

# UNIVERSIDAD DE GRANADA

## MASTER PROFESIONAL EN INGENIERÍA INFORMÁTICA

### PRÁCTICA 4

---

## Hadoop

---

*Autor:*

Manuel Jesús García Manday  
(nickter@correo.ugr.es)

Master en Ingeniería Informática

21 de mayo de 2017

## Índice

1. Objetivo.	3
2. Introducción.	3
3. Calcula el valor mínimo de la variable (columna) 5.	4
4. Calcula el valor máximo de la variable (columna) 5.	6
5. Calcula al mismo tiempo los valores máximo y mínimo de la variable 5.	9
6. Calcula los valores máximo y mínimo de todas las variables (salvo la última, que es la etiqueta de la clase).	12
7. Realizar la media de la variable 5.	15
8. Obtener la media de todas las variables (salvo la clase).	18
9. Comprobar si el conjunto de datos ECBDL es balanceado o no balanceado, es decir, que el ratio entre clases sea menor o mayor que 1.5 respectivamente.	20
10.Cálculo del coeficiente de correlación entre todas las parejas de variables.	23

## 1. Objetivo.

El objetivo de esta práctica es realizar programas escalables para mejorar la eficiencia en entornos Big Data.

## 2. Introducción.

Para comenzar a realizar las tareas que se piden en esta práctica, es necesario en primer lugar realizar una serie de pasos iniciales que se describen a continuación.

Realizamos una conexión remota hacia el servidor **hadoop.ugr.es** y una vez dentro creamos una carpeta nueva donde descargaremos el código Java de los programas. Comprobamos tambien que los datos de entrada se encuentran disponibles.

```
[MacBook-Pro-de-Jesus:~ jesusgarciamanday]$ ssh mcc48893432@hadoop.ugr.es  
Password:  
Last login: Tue May  9 18:48:58 2017 from cvi063087.ugr.es
```

Figura 1: Conexión remota a **hadoop.ugr.es**.

```
[[mcc48893432@hadoop-master ~]$ mkdir stat  
[[mcc48893432@hadoop-master ~]$ cp /tmp/Min* ./stat/  
[[mcc48893432@hadoop-master ~]$ ls stat  
Min.java  MinMapper.java  MinReducer.java
```

Figura 2: Copiando los ficheros Java.

```
[[mcc48893432@hadoop-master ~]$ hdfs dfs -ls /tmp/BDCC/datasets/ECBTL14  
Found 1 items  
-rw-r--r--  2 root supergroup _102747144 2017-03-16 11:06 /tmp/BDCC/datasets/ECBTL14/ECBTL14_10tst.data
```

Figura 3: Datos de entrada.

Viendo que están disponibles, ahora nos creamos un directorio local para las clases de java.

```
[[mcc48893432@hadoop-master ~]$ cd stat/  
[[mcc48893432@hadoop-master stat]$ mkdir java_classes
```

Figura 4: Directorio local.

Con la preconfiguración realizada pasamos a realizar las diferentes tareas que es exponen en la práctica.

### 3. Calcula el valor mínimo de la variable (columna) 5.

La primera de ellas es calcular el mínimo sobre el conjunto de valores del dataset, por lo que una vez que tenemos todos los ficheros java correspondientes, ahora toca compilarlos para crear el fichero **.jar** a continuación y ejecutarlo en **hadoop**.

```
[mcc48893432@hadoop-master stat]$ javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop-mapreduce/* -d java_classes Min*
[mcc48893432@hadoop-master stat]$ /usr/java/jdk1.8.0_25/bin/jar -cvf stat.jar -C java_classes /
added manifest
adding: oldapi/(in = 0) (out= 0)(stored 0%)
adding: oldapi/Min.class(in = 1420) (out= 782)(deflated 44%)
adding: oldapi/MinMapper.class(in = 1801) (out= 706)(deflated 60%)
adding: oldapi/MinReducer.class(in = 1717) (out= 709)(deflated 58%)
adding: MinMapper.java(in = 916) (out= 367)(deflated 59%)
adding: java_classes/(in = 0) (out= 0)(stored 0%)
adding: java_classes.oldapi/(in = 0) (out= 0)(stored 0%)
adding: java_classes.oldapi/Min.class(in = 1420) (out= 782)(deflated 44%)
adding: java_classes.oldapi/MinMapper.class(in = 1801) (out= 706)(deflated 60%)
adding: java_classes.oldapi/MinReducer.class(in = 1717) (out= 709)(deflated 58%)
adding: Min.java(in = 982) (out= 396)(deflated 59%)
adding: MinReducer.java(in = 816) (out= 334)(deflated 59%)
```

Figura 5: Compilamos y ejecutamos (I).

```
[mcc48893432@hadoop-master stat]$ javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop-mapreduce/* -d java_classes Min*
[mcc48893432@hadoop-master stat]$ /usr/java/jdk1.8.0_25/bin/jar -cvf stat.jar -C java_classes /
added manifest
adding: oldapi/(in = 0) (out= 0)(stored 0%)
adding: oldapi/Min.class(in = 1420) (out= 782)(deflated 44%)
adding: oldapi/MinMapper.class(in = 1801) (out= 706)(deflated 60%)
adding: oldapi/MinReducer.class(in = 1717) (out= 709)(deflated 58%)
adding: MinMapper.java(in = 916) (out= 367)(deflated 59%)
adding: java_classes/(in = 0) (out= 0)(stored 0%)
adding: java_classes.oldapi/(in = 0) (out= 0)(stored 0%)
[mcc48893432@hadoop-master stat]$ hadoop jar stat.jar oldapi.Min /tmp/BDCC/datasets/ECBDL14/ECBDL14_10tst.data ./stat/output/
17/05/19 17:29:17 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/192.168.10.1:8032
17/05/19 17:29:17 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/192.168.10.1:8032
17/05/19 17:29:17 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface
and execute your application with ToolRunner to remedy this.
[17/05/19 17:29:18 INFO mapred.FileInputFormat: Total input paths to process : 1
17/05/19 17:29:18 INFO mapreduce.JobSubmitter: number of splits:2
[17/05/19 17:29:18 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1494408081774_0178
17/05/19 17:29:18 INFO impl.YarnClientImpl: Submitted application application_1494408081774_0178
17/05/19 17:29:18 INFO mapreduce.Job: The url to track the job: http://hadoop.ugr.es:8088/proxy/application_1494408081774_0178/
17/05/19 17:29:18 INFO mapreduce.Job: Running job: job_1494408081774_0178
17/05/19 17:29:22 INFO mapreduce.Job: Job job_1494408081774_0178 running in uber mode : false
17/05/19 17:29:22 INFO mapreduce.Job: map 0% reduce 0%
17/05/19 17:29:32 INFO mapreduce.Job: map 100% reduce 0%
17/05/19 17:29:38 INFO mapreduce.Job: map 100% reduce 75%
17/05/19 17:29:39 INFO mapreduce.Job: map 100% reduce 81%
17/05/19 17:29:41 INFO mapreduce.Job: map 100% reduce 88%
17/05/19 17:29:42 INFO mapreduce.Job: map 100% reduce 100%
17/05/19 17:29:43 INFO mapreduce.Job: Job job_1494408081774_0178 completed successfully
17/05/19 17:29:43 INFO mapreduce.Job: Counters: 49
```

Figura 6: Compilamos y ejecutamos (II).

```

File System Counters
    FILE: Number of bytes read=2142847
    FILE: Number of bytes written=6470142
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=102749934
    HDFS: Number of bytes written=8
    HDFS: Number of read operations=54
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=32
Job Counters
    Launched map tasks=2
    Launched reduce tasks=16
    Rack-local map tasks=2
    Total time spent by all maps in occupied slots (ms)=112574
    Total time spent by all reduces in occupied slots (ms)=1894977
    Total time spent by all map tasks (ms)=16082
    Total time spent by all reduce tasks (ms)=38673
    Total vcore-seconds taken by all map tasks=16082
    Total vcore-seconds taken by all reduce tasks=38673
    Total megabyte-seconds taken by all map tasks=112574000
    Total megabyte-seconds taken by all reduce tasks=1933650000

```

Figura 7: Compilamos y ejecutamos (III).

```

Map-Reduce Framework
    Map input records=2897917
    Map output records=2897917
    Map output bytes=28979170
    Map output materialized bytes=2143005
    Input split bytes=234
    Combine input records=0
    Combine output records=0
    Reduce input groups=1
    Reduce shuffle bytes=2143005
    Reduce input records=2897917
    Reduce output records=1
    Spilled Records=5795834
    Shuffled Maps =32
    Failed Shuffles=0
    Merged Map outputs=32
    GC time elapsed (ms)=347
    CPU time spent (ms)=37010
    Physical memory (bytes) snapshot=7926947840
    Virtual memory (bytes) snapshot=984134000640
    Total committed heap usage (bytes)=19421724672
Shuffle Errors
    BAD_ID=0
    CONNECTION=0
    IO_ERROR=0
    WRONG_LENGTH=0
    WRONG_MAP=0
    WRONG_REDUCE=0
File Input Format Counters
    Bytes Read=102749700
File Output Format Counters
    Bytes Written=8

```

Figura 8: Compilamos y ejecutamos (IV).

Por último comprobamos el resultado para ver si se ha realizado correctamente.

```

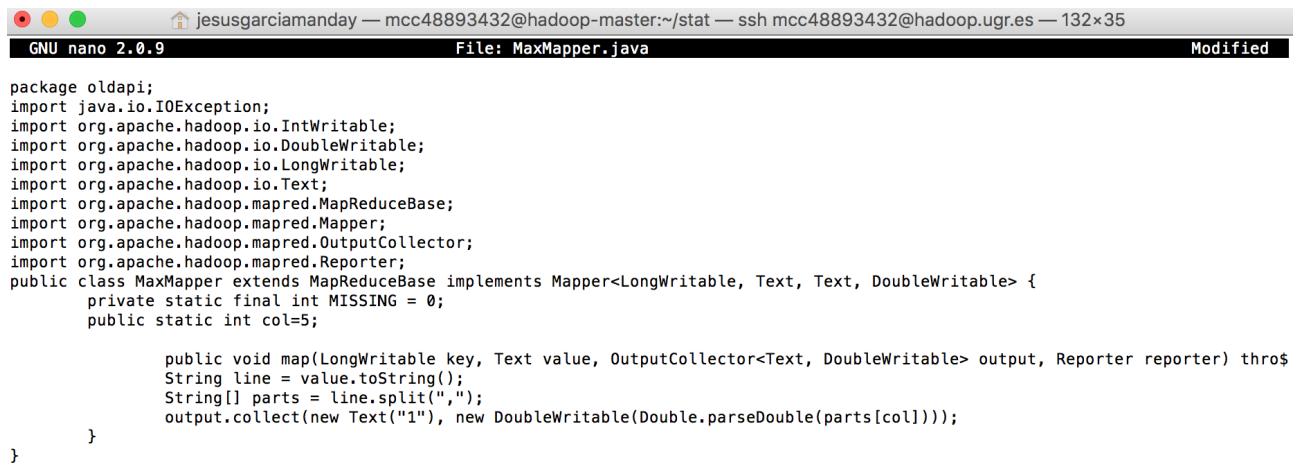
[jesusgarciamanday — mcc48893432@hadoop-master:~/stat — ssh mcc48893432@hadoop.ugr.es]
[[mcc48893432@hadoop-master stat]$ hdfs dfs -cat stat/output/*
1      -11.0

```

Figura 9: Comprobando resultado.

## 4. Calcula el valor máximo de la variable (columna) 5.

Para calcular el valor máximo de la variable (columna) 5 vamos a crear clases de java correspondientes para hacer dicha operación. Comenzaremos por crear la clase **MaxMapper** con su fichero java correspondiente.



```

package oldapi;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;
public class MaxMapper extends MapReduceBase implements Mapper<LongWritable, Text, Text, DoubleWritable> {
    private static final int MISSING = 0;
    public static int col=5;

    public void map(LongWritable key, Text value, OutputCollector<Text, DoubleWritable> output, Reporter reporter) throws IOException {
        String line = value.toString();
        String[] parts = line.split(",");
        output.collect(new Text("1"), new DoubleWritable(Double.parseDouble(parts[col])));
    }
}

```

Figura 10: Clase MaxMapper.

A continuación creamos la clase correspondientes para el Reducer (**MaxReducer**) y la clase principal **Max** donde estará el main.



