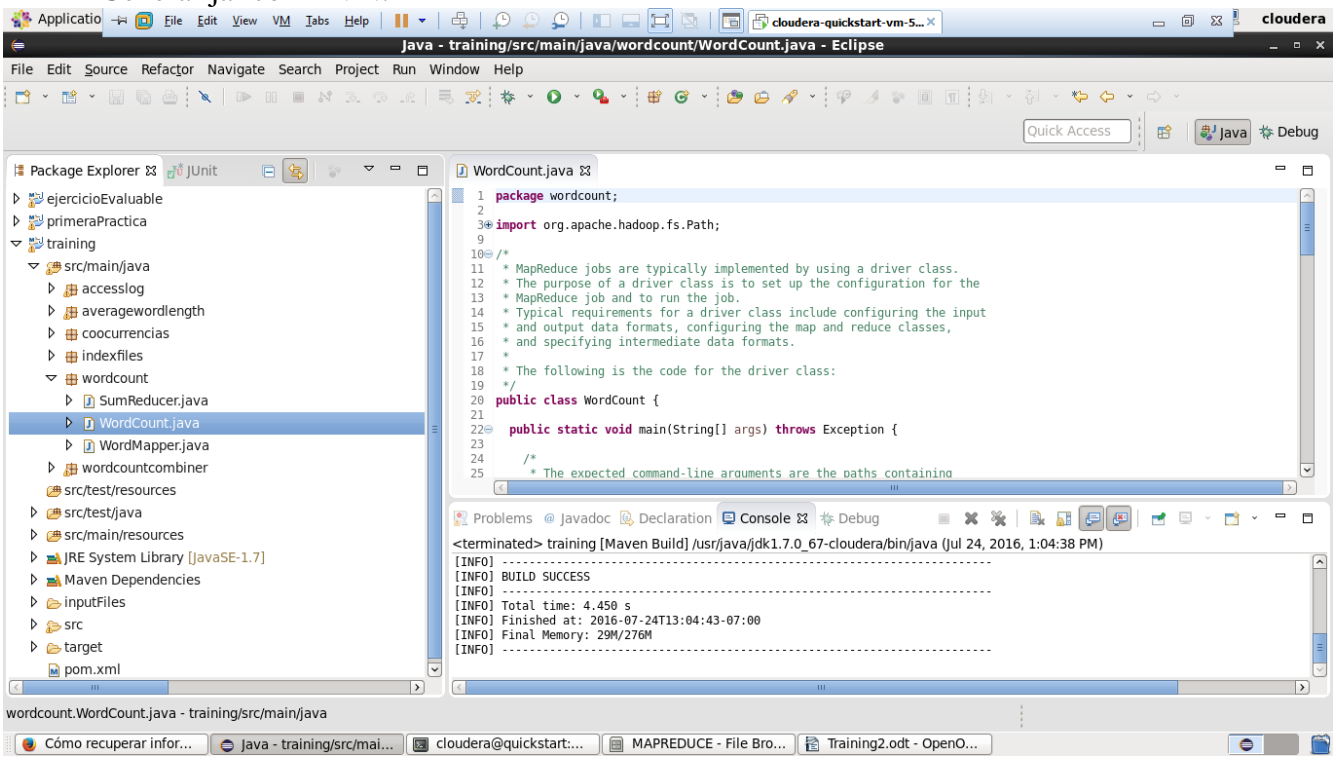
- NOTAS a tener en cuenta:

  1. Maven necesita una estructura fija para poder generar correctamente el .jar (src/main y src/test y dentro de cada uno src/main/java src/main/resources y src/test/main src/test/resources y cada uno va a ir a una /classes diferente)
  2. El log debe estar situado en src/main/resources y debe compilarse hacia target/classes y por lotanto el log4java.propertis tiene que estar a primer nivel en target/classes
  3. Cuando se ejecute hadoop o yarn y se le indique la clase se debe indicar toda la ruta, por ejemplo com.mbit.WordCount o wordcount.WordCount sino no la encuentra.
  4. Si por ejemplo se quiere procesar un archivo (de forma distribuida) pero no es necesario reducir informacion se puede planificar un Job sin Reducer, solo con Mappers e incluso se pueden incluir Contadores. Ejemplo: log con imagenes.

- WORDCOUNT:

  Generar jar con MVN:



Ejecutar tarea hadoop o yarn:

Previamente ha sido movido los ficheros de entrada a HDFS y no se ha creado la carpeta de salida.

```
[cloudera@quickstart target]$ hadoop jar training-0.0.1-SNAPSHOT.jar wordcount.WordCount
hdfs:///user/cloudera/training/inputFiles/shakespeare hdfs:///user/cloudera/training/output/wordCount
16/07/23 12:41:11 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
16/07/23 12:41:11 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed.
Implement the Tool interface and execute your application with ToolRunner to remedy this.
16/07/23 12:41:12 INFO input.FileInputFormat: Total input paths to process : 5
16/07/23 12:41:12 INFO mapreduce.JobSubmitter: number of splits:5
16/07/23 12:41:12 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1469299672451_0001
16/07/23 12:41:12 INFO impl.YarnClientImpl: Submitted application application_1469299672451_0001
16/07/23 12:41:12 INFO mapreduce.Job: The url to track the job:
http://quickstart.cloudera:8088/proxy/application_1469299672451_0001/
16/07/23 12:41:12 INFO mapreduce.Job: Running job: job_1469299672451_0001
16/07/23 12:41:19 INFO mapreduce.Job: Job job_1469299672451_0001 running in uber mode : false
16/07/23 12:41:19 INFO mapreduce.Job:  map 0% reduce 0%
```

