

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Seminario de Sistemas 2 Sección A

Primer Semestre 2023



Nombre: William Alejandro Borrayo Alarcón

Carné: 201909103

## Contenido

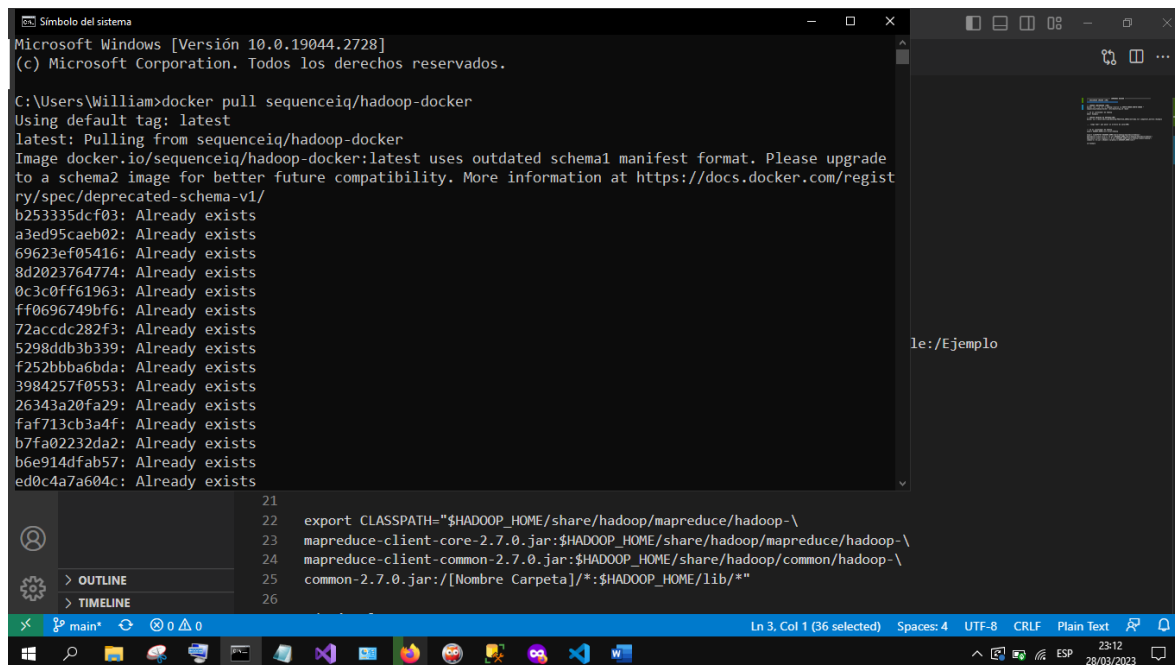
Procedimiento realizado .....	2
Preparación de Hadoop en contenedor .....	2
Procedimiento para el conteo de palabras en el archivo Correos.txt .....	7
Procedimiento para el conteo de palabras en el archivo Punteos.txt .....	14
Archivo de comandos utilizados .....	20
Análisis acerca de los resultados de cada archivo .....	20
Conclusiones acerca de los resultados de cada archivo .....	22
Conclusiones acerca del uso de Hadoop en BigData .....	22
Bibliografía .....	22

## Procedimiento realizado

Para realizar este procedimiento es necesario tener Docker instalado.

### Preparación de Hadoop en contenedor

1. Descarga de imagen de Hadoop: Si vemos el mensaje 'Already exists' es porque ya la tenemos descargada.



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19044.2728]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\William>docker pull sequenceiq/hadoop-docker
Using default tag: latest
latest: Pulling from sequenceiq/hadoop-docker
Image docker.io/sequenceiq/hadoop-docker:latest uses outdated schema1 manifest format. Please upgrade
to a schema2 image for better future compatibility. More information at https://docs.docker.com/regist
ry/spec/deprecated-schema-v1/
b253335dcf03: Already exists
a3ed95caeb02: Already exists
69623ef05416: Already exists
8d2023764774: Already exists
0c3c0ff61963: Already exists
ff0696749bf6: Already exists
72accdc282f3: Already exists
5298ddb3b339: Already exists
f252bbb6bda: Already exists
3984257f0553: Already exists
26343a20fa29: Already exists
faf713cb3a4f: Already exists
b7fa02232da2: Already exists
b6e914dfab57: Already exists
ed0c4a7a604c: Already exists
```

2. Creación del contenedor (ejecución de la imagen descargada): En el comando se utiliza un parámetro para darle un nombre específico al contenedor: 'hadoop'. Si se creó correctamente veremos la consola de esta manera:

```
Símbolo del sistema - docker run -it --name hadoop -v hadoop:/source -p 50070-50080:50070-50080 sequenceiq/hadoop-docker /etc/bootstrap.sh -bash
C:\Users\William>docker run -it --name hadoop -v hadoop:/source -p 50070-50080:50070-50080 ^
¿Más? sequenceiq/hadoop-docker /etc/bootstrap.sh -bash
/
Starting sshd: [ OK ]
Starting namenodes on [5bccd9e2bf2b]
5bccd9e2bf2b: starting namenode, logging to /usr/local/hadoop/logs/hadoop-root-namenode-5bccd9e2bf2b.out
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-root-datanode-5bccd9e2bf2b.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-root-secondarynamenode-5bccd9e2bf2b.out
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn--resourcemanager-5bccd9e2bf2b.out
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root-nodemanager-5bccd9e2bf2b.out
bash-4.1#
```

También en localhost:50070 veremos esto:

Namenode information

localhost:50070/dfshealth.html#tab-overview

110%

Hadoop

Overview

Datanodes

Datanode Volume Failures

Snapshot

Startup Progress

Utilities

## Overview '5bccd9e2bf2b:9000' (active)

Started:	Wed Mar 29 01:19:32 EDT 2023
Version:	2.7.0, rd4c8d4d4d203c934e8074b31289a28724c0842cf
Compiled:	2015-04-10T18:40Z by jenkins from (detached from d4c8d4d)
Cluster ID:	CID-f371ad4-5145-47ad-ba01-959bc79aecbe
Block Pool ID:	BP-581371184-172.17.13.14-1437578119536

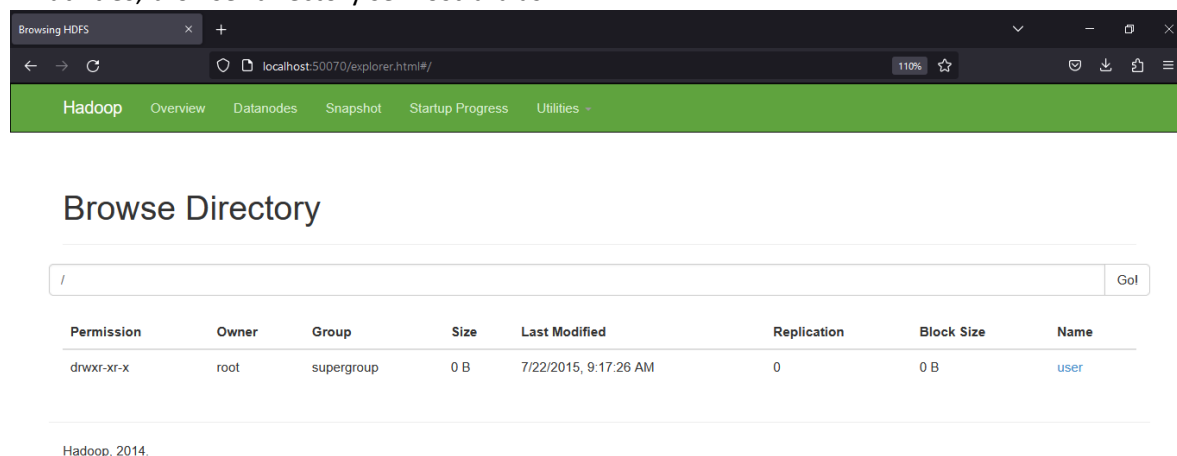
## Summary

Security is off.

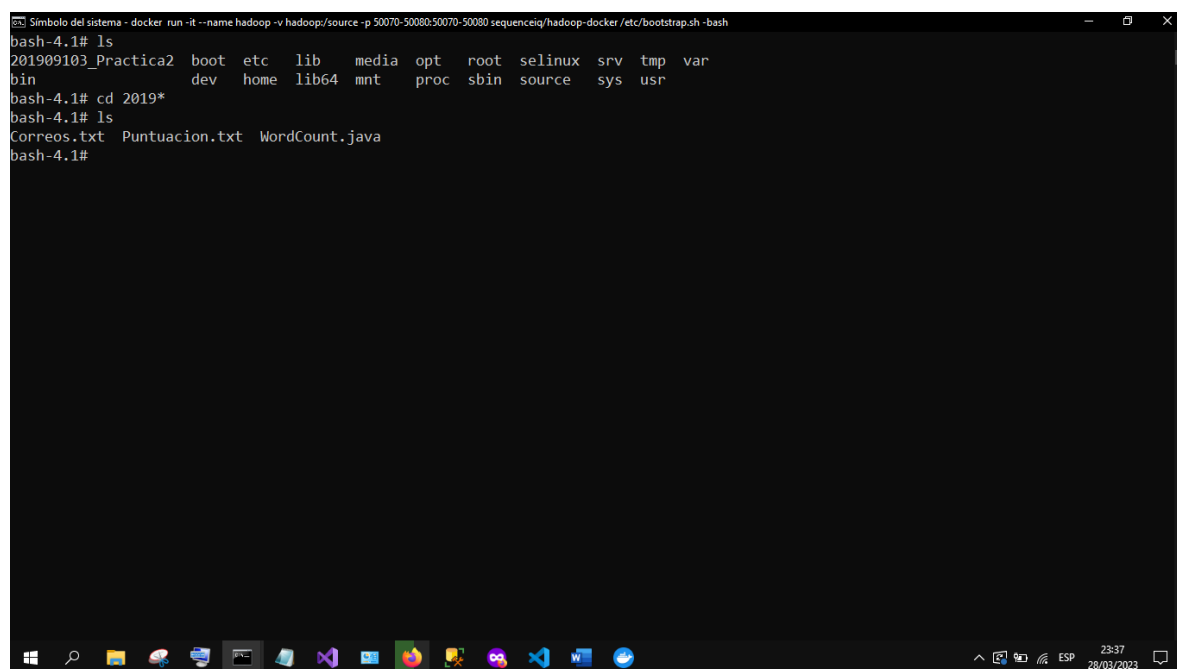
Safemode is off.

35 files and directories, 31 blocks = 66 total filesystem object(s).

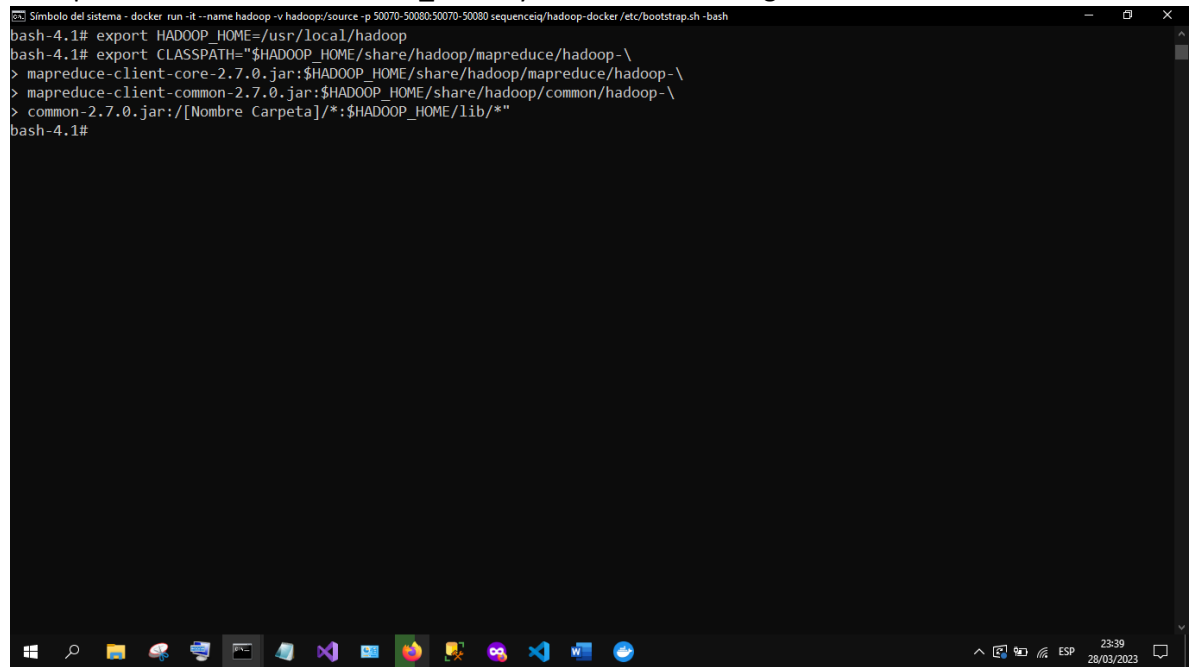
En utilities, browser directory se mostrará así:



3. Luego de esto se crea una carpeta con el nombre '201909103\_Practica2' en el contenedor, aquí se guardarán algunos archivos necesarios (archivos de entrada y el WordCount.java). Para la copia de los archivos se utiliza el comando: `docker cp <Ruta en computadora> hadoop:/201909103_Practica2` en una consola aparte.

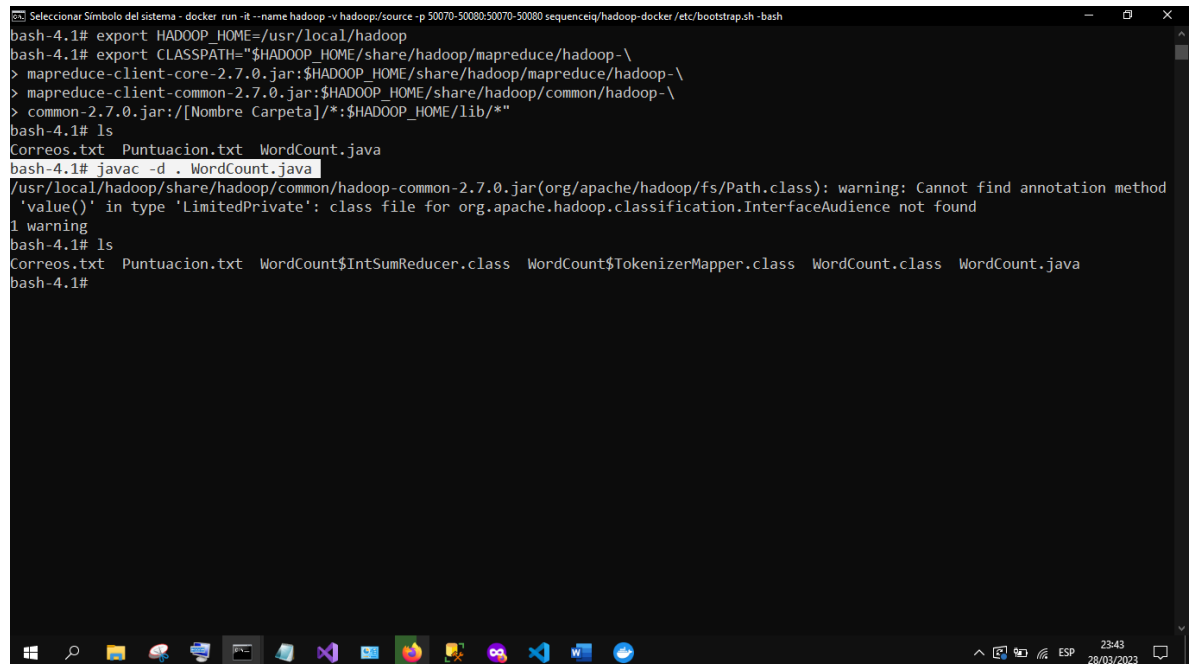


4. Se exportan las variables HADOOP\_HOME y CLASSPATH de la siguiente forma:

A terminal window titled 'Simbolo del sistema - docker run -it --name hadoop -v hadoop:/source -p 50070-50080:50070-50080 sequenceiq/hadoop-docker /etc/bootstrap.sh -bash'. The terminal shows the following commands and output:

```
bash-4.1# export HADOOP_HOME=/usr/local/hadoop
bash-4.1# export CLASSPATH="$HADOOP_HOME/share/hadoop/mapreduce/hadoop-
> mapreduce-client-core-2.7.0.jar:$HADOOP_HOME/share/hadoop/mapreduce/hadoop-
> mapreduce-client-common-2.7.0.jar:$HADOOP_HOME/share/hadoop/common/hadoop-
> common-2.7.0.jar:[Nombre Carpeta]/*:$HADOOP_HOME/lib/*"
bash-4.1#
```

5. Ejecutamos el archivo .java, si se ejecuta bien deberá crear 3 archivos .class como se muestra en la imagen, la advertencia que se muestra no representa mayor problema:

A terminal window titled 'Seleccionar Símbolo del sistema - docker run -it --name hadoop -v hadoop:/source -p 50070-50080:50070-50080 sequenceiq/hadoop-docker /etc/bootstrap.sh -bash'. The terminal shows the following commands and output:

```
bash-4.1# export CLASSPATH="$HADOOP_HOME/share/hadoop/mapreduce/hadoop-
> mapreduce-client-core-2.7.0.jar:$HADOOP_HOME/share/hadoop/mapreduce/hadoop-
> mapreduce-client-common-2.7.0.jar:$HADOOP_HOME/share/hadoop/common/hadoop-
> common-2.7.0.jar:[Nombre Carpeta]/*:$HADOOP_HOME/lib/*"
bash-4.1# ls
Correos.txt Puntuacion.txt WordCount.java
bash-4.1# javac -d . WordCount.java
/usr/local/hadoop/share/hadoop/common/hadoop-common-2.7.0.jar(org/apache/hadoop/fs/Path.class): warning: Cannot find annotation method
'value()' in type 'LimitedPrivate': class file for org.apache.hadoop.classification.InterfaceAudience not found
1 warning
bash-4.1# ls
Correos.txt Puntuacion.txt WordCount$IntSumReducer.class WordCount$TokenizerMapper.class WordCount.class WordCount.java
bash-4.1#
```

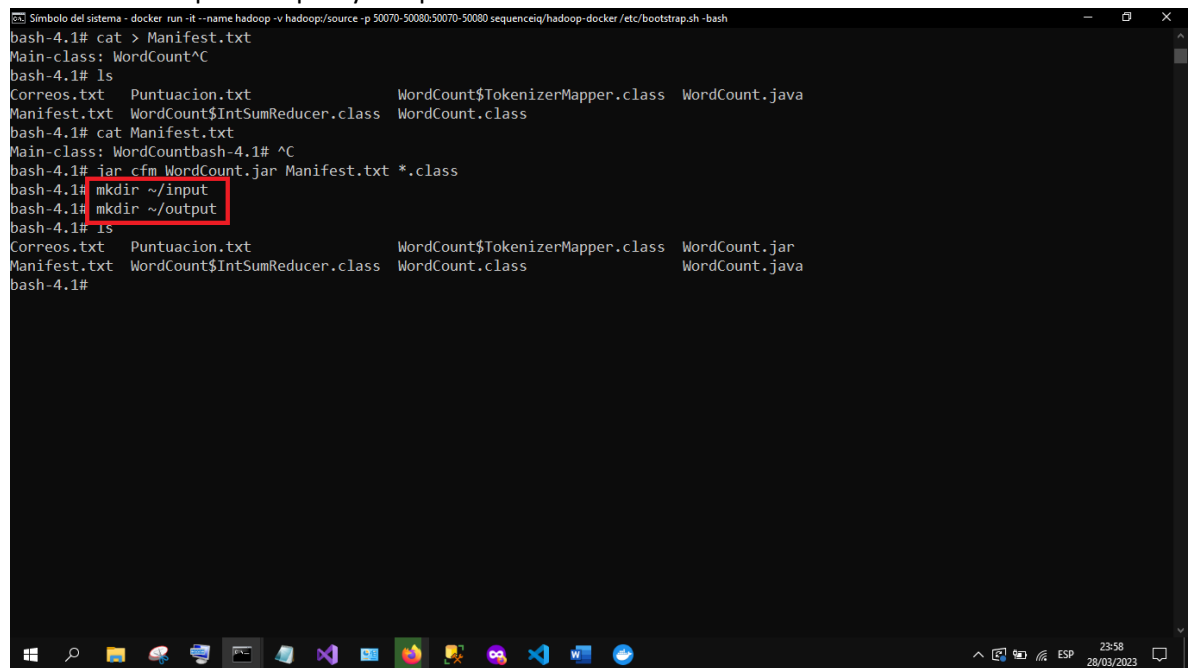
6. Se crea el archivo Manifest.txt con el contenido “Main-class: WordCount”. Se puede utilizar cat y luego de escribir el contenido la combinación Ctrl + D.

