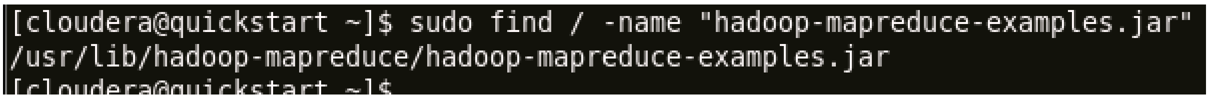
Mapreduce conteo de palabras

1. Verificar los Ejemplos de Hadoop Disponibles en Cloudera

Los ejemplos se encuentran en el archivo hadoop-mapreduce-examples.jar, que generalmente se instala con Hadoop. Para verificar su ubicación: bash

```
sudo find /-name "hadoop-mapreduce-examples.jar"
```

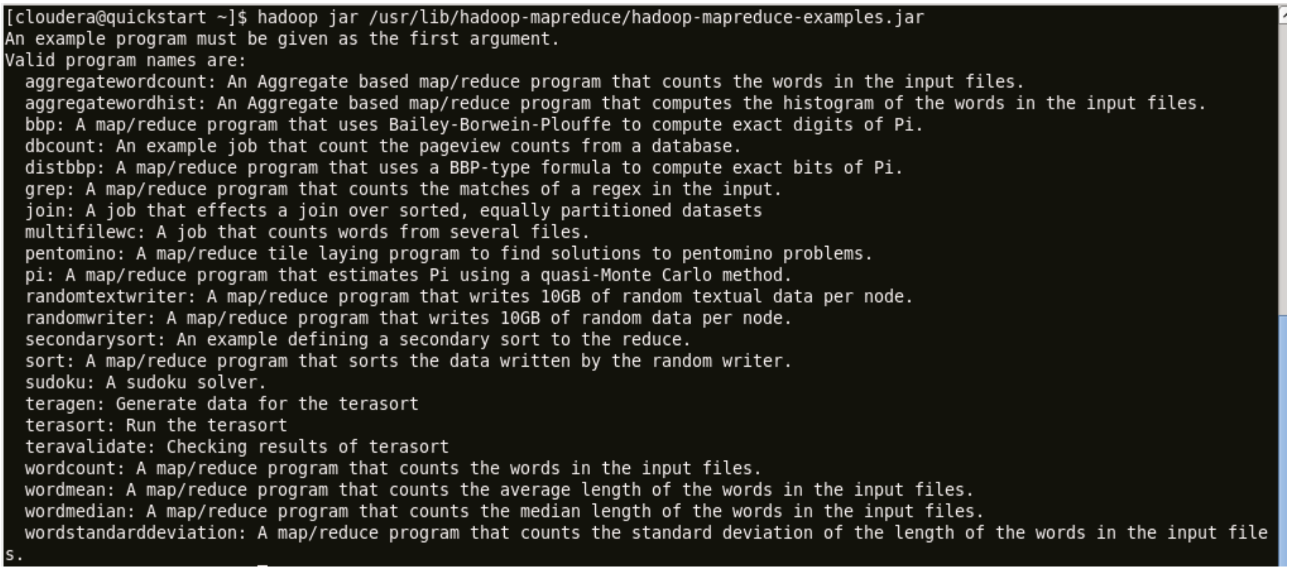


```
[cloudera@quickstart ~]$ sudo find / -name "hadoop-mapreduce-examples.jar"
/usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar
[cloudera@quickstart ~]$
```

Vemos que se encuentra en: /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar

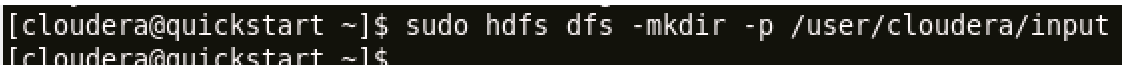
Consulta los ejemplos disponibles: bash

```
hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar
```