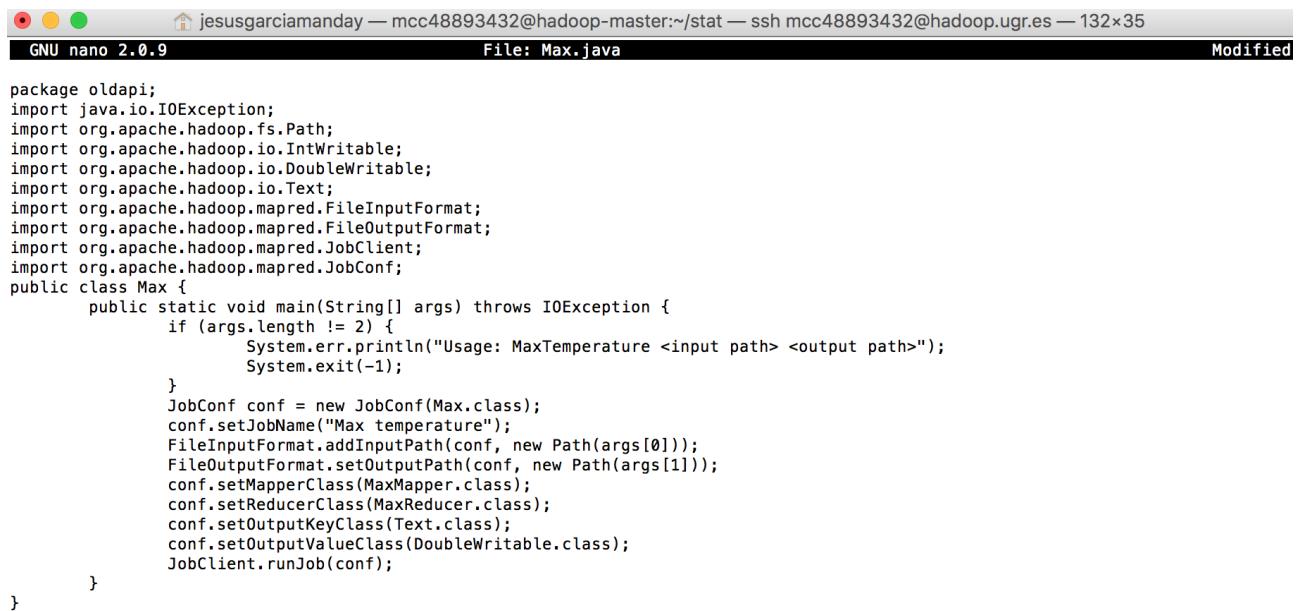
```

package oldapi;
import java.io.IOException;
import java.util.Iterator;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;
public class MaxReducer extends MapReduceBase implements Reducer<Text, DoubleWritable, Text, DoubleWritable> {

    public void reduce(Text key, Iterator<DoubleWritable> values, OutputCollector<Text, DoubleWritable> output, Reporter reporter) throws IOException {
        Double maxValue = Double.MIN_VALUE;
        while (values.hasNext()) {
            maxValue = Math.max(maxValue, values.next().get());
        }
        output.collect(key, new DoubleWritable(maxValue));
    }
}

```

Figura 11: Clase MaxReducer.



```

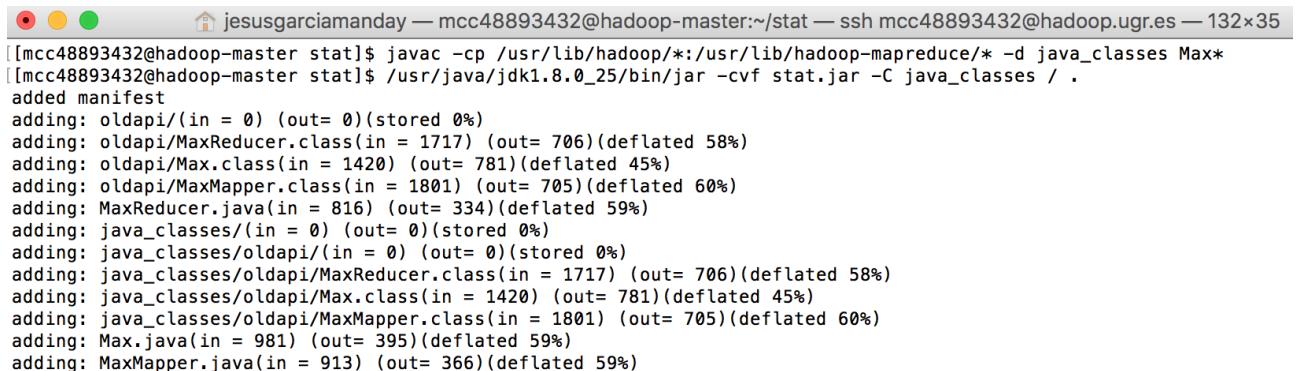
[jesusgarciamanday — mcc48893432@hadoop-master:~/stat — ssh mcc48893432@hadoop.ugr.es — 132x35]
GNU nano 2.0.9                               File: Max.java                         Modified

package oldapi;
import java.io.IOException;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
public class Max {
    public static void main(String[] args) throws IOException {
        if (args.length != 2) {
            System.err.println("Usage: MaxTemperature <input path> <output path>");
            System.exit(-1);
        }
        JobConf conf = new JobConf(Max.class);
        conf.setJobName("Max temperature");
        FileInputFormat.addInputPath(conf, new Path(args[0]));
        FileOutputFormat.setOutputPath(conf, new Path(args[1]));
        conf.setMapperClass(MaxMapper.class);
        conf.setReducerClass(MaxReducer.class);
        conf.setOutputKeyClass(Text.class);
        conf.setOutputValueClass(DoubleWritable.class);
        JobClient.runJob(conf);
    }
}

```

Figura 12: Clase Max.

Una vez que tenemos todos los ficheros procedemos a realizar los mismos pasos que con el ejercicio anterior compilando las clases, creando el fichero jar y ejecutandolo en **hadoop**.



```

[jesusgarciamanday — mcc48893432@hadoop-master:~/stat — ssh mcc48893432@hadoop.ugr.es — 132x35]
[[mcc48893432@hadoop-master stat]$ javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop-mapreduce/* -d java_classes Max*
[[mcc48893432@hadoop-master stat]$ /usr/java/jdk1.8.0_25/bin/jar -cvf stat.jar -C java_classes /
added manifest
adding: oldapi/(in = 0) (out= 0)(stored 0%)
adding: oldapi/MaxReducer.class(in = 1717) (out= 706)(deflated 58%)
adding: oldapi/Max.class(in = 1420) (out= 781)(deflated 45%)
adding: oldapi/MaxMapper.class(in = 1801) (out= 705)(deflated 60%)
adding: MaxReducer.java(in = 816) (out= 334)(deflated 59%)
adding: java_classes/(in = 0) (out= 0)(stored 0%)
adding: java_classes.oldapi/(in = 0) (out= 0)(stored 0%)
adding: java_classes.oldapi/MaxReducer.class(in = 1717) (out= 706)(deflated 58%)
adding: java_classes.oldapi/Max.class(in = 1420) (out= 781)(deflated 45%)
adding: java_classes.oldapi/MaxMapper.class(in = 1801) (out= 705)(deflated 60%)
adding: Max.java(in = 981) (out= 395)(deflated 59%)
adding: MaxMapper.java(in = 913) (out= 366)(deflated 59%)

```

Figura 13: Compilamos y ejecutamos (I).

```

[jesusgarciamanday — mcc48893432@hadoop-master:~] ssh mcc48893432@hadoop.ugr.es — 132x35
[mcc48893432@hadoop-master stat]$ hadoop jar stat.jar oldapi.Max /tmp/BDCC/datasets/ECBDL14/ECBDL14_10tst.data ./stat/output2/
17/05/20 12:57:54 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/192.168.10.1:8032
17/05/20 12:57:54 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/192.168.10.1:8032
17/05/20 12:57:55 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface
and execute your application with ToolRunner to remedy this.
17/05/20 12:57:55 INFO mapred.FileInputFormat: Total input paths to process : 1
17/05/20 12:57:55 INFO mapreduce.JobSubmitter: number of splits:2
17/05/20 12:57:55 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1494408081774_0246
17/05/20 12:57:55 INFO impl.YarnClientImpl: Submitted application application_1494408081774_0246
17/05/20 12:57:55 INFO mapreduce.Job: The url to track the job: http://hadoop.ugr.es:8088/proxy/application_1494408081774_0246/
17/05/20 12:57:55 INFO mapreduce.Job: Running job: job_1494408081774_0246
17/05/20 12:58:01 INFO mapreduce.Job: Job job_1494408081774_0246 running in uber mode : false
17/05/20 12:58:01 INFO mapreduce.Job: map 0% reduce 0%
17/05/20 12:58:11 INFO mapreduce.Job: map 100% reduce 0%
17/05/20 12:58:16 INFO mapreduce.Job: map 100% reduce 81%
17/05/20 12:58:19 INFO mapreduce.Job: map 100% reduce 88%
17/05/20 12:58:20 INFO mapreduce.Job: map 100% reduce 100%
17/05/20 12:58:20 INFO mapreduce.Job: Job job_1494408081774_0246 completed successfully
17/05/20 12:58:20 INFO mapreduce.Job: Counters: 49
  File System Counters
    FILE: Number of bytes read=2171063
    FILE: Number of bytes written=6526311
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=102749934
    HDFS: Number of bytes written=6
    HDFS: Number of read operations=54
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=32

Job Counters
  Launched map tasks=2
  Launched reduce tasks=16
  Rack-local map tasks=2
  Total time spent by all maps in occupied slots (ms)=111125
  Total time spent by all reduces in occupied slots (ms)=1915900
  Total time spent by all map tasks (ms)=15875
  Total time spent by all reduce tasks (ms)=39100
  Total vcore-seconds taken by all map tasks=15875
  Total vcore-seconds taken by all reduce tasks=39100
  Total megabyte-seconds taken by all map tasks=111125000
  Total megabyte-seconds taken by all reduce tasks=1955000000

Map-Reduce Framework
  Map input records=2897917
  Map output records=2897917
  Map output bytes=28979170
  Map output materialized bytes=2170940
  Input split bytes=234
  Combine input records=0
  Combine output records=0
  Reduce input groups=1
  Reduce shuffle bytes=2170940
  Reduce input records=2897917
  Reduce output records=1
  Spilled Records=5795834
  Shuffled Maps =32
  Failed Shuffles=0
  Merged Map outputs=32
  GC time elapsed (ms)=327
  CPU time spent (ms)=36200
  Physical memory (bytes) snapshot=7933112320
  Virtual memory (bytes) snapshot=984107524096
  Total committed heap usage (bytes)=19421724672

```

Figura 14: Compilamos y ejecutamos (II).

```

Job Counters
  Launched map tasks=2
  Launched reduce tasks=16
  Rack-local map tasks=2
  Total time spent by all maps in occupied slots (ms)=111125
  Total time spent by all reduces in occupied slots (ms)=1915900
  Total time spent by all map tasks (ms)=15875
  Total time spent by all reduce tasks (ms)=39100
  Total vcore-seconds taken by all map tasks=15875
  Total vcore-seconds taken by all reduce tasks=39100
  Total megabyte-seconds taken by all map tasks=111125000
  Total megabyte-seconds taken by all reduce tasks=1955000000

Map-Reduce Framework
  Map input records=2897917
  Map output records=2897917
  Map output bytes=28979170
  Map output materialized bytes=2170940
  Input split bytes=234
  Combine input records=0
  Combine output records=0
  Reduce input groups=1
  Reduce shuffle bytes=2170940
  Reduce input records=2897917
  Reduce output records=1
  Spilled Records=5795834
  Shuffled Maps =32
  Failed Shuffles=0
  Merged Map outputs=32
  GC time elapsed (ms)=327
  CPU time spent (ms)=36200
  Physical memory (bytes) snapshot=7933112320
  Virtual memory (bytes) snapshot=984107524096
  Total committed heap usage (bytes)=19421724672

```

Figura 15: Compilamos y ejecutamos (III).

```

Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=102749700
File Output Format Counters
  Bytes Written=6

```

Figura 16: Compilamos y ejecutamos (IV).

Comprobamos que el resultado nos arroja el valor máximo.

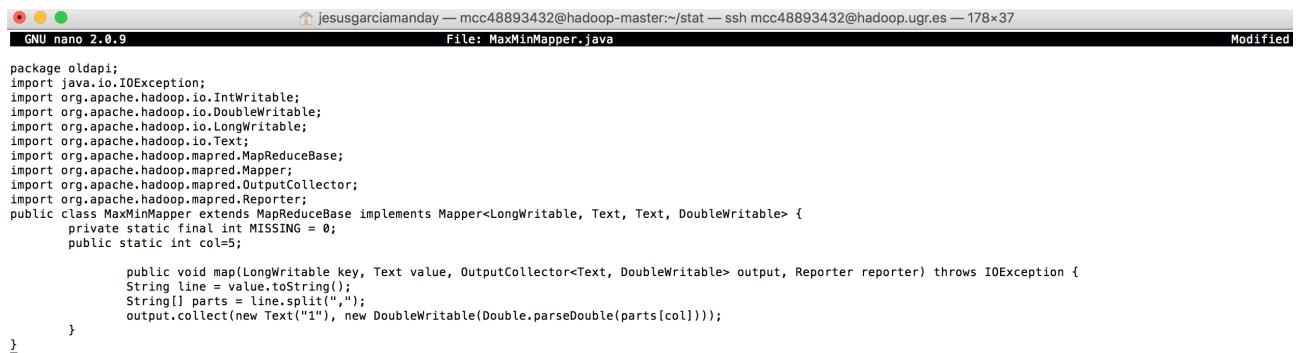


```
[mcc48893432@hadoop-master stat]$ hdfs dfs -cat stat/output2/*
1      9.0
```

Figura 17: Comprobando resultado.

## 5. Calcula al mismo tiempo los valores máximo y mínimo de la variable 5.

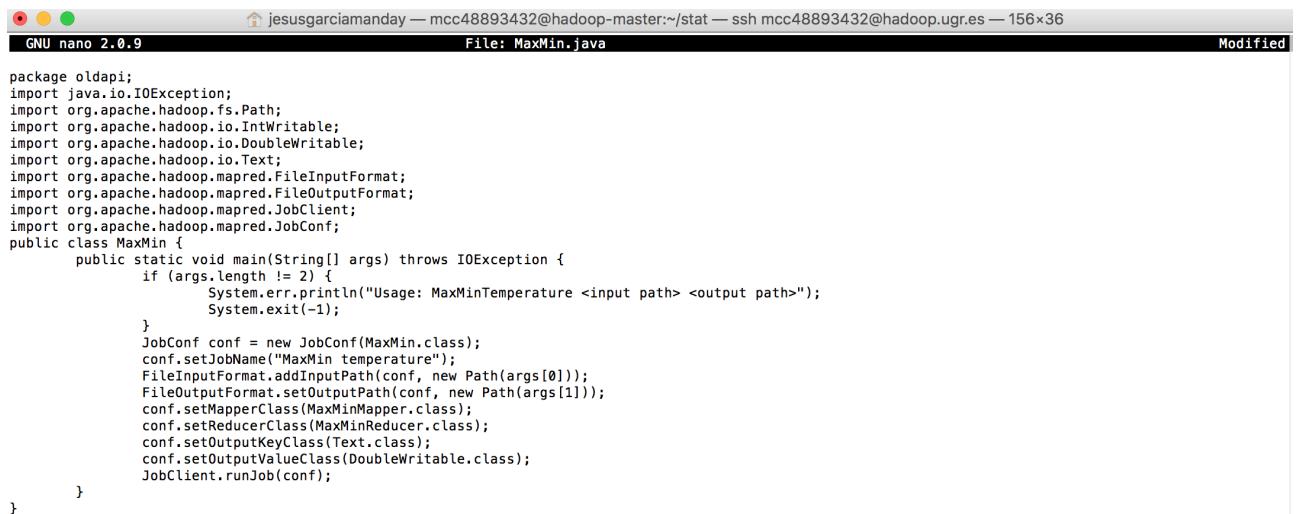
Al igual que para los anteriores cálculos creamos los ficheros correspondientes, aunque el fichero con la función **Mapper** y el principal con el main solo cambia el nombre, en la clase con la función **Reducer** es donde se realizan los cambios necesarios para obtener los resultados esperados.



```

jeshusgarciamanday — mcc48893432@hadoop-master:~/stat — ssh mcc48893432@hadoop.ugr.es — 178x37
GNU nano 2.0.9          File: MaxMinMapper.java          Modified
package oldapi;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;
public class MaxMinMapper extends MapReduceBase implements Mapper<LongWritable, Text, Text, DoubleWritable> {
    private static final int MISSING = 0;
    public static int col=5;
    public void map(LongWritable key, Text value, OutputCollector<Text, DoubleWritable> output, Reporter reporter) throws IOException {
        String line = value.toString();
        String[] parts = line.split(",");
        output.collect(new Text("1"), new DoubleWritable(Double.parseDouble(parts[col])));
    }
}
```

Figura 18: Clase MaxMinMapper.