```
Simbolo del sistema - docker run -it --name hadoop -v hadoop:/source -p 50070-50080:50070-50080 sequenceiq/hadoop-docker /etc/bootstrap.sh -bash
bash-4.1# cat > Manifest.txt
Main-class: WordCount^C
bash-4.1# ls
Correos.txt  Puntuacion.txt  WordCount$TokenizerMapper.class  WordCount.java
Manifest.txt  WordCount$IntSumReducer.class  WordCount.class
bash-4.1# cat Manifest.txt
Main-class: WordCountbash-4.1# ^C
bash-4.1#
```

7. Se debe crear un archivo .jar utilizando el archivo Manifest.txt previamente creado:

```
Simbolo del sistema - docker run -it --name hadoop -v hadoop:/source -p 50070-50080:50070-50080 sequenceiq/hadoop-docker /etc/bootstrap.sh -bash
bash-4.1# cat > Manifest.txt
Main-class: WordCount^C
bash-4.1# ls
Correos.txt  Puntuacion.txt  WordCount$TokenizerMapper.class  WordCount.java
Manifest.txt  WordCount$IntSumReducer.class  WordCount.class
bash-4.1# cat Manifest.txt
Main-class: WordCountbash-4.1# ^C
bash-4.1# jar cfm WordCount.jar Manifest.txt *.class
bash-4.1#
```

8. Se crean las carpetas 'input' y 'output' en el home del usuario root:

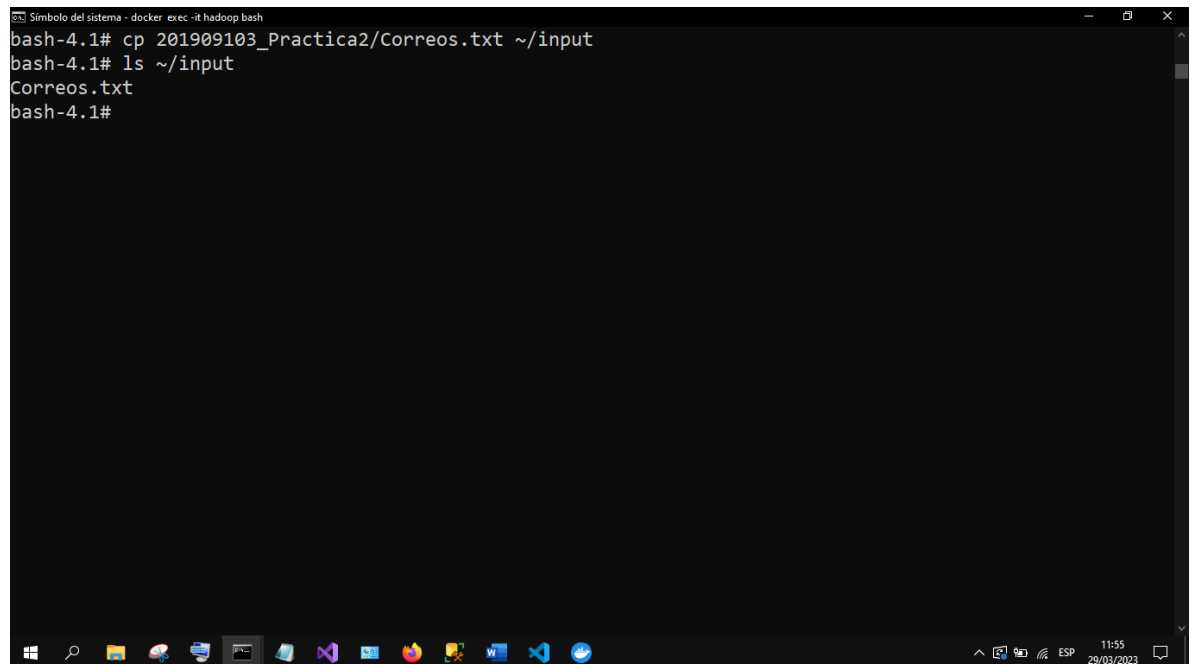
A terminal window titled 'Simbolo del sistema - docker run -it --name hadoop -v hadoop/source -p 50070-50080:50070-50080 sequenceiq/hadoop-docker /etc/bootstrap.sh -bash'. The terminal shows the following commands and output:

```
bash-4.1# cat > Manifest.txt
Main-class: WordCount^C
bash-4.1# ls
Correos.txt  Puntuacion.txt          WordCount$TokenizerMapper.class  WordCount.java
Manifest.txt  WordCount$IntSumReducer.class  WordCount.class
bash-4.1# cat Manifest.txt
Main-class: WordCount
bash-4.1# jar cfm WordCount.jar Manifest.txt *.class
bash-4.1# mkdir ~/input
bash-4.1# mkdir ~/output
bash-4.1# ls
Correos.txt  Puntuacion.txt          WordCount$TokenizerMapper.class  WordCount.jar
Manifest.txt  WordCount$IntSumReducer.class  WordCount.class                  WordCount.java
bash-4.1#
```

The 'mkdir ~/input' and 'mkdir ~/output' commands are highlighted with a red box.

### Procedimiento para el conteo de palabras en el archivo Correos.txt

1. Se copia el archivo de entrada hacia la carpeta 'input' creada en el paso anterior:

A terminal window titled 'Simbolo del sistema - docker exec -it hadoop bash'. The terminal shows the following commands and output:

```
bash-4.1# cp 201909103_Practica2/Correos.txt ~/input
bash-4.1# ls ~/input
Correos.txt
bash-4.1#
```

2. Se copia el archivo de entrada hacia el sistema de archivos de hadoop. Y se verifica la correcta copia del archivo:

```
Seleccionar Símbolo del sistema - docker exec -it hadoop bash
Correos.txt
bash-4.1# ${HADOOP_HOME}/bin/hdfs dfs -copyFromLocal ~/input /
bash-4.1# ${HADOOP_HOME}/bin/hdfs dfs -ls /input
Found 1 items
-rw-r--r-- 1 root supergroup 31354 2023-03-29 13:55 /input/Correos.txt
bash-4.1# cd 201909103_Practica2
bash-4.1# ${HADOOP_HOME}/bin/hadoop jar WordCount.jar WordCount /input/Correos.txt /output
23/03/29 13:57:30 INFO client.RMPProxy: Connecting to ResourceManager at /0.0.0.0:8032
23/03/29 13:57:31 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed.
Implement the Tool interface and execute your application with ToolRunner to remedy this.
23/03/29 13:57:31 INFO input.FileInputFormat: Total input paths to process : 1
23/03/29 13:57:31 INFO mapreduce.JobSubmitter: number of splits:1
23/03/29 13:57:31 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1680108686539_0002
23/03/29 13:57:32 INFO impl.YarnClientImpl: Submitted application application_1680108686539_0002
23/03/29 13:57:32 INFO mapreduce.Job: The url to track the job: http://5bcd9e2bf2b:8088/proxy/application_1680108686539_0002/
23/03/29 13:57:32 INFO mapreduce.Job: Running job: job_1680108686539_0002
23/03/29 13:57:38 INFO mapreduce.Job: Job job_1680108686539_0002 running in uber mode : false
23/03/29 13:57:38 INFO mapreduce.Job: map 0% reduce 0%
23/03/29 13:57:43 INFO mapreduce.Job: map 100% reduce 0%
23/03/29 13:57:49 INFO mapreduce.Job: map 100% reduce 100%
23/03/29 13:57:49 INFO mapreduce.Job: Job job_1680108686539_0002 completed successfully
23/03/29 13:57:49 INFO mapreduce.Job: Counters: 49
File System Counters
FILE: Number of bytes read=26259
```

3. Para iniciar con el conteo de palabras es necesario estar en la carpeta que tiene el archivo WordCount.java (201909103\_Practica2). Desde ahí se ejecuta el comando: `${HADOOP_HOME}/bin/hadoop jar WordCount.jar WordCount /input/Correos.txt /output`:

```
Seleccionar Símbolo del sistema - docker exec -it hadoop bash
Correos.txt
bash-4.1# ${HADOOP_HOME}/bin/hdfs dfs -copyFromLocal ~/input /
bash-4.1# ${HADOOP_HOME}/bin/hdfs dfs -ls /input
Found 1 items
-rw-r--r-- 1 root supergroup 31354 2023-03-29 13:55 /input/Correos.txt
bash-4.1# cd 201909103_Practica2
bash-4.1# ${HADOOP_HOME}/bin/hadoop jar WordCount.jar WordCount /input/Correos.txt /output
23/03/29 13:57:30 INFO client.RMPProxy: Connecting to ResourceManager at /0.0.0.0:8032
23/03/29 13:57:31 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed.
Implement the Tool interface and execute your application with ToolRunner to remedy this.
23/03/29 13:57:31 INFO input.FileInputFormat: Total input paths to process : 1
23/03/29 13:57:31 INFO mapreduce.JobSubmitter: number of splits:1
23/03/29 13:57:31 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1680108686539_0002
23/03/29 13:57:32 INFO impl.YarnClientImpl: Submitted application application_1680108686539_0002
23/03/29 13:57:32 INFO mapreduce.Job: The url to track the job: http://5bcd9e2bf2b:8088/proxy/application_1680108686539_0002/
23/03/29 13:57:32 INFO mapreduce.Job: Running job: job_1680108686539_0002
23/03/29 13:57:38 INFO mapreduce.Job: Job job_1680108686539_0002 running in uber mode : false
23/03/29 13:57:38 INFO mapreduce.Job: map 0% reduce 0%
23/03/29 13:57:43 INFO mapreduce.Job: map 100% reduce 0%
23/03/29 13:57:49 INFO mapreduce.Job: map 100% reduce 100%
23/03/29 13:57:49 INFO mapreduce.Job: Job job_1680108686539_0002 completed successfully
23/03/29 13:57:49 INFO mapreduce.Job: Counters: 49
File System Counters
FILE: Number of bytes read=26259
```



Se iniciará el proceso de conteo, al final veremos detalles como la cantidad de bytes que fueron escritos, leídos, si hubieron errores u otras cosas:

```
Simbolo del sistema - docker exec -it hadoop bash
Reduce input records=1923
Reduce output records=1923
Spilled Records=3846
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=44
CPU time spent (ms)=1880
Physical memory (bytes) snapshot=423301120
Virtual memory (bytes) snapshot=1497374720
Total committed heap usage (bytes)=394264576

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=31354
File Output Format Counters
Bytes Written=18624
bash-4.1# clear
bash-4.1# cd ..
bash-4.1#
```

4. Podremos ver los archivos que fueron generados con el primer comando, el segundo hará que se muestre el contenido del archivo de resultado:

```
Simbolo del sistema - docker exec -it hadoop bash
bash-4.1# ${HADOOP_HOME}/bin/hdfs dfs -ls /output
Found 2 items
-rw-r--r--  1 root supergroup          0 2023-03-29 13:57 /output/_SUCCESS
-rw-r--r--  1 root supergroup    18624 2023-03-29 13:57 /output/part-r-00000
bash-4.1# ${HADOOP_HOME}/bin/hdfs dfs -cat /output/part-r-00000
1      5
1,000  1
1/2    1
10     3
100-   1
10am   2
10am,  1
10th   1
11     1
12     2
12/1a  1
125    1
12th   1
150    1
175.   1
17th   2
18     1
18-19  1
1970   1
1:30   1
```

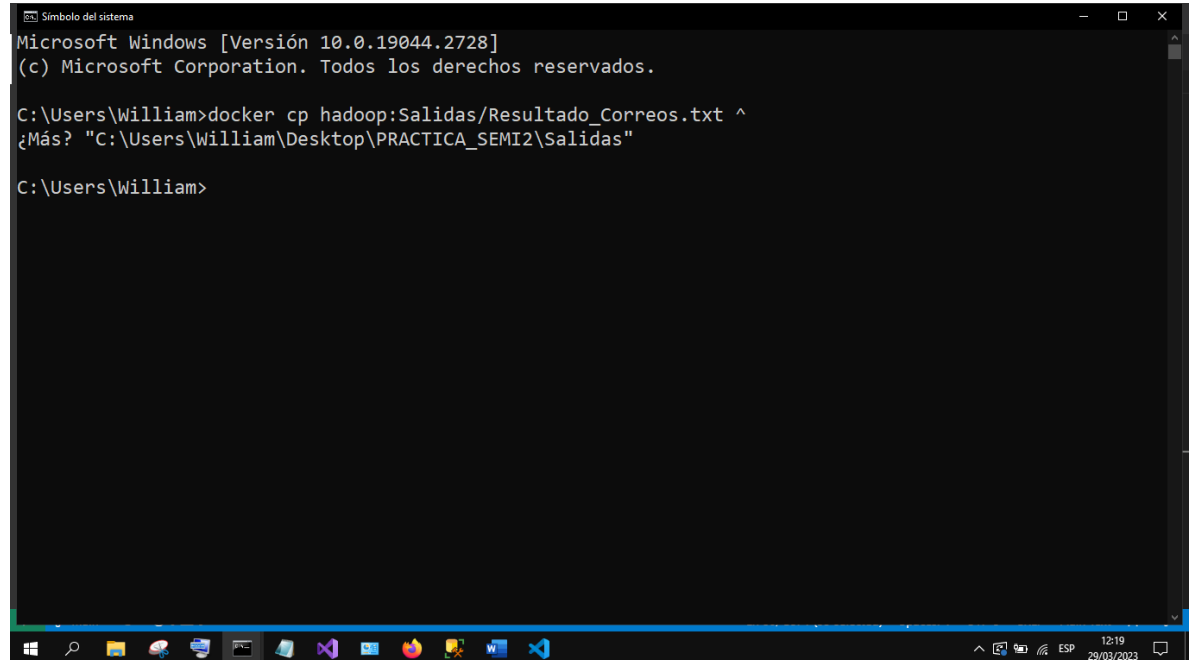
5. Renombramos el archivo a 'Resultado\_Correos.txt' y verificamos que tenga el mismo contenido:

```
Símbolo del sistema - docker exec -it hadoop bash
bash-4.1# clear
bash-4.1# ${HADOOP_HOME}/bin/hdfs dfs -mv /output/part-r-00000 /output/Resultado_Correos.txt
bash-4.1# ${HADOOP_HOME}/bin/hdfs dfs -cat /output/Resultado_Correos.txt
1          5
1,000      1
1/2        1
10         3
100-       1
10am       2
10am,      1
10th       1
11         1
12         2
12/1a      1
125        1
12th       1
150        1
175.       1
17th       2
18         1
18-19      1
1970       1
1:30       1
1st        2
1st,       1
```

6. Copiamos el archivo de resultado a la carpeta output del home del usuario root, luego hacia la carpeta Salidas (es necesario crearla en el directorio inicial del contenedor), para su fácil extracción del contenedor.

```
Símbolo del sistema - docker exec -it hadoop bash
bash-4.1# ${HADOOP_HOME}/bin/hdfs dfs -copyToLocal /output/Resultado_Correos.txt ~/output
23/03/29 14:18:43 WARN hdfs.DFSCClient: DFSInputStream has been closed already
bash-4.1# cp ~/output/Resultado_Correos.txt /Salidas
bash-4.1# ls Salidas
Resultado_Correos.txt
bash-4.1# docker cp hadoop:Salidas/Resultado_Correos.txt ^
bash: docker: command not found
bash-4.1# "C:\Users\William\Desktop\PRACTICA_SEMI2\Salidas"^C
bash-4.1#
```

7. Finalmente pasamos el archivo de resultado desde el contenedor hacia nuestra computadora, esto desde una nueva consola y hacia el directorio de nuestra preferencia.

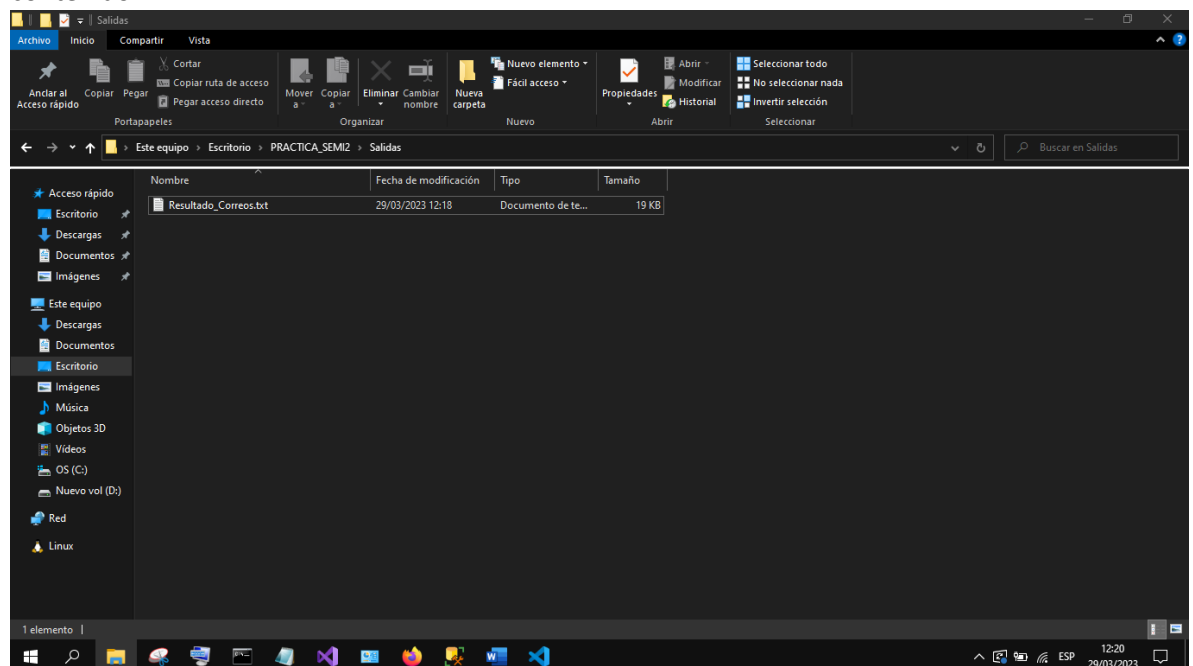


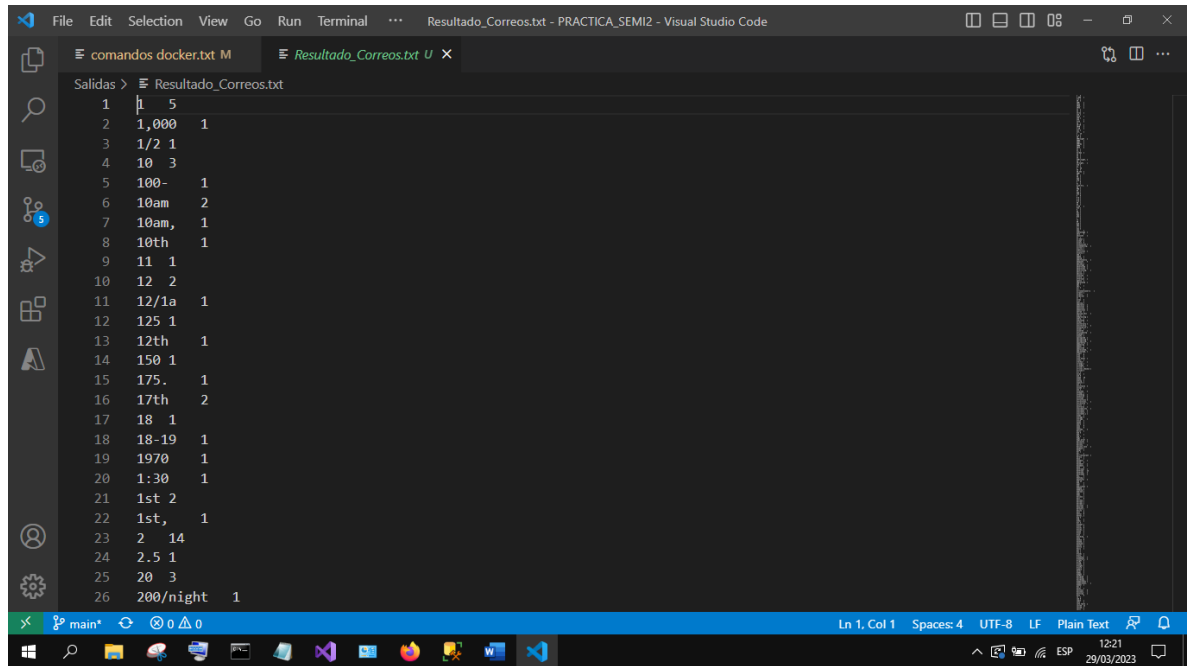
```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19044.2728]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\William>docker cp hadoop:Salidas/Resultado_Correos.txt ^
¿Más? "C:\Users\William\Desktop\PRACTICA_SEMI2\Salidas"