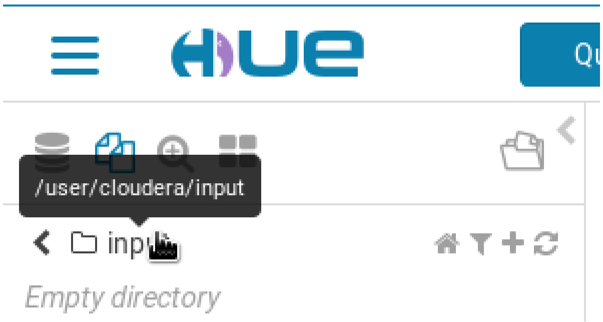


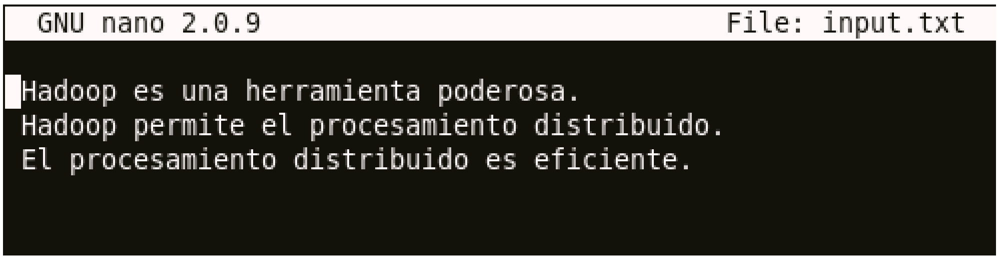
```
[cloudera@quickstart ~]$ hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar
An example program must be given as the first argument.
Valid program names are:
  aggregatewordcount: An Aggregate based map/reduce program that counts the words in the input files.
  aggregatewordhist: An Aggregate based map/reduce program that computes the histogram of the words in the input files.
  bbp: A map/reduce program that uses Bailey-Borwein-Plouffe to compute exact digits of Pi.
  dbcount: An example job that count the pageview counts from a database.
  distbbp: A map/reduce program that uses a BBP-type formula to compute exact bits of Pi.
  grep: A map/reduce program that counts the matches of a regex in the input.
  join: A job that effects a join over sorted, equally partitioned datasets
  multifilewc: A job that counts words from several files.
  pentomino: A map/reduce tile laying program to find solutions to pentomino problems.
  pi: A map/reduce program that estimates Pi using a quasi-Monte Carlo method.
  randomtextwriter: A map/reduce program that writes 10GB of random textual data per node.
  randomwriter: A map/reduce program that writes 10GB of random data per node.
  secondarysort: An example defining a secondary sort to the reduce.
  sort: A map/reduce program that sorts the data written by the random writer.
  sudoku: A sudoku solver.
  teragen: Generate data for the terasort
  terasort: Run the terasort
  teravalidate: Checking results of terasort
  wordcount: A map/reduce program that counts the words in the input files.
  wordmean: A map/reduce program that counts the average length of the words in the input files.
  wordmedian: A map/reduce program that counts the median length of the words in the input files.
  wordstandarddeviation: A map/reduce program that counts the standard deviation of the length of the words in the input file
s.
```

2. Prepara los Datos de Entrada en HDFS Crea un Directorio de Entrada en HDFS: bash



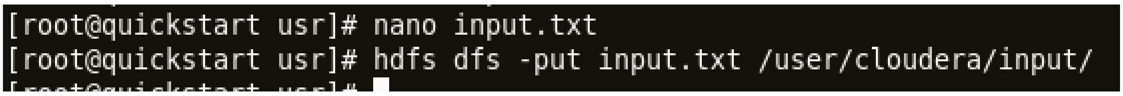
```
[cloudera@quickstart ~]$ sudo hdfs dfs -mkdir -p /user/cloudera/input
[cloudera@quickstart ~]$
```





```
  GNU nano 2.0.9                                File: input.txt

Hadoop es una herramienta poderosa.
Hadoop permite el procesamiento distribuido.
El procesamiento distribuido es eficiente.
```

