UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN
CENTRO DE INVESTIGACIÓN EN SISTEMAS DE INFORMACIÓN

Implementación del módulo de indexación y búsqueda para el prototipo de Archivo Web Venezuela para la búsqueda de los contenidos Web bajo el formato WARC.

Trabajo Especial de Grado presentado ante la ilustre
Universidad Central de Venezuela por
Br. Ericka A. Montero Hernández.
Br. Hilda C. Pérez Laya
Para optar al título de Licenciado en Computación
Tutores:
Profa. Mercy Ospina

Caracas, 20/10/2016

UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN

ACTA

Quienes suscriben, miembros del jurado designado por el Consejo de la Escuela de Computación, para examinar el Trabajo Especial de Grado titulado "Implementación del módulo de indexación y búsqueda para el prototipo de Archivo Web Venezuela para la búsqueda de los contenidos Web bajo el formato WARC." y presentado por las bachilleres: Br. Ericka A. Montero Hernández. CI: 20.593.044 y Br. Hilda C. Pérez Laya CI: 20.131.929, a los fines de optar al título de **Licenciado en Computación**, dejamos constancia de lo siguiente:

Leído como fue dicho trabajo, por cada uno de los miembros del jurado, se fijó el día 20 de Octubre de 2016, a las 3:00 pm horas, para que los autores lo defendieran en forma pública, lo que estos hicieron en la Sala PAIII de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual respondieron a las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobar con la nota de 90 puntos.

En fe de lo cual se levanta la presente Acta, en Caracas el día 20 de Octubre de 2016

Profa. Mercy Ospina

(Tutora)

Prof. Andrés Sanoja

(Jurado)

Prof. Hector Navarro

(Jurado)

DEDICATORIA

Este TEG se lo dedico

A ustedes Jacqueline Hernández, (mi mami, mi hermosa), Alexis Montero (mi papi, mi gordito) gracias por estar día a día a mi lado, por consentirme, guiarme y centrarme a la vez.

A ti Barbara Montero, mi hermana, mi motor, la razón por la cual me esfuerzo cada día para ser un buen ejemplo para ti.

A ustedes abuelos, tías, madrina y primas, que siempre han creído incondicionalmente en mi

Y a ti Dios, Santa Babara, tía Magaly y abuela Lucia que desde el cielo siempre me estuvieron apoyando y nunca me dejaron caer.

Ericka Montero

Este TEG se lo dedico

A mis 3 madres Vilma Laya, Teresa Laya y Cruz Laya sin ustedes nunca hubiera podido lograr esta meta, han sido mi apoyo, mi inspiración, mi jalada de oreja cuando algo no está bien, este logro tiene el nombre de ustedes.

A mis tíos que han sido ese padre que en todo los momentos han estado ahí para mí para darme su consejo, su apoyo, esto logro también es por ustedes, Edgar Laya y a José Laya que de seguro está celebrando y corrigiéndome más de un acento y coma en su otra vida.

Y al resto de mi familia por darme su apoyo incondicional, esas palabras de aliento cuando crees que todo está perdido.

Y a ti mi abuela Carmen Laya que desde su otra vida me esta dando todo tu apoyo.

Hilda Pérez.

AGRADECIMIENTOS

Gracias a ti Dios, Santa Barbara por brindarme la fuerza y la esperanza que necesitaba para seguir adelante en este largo camino y sobre todo darme la bendición de tener a mi papá junto a mí.

A ti mami por ser tan especial, atenta y dedicada, gracias por escuchar mis locuras cuando te explicaba mis problemas de la universidad, gracias por hacerme creer en mi cada vez que sentía que ya no podía. A ti papi gracias por estar siempre a mi lado, por dejarme dormir en la camioneta mientras subíamos a Caracas, por tratar de entender lo que hago para poder tener temas en común. Para ustedes no existe palabras de agradecimiento que abarque todo lo que han hecho por mí.

Gracias a ti Barbara Alexandra, mi hermana, mi gordita, la niña que me saca canas verdes, gracias por todas las cosas que me has enseñado sin darte cuenta, por molestarte cada vez que te decía que no podía salir o jugar contigo ya que tenía que estudiar y a pesar de siempre dejarte vestida y alborotada siempre estuviste a mi lado.

Gracias a ustedes mis viejos, por preocuparse, por apoyarme y sentirse orgulloso de mi.

Gracias Tia Virginia por el apoyo incondicional que me dio en el momento que más lo necesite.

Gracias a ustedes primas Susan Tovar y Magaly Tovar por apoyarme y nunca decirme que no.

Gracias a mis amigos Angelica Rojas , Jose Esparragoza, a mi novio Antolin Pereira por siempre animarme cuando les decía que ya no podía más, por creer en mí y decirme que esto es lo mio.

A ti Hilda Pérez mi hermana, mi bffu, sin ti esta travesía no fuera sido posible de realizar, gracias por esos buenos y malos momentos. Gracias por enseñarme tantas cosas, no solo a nivel académico sino a nivel personal. Cuando se confía y se cree en alguien sueños como estos son capaces de realizarse.

Gracias a cada uno de los profesores que formaron parte de mi formación académica, gracias por hacerme enamorar de esta carrera tan difícil y a la vez divertida

A todos los que nombre y los que me falto por nombrar gracias infinitas por realizar y formar parte de este sueño de ser Licenciada en Computación de la excelentísima Universidad Central de Venezuela

Ericka Montero

A ti mama Vilma por ser mi mayor ejemplo, que cuando se quiere no hay ningún obstáculo para evitar cumplir la meta, por ser mi motor para cada día superarme, solo por ti GRACIAS.

A ti mama Cruz por darme mi nalgadas cuando algo estaba haciendo mal, gracias a eso soy la persona que soy hoy en día, gracias por ser también ser una madre para mi, gracias por todos tus consejos, gracias por ser mi ejemplo.

A ti mi vieja, mi madre, mi tía, mi amiga, mi compiche, mi protectora Teresa, gracias por siempre estar para mi , gracias por entenderme , por ser mi profesora en la carrera más difícil la VIDA, te agradezco por simplemente por estar en mi vida y ayudarme a cumplir este sueño.

A ti mi padre, mi tío Edgar, gracias por todos los momentos en que has estado ahí para brindarme todo tu cariño, tus consejos de que los limites se los pones uno y sentir orgullo de mi.

Gracias a ti Reinaldo Laya por ser el hermano que siempre me criticas en positivo pero nunca me falta y regalarme a los dos sobrinos que con sus abrazos me daban fuerza para seguir cuando sentía que no había solución.

Gracias Tía Mercedes, primas Betty, Maria y Nastassjha por creer siempre en mi, por todo el apoyo que me han dado.

Gracias Tía Milena por ser tan especial conmigo, por regalarme la dicha de tenerte como mi familia, gracias por estar siempre pendiente de mi , de darme tu apoyo y ser un ejemplo para mi.

Gracias Tía y Madrina Luisa por ser incondicional para mi, por todo el apoyo que me has dado, y a mi primo querido Daniel por siempre estar pendiente de mi.

Gracias a mis amigas Nancy Benítez, Andrea Pereira por siempre animarme cuando les decía que ya no podía más. Gracias mi playita Gabriela Jurado por la jalada de oreja y enseñarme a usar la borra cuando cometía un error en matemática. Gracias a mi hermana Rocio Tovar, que nos tratamos justo en el momento perfecto, por ser mi amiga, por sus consejos, por todo el apoyo , por creer en mí incondicionalmente. GRACIASS!!!.

A ti Ericka Montero mi bffucita, mi COMPAÑERA de trabajo, mi COMPAÑERA de clase, mi COMPAÑERA DE TESIS, y sobretodo MI HERMANA, gracias por todos los momentos que hemos compartidos juntas, sin ti nada de esto hubiera sido posible, gracias por creer en mí y darme todo tu apoyo.

Gracias a cada uno de los profesores que formaron parte de mi formación académica.

A todos los que nombre y los que me falto por nombrar gracias infinitas por realizar y formar parte de este sueño de ser Licenciada en Computación de la excelentísima Universidad Central de Venezuela.

Hilda Pérez.

Le agradecemos a Dios por habernos acompañado y guiado a lo largo de nuestra carrera, por ser nuestra fortaleza en los momentos de debilidad.

Agradecemos a cada uno de los profesores que fueron parte de nuestra formación académica como profesional. En especial a nuestra tutora Mercy Ospina por confiar y creer en nosotras, por darnos ese voto de confianza, gracias por su apoyo y su dedicación. Al profesor German gracias por su apoyo incondicional y preocupación para que este trabajo fuese logrado con éxito. Gracias al profesor Andres Sanoja por estar siempre a disposición y guiarnos para realizar cada pasó de manera correcta. Gracias a la profesora Paola Saputelli por todos sus consejos, por escucharnos y por ser más que una profesora una amiga. A todos les damos las gracias.

A nuestros familiares, a nuestros amigos, a los nuevos compañeros, a nuestra empresa de trabajo TCS, gracias por creer en nosotras, por darnos su voto de confianza y estar en los momentos buenos y malos, gracias por alentarnos a seguir adelante, en confiar en nosotras, en fin, gracias infinitas.

Este trabajo especial de grado lleva el nombre de todos ustedes

Ericka Montero y HildaPerez

Universidad Central de Venezuela.
Facultad de Ciencias
Escuela de Computación
Centro de Investigación de Sistemas de Información

Implementación del módulo de indexación y búsqueda para el prototipo de Archivo Web Venezuela para la búsqueda de los contenidos Web bajo el formato WARC.

Autor(es): Ericka A. Montero Hernández.
Hilda C. Pérez Laya.

Tutora: Profa.Mercy Ospina.

Fecha: 20/10/2016.

RESUMEN

El patrimonio cultural define a los pueblos y representa su herencia histórica, por lo cual es importante su preservación. El patrimonio cultural se ha clasificado como tangible (obras materiales) o intangible (generación y transmisión de conocimiento, costumbres, etc.), dentro de este último se define el patrimonio digital como aquel que ha sido originado de manera digital y abarca recursos como páginas Web, bases de datos, libros digitales, material multimedia, grabaciones, programas informáticos, entre otros. Entre estos se tienen los recursos Web que se diferencian de cualquier otro tipo de recurso digital por su naturaleza cambiante y su estructura de hiperenlaces, por lo que su preservación tiene características propias y retos que deben ser abarcados por los Archivos Web, los cuales son sistemas de información cuya finalidad es la preservación histórica de estos recursos conocidos como patrimonio Web.

Una de las actividades importantes en el proceso de preservación Web es la indexación de los contenidos almacenados, ya que permite búsquedas más rápidas y eficientes. En la actualidad se está desarrollando un prototipo de Archivo Web para la preservación de Páginas web en Venezuela, actualmente en su versión 2. El presente Trabajo Especial de Grado propone una mejora del módulo de gestión de almacenamiento e indexación del prototipo, donde el contenido se está almacenando en un clúster Hadoop en el formato de almacenamiento para Archivos Web denominado WARC, con búsquedas por URL y por palabras clave que requieren del procesamiento de los WARC. Para la gestión de los índices se usa la herramienta de búsqueda SolrCloud la cual se puede integrar al sistema Hadoop. Este módulo se desarrolló usando la metodología de desarrollo basado en componente.

Palabras Claves: Archivo Web, preservación Web, Formato WARC, indexación, almacenamiento, rastreo, SolrCloud, Hadoop.

Índice de Contenido

Índice de Figuras

Índice de Tablas

INTRODUCCIÓN

La herencia cultural de una comunidad son sus creencias, su costumbres, doctrinas, son los bienes culturales que la historia le ha legado a una nación las cuales se transmite de generación a generación, a esto se le denomina patrimonio cultural, es la herencia recibida de los antepasados, y que viene a ser el testimonio de su existencia, de su visión de mundo, de sus formas de vida y de su manera de ser, y es también el legado que se deja a las generaciones futuras. Este patrimonio puede ser tanto tangible (objetos arqueológicos, artísticos, religiosos, entre otros) como intangible (costumbres, tradiciones, conocimiento científico, entre otros). En este último entraría la categoría de patrimonio digital, el cual está en formato digital y abarca muchos ámbitos como páginas Web, material multimedia, documentos digitales, grabaciones, programas informáticos, entre otros. Dentro del patrimonio digital se puede resaltar aquel que es publicado por media de la Web, denominado patrimonio Web, el cual ha tomado una gran relevancia debido a que se ha convertido en una valiosa fuente de información, pero que tiene una naturaleza inestable y pasajera, por lo que no existe garantía de que un contenido que hoy se encuentre en línea se mantenga en el tiempo, esto es lo que hace imprescindible que se busquen mecanismos para la preservación del patrimonio Web.

El contenido que puede almacenar el patrimonio web, tiende a cambiar con facilidad por lo que puede perderse en algún momento, sea por modificaciones o eliminación del mismo, ya que este no es autopreservable. Es por esta razón, que se han desarrollado mecanismos para conservar los recursos digitales que están en la web. A las actividades asociadas a este proceso de conservación, se les ha dado el nombre de preservación web y, como su principal objetivo, busca almacenar la información web relevante para cualquier rama del saber en un lugar alterno, a su localidad actual, de manera segura. La acción de preservar la web implica muchas cosas y entre ellas están, seleccionar un conjunto de contenidos web (páginas web), esto se hace por la razón de que es inviable preservar toda la web, por lo tanto hay que decidir cuáles son los contenidos a preservar, luego de esto hay que realizar la adquisición de dicho contenido de manera periódica, lo que se busca con esto es poder capturar posibles cambios en el contenido de la web, esto se debe a que los contenidos web son de naturaleza dinámica por lo que tienden a cambiar con el tiempo.

En las últimas décadas, a nivel mundial, se han ido implementando iniciativas para realizar la actividad de preservación de la web, por medio de técnicas y métodos que permitan almacenar la información de forma que esta pueda ser accedida posteriormente por medio de un navegador web. Para poder realizar esta tarea se crearon los sistemas para la Preservación Web los cuales se denominan Archivos Web, siendo este el tema de investigación a desarrollar, temática importante pero poco abordada a nivel nacional.

Actualmente en el país, en la Escuela de Computación de la Facultad de Ciencias de la Universidad Central de Venezuela se está desarrollando un prototipo de Archivo Web como una iniciativa que busca preservar los sitios web del país y que se encuentra funcionando en

un servidor para poder rastrear y preservar dichos sitios web. Estos rastreos realizados han producidos una gran cantidad de información relacionada al contenido preservado, que hasta la fecha no están siendo indexados para aprovecharlo en su totalidad, y para lograr esto se ha hecho necesario conocer cómo acceder a dicha información, su estructura y su semántica, para que de esta de manera se pueda crear índices que puedan ser utilizados para facilitar y mejorar los tiempos de respuesta de las búsquedas realizadas al prototipo.

Por lo expuesto anteriormente este Trabajo Especial de Grado propone estudiar la información, estructura y semántica de las páginas web preservadas para poder generar una solución de estructuras de índices que permita mejorar los tiempos de respuesta de las búsquedas del Prototipo de Archivo Web de Venezuela.

Este trabajo está estructurado en cuatros capítulos del siguiente modo:

- Capítulo I - Planteamiento del Problema: Contiene el planteamiento del problema donde se describe la problemática general asociada a la preservación Web que esta investigación identificó para dar solución. De la misma forma se justifica y se establece el alcance de este T.E.G.
- Capítulo II - Marco Conceptual: Donde se realiza la explicación de los distintos conceptos que permiten comprender la investigación.
- Capítulo III- Marco Metodológico: Sección donde se describe la metodología de desarrollo utilizada para el desarrollo e implementación del Módulo propuesto.
- Capítulo IV Desarrollo de la Aplicación: En este capítulo se detallan los componentes y las actividades ejecutadas, técnicas e instrumentos referentes al desarrollo de la aplicación.
- Posteriormente se dan las conclusiones y recomendaciones consideradas por el equipo de esta investigación a tomar en cuenta para futuras extensiones de este proyecto o futuras investigaciones.
- Por último se listan las referencias bibliográficas consultadas para la investigación, desarrollo e implementación del presente T.E.G.

CAPÍTULO 1
PROBLEMA DE INVESTIGACIÓN

En este capítulo se plantea el problema a solventar y se explica el contexto actual. También, se especifican los objetivos, la justificación y el alcance definidos para esta investigación.

1.1. Planteamiento del problema

Actualmente se está desarrollando en la Universidad Central de Venezuela, un Prototipo de Archivo Web para la preservación de sitios web de venezolanos, el cual se ha ido realizando de manera modular e incremental. En la versión 1 se desarrollaron los siguientes módulos: un módulo de adquisición (productor), un módulo de indexación y almacenamiento (archivo), y un módulo de acceso al archivo web (consumidor), los cuales se puede apreciar en la **Figura 1.**

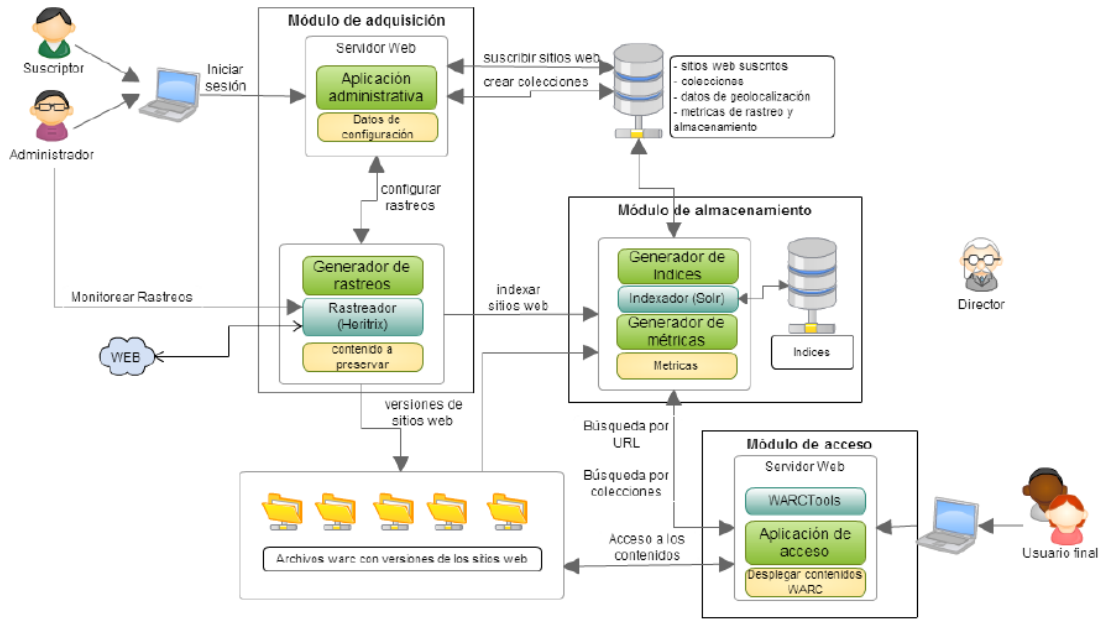

Figura 1 - Arquitectura Del Prototipo De Archivo Web.
Fuente: Un Marco De Referencia Para La Implementación De Archivos Web (Ospina Torres M. H., 2014)

El módulo de adquisición se encuentran las herramientas de rastreo, las cuales se encargan de inspeccionar la World Wide Web de forma metódica y automatizada, con la finalidad de crear una copia de todas las páginas Web visitadas para su posterior procesamiento, a este proceso se le denomina rastreo o cosecha. (Rivero & García, 2013)

El módulo de indexación y almacenamiento es donde se almacenan las páginas web cosechadas, para luego ser indexadas, y así facilitar futuros accesos a dichas páginas. (Rivero & García, 2013)

El módulo de acceso permite al usuario ingresar y ver el contenido de los archivos almacenados, de acuerdo a la búsqueda que hayan realizado. (Kabchi & Martínez, 2013)

En esta iniciativa, se está generando una gran cantidad de información de manera constante, a partir de los rastreos realizados a un conjunto de sitios web venezolanos seleccionados y ha presentado problemas de escalamiento, por lo que se comenzó a desarrollar la versión 2 que busca que los contenidos sean almacenados en un cluster Hadoop. Además solo se efectúa la búsqueda del contenido por medio de la URL del sitio web y el lapso tiempo de respuesta que se tiene de dicha consulta no es el esperado.

La problemática anteriormente presentada se debe a funcionalidades aun no implementadas:

1. No se posee un servidor con arquitectura distribuida.
2. No se realiza búsquedas por otro atributo que no sea la url del sitio web
3. No se tienen estructuras distribuidas de índices.

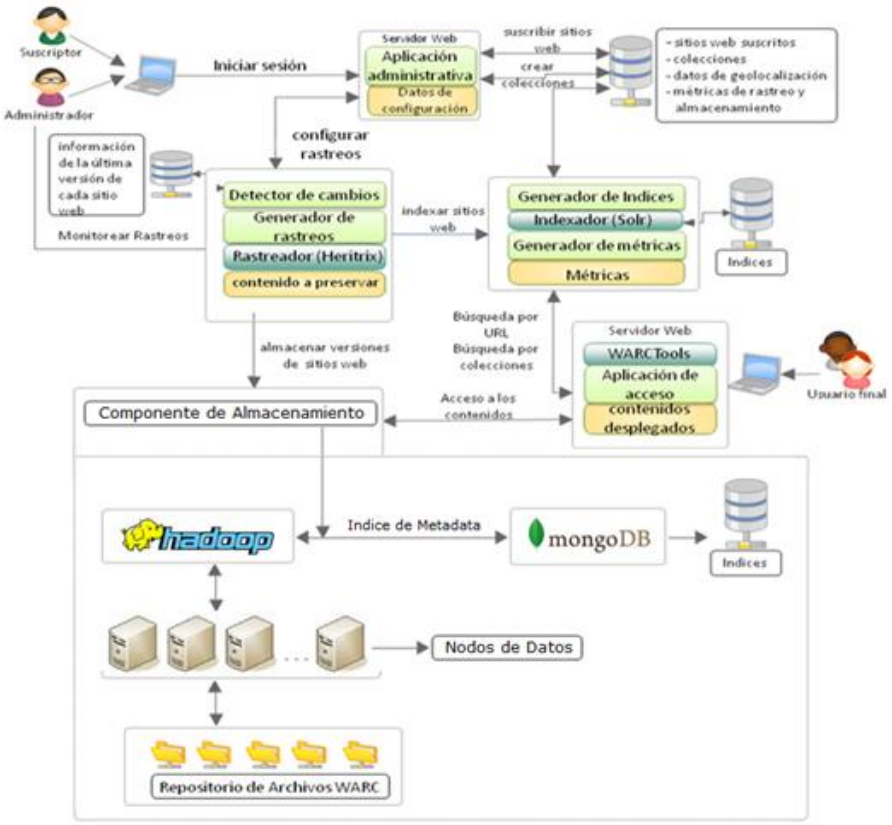

Figura 2 - Arquitectura del Prototipo de Archivo Web con Escalabilidad Horizontal
Fuente: (Estaba Fernández-Trujillo. & Ciancia Biondo, 2015)

En la **Figura 2** se muestra la Arquitectura con la modificación añadiendo el nuevo Módulo de Gestión de Almacenamiento. Este módulo se desarrolló pensando en modificar lo menos posible la Arquitectura del Prototipo de Archivo Web, por lo cual permite escalar horizontalmente y replicar los archivos WARC, para perdurar en el tiempo, utilizando un clúster de Hadoop, donde se puede manejar gran volumen de información y obtener un resultado aceptable de respuesta en caso de que se requiera y Mongo como Base de Datos NoSQL para generar un índice sobre los WARC almacenados. El proceso de carga de archivos en los Nodos de Datos de Hadoop y la carga de información en Mongo es totalmente transparente para los demás componentes y fue implementado en el TEG (Estaba Fernández-Trujillo. & Ciancia Biondo, 2015).

En el presente T.E.G. se plantea la implementación del módulo de indexación y búsqueda donde se efectúan las funciones de indexación y búsqueda mejoradas que darían solución a los problemas planteados en aras de contribuir con el adecuado proceso de preservación de Archivos Web en Venezuela.

1.2. Objetivos del TEG

1.2.1. Objetivo General

Implementar el módulo de indexación y búsqueda para el prototipo de Archivo Web de Venezuela que permita la búsqueda de los contenidos Web almacenados en un clúster Hadoop bajo el formato WARC.

1.2.2. Objetivos Específicos

- Adaptar el método de desarrollo basado en componentes para la implementación del módulo de Indexación y Búsqueda.
- Analizar el contenido de los WARC existentes en el repositorio para conocer detalladamente la información almacenada, para su posterior uso en los índices.
- Establecer las estrategias de indexación adecuadas según las técnicas y tipos de indexación ofrecidas por SolrCloud.
- Realizar pruebas funcionales y de rendimiento del módulo de indexación y búsqueda.
- Integrar la solución de indexación y búsqueda con los módulos existentes de la versión 2 del prototipo de Archivo Web en desarrollo.

1.3. Justificación y Alcance

Debido al desarrollo de la versión 2 del prototipo donde se cambia el esquema de almacenamiento y a que en la primera versión no se contemplaron funcionalidades de búsqueda avanzadas sobre los contenidos almacenados en formato WARC, consideramos que este trabajo de investigación se justifica de manera que el módulo a desarrollar permita:

- Mejorar el tiempo de respuesta a las consultas realizadas al servidor de Archivo Web Venezuela.

- Mejorar los tipos de búsquedas realizadas a los contenidos almacenados en el servidor.

Este Trabajo Especial de Grado propone una nueva versión del Módulo de indexación y búsqueda del Prototipo de Archivo Web de Venezuela, dentro del alcance de este T.E.G. se contempla proveer una nueva estructura de índices, así como permitir las búsquedas por la URL del sitio web, su contenido. Dichas búsqueda será implementada por medio de una interfaz web que permita realizar la consulta al servidor donde se encuentre alojado el prototipo de Archivo Web Venezuela.

En este trabajo no se prevé el despliegue de los WARC para mostrar la página resultado de la búsqueda.

Una limitante existente es la escasa documentación que existe sobre cómo realizar el tratamiento del contenido web en los WARC, debido a que dicho contenido está formado por un conjunto de tecnologías que van cambiando de manera constante en el tiempo, y muchas veces este contenido es generado de forma dinámica, característica que dificulta aun poder realizar el tratado de la información contenida en dichos archivos para su posterior clasificación, almacenamiento y generación de índices para el Prototipo de Archivo Web.

1.4. Antecedentes

A nivel mundial muchos países se han esforzado en desarrollar lo que se ha denominado iniciativas de preservación Web cuyo objetivo es preservar la Web de sus propios países, aunque algunas tienen un alcance mayor. En el portal de la International Internet Preservation Consortium (IIPC) están reseñadas las iniciativas de los países miembros que han construido sus Archivos Digitales (**Figura 3**).

