



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
SEMINARIO DE SISTEMAS 2
SECCIÓN N
ING. LUIS VETTORAZZI
AUX. ESCARLETH VELASCO

Práctica #2

Nombre: Pablo Fernando Cabrera Pineda

Carnet: 2019012698

Manual para procesamiento de datos utilizando Hadoop

Instalación de Hadoop

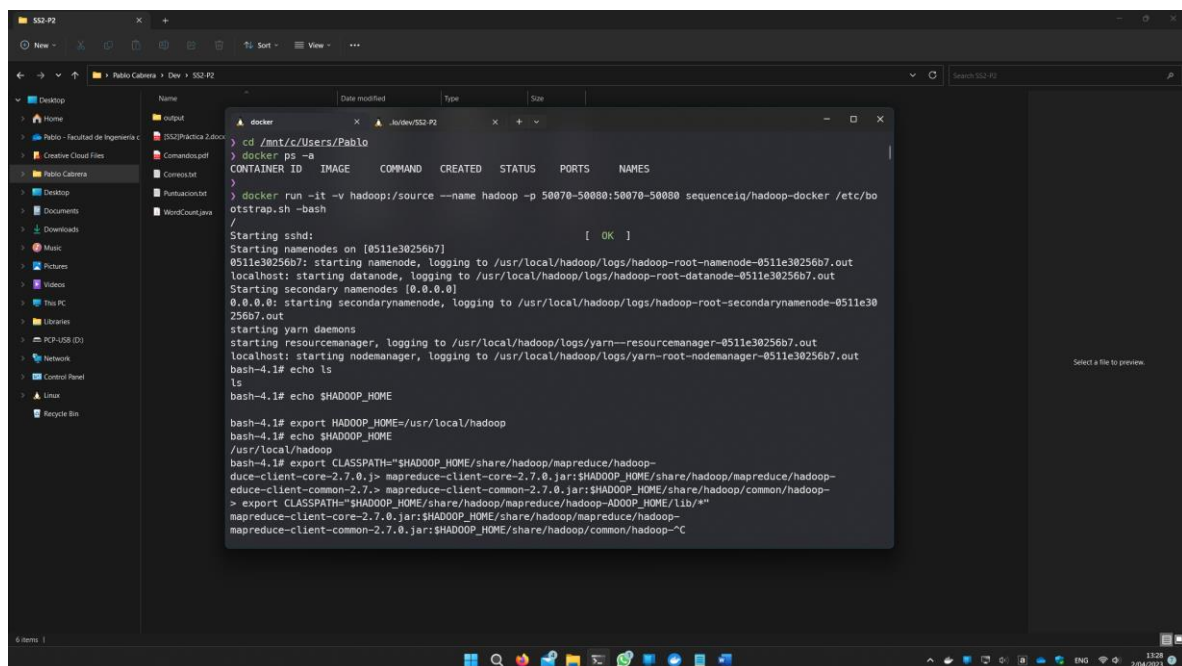
Existen dos formas de instalar hadoop:

- Instalar en el dispositivo
- Correr un contenedor con docker

En este manual se utilizó docker para hadoop por lo que se incluyen los pasos para correr el contenedor en cuestión.

Correr el contenedor

Para correr el contenedor se debe ejecutar el siguiente comando:



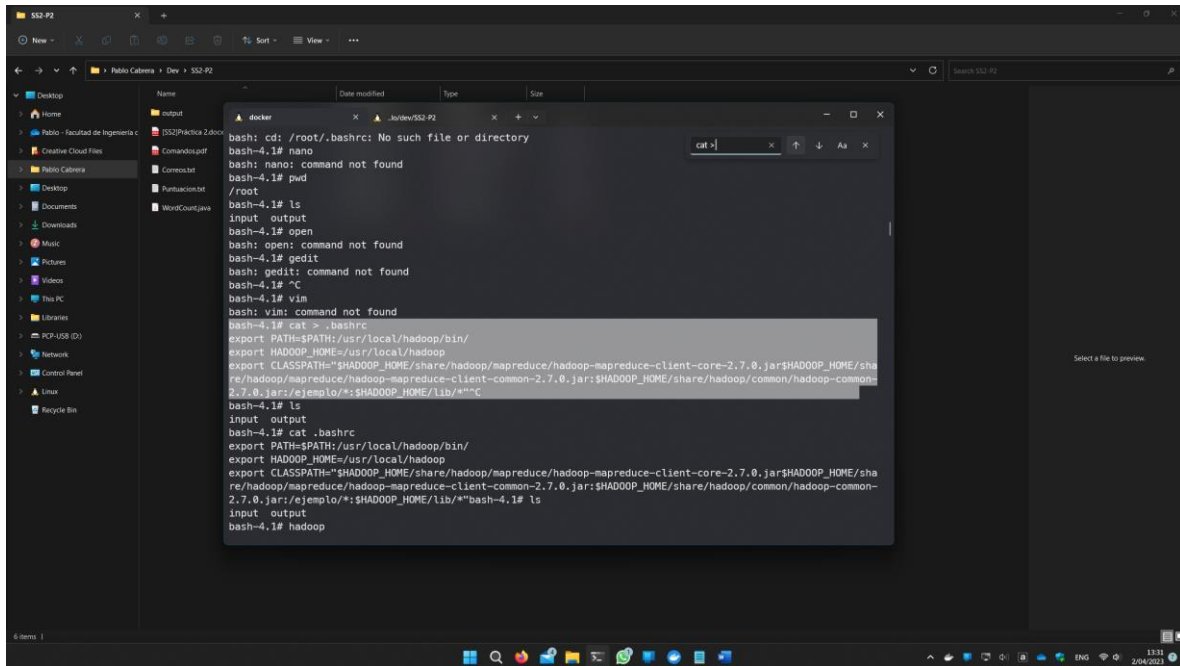
```
cd /mnt/c/Users/Pablo
> docker ps -a
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS        NAMES
> docker run -it -v hadoop:/source --name hadoop -p 50070-50080:50070-50080 sequenceiq/hadoop-docker /etc/bo
otstrap.sh -bash
Starting sshd: [ OK ]
Starting namenodes on [0511e30256b7]
0511e30256b7: starting namenode, logging to /usr/local/hadoop/logs/hadoop-root-namenode-0511e30256b7.out
localhost: starting datanode, logging to /usr/local/hadoop/logs/hadoop-root-datanode-0511e30256b7.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-root-secondarynamenode-0511e30
256b7.out
starting yarn daemons
Starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-root-resourcemanager-0511e30256b7.out
localhost: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-root-nodemanager-0511e30256b7.out
bash-4.1# echo ls
ls
bash-4.1# echo $HADOOP_HOME
/usr/local/hadoop
bash-4.1# export HADOOP_HOME=/usr/local/hadoop
bash-4.1# echo $HADOOP_HOME
/usr/local/hadoop
bash-4.1# export CLASSPATH=$HADOOP_HOME/share/hadoop/mapreduce/hadoop-
duce-client-core-2.7.0.jar:mapreduce-client-core-2.7.0.jar:$HADOOP_HOME/share/hadoop/mapreduce/hadoop-
duce-client-common-2.7.0.jar:mapreduce-client-common-2.7.0.jar:$HADOOP_HOME/share/hadoop/common/hadoop-
> export CLASSPATH=$HADOOP_HOME/share/hadoop/mapreduce/hadoop-ADOOP_HOME/lib/*
mapreduce-client-core-2.7.0.jar:$HADOOP_HOME/share/hadoop/mapreduce/hadoop-
mapreduce-client-common-2.7.0.jar:$HADOOP_HOME/share/hadoop/common/hadoop-~C
```