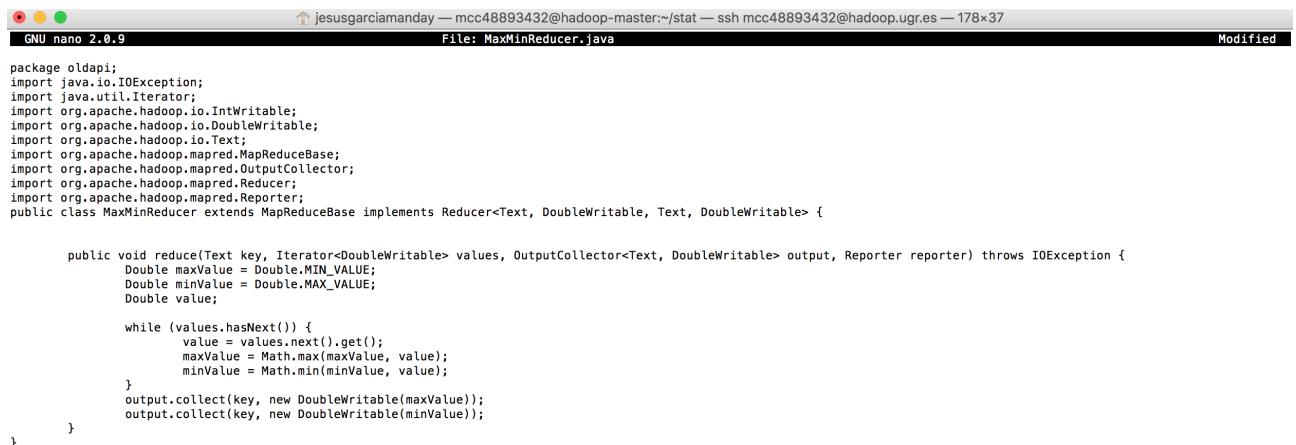
```

GNU nano 2.0.9          File: MaxMin.java          Modified

package oldapi;
import java.io.IOException;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
public class MaxMin {
    public static void main(String[] args) throws IOException {
        if (args.length != 2) {
            System.err.println("Usage: MaxMinTemperature <input path> <output path>");
            System.exit(-1);
        }
        JobConf conf = new JobConf(MaxMin.class);
        conf.setJobName("MaxMin temperature");
        FileInputFormat.addInputPath(conf, new Path(args[0]));
        FileOutputFormat.setOutputPath(conf, new Path(args[1]));
        conf.setMapperClass(MaxMinMapper.class);
        conf.setReducerClass(MaxMinReducer.class);
        conf.setOutputKeyClass(Text.class);
        conf.setOutputValueClass(DoubleWritable.class);
        JobClient.runJob(conf);
    }
}

```

Figura 19: Clase MaxMin.



```

GNU nano 2.0.9          File: MaxMinReducer.java          Modified

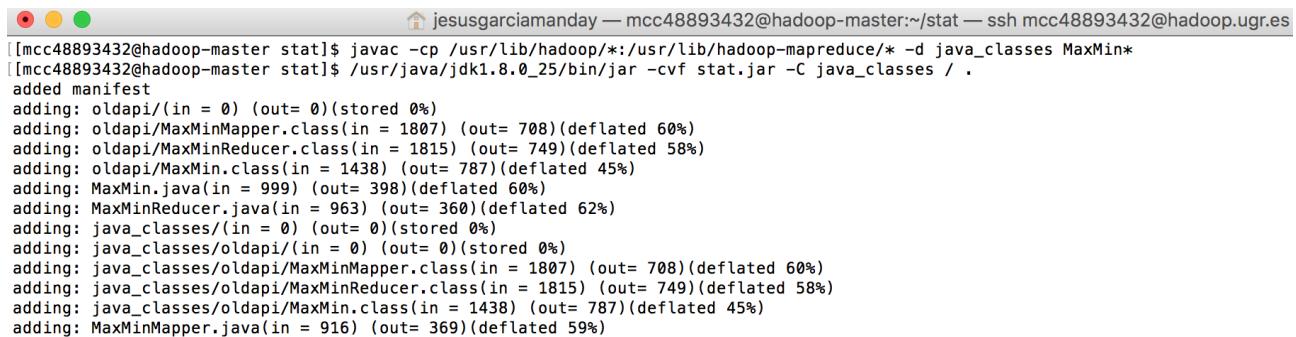
package oldapi;
import java.io.IOException;
import java.util.Iterator;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;
public class MaxMinReducer extends MapReduceBase implements Reducer<Text, DoubleWritable, Text, DoubleWritable> {

    public void reduce(Text key, Iterator<DoubleWritable> values, OutputCollector<Text, DoubleWritable> output, Reporter reporter) throws IOException {
        Double maxValue = Double.MIN_VALUE;
        Double minValue = Double.MAX_VALUE;
        Double value;
        while (values.hasNext()) {
            value = values.next().get();
            maxValue = Math.max(maxValue, value);
            minValue = Math.min(minValue, value);
        }
        output.collect(key, new DoubleWritable(maxValue));
        output.collect(key, new DoubleWritable(minValue));
    }
}

```

Figura 20: Clase MaxMinReducer.

Ahora toca compilarlos para adjuntarlos en el fichero **jar** y realizar la ejecución en **hadoop**.



```

[jesugarciamanday@mcc48893432 hadoop-master]$ javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop-mapreduce/* -d java_classes MaxMin*
[jesugarciamanday@mcc48893432 hadoop-master]$ /usr/java/jdk1.8.0_25/bin/jar -cvf stat.jar -C java_classes /
added manifest
adding: oldapi/(in = 0) (out= 0)(stored 0%)
adding: oldapi/MaxMinMapper.class(in = 1807) (out= 708)(deflated 60%)
adding: oldapi/MaxMinReducer.class(in = 1815) (out= 749)(deflated 58%)
adding: oldapi/MaxMin.class(in = 1438) (out= 787)(deflated 45%)
adding: MaxMin.java(in = 999) (out= 398)(deflated 60%)
adding: MaxMinReducer.java(in = 963) (out= 360)(deflated 62%)
adding: java_classes/(in = 0) (out= 0)(stored 0%)
adding: java_classes.oldapi/(in = 0) (out= 0)(stored 0%)
adding: java_classes.oldapi/MaxMinMapper.class(in = 1807) (out= 708)(deflated 60%)
adding: java_classes.oldapi/MaxMinReducer.class(in = 1815) (out= 749)(deflated 58%)
adding: java_classes.oldapi/MaxMin.class(in = 1438) (out= 787)(deflated 45%)
adding: MaxMinMapper.java(in = 916) (out= 369)(deflated 59%)

```

Figura 21: Compilamos y ejecutamos (I).

```
[mcc48893432@hadoop-master stat]$ hadoop jar stat.jar oldapi.MaxMin /tmp/BDCC/datasets/ECBOL14/ECBOL14_10tst.data ./stat/output5/
17/05/20 17:03:15 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/192.168.10.1:8032
17/05/20 17:03:15 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/192.168.10.1:8032
17/05/20 17:03:15 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
17/05/20 17:03:15 INFO mapred.FileInputFormat: Total input paths to process : 1
17/05/20 17:03:16 INFO mapreduce.JobSubmitter: number of splits:2
17/05/20 17:03:16 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1494408081774_0267
17/05/20 17:03:16 INFO impl.YarnClientImpl: Submitted application application_1494408081774_0267
17/05/20 17:03:16 INFO mapreduce.Job: The url to track the job: http://hadoop.ugr.es:8088/proxy/application_1494408081774_0267/
17/05/20 17:03:16 INFO mapreduce.Job: Running job: job_1494408081774_0267
17/05/20 17:03:20 INFO mapreduce.Job: Job job_1494408081774_0267 running in uber mode : false
17/05/20 17:03:20 INFO mapreduce.Job: map 0% reduce 0%
17/05/20 17:03:30 INFO mapreduce.Job: map 100% reduce 0%
17/05/20 17:03:35 INFO mapreduce.Job: map 100% reduce 63%
17/05/20 17:03:36 INFO mapreduce.Job: map 100% reduce 81%
17/05/20 17:03:38 INFO mapreduce.Job: map 100% reduce 88%
17/05/20 17:03:40 INFO mapreduce.Job: map 100% reduce 100%
17/05/20 17:03:40 INFO mapreduce.Job: Job job_1494408081774_0267 completed successfully
17/05/20 17:03:40 INFO mapreduce.Job: Counters: 50
File System Counters
  FILE: Number of bytes read=2121182
  FILE: Number of bytes written=6427147
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=102749934
  HDFS: Number of bytes written=14
  HDFS: Number of read operations=54
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=32
System Counters
Job Counters
  Launched map tasks=2
  Launched reduce tasks=16
  Data-local map tasks=1
  Rack-local map tasks=1
  Total time spent by all maps in occupied slots (ms)=111153
  Total time spent by all reduces in occupied slots (ms)=1901494
  Total time spent by all map tasks (ms)=15879
  Total time spent by all reduce tasks (ms)=38806
  Total vcore-seconds taken by all map tasks=15879
  Total vcore-seconds taken by all reduce tasks=38806
  Total megabyte-seconds taken by all map tasks=111153000
  Total megabyte-seconds taken by all reduce tasks=1940300000
Map-Reduce Framework
  Map input records=2897917
  Map output records=2897917
  Map output bytes=28979170
  Map output materialized bytes=2121495
  Input split bytes=234
  Combine input records=0
  Combine output records=0
  Reduce input groups=1
  Reduce shuffle bytes=2121495
  Reduce input records=2897917
  Reduce output records=2
  Spilled Records=5795834
  Shuffled Maps =32
  Failed Shuffles=0
  Merged Map outputs=32
  GC time elapsed (ms)=367
  CPU time spent (ms)=36140
  Physical memory (bytes) snapshot=914434560
  Virtual memory (bytes) snapshot=984147337216
  Total committed heap usage (bytes)=19421724672
```

Figura 22: Compilamos y ejecutamos (II).

```
Job Counters
  Launched map tasks=2
  Launched reduce tasks=16
  Data-local map tasks=1
  Rack-local map tasks=1
  Total time spent by all maps in occupied slots (ms)=111153
  Total time spent by all reduces in occupied slots (ms)=1901494
  Total time spent by all map tasks (ms)=15879
  Total time spent by all reduce tasks (ms)=38806
  Total vcore-seconds taken by all map tasks=15879
  Total vcore-seconds taken by all reduce tasks=38806
  Total megabyte-seconds taken by all map tasks=111153000
  Total megabyte-seconds taken by all reduce tasks=1940300000
Map-Reduce Framework
  Map input records=2897917
  Map output records=2897917
  Map output bytes=28979170
  Map output materialized bytes=2121495
  Input split bytes=234
  Combine input records=0
  Combine output records=0
  Reduce input groups=1
  Reduce shuffle bytes=2121495
  Reduce input records=2897917
  Reduce output records=2
  Spilled Records=5795834
  Shuffled Maps =32
  Failed Shuffles=0
  Merged Map outputs=32
  GC time elapsed (ms)=367
  CPU time spent (ms)=36140
  Physical memory (bytes) snapshot=914434560
  Virtual memory (bytes) snapshot=984147337216
  Total committed heap usage (bytes)=19421724672
```

Figura 23: Compilamos y ejecutamos (III).

```
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=102749700
File Output Format Counters
  Bytes Written=14
```

Figura 24: Compilamos y ejecutamos (IV).

Por último comprobamos el resultado obtenido.

```
[mcc48893432@hadoop-master stat]$ hdfs dfs -cat stat/output5/*
1      9.0
1     -11.0
```

Figura 25: Comprobando resultado.

## 6. Calcula los valores máximo y mínimo de todas las variables (salvo la última, que es la etiqueta de la clase).

Para este cálculo solo necesitaremos adaptar la clase con la función **Mapper** para cada columna ya que el **Reducer** es el mismo. Ahora la función **Mapper** asignará diferentes key, uno por cada columna.

```
GNU nano 2.0.9 jesusgarciamanday — mcc48893432@hadoop-master:~/stat — ssh mcc48893432@hadoop.ugr.es — 132x35 Modified
File: MaxMinMapper.java

package oldapi;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;
public class MaxMinMapper extends MapReduceBase implements Mapper<LongWritable, Text, Text, DoubleWritable> {
    private static final int MISSING = 0;
    public static int col=5;

    public void map(LongWritable key, Text value, OutputCollector<Text, DoubleWritable> output, Reporter reporter) throws IOException {
        String line = value.toString();
        String[] parts = line.split(",");
        for(int i = 1; i < parts.length; i++){
            output.collect(new Text(String.valueOf(i)), new DoubleWritable(Double.parseDouble(parts[i-1])));
        }
    }
}
```

Figura 26: Clase MaxMinMapper.