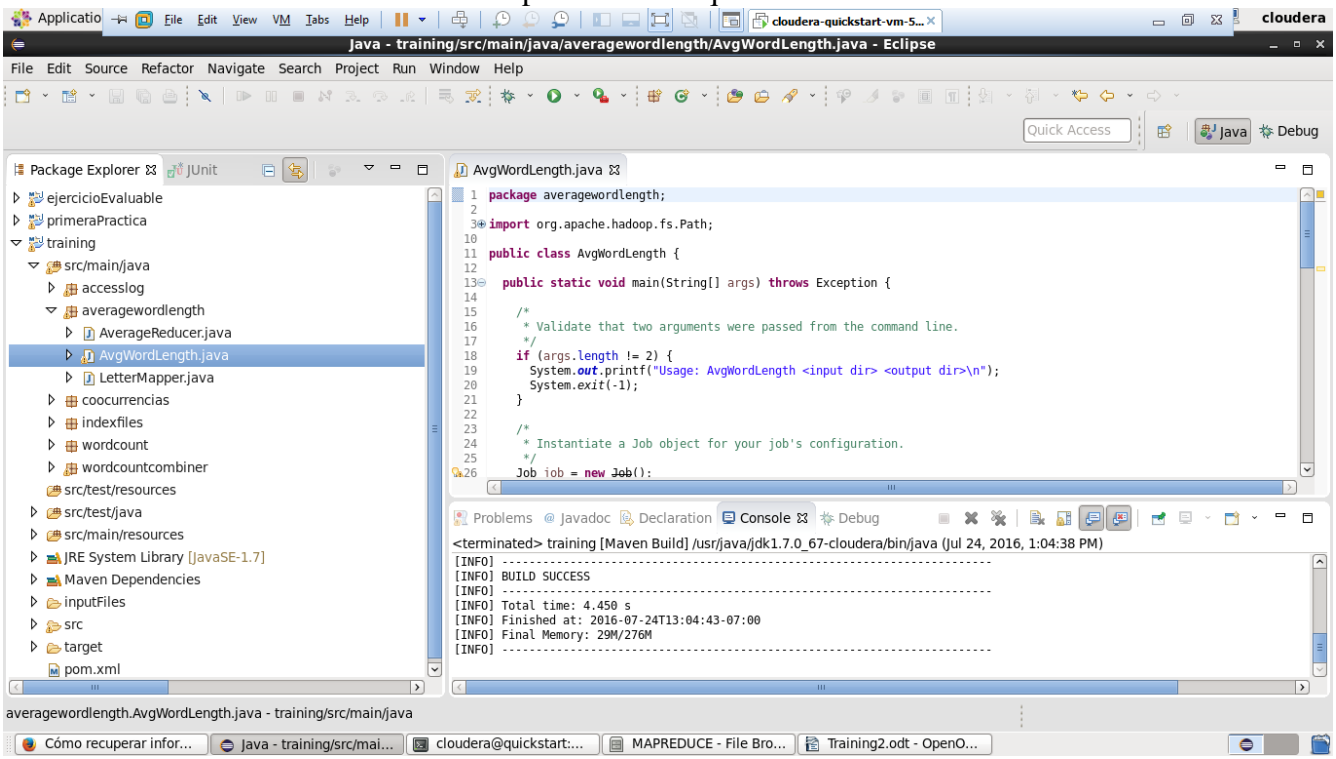
```
16/07/23 12:41:29 INFO mapreduce.Job:  map 20% reduce 0%
16/07/23 12:41:32 INFO mapreduce.Job:  map 80% reduce 0%
16/07/23 12:41:33 INFO mapreduce.Job:  map 100% reduce 0%
16/07/23 12:41:37 INFO mapreduce.Job:  map 100% reduce 100%
16/07/23 12:41:37 INFO mapreduce.Job: Job job_1469299672451_0001 completed successfully
16/07/23 12:41:37 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=10828596
                FILE: Number of bytes written=22337897
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=5343961
                HDFS: Number of bytes written=324841
                HDFS: Number of read operations=18
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=5
                Launched reduce tasks=1
                Data-local map tasks=5
                Total time spent by all maps in occupied slots (ms)=38507
                Total time spent by all reduces in occupied slots (ms)=5269
                Total time spent by all map tasks (ms)=38507
                Total time spent by all reduce tasks (ms)=5269
                Total vcore-seconds taken by all map tasks=38507
                Total vcore-seconds taken by all reduce tasks=5269
                Total megabyte-seconds taken by all map tasks=39431168
                Total megabyte-seconds taken by all reduce tasks=5395456
        Map-Reduce Framework
                Map input records=175558
                Map output records=974078
                Map output bytes=8880434
                Map output materialized bytes=10828620
                Input split bytes=754
                Combine input records=0
                Combine output records=0
                Reduce input groups=31809
                Reduce shuffle bytes=10828620
                Reduce input records=974078
                Reduce output records=31809
                Spilled Records=1948156
                Shuffled Maps =5
                Failed Shuffles=0
                Merged Map outputs=5
                GC time elapsed (ms)=899
                CPU time spent (ms)=15170
                Physical memory (bytes) snapshot=1832407040
                Virtual memory (bytes) snapshot=9365557248
                Total committed heap usage (bytes)=1917845504
        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=5343207
        File Output Format Counters
                Bytes Written=324841
[cloudera@quickstart target]$
```

Se recogen todos los ficheros ubicados en **hdfs:///user/cloudera/training/inputFiles/shakespeare** y se deja el resultado en la carpeta, la cual se crea, en **hdfs:///user/cloudera/training/output/wordCount**

```
[cloudera@quickstart target]$ hadoop fs -ls hdfs:///user/cloudera/training/output/wordCount
Found 2 items
-rw-r--r--   1 cloudera cloudera          0 2016-07-23 12:41 hdfs:///user/cloudera/training/output/wordCount/_SUCCESS
-rw-r--r--   1 cloudera cloudera     324841 2016-07-23 12:41 hdfs:///user/cloudera/training/output/wordCount/part-r-
00000
```

- MEDIA DE PALABRAS: averagewordlength

  Hacemos exactamente lo mismo para la media que hemos hecho antes:



Y ejecutamos:

```
[cloudera@quickstart target]$ hadoop jar training-0.0.1-SNAPSHOT.jar averagewordlength.AvgWordLength
hdfs:///user/cloudera/training/inputFiles/shakespeare hdfs:///user/cloudera/training/output/avgWordLength
16/07/23 16:53:51 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
16/07/23 16:53:51 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the
Tool interface and execute your application with ToolRunner to remedy this.
16/07/23 16:53:51 INFO input.FileInputFormat: Total input paths to process : 5
16/07/23 16:53:51 INFO mapreduce.JobSubmitter: number of splits:5
16/07/23 16:53:52 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1469299672451_0002
16/07/23 16:53:52 INFO impl.YarnClientImpl: Submitted application application_1469299672451_0002
16/07/23 16:53:52 INFO mapreduce.Job: The url to track the job:
http://quickstart.cloudera:8088/proxy/application_1469299672451_0002/
16/07/23 16:53:52 INFO mapreduce.Job: Running job: job_1469299672451_0002
16/07/23 16:53:57 INFO mapreduce.Job: Job job_1469299672451_0002 running in uber mode : false
16/07/23 16:53:57 INFO mapreduce.Job:  map 0% reduce 0%
16/07/23 16:54:12 INFO mapreduce.Job:  map 13% reduce 0%
16/07/23 16:54:13 INFO mapreduce.Job:  map 20% reduce 0%
16/07/23 16:54:16 INFO mapreduce.Job:  map 40% reduce 0%
16/07/23 16:54:17 INFO mapreduce.Job:  map 60% reduce 0%
16/07/23 16:54:18 INFO mapreduce.Job:  map 100% reduce 0%
16/07/23 16:54:22 INFO mapreduce.Job:  map 100% reduce 100%
16/07/23 16:54:22 INFO mapreduce.Job: Job job_1469299672451_0002 completed successfully
16/07/23 16:54:22 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=7792630
                FILE: Number of bytes written=16268203
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=5343961
                HDFS: Number of bytes written=1113
                HDFS: Number of read operations=18
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=5
                Launched reduce tasks=1
                Data-local map tasks=5
```

```
                Total time spent by all maps in occupied slots (ms)=76325
                Total time spent by all reduces in occupied slots (ms)=6254
                Total time spent by all map tasks (ms)=76325
                Total time spent by all reduce tasks (ms)=6254
                Total vcore-seconds taken by all map tasks=76325
                Total vcore-seconds taken by all reduce tasks=6254
                Total megabyte-seconds taken by all map tasks=78156800
                Total megabyte-seconds taken by all reduce tasks=6404096
        Map-Reduce Framework
                Map input records=175558
                Map output records=974078
                Map output bytes=5844468
                Map output materialized bytes=7792654
                Input split bytes=754
                Combine input records=0
                Combine output records=0
                Reduce input groups=60
                Reduce shuffle bytes=7792654
                Reduce input records=974078
                Reduce output records=60
                Spilled Records=1948156
                Shuffled Maps =5
                Failed Shuffles=0
                Merged Map outputs=5
                GC time elapsed (ms)=7873
                CPU time spent (ms)=29710
                Physical memory (bytes) snapshot=1695772672
                Virtual memory (bytes) snapshot=9386377216
                Total committed heap usage (bytes)=1645740032
        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=5343207
        File Output Format Counters
                Bytes Written=1113
```

