



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**Facultad de Ingeniería en Electricidad y Computación**

“Búsquedas optimizadas en la página de la ESPOL”

**TESINA DE SEMINARIO**

Previo la obtención de los Títulos de:

**INGENIERO EN CIENCIAS COMPUTACIONALES**  
**ESPECIALIZACIÓN SISTEMAS MULTIMEDIA**  
**INGENIERO EN CIENCIAS COMPUTACIONALES**  
**ESPECIALIZACIÓN SISTEMAS DE INFORMACIÓN**

Presentada por:

**Jorge Rafael Herrera Medina**

**Carlos Alberto Rodríguez Rivera**

**GUAYAQUIL – ECUADOR**  
**2012**

# DEDICATORIA

A mi familia quienes siempre me apoyaron en todo para poder poner alas a mis sueños.

*Jorge Herrera Medina*

Este trabajo lo dedico con mucho cariño a mis padres quienes me han apoyado durante toda mi carrera estudiantil. A mis hermanos y todos mis amigos quienes han sido fuente de motivación para culminar mis estudios con éxito.

*Carlos Rodríguez Rivera*

# AGRADECIMIENTO

Agradecemos principalmente a Dios quien ha sido el pilar principal para terminar nuestros estudios.

A nuestros padres por ser el apoyo incondicional, a nuestra familia, nuestros amigos quienes nos han ayudado durante toda la carrera universitaria.

Un agradecimiento especial a la Ing. Vanessa Cedeño por su ayuda para culminar con éxito este proyecto.

# DECLARATORIA EXPRESA

“La responsabilidad del contenido de este Trabajo de Graduación, nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la

**Escuela Superior Politécnica del Litoral”**

(Reglamento de Graduación de la ESPOL)

---

JORGE HERRERA MEDINA

---

CARLOS RODRÍGUEZ RIVERA

# TRIBUNAL DE SUSTENTACIÓN

---

Ing. Vanessa Cedeño.

**PROFESOR DEL SEMINARIO DE GRADUACIÓN**

---

Ing. Xavier Ochoa.

**PROFESOR DELEGADO POR EL DECANO**

# RESUMEN

Este proyecto 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, para lo cual 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 las diferentes sitios web asociadas a la ESPOL como ICM, IFC, los diferentes institutos y facultades.

Para lograr este objetivo se usara Hadoop como herramienta de procesamiento masivo y escalable de datos para analizar e indexar información de la Web de la ESPOL, entregar resultados de búsquedas útiles y más relevantes para el usuario.

Llevar un registro de qué resultados fueron escogidos y cuáles ignorados para refinar los resultados mostrados a otros usuarios con consultas similares. Una vez realizado las búsquedas se analizara el tiempo que se llevó en mostrar el resultado y compararlo con los otros buscadores

# INDICE GENERAL

RESUMEN .....	VI
INDICE GENERAL .....	VII
INDICE DE FIGURAS .....	VIII
INDICE DE TABLAS .....	IX
INTRODUCCIÓN .....	X
<b>CAPÍTULO 1 .....</b>	<b>XIII</b>
<b>1. ANTECEDENTES Y JUSTIFICACIÓN .....</b>	<b>XIII</b>
1.1. ANTECEDENTES Y DESCRIPCIÓN DEL PROBLEMA.....	XIII
1.2. JUSTIFICACIÓN .....	XIII
1.3. OBJETIVOS .....	2
1.4. ALCANCE .....	3
<b>CAPÍTULO 2 .....</b>	<b>4</b>
<b>2. FUNDAMENTOS TEÓRICOS HADOOP.....</b>	<b>4</b>
2.1. HADOOP .....	4
2.2. HDFS.....	5
2.3. MAP/REDUCE - JOB TRACKER Y TASK TRACKER .....	6
2.3.1. FUNCIÓN MAP().....	7
2.3.2. FUNCIÓN REDUCE() .....	8
2.4. GREP .....	8
2.4.1. SUS OPCIONES SON: .....	9
<b>CAPÍTULO 3 .....</b>	<b>11</b>
<b>3. ANÁLISIS DE LA SOLUCIÓN .....</b>	<b>11</b>
<b>3.1. REQUERIMIENTOS .....</b>	<b>11</b>
• CENTOS LINUX.....	11
• VIRTUAL BOX .....	112
• JAVA .....	112
• HADOOP .....	13
• JDK.....	13
• NETBEANS .....	13
• LYNX.....	14
• GREP .....	14
• SSH.....	14
<b>3.2. REQUERIMIENTOS FUNCIONALES.....</b>	<b>15</b>

