la planificación y en el desarrollo del software.
- Define qué hacer, cómo y cuándo durante todo el desarrollo y mantenimiento de un proyecto.

1.4.1 Metodología de Software Ágil

Son un conjunto de métodos y metodologías que ayudan al equipo de trabajo a pensar de forma más efectiva, eficiente y tomar mejores decisiones. Estas metodologías son adaptaciones de todas las áreas de la ingeniería de software tradicional, incluyendo la administración de proyectos, diseño de software, arquitectura y mejoramiento de procesos.(Greene 2015)

Estas metodologías promueven el mantenimiento de una continua retroalimentación y la integración de nuevos cambios en los requerimientos de software a través del ciclo de vida del desarrollo de la aplicación. Integra la colaboración cercana entre el cliente y el desarrollador. Implementa las entregas en cortos periodos de tiempo de pequeñas porciones terminadas del software en desarrollo.(Babar, Brown et al. 2014)

1.4.2 Metodología SCRUM

Es un marco de trabajo a través del cual las personas pueden abordar problemas complejos adaptativos, a la vez que se entregan productos productivamente y creativamente con el máximo valor. Es ligero y simple de entender.

Está compuesto de procesos que se ha sido utilizado para gestionar el desarrollo de productos complejos desde principios de los años 90. No es un proceso o una técnica para construir productos, es un marco de trabajo donde se pueden emplear

un conjunto de diferentes procesos y técnicas. Scrum muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo de modo que se pueda mejorar.

Se compone por los Equipos Scrum, sus Roles, Eventos, Artefactos yReglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico lo cual es esencial para el éxito.(Schwaber and Sutherland 2016)

Por todas estas características y su gran utilización y respaldo de excelentes empresas como Adobe, Google, Spotify y Apple que utilizan esta metodología para el desarrollo de sus proyectos, se escoge esta opción para implementar la solución del presente trabajo.

1.4.2.1 Roles de la Metodología

Estos son los roles esenciales para poner en marcha un equipo Scrum según (Schwaber and Sutherland 2016)

Product Owner

El Propietario del Producto (Product Owner) es el responsable de maximizar el valor del producto y el trabajo del Equipo de Desarrollo (Development Team). Cómo se lleva a cabo puede variar ampliamente entre distintas organizaciones, equipos Scrum e individuos. El Product Owner es la única persona responsable de gestionar la Pila del Producto (Product Backlog).

Scrum Master

Es el responsable en asegurar que se entienda y se adopte Scrum. Los Scrum Masters hacen esto asegurándose de que el Equipo Scrum trabaja ajustándose a la teoría, prácticas y reglas de Scrum. Es un sirviente líder que está al servicio del, y para el Equipo Scrum. Ayuda a las personas externas al Equipo Scrum a entender qué interacciones con el Equipo Scrum pueden ser útiles y cuáles no. El Scrum

Master ayuda a todos a modificar estas interacciones para maximizar el valor creado por el Equipo Scrum.

Development Team

El Equipo de Desarrollo (Development Team) se compone en base a los profesionales que realizan el trabajo de entregar un Incremento de producto "Terminado" que potencialmente se pueda poner en producción al final de cada Sprint. Solo los miembros del Equipo de Desarrollo participan en la creación del Incremento.

1.4.2.2 Artefactos de Scrum

Los artefactos de Scrum representan el trabajo o el valor en diversas formas que son útiles para proporcionar transparencia y oportunidades para la inspección y adaptación. Los artefactos definidos por Scrum están diseñados específicamente para maximizar la transparencia de la información clave, necesaria para asegurar que todos tengan el mismo entendimiento del artefacto. Estos son los artefactos según (Schwaber and Sutherland 2016)

Product Backlog

La Pila del Producto (Product Backlog) es una lista ordenada de todo lo que podría ser necesario en el producto y es la única fuente de requisitos para cualquier cambio a realizarse en el producto. El Propietario del Producto (Product Owner) es el responsable de la Pila del Producto, incluyendo su contenido, disponibilidad y ordenación.

Una Pila del Producto nunca está completa. El desarrollo más temprano de la misma solo refleja los requisitos conocidos y mejor entendidos al principio. Evoluciona a medida que el producto y el entorno en el que se usará también lo hacen. Es dinámica, cambia constantemente para identificar lo que el producto necesita para ser adecuado, competitivo y útil.

Sprint Backlog

La Pila del Sprint (Sprint Backlog) es el conjunto de los elementos de la Pila del Producto (Product Backlog) seleccionados para el Sprint, más un plan para entregar el Incremento de producto y conseguir el objetivo del Sprint. El Sprint Backlog es una predicción hecha por el Equipo de Desarrollo (Development Team) acerca de qué funcionalidad formará parte del próximo Incremento y del trabajo necesario para entregar esa funcionalidad en un Incremento Terminado.

El Sprint Backlog hace visible todo el trabajo que el Equipo de Desarrollo identifica como necesario para alcanzar el objetivo del Sprint. Es un plan con un nivel de detalle suficiente como para que los cambios en el progreso se puedan entender en el Scrum Diario (DailyScrum). El Equipo de Desarrollo modifica la Pila del Sprint durante el Sprint y esta Pila del Sprint emerge a lo largo del Sprint. Esto ocurre a medida que el Equipo de Desarrollo trabaja en lo planeado y aprende más acerca del trabajo necesario para conseguir el objetivo del Sprint.

1.5 Patrones Utilizados

Un patrón es la mejor práctica probada para darle solución a un conocimiento o a un problema recurrente dentro de un contexto. Describe un problema cuando ocurre una y otra vez, y luego describe el núcleo de la solución de ese problema, de esa forma se puede utilizar la solución las veces necesarias sin hacerlo de la misma forma.(Ackerman and Gonzalez 2010)

Síntesis

- Proporcionar catálogos de elementos reusables en el diseño de sistemas software.
- Evitar la reiteración en la búsqueda de soluciones a problemas ya conocidos y solucionados anteriormente.
- Formalizar un vocabulario común entre diseñadores.
- Estandarizar el modo en que se realiza el diseño.

- Facilitar el aprendizaje de las nuevas generaciones de diseñadores condensando conocimiento ya existente.

No Pretenden

- Imponer ciertas alternativas de diseño frente a otras.
- Eliminar la creatividad inherente al proceso de diseño.

1.5.1 Patrón de Arquitectura

Un patrón de arquitectura expresa un esquema de organización estructural fundamental para sistemas de software. Proporciona un conjunto de subsistemas predefinidos, especifica sus responsabilidades e incluye reglas y directrices para organizar las relaciones entre ellos.(Buschmann, Meunier et al. 1996)

Estos son los patrones de arquitectura utilizados en el presente trabajo con el objetivo de ganar en claridad de código y seguir las buenas prácticas de la programación.

MVC

El patrón Modelo-Vista-Controlador es un patrón arquitectural que promueve el aislamiento estricto entre las partes individuales de la aplicación. Este aislamiento es mejor conocido como Separación de Responsabilidades (Separation of Concerns) o en términos más generales, bajo acoplamiento. (Chadwick, Snyder et al. 2012)

Este patrón no es nuevo data de 1978 del proyecto Xerox PARC, pero ha ganado enorme popularidad como patrón de aplicaciones web por la siguiente razón:

- La interacción del usuario con aplicaciones MVC siguen un ciclo natural: el usuario realiza una acción y en respuesta la aplicación cambia el modelo de datos y entrega una vista actualizada al usuario. Esto en muy conveniente

para las aplicaciones web las cuales realizan una serie de pedidos y respuestas HTTP.(Freeman 2013)

1.5.2 Patrón de Diseño

Los patrones de diseño son soluciones para problemas típicos y recurrentes que nos podemos encontrar a la hora de desarrollar una aplicación. Aunque nuestra aplicación sea única, tendrá partes comunes con otras aplicaciones: acceso a datos, creación de objetos, operaciones entre sistemas, entre otras. En lugar de todo el mismo código, se puede solucionar el problema utilizando algún patrón, al ser estos soluciones probadas y documentadas por la comunidad de programadores.