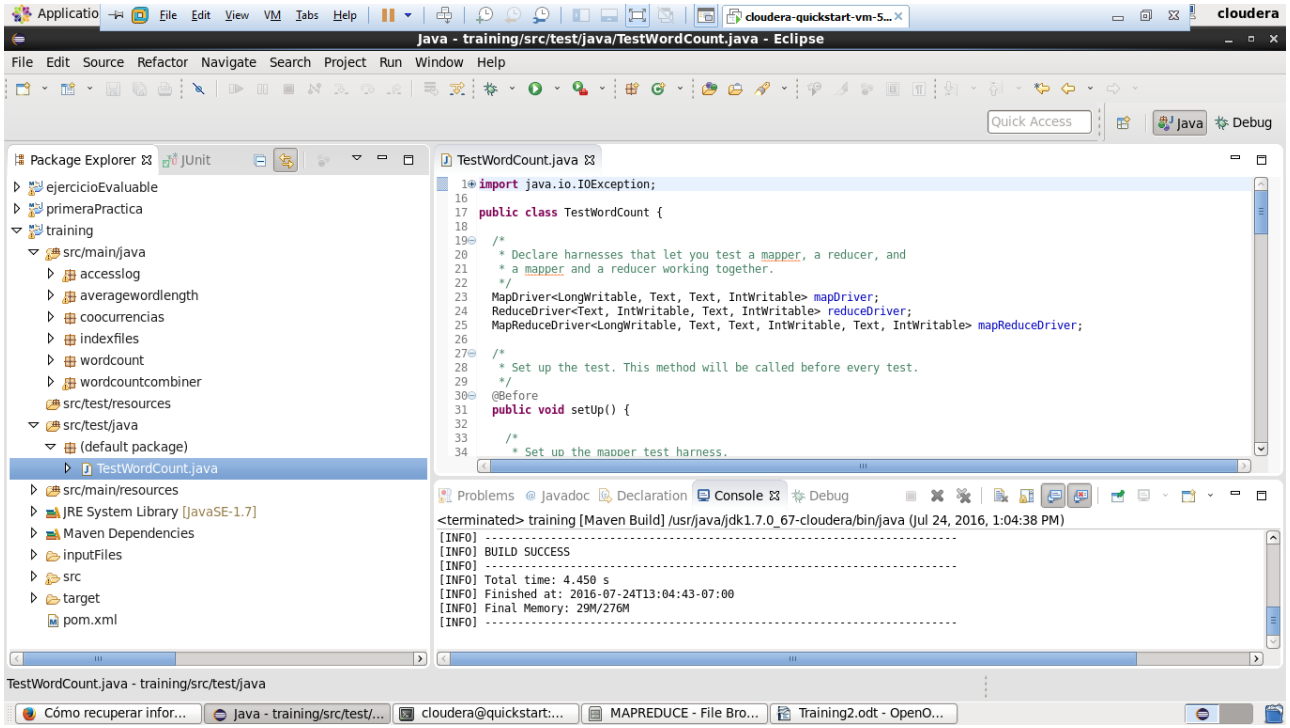
Visualizamos el resultado:

```
9       1.0
A       3.901754225255347
B       5.143532818532819
C       6.634214463840399
D       5.221781152916811
E       5.53018939875429
F       5.265583343912657
G       5.810282153366799
H       4.428398058252427
I       1.4687346778674861
J       4.97845507094062
K       4.659987476518472
L       5.116772823779193
M       5.451585352834419
N       3.991723259762309
O       2.8934336691346036
P       6.502230031085282
Q       5.536977491961415
R       5.930306748466258
S       5.307761868877167
T       3.965374320006908
U       5.421190893169878
V       5.2165160230073955
W       4.471301066686017
X       3.2211538461538463
Y       3.448119498532942
Z       6.0
a       3.0712166172106823
b       4.252546094225326
c       6.06068652351266
d       4.163519460657324
e       5.206521739130435
f       4.784952757916241
g       4.940715543947033
h       3.883610494523489
i       2.7480451279683757
j       5.341365461847389
k       4.6065459610027855
l       4.280937316068275
m       3.728475485549483
n       3.708169228814636
```