Este comando se encarga de varias cosas:

- Descargar la imagen hadoop:latest, si no se tiene actualmente.
- Correr un contenedor de hadoop con los puertos 50070 – 50080 mapeados a los puertos del dispositivo.
- Ejecutar bash en el contenedor.

Preparación del contenedor

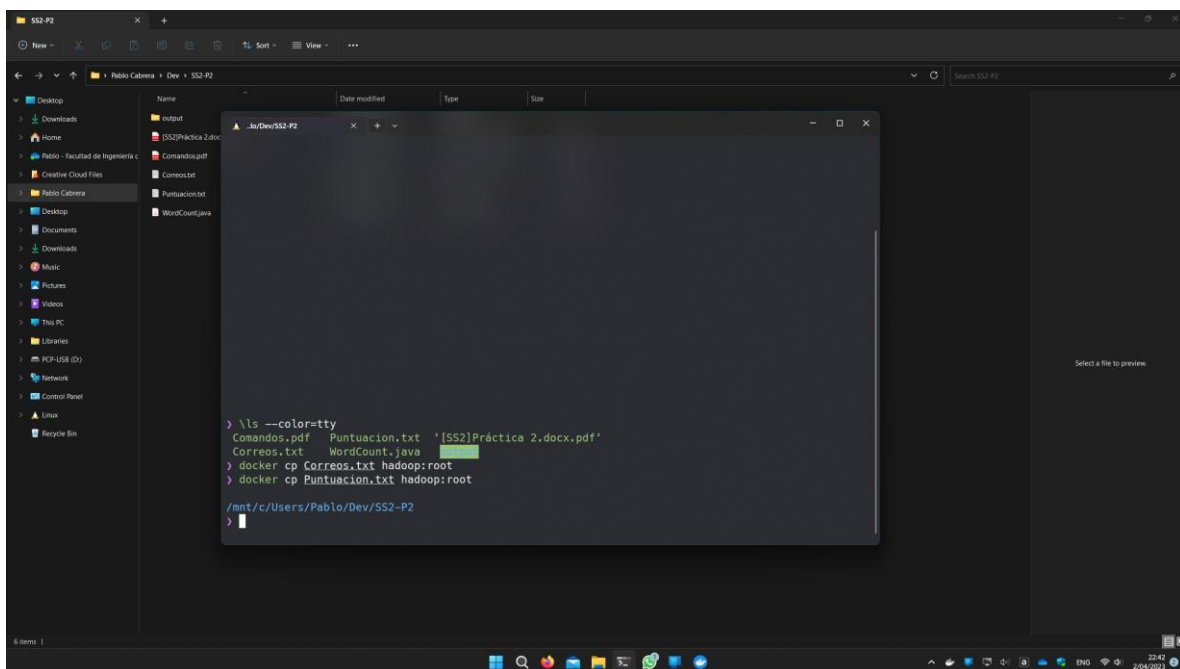
Configurar las variables de entorno necesarias para ejecutar hadoop. Estas se escriben en `/root/.bashrc` para que siempre que se entre al contenedor estén disponibles.



```
bash: cd: /root/.bashrc: No such file or directory
bash-4.1# nano
bash: nano: command not found
bash-4.1# pwd
/root
bash-4.1# ls
input output
bash-4.1# open
bash: open: command not found
bash-4.1# gedit
bash: gedit: command not found
bash-4.1# vim
bash: vim: command not found
bash-4.1# cat > .bashrc
export PATH=$PATH:/usr/local/hadoop/bin/
export HADOOP_HOME=/usr/local/hadoop
export CLASSPATH=$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:$HADOOP_HOME/lib/*
bash-4.1# ls
input output
bash-4.1# cat .bashrc
export PATH=$PATH:/usr/local/hadoop/bin/
export HADOOP_HOME=/usr/local/hadoop
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:$HADOOP_HOME/share/hadoop/common/hadoop-common-2.7.0.jar:$HADOOP_HOME/lib/*
bash-4.1# ls
input output
bash-4.1# hadoop
```