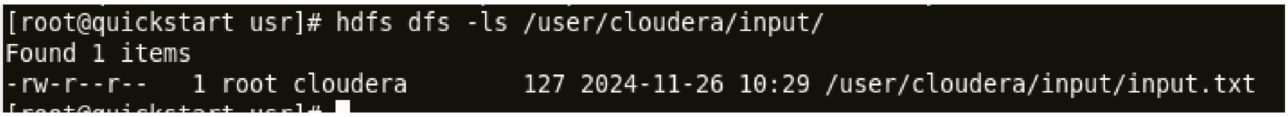
Lo pasamos a cloudera:



```
[root@quickstart usr]# nano input.txt
[root@quickstart usr]# hdfs dfs -put input.txt /user/cloudera/input/
[root@quickstart usr]#
```

Verifica que el archivo esté disponible en HDFS:

Usamos el comando más moderno de entre los dos recomendados:



```
[root@quickstart usr]# hdfs dfs -ls /user/cloudera/input/
Found 1 items
-rw-r--r--   1 root cloudera        127 2024-11-26 10:29 /user/cloudera/input/input.txt
[root@quickstart usr]#
```

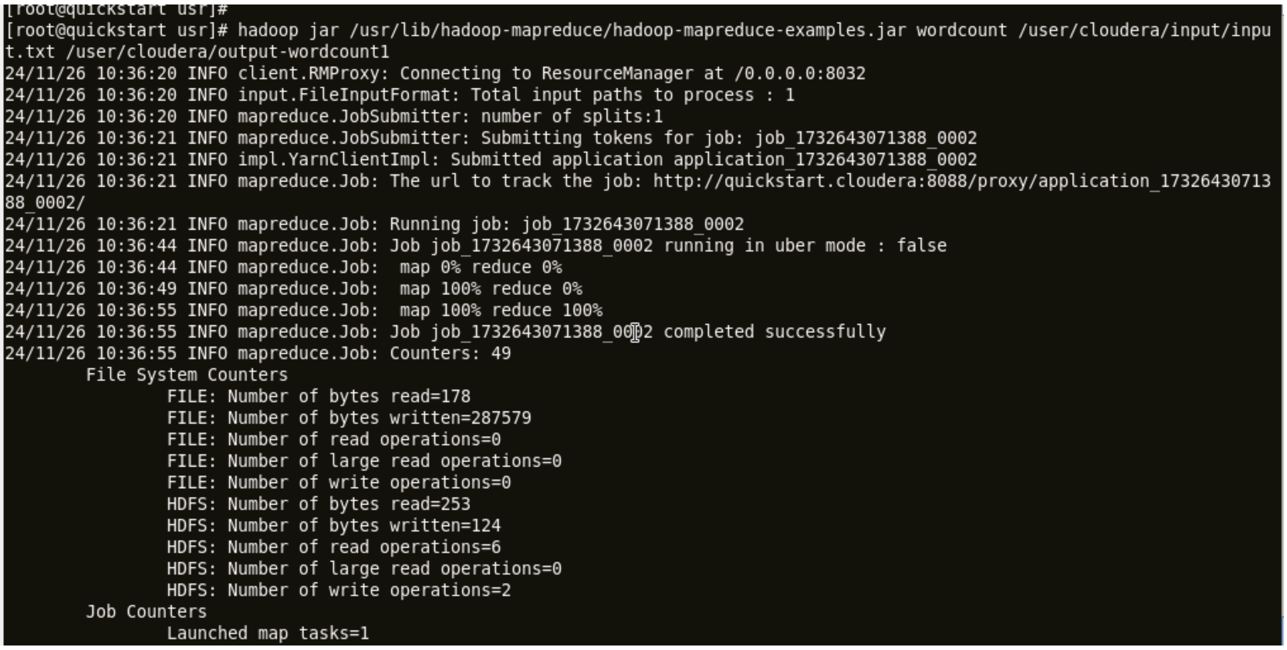
3. Ejecuta un Ejemplo MapReduce

Ejemplo 1: Contar Palabras con wordcount

Ejecuta el Ejemplo:

bash

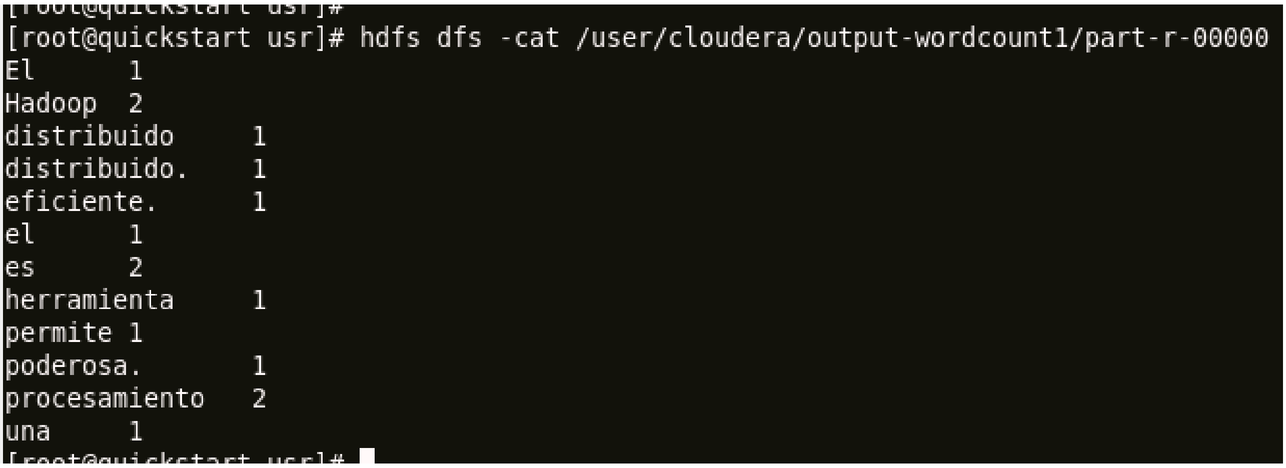
```
hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar
wordcount /user/cloudera/input/input.txt /user/cloudera/output-wordcount
```



```
[root@quickstart usr]#
[root@quickstart usr]# hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar wordcount /user/cloudera/input/inpu
t.txt /user/cloudera/output-wordcount1
24/11/26 10:36:20 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
24/11/26 10:36:20 INFO input.FileInputFormat: Total input paths to process : 1
24/11/26 10:36:20 INFO mapreduce.JobSubmitter: number of splits:1
24/11/26 10:36:21 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1732643071388_0002
24/11/26 10:36:21 INFO impl.YarnClientImpl: Submitted application application_1732643071388_0002
24/11/26 10:36:21 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_17326430713
88_0002/
24/11/26 10:36:21 INFO mapreduce.Job: Running job: job_1732643071388_0002
24/11/26 10:36:44 INFO mapreduce.Job: Job job_1732643071388_0002 running in uber mode : false
24/11/26 10:36:44 INFO mapreduce.Job:  map 0% reduce 0%
24/11/26 10:36:49 INFO mapreduce.Job:  map 100% reduce 0%
24/11/26 10:36:55 INFO mapreduce.Job:  map 100% reduce 100%
24/11/26 10:36:55 INFO mapreduce.Job: Job job_1732643071388_0002 completed successfully
24/11/26 10:36:55 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=178
                FILE: Number of bytes written=287579
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=253
                HDFS: Number of bytes written=124
                HDFS: Number of read operations=6
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
```

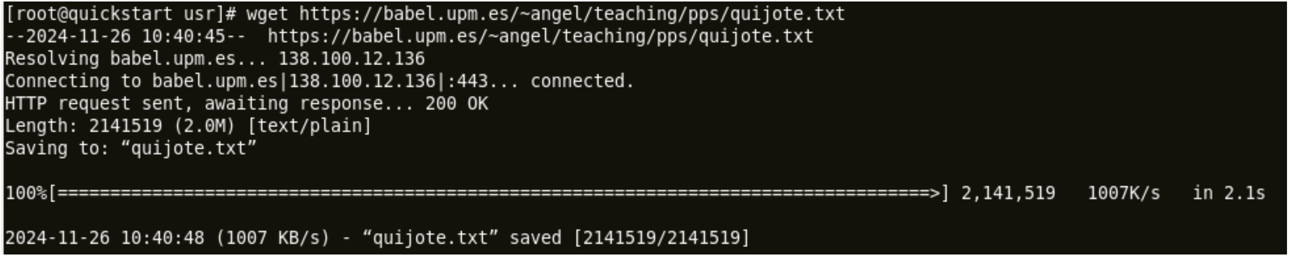
Verifica el Resultado: Después de que el trabajo se complete, los resultados estarán en /user/cloudera/output-wordcount.

```
hdfs dfs-cat /user/cloudera/output-wordcount/part-r-00000
```