Figura 3 - Países que hospedan iniciativas de Preservación Web
Fuentes: (Proyecto mundiales que trabajan en la Preservacion Web, 2016)

Además para garantizar funciones de preservación se han desarrollado formatos de archivos contenedores específicos, cuyo objetivo principal es superar la limitación de los sistemas de archivos propios de los sistemas operativos, donde se alojan los Archivos Web, y almacenar a su vez el contenido a preservar, siendo uno de los estándar más reconoció el formato WARC, desarrollado por el IIPC (International Internet Preservation Consortium) para la preservación Web como una mejora del formato ARC. A pesar de que los archivos WARC disminuyen la cantidad de documentos a procesar y almacenar, se generan nuevos inconvenientes si estos son de gran tamaño, ya que aumenta el tiempo de respuesta al tratar de extraer la información para futuros accesos, por lo que varias iniciativas o proyectos proponen trabajar bajo un framework distribuido como lo es Hadoop y/o Solr como herramienta para la indexación y búsquedas, estas los podemos visualizar en la tabla **Tabla 1**.

Tabla 1 – Iniciativas Y Proyectos que usan Hadoop y/o Solr

Iniciativas / Proyectos	Almacenamiento	Formato de almacenamiento	Indexador
Padicat (Ver 2.2.2.)	Hadoop y Wayback	ARC o WARC	NutchWax
Archivos Web de la Biblioteca del Congreso	Wayback	ARC o WARC	Curl
Webarchive-indexing	Hadoop	WARC	Sorl
Discover HDP 2.1	Hadoop		Sorl
Cloudera	Hadoop		Solr
Hue	Hadoop		Solr
Portuguese Web Archive (Ver 2.2.3.)	Hadoop y Wayback	WARC	Nutchwax
Internet Memories Foundation (Ver 2.2.4.)	Hadoop	WARC	HBASE

CAPÍTULO 2
MARCO CONCEPTUAL

En este capítulo se van a describir los conceptos y las bases teóricas del tema que permiten entender el trabajo realizado.

Se da inicio del capítulo haciendo una pequeña introducción a los procesos de Preservación Digital y Preservación Web, posterior a esto adentraremos en los Archivos Web como herramientas de tales procesos, tanto a lo referente a su definición como a sus antecedentes.

Seguidamente ya ubicados en el contexto Archivo Web de Venezuela se describe el estado actual de avance del mencionado prototipo, el cual constituye el marco referencial de esta investigación. En este sentido se presentan aspectos como los procesos de adquisición, almacenamiento, indexación y los motores de búsqueda, la herramienta de indexación y búsqueda, tecnologías usadas, la arquitectura de la aplicación, las interfaces y el lenguaje de programación.

2.1. El Patrimonio Web y La Preservación Web

2.1.1. El Patrimonio Web

El patrimonio cultural de una comunidad está formado por los bienes culturales que la historia le ha otorgado a una nación y a los que la sociedad les otorga una especial importancia histórica, científica, simbólica o estética. Es la herencia recibida de los antepasados, y que viene a ser el testimonio de su existencia, de su visión de mundo, de sus formas de vida y de su manera de ser, y es también el legado que se deja a las generaciones futuras.

En búsqueda de mantener y preservar la información que se encuentra presente en Internet, la preservación digital juega un papel importante. Podemos definir a la preservación digital como el almacenamiento de recursos que permiten la consulta posterior de los mismos, la cual puede ser de índole académica, profesional, geopolítica, económica, social o para investigación personal.

"La preservación digital puede definirse como el conjunto de los procesos destinados a garantizar la continuidad de los elementos del patrimonio digital durante todo el tiempo que se consideren necesarios.

El patrimonio digital está formado por los materiales informáticos de valor perdurable dignos de ser conservados para las generaciones futuras, y que proceden de comunidades, industrias, sectores y regiones diferentes. No todos los materiales digitales poseen valor perdurable, pero los que lo tienen exigen metodologías de conservación activas para mantener la continuidad del patrimonio digital." (UNESCO, 2003)."

El patrimonio digital se encuentra conformado a su vez por el patrimonio Web, de hecho se puede decir que el patrimonio Web es un subconjunto del patrimonio digital, que incluye a los documentos Web así como también contendrá todos los elementos que la conforman,

como lo son las hojas de estilo, imágenes, scripts. Para preservar este tipo de patrimonio hay que contar con técnicas y la infraestructura adecuada

Existe una clasificación del patrimonio Web establecida por la UNESCO (Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura) dicha clasificación se detalla a continuación:

- **Patrimonio Intangible:** son expresiones y prácticas culturales, que puede estar constituido, por la poesía, los ritos, los modos de vida, investigación científica, la medicina tradicional, la religiosidad popular y las tecnologías tradicionales de cada país.
- **Patrimonio Tangible:** Comprende los objetos arqueológicos, históricos, artísticos, etnográficos, tecnológicos, religiosos y aquellos de origen artesanal o folklórico que constituyen colecciones importantes para las ciencias, la historia del arte y la conservación de la diversidad cultural del país.

El patrimonio Web y en general el patrimonio digital posee problemas asociados a su preservación, estos problemas no son sólo de carácter técnico sino que además tienen dimensiones sociales y de organización, por esta razón la UNESCO se interesó, ya que entre sus objetivos tiene el fomentar y permitir la preservación del patrimonio cultural, científico e informativo de los pueblos del mundo, por ello difícilmente podía ignorar el crecimiento y la vulnerabilidad que amenaza al patrimonio digital.

2.1.2. La Preservación Web

La preservación Web, conocida también como Web Archiving es el proceso de coleccionar parte del patrimonio web y asegurar que esta colección sea preservada para futuras consultas e investigaciones.

Dicha preservación tiene la responsabilidad de garantizar que el contenido logre ser reproducido de manera similar a como funcionaba en su entorno operativo, es por esto que el conjunto de páginas seleccionadas para ser conservadas son mantenidas completas, en la preservación web se va a preservar diversos tipos de contenido web, incluyendo páginas web HTML, hojas de estilo (CSS), archivos de JavaScript, imágenes y vídeo. Los lugares donde se realiza la preservación y se almacenan los contenidos se denominan Archivos de la Web o Archivos Web. Para llevar a cabo la preservación web hay que llevar a cabo las siguientes tareas:

La IIPC ha definido cuatro (4) actividades para la preservación web, las cuales se pueden visualizar en la **Figura 4.**

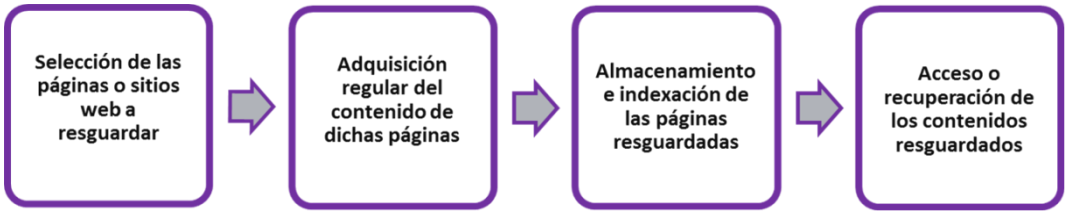

Figura 4 - Actividades para la Preservación Web
Fuente: Desarrollo De Una Aplicación Para Acceder A Contenidos De Un Archivo Web En Formato Warc (Ospina, Martínez, Kabchi, & León, 2014)

A continuación, se presenta una breve descripción de estas actividades:

- La selección permite limitar el ámbito del archivo, pudiendo preservar contenidos locales o de un tipo en particular, como por ejemplo, contenidos de un país o educativos solamente. (Ospina, Martínez, Kabchi, & León, 2014)
- La adquisición logra que se puedan almacenar los cambios generados sobre los contenidos que se preservan a través del tiempo. (Ospina, Martínez, Kabchi, & León, 2014).
- El almacenamiento requiere estrategias que permitan preservar grandes volúmenes de información (del orden de los Terabytes), millones de archivos y diferentes formatos. Para este fin, se han desarrollado formatos de archivos contenedores específicos, cuyo objetivo principal es superar la limitación de los sistemas de archivos propios de los sistemas operativos donde se alojan los Archivos Web. (Ospina, Martínez, Kabchi, & León, 2014).
- El acceso o recuperación de los contenidos está estrechamente ligado a la forma en que se encuentran almacenados, pero debido a la naturaleza hipertextual y multimedia de la web, se espera que el usuario final pueda acceder a este contenido de manera similar a cuando lo hace en los servidores originales. (Ospina, Martínez, Kabchi, & León, 2014).

Por lo expuesto anteriormente, se puede observar que el proceso de realizar la preservación web no es sencillo. Planear y llevar a cabo la preservación web es un proceso que puede resultar costoso, ya que requiere la dedicación de recursos humanos, financieros y técnicos, sin embargo, es necesario tomar en cuenta que la información y el conocimiento son los principales puntos de valor que se quiere preservar, ¿qué pasaría si se pierde el acceso a la colección de tesis, artículos científicos o libros? ¿o si desaparece la información de un proyecto de investigación?, sin duda los beneficios a largo plazo justifican cualquier inversión que se realice en esta área, pero aunque que estemos consiente de que preservar la web es necesario, no se escapa de la realidad la gran complejidad que acarreada este proceso y como aumenta de manera exponencial puesto que no se puede distinguir la gran cantidad de información digital que el mundo produce cada año y es que en realidad se ignora la verdadera y enorme dimensión de tratar de preservar la web.

2.2. Archivo Web

Conociendo la importancia de preservar la Web, muchos países se han comprometido y esforzado en hacer iniciativas para preservar la misma en su localidad, actualmente algunas de estas iniciativas se han ido trazando la idea de ampliar un poco más el dominio o alcance de su sistema, dando a surgir el concepto de Archivo Web.

Los Archivos Web son sistemas de información que surgen para archivar, de manera histórica, documentos que están publicados en la Web, considerados parte del patrimonio digital de las naciones. Se puede definir un documento web como un documento basado en el lenguaje de marcas HTML1 (HyperText Markup Lenguage), también llamado página web, y los demás archivos asociados en su composición como imágenes, videos, hojas de estilo, scripts, entre otros, que puede ser localizado a través de un URL y que, normalmente, forma parte de un sitio web (Ospina Torres M. H., 2014).

El archivado de la web es una práctica de la preservación, La intención del archivo web es preservar la forma original del contenido cosechado sin modificaciones. Para lograr este objetivo los instrumentos, normas, políticas y buenas prácticas deben estar en el lugar que va a garantizar la gestión de archivos web a través del tiempo (IIPC, 2016), la IIPC considera el archivo web como una práctica de preservación digital.

La intención de Archivo Web es conservar la forma original del contenido cosechado sin modificaciones. Para lograr este objetivo los instrumentos, normas, políticas y buenas prácticas deben estar en el lugar que asegure la gestión de Archivos Web (IIPC, Netpreserve, 2012). Los sitios replicados son mantenidos completos, es decir, acompañados de los archivos, imágenes, gráficas y aspecto visual, y son almacenados en servidores de preservación en un ambiente seguro (Masanès, 2006).

Figura 5 - Resume la definición de archivos web
Fuente: Elaboración Propia

El querer preservar la Web es una acción que se vienen haciendo desde hace ya más de dos décadas. Para estas iniciativas de archivado Web se han establecidos estándares tanto para el tamaño como para el alcance del sistema. Generalmente estos Archivos Web son creados para almacenar los archivos acerca de su país de creación, aunque en algunos casos contienen información de entidades de distintos orígenes. La creación y mantenimiento de este tipo de archivadores es costosa y compleja.

El primer Archivo Web del mundo lo creo Brewster Kale en Estados Unidos en el año de 1996 y recibió el nombre de Internet Archive (Archivo de Internet). Posterior a esto surgieron tres iniciativas en ese mismo año: Australia's Web Archive (Archivo Web de Australia), Tasmanian's Web Archive (Archivo Web de Tasmania) y Kulturarw de Suecia. Solo otras cinco iniciativas surgieron en los siguientes 6 años: New Zealand WA (Nueva Zelanda, 1999), WebArchiv (República Checa, 2000), Library of Congress (EEUU, 2000), University of Michigan (EEUU, 2000) y OASIS (Korea, 2001). El emprendimiento de estos sistemas para preservar la web no ha cesado, actualmente existen alrededor de ochenta y dos (82) archivos web en el mundo. En los últimos 5 años se han creado alrededor de 17 Archivos Web, de los cuales se puede nombrar Internet Memory Research (Francia, 2011), Deutsche Nationalbibliothek (Alemania, 2012), Australian Government Web Archive (Australia, 2013), Digital Resources (Eslovaquia, 2015)

2.2.1. PANDORA (Australia's Web Archive)

Pandora es un desarrollo realizado íntegramente por Cran Consulting, S.L. que, basándose en el motor de indexación Full Text Lucene (desarrollo de dominio público bajo licencia de software libre de Apache), permite la indexación de archivos digitales, para su inclusión en una base de datos documental, almacenados en repositorios ARC y WARC. Para realizar búsquedas en la base de datos documental, Pandora es accesible desde cualquier navegador HTML estándar. (Estaba Fernández-Trujillo. & Ciancia Biondo, 2015)

Figura 6 - Página Principal de Pandora
Fuente: http://pandora.nla.gov.au/, 2014.

Pandora posee un repositorio de documentos, permitiendo a sus usuarios la recuperación de los mismos a partir de avanzados criterios de búsqueda tales como tipo de documento, cabecera, la fecha o rango de fechas y/o términos contenidos en el texto completo de los documentos indexados.

Fue establecida inicialmente por la biblioteca nacional de Australia en el año 1996, es un producto informático orientado a satisfacer las necesidades de difusión de contenidos de bibliotecas, centros de documentación y otras entidades donde, cada vez más, la información digitalizada forma parte de sus fondos documentales a gestionar.

Desde el punto de vista de los usuarios que acceden a Pandora, Pandora no solo pretende satisfacer las necesidades de búsqueda de instituciones como bibliotecas o archivos, ni solo de investigadores especializados si no que, con una interfaz de acceso Web sencilla e intuitiva, pretende hacer llegar los fondos documentales a cualquier usuario de Internet interesado en los mismos. Pandora es, por tanto, un instrumento que permite la recopilación, conservación y difusión de fondos documentales digitalizados mediante una interfaz de accesos Web (Internet/Intranet) de acceso simultáneo y concurrente.

La biblioteca donde se almacenan los objetos digitales asociados con PANDORA es denominada DOSS, las páginas Web preservadas son almacenadas en sistemas de archivos de Unix en un formato de esta biblioteca, luego de ser archivados son enviados fuera del sitio para su custodia. El sistema de acceso al público también está construido utilizando Apache/WebObjects/Java y Oracle para proporcionar la localización de recursos, navegación y control de acceso. Los elementos de contenidos digitales se entregan como contenido estático a través de Apache. El servicio está alojado en servidores Sun Solaris.

2.2.2. PADICAT

El PADICAT (acrónimo de Patrimonio Digital de Cataluña) es el archivo Web de Cataluña el cual fue creado el 2005 por la Biblioteca de Cataluña, la institución pública responsable de capturar, conservar y difundir el patrimonio bibliográfico de Cataluña, y por extensión el patrimonio digital. El sistema se basa en la aplicación de una serie de programas informáticos que permiten la captura, el almacenamiento, la organización y el acceso permanente a las páginas Web publicadas en Internet. Posteriormente a la fase de análisis y test de software, los desarrolladores determinaron que utilizarían el programa informático Heritrix, el cual es el encargado de compilar las páginas Web tal y como las ve el usuario que navega por Internet y almacenarlas en archivos comprimidos en formato .ARC o WARC. El programa Heritrix se complementa con NutchWax, o bien la combinación de Hadoop y Wayback, que llevan a cabo unos procesos de indexación de la información compilada, permitiendo utilizar estos índices para localizar los recursos dentro de la colección mediante sus respectivas interfaces de consulta: Wera, que permite la búsqueda por palabras clave a través de los índices generados por NutchWax; y Wayback, que permite la consulta directa por URL en los índices generados por Hadoop y el mismo Wayback. (Padicat, 2015)

2.2.3. Portuguese Web Archive

El Portuguese Web Archive o conocido también por sus siglas PWA tiene como objetivo principal la conservación y el acceso de los contenidos de la Web que ya no están disponibles en línea. El sistema Portuguese Web Archive recoge y almacena automáticamente la información publicada en la Web, en archivos contenedores ARC. Para realizar una búsqueda de alguna página Web preservada, PWA utiliza un servicio proporcionado por Google, que permite realizar búsquedas dentro de una colección de metadatos. (Estaba Fernández-Trujillo. & Ciancia Biondo, 2015)

Figura 7 - Página Principal Del Portuguese Web Archive
Fuente: www.archive.pt, 2014.

Durante el desarrollo del sistema PWA IR (recuperación de información) se enfrentaron a limitaciones en la velocidad de búsqueda, la calidad de los resultados, escalabilidad y facilidad de uso. Para hacer frente a ello, se modificó el método de acceso para mejorar la recuperación de información utilizando Nutchwax, Nutch y el código de Wayback que fueron adaptados para satisfacer éstas necesidades. Además, se agregaron varias optimizaciones, como simplificaciones en el camino de búsqueda de versiones de documentos y se resolvieron varios cuellos de botella.

El motor de búsqueda PWA es un servicio público en http://archive.pt basado en Heritrix y una plataforma de investigación para el Archivo Web. Como su predecesor Nutch, almacena las Páginas Web preservadas a través de clúster de Hadoop. Sus principales características incluyen la búsqueda rápida de texto completo, búsqueda de URL, buscar frases, búsqueda de fechas (fecha, formato, sitio Web), y la clasificación por relevancia. También el motor de búsqueda PWA es altamente escalable y su arquitectura es lo suficientemente flexible como para permitir el despliegue de configuraciones diferentes para responder a las diferentes necesidades. Actualmente, hay más de 1.875 millones contenidos archivados desde 1996. El motor de búsqueda de PWA (Portuguese Web Archive) se basa en Nutchwax 0.11.0 y 1.2.1 y Wayback también utilizan Hadoop para la automatización distribuida de los sitios web.

Las principales características del motor de búsqueda PWA son:

- Búsqueda de texto completo
- URL de búsqueda
- Búsqueda de frases
- Facetas de búsqueda (fecha, el formato, el sitio)

2.2.4. Internet Memories Foundation (IMF)

Se definen como una institución sin fines de lucro, fue fundada en 2004 en Ámsterdam bajo el nombre de Fundación Europea de Archivos, para apoyar y desarrollar los archivos digitales con acceso libre. En 2010, cambió su nombre por el de Internet Memories Foundation para expresar su interés en la preservación de contenidos Web como un nuevo medio para las generaciones actuales y futuras. Actualmente la Fundación alberga cientos de terabytes de sitios Web archivados (en formato WARC) con acceso libre, incluyendo su propia colección y colecciones de las instituciones asociadas. Esto incluye organizaciones como el Archivo Nacional del Reino Unido, la Biblioteca Nacional de Irlanda y el CERN. La IMF utiliza Hadoop, HDFS y HBase para almacenar e indexar los datos, y asocia este almacenamiento con un servidor Web que permite a los usuarios navegar por cada archivo y recuperar los documentos. (Estaba Fernández-Trujillo. & Ciancia Biondo, 2015).

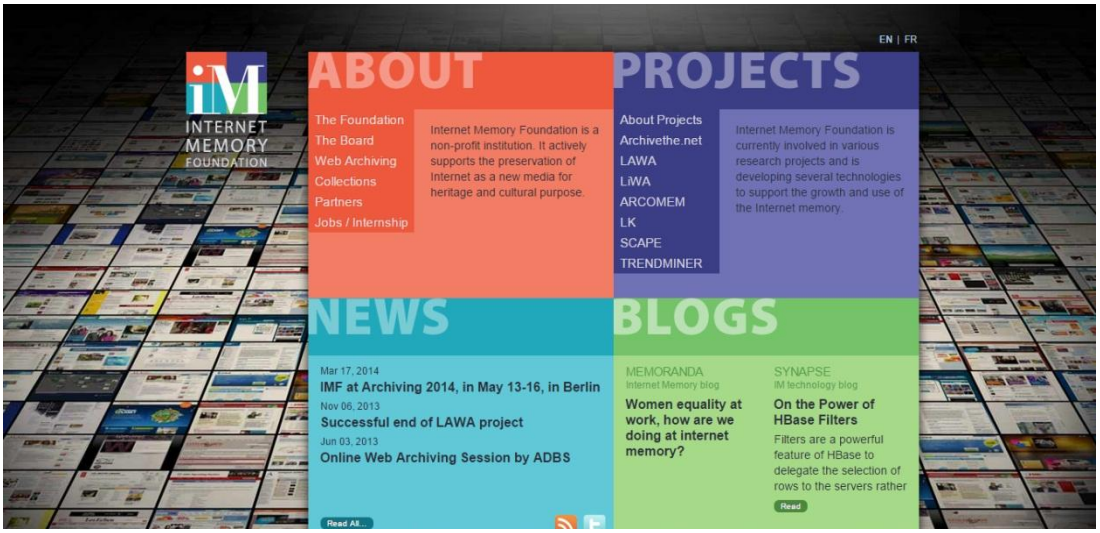

Figura 8 - Página Principal de Internet Memories Foundation
Fuente: www.internetmemory.org, 2014.

La IMF ha recogido una de las mayores colecciones en línea, de música clásica (más de 800 piezas, desde Mozart a Dvorak) y la información pública de varios Films del Gobierno británico, realizado en colaboración con el Instituto Holandés para el Sonido y Visión y el Archivo Nacional del Reino Unido. El rastreador Web utilizado por el proyecto es Heritrix versión 3.

Una forma de almacenar los contenidos dentro de un Archivo Web es el Formato WARC lo cual se describirá en la siguiente sección.

2.3. Formato WARC

Un archivo de formato WARC es una extensión del formato ARC, que contiene una cabecera la cual va a estar seguida de un bloque de contenido (UNESCO, 2003). La cabecera contiene información como la versión del WARC, el tipo de registro, el tamaño de los datos y otros que dependen del tipo del registro.

Este formato permite relacionar y almacenar múltiples recursos u objetos de datos que contienen cabeceras de textos y bloques de datos en un solo archivo de gran tamaño.

Cada archivo WARC contiene uno o más objetos digitales y una vez que el objeto es escrito el archivo se cierra y solo se abre para lecturas, con el formato WARC se busca añadir nuevas características, especialmente en:

- El almacenamiento de las cabeceras de petición del protocolo HTTP.
- El almacenamiento de metadata arbitraria.
- La asignación de un identificador para cada archivo de contenido.

El formato WARC está hecho para almacenar todo tipo de contenido Web, ya sea obtenido por el protocolo HTTP o por cualquier otro protocolo (**Figura 9**). Entre sus principales características se tienen:

- Guarda contenido junto a la información de los protocolos de cosecha
- Guarda metadatos enlazados a otros datos
- Permite compresión de datos y preservar integridad de los registros

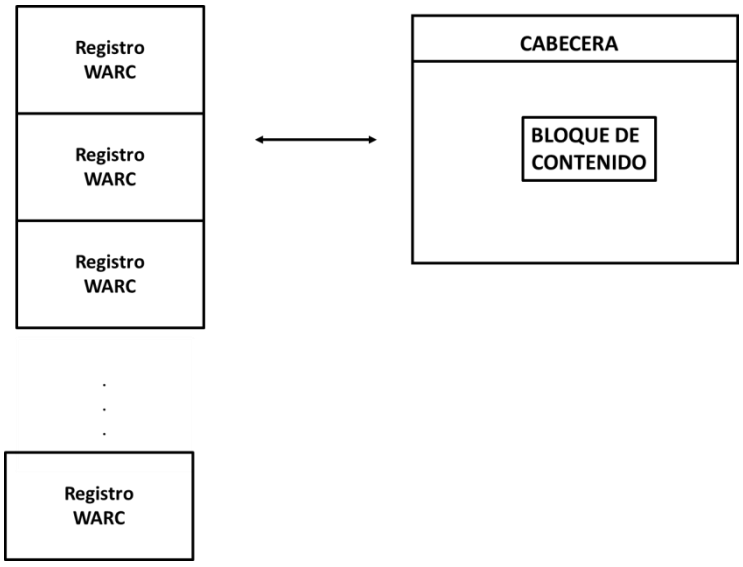

Figura 9- Formato Registro Warc
FUENTE: (ISO 28500, 2009)

El formato de archivo WARC está hecho lo suficientemente diferente al formato de archivo ARC, para que de esta manera las herramientas de software puedan detectarlo de formar

única y procesarlo correctamente. Dada la gran cantidad de datos de archivo existente en el formato ARC, es importante que el acceso y uso de éstos no sea interrumpido durante la transición al formato WARC.

Los WARC establecen un método para la combinación de múltiples recursos digitales en un archivo de archivos añadido, junto con información relacionada. Los recursos que son almacenados son anticuados, identificados por los URIs y precedidos de cabeceras de texto simple. Por convención, los archivos con este formato se nombran con extensión ".warc" y tienen la extensión MIME definida como aplicación/warc.

Existen actualmente ocho (8) tipos de registro WARC actualmente definidos, 'warcinfo', 'response', 'resourse', 'request', 'metadata', 'revisit', 'conversion', y 'continuation'. El propósito y uso de cada tipo se describen a continuación.

2.3.1. Tipos de Registros WARC

Un registro WARC se puede clasificar en 8 tipos de registros:

- Warcinfo: describe los registros que le siguen hasta el final del archivo o hasta el siguiente registro del tipo warcinfo. Típicamente aparece al inicio de cada archivo WARC contiene información acerca de la cosecha que generó los siguientes registros. Para un Archivo Web, comúnmente contiene la descripción de un rastreo web, es decir, puede contener información como la duración, el propósito. El formato de la descripción puede contener datos como el tamaño máximo del archivo o la tasa de rastreo.
- Resource: contiene la respuesta completa de un protocolo, tal como una respuesta HTTP, incluyendo cabeceras y contenido-cuerpo, de una recuperación de Internet.
- Request: contiene una esquematización completa y específica de la petición al servidor incluyendo la información del protocolo de red.
- Metadata: registro creado para describir, explicar o acompañar a otro registro. Un registro "metadata" casi siempre se referirá a un registro de otro tipo, teniendo este otro registro el contenido original, o transformado, que había sido recolectado.
- Revisit: describe la acción de volver a visitar el contenido previamente archivado e inclute el cuerpo de contenido abreviado que tiene que estar enlazado a algún registro previo. Un registro "revisit" debería ser manejado únicamente cuando, al descifrar un registro, se requiere consultar un registro previo; otros tipos de registros deberían ser preferidos si el registro actual es descifrable por sí solo. No se requiere que algún "revisit" de un URI previamente visto utilice un "revisit", solo aquellos que se refieran a otros registros.
- Conversion: contiene una versión alternativa del contenido de otro registro que se creó como resultado del proceso de archivado. Cada transformación de una versión alternativa debe ser independiente de su versión original.
- Continuation: los bloques de registros de este tipo deben estar relacionados a su correspondiente registro anterior de otros archivos WARC para crear el registro original de tamaño completo. Es usado cuando un registro hace que el WARC exceda

el tamaño límite establecido y debe ser separado en segmentos. Un registro de continuación va a necesitar de los parámetros "Segment-Origin-ID" (ID-Segmento-Origen) y "Segment-Number" (Número-Segmento), y normalmente incluye el parámetro "Related-Record-ID" (IDRegistro-Relacionado).

