ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
CCENTRO DE INVESTIGACIÓN CIENTÍFICA Y TECNOLÓGICA

# Módulo de recomendaciones de páginas a visitar en la Wikipedia, basado en las aportaciones efectuadas por la comunidad de usuarios usando Hadoop

Bolívar Alberto Elbert Pontón [(1)] Andrés Martin Cantos Rivadeneira [(2)] Cristina Abad [(3)]
Facultad de Ingeniería en Electricidad y Computación (FIEC)
ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL (ESPOL)
Campus Gustavo Galindo, vía Perimetral Km. 30.5, Apartado 09-01-5863, Guayaquil, Ecuador
bolívar.elbert@gmail.com [(1)] ancantos99@gmail.com [(2)] cabad@fiec.espol.edu.ec [(3)]

## Resumen

*Los sistemas de recomendaciones cada día se vuelven indispensables para filtrar la gran cantidad de información disponible en la Web. Uno de los objetivos principales de estos sistemas es asistir a los usuarios en sus procesos de búsqueda de información en la Web.*

*En este trabajo se presenta una revisión básica de los aspectos fundamentales relacionados con el diseño, implementación y estructura de un módulo de recomendación para la Wikipedia, utilizando Map/Reduce implementado en el Framework Apache Hadoop para procesar grandes cantidades de información. En la Wikipedia se puede catalogar dos tipos de usuarios, el usuario aportador y el usuario que realiza la consulta, los usuarios suelen aportar a diversas wikis sean estas o no de la misma categoría o contenido.*

*Se creó el módulo de recomendaciones basándose en las wikis a la que han aportado los usuarios. Dada una wiki se consultó que usuarios aportaron a esa wiki, para luego buscar por cada usuario a que otras wikis han aportado y de esta manera obtener para esa wiki una lista de posibles wikis a recomendar.*

*Para tratar en lo posible de no perder la consistencia de la información entre la wiki que el usuario consultó y las wikis recomendadas se aplicó el coeficiente de similitud de Jaccard entre la wiki consultada y cada una de las posibles wikis a recomendar. Este coeficiente nos permite obtener un valor que nos indica el porcentaje de similitud que hay entre dos wikis basándose en el número de usuarios que han aportado tanto a una o ambas wikis, al final son presentadas al usuario aquellas wikis que obtuvieron el porcentaje de similitud más elevado.*

*Podemos decir que el resultado de las recomendaciones para una wiki muestra otros tópicos que también gustan a los usuarios que aportaron con dicha wiki, pero que no necesariamente traten sobre la misma temática.*

**Palabras Claves:** *Sistemas de recomendaciones, Wikipedia, Map / Reduce, Apache Hadoop, Coeficiente de similitud de Jaccard.*

## Abstract

*The daily recommendation systems become essential to filter the vast amount of information available on the Web. One of the main objectives of these systems is to assist users in their information seeking process on the Web.*

*This paper presents a basic overview of key aspects related to design, implementation and module structure of a recommendation to Wikipedia, using Map / Reduce Framework implemented in the Apache Hadoop to process large amounts of information. At Wikipedia you can categorize two types of users, the contributor and the user making the query, users often contribute to several wikis are these or not in the same category or content.*

*Module was created based on recommendations wikis that have contributed to the users. Given a wiki that users were consulted contributed to this wiki, then search for each user to have contributed to other wikis and thereby get for such a list of possible wiki wikis recommended.*

*To do whatever is possible not to lose the consistency of information between the user wiki wikis recommended consulted and applied the Jaccard similarity coefficient between the wiki and consulted each of the wikis to recommend possible. This ratio allows us to obtain a value that indicates the percentage of similarity between two wikis based on the number of users who have contributed so much to one or both wikis, the end user are presented to those wikis that the percentage of similarity obtained more high.*

*We can say that the result of the recommendations for a wiki topics also shows other users like you contributed to the wiki, but not necessarily treated on the same topic.*

**Keywords:** *Recommendations systems, Wikipedia, Map / Reduce, Apache Hadoop, Jaccard similarity coefficients*

## 1. Introducción

La Wikipedia es actualmente la enciclopedia libre disponible en la Web con mayor aceptación entre los usuarios de todo el mundo. Su popularidad se debe, en parte, a su rápido y constante crecimiento, en cuanto a contenidos se refiere.

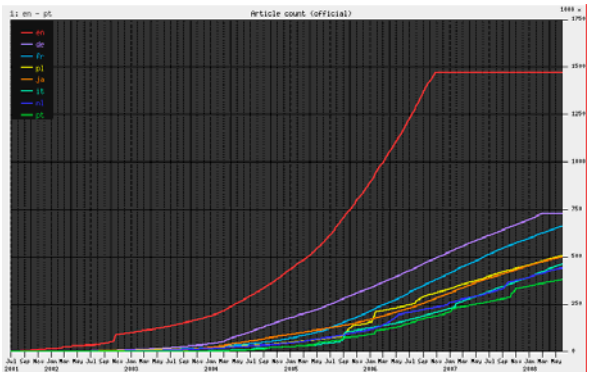

**Figura 1.** Crecimiento de la Wikipedia [1]

Así como crece la Wikipedia, también crece la información disponible en otros sitios Web, como Facebook, YouTube, y Flickr. Con un volumen de información que crece de forma desmedida día a día, los centros de datos empiezan a ganar protagonismo junto a los sistemas de recomendaciones para evaluar y filtrar la gran cantidad de información disponible en la Web y guiar a los usuarios a la información que buscan.

Para estas empresas lograr hacerlo con un bajo presupuesto sin perder características indispensables como la tolerancia a fallos, alta disponibilidad, recuperación de fallos, consistencia en los datos, escalabilidad y seguridad es uno de los problemas a los que se enfrentan día a día. Es aquí donde la plataforma Apache Hadoop [2] entra a ganar mercado, sobre todo en cuanto al uso del paradigma de programación distribuida popularizado por Google llamado Map/Reduce [3], que esta plataforma implementa de manera libre.