Copiar archivos de entrada

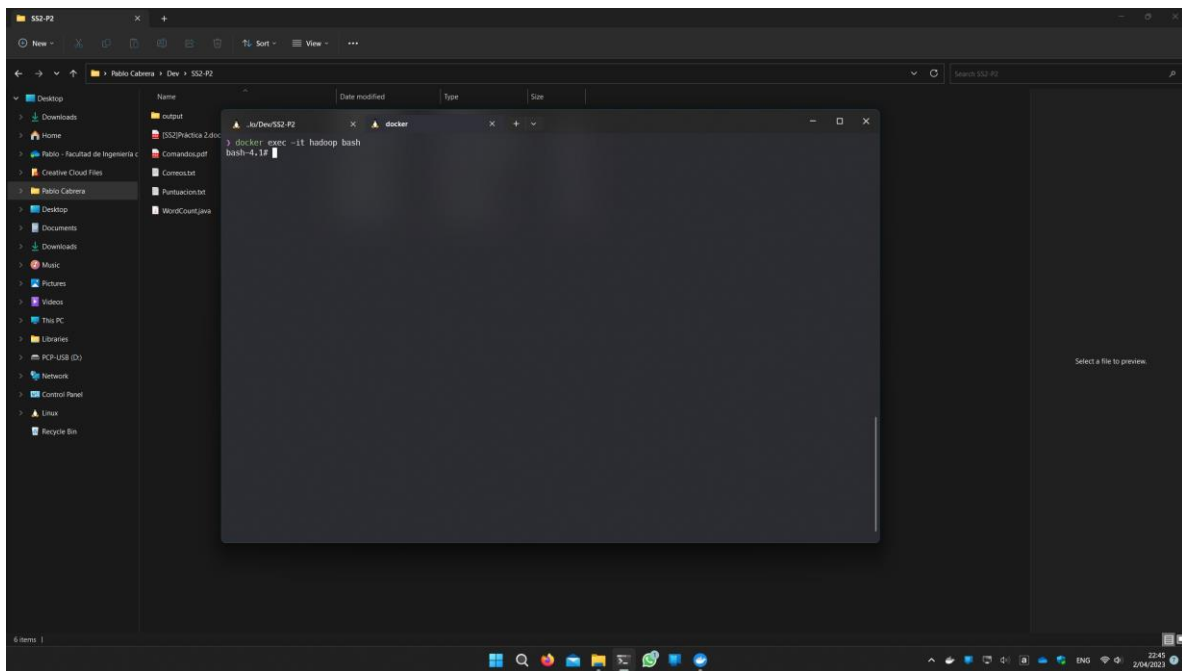
Para copiar los archivos de entrada hacia el contenedor se debe utilizar el siguiente comando:



```
> \ls --color=tty
Comandos.pdf  Puntuacion.txt  '[SS2]Práctica 2.docx.pdf'
Correos.txt   WordCount.java
> docker cp Correos.txt hadoop:root
> docker cp Puntuacion.txt hadoop:root

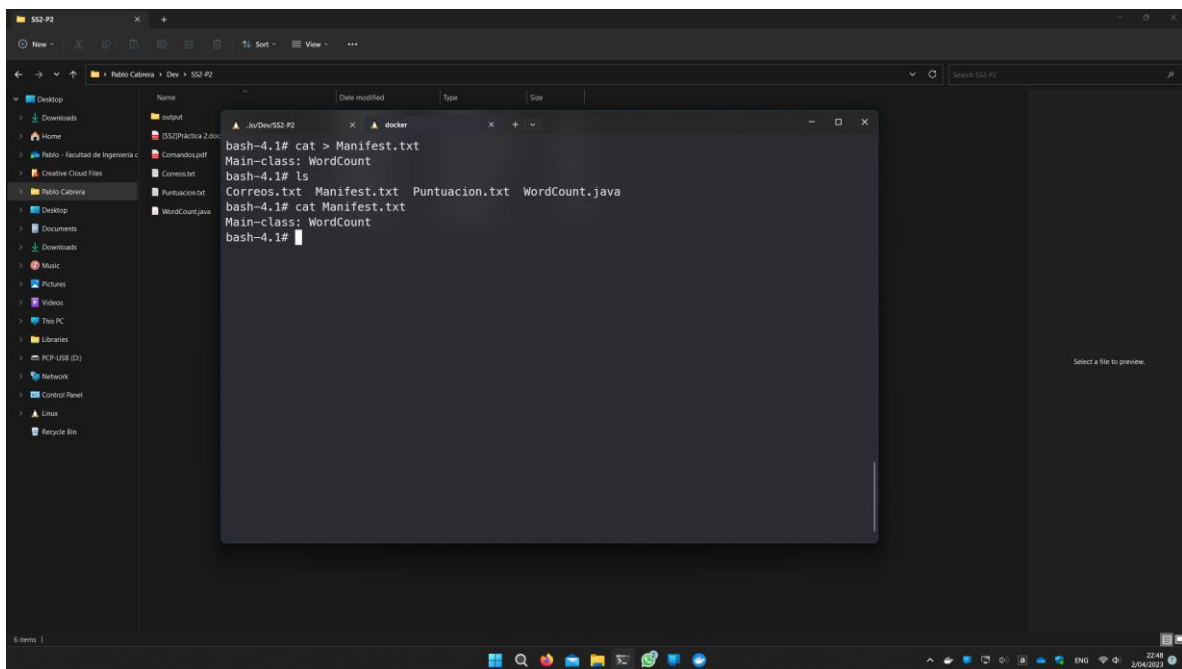
/mnt/c/Users/Pablo/Dev/SS2-P2
>
```