<b>3.3. REQUISITOS NO FUNCIONALES.....</b>	<b>17</b>
<b>CAPÍTULO 4 .....</b>	<b>18</b>
<b>4. DISEÑO E IMPLEMENTACIÓN DE LA OPCION DE BUSQUEDA DE LA ESPOL .....</b>	<b>18</b>
4.1. DESCRIPCIÓN DEL DISEÑO.....	19
4.1.1. OBTENCIÓN DE LOS DATOS.....	19
4.1.2. CONFIGURACIÓN DE HADOOP.....	21
4.1.3. PROCESO STANDALONE.....	22
4.1.4. PROCESO PSEUDO DISTRIBUIDO.....	22
4.1.5. CLASES EN JAVA.....	27
4.1.6. CREACIÓN DE LA INTERFACE WEB.....	28
<b>CAPÍTULO 5 .....</b>	<b>31</b>
<b>5. PRUEBAS Y RESULTADOS.....</b>	<b>31</b>
5.1. EJECUCIÓN DE LAS PRUEBAS .....	31
5.2. ANÁLISIS DE LOS RESULTADOS .....	33
<b>CONCLUSIONES.....</b>	<b>35</b>
<b>RECOMENDACIONES.....</b>	<b>37</b>
<b>BIBLIOGRAFÍA.....</b>	<b>38</b>



# INDICE DE FIGURAS

Figura 2-1 – Esquema Hadoop – Computación distribuida .....	5
Figura 2-2 – Arquitectura HDFS .....	6
Figura 2-3 – Proceso MAP/REDUCE .....	7
Figura 3-1 – Esquema del comando Lynx .....	15
Figura 3-2 – Esquema del comando Grep.....	16
Figura 4-1 – Gráfico del archivo de salida .....	20
Figura 4-2 – Gráfico del archivo core-site .....	22
Figura 4-3 – Gráfico del archivo hdfs-site .....	23
Figura 4-4 – Gráfico del archivo mapred-site .....	23
Figura 4-5 – Gráfico del formateo del filesystem .....	24
Figura 4-6 – Gráfico de ejecución de nodos.....	25
Figura 4-7 – Gráfico de ejecución de MapReduce.....	25
Figura 4-8 – Gráfico de salida del grep .....	26
Figura 4-9 – Gráfico del IDE NetaBeans.....	28
Figura 4-10 – Gráfico de la interface web .....	29
Figura 4-11 – Gráfico de la interface web del proyecto .....	30

# INDICE DE TABLAS

Tabla 4-1 Sitios Web de Espol .....	21
Tabla 5.1. Gráfico comparativo de hadoop .....	33
Tabla 5.2. Gráfico comparativo de google.....	34

# 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.

En el capítulo uno de este trabajo se detalla la descripción del problema y antecedentes sobre el cual se plantean los objetivos, justificación y alcance de la opción de búsqueda que actualmente contiene el sitio web de ESPOL.

En el capítulo dos se presentan los fundamentos teóricos, las herramientas tecnológicas y algoritmos que se utilizaran para el desarrollo del proyecto tales como HADOOP, Map-Reduce, Grep, Lynx.

En el capítulo tres se describe el análisis de la solución y los requerimientos de información necesarios tales como Linux, Virtualización, librerías de java, Netbeans y todo lo necesario para el desarrollo del módulo de búsqueda en el sitio web de la ESPOL.

En el capítulo 4 se muestra el diseño y se detalla los pasos necesarios para la implementación de la solución presentada.

Las pruebas y resultados del análisis de los tiempos de respuestas de las diferentes búsquedas realizadas son mostrados en el capítulo 5.

Y para finalizar, al final de los capítulos detallados se mostraran las recomendaciones y conclusiones propuestas por el grupo de trabajo.

# **CAPÍTULO 1**

## **1. ANTECEDENTES Y JUSTIFICACIÓN.**

### **1.1. Antecedentes y 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.

### **1.2. Justificación**

Al ser implementado el sistema podrá permitir a los usuarios de la página de la ESPOL realizar búsqueda con mayor rapidez (en relación con la cantidad de nodos disponibles) y relevancia, basado en la información previamente almacenada una base de datos, donde

estará la información relacionada al sitio que están en las páginas de la universidad, utilizando Hadoop como plataforma de procesamiento masivo y escalable de datos.

### **1.3. 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.

## 1.4. 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

# CAPÍTULO 2

## 2. FUNDAMENTOS TEÓRICOS HADOOP.

En este capítulo se detallan todos los requerimientos necesarios para el desarrollo e implementación del proyecto propuesto. A continuación detallaremos cada uno de los conceptos principales utilizados:

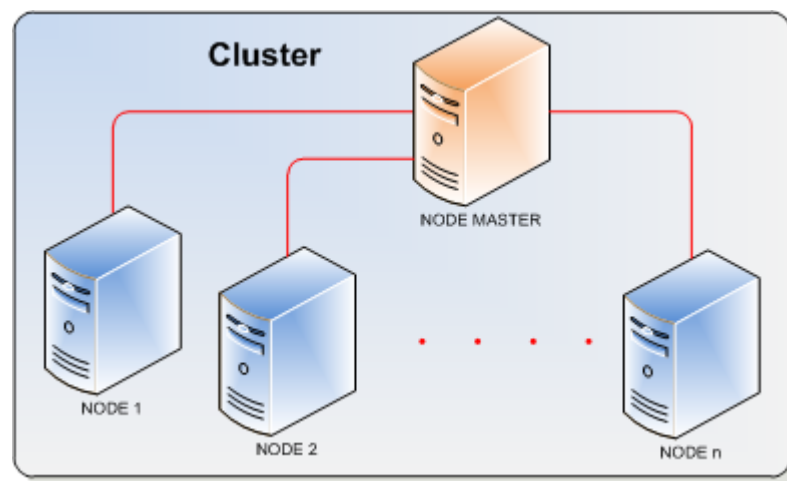
### 2.1. Hadoop

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.