Al tener herramientas que nos permiten manejar grandes cantidades de información la forma de presentarlas a los usuarios al momento de realizar búsquedas es otra meta de nuestros días, y es aquí donde los sistemas de recomendación juegan un papel crucial. La idea tras ellos es encontrar usuarios con gustos similares a los de otro determinado y recomendar a este cosas que desconoce pero que gustan a aquellos con los que se tiene similitud.

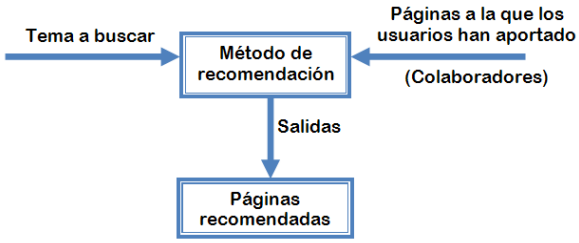


**Figura 2.** Estructura de un Sistemas de recomendaciones [4]

El gráfico de la figura 2 muestra una estructura simple de un sistema de Recomendación adaptado específicamente a nuestro proyecto. El tema central del presente trabajo es brindar una panorámica general del uso y aplicación del paradigma Map/Reduce y de los sistemas de recomendación, así como resultados obtenidos empleando el coeficiente de similitud de Jaccard como parte del método de recomendación para generar nuestras salidas. El campo específico de aplicación de estos conceptos es en la generación de recomendaciones para la Wikipedia, en base a las contribuciones de sus usuarios.

## 2. Materiales y métodos

### 2.1. Dataset

Como Dataset elegimos trabajar con los respaldos de la Wikipedia, todo el contenido de la Wikipedia se encuentra bajo licencia GNU de documentación libre, por lo tanto puede ser copiado, modificado y redistribuido. La Fundación Wikimedia es la encargada de distribuir periódicamente copias de seguridad de todos los contenidos de las diferentes bases de datos de Wikipedia y a estos se los llama dumps.

Todo este contenido puede descargarse desde http://download.wikimedia.org/. Estos dumps están disponibles en formato XML y comprimidos con bzip2.

El contenido de las páginas se encuentra entre los tags <page></page> y las revisiones entre los tags <revisión></revisión> el formato de estos archivos XML es de la siguiente forma.

```
<page>
    <title>Gómez Plata</title>
    <id>454035</id>
    <revision>
      <id>25156038</id>
      <timestamp>2009-03-28T06:38:04Z</timestamp>
      <contributor>
        <username>SajoR</username>
        <id>130444</id>
      </contributor>
      <minor />
      <comment>leve mejora</comment>
      <text xml:space="preserve">'''Montserrat
Domínguez''' ([[Madrid]],
[[1963]]) es una [[periodismo|periodista]]
[[España|español]]
```

**Figura 3.** Formato de los Dumps de Wikipedia

En la página de descargas, hay varios ficheros para descarga, pero los más importantes para nuestro proyecto son:

**Pages-articles.xml.bz2** que contienen las últimas revisiones de los artículos, este archivo es un poco pesado debido a que guarda todo el contenido de la página.**pages-stub-meta-history.xml.7z/bz2** que solo ofrece información sobre el historial de las modificaciones y guarda el usuario que realizó dicha modificación.

Los archivos que decidimos utilizar para realizar las pruebas fueron los siguientes: **eswiki-20090710-pages-articles.xml** (3.18 GB) y **eswiki-20090710-stub-meta-history.xml** (7,04 GB) es decir los respaldos del 10 de Julio del 2009 de la edición de la Wikipedia en español.

El archivo eswiki-20090710-stub-meta-history.xml fue el dataset utilizado para realizar todos los procesos Map/Reduce necesarios para obtener las recomendaciones mientras que el archivo eswiki-20090710-pages-articles.xml se lo utilizó para importar a la base de datos del MediaWiki todo el contenido de la Wikipedia y poder verla offline. El MediaWiki es un software de wiki libre del cual hablaremos más adelante, esta importación se la realiza con un programa hecho en java llamado mwdumper.jar [5] que es gratuito y está disponible en Internet.

## 2.2. Software Wiki (mediawiki)

MediaWiki [6] es un software wiki libre escrito originalmente para Wikipedia, actualmente es utilizado por otros proyectos wikis de la Fundación Wikimedia y también por otras wikis, incluyendo la wiki de la ESPOL. Se lo utilizó en nuestro proyecto como software para visualizar las wikis y resultados de las recomendaciones.

MediaWiki está desarrollado en PHP y puede ser usado con una base de datos MySql o PostgreSQL.

La versión del MediaWiki utilizada por nosotros es la 1.15.0 y puede descargarse desde la siguiente dirección:
http://download.wikimedia.org/mediawiki/1.15/mediawiki-1.15.0.tar.gz

La principal ventaja del MediaWiki es que al ser licencia GNU, este software ofrece todo el código fuente de tal manera que se lo puede modificar y ajustar a los requerimientos de nuestro proyecto.

## 2.3. Cloud[9]

Cloud[9] es una librería MapReduce para Hadoop, de código abierto que se encuentra en un repositorio Subversión, para obtenerlo se lo realizó desde Eclipse abriendo la perspectiva SVN Repository Exploring y añadiendo el siguiente repositorio: https://subversion.umiacs.umd.edu/umd-hadoop/core
Toda esta y más información del Cloud9 como el API Javadoc la obtuvimos desde la Web del autor, Jimmy Lin [7].

Utilizamos esta librería ya que entre sus varias opciones ofrece un paquete que permite trabajar con los dumps de la Wikipedia. Este paquete es edu.umd.cloud9.collection.wikipedia. Entre las clases que contiene este paquete y la que utilizamos con mayor frecuencia es WikipediaPage que representa una página Wikipedia (contenido que se encuentra entre los tags <page></page>) y nos permite obtener–entre otras–cosas el título de la página, el id y el texto XML de la página en sí. Utilizamos WikipediaPage como dato de entrada en algunos Mapper del proyecto.