Para poder realizar la actividad de acceso o recuperación de los contenidos de un Archivo Web es necesario el uso de índices, motores de búsquedas, y el conocimiento de diversas técnicas que se explican a continuación.

2.4. Indexación

El proceso de localizar y recuperar rápidamente cada contenido de un archivo se le conoce como indexación, el cual nos va a permitir el uso de diversos métodos que mediante la utilización de una palabra (clave) asociada a un identificador de un archivo especifico incluirlo a un índice para posteriormente conocer la posición exacta de cada archivo y posibilitar el análisis de frecuencias el cual es accedido.

La mayoría de las herramientas de acceso a la Web están basadas en indexación automática, que no es más que la indexación que se realiza a través de procedimientos algorítmicos. Existen varios tipos de indexación automática y cuando la Web fue ganando popularidad se utilizaron tres tipos diferentes: el primero, fue un simple estilo de índice parte-de-atrás-del-libro, como en muchos libros escolares donde las palabras claves están vinculadas a una página o un capítulo. Cabe destacar que para la Web las palabras claves están vinculadas a una URL. El segundo tipo, sería indexando las palabras claves en conjunto con árboles objetos de los sitios revisados y el tercero, se realizaría con los motores de búsqueda más la indexación.

La indexación en el Archivo Web desarrollado, está basada en palabras clave con la ayuda de una plataforma de búsqueda, siendo la URL del sitio cosechado la clave de indexación. Al momento de la consulta, se coloca la URL deseada y se obtendrá como resultado todas las versiones correspondientes a la misma (Rivero & García, 2013).

La utilización de indexación puede mejorar el rendimiento de las consultas, ya que los datos necesarios para satisfacer las necesidades de la consulta existen en el propio índice y se reduce al máximo el tamaño de los archivos debido a que va se necesitar las páginas de índice y no las páginas de datos de la tabla o el índice agrupado para recuperar los datos solicitados; por tanto, se reduce también la E/S global en el disco.

Cuando el número de archivos para buscar es potencialmente de gran tamaño o la cantidad de consultas de búsqueda por realizar es considerable, el problema de búsqueda a menudo se divide en dos tareas: la indexación y la búsqueda. La etapa de indexación analizará el contenido de todos los archivos y creará una lista de los términos de búsqueda, a menudo llamado un índice, pero más correctamente llamado una concordancia. En la etapa de búsqueda, al realizar una consulta específica, sólo el índice se hace referencia en lugar del contenido de los archivos originales.

En el proceso de indexación para agilizar la búsqueda de grandes cantidades de datos es necesario hacer el uso de índices, ya que el mismo mejora la velocidad de las operaciones, de modo que el acceso a la información sea más rápido (Cardoso M, 2016).

Tomando como base la sección anterior, donde se expresa que el proceso de indexación trata de reducir al máximo el tamaño de los archivos o tablas de la base de datos para conseguir la mejor relación entre tiempo de ejecución de las consultas y exhaustividad del fichero inverso, a esta gestión se le denomina técnicas de Indexación, la cual se explicaran a continuación.

2.4.1. StopWords

Las palabras vacías son palabras que no contienen un significado importante para ser utilizado en las consultas de búsqueda. Por lo general, estas palabras se filtran de las consultas de búsqueda, ya que regresa una extensa cantidad de información innecesaria. Una mejor definición se proporciona a continuación:

Es una técnica de indexación que genera una lista de palabras de uso frecuente que no se tendrán en consideración, se omiten tanto en el momento del proceso de indexación como en el proceso de búsqueda (L.C., 2013).

Las palabras que no aparecen en el índice en una base de datos es porque son insignificantes (es decir, los artículos, preposiciones). Estas palabras son excluidas de las búsquedas para agilizar el proceso de indexar y analizar las páginas web. Algunos ejemplos de palabras vacías son: "un", "y", "pero", "cómo", "o" y "qué". Los buscadores en internet no pueden impedir el uso de estas palabras, por lo tanto las ignoran.

Por ejemplo, si buscamos "¿Qué es una placa base?", el motor de búsqueda ignorara las palabras "qué", "es" y "una" y sólo buscaría el término "placa base". Las palabras vacías varían de un sistema a otro. Además, algunos sistemas simplemente ignorará dejar de palabras donde el uso de palabras funcionales en otros sistemas se traducirá en la recuperación de cero aciertos. "

2.4.2. Stemming

Stemming es un método para reducir una palabra a su raíz o (en inglés) a un *stem*. Hay algunos algoritmos de stemming que ayudan en sistemas de recuperación de información. Stemming aumenta el *recall* que es una medida sobre el número de documentos que se pueden encontrar con una consulta. Por ejemplo una consulta sobre "bibliotecas" también encuentra documentos en los que solo aparezca "bibliotecario" porque el stem de las dos palabras es el mismo ("bibliotec").

Término	Stem		Término	Stem
che	che		consign	consign
checa	chec		consigned	consign
checar	chec		consigning	consign
checo	chec		consignment	consign

Figura 10 - Ejemplo de Stemming
Fuente: (Blázquez Ochando, webcache, 2012)

Analizando los ejemplos de la **Figura 10**, tanto en inglés como en español y en cualquier idioma, un término puede ser reducido a su común denominador, permitiendo la recuperación de todos los documentos cuyas palabras tengan la misma raíz común, por ejemplo (catálogo, catálogos, catalogación, catalogador, catalogar, catalogando, catalogado, catalogándonos).

2.5. Motores De Búsqueda

En la actualidad para agilizar el tiempo de respuesta en el proceso de búsqueda se hace uso de motores de búsqueda los cuales suelen utilizar palabras claves y metadatos, que provienen de las mismas páginas para de esta manera obtener un vocabulario y mejorar la búsqueda en el sitio. Los motores de búsqueda han recorrido un largo camino desde sus primeros prototipos. A partir de las mejoras en los rastreadores web y la categorización y la indexación de la web, a la introducción de nuevos.

El desarrollo de motores de búsqueda ha llevado a la investigación de múltiples tecnologías de búsqueda, las cuales se desarrollaron a partir diferentes motores de búsqueda. Alta Vista fue el primer motor de búsqueda para procesar las consultas en lenguaje natural; Lycos comenzó fuerte con un sistema de categorización de relevancia señales, palabras clave con prefijos y la proximidad palabra encontrada; y Ask Jeeves introdujo el uso de editores humanos para que coincida con las consultas de búsqueda de los usuarios reales.

Un motor de búsqueda es un software o una secuencia de comandos disponibles a través del Internet que busca en documentos o en bases de datos las palabras clave y devuelve los resultados de todos los archivos que contengan esas palabras clave. Hoy en día, existen miles de motores de búsqueda diferentes disponibles en Internet, cada uno con sus propias habilidades y características. El primer motor de búsqueda que se haya desarrollado se considera Archie, que fue utilizado para buscar FTP archivos y el primer motor de búsqueda de texto se considera Verónica. Hoy en día, el motor de búsqueda más popular y bien conocido es Google.

2.5.1. Tipos de Motores de Búsqueda

Actualmente el prototipo de Archivo Web de Venezuela cuenta con un buscador, el cual permite realizar la búsqueda por medio de la URL del sitio, a este tipo de buscadores se les conoce como buscadores de directorios.

2.5.1.1. Directorios

Los algoritmos para este tipo de motor de búsqueda son más sencillos, y los sitios son presentando como enlaces, los cuales representan los sitios registrados. Estos tipos de buscadores no recorren los sitios web ni almacenan sus contenidos, lo que hacen es registrar algunos de los datos de la página web, como el título y la descripción que se introduzcan en el momento de registrar el sitio en el directorio.

Los buscadores de directorio buscan información sobre contenidos de la página, los resultados serán presentados haciendo referencia a los contenidos y temática del sitio.

Para el nuevo módulo de indexación y búsqueda de archivo Web de Venezuela, se quiere implementar un motor de búsqueda jerárquico.

2.5.1.2. Buscadores Jerárquicos

Estos tipos de motores de búsqueda, recorren las páginas coleccionando información sobre los contenidos de las páginas. Cuando se inicia una búsqueda de información concreta en los buscadores, ellos consultan su base de datos y presentan resultados clasificados por su distinción para esa búsqueda concreta. Los buscadores pueden almacenar en sus bases de datos desde la página de entrada de cada web, hasta todas las páginas que residan en el servidor, una vez que las que el buscador la haya reconocido e indexado.

Eventualmente los motores de búsqueda revisan los sitios web, para actualizar los contenidos de su base de datos, por tanto puede que los resultados de la búsqueda estén desactualizados en algunos casos.

Debido a que los motores de búsqueda contienen millones y a veces miles de millones de páginas, muchos motores de búsqueda no sólo terminan de buscar las páginas, sino también muestran los resultados en función de su importancia. Esta importancia se determina comúnmente mediante el uso de diversos algoritmos.

2.5.2. ¿Cómo funciona un motor de búsqueda?

Como se puede observar en la **Figura 11** que la fuente de todos los datos del motor de búsqueda es un rastreador, que visita automáticamente las páginas y los índices de sus contenidos.

Una vez que una página ha sido rastreada, los datos contenidos en la página se procesan. A menudo, esto puede implicar los siguientes pasos.

- Anotar palabras vacías.
- Anotar las palabras que quedan en la página y la frecuencia con las que se producen.
- Enlaces a otras páginas de registro.
- Registro de la información sobre imágenes u otros medios embebidos.
- Los datos recogidos anteriormente se utilizan para clasificar la página y es el método principal de un motor de búsqueda utiliza para determinar si una página se debe mostrar y en qué orden.
- Por último, una vez que se procesa los datos se divide en uno o más archivos, de trasladó a diferentes ordenadores o carga en memoria en el que se puede acceder cuando se realiza una búsqueda.

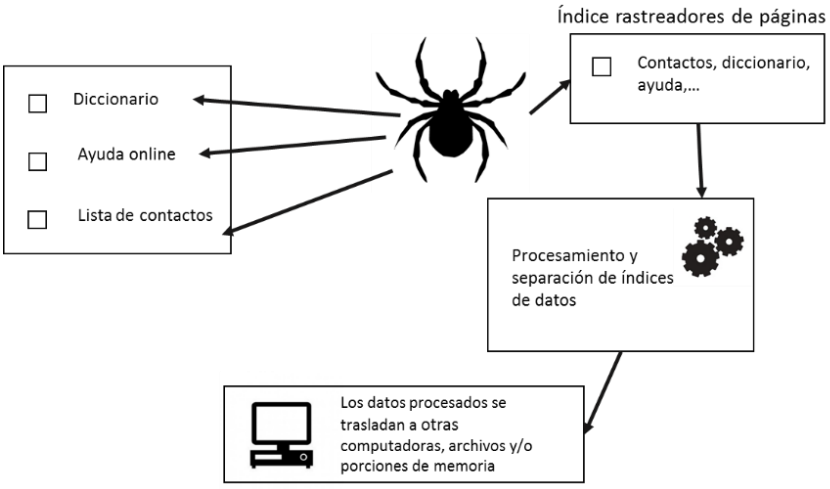

Figura 11 - Motor de Búsqueda
Fuentes:

Los motores de búsqueda pueden realizar distintos tipos de búsquedas, ya sea por fecha, por un campo en específico, por temas, en la siguiente tabla se muestra y explican algunos tipos de búsquedas.

Tabla 2 - Tipos de búsqueda

Restringido Campo de Búsqueda.	Permite a los usuarios realizar su búsqueda sobre un determinado campo dentro de un registro almacenado de datos, por ejemplo "Título" o "Autor".
Consultas Booleanas	Se hace uso de operadores booleanos para aumentar drásticamente la precisión de una búsqueda. El operador lógico dice, en efecto que se debería de traer.
Búsqueda de Concordancia	Produce una lista alfabética de todas las palabras principales que se producen en un texto con su contexto inmediato.
Búsqueda de Proximidad	Incluye solo los documentos que contienen dos o más palabras separadas por un número determinado de palabras.
Expresión Regular	Emplea una potente consulta compleja sintaxis que se puede utilizar para especificar las condiciones de recuperación de manera precisa.

Búsqueda Facetada	Consiste en encontrar elementos o contenidos restringiendo el conjunto global de resultados a través de múltiples criterios o facetas, se realizará la búsqueda por cualquier metadato del grafo semántico de una entidad. Estos criterios o facetas pueden aplicarse simultánea o sucesivamente.

En el nuevo módulo de indexación y búsqueda se implementaran la búsqueda por concordancia.

Para poder desarrollar el nuevo módulo de indexación y búsqueda para el prototipo de archivo Web de Venezuela, se harán uso de las siguientes tecnologías existentes:

2.6. Solr

Es una plataforma o motor de búsqueda de código abierto basado en el proyecto Apache Lucene, es una librería que facilita las búsquedas de texto con alto rendimiento haciendo uso de índices invertidos, obteniendo de este modo una mayor velocidad y flexibilidad en la búsqueda de cadenas de texto y menor dependencia del tamaño del índice. Algunas de las características principales de Solr son las siguientes:

- Búsquedas de texto completo
- Provee búsquedas dinámicas y distribuidas
- La integración de bases de datos y replicación de índices.

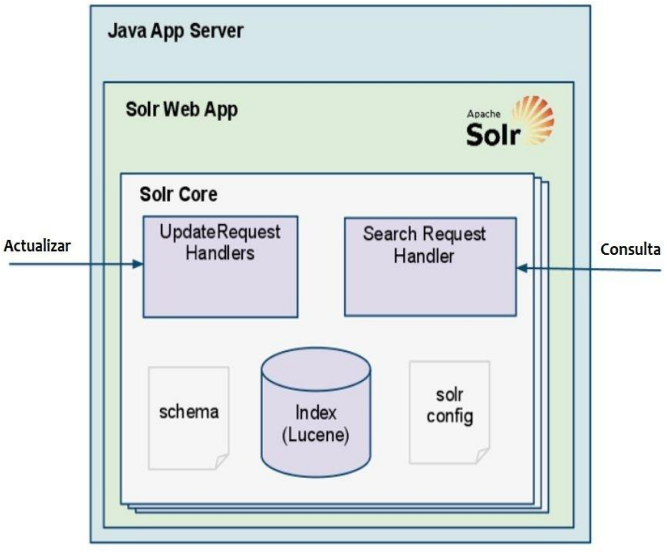

Figura 12 - Arquitectura Solr
Fuente: (Hoydahl)

Otra característica fundamental de Solr es que es altamente escalable y actualmente proporciona los mecanismos de búsqueda y funciones de navegación de muchos de los sitios más grandes del mundo de Internet como Instagram, Netflix, eBay, Internet Archive (Apache Solr, 2011.).

Solr está desarrollado en Java, otorgándole este lenguaje extender y modificar a Solr por medio de interfaces. Sin embargo, Solr es una tecnología que se comunica con distintos componentes por medio de peticiones HTTP, XML y JSON.

2.6.1. Índices en Solr

El índice creado por Solr es conocido como un índice invertido. Un índice invertido contiene estadísticas e información sobre los términos de un documento. Esto hace una búsqueda basada en términos muy eficiente. El índice creado por Solr se puede utilizar para enumerar los documentos que contienen el término buscado. Un ejemplo de un índice invertido, lo podemos obtener cuando usamos el índice en la parte posterior de cualquier libro, en este podemos ver términos significativos asociados con las páginas en las que se encuentran dentro del libro. Del mismo modo, en el caso de un índice invertido, los términos sirven para señalar o hacer referencia a los documentos en los que ocurren.

A continuación estudiaremos los índices en Sorl en profundidad. Un índice de Solr consta de documentos, campos y términos. Un documento está formado por strings (cadenas) o frases conocidas como términos, estos términos se refieren al contexto pueden ser agrupados juntos en un campo.

Por ejemplo, consideremos un producto en cualquier sitio de comercio electrónico, la información del producto se puede dividir en varios campos como el nombre del producto, descripción del producto, categoría de producto y precio del producto. Los campos pueden ser almacenados o indexados o ambos inclusive. Un campo almacenado contiene el texto original, sin analizar relacionado con dicho campo. El texto en los campos indexados se puede dividir en términos. El proceso de ruptura del texto en términos se le conoce como tokenization (tokenización). Los términos creados después de tokenization se llaman tokens (trozos de textos), que luego se utilizan para crear el índice invertido. El proceso de tokenization emplea una lista de filtros de tokens que se ocupan de diversos aspectos del proceso de tokenization.

Veamos un ejemplo **Figura 13** del proceso de indexación en Sorl con dos documentos que tienen solamente un solo campo.

Id_Documento	Texto
1	El señor de los anillos la comunidad del anillo
2	El señor de los anillos las dos torres

Figura 13 – Ejemplo Documento Sorl

Supongamos que decimos que la tokenization (ruptura de los términos) se va a aplicar en los espacios en blanco. Los espacios en blanco se definen como uno o más espacios o tabuladores. Los tokens son formados después de la tokenization de los documentos. Los tokens obtenidos de los documentos anteriores quedarían como se puede visualizar en la **Figura 14**.

Id_Documento	Texto								
1	El	señor	de	los	anillos	la	comunidad	del	anillo
2	El	señor	de	los	anillos	las	dos	torres	

Figura 14 - Ejemplo Tokenization Sorl.

El índice invertido formado de esta manera contendrá los siguientes términos y asociaciones que se puede observar en la **Figura 15**.

Token	Id_Documento
El	1,2
Señor	1,2
De	1,2
Los	1,2
Anillos	1,2
la	1
comunidad	1
del	1
anillo	1
las	2
dos	2
torres	2

Figura 15 – Ejemplo Índice Invertido Sorl

Posteriormente pasa por el proceso de stop Words, quedado de la siguiente manera ver **Tabla 3**.

Tabla 3 – Ejemplo Índice Invertido Solr con StopWors

Stop Words	Id_Documento
Señor	1,2
Anillos	1,2
Comunidad	1
Anillo	1
Dos	2
Torres	2

En el índice, podemos ver que el token de **Señor** aparece en ambos documentos. Si buscamos **Señor** en el índice que hemos creado, el resultado contendrá los documentos 1 y

2. Por otro lado, el token **Comunidad** tiene 1 único documento asociado a él en el índice. Una búsqueda de **Comunidad** regresará único documento 1.

2.6.2. Indexación en Sorl

La indexación de datos es una de las cosas más importantes en todas las implementaciones de Lucene y Solr. Cuando los datos no están indexado correctamente los resultados de las búsquedas serán pobres. Cuando los resultados de la búsqueda son pobres, es casi seguro que los usuarios no estarán satisfecho con la aplicación que utiliza. Es por eso que necesitamos que nuestros datos estén preparados y catalogados, de la mejor forma posible.

Por otra parte, la preparación de los datos no es una tarea fácil. Hoy en día tenemos más y más datos. Necesitamos índice de con múltiples formatos de datos que proviene de múltiples fuentes.

Para llevar a cabo la indexación es necesario colocar dentro del archivo que define las estructuras de datos a indexar llamado schema.xml el campo que se desea indexar. La definición de este se realiza siguiendo la siguiente sintaxis:

<field name="nombre de campo" type="tipo de dato" />

Los tipos de datos están definidos por las clases de java, sin embargo Solr da la libertad de crear un nuevo tipo de dato en caso que se necesite (Rivero & García, 2013). Además de los parámetros ya visto, la plataforma acepta parámetros adicionales para permitir la flexibilidad de la indexación:

- **default:** Valor a usar si no se recibe ninguno
- **required:** Define si un campo es obligatorio.
- **indexed:** Determina si un campo es buscable u ordenable.
- **stored:** Determina si un campo se puede recuperar en una consulta.
- **multiValued:** El campo contiene más de un valor.

En la **Figura 16** se puede observar la arquitectura de Solr. Como primer nivel tenemos las librerías, estas son el pilar de Solr ya que permiten realizar las búsquedas que como se ha mencionado anteriormente lo hacen por medio de la librería de Lucene. En el nivel superior Solr se divide en módulos, a continuación se describen los módulos más importantes:

- **Manejador de Peticiones o Request Handlers:** Este módulo se encargan de manejar las peticiones a través del protocolo HTTP, el componente que recibe esta petición es el denominado componente select.
- **Componentes de Búsquedas o Search Components:** forman parte del manejador de Búsquedas, su función es encargarse de las consultas, las búsquedas en facetas, el resaltado de consultas y las estadísticas.
- **Manejador de Actualización o Update Handlers:** maneja las peticiones de indexación.

- **Escritores de Respuesta o Responde Writers:** Este módulo se encarga de dar la respuesta en el estándar que se desee o se solicite, se dispone de XML, JSON y binario.
- **Manejador de Peticiones de Extracción**: se encarga de la indexación de formatos enriquecidos, como los documentos PDF o documentos WORD, a través de la ayuda de un componente propio de Lucene llamado Apache Tika.

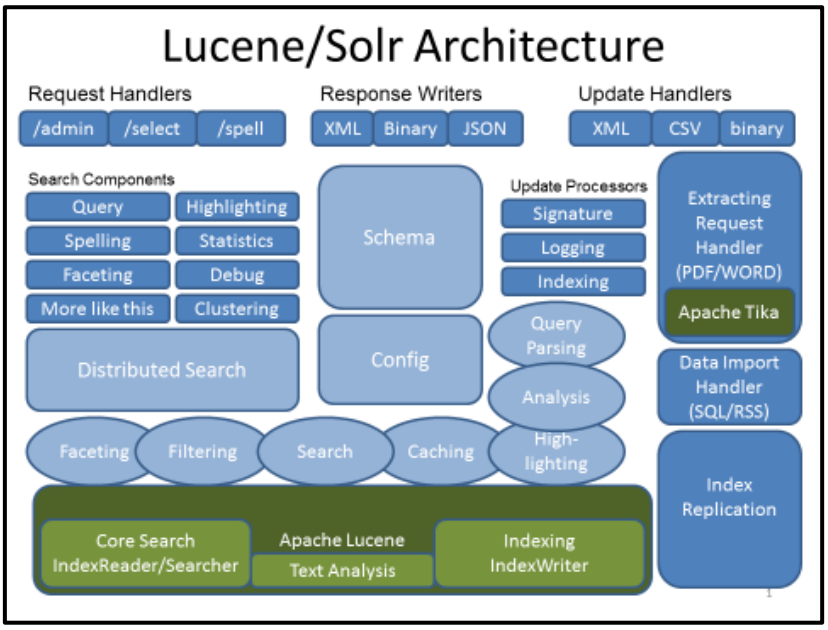

Figura 16 – Arquitectura Solr.
Fuentes: (Apache Solr, 2011.)

Por medio de estos módulos se puede hacer uso de todas las características de Solr, como la habilidad de esta plataforma para retornar la lista de resultados de la búsqueda de un query, su completitud en cuanto al esquema de datos puesto que maneja tipos numéricos, campos dinámicos y claves únicas entre otros. Solr permite realizar búsqueda por facetas basadas en valores únicos de campo, consultas explícitas, rangos de fechas y rangos numéricos. Da la posibilidad de dar al usuario sugerencias sobre la ortografía en las consultas

La plataforma de Búsqueda se divide en dos grandes partes como se puede observar el la **Figura 17**, el Índice y el Servidor:

- **Índice:** Sistema de archivos que almacenan la información. Contiene la configuración de Solr y la definición de la estructura de datos.
- **Servidor**: Proporciona el acceso a los índices y las características adicionales. Admite plugins para añadir funcionalidades.

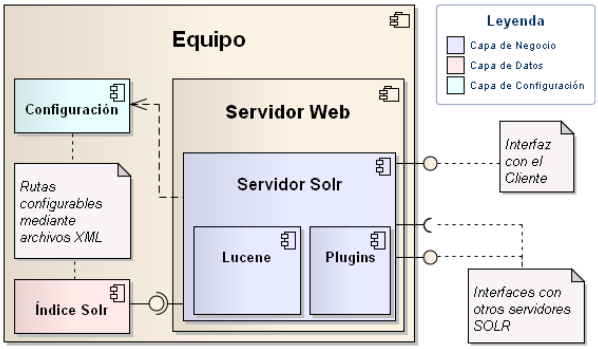

Figura 17 - Plataforma de Búsqueda
Fuentes: (Apache Solr, 2011.)

La arquitectura de Sorl permite búsquedas distribuidas: Uno de los servidores actúa como maestro, consultando al resto y componiendo la respuesta (**Figura 18**).

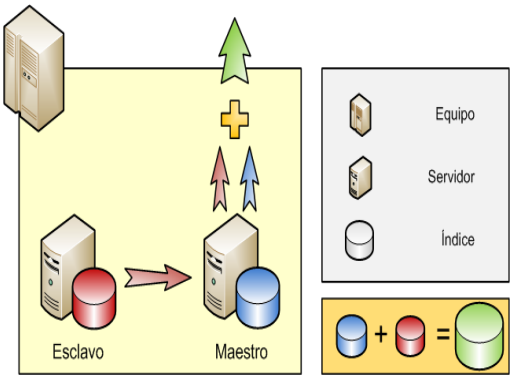

Figura 18 - Búsquedas Distribuidas
Fuentes: (Apache Solr, 2011.)

Ahora bien si se quiere trabajar con índices distribuidos Solr ofrece una nueva versión que proporciona alta escalabilidad, tolerancia a fallos, las capacidades de indexación y búsqueda distribuida, búsqueda en tiempo real, configuración centralizada del clúster y gestión. La cual se describirá en la siguiente sección.

