ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Sistema de reportes y análisis sobre tendencias en la Web de la ESPOL usando Hadoop para el procesamiento masivo de los datos.

TESINA DE SEMINARIO

Previo la obtención de los Títulos de:

INGENIERO EN CIENCIAS COMPUTACIONALES

ESPECIALIZACIÓN SISTEMAS DE INFORMACION

INGENIERO EN CIENCIAS COMPUTACIONALES

ESPECIALIZACIÓN SISTEMAS TECNOLÓGICOS

Presentada por:

LUIS ALFREDO GALLARDO MALDONADO

FABRICIO BOLIVAR BERMEO ROMERO

GUAYAQUIL – ECUADOR

2011

DEDICATORIA

A mis amigos y familiares, que de una u otra manera estuvieron presentes en todo este proceso académico.

En especial va dedicado a mis padres Luis Alfredo y Mercedes por todo ese apoyo incondicional y por todas esas palabras de aliento en los momentos más difíciles. Gracias por toda esa confianza depositada en mí, ayudándome a cumplir mis metas como persona, además gracias por todas las enseñanzas, consejos y amor brindado.

LUIS ALFREDO GALLARDO MALDONADO

Primero quiero agradecerle a Dios por darme la sabiduría, fortaleza y nunca dejarme caer ante las adversidades que la vida nos conlleva.

Segundo agradecer a mi madre, es por ella, que con su sacrificio y su constancia me ayudo para seguir adelante en esta dura carrera universitaria, quien con su paciencia, su alegría, sus consejos supo guiarme por el buen camino para poder terminar mis estudios ,este título no lleva mi nombre sino el suyo Carmen Elena Romero Reinoso.

Sin dejar atrás a mis hermanas, Marcia que con su apoyo desde el inicio supo encaminarme para mi desarrollo universitario, Gisella, con

su ayuda en los momentos difíciles siempre estuvo allí, Yulissa, quien siempre me ayudo en todo lo que necesite en mi carrera, la que nunca me dijo no y siempre estuvo pendiente de mi, Mercy, con sus oraciones supo guiarme y estar junto a Dios con sus plegarias, a mi Padre por sus enseñanzas del día a día y a todos quienes siempre estuvieron pendiente de mi durante todo este tiempo, se los agradezco mucho.

FABRICIO BOLIVAR BERMEO ROMERO

AGRADECIMIENTO

Este proyecto es el resultado del esfuerzo conjunto de los que formamos el grupo de trabajo. Agradecemos a todas las personas que de una u otra manera colaboraron en la realización de este trabajo, en especial a la Ing. Vanessa Cedeño Directora de Proyecto y al Ing. Lenín Freire Miembro Principal.

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)

Luis Alfredo Gallardo Maldonado

Fabricio Bolívar Bermeo Romero

TRIBUNAL DE SUSTENTACIÓN

Ing. Vanessa Cedeño

PROFESORA DEL SEMINARIO DE GRADUACION

Ing. Lenín Freire

PROFESOR DELEGADO DEL DECANO

RESUMEN

Los sistemas de reportes y análisis sobre tendencias son ampliamente utilizados hoy en día gracias a su capacidad de analizar las palabras más usadas por los usuarios en la web, por ejemplo se han usado estos sistemas en las redes sociales, ya que las mismas en los últimos años han tenido gran acogida.

En la web de la ESPOL existe una gran cantidad de datos, y no existe una herramienta que permita recolectarlos y realizar un análisis de su contenido, para resolver este problema usamos Hadoop que es una plataforma que nos permite desarrollar aplicaciones que tengan que tratar con grandes cantidades de datos, hasta petabytes.

Los programas MapReduce de Hadoop están diseñados para computar grandes volúmenes de datos en paralelo. El ejemplo más claro de esto, lo que va a ser de gran ayuda y que se usa para este tipo de problemas es el WordCount, que lee archivos de texto y cuenta con qué frecuencia ocurren las palabras.

En el WordCount la entrada es un archivo de texto y la salida es otro archivo de texto; donde el primero es un archivo donde se encuentran todas las palabras, y el segundo en cada línea tiene una palabra y el número de frecuencia de la misma. Se usa esta función para que nos permita conocer las palabras que más usan en la web los miembros de la comunidad de la ESPOL. Con todos estos datos recogidos vamos a mostrar el reporte y análisis en un gráfico que mostrará las tendencias de las palabras dinámicamente.

INDICE GENERAL

INDICE DE FIGURAS

ABREVIATURAS

ESPOL: Escuela Superior Politécnica del Litoral

URL: Uniform Resource Locator (Localizador de Recurso Uniforme)

FTP: File Transfer Protocol (Protocolo de Transferencia de Archivo)

HTTP: Hypertext Transfer Protocol (Protocolo de Transferencia de Hipertexto)

HTTPS: Hypertext Transfer Protocol Secure (Protocolo Seguro de Transferencia de Hipertexto)

TCP/IP: Transmission Control Protocol/Internet Protocol (Protocolo de Control de Transmisión/Protocolo de Internet)

INTRODUCCION

Durante los últimos años, los sistemas de reportes y análisis sobre tendencias han sido herramientas que han contribuido a mejorar la experiencia de los usuarios en las aplicaciones que estos utilizan, dado que los mismos nos permiten conocer acerca de los temas que son de mayor popularidad en los diferentes sitios.

Numerosas empresas han apostado por el uso de estas herramientas para lograr que sus sitios web posean un ambiente personalizado, en donde los usuarios puedan observar las palabras más usadas por ellos mismos. Algunos de los sitios web en donde se puede encontrar el uso de estos sistemas son Google, Facebook, Twitter, etc. Los cuales permiten ver un análisis estadístico de las palabras más usadas por los usuarios de estos sitios.

El contenido de este trabajo se distribuye de la siguiente manera:

En el capítulo 1 se describe el problema, antecedentes, objetivos, justificación y alcance del presente trabajo.

En el capítulo 2 se presentan los fundamentos teóricos de sobre las tecnologías utilizadas, como HADOOP que es una plataforma la cual permite desarrollar aplicaciones que tengan que tratar con grandes cantidades de datos.

En el capítulo 3 se describe el análisis de la solución, con la ayuda de dichas herramientas, para esto se usa los diferentes métodos que utiliza HADOOP para el muestreo de los datos, luego en el cual se usa Adobe Flex Builder para el análisis de los resultados.

En el capítulo 4 se muestra el diseño y se detalla la implementación de la solución presentada, tanto para el procesamiento masivo de datos (HADOOP) como el diseño de la interfaz con el usuario que se lo realizará en la herramienta gráfica Flex, por medio de componentes que tiene dicha herramienta.

Finalmente, las pruebas y resultados son mostrados en el capítulo 5.

CAPITULO 1

1. Antecedentes y Justificación

1.1. Antecedentes y Descripción del Problema

Antecedentes

En el ámbito del internet los sistemas de reportes y análisis sobre tendencias han sido ampliamente usados para el beneficio de sus propios usuarios, que les permite conocer sobre sus propios temas de interés.

Como ejemplo de esto tenemos a Trendsmap que es una interesante herramienta para conocer en tiempo real que es lo que está sucediendo en tu región, cuales son los hashtag más populares y sobre que tendencias se está hablando en Twitter, pero Trendsmap no es una herramienta de tendencias cualquiera, lo caracteriza su tecnología de GeoIP gracias a Google, lo cual nos permite tener cierta precisión sobre la ubicación de un hashtag en una determinada región.

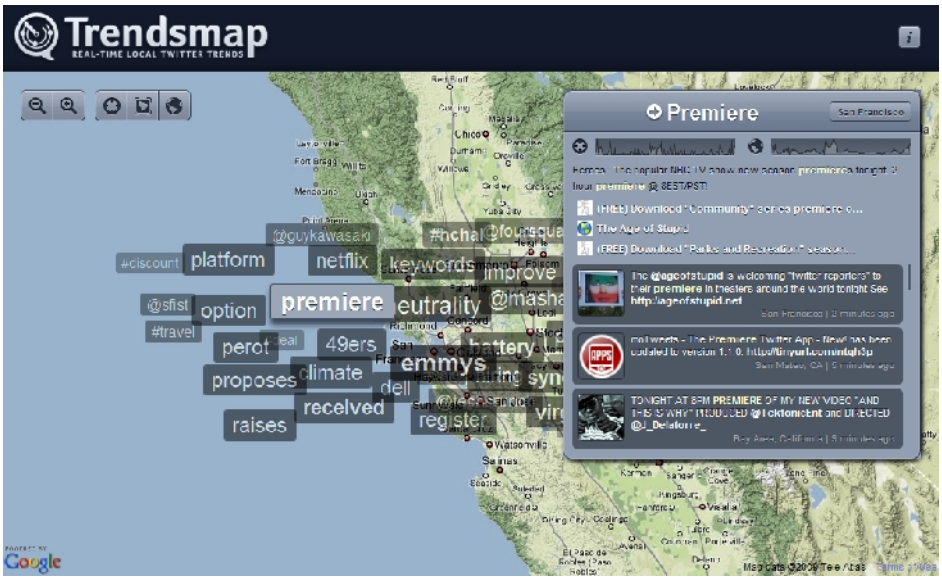

Figura 1. Trendsmap de Twitter

También existe Google Trends que es una herramienta de Google Labs que muestra los términos de búsqueda más populares del pasado reciente.

Las gráficas de Google Trends representan con cuánta frecuencia se realiza una búsqueda particular en varias regiones del mundo y en varios idiomas. Una característica adicional de Google Trends es la posibilidad de mostrar noticias relacionadas con el término de búsqueda encima de la gráfica, mostrando cómo afectan los eventos a la popularidad.

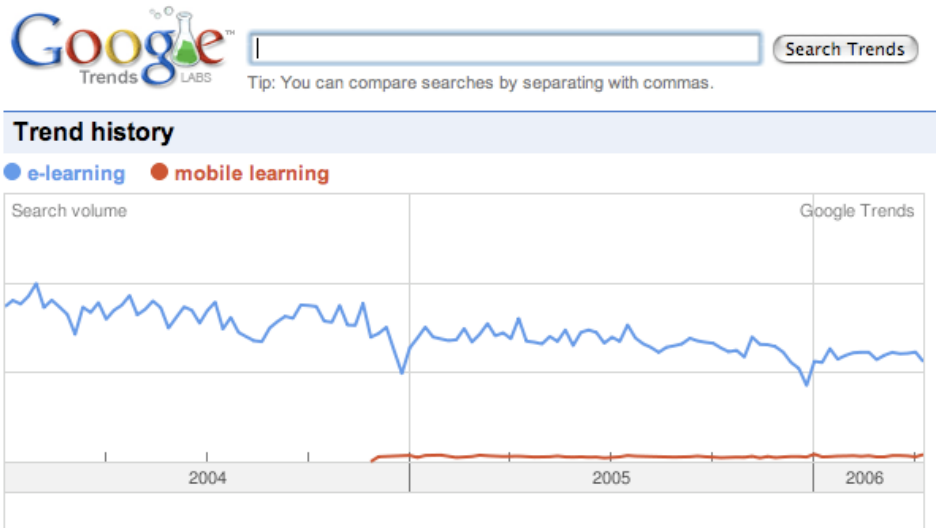

Figura 2. Google Trends

Otro ejemplo de estas herramientas es el Lexicon de Facebook, que fue usada hace algún tiempo y que por el momento se encuentra deshabilitada, esta es una herramienta para seguir las tendencias lingüísticas en esta red social, consulta el uso de palabras y frases en los muros de perfiles, grupos y eventos. Por ejemplo, puedes

introducir “amor, odio” para comparar el uso de estas dos palabras en los muros de Facebook.

Por último, cabe agregar que esta aplicación nos permite determinar (o al menos tener una aproximación) del éxito que una campaña viral ha tenido en Facebook.