Ejecutar bash en el contenedor

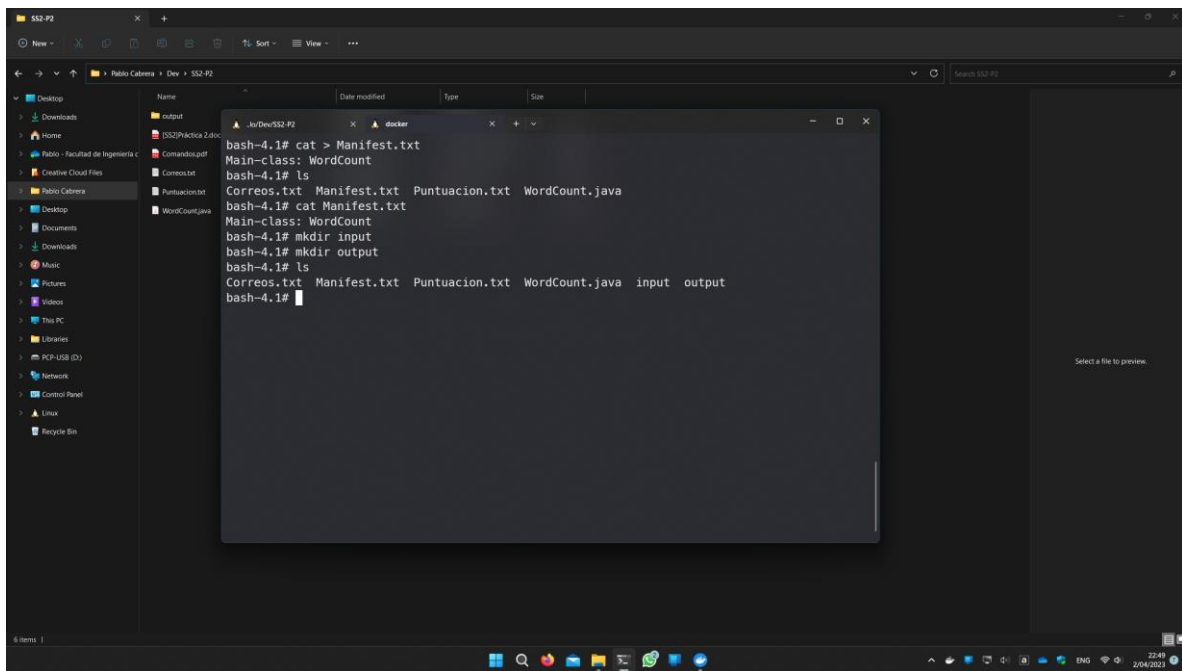


Crear archivo Manifest

El archivo manifest es necesario para crear el jar para procesar los archivos de entrada



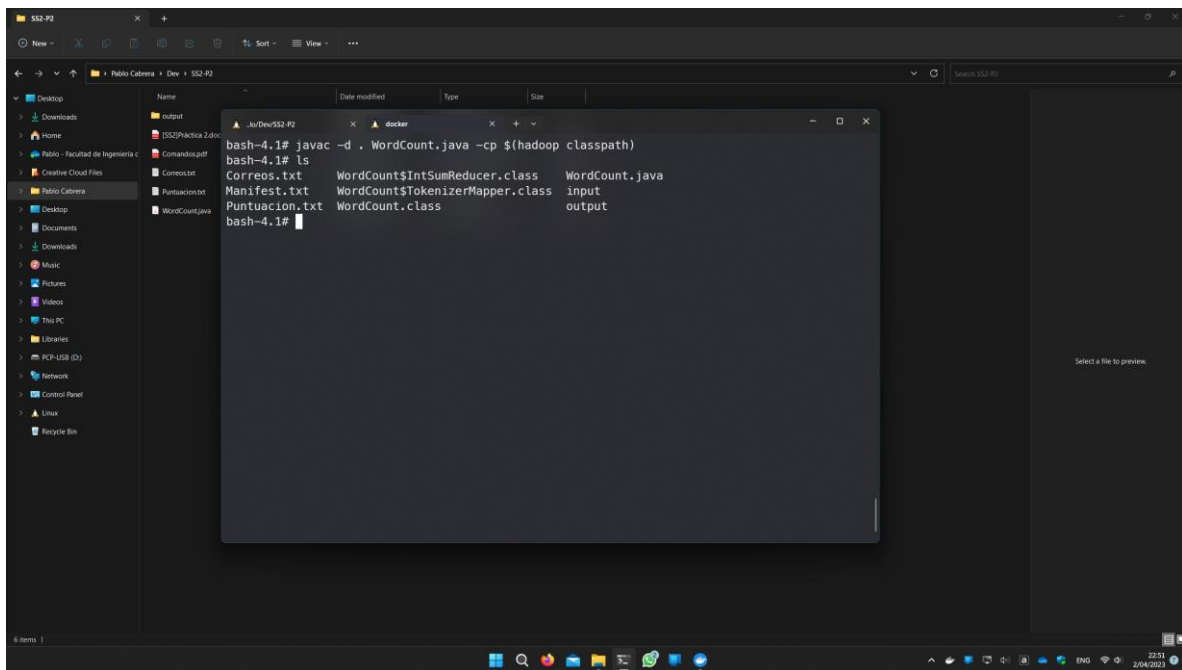
Crear carpetas en el contenedor para entrada y salida de archivos



The screenshot shows a Docker container terminal with the following commands and output:

```
bash-4.1# cat > Manifest.txt
Main-class: WordCount
bash-4.1# ls
Correos.txt  Manifest.txt  Puntuacion.txt  WordCount.java
bash-4.1# cat Manifest.txt
Main-class: WordCount
bash-4.1# mkdir input
bash-4.1# mkdir output
bash-4.1# ls
Correos.txt  Manifest.txt  Puntuacion.txt  WordCount.java  input  output
bash-4.1#
```