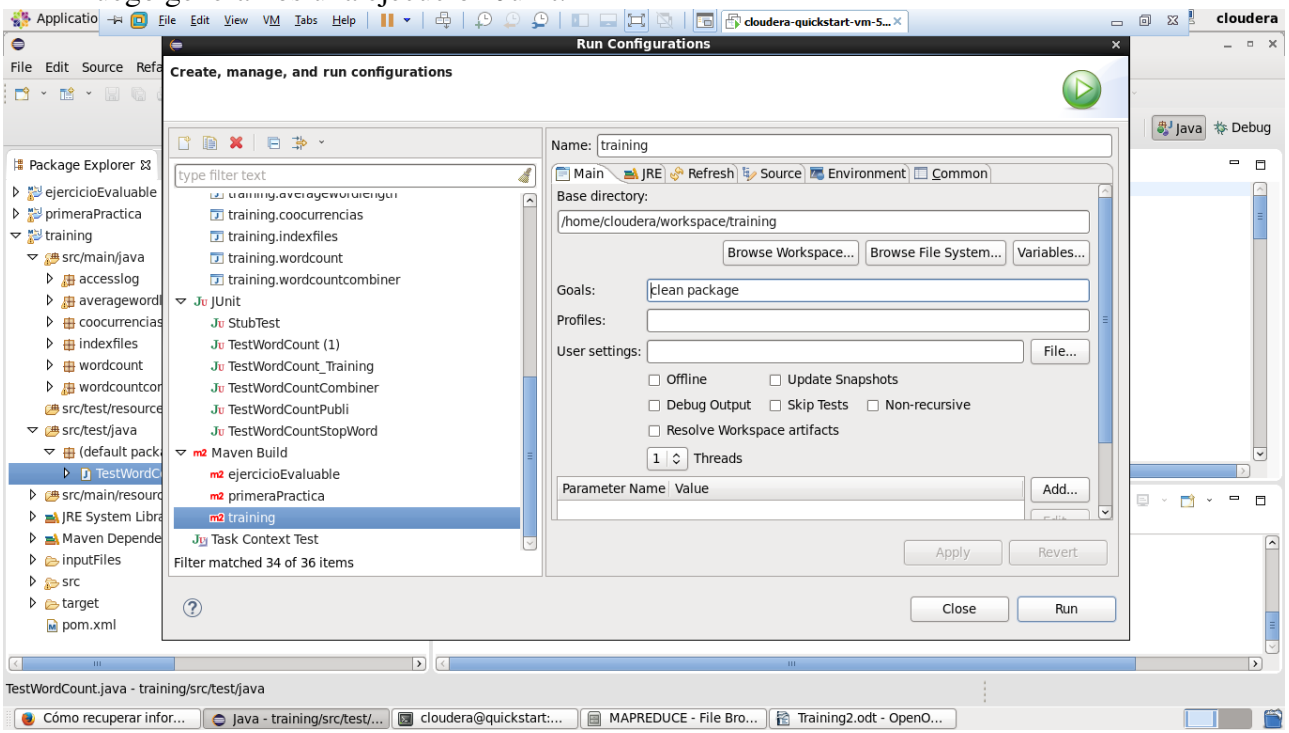
```
o       2.7891238670694865
p       6.108971062596419
q       6.034207525655645
r       5.8439986163957105
s       4.339266369764083
t       3.7265960492413397
u       4.511729670596815
v       5.734653024911032
w       4.350099946966916
y       3.5301620582710873
z       4.672727272727273
```

- MRUnits:

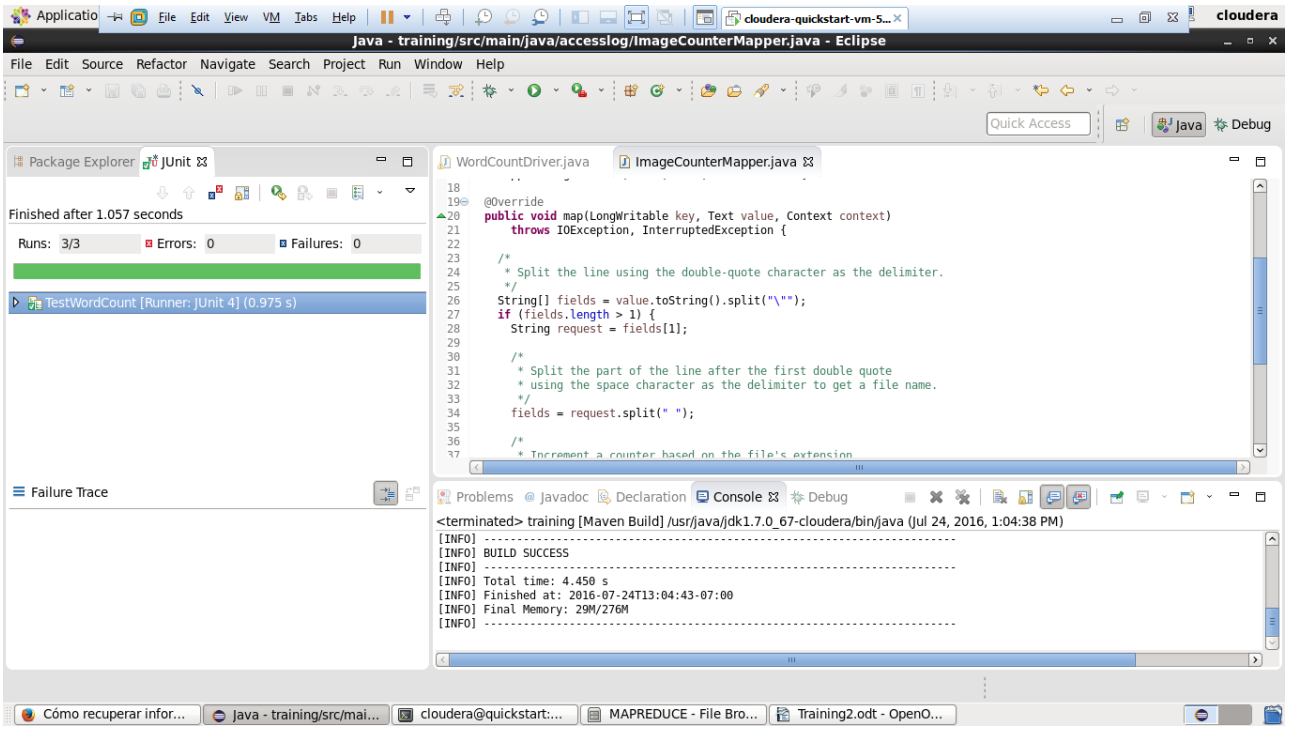
  Incluimos clases en proyecto y las adaptamos para que tiren de SumReducer y WordMapper correcto.