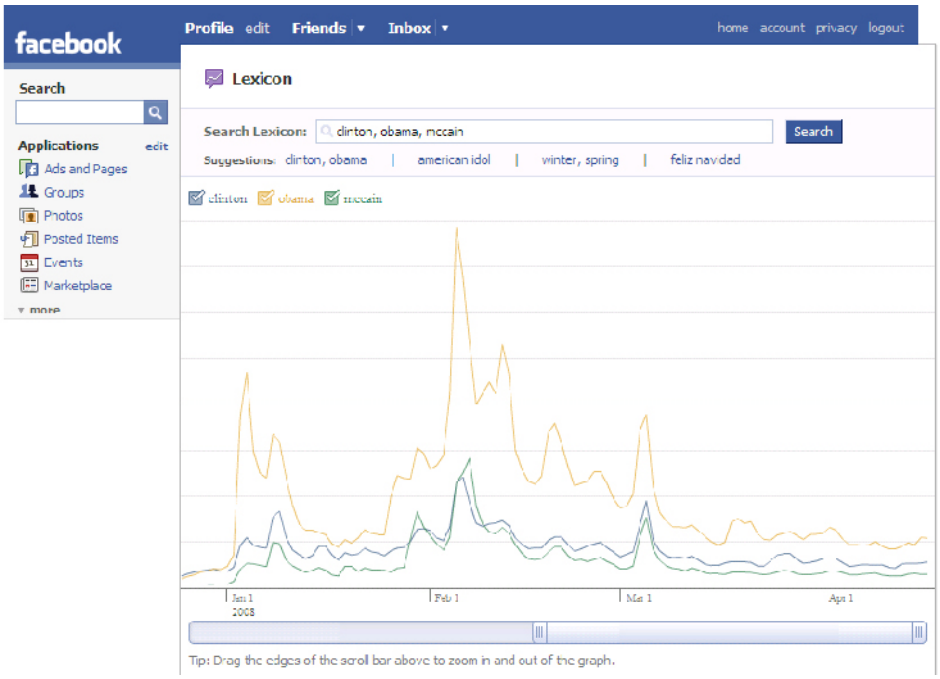

Figura 3. Lexicon de Facebook

Descripción del Problema

De acuerdo a lo citado en la sección anterior, vemos que podemos tener un gran universo de palabras en cada sitio web, por este motivo

se ha llevado a construir este tipo de herramientas que nos ayude a obtener una visión de los temas que más están de moda. Cada sitio tiene su propia idea de construcción de estos sistemas, podemos tener un resultado de temas pasados o actuales, cuyos resultados se nos va a mostrar en un gráfico sencillo o gráficos más complejos con mapas y ubicaciones de donde surgieron dichos temas.

Este escenario permite plantear la construcción de un sistema de reportes y análisis sobre tendencias de las palabras más usadas en la página de facebook de la ESPOL (http://www.facebook.com/espol - imagen de la página en el **ANEXO A**), para que a los estudiantes les permita conocer sobre los temas que más escriben.

1.2. Justificación

Esta herramienta de reportes y análisis sobre tendencias constituye una herramienta de gran interés, ya que en la actualidad no existe una herramienta que nos permita conocer los temas de interés de los estudiantes.

El presente trabajo nace como una iniciativa que busca explorar la viabilidad de la aplicación de los sistemas de reportes y análisis sobre

tendencias en la web de la universidad, y hallar un escenario para la implementación de este tipo de sistemas dentro de ella. De esta manera construir un trabajo que sirva de base a futuros proyectos dentro de ESPOL.

Para esto se propone el desarrollo de este tipo de sistema usando Hadoop para el procesamiento masivo de los datos, además de otras herramientas que nos servirán para una mayor visibilidad de los resultados, todo esto usado en conjunto nos permitirá conocer las tendencias de las palabras más usadas en la página de la ESPOL en Facebook.

1.3. Objetivos

- Crear una aplicación usando Hadoop que permita conocer lo que buscan o discuten en la web los miembros de la comunidad de la ESPOL.
- Identificar las tendencias sobre temas populares mediante un análisis estadístico de palabras a través de la página de facebook de la ESPOL.

1.4. Alcance

- Poder tener una tendencia sobre los temas más leídos y revisados por parte de las personas que son miembros de la comunidad de la ESPOL en facebook.

- Poder visualizar en que período las entradas tuvieron más acogida por los usuarios.

- Con los resultados obtenidos, poder dar una orientación al administrador de la cuenta de facebook de la ESPOL que temas son los más interesados en leer por parte de los estudiantes o usuarios, para así tener una gran cantidad de usuarios registrados en sus cuentas.

CAPITULO 2

2. Fundamentos Teóricos

2.1. Hadoop

Figura 4. Logotipo de Hadoop

Hadoop es una plataforma que nos permite desarrollar aplicaciones que tengan que tratar con grandes cantidades de datos, hasta petabytes. Se trata de un sub-proyecto de Lucene, un proyecto de

Apache que desarrolla software para realizar búsquedas. Hadoop es muy útil cuando vamos a realizar proyectos que necesiten de escalabilidad, ya hemos dicho que puede almacenar y procesar petabytes de información. A su vez, es perfecto para un clúster de servidores, distribuyendo la información entre los nodos, siendo posible disponer de miles de nodos. Al disponer los datos de forma distribuida, la búsqueda se puede realizar muy rápidamente ya que Hadoop puede acceder a ella de forma paralela. Y aunque los datos estén distribuidos, no hay que preocuparse de fallos ya que dispone de un sistema de seguridad.

2.1.1. Arquitectura

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 Hadoop. El paquete también proporciona código fuente, documentación, y una sección de contribución que incluye proyectos de la Comunidad Hadoop.

2.1.1.1. Sistema de Archivos

2.1.1.1.1. Hadoop Distributed File System

El Hadoop Distributed File System (HDFS) es un sistema de archivos distribuido, escalable y portátil escrito en Java para el framework Hadoop. Cada nodo en una instancia Hadoop típicamente tiene un único nodo de datos; un clúster de datos forma el clúster HDFS. La situación es típica porque cada nodo no requiere un nodo de datos para estar presente. Cada nodo sirve bloques de datos sobre la red usando un protocolo de bloqueo específico para HDFS.

El sistema de archivos usa la capa TCP/IP para la comunicación; los clientes usan RPC para comunicarse entre ellos. El HDFS almacena archivos grandes (el tamaño ideal de archivos es de 64 MB), a través de múltiples máquinas. Consigue fiabilidad mediante replicada de datos a través de múltiples hosts, y no requiere almacenamiento RAID en ellos. Con el valor de replicación por defecto, 3, los datos se almacenan en 3 nodos: dos en el mismo rack, y otro en un rack distinto. Los nodos de datos pueden hablar

entre ellos para reequilibrar datos, mover copias, y conservar alta la replicación de datos.

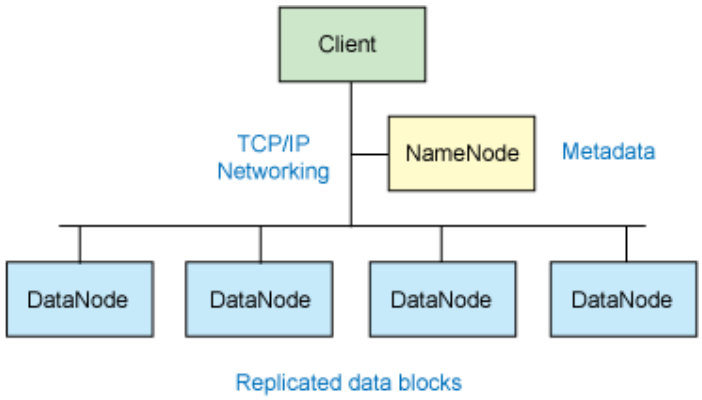

Figura 5. Visión simplificada de un clúster Hadoop

2.1.1.1.2. Otros sistemas de archivos

A junio de 2010, la lista de sistemas de archivos soportados incluye:

- HDFS: El sistema propio de Hadoop. Está diseñado para la escala de decenas petabytes de almacenamiento y funciona sobre los sistemas de archivos de base.
- Amazon S3: éste se dirige a clústeres almacenados en la infraestructura del servidor bajo demanda Amazon Elastic Compute Cloud. No hay conciencia de racks en este sistema de archivos, porque todo es remoto.

- CloudStore (previamente llamado KosmosDistributed File System), el cual es consciente de los racks.
- FTP: éste almacena todos sus datos en un servidor FTP accesible remotamente.
- HTTP y HTTPS de solo lectura.

2.1.2. Usuarios prominentes

El 19 de febrero de 2008, Yahoo! Inc. lanzó lo que pretendía era la más grande aplicación de producción Hadoop. El Yahoo! SearchWebmap es una aplicación de Hadoop que se ejecuta en más de 10.000 núcleos Linux Clústeres de racimo y produce datos que se utilizan actualmente en todos los resultados de búsqueda de Yahoo!. En junio de 2009, Yahoo! hizo disponible el código fuente de la versión de Hadoop que usa en producción.

Aparte de Yahoo!, otras organizaciones usan Hadoop para ejecutar cómputos enormes distribuidos, como por ejemplo Facebook, Amazon, Google, etc.

Figura 6. Usuarios de Hadoop

2.1.3. MapReduce

MapReduce es un modelo de programación diseñado para procesar grandes volúmenes de datos en paralelo dividiendo el trabajo en un conjunto de tareas independientes. Los programas MapReduce son escritos en un particular estilo influenciado por construcciones de *programación funcional,* específicamente lenguajes para el procesamiento de listas de datos. En este módulo se explica la naturaleza de este modelo de programación y cómo puede ser utilizado para escribir programas que se ejecutan en el entorno de Hadoop.

2.1.3.1. Requisitos

Este módulo requiere que haya configurado un entorno de desarrollo con Hadoop instalado y configurado correctamente, esto se muestra en el **ANEXO D**.

2.1.3.2. Conceptos de programación funcional

Los programas MapReduce están diseñados para calcular grandes cantidades de datos en forma paralela. Para ello es necesario dividir la cantidad de trabajo entre un gran número de máquinas. Este modelo no escalaría a clústeres (cientos o miles de nodos) si a los componentes se les permitiría compartir los datos de forma arbitraria. Los gastos generales de comunicación necesarios para mantener los datos en los nodos sincronizados en todo momento evitarán que el sistema funcione de forma fiable y eficiente a gran escala.

Por el contrario, todos los elementos de datos en MapReduce son *inmutables,* es decir, que no pueden ser actualizados. Si en una tarea de mapeo se cambia una entrada (clave, valor) par, no se refleja en los archivos de entrada, la comunicación se

produce únicamente mediante la generación de una nueva salida (clave, valor) pares que después son remitidos por el sistema de Hadoop en la siguiente fase de ejecución.

2.1.3.3. Lista de procesamiento

Conceptualmente, los programas de MapReduce transforman listas de elementos de datos de entrada en listas de elementos de datos de salida. Un programa de MapReduce hará esto dos veces, usando dos diferentes lenguajes de procesamiento de listas: *Map* y *Reduce.* Estos términos son tomados de varios lenguajes de procesamiento de listas, tales como LISP, Scheme o ML.

2.1.3.4. Mapeo de listas

La primera fase de un programa de MapReduce se llama *mapeo.* Una lista de elementos de datos son proporcionados, de uno en uno, a una función llamada *Mapper,* que transforma cada elemento de forma individual a un elemento de datos de salida.

