# *Reporte técnico y Reporte de Avance, fases*

## *Tópicos en Telemática*

## *Profesor:*

## *Edwin Montoya*


***por:***

***Carlos Mario Blanco Pérez***

***Santiago Vásquez Zuluaga***



*Eafit*

*2017*

## ***Procesos:***

***ETL****:* En el proceso de extracción de los datos se procederá a descargar el *dataset* de la biblioteca de Gubenberg el cual incluye una colección de documentos tanto en inglés como en español. (Dado que el *dataset* se encuentra en la carpeta raíz su avance es del 100%)

En el proceso de transformación se llevaran todos los elementos que integran el *dataset* a un solo formato .txt (dado que los documentos dentro del *dataset* tienen todos una extensión .txt su avance es del 100%)

En el proceso de carga se llevara el *dataset* a HDFS (Hadoop Distributed File System, cuyo entorno ya está configurado por defecto con un master en la dirección 10.131.137.188 y sus respectivos slaves en los cuales estará distribuida la carga), avance 100%

***Procesamiento:*** El procesamiento se realizara bajo una arquitectura de datos en *batch* donde se integrara al lenguaje de programación *Python* junto con una librería *mrjob* (*map/reduce)* para el manejo y procesamiento del *dataset* bajo un *índice invertido* para clasificar, ordenar y consultas posteriores. Avance 100%

***Aplicación:*** Se desarrollara una aplicación en *Ruby on Rails (RoR),* la cual será la que tendrá interacción directa con el usuario que en forma de búsqueda el agregara en un *text área* una serie de palabras las cuales retornaran como resultado el documento(s) .txt los cuales tengan la mayor ocurrencia de las palabras ingresadas por el usuario. Avance 100%.

**Github:** https://github.com/santivasquez/bigDataProyecto3.git

***Algoritmo Map/reduce, programado en Python***

```python
# -*- coding: utf-8 -*-
from mrjob.compat import jobconf_from_env
from mrjob.job import MRJob
from pymongo import MongoClient

# PASO 1: Conexión al Server de MongoDB Pasandole el host y el puerto
mongoClient = MongoClient('10.131.137.188',27017)

# PASO 2: Conexión a la base de datos
db = mongoClient["grupo_07"]
db.authenticate("user1", "eafit.2017")

# PASO 3: Obtenemos una coleccion para trabajar con ella
collection = db.gutenberg

class MRWordFrequencyCount(MRJob):

    def mapper(self, _, line):
        value = line.decode('utf-8','ignore').split()

        for word in value:
            yield (jobconf_from_env('mapreduce.map.input.file'),word), 1

    def reducer(self, key, values):
        collection.insert({ 'nameFile': key[0],'word': key[1], 'numero':sum(values)})
        yield key,sum(values)

if __name__ == '__main__':
    MRWordFrequencyCount.run()

# PASO FINAL: Cerrar la conexion
mongoClient.close()
```

***La estructura y outputs en la base de datos Mongodb son:***

```
> db.gutenberg.find({"word": "todos"}).limit(2).pretty()
{
        "_id" : ObjectId("59069beed03b4f1fdafa7d09"),
        "word" : "todos",
        "numero" : 115,
        "nameFile" : "hdfs://master:8020/user/st0263/cblanco/gutenberg-txt-es/10293-8.txt"
}
{
        "_id" : ObjectId("59069c70d03b4f1fdafaa9cb"),
        "word" : "todos",
        "numero" : 115,
        "nameFile" : "hdfs://master:8020/user/st0263/cblanco/gutenberg-txt-es/10293.txt"
}
```

***Comandos usados***

*python doc_mapreduce.py -r hadoop hdfs:///user/st0263/cblanco/gutenberg-txt-es/**

*// Correr el algoritmo Map/reduce pasandole como parametros todos los textos planos dentro de la carpeta Gutenberg-txt-es*

*mongo grupo_07 -u user1 –p*

*// Conectarse a la base de datos MongoDB*

*hadoop fs -ls "hdfs://master:8020/user/st0263/cblanco/"*

*//Inspectionar*

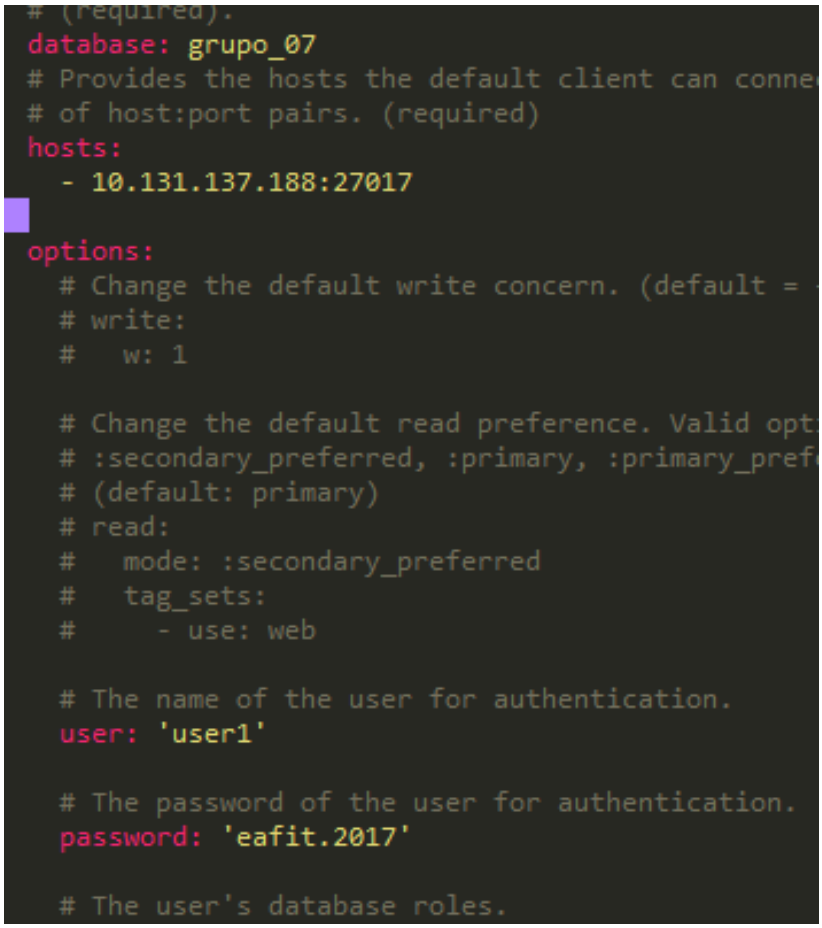
*hadoop fs -put /home/cblanco/gutenberg-txt-es /user/st0263/cblanco*

*//agregar*

*hadoop fs -cat "hdfs://master:8020/user/st0263/cblanco/gutenberg-txt-es/10293.txt"*

*//Mostrar en pantalla el contenido de “10293.txt”*

***Configuración – MongoDB – Ruby on Rails – mongoid.yml***




**Indicaciones:**
Para iniciar la aplicación, se debe ingresar por la URL 10.131.137.166:3000. Donde se encuentra con una aplicación web elaborada en Ruby on Rails. La interacción se hace mediante el botón de búsqueda, donde se ingresan las palabras a buscar y a continuación se listarán los primeros 20 archivos (traídos de la biblioteca Gutenberg) en donde más se repitan las palabras ingresadas. Los archivos listados contienen un enlace hacia su ubicación, donde se puede tener más información del archivo.

URL: 10.131.137.166:3000