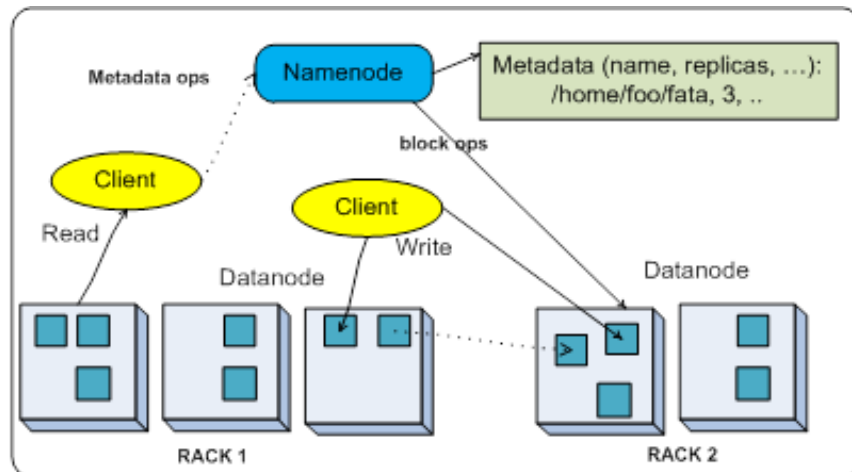


**Figura 2-1** – Esquema Hadoop – Computación distribuida.

## 2.2. 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.

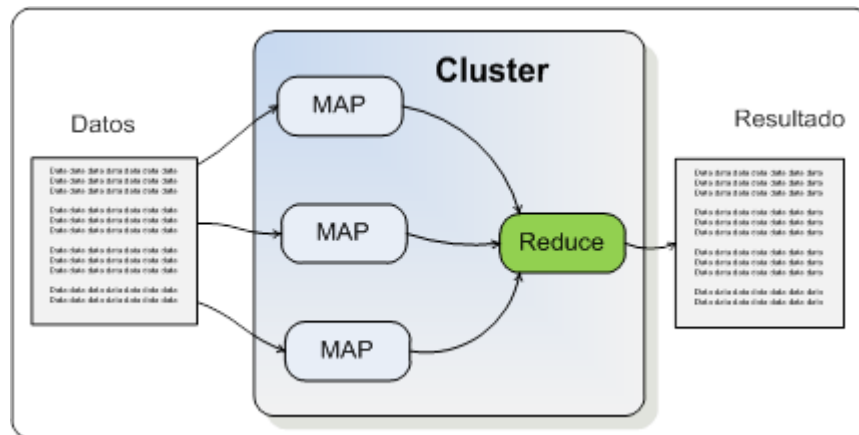


**Figura 2-2 – Arquitectura HDFS.**

### **2.3. 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 2-3 – 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.

### 2.3.1. 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

### 2.3.2. 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.

## 2.4. 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**primirlas [5]

¿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.

### 2.4.1. 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:** Obtene 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.

# CAPÍTULO 3

## 3. ANÁLISIS DE LA SOLUCIÓN.

Este capítulo se detalla el análisis de la solución de búsquedas personalizadas en el sitio web de ESPOL, a partir del cual se realizó el diseño del esquema de la solución del proyecto.

### 3.1. 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 que estaba instalado en los

laboratorios de computación de la Facultad de Eléctrica y Computación donde se realizaron las pruebas de funcionamiento en los diferentes nodos configurados.

- **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. Para la correcta configuración del sistema operativo se realizó la instalación de Centos en modo texto, esto nos permite asignarle el tamaño de los discos virtuales y tener el espacio requerido para la instalación de los requerimientos necesarios para el proyecto.

- **Java**

Para poder utilizar hadoop dentro del ambiente Linux debe tener instalado la versión o una superior, de esta forma garantizamos que todos los procesos que involucren o necesiten de este componentes funciones de forma correcta.

```
-----  
[hadoop@localhost ~]$ java -version  
java version "1.6.0"  
OpenJDK Runtime Environment (build 1.6.0-b09)  
OpenJDK Client VM (build 1.6.0-b09, mixed mode)
```



- **Hadoop**

La versión de hadoop que se utilizó en las pruebas del proyecto es hadoop-0.20.203.0, este componente viene con todos los paquetes, ejemplos y documentación necesaria para la configuración e pruebas del sistema.

- **JDK**

La versión del JDK es la que tiene por defecto la plataforma de distribución de Centos Linux, para las pruebas se necesita este módulo bien configurado ya que es necesario configurar las variables de entorno para que los ejemplos de hadoop funcionen.