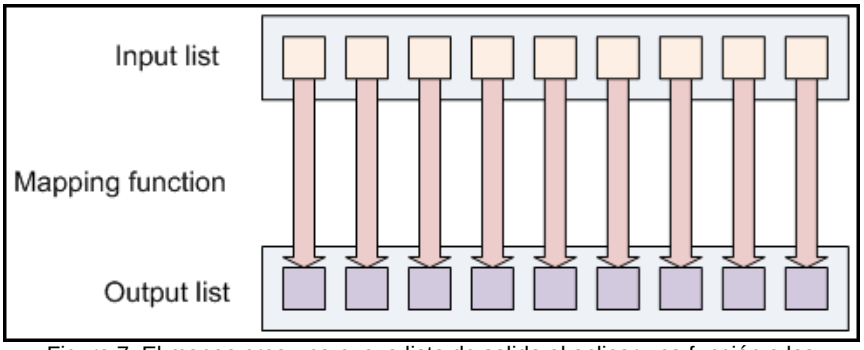

Figura 7. El mapeo crea una nueva lista de salida al aplicar una función a los elementos individuales de una lista de entrada

Como ejemplo de la utilidad del Map: Supongamos que se tiene una función `toUpper(str)` que devuelve una versión en mayúsculas de su cadena de entrada. Se podría utilizar esta función con el Map para convertir una lista de cadenas en una lista de cadenas mayúsculas. Tenga en cuenta que no estamos modificando la cadena de entrada aquí: estamos volviendo una nueva cadena que formará parte de una nueva lista de salida.

2.1.3.5. Reducción de listas

La reducción permite agregar valores juntos. Una función de *Reduce* recibe de una lista de entrada un iterador de valores de entrada. Luego combina estos valores juntos, devolviendo un sólo valor de salida.

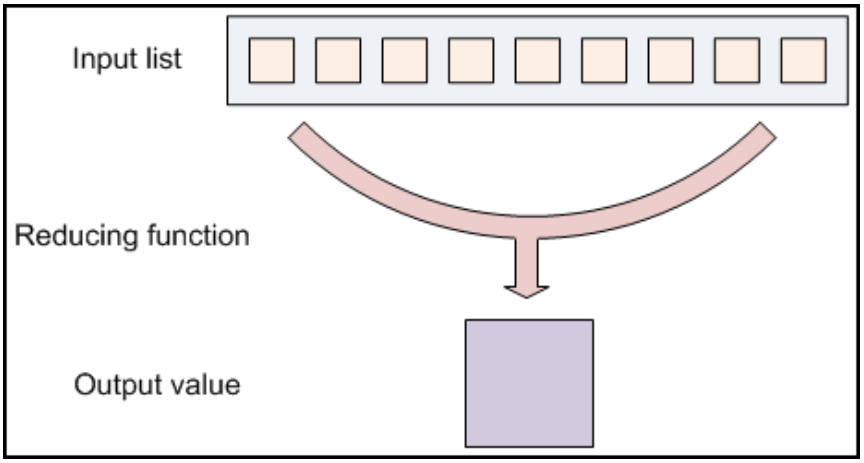

Figura 8. Reducir una lista se repite en los valores de entrada para producir un valor agregado como salida

La reducción se utiliza a menudo para producir "resumen" de datos, convirtiendo un gran volumen de datos en un resumen más pequeño de sí mismo. Por ejemplo, "+" puede ser utilizado como una función de reducción, para devolver la suma de una lista de valores de entrada.

2.1.3.6. Ponerlos juntos en MapReduce

El marco Hadoop MapReduce toma estos conceptos y los utiliza para procesar grandes cantidades de información. Un programa MapReduce tiene dos componentes: uno que implementa el Mapper, y otro que implementa el Reductor. Los idiomas Mapper y Reductor descritos anteriormente se

extienden ligeramente para trabajar en este entorno, pero los principios básicos son los mismos.

Claves y valores: En MapReduce, ningún valor se encuentra por sí solo. Cada valor tiene una *clave* asociada a él. Las claves identifican valores relacionados. Por ejemplo, lecturas de velocímetro con registro de código de tiempo desde varios autos podrían ser introducidos por el número de la matrícula, se vería así:

```
AAA-123    65 mph, 12:00 pm
ZZZ-789    50 mph, 24:02
AAA-123    40 mph, 24:05
CCC-456    25 mph, 12:15 pm
...
```

La funciones de mapeo y reducción no sólo reciben valores, sino (clave, valor) pares. La salida de cada una de estas funciones es la misma: una clave y un valor deben ser emitidas a la siguiente lista en el flujo de datos.

MapReduce también es menos estricto que otros lenguajes sobre cómo el Mapper y el Reductor trabajan. En un mapeo funcional más formal, y en reducir marcos, un Mapper debe producir exactamente un elemento de salida para cada elemento de entrada, y un Reductor debe producir exactamente

un elemento de salida para cada lista de entrada. En MapReduce, un número arbitrario de valores puede ser la salida de cada fase, un Mapper puede asignar una entrada en cero, uno, o cien salidas. Un Reductor puede calcular más de una lista de entrada y emitir uno o una docena de salidas diferentes.

Las claves dividen el espacio reducido: Una función de reducción convierte una larga lista de valores en uno (o pocos) valores de salida. En MapReduce, todos los valores de salida generalmente no se reducen juntos. Todos los valores *con la misma clave* se presentan a un sólo reductor juntos. Esto se realiza de forma independiente de cualquier operación de reducción que ocurre en otras listas de valores, con diferentes claves adjuntas.

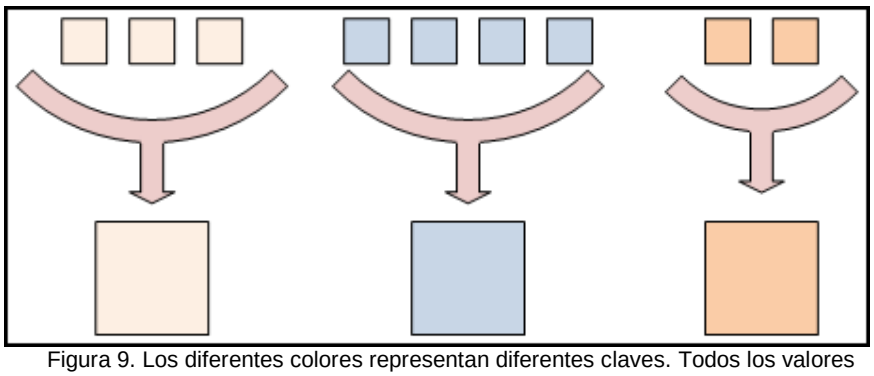

Figura 9. Los diferentes colores representan diferentes claves. Todos los valores con la misma clave se presentan a una sola tarea de reducir

2.1.3.7. Un ejemplo de aplicación: WordCount

Un simple programa MapReduce puede ser escrito para determinar cuántas veces aparecen diferentes palabras en un conjunto de archivos. Por ejemplo, si tenemos los archivos:

foo.txt: *Dulce, este es el archivo foo*

bar.txt: *Este es el archivo de barras*

Es de esperar que la salida sea:

dulce	1
este	2
es	2
el	2
foo	1
barras	1
archivo	2
de	1

Lógicamente, se puede escribir un programa en MapReduce para calcular esta salida. La estructura de alto nivel se vería así:

```
mapper (nombre de archivo, archivo-contenido):
  para cada palabra en archivo-contenido:
       emiten (palabra, 1)


reductor (palabra, valores):
  suma = 0
  para cada valor en valores:
       suma = suma + valor
  emiten (word, sum)
```

Varios casos de la función de mapeo se crean en las diferentes máquinas en nuestro clúster. Cada caso recibe un archivo de entrada diferente (se supone que tenemos muchos archivos de este tipo). La salida de mappers (palabra, 1) pares que se envían a los reductores. Varios casos del método reductor también son ejemplificados en las diferentes máquinas. Cada reductor es responsable de procesar la lista de valores asociados con una palabra diferente. La lista de valores será una lista de números 1, el reductor resume aquellos en un recuento final asociado con una sola palabra. El reductor emite la (palabra, número) final de salida que está escrito en un archivo de salida.

Podemos escribir un programa muy similar a este en Hadoop MapReduce, está incluido en la distribución Hadoop en `src/examples/org/apache/hadoop/examples/WordCount.java`
Está reproducido parcialmente a continuación:

```
public static class MapClass extends MapReduceBase
    implements Mapper<LongWritable, Text, Text, IntWritable> {

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(LongWritable key, Text value,
                    OutputCollector<Text, IntWritable> output,
                    Reporter reporter) throws IOException {
      String line = value.toString();
      StringTokenizer itr = new StringTokenizer(line);
      while (itr.hasMoreTokens()) {
```

```
        word.set(itr.nextToken());
        output.collect(word, one);
      }
    }
  }

  /**
   * A reducer class that just emits the sum of the input
   * values.
   */
  public static class Reduce extends MapReduceBase
    implements Reducer<Text, IntWritable, Text, IntWritable> {

    public void reduce(Text key, Iterator<IntWritable> values,
                       OutputCollector<Text, IntWritable>
output,
                       Reporter reporter) throws IOException {
      int sum = 0;
      while (values.hasNext()) {
        sum += values.next().get();
      }
      output.collect(key, new IntWritable(sum));
    }
  }
```

Hay algunas pequeñas diferencias entre la implementación actual de Java y el pseudo-código que se muestra arriba. En primer lugar, Java no tiene la palabra clave nativa `emitir`, el objeto `OutputCollector` dado como una entrada recibirá los valores para emitir a la siguiente etapa de ejecución. Y en segundo lugar, el formato de entrada por defecto utilizado por Hadoop presenta cada línea de un archivo de entrada como una entrada independiente a la función de mapeo, no todo el archivo a la vez. También utiliza un objeto `StringTokenizer` para romper la línea en palabras. Esto no lleva a cabo la normalización de la entrada, por lo que "gato", "Gato" y "gato", son considerados como diferentes cadenas. Tenga en cuenta

que la variable `word` se reutiliza cada vez que el mapper produzca otra (la palabra, 1) en par, lo que ahorra tiempo al no asignar una nueva variable para cada salida. El método `output.collect()` copiará los valores que recibe como entrada, para que esté pueda sobrescribir las variables que se utiliza.

CAPITULO 3

3. Análisis de la solución

La página de facebook de la ESPOL nos ofrece todos los registros de las publicaciones y comentarios de los estudiantes, acerca de los temas de actualidad en la universidad. Estos datos nos pueden ayudar a conocer sobre qué temas son los más populares, es decir los que más son de objetivo de comentarios.

Se Utilizan estos datos para analizarlos y buscar que palabras son las más comunes en la comunidad politécnica, todos estos datos los

llevamos a gráficos para mostrar los detalles de las tendencias de dichas palabras.

3.1. Requerimientos

Teniendo en consideración que la solución propuesta se centra en la construcción de un sistema de reportes y análisis basado en los comentarios del usuario, se definen las características principales que éste debe cumplir, en las sub-secciones a continuación.

3.1.1. Requerimientos Funcionales

- El sistema tendrá el objetivo de poder comparar diferentes palabras que son usadas en la página de facebook de la ESPOL.
- La comparación de dichas palabras se lo realizará mediante gráficos estadísticos, donde se presentará el número de repeticiones que haya tenido cada palabra.
- Podremos ver la frecuencia por fecha, agrupadas por año y mes, en la que cada palabra fue mencionada para las entradas en su cuenta de facebook.
- El usuario podrá ingresar hasta un máximo de 5 palabras para poder realizar la comparación.

3.1.2. Requerimientos No Funcionales

- La aplicación podrá ser visualizada en diferentes exploradores como: Mozilla Firefox, Internet Explorer, Google Chrome y cualquier otro navegador web.
- La aplicación podrá ser utilizada por cualquier persona que esté interesada en ver las tendencias de las diferentes palabras más utilizadas en su cuenta de facebook de la ESPOL.
- El sistema debe ser escalable, es decir que debe ser construido de tal manera que se puedan añadir nuevos criterios de análisis de las palabras.

3.2. Análisis de las Capacidades del Sistema de reportes y análisis sobre tendencias en la Web de la ESPOL

Analizando los capítulos anteriores y observando las herramientas que podemos utilizar para implementar este sistema, nos damos cuenta que podemos hacer uso de las funciones básicas de cada una de las herramientas y mezclándolas entre sí, para obtener un resultado con grandes cantidades de información, para así luego mostrar los detalles de los mismos.