1.5.2.1 Inversión de Control

Inversión de control es un principio en la ingeniería de software mediante el cual el control de objetos o partes de un programa se transfiere a un contenedor o framework. Se usa con mayor frecuencia en el contexto de la programación orientada a objetos. A diferencia de la programación tradicional, en la que nuestro código realiza llamadas a una biblioteca, IoC permite que un framework controle el flujo de un programa y realice llamadas a nuestro código.

Ventajas de utilizar IoC

- Desacoplamiento de la ejecución de una tarea desde su implementación, lo que facilita el cambio entre diferentes implementaciones
- Mayor modularidad de un programa
- Mayor facilidad para realizar Test un programa, aislando un componente o realizando mocking de sus dependencias, permitiendo que los componentes se comuniquen a través de contracts.

1.5.2.2 Inyección de Dependencias

El patrón de Inyección de Dependencia se utiliza en casi todos los framework y se basa en el principio de "inversión de control" (IoC). Se relaciona con la forma en que

un objeto obtiene referencias a sus dependencias, el objeto pasa sus dependencias a través de argumentos constructor o después de la construcción a través de métodos setter o métodos de interfaz. Se llama inyección de dependencia ya que las dependencias de un objeto se "inyectan" en él, el término dependencia es un poco engañoso, ya que no se trata de una nueva "dependencia" que se inyecta, sino más bien de un "proveedor" de esa capacidad particular.

1.5.2.3 Patrón Repositorio

Un repositorio, media entre el dominio y las capas de mapeo de datos, actuando como una colección de objetos de dominio en memoria. Los objetos del cliente construyen especificaciones de consulta de forma declarativa y las envían al repositorio para su satisfacción. Los objetos se pueden agregar y eliminar del repositorio, como se podría hacer desde una simple colección de objetos, y el código de mapeo encapsulado por el repositorio lleva a cabo las operaciones apropiadas detrás de escena.

1.6 Herramientas Utilizadas en la Implementación

En la construcción de la aplicación desarrollada en esta investigación se utilizaron varias herramientas de muy alto nivel que son utilizadas a nivel mundial por millones de desarrolladores, los cuales garantizan a través de la retroalimentación con los respectivos dueños, que funcionen de manera estable y precisa.

1.6.1 Intellij IDEA

Es un IDE (Entorno de Desarrollo Integrado) inteligente de Java que provee una combinación robusta de herramientas de desarrollo. Entre sus funcionalidades se encuentra la asistencia inteligente de código, navegación inteligente, búsqueda, varios tipos de refactorización, análisis de código. Sus funcionalidades son continuamente extendidas por sus clientes a través de los plugin. Posee una interfaz de trabajo muy intuitiva de fácil manejo sin perder la robustez de software profesional que ha ido adoptando a lo largo de los años.

1.6.2 Apache Maven

Es un framework de código abierto gestor de proyectos basado en estándar, que simplifica la construcción, las pruebas, los reportes y el empaquetado de proyectos.(Varanasi and Belida 2014)

Antes de la creación de Maven los desarrolladores gastaban mucho tiempo construyendo los proyectos y cada vez que pasaban de un proyecto a otro había una curva de aprendizaje. Maven solucionó este problema introduciendo una interfaz común. (Siriwardena 2015)

1.6.3 Apache Tomcat

Es el más común y popular contenedor web disponible en el mercado. Fue originalmente creado como Sun Java Web Server por ingenieros de Sun Microsystems y se convirtió en la implementación de referencia de Java EE Servlet.

Su principal ventaja es la pequeña footprint, simple configuración y su larga historia de comunidades envuelta en el mejoramiento de este. Típicamente los desarrolladores pueden tener listo y ejecutándose una instalación de Tomcat en 5 o 10 minutos.(Williams 2014)

1.6.4 MySQL

Es un sistema gestor de base de datos relacionales de código abierto, multihilo creado por Michael Widenius en el 1995. (Dyer 2015) Posee una arquitectura cliente servidor. Soporta el lenguaje estándar de base de datos SQL (Structured Query Language) para realizar consultas, actualizar datos y la administración de la base de datos. Es considerado como uno de los programas de base de datos más rápidos que existen.(Kofler 2005)

1.6.5 Sprintometer

Es una simple pero poderosa herramienta para administrar proyectos basados en las metodologías agiles SCRUM o XP. Permite configurar de forma fácil pero muy profesional los componentes de cada una de estas metodologías y exportar los resultados a formatos Excel y ODF.

1.7 Framework Utilizados

Conjunto de clases cooperativas que construyen un diseño reutilizable para un tipo específico de software. Un Framework proporciona la arquitectura partiendo el diseño en clases abstractas y definiendo sus responsabilidades y colaboraciones. Un desarrollador realiza una aplicación haciendo subclases y componiendo instancias a partir de las clases definidas por el Framework.(Riba 2008)

1.7.1 Spring Framework

Es un contenedor de aplicación para Java que provee varias características entre las que se encuentran, Control de Inversión, Inyección de Dependencias, acceso a datos abstractos. Fue concebido en el 2002 en respuesta a quejas en la industria a que las especificaciones de Java EE eran severamente deficientes y muy difíciles de utilizar. Spring Framework es una herramienta de productividad para desarrollar aplicaciones de todo tipo de forma rápida, modular y sencillos códigos de prueba.(Williams 2014)

1.7.2 Springboot Framework

Es una solución configuración-sobre-convención (convention-over-configuration) para crear proyectos stand-alone basados en aplicaciones Spring que pueden simplemente ser ejecutadas. Es un framework pre configurado utilizando librerías de terceros para comenzar a desarrollar lo más rápido posible sin perder tiempo ajustando los elementos que la componen.(Pivotal-Software 2018)

1.7.3 Thymeleaf

Es un motor de plantillas Java moderno del lado del servidor funciona tanto para web como para aplicaciones independientes. Es capaz de procesar HTML, XML, JavaScript, CSS, incluso texto plano.

Su principal objetivo es proveer de una forma elegante y altamente sostenible la creación de plantillas. Para lograr esto implementa el concepto de Plantilla Natural para insertar su lógica dentro de los archivos de plantilla de forma que no afecte la plantilla de ser utilizada como prototipo de diseño. Esto mejora la comunicación de diseño y salva la brecha entre diseño y equipo de desarrollo.(Thymeleaf-Team 2018)

1.7.4 Bootstrap

Es un excelente framework CSS que ofrece gran cantidad de elementos de interfaz de usuario cuidadosamente elaborados, layouts y plugin jQuery. Es de código abierto y se ha convertido en uno de los proyectos más populares de todos los tiempos. Cuenta con un diseño adaptativo de primera calidad capaz de implementar complejos diseños de forma increíblemente rápida. En su implementación contiene de forma preconfigurada estilos para la tipografía, navegación, tablas, formularios, botones y mucho más. Todas estas características están diseñadas de tal forma de que no interfieran en las futuras actualizaciones del producto. (Niska 2014)

1.8 Lenguajes Informáticos Utilizados

1.8.1 Java

Es un poderoso leguaje de programación orientado a objeto desarrollado por Sum Microsystems Inc. en 1991. Fue desarrollado para dispositivos electrónicos de consumo, pero luego fueron redirigidos sus objetivos hacia el desarrollo de aplicaciones para internet. (Chaudhary 2014)

Es un lenguaje de programación de propósito general con características muy relevantes como, sencillo, robusto, multitarea, dinámico y de plataforma independiente.(Spell 2015)

1.8.2 HTML

El Lenguaje de Marcas de Hipertexto (***Hyper Text Markup Language)*** es un lenguaje de marcado utilizado para convertir documentos de texto en páginas web y aplicaciones. Su propósito fundamental es proveer la descripción semántica de los contenidos y establecer la estructura del documento.(Robbins 2013)

1.8.3 CSS

Hojas de estilo en cascada (Cascading Style Sheets) es un lenguaje simple para definir estilos que puede ser aplicado a HTML, donde el HTML describe la estructura de la página web y CSS describe su presentación.