Otra clase importante es la que nos permite indicar en el Job que el formato de entrada sea de tipo WikipediaPageInputFormat, de tal manera que enviando al Job como entrada un solo archivo XML donde están todas las páginas de la Wikipedia. Esta librería se encarga de enviarnos al map una sola página para ser procesada.

## 2.4. Hadoop

Hadoop [8] es una plataforma que nos permite desarrollar aplicaciones que tengan que tratar con grandes cantidades de datos. Es un subproyecto de Apache open – source que implementa en Java algunas de las tecnologías de Google, como MapReduce y Google File System.

En Hadoop tenemos en lugar de GFS (Google File System) [3] el HDFS que es el sistema de archivos distribuidos pensado para grandes tamaños (GBs a TBs), tolerante a fallos y diseñado para ser instalado

en máquinas de bajo coste que puedan ejecutar una instancia de la JVM.

Nosotros en nuestro proyecto utilizamos la versión 0.18.3 de Hadoop [9].

Todo el proceso de instalación de Hadoop lo evitamos ya que para correr nuestros programas lo realizamos sobre los Amazon Web Services (EC2, S3).

Utilizamos el servicio EC2 con AMIs (Amazon Machine Image) [10] que ya traen instalado Hadoop sobre un sistema operativo Fedora. S3 lo utilizamos como repositorio para los datos de entrada y para la salida final de los procesos MapReduce.

## 2.5. Filtrado de información

Teniendo como hipótesis que los usuarios de la Wikipedia la mayoría de las veces opina sobre temas similares, el primer paso consiste en obtener los usuarios que han contribuido con las ediciones de una página para luego recuperar por cada usuario las páginas a las que ellos también han hecho aportaciones.

Entre los diferentes tipos de usuario que frecuentan la Wikipedia encontramos los usuarios anónimos y los usuarios registrados.

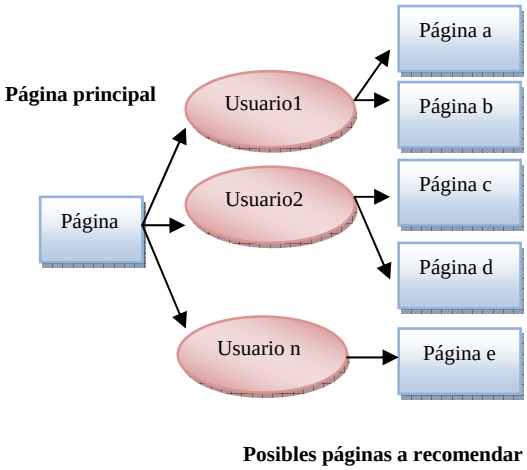


**Figura 4.** Filtrado de información

Los Usuarios anónimos son aquellos que no se han registrado y la Wikipedia en sus revisiones los identifica con su dirección IP, mientras que los usuarios registrados son quienes han seguido un proceso muy simple en el que solo es necesario dar el nombre y contraseña.

La Wikipedia cuenta con «programas robot» (bots) que detectan cambios de tipo vandálico y que actúan en consecuencia, a veces de forma automática, a veces preventiva. Por ejemplo si en un artículo aparecen insultos estos bots se encargan de eliminarlos, estos cambios también son almacenados en las revisiones de la Wikipedia. Una forma sencilla de identificar si un usuario es un robot es verificando si contiene la palabra bot en su nombre de usuario; por ejemplo, el robot principal de la Wikipedia se llama AVBOT [11].

Durante el proceso de filtrado los usuarios anónimos y los robot no son considerados, debido a que estos no siguen una línea de aportaciones sobre un tema definido y su inclusión traería como resultados páginas muy aleatorias.

También son descartadas las páginas que no corresponden a un artículo, con el fin de reducir el número de páginas a procesar. La Wikipedia entre sus tipos de páginas tiene las páginas de discusión, redirects (páginas que se encargan de direccionar a otras páginas), páginas de configuración de la Wikipedia y artículos.

Para comprobar si una página no corresponde a un artículo se verificó si contiene el signo dos puntos ( : ) en su título; por ejemplo, la página con título Wikipedia Discusión: Café (noticias) no corresponde a un artículo.

Este proceso no es de vital importancia, al no intervenir en la obtención del resultado final; sin embargo, lo describimos ya que fue el que nos dio la pauta para comenzar la investigación, entender la estructura de la Wikipedia, comprender cómo se comporta Hadoop con grandes datos de entrada y verificar que la hipótesis planteada por nosotros acerca de que los usuarios aportan generalmente a temas en común es valedera aunque no en un 100% ya que se observó que en algunos casos las posibles páginas a recomendar no se alejaban mucho del tema principal.

Observando esta primera salida pudimos conocer la magnitud de los datos a procesar, darnos cuenta de qué datos dentro del XML servían, y qué datos no; todo esto con el objetivo de reducir los tamaños de la salida para optimizar los procesos.

## 2.6. Valoración de páginas a recomendar

Para valorar qué páginas son las mejores páginas a recomendar para la página principal, hemos decidido utilizar el coeficiente de similitud de Jaccard [12].

### 2.6.1. Coeficiente de similitud de Jaccard

También conocido como el índice de Jaccard, el coeficiente de similitud de Jaccard es una medida estadística de la similitud entre los conjuntos de la muestra.

Para los dos conjuntos, esto es definido como la cardinalidad de su intersección dividido por la cardinalidad de su unión, matemáticamente es expresada de la siguiente manera.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Para el caso de la Wikipedia es necesario un enfoque ligeramente diferente. Tomando como conjunto los usuarios de la Wikipedia que han aportado a dos páginas específicas, es posible determinar la similitud que hay entre estas dos páginas.

Para el cálculo de la similitud de pares de páginas, el numerador es el número de usuarios que han editando ambas páginas mientras que el denominador es el número de usuarios que han editado una o las dos páginas. Matemáticamente, tenemos la siguiente fórmula.

$$J(A,B) = \frac{|X \cap Y|}{|X \cup Y|}$$

Donde A y B son las páginas a las que se desea encontrar la similitud, X y Y corresponden al conjunto de usuarios que se producen con A y B respectivamente.