Con ésta podemos asegurarnos de tener un sistema que será capaz de mostrar las palabras que son tendencias en la web de la ESPOL a través de gráficos detallados que mostrarán su análisis. Los gráficos serán mostrados con la herramienta especificada, con los datos que recoge de una base de datos, dicha base tendrá la capacidad de almacenar grandes cantidades de datos que serán fácilmente procesados.

Por ejemplo tenemos un gran universo de palabras del cual vamos a escoger una, a ésta se la analizará con respecto al número de veces que se ha nombrado en la web y se mostrará el respectivo reporte de la misma. Además también se podrá comparar dos o más palabras para ver cuál de ellas ha sido de más popularidad entre los usuarios.

Figura 10. Ejemplo de un universo de palabras

De ésta manera, la única forma de determinar la popularidad de cada palabra será por el número de veces que los usuarios la escriben y la fecha en que han sido escritas.

CAPITULO 4

4. Diseño e Implementación del Sistema de reportes y análisis sobre tendencias en la Web de la ESPOL

En este capítulo se describe la metodología usada para la implementación del sistema de reporte y análisis sobre las tendencias en la web de la ESPOL. Posteriormente se detalla el plan de pruebas utilizado.

4.1. Modelo del Sistema de reportes y análisis

El sistema para llegar a mostrar sus resultados finales se lo tuvo que realizar por diferentes etapas y con diferentes herramientas. Los trabajos de implementación realizados se explican en las etapas descritas en la siguiente sección.

4.1.1. Modelo Lógico del Sistema de reportes y análisis

Scripts para la obtención del contenido

El procedimiento para obtener el análisis y reporte de las tendencias se centra en la obtención de los todos los comentarios que existan en la página de facebook de la ESPOL. Para esto se realizó dos scripts, a continuación se detalla cada uno de ellos.

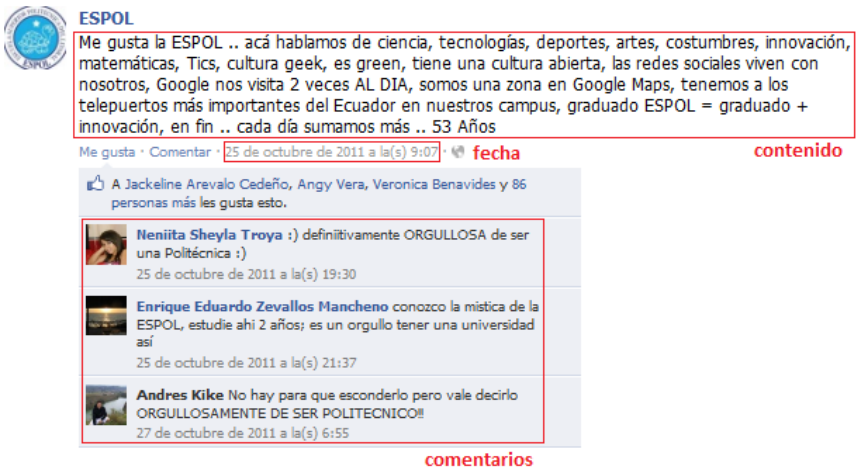

Figura 11. Post de la página de facebook de la ESPOL

El primer script se trata de la obtención de los URL de todos los post publicados en la página web (El código java se encuentra en el **ANEXO B**). A continuación presentamos el pseudocódigo que ilustra los pasos de obtención de los mismos:

```
1   INICIO
2
3   Variables: posts,result,guardados
4   Conectarse con la URL: "http://www.facebook.com/espol"
5   Almacenar todo el contenido de la pagina en: result
6   Variable: busqueda = "\\/espol\\/posts\\/";
7
8   Bucle hasta que en result no se encuentre la busqueda
9       Obtener la URL del post
10      Añadir la URL a la variable posts
11  Fin Bucle
12
13  Leer el archivo: "posts.txt"
14  Almacenar los datos del archivo en la variable guardados
15  Para i = 1 hasta el tamaño de posts
16      Añadir en guardados los nuevos posts
17
18  Abrir el archivo: "posts.txt"
19  Para i = 1 hasta el tamaño de guardados
20      Grabar los datos de guardados en el archivo
21
22  FIN
```

Figura 12. Pseudocódigo para obtener la URL de los posts

El segundo script es para obtener el contenido de todos los post. Aquí nos conectamos con cada URL del post y obtenemos la fecha, el contenido y los comentarios de cada uno (El código java se encuentra en el **ANEXO C**). A continuación

presentamos el pseudocódigo que ilustra los pasos de obtención del contenido:

```
INICIO

Variables: posts,result,guardados,total,arrayTotal
Leer el archivo: "posts.txt"
Almacenar los datos del archivo en la variable guardados

Para i = 1 hasta el tamaño de guardados
    Variables: comment,message,fecha,datalink,post_url
    Almacenar en post_url: guardados.get(i)
    Conectarse con la URL: post_url // URL del post
    Obtener la fecha del post (año;mes;dia)
    Almacenarla en la variable: fecha
    Obtener el contenido del post
    Almacenarlo en la variable: message
    Obtener el link publicado en el post
    Almacenarlo en la variable: datalink
    Obtener los comentarios del post
    Almacenarlos en la variable: comment
    Unir todo esto en la varible: total
    total: message + datalink + comment
    Dividir el total y guardarlo en arrayTotal
    Abrir el archivo: "resultado_posts.txt"
    Para j = 1 hasta el tamaño de arrayTotal
        Escribir en el archivo: fecha + arrayTotal[j]

FIN
```

Figura 13. Pseudocódigo para obtener el contenido de los posts

Aplicación de WordCount

Como de la página de facebook de la ESPOL obtenemos una gran cantidad de palabras las cuales se repiten muchas veces, para este tipo de casos usamos el WordCount, para que nos

agrupe en una sola palabra y el número de repeticiones de la misma.

WordCount es una sencilla aplicación que se encarga de analizar un texto, señalando en una lista: las palabras utilizadas y el número de repeticiones existentes.

El formato de los datos es:

Entrada: Archivo texto plano (.txt)

Salida: [Palabra] -> Clave
[#Repetición] -> Valor

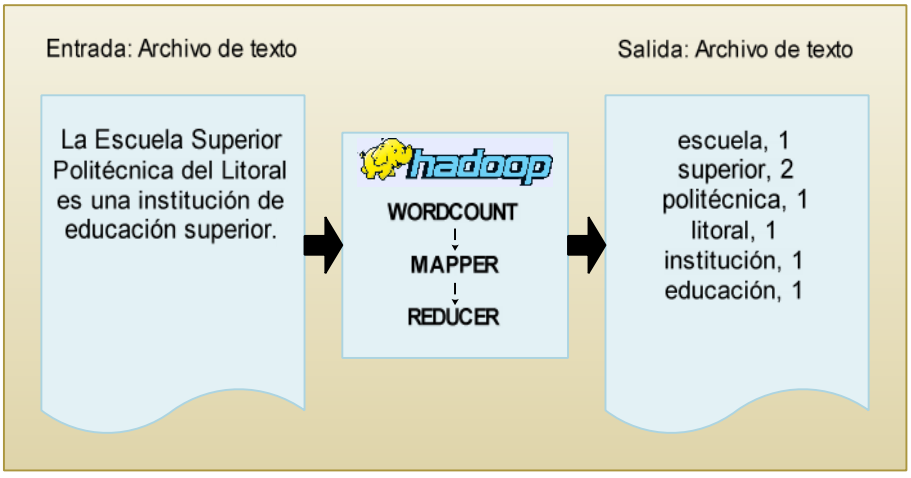

Figura 14. Aplicación del WordCount en un archivo de texto

La implementación del WordCountMapper, vía el método mapper, toma cada línea del archivo de texto, según lo dispuesto por el TextInputFormat. La línea es dividida en palabras (tokens) y emite pares clave/valor (palabra, 1).

La implementación del WordCountReducer, vía el método reducer, recibe la salida del mapper que contiene todas las palabras existentes en el archivo de entrada y cuenta la frecuencia de ellas emitiendo un nuevo par clave/valor (palabra,#repetición).

4.1.2. Estructura del Sistema de reportes y análisis

La estructura del sistema de reporte y análisis de tendencias, se la realizó de la siguiente manera:

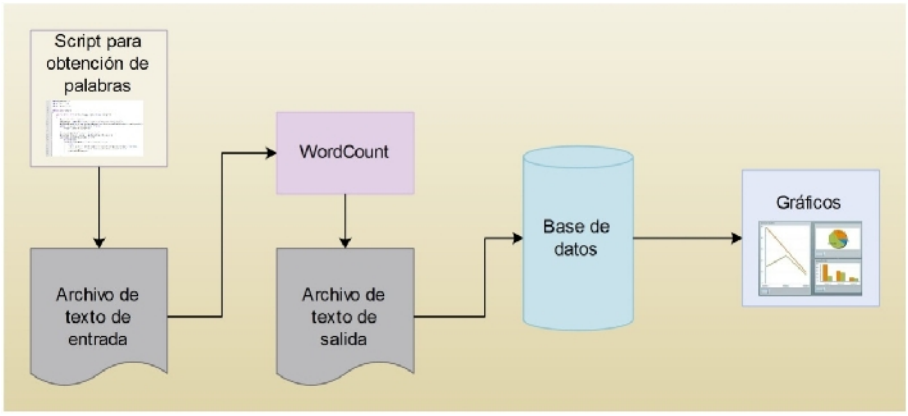

Figura 15. Estructura del sistema de reportes y análisis

4.2. Modelamiento de los Datos para su Procesamiento

Los datos para que se sean procesados correctamente tienen la siguiente estructura:

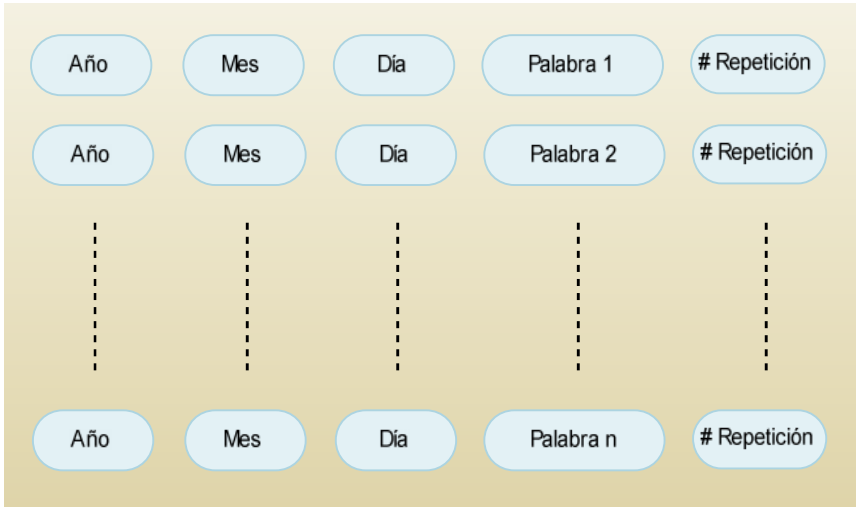

Figura 16. Estructura de los datos

Como un ejemplo de este esquema, tenemos el siguiente, donde se muestran 6 palabras diferentes con distintas fechas y con su número de repeticiones:

Figura 17. Ejemplo de la estructura de datos

4.3. Plan de Pruebas

Usando el Sistema de reportes y análisis sobre tendencias implementado, se hicieron la comparación de un máximo de 5 palabras usadas por los usuario de la comunidad de la universidad, mostrando cuales fueron las palabras más mencionadas por los estudiantes.

Para obtener un resultado actualizado, podemos seguir paso a paso las pruebas que se fueron realizando hasta obtener los resultados finales. Para eso se realizó lo siguiente:

a) Prueba del script de obtención de las palabras de la web de la ESPOL, esto abarca la adición de la fecha en que se publicó una palabra.
b) Prueba del WordCount de las palabras que se obtienen de la web, esto analizará y obtendrá el número de repeticiones de cada palabra.
c) Prueba de la subida a la base de datos MySQL de las palabras y sus repeticiones.
d) Prueba de obtener los datos de la base, desde Adobe Flex.
e) Prueba de la generación de los gráficos correspondientes.