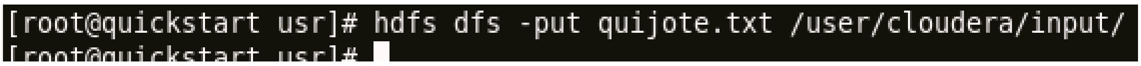
```
[root@quickstart usr]#
[root@quickstart usr]# hdfs dfs -cat /user/cloudera/output-wordcount1/part-r-00000
El      1
Hadoop  2
distribuido     1
distribuido.    1
eficiente.      1
el      1
es      2
herramienta     1
permite 1
poderosa.       1
procesamiento   2
una     1
[root@quickstart usr]#
```

Realizar la misma operación con el siguiente archivo: https://babel.upm.es/~angel/teaching/pps/quijote.txt
Descargamos el archivo en cuestión:



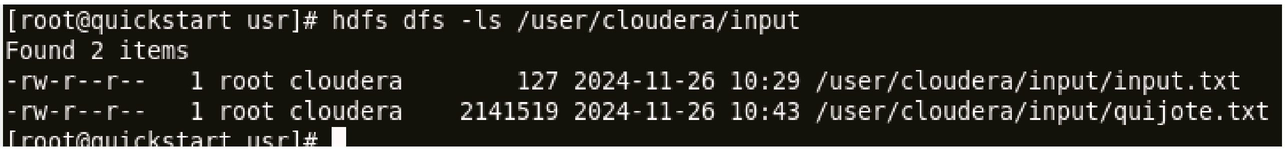
```
[root@quickstart usr]# wget https://babel.upm.es/~angel/teaching/pps/quijote.txt
--2024-11-26 10:40:45--  https://babel.upm.es/~angel/teaching/pps/quijote.txt
Resolving babel.upm.es... 138.100.12.136
Connecting to babel.upm.es|138.100.12.136|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2141519 (2.0M) [text/plain]
Saving to: “quijote.txt”

100%[=====================================================================================>] 2,141,519   1007K/s   in 2.1s

2024-11-26 10:40:48 (1007 KB/s) - “quijote.txt” saved [2141519/2141519]
```

Subimos el archivo a cloudera:



```
[root@quickstart usr]# hdfs dfs -put quijote.txt /user/cloudera/input/
[root@quickstart usr]#
```

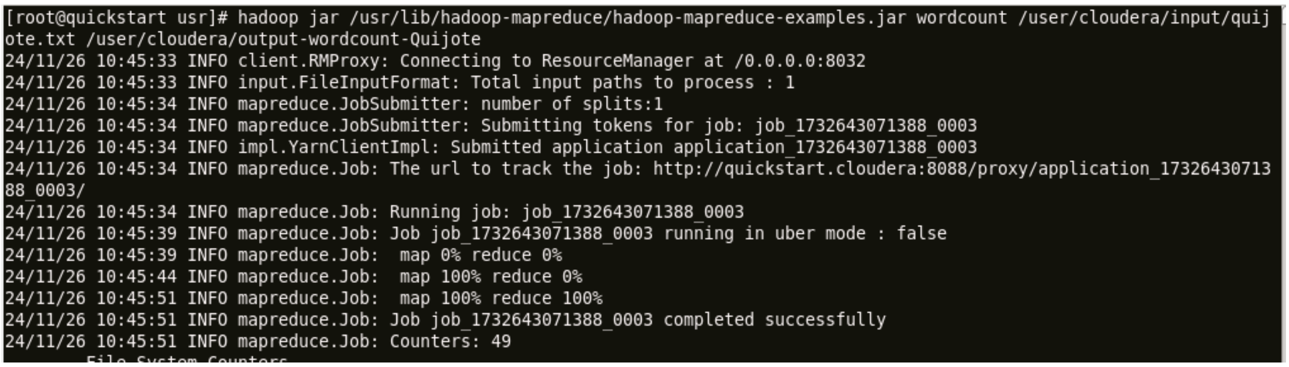
Verificamos que también se ha subido correctamente:



```
[root@quickstart usr]# hdfs dfs -ls /user/cloudera/input
Found 2 items
-rw-r--r--   1 root cloudera        127 2024-11-26 10:29 /user/cloudera/input/input.txt
-rw-r--r--   1 root cloudera    2141519 2024-11-26 10:43 /user/cloudera/input/quijote.txt
[root@quickstart usr]#
```