Como en los anteriores ejercicios, compilamos y ejecutamos las clases.

```
[mcc48893432@hadoop-master stat]$ javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop-mapreduce/* -d java_classes MaxMin*
[mcc48893432@hadoop-master stat]$ /usr/java/jdk1.8.0_25/bin/jar -cvf stat.jar -C java_classes /
added manifest
adding: oldapi/(in = 0) (out= 0)(stored 0%)
adding: oldapi/MaxMinMapper.class(in = 1938) (out= 790)(deflated 59%)
adding: oldapi/MaxMinReducer.class(in = 1824) (out= 762)(deflated 58%)
adding: oldapi/MaxMin.class(in = 1438) (out= 787)(deflated 45%)
adding: MaxMin.java(in = 999) (out= 398)(deflated 60%)
adding: MaxMinReducer.java(in = 986) (out= 372)(deflated 62%)
adding: java_classes/(in = 0) (out= 0)(stored 0%)
adding: java_classes/oldapi/(in = 0) (out= 0)(stored 0%)
adding: java_classes/oldapi/MaxMinMapper.class(in = 1938) (out= 790)(deflated 59%)
adding: java_classes/oldapi/MaxMinReducer.class(in = 1824) (out= 762)(deflated 58%)
adding: java_classes/oldapi/MaxMin.class(in = 1438) (out= 787)(deflated 45%)
adding: MaxMinMapper.java(in = 964) (out= 410)(deflated 57%)
```

Figura 27: Compilamos y ejecutamos (I).

```
[mcc48893432@hadoop-master stat]$ hadoop jar stat.jar oldapi.MaxMin /tmp/BDCC/datasets/ECBDL14/ECBDL14_10tst.data ./stat/output6/
17/05/20 17:46:17 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/192.168.10.1:8032
17/05/20 17:46:17 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/192.168.10.1:8032
17/05/20 17:46:17 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and
execute your application with ToolRunner to remedy this.
17/05/20 17:46:17 INFO mapred.FileInputFormat: Total input paths to process : 1
17/05/20 17:46:18 INFO mapreduce.JobSubmitter: number of splits:2
17/05/20 17:46:18 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1494408081774_0283
17/05/20 17:46:18 INFO impl.YarnClientImpl: Submitted application application_1494408081774_0283
17/05/20 17:46:18 INFO mapreduce.Job: The url to track the job: http://hadoop.ugr.es:8088/proxy/application_1494408081774_0283/
17/05/20 17:46:18 INFO mapreduce.Job: Running job: job_1494408081774_0283
17/05/20 17:46:22 INFO mapreduce.Job: Job job_1494408081774_0283 running in uber mode : false
17/05/20 17:46:22 INFO mapreduce.Job: map 0% reduce 0%
17/05/20 17:46:32 INFO mapreduce.Job: map 32% reduce 0%
17/05/20 17:46:35 INFO mapreduce.Job: map 45% reduce 0%
17/05/20 17:46:38 INFO mapreduce.Job: map 58% reduce 0%
17/05/20 17:46:41 INFO mapreduce.Job: map 65% reduce 0%
17/05/20 17:46:44 INFO mapreduce.Job: map 71% reduce 0%
17/05/20 17:46:47 INFO mapreduce.Job: map 79% reduce 0%
17/05/20 17:46:49 INFO mapreduce.Job: map 89% reduce 0%
17/05/20 17:46:50 INFO mapreduce.Job: map 92% reduce 0%
17/05/20 17:46:51 INFO mapreduce.Job: map 100% reduce 0%
17/05/20 17:46:54 INFO mapreduce.Job: map 100% reduce 25%
17/05/20 17:46:55 INFO mapreduce.Job: map 100% reduce 31%
17/05/20 17:46:57 INFO mapreduce.Job: map 100% reduce 81%
17/05/20 17:46:58 INFO mapreduce.Job: map 100% reduce 100%
17/05/20 17:46:59 INFO mapreduce.Job: Job job_1494408081774_0283 completed successfully
17/05/20 17:46:59 INFO mapreduce.Job: Counters: 50
  File System Counters
    FILE: Number of bytes read=45717534
    FILE: Number of bytes written=70501121
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=102749934
    HDFS: Number of bytes written=146
    HDFS: Number of read operations=54
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=32
```

Figura 28: Compilamos y ejecutamos (II).

```

Job Counters
    Launched map tasks=2
    Launched reduce tasks=16
    Data-local map tasks=1
    Rack-local map tasks=1
    Total time spent by all maps in occupied slots (ms)=368760
    Total time spent by all reduces in occupied slots (ms)=3107090
    Total time spent by all map tasks (ms)=52680
    Total time spent by all reduce tasks (ms)=63410
    Total vcore-seconds taken by all map tasks=52680
    Total vcore-seconds taken by all reduce tasks=63410
    Total megabyte-seconds taken by all map tasks=368760000
    Total megabyte-seconds taken by all reduce tasks=3170500000
Map-Reduce Framework
    Map input records=2897917
    Map output records=28979170
    Map output bytes=292689617
    Map output materialized bytes=22770570
    Input split bytes=234
    Combine input records=0
    Combine output records=0
    Reduce input groups=10
    Reduce shuffle bytes=22770570
    Reduce input records=28979170
    Reduce output records=20
    Spilled Records=86937510
    Shuffled Maps =32
    Failed Shuffles=0
    Merged Map outputs=32
    GC time elapsed (ms)=438
    CPU time spent (ms)=126880
    Physical memory (bytes) snapshot=10873622528
    Virtual memory (bytes) snapshot=984140087296
    Total committed heap usage (bytes)=21354250240
Shuffle Errors
    BAD_ID=0
    CONNECTION=0
    IO_ERROR=0
    WRONG_LENGTH=0
    WRONG_MAP=0
    WRONG_REDUCE=0
File Input Format Counters
    Bytes Read=102749700
File Output Format Counters
    Bytes Written=146

```

Figura 29: Compilamos y ejecutamos (III).

Vemos el resultado que se ha obtenido.

```

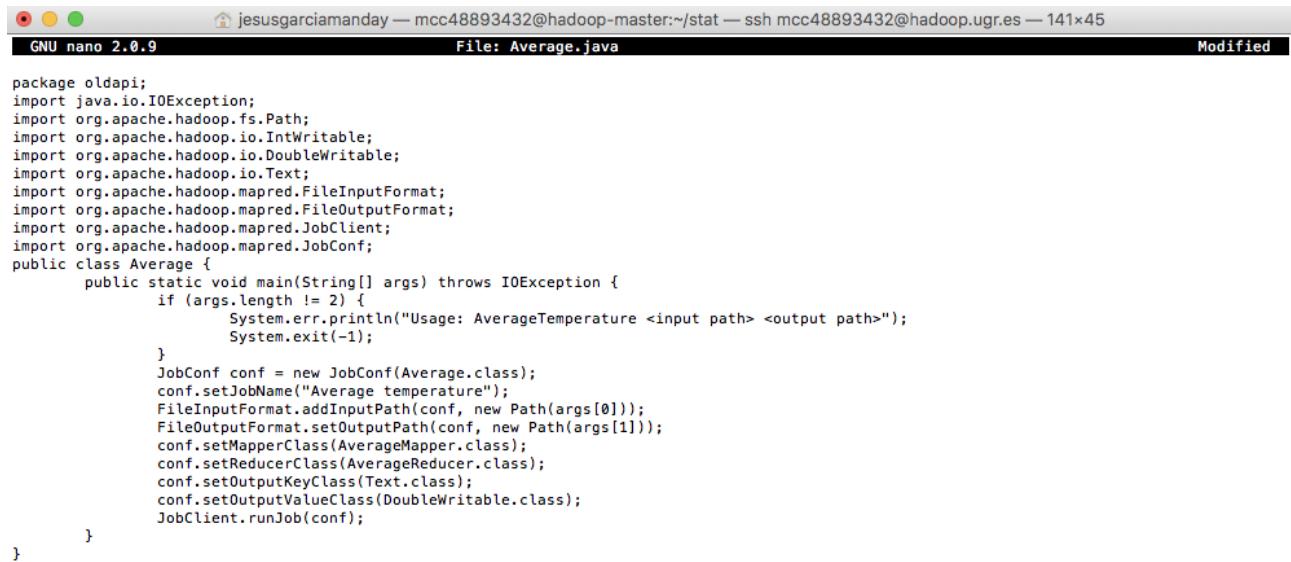
[jesusgarciamanday — mcc48893432@hadoop-master:~/stat — ssh mcc48893432@hadoop.ugr.es]
[[mcc48893432@hadoop-master stat]$ hdfs dfs -cat stat/output6/*
1      0.768
1      0.094
10     10.0
10     -13.0
2      0.154
2      0.0
3      10.0
3      -12.0
4      8.0
4      -11.0
5      9.0
5      -12.0
6      9.0
6      -11.0
7      9.0
7      -13.0
8      9.0
8      -12.0
9      7.0
9      -12.0

```

Figura 30: Comprobando resultado.

## 7. Realizar la media de la variable 5.

Para realizar esta tarea será necesario crear dos funciones nuevas para **Mapper** y **Reducer** con sus respectivas clases ya que difieren de las tareas anteriores.



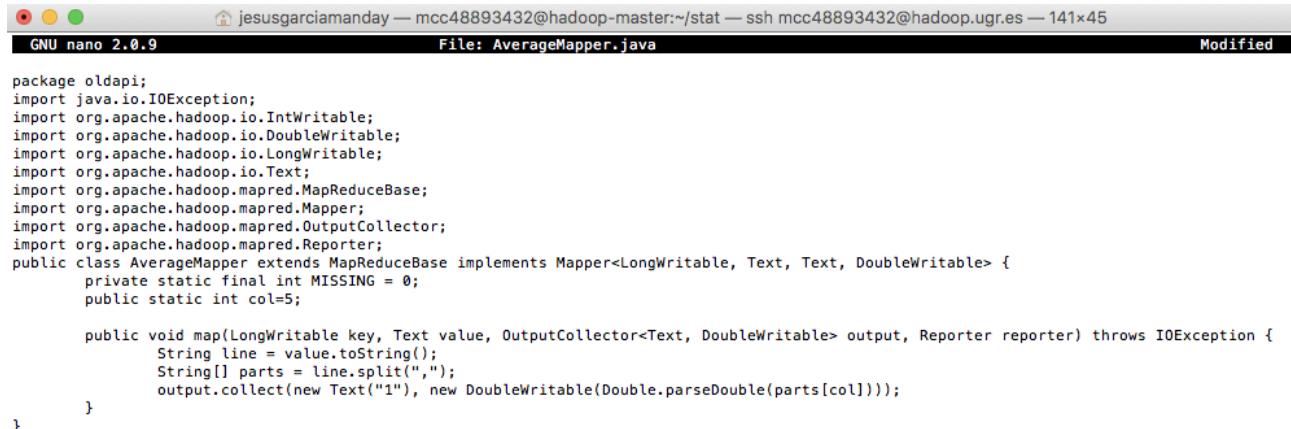
```

[jesugarciamanday — mcc48893432@hadoop-master:~/stat — ssh mcc48893432@hadoop.ugr.es — 141x45]
GNU nano 2.0.9 File: Average.java Modified

package oldapi;
import java.io.IOException;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
public class Average {
    public static void main(String[] args) throws IOException {
        if (args.length != 2) {
            System.err.println("Usage: AverageTemperature <input path> <output path>");
            System.exit(-1);
        }
        JobConf conf = new JobConf(Average.class);
        conf.setJobName("Average temperature");
        FileInputFormat.addInputPath(conf, new Path(args[0]));
        FileOutputFormat.setOutputPath(conf, new Path(args[1]));
        conf.setMapperClass(AverageMapper.class);
        conf.setReducerClass(AverageReducer.class);
        conf.setOutputKeyClass(Text.class);
        conf.setOutputValueClass(DoubleWritable.class);
        JobClient.runJob(conf);
    }
}

```

Figura 31: Clase Average.



```

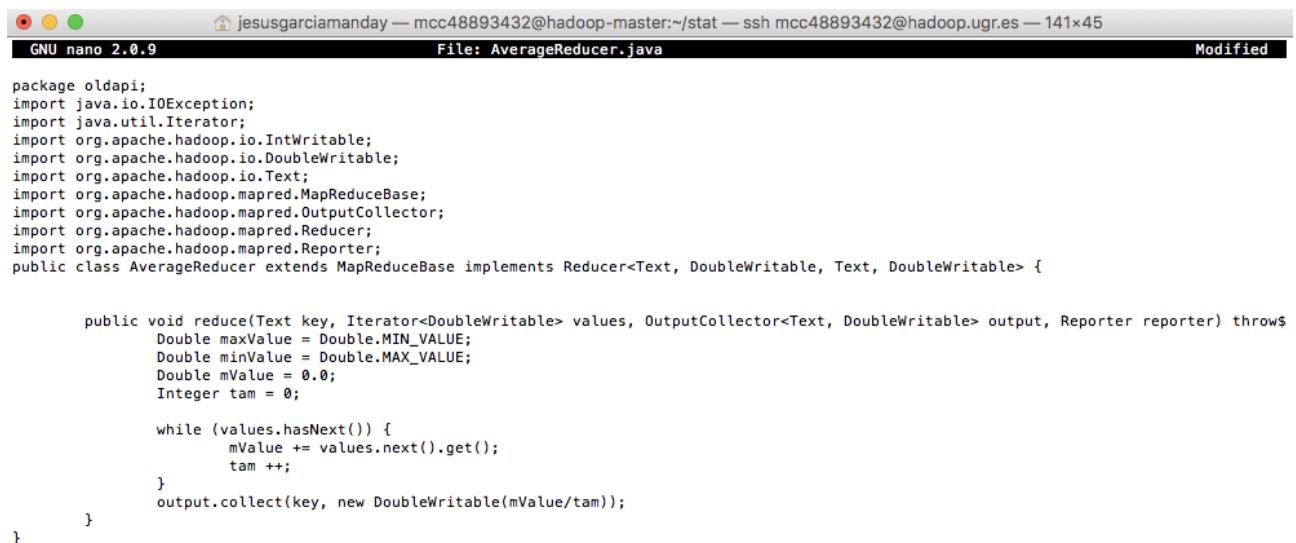
[jesugarciamanday — mcc48893432@hadoop-master:~/stat — ssh mcc48893432@hadoop.ugr.es — 141x45]
GNU nano 2.0.9 File: AverageMapper.java Modified

package oldapi;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;
public class AverageMapper extends MapReduceBase implements Mapper<LongWritable, Text, Text, DoubleWritable> {
    private static final int MISSING = 0;
    public static int col=5;

    public void map(LongWritable key, Text value, OutputCollector<Text, DoubleWritable> output, Reporter reporter) throws IOException {
        String line = value.toString();
        String[] parts = line.split(",");
        output.collect(new Text("1"), new DoubleWritable(Double.parseDouble(parts[col])));
    }
}

```

Figura 32: Clase AverageMapper.



```

GNU nano 2.0.9          File: AverageReducer.java           Modified

package oldapi;
import java.io.IOException;
import java.util.Iterator;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;
public class AverageReducer extends MapReduceBase implements Reducer<Text, DoubleWritable, Text, DoubleWritable> {

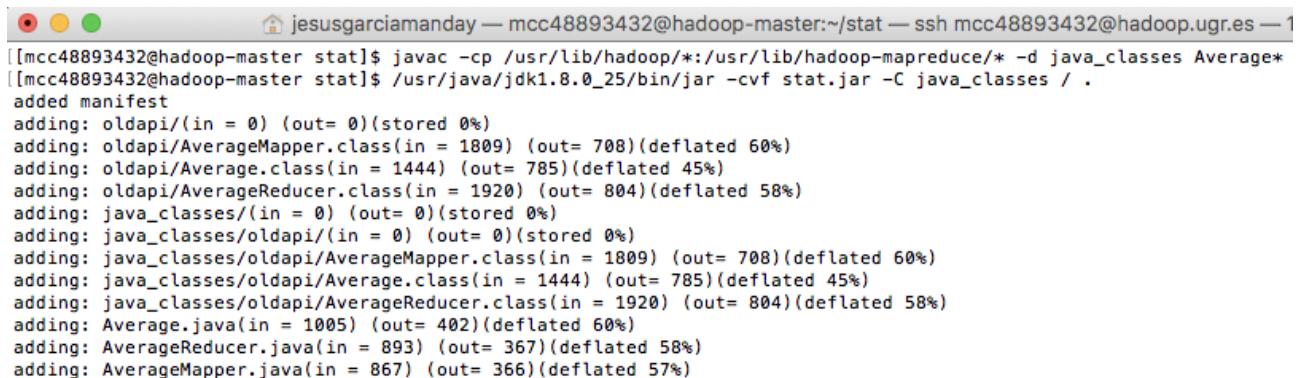
    public void reduce(Text key, Iterator<DoubleWritable> values, OutputCollector<Text, DoubleWritable> output, Reporter reporter) throws IOException {
        Double maxValue = Double.MIN_VALUE;
        Double minValue = Double.MAX_VALUE;
        Double mValue = 0.0;
        Integer tam = 0;

        while (values.hasNext()) {
            mValue += values.next().get();
            tam++;
        }
        output.collect(key, new DoubleWritable(mValue/tam));
    }
}

```

Figura 33: Clase AverageReducer.