Gráficamente podemos verlo de la siguiente manera:

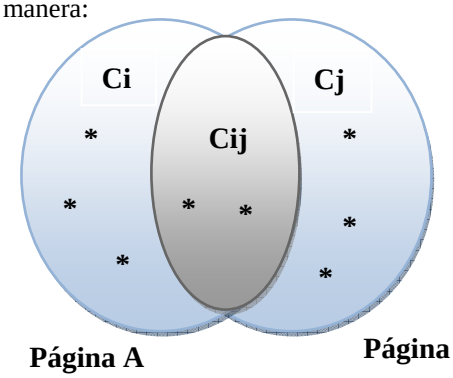


**Figura 5.** Diagrama de Venn para dos páginas de la Wikipedia con sus usuarios

Tomando en cuenta que Ci es el número de usuarios que han aportado a la página A, Cj es el número de usuarios que han aportado a la página B y Cij es el número de usuarios que han aportado tanto a la página A como a la página B, tenemos que la similitud entre estas dos páginas (Coeficiente de similitud de Jaccard) se calcula de la siguiente manera:

$$Sim(A,B) = \frac{Cij}{Ci + Cj - Cij}$$

Sim(A, B) es un valor porcentual y va desde 0 hasta 1 siendo este el valor más alto de coincidencia. En el caso de que no exista ningún usuario que haya aportado tanto a la página A como a la página B, Cij valdría cero y el resultado del la Sim(A,B) será cero; es decir, que es probable que estas dos páginas no tengan absolutamente nada en común.

Otro caso extremo se produce cuando Ci y Cj son cero y todos los usuarios tanto de la página A como de la página B han aportado a ambas páginas; en este caso, la Sim (A, B) será 1, es decir es probable que estas dos páginas tengan mucho en común.

## 2.7. Selección de páginas a recomendar

En el proceso de valoración de datos obtuvimos todas las combinaciones posibles de las páginas a recomendar con sus respectivos valores de similitud. En esta etapa se toma como entrada estas combinaciones y se genera para cada página una lista de páginas a recomendar ordenadas de mayor a menor dependiendo del valor de similitud que tengan con respecto a la página siendo analizada. La Figura 8 presenta nuestra salida final, lista para ser ingresada a la base de datos MySql.

| | |
|---|---|
| (Pag1, Pág. a) | 0.20 |
| (Pag1, Pág. b) | 0.40 |
| (Pag1, Pág. c) | 0.70 |
| (Pág. a, Pág. b) | 0.10 |
| (Pág. a, Pág. c) | 0.01 |
| (Pág. b, Pág. c) | 0.03 |

**Figura 6.** Formato de salida del proceso de valoración de datos

Para elegir las 5 mejores páginas a recomendar, se seleccionan las páginas que tengan la similitud (coeficiente de similitud de Jaccard) más cercano a uno.

Por ejemplo

Sim (Pag1, Pág. a) = 0.2
Sim (Pag1, Pág. b) = 0.4
Sim (Pag1, Pág. c) = 0.7

Entonces tenemos que para la Pag1 la mejor página a recomendar sería la Pág. C

Pag1 Pág. c, 0.7 ^Pág. b, 0.4 ^Pág. a, 0.2
Pág. a Pag1, 0.2 ^Pág. b, 0.10 ^Pág. c, 0.01
Pág. b Pag1, 0.4 ^Pág. a, 0.10 ^Pág. c, 0.03

**Figura 7.** Formato de salida final

## 2.8. Algoritmo utilizado

Para la implementación del algoritmo se utiliza la metodología Map/Reduce.

### 2.8.1. Algoritmo utilizado para el filtrado de información

Para obtener el universo de posibles páginas a recomendar se necesitaron de dos procesos Map/Reduce, primero hay que obtener un listado de páginas por usuario, luego por cada página de esa lista se genera una nueva lista de páginas. Al final se concatenan todas las listas de determinada página y se obtiene como resultado un archivo donde cada línea contiene una página y una lista de posibles páginas a recomendar. Los procesos Map y Reduce se detallan a continuación (ver Figura 8).

#### 2.8.1.1. Procesos map/reduce para filtrado de información

**Primer Proceso Map/Reduce**
**Map**

El método Map consiste generar tuplas <usuarios, página>, teniendo como dato de entrada las páginas de la Wikipedia con la información de las revisiones en formato XML.

La librería Cloud9 nos permite procesar el XML de la Wikipedia y tener como dato de entrada un objeto de tipo WikipediaPage.

La clase WikipediaPage nos proporciona ya gran parte de la información de la página como el título, el id, el tipo de página, el XML en sí de la página, el texto, entre otros.

**Entrada:** WikipediaPage
**Salida:** Text, Text
WikipediaPage → <Usuario, Título Página>

**Reduce**

El reduce recibe una lista de páginas y se encarga de generar otra lista de páginas por cada una de las páginas de la lista recibida. Esta nueva lista generada es la misma lista recibida pero sin tomar en cuenta la página a la que le está generando la lista, ya que esta página es la clave.

Por ejemplo, un reduce recibió los siguientes valores:
Usuario1, {pág. 1, pág. 2, pág. 3, pág. 4}
Se generó una salida por cada página de la siguiente manera:

pág. 1, {pág. 2, pág. 3, pág. 4}
pág. 2, {pág. 1, pág. 3, pág. 4}
pág. 3, {pág. 1, pág. 2, pág. 4}
pág. 4, {pág. 1, pág. 2, pág. 3}

Entrada: Text, {Text}
Salida: Text, Text
Usuario, {Título Página} → Título Página, {Título Página}

**Segundo Proceso Map/Reduce**
**Map**

El método Map recibe una línea del archivo generado en el proceso MapReduce1. Esta línea tiene la forma [Título Página, {Título Página}] y simplemente lo que hace es emitir un map por cada línea del archivo.

