Búsquedas optimizadas en la página web de la ESPOL

Jorge Herrera Medina [(1)], Carlos Rodríguez Rivera [(2)], Vanesa Cedeño Mieles [(3)]
Facultad de Ingeniería en Electricidad y Computación (FIEC)
Escuela Superior Politécnica del Litoral (ESPOL)
Campus Gustavo Galindo, Km 30.5 vía Perimetral
Apartado 09-01-5863. Guayaquil, Ecuador
rherrera@espol.edu.ec [(1)], alrodrig @fiec.espol.edu.ec[(2)]
Escuela Superior Politécnica del Litoral (ESPOL) [(3)], Master In Computer Science [(3)], vcedeno@fiec.espol.edu.ec [(3)]

Resumen

El presente documento ilustra el análisis y la implementación de una aplicación que se enfoca en realizar búsquedas optimizadas en la página de la ESPOL. Actualmente el sitio web de la universidad no cuenta con un proceso de búsqueda propio que permita obtener resultados de contenidos dentro del sitio de la ESPOL. Por esto se desea desarrollar un módulo que permita realizar búsquedas en los diferentes contenidos que están publicados en el sitio web de la universidad, así como también realizar búsquedas de artículos en los diferentes sitios web de institutos y facultades asociados a la ESPOL.

Para lograr este objetivo se usará Hadoop como herramienta de procesamiento masivo y escalable de datos para analizar e indexar información de la Web de la ESPOL, también entregar al usuario resultados de búsquedas útiles y más relevantes. También compararemos los tiempos de respuesta de las búsquedas realizadas con Hadoop y con el buscador actual que contiene el sitio de la ESPOL.

Palabras Clave: *Hadoop, MapReduce, Lynx, Búsquedas, Optimización.*

Abstract

This paper illustrates the analysis and implementation of an application that focuses on optimized search in the ESPOL webpage. Currently the website does not have a search process of its own to obtain results contained within the site ESPOL. For this reason a module will be developed that allows to search different content that is published on the website of the university, as well as for items in institutes and faculties websites associated with ESPOL.

To achieve this goal, Hadoop will be used as a tool of massive and scalable processing of data, to analyze and index information from the Web ESPOL to deliver more useful and relevant search results to the user. We will also compare the response times of the searches made with Hadoop to the current search engine in the ESPOL website.

Key Words: *Hadoop, MapReduce, Lynx, Searches, Optimization.*

1. Introducción

La importancia de este proyecto es garantizar una búsqueda relevante en base a los contenidos de los diferentes sitios, artículos, documentos, proyectos, etc. que brinda la ESPOL dentro de su sitio en internet.

También se va a optimizar el tiempo de búsqueda y visualizar resultados con la utilización del comando GREP dentro de la opción búsqueda en el sitio web de la ESPOL.

2. Generalidades

2.1. Descripción del problema

La opción de búsqueda de la página de la ESPOL a pesar de que realiza su función gracias al API de Google, no optimiza sus resultados en las preferencias, ni la presenta organizada por algún parámetro implícito como fecha u orden alfabético de las páginas coincidentes con la búsqueda.

2.2 Objetivos

- Implementar una opción de búsqueda de calidad con los contenidos de la página de la ESPOL usando Hadoop como plataforma de procesamiento masivo y escalable de datos.
- Optimizar el tiempo de búsqueda con la utilización de los nodos configurados previamente en el cluster de hadoop.
- Comparar los tiempos de respuesta de las búsquedas realizadas con Hadoop y con el buscador actual que contiene el sitio de la ESPOL.
- Realizar recomendaciones y sugerencias para futuras modificaciones dentro de la plataforma de búsqueda.

