

viu  
.es

2  
0  
2  
2  
-  
2  
0  
2  
3



# ACTIVIDAD 1: Programación con Hadoop MapReduce.

Máster en Big Data y Data Science

01MBID – Fundamentos de la Tecnología de Big Data

Autores: Adrián Hernández Padrón y Andrea San Blas Hernández.

Fecha: 24/06/2022

### Actividades a elaborar:

Dado un dataset que contenga entradas con la forma “persona gasto”, crea un programa llamado ModaGastoPorPersona que indique para cada persona cuál es el gasto más frecuente (la moda). Se valorará positivamente la optimización del programa, por ejemplo a través de la funcionalidad Combiner. Ejemplo:

Entrada	Salida
Alice 10	Alice 10
Alice 3	Bob 5
Alice 10	
Bob 5	
Bob 5	
Bob 1	

Notar que Alice y Bob gastaron 2 veces 10 y 5, y sólo una vez 3 y 1, respectivamente, por lo tanto la salida es Alice 10 y Bob 5. Se proporciona un fichero de entrada (casoDePruebaEJ1.txt).

### Solución:

Antes de comenzar con el desarrollo de la actividad se debe realizar la instalación de Hadoop. En el caso del presente trabajo los dos autores han trabajado en entornos distintos. Estos son: Google Colab y Hadoop en local. Para el primer caso, basta con ejecutar el siguiente código en una celda de Google Colab:

```
!apt-get install -y openjdk-11-jdk-headless -qq > /dev/null
!wget https://downloads.apache.org/hadoop/common/hadoop-3.3.3/hadoop-3.3.3.tar.gz
!tar -xzf hadoop-3.3.3.tar.gz
!mv hadoop-3.3.3/ /usr/local/
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-11-openjdk-amd64"
os.environ["HADOOP_HOME"] = "/usr/local/hadoop-3.3.3"
os.environ["PATH"] += os.pathsep + "/usr/local/hadoop-3.3.3/bin"
```

Ejecutando dicho código, se obtiene:

```
!apt-get install -y openjdk-11-jdk-headless -qq > /dev/null
!wget https://downloads.apache.org/hadoop/common/hadoop-3.3.3/hadoop-3.3.3.tar.gz
!tar -xzf hadoop-3.3.3.tar.gz
!mv hadoop-3.3.3/ /usr/local/
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-11-openjdk-amd64"
os.environ["HADOOP_HOME"] = "/usr/local/hadoop-3.3.3"
os.environ["PATH"] += os.pathsep + "/usr/local/hadoop-3.3.3/bin"

--2022-06-22 22:32:17-- https://downloads.apache.org/hadoop/common/hadoop-3.3.3/hadoop-3.3.3.tar.gz
Resolving downloads.apache.org (downloads.apache.org)... 88.99.95.219, 135.181.214.104, 2a01:4f8:10a:201a::2, ...
Connecting to downloads.apache.org (downloads.apache.org)|88.99.95.219|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 645040598 (615M) [application/x-gzip]
Saving to: 'hadoop-3.3.3.tar.gz'

hadoop-3.3.3.tar.gz 100%[=====] 615.16M 19.7MB/s in 32s

2022-06-22 22:32:49 (19.3 MB/s) - 'hadoop-3.3.3.tar.gz' saved [645040598/645040598]
```

Se observa que se ha descargado correctamente la versión de Hadoop 3.3.3.

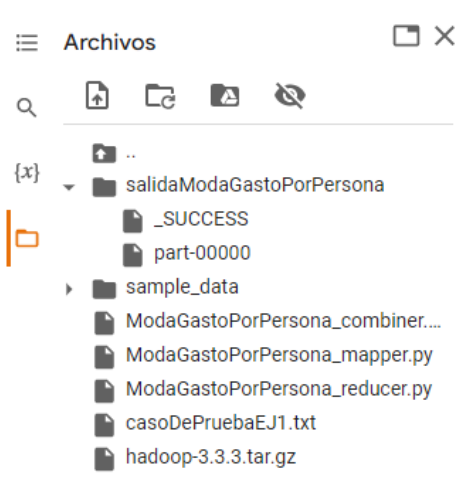
La instalación de Hadoop localmente en el mac se hizo siguiendo los pasos del siguiente video:  
<https://www.youtube.com/watch?v=H999fluymqc>

Tras la instalación de Hadoop se tiene el entorno preparado para la implementación del programa ModaGastoPorPersona que se solicita. El programa consta de tres archivos en lenguaje python. Veamos a continuación qué función cumplen cada uno de esos archivos.