La comunidad internacional World Wide Web Consortium (W3C) escribe y mantiene las especificaciones CSS, a la vez de definir y estandarizar la forma en la que las personas deberían escribir el CSS y los navegadores deberían implementarlo.(Lunn 2013)

1.8.4 JavaScript

Es un lenguaje de alto nivel que es compilado en tiempo de ejecución, esto significa que requiere un motor (engine) que es el responsable de interpretar el programa y ejecutarlo. Los motores más comunes se encuentran en los navegadores, tal como Chrome, Firefox o Internet Explorer. JavaScript es además un lenguaje dinámico, lo que significa que los elementos en el programa pueden cambiar mientras está corriendo. (Jones 2014)

1.8.5 XML

Extensible Markup Language (XML) es un formato universal para datos y documentos estructurados. Los archivos XML tienen una extensión de archivo de xml. Al igual que HTML, XML utiliza etiquetas (palabras delimitadas por los caracteres > y <) para estructurar los datos del documento.(IBM 2018) Fue diseñado para almacenar y transportar datos, es autodescriptivo y cuenta con la recomendación directa de uso de la W3C. (W3schools 2018)

1.9 Conclusión Parcial

Después de analizar una gran variedad de software se demuestra que ninguno de estos cumple completamente con las necesidades del cliente, por tal motivo se decide realizar la implementación de uno para satisfacer todos los requerimientos.

Tras analizar la teoría relacionada con el objeto de estudio, se obtiene un mejor entendimiento de la investigación evitado perder el objetivo de la misma. Estos conocimiento teóricos adquiridos sirvieron de base para seleccionar las herramientas y lenguajes ideales para el desarrollo de la aplicación.

CAPÍTULO II: PROPUESTA DE SOLUCIÓN

2.1 Introducción

En este capítulo será descrito el proceso de desarrollo del software Electrom el cual comienza con el análisis de las herramientas seleccionadas, unido a la descripción del funcionamiento, su arquitectura y la integración entre estas para lograr un software completamente funcional. Se expondrán todo el proceso de trabajo basado en la metodología SCRUM para la fase de planificación e implementación del software y su correspondiente prueba de validación.

2.2 Análisis de Herramientas Utilizadas

2.2.1 Norconex HTTP Collector

2.2.1.1 Características

Las siguientes características fueron extraídas de la página web oficial de Norconex(Norconex-Inc. 2018) y representan una síntesis de por qué seleccionar esta herramienta para explotar sus funcionalidades.

- **Crawler universal:** Los documentos pueden ser enviados a cualquier motor de búsqueda u otros repositorios de datos, con el Commiter apropiado.
- **Facilidad de uso para los desarrolladores:** Casi cualquier característica puede ser cambiada por la implementación de una propia.
- **Fácil de ejecutar:** Completamente documentado y con configuración lista para utilizar.
- **Insertable:** Puede ser insertado en cualquier proyecto java manteniendo o no, la configuración basada en archivo.
- **Fácil mantenimiento:** La configuración puede ser dividida en varios archivos para lograr una mejor sostenibilidad.
- **Reanudable en caso de falla del sistema:** en caso de un error, se puede continuar por donde se afectó el trabajo.

- **Varias plataformas:** Basado completamente en Java, por lo que puede ser utilizado en varios sistemas operativos como Windows, Linux, Unix, Mac o cualquier otro sistema operativo que soporte Java.
- **Código Abierto**: Su código fuente está disponible para la descarga en Github y responde a la licencia de Apache 2.0.

2.2.1.2 Funcionalidades ofrecidas

Estas son algunas de las características más importantes ofrecidas por los colectores de Norconex.

- Aplicación multihilo.
- Admite diferentes intervalos de tarea programadas.
- Puede crawl millones de datos en un solo servidor de capacidad promedio.
- Extraer texto de gran cantidad de formatos de documentos.
- Extraer metadatos asociados al documento.
- Manipulación de metadatos.
- Reconocimiento del lenguaje.
- Soporte para OCR, en imágenes y PDF.
- Captura de pantalla (screenshots) de páginas web.
- Traducción de contenido.
- Generación de título dinámico.
- Configuración de la velocidad del crawler.
- Normalización de URL.
- Soporte para diferentes frecuencias de re-crawling de ciertas páginas.
- Soporte para varios esquemas de autentificación de páginas web.
- Puede almacenar las URL extraídas en diferentes motores de base de datos.
- Entre otras muchas.

2.2.1.3 Formatos soportados

HTTP Collector soporta una larga lista de formatos de archivo, en su página web exponen una lista de 100 tipos diferentes, entre los que se encuentran audio, video, documentos, imágenes, entre otros muchos. Esta lista es solo una parte de los formatos soportados, Norconex utiliza la herramienta Apache Tika para ejecutar esta función, la cual soporta un aproximado de 350 formatos de archivos en su funcionamiento.

2.2.1.4 Distribución

Este crawler se distribuye de forma comprimida en su página oficial (Norconex-Inc. 2018), con un peso de archivo de 80Mb. La actualización de este software se produce de forma regular, alimentada por la comunidad que respalda su funcionamiento cada día. Su código fuente se encuentra en GitHub, donde puede ser descargado y utilizado a conveniencia.

Dentro de la página web de GitHub dedicada a este software se encuentra un chat público dedicado a responder cada una de las preguntas que puedan surgir en la utilización de este crawler. Las preguntas son contestadas por los mismos desarrolladores del software y la respuesta es casi inmediata. Durante el desarrollo del software de esta investigación se hizo necesario utilizar esta vía de comunicación y la respuesta se presenta en el Anexo 1.

El paquete una vez descargado y extraído su contenido, cuenta con una distribución de carpetas muy sencilla, la cual se explica a continuación.

- **apidocs:** es una pequeña documentación que viene integrada en caso de surgir alguna duda durante la implementación.
- **clases:** Este archivo viene por defecto vacío y su función es albergar las nuevas clases desarrolladas por la persona que está haciendo uso del crawler para brindar nuevas funciones.

- **example:** Es una preconfiguración hecha de ejemplo, lista para funcionar y realizar crawl a un sitio web de internet. Cuenta con dos configuraciones, una mínima, con pocos elementos de configuración y una compleja, con muchos más elementos. Se incluye un pequeño archivo de texto donde explica como ejecutar el crawler desde una terminal de comandos.
- **lib:** Es el corazón del HTTP Collector, dentro se encuentran todas las librerías necesarias para que todo funcione de forma correcta.
- **script:** Pequeños script de ejecución.
- **third-party:** En esta carpeta están contenidas todas las licencias a las cuales responde este software.
- Además de las carpetas, vienen en la raíz del paquete de distribución, dos archivos con el nombre “collector-http”, uno para iniciar la ejecución del crawler en Windows y otro en sistemas basados en UNIX.

2.2.1.5 Configuración del Crawler

La configuración se realiza en un archivo XML, que preferiblemente y por organización se coloca dentro de una carpeta en la raíz del crawler. Dentro de este archivo se configuran todas las características necesarias para realizar el web spider. Entre estas características se encuentran, la dirección URL a resolver, la profundidad de búsqueda sobre esta URL y los tipos de archivos a recuperar. Estos son solo algunos de los muchos elementos que se pueden configurar en este archivo XML.

2.2.1.6 Ejecutar Crawler Norconex

La ejecución se realiza de forma muy sencilla, solo ejecutando el siguiente comando en la consola. Este solo se aplica al ejemplo por defecto que trae el crawler, si se quisiera ejecutar otra configuración solo hay que cambiar la dirección donde se encuentra el XML.

Ejecutar la configuración mínima:

collector-http.bat –a start –c examples/minimun-config.xml

Esta primera parte siempre es igual “collector-http.bat –a start –c” porque hace referencia al archivo que se encuentra en la raíz del sistema de archivos. Lo único que cambia en caso de haber introducido un nuevo archivo de configuración es la dirección relativa a la carpeta raíz.

2.2.1.7 Proceso de ejecución del crawl

Los desarrolladores de Norconex han sabido desarrollar un gran producto y demostrar la sencillez de su utilización, pero esto es solo la vista general del software, en su código se esconde muchos años de ingeniería y procesos complejos.

En el momento en que se ejecuta el crawl, comienza la ejecución de una cadena de eventos que propician el objetivo final, la extracción de los datos. Esta secuencia inicia con la URL introducida para procesar, a partir de este punto comienza una secuencia lógica de condiciones a analizar para desencadenar acciones. Entre estas acciones se encuentra:

- La normalización de las URLs.
- El tiempo de espera entre peticiones.
- La extracción de los documentos del sitio.
- El análisis de los documentos.
- Extracción de metadatos y links.
- Guardado de los datos.