2.3 Alcance

- Para la realización de este proyecto nos limitaremos a usar ciertas páginas de la ESPOL: ICM, FIEC ICF.
- Para obtener la información de la página web utilizaremos las propiedades del comando lynx –dump el cual devuelve el contenido e hipervínculos en texto plano, aunque es limitado dado que no permite a su contenido si la página cuenta con sesiones o Id en su cabecera o dirección URL.
- El diseño de la interfaz web desarrollada para el proyecto está compuesta de forma sencilla con un botón de búsqueda y la opción donde se ingresa el contenido del texto

3 Fundamentos Teóricos Hadoop.

3.1 Hadoop

Hadoop consiste básicamente en el Hadoop Common, que proporciona acceso a los sistemas de archivos soportados por Hadoop. El paquete de software The Hadoop Common contiene los archivos .jar y los scripts necesarios para hacer correr el software de hadoop. El paquete también proporciona código fuente, documentación, y una sección de contribución que incluye proyectos de la Comunidad Hadoop.

Una funcionalidad clave es que para la programación efectiva de trabajo, cada sistema de archivos debe conocer y proporcionar su ubicación: el nombre del rack (más precisamente, del switch) donde está el nodo trabajador. 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 tráfico de red troncal.

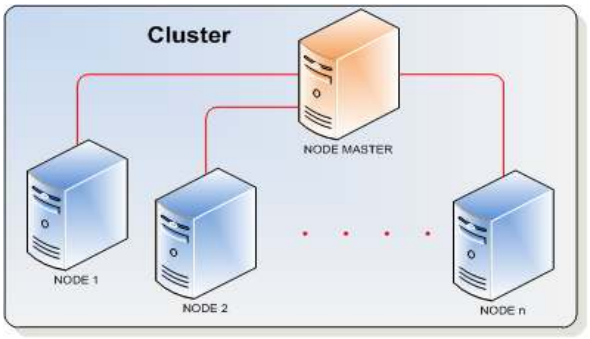

Figura 1 – Esquema Hadoop – Computación distribuida**.**

3.2 HDFS

El sistema de archivos HDFS (*Hadoop Distributed File System*) usa esto cuando replica datos, para intentar conservar copias diferentes de los datos en racks diferentes. El objetivo es reducir el impacto de un corte de energía de rack o de fallo de interruptor de modo que incluso si se producen estos eventos, los datos todavía puedan ser legibles y procesados.

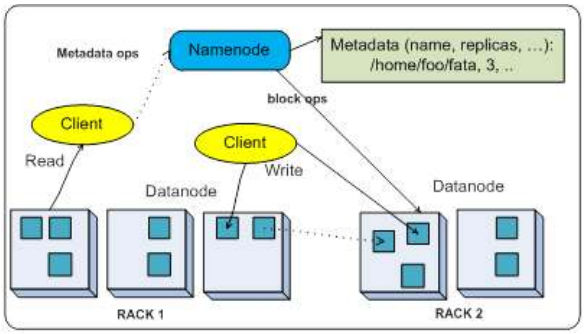

Figura 2 – Arquitectura HDFS**.**

3.3 MAP/REDUCE - Job Tracker y Task Tracker

Un clúster típico Hadoop (Map – Reduce Framework) incluye un nodo maestro y múltiples nodos esclavo. El nodo maestro consiste en jobtracker (rastreador de trabajo), tasktracker (rastreador de tareas), namenode (nodo de nombres), y datanode (nodo de datos).

Un esclavo o compute node (nodo de cómputo) consisten en un nodo de datos y un rastreador de tareas.

Hadoop requiere tener instalados entre nodos en el clúster JRE 1.6 o superior, y SSH.

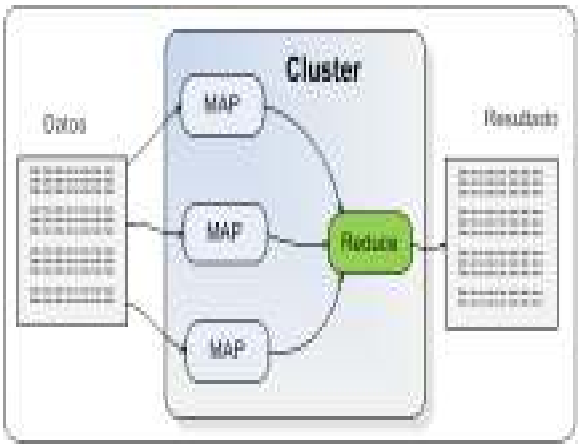

Figura 3 – Proceso MAP/REDUCE**.**

Map/reduce es la parte que realiza los cálculos y transformaciones sobre los datos. Básicamente se trata de una serie de componentes software que ejecutan un programa realizado en Java que sigue el modelo de programación del mismo nombre (map/reduce). Esto es un esquema de programación paralela que tiene sus orígenes en la programación funcional.