Lo mostrado anteriormente fue el plan de pruebas desde el inicio hasta obtener el resultado final del sistema.

El siguiente paso fue analizar que palabras fueron las más mencionadas, para esto debemos ver los resultados del análisis de cada palabra en los gráficos de reportes, además podemos comparar entre palabras que pueden ser sinónimos o palabras completamente

diferentes, todos estos escenarios tendrán que estar presentes en la pruebas finales a realizarse, para obtener un informe detallado e ir comprobando cual tema es la tendencia actual.

4.4. Implementación del Sistema de reportes y análisis

4.4.1. Herramientas a Utilizarse

Geany

Figura 18. Logo de Geany

Usado para desarrollar el código en java. Es un editor de texto ligero basado en Scintilla con características básicas de entorno de desarrollo integrado (IDE).

Está disponible para distintos sistemas operativos, como GNU/Linux, Mac OS X, BSD, Solaris y Microsoft Windows. Es distribuido como software libre bajo la Licencia Pública General de GNU.

WordCount de Hadoop

WordCount lee archivos de texto y cuenta con que frecuencia ocurren las palabras. La entrada es archivos de texto y la salida es otro archivo de texto; donde el primero es un archivo donde se encuentran todas las palabras, y el segundo en cada línea tiene una palabra y el número de frecuencia de la misma.

Adobe Flex Builder

Figura 19. Logo de Flex

Flex es un marco de trabajo gratuito de código abierto y altamente productivo para crear aplicaciones web, para ordenadores de escritorio y para dispositivos móviles. Flex permite crear aplicaciones web y para dispositivos móviles que comparten una base de código común, lo que reduce el tiempo y el coste de creación de aplicaciones y el mantenimiento a largo plazo.

MySQL

Figura 20. Logo de MySQL

MySQL es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales.

MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interactuación con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos.

CAPITULO 5

5. Pruebas y Resultados

5.1. Ejecución de las Pruebas

Prueba del script para la obtención de los URL de los posts, aquí nos vamos a conectar a la página de Facebook de la ESPOL para obtener todas las URL de los contenidos publicados en su muro. El siguiente script a probar es dadas las URL nos conectamos a ellas para obtener todo el contenido de la publicación, sus comentarios y la fecha en que fue publicado, para luego almacenar todo eso en un archivo de texto.

Con ese archivo de texto utilizamos el WordCount para quitar las palabras repetidas y luego grabarlo en un archivo de texto con su número de repeticiones.

Después de todo ese proceso procedemos a grabar todos esos datos generados en la base de datos. Dichos datos los vamos a leer con la ayuda del Adobe Flex que nos generara los gráficos de reportes.

5.2. Análisis de los Resultados

Analizamos los datos de la generación del archivo de texto con todo ese proceso para garantizar un resultado óptimo.

Dado esto podemos tomar muestras de ciertas palabras para comprobar la existencia de las mismas y su número de repeticiones, además con el reporte de gráficos podemos tener más claridad, el cual es mostrado a continuación.

Ingreso de las palabras para su análisis

Esta es la pantalla gráfica en donde ingresaremos las palabras para su correspondiente análisis:

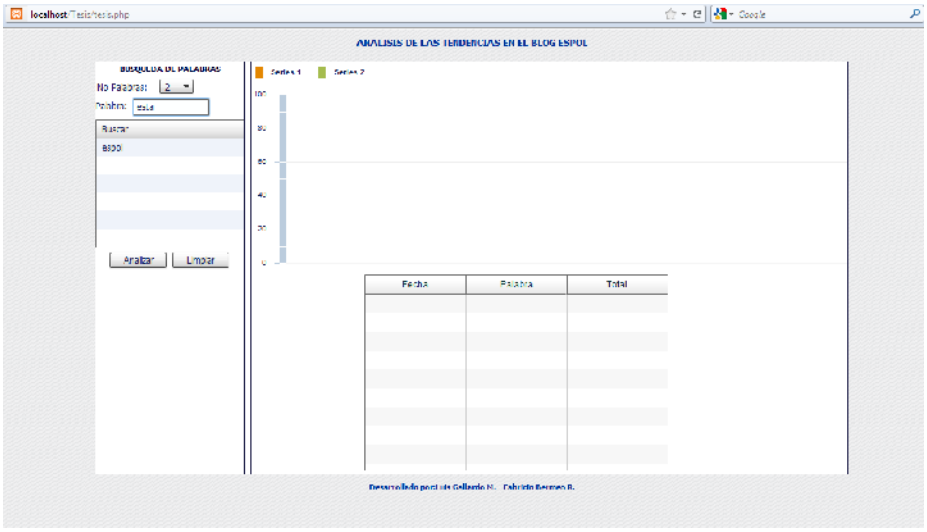

Figura 21. Ingreso de las palabras

Muestra de los resultados, agrupados por año y mes

Aquí nos muestra un gráfico de las tendencias de las palabras por año y por mes:

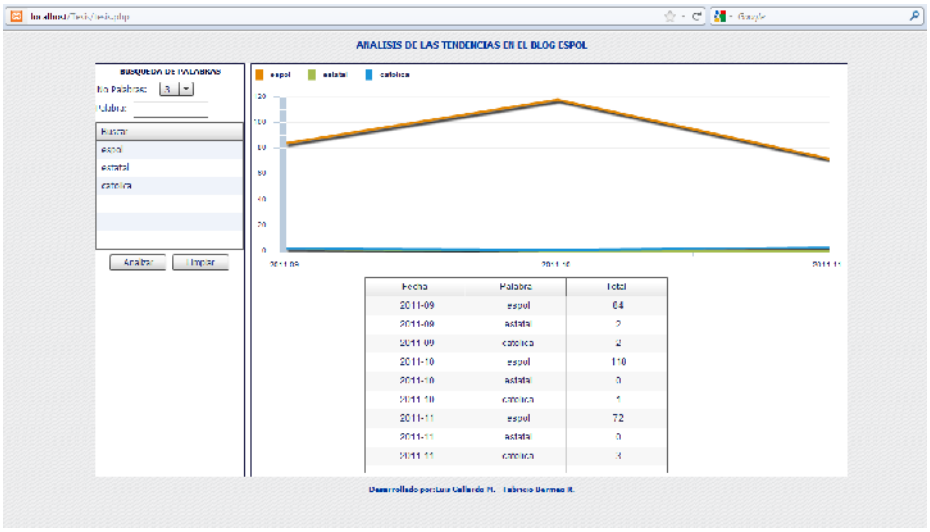

Figura 22. Gráfico de resultado por año y mes

Detalle Mensual

Aquí nos muestra un gráfico de las tendencias de las palabras del mes escogido:

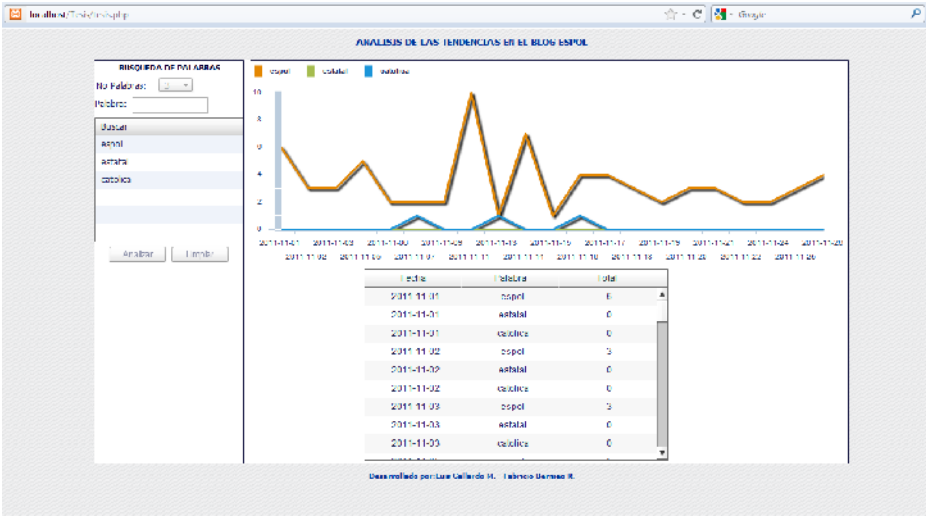

Figura 23. Gráfico de resultado mensual

CONCLUSIONES

Las conclusiones son:

1) Java nativo de Hadoop permite maximizar el excelente uso de recursos, para obtener el resultado más óptimo, pero sacrifica:
    - Facilidad en escritura de código
    - Tiempo empleado en la implementación de la solución

2) El procesamiento de archivos pequeños (en el orden de los KB) demora la tarea de ejecución debido a que Hadoop está desarrollado para ser más eficiente manipulando archivos de gran tamaño.

3) El sistema de reportes y análisis nos permitió conocer sobre los temas más comunes en la comunidad de la ESPOL de los últimos meses, ayudándonos a ver en detalle, con gráficos correspondientes que nos muestras las tendencias y estadísticas de éstos temas.

RECOMENDACIONES

Las recomendaciones son:

1) El procesamiento de grandes cantidades de datos y su posterior utilización en una tarea específica, recordando que es más sencillo manipular datos agrupados, que tener los datos dispersos.

2) Se propone realizar con un grupo de estudiantes pruebas, que cada uno comente seguidamente la página de facebook de la ESPOL, para luego tomar esos datos y procesarlos, para luego visualizar sus tendencias en un grafico estadístico.

3) Como trabajo a futuro se puede considerar éste proyecto como la base para la implementación de una aplicación al alcance del uso de los estudiantes para conocer lo común entre ellos.

4) La toma de los datos de la página deberían ser regularmente a diario, para obtener todos los posibles comentarios hechos en la misma, se podría hacer eso para mostrar unos resultados casi en tiempo real.

ANEXOS

ANEXO A

Página de facebook de la ESPOL

Figura 24. Anexo A, Página web de facebook de la ESPOL

ANEXO B

Script para obtener los posts de la página web de facebook de la ESPOL

```
import java.net.*;
import java.io.*;
import java.util.*;

public class GetPosts
{
	public static void main(String[] args) throws Exception
	{
		String entrada,result="";
		URL pagina = new URL("file:///home/luis/Descargas/espol.html");
		BufferedReader in = new BufferedReader(new
InputStreamReader(pagina.openStream()));
		while ((entrada = in.readLine()) != null)
			result = result + entrada;

		String temp = result;

		String busqueda = "\\/espol\\/posts\\/";
		ArrayList <String> posts = new ArrayList <String> ();
		if(result.indexOf(busqueda) >= 0){
			while(true){
			if(result.indexOf(busqueda) == -1)
				break;
				result =
result.substring(result.indexOf(busqueda)+busqueda.length());
				String num_post =
result.substring(0,result.indexOf("\\"));
				posts.add("http://www.facebook.com/espol/posts/" +
num_post);
			}
		}
		result = temp;
		busqueda = "photo.php?fbid=";
		ArrayList <String> postsPhotos = new ArrayList <String> ();
		if(result.indexOf(busqueda) >= 0){
			while(true){
				if(result.indexOf(busqueda) == -1)
					break;
				String str_photo =
result.substring(result.indexOf(busqueda));
				str_photo =
str_photo.substring(0,str_photo.indexOf("type=1")) + "type=1";
				result =
result.substring(result.indexOf(busqueda)+busqueda.length());
				if(!posts.contains(str_photo) &&
str_photo.contains("set=pt"))
					posts.add("http://www.facebook.com/" +
str_photo);
			}
		}

	  ArrayList <String> guardados = new ArrayList <String> ();
```

```
        /*** Lectura del archivo ***/
        FileReader fr = null;
        try {
                    File archivo = new File ("posts.txt");
            fr = new FileReader (archivo);
            BufferedReader br = new BufferedReader(fr);
            String linea;
            while((linea=br.readLine())!=null)
                           guardados.add(linea);
                    for(int i=0; i<posts.size(); i++){
                           if(!guardados.contains(posts.get(i)))
                           guardados.add(posts.get(i));
                    }
             }catch(Exception e){
             e.printStackTrace();
             }finally{
                    try{
                           if( null != fr )
                                  fr.close();
                    }catch (Exception e2){
                           e2.printStackTrace();
                    }
             }

             /*** Escritura del archivo ***/
             FileWriter fichero = null;
             try {
                    fichero = new FileWriter("posts.txt");
                    PrintWriter pw = new PrintWriter(fichero);
                    for(int i=0; i<guardados.size(); i++){
                           //System.out.println(""/*guardados.get(i)*/);
                           pw.println(guardados.get(i));
                    }
        }catch(Exception e){
             e.printStackTrace();
        }finally {
                    try {
                           if (null != fichero)
                                  fichero.close();
                    }catch (Exception e2) {
                           e2.printStackTrace();
           }
        }

             in.close();
    }
}
```