**Entrada:** Text, Text
**Salida:** Text, Text
Título Página, {Título Página} → <Título Página, {Título Página}>

**Reduce**

El reduce se encarga de concatenar la lista de páginas recibidas y este sería el universo de posibles páginas a recomendar.

**Entrada:** Text, {Text}
**Salida:** Text, Text
Título Página, {Título Página}→ Título Página, {Título Página}

### 2.8.2. Algoritmo utilizado para obtener el coeficiente de similitud de Jaccard

Para calcular el coeficiente de similitud de Jaccard para un par de elementos X, es necesario el cálculo de dos valores. El primero es el número de elementos de Y que se produce con los dos elementos X, y el segundo es el número de elementos de Y que se producen con uno o ambos elementos X.

Para el cálculo de la segunda cantidad, primero hay que calcular el número de elementos de Y que se produce con el elemento de primer X más el número de elementos de Y que se produce con el segundo elemento X, y luego restar el número de elementos de Y que se producen con ambos.

El valor de la primera cantidad es obtenido mientras se calcula la segunda cantidad y por último este valor se lo divide para la segunda cantidad

obteniendo así el valor del coeficiente de similitud de Jaccard para un par de páginas.

La implementación se la realizó utilizando tres procesos Map/Reduce que se describen a continuación (ver Figura 9).

#### 2.8.2.1. Procesos map/reduce para obtener el coeficiente de similitud de Jaccard

**Primer Proceso Map/Reduce**

**Map**

Este método simplemente genera una tupla <clave, valor> utilizando como clave la página y como valor el usuario que realizó la aportación; teniendo como dato de entrada las páginas de la Wikipedia con la información de las revisiones en formato XML.

**Entrada:** WikipediaPage
**Salida:** Text, Text
WikipediaPage → <Título Página, Usuario>

**Reduce**

El método reduce recibe una página con la lista de usuarios que han aportado a esa página, y calcula la cardinalidad para esa página que no es más que el número de usuarios que aportaron a dicha página (a este valor le llamaremos C). Este valor de C será usado después para calcular el denominador del coeficiente de Jaccard.

El reduce retorna un par donde la clave es la página y el valor es una nueva tupla <Usuario, C> a la que le llamaremos CPar. Por cada usuario de la página se genera una salida de este tipo.

**Entrada:** Text, {Text}
**Salida:** Text, CPar
Título Página, {Usuario} → Título Página, <Usuario, C>;
Donde C = | {lista de Usuarios} |

**Segundo Proceso Map/Reduce**

**Map**

Aquí se tiene como dato de entrada la salida del Proceso Map/Reduce 1, cada línea de los archivos de salida de dicho proceso tiene la forma [Título Página Usuario C].

Por cada línea del archivo se generó un map donde la clave es el Usuario y la salida es una nueva tupla <Página, C>; en otras palabras, la salida de este método map es la salida del método reduce del proceso MapReduce1 pero con los valores de Página y Usuario intercambiados.

**Entrada:** Text, CPar
**Salida:** Text, CPar
Título Página, <Usuario, C> → Usuario, <Título Página, C>

**Reduce**

Este recibe una lista de tuplas <Título Página, C> por cada usuario, y la salida de este reduce es muy peculiar ya que por clave se utilizó un string vacío y por valor simplemente se concatena la lista de valores CPar recibidos, de tal manera que cada línea del archivo de salida es una lista de valores CPar con la forma <Título Página, C>.

**Entrada:** Text, {CPar}
**Salida:** Text, {CPar}
Usuario, {<Título Página, C>} → R, {<Título Página, C>};
Donde R = "" (String vacío)

**Tercer Proceso Map/Reduce**

**Map**

En este método se recibe la salida del proceso Map/Reduce 2 cada línea del archivo de salida tiene una lista de tuplas de la forma <Título Página, C>. Este proceso es el encargado de generar las tuplas de página <Xa, Xb> con el primer elemento de la lista y cada uno de los siguientes elementos, es decir una lista de tamaño n producirá n-1 tuplas, uno por cada elemento de la lista exceptuando el primero. Las tuplas generadas son la clave del map y el valor es la suma de los valores de C asociado a cada página.

Entrada: Text, {CPar}
Salida: Text, Intwritable
R, {<Título Página, C>} → <Xa, Xb>, Cij;
Donde Cij = Ci + Cj

**Reduce**

Finalmente, este es el reduce que calcula el coeficiente de similitud de Jaccard. Primero cuenta cuantas veces se generó la tupla <Xa, Xb> y este valor representa la intersección de Xa con Xb, es decir el numerador del coeficiente de similitud de Jaccard. El valor del denominador se obtiene restando el valor de Cij asociado al par con el de la intersección.

**Entrada:** Text, {Intwritable}
**Salida:** Text, Floatwritable
<Xa, Xb>, {Cij} → <Xi, Xj>, $\frac{|\{Cij\}|}{Cij - |\{Cij\}|}$

#### 2.8.3. Algoritmo utilizado para seleccionar las páginas a recomendar

Este proceso es muy sencillo pero muy importante, ya que es el que nos permite obtener las recomendaciones para cada página ordenadas de mayor a menor dependiendo del valor del coeficiente de similitud de Jaccard. Consta de un solo proceso Map/Reduce. Se recibe como entrada la lista de todas las posibles combinaciones de páginas con su valor de

similitud, y simplemente se emiten dos maps por cada combinación de página y en el Reduce se procede a ordenar y concatenar las páginas. Su proceso Map/Reduce se describe a continuación (ver Figura 10).

#### 2.8.3.1. Proceso Map/Reduce para seleccionar las páginas a recomendar

**Map**

Este método recibe una línea de la forma [Página A, Página B  0.2] que contiene un par de página con su respectivo valor de coeficiente de similitud entre ellas.

Se genera una tupla <clave, valor> utilizando como clave una página y como valor la otra página concatenada con el valor de similitud.