3.3.1 Función Map()

Esta función trabaja sobre grandes cantidades de datos los cuales son divididos en más de dos partes que contienen datos o registros:

dato1 --> Map() --> dato '1
dato 2 --> Map() --> dato '2

...

dato x --> Map() --> dato 'x

3.3.2 Función Reduce()

Esta función sabe interpretar y unir los datos que fueron realizados con la función Map:

[dato '1, dato '2, ... , dato 'x] --> Reduce() --> Resultado.

3.4 GREP

Este comando es uno de los más útiles a nivel de Linux el cual nos ahorra mucho tiempo para realizar búsquedas de archivos, documentos, palabras etc. con hadoop. ¿Qué significa Grep? ***g/re/p*** significa hacer una búsqueda **g**lobal para las líneas que encajen con la expresión **re**gular, e im**p**rimirlas.

¿Pero qué hace el comando grep? Busca determinada palabra o frase entre los archivos de texto. Si el término buscado aparece varias veces en un mismo archivo, nos muestra varias líneas de resultado, una por cada coincidencia. Por ejemplo:

Sintaxis: **grep** [opciones] [expresión regular] [archivo]

grep -r curso /home/hadoop/Documentos/*

Con ese comando, buscamos la palabra curso en cualquier fichero del directorio Documentos. Esto incluye las carpetas que existan dentro de Documentos (hemos indicado esto al escribir -r).

Si deseamos buscar en un fichero concreto, sustituimos * por el nombre del fichero. Hay un detalle importante, el comando anterior diferencia entre mayúsculas y minúsculas.

grep -ir curso /home/hadoop/Documentos/*

grep -i "curso linux" /home/hadoop/Documentos/notas.txt

Al incluir -i no hace distinción entre mayúsculas o minúsculas.

3.4.1 Sus opciones son:

-c: Modificar la salida normal del programa, en lugar de imprimir por salida estándar las líneas coincidentes, imprime la cantidad de líneas que coincidieron en cada archivo.

-e PATRÓN: Usar PATRÓN como el patrón de búsqueda, muy útil para proteger aquellos patrones de búsqueda que comienzan con el signo «-».

-f ARCHIVO: Obtiene los patrones del archivo ARCHIVO

-H: Imprimir el nombre del archivo con cada coincidencia.

-r: Buscar recursivamente dentro de todos los subdirectorios del directorio actual.

4 Análisis de la Solución.

4.1 Requerimientos.

Tomando en consideración que la solución propuesta se centra en la construcción de un motor de búsqueda

utilizando el contenedor de datos de hadoop, a continuación se detalla los requerimientos principales para el funcionamiento del proyecto.

- **Centos Linux**

La versión de la distribución Centos de Linux que se utilizó en la elaboración del proyecto fue 5.6.

- **Virtual Box**

Para las pruebas de configuración e instalación de Centos en un ambiente virtual se utilizó esta herramienta, en la cual se instaló la distribución de Centos de Linux versión 5.5..

- **Java**

Para poder utilizar hadoop dentro del ambiente Linux debe tener instalado la versión o una superior.

- **Hadoop**

La versión de hadoop que se utilizó en las pruebas del proyecto es hadoop-0.20.203.0.

- **JDK**

La versión del JDK es la que tiene por defecto la plataforma de distribución de Centos Linux.

- **NetBeans**

Para la programación de las clases en java se utilizó la versión NetBeans IDE 6.9.1.