- **Archivo 1:** ModaGastoPorPersona\_mapper.py. Cada línea del fichero de entrada 'casoDePruebaEJ1.txt' es procesada por un 'Map'. De ese string que recibe el 'Map' se construye un par <clave, valor>. Luego, este par es emitido por el 'Map' y recibido por el 'Combiner'. En este problema en concreto la clave es 'persona-gasto' y el valor '1'. Porque como se comenta en el desarrollo del código este valor '1' inicia un contador que almacenará en el 'Combiner' el número de veces que se repite la clave 'persona-gasto'.
- **Archivo 2:** ModaGastoPorPersona\_combiner.py. El 'Combiner' se ejecuta en el mismo nodo en el que se ejecuta el 'Mapper'. Y por cada 'Mapper' aplica el código de este segundo archivo. Es decir, se reciben los pares <persona-gasto, 1> emitidos en el 'Mapper' y se devuelven las tuplas <persona-gasto, suma\_conteos>, donde 'suma\_conteos' almacena la frecuencia de cada par <persona, gasto>.
- **Archivo 3:** ModaGastoPorPersona\_reducer.py. Cada 'Reducer' recibe clave y su lista de valores. Y genera un único resultado que es la tupla <persona, gasto, freq> indicando para cada persona cuál es el gasto más frecuente (la moda) y el número de veces que aparece (freq).

Se tiene en cuenta que la explicación de cada uno de los códigos se encuentra explícita dentro de cada archivo.

Luego, se cargan los archivos en Google Colab:



Realizando una llamada al sistema damos los permisos de acceso a los ficheros:

```
[2] !chmod u+x ./ModaGastoPorPersona_mapper.py
!chmod u+x ./ModaGastoPorPersona_combiner.py
!chmod u+x ./ModaGastoPorPersona_reducer.py
```

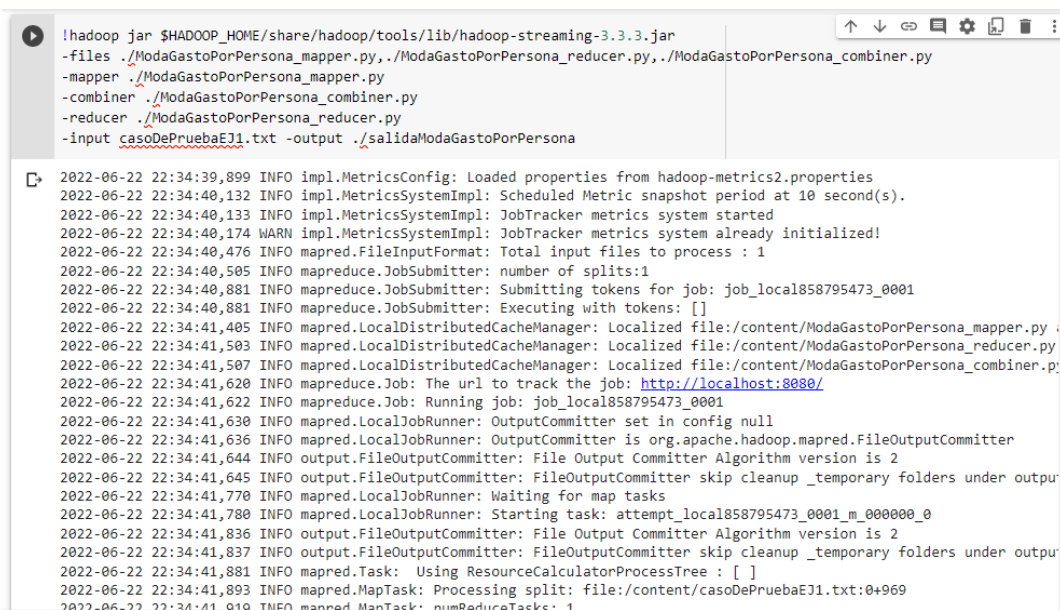
Ejecutamos en Hadoop el programa MapReduce:

```
!hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-3.3.3.jar -files
./ModaGastoPorPersona_mapper.py,./ModaGastoPorPersona_reducer.py,./ModaGastoPorPersona_combiner.
py
-mapper ./ModaGastoPorPersona_mapper.py
-combiner ./ModaGastoPorPersona_combiner.py
-reducer ./ModaGastoPorPersona_reducer.py
-input casoDePruebaEJ1.txt -output ./salidaModaGastoPorPersona
```

Una explicación breve de este comando es la siguiente:

- **!hadoop jar:** Hadoop es una estructura de software de código abierto escrita en Java, por tanto debe ser ejecutado con `!hadoop jar`.
- **\$HADOOP\_HOME/share/hadoop/tools/lib/hadoop-streaming-3.3.3.jar:** Con esta línea de comando ejecutamos con `'hadoop-streaming'` y con la versión de Hadoop 3.3.3. El comando `'hadoop-streaming'` no quiere decir que los datos se procesan en tiempo real. Hadoop tiene procesamiento batch por defecto. Lo que indica este comando es la manera en la que se pasan los `'-input'` y los `'-output'`. Permite desarrollar ejecutables de MapReduce en lenguajes que no sean Java. En este caso, en lenguaje python.
- **-files**  
`./ModaGastoPorPersona_mapper.py,./ModaGastoPorPersona_reducer.py,`  
`./ModaGastoPorPersona_combiner.py:` Indicamos cuales son los ficheros que vamos a utilizar. Estos ficheros se han subido previamente al entorno de trabajo.
- **-mapper** `./ModaGastoPorPersona_mapper.py:` Indicamos cual de esos archivos es el `'Mapper'`.
- **-combiner** `./ModaGastoPorPersona_combiner.py:` Indicamos cual es el `'Combiner'`.
- **-reducer** `./ModaGastoPorPersona_reducer.py:` Indicamos cual es el `'Reducer'`.
- **-input** `casoDePruebaEJ1.txt:` Indicamos el archivo de entrada.
- **-output** `./salidaModaGastoPorPersona:` Indicamos el archivo de salida.

Ejecutamos:



```
!hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-3.3.3.jar
-files ./ModaGastoPorPersona_mapper.py,./ModaGastoPorPersona_reducer.py,./ModaGastoPorPersona_combiner.py
-mapper ./ModaGastoPorPersona_mapper.py
-combiner ./ModaGastoPorPersona_combiner.py
-reducer ./ModaGastoPorPersona_reducer.py
-input casoDePruebaEJ1.txt -output ./salidaModaGastoPorPersona

2022-06-22 22:34:39,899 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2022-06-22 22:34:40,132 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
2022-06-22 22:34:40,133 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2022-06-22 22:34:40,174 WARN impl.MetricsSystemImpl: JobTracker metrics system already initialized!
2022-06-22 22:34:40,476 INFO mapred.FileInputFormat: Total input files to process : 1
2022-06-22 22:34:40,505 INFO mapreduce.JobSubmitter: number of splits:1
2022-06-22 22:34:40,881 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local858795473_0001
2022-06-22 22:34:40,881 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-06-22 22:34:41,405 INFO mapred.LocalDistributedCacheManager: Localized file:/content/ModaGastoPorPersona_mapper.py
2022-06-22 22:34:41,503 INFO mapred.LocalDistributedCacheManager: Localized file:/content/ModaGastoPorPersona_reducer.py
2022-06-22 22:34:41,507 INFO mapred.LocalDistributedCacheManager: Localized file:/content/ModaGastoPorPersona_combiner.py
2022-06-22 22:34:41,620 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2022-06-22 22:34:41,622 INFO mapreduce.Job: Running job: job_local858795473_0001
2022-06-22 22:34:41,630 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2022-06-22 22:34:41,636 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapred.FileOutputCommitter
2022-06-22 22:34:41,837 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2022-06-22 22:34:41,644 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output
2022-06-22 22:34:41,645 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output
2022-06-22 22:34:41,770 INFO mapred.LocalJobRunner: Waiting for map tasks
2022-06-22 22:34:41,780 INFO mapred.LocalJobRunner: Starting task: attempt_local858795473_0001_m_000000_0
2022-06-22 22:34:41,836 INFO output.FileOutputCommitter: File Output Committer Algorithm version is 2
2022-06-22 22:34:41,837 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporary folders under output
2022-06-22 22:34:41,881 INFO mapred.Task: Using ResourceCalculatorProcessTree : [ ]
2022-06-22 22:34:41,893 INFO mapred.MapTask: Processing split: file:/content/casoDePruebaEJ1.txt:0+969
2022-06-22 22:34:41,910 INFO mapred.MapTask: numReduceTasks: 1
```

Al finalizar la ejecución se obtiene como salida la creación de la carpeta: 'salidaModaGastoPorPersona'. Pedimos que se nos muestre la salida mediante el código:

```
!cat ./salidaModaGastoPorPersona/*
```

Finalmente, se muestra el resultado esperado para el fichero de entrada 'casoDePruebaEJ1.txt' proporcionado. En la primera columna se tienen a las personas que figuran en el fichero de entrada. Luego, en la segunda columna el gasto más frecuente (la moda) asociado a la persona que se encuentra en la misma fila. Y aunque no se solicita, se ha añadido una tercera columna que nos enseña la frecuencia de repetición del gasto moda.

```
[ ] !cat ./salidaModaGastoPorPersona/*
```

Alice	10	2
Bob	91	3

Se tiene que el gasto más frecuente de Alice toma el valor 10 y la frecuencia de este gasto es de dos veces. Y el gasto más frecuente de Bob es 91 y la frecuencia de dicho gasto es de tres veces.