Con las funciones creadas en su correspondiente clase pasamos a compilar y a ejecutar.



```

[jesugarciamanday@mcc48893432 hadoop-master]$ javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop-mapreduce/* -d java_classes Average*
[jesugarciamanday@mcc48893432 hadoop-master]$ /usr/java/jdk1.8.0_25/bin/jar -cvf stat.jar -C java_classes /
added manifest
adding: oldapi/(in = 0) (out= 0)(stored 0%)
adding: oldapi/AverageMapper.class(in = 1809) (out= 708)(deflated 60%)
adding: oldapi/Average.class(in = 1444) (out= 785)(deflated 45%)
adding: oldapi/AverageReducer.class(in = 1920) (out= 804)(deflated 58%)
adding: java_classes/(in = 0) (out= 0)(stored 0%)
adding: java_classes.oldapi/(in = 0) (out= 0)(stored 0%)
adding: java_classes.oldapi/AverageMapper.class(in = 1809) (out= 708)(deflated 60%)
adding: java_classes.oldapi/Average.class(in = 1444) (out= 785)(deflated 45%)
adding: java_classes.oldapi/AverageReducer.class(in = 1920) (out= 804)(deflated 58%)
adding: Average.java(in = 1005) (out= 402)(deflated 60%)
adding: AverageReducer.java(in = 893) (out= 367)(deflated 58%)
adding: AverageMapper.java(in = 867) (out= 366)(deflated 57%)

```

Figura 34: Compilamos y ejecutamos (I).

```
[mcc48893432@hadoop-master stat]$ hadoop jar stat.jar Average /tmp/BDCC/datasets/ECBDL14/ECBDL14_10tst.data ./stat/output8/
17/05/20 18:21:37 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/192.168.10.1:8032
17/05/20 18:21:37 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/192.168.10.1:8032
17/05/20 18:21:37 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and exec
ute your application with ToolRunner to remedy this.
17/05/20 18:21:37 INFO mapred.FileInputFormat: Total input paths to process : 1
17/05/20 18:21:38 INFO mapreduce.JobSubmitter: number of splits:2
17/05/20 18:21:38 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1494408081774_0290
17/05/20 18:21:38 INFO impl.YarnClientImpl: Submitted application application_1494408081774_0290
17/05/20 18:21:38 INFO mapreduce.Job: The url to track the job: http://hadoop.ugr.es:8088/proxy/application_1494408081774_0290/
17/05/20 18:21:38 INFO mapreduce.Job: Running job: job_1494408081774_0290
17/05/20 18:21:42 INFO mapreduce.Job: Job job_1494408081774_0290 running in uber mode : false
17/05/20 18:21:42 INFO mapreduce.Job: map 0% reduce 0%
17/05/20 18:21:52 INFO mapreduce.Job: map 100% reduce 0%
17/05/20 18:21:56 INFO mapreduce.Job: map 100% reduce 75%
17/05/20 18:21:57 INFO mapreduce.Job: map 100% reduce 81%
17/05/20 18:21:59 INFO mapreduce.Job: map 100% reduce 88%
17/05/20 18:22:00 INFO mapreduce.Job: map 100% reduce 100%
17/05/20 18:22:01 INFO mapreduce.Job: Job job_1494408081774_0290 completed successfully
17/05/20 18:22:01 INFO mapreduce.Job: Counters: 50
File System Counters
    FILE: Number of bytes read=2168431
    FILE: Number of bytes written=6521412
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=102749934
    HDFS: Number of bytes written=21
    HDFS: Number of read operations=54
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=32
Job Counters
    Launched map tasks=2
    Launched reduce tasks=16
    Data-local map tasks=1
    Rack-local map tasks=1
    Total time spent by all maps in occupied slots (ms)=108864
    Total time spent by all reduces in occupied slots (ms)=1851759
    Total time spent by all map tasks (ms)=15552
    Total time spent by all reduce tasks (ms)=37791
    Total vcore-seconds taken by all map tasks=15552
    Total vcore-seconds taken by all reduce tasks=37791
    Total megabyte-seconds taken by all map tasks=108864000
    Total megabyte-seconds taken by all reduce tasks=1889550000
```

Figura 35: Compilamos y ejecutamos (I).

```
Map-Reduce Framework
    Map input records=2897917
    Map output records=2897917
    Map output bytes=28979170
    Map output materialized bytes=2168457
    Input split bytes=234
    Combine input records=0
    Combine output records=0
    Reduce input groups=1
    Reduce shuffle bytes=2168457
    Reduce input records=2897917
    Reduce output records=1
    Spilled Records=5795834
    Shuffled Maps =32
    Failed Shuffles=0
    Merged Map outputs=32
    GC time elapsed (ms)=337
    CPU time spent (ms)=35700
    Physical memory (bytes) snapshot=7900184576
    Virtual memory (bytes) snapshot=984125603840
    Total committed heap usage (bytes)=19167969280
Shuffle Errors
    BAD_ID=0
    CONNECTION=0
    IO_ERROR=0
    WRONG_LENGTH=0
    WRONG_MAP=0
    WRONG_REDUCE=0
File Input Format Counters
    Bytes Read=102749700
File Output Format Counters
    Bytes Written=21
```

Figura 36: Compilamos y ejecutamos (III).

Y por último comprobamos el resultado.

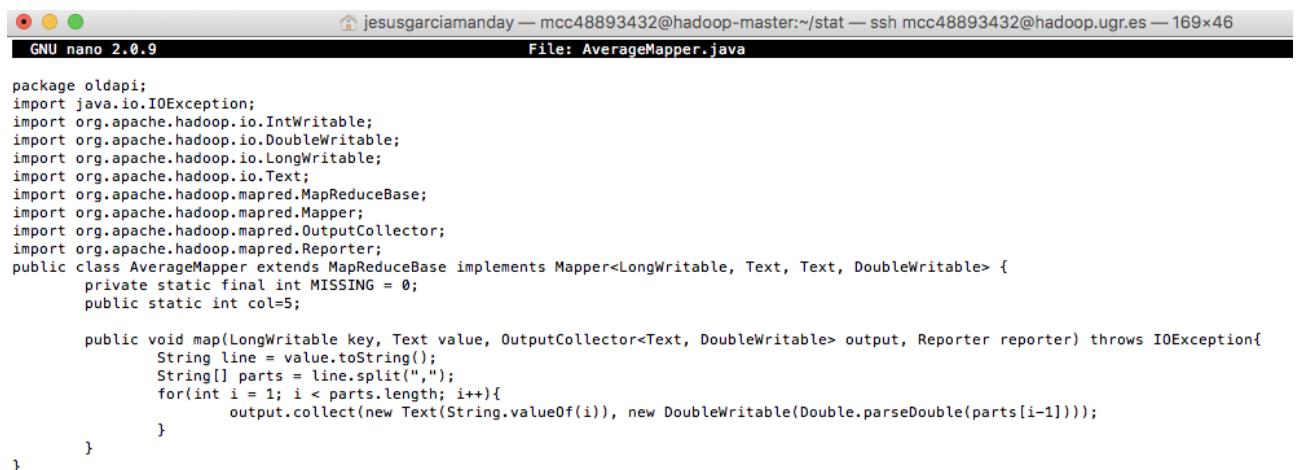


```
[mcc48893432@hadoop-master stat]$ hdfs dfs -cat stat/output8/*
1      -1.282261707288373
```

Figura 37: Comprobando resultado.

## 8. Obtener la media de todas las variables (salvo la clase).

En esta tarea solo necesitamos modificar la función **Mapper** con respecto al ejercicio anterior donde la función **Reduce** será la misma, por lo que será necesario modificar únicamente un fichero.



```
GNU nano 2.0.9          File: AverageMapper.java

package oldapi;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;
public class AverageMapper extends MapReduceBase implements Mapper<LongWritable, Text, Text, DoubleWritable> {
    private static final int MISSING = 0;
    public static int col=5;

    public void map(LongWritable key, Text value, OutputCollector<Text, DoubleWritable> output, Reporter reporter) throws IOException{
        String line = value.toString();
        String[] parts = line.split(",");
        for(int i = 1; i < parts.length; i++){
            output.collect(new Text(String.valueOf(i)), new DoubleWritable(Double.parseDouble(parts[i-1])));
        }
    }
}
```

Figura 38: Clase AverageMapper.

Una vez modificado el método **Mapper** podemos compilar y ejecutar.

```
[mcc48893432@hadoop-master stat]$ hadoop jar stat.jar oldapi.Average /tmp/BDCC/datasets/ECBDL14/ECBDL14_10tst.data ./stat/output9/
17/05/20 18:39:41 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/192.168.10.1:8032
17/05/20 18:39:41 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/192.168.10.1:8032
17/05/20 18:39:42 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
17/05/20 18:39:42 INFO mapred.FileInputFormat: Total input paths to process : 1
17/05/20 18:39:42 INFO mapreduce.JobSubmitter: number of splits:2
17/05/20 18:39:42 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1494408081774_0291
17/05/20 18:39:43 INFO impl.YarnClientImpl: Submitted application application_1494408081774_0291
17/05/20 18:39:43 INFO mapreduce.Job: The url to track the job: http://hadoop.ugr.es:8088/proxy/application_1494408081774_0291/
17/05/20 18:39:43 INFO mapreduce.Job: Running job: job_1494408081774_0291
17/05/20 18:39:47 INFO mapreduce.Job: Job job_1494408081774_0291 running in uber mode : false
17/05/20 18:39:47 INFO mapreduce.Job: map 0% reduce 0%
17/05/20 18:39:57 INFO mapreduce.Job: map 32% reduce 0%
17/05/20 18:40:01 INFO mapreduce.Job: map 45% reduce 0%
17/05/20 18:40:03 INFO mapreduce.Job: map 58% reduce 0%
17/05/20 18:40:04 INFO mapreduce.Job: map 66% reduce 0%
17/05/20 18:40:05 INFO mapreduce.Job: map 72% reduce 0%
17/05/20 18:40:11 INFO mapreduce.Job: map 79% reduce 0%
17/05/20 18:40:15 INFO mapreduce.Job: map 89% reduce 0%
17/05/20 18:40:15 INFO mapreduce.Job: map 92% reduce 0%
17/05/20 18:40:16 INFO mapreduce.Job: map 100% reduce 0%
17/05/20 18:40:18 INFO mapreduce.Job: map 100% reduce 25%
17/05/20 18:40:20 INFO mapreduce.Job: map 100% reduce 31%
17/05/20 18:40:21 INFO mapreduce.Job: map 100% reduce 81%
17/05/20 18:40:22 INFO mapreduce.Job: map 100% reduce 94%
17/05/20 18:40:23 INFO mapreduce.Job: map 100% reduce 100%
17/05/20 18:40:24 INFO mapreduce.Job: Job job_1494408081774_0291 completed successfully
17/05/20 18:40:24 INFO mapreduce.Job: Counters: 50
File System Counters
  FILE: Number of bytes read=45535635
  FILE: Number of bytes written=70221779
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=102749934
  HDFS: Number of bytes written=215
  HDFS: Number of read operations=54
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=32

Job Counters
  Launched map tasks=2
  Launched reduce tasks=16
  Data-local map tasks=1
  Rack-local map tasks=1
  Total time spent by all maps in occupied slots (ms)=364637
  Total time spent by all reduces in occupied slots (ms)=3090920
  Total time spent by all map tasks (ms)=52091
  Total time spent by all reduce tasks (ms)=63080
  Total vcore-seconds taken by all map tasks=52091
  Total vcore-seconds taken by all reduce tasks=63080
  Total megabyte-seconds taken by all map tasks=364637000
  Total megabyte-seconds taken by all reduce tasks=3154000000
```

Figura 39: Compilamos y ejecutamos (I).

```
Job Counters
  Launched map tasks=2
  Launched reduce tasks=16
  Data-local map tasks=1
  Rack-local map tasks=1
  Total time spent by all maps in occupied slots (ms)=364637
  Total time spent by all reduces in occupied slots (ms)=3090920
  Total time spent by all map tasks (ms)=52091
  Total time spent by all reduce tasks (ms)=63080
  Total vcore-seconds taken by all map tasks=52091
  Total vcore-seconds taken by all reduce tasks=63080
  Total megabyte-seconds taken by all map tasks=364637000
  Total megabyte-seconds taken by all reduce tasks=3154000000
Map-Reduce Framework
  Map input records=2897917
  Map output records=28979170
  Map output bytes=292689617
  Map output materialized bytes=22679051
  Input split bytes=234
  Combine input records=0
  Combine output records=0
  Reduce input groups=10
  Reduce shuffle bytes=22679051
  Reduce input records=28979170
  Reduce output records=10
  Spilled Records=86937510
  Shuffled Maps=32
  Failed Shuffles=0
  Merged Map outputs=32
  GC time elapsed (ms)=432
  CPU time spent (ms)=126640
  Physical memory (bytes) snapshot=10865704960
  Virtual memory (bytes) snapshot=984135553024
  Total committed heap usage (bytes)=21401436160
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=102749700
File Output Format Counters
  Bytes Written=215 _
```

Figura 40: Compilamos y ejecutamos (II).