- **Lynx**

Con este comando nos permite obtener la información y contenido de cada página web, esta herramienta también es de gran utilidad ya que nos permite capturar la información en texto plano.

- **Grep**

Con el comando grep podemos realizar las búsquedas necesarias dentro de los archivos planos que se obtienen con el comando Lynx.

- **SSH**

Con las propiedades de SSH nos permite tener un cluster de nodos entre master y esclavo, este paquete se debe configurar en cada nodo para que no nos pida las credenciales a la hora de realizar una conexión con uno de sus nodos directamente desde el master. La versión utilizada es la 4.3p2

A continuación se detalla el procedimiento para analizar los datos de la búsqueda dentro del sitio de ESPOL:

4.2 Requerimientos Funcionales.

- Obtener datos de las páginas donde se desea realizar la búsqueda mediante el comando Lynx y almacenarlas en archivos planos organizados por índices.

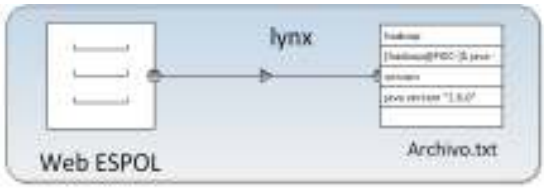

Figura 4 – Esquema del comando Lynx**.**

- Para realizar la búsqueda de forma ordenada en los diferentes nodos se realizara mediante el comando grep que nos permite ordenar y agrupar la gran cantidad de datos, los cuales serán almacenados en un archivo de salida.

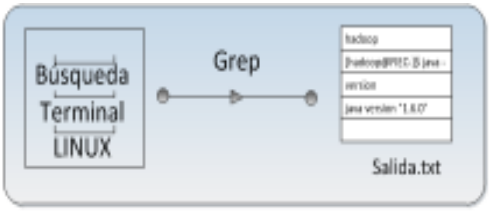

Figura 5– Esquema del comando Grep**.**

- Para el análisis se utiliza la siguiente sintaxis del comando grep: **bin/hadoop jar hadoop-examples-*.jar grep input output 'dfs[a-z.]+'**
Los datos almacenados en el archivo de salida dependen de la palabra o frase que desea buscar.

- Luego de obtener los datos de salida, estos son procesados en una clase de java mediante netbeans que permite ejecutar los comandos desde el terminal de Linux previa la creación de un script donde se registraran los comandos necesarios para la ejecución del grep

- Para mostrar los datos en el browser se procedió con la creación de una página de prueba mediante una aplicación web donde se van a mostrar los resultados de la búsqueda realizada, para lo cual debe estar levantado el tomcat que permite ejecutar servicios webs.

4.3 Requisitos No Funcionales.

Para desarrollar el proyecto con hadoop se requiere que los equipos donde se van a realizar la distribución de información tengan las siguientes características:

- PCs para que trabajen como nodos
- Procesador Dual-Core Intel Xeon 2.0 GHZ
- 8GB de memoria RAM
- Discos SATA de 41TB
- Tarjeta de red Gigabit Ethernet
    - Uplink from rack is 3-4 gigabit
    - Rack-internal is 1 gigabit

5 Diseño e Implementación de la Opción de Búsqueda de la ESPOL.

5.1 Descripción del diseño

El diseño de la solución de la opción de búsqueda de calidad de la información que se encuentra dentro del sitio de la ESPOL es el siguiente:

5.1.1 Obtención de los datos.

Para la obtención de los datos se instalaron los paquetes de la librería LYNX que tiene la distribución Centos de lunix, el cual nos permite obtener la información de cada página web en texto plano y los links de la webs presentes en la página.

La sintaxis para obtener los datos es el siguiente:

lynx -dump [web address] > webpagename.txt

Donde:

[web address]: es la dirección web de la que se desea bajar el contenido de toda la página.

webpagename: es el nombre del archivo que contiene toda la información y links del sitio web, el cual está almacenado en una archivo txt.

