Desarrollo de programa MapReduce en Python

1. Desarrollo de programa Wordcount:
   Problema: contar el número de veces que aparece cada palabra
   Subproblema: por cada palabra contamos el número de veces que aparece
      Clave: palabra
      Valor: el número de veces que aparece la palabra
   Reduce: recibe una palabra junto con las veces que apareció ej <hola, [1,7,1,3]> significa que la palabra "hola" aparece 1 vez, 7 veces, 1 vez y 3 veces. Entonces la palabra "hola" aparece 12 veces
   Map: recibe una línea de texto y le envía a cada Reduce la información que necesita de esa línea. Ej: "hola Big Data" aquí tenemos 3 palabras, cada una le interesa a un subproblema. Entonces tenemos que enviar información a 3 subrproblemas. Al subproblema "hola" le indicamos que aparece una vez. Al subproblema "Big" le indicamos que aparece una vez. Y finalmente al subproblema "Data" le indicamos que aparece 1 vez. Por tanto, Map emite <palabra, 1> o lo que es lo mismo <subproblema, 1>. Por tanto:
   Map(recibe una línea): por cada palabra emite <palabra, 1>
   Reduce(recibe una palabra junto con las veces que apareció): suma las veces que apareció esa palabra
   a. Creamos una carpeta: mkdir /home/moranjesus/Desktop/wordcountPython

```
[moranjesus@localhost ~]$ mkdir /home/moranjesus/Desktop/wordcountPython
[moranjesus@localhost ~]$
```

   b. Nos ubicamos en esa carpeta desde la terminal:
      cd /home/moranjesus/Desktop/wordcountPython

```
[moranjesus@localhost ~]$ cd /home/moranjesus/Desktop/wordcountPython
[moranjesus@localhost wordcountPython]$
```

   c. Creamos un archivo llamado mapperWordCount.py:
      touch mapperWordCount.py

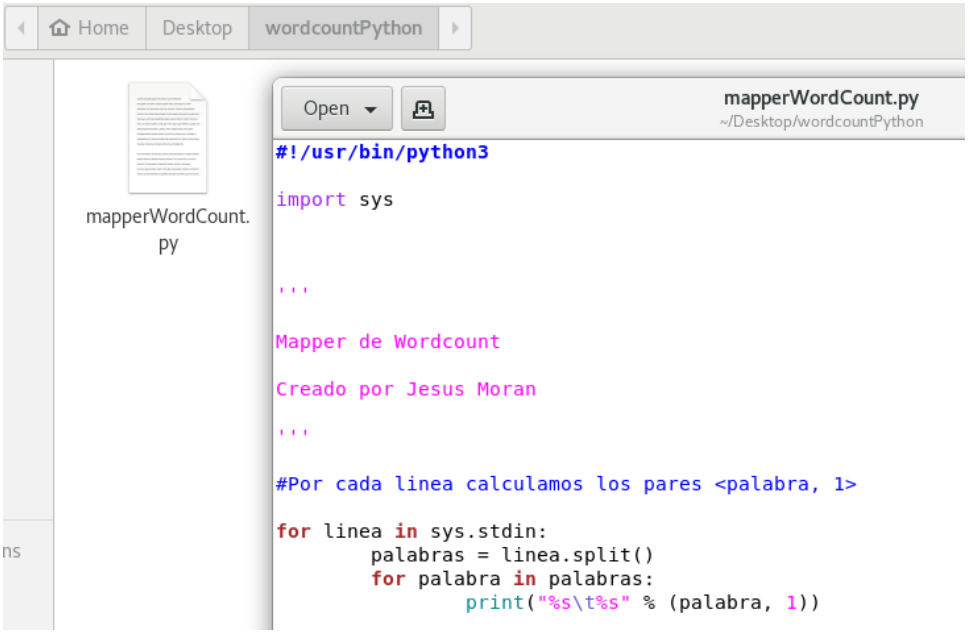
```
[moranjesus@localhost wordcountPython]$ touch mapperWordCount.py
[moranjesus@localhost wordcountPython]$
```