Todo este complejo proceso se puede apreciar a plenitud en el siguiente diagrama o en su sitio oficial (Norconex-Inc. 2018), donde se ofrece como multimedia, brindando la posibilidad de interactuar con el mismo.

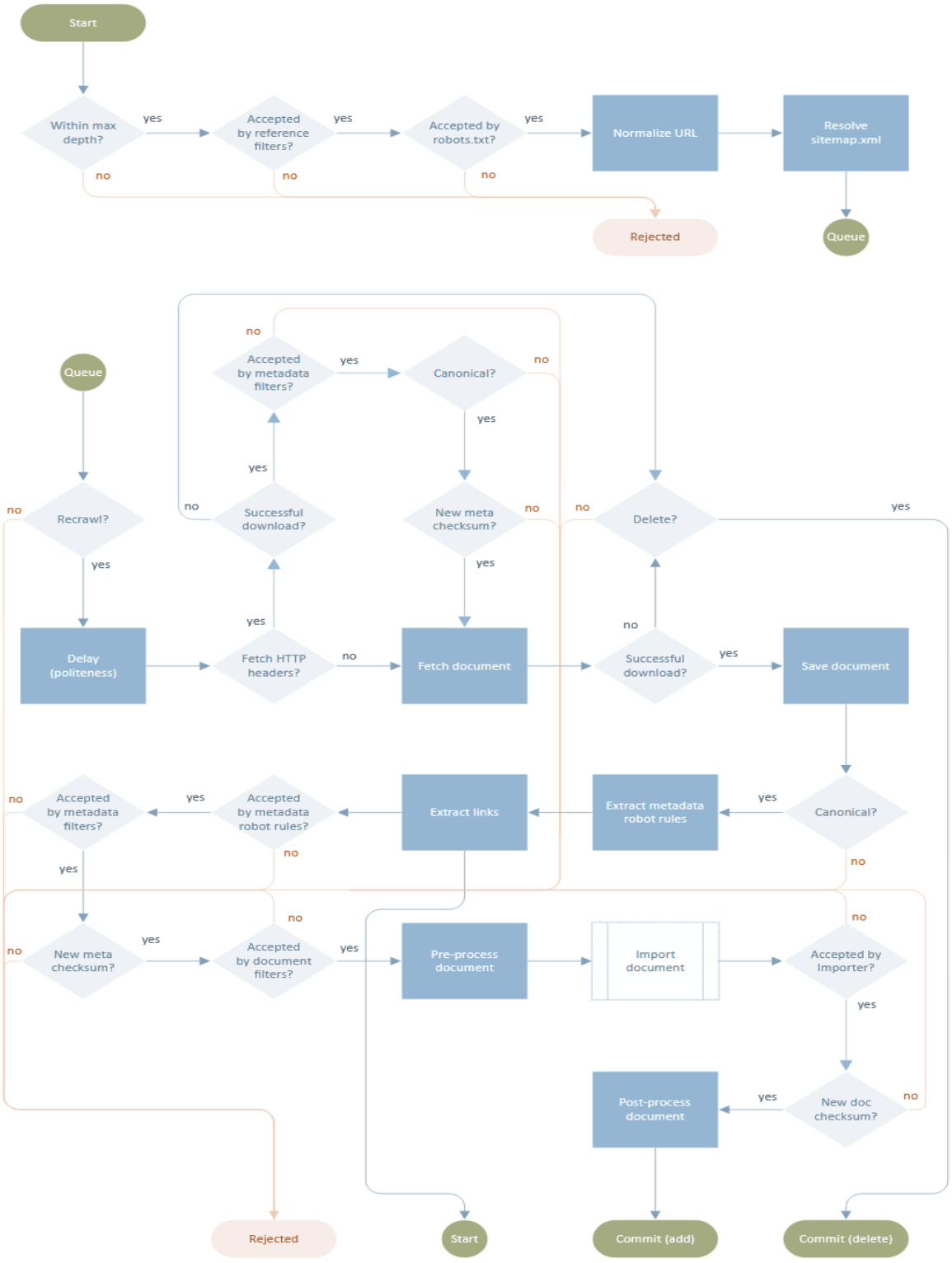


Diagrama 2 Diagrama de Procesos Norconex HTTP Collector

2.2.1.8 Configuración de Norconex en Electrom

El proceso de configuración del crawler para la aplicación Electrom comenzó con la creación de una carpeta de nombre “electrom” en el directorio raíz de Norconex. En el interior de esta carpeta se creó un archivo XML de nombre electrom-config con el objetivo de contener todas las configuraciones necesarias. A partir de este punto se comenzó a introducir configuraciones esenciales para el funcionamiento del crawler, estas configuraciones fueron escogidas de los archivos dela carpeta ejemplo y de la documentación técnica. Estas configuraciones en un primer estado se almacenan de forma estática, para luego ser cambiadas por la aplicación de forma dinámica por la interfaz de usuario. A continuación, se muestran algunas de las más importantes:

URL a realizar crawl

```
<startURLsstayOnDomain="true" stayOnPort="true" stayOnProtocol="true">
<url>http://localhost/tocrawl</url>
</startURLs>
```

Profundidad de búsqueda Máxima

```
<maxDepth>3</maxDepth>
```

Tipos de documentos a buscar

```
<documentFilters>
<filterclass="$filterExtension">pdf,ppt,docx</filter>
</documentFilters>
```

URL del servidorSolr

```
<committer class="com.norconex.committer.solr.SolrCommitter">
<solrURL>http://localhost:8983/solr/electrom</solrURL>
</committer>
```

2.2.2 Apache Solr

2.2.2.1 Características

Apache Solr ofrece gran cantidad de características que pueden ser utilizadas en su implementación, a continuación, se exponen algunas:

- **Búsqueda avanzada de texto:** Desarrollado por Lucene, Solr permite potentes funciones de búsqueda que incluyen frases, comodines, uniones, agrupaciones y mucho más en cualquier tipo de datos.
- **Optimizado para grandes volúmenes de datos:** Es una herramienta probada a nivel mundial por soportar volúmenes de datos extremadamente grandes.
- **Estándar basado en interfaces abiertas:** Utiliza estándar como XML, JSON y HTTP para construir aplicaciones de forma muy rápida.
- **Sencilla interfaz de administración:** Posee una interfaz de administración muy intuitiva que ayuda en el control de las instancias de Sol.
- **Fácil monitoreo:** Solr publica el estado de sus instancias a través de métricas administradas por Java Management eXtensions (JMX).
- **Flexible y adaptable:** Diseñado para adaptarse a las necesidades y simplificar la configuración.
- **Configuración avanzada de análisis de texto:** Solr contiene con soporte para la mayoría de los idiomas ampliamente hablados en el mundo (inglés, chino, japonés, alemán, francés y muchos más) y muchas otras herramientas de análisis diseñadas para hacer que la indexación y la consulta de su contenido sean lo más flexibles posible
- **Seguridad integrada:** Los datos pueden ser protegidos con SSL, autentificación y autentificación basada en roles

2.2.2.2 Distribución

Apache Solr se distribuye desde su página oficial (Apache-Software-Foundation 2018) de forma comprimida en un archivo de 150Mb. En el momento de la creación de este documento se encontraba disponible la versión 7.2. Estas actualizaciones son posibles en gran medida por las contribuciones de la comunidad y su retroalimentación constante.

En el interior del archivo comprimido se encuentra una distribución de directorios como la mostrada a continuación:

- **bin:** Este directorio incluye varios scripts muy importantes para el manejo de Apache Solr, estos son los encargados de iniciar, parar o comprobar el estado de la aplicación, o la creación de nuevos núcleos.
- **contrib:** Contiene plugin para características especializadas de Solr.
- **dist:** Están almacenadas las librerías base para el funcionamiento.
- **docs:** Contiene una página de acceso directo a la documentación online.
- **example:** Este directorio incluye varios tipos de ejemplo de las distintas capacidades de Solr.
- **licenses:** Son incluidas todas las licencias de librerías de terceros.
- **server:** Es el corazón de la aplicación, en este directorio son almacenados los núcleos creados, la administración de Solr, los datos guardados, entre otros elementos.