C:\Users\William>
```

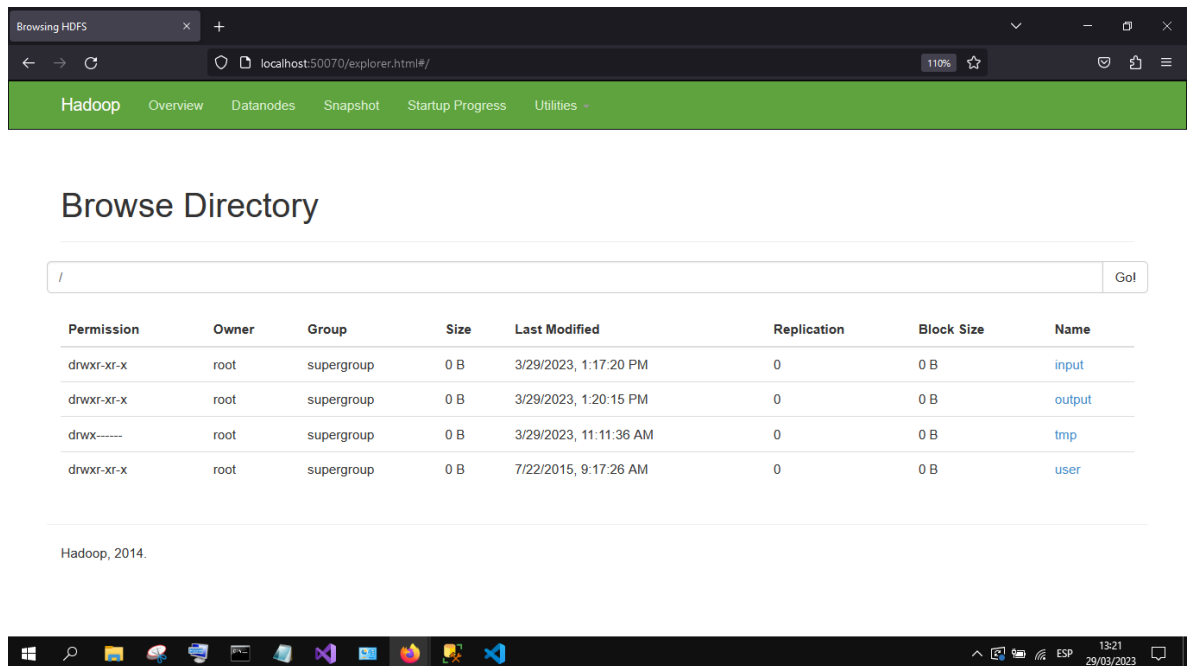
Podremos comprobar que el archivo ya está en nuestra computadora y con el mismo contenido:





```
1 | 5
2 1,000 1
3 1/2 1
4 10 3
5 100- 1
6 10am 2
7 10am, 1
8 10th 1
9 11 1
10 12 2
11 12/1a 1
12 125 1
13 12th 1
14 150 1
15 175. 1
16 17th 2
17 18 1
18 18-19 1
19 1970 1
20 1:30 1
21 1st 2
22 1st, 1
23 2 14
24 2.5 1
25 20 3
26 200/night 1
```

8. Desde el navegador en localhost:50070, utilities, browse directory se pueden ver los archivos de esta manera:



Browsing HDFS

localhost:50070/explorer.html#/

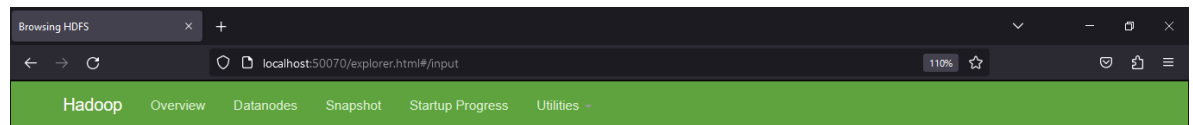
Hadoop Overview Datanodes Snapshot Startup Progress Utilities

### Browse Directory

/ Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	root	supergroup	0 B	3/29/2023, 1:17:20 PM	0	0 B	<a href="#">input</a>
drwxr-xr-x	root	supergroup	0 B	3/29/2023, 1:20:15 PM	0	0 B	<a href="#">output</a>
drwx-----	root	supergroup	0 B	3/29/2023, 11:11:36 AM	0	0 B	<a href="#">tmp</a>
drwxr-xr-x	root	supergroup	0 B	7/22/2015, 9:17:26 AM	0	0 B	<a href="#">user</a>

Hadoop, 2014.

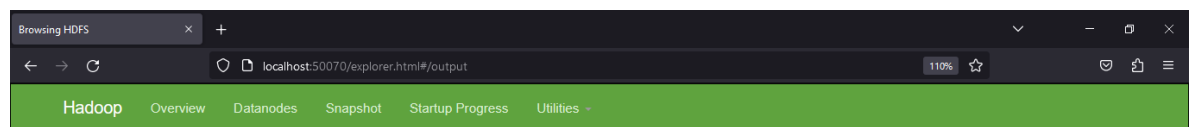


## Browse Directory

/input Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	root	supergroup	30.62 KB	3/29/2023, 1:17:20 PM	1	128 MB	<a href="#">Correos.txt</a>

Hadoop, 2014.



## Browse Directory

/output Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	root	supergroup	18.19 KB	3/29/2023, 1:17:53 PM	1	128 MB	<a href="#">Resultado_Correos.txt</a>
-rw-r--r--	root	supergroup	0 B	3/29/2023, 1:17:53 PM	1	128 MB	<a href="#">_SUCCESS</a>

Hadoop, 2014.



## Procedimiento para el conteo de palabras en el archivo Punteos.txt

1. Es necesario eliminar los archivos en las carpetas output e input, tanto del sistema de archivos de Hadoop como del home del usuario root de esta manera:

[illegible]

- Se copia el archivo de entrada hacia la carpeta 'input' en el home del usuario root y luego hacia el sistema de archivos de hadoop:

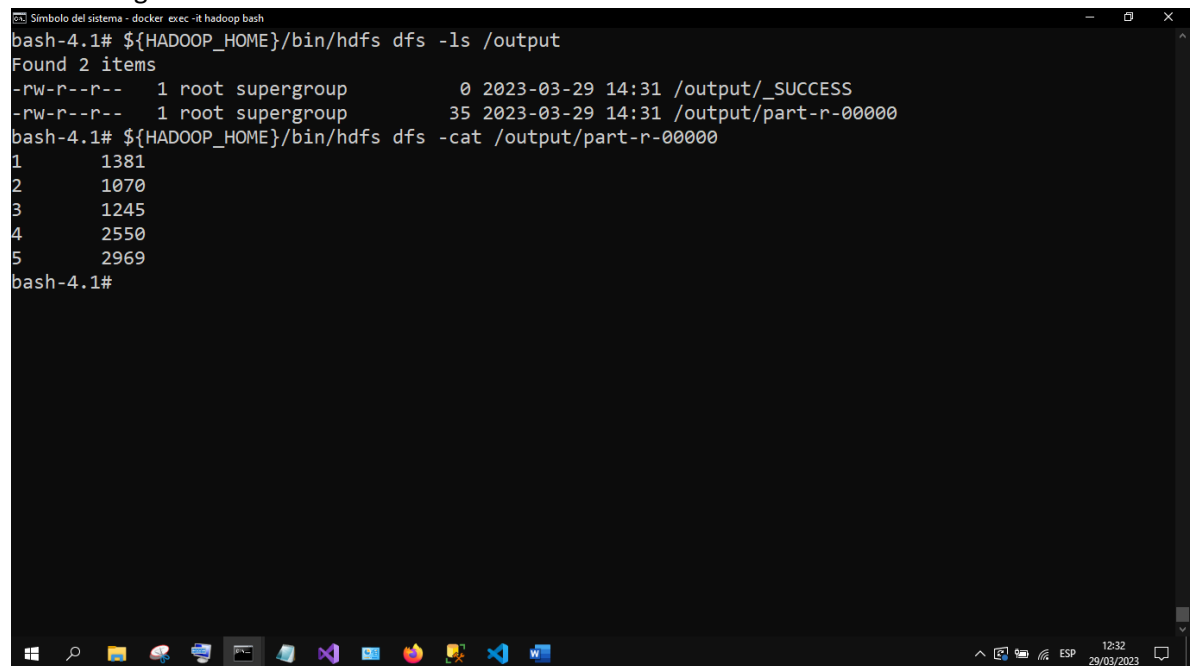
```
bash-4.1# rm ~/output/*
bash-4.1# ${HADOOP_HOME}/bin/hdfs dfs -rm -skipTrash /input/*
Deleted /input/Correos.txt
bash-4.1#
bash-4.1#
bash-4.1#
bash-4.1#
bash-4.1#
bash-4.1#
bash-4.1#
bash-4.1#
bash-4.1#
bash-4.1#
bash-4.1#
bash-4.1#
bash-4.1#
bash-4.1#
bash-4.1#
bash-4.1#
bash-4.1# cp 201909103_Practica2/Puntuacion.txt ~/input
bash-4.1# ${HADOOP_HOME}/bin/hdfs dfs -copyFromLocal ~/input /
bash-4.1# ${HADOOP_HOME}/bin/hdfs dfs -ls /input
Found 1 items
-rw-r--r-- 1 root supergroup 18429 2023-03-29 14:28 /input/Puntuacion.txt
```

3. Se inicia el conteo de palabras de la misma manera que en el archivo anterior:

```
Símbolo del sistema - docker exec -it hadoop bash
bash-4.1# clear
bash-4.1# cd 201909103_Practica2
bash-4.1# ${HADOOP_HOME}/bin/hadoop jar WordCount.jar WordCount /input/Puntuacion.txt /output
23/03/29 14:30:42 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
23/03/29 14:30:43 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed.
Implement the Tool interface and execute your application with ToolRunner to remedy this.
23/03/29 14:30:43 INFO input.FileInputFormat: Total input paths to process : 1
23/03/29 14:30:43 INFO mapreduce.JobSubmitter: number of splits:1
23/03/29 14:30:44 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1680108686539_0003
23/03/29 14:30:44 INFO impl.VarnClientImpl: Submitted application application_1680108686539_0003
23/03/29 14:30:44 INFO mapreduce.Job: The url to track the job: http://5bcd9e2bf2b:8088/proxy/application_1680108686539_0003/
23/03/29 14:30:44 INFO mapreduce.Job: Running job: job_1680108686539_0003
23/03/29 14:30:50 INFO mapreduce.Job: Job job_1680108686539_0003 running in uber mode : false
23/03/29 14:30:50 INFO mapreduce.Job: map 0% reduce 0%
23/03/29 14:30:55 INFO mapreduce.Job: map 100% reduce 0%
23/03/29 14:31:01 INFO mapreduce.Job: map 100% reduce 100%
23/03/29 14:31:01 INFO mapreduce.Job: Job job_1680108686539_0003 completed successfully
23/03/29 14:31:01 INFO mapreduce.Job: Counters: 49
File System Counters
  FILE: Number of bytes read=46
  FILE: Number of bytes written=229879
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
```

```
Símbolo del sistema - docker exec -it hadoop bash
Reduce shuffle bytes=46
Reduce input records=5
Reduce output records=5
Spilled Records=10
Shuffled Maps =1
Failed Shuffles=0
Merged Map outputs=1
GC time elapsed (ms)=63
CPU time spent (ms)=1980
Physical memory (bytes) snapshot=426500096
Virtual memory (bytes) snapshot=1505546240
Total committed heap usage (bytes)=394264576
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=18429
File Output Format Counters
  Bytes Written=35
bash-4.1# cd ..
bash-4.1#
```

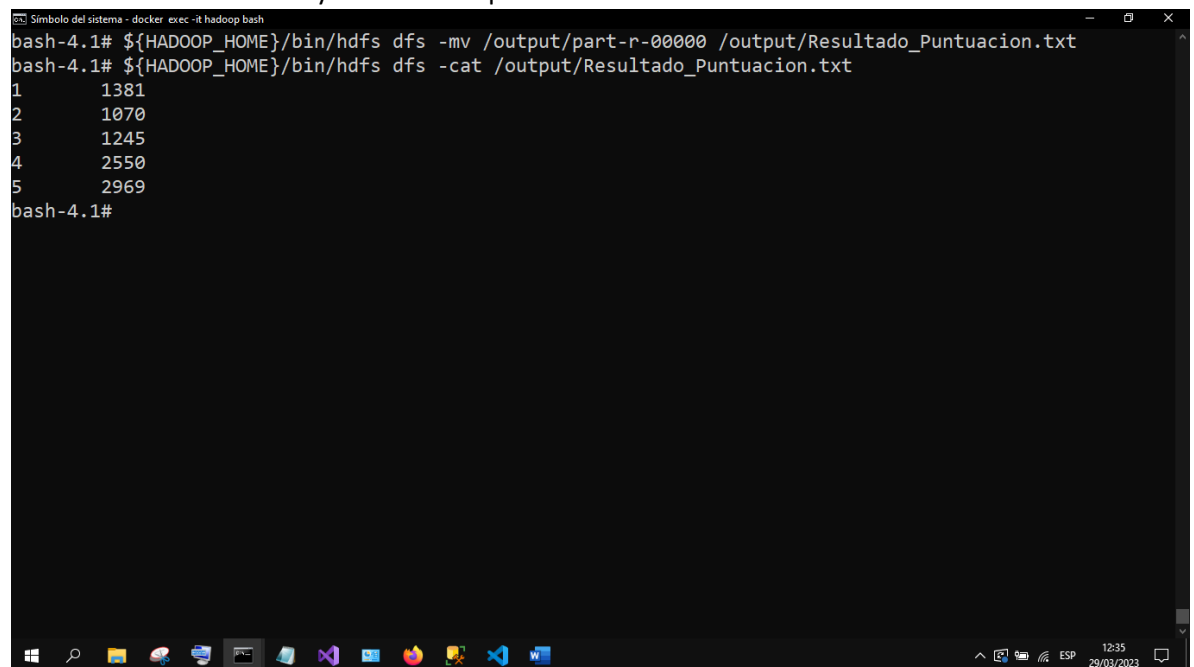
4. El archivo generado como resultado se verá de esta manera:



The screenshot shows a terminal window titled "Símbolo del sistema - docker exec -it hadoop bash". The user runs the command `bash-4.1# ${HADOOP_HOME}/bin/hdfs dfs -ls /output`, which returns "Found 2 items" and a list of files: `-rw-r--r-- 1 root supergroup 0 2023-03-29 14:31 /output/_SUCCESS` and `-rw-r--r-- 1 root supergroup 35 2023-03-29 14:31 /output/part-r-00000`. Then, the user runs `bash-4.1# ${HADOOP_HOME}/bin/hdfs dfs -cat /output/part-r-00000`, which outputs the numbers 1381, 1070, 1245, 2550, and 2969 on separate lines. The terminal window has a Windows taskbar at the bottom with various application icons and a system clock showing 12:32 on 29/03/2023.