La ejecución de este comando va a la dirección web que se necesite y baja toda la información en un archivo txt. Esta información esta almacenada en un directorio llamado DB donde se encuentran todos los sitios para realizar las pruebas de búsqueda. Para nuestro propósito solo se ha realizado la extracción de la información de tres sitios web que están dentro de la página de la ESPOL

Tabla de contenido de la información de cada sitio web:

Nombre Sitio	*Nombre del archivo*	*Numero de palabras*
www.espol.edu.ec	espol.txt	
www.icm.espol.edu.ec	icm.txt	
www.fiec.espol.edu.ec	fiec.txt	

Tabla 1– Sitios Web de Espol

Esta información esta almacenada en un directorio llamado DB donde están guardados todas las páginas con sus respectivos links internos.

5.1.2 Configuración de hadoop.

- Instalación de software de hadoop
- Configuración de las variables de entorno de java con hadoop
- Configuración de SSH en cada uno de los nodos
- Verificar el funcionamiento del comando grep de linux

Iniciar el levantamiento del demonio de hadoop
$ bin/start-all.sh

Una vez configurado se puede observar si los nodos están levantados a través de la interface web para el NameNode y para el JobTracker
NameNode - http://localhost:50070/

JobTracker - http://localhost:50030/

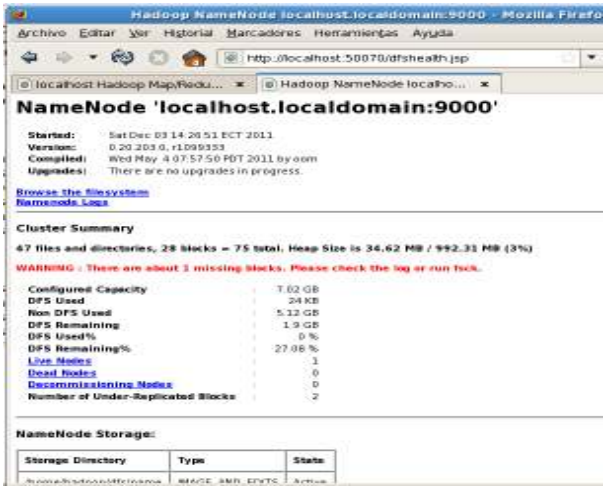

Figura 6 – Gráfico de ejecución de nodos

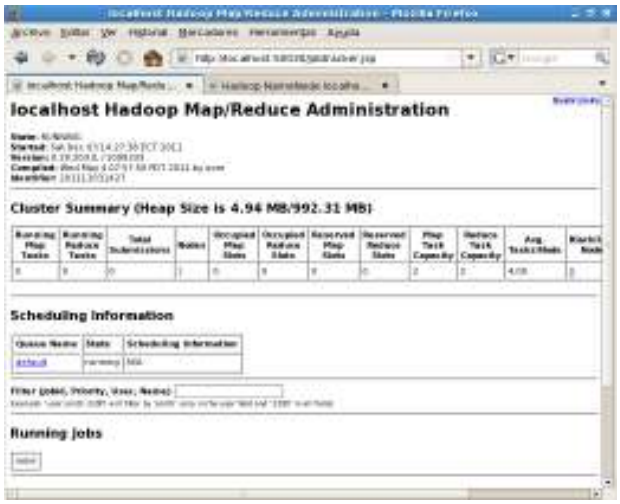

Figura 7 – Gráfico de ejecución de MapReduce

Luego copiar los archivos de entrada en filesystem
$ bin/hadoop fs -put conf input

Ejecutar nuevamente la búsqueda mediante el grep de hadoop
$ bin/hadoop jar hadoop-examples-*.jar grep input salida 'palabra a buscar'

Verificar que la información este en el directorio de salida y copiar los archivos de salida en el filesystem distribuido al local:
$ bin/hadoop fs -get salida output
$ cat output/*

$ bin/hadoop fs -cat output/*

Luego se debe poner en stop el demonio de hadoop
$ bin/stop-all.sh

Luego re realizar las pruebas del funcionamiento de proceder con la creación de las clases en java que van a procesar la información en un la interface web