2.2.2.3 Formas de utilizar Solr

Solr puede ser utilizado de dos formas principalmente, la primera, stand-alone diseñada para tener la información almacenada en un solo núcleo (core) y la segunda, en varios núcleos con varios cluster, este diseño es implementado para distribuir la información en diferentes computadoras para evitar la pérdida de información en caso de algún problema y para distribuir las cargas de trabajo. Este proceso es transparente para el desarrollador, este solo tiene que realizar las

configuraciones necesarias y Solr se encarga de distribuir y balancear la información. En la implementación de Electrom se utilizó el método stand-alone para almacenar el contenido.

2.2.2.4 Comando Utilizados

Durante el proceso de explotación del servidor Solr se utilizaron algunos comandos esenciales para el trabajo y puesta en funcionamiento. Estos se ejecutan en una terminal de comandos desde la carpeta bin ubicada en la raíz de la distribución de carpetas. A continuación se muestran los utilizados:

Ejecutar el Servidor

- *"solrstart"*

Parar servidor

- *"solr stop –p 8983"*
- *"solr stop -all"*

El primer comando para el servidor solr de un puerto en específico, mientras el segundo para todos los servidores.

Comprobar estado del servidor

- *"solr status"*

Crear nuevo núcleostand-alone

- "solrcreate –c electrom"

Borrar núcleo

- "solrdelete –c electrom"

2.2.2.5 Configuración

Cuando es creado un núcleo en Solr este ya tiene las configuraciones necesarias para su uso, esta configuración puede ser modificada a conveniencia al ser el archivo de configuración un XML. Se recomienda por los desarrolladores de Solr que este archivo sea modificado solo cuando se tenga un buen entendimiento de la aplicación. La configuración de Electrom, mantuvo los datos creados por defecto.

2.2.3 Relación de trabajo Norconex HTTP Colector y Apache Solr

Norconex de acuerdo a su configuración va a realizar el proceso de extracción de la información y almacenarla localmente a en la carpeta de salida, esta carpeta es configurada en el XML de configuración general. Para poder enviar la información hacia el servidor de indexado es necesario utilizar el committer específicamente diseñado para apache Solr, proporcionado por Norconex para poder estableces la comunicación entre ambas aplicaciones. Para realizar esta tarea es necesario descargar esta herramienta desde su página principal (Norconex-Inc. 2018), y ejecutar el script de acuerdo al sistema operativo que corresponda. Una vez realizada esta acción se mostrará una terminal de comandos en la cual hay que poner el path completo hasta la carpeta “lib” dentro del directorio de Norconex. Terminado el proceso solo hay que configurar Apache Solr en la configuración global.

2.3 Metodología SCRUM

2.3.1 Descripción de la aplicación

Electrom es una aplicación que cuenta con la habilidad de analizar el contenido de repositorios de documentos compartidos por el protocolo HTTP, extraer los datos previamente configurado y realizar búsquedas en el contenido de los documentos.

Al ser una aplicación para desplegarse en un entorno de servidor web, brinda la posibilidad de ser consultada desde cualquier computadora, teléfono o tablet nodo de la red donde haya sido instalada.

Todas estas características se encuentran agrupadas en un entorno ligero del lado del usuario final en el momento de realizar la consulta, así como de fácil manejo para los administradores, está desarrollada para ser muy intuitiva en su uso y configuración.

2.3.2 Roles en el período de desarrollo

Siguiendo las bases de la metodología SCRUM se asignaron los roles a cada una de las personas que estuvieron presentes en el desarrollo de la aplicación.

Roles en la Metodología SCRUM	Miembro
ProductOwner (Propietario del Producto)	Giovel Peralta Díaz (Director de Electromedicina)
Scrum Master	JosvalDiaz Blanco
DevelopmentTeam (Equipo de desarrollo)	Javier Peralta Díaz

Tabla 2 Roles de la Metodología SCRUM

2.3.3 Usuarios del Sistema

La aplicación está diseñada para trabajar para dos tipos de usuarios los cuales se presentan a continuación:

Tipo de Usuario	Descripción
Administrador	Es el encargado de que funcionen todas las características del sistema. Supervisa el estado de los servidores internos de la aplicación y su configuración.
Usuario del sistema	Persona a la cual va dirigida el sistema. Realiza las consultas sobre los contenidos guardados en el sistema.

Tabla 3 Usuarios del Sistema

2.3.4 Requisitos del Sistema

Electrom es un sistema con funcionalidades de crawl e indexado, pero estas son funciones que solo realiza en un periodo corto de tiempo. Para ejecutar estas funciones, se recomienda realizarlas en un horario donde el servidor donde se instale Electrom no tenga mucha carga de trabajo y pueda explotar todos los recursos de la computadora. Durante el proceso es normal que se consuman todos los recursos disponibles de RAM y CPU. En el trabajo normal de la aplicación solo es necesario 500Mb de RAM libre, por lo que las siguientes especificaciones pueden ser las mínimas recomendadas para su uso:

Requerimiento de Hardware	
Servidor	**Cliente**
CPU: Core I3 4ta Generación **RAM:** 4GB **Disco Duro:** Proporcional a la información a utilizar	Cualquier dispositivo que pueda ejecutar un navegador web

Tabla 4 Requerimiento de Hardware

Requerimiento de Software	
Servidor	**Cliente**
• **Java Development Kit** • **MySQL** • **Apache Tomcat 8+**	Navegador Web

Tabla 5 Requerimiento de Software

Requisitos Funcionales

- Herramienta que permita realizar consultas sobre el repositorio de documentación técnica del Taller de Electromedicina de Matanzas y permita recuperar la información deseada analizando el contenido de los documentos.
- Que la consulta se pueda realizar desde cualquier maquina nodo de la red interna de Electromedicina.

Requisitos no Funcionales

- El software utilizado no debe consumir muchos recursos del servidor central de la institución

Analizando los requerimientos del Product Owner y después de la investigación reflejada en el Capítulo 1, se decide realizar una aplicación web, cumpliendo con uno de los requisitos funcionales descritos. De esta forma se podrá tener acceso a la aplicación desde cualquier punto dentro de la red de la institución.

Dado que los requerimientos no son muy estrictos y solo reflejan una necesidad del cliente, se procede a desarrollar una posible solución y presentársela al Product Owner para comprobar si es correcta la idea del equipo de desarrollo.

Se le sugiere el desarrollo de una aplicación que integre dos aplicaciones escogidas tras una extensa investigación, con funcionalidades muy específicas e importantes para proporcionar el resultado deseado y la integración con una tercera que se encargaría del control de las dos previamente mencionadas.

Se desarrolla junto con el Product Owner una lista ordenada de todo lo que podría ser necesario para el desarrollo del producto, dando como resultado los siguientes requisitos.

2.3.5 Product Backlog

Este es el primer acercamiento a los requisitos para el desarrollo del sistema. De ser necesario agregar más requisitos, se le daría la prioridad requerida y se reordenaría la lista.

Prioridad	Requisitos
1	Diseño del FrontEnd de la aplicación.

2	Diseño del BackEnd de la aplicación.
3	Estudio e investigación del crawler.
4	Configuración del crawler.
5	Estudio e investigación de Apache Solr.
6	Configuración de Apache Solr.
7	Crear de Controlador view Administración.
8	Crear de Controlador view Crawler.
9	Crear de Controlador view Usuario.
10	Servicios de control de server en BackEnd.
11	Servicio XML.
12	Repositorios de interconexión
13	Seguridad de la Aplicación.
14	Servicio de Búsqueda

Tabla 6 Product Backlog

2.3.6 Sprint Backlog

La lista ordenada desarrollada en conjunto entre el Product Owner y el Development Team es la base para la planificación de los Sprint, los cuales dividen esta lista en secciones de trabajo a las cuales se le asigna una fecha de inicio y de fin para el cumplimiento de esta. De la semana solo se cuentan 5 días laborables, 8 horas al día.