Luego generamos una ejecucion Junit:



Y podemos lanzarla desde eclipse y el resultado debe darnos 0 errores y tantas pruebas como @Test tenemos:

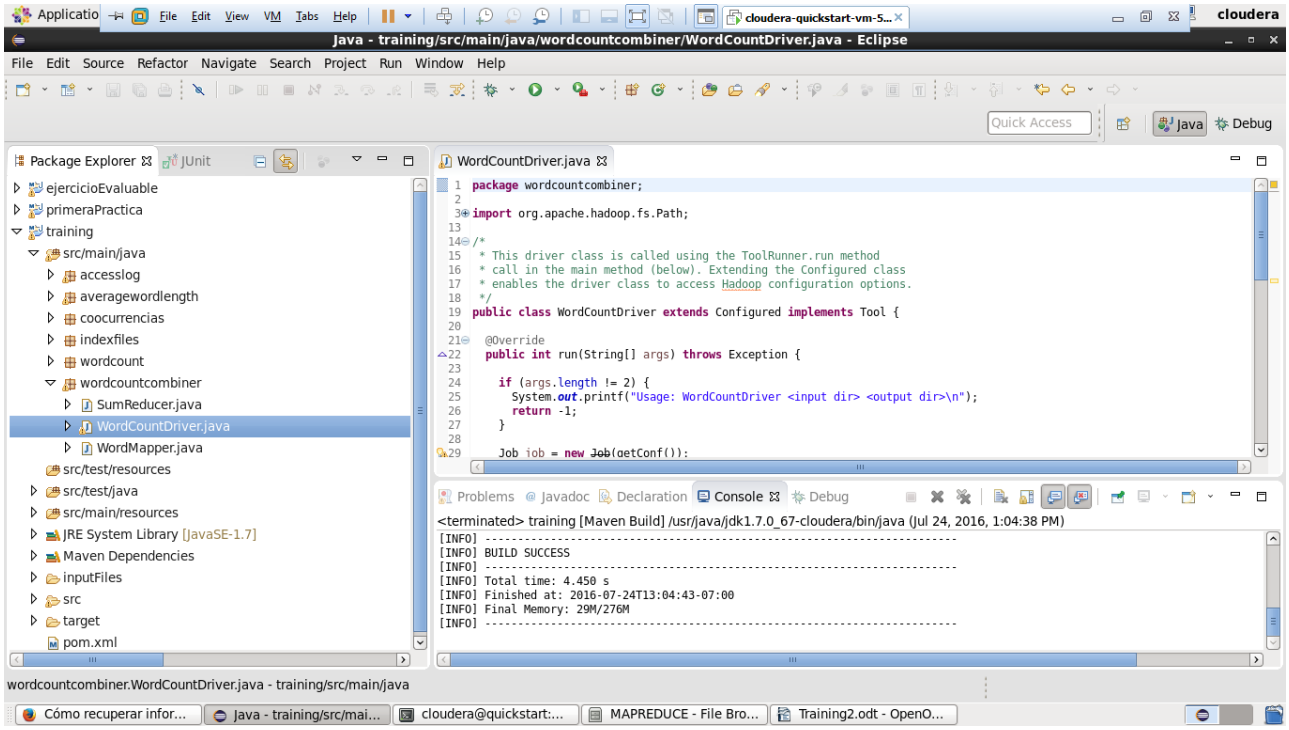


- Si lanzamos la compilacion por maven tambien lanza los Junit:

```
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building training 0.0.1-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ training ---
[INFO] Deleting /home/cloudera/workspace/training/target
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ training ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 1 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ training ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 6 source files to /home/cloudera/workspace/training/target/classes
[WARNING] /home/cloudera/workspace/training/src/main/java/averagewordlength/AvgWordLength.java:
/home/cloudera/workspace/training/src/main/java/averagewordlength/AvgWordLength.java uses or overrides a
deprecated API.
[WARNING] /home/cloudera/workspace/training/src/main/java/averagewordlength/AvgWordLength.java: Recompile with
-Xlint:deprecation for details.
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ training ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ training ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to /home/cloudera/workspace/training/target/test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ training ---
[INFO] Surefire report directory: /home/cloudera/workspace/training/target/surefire-reports

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running TestWordCount
Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.019 sec

Results :

Tests run: 3, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ training ---
[INFO] Building jar: /home/cloudera/workspace/training/target/training-0.0.1-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 4.590 s
[INFO] Finished at: 2016-07-23T17:22:31-07:00
[INFO] Final Memory: 29M/274M
[INFO] ------------------------------------------------------------------------
```

- UTILIZACION DE COMBINERS

  Vamos a lanzar el WordCount pero utilizando Combiners. Para ello cremos un nuevo paquete y copiamos las clases dentro. Lanzamos una ejecucion y revisamos el fichero de salida.



Tras este punto generamos de nuevo el jar con MVN y volvemos a lanzarlo en HDFS con otra ruta de salida.

```
[cloudera@quickstart target]$ hadoop jar training-0.0.1-SNAPSHOT.jar wordcountcombiner.WordCountDriver
hdfs:///user/cloudera/training/inputFiles/shakespeare hdfs:///user/cloudera/training/output/wordcountcombiner
16/07/24 05:16:50 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
16/07/24 05:16:50 INFO input.FileInputFormat: Total input paths to process : 5
16/07/24 05:16:50 INFO mapreduce.JobSubmitter: number of splits:5
16/07/24 05:16:50 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1469299672451_0003
16/07/24 05:16:51 INFO impl.YarnClientImpl: Submitted application application_1469299672451_0003
16/07/24 05:16:51 INFO mapreduce.Job: The url to track the job:
http://quickstart.cloudera:8088/proxy/application_1469299672451_0003/
16/07/24 05:16:51 INFO mapreduce.Job: Running job: job_1469299672451_0003
16/07/24 05:16:56 INFO mapreduce.Job: Job job_1469299672451_0003 running in uber mode : false
16/07/24 05:16:56 INFO mapreduce.Job:  map 0% reduce 0%
16/07/24 05:17:08 INFO mapreduce.Job:  map 20% reduce 0%
16/07/24 05:17:14 INFO mapreduce.Job:  map 33% reduce 0%
16/07/24 05:17:15 INFO mapreduce.Job:  map 40% reduce 0%
16/07/24 05:17:20 INFO mapreduce.Job:  map 60% reduce 0%
16/07/24 05:17:21 INFO mapreduce.Job:  map 100% reduce 0%
16/07/24 05:17:22 INFO mapreduce.Job:  map 100% reduce 100%
16/07/24 05:17:22 INFO mapreduce.Job: Job job_1469299672451_0003 completed successfully
16/07/24 05:17:22 INFO mapreduce.Job: Counters: 50
        File System Counters
                FILE: Number of bytes read=838036
                FILE: Number of bytes written=2358865
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=5343961
                HDFS: Number of bytes written=324841
                HDFS: Number of read operations=18
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Killed map tasks=1
                Launched map tasks=6
                Launched reduce tasks=1
                Data-local map tasks=6
```

```
                Total time spent by all maps in occupied slots (ms)=85444
                Total time spent by all reduces in occupied slots (ms)=11665
                Total time spent by all map tasks (ms)=85444
                Total time spent by all reduce tasks (ms)=11665
                Total vcore-seconds taken by all map tasks=85444
                Total vcore-seconds taken by all reduce tasks=11665
                Total megabyte-seconds taken by all map tasks=87494656
                Total megabyte-seconds taken by all reduce tasks=11944960
        Map-Reduce Framework
                Map input records=175558
                Map output records=974078
                Map output bytes=8880434
                Map output materialized bytes=838060
                Input split bytes=754
                Combine input records=974078
                Combine output records=61369
                Reduce input groups=31809
                Reduce shuffle bytes=838060
                Reduce input records=61369
                Reduce output records=31809
                Spilled Records=122738
                Shuffled Maps =5
                Failed Shuffles=0
                Merged Map outputs=5
                GC time elapsed (ms)=16570
                CPU time spent (ms)=29110
                Physical memory (bytes) snapshot=1719341056
                Virtual memory (bytes) snapshot=9391128576
                Total committed heap usage (bytes)=1570766848
        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=5343207
        File Output Format Counters
                Bytes Written=324841
```