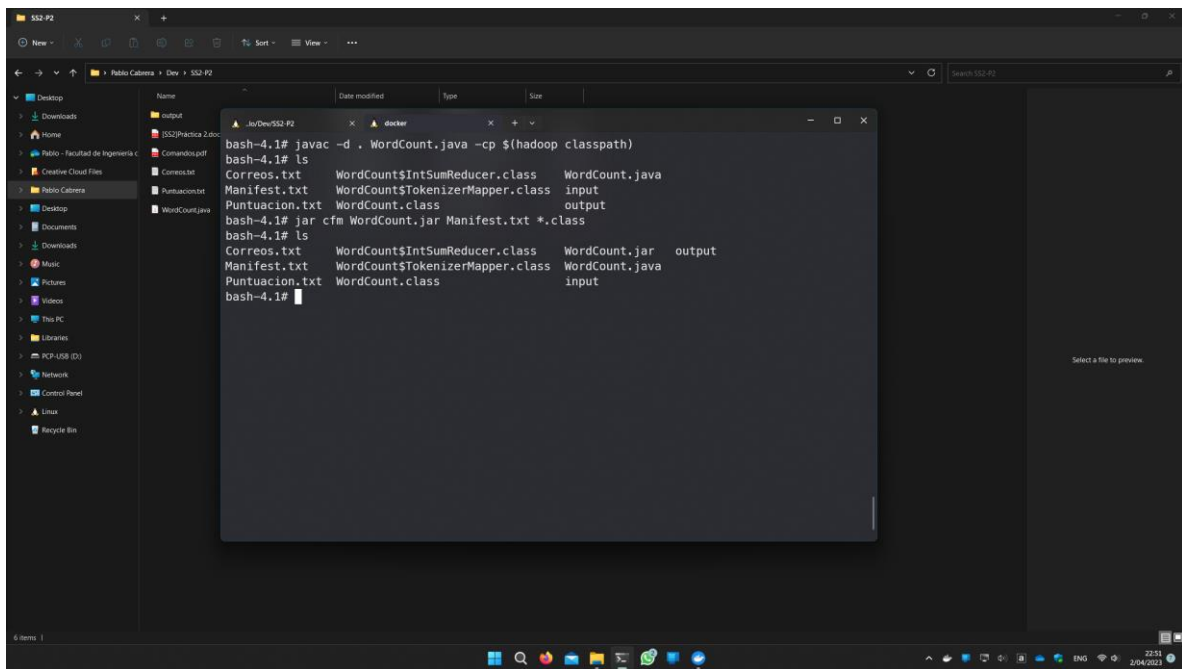
Compilar el java de WordCount



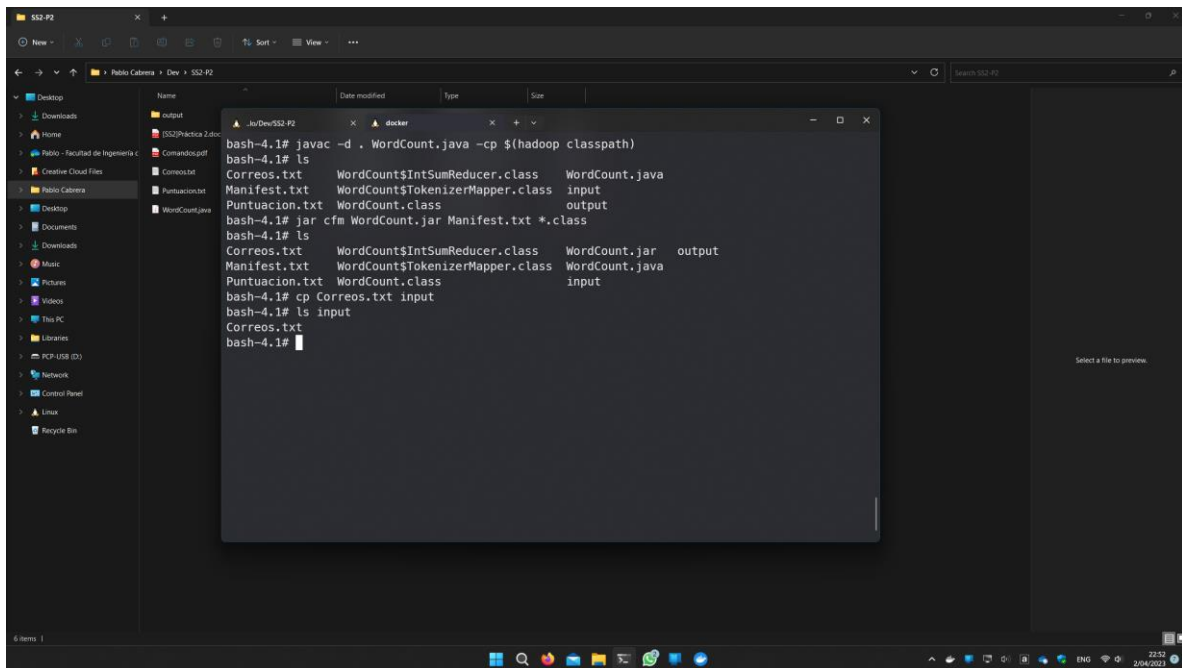
The screenshot shows a Docker container terminal with the following commands and output:

```
bash-4.1# javac -d . WordCount.java -cp $(hadoop classpath)
bash-4.1# ls
Correos.txt  WordCount$IntSumReducer.class  WordCount.java
Manifest.txt  WordCount$TokenizerMapper.class  input
Puntuacion.txt  WordCount.class  output
bash-4.1#
```