Ejecutamos el ejemplo:

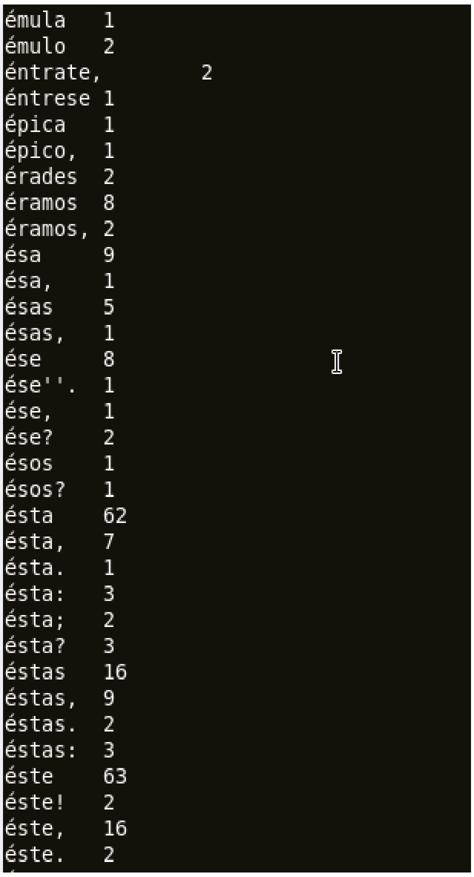
```
hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar
wordcount /user/cloudera/input/quijote.txt /user/cloudera/output-
wordcount-Quijote
```



```
[root@quickstart usr]# hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples.jar wordcount /user/cloudera/input/quij
ote.txt /user/cloudera/output-wordcount-Quijote
24/11/26 10:45:33 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
24/11/26 10:45:33 INFO input.FileInputFormat: Total input paths to process : 1
24/11/26 10:45:34 INFO mapreduce.JobSubmitter: number of splits:1
24/11/26 10:45:34 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1732643071388_0003
24/11/26 10:45:34 INFO impl.YarnClientImpl: Submitted application application_1732643071388_0003
24/11/26 10:45:34 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_17326430713
88_0003/
24/11/26 10:45:34 INFO mapreduce.Job: Running job: job_1732643071388_0003
24/11/26 10:45:39 INFO mapreduce.Job: Job job_1732643071388_0003 running in uber mode : false
24/11/26 10:45:39 INFO mapreduce.Job:  map 0% reduce 0%
24/11/26 10:45:44 INFO mapreduce.Job:  map 100% reduce 0%
24/11/26 10:45:51 INFO mapreduce.Job:  map 100% reduce 100%
24/11/26 10:45:51 INFO mapreduce.Job: Job job_1732643071388_0003 completed successfully
24/11/26 10:45:51 INFO mapreduce.Job: Counters: 49
        File System Counters
```

Comprobamos el resultado:

```
hdfs dfs -cat /user/cloudera/output-wordcount-Quijote/part-r-00000
```



```
émula   1
émulo   2
éntrate,        2
éntrese 1
épica   1
épico,  1
érades  2
éramos  8
éramos, 2
ésa     9
ésa,    1
ésas    5
ésas,   1
ése     8
ése''.  1
ése,    1
ése?    2
ésos    1
ésos?   1
ésta    62
ésta,   7
ésta.   1
ésta:   3
ésta;   2
ésta?   3
éstas   16
éstas,  9
éstas.  2
éstas:  3
éste    63
éste!   2
éste,   16
éste.   2
```