Si nos fijamos en los counters predefinidos, veremos que se ha procesado una tarea combine, y que las tareas map han tardado mas y las reduce menos con respecto al wordcount sin combiners. Podemos revisar el fichero de salida y el resultado es el mismo, de hecho los bytes de salida y de entrada son los mismos:

```
[cloudera@quickstart target]$ hadoop fs -tail hdfs:///user/cloudera/training/output/wordcountcombiner/part-r-00000
1
writ    104
write   107
writer  2
writers 7
writes  23
writhled 1
writing 22
writings 1
writs   1
written 33
wrong   287
wronged 25
wronger 4
wrongful 2
wrongfully      8
wronging 1
wrongly 1
wrongs  73
wrote   10
wroth   1
wrought 43
wrung   5
wry     1
wrying  1
y       4
yard    14
yards   5
yare    9
yarely  2
yarn    2
yaw     1
yawn    5
yawning 3
```

ycliped 1
ye 287
yea 72
yeanling 1
year 100
yearly 4
yearn 4
yearns 2
years 202
yeas 1
yeast 1
yell 4
yellow 30
yellowness 1
yellows 2
yells 1
yelping 4
yeoman 12
yeomen 1
yerk 1
yes 37
yest 1
yesterday 24
yesterdays 1
yesternight 12
yesty 2
yet 1283
yew 6
yield 143
yielded 25
yielder 2
yielders 1
yielding 20
yieldings 1
yields 15
yoemen 1
yoke 35
yoked 5
yokes 3
yoketh 1
yoking 1
yon 19
yond 36
yonder 60
yore 1
you 12702
young 432
younger 33
youngest 23
youngling 1
younglings 1
youngly 2
youngster 1
younker 3
your 6246
yours 258
yourself 281
yourselves 74
youth 288
youthful 32
youths 5
zanies 1
zany 1
zeal 33
zealous 6
zeals 1
zed 1
zenith 1
zephyrs 1
zir 2
zo 1
zodiac 1
zodiacs 1
zone 1
zounds 3
zwaggered 1

- PROCESAR ACCESS LOG CON CONTADORES:

  Vamos a procesar un archivo de log, con Mappers (no es necesario Reducer) y utilizando Counters para identificar el numero de imagenes.

  Creamos nuevo paquete e incluimos clases. Ejecutamos indicandole la carpeta de accesslog como entrada y vemos que aparecen nuevos counter aÃ±adidos a los predefinidos y que el archivo de salida es part-m-0000 y esta vacio puesto que al contexto no se le añade nada.

  El fichero de entrada tiene esta pinta:

```
10.38.181.147 - - [13/Nov/2011:01:52:51 -0800] "GET /images/filmpics/0000/5129/SK27_thumb.jpg HTTP/1.1" 200 42287
10.38.181.147 - - [13/Nov/2011:01:52:52 -0800] "GET /images/filmpics/0000/5133/SK32_thumb.jpg HTTP/1.1" 200 38147
10.38.181.147 - - [13/Nov/2011:01:52:51 -0800] "GET /images/filmpics/0000/5123/SK12_thumb.jpg HTTP/1.1" 200 45645
```

  Por eso se hacen dos split en el map. Uno con \ª , y otro con el espacio para quitar el GET, y luego ya coges la terminacion. O .jpg o .gif u otro.

  Luego generamos jar con MVN y ejecutamos sobre HDFS:

```
[cloudera@quickstart target]$ hadoop jar training-0.0.1-SNAPSHOT.jar accesslog.ImageCounter
hdfs:///user/cloudera/training/inputFiles/accesslog hdfs:///user/cloudera/training/output/accesslog
16/07/24 06:14:21 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
16/07/24 06:14:22 INFO input.FileInputFormat: Total input paths to process : 1
16/07/24 06:14:22 INFO mapreduce.JobSubmitter: number of splits:1
16/07/24 06:14:22 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1469299672451_0004
16/07/24 06:14:22 INFO impl.YarnClientImpl: Submitted application application_1469299672451_0004
16/07/24 06:14:22 INFO mapreduce.Job: The url to track the job:
http://quickstart.cloudera:8088/proxy/application_1469299672451_0004/
16/07/24 06:14:22 INFO mapreduce.Job: Running job: job_1469299672451_0004
16/07/24 06:14:29 INFO mapreduce.Job: Job job_1469299672451_0004 running in uber mode : false
16/07/24 06:14:29 INFO mapreduce.Job:  map 0% reduce 0%
16/07/24 06:14:35 INFO mapreduce.Job:  map 100% reduce 0%
16/07/24 06:14:35 INFO mapreduce.Job: Job job_1469299672451_0004 completed successfully
16/07/24 06:14:35 INFO mapreduce.Job: Counters: 33
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=113298
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=16779190
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=5
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=3468
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=3468
                Total vcore-seconds taken by all map tasks=3468
                Total megabyte-seconds taken by all map tasks=3551232
        Map-Reduce Framework
                Map input records=150000
                Map output records=0
                Input split bytes=156
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=56
                CPU time spent (ms)=1550
                Physical memory (bytes) snapshot=213319680
                Virtual memory (bytes) snapshot=1574699008
                Total committed heap usage (bytes)=282066944
        ImageCounter
                gif=1769
                jpg=87210
                other=61021
        File Input Format Counters
                Bytes Read=16779034
        File Output Format Counters
                Bytes Written=0
JPG   = 87210
GIF   = 1769
OTHER = 61021
```

Vemos que los counter se recuperan en el Driver y se puede trabajar con ellos e imprimir, aunque Hadoop ya los muestra con su grupo:

```
ImageCounter
        gif=1769
        jpg=87210
        other=61021
```

Por ultimo revisamos el fichero de salida que debe estar vacio:

```
[cloudera@quickstart target]$ hadoop fs -ls hdfs:///user/cloudera/training/output/accesslog/
Found 2 items
-rw-r--r--   1 cloudera cloudera          0 2016-07-24 06:14 hdfs:///user/cloudera/training/output/accesslog/_SUCCESS
-rw-r--r--   1 cloudera cloudera          0 2016-07-24 06:14 hdfs:///user/cloudera/training/output/accesslog/part-m-
00000
[cloudera@quickstart target]$ hadoop fs -text hdfs:///user/cloudera/training/output/accesslog/part-m-00000
[cloudera@quickstart target]$
```

- INDEXACION DE FICHEROS:

  Creamos un MapReduce para indexar cada palabra e identificar en que ficheros aparece.