2.7. SolrCloud

SolrCloud es el nombre de un conjunto de nuevas capacidades distribuidas en Solr la cual proporciona la solución de alta disponibilidad y con mutación por error para un índice que abarca más de varios servidores Solr. Se sigue con el modelo maestro-esclavo tradicional pero ahora se añade la implementación de un clúster de Solr fragmentados, se crea varios servidores maestros Solr, uno para cada fragmento y los esclavos para estos servidores maestros. **Figura 19**

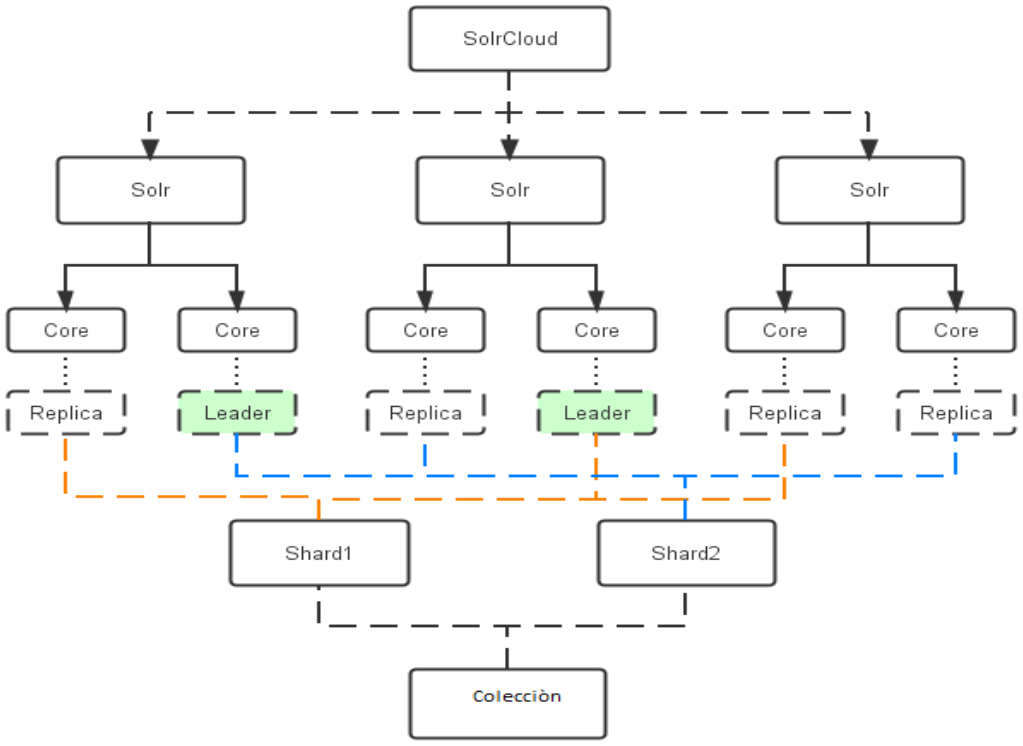

Figura 19 – Diagrama de la arquitectura SolrCloud
Fuente: http://www.tqcto.com/article/internet/3173.html.

Con SolrCloud se resuelve el problema de distribución de las anteriores versiones de Solr:

- Había que tener cuidado de que el algoritmo sharding de manera que los datos se distribuyeran a través de múltiples fragmentos.
- Además, se tenía que tener cuidado de los servidores que se caían, había que crear una configuración de conmutación por error para el mismo. El equilibrio de carga de consultas de búsqueda era manual.

2.7.1. Indexación distribuida

Cuando se tiene grandes cantidades de datos para ser indexados, se quiere acelerar el proceso de indexación, para lograr esto podemos trabajar en la indexación distribuida o también llamado índices distribuidos. Los índices distribuidos pueden hacerse mediante la creación de múltiples índices en diferentes máquinas y finalmente hacer una fusión en un único índice **Figura 20**. Aunque mejor sería crear los índices separados en diferentes máquinas cada una con una instancia Solr y utilizar Solr sharding para consultar los índices a través de múltiples fragmentos. Por ejemplo, un índice de 10 millones de productos se puede romper en trozos más pequeños en base a la identificación del producto y puede ser indexado en más de 10 máquinas, con cada indexación de un millón de productos. Durante

la búsqueda, podemos añadir estos 10 servidores Solr como fragmentos y distribuir nuestras consultas de búsqueda sobre estas máquinas.

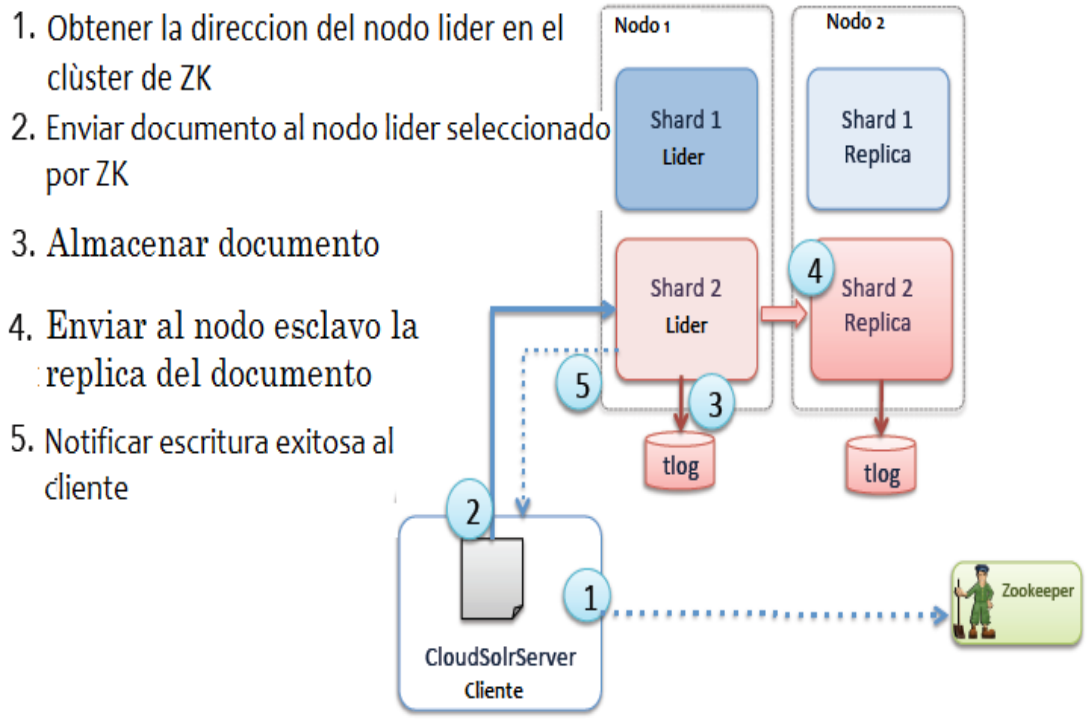

Figura 20 – Distribución de los Índices en SolrCloud
Fuente: (Potter, 2014)

SolrCloud maneja escalabilidad para grandes índices, ya que se constituye por un grupo de servidores Solr o núcleos que se pueden unir juntos como un solo servidor SolrCloud. SolrCloud se utiliza cuando hay una necesidad de capacidades de búsqueda altamente escalable, tolerante a fallos, indexación y distribuido. Con SolrCloud, un índice único puede extenderse a lo largo de múltiples núcleos Solr que pueden estar en diferentes servidores Solr. En la sección siguiente se estudiaran algunos conceptos de SolrCloud.

2.7.2. Terminología Usada en SolrCloud

Tabla 4 – Terminología Usada en SorlCloud

Cluster	Conjunto de Solr nodos gestionados como una sola unidad
Nodo	Una instancia JVM corriendo Solr
Partición	subconjunto de toda la colección de documentos, Simplemente un solo núcleo Solr

Fragmento	Una partición tiene que ser almacenado en múltiples nodos como se especifica por el factor de replicación. Todos estos nodos forman colectivamente un fragmento. Un nodo puede ser una parte de múltiples fragmentos
Líder	Cada grupo tiene un nodo identificado como su líder. Todas las escrituras de documentos pertenecientes a una partición enrutan a través del líder
Colección	Uno o más fragmentos / Réplicas están dispuestos en Colección
Factor de replicación	Número mínimo de copias de un documento que mantiene el grupo
Registro de transacciones	Un registro sólo de adición de operaciones de escritura mantenida por cada nodo

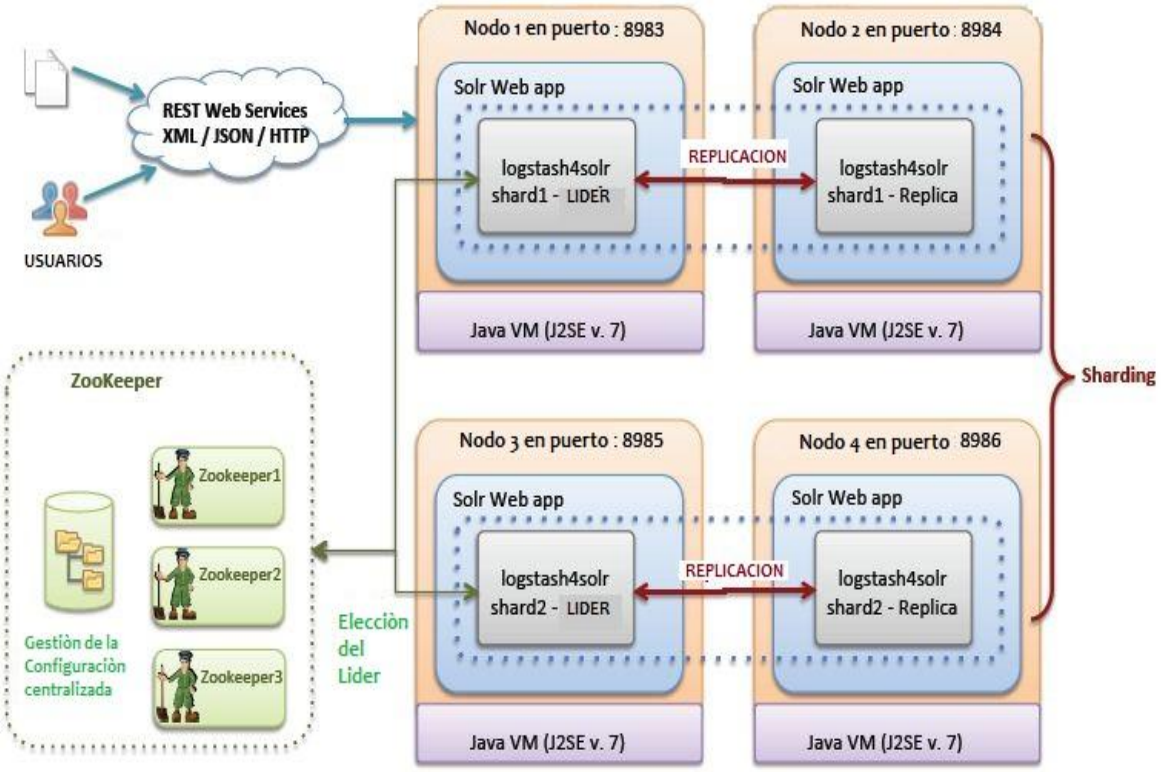

Figura 21 - Arquitectura SolrCloud
Fuente: (Potter, 2014)

2.7.3. Ventajas de SolrCloud

Las ventajas que ofrece SolrCloud principalmente se centra en:

- **Sharding.** Es el acto de la división de un solo índice Solr en varios equipos, también conocido como fragmentos o rodajas en la terminología Solr. Sharding es más a menudo necesario porque un índice ha crecido demasiado para estar contenido en un solo servidor. Un fragmento dado puede contener múltiples servidores físicos / virtuales, lo que significa que todas las máquinas / réplicas en ese fragmento contienen los mismos datos de índice y sirven consultas para esos datos. A través sharding, puede dividir el índice a través de múltiples máquinas y seguir creciendo sin caer en problemas, visualizar **Figura 22.**

Figura 22 – Shard en SorlCloud
Fuente: (Potter, 2014)

- **Distribución de Consultas y Balanceo de Carga.** Con SolrCloud, la búsqueda distribuida se maneja automáticamente por los nodos en la nube consultar cualquier nodo provocará que el nodo para enviar la consulta a un nodo en todos los demás fragmentos, devolver una respuesta sólo cuando se ha agregado los resultados de todos los fragmentos. Además, ZooKeeper y todas las réplicas son conscientes de los nodos que no respondieron, y por lo tanto no van a dirigir las consultas a los nodos que se consideran caídos. En el caso de que un nodo caído aún no ha sido detectado por ZK y se envía una consulta, el nodo de consulta reportará el nodo como caído para ZK, y vuelva a enviar la consulta a otro nodo. De esta manera, las consultas a un SolrCloud son muy duraderas y tendrá casi nunca a ser interrumpidas por un nodo derribado, SolrCloud puede manejar su propio equilibrio de la carga si se utiliza un cliente inteligente como SolrJ. SolrJ utilizará un sencillo equilibrador de carga round robin, distribuyendo las consultas de manera uniforme a todos los nodos en SolrCloud. Además, SolrJ es ZooKeeper conscientes y por lo tanto nunca se enviará una consulta a un nodo que se conoce como caído. **Figura 23**

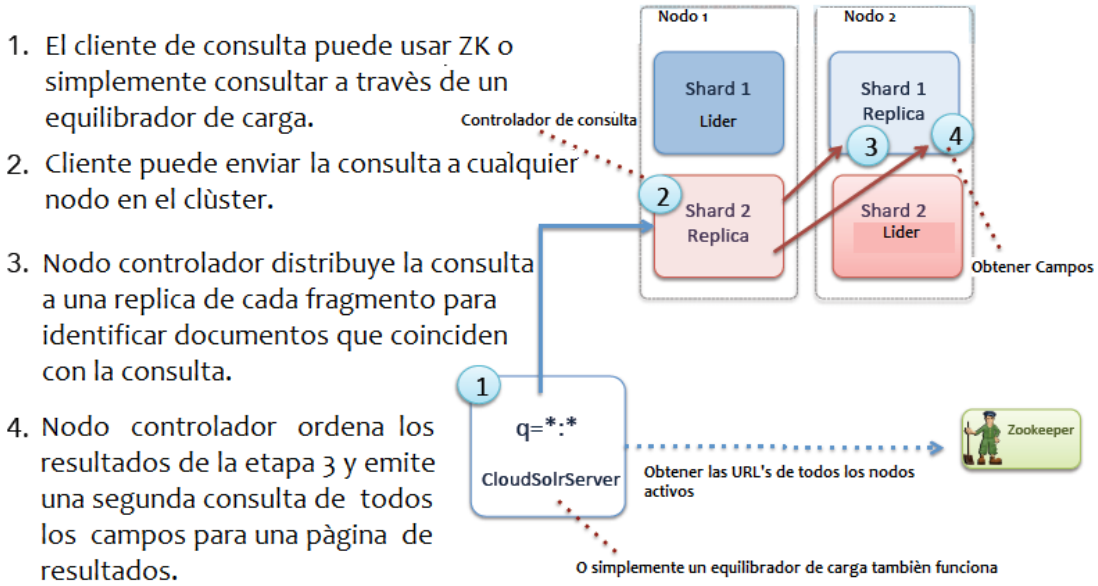

Figura 23 - Distribución de Consultas y Balanceo de Carga Solr
Fuente: (Sain Technology Solutions, 2015)

- **Alta Disponibilidad.** En SolrCloud, cuando ZooKeeper detecta un líder caído, se iniciará el proceso de elección de líder de forma instantánea, la selección de un nuevo líder para comenzar a distribuir los documentos de nuevo. Desde el registro de transacciones se asegura de que todos los nodos del fragmento están sincronizados, todas las actualizaciones son. En algunos casos, si en una réplica han perdido demasiados cambios, se llevará a cabo una replicación estándar, así como reproducir el registro de transacciones antes de servir consultas. Se manejara automáticamente los cambios de recorrido y la recuperación cuando un nodo del clúster deja de funcionar, es decir, nunca se perderá las actualizaciones y la ruta automáticamente alrededor de los nodos que no responden, visualizar **Figura 24**.

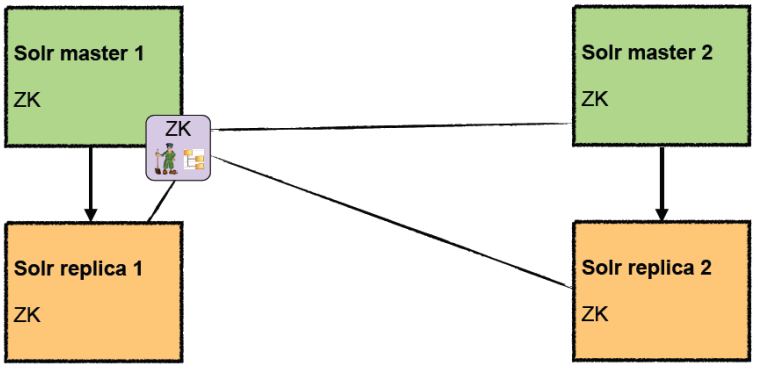

Figura 24 - Alta Disponibilidad SolrCloud
Fuente: (Hoydahl)

SolrCloud está enfocado para el almacenamiento y búsqueda distribuida, lo más común es utilizar SolrCloud para guardar/cargar los índices de los archivos procesado en un framework distribuido, para posteriormente permitir que un resultado procesado pueda buscar a través de SolrCloud. Algunas de los framework distribuidos que se puede integrar con SolrCloud es Hadoop (**Figura 25**), el cual se estudiara a continuación.

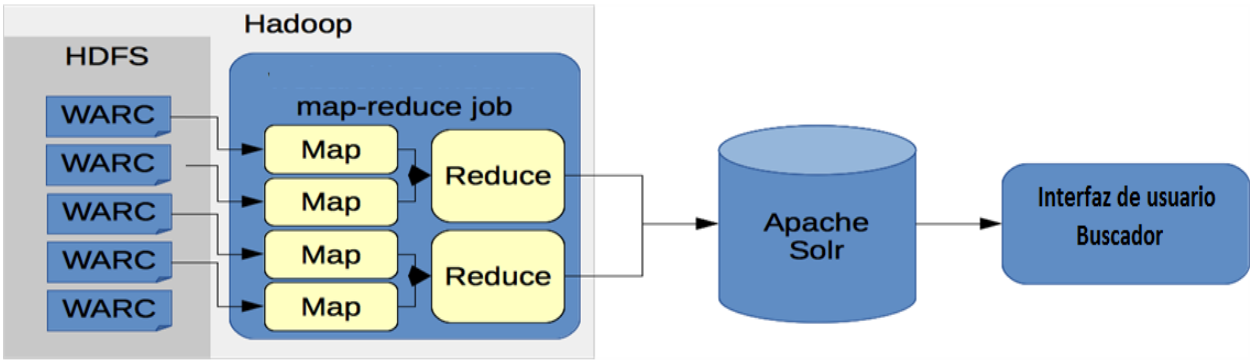

Figura 25 - Solr Integrado con Hadoop
Fuente: (Jackson, 2015)

2.8. Apache Hadoop

Es un proyecto de software para desarrolladores de código abierto que es confiable, escalable y distribuido. Sus librerías permiten el procesamiento distribuido de grandes conjuntos de datos a través de grupos de computadores que utilizan modelos de programación simples (The Apache Software Foundation, 2014). Está diseñado para pasar de servidores individuales a miles de máquinas cada una con un procesamiento y almacenamiento local. En vez de confiar en el hardware para ofrecer alta disponibilidad que está en constante crecimiento, está diseñado para detectar y controlar los errores en la capa de aplicación, que es más propensa a errores.

Las aplicaciones Hadoop pueden usar esta información para ejecutar trabajo en el nodo donde están los datos y en su defecto en el mismo rack/switch, reduciendo así el llamado tráfico de red troncal (Backbone Traffic).

Un clúster típico Hadoop incluye un nodo maestro, que consiste en jobtracker (rastreador de trabajo), tasktracker (rastreador de tareas), namenode (nodo de nombres) y datanode (nodo de datos), y múltiples nodos esclavo o compute node (nodo de cómputo) que consisten en un nodo de datos y un rastreador de tareas, visualizar **Figura 26** .

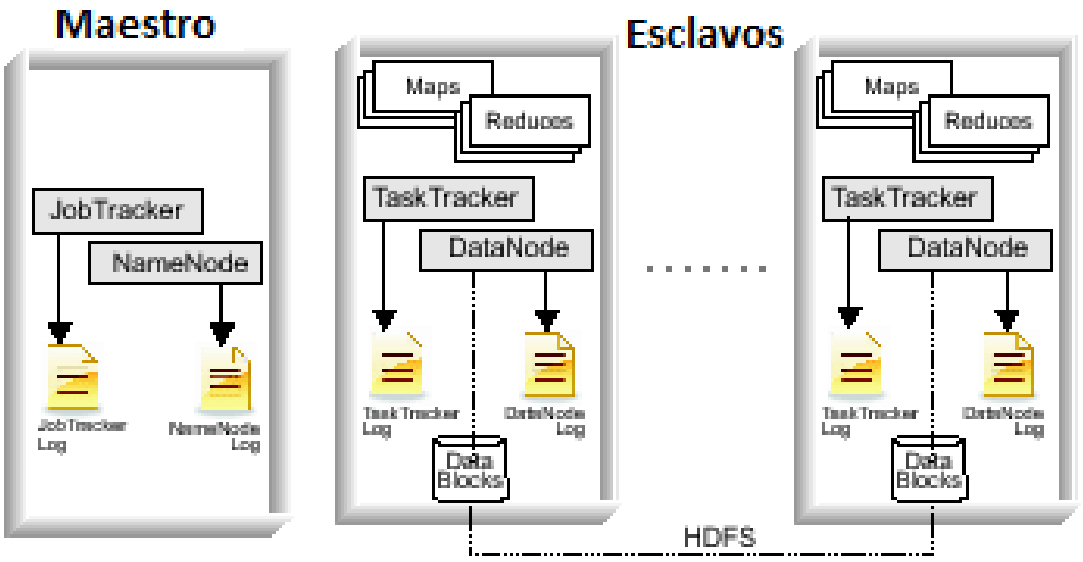

Figura 26 - Cluster Hadoop
Fuente: https://www.kopo.com/bd/architecture-of-hadoop/

2.8.1. Arquitectura principal de Hadoop

En su núcleo, Hadoop tiene dos capas principales (**Figura 29**):

- Capa de almacenamiento (Hadoop Distributed File System).
- Informática/Computación capa (MapReduce)

2.8.1.1. HDFS

HDFS es el sistema de almacenamiento, es un sistema de ficheros distribuido. Fue creado a partir del Google File System (GFS). HDFS se encuentra optimizado para grandes flujos y trabajar con ficheros grandes en sus lecturas y escrituras. Su diseño reduce la E/S en la red. La escalabilidad y disponibilidad son otras de sus claves, gracias a la replicación de los datos y tolerancia a los fallos (The Apache Software Foundation, 2014). Los elementos importantes del cluster:

- NameNode: Sólo hay uno en el cluster. Regula el acceso a los ficheros por parte de los clientes. Mantiene en memoria la metadata del sistema de ficheros y control de los bloques de fichero que tiene cada DataNode. Ver **Figura 27**
- DataNode: Son los responsables de leer y escribir las peticiones de los clientes. Los ficheros están formados por bloques, estos se encuentran replicados en diferentes nodos. Ver **Figura 27**

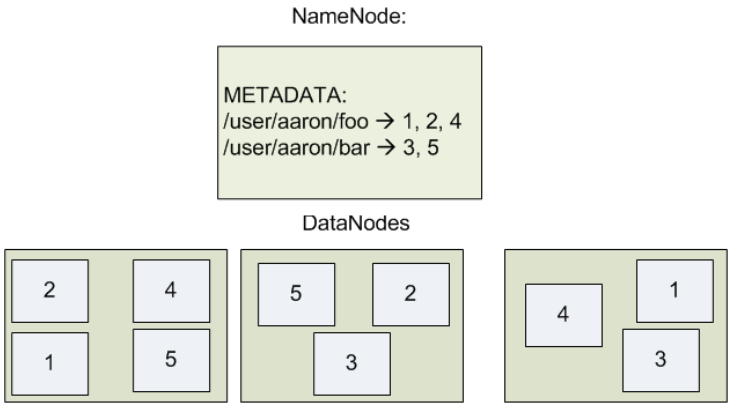

Figura 27 – Tipos de elementos en Hadoop
Fuente: (Sain Technology Solutions, 2015)

2.8.1.2. MapReduce

MapReduce es un proceso batch, creado para el proceso distribuido de los datos. Permite de una forma simple, paralelizar trabajo sobre los grandes volúmenes de datos, como combinar web logs con los datos relacionales de una base de datos OLTP, de esta forma ver como los usuarios interactúan con el website.

El modelo de MapReduce simplifica el procesamiento en paralelo, abstrayéndonos de la complejidad que hay en los sistemas distribuidos. Básicamente las funcionesMap transforman un conjunto de datos a un número de pares key/value. Cada uno de estos elementos se encontrará ordenado por su clave, y la función reduce es usada para combinar los valores (con la misma clave) en un mismo resultado (The Apache Software Foundation, 2014).

Un programa en MapReduce, se suele conocer como Job, la ejecución de un Job empieza cuando el cliente manda la configuración de Job al JobTracker, esta configuración especifica las funciones Map, Combine (shuttle) y Reduce, además de la entrada y salida de los datos.

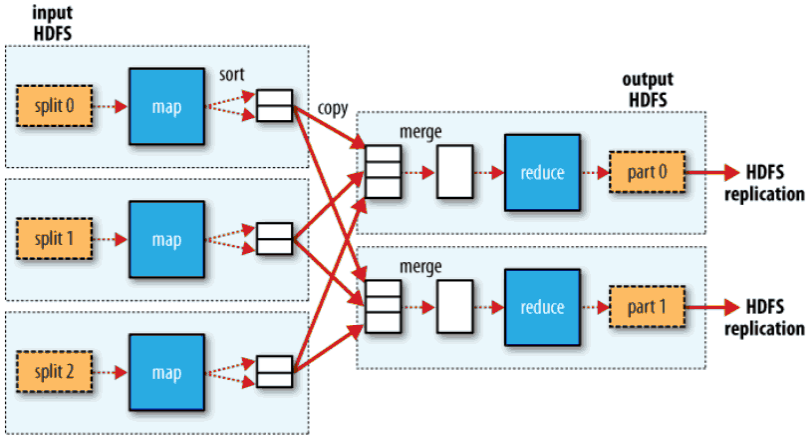

Figura 28 - MapReduce Hadoop
Fuente: http://www.cnblogs.com/forfuture1978/archive/2010/02/27/1674955.html

Aparte de los mencionados dos componentes básicos, Hadoop incluye también los dos módulos siguientes:

- Hadoop común: Estas son las bibliotecas de Java y las utilidades requeridas por otros módulos Hadoop.
- Hadoop HILO: Este es un marco para la planificación de tareas y administración de recursos de clúster (The Apache Software Foundation, 2014).

Figura 29 – Arquitectura Principal Hadoop
Fuente: http://hadoop.apache.org/

La arquitectura de Apache Hadoop consiste básicamente en el Hadoop Común (Hadoop Common) es decir, tiene acceso a los sistemas de archivos soportados por Hadoop (The Apache Software Foundation, 2014). El paquete de software Hadoop Common contiene los archivos .jar y los scripts necesarios para hacer correr Hadoop, también proporciona código fuente, documentación, y una sección de contribución que incluye proyectos de la Comunidad Hadoop.

2.8.2. El ecosistema de Hadoop

En Hadoop se posee un ecosistema muy diverso, que crece día tras día, por lo que es difícil saber de todos los proyectos que interactúan con Hadoop de alguna forma. A continuación sólo mostraremos los más comunes **Figura 30.**

- **Hive:** Infraestructura de Dataware que proporciona mecanismos para el almacenamiento, extracción, transformación y carga (ETL), y un lenguaje similar a SQL para realizar consultas y análisis.
- **HBase**: Es una base de datos NoSQL diseñada para almacenamiento en tiempo real, recuperación y búsqueda en tablas muy grandes (millones de columnas, billones de filas) que se ejecuta sobre HDFS.