Sólo nos queda comprobar el resultado.

```
[mcc48893432@hadoop-master stat]$ hdfs dfs -cat stat/output9/*
1 0.2549619599157577
10 -1.6989002790625127
2 0.0521277659095442
3 -2.188240380935686
4 -1.408876789776933
5 -1.7528724942777865
6 -1.282261707288373
7 -2.293434905140485
8 -1.5875789403216172
9 -1.7390052924221087
```

Figura 41: Comprobando resultado.

## 9. Comprobar si el conjunto de datos ECBDL es balanceado o no balanceado, es decir, que el ratio entre clases sea menor o mayor que 1.5 respectivamente.

Para calcular si el conjunto de datos es balanceado o no balanceado a partir del ratio entre las clases es necesario definirse dos nuevas funciones **Mapper** y **Reducer** como se muestra en las imágenes.

```
GNU nano 2.0.9 jesusgarciamanday — mcc48893432@hadoop-master:~/stat — ssh mcc48893432@hadoop.ugr.es
File: Balancer.java

package oldapi;
import java.io.IOException;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
public class Balancer {
    public static void main(String[] args) throws IOException {
        if (args.length != 2) {
            System.err.println("Usage: Balancer <input path> <output path>");
            System.exit(-1);
        }
        JobConf conf = new JobConf(Balancer.class);
        conf.setJobName("Balancer");
        FileInputFormat.addInputPath(conf, new Path(args[0]));
        FileOutputFormat.setOutputPath(conf, new Path(args[1]));
        conf.setMapperClass(BalancerMapper.class);
        conf.setReducerClass(BalancerReducer.class);
        conf.setOutputKeyClass(Text.class);
        conf.setOutputValueClass(DoubleWritable.class);
        JobClient.runJob(conf);
    }
}
```

Figura 42: Clase Balancer.

```

package oldapi;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;
public class BalancerMapper extends MapReduceBase implements Mapper<LongWritable, Text, Text, DoubleWritable> {

    public void map(LongWritable key, Text value, OutputCollector<Text, DoubleWritable> output, Reporter reporter) throws IOException{
        String line = value.toString();
        String[] parts = line.split(",");
        int tam = parts.length - 1;
        output.collect(new Text("11"), new DoubleWritable(Double.parseDouble(parts[tam])));
    }
}

```

Figura 43: Clase BalancerMapper.

```

package oldapi;
import java.io.IOException;
import java.util.Iterator;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;
public class BalancerReducer extends MapReduceBase implements Reducer<Text, DoubleWritable, Text, DoubleWritable> {

    public void reduce(Text key, Iterator<DoubleWritable> values, OutputCollector<Text, DoubleWritable> output, Reporter reporter) throws IOException {
        Double c0 = 0.0, c1 = 0.0;
        Double bal = 0.0;

        while (values.hasNext()) {
            if(values.next().get() == 0.0)
                c0++;
            else
                c1++;
        }

        bal = (c0 > c1)? c0/c1 : c1/c0;
        output.collect(key, new DoubleWritable(bal));
    }
}

```

Figura 44: ClaseBalancerReducer.

De nuevo como en las tareas anteriores compilamos y ejecutamos.

```

[jesusgarciamanday@mcc48893432 hadoop-master:~/stat]$ javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop-mapreduce/* -d java_classes Balancer*
[jesusgarciamanday@mcc48893432 hadoop-master:~/stat]$ /usr/java/jdk1.8.0_25/bin/jar -cvf stat.jar -C java_classes /
added manifest
adding: oldapi/(in = 0) (out= 0)(stored 0%)
adding: oldapi/BalancerMapper.class(in = 1693) (out= 648)(deflated 61%)
adding: oldapi/Balancer.class(in = 1427) (out= 774)(deflated 45%)
adding: oldapi/BalancerReducer.class(in = 1793) (out= 743)(deflated 58%)
adding: BalancerMapper.java(in = 896) (out= 345)(deflated 61%)
adding: BalancerReducer.java(in = 870) (out= 363)(deflated 58%)
adding: java_classes/(in = 0) (out= 0)(stored 0%)
adding: java_classes.oldapi/(in = 0) (out= 0)(stored 0%)
adding: java_classes.oldapi.oldapi/BalancerMapper.class(in = 1693) (out= 648)(deflated 61%)
adding: java_classes.oldapi.oldapi.Balancer.class(in = 1427) (out= 774)(deflated 45%)
adding: java_classes.oldapi.oldapi.BalancerReducer.class(in = 1793) (out= 743)(deflated 58%)
adding: Balancer.java(in = 988) (out= 394)(deflated 60%)

```

Figura 45: Compilamos y ejecutamos (I).

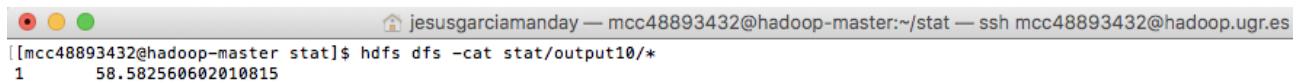
```
[mcc48893432@hadoop-master stat]$ hadoop jar stat.jar oldapi.Balancer /tmp/BDCC/datasets/ECBDL14/ECBDL14_10tst.data ./stat/output10/
17/05/20 19:37:09 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/192.168.10.1:8082
17/05/20 19:37:09 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/192.168.10.1:8082
17/05/20 19:37:10 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
17/05/20 19:37:10 INFO mapred.FileInputFormat: Total input paths to process : 1
17/05/20 19:37:10 INFO mapreduce.JobSubmitter: number of splits:2
17/05/20 19:37:10 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1494408081774_0307
17/05/20 19:37:11 INFO mapreduce.Job: The url to track the job: http://hadoop.ugr.es:8088/proxy/application_1494408081774_0307/
17/05/20 19:37:11 INFO mapreduce.Job: Running job: job_1494408081774_0307
17/05/20 19:37:15 INFO mapreduce.Job: Job job_1494408081774_0307 running in uber mode : false
17/05/20 19:37:15 INFO mapreduce.Job: map 0% reduce 0%
17/05/20 19:37:25 INFO mapreduce.Job: map 100% reduce 0%
17/05/20 19:37:29 INFO mapreduce.Job: map 100% reduce 6%
17/05/20 19:37:30 INFO mapreduce.Job: map 100% reduce 81%
17/05/20 19:37:31 INFO mapreduce.Job: map 100% reduce 88%
17/05/20 19:37:33 INFO mapreduce.Job: map 100% reduce 100%
17/05/20 19:37:34 INFO mapreduce.Job: Job job_1494408081774_0307 completed successfully
17/05/20 19:37:34 INFO mapreduce.Job: Counters: 50
  File System Counters
    FILE: Number of bytes read=143469
    FILE: Number of bytes written=2471440
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=102749934
    HDFS: Number of bytes written=21
    HDFS: Number of read operations=54
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=32
  Job Counters
    Launched map tasks=2
    Launched reduce tasks=16
    Data-local map tasks=1
    Rack-local map tasks=1
    Total time spent by all maps in occupied slots (ms)=103152
    Total time spent by all reduces in occupied slots (ms)=1821575
    Total time spent by all map tasks (ms)=14736
    Total time spent by all reduce tasks (ms)=37175
    Total vcore-seconds taken by all map tasks=14736
    Total vcore-seconds taken by all reduce tasks=37175
    Total megabyte-seconds taken by all map tasks=103152000
    Total megabyte-seconds taken by all reduce tasks=1858750000
```

Figura 46: Compilamos y ejecutamos (II).

```
Map-Reduce Framework
  Map input records=2897917
  Map output records=2897917
  Map output bytes=28979170
  Map output materialized bytes=143591
  Input split bytes=234
  Combine input records=0
  Combine output records=0
  Reduce input groups=1
  Reduce shuffle bytes=143591
  Reduce input records=2897917
  Reduce output records=1
  Spilled Records=5795834
  Shuffled Maps =32
  Failed Shuffles=0
  Merged Map outputs=32
  GC time elapsed (ms)=330
  CPU time spent (ms)=34160
  Physical memory (bytes) snapshot=7885582336
  Virtual memory (bytes) snapshot=904114311168
  Total committed heap usage (bytes)=18992332800
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=102749700
File Output Format Counters
  Bytes Written=21 _
```

Figura 47: Compilamos y ejecutamos (III).

Y comprobamos el resultado donde se puede ver que el ratio es mucho mayor que 1.5 por lo que las clases no están balanceadas.

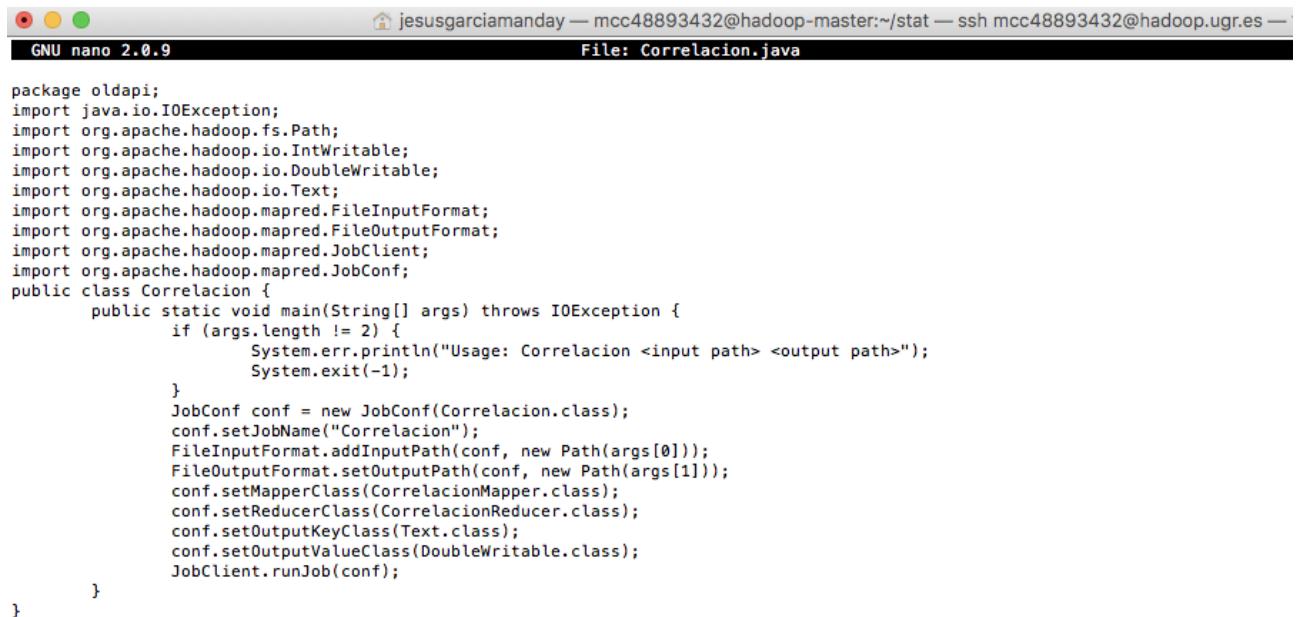


```
[mcc48893432@hadoop-master stat]$ hdfs dfs -cat stat/output10/*
1      58.582560602010815
```

Figura 48: Comprobando resultado.

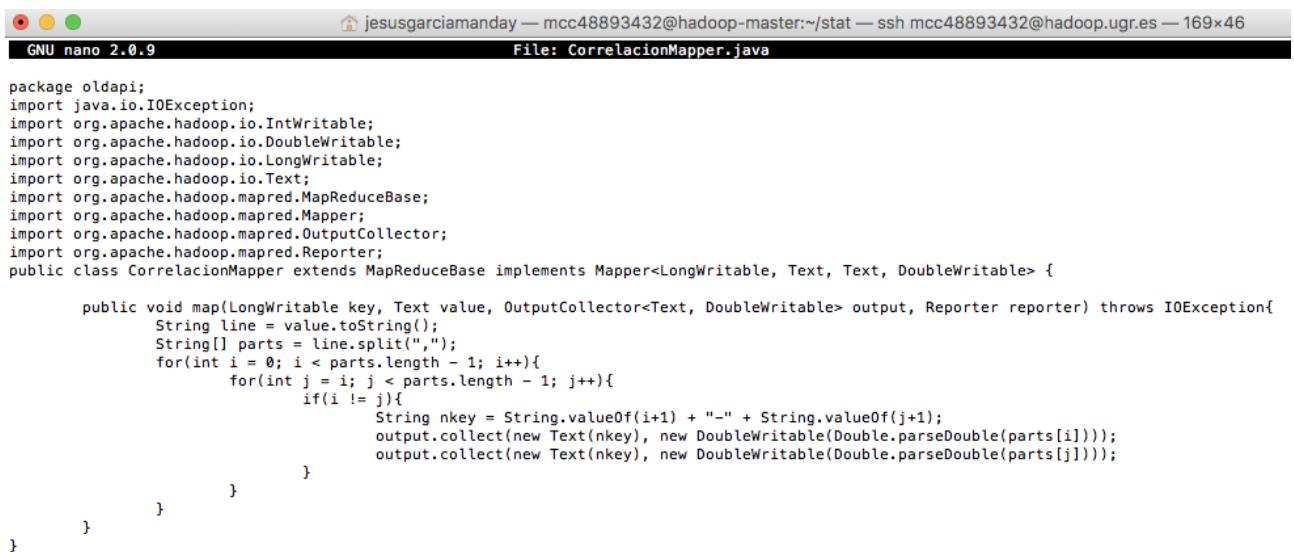
## 10. Cálculo del coeficiente de correlación entre todas las parejas de variables.