5.1.3 Clases en java.

En esta parte se realiza la creación de las clases en java gracias a IDE de NetBeans que van a permitir la ejecución de los comandos en los terminales de Linux, para este propósito se van a ejecutar los siguientes procesos:
Código en java:

```
import java.io.*;
public class Main {
      public static void main(String args[]) {
          try {
            Runtime rt = Runtime.getRuntime();
            Process pr =
rt.exec("/home/hadoop/commandos.txt");
            BufferedReader input = new
BufferedReader(new
InputStreamReader(pr.getInputStream()));
            String line=null;
            while((line=input.readLine()) != null) {
               System.out.println(line);
            }
             int exitVal = pr.waitFor();
                              System.out.println("Exited
                    with error code "+exitVal);
                              } catch(Exception e) {

                    System.out.println(e.toString());
                              e.printStackTrace();
                          }
                      }
                    }
```

5.1.4 Creación de la interface web.

Para la creación de la interface web se lo realizó en netBeans ya que esta herramienta cuenta con los recursos necesarios para realizar un servidor web.

La opción de búsqueda que será incorporado en el sitio web tiene un diseño muy sencillo en un formulario donde se va a ingresar las opciones de búsqueda, ya que lo más importante es el proceso que se realiza a nivel interno ya que el sitio web es desarrollado con tecnología JSP que nos permite realizar procesos web asegurándonos legibilidad y seguridad de la información.

El resultado de la búsqueda de la información se muestra de la siguiente forma indicando el tiempo que demoro la búsqueda del contenido, una pequeña descripción de la información buscada así como también el link donde se encuentra el contenido de los datos que se desea consultar.

La información que se muestra información buscada que coincide con los términos que se ingresaron en la búsqueda.

5. Pruebas y Resultados.

5.1. Ejecución de las pruebas

Debido a que la búsqueda en el sitio web de ESPOL no muestra resultados óptimos a nivel del contenido de los

sitios relacionados en dicha web, se procedió con la implementación de optimización de la búsqueda de los contenidos que estén dentro del sitio web de la ESPOL con la tecnología de hadoop.

Para las pruebas realizadas se instaló hadoop en 4 nodos en una plataforma de distribución de Centos Linux versión 5.6 en el cual se configuró y se instaló las librerías necesarias para la ejecución de las pruebas.
A continuación se detalla el procedimiento de ejecución de las pruebas:

- Captura de los información de los link asociados al sitio web de ESPOL y sus diferentes contenidos.

- Los resultados de la información antes mencionada es guardada en una base de datos, para nuestro caso la base de datos están almacenados en un directorio llamado DB que contiene archivos en formato txt donde están almacenados los índices, contenidos y link asociados a cada uno de ellos.

- Luego se realiza la ejecución del GREP con hadoop desde una clase en java el cual inicia el proceso de map-reduce. Esto permite almacenar la información de búsqueda en un archivo de salida indicando el número de veces y la ubicación de la palabra en sus respectivos archivos.

- Los resultados son mostrados en la aplicación web desarrollada con tecnología JSP indicando el tiempo que tomo realizar la búsqueda y los diferentes contenidos, los archivos que contienen la palabra de búsqueda y sus links respectivos.

5.2. Análisis de los resultados

Las pruebas realizadas muestran que los tiempos de búsqueda en el sitio web de ESPOL son menores a medida que se aumentan los nodos para el procesamiento de la búsqueda.

Gracias al gran poder de procesamiento que tiene Hadoop de trabajar con diferentes nodos, se puede optimizar las búsquedas de los contenidos a gran escala y de forma rápida.

En el siguiente gráfico se observa la relación del tiempo de respuesta de la búsqueda versus el número de nodos utilizados para realizar la búsqueda.