- **HCatalog:** Es un servicio de gestión de tablas y almacenamiento para usar con Hadoop, que provee un esquema compartido y una abstracción de tablas para que los usuarios no deban saber cómo se almacenan sus datos.
- **Pig:** Es un conjunto de herramientas para el análisis programático de análisis de ficheros planos. Ofrece un lenguaje de programación, transformación de datos y procesamiento en paralelo.
- **Zookeeper**: Es una herramienta de gestión de aplicaciones distribuidas para la configuración, sincronización de eventos, agrupación de servicios utilizados para la gestión de los nodos en una red Hadoop. Esta herramienta es parte fundamental del ecosistema Haddop ya que es la encargada de balancear la carga entre los nodos, y se estudiara en la siguiente sección.

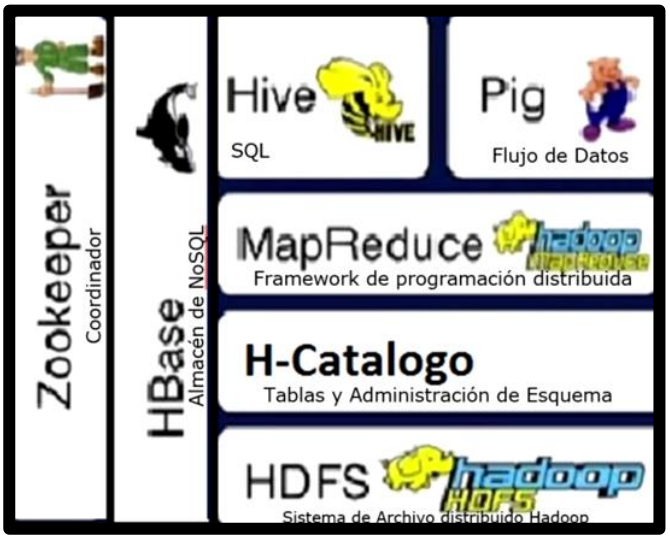

Figura 30 - Ecosistema Hadoop
Fuente: (The Apache Software Foundation, 2014)

2.9. Zookeeper

Apache ZooKeeper es un proyecto de software de la Apache Software Foundation, que provee un servicio de configuración centralizada y registro de nombres de código abierto para grandes sistemas distribuidos.

Un sistema como ZooKeeper puede hacer por nosotros tareas como la coordinación de los sistemas distribuidos. Una tarea de coordinación es una tarea que implica múltiples procesos. Esa tarea puede ser para los fines de la cooperación o para regular la contención de los datos. La cooperación significa que los procesos tienen que hacer algo juntos, y los procesos se movilizan para que otros procesos puedan progresar. Por ejemplo, en las arquitecturas típicas maestro-esclavo, el esclavo informa al maestro que esté disponible para realizar alguna tarea. El maestro como respuesta asigna tareas al esclavo. La contención es diferente: se refiere a situaciones en las que dos procesos no pueden avanzar al mismo tiempo, por lo que uno debe esperar por el otro. Usando el mismo ejemplo principal de trabajo, que realmente quiere tener un solo maestro, pero varios procesos

pueden tratar de convertirse en el maestro. Los procesos múltiples necesitan implementar la exclusión mutua. De hecho, podemos pensar que la tarea de adquirir el puesto de maestro como la de adquirir un bloqueo: el proceso que adquirió el bloqueo ejerce el papel de maestro.

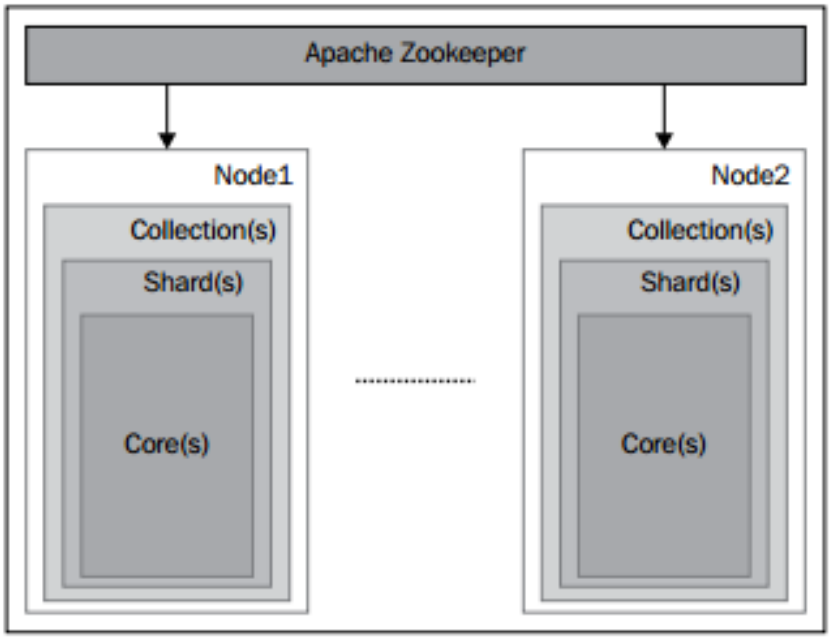

Figura 31 - ZooKeeper integrado con Solr
Fuente: (Karambelkar, 2013)

Algunos ejemplos en los que ZooKeeper ha sido útil para obtener una mejor idea de donde es aplicable:

Tabla 5 - Aplicaciones que usan ZooKeeper

Apache HBase	HBase es un almacén de datos que se utiliza normalmente junto con Hadoop. En HBase, ZooKeeper se utiliza para elegir un patrón de clúster, para realizar un seguimiento de servidores disponibles, y para mantener los metadatos clúster.
Apache Solr	Solr es una plataforma de búsqueda empresarial. En su forma distribuida, denominada SolrCloud, se utiliza ZooKeeper para almacenar metadatos sobre el cluster y coordinar los cambios a estos metadatos.
Facebook Messages	Se trata de una aplicación de Facebook que integra canales de comunicación: correo electrónico, SMS, chat de Facebook, y el vigente Facebook Bandeja de entrada. Utiliza ZooKeeper como un controlador para la aplicación de sharding y conmutación por error, y también para la detección de servicios.

La arquitectura de ZooKeeper soporta alta disponibilidad a través de servicios redundantes. Los clientes pueden así preguntar a otro maestro ZooKeeper si el primero falla al responder. Los nodos ZooKeeper guardan sus datos en un espacio de nombres jerárquico, como hace un sistema de archivos o una estructura de datos. Los clientes pueden leer y escribir desde y hacia los nodos y de esta forma tienen un servicio de configuración compartido. ZooKeeper es usado por varias compañías, incluyendo Rackspace y Yahoo!, así como sistemas de búsqueda empresarial open source como Solr (T. A. S. Foundation, 2010)

ZooKeeper tiene como objetivo proporcionar un núcleo de rendimiento simple y alto para la construcción de primitivas de coordinación más complejas en el cliente (P. Hunt, 2010). La interfaz de ZooKeeper permite un alto rendimiento de la implementación del servicio. ZooKeeper ofrece una garantía por cliente ya que las solicitudes que recibe se ejecutan de forma FIFO primero en entrar, primero en salir. Esto permite la implementación de un alto rendimiento ya que las solicitudes se procesan en servidores locales (P. Hunt, 2010).

2.9.1. Descripción General del Servicio

ZooKeeper ofrece a sus clientes la abstracción de un conjunto de nodos de datos llamados znodes, organizados de acuerdo a un espacio de nombres jerárquico. Los znodes en esta jerarquía son los datos que manipulan los clientes a través del Api de ZooKeeper. Los espacios de nombres jerárquicos se utilizan comúnmente en sistemas de archivos. Es una forma de organización de los datos, ya que los usuarios están acostumbrados a esta abstracción y permite una mejor organización de la aplicación de metadatos (P. Hunt, 2010).

Para referirse a un znode dado, usamos la notación estándar de UNIX para archivos de rutas del sistema. Por ejemplo, se utiliza /A /B /C para saber la ruta del sistema en el que se encuentra (P. Hunt, 2010) **Figura 32**.

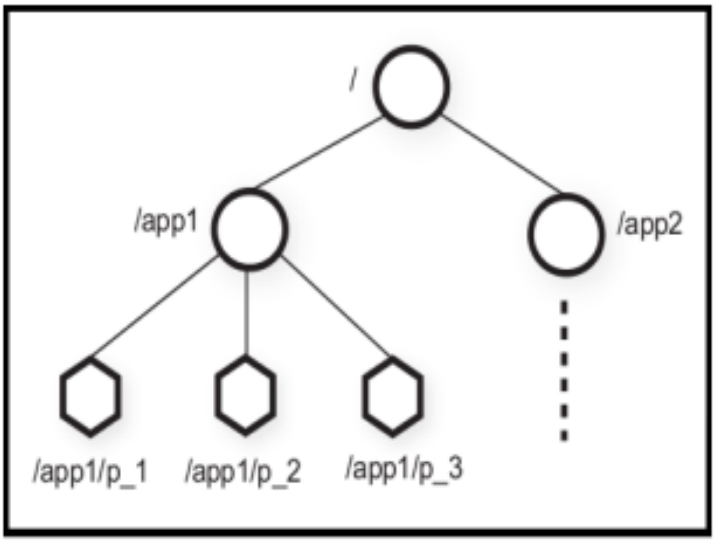

Figura 32- Espacio de Nombres Jerárquicos
Fuentes: (T. A. S. Foundation, 2010)

Hay dos tipos de nodos que se puede distinguir en la arquitectura de Zookeeper:

- **Nodo líder:** el nodo líder es el único nodo responsable de procesar las solicitudes de escritura. Todos los otros nodos denominados seguidores simplemente delegan las llamadas de clientes de escritura al nodo líder.
  No se marca cualquier nodo de líder durante la configuración de clúster de Apache ZooKeeper. En su lugar, es elegido internamente entre todos los nodos de clúster. Apache ZooKeeper utiliza el concepto de mayoría por el mismo nodo es decir, que obtiene mayor número de votos es elegido como líder.
  Esto sirve como la base de la recomendación que sugiere que tener un número impar de nodos en un clúster de conmutación por error a la mejor y la disponibilidad. Por ejemplo, si se crea el clúster de cuatro nodos y dos nodos de ir fuera de línea por alguna razón. Apache ZooKeeper estará abajo como la mitad de los nodos han ido fuera de línea, ya que no es posible ganar la mayoría en las elecciones nodo líder. Sin embargo, si creamos el grupo de cinco nodos, incluso si dos nodos se desconectan, Apache ZooKeeper todavía será funcional ya que aún tenemos mayoría de los nodos de servicio.
- **Nodos seguidores:** Todos los nodos que no sean líder son llamados nodos de seguidor. Un nodo de seguimiento es capaz de atender las solicitudes de leer por su cuenta. Para las solicitudes de escritura, se las trae a envía al nodo líder. Los seguidores también juegan un papel importante de elegir a un nuevo líder si el nodo líder existente se cae.

Se presenta una breve descripción de los componentes del nodo, como se muestra en el diagrama de la arquitectura de la siguiente **Figure 33.**

Hay que tener en cuenta que estos no son los únicos componentes del nodo.

- **Procesador de Peticiones:** Este componente sólo se activa en el nodo líder y es responsable de la tramitación de solicitudes de escritura procedentes de nodos cliente o seguidor. Una vez procesador de peticiones processes la solicitud de escritura, difunde los cambios al nodo seguidor para que puedan actualizar su estado en consecuencia.
- **Atómica Broadcast:** Este componente está presente en ambos nodos, tanto en el nodo líder y como en el nodo seguidor. Este componente es el responsable de la transmisión de los cambios a otros nodos.
- **Base de datos en memoria:** Es responsable de almacenar los datos en ZooKeeper. Cada nodo contiene su propia base de datos que les permite a las peticiones de lectura servidor poder responder. Además de esto, los datos también se escriben en el sistema que proporciona capacidad de recuperación en caso de cualquier problema con el clúster del archivo. En caso de solicitudes de escritura, la base de datos en memoria se actualiza sólo después de que se ha sido escrito con éxito en el sistema de archivos.

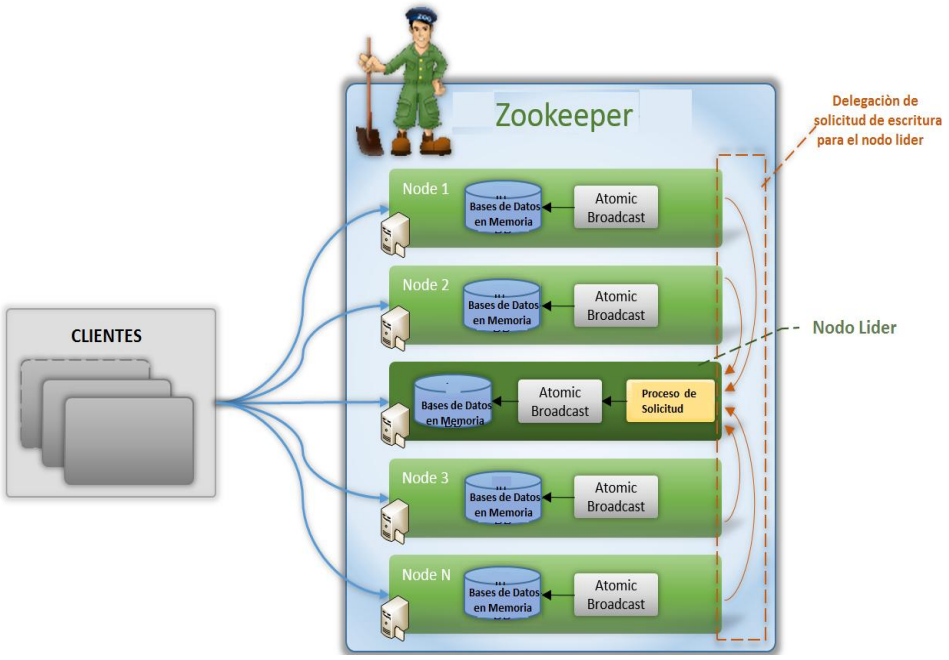

Figure 33 - Arquitectura de Zookeeper
Fuente: (Sain Technology Solutions, 2015)

CAPÍTULO 3
MARCO METODOLÓGICO

El proceso de desarrollo de software es un proceso afectado por la creatividad y juicio de las personas involucradas. En el desarrollo de software hay una serie de desafíos adicionales, relativos esencialmente a la naturaleza del producto obtenido. Un proceso de desarrollo de software tiene como propósito la producción eficaz y eficiente de un producto software que reúna los requisitos del cliente. Se puede definir de manera más forma al proceso de desarrollo de software como " aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos son transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo" (Jacobson ,1998)

3.1. Metodologías de Desarrollo de Software

Un área importante de la ingeniería de software es el desarrollo de metodologías y modelos. En la actualidad ha habido muchos esfuerzos que se han encaminado al estudio de los métodos y técnicas para lograr una aplicación más eficiente de las metodologías y lograr sistemas más eficientes y de mayor calidad con la documentación necesaria en perfecto orden y en el tiempo requerido. Una metodología impone un proceso de forma disciplinada sobre el desarrollo de software con el objetivo de hacerlo más predecible y eficiente. Una metodología define una representación que permite facilitar la manipulación de modelos, y la comunicación e intercambio de información entre todas las partes involucradas en la construcción de un sistema.

Existen diferentes modelos y metodologías que han sido en los últimos años herramientas de apoyo para el desarrollo del software. (Sommerville, Ingenieria del Software, 2005) Menciona que:

- Modelo de desarrollo de software: es una representación simplificada del proceso para el desarrollo de software, presentada desde una perspectiva específica.
- Metodología de desarrollo de software: es un enfoque estructurado para el desarrollo de software que incluye modelos de sistemas, notaciones, reglas, sugerencias de diseño y guías de procesos.

3.2. Desarrollo Basado en Componentes

La metodología de software basada en Componentes (CBSE) surgió a finales de los 90's como una aproximación basada en la reutilización al desarrollo de sistemas de software. Está metodología fue motivada por la frustración de los desarrolladores de que el modelo orientados a objetos no aplicaba una reutilización extensiva, tal como ésta sugería originalmente, debido a que la utilización de clases implica el conocimiento detallado de ellas, lo cual significa que se debía tener acceso al código fuente lo que imposibilitaba el marketing de clases para su reutilización.

En la actualidad la complejidad de los sistemas computacionales ha aumentado y nos ha llevado a buscar la reutilización del software existente. El desarrollo de software basado en componentes permite reutilizar piezas de código preelaborado que permiten realizar diversas tareas, conllevando a diversos beneficios como las mejoras a la calidad, la disminución del ciclo de desarrollo y el mayor retorno sobre la inversión.

En efecto, al reutilizar código, artefactos, no solo nos aseguramos de no cometer los mismos errores del pasado, sino que logramos construir cosas cada vez más grandes y maravillosas, con bases firmes y calidad incomparable. Este concepto de la reutilización, es uno de los primeros que se nos enseñan a quienes entramos al mundo del desarrollo de software, tendremos de utilizarlo desde el mismo instante en que escribamos nuestra primera línea de código.

Este enfoque basado en la reutilización se compone de una gran base de componentes software reutilizables y de algunos marcos de trabajo de integración para éstos. Algunas veces estos componentes son sistemas por sí mismos (COTS o sistemas comerciales) que se pueden utilizar para proporcionar una funcionalidad específica, como dar formato al texto o efectuar cálculos numéricos. En la **Figura 34** se muestra el modelo del proceso genérico para CBSE.

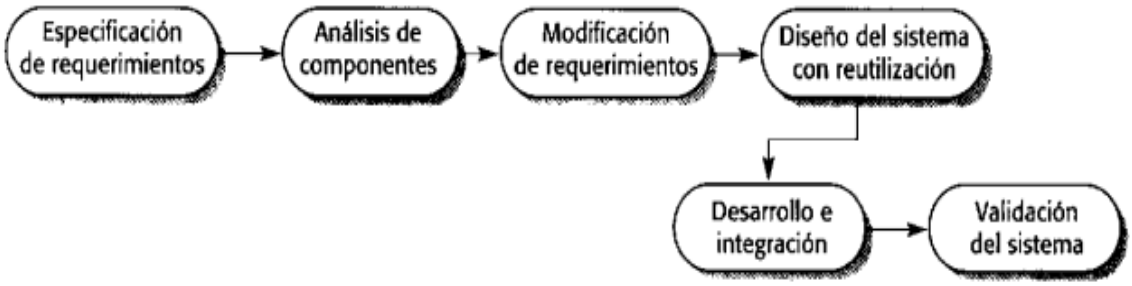

Figura 34 - Ingeniería del Software Basada en Componentes
Fuente: (Sommerville, Ingenieria del Software, 2005)

Aunque la etapa de especificación de requerimientos y la de validación son comparables con otros procesos, las etapas intermedias en el proceso orientado a la reutilización son diferentes. Estas etapas son:

- **Análisis de Componentes.** Dada la especificación de requerimientos, se buscan los componentes para realizar esta especificación. Por lo general, no existe una concordancia exacta y los componentes que se utilizan sólo proporcionan parte de la funcionalidad requerida.
- **Modificación de Requerimientos.** En esta etapa, los requerimientos se analizan utilizando información acerca de los componentes que se han descubierto. Entonces, estos componentes se modifican para reflejar los componentes disponibles. Si las

modificaciones no son posibles, la actividad de análisis de componentes se puede llevar a cabo nuevamente para buscar soluciones alternativas.

- **Diseño del Sistema con Reutilización.** En esta fase se diseña o se reutiliza un marco de trabajo para el sistema. Los diseñadores tienen en cuenta los componentes que se reutilizan y organizan el marco de trabajo para que los satisfaga. Si los componentes reutilizables no están disponibles, se puede tener que diseñar nuevo software.
- **Desarrollo e Integración.** Para crear el sistema, el software que no se puede adquirir externamente se desarrolla, y los componentes y los sistemas se integran. En este modelo, la integración de sistemas es parte del proceso de desarrollo, más que una actividad separada.

  Los fundamentos de la ingeniería del software basada en componentes son (Sommerville, Ingenieria del Software, 2005):

- **Componentes Independientes**, que son completamente especificados por sus interfaces. Debería haber una clara separación entre la interfaz de los componentes y su implementación para que una implementación de un componente pueda reemplazar por otro sin cambiar el sistema.
- **Estándares de Componentes,** que facilitan la integración de los componentes. Estos estándares se incluyen en un modelo de componentes y definen, en el nivel más bajo cómo las interfaces de componentes deberían especificarse y cómo se comunican sus componentes. Algunos modelos definen interfaces que deberían implementarse conforme a todos los componentes. Si los componentes cumplen con los estándares, entonces su funcionamiento es independiente de su lenguaje de programación. Los componentes escritos en diferentes lenguajes pueden integrarse en el mismo sistema.
- **El Middleware**, que proporciona soportes software para la integración de componentes. Para conseguir que componentes independientes y distribuidos trabajen juntos, necesita un soporte middleware que maneje las comunicaciones de los componente Un producto middleware como CORBA (Pope, 1998), maneja cuestiones de bajo nivel de forma eficiente y permite al diseñador centrarse en problemas relacionados con la aplicación. Además, un producto middleware utilizado para implementar un modelo de componentes puede proporcionar soporte para la asignación de recursos, gestión de transacciones, seguridad y concurrencia.
- **Un Proceso de Desarrollo**, que se adapta a la ingeniería del software basada en componentes. Si se intenta añadir una aproximación basada en componentes a un proceso de desarrollo que está adaptado a la producción de software original, se puede observar que las suposiciones inherentes al proceso limitan el potencial del CBSE.

  El desarrollo basado en componentes se está admitiendo cada vez más como una aproximación fundamental a la ingeniería del software incluso aunque los componentes reutilizables no estén disponibles. CBSE está estableciendo sobre unos principios de diseño sólidos que soportan la construcción de software comprensible y

sostenible. Los componentes son independientes para que no interfieran en su funcionamiento unos con otros.

Los detalles de implementación se ocultan, por lo que la implementación de los componentes puede cambiarse sin afectar al resto del sistema mientras que los componentes se comunican a través de interfaces bien definidas, de forma que si estas interfaces se mantienen, un componente puede reemplazarse por otro que proporcione una funcionalidad adicional o mejorada (Sommerville, Ingenieria del Software, 2005).

Además, las infraestructuras de componentes proporcionan plataformas de alto nivel que reducen los costes del desarrollo de aplicaciones.

Aunque CBSE se está convirtiendo en una aproximación fundamental para el desarrollo del software, presenta varios problemas:

- **Confiabilidad de los Componentes.** Los componentes son cajas negras de unidades de programas, y el código de los componentes puede no estar disponible para los usuarios de dichos componentes. En tales casos, ¿cómo sabe un usuario que un componente es confiable? El componente puede no tener documentados modos de fallo que comprometen el sistema en el que se utiliza dicho componente. Su comportamiento no funcional puede no ser el esperado, y un problema más grave es que el componente podría ser un caballo de Troya que oculta código dañino que viola la seguridad del sistema.
- **Certificación de Componentes.** Estrechamente relacionada con la confiabilidad se halla la cuestión de la certificación. Se ha propuesto que asesores independientes deberían certificar los componentes para asegurar a los usuarios que los componentes son confiables. Sin embargo, no está claro cómo se puede hacer esto. ¿Quién pagaría por la certificación?, ¿quién sería responsable si el componente no funciona de acuerdo con la certificación?, y ¿cómo podrían los certificadores limitar su responsabilidad? Desde nuestro punto de vista, la única solución viable es certificar que los componentes cumplen una especificación formal. Sin embargo, la industria no parece dispuesta a pagar por esto.
- **Predicción de Propiedades Emergentes.** Todos los sistemas tienen propiedades emergentes, y el intentar predecir y controlar estas propiedades emergentes es importante en el proceso de desarrollo del sistema. Debido a que los componentes son opacos, predecir sus propiedades emergentes es particularmente difícil. Como consecuencia, es posible encontrarse con que, cuando se integran componentes, el sistema resultante tiene propiedades no deseadas que limitan su uso.
- **Equilibrio de Requerimientos.** Normalmente se tiene que encontrar un equilibrio entre los requerimientos ideales y los componentes disponibles en el proceso de especificación y diseño del sistema. Por el momento, alcanzar este equilibrio es un proceso intuitivo. Necesitamos un método de análisis de equilibrio sistemático y más estructurado para ayudar a los diseñadores a seleccionar y configurar componentes.

3.2.1. Ventajas de Desarrollo Basado en Componentes

El uso de este paradigma posee algunas ventajas:

Tabla 6- Ventajas de desarrollo basado en componentes

Reutilización del software	Nos lleva a alcanzar un mayor nivel de reutilización de software.
Simplifica las pruebas.	Permite que las pruebas sean ejecutadas probando cada uno de los componentes antes de probar el conjunto completo de componentes ensamblados.
Simplifica el mantenimiento del sistema.	Cuando existe un débil acoplamiento entre componentes, el desarrollador es libre de actualizar y/o agregar componentes según sea necesario, sin afectar otras partes del sistema.
Mayor calidad	Dado que un componente puede ser construido y luego mejorado continuamente por un experto u organización, la calidad de una aplicación basada en componentes mejorará con el paso del tiempo

La ingeniería del software basada en componentes tiene la ventaja obvia de reducir la cantidad de software a desarrollarse y así reduce los costos y los riesgos. Por lo general, también permite una entrega más rápida del software. Sin embargo, los compromisos en los requerimientos son inevitables, y esto puede dar lugar a un sistema que no cumpla las necesidades reales de los usuarios. Más aún: si las nuevas versiones de los componentes reutilizables no están bajo el control de la organización que los utiliza, se pierde parte del control sobre la evolución del sistema. La CBSE tiene mucho en común con un enfoque que está surgiendo para el desarrollo de sistemas que se basa en la integración de servicios web de una serie de proveedores.

3.2.2. Modelo de Componentes

Un modelo de componentes es una definición de los estándares para la definición de componentes, documentación y despliegue. Estos estándares son utilizados por los desarrolladores de componentes para asegurar que los componentes puedan interoperar. También son utilizados por los proveedores de infraestructuras de comunicación de los componentes, quienes proporcionan middleware para soportar el funcionamiento de éstos (Sommerville, Ingenieria del Software, 2005).

El diagrama muestra que en la **Figura 35** los elementos en un modelo de componentes pueden clasificarse como elementos relacionados con las interfaces de los componentes, elementos relacionados con la información que se necesita para utilizar el componente en un programa y elementos relacionados con el despliegue del componente. Los elementos que definen a un componente son sus interfaces. El modelo de componentes especifica cómo deberían definirse las interfaces y los elementos, tales como nombres de operaciones, parámetros y excepciones que deberían incluirse en la definición de una interfaz. El modelo también debería especificar el lenguaje utilizado para definir las interfaces.