  1. Tenemos que crear un Mapper que a cada parabra le mapee el fichero desde el que viene:

```
public void map(LongWritable key, Text value, Context context)
 throws IOException, InterruptedException {

      String line = value.toString();
      String nombreFile = null;

      for (String word : line.split("\\W+")) {
              if (word.length() > 0) {
                      nombreFile = ((FileSplit) context.getInputSplit()).getPath().getName();
                      context.write(new Text(word), new Text(nombreFile));
              }
      }
}
```

2. Para el Reducer lo intentamos con un IdentityReducer, pero no sale el resultado esperado porque el IdentitiReducer genera tantas lineas como apariciones de la palabra en cada fichero
el, file1,file1,file2,file3 → el,file1
el,file1
el,file2
el,file3

Debido a esto utilizamos un Reducer a medida:

a. Si no te importa que los ficheros se repitan puedes hacer esto:

```
@Override
public void reduce(Text key, Iterable<Text> ficheros, Context context)
throws IOException, InterruptedException {

String cad_ficheros = "";

for (Text fichero : ficheros) {
cad_ficheros += "|" + fichero.toString();
}
context.write(key, new Text(cad_ficheros));
}
```

Y tendras lineas asi:

```
A       histories|histories|histories|histories|histories|histories|histories|histories|histories|histories|histories|
histories|histories|histories|histories|histories|histories|histories|histories|histories|histories|histories|
histories|histories|histories|histories|histories|histories|histories|histories|histories|histories|histories|
histories|histories|poems|poems|poems|poems|comedies|comedies|comedies|comedies|comedies|comedies|comedies|comedies|
comedies|comedies|comedies|comedies|comedies|comedies|comedies|comedies|comedies|comedies|comedies|comedies|comedies|
comedies|comedies|comedies|comedies|comedies|comedies|comedies|comedies|comedies|comedies|comedies|comedies|comedies|
```

```
comedies|comedies|comedies|comedies|comedies|comedies|comedies|tragedies|tragedies|tragedies|tragedies|tragedies|
tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|
tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|
tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|
tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|tragedies|
tragedies|tragedies||tragedies|tragedies|tragedies|glossary|glossary|glossary|glossary|glossary|glossary|glossary|
glossary|glossary|glossary|glossary|glossary|glossary|glossary|glossary|glossary|glossary|
```

b. Si se quiere sin repeticiones la mejor opcion es utilizar un Set:

```
   @Override
  public void reduce(Text key, Iterable<Text> ficheros, Context context)
                          throws IOException, InterruptedException {

    Set<String> setFiles = new HashSet<String>();

    for (Text fichero : ficheros) {
      setFiles.add(fichero.toString());
    }

    context.write(key, new Text(setFiles.toString()));

  }
```

Lanzamos MVN y lo ejecutamos en HDFS real y revisamos la salida:

```
[cloudera@quickstart target]$ hadoop jar training-0.0.1-SNAPSHOT.jar indexfiles.IndexFilesDriver
hdfs:///user/cloudera/training/inputFiles/shakespeare hdfs:///user/cloudera/training/output/indexFiles
â€¦
â€¦
[cloudera@quickstart target]$ hadoop fs -ls hdfs:///user/cloudera/training/output/indexFiles
Found 2 items
-rw-r--r--   1 cloudera cloudera          0 2016-07-24 11:37 hdfs:///user/cloudera/training/output/indexFiles/_SUCCESS
-rw-r--r--   1 cloudera cloudera     909636 2016-07-24 11:37 hdfs:///user/cloudera/training/output/indexFiles/part-r-
00000
[cloudera@quickstart target]$ hadoop fs -tail hdfs:///user/cloudera/training/output/indexFiles/part-r-00000
gedies]
yonder   [glossary, comedies, histories, tragedies]
yore     [poems]
you      [glossary, poems, comedies, histories, tragedies]
young    [glossary, poems, comedies, histories, tragedies]
younger  [comedies, histories, tragedies]
youngest [comedies, histories, tragedies]
youngling        [poems]
younglings       [tragedies]
youngly  [poems, tragedies]
youngster        [poems]
younker  [comedies, histories]
your     [poems, comedies, histories, tragedies]
yours    [poems, comedies, histories, tragedies]
yourself [poems, comedies, histories, tragedies]
yourselves       [poems, comedies, histories, tragedies]
youth    [glossary, poems, comedies, histories, tragedies]
youthful [poems, comedies, histories, tragedies]
youths   [comedies, histories, tragedies]
zanies   [comedies]
zany     [comedies]
zeal     [comedies, histories, tragedies]
zealous  [poems, comedies, histories]
zeals    [tragedies]
zed      [tragedies]
zenith   [comedies]
zephyrs  [tragedies]
zir      [tragedies]
zo       [tragedies]
zodiac   [tragedies]
zodiacs  [comedies]
zone     [tragedies]
zounds   [histories]
zwaggered        [tragedies]
[cloudera@quickstart target]$
```

- CO OCURRENCIA DE PALABRAS:

  Lo que se va a hacer es medir cuantas veces aparece una palabra x creca de otra palabra y.
  Para eso crearemos un Mapper que vaya recogiendo las lineas, quitandoles caracteres extraños y luego splitear en palabras. Para despues generar pares con cuenta=1.

```
String line = value.toString();
line = line.replace(",", "");
line = line.replace(".", "");
line = line.replace(";", "");
line = line.replace(":", "");
line = line.replace("!", "");
line = line.replace("?", "");
line = line.replace("\\s", "");

String[] words = line.split("\\W+");

for (int i = 0; i < words.length - 1; i++) {
  if (!words[i].equals("")) {
    first.set(words[i]);
    second.set(words[i + 1]);
    textPair.set(first, second);
    context.write(textPair, one);
  }
}
```