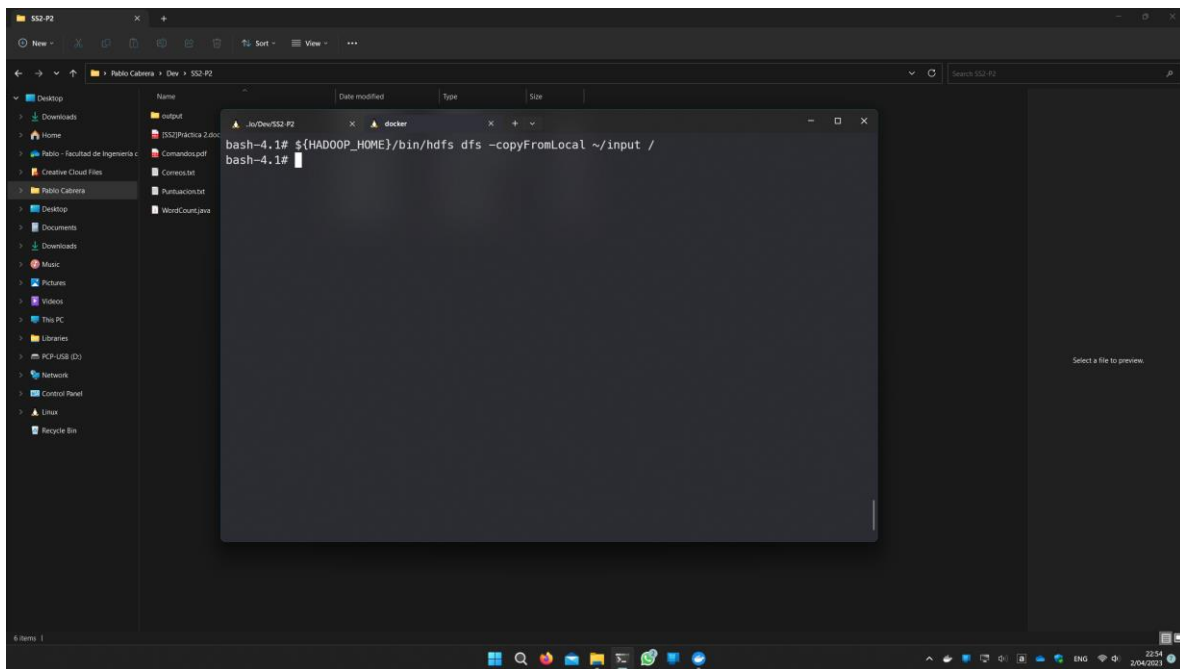
Crear el jar de WordCount



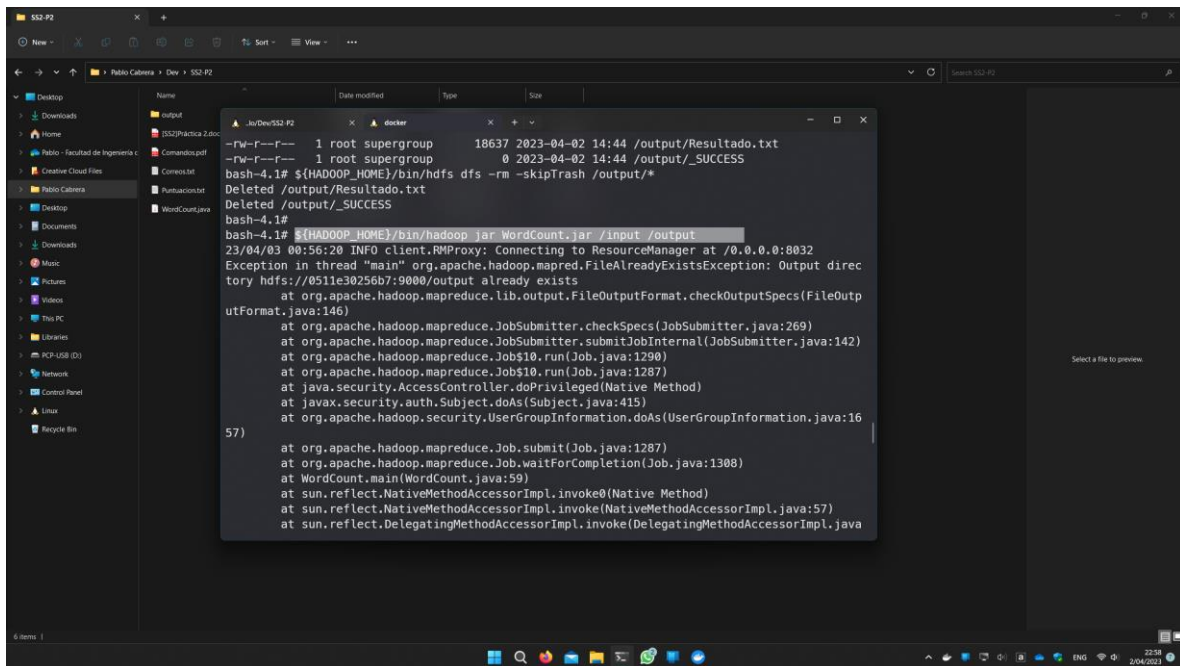
Copiar el archivo de entrada para hadoop