- **NetBeans**

Para la programación de las clases en java se utilizó la versión NetBeans IDE 6.9.1. este programa bien con sus propios plugin de hadoop, pero para nuestro proyecto no es necesario utilizarlo ya que nuestra clase ha sido desarrollada para que ejecute directamente los comandos del map-reduce de hadoop

- **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. Gracias a las propiedades del comando grep y las herramientas de hadoop se puede realizar una búsqueda a gran escala y con grandes cantidades de datos.

- **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

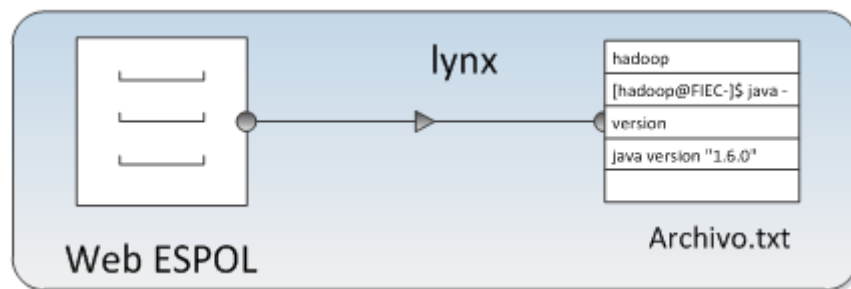
```
[hadoop@localhost ~]$ ssh -version
OpenSSH_4.3p2, OpenSSL 0.9.8e-fips-rhel5 01 Jul 2008
Bad escape character 'rsion'.
[hadoop@localhost ~]$ █
```

---

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

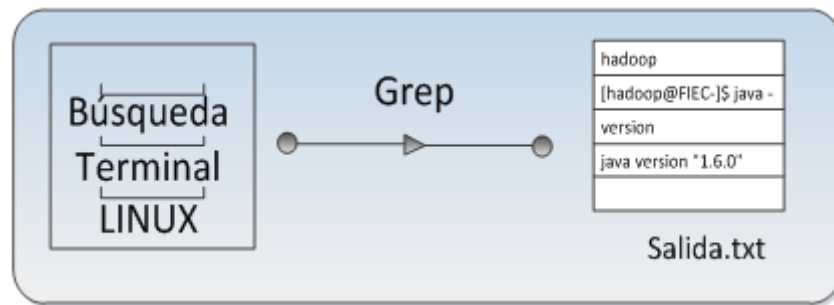
### 3.2. Requerimientos Funcionales.

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



**Figura 3-1** – 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 3-2 – 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.

### **3.3. 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

# CAPÍTULO 4

## 4. DISEÑO E IMPLEMENTACIÓN DE LA OPCION DE BUSQUEDA DE LA ESPOL.

En este capítulo se describe la metodología usada para la implementación de la opción de búsqueda en el sitio web de ESPOL realizado con la tecnología JSP desarrollado mediante NetBeans y hadoop. En el capítulo 5 se detalla el plan de pruebas realizadas.

Para el desarrollo de la interfaz web solo se realizó el diseño de la opción de búsqueda, ya que este es el objetivo principal del proyecto a implementar.

A continuación se presentan los pasos para el diseño e implementación de la solución de búsqueda propuesta en el proyecto para la opción de búsqueda dentro del sitio web de la ESPOL.

## 4.1. 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:

### 4.1.1. Obtención de los datos.

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

La sintaxis para obtener los datos es la siguiente:

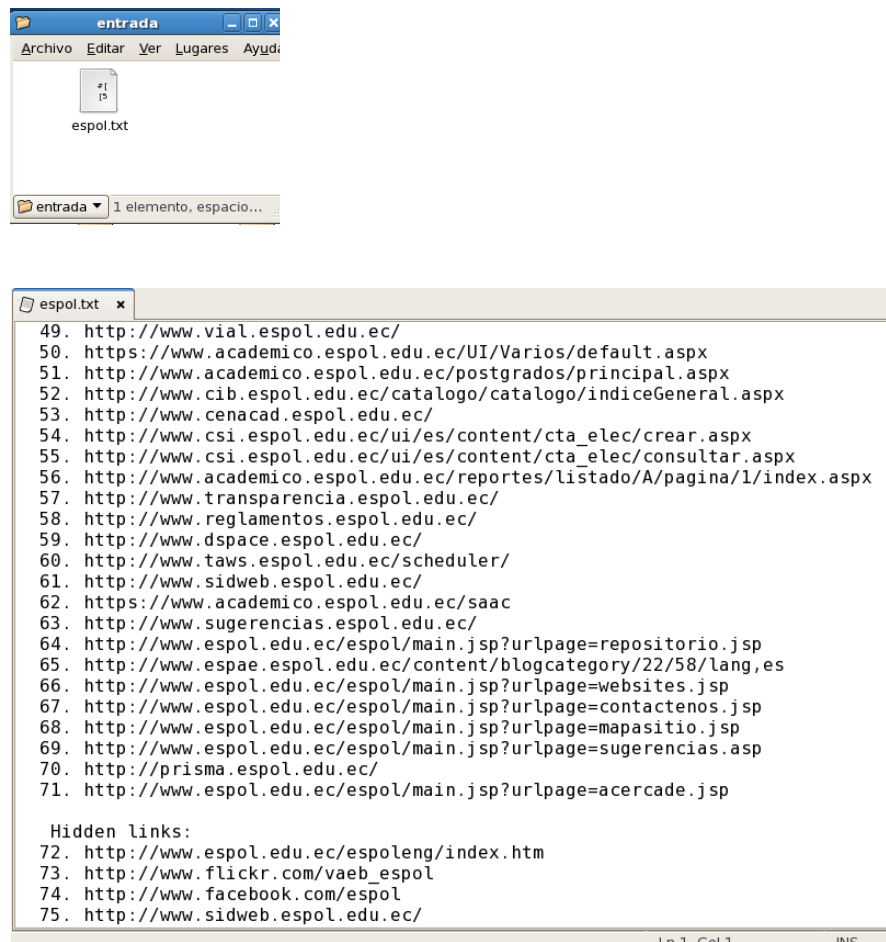
```
lynx -dump [web address] > webpagename.txt
```