   d. Damos doble click en ese archivo y escribimos:

```
#!/usr/bin/python3
import sys

'''
Mapper de Wordcount
Creado por Jesus Moran
'''
#Por cada linea calculamos los pares <palabra, 1>
for linea in sys.stdin:
      palabras = linea.split()

      for palabra in palabras:
            print("%s\t%s" % (palabra, 1))
```



e. Le damos permisos de ejecución: chmod u+x mapperWordCount.py

```
[moranjesus@localhost wordcountPython]$ chmod u+x mapperWordCount.py
[moranjesus@localhost wordcountPython]$
```

f. Creamos un archivo llamado reducerWordCount.py:

touch reducerWordCount.py

```
[moranjesus@localhost wordcountPython]$ touch reducerWordCount.py
[moranjesus@localhost wordcountPython]$
```

g. Damos doble click en ese archivo y escribimos:

```
#!/usr/bin/python3
import sys

'''
Reducer de Wordcount
Creado por Jesus Moran
'''


subproblema = None
suma_conteos = 0

for claveValor in sys.stdin:
      palabra, conteo_palabra = claveValor.split("\t", 1)

      #convertimos conteo a entero
      conteo_palabra = int(conteo_palabra)

      #El primer subproblema es la primer palabra
      if subproblema == None:
            subproblema = palabra

      #si la palabra es del subproblema actual, sumamos
```

```
        if subproblema == palabra:
                suma_conteos = suma_conteos + conteo_palabra
        else: #si ya acabamos con el subproblema, emitimos
                print("%s\t%s" % (subproblema, suma_conteos))
                #Pasamos al siguiente subproblema
                subproblema = palabra
                suma_conteo = conteo

#el anterior bucle no emite el ultimo subproblema
print("%s\t%s" % (subproblema, suma_conteos))
```

```
#!/usr/bin/python3

import sys

'''
Reducer de Wordcount
Creado por Jesus Moran
'''

subproblema = None
suma_conteos = 0

for claveValor in sys.stdin:
        palabra, conteo_palabra = claveValor.split("\t", 1)

        #convertimos conteo a entero
        conteo_palabra = int(conteo_palabra)

        #El primer subproblema es la primer palabra
        if subproblema == None:
                subproblema = palabra

        #si la palabra es del subproblema actual, sumamos
        if subproblema == palabra:
                suma_conteos = suma_conteos + conteo_palabra
        else: #si ya acabamos con el subproblema, emitimos
                print("%s\t%s" % (subproblema, suma_conteos))

                #Pasamos al siguiente subproblema
                subproblema = palabra
                suma_conteo = conteo_palabra

#el anterior bucle no emite el ultimo subproblema
print("%s\t%s" % (subproblema, suma_conteos))
```

h. Le damos permisos de ejecución: chmod u+x reducerWordCount.py

```
[moranjesus@localhost wordcountPython]$ chmod u+x reducerWordCount.py
[moranjesus@localhost wordcountPython]$
```

2. Descargamos el Quijote:

   curl https://www.gutenberg.org/cache/epub/996/pg996.txt > quijote.txt

```
[moranjesus@localhost wordcountPython]$ curl https://www.gutenberg.org/cache/epu
b/996/pg996.txt > quijote.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 2335k  100 2335k    0     0  1334k      0  0:00:01  0:00:01 --:--:-- 1334k
[moranjesus@localhost wordcountPython]$
```

3. Ejecutamos en Hadoop el programa MapReduce:

hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-3.3.0.jar -files ./mapperWordCount.py,./reducerWordCount.py -mapper ./mapperWordCount.py -reducer ./reducerWordCount.py -input quijote.txt -output ./salidaQuijote

```
[moranjesus@localhost wordcountPython]$ hadoop jar $HADOOP_HOME/share/hadoop/too
ls/lib/hadoop-streaming-3.3.0.jar -files ./mapperWordCount.py,./reducerWordCount
.py -mapper ./mapperWordCount.py  -reducer ./reducerWordCount.py -input quijote.
txt -output ./salidaQuijote
```

```
File System Counters
        FILE: Number of bytes read=13190504
        FILE: Number of bytes written=14243332
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
Map-Reduce Framework
        Map input records=43276
        Map output records=430283
        Map output bytes=3200561
        Map output materialized bytes=4061133
        Input split bytes=109
        Combine input records=0
        Combine output records=0
        Reduce input groups=33436
        Reduce shuffle bytes=4061133
        Reduce input records=430283
        Reduce output records=33436
        Spilled Records=860566
        Shuffled Maps =1
        Failed Shuffles=0
        Merged Map outputs=1
        GC time elapsed (ms)=44
        Total committed heap usage (bytes)=296951808
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=2391235
File Output Format Counters
        Bytes Written=522631
```