  Hay que crear una clase TextPair, que implemente WritableComparable, puesto que luego el Shuffle la utilizara para ordenar y comparar. Importante los metodos compare y equals.

```
public class TextPair implements WritableComparable<TextPair> {
 private Text first;
 private Text second;

 public TextPair(Text first, Text second) {
 set(first, second);
 }

 public TextPair() {
 set(new Text(), new Text());
 }

 public TextPair(String first, String second) {
 set(new Text(first), new Text(second));
 }

 public Text getFirst() {
 return first;
 }

 public Text getSecond() {
 return second;
 }

 public void set(Text first, Text second) {
 this.first = first;
 this.second = second;
 }

 @Override
 public void readFields(DataInput in) throws IOException {
 first.readFields(in);
 second.readFields(in);
 }

 @Override
 public void write(DataOutput out) throws IOException {
 first.write(out);
 second.write(out);
 }

 @Override
 public String toString() {
 return "<" + first + "," + second + ">";
 }

 @Override
 public int compareTo(TextPair tp) {
 if (this.equals(tp)) {
 return 0;
 }
```

```
    return -1;
  }

  @Override
  public int hashCode() {
    return first.hashCode() * 163 + second.hashCode();
  }

  @Override
  public boolean equals(Object o) {
    if (o instanceof TextPair) {
      TextPair tp = (TextPair) o;
      return (first.equals(tp.first) && second.equals(tp.second))
          || (first.equals(tp.second) && second.equals(tp.first));
    }
    return false;
  }
}
```

Por ultimo un reducer que simplemente sumara las contabilizaciones:

```
   public void reduce(TextPair key, Iterable<IntWritable> count,
      Context context)
                        throws IOException, InterruptedException {

    int countCoocurs = 0;

    for (IntWritable coocur : count) {
      countCoocurs += coocur.get();
    }

    context.write(key, new IntWritable(countCoocurs));

  }
```

Ejecutamos con fichero pequeño sobre local y comprobamos que si compara bien, y despues lanzamos MVN y lanzamos la tarea hadoop sobre HDFS real sobre todos los ficheros de shakespeare:

```
[cloudera@quickstart target]$ hadoop jar training-0.0.1-SNAPSHOT.jar coocurrencias.CoocurrenciasCountDriver
hdfs:///user/cloudera/training/inputFiles/shakespeare hdfs:///user/cloudera/training/output/coocurrencias
16/07/24 13:05:53 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
16/07/24 13:05:53 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed.
Implement the Tool interface and execute your application with ToolRunner to remedy this.
16/07/24 13:05:53 INFO input.FileInputFormat: Total input paths to process : 5
16/07/24 13:05:53 INFO mapreduce.JobSubmitter: number of splits:5
16/07/24 13:05:54 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1469299672451_0006
16/07/24 13:05:54 INFO impl.YarnClientImpl: Submitted application application_1469299672451_0006
16/07/24 13:05:54 INFO mapreduce.Job: The url to track the job:
http://quickstart.cloudera:8088/proxy/application_1469299672451_0006/
16/07/24 13:05:54 INFO mapreduce.Job: Running job: job_1469299672451_0006
16/07/24 13:06:00 INFO mapreduce.Job: Job job_1469299672451_0006 running in uber mode : false
16/07/24 13:06:00 INFO mapreduce.Job:  map 0% reduce 0%
16/07/24 13:06:10 INFO mapreduce.Job:  map 13% reduce 0%
16/07/24 13:06:12 INFO mapreduce.Job:  map 27% reduce 0%
16/07/24 13:06:14 INFO mapreduce.Job:  map 80% reduce 0%
16/07/24 13:06:16 INFO mapreduce.Job:  map 87% reduce 0%
16/07/24 13:06:18 INFO mapreduce.Job:  map 100% reduce 0%
16/07/24 13:06:23 INFO mapreduce.Job:  map 100% reduce 100%
16/07/24 13:06:24 INFO mapreduce.Job: Job job_1469299672451_0006 completed successfully
16/07/24 13:06:24 INFO mapreduce.Job: Counters: 49
        File System Counters
                FILE: Number of bytes read=13577019
                FILE: Number of bytes written=27834959
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=5343961
                HDFS: Number of bytes written=11840823
                HDFS: Number of read operations=18
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
        Job Counters
                Launched map tasks=5
                Launched reduce tasks=1
                Data-local map tasks=5
                Total time spent by all maps in occupied slots (ms)=62791
                Total time spent by all reduces in occupied slots (ms)=6817
                Total time spent by all map tasks (ms)=62791
```

```
                Total time spent by all reduce tasks (ms)=6817
                Total vcore-seconds taken by all map tasks=62791
                Total vcore-seconds taken by all reduce tasks=6817
                Total megabyte-seconds taken by all map tasks=64297984
                Total megabyte-seconds taken by all reduce tasks=6980608
        Map-Reduce Framework
                Map input records=175558
                Map output records=846688
                Map output bytes=11883637
                Map output materialized bytes=13577043
                Input split bytes=754
                Combine input records=0
                Combine output records=0
                Reduce input groups=843251
                Reduce shuffle bytes=13577043
                Reduce input records=846688
                Reduce output records=843251
                Spilled Records=1693376
                Shuffled Maps =5
                Failed Shuffles=0
                Merged Map outputs=5
                GC time elapsed (ms)=1492
                CPU time spent (ms)=37330
                Physical memory (bytes) snapshot=1957122048
                Virtual memory (bytes) snapshot=9396256768
                Total committed heap usage (bytes)=2008023040
        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=5343207
        File Output Format Counters
                Bytes Written=11840823
```

Una pequeña traza del fichero podria ser:

```
[cloudera@quickstart target]$ hadoop fs -tail hdfs:///user/cloudera/training/output/coocurrencias/part-r-00000
US,And>  2
<my,absence>    2
<endure,my>     1
<Cannot,endure>  1
<in,an>  15
<new,a>  7
<brings,forth>   2
<smock,brings>   1
<your,old>      2
<consolation,your>      1
<with,consolation>      1
<grief,is>      2
<this,grief>    1
<be,lamented>   2
<case,to>       2
<If,there>      14
<new,If> 1
<take,the>      28
<in,her> 72
<should,be>     88
<for,nothing>   9
<pity,to>       5
<it,die> 2
<women,die>     1
<let,women>     1
<occasion,let>   1
<compelling,occasion>   1
<to,thee>       126
<Importeth,thee> 1
<more,serious>   1
<Second,Messenger>      10
<upon,your>     30
<stays,upon>    1
<He,stays>      1
<Second,Attendant>      2
<to,and> 94
<Lydia,and>     1
<To,Lydia>      1
<from,Syria>    1
<shook,from>    1
<My,lord>       122
<in,the> 494
```

<Enter,DOMITIUS> 5
<him,your> 2
<Show,him> 1
<ALEXAS,Show> 1
<book,of> 6
<infinite,book> 1
<s,infinite> 1
<In,nature> 2
<any,thing> 43
<most,any> 1
<Alexas,most> 1
<sweet,Alexas> 2
<go,with> 44
<Which,still> 3
<of,the> 444
<musters,of> 1
<A,room> 29
<I,Alexandria> 2
<parts,of> 8
<In,several> 2
<SCENE,In> 1
<wife,to> 19
<queen,of> 7
<Clown,A> 3
<A,Soothsayer> 2
<ANTONY,AND> 43
<my,diseases> 1