# Sprint	ProductBacklog	Fecha Inicio	Fecha Fin
1	Diseño del FrontEnd de la aplicación.	1 Feb 2018	16 Feb 2018
	Diseño del BackEnd de la aplicación.		
2	Estudio e investigación del crawler.	19 Feb 2018	2 Mar 2018
	Configuración del crawler.		
3	Estudio e investigación de Apache Solr.	5 Mar 2018	16 Mar 2018
	Configuración de Apache Solr.		
4	Crear de Controlador view Administración.	19 Mar 2018	6 Abril 2018
	Crear de Controlador view Crawler.		

	Crear de Controlador view Usuario Administrador.		
5	Servicios de control de server en BackEnd.	9 Abril 2018	27 Abril 2018
	Servicio XML.		
6	Repositorios de interconexión	30 Abril 2018	25 May 2018
	Seguridad de la Aplicación.		
	Servicio de Búsqueda		

Tabla 7 Sprint Backlog

Cada una de estos requerimientos contiene una o varias tareas de ingeniería para lograr el completamiento de estos. Las tareas fueron descritas en su correspondiente requerimiento con la ayuda del software gratuito Sprintometer, el cual es un gestor muy completo para la metodología SCRUM y XP.

Sprintometer utiliza plantillas de tareas como la mostrada en la imagen 1.

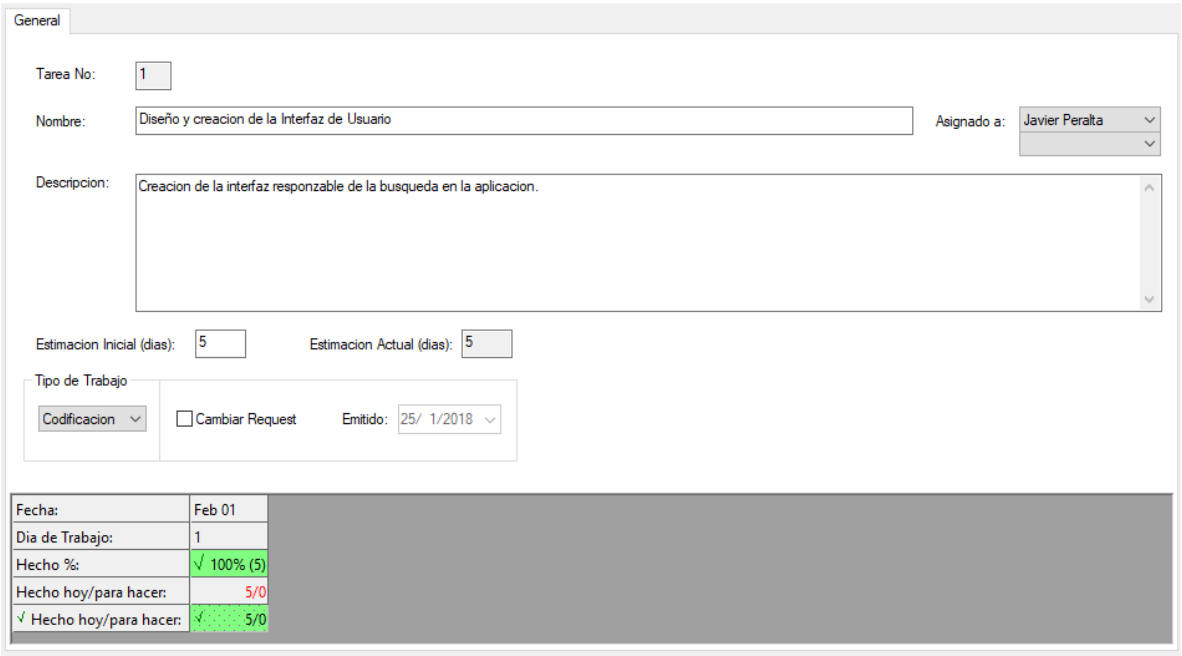


Imagen 1 Plantilla de tarea de Sprintometer

2.4 Validación del Software

En esta sección expondremos las pruebas realizadas para validar el correcto funcionamiento del software desarrollado. Estas tendrán como guía las pruebas de aceptación, las cuales son las últimas pruebas a realizar para ratificar desde el punto de vista del usuario final el correcto funcionamiento del software.

2.4.1 Instalación del software Electrom

La instalación del software fue realizada por el administrador de redes de Electromedicina Provincial de Matanzas, Ing. Enier Molina, en una computadora con características de hardware, motherboard de 4ta generación, con micro i3 y 4gb de memoria RAM. Se siguió el proceso descrito en el manual de usuario. No se encontraron dificultades en el proceso de instalación.

2.4.2 Documentos a indexar

En el proceso de tomar una muestra de los documentos a indexar se contó con el asesoramiento de especialistas de tres departamentos de Electromedicina, los

cuales proporcionaron los manuales técnicos de diferentes equipos afines a su especialidad. En la siguiente tabla se resume todos los datos.

Nombre	Especialidad	Cargo	Tamaño de documentación
Ing. Donald Yáñez	Imaginología	Especialista en equipos de Rayos X	4.58Gb
Ing. Dariel Rivero	Electrónica	Jefe del departamento de Electrónica Medica.	718Mb
Lic. Roberto Ávila	Esterilización	Especialista en equipos de Esterilización.	2.45Gb

Tabla 8 Datos de muestra para indexar

Los datos proporcionados se publicaron en una carpeta compartida en la misma computadora donde se instaló la aplicación y se comenzó el proceso de crawl e indexación. Se realizó un indexado diferente por cada una de las carpetas de documentación para analizar el proceso de trabajo. Los datos proporcionados se registran en la siguiente tabla.

	Tamaño de Documentación	Tiempo de crawl + indexado	Total de archivos indexados
Imaginología	4.58Gb	16min 42seg	286
Esterilización	2.45Gb	30min 22seg	348
Electrónica	718Mb	17min	122

Tabla 9 Tiempo de trabajo de indexación

Los datos de la tabla demuestran la eficiencia del software Electrom, el cual es capaz de procesar grandes volúmenes de datos en un corto periodo de tiempo.

2.4.3 Prueba de búsqueda de información

Después de terminado el proceso de indexado de toda la documentación se realizaron pruebas de búsqueda de información a través de la página que proporciona Electrom para realizar consultas. Las pruebas se reflejan a continuación.

Consultas realizadas a la documentación de Imaginología por su especialista.

#	Consulta realizada	Satisfactorio / Insatisfactorio
1	Ionization chambers	**satisfactorio**
2	MAMMOMAT 1000	**satisfactorio**

3	Quality check	**satisfactorio**
4	Quality check MGU	**insatisfactorio**
5	SCT-7800	**satisfactorio**
6	Tomografía axial	**insatisfactorio**
7	vertical bucky stand	**satisfactorio**
8	Neuviz 16	**satisfactorio**

Consultas realizadas a la documentación de Esterilización por su especialista.

#	Consulta realizada	Satisfactorio / Insatisfactorio
1	Medallist Series	**satisfactorio**
2	Wáter softener	**satisfactorio**
3	Aire control	**satisfactorio**
4	s-140	**insatisfactorio**
5	Calor seco	**satisfactorio**

6	hirayama	**satisfactorio**
7	Sakura FI-371E	**satisfactorio**

Consultas realizadas a la documentación de Electrónica por su especialista.

#	Consulta realizada	Satisfactorio / Insatisfactorio
1	Atom Infusion Pump	**satisfactorio**
2	OT-701	**satisfactorio**
3	Desfibrilador TEC	**satisfactorio**
4	La ley de Joule	**satisfactorio**
5	DOCTUS VI	**insatisfactorio**
6	Transistor PNP	**insatisfactorio**
7	BSM-2300A	**satisfactorio**
8	diode	**satisfactorio**

2.4.4 Análisis de los resultados

Después de realizar diferentes consultas a la documentación indexada, por parte de los especialistas, se contabilizaron los resultados para obtener una perspectiva real del funcionamiento del software.

Resumen	Valor
Cantidad de consultas realizadas	23
Cantidad de respuestas aceptadas devueltas por el software	17
Cantidad de respuestas erróneas o sin resultados	6
Porciento de respuestas correctas	73.91%

El análisis de los datos demuestra la utilidad del software para la búsqueda de información en un repositorio de documentación indexado. Quedando como recomendación para atender las respuestas erróneas o sin resultado, mejorar el reconocimiento de la Expresiones Regulares en la búsqueda de información.