```
Símbolo del sistema - docker exec -it hadoop bash
bash-4.1# ${HADOOP_HOME}/bin/hdfs dfs -ls /output
Found 2 items
-rw-r--r-- 1 root supergroup      0 2023-03-29 14:31 /output/_SUCCESS
-rw-r--r-- 1 root supergroup    35 2023-03-29 14:31 /output/part-r-00000
bash-4.1# ${HADOOP_HOME}/bin/hdfs dfs -cat /output/part-r-00000
1      1381
2      1070
3      1245
4      2550
5      2969
bash-4.1#
```

5. Renombramos el archivo y verificamos que el contenido sea el mismo:

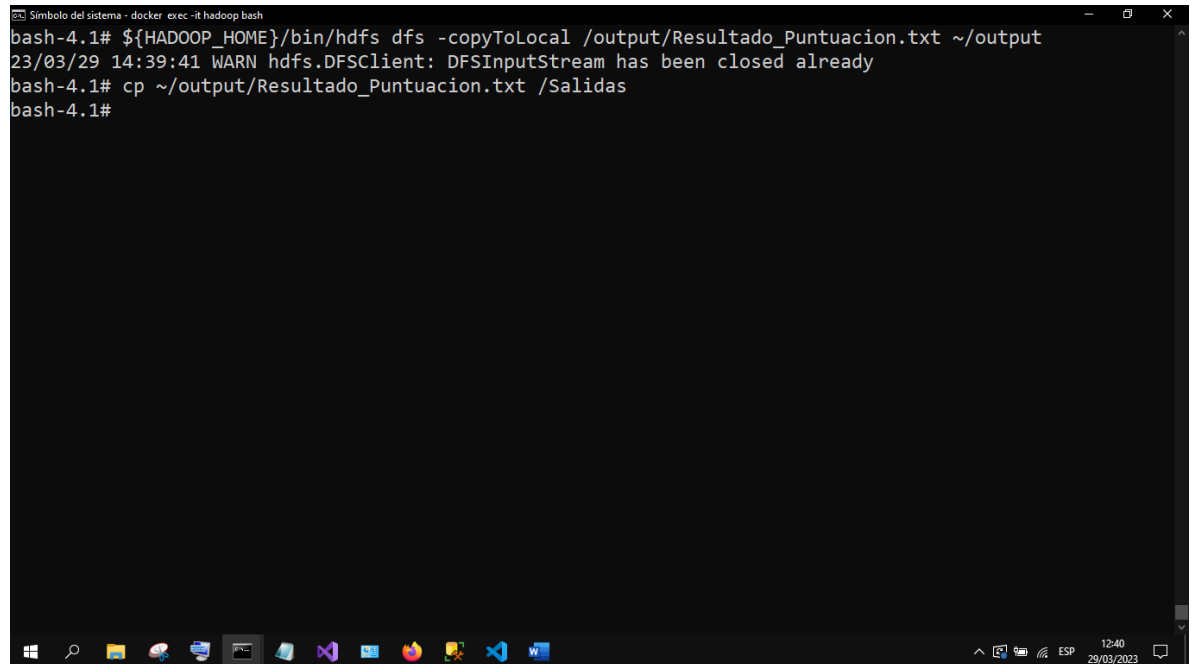


The screenshot shows a terminal window titled "Símbolo del sistema - docker exec -it hadoop bash". The user runs the command `bash-4.1# ${HADOOP_HOME}/bin/hdfs dfs -mv /output/part-r-00000 /output/Resultado_Puntuacion.txt`. Then, the user runs `bash-4.1# ${HADOOP_HOME}/bin/hdfs dfs -cat /output/Resultado_Puntuacion.txt`, which outputs the same numbers as before: 1381, 1070, 1245, 2550, and 2969. The terminal window has a Windows taskbar at the bottom with various application icons and a system clock showing 12:35 on 29/03/2023.

```
Símbolo del sistema - docker exec -it hadoop bash
bash-4.1# ${HADOOP_HOME}/bin/hdfs dfs -mv /output/part-r-00000 /output/Resultado_Puntuacion.txt
bash-4.1# ${HADOOP_HOME}/bin/hdfs dfs -cat /output/Resultado_Puntuacion.txt
1      1381
2      1070
3      1245
4      2550
5      2969
bash-4.1#
```

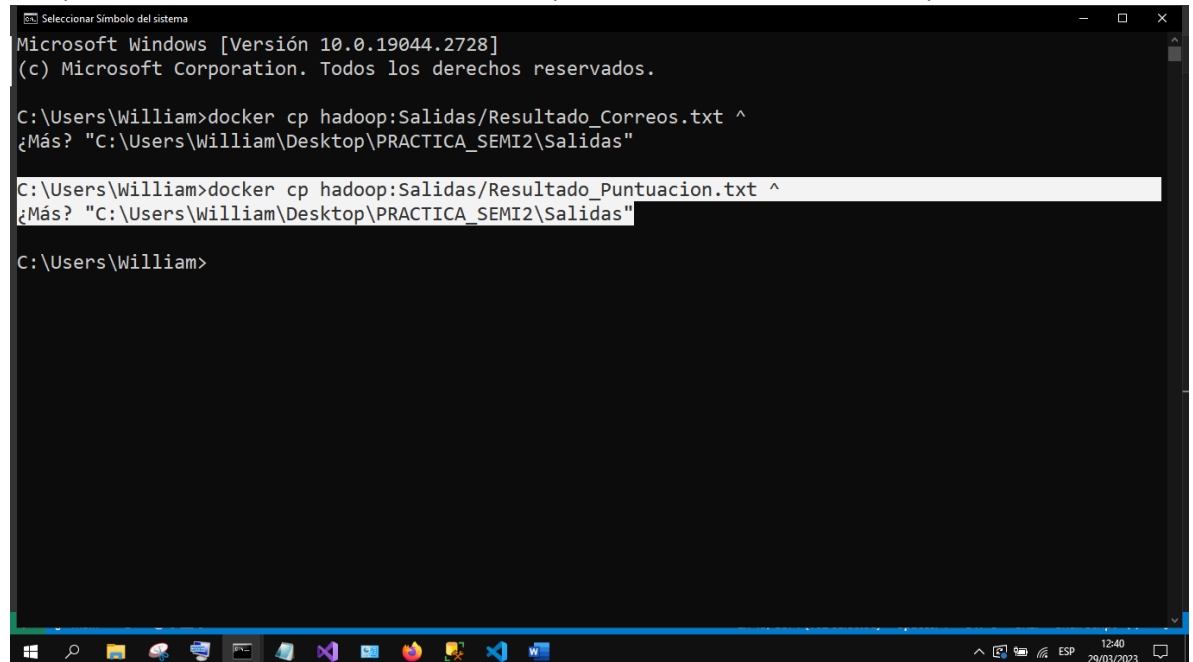


6. Copiamos el archivo de resultado a la carpeta output del home del usuario root, luego hacia la carpeta Salidas (es necesario crearla en el directorio inicial del contenedor), para su fácil extracción del contenedor.



```
Símbolo del sistema - docker exec -it hadoop bash
bash-4.1# ${HADOOP_HOME}/bin/hdfs dfs -copyToLocal /output/Resultado_Puntuacion.txt ~/output
23/03/29 14:39:41 WARN hdfs.DFSClient: DFSInputStream has been closed already
bash-4.1# cp ~/output/Resultado_Puntuacion.txt /Salidas
bash-4.1#
```

7. Finalmente pasamos el archivo de resultado desde el contenedor hacia nuestra computadora, esto desde una nueva consola y hacia el directorio de nuestra preferencia.



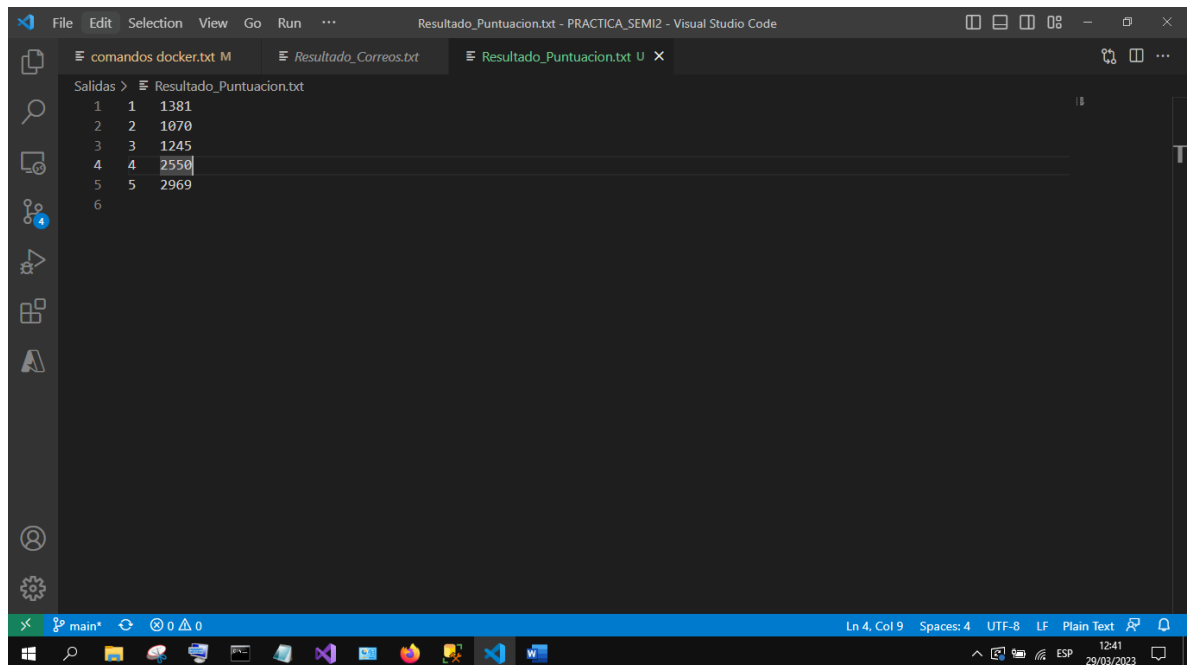
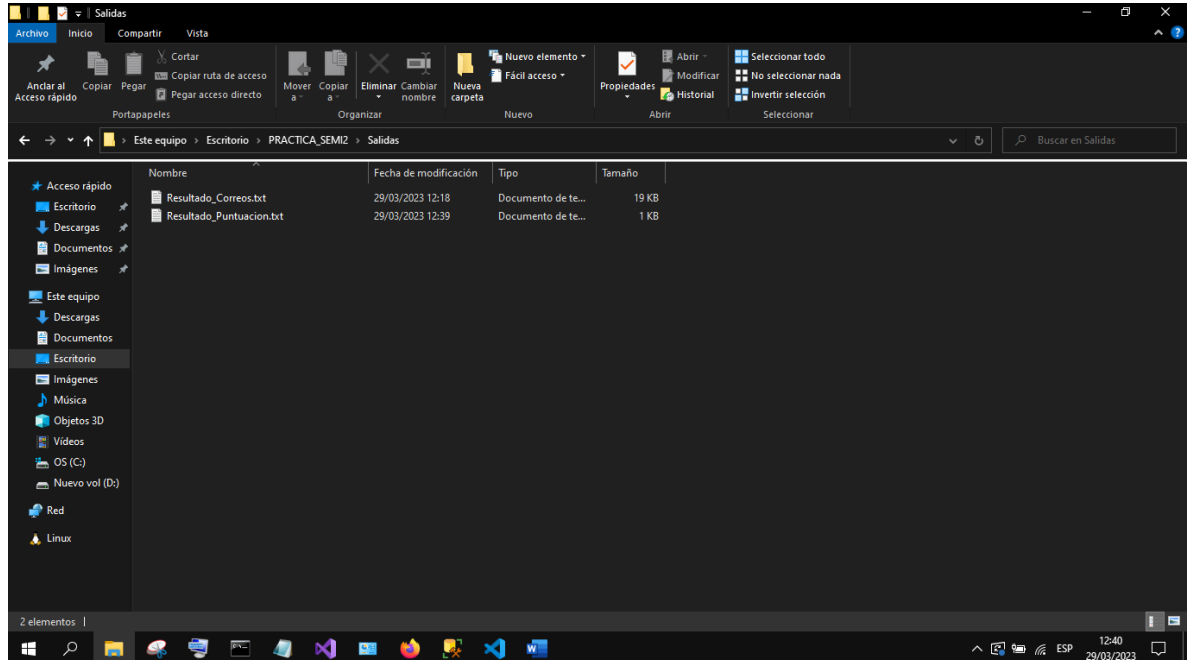
```
Selecciónar Símbolo del sistema
Microsoft Windows [Versión 10.0.19044.2728]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\William>docker cp hadoop:Salidas/Resultado_Correo.txt ^
¿Más? "C:\Users\William\Desktop\PRACTICA_SEMI2\Salidas"