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 un 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:



**Figura 4-1 – Gráfico del archivo de salida**



Tabla de contenido de la información de cada sitio web:

<i>Nombre Sitio</i>	<i>Nombre del archivo</i>	<i>Numero de palabras</i>
<b>www.espol.edu.ec</b>	espol.txt	
<b>www.icm.espol.edu.ec</b>	icm.txt	
<b>www.fiec.espol.edu.ec</b>	fiec.txt	

**Tabla 4-1 – Sitios Web de Espol**

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

#### **4.1.2. Configuración de hadoop.**

La configuración de hadoop más detallado se encuentra en el anexo A al final de documento. Por ahora solo se detallara lo más relevante para la implementación.

- 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

### 4.1.3. Proceso standalone.

Aquí se realiza el proceso de creación de directorios donde se realizara el proceso map-reduce:

```
$ mkdir input
$ cp conf/*.xml input
$ bin/hadoop jar hadoop-examples-*.jar
grep input output 'palabra a buscar'
$ cat output/*
```

### 4.1.4. Proceso pseudo distribuido.

Esta operación permite que hadoop ejecute los procesos de java de forma distribuida en los diferentes nodos. A continuación se indica que archivos se deben configurar:

- `conf/core-site.xml`
- `conf/hdfs-site.xml`
- `conf/mapred-site.xml`

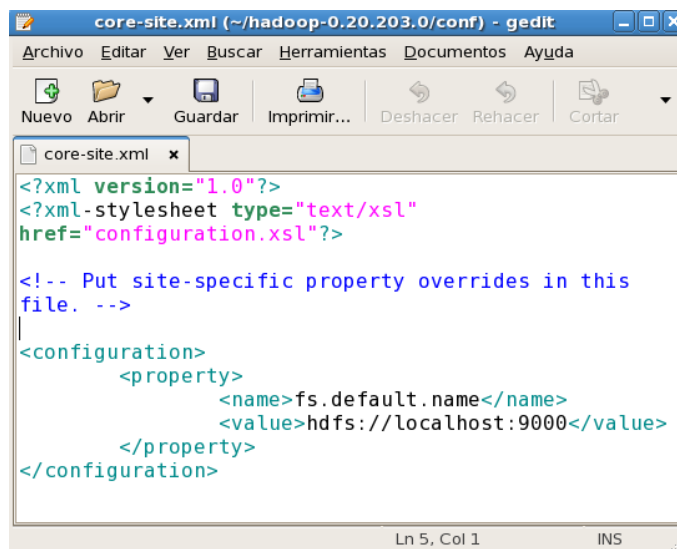


Figura 4-2 – Gráfico del archivo core-site

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl"
href="configuration.xsl"?>