Como en el ejemplo la similitud entre la página A y la página B es 0.2 entonces se generan las dos siguientes salidas:

Una salida indica que la Página B es posible recomendación para la Página  A con un grado de similitud de 0.2 <Página A, (Página B, 02)>.

La otra salida indica que la Página A es posible recomendación para la Página B con un grado de similitud de 0.2 <Página B, (Página A, 02)>.

**Entrada:** Text, Text
**Salida:** Text, Text
[Título Página 1, Título Página 2, Valor Coeficiente] → <Título Página 1, (Título Página 2, Valor Coeficiente)> y <Título Página 2, (Título Página 1, Valor Coeficiente)>

**Reduce**

El reduce recibe una página con la lista de posibles páginas a recomendar asociadas con su valor de similitud. Este método se encarga de ordenar de mayor a menor esa lista dependiendo de su valor de similitud, se emite una salida con la página y la misma lista recibida pero ordenada.

**Entrada:** Text, {Text}
**Salida:** Text, {Text}
Título Página, {<Título Página, Coeficiente>} → Título Página, {<Título Página, Coeficiente>}

## 3. Resultados

Como se mencionó anteriormente se utilizó el MediaWiki para presentar los resultados. Este software utiliza una base de datos MySql para almacenar toda su información, la salida final de los diferentes procesos Map/Reduce es introducido a esta base de datos, ya que la salida es simplemente un archivo de texto donde cada línea del archivo representa un registro, nosotros obtenemos todas las recomendaciones posibles para una página dependiendo del número de usuarios que ha aportado a esa página, pero solo son introducidos a la base de datos las 5 recomendaciones que tengan el índice de similitud de Jaccard más cercano a uno y estas son presentadas en la pestaña de recomendaciones. Esto con el objetivo de no saturar la base con datos innecesarios.

La tabla que almacena las recomendaciones es simple: contiene 6 campos, de los cuales, el primero corresponde a la página principal y los 5 restantes cada uno corresponde a una página recomendada.

Se modificó el MediaWiki y se le añadió para cada página una pestaña de recomendaciones que es donde se mostrarán los resultados.

La arquitectura final del sistema es la siguiente (ver Figura 10).


**Figura 11.** Arquitectura final del sistema

Como se puede observar la arquitectura final del sistema es muy simple y en ningún momento se encuentra conexión con el EC2 o el S3 que son los servicios de Amazon que se utilizaron para generar las recomendaciones.

Esto se debe a que el proceso de generar las recomendaciones no es un proceso en línea que se ejecuta cada vez que el usuario busca una página. Este es un proceso que se encuentra realizado con anterioridad y debería realizarse periódicamente dependiendo de cómo sea la tasa de cambio de los datos en la Wikipedia, o en su debido caso se lo puede realizar cada vez que se libere un nuevo respaldo de la base de datos de la Wikipedia.

**Tabla 1.** Tiempos tomados para la obtención del coeficiente de Jaccard

| | Nº PÁGINAS ENTRADA | | | | |
|---|---|---|---|---|---|
| | 57.000 | 112.468 | 216.903 | 459.866 | 1.064.418 |
| 1° Map Reduce | 3,07 | 3,15 | 3,27 | 3,45 | 4,39 |
| 2° Map Reduce | 1,21 | 3,56 | 11,05 | 29,39 | 52,25 |
| 3° Map Reduce | 5,56 | 21,21 | 67,57 | 209,56 | 395,42 |
| Tiempo Total | 9,84 | 27,92 | 81,89 | 242,40 | 452,06 |

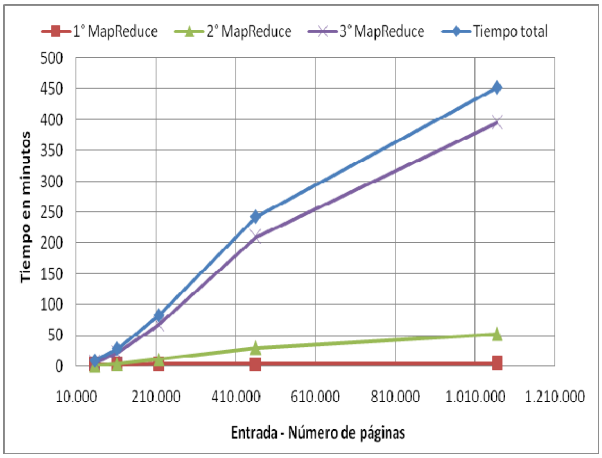


**Figura 12.** Resultado de pruebas para la obtención del coeficiente de Jaccard

Como se observa en el gráfico el primer proceso Map/Reduce no varía casi nada al incrementar el número de entrada. Sigue una tendencia casi constante, tanto así que de la primera a la última prueba solo tuvo un incremento de 1,32 minutos.

El segundo proceso si presenta una tendencia creciente evolutiva pero no muy significativa ya que entre la primera y última prueba hubo un incremento de menos de una hora. Esto es normal ya que el segundo proceso tiene que iterar sobre una lista cuyo tamaño depende de los datos de entrada.

Otra observación importante es que a pequeñas cantidades de entradas del orden no mayor a las 200.000 páginas el primero y segundo proceso se ejecuta casi al mismo tiempo.

Como era de esperarse el tercer proceso Map reduce es el que más incide sobre el tiempo total; siendo así, que es el que marca la tendencia creciente casi exponencial para el tiempo total del proceso, presenta variaciones bruscas con respecto a la entrada.

Tal comportamiento era esperado, ya que durante este proceso los método map y reduce realizan operaciones que a pesar de que no son tan complejas se realizan al iterar sobre grandes conjuntos de datos, el map tiene que crear los pares de páginas y para esto recorre más de una vez sobre una lista, mientras que el reduce realiza el cálculo final para obtener el coeficiente de Jaccard. En resumen, se recorre varias veces una lista cuyo tamaño depende de los datos de entrada.