4. Ejecutamos el programa con Combiner:

   hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-3.3.0.jar -files ./mapperWordCount.py,./reducerWordCount.py -mapper ./mapperWordCount.py -reducer ./reducerWordCount.py -combiner ./reducerWordCount.py -input quijote.txt -output ./salidaQuijote2

```
ote2
[moranjesus@localhost wordcountPython]$ hadoop jar $HADOOP_HOME/share/hadoop/too
ls/lib/hadoop-streaming-3.3.0.jar -files ./mapperWordCount.py,./reducerWordCount
.py -mapper ./mapperWordCount.py -reducer ./reducerWordCount.py -combiner ./redu
cerWordCount.py -input quijote.txt -output ./salidaQuijote2
```

```
File System Counters
        FILE: Number of bytes read=6239136
        FILE: Number of bytes written=3662330
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
Map-Reduce Framework
        Map input records=43276
        Map output records=430283
        Map output bytes=3200561
        Map output materialized bytes=585449
        Input split bytes=109
        Combine input records=430283
        Combine output records=33436
        Reduce input groups=33436
        Reduce shuffle bytes=585449
        Reduce input records=33436
        Reduce output records=33436
        Spilled Records=66872
        Shuffled Maps =1
        Failed Shuffles=0
        Merged Map outputs=1
        GC time elapsed (ms)=53
        Total committed heap usage (bytes)=296869888
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=2391235
File Output Format Counters
        Bytes Written=366937
```

Notar que Combine reduce “muchos” datos

5. Hacemos pruebas del programa sin utilizar Hadoop:
    a. Ejecutamos todo el programa Mapper-Reducer sin Hadoop:

        cat quijote.txt | ./mapperWordCount.py | sort -k1,1 |
        ./reducerWordCount.py > salidaQuijote3

        ```
        [moranjesus@localhost wordcountPython]$ cat quijote.txt | ./mapperWordCount.py | sort -k1,1 | ./reduce
        rWordCount.py > salidaQuijote3
        ```

        En el archivo SalidaQuijote3 tenemos el resultado

    b. Si queremos ver que hace el programa para una línea específica:

        printf "esta línea es una prueba es para contar" |
        ./mapperWordCount.py | sort -k1,1 | ./reducerWordCount.py

        ```
        [moranjesus@localhost wordcountPython]$ printf "esta linea es una prueba es para contar" | ./mapperWor
        dCount.py | sort -k1,1 | ./reducerWordCount.py
        contar  1
        es      2
        esta    1
        linea   1
        para    1
        prueba  1
        una     1
        [moranjesus@localhost wordcountPython]$
        ```

    c. Para probar varias líneas:

        printf "esta línea es una prueba es para contar\n esta es otra linea" |
        ./mapperWordCount.py | sort -k1,1 | ./reducerWordCount.py

```
[moranjesus@localhost wordcountPython]$ printf "esta linea es una prueba es para contar\n esta es otra
 linea" | ./mapperWordCount.py | sort -k1,1 | ./reducerWordCount.py
contar  1
es      3
esta    2
linea   2
otra    1
para    1
prueba  1
una     1
[moranjesus@localhost wordcountPython]$
```

d. Si queremos ver únicamente qué es lo que emite Mapper:

printf "esta línea es una prueba es para contar\n esta es otra linea" | ./mapperWordCount.py

```
[moranjesus@localhost wordcountPython]$ printf "esta linea es una prueba es para contar\n esta es otra
 linea" | ./mapperWordCount.py
esta    1
linea   1
es      1
una     1
prueba  1
es      1
para    1
contar  1
esta    1
es      1
otra    1
linea   1
[moranjesus@localhost wordcountPython]$
```