<!-- Put site-specific property overrides in
this file. -->

<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <property>
    <name>dfs.name.dir</name>
    <value>/home/hadoop/dfs/name</value>
  </property>
  <property>
    <name>dfs.data.dir</name>
    <value>/home/hadoop/dfs/data</value>
  </property>
</configuration>
```

Figura 4-3 – Gráfico del archivo hdfs-site

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl"
href="configuration.xsl"?>

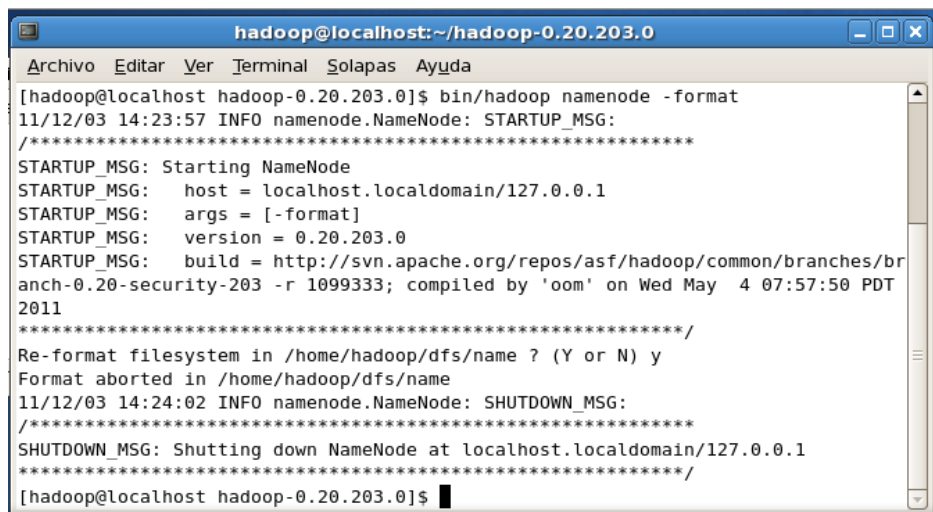
<!-- Put site-specific property overrides in
this file. -->

<configuration>
  <property>
    <name>mapred.job.tracker</name>
    <value>localhost:9001</value>
  </property>
  <property>
    <name>mapred.system.dir</name>
    <value>/hadoop/mapred/system</value>
  </property>
</configuration>
```

Figura 4-4 – Gráfico del archivo mapred-site

Luego de configurar los archivos, se procede a formatear el filesystem:

```
$ bin/hadoop namenode -format
```



```
hadoop@localhost:~/hadoop-0.20.203.0
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
[hadoop@localhost hadoop-0.20.203.0]$ bin/hadoop namenode -format
11/12/03 14:23:57 INFO namenode.NameNode: STARTUP_MSG:
/*****
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = localhost.localdomain/127.0.0.1
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 0.20.203.0
STARTUP_MSG:   build = http://svn.apache.org/repos/asf/hadoop/common/branches/branch-0.20-security-203 -r 1099333; compiled by 'oom' on Wed May  4 07:57:50 PDT 2011
*****/
Re-format filesystem in /home/hadoop/dfs/name ? (Y or N) y
Format aborted in /home/hadoop/dfs/name
11/12/03 14:24:02 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at localhost.localdomain/127.0.0.1
*****/
[hadoop@localhost hadoop-0.20.203.0]$
```

**Figura 4-5** – Gráfico del formateo del filesystem

Iniciar el levantamiento del demonio de hadoop

```
$ bin/start-all.sh
```

Una vez realizado los pasos anteriores 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/
```

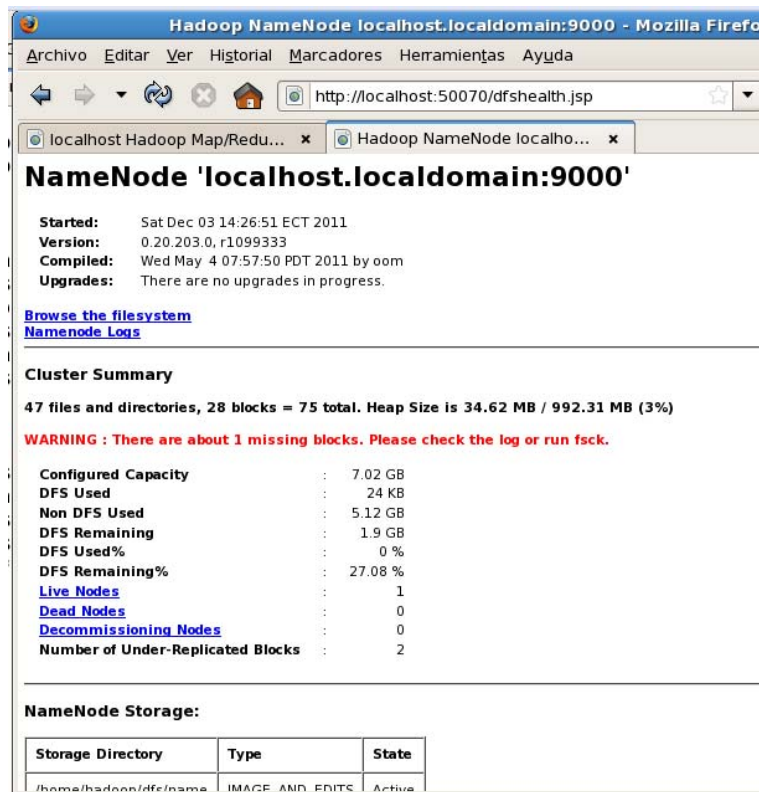


Figura 4-6 – Gráfico de ejecución de nodos

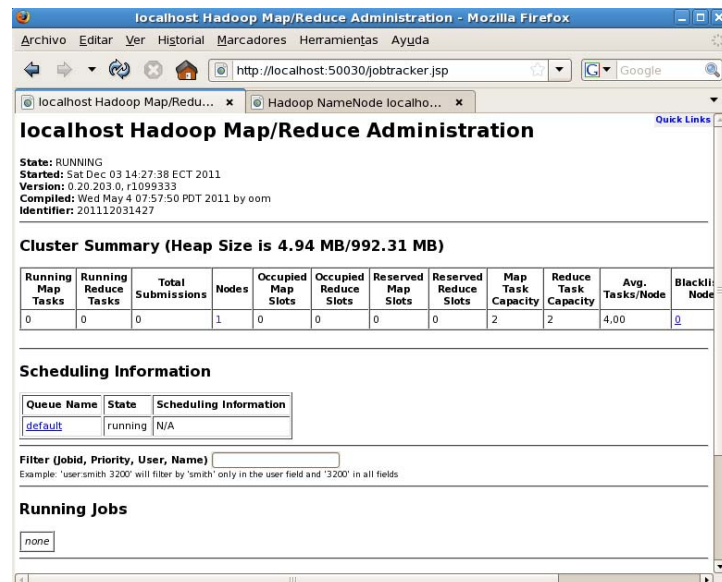


Figura 4-7 – Gráfico de ejecución de MapReduce

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'
```