## 4. Conclusiones

Utilizamos el coeficiente de similitud Jaccard como método para encontrar similitudes entre páginas. Este coeficiente utiliza teoría de conjuntos para hallar un porcentaje de similitud entre dos elementos de un conjunto. Logramos enfocar la teoría del Coeficiente de Jaccard para utilizarlo sobre los datos de la Wikipedia valiéndonos de los usuarios como elementos del conjunto página, implementamos este coeficiente en su versión MapReduce y de esta manera pudimos obtener todas las combinaciones posibles de páginas con su valor de similitud. Un proceso que no resulto muy costoso ya que para generar las recomendaciones se necesitan de tan solo $0,50 y para almacenar la salida de los procesos $1,50.

Observando los resultados obtenidos que se muestran en el apartado 7.4 del presente trabajo, podemos concluir que los usuarios de la Wikipedia no siempre pero si en algunas ocasiones aporta mayormente a wikis que tienen que ver con la misma temática y esto nos permite mantener una consistencia de información entre la wiki buscada y las recomendaciones pero no siempre será del 100% y en algunos casos puede que sea del 0%. Si lo que se quiere es tener recomendaciones basadas totalmente en contenido con un 100% de consistencia de datos, no es suficiente basar las recomendaciones en las aportaciones de los usuarios.

Para el caso particular de la Wikipedia que no almacena nada de información sobre gustos o preferencias de los usuarios, utilizar las aportaciones de los usuarios es un método valedero y rápido para generar recomendaciones y en algunos de los casos sin perder mucho la consistencia de datos.

En las pruebas realizadas se observó que el algoritmo obtiene su mayor eficiencia con un número de entradas del orden de las miles de páginas, al aumentar el tamaño a las millones de páginas los tiempos se disparan y se observa que se produce una escalabilidad lineal, estos tiempos podrían mejorarse aumentando el tamaño del clúster. Los procesos fueron desarrollados pensando en que se realizarían periódicamente con el fin de actualizar una base de

datos, es por eso que consideramos que el mayor tiempo obtenido que fue de 15 horas sumando el proceso de obtención del coeficiente de similitud y el proceso de selección de páginas es un buen tiempo tomando en cuenta que se trabajó con 10 nodos para una entrada de más de un millón de páginas.

## 5. Recomendaciones

Al realizar las salidas de los diferentes Map/Reduce, al principio utilizábamos los títulos de las páginas para identificarlas, estos eran concatenados y escritos en las diferentes salidas, comenzamos a notar un crecimiento desmesurado del tamaño de las salidas incluso para pequeñas entradas, llegando a tener archivos de hasta 1G para entradas de 1000 páginas. Esto retrasaba los procesos Map/Reduce que utilizaban esas salidas como entradas para sus operaciones. En base a esto, recomendamos evitar utilizar cadenas de caracteres tan largas para las salidas. En nuestro caso utilizamos los Id de las páginas que son enteros de longitud 10 y notamos una mejora en el tiempo de los procesos.

Los bucles internos dentro de los procesos Map/Reduce dependen mucho del tamaño de la entrada, conforme aumentábamos el tamaño de la entrada, estos bucles tomaban más de 10 minutos en terminar, durante este tiempo no hay actividad entre Hadoop y el proceso Map/Reduce ya que no se lee ni escribe datos en el HDFS pero si se realizan operaciones, Hadoop después de 10 minutos de detectar inactividad en un proceso procede a matar el proceso, es recomendable utilizar la instrucción reporter.progress() dentro de estos bucles para indicarle a Hadoop que el proceso sigue ahí.

Además de utilizar el índice de similitud de Jaccard, se podría mejorar el sistema de recomendación añadiéndole características como retroalimentación y heurísticas que permitan obtener recomendaciones basándose en otros criterios además de las aportaciones hechas por los usuarios.