ANEXO C

Script para obtener el contenido de los posts de la página

```
import java.net.*;
import java.io.*;
import java.util.*;

public class GetContenido
{
	public static void main(String[] args) throws Exception
	{
		String entrada;
		ArrayList <String> guardados = new ArrayList <String> ();

	 /*** Lectura del archivo ***/
	 FileReader fr = null;
	 try {
			File archivo = new File ("posts.txt");
		    fr = new FileReader (archivo);
		    BufferedReader br = new BufferedReader(fr);
		    String linea;
		    while((linea=br.readLine())!=null)
				guardados.add(linea);
		     }catch(Exception e){
		     e.printStackTrace();
		     }finally{
				try{
					if( null != fr )
						fr.close();
				}catch (Exception e2){
					e2.printStackTrace();
				}
		     }
		     String data="";
		     FileWriter fichero = null;
		     try {
				fichero = new FileWriter("resultado_posts.txt");
				PrintWriter pw = new PrintWriter(fichero);
				for(int i=0; i<guardados.size(); i++){
					String
result="",comment="",message="",fecha="",datalink="";
					String post_url = guardados.get(i);
					//String post_url =
"http://www.facebook.com/espol/posts/316634881707935";
					System.out.println(post_url);
					BufferedReader in = null;
					try{
						URL pagina = new URL(post_url);
						in = new BufferedReader(new
InputStreamReader(pagina.openStream()));
					}catch (Exception e){
						System.out.println("Error de Lectura: " +
post_url);
						break;
					}
```

```
                        while ((entrada = in.readLine()) != null)
                                result = result + entrada;

                        String searchMessage = "messageBody";
                        String searchDate = "uiStreamSource";
                        if (post_url.indexOf("photo.php") > 0){
                                searchMessage = "fbPhotosPhotoCaption";
                                searchDate = "commentActions";
                        }

                        // Fecha del post
                        if(result.indexOf(searchDate) >= 0){
                                fecha =
result.substring(result.indexOf(searchDate)+searchDate.length());
                                fecha =
fecha.substring(fecha.indexOf("title=")+"title=".length());
                                fecha =
fecha.substring(0,fecha.indexOf("data"));
                                fecha = fecha.substring(0,fecha.indexOf("a
la"));
                                fecha = fecha.replaceAll(" |\"","");
                                fecha = fecha.replaceAll("de|,",";");
                                String [] arrFecha = fecha.split(";");
                                String dia = arrFecha[1];
                                String mes = getNumeroMes(arrFecha[2]);
                                String anio = arrFecha[3];
                                fecha = anio + ";" + mes + ";" + dia;
                        }

                        // Contenido del post
                        if(result.indexOf(searchMessage) >= 0){
                                message =
result.substring(result.indexOf(searchMessage));
                                message =
message.substring(message.indexOf(">")+1);
                                message =
message.substring(0,message.indexOf("<"));
                        }

                        // Link del post
                        if(result.indexOf("uiAttachmentDesc") >= 0){
                                String cad = "";
                                String result_link = result;
                                while(true){

      if(result_link.indexOf("uiAttachmentDesc") == -1)
                                                break;
                                        result_link =
result_link.substring(result_link.indexOf("uiAttachmentDesc")+"uiAttachmentD
esc".length());
                                        datalink =
result_link.substring(result_link.indexOf(">")+1);
                                        datalink =
datalink.substring(0,datalink.indexOf("<"));
                                }
                        }

                        // Comentarios del post
                        if(result.indexOf("commentBody") >= 0){
```

```
                                    String cad = "";
                                    String result_com = result;
                                    while(true){
                                          if(result_com.indexOf("commentBody")
== -1)
                                                break;
                                          cad =
result_com.substring(result_com.indexOf("commentBody"));
                                          cad =
cad.substring(cad.indexOf(">")+1);
                                          result_com =
cad.substring(cad.indexOf("<"));
                                          cad =
cad.substring(0,cad.indexOf("<"));
                                          cad = cad.replaceAll("<br />","");
                                          comment = comment + cad + ",";
                                    }
                              }

                              // Unir todo el contenido
                              if(fecha.length() > 0){
                                    String total = message + " " + datalink + "
" + comment;
                                    total = total.toLowerCase();
                                    total =
total.replaceAll("los|las|que|una|más|del|con|como|sus|son|este|esto|para|ht
tp|www|por|&|amp|quot|!|¡|[?]|[¿]|<br />", "");
                                    total = total.replaceAll(" |#|[.]|[;]|[_]|–
|-|[|]|[*]|[\"]|[/]|[“]|[”]|[¨]|[¨]|[´]|[’]|[:]|[(]|[)]|[’]|‘|’|%|[+]",
",");
                                    total = total.replaceAll("&#64;","@");
                                    total = total.replaceAll("á","a");
                                    total = total.replaceAll("é","e");
                                    total = total.replaceAll("í","i");
                                    total = total.replaceAll("ó","o");
                                    total = total.replaceAll("ú","u");
                                    String [] arrayTotal = total.split(",");
                                    for(int j=0; j<arrayTotal.length; j++){
                                          if(arrayTotal[j].length() > 2 &&
!(isNumeric(arrayTotal[j])))

      pw.println(fecha+";"+arrayTotal[j]);
                                    }
                              }
                        in.close();
                  }
                  }catch(Exception e){
                  e.printStackTrace();
            }finally {
                        try {
                              if (null != fichero)
                                    fichero.close();
                        }catch (Exception e2) {
                              e2.printStackTrace();
               }
            }
      }

      /*** Dado el nombre de un mes obtener su numero ***/
      private static String getNumeroMes(String mes){
```

```
            String [] meses = new String[]
{"enero","febrero","marzo","abril","mayo","junio",

      "julio","agosto","septiembre","octubre","noviembre","diciembre"};
            for(int i=0; i<meses.length; i++){
                  if(meses[i].equalsIgnoreCase(mes))
                        return i+1+"";
            }
            return "";
      }

      /*** Comprobar si una cadena es numero ***/
      private static boolean isNumeric(String cadena){
            try {
                  cadena = cadena.replaceAll(" ","u");
                  Integer.parseInt(cadena);
                  return true;
            } catch (NumberFormatException nfe){
                  return false;
            }
   }
}
```

ANEXO D

Instalación y Configuración de Hadoop

Requisitos previos

- **Plataformas compatibles**
  - GNU / Linux se admite como una plataforma de desarrollo y producción. Hadoop ha sido demostrada en GNU / Linux clústeres con 2000 nodos.
  - Win32 se admite como una plataforma de desarrollo. La operación distribuida no ha sido bien probada en Win32, por lo que no se admite como una plataforma de producción.
- **Software necesario**

  Software necesario para Linux y Windows son:

  1. Java 1.6.x, preferiblemente de Sun, debe estar instalado.
  2. ssh debe estar instalado y sshd debe estar en ejecución para utilizar los scripts de Hadoop que gestionan a distancia los demonios Hadoop.

  Requisitos adicionales para Windows son:

  1. Cygwin - necesario para capa de soporte, además del software requerido anteriormente.

- **Instalación del software**

  Si el clúster no tiene el software requerido tendrá que instalarlo.

  Por ejemplo, en Ubuntu Linux:

  ```
  $ sudo apt-get install ssh
  $ sudo apt-get install rsync
  ```

  En Windows, si no ha instalado el software necesario durante la instalación de cygwin, inicie la instalación de cygwin y seleccione los paquetes:

  - openssh - la categoría Net

Preparar para iniciar el clúster Hadoop

Desempaquetar el archivo descargado de distribución de Hadoop (http://hadoop.apache.org/common/releases.html). En la distribución, editar el fichero `conf/hadoop-env.sh` para definir el `JAVA_HOME` con el lugar de la instalación de Java.

Pruebe el siguiente comando:

```
$ bin/hadoop
```

Esto mostrará la documentación usada por el script de **hadoop**.

Ahora ya está listo para iniciar su clúster Hadoop en uno de los tres modos disponibles:

- Modo Local (independiente)

- Modo Pseudo-distribuido
- Modo totalmente-distribuido

Funcionamiento Independiente

Por defecto, Hadoop está configurado para ejecutarse en un modo no distribuido, como un solo proceso de Java, esto es útil para la depuración.

El ejemplo siguiente copia el directorio descomprimido `conf` a utilizar como entrada, luego busca y muestra todos los emparejamientos de la expresión regular dada. La salida se escribe en el directorio de `output` dado.

```
$ mkdir input
$ cp conf/*.xml input
$ bin/hadoop jar hadoop-examples-*.jar grep input output 'dfs[a-z.]+'
$ cat output/*
```

Funcionamiento Pseudo-distribuido

Hadoop también se puede ejecutar en un solo nodo en un modo de pseudo-distribuido donde cada demonio Hadoop se ejecuta en un proceso separado de Java.

Configuración

Utilice lo siguiente:

conf/core-site.xml:

```
<configuration>
     <property>
         <name>fs.default.name</name>
         <value>hdfs://localhost:9000</value>
     </property>
</configuration>
```

conf/hdfs-site.xml:

```
<configuration>
     <property>
         <name>dfs.replication</name>
         <value>1</value>
     </property>
</configuration>
```

conf/mapred-site.xml:

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

Configuración de ssh passphraseless

Ahora comprobamos que se puede hacer ssh al localhost sin contraseña:

```
$ ssh localhost
```

Si no puedes hacer ssh a localhost sin contraseña, ejecute los siguientes comandos:

```
$ ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa
$ cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
```

Ejecución

Dar formato a un nuevo sistema de archivos distribuido:

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

Iniciar los demonios Hadoop:

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

El demonio hadoop salida del registro se escriben en el directorio `${HADOOP_LOG_DIR}` (por defecto es `${HADOOP_HOME}/logs`).

Navegar por la interfaz web para el NameNode y la JobTracker, por defecto están disponibles en:

- `NameNode` - http://localhost:50070/
- `JobTracker` - http://localhost:50030/

Copie los archivos de entrada en el sistema de archivos distribuido:

```
$ bin/hadoop fs -put conf input
```

Ejecutar algunos de los ejemplos:

```
$ bin/hadoop jar hadoop-examples-*.jar grep input output 'dfs[a-z.]+'
```

Examine los archivos de salida:

Copie los archivos de salida desde el sistema de archivos distribuidos al sistema de archivos locales y examinarlas:

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

o

Ver los archivos de salida del sistema de archivos distribuido:

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

Cuando hayas terminado, detener los demonios con:

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

Operación totalmente distribuida

Para más información sobre la configuración totalmente distribuida, clusteres no triviales ver:

(http://hadoop.apache.org/common/docs/stable/cluster_setup.html).