**Figura 4-8** – Gráfico de salida del grep

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/*
```

O

```
$ bin/hadoop fs -cat output/*
```

Luego se debe poner en stop el demonio de hadoop

```
$ bin/stop-all.sh
```

Luego se realizarán las pruebas del funcionamiento de proceder con la creación de las clases en Java que van a procesar la información en una interfaz web.

#### 4.1.5. Clases en Java.

En esta parte se realiza la creación de las clases en Java gracias al 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:

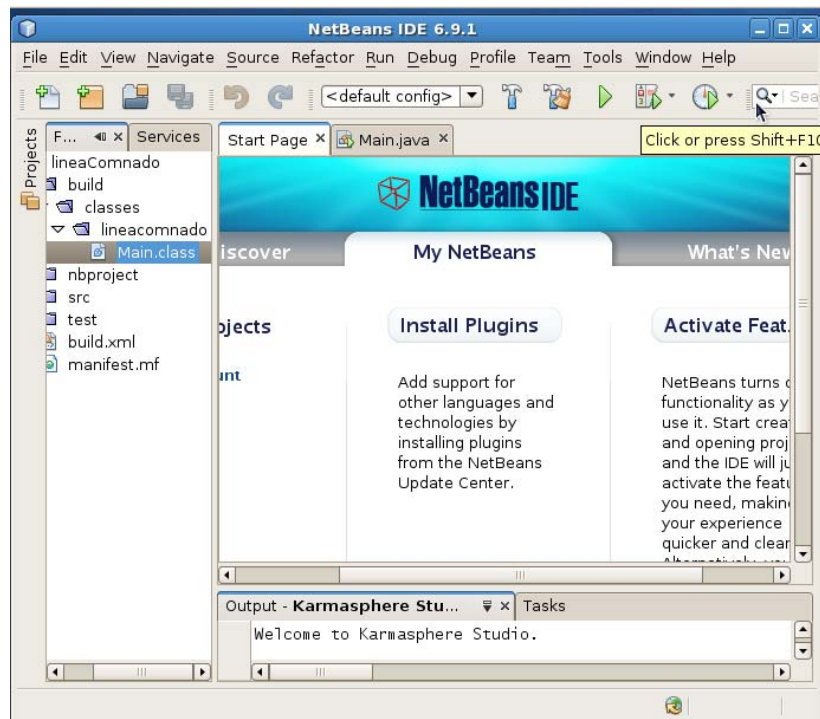
Código en Java:

```
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();
        }
    }
}
```

#### 4.1.6. 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.



**Figura 4-9 – Gráfico del IDE NetaBeans**

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.

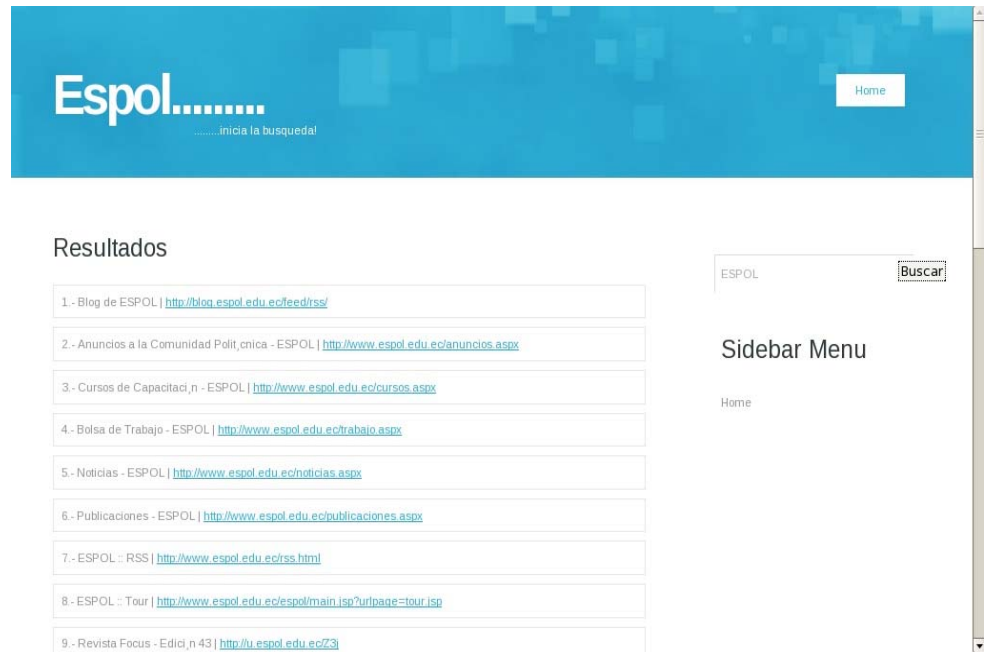
A continuación se muestra una pantalla grafica de la aplicación via web.

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.



Figura 4-10 – Gráfico de la interface web de la ESPOL

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



**Figura 4-11** – Gráfico de la interface web del proyecto

# CAPÍTULO 5

## 5. PRUEBAS Y RESULTADOS.

En el presente capítulo mostraremos las pruebas realizadas así mismo los resultados de las búsquedas en el sitio de la ESPOL y comparar el tiempo con el buscador de Google. Al final explicaremos el resultado de las pruebas.

### 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 es menor 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 5.1.** Gráfico comparativo de hadoop(1 observación)

<b>Tiempo Vs Nodos</b>	
<b>Nodos</b>	<b>Tiempo de búsqueda hadoop</b>
<b>2</b>	12 segundos
<b>4</b>	10 segundos

**Tabla 5.2.** Gráfico comparativo de hadoop(2 observación)

<b>Tiempo Vs Nodos</b>	
<b>Nodos</b>	<b>Tiempo de búsqueda hadoop</b>
<b>2</b>	12 segundos
<b>4</b>	9 segundos

**Tabla 5.3.** Gráfico comparativo de hadoop(3 observación)

<b>Tiempo Vs Nodos</b>	
<b>Nodos</b>	<b>Tiempo de búsqueda google</b>
<b>Índice web de Google</b>	0.46 segundos

**Tabla 5.4.** Gráfico comparativo de google

# CONCLUSIONES

Las conclusiones son:

1. 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 aprecia en los tiempos se requiere mayor cantidad de nodos y datos para que hadoop pueda ser utilizado de manera eficiente.
2. 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.
3. 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.

4. 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.
5. 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.



# RECOMENDACIONES

Las recomendaciones son:

1. Se recomienda que para futuras pruebas, se debe extraer la información completa de cada uno de los sitios web de ESPOL y realizar búsquedas más exhaustivas.
2. También se debe realizar una actualización de las versiones de hadoop para ver las mejoras en el análisis escalable de datos en los diferentes nodos del cluster.
3. Se recomienda incorporar la opción de búsqueda personalizada de hadoop dentro del sitio web de ESPOL para que los usuarios puedan ver los resultados de su búsqueda más detallada y precisa. Y posiblemente la creación de sesiones para personalizar aun más las búsquedas.

# BIBLIOGRAFÍA

- [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/>, fecha de consulta 2 de Octubre de 2011
- [2] **Yahoo**, Module 3: Getting Started With Hadoop  
<http://developer.yahoo.com/hadoop/tutorial/module3.html>, fecha de consulta 13 Octubre de 2011
- [3] **Faud NAHDI**, How to install JDK,  
<http://www.techonia.com/install-jdk-java-linux> , fecha de consulta 15 de Octubre de 2011
- [4] **VIVEK GITE**, HowTo: Use grep Command In Linux / UNIX,  
<http://www.cyberciti.biz/faq/howto-use-grep-command-in-linux-unix/> , fecha de consulta 22 de Octubre de 2011
- [5] **Arif N., Programming Hadoop in Netbeans**,  
<http://arifn.web.id/blog/2010/01/23/hadoop-in-netbeans.html> , 22 de Octubre de 2010

- [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> , fecha de consulta 30 de Octubre de 2011
- [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/> , fecha de consulta 8 de Noviembre de 2011
- [8] **Luciano**, Varios ejemplos de uso del comando grep,  
<http://luauf.com/2009/05/04/varios-ejemplos-de-uso-del-comando-grep/>, fecha de consulta 22 de Noviembre de 2011
- [9] **Brandeis University**, Hadoop Example Program,  
<http://pages.cs.brandeis.edu/~cs147a/lab/hadoop-example/>, fecha de consulta 22 de Octubre de 2011
- [10] **Leons Petražickis**, Hadoop Lab 1 - HDFS Instructions,  
<http://www.slideshare.net/leonsp/hadoop-lab-1-hdfs-instructions> , fecha de consulta 21 Noviembre 2011
- [11] **linglom**, How to run command-line or execute external application from Java  
<http://www.linglom.com/2007/06/06/how-to-run-command-line-or-execute-external-application-from-java/>, fecha de consulta 30 Noviembre 2011