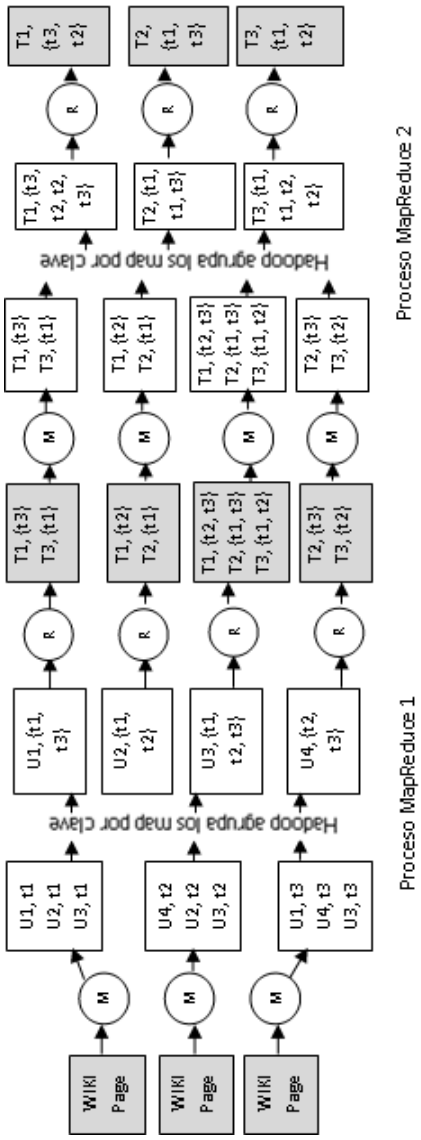


**Figura 8.** Procesos Map/Reduce para el Filtrado de Información

## 6. Apéndice

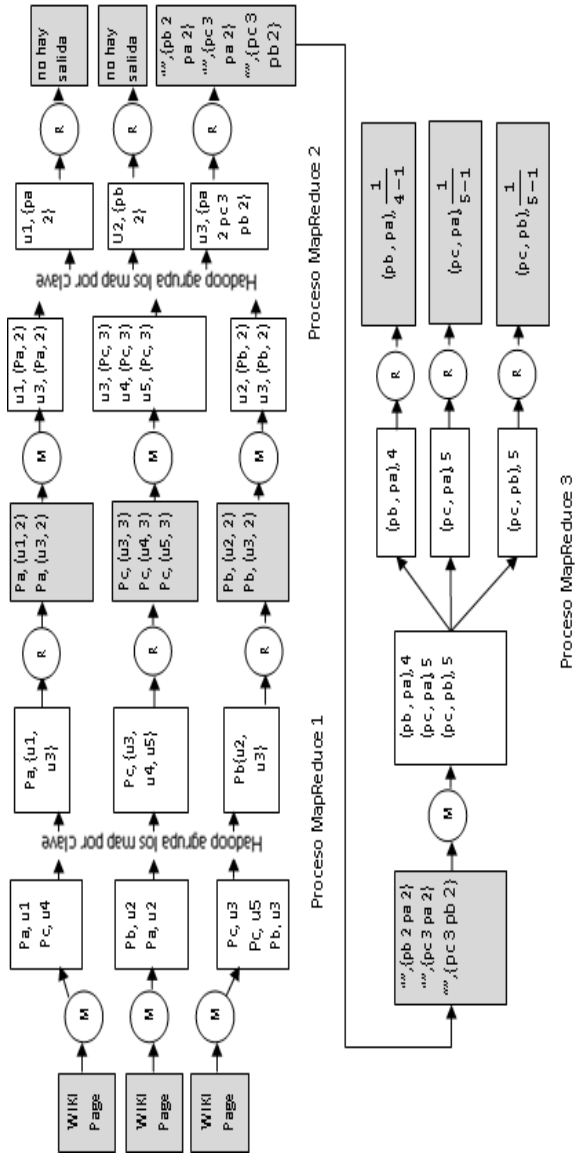


**Figura 9.** Procesos Map/Reduce para obtener el Coeficiente de similitud de Jaccard

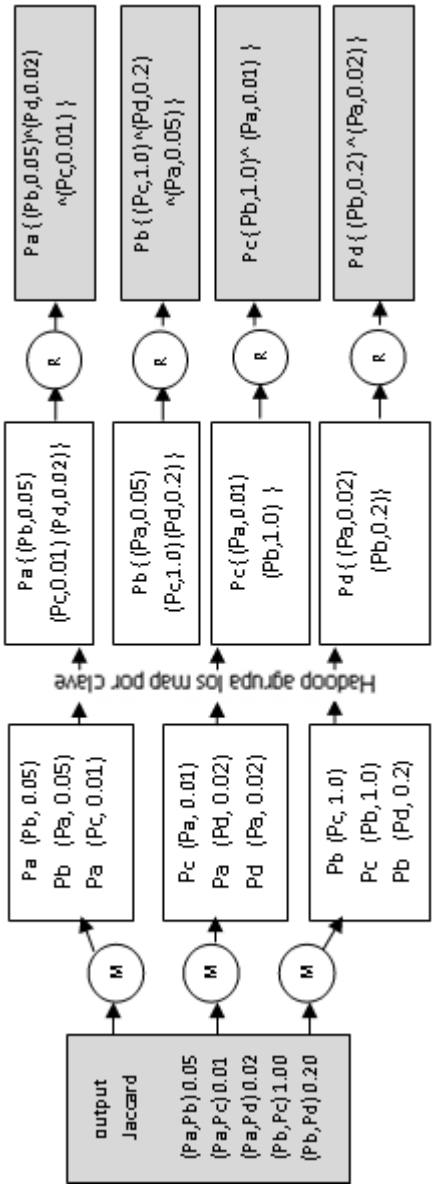


**Figura 10.** Proceso Map/Reduce para seleccionar las páginas a recomendar

## 7. Referencias bibliográficas

[1] Wikimedia Commons, "Archivo: Wikipedia growth.png - Wikipedia, la enciclopedia libre," Archivo: Wikipedia growth.png, <http://es.wikipedia.org/wiki/Archivo:Wikipedia_growth.png> [Consultado: Jueves 27 de agosto del 2009]

[2] The Apache Software Foundation, "Apache Hadoop." <http://hadoop.apache.org/> [Consultado: viernes 4 de septiembre del 2009]

[3] Dean Jeffrey y Ghemawat Sanjay, "MapReduce: Simplified Data Processing on Large Clusters ," OSDI'04: Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, December, 2004.

[4] "Sistemas de recomendaciones: herramientas para el filtrado de información en Internet", <http://www.hipertext.net/web/pag227.htm> [Consultado: Lunes, 24 de Agosto del 2009]

[5] "MWDumper - MediaWiki. "<http://www.mediawiki.org/wiki/MWDumper> [Consultado: martes 8 de septiembre del 2009]

[6] "MediaWiki/es - MediaWiki." <http://www.mediawiki.org/wiki/MediaWiki/es> [Consultado: martes 8 de septiembre del 2009]

[7] "Cloud9: A Library for Hadoop." <http://www.umiacs.umd.edu/~jimmylin/cloud9/docs/index.html> [Consultado: lunes 3 de agosto del 2009]

[8] Borthakur Dhruba, "The Hadoop Distributed File System: Architecture and Design." <http://hadoop.apache.org/common/docs/r0.15.3/hdfs_design.pdf > [Consultado lunes 3 de agosto del 2009]

[9] "Hadoop Common Releases." <http://hadoop.apache.org/common/releases.html> [Consultado: lunes 3 de agosto del 2009]

[10] "Amazon Machine Image - Wikipedia, the free encyclopedia." <http://en.wikipedia.org/wiki/Amazon_Machine_Image> [Consultado: martes 15 de septiembre del 2009]

[11] "Usuario:AVBOT - Wikipedia, la enciclopedia libre." <http://es.wikipedia.org/wiki/Usuario:AVBOT> [Consultado: martes 15 de septiembre del 2009]

[12] Bank Jacob y Cole Benjamin, "Calculating the Jaccard Similarity Coefficient with Map Reduce for Entity Pairs in Wikipedia," December 16, 2008.