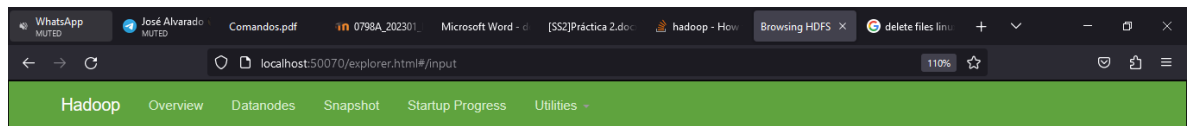
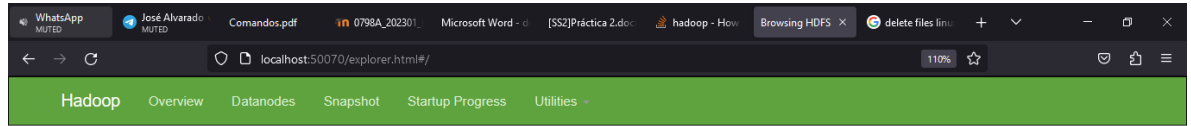
C:\Users\William>docker cp hadoop:Salidas/Resultado_Puntuacion.txt ^
¿Más? "C:\Users\William\Desktop\PRACTICA_SEMI2\Salidas"

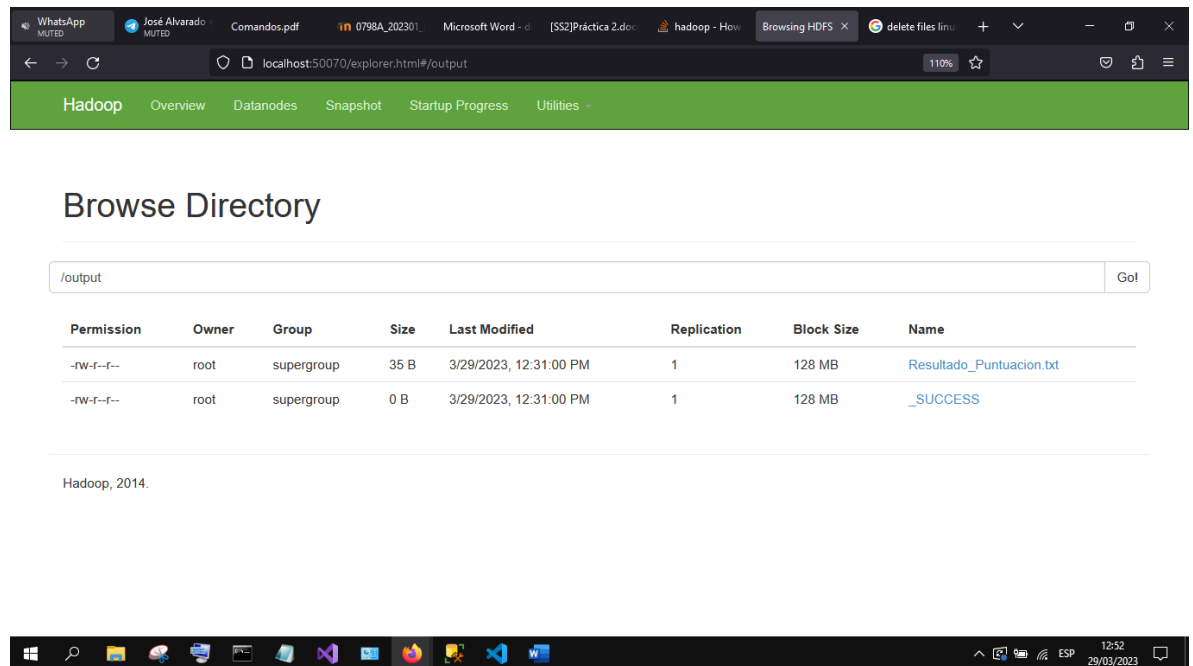
C:\Users\William>
```

Podremos comprobar que el archivo ya está en nuestra computadora y con el mismo contenido:



9. Desde el navegador en localhost:50070, utilities, browse directory se pueden ver los archivos de esta manera:





Archivo de comandos utilizados

<https://drive.google.com/file/d/1LnGh4iACTDktsCw8ECojQKlctKYfFOD/view?usp=sharing>

Análisis acerca de los resultados de cada archivo

**Respecto al archivo Correos.txt:**

Estas son las 5 palabras encontradas en los correos con mayor frecuencia:

Palabra	Conteo
great	46
hotel	114
not	52
room	77
staff	41

Son palabras que se relacionan mucho a la administración de hoteles.

Dado que la empresa SG-Food es una megaempresa destinada a la compra, distribución y comercialización de productos de diferentes marcas y categorías, y que en estos años debido a la crisis que atraviesa el país por el COVID-19 la industria alimentaria o de productos de uso personal han sido las más demandadas, se puede hacer la hipótesis que los compradores están abasteciendo

sus hoteles con este tipo de productos, muy probablemente porque las personas no han podido viajar mucho y hospedarse en estos hoteles, entonces los dueños están buscando formas de mantener la seguridad de sus visitantes y que así las personas puedan hospedarse nuevamente.

Palabra	Conteo
did	35
good	32
n't	38
nice	37
seattle	31
stay	32

Las siguientes 5 palabras más frecuentes también se relacionan a la idea anterior, algunas pueden tener contextos un poco abiertos y variados, pero tampoco representan cantidades demasiado considerables en los conteos ya que según el archivo de Puntuaciones.txt se tienen 9215 opiniones de clientes.

El resto de las palabras de igual forma pueden tener contextos muy variados y son menos utilizadas, aunque se tiene el caso también que Hadoop distinguió aquellas palabras acompañadas de comas o mal escritas. En este caso esas palabras tampoco representan una cantidad muy significativa, pero es conveniente que se realice un agrupamiento de estas palabras con la ayuda de otras herramientas y programas de software.

Para saber si los comentarios son positivos o negativos podemos considerar también el archivo de puntuaciones.

#### Respecto al archivo Puntuacion.txt:

Puntuacion	Conteo
5	2969
4	2550
1	1381
3	1245
2	1070

Las puntuaciones con más frecuencia son las de 5 y 4, esto da a entender que la empresa está brindando un buen servicio a sus clientes. En la tercera posición se encuentra la valoración de 1, esto también indica que una cantidad considerable no está satisfecha con los servicios de la empresa, para analizar estos motivos puede ser necesario acudir a los comentarios de los clientes y buscar aquellos que utilicen palabras negativas como: n't, not, aggravated, bad, entre otras, ya que son este tipo de palabras las que generalmente se emplean al expresar disgusto. Con una base de datos que relacione esta información puede ser más sencillo, pero en este caso los datos no están estructurados.

El último orden de las puntuaciones es 3 y 2, por lo que se sigue considerando que en su mayoría los clientes quedan satisfechos.

## Conclusiones acerca de los resultados de cada archivo

### Correos.txt

- Se deben utilizar herramientas y programas para agrupar palabras similares (mal escritas o con comas), porque con mayores cantidades de datos sí puede perjudicar la toma correcta de decisiones.
- La identificación de palabras negativas ayuda a descubrir motivos de disgusto en los clientes.
- La mayoría de los consumidores son personas con trabajos relacionados a los hoteles.

### Puntuación.txt

- La mayoría de las clientes está satisfecha con el servicio que la empresa les brinda.
- Una cantidad considerable de personas no está satisfecha con el servicio que les provee la empresa.
- Estos datos ayudan a tener conclusiones generales de manera más rápida gracias a que son numéricos y más directos. Para su profunda investigación se acude a archivos como el de Correos.txt

## Conclusiones acerca del uso de Hadoop en BigData

- Es una herramienta capaz de realizar diferentes tipos de análisis gracias a su funcionamiento con códigos de java.
- Ayuda a visualizar de una mejor forma los datos no estructurados para generar hipótesis y/o brindar conclusiones.
- Si el contexto y la complejidad de los datos es muy amplia para realizar análisis específicos, Hadoop funciona bien como guía para saber por dónde empezar.

## Bibliografía

- <https://hadoop.apache.org/>
- <https://www.docker.com/>