ANEXO E

Código Flex, presentación gráficos y tablas.

```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
                      xmlns:s="library://ns.adobe.com/flex/spark"
                      xmlns:mx="library://ns.adobe.com/flex/mx"
width="100%" height="99%" creationComplete="inicializaGrafico()" >
       <fx:Style source="fabricio.css"/>

       <fx:Declarations>
              <mx:HTTPService id="cnxPhp" result="cargaDatos(event)"
fault="errorDatos(event)"
                                     useProxy="false" url =
"http://localhost/Tesis/prueba.php" resultFormat="e4x"
                                     showBusyCursor="true" method="POST"/>

              <mx:SeriesSlide id="slideIn" duration="1500" direction="left"/>
              <mx:SeriesSlide id="slideOut" duration="1500"
direction="right"/>

              <mx:Parallel id="showEffects">
                     <mx:WipeRight duration="1000"/>
                     <mx:Fade alphaFrom="0" alphaTo="1" duration="2000"/>
              </mx:Parallel>

              <mx:Parallel id="hideEffects">
                     <mx:Fade alphaFrom="1" alphaTo="0" duration="1000"/>
                     <mx:WipeUp duration="1000"/>
              </mx:Parallel>


       </fx:Declarations>


       <fx:Script>
              <![CDATA[
                     import mx.charts.HitData;
                     import mx.charts.series.LineSeries;
                     import mx.charts.series.items.LineSeriesItem;
                     import mx.collections.ArrayCollection;
                     import mx.controls.*;
                     import mx.rpc.events.FaultEvent;
                     import mx.rpc.events.ResultEvent;
                     import mx.rpc.remoting.RemoteObject;
                     import mx.utils.ObjectProxy;

                     import org.osmf.traits.IPlayable;

                     [Bindable] public var arr:ArrayCollection = new
ArrayCollection();
                     private var titulo:String = "ERROR";
                     [Bindable] public var total_registros:int = 0;
                     private var num_graficas:int = 0;
                     [Bindable] public var resultado:ArrayCollection = new
ArrayCollection();
```

```
                  [Bindable] public var arr_tabla:ArrayCollection = new
ArrayCollection();
                  private var array_resultado:Array = new Array();
            /*-------------------------------------------------------------
---*/
                  private function inicializaGrafico():void{
            /*-------------------------------------------------------------
---*/


      txt_palabra.addEventListener(KeyboardEvent.KEY_DOWN, keyHandler);
                        //var par:Object = new Object();
                        //par.id = "";
                        //cnxPhp.send(par);
                        total_registros = cb3.selectedItem.label;
                  }//endfunction



            /*-------------------------------------------------------------
--*/
                  private function cargaDatos(res:ResultEvent):void{
            /*-------------------------------------------------------------
*/

                        var xml:XML = res.result as XML;
                        var xmllist:XMLList;
                        var i:int = 0;
                        var flag:Boolean = false;
                        var arr_palabras:ArrayCollection = new
ArrayCollection();
                        arr_tabla.removeAll();
                        resultado.removeAll();
                        for each (var element:XML in xml.elements()){
                              //var oElemento:Object = new Object();
                              //oElemento.nombre = element.;
                              //arr_palabras.addItem(oElemento);
                              flag = true;
                              var fecha:String = "";
                              fecha = element.@id;
                              xmllist = element.children();
                              i = 0;
                              while(i < xmllist.length()){
                                    var xmlnode:XML = xmllist[i];
                                    var palabra:String = xmlnode.@id;
                                    var valor:String = xmlnode.@valor;
                                    actualizaCampos(palabra,valor);
                                    i++;
                              }
                              llenaDataProvider(fecha);
                              setFlagPalabras();
                        }
                        hbox_principal.visible = flag;
                        if(flag == true){
                              creaLineas();
                        }else{
                              Alert.show("No existen Datos para su
Busqueda","A L E R T A");
                        }
```

```
                }

            /*-------------------------------------------------------------
-----------------------*/
                private function
actualizaCampos(palabra:String,valor:String):void{
            /*-------------------------------------------------------------
-----------------------*/
                        var i:int = 0;
                        var len:int = 0;
                        var pal:String = "";
                        len = arr.length;
                        while(i < len){
                                pal = arr.getItemAt(i).palabra;
                                if(pal == palabra){
                                        arr.getItemAt(i).flag = "1";
                                        arr.getItemAt(i).numveces = valor;
                                        break;
                                }
                                i++;
                        }

                }

            /*-------------------------------------------------------------
--------------*/
                private function llenaDataProvider(fecha:String):void{
            /*-------------------------------------------------------------
--------------*/
                        var obj:Object = new Object();
                        var obj_tabla:Object;
                        var i:int = 0;
                        var len:int = 0;
                        len = arr.length;
                        var tipo:String = "";
                        while(i < len){
                                obj_tabla = new Object();
                                obj_tabla.fecha   = fecha;
                                obj_tabla.palabra =
arr.getItemAt(i).palabra;
                                obj_tabla.total   =
arr.getItemAt(i).numveces;

                                tipo = arr.getItemAt(i).variable;
                                switch(tipo){
                                        case "palabra1":
                                                obj.palabra1 =
arr.getItemAt(i).numveces;
                                                break;
                                        case "palabra2":
                                                obj.palabra2 =
arr.getItemAt(i).numveces;
                                                break;
                                        case "palabra3":
                                                obj.palabra3 =
arr.getItemAt(i).numveces;
                                                break;
                                        case "palabra4":
                                                obj.palabra4 =
arr.getItemAt(i).numveces;
```

```
                                          break;
                                    case "palabra5":
                                          obj.palabra5 =
arr.getItemAt(i).numveces;
                                          break;
                                    }
                                    arr_tabla.addItem(obj_tabla);
                                    i++;
                              }
                              obj.fecha = fecha;
                              resultado.addItem(obj);
                        }

                  /*-------------------------------------------------------------
---------*/
                        private function setFlagPalabras():void{
                  /*-------------------------------------------------------------
---------*/
                              var i:int = 0;
                              var len:int = 0;
                              while(i < len){
                                    arr.getItemAt(i).flag = "0";
                                    i++;
                              }


                        }

                  /*-------------------------------------------------------------
------------*/
                        private function creaLineas():void{
                  /*-------------------------------------------------------------
------------*/

                              var array:Array = this.linechart1.series;
                              var len:int = array.length;
                              var i:int = 0;
                              while(len > i){
                                    array.pop();
                                    len--;
                              }//endwhile
                              anadeColumna(array);
                        }

                  /*-------------------------------------------------------------
-----------*/
                        public function anadeColumna(array:Array):void{
                  /*-------------------------------------------------------------
----------*/
                              var i:int = 0;
                              var len:int = 0;
                              len = arr.length;
                              var tipo:String = "";
                              while(i < len){
                                    var columna:LineSeries = new LineSeries();
                                    var j:int = 0;
                                    tipo = arr.getItemAt(i).variable;

                                    switch(tipo){
                                          case "palabra1":
```

```
                                                columna.yField =
"palabra1";
                                                break;
                                          case "palabra2":
                                                columna.yField =
"palabra2";
                                                break;
                                          case "palabra3":
                                                columna.yField =
"palabra3";
                                                break;
                                          case "palabra4":
                                                columna.yField =
"palabra4";
                                                break;

                                          case "palabra5":
                                                columna.yField =
"palabra5";
                                                break;
                                    }// end switch

                                    columna.displayName =
arr.getItemAt(i).palabra;
                                    columna.setStyle("showDataEffect",slideIn);

      columna.setStyle("hideDataEffect",slideOut);
                                    columna.setStyle("fill","#006699");
                                    columna.setStyle("stroke","#FF6600");
                                    array.push(columna);


                                    i++;
                              }//end while
                              linechart1.dataProvider   = resultado;

                              linechart1.series               = array;

                              tabla_datos.dataProvider  = arr_tabla;
                        }


                  /*-------------------------------------------------------------
--*/
                        private function errorDatos(res:FaultEvent):void{
                  /*-------------------------------------------------------------
*/
                              Alert.show(res.fault.toString());
                        }

                  /*-----------------------------------------------------*/
                        private function focusInput():void{
                  /*-----------------------------------------------------*/
                              focusManager.setFocus(txt_palabra);
                        }

                  /*-----------------------------------------------------*/
                        private function ingresaPalabra():void{
                  /*-----------------------------------------------------*/
                              var obj:Object = new Object();
```

```
                        if(txt_palabra.text != ""){
                              if(verificaTotalIngresos()){

      if(validaPalabraRepetida(txt_palabra.text)){
                                          var len:int = 0;
                                          len = arr.length + 1;
                                          obj.palabra =
txt_palabra.text.toLowerCase();
                                          obj.variable = "palabra"+len;
                                          obj.flag     = "0";
                                          obj.numveces = "0";
                                          arr.addItem(obj);
                                          txt_palabra.text = "";

                                    }else{

      Alert.show('Palabra'+txt_palabra.text+' ya fue Ingresada',titulo);
                                          }
                                    }else{
                                          Alert.show('Supero el Max de Palabras
',titulo);
                                    }
                              }else{
                                    Alert.show('Ingrese una Palabra',titulo);
                              }
                              focusInput();
                        }

            /*-------------------------------------------------------------
------*/
                  private function
validaPalabraRepetida(ipalabra:String):Boolean{
            /*-------------------------------------------------------------
-------*/
                        var flag:Boolean = true;
                        var i:int = 0;
                        var len:int = 0;
                        var palabra:String = "";
                        len = arr.length;
                        while(i < len){
                              palabra = arr.getItemAt(i).palabra;
                              if(ipalabra == palabra){
                                    flag = false;
                                    break;
                              }
                              i++;
                        }

                        return flag;

                  }

            /*-----------------------------------------------------*/
                  private function prueba():void{
            /*-----------------------------------------------------*/


                        //Alert.show('figura');
```

```
		}
		/*-------------------------------------------------------------
--------*/
		private function keyHandler(event:KeyboardEvent):void {
		/*-------------------------------------------------------------
-------*/
			if(event.keyCode == 13){
				ingresaPalabra();
			}//endif
		}
		/*-------------------------------------------------------------
----------*/
		private function enviaConsulta():void{
		/*-------------------------------------------------------------
----------*/
			var i:int = 0;
			var len:int = 0;
			len = arr.length;
			var s:String = "";
			while(i < len){
				s += arr.getItemAt(i).palabra+';';
				i++;
			}

			var par:Object = new Object();
			par.id = s;
			cnxPhp.send(par);

		}
		/*-------------------------------------------------------------
----------------*/
		private function changeEvt(event:Event):void {
		/*-------------------------------------------------------------
----------------*/
			var len_tabla:int = arr.length + 1;
			var max_palabra:int = cb3.selectedItem.label;
			if(len_tabla > max_palabra){
				Alert.show('El maximo de palabras a
Ingresar NO debe ser Menor al total de palabras Ingresadas',titulo);
			}else{
				total_registros
=event.currentTarget.selectedItem.label;
				verificaTotalIngresos();
			}
		}

		/*-------------------------------------------------------------
----------------*/
		private function verificaTotalIngresos():Boolean{
		/*-------------------------------------------------------------
----------------*/
			var flag:Boolean = true;
			var len_tabla:int = arr.length + 1;
			if(arr.length >= total_registros){
				flag = false;
			}
			return flag;
		}
		/*-------------------------------------------------------------
----------------*/
```

```
                  private function limpiaTabla():void {
            /*-------------------------------------------------------------
----------------*/
                        arr.removeAll();
                        txt_palabra.text = "";
                  }
            /*-------------------------------------------------------------
-------------*/
                  private function
lcDataTipFunction(hitData:HitData):String {
            /*-------------------------------------------------------------
-------------*/
                        var s:String;
                        var cabecera:String = "";
                        cabecera =
LineSeries(hitData.element).displayName;
                        cabecera = cabecera.toUpperCase();
                        s = "<b>"+cabecera+"</b>\n";
                        s += "<b>Fecha: </b>" + hitData.item.fecha+ "\n";

                        var tipo:String = "";
                        tipo = LineSeries(hitData.element).yField;
                        switch(tipo){
                              case "palabra1":
                                    s += "<b>Veces: </b>" +
hitData.item.palabra1 + "\n";
                                    break;
                              case "palabra2":
                                    s += "<b>Veces: </b>" +
hitData.item.palabra2 + "\n";
                                    break;
                              case "palabra3":
                                    s += "<b>Veces: </b>" +
hitData.item.palabra3 + "\n";
                                    break;
                              case "palabra4":
                                    s += "<b>Veces: </b>" +
hitData.item.palabra4 + "\n";
                                    break;
                              case "palabra5":
                                    s += "<b>Veces: </b>" +
hitData.item.palabra5 + "\n";
                                    break;
                        }

                        return s;
                  }

            ]]>
      </fx:Script>

      <mx:VBox width="100%" height="100%" color="#1D2556">
            <mx:HBox x="0" y="30" width="100%" height="100%">
                  <mx:HBox width="20%" height="100%" fontFamily="Tahoma,
Geneva, sans-serif" fontSize="12" borderVisible="true" borderStyle="solid"
borderColor="#1D1E48">
                        <mx:VBox id="panel_busqueda" x="0" y="0"
width="100%" height="50%">
                              <mx:HBox width="100%" height="10%"
paddingTop="5">
```

```
                                        <s:Label width="100%" height="100%"
text="BUSQUEDA DE PALABRAS" fontSize="10" fontWeight="bold" color="#000039"
textAlign="center"/>
                                </mx:HBox>
                                <mx:HBox>
                                        <mx:FormItem label="No Palabras:">
                                                <mx:ComboBox id="cb3"
selectedIndex="0" change="changeEvt(event)">
                                                        <mx:dataProvider>
                                                                <fx:Array>
                                                                        <fx:Object
label="2" />
                                                                        <fx:Object
label="3" />
                                                                        <fx:Object
label="4" />
                                                                        <fx:Object
label="5" />

                                                                </fx:Array>
                                                        </mx:dataProvider>
                                                </mx:ComboBox>
                                        </mx:FormItem>
                                </mx:HBox>
                                <mx:HBox>
                                        <s:Label text="Palabra:"
paddingTop="5"  height="22"/>
                                        <s:TextInput id="txt_palabra"
width="100" restrict="A-Z\\a-z"/>
                                        <!--<s:Button id = "btn_ok"
label="Ok" width="50" click="ingresaPalabra()"/>-->
                                </mx:HBox>

                                <mx:HBox width="100%">
                                        <mx:DataGrid id="dg_palabra" x="24"
y="35" dataProvider="{arr}" width="100%"

itemRollOver="prueba()" >
                                                <mx:columns>
                                                        <mx:DataGridColumn
headerText="Buscar" dataField="palabra"  />

                                                </mx:columns>
                                        </mx:DataGrid>
                                </mx:HBox>
                                <mx:HBox width="100%"
horizontalAlign="center">
                                        <s:Button id = "btn_analizar"
label="Analizar" width="75" click="enviaConsulta()"/>
                                        <s:Button id = "btn_limpiar"
label="Limpiar" width="75" click="limpiaTabla()"/>
                                </mx:HBox>
                        </mx:VBox>

                </mx:HBox>
                <mx:VBox id="hbox_principal" visible="false" width="80%"
height="100%" borderColor="#1D1E48" borderStyle="solid">
                        <mx:VBox width="100%" height="51%">
                                <mx:VBox width="100%" height="100%">
```

```
                                        <mx:Legend
dataProvider="{linechart1}" width="100%" height="10%"/>
                                        <mx:LineChart width="100%"
height="90%" id="linechart1" showDataTips="true"

creationCompleteEffect="showEffects"

dataTipFunction="lcDataTipFunction"

showEffect="showEffects">
                                              <mx:horizontalAxis>
                                                    <mx:CategoryAxis id="a1"
categoryField="fecha" />
                                              </mx:horizontalAxis>
                                              <mx:series>
                                                    <!--
                                                    <mx:LineSeries
id="palabra1" displayName="Series 1" yField="numveces"

showDataEffect="{slideIn}"

hideDataEffect="{slideOut}"/>
                                              <mx:LineSeries
id="palabra2" displayName="Series 2" yField="numveces"

showDataEffect="{slideIn}"

hideDataEffect="{slideOut}"/>
                                                    -->
                                              </mx:series>
                                        </mx:LineChart>
                                  </mx:VBox>

                            </mx:VBox>
                            <mx:VBox width="70%" height="48%"
textAlign="center">
                                  <!--
                                  <mx:DataGrid id="dgUserRequest"
width="100%" height="100%"  sortableColumns="true">
                                        <mx:columns>
                                              <mx:DataGridColumn
headerText="Palabra" dataField="nombre" />
                                              <mx:DataGridColumn
headerText="Veces" dataField="numveces"/>
                                              <mx:DataGridColumn
headerText="Fecha" dataField="fecha"/>
                                        </mx:columns>
                                  </mx:DataGrid>
                                  -->
                                  <mx:AdvancedDataGrid id="tabla_datos"

height="100%" width="100%"

sortableColumns="true"

sortExpertMode="true" textAlign="center" >
                                        <mx:GroupingCollection2 id="gc">
                                              <mx:Grouping label="fecha">
                                                    <mx:GroupingField
name="fecha">
```

```
                                                                <mx:summaries>

        <mx:SummaryRow summaryPlacement="group">

        <mx:fields>

        <mx:SummaryField2 dataField="total" summaryOperation="SUM" />



        </mx:fields>

        </mx:SummaryRow>
                                                                </mx:summaries>
                                                            </mx:GroupingField>
                                                        </mx:Grouping>
                                                    </mx:GroupingCollection2>


                                                    <mx:columns>

                                                        <mx:AdvancedDataGridColumn
width="50" headerText="Fecha" sortable="true" dataField="fecha"
showDataTips="true"/>
                                                        <mx:AdvancedDataGridColumn
width="50" headerText="Palabra" id="ht2" sortable="true" dataField="palabra"
showDataTips="true"/>
                                                        <mx:AdvancedDataGridColumn
width="50" headerText="Total" id="ht3" textAlign="center" sortable="true"
dataField="total"  />

                                                    </mx:columns>

                                                </mx:AdvancedDataGrid>

                                            </mx:VBox>
                                        </mx:VBox>
                                    </mx:HBox>
        </mx:VBox>

</s:Application>
```