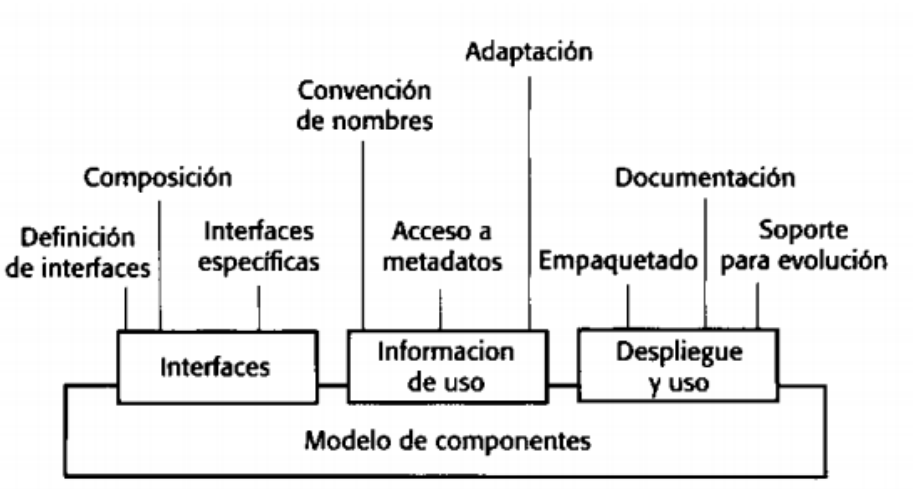

Figura 35 - Elementos básicos de un Modelo de Componentes
Fuente: (Sommerville, Ingenieria del Software, 2005)

Los elementos básicos de un modelo de componentes se pueden clasificarse en tres grandes grupos según su relación:

- Las interfaces
- La información que se necesita para utilizar el componente
- El despliegue del componente.

3.2.3. Componentes

Un componente software es una unidad de composición con interfaces especificadas contractualmente y dependencias de contexto explícitas únicamente. Un componente software puede ser desplegado de forma independiente y está sujeto a la composición por terceras partes (Szyperski, 2002)

Szyperski también constata que un componente no tiene un estado externamente observable. Esto significa que las copias de componentes son indistinguibles.

La **Figura 36** muestra las características esenciales de un componente para ser usado en CBSE.

Características del componente	Descripción
Estandarizado	La estandarización de componentes significa que un componente usado en un proceso CBSE tiene que ajustarse a algún modelo estandarizado de componentes. Este modelo puede definir interfaces de componentes, metadatos de componentes, documentación, composición y despliegue.
Independiente	Un componente debería ser independiente, debería ser posible componerlo y desplegarlo sin tener que utilizar otros componentes específicos. En las situaciones en las que el componente necesita servicios proporcionados externamente, éstos deberían hacerse explícitos en una especificación de interfaz del tipo «requiere».
Componible	Para que un componente sea componible, todas las interacciones externas deben tener lugar a través de interfaces definidas públicamente. Además debe proporcionar acceso externo a la información sobre sí mismo, como por ejemplo a sus métodos y atributos.
Desplegable	Para ser desplegable, un componente debe ser independiente y debe ser capaz de funcionar como una entidad autónoma o sobre una plataforma de componentes que implemente el modelo de componentes. Esto normalmente significa que el componente es binario y que no tiene que compilarse antes de ser desplegado.
Documentado	Los componentes tienen que estar completamente documentados para que los usuarios potenciales puedan decidir si los componentes satisfacen o no sus necesidades. La sintaxis e, idealmente, la semántica de todas las interfaces de componentes tienen que ser especificadas.

Figura 36 - Características de los Componentes
Fuente: (Sommerville, Ingenieria del Software, 2005)

3.2.4. Interfaces de los Componentes

Las interfaces son aquellas que separan el comportamiento de los componentes de la implementación, de esta forma tenemos dos interfaces que están relacionadas, la primera es la interfaz que proporciona y la segunda es la interfaz que requiere, como se muestra en la **Figura 37** (Sommerville, Ingenieria del Software, 2005)

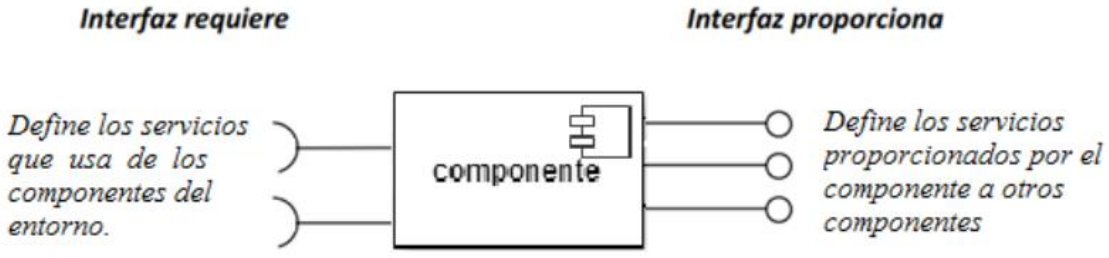

Figura 37 - Interfaces de los Componentes
Fuente: (Sommerville, Ingenieria del Software, 2005)

La interfaz que **proporciona** como su mismo nombre indica define los servicios que proporciona el componente, en otras palabras, definen los métodos que pueden ser llamados por el usuario y que pertenecen al componente, también es conocido como el API del componente. La interfaz que requiere describe los servicios que son necesarios para que el componente funcione correctamente, estos a su vez son requeridos por otros componentes del sistema. Sin alterar la independencia del componente, ya que, los servicios que se requieren no son solicitados a un componente específico.

Los fundamentos del desarrollo basada en componentes son:

- Componentes independientes completamente especificados por sus interfaces.
- Estándares de componentes que facilitan su integración, estos definen la descripción de las interfaces y como se comunican los componentes.
- Middleware provee soporte de software para la unificación de componentes, permite que los componentes independientes y distribuidos trabajen juntos, maneja las comunicaciones entre componentes, proporciona soporte para la asignación de recursos, gestión de transacciones, seguridad y concurrencia.

3.2.5. Creación del Sistema con la Composición de Componentes

El proceso de enlazar componentes para crear un sistema es conocido como la creación del sistema con la composición de componentes, este proceso se lleva a cabo en cuatro actividades:

- **Análisis de los Componentes:** se establece qué tan adecuado es un componte específico para la construcción del sistema.
- **Adaptación de los Componentes:** Cuando un componente está orientado hacia un contexto determinado y tiene funcionalidades determinadas y es necesario llevarlo en otra dirección o enfoque es necesario adaptar el componente.
- **Ensamblaje de los Componentes:** En este punto se integran los componentes, esto se efectúa según la estructura mediante la cual fueron definidos. Existen tres tipos de composiciones como se puede apreciar en la **Figura 38.**
    a) Composición secuencial: ocurre cuando los componentes que forman el componente compuesto se ejecutan en secuencia y se requiere algún código extra para enlazar los componentes.
    b) Composición jerárquica: sucede cuando un componente realiza una llamada directamente a los servicios prestados a otro componente, se corresponde con la situación.
    c) Composición aditiva: las interfaces de dos o más componentes se fusionan para crear las interfaces de un nuevo componente.
- **Mantenimiento:** Inmediatamente de que el sistema ya se encuentra en funcionamiento lo más factible es que sea necesario hacer cambios en los componentes bien sea, por nuevos requerimientos o inconvenientes con los mismos, estos cambios logran consistir en la reescritura del componente o en el peor de los casos ocasiona la sustitución del mismo

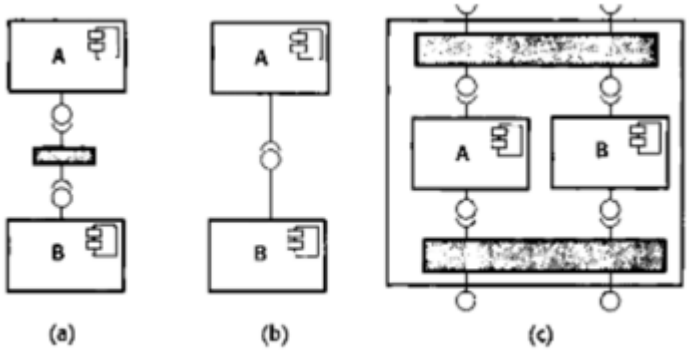

Figura 38 - Tipos de Composición de Componentes
Fuente: (Sommerville, Ingenieria del Software, 2005)

CAPÍTULO 4
MARCO APLICATIVO

En esta sección se explicará el desarrollo del Módulo de Indexación y Búsqueda, siguiendo la metodología descrita en el capítulo anterior, así como también las tecnologías utilizadas en cada actividad del proyecto, en caso de aplicar.

4.1. Definición de Requerimientos

Para este Trabajo Especial de Grado se agrega una solución a un sistema que actualmente se encuentra en funcionamiento, el cual ya preserva páginas Web venezolanas, alojando los archivos WARC en un Cluster Hadoop. El requerimiento principal es lograr la escalabilidad horizontal a fin de poder expandir todo el sistema de almacenamiento, agregando en este caso un módulo para poder indexar y buscar por palabras claves en los contenido almacenado en un formato WARC en Hadoop (donde se almacenan los archivos WARC de manera distribuida en los Nodos de Datos) permitiendo manejar gran volumen de información con un resultado aceptable de respuesta en caso de que se requiera.

4.2. Diseño Técnico

Existen varios sitios que se dedican a la preservación Web, como por ejemplo Portuguese Web Archive, Internet Memories Foundation, Padicat y entre otros, la mayoría de ellos utilizan a Hadoop, como el framework distribuido, y como indexador utilizan herramienta como Solr.

Inicialmente se pensó utilizar ElasticSearch como herramienta para indexar, sin embargo al realizar pruebas se notó que la etapa de indexación de documentos era un proceso más tedioso y no existe suficiente documentación de integración con Hadoop a comparación de SolrCloud.

Tomando en consideración estas premisas y además que se requiere acceder a los contenidos almacenado en los Warc's, se desarrolló un Módulo de indexación y búsqueda para la búsqueda de los contenidos Web utilizando Hadoop para el almacenamiento de los archivos bajo el formato WARC, ambas herramientas se encuentran disponibles en la Web como Software Libre.

En el Desarrollo del Módulo, se realizó una modificación de la arquitectura del Prototipo de Archivo Web de la Facultad de Ciencias. En la **Figura 39** se muestra la Arquitectura con la modificación añadiendo el nuevo Módulo de Indexación y Búsqueda.

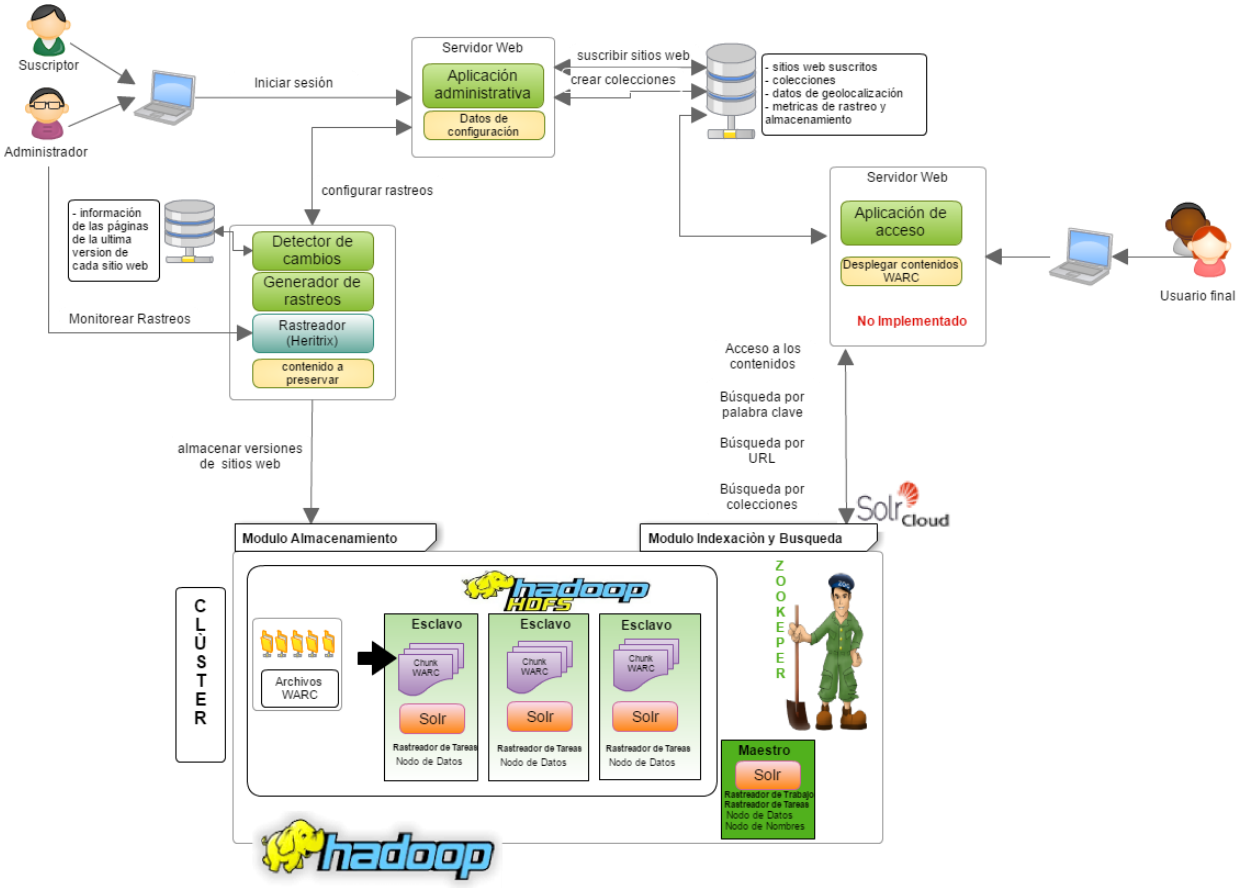

Figura 39 - Arquitectura Archivo Web con el Modulo de Indexación y Búsqueda
Fuente: Elaboración Propia

En este nuevo módulo podemos distinguir dos etapas:

- **Etapa de Indexación:** Una vez que todos los archivos Warc's son obtenido de Hadoop, se almacenan de manera temporal, para analizar y extraer las palabras claves de los contenidos en los HTML almacenado en los Warc's, y posteriormente ser indexado en SolrCloud pasándole la palabra clave, el URL, la versión y el WARC donde está contenido.
- **Etapa de Búsqueda:** Dado una palabra clave se realiza el proceso de consulta en SorlCloud, donde este retorna un JSON con los resultados donde la palabra clave coindice con los índices de SorlCloud, este JSON estará estructurado con la URL, la versión , el WARC y la palabra clave.

4.2.1. Diagrama de Componentes

Como se estudió en la sección anterior para esta nueva funcionalidad se decidió utilizar una arquitectura basada en componentes, el Diagrama de Componentes se encuentra dividido en dos niveles, de esta forma se puede explicar de manera más clara y sencilla. El Nivel 1 referente a la **Figura 40** explica como el Módulo de Indexación y Búsqueda interactúa con

el sistema utilizando interfaces en PYTHON para realizar la búsqueda e indexación por palabra clave en SolrCloud.

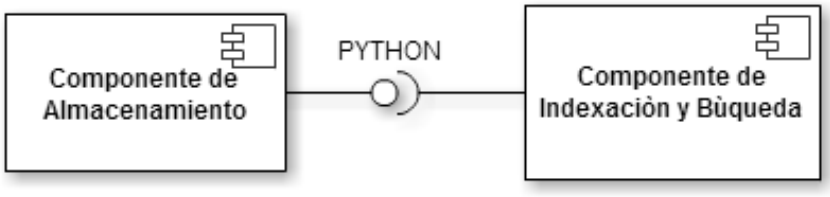

Figura 40 – Nivel 1 Diagramas de Componentes
Fuente: Elaboración Propia

El Nivel 2 referente a la **Figura 41** nos muestra de manera minimalista como esta implementado el Módulo de Indexación y Búsqueda, en este caso se encuentra un Componente Analizador, el cual está desarrollado en código PYTHON, que se va a encargar de analizar los archivos de Hadoop y pasarlo a SolrCloud, para realizar las acciones de descarga e indexación.

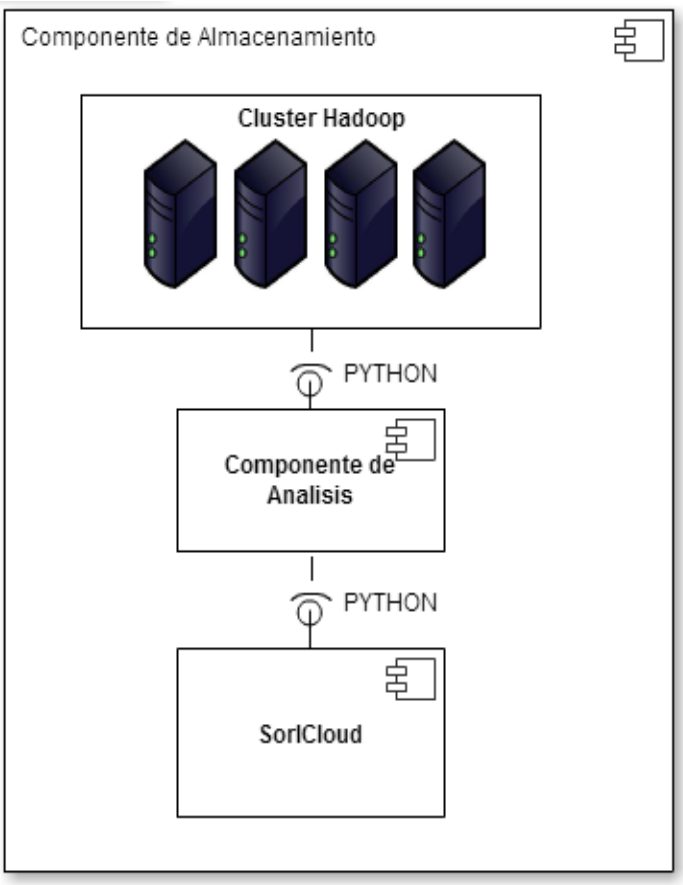

Figura 41 - Nivel 2 Diagramas de Componentes
Fuente: Elaboración Propia

4.2.2. Diagrama de Secuencias

En el Diagrama de Secuencias del nuevo Componente de Indexación y Búsqueda que se muestra en la **Figura 42**, se encuentran 4 objetos para mostrar el proceso de indexación de los contenidos de los archivos WARC de Hadoop para SolrCloud.

Con respecto al proceso de indexación, los archivos son buscados en Hadoop y se procede a retorna todos los archivos WARC que no ha sido indexados, almacenándolos de manera temporal para su análisis, una vez almacenado automáticamente comienza el proceso de análisis de los mismos, primero se procede abrir los Warc's, obtener los HTML para realizar el proceso de limpieza donde va obtener las palabras clave que serán guardadas en el documento que se indexan en SolrCloud.

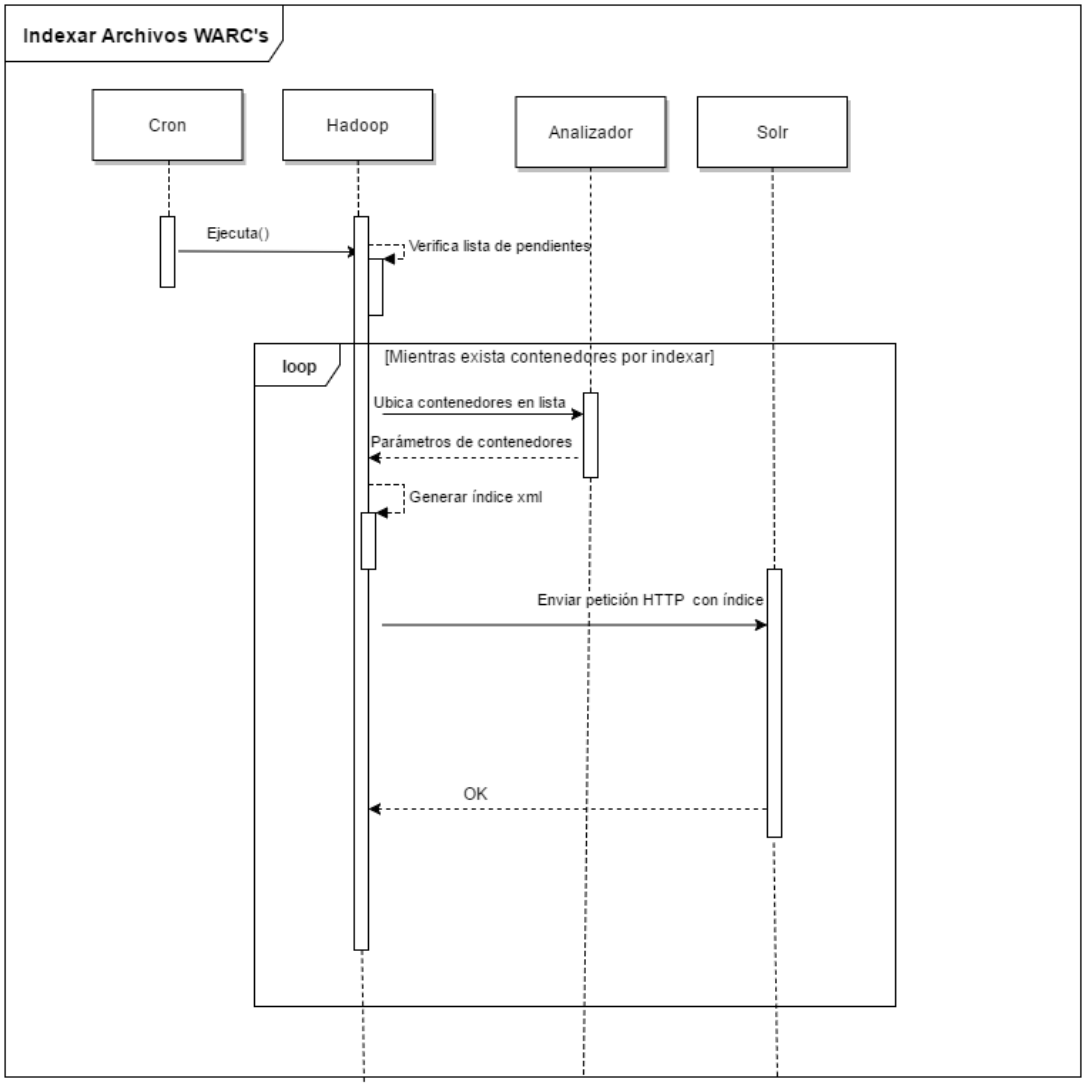

Figura 42 - Diagrama de Secuencias
Fuente: Elaboración Propia.

Con respecto a la fase de búsqueda por URL o palabra clave dentro de los contenidos se va utilizar 3 objetos para modelar el proceso de búsqueda en Solr lo cual podemos presenciar el modelaje del mismo en el diagrama de secuencias de la **Figura 43**.

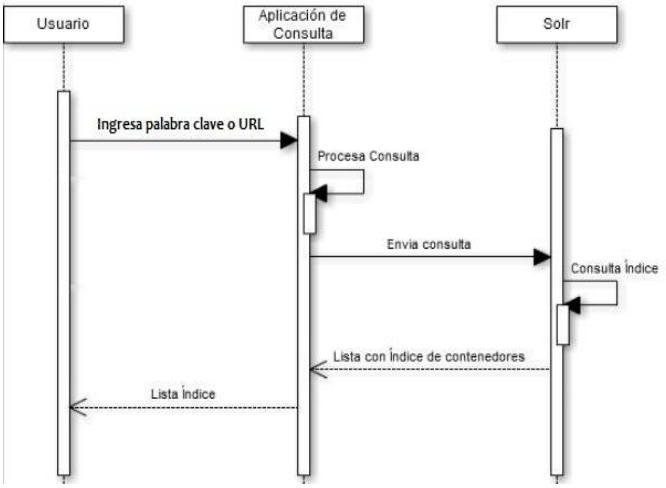

Figura 43 - Diagrama de Secuencia Búsqueda
Fuente: Elaboración Propia

4.3. Definición de Herramientas

A continuación se indicarán cada una de las herramientas utilizadas en el Desarrollo del Módulo de Indexación y Búsqueda:

4.3.1. WarcTools

Está compuesto por una serie de scripts en lenguaje Python, que se utiliza para crear y manipular los archivos WARC. Esta herramienta se va utilizar en este módulo para filtrar las paginas HTML y extraer la información contenida en los HTML.

4.3.2. Secure Shell (SSH)

SSH o en español, Intérprete de Órdenes Seguro, es un protocolo que sirve para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos, el cual permite la conexión entre lo múltiples nodos de Hadoop.

4.3.3. Hadoop

Apache Hadoop es un Framework que soporta aplicaciones distribuidas bajo una licencia libre. Permite a las aplicaciones trabajar con miles de nodos y Petabytes de datos. Hadoop es un proyecto de alto nivel que está siendo construido y usado por una comunidad global, mediante el lenguaje de programación Java. Este framework es el que se usa en el clúster donde se almacena los archivos Warc's.

4.3.4. SolrCloud

Apache Solr incluye la capacidad de configurar un clúster de servidores Solr que combina la tolerancia a fallos y alta disponibilidad. Llaman a SolrCloud**,** a las capacidades de indexación y búsqueda distribuida. Una vez instalado SolrCloud, se debe proceder a la configuración del mismo, ya que por defecto viene para trabajar en Modo Single, es decir,

de manera local, con un solo nodo que es tanto Nodo Maestro (donde se ejecutan todos los comandos de SolrCloud) como Nodo de Datos (donde se almacenan los índices). Para el desarrollo del Módulo de Indexación y Búsqueda se requiere la configuración Multi-Nodo, para ello se debe ejecutar el comando que se puede visualizar en la **Figura 44**, que este brindara una interfaz por consola que se encargara de modificar una serie de archivos relacionados con SolrCloud.

```
hilda@hilda-P1400:/opt/solr-6.2.1$ sudo bin/solr -e cloud
```

Figura 44 - Comando SolrCloud
Fuente: Elaboración Propia

A continuación se indicará en la **Figura 45** la configuración realizada en el desarrollo del Módulo de Indexación y Búsqueda, donde se configuró 1 Nodo Maestro y 2 Nodos de Datos.

```
Welcome to the SolrCloud example!

This interactive session will help you launch a SolrCloud cluster on your local
workstation.
To begin, how many Solr nodes would you like to run in your local cluster? (spec
ify 1-4 nodes) [2]:
2
Ok, let's start up 2 Solr nodes for your example SolrCloud cluster.
Please enter the port for node1 [8983]:
8987
Please enter the port for node2 [7574]:
7578
Creating Solr home directory /opt/solr-6.2.1/example/cloud/node1/solr
Cloning /opt/solr-6.2.1/example/cloud/node1 into
   /opt/solr-6.2.1/example/cloud/node2

Starting up Solr on port 8987 using command:
bin/solr start -cloud -p 8987 -s "example/cloud/node1/solr"

Waiting up to 30 seconds to see Solr running on port 8987 [-]
Started Solr server on port 8987 (pid=14711). Happy searching!


Starting up Solr on port 7578 using command:
bin/solr start -cloud -p 7578 -s "example/cloud/node2/solr" -z localhost:9987

Waiting up to 30 seconds to see Solr running on port 7578 [/]
Started Solr server on port 7578 (pid=14926). Happy searching!


Now let's create a new collection for indexing documents in your 2-node cluster.
Please provide a name for your new collection: [gettingstarted]
archiveWebVenezuela
How many shards would you like to split archiveWebVenezuela into? [2]
2
How many replicas per shard would you like to create? [2]
2
Please choose a configuration for the archiveWebVenezuela collection, available
```

Figura 45 - Configuración SolrCloud
Fuente: Elaboración Propia

4.3.5. Proceso de Análisis

Para generar los índices en SolrCloud por palabra clave de los contenido, se procede a generar un documento JSON que va contener las palabras claves de los HTML contenido en los archivos WARC, para este análisis se crea un algoritmo en PYTHON que realiza el proceso de análisis, que sigue los pasos mostrados en la **Figura 46.** Para la ejecución de este proceso se debe cumplir con la condición de que exista uno o varios WARCS a indexar en Hadoop.