Creamos los ficheros correspondientes donde se definen las funciones **Mapper** y **Reducer** para calcular la correlación entre cada par de variables del conjunto de datos.



```
File: Correlacion.java
package oldapi;
import java.io.IOException;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
public class Correlacion {
    public static void main(String[] args) throws IOException {
        if (args.length != 2) {
            System.err.println("Usage: Correlacion <input path> <output path>");
            System.exit(-1);
        }
        JobConf conf = new JobConf(Correlacion.class);
        conf.setJobName("Correlacion");
        FileInputFormat.addInputPath(conf, new Path(args[0]));
        FileOutputFormat.setOutputPath(conf, new Path(args[1]));
        conf.setMapperClass(CorrelacionMapper.class);
        conf.setReducerClass(CorrelacionReducer.class);
        conf.setOutputKeyClass(Text.class);
        conf.setOutputValueClass(DoubleWritable.class);
        JobClient.runJob(conf);
    }
}
```

Figura 49: Clase Correlacion.



```

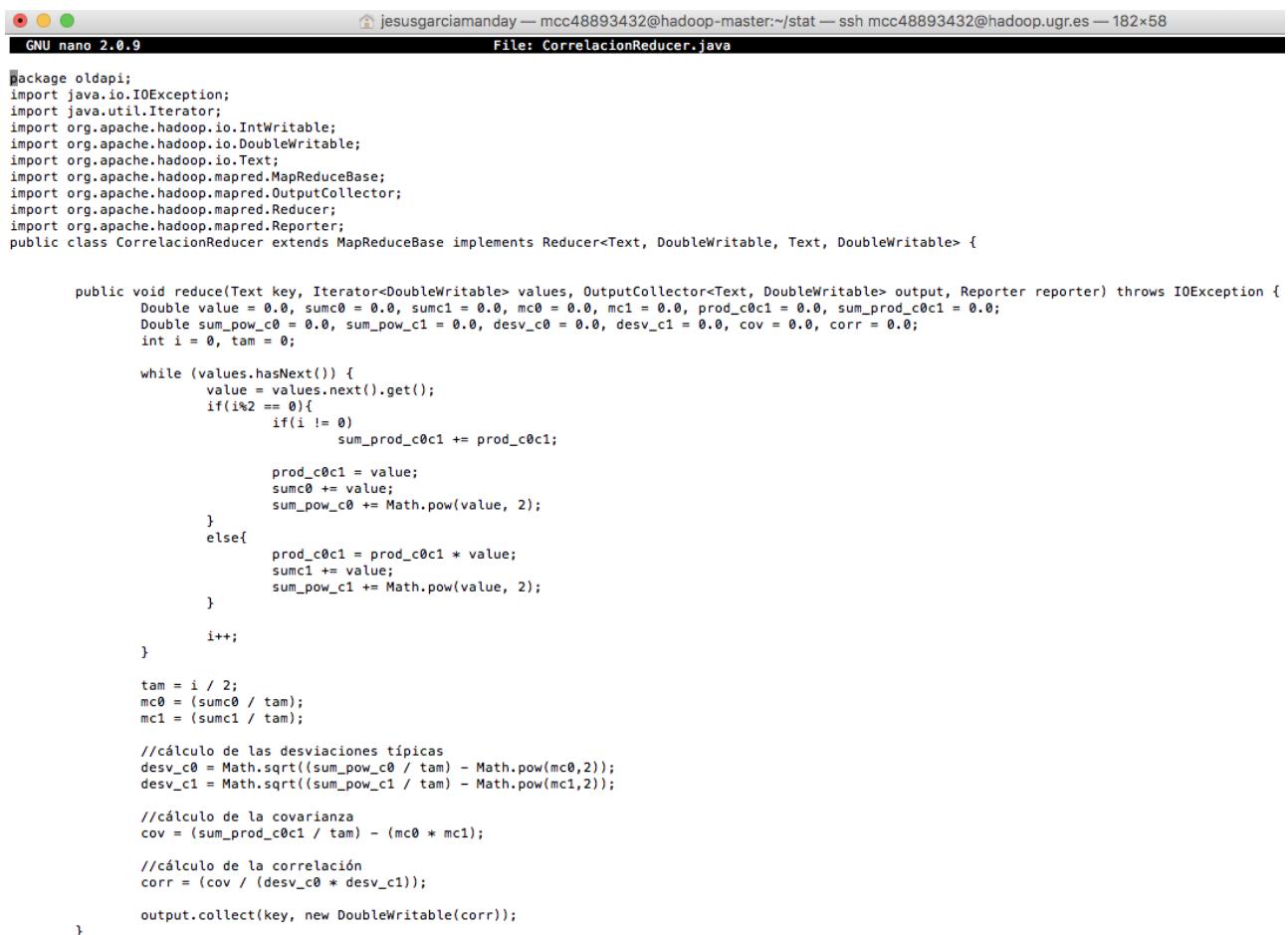
GNU nano 2.0.9          File: CorrelacionMapper.java

package oldapi;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;
public class CorrelacionMapper extends MapReduceBase implements Mapper<LongWritable, Text, Text, DoubleWritable> {

    public void map(LongWritable key, Text value, OutputCollector<Text, DoubleWritable> output, Reporter reporter) throws IOException{
        String line = value.toString();
        String[] parts = line.split(",");
        for(int i = 0; i < parts.length - 1; i++){
            for(int j = i; j < parts.length - 1; j++){
                if(i != j){
                    String nkey = String.valueOf(i+1) + "-" + String.valueOf(j+1);
                    output.collect(new Text(nkey), new DoubleWritable(Double.parseDouble(parts[i])));
                    output.collect(new Text(nkey), new DoubleWritable(Double.parseDouble(parts[j])));
                }
            }
        }
    }
}

```

Figura 50: Clase CorrelacionMapper.



```

GNU nano 2.0.9          File: CorrelacionReducer.java

package oldapi;
import java.io.IOException;
import java.util.Iterator;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.DoubleWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;
public class CorrelacionReducer extends MapReduceBase implements Reducer<Text, DoubleWritable, Text, DoubleWritable> {

    public void reduce(Text key, Iterator<DoubleWritable> values, OutputCollector<Text, DoubleWritable> output, Reporter reporter) throws IOException {
        Double value = 0.0, sumc0 = 0.0, sumc1 = 0.0, mc0 = 0.0, mc1 = 0.0, prod_c0c1 = 0.0, sum_prod_c0c1 = 0.0;
        Double sum_pow_c0 = 0.0, sum_pow_c1 = 0.0, desv_c0 = 0.0, desv_c1 = 0.0, cov = 0.0, corr = 0.0;
        int i = 0, tam = 0;

        while (values.hasNext()) {
            value = values.next().get();
            if(i%2 == 0){
                if(i != 0)
                    sum_prod_c0c1 += prod_c0c1;

                prod_c0c1 = value;
                sumc0 += value;
                sum_pow_c0 += Math.pow(value, 2);
            } else{
                prod_c0c1 = prod_c0c1 * value;
                sumc1 += value;
                sum_pow_c1 += Math.pow(value, 2);
            }
            i++;
        }

        tam = i / 2;
        mc0 = (sumc0 / tam);
        mc1 = (sumc1 / tam);

        //cálculo de las desviaciones típicas
        desv_c0 = Math.sqrt((sum_pow_c0 / tam) - Math.pow(mc0,2));
        desv_c1 = Math.sqrt((sum_pow_c1 / tam) - Math.pow(mc1,2));

        //cálculo de la covarianza
        cov = (sum_prod_c0c1 / tam) - (mc0 * mc1);

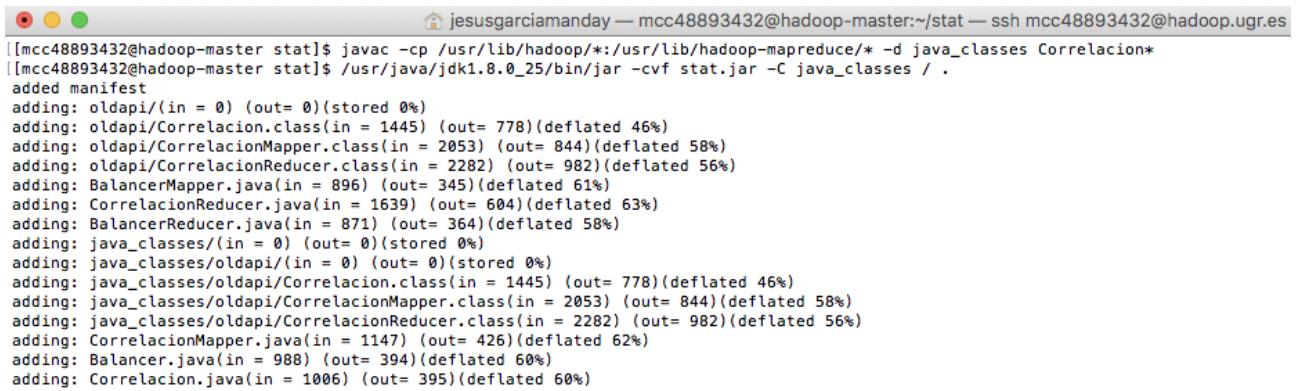
        //cálculo de la correlación
        corr = (cov / (desv_c0 * desv_c1));
        output.collect(key, new DoubleWritable(corr));
    }
}

```

Figura 51: Clase CorrelacionReducer.

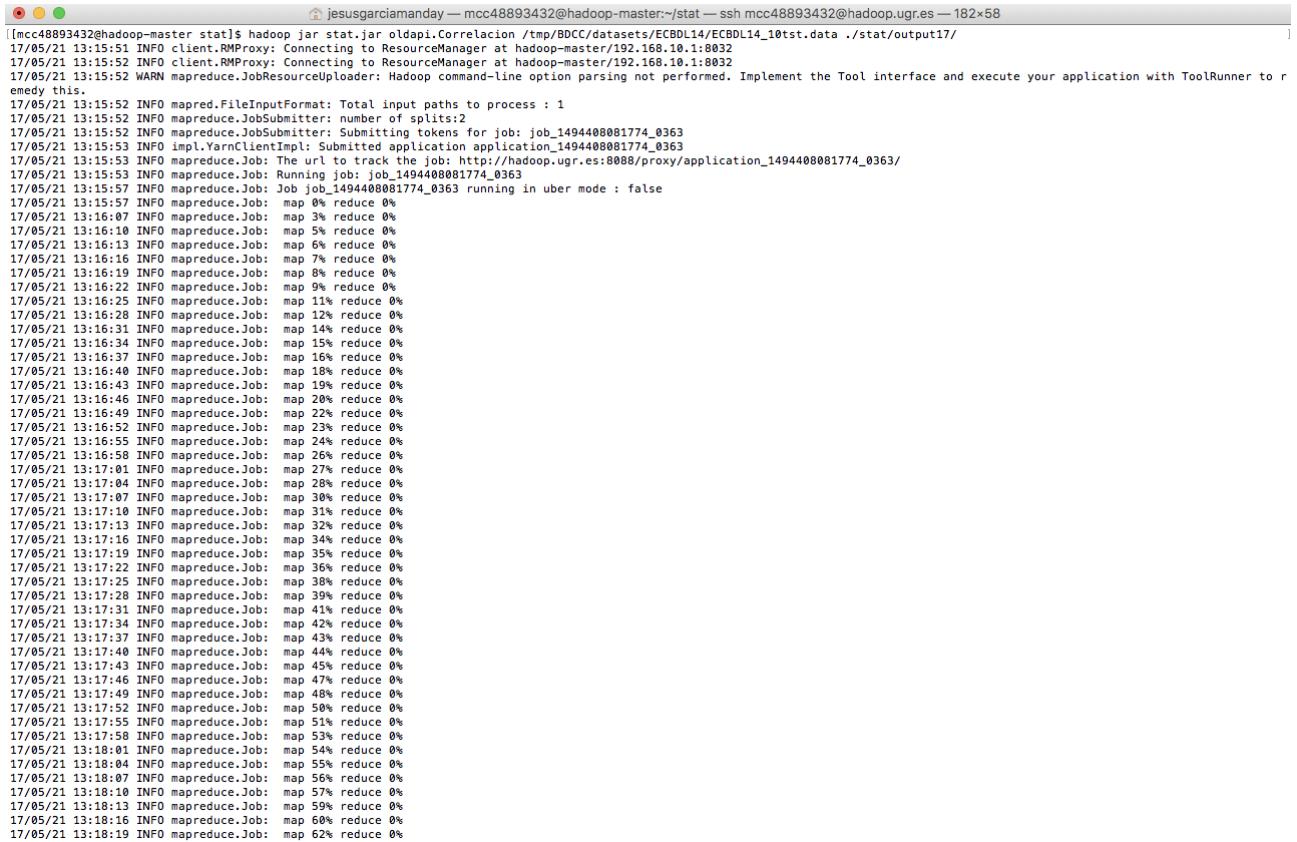
Una vez que tenemos los ficheros con las funciones implementadas, pasamos a compilar las clases para em-

paquetarlas en el archivo **jar** correspondiente y ejecutarlo en **hadoop**.



```
[mcc48893432@hadoop-master stat]$ javac -cp /usr/lib/hadoop/*:/usr/lib/hadoop-mapreduce/* -d java_classes Correlacion
[mcc48893432@hadoop-master stat]$ /usr/java/jdk1.8.0_25/bin/jar -cvf stat.jar -C java_classes /
added manifest
adding: oldapi/(in = 0) (out= 0)(stored 0%)
adding: oldapi/Correlacion.class(in = 1445) (out= 778)(deflated 46%)
adding: oldapi/CorrelacionMapper.class(in = 2053) (out= 844)(deflated 58%)
adding: oldapi/CorrelacionReducer.class(in = 2282) (out= 982)(deflated 56%)
adding: BalancerMapper.java(in = 896) (out= 345)(deflated 61%)
adding: CorrelacionReducer.java(in = 1639) (out= 604)(deflated 63%)
adding: BalancerReducer.java(in = 871) (out= 364)(deflated 58%)
adding: java_classes/(in = 0) (out= 0)(stored 0%)
adding: java_classes.oldapi/(in = 0) (out= 0)(stored 0%)
adding: java_classes.oldapi/Correlacion.class(in = 1445) (out= 778)(deflated 46%)
adding: java_classes.oldapi/CorrelacionMapper.class(in = 2053) (out= 844)(deflated 58%)
adding: java_classes.oldapi/CorrelacionReducer.class(in = 2282) (out= 982)(deflated 56%)
adding: CorrelacionMapper.java(in = 1147) (out= 426)(deflated 62%)
adding: Balancer.java(in = 988) (out= 394)(deflated 60%)
adding: Correlacion.java(in = 1006) (out= 395)(deflated 60%)
```