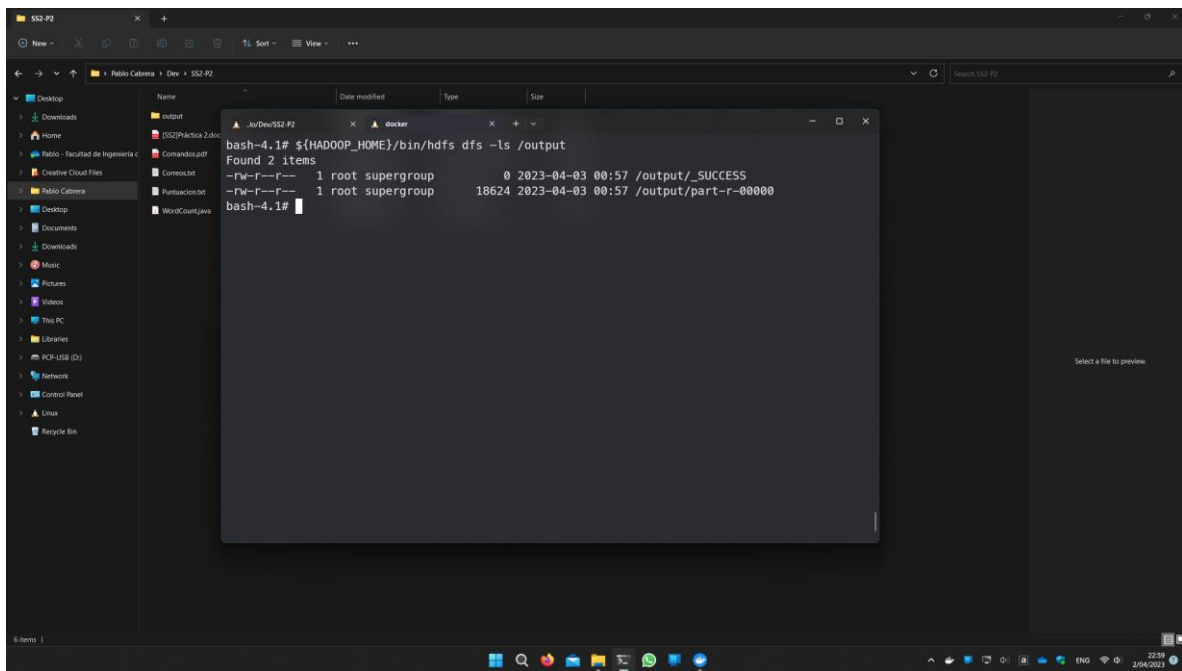
Copiar el archivo de entrada al sistema de archivos de hadoop



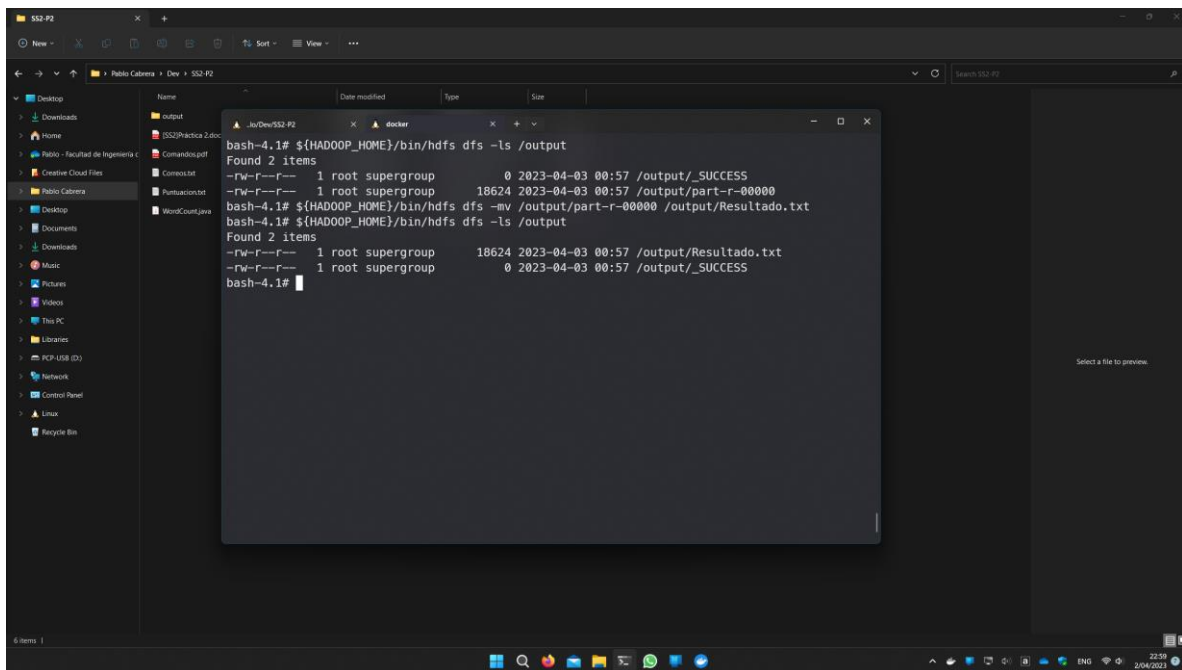
Ejecutar el análisis con hadoop



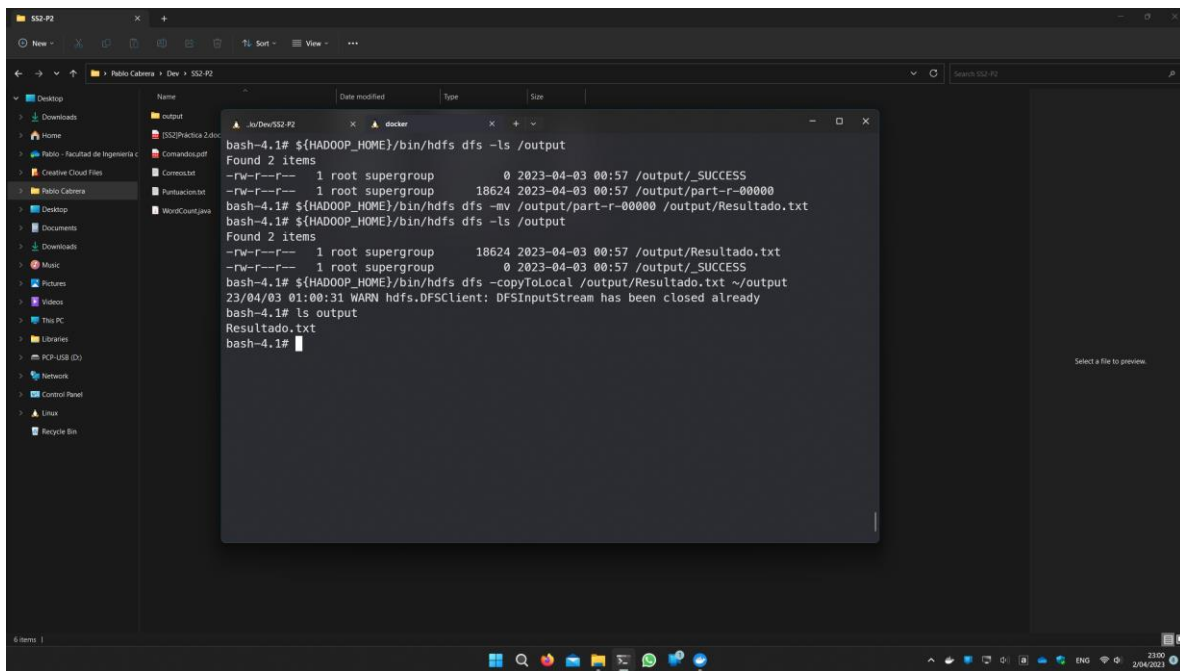
Verificar que se creó el archivo de salida