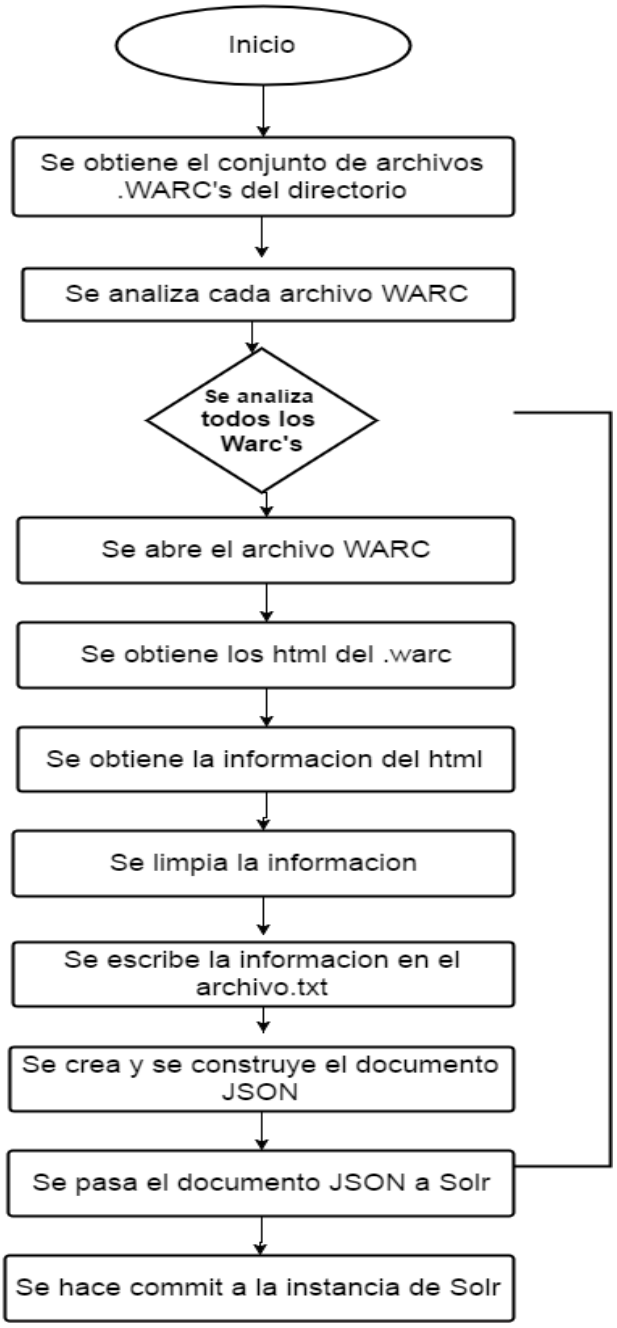

Figura 46 - Diagrama De Flujo Proceso de Análisis
Fuente: Elaboración Propia

El primer paso es realizar la extracción de los datos - correspondientes a los archivos WARC rastreados previamente – del Cluster distribuido, la cual se encuentra en *Hadoop*. Se procede a ejecutar el job de Hadoop para acceder al Cluster, con el objetivo de ejecutar las sentencias que extrae todos los Warc's, ésta última se puede visualizar en la **Figura 47**.

```
shell_exec('/usr/local/hadoop/bin/hdfs dfs -get /input/uploads/
'.$file[$i].' /var/www/html/uploads 2>&1');
```

Figura 47 - Código para descargar archivos desde Hadoop
Fuentes: (Estaba Fernández-Trujillo. & Ciancia Biondo, 2015)

Posteriormente, se procede abrir los Archivos Warc's utilizando la biblioteca *Hanzo Warc Tools* y el siguiente comando: *"ArchiveRecord.open_archive".* Una vez ejecutado, se procede a identificar los archivos HTML resultante de la extracción y extraer los archivos en varios archivos/directores, que serán analizados cada uno para extraer la información de los HTML utilizando la biblioteca *BeautifulSoup* **Figura 48***,* donde vamos a obtener las cabecera y el cuerpo del HTML que luego de hacerle una limpieza quitándoles los javascript embebidos será insertado en SolrCloud. En paralelo a este procedimiento se genera un archivo txt que contiene la información que se le va pasar a Solr para llevar un registro, el cual sirve como un log de los datos que se están pasando.

```
try:
    soup = BeautifulSoup(html_string)
except:
    print "Error: No se puede transforma el html a archivo: " + html_file
    return

if len(html_string) < 1:
    print "error transformando html a archivo " + str(html_file)
    return

#extracion de informacion usando soup
html_body = soup.get_text()
html_id = hashlib.md5(file_url).hexdigest()

text_file = open(html_file + ".txt", "w+b")
text_file.write(html_body)
text_file.close()
```

Figura 48 - Extracción de Información BeautifulSoup
Fuente: Elaboración Propia

Finalmente, se manda un documento con la estructura que se puede visualizar en la **Figura 49** a la herramienta de SolrCloud el cual fue accedido utilizando el conector *Solr*, para que este haga el proceso de indexación de la información contenida en el documento generado en el proceso descrito anteriormente.

```
doc = {"id":html_id, "content":html_body, "nombreWarc":nombreWarc, "URL":file_url, "_version_":version}

try:
    solr_instance.add(doc)
except Exception as inst:
    print "Error indexando el archivo en Solr" + str(inst)
```

Figura 49 – Código Fuente pasando Documento a Solr
Fuente: Elaboración Propia

4.3.6. Programar una aplicación con lenguaje PYTHON

Para permitir la interacción entre las aplicaciones instaladas, se elaboraron un prototipo que fue ajustado a la interfaz actual del prototipo de Archivo Web existente.

4.3.6.1. Desarrollo del Prototipo para el Usuario Final

En este prototipo se modificó además de la interfaz, la forma en que se realizaba la búsqueda que era solo por URL, ahora se dispone de la funcionalidad de realizar búsqueda dentro de los contenido Warc's.

Adicionalmente, se implementa el uso de Logs, en el proceso de realizar la indexación, se crea un archivo txt en la ruta /var/indexación/www/logs donde se almacena toda la información que se está procesando detrás de la interfaz, de tal manera que si ocurre un fallo en el log.txt se observara toda la corrida y se podrá detectar el error.

A continuación se muestra la interfaz final creada para este prototipo **Figura 50**.

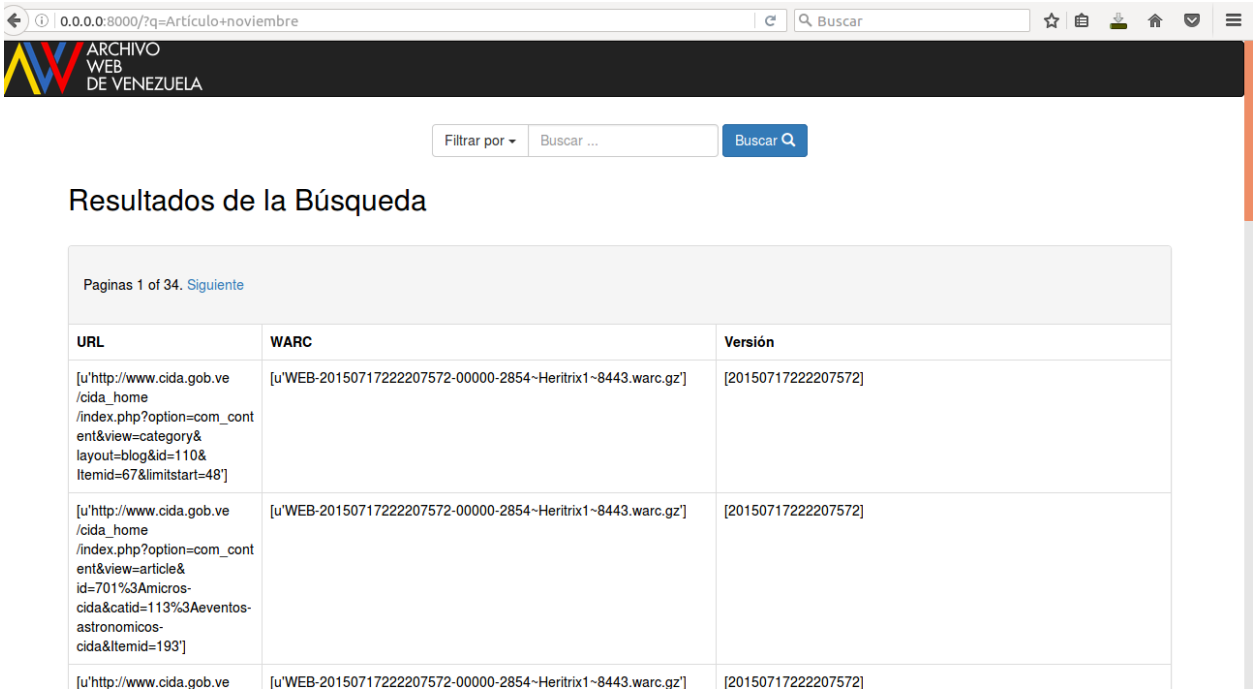

Figura 50 - Interfaz de Búsqueda
Fuente: Elaboración Propia

4.4. Pruebas

4.4.1. Prueba de rendimiento

Debido a que existen procesos que son ejecutados en background, el modulo fue desarrollado para crear varios archivo de texto plano "Logs", lo cual van a permitir llevar un seguimiento del proceso de análisis de los Warc's en la fase de indexación, para así, realizar validaciones y encontrar fallos en casos que ocurriesen. Como los logs no se sobrescriben, se puede saber que ha ocurrido, por ejemplo, se ejecuta el proceso de análisis del archivo WARC, y en ese momento presento una falla el nodo donde se va a mandar el documento,

se procede a generar un archivo con el nombre del WARC el cual va servir para que un paso siguiente se mande a indexar el documentos de nuevo, se puede observar en la **Figura 51**.

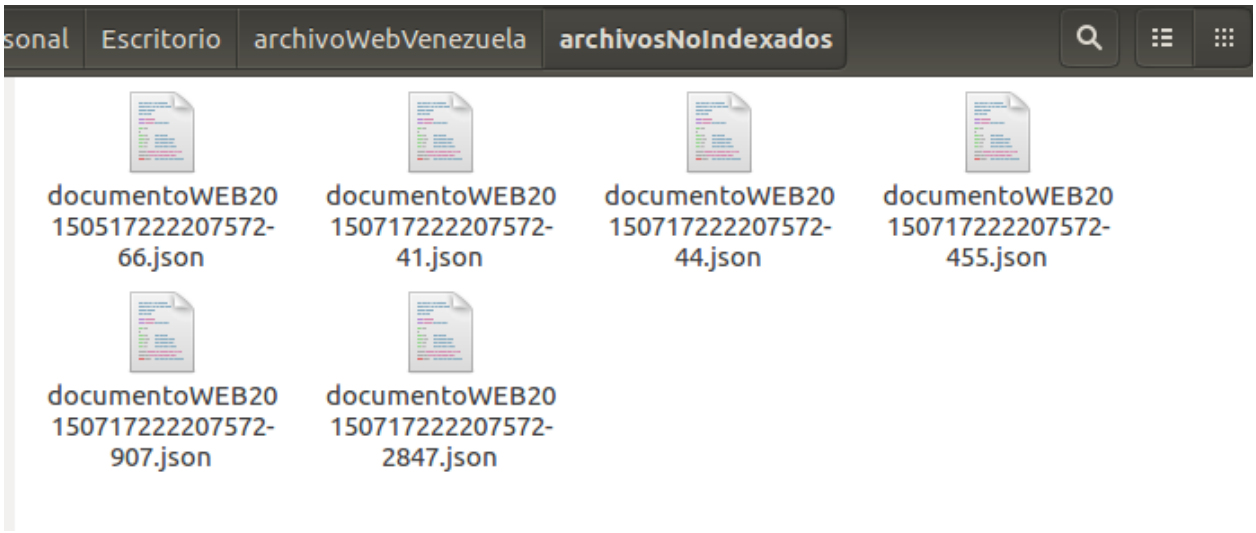

Figura 51 - Archivos no Indexados
Fuente: Elaboración Propia

De igual forma también se va generar un archivo log con el nombre "log advertencia" en el caso que no existan Warc's para indexar este archivo indicara que no existen archivos para realizar el proceso de indexación en Solr.

En caso contrario, de que se encuentre al menos 1 archivo WARC, se creara un archivo "logindexacion" **(Figura 52**), el cual indicara que en ese instante de tiempo(se especifica la fecha de la corrida) se está ejecutando el proceso de indexación, mostrando las diferentes etapas por la cual pasa el WARC para realizar la extracción e indexación de palabras claves contenida en dicho archivos, como se aprecia en la **Figura 53**.

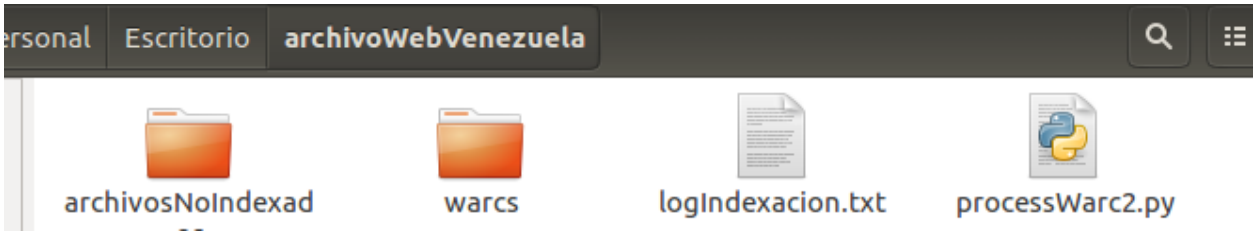

Figura 52 - Log Indexación en el Directorio
Fuente: Elaboración Propia

Figura 53 - Log Indexación
Fuente: Elaboración Propia

4.4.2. Análisis de tiempos de indexación

Para realizar el análisis del procesos de indexación se realizaron pruebas de pesos, cantidad de páginas contenida en los formatos Warc's y etapa de análisis por las que pasa un archivo WARC, con un recurso de 2 GB de RAM visualizar **Figura 54**, el cual hace que la indexación resulte un poco más lenta en comparación con otras máquinas que cuentan con mayor recursos, debido a que las instancias de SOLR requieren un mínimo de 8 GB de RAM para obtener un tiempo de ejecución aceptable, estas pruebas se hicieron bajo un ambiente de una 1 colección de SOLR con 2 nodo shard y 2 nodos de replicación por cada shard visualizar

Figura 55.

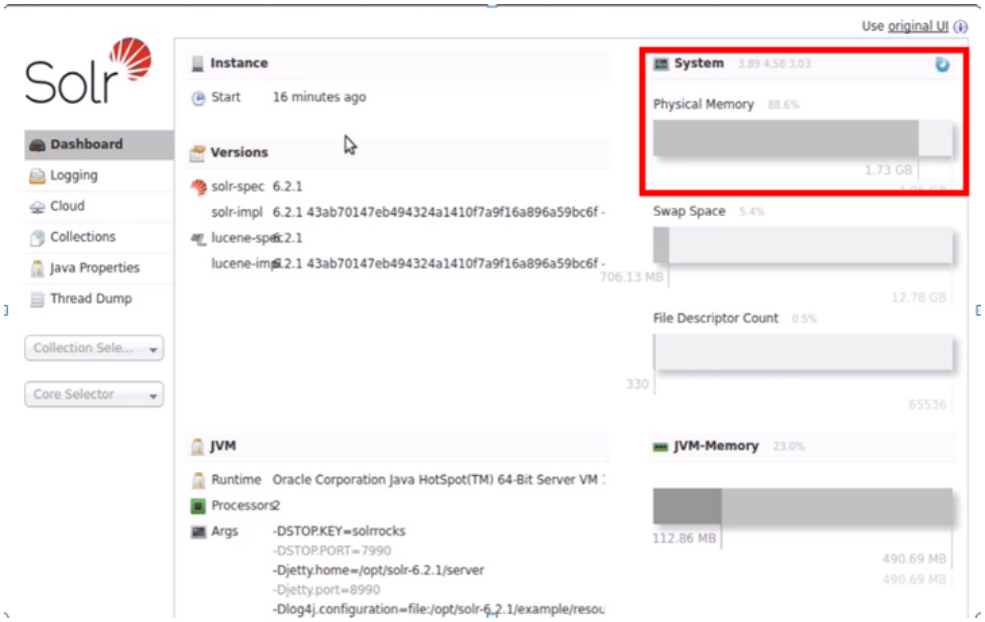

Figura 54- Ambiente de Prueba con un recurso de 2GB de RAM
Fuente: Elaboración Propia

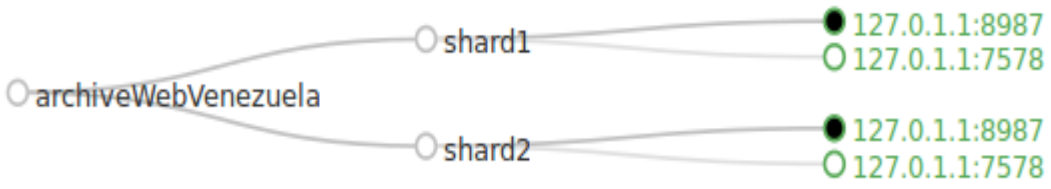

Figura 55 - Ambiente SolrCloud Prueba
Fuente: Elaboración Fuente

A continuación, se describe dichas pruebas.

4.4.2.1. Casos Peso

Para estas pruebas el proceso de indexación fue realizado en tres casos, el primer caso fue realizado cargando 5 WARC, el segundo caso fue realizado cargando 7 archivos WARC y el último caso se probó cargando 10 WARC de diferentes tamaños (desde unos pocos megas 167 MB hasta máximo 1GB), simulando que proviene del Módulo de Gestión de almacenamiento, dando como resultado todas las ejecuciones de manera exitosa

- **Cantidad de 5 Warc's**

Una vez realizada las pruebas pertinentes, se muestra el resultado de rendimiento de las misma. Cabe destacar para estos resultados se cuenta con un universo de 5 contenedores de distintos tamaños lo cual podemos visualizar en la **Tabla 7**.

Tabla 7 – Datos 5 Warc's peso

PAGINA WEB	PESO MB	CANTIDAD DE PAGINAS	TIEMPO
www.inzit.go.ve	5,8	502	0:01:50
www.fundacredesa.go.ve	0,003320313	0	0
www.cendit.gob.ve	28	2102	0:04:00
www.inzit.go.ve	7,1	618	0:02:50
www.cida.go.ve	126,9	3135	0:06:00
Total	**167,80**	**6357**	**0:14:00**

En cuanto al tiempo de indexación de los contenedores en la **Figura 56**, se muestra cuantos minutos tardó en llevarse a cabo el proceso de indexación.

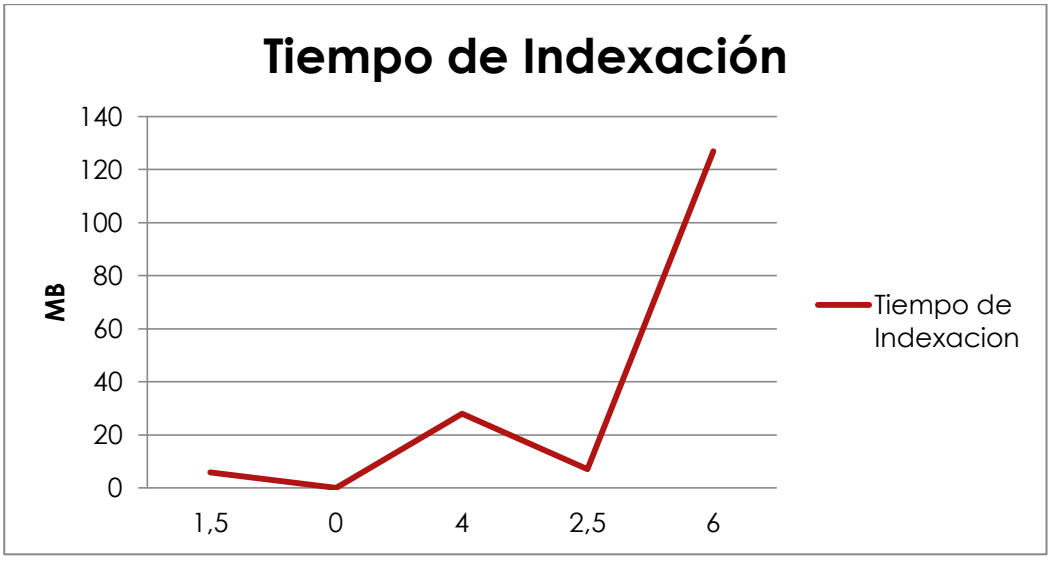

Figura 56 - Tiempo en Indexación warc's 5 por peso
Fuente: Elaboración Propia

- **Cantidad de 7 Warc's**

Una vez realizada las pruebas pertinentes, se muestra el resultado de rendimiento de las misma. Cabe destacar para estos resultados se cuenta con un universo de 7 contenedores de distintos tamaños lo cual podemos visualizar en la **Tabla 8**.

Tabla 8 -Datos 7 Warc's peso

PAGINA WEB	PESO MB	CANTIDAD DE PAGINAS	TIEMPO
www.inzit.go.ve	5,8	502	0:01:30
www.fundacredesa.go.ve	0,003320313	0	0
www.cendit.gob.ve	28	2102	0:03:00

PAGINA WEB	PESO MB	CANTIDAD DE PAGINAS	TIEMPO
www.inzit.go.ve	7,1	618	0:02:50
www.cida.go.ve	126,9	3135	0:05:00
www.fundacredesa.go.ve	4,4	400	0:01:00
www.cendit.gob.ve	256,9	5236	0:08:10
Total	**429,10**	**11993**	**0:22:00**

En cuanto al tiempo de indexación de los contenedores en la **Figura 57**, se muestra cuantos minutos tardó en llevarse a cabo el proceso de indexación

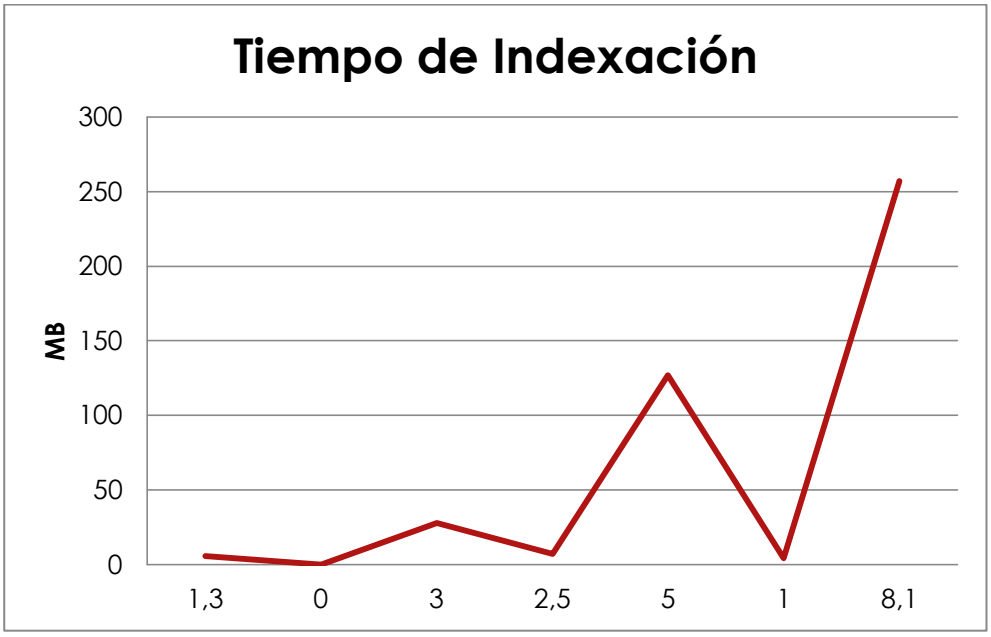

Figura 57 -Tiempo en Indexación warc's 7 por peso
Fuente: Elaboracion Propia

- **Cantidad de 10 Warc's**

Una vez realizada las pruebas pertinentes, se muestra el resultado de rendimiento de las misma. Cabe destacar para estos resultados se cuenta con un universo de 10 contenedores de distintos tamaños lo cual podemos visualizar en la **Tabla 9**

Tabla 9 - Datos 10 Warc's peso

PAGINA WEB	PESO MB	CANTIDAD DE PAGINAS	TIEMPO
www.inzit.go.ve	5.8	502	0:01:30
www.fundacredesa.go.ve	0,003320313	0	0

PAGINA WEB	PESO MB	CANTIDAD DE PAGINAS	TIEMPO
www.inzit.go.ve	5.8	502	0:01:30
www.cendit.gob.ve	28	2102	0:04:00
www.inzit.go.ve	7,1	618	0:02:30
www.cida.go.ve	126,9	3135	0:06:00
www.fundacredesa.go.ve	4,4	400	0:01:00
www.cendit.gob.ve	256,9	5236	0:08:10
www.inzit.go.ve	6,8	618	0:06:00
www.inzit.go.ve	7,8	630	0:02:50
www.fundacredesa.go.ve	356,9	6852	0:10:10
Total	**794,80**	**20093**	**0:42:10**

En cuanto al tiempo de indexación de los contenedores en la **Figura 58**, se muestra cuantos minutos tardó en llevarse a cabo el proceso de indexación

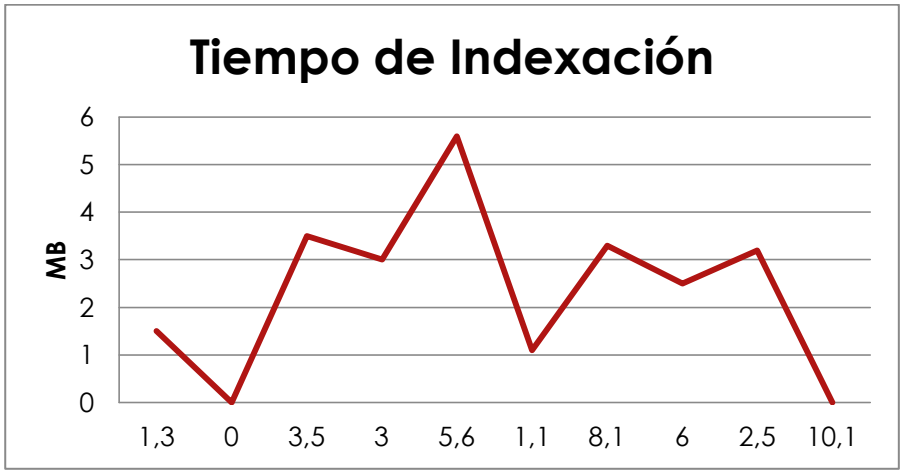

Figura 58 - Tiempo en Indexación warc's 10 por peso
Fuente: Elaboración Propia

Resultado Final

Tabla 10 - Resultado Final Datos casos por peso

Cantidad de WARC'S	Mb	Tiempo (Minutos)	Cantidad de Paginas
5	167	14	**6357**
7	429	21	**11993**

	Cantidad de WARC'S	Mb	Tiempo (Minutos)	Cantidad de Paginas
	10	794	42	**20093**
Total	**22**	**1390**	**77**	**38443**

Los casos de pruebas corridos para el proceso de indexación permiten constatar que a mayor tamaño que tengan los Warc's estos aumentan su tiempo de procesamiento y extracción de la información para su posterior indexación. Todo esto se encuentra estrechamente ligado a la cantidad de páginas que compongan al archivo Warc.