2.4.5 Prueba de Estrés

Al tener la veracidad de que el software funciona de manera correcta se hiso necesario comprobar la estabilidad del software en un proceso mucho más largo de trabajo. Las condiciones y los resultados de la prueba se describes a continuación:

- La prueba se realizó sobre la misma computadora donde se instaló el sistema para realizar las pruebas anteriores
- Tamaño de información a procesar: **20 Gb**
- Tiempo transcurrido para procesar la información: **51 min 42 seg**

- Cantidad de documentos indexados: **1395**

El proceso transcurrió de forma correcta sin mostrarse ningún tipo de error que afectara la tarea, demostrando que el sistema se comporta estable ante grandes volúmenes de información.

Comparando estos resultados con el enfoque tradicional de gestión documental, considerando solo un especialista para llevar a cabo esta tarea de describir meta-datos de al menos 20 documentos diarios, harían falta en término de tiempo para esta colección no menos de 70 días.

Para afrontar la tarea real de 200 Gb de información serían necesarios aproximadamente 2 años, sin contar el incremento periódico de la colección. Para garantizar mejor rendimiento en este enfoque haría falta más personal y equipamiento, no resolviendo el problema con la celeridad necesaria, así como la búsqueda por contenido.

2.5 Conclusión Parcial

Después de analizar las herramientas Norconex HTTP Collector y Apache Solr, quedó demostrada la eficiencia de la unión de estos softwares para recolectar e indexar datos y su posterior recuperación por parte de la herramienta implementada, la cual permite como otra funcionalidad, la simplificación del proceso de configuración.

CONCLUSIONES

El estudio de varias herramientas dedicadas a la gestión de grandes volúmenes de datos, demostró que ninguna de estas resuelve forma total los requerimientos del cliente, por lo que se optó por creación de Electrom, una herramienta basada en los enfoques de los grandes buscadores de contenido, pero implementada aprovechando la existencia de varias herramientas independientes unidas y configuradas como un sistema.

Las pruebas realizadas a Electrom confirmaron la validez de la propuesta y el cumplimiento del objetivo de investigación, permitiendo el procesamiento y recuperación de volúmenes de información considerable en tiempos relativamente cortos con respectos a los sistemas de gestión documental tradicionales.

La mejora del proceso de búsqueda de información tiene un impacto directo en la calidad del proceso de reparación del equipamiento médico en el Taller de Electromedicina de Matanzas, confirmando la veracidad de la hipótesis planteada.

RECOMENDACIONES

- Mejorar el proceso de consultas a la documentación indexada, utilizando lenguajes de descripción de búsqueda más complejo.
- Extender las funcionalidades de Norconex HTTP Collector, como por ejemplo el seguimiento de los procesos del crawler.
- Ajustar el consumo de RAM de acuerdo a los servidores utilizados.
- Enriquecer y actualizar los componentes del modelo propuesto para buscar un mejor índice de recuperación y pertinencia.

BIBLIOGRAFÍA

Ackerman, L. and C. Gonzalez (2010). Patterns-Based Engineering, Addison Wesley.

Aguilar, B. E. C. and C. R. S. Garcia (2013). Repositorio digital de trabajos reeccionales de las licenciaturas de la Escuela Nacional de Biblioteconomia y Archivonomia: Propuesta con uso de software libre Maxico D.F., Escuela Nacional de Biblioteconomia y Archivonomia. **Licenciado**.

Apache-Software-Foundation (2018). "Apache Solr." 2018, from http://lucene.apache.org/solr/.

athento.com (2018). "CARACTERÍSTICAS NUXEO." 2018, from http://www.athento.com/es/nuxeo/caracteristicas/.

Babar, M. A., et al. (2014). Agile Software Architecture. Aligning Agile Processes and Software Architectures. L. Lawrence, Morgan Kaufmann.

Buschmann, F., et al. (1996). Pattern Oriented Software Architecture. A System of Patterns. New York, John Wiley.

Cobos, J. S. (2002). "Gestión documental versus gestión de contenido." El profesional de la información **11**.

Chadwick, J., et al. (2012). Programming ASP.NET MVC 4. R. Roumeliotis, O'Reilly.

Chaudhary, H. H., Ed. (2014). Introduction to Java Programming. Advanced Features (Core Series) Updated To Java 8, Programmer's Mind.

DuraSpace (2018). "About DSpace." from https://duraspace.org/dspace/about/.

Dyer, R. J. T. (2015). Learning MySQL and MariaDB, O'Reilly.

Electromedicina-Matanzas (2017). Dirección Estratégica. OBJETIVOS 2015-2017.

Fagundo, A. M. C. (2016). La evaluación de las Bibliotecas Digitales de Artes y Humanidades cubanas: El caso de la Biblioteca de la Universidad de las Artes. Dpto. de Información y Comunicación de la Universidad de Granada. Granada, Universidad de Zaragoza. **Doctorado**.

Freeman, A. (2013). Pro ASP.NET MVC 5. E. Buckingham, Apress.

GOYAL, K. (2018). "Calibre About." from https://calibre-ebook.com/es/about#history.

Gray, J. and R. Belew (2018). What is Information Retrieval (IR)?

Greene, A. S. J. (2015). Learning Agile. O'Reilly.

Hiberus (2018). "NoSQL y los motores de búsqueda: Apache Solr vs Elasticsearch." from https://www.hiberus.com/crecemos-contigo/nosql-y-los-motores-de-busqueda-apache-solr-vs-elasticsearch/.

IBM (2018). "¿Qué es XML?". from https://www.ibm.com/support/knowledgecenter/es/SSEPGG_8.2.0/com.ibm.db2.ii.doc/opt/c0007799.htm.

Jones, D. (2014). Javascript: Novice to ninja. C. Buckler, SitePoint.

Kitchin, R. (2014). The Data Revolution. SAGE.

Kofler, M. (2005). The Definitive Guide to MySQL5. J. Gilmore, Apress.

Lunn, I. (2013). CSS3 Foundations, John Wiley.

Maida, E. G. and J. Pacienzia (2015). Metodologías de desarrollo de software. FACULTAD DE QUÍMICA E INGENIERIA "FRAY ROGELIO BACON", PONTIFICIA UNIVERSIDAD CATÓLICA ARGENTINA SANTA MARIA DE LOS BUENOS AIRES. **Licenciatura en Sistemas y Computación**.

Manning, C. D., et al. (2008). Boolean retrieval. Cambridge University Pres.

New_Zealand_Digital_Library_Project (2018). "Greenstone." from http://www.greenstone.org.

Niska, C. (2014). Extending Bootstrap. Understand Bootstrap and unlock its secrets to build a truly customized project! S. Pant, H. Shaikh and R. Singh, Packt Publishing.

Norconex-Inc. (2018). "Flow Diagram." from http://www.norconex.com/collectors/collector-http/flow.

Norconex-Inc. (2018). "Norconex Apache Solr Committer Installation." Retrieved 15/3/2018, 2018, from https://www.norconex.com/collectors/committer-solr/install.

Norconex-Inc. (2018). "Norconex features." 2018, from https://www.norconex.com/collectors/features.

Norconex-Inc. (2018). "Norconex HTTP Collector." 2018, from http://www.norconex.com/collectors/collector-http/.

Pivotal-Software (2018). "Spring Boot." from https://projects.spring.io/spring-boot/.

Reyes, R. (2009). Terminología Científico-Social. Diccionario Crítico de Ciencias Sociales. P. y. Valdés.

Riba, J. M. C. (2008). Diseño e implementación de un marco de trabajo (framework) de presentación para aplicaciones JEE.

Robbins, J. N. (2013). HTML5 Pocket Reference, O'Reilly.

Schwaber, K. and J. Sutherland (2016). La Guía Definitiva de Scrum: Las Reglas del Juego.

Siriwardena, P. (2015). Maven Essentials. A. M. Pérez, Packt Publishing.

Spell, B. (2015). Pro Java 8 Programming. S. Anglin, APress.

Structure_Dynamics_LLC (2018). "Open Semantic Framework." Retrieved 2018-02-17, 2018, from http://opensemanticframework.org.

Thymeleaf-Team (2018). Using Thymeleaf**:** 1-104.