Tiempo Vs Nodos	
Nodos	**Tiempo de búsqueda hadoop**
2	16 segundos
4	11 segundos

Tabla 2. Gráfico comparativo de hadoop (1 observación)

Tiempo Vs Nodos	
Nodos	**Tiempo de búsqueda hadoop**
2	12 segundos
4	10 segundos

Tabla 3. Gráfico comparativo de hadoop (2 observaciones)

Tiempo Vs Nodos	
Nodos	**Tiempo de búsqueda hadoop**
2	12 segundos
4	9 segundos

Tabla 4. Gráfico comparativo de hadoop (3 observaciones)

Tiempo Vs Nodos	
Nodos	**Tiempo de búsqueda google**
Índice web de Google	0.46 segundos

Tabla 5. Gráfico comparativo de google

6. Conclusiones

Hadoop es un framework muy potente y realmente sencillo de utilizar, sin embargo, debemos tener muy claro que se quiere resolver y no intentar resolver todos nuestros problemas con él. Tiene que ser un sistema distribuido con gran cantidad de datos y nodos para procesar dichos datos. Como se aprecio en los tiempos se requiere mayor cantidad de nodos y datos para que hadoop pueda ser utilizado de manera eficiente.

Debido a la gran cantidad de datos, se debe realizar una extracción de la información de los sitios webs asociados al sitio web de la ESPOL gracias a la librería LYNX de Linux el cual nos facilitó la obtención de la información de cada sitio web (nombre.espol.edu.ec), para las pruebas realizadas solo se tomó como referencia tres sitios web como ICM, FIEC y ESPOL.

Una vez obtenidos la gran cantidad de datos, se utilizó la tecnología MapReduce de hadoop, el cual nos permite realizar un análisis escalable de datos en tiempos mínimos a través de los diferentes nodos del cluster y administrados por el nodo master.

La utilización de la librería GREP de Linux y hadoop nos permite realizar la búsqueda del contenido dentro del directorio de información almacenada en los archivos extraídos de los diferentes sitios.

Para mostrar los datos de las búsquedas se implementó una aplicación web que nos permite observar los resultados que los usuarios han realizado en la búsqueda dentro del contenido de los sitios de ESPOL, esta aplicación fue desarrollada gracias a la tecnología de NetBeans y las librerías de hadoop.

7. Referencias Bibliográficas

[1] **Admin of ihaveapc**, How to Save Web Pages as Text Files through Linux Mint / Ubuntu Terminal, http://www.ihaveapc.com/2011/03/how-to-save-web-pages-as-text-files-through-linux-mint-ubuntu-terminal/

[2] **Yahoo**, Module 3: Getting Started With Hadoop http://developer.yahoo.com/hadoop/tutorial/module3.html.

[3] **Faud NAHDI**, How to install JDK, http://www.techonia.com/install-jdk-java-linux .

[4] **VIVEK GITE**, HowTo: Use grep Command In Linux / UNIX, http://www.cyberciti.biz/faq/howto-use-grep-command-in-linux-unix/.

[5] **Arif N., Programming Hadoop in Netbeans,** http://arifn.web.id/blog/2010/01/23/hadoop-in-netbeans.html.

[6] **SKYMYRKA**, Installing Java (Sun JDK 1.6.0) on CentOS 5, http://de0ris.blogspot.com/2008/08/installing-java-sun-jdk-160-on-centos5.html.

[7] **JR**, Install NetBeans IDE 6.9.1, http://www.if-not-true-then-false.com/2010/install-netbeans-6-9-on-fedora-centos-red-hat-rhel/.

[8] **Luciano,** Varios ejemplos de uso del comando grep, http://luauf.com/2009/05/04/varios-ejemplos-de-uso-del-comando-grep/.

[9] **Brandeis University**, Hadoop Example Program, http://pages.cs.brandeis.edu/~cs147a/lab/hadoop-example/.

[10] **Leons Petražickis**, Hadoop Lab 1 - HDFS Instructions, http://www.slideshare.net/leonsp/hadoop-lab-1-hdfs-instructions.

[11] **linglom**, How to run command-line or execute external application from Jav http://www.linglom.com/20 6/how-to-run-command-line-or-execut ıal-application-from-java/.