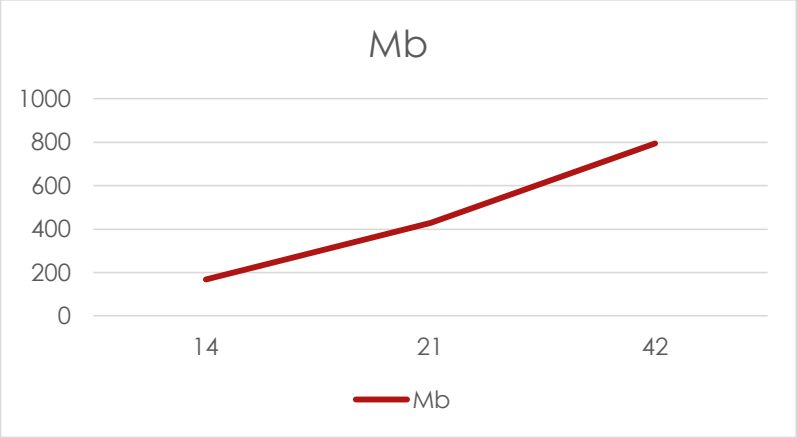

Figura 59 - Total en Indexación en casos por peso
Fuente: Elaboración Propia

4.4.2.2. Casos de limpieza por páginas

Una vez realizada las pruebas pertinentes, se muestra el resultado de rendimiento de las misma. Cabe destacar para estos resultados se cuenta con un universo de 1 contenedor ***"WEB-20150717222207572-00000-2854~Heritrix1~8443.warc.gz"*** el cual almacena la página web ***"www.inzit.go.ve"*** y posee la cantidad de páginas de **502**. En cuanto al tiempo de indexación del contenedor en la **Figura 60**, se muestra cuantos milisegundos tardó en llevarse a cabo el proceso de indexación específicamente la etapa de análisis de extracción del contenido por página.

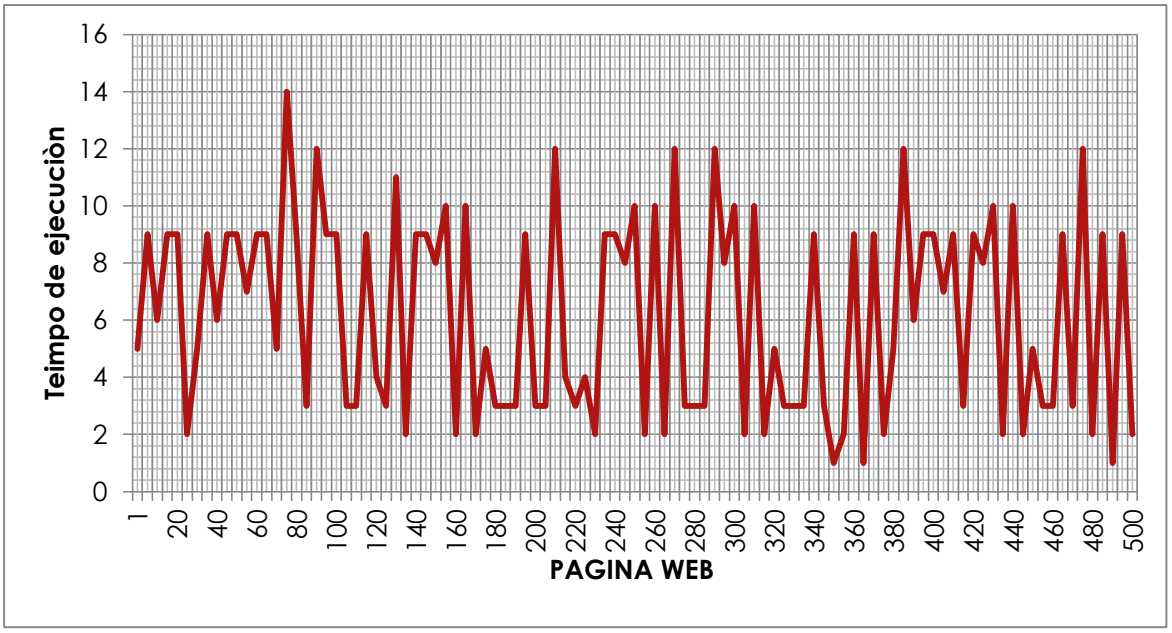

Figura 60 - Tiempo de Indexación de las páginas de un WARC en la etapa de análisis
Fuente: Elaboración Propia

Se puede observar por medio de este caso de prueba, que la duración del análisis del contenido de las paginas va a variar entre una página y otra, ya que está ligado al contenido que tiene cada página de forma individual, se percibe que no es un comportamiento lineal, es decir, que no todas las paginas se tardan la misma cantidad de tiempo en el proceso de limpieza del documento HTML.

4.4.3. Análisis de tiempos de búsqueda

Para realizar el análisis del proceso de búsqueda se realizaron pruebas con una sola y dos palabras claves en los contenidos WARCS, teniendo en cuenta que las frecuencias de dichas palabras son obtenidas por la temática del sitio web indexado, para ejecutar estos casos de prueba se contó con el mismo ambiente de desarrollo que se describió en la sección anterior en tiempo de indexación, en estas pruebas se usara como unidad de medida Milisegundos que trae por defecto Solr. En la **Figura 61** se puede observar la cantidad total de documentos indexados.

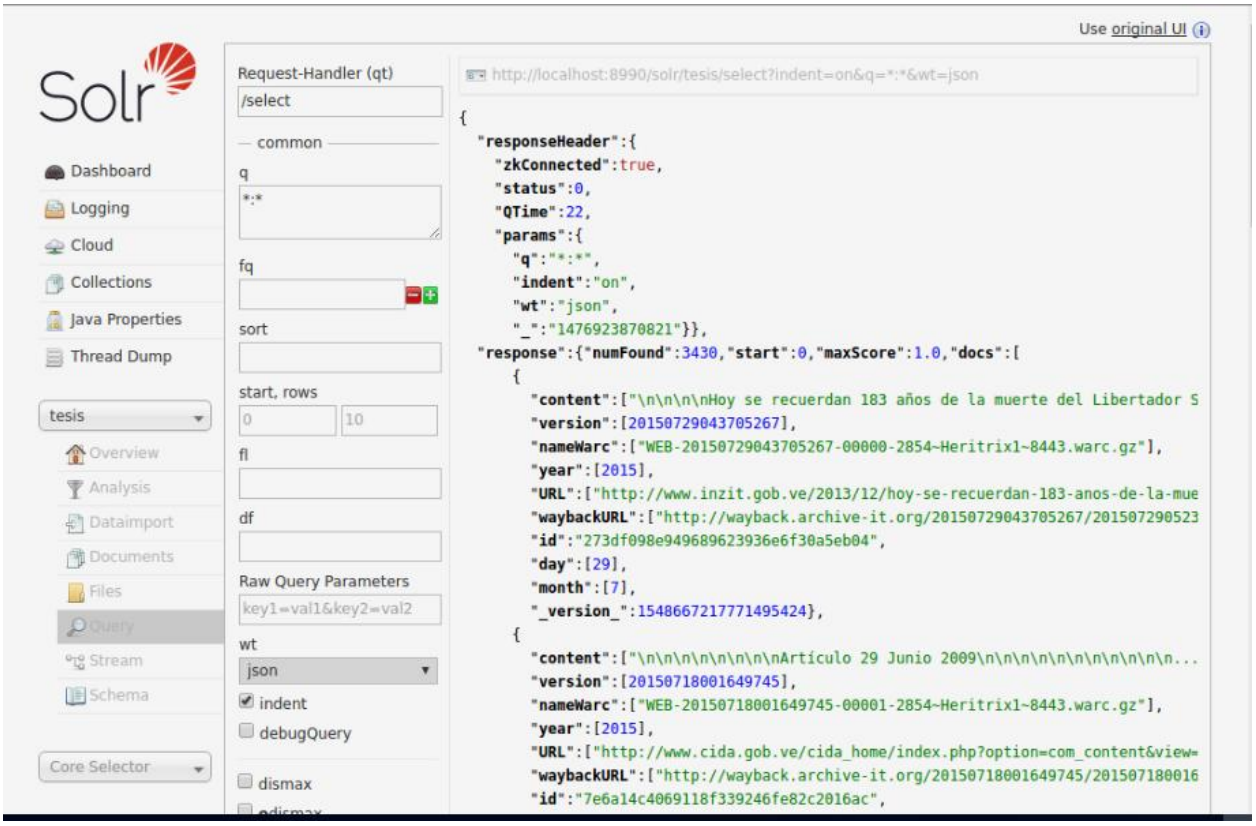

Figura 61- Total, de documentos indexados en la colección
Fuente: Elaboración propia

4.4.3.1. Caso con una palabra clave

Tabla 11 - Resultado final de los datos casos por una palabra clave

Palabra	Número de documentos encontrados	Tiempo (Milisegundos)
Venezuela	1	140
Chavez	1	140
Gobierno	76	237
Chávez	209	263
Presidente	244	203
Eduacación	265	916
Eduacacion	9	284
Zuliano	541	244
Colección Completa en Solr	3430	22

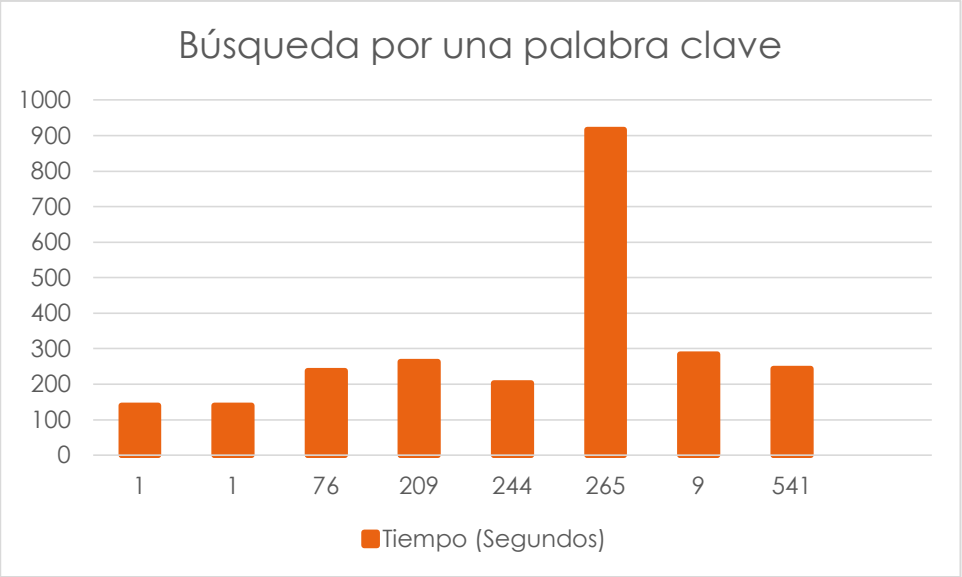

Figura 62 - Tiempo de búsqueda una palabra clave
Fuente: Elaboración Propia

4.4.3.2. Caso con dos palabras clave

Tabla 12 Resultado final de los datos casos por dos palabras claves

Palabra	Número de documentos encontrados	Tiempo (Milisegundos)
Articulo noviembre	323	121
presidente hugo	160	145
zuliano venezuela	2608	180
Zuliano Chávez	209	126
investigaciones venezolanas	58	230
caracas zulia	566	51
Simón Bolivar	879	68
libertador simon	28	157
Colección Completa en Solr	**3430**	**22**

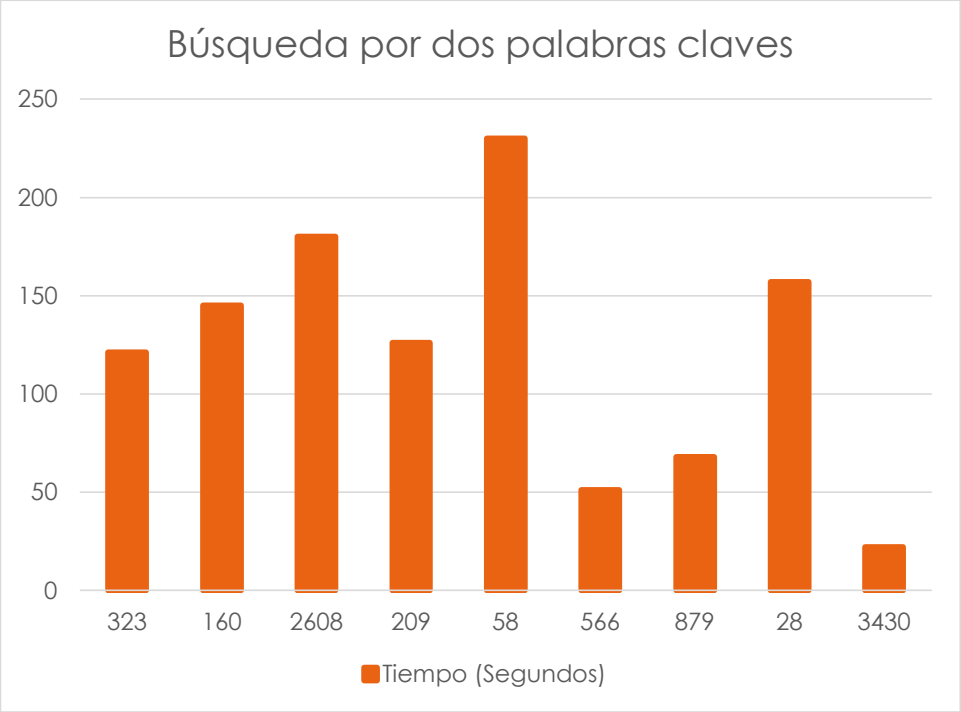

Figura 63- Tiempo de búsqueda por dos palabras claves
Fuente: Elaboración Propia

Se puede observar que el tiempo de respuesta de una búsqueda que se realice tanto por una sola palabra clave como por dos palabras claves, va a depender de la cantidad de veces que la palabra aparezca en los documentos previamente indexados, así como también dependerá de la cantidad de procesamiento libre que posea la máquina. Se puede observar también el buen rendimiento que tiene SolrCloud al momento de realizar la búsqueda, ya que a pesar de que se tiene un volumen de datos considerable como se pudo observar en la Figura 61 y los índices se encuentran distribuidos, los resultados de las consultas se obtuvieron de forma aceptable.

CONCLUSIONES Y RECOMENDACIONES

Al culminar la realización de este trabajo especial de grado denominado "Implementación del módulo de indexación y búsqueda para el prototipo de Archivo Web Venezuela para la búsqueda de los contenidos Web bajo el formato WARC." se estará contando con un prototipo funcional y operativo para los usuarios de Archivo Web Venezuela, para realizar búsqueda por palabras claves dentro del contenido almacenado en los WARC que se generan en el rastreo y almacenamiento de sitios Web en Venezuela.

La implementación fue desarrollada en una arquitectura basada en componentes tal como se pudo observar en la etapa de elaboración de desarrollo, el prototipo está basado en el modelo que se muestra en la Figura 39 , el mismo exhibe como está conformado Archivo Web Venezuela y como será implementado por medio de la plataforma de Búsqueda Solr Cloud el nuevo módulo de indexación y búsqueda, el cual va a permitir búsqueda por palabras claves en el contenido y búsqueda por url, la búsqueda facetadas no está dentro del alcance de este trabajo y se considera en trabajos futuros que se integrarán a este prototipo.

La principal ventaja que tiene los índices de Solr sobre los índices de las bases de datos relacionales, radica en el modelo de datos. La característica que define a las bases de datos relacionales es su modelo de datos basado en tablas y la posibilidad de relacionarlas entre sí mediante llaves. De este modo, para una consulta que involucre varías tablas, es necesario el uso de la operación JOIN para unir las tablas implicadas, aunque dicha operación puede ser muy lenta si las tablas son muy grandes. En contraste, Solr utiliza un modelo de datos orientado a documentos que pueden pensarse como una base de datos de una sola tabla, por lo que una operación análoga a JOIN es inexistente. Un documento en Solr es simplemente un conjunto de campos, como una túpla en una tabla de una base de datos, con la diferencia de que cada columna puede ser multi valuada. Otra diferencia importante entre Solr y una base de datos relacional es que en Solr no existe la operación equivalente a la operación UPDATE. Si cierta parte de un documento necesita ser actualizada, el documento debe ser eliminado y agregar en su lugar el documento actualizado.

La nueva versión de Solr que contempla el uso de Solr Cloud permite reducir la carga de los servidores, es decir las costosas operaciones de indexación ahora se realizarán en un servidor a parte por lo que los servidores dispondrán de mayor capacidad para servir peticiones. Nos otorga mayor escalabilidad, se puede dar mayor potencia a la herramienta de indexación y búsqueda cuando se necesite, sin tener que aumentar el número de servidores, nos da mayor fiabilidad en los clústers, ya que al tener unos índices compartidos por las diferentes instancias de Solr se reduce el riesgo de sufrir problemas de sincronización y de corrupción de índices.

Por esta razón para el desarrollo de esta funcionalidad se implementó un proceso de indexación el cual se realiza través de la plataforma de SolrCloud, manipulando los documentos almacenados dentro del Archivo Web para poder consultarlos y extraer su informaciónón. Además, la aplicación procesa los distintos archivos dentro de los componentes y tiene la funcionalidad de buscar información en contenedores de rastreo ubicados en el módulo de almacenamiento.

Esta solución para el módulo de indexación y búsqueda ofrece a los usuarios del Prototipo una interfaz que los ayudará a realizar las búsquedas por url y por palabra clave en el contenido de los archivos HTML , así como también facilitar la visualización del contenido buscado por medio de una tabla donde se podrá observar, la URL, nombre del Warc's y la versión del mismo, si la búsqueda realizada arroja más de 10 registro se activa la paginación, esto con la finalidad de hacer la interfaz visualmente más cómoda y usable para el usuario. De igual forma se cuenta con la interfaz de administración que provee SolrCloud, donde se podrá observar los nodos creados, ejecutar y ver los resultados de los querys de las búsquedas realizadas.

Durante el desarrollo de la funcionalidad, se presentaron algunas limitantes al momento de construir la solución debido a que no se contaba con la experiencia necesaria para trabajar con los archivos Warc's y se tenía poca documentación sobre la manipulación de los mismos, también se dificulto el proceso de análisis para obtener el contenido de los Warc's y como sería el proceso de limpieza de la información en los archivos HTML. Esto ocasionó que, al inicio, tuviéramos una curva de aprendizaje un poco elevada, pero cuando se comprendió los archivos Warc's y gracias a las distintas herramientas existentes que trabajan con este formato se pudieron realizar las primeras extracciones del contenido de las paginas, lo cual permitió que el desarrollo de la solución fuese más rápido.

Con las pruebas que se realizaron al prototipo se pudo comprobar que sus procesos funcionan de forma correcta, permitiendo abrir los archivos Warc's, extraer sus contenidos, limpiar el contenido de los archivos HTML, para así poder realizar la indexación y posteriormente realizar las búsquedas y aunque las pruebas salieron exitosas siempre hay que tener en cuenta que todo software puede ser mejorado y que difícilmente exista un software infalible.

Adicionalmente, se pudo constatar las mejoras en los tiempos de respuesta en el proceso de indexación y búsqueda con el tratado de un escaso volumen de archivos Warc's.

En la aplicación actualmente no se pueden realizar búsqueda facetadas, por lo que se propone como trabajo futuro, seleccionar nuevas tecnologías o seguir usando SolrCloud para construir este tipo de búsqueda, así mismo se incita a mejorar la limpieza del contenido de los archivos HTML que contiene los archivos Warc's, con la finalidad de indexar la información relevante de la página, para que de esta manera poder hacer una clasificación del contenido de las misma y poder tener un tipo de búsqueda facetada por contenido , por último se sugiere realizar un análisis de los script y hojas de estilos que tengan las paginas almacenadas para su posterior indexación que contienen dichos archivos.

También otro trabajo futuro que recomendamos seria agregar la funcionalidad de realizar el despliegue del contenido del WARC dado la búsqueda que se realice.

BIBLIOGRAFÍA

Apache Solr. (octubre de 2011.). Obtenido de http://lucene.apache.org/solr/

Blázquez Ochando, M. (2012 йил 8-11). *El proceso de indexación.* From http://ccdoc-tecnicasrecuperacioninformacion.blogspot.com/2012/11/el-proceso-de-indexacion.html

Blázquez Ochando, M. (8 de Noviembre de 2012). *webcache*. Obtenido de Técnicas avanzadas de recuperación de información: http://webcache.googleusercontent.com/search?q=cache:HHAp_oRRcKkJ:ccdoc-tecnicasrecuperacioninformacion.blogspot.com/2012/11/el-proceso-de-indexacion.html+&cd=1&hl=es&ct=clnk&gl=ve

Burner , M., & Kahle, B. (1996). *Internet Archive*. Obtenido de http://archive.org/web/researcher/ArcFileFormat.php

Cardoso M, L. I. (2016). *Sistemas de Bases de Datos II.* Caracas: Universidad Catolica Andres Bello.

Espiñeira, Sheldon y Asociados. (2008). *PWC.* Obtenido de www.pwc.com/

Estaba Fernández-Trujillo., J., & Ciancia Biondo, V. (2015). *Módulo de Almacenamiento y Gestión de Datos para Preservación de Archivos Web Usando Bases de Datos NoSQL.* caracas.

Gomes, D. M. (2013). *M.A survey on web archiving initiatives,.* Foundation for National Scientific Computing (FCCN).

Hoydahl, J. *Scaling search with SolrCloud.*

IIPC. (2016). Obtenido de http://netpreserve.org/

IIPC Framework Working Group. (28 de Marzo de 2008). Obtenido de http://archive-access.sourceforge.net/warc/warc_file_format-0.9.html

IIPC. (2012). *Netpreserve*. Obtenido de Netpreserve: http://netpreserve.org/

ISO 28500. (2006). *Information and documentation — The WARC File Format.*

ISO 28500. (2009). *ISO.* Obtenido de https://www.iso.org/obp/ui/#iso:std:iso:28500:ed-1:v1:en

Jackson, A. (14 de julio de 2015). Obtenido de Introduction to Apache Solr: http://www.slideshare.net/andrewnjackson/iipc-trainingeventjan2014solrintroduction

Kabchi, M., & Martínez, M. (2013). *Definición de las estrategias para el desarrollo del Módulo de Acceso a los Contenidos Web Preservados en formato WARC para el Prototipo de Archivo Web para la Preservación del Patrimonio Web de Venezuela.* Caracas.

Kahle, B., & Burner, M. (1996). *Internet Archive.* Obtenido de http://archive.org/Web/researcher/ArcFileFormat.php

Karambelkar, H. (2013). *Scaling Big Data with Hadoop and Solr.* Packt Publishing.

LOC. (2015). Obtenido de https://www.loc.gov/websites/

Masanès, J. (2006). *Web Archive.* New York: Springer.

National Information Standards Organization. (2004). *Understanding Metadata.* Bethesda, Maryland, United States of America: NISO Press.

Ospina Torres, M. H. (2014). *Un Marco de Referencia para la Implementación de Archivos Web.* Universidad Central de Venezuela. Caracas: .

Ospina Torres, M. M., & León Luna, C. P. (2013). Una arquitectura basada en software libre para archivos web. *Enlace* , 53-72.

Ospina, M. (Septiembre de 2011). Modelo de Archivo de la Web para Venezuela. Caracas, Distrito Capital, Venezuela.

Ospina, M., Martinez, M., Kabchi, M., & León, C. (2014). *Desarrollo de una Aplicación para Acceder a Contenidos de un Archivo Web en Formato WARC.* Caracas.

P. Hunt, F. P. (2010). *Zookeeper: Wait-free coordination for internet-scale systems,.* distributed systems.

Padicat. (2015). Obtenido de www.padicat.cat

Potter, T. (2014). *Introduction to SolrCloud ApacheCon.*

Proyecto mundiales que trabajan en la Preservacion Web. (2016). Obtenido de https://www.google.com/maps/d/u/0/viewer?hl=en_US&mid=1bl4aWdl3c51-iG4DZePrQNMfYNs

Rivero, L., & García, J. (2013). *Implementación de los móulos de adquisición y almacenamiento de un prototipo para el archivado de sitios Web en Venezuela.* Caracas.

Sain Technology Solutions. (7 de Abril de 2015). *Introduction to Apache ZooKeeper*. Obtenido de http://www.allprogrammingtutorials.com/tutorials/introduction-to-apache-zookeeper.php

Sommerville, I.

Sommerville, I. (2005). *Ingenieria del Software.* PEARSON.

Szyperski, C. (2002). *Component Software: Beyond Object-Oriented Programming.* Boston: Addison-Wesley Longman Publishing Co.

T. A. S. Foundation. (2010). *Apache zookeeper.* Obtenido de http://zookeeper.apache.org

The Apache Software Foundation. (2014). *Apache Hadoop.* Obtenido de http://hadoop.apache.org/

UNESCO. (2003). *Directrices para la preservación del patrimonio digital.* Preparado por la Biblioteca Nacional de Australia.