Varanasi, B. and S. Belida (2014). Introducing Maven. D. Vohra, Apress.

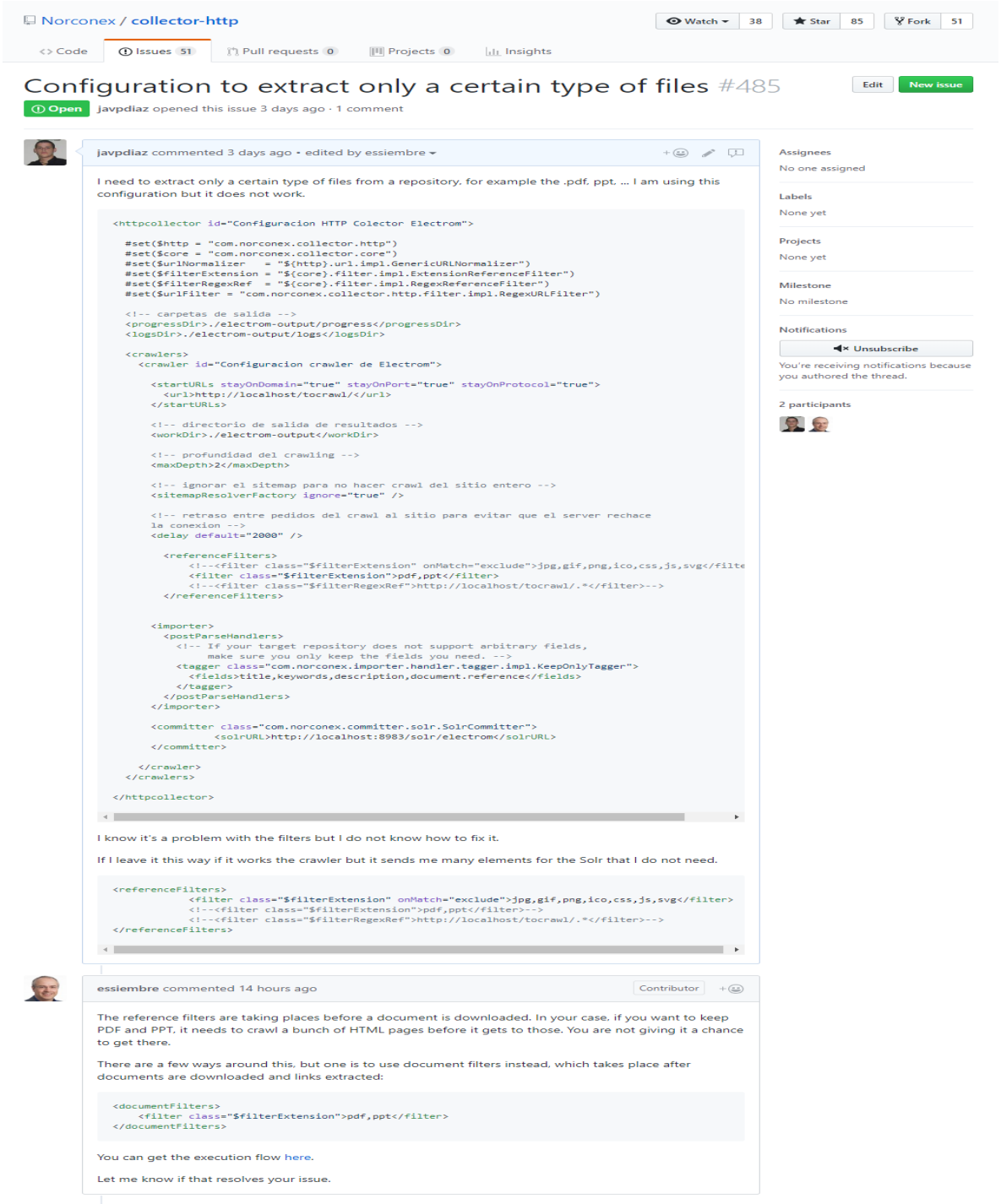
W3schools (2018). "Introduction to XML." from https://www.w3schools.com/xml/xml_whatis.asp.

Williams, N. S. (2014). Professional Java® for Web Applications. J. R. dakovich and M. J. Elera, Wrox Press.

Zayas, D. C. C. M. A. d. and D. C. V. M. S. Lombardía Metodología de la Investigación Científica. La investigacion cientifica en la sociedad del conocimiento.

ANEXOS

Anexo 1



Norconex / collector-http

Watch 38 Star 85 Fork 51

Code Issues 51 Pull requests 0 Projects 0 Insights

Configuration to extract only a certain type of files #485

Edit New issue

Open javpdiaz opened this issue 3 days ago · 1 comment

javpdiaz commented 3 days ago • edited by essiembre

I need to extract only a certain type of files from a repository, for example the .pdf, ppt, ... I am using this configuration but it does not work.

```
<httpcollector id="Configuracion HTTP Colector Electrom">

  #set($http = "com.norconex.collector.http")
  #set($core = "com.norconex.collector.core")
  #set($urlNormalizer    = "${http}.url.impl.GenericURLNormalizer")
  #set($filterExtension = "${core}.filter.impl.ExtensionReferenceFilter")
  #set($filterRegexRef  = "${core}.filter.impl.RegexReferenceFilter")
  #set($urlFilter = "com.norconex.collector.http.filter.impl.RegexURLFilter")

  <!-- carpetas de salida -->
  <progressDir>./electrom-output/progress</progressDir>
  <logsDir>./electrom-output/logs</logsDir>

  <crawlers>
    <crawler id="Configuracion crawler de Electrom">

      <startURLs stayOnDomain="true" stayOnPort="true" stayOnProtocol="true">
        <url>http://localhost/tocrawl/</url>
      </startURLs>

      <!-- directorio de salida de resultados -->
      <workDir>./electrom-output</workDir>

      <!-- profundidad del crawling -->
      <maxDepth>2</maxDepth>

      <!-- ignorar el sitemap para no hacer crawl del sitio entero -->
      <sitemapResolverFactory ignore="true" />

      <!-- retraso entre pedidos del crawl al sitio para evitar que el server rechace
      la conexion -->
      <delay default="2000" />

        <referenceFilters>
            <!--<filter class="$filterExtension" onMatch="exclude">jpg,gif,png,ico,css,js,svg</filte
            <filter class="$filterExtension">pdf,ppt</filter>
            <!--<filter class="$filterRegexRef">http://localhost/tocrawl/.*</filter>-->
        </referenceFilters>


      <importer>
        <postParseHandlers>
          <!-- If your target repository does not support arbitrary fields,
               make sure you only keep the fields you need. -->
          <tagger class="com.norconex.importer.handler.tagger.impl.KeepOnlyTagger">
            <fields>title,keywords,description,document.reference</fields>
          </tagger>
        </postParseHandlers>
      </importer>

      <committer class="com.norconex.committer.solr.SolrCommitter">
                <solrURL>http://localhost:8983/solr/electrom</solrURL>
      </committer>

    </crawler>
  </crawlers>

</httpcollector>
```

I know it's a problem with the filters but I do not know how to fix it.

If I leave it this way if it works the crawler but it sends me many elements for the Solr that I do not need.

```
<referenceFilters>
            <filter class="$filterExtension" onMatch="exclude">jpg,gif,png,ico,css,js,svg</filter>
            <!--<filter class="$filterExtension">pdf,ppt</filter>-->
            <!--<filter class="$filterRegexRef">http://localhost/tocrawl/.*</filter>-->
</referenceFilters>
```

Assignees: No one assigned

Labels: None yet

Projects: None yet

Milestone: No milestone

Notifications: Unsubscribe

You're receiving notifications because you authored the thread.

2 participants

essiembre commented 14 hours ago — Contributor

The reference filters are taking places before a document is downloaded. In your case, if you want to keep PDF and PPT, it needs to crawl a bunch of HTML pages before it gets to those. You are not giving it a chance to get there.

There are a few ways around this, but one is to use document filters instead, which takes place after documents are downloaded and links extracted:

```
<documentFilters>
    <filter class="$filterExtension">pdf,ppt</filter>
</documentFilters>
```

You can get the execution flow here.

Let me know if that resolves your issue.