ANEXO F

Código PHP, recibe la petición desde Flex y retorna un XML.

```
<?php
		$par = $_POST["id"];
		$sql = "Select palabra,sum(numveces) as numveces,anio,mes,dia FROM tesis\n";
		if($par == null){
			$sql .= "ORDER BY numveces DESC  LIMIT 0,5 ";
		}else{
			$sql .= "WHERE ";
			$pal = explode(";",$par);
			$i = 0;

			while($i < count($pal)){
				if($pal[$i] <> ""){
					$sql .= "palabra = '".$pal[$i]."' or ";
				}
				$i++;
			}

			$sql = substr($sql,0,strlen($sql)-4);
			$sql .= " group by palabra,anio,mes\n";
			$sql .= "order by anio,mes,dia asc";
		}

		$conexion = mysql_connect("localhost", "root", "") or die(mysql_error());
		 mysql_select_db("test", $conexion);

		$result = mysql_query($sql, $conexion);
		$pal = "<?xml version ='1.0' encoding = 'utf-8' ?>\n";
		$pal  .= "<tesis>\n";
		$v_archivo		= "archivo.xml";
		$ar=fopen($v_archivo,"w") or die("Problemas en la creacion");
		$palabra_ant = "";
		$fecha_ant = "";
		$i = 0;
		$mes = "";
		$dia = "";
		while($row = mysql_fetch_array($result)){
			$mes = $row['mes'];
			$dia = $row['dia'];
			if(strlen($mes) == 1){
				$mes = "0".$row['mes'];
			}

			if(strlen($dia) == 1){
				$dia = "0".$row['dia'];
			}
			$fecha = $row['anio']."-".$mes;
			$i++;

			if($fecha_ant != ""){
				if($fecha != $fecha_ant){
					$pal  .= "		</fecha>\n";
					$pal  .= "		<fecha id = '".$fecha."'>\n";
```

```
                                        $pal  .= "                    <palabra id = "'.$row['palabra'].'"
        valor = "'.$row['numveces'].'">0</palabra>\n";
                                        $fecha_ant = $fecha;


                                    }else{
                                        $pal  .= "                    <palabra id = "'.$row['palabra'].'"
valor = "'.$row['numveces'].'">0</palabra>\n";
                                        $fecha_ant = $fecha;
                                    }

                                }else{
                                    //echo $i;
                                    $pal  .= "          <fecha id = "'.$fecha.'">\n";
                                    $pal  .= "                    <palabra id = "'.$row['palabra'].'" valor =
"'.$row['numveces'].'">0</palabra>\n";
                                    $fecha_ant = $fecha;
                                }
                            }
                            $pal  .= "          </fecha>\n";
                            $pal  .= "</tesis>\n";

                            fputs($ar,$pal."\n");
                            fclose($ar);
                            echo $pal;
```

ANEXO G

Código PHP, presentación de la página principal.

```
<html>
	<head>
		<title>:::: TESIS ::::</title>

		<style type="text/css" media="screen">
			html, body {
				height:100%;
			}
			body {
				margin:0;
				padding:0;
				overflow:hidden;
				background-image:url(imagenes/fondo.gif);
				background-repeat:repeat-x;s
			}
			h2{
				text-align:center;
				font-size:13px;
				font-weight:bold;
				color:#039;
				font-family:Tahoma, Geneva, sans-serif;
				text-transform:uppercase;
			}

			p{
				text-align:center;
				font-size:10px;
				font-weight:bold;
				color:#039;
				font-family:Tahoma, Geneva, sans-serif;

			}
		</style>

		<script type="text/javascript" src="js/swfobject.js"></script>
		<script type="text/javascript">
			swfobject.embedSWF("bin-debug/fabricio.swf", "miFlash", "1000", "550",
"8.0.0", "js/expressInstall.swf", {}, {menu:"false", scale:"noscale", wmode:"opaque"}, {} );

		</script>

	</head>
	<body>
		<div id = "titulo" style = "width:100%;height:5%">
			<h2>Analisis de las Tendencias en el Blog Espol</h2>
		</div>
		<div align="center">
			<div align="center" id="miFlash"
style="float:left;width:100%;height:85%;border-color:#333333"></div>
			<!--<div id="miFlash2" style="float:right;width:50%;height:90%"></div>-->
		</div>
	<div id = "pie" style = "width:100%;height:5%">
```

```
                    <p>Desarrollado por:Luis Guillermo Gallardo - Fabricio Bermeo R.</p>
                </div>
        </body>
</html>
```

BIBLIOGRAFÍA

[1] Wikipedia, Hadoop, http://es.wikipedia.org/wiki/Hadoop, Agosto 2011

[2] Wikipedia, Adobe Flex, http://es.wikipedia.org/wiki/Adobe_Flex, Mayo 2010

[3] Wikipedia, Geany, http://es.wikipedia.org/wiki/Geany, Octubre 2005

[4] Adobe, What is Flex? , http://www.adobe.com/products/flex.html, Noviembre 2011

[5] José Manuel Pérez, Que es MySQL?, http://www.esepestudio.com/articulo/desarrollo-web/bases-de-datos-mysql/Que-es-MySQL.htm, Agosto 2005

[6] Alejandro, Trendsmap la herramienta de tendencias en tiempo real para twitter con GeoIP, http://www.debubuntu.com/trendsmap-la-herramienta-de-tendencias-en-tiempo-real-para-twitter-con-geoip/, Septiembre 2011

[7] Wikipedia, Google Trends, http://es.wikipedia.org/wiki/Google_Trends

[8] Milagros, Facebook Lexicon compara palabras en Facebook, http://www.chicaseo.com/facebook-lexicon/, Junio 2007

[9] Daniel Pecos, MySQL, http://danielpecos.com/docs/mysql_postgres/x57.html, Junio 2005

[10] Wikipedia, Adobe Flex, http://es.wikipedia.org/wiki/Adobe_Flex , Octubre 2011

[11] Yahoo, MapReduce, http://developer.yahoo.com/hadoop/tutorial/module4.html , Octubre 2011