Figura 52: Compilamos y ejecutamos (I).



```
[mcc48893432@hadoop-master stat]$ hadoop jar stat.jar oldapi.Correlacion /tmp/BDCC/datasets/ECBDL14/ECBDL14_10st.data ./stat/output17
17/05/21 13:15:51 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/192.168.10.1:8032
17/05/21 13:15:52 INFO client.RMProxy: Connecting to ResourceManager at hadoop-master/192.168.10.1:8032
17/05/21 13:15:52 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
17/05/21 13:15:52 INFO mapred.FileInputFormat: Total input paths to process : 1
17/05/21 13:15:52 INFO mapreduce.JobSubmission: number of splits:2
17/05/21 13:15:52 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1494408001774_0363
17/05/21 13:15:53 INFO impl.YarnClientImpl: Submitted application application_1494408001774_0363
17/05/21 13:15:53 INFO mapreduce.Job: Running job: job_1494408001774_0363
17/05/21 13:15:57 INFO mapreduce.Job: Job job_1494408001774_0363 running in uber mode : false
17/05/21 13:16:07 INFO mapreduce.Job: map 0% reduce 0%
17/05/21 13:16:07 INFO mapreduce.Job: map 3% reduce 0%
17/05/21 13:16:10 INFO mapreduce.Job: map 5% reduce 0%
17/05/21 13:16:13 INFO mapreduce.Job: map 6% reduce 0%
17/05/21 13:16:16 INFO mapreduce.Job: map 7% reduce 0%
17/05/21 13:16:19 INFO mapreduce.Job: map 8% reduce 0%
17/05/21 13:16:22 INFO mapreduce.Job: map 9% reduce 0%
17/05/21 13:16:25 INFO mapreduce.Job: map 11% reduce 0%
17/05/21 13:16:28 INFO mapreduce.Job: map 12% reduce 0%
17/05/21 13:16:31 INFO mapreduce.Job: map 14% reduce 0%
17/05/21 13:16:34 INFO mapreduce.Job: map 15% reduce 0%
17/05/21 13:16:37 INFO mapreduce.Job: map 16% reduce 0%
17/05/21 13:16:40 INFO mapreduce.Job: map 18% reduce 0%
17/05/21 13:16:43 INFO mapreduce.Job: map 19% reduce 0%
17/05/21 13:16:46 INFO mapreduce.Job: map 20% reduce 0%
17/05/21 13:16:49 INFO mapreduce.Job: map 22% reduce 0%
17/05/21 13:16:52 INFO mapreduce.Job: map 23% reduce 0%
17/05/21 13:16:55 INFO mapreduce.Job: map 24% reduce 0%
17/05/21 13:16:58 INFO mapreduce.Job: map 26% reduce 0%
17/05/21 13:17:01 INFO mapreduce.Job: map 27% reduce 0%
17/05/21 13:17:04 INFO mapreduce.Job: map 28% reduce 0%
17/05/21 13:17:07 INFO mapreduce.Job: map 30% reduce 0%
17/05/21 13:17:10 INFO mapreduce.Job: map 31% reduce 0%
17/05/21 13:17:13 INFO mapreduce.Job: map 32% reduce 0%
17/05/21 13:17:16 INFO mapreduce.Job: map 34% reduce 0%
17/05/21 13:17:19 INFO mapreduce.Job: map 35% reduce 0%
17/05/21 13:17:22 INFO mapreduce.Job: map 36% reduce 0%
17/05/21 13:17:25 INFO mapreduce.Job: map 38% reduce 0%
17/05/21 13:17:28 INFO mapreduce.Job: map 39% reduce 0%
17/05/21 13:17:31 INFO mapreduce.Job: map 41% reduce 0%
17/05/21 13:17:34 INFO mapreduce.Job: map 42% reduce 0%
17/05/21 13:17:37 INFO mapreduce.Job: map 43% reduce 0%
17/05/21 13:17:40 INFO mapreduce.Job: map 44% reduce 0%
17/05/21 13:17:43 INFO mapreduce.Job: map 45% reduce 0%
17/05/21 13:17:46 INFO mapreduce.Job: map 47% reduce 0%
17/05/21 13:17:49 INFO mapreduce.Job: map 48% reduce 0%
17/05/21 13:17:52 INFO mapreduce.Job: map 50% reduce 0%
17/05/21 13:17:55 INFO mapreduce.Job: map 51% reduce 0%
17/05/21 13:17:58 INFO mapreduce.Job: map 53% reduce 0%
17/05/21 13:18:01 INFO mapreduce.Job: map 54% reduce 0%
17/05/21 13:18:04 INFO mapreduce.Job: map 55% reduce 0%
17/05/21 13:18:07 INFO mapreduce.Job: map 56% reduce 0%
17/05/21 13:18:10 INFO mapreduce.Job: map 57% reduce 0%
17/05/21 13:18:13 INFO mapreduce.Job: map 59% reduce 0%
17/05/21 13:18:16 INFO mapreduce.Job: map 60% reduce 0%
17/05/21 13:18:19 INFO mapreduce.Job: map 62% reduce 0%
```

Figura 53: Compilamos y ejecutamos (II).

```
17/05/21 13:18:22 INFO mapreduce.Job: map 63% reduce 0%
17/05/21 13:18:25 INFO mapreduce.Job: map 64% reduce 0%
17/05/21 13:18:28 INFO mapreduce.Job: map 65% reduce 0%
17/05/21 13:18:31 INFO mapreduce.Job: map 66% reduce 0%
17/05/21 13:18:34 INFO mapreduce.Job: map 67% reduce 0%
17/05/21 13:18:40 INFO mapreduce.Job: map 68% reduce 0%
17/05/21 13:18:43 INFO mapreduce.Job: map 70% reduce 0%
17/05/21 13:18:52 INFO mapreduce.Job: map 72% reduce 0%
17/05/21 13:18:55 INFO mapreduce.Job: map 74% reduce 0%
17/05/21 13:18:58 INFO mapreduce.Job: map 75% reduce 0%
17/05/21 13:19:01 INFO mapreduce.Job: map 76% reduce 0%
17/05/21 13:19:04 INFO mapreduce.Job: map 77% reduce 0%
17/05/21 13:19:07 INFO mapreduce.Job: map 78% reduce 0%
17/05/21 13:19:11 INFO mapreduce.Job: map 79% reduce 0%
17/05/21 13:19:14 INFO mapreduce.Job: map 80% reduce 0%
17/05/21 13:19:20 INFO mapreduce.Job: map 81% reduce 0%
17/05/21 13:19:26 INFO mapreduce.Job: map 82% reduce 0%
17/05/21 13:19:29 INFO mapreduce.Job: map 84% reduce 0%
17/05/21 13:19:32 INFO mapreduce.Job: map 83% reduce 0%
17/05/21 13:19:35 INFO mapreduce.Job: map 85% reduce 0%
17/05/21 13:19:38 INFO mapreduce.Job: map 84% reduce 0%
17/05/21 13:19:41 INFO mapreduce.Job: map 86% reduce 0%
17/05/21 13:19:44 INFO mapreduce.Job: map 85% reduce 0%
17/05/21 13:19:50 INFO mapreduce.Job: map 87% reduce 0%
17/05/21 13:19:53 INFO mapreduce.Job: map 86% reduce 0%
17/05/21 13:19:56 INFO mapreduce.Job: map 88% reduce 0%
17/05/21 13:20:05 INFO mapreduce.Job: map 89% reduce 0%
17/05/21 13:20:15 INFO mapreduce.Job: map 90% reduce 0%
17/05/21 13:20:27 INFO mapreduce.Job: map 91% reduce 0%
17/05/21 13:20:33 INFO mapreduce.Job: map 93% reduce 0%
17/05/21 13:20:36 INFO mapreduce.Job: map 92% reduce 0%
17/05/21 13:20:45 INFO mapreduce.Job: map 93% reduce 0%
17/05/21 13:20:54 INFO mapreduce.Job: map 94% reduce 0%
17/05/21 13:21:00 INFO mapreduce.Job: map 95% reduce 0%
17/05/21 13:21:09 INFO mapreduce.Job: map 96% reduce 0%
17/05/21 13:21:12 INFO mapreduce.Job: map 97% reduce 0%
17/05/21 13:21:21 INFO mapreduce.Job: map 99% reduce 0%
17/05/21 13:21:29 INFO mapreduce.Job: map 100% reduce 0%
17/05/21 13:21:31 INFO mapreduce.Job: map 100% reduce 29%
17/05/21 13:21:34 INFO mapreduce.Job: map 100% reduce 41%
17/05/21 13:21:36 INFO mapreduce.Job: map 100% reduce 45%
17/05/21 13:21:37 INFO mapreduce.Job: map 100% reduce 53%
17/05/21 13:21:39 INFO mapreduce.Job: map 100% reduce 56%
17/05/21 13:21:40 INFO mapreduce.Job: map 100% reduce 63%
17/05/21 13:21:41 INFO mapreduce.Job: map 100% reduce 64%
17/05/21 13:21:42 INFO mapreduce.Job: map 100% reduce 65%
17/05/21 13:21:43 INFO mapreduce.Job: map 100% reduce 71%
17/05/21 13:21:45 INFO mapreduce.Job: map 100% reduce 75%
17/05/21 13:21:46 INFO mapreduce.Job: map 100% reduce 81%
17/05/21 13:21:48 INFO mapreduce.Job: map 100% reduce 83%
17/05/21 13:21:49 INFO mapreduce.Job: map 100% reduce 86%
17/05/21 13:21:51 INFO mapreduce.Job: map 100% reduce 87%
17/05/21 13:21:52 INFO mapreduce.Job: map 100% reduce 90%
17/05/21 13:21:53 INFO mapreduce.Job: map 100% reduce 91%
17/05/21 13:21:54 INFO mapreduce.Job: map 100% reduce 92%
17/05/21 13:21:55 INFO mapreduce.Job: map 100% reduce 95%
17/05/21 13:21:57 INFO mapreduce.Job: map 100% reduce 96%
17/05/21 13:21:58 INFO mapreduce.Job: map 100% reduce 98%
```

Figura 54: Compilamos y ejecutamos (III).

```

17/05/21 13:21:59 INFO mapreduce.Job: map 100% reduce 99%
17/05/21 13:22:00 INFO mapreduce.Job: map 100% reduce 100%
17/05/21 13:22:01 INFO mapreduce.Job: Job job_1494408081774_0363 completed successfully
17/05/21 13:22:01 INFO mapreduce.Job: Counters: 50
  File System Counters
    FILE: Number of bytes read=674242832
    FILE: Number of bytes written=908984921
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=102749934
    HDFS: Number of bytes written=1138
    HDFS: Number of read operations=54
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=32
  Job Counters
    Killed reduce tasks=1
    Launched map tasks=2
    Launched reduce tasks=17
    Rack-local map tasks=2
    Total time spent by all maps in occupied slots (ms)=4566345
    Total time spent by all reduces in occupied slots (ms)=18715011
    Total time spent by all map tasks (ms)=652335
    Total time spent by all reduce tasks (ms)=381939
    Total vcore-seconds taken by all map tasks=652335
    Total vcore-seconds taken by all reduce tasks=381939
    Total megabyte-seconds taken by all map tasks=4566345000
    Total megabyte-seconds taken by all reduce tasks=19096950000
  Map-Reduce Framework
    Map input records=2897917
    Map output records=260812530
    Map output bytes=3181912866
    Map output materialized bytes=235271277
    Input split bytes=234
    Combine input records=0
    Combine output records=0
    Reduce input groups=45
    Reduce shuffle bytes=235271277
    Reduce input records=260812530
    Reduce output records=45
    Spilled Records=1007553869
    Shuffled Maps =32
    Failed Shuffles=0
    Merged Map outputs=32
    GC time elapsed (ms)=1708
    CPU time spent (ms)=1160080
    Physical memory (bytes) snapshot=15827271680
    Virtual memory (bytes) snapshot=984168431616
    Total committed heap usage (bytes)=21418737664
  Shuffle Errors
    BAD_ID=0
    CONNECTION=0
    IO_ERROR=0
    WRONG_LENGTH=0
    WRONG_MAP=0
    WRONG_REDUCE=0

```

Figura 55: Compilamos y ejecutamos (IV).

```

File Input Format Counters
  Bytes Read=102749700
File Output Format Counters
  Bytes Written=1138

```

Figura 56: Compilamos y ejecutamos (V).

Con la ejecución terminada correctamente pasamos a ver los resultados obtenidos.

```
[mcc48893432@hadoop-master stat]$ hdfs dfs -cat stat/output17/*
5-9      0.014324448501259272
6-8      0.030140080219938088
6-9      0.017136767192123536
7-8      0.05571437618018473
7-9      0.04402815102922523
8-9      -0.05428750966354933
9-10     0.041153864923909824
1-2      -0.10169351129527973
8-10     0.050285929580401664
1-3      -0.020377723341046976
7-10     0.049746017198488625
1-4      -0.011624387631553858
2-3      -0.02653571477366161
6-10     0.024482948272585636
1-5      -0.022260906360211014
2-4      -0.0036758231137280512
5-10     0.018170618158992728
1-6      -0.021503204158265368
2-5      -0.020828563832617546
3-4      0.018759637459232256
4-10     0.023848271908014407
1-7      -0.029731428543233376
2-6      -0.014035338986913766
3-10     0.03007864600478592
3-5      0.023809450337523033
1-8      -0.013085492698102429
2-10     -0.011506904691958485
2-7      -0.02202380585492076
3-6      0.025898456271905377
4-5      0.020231144275277266
1-10     -0.017501980695027685
1-9      -0.026464584364562012
2-8      -0.01125058199734558
3-7      0.049164427273738514
4-6      0.025016444928476433
2-9      -0.0237847451706592
3-8      0.03939354998201235
4-7      0.030373268279491532
5-6      0.03037779448659513
3-9      0.029625134765712394
4-8      0.02794316508917126
5-7      0.021689158289262128
4-9      0.022736286781502496
5-8      0.0218292185175135
6-7      0.02666870242550753
```

Figura 57: Comprobando resultado.