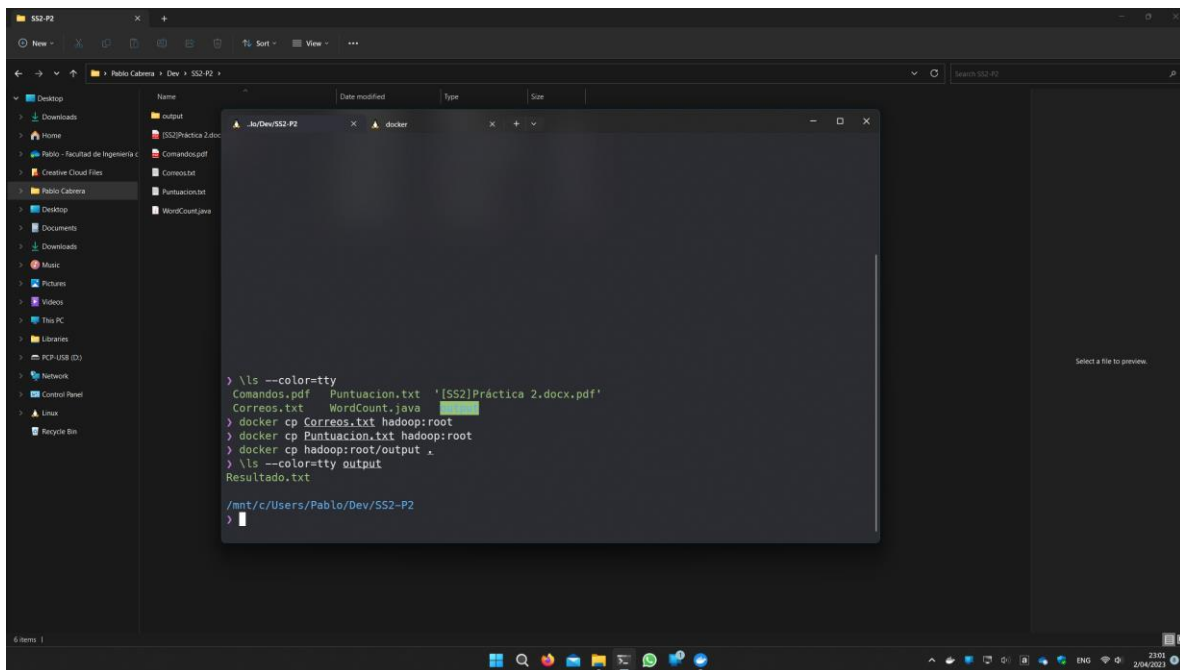
Cambiar el nombre del archivo de salida a Resultado.txt



Copiar el archivo de salida al contenedor



Copiar el archivo de salida afuera del contenedor



Sistema de archivos de HDFS

Browse Directory

/output

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	root	supergroup	18.19 KB	2/4/2023, 22:57:20	1	128 MB	Resultado_Correo.txt
-rw-r--r--	root	supergroup	18.19 KB	2/4/2023, 23:05:20	1	128 MB	Resultado_Puntuacion.txt
-rw-r--r--	root	supergroup	0 B	2/4/2023, 22:57:20	1	128 MB	_SUCCESS

Hadoop, 2014.

Conclusiones

Archivo Correos.txt

En la lista se puede observar la cantidad de veces que se repite cada palabra o frase en los comentarios. Algunas de las palabras más repetidas son "room" (39 veces), "hotel" (33 veces), "staff" (23 veces), "location" (18 veces), "great" (17 veces), "clean" (16 veces), "good" (14 veces), "service" (13 veces) y "stay" (12 veces).

Al analizar estas repeticiones, se puede inferir que los aspectos más importantes para los huéspedes son la calidad de la habitación y la limpieza del hotel, seguido de la atención del personal y la ubicación. Los huéspedes también parecen estar satisfechos con el servicio del hotel en general.

Sin embargo, también hay algunas palabras negativas que se repiten con frecuencia, como "bad" (5 veces), lo que sugiere que algunos huéspedes tuvieron experiencias negativas en el hotel. Además, algunos huéspedes se quejan de problemas con el aire acondicionado y del ruido en el hotel.

En general, el análisis de las palabras más repetidas en los comentarios sugiere que el hotel tiene algunas áreas de oportunidad para mejorar, pero en general parece ser un lugar agradable para hospedarse.

Archivo Puntuacion.txt

De acuerdo con los datos, se puede observar que la mayoría de los evaluadores han calificado al hotel con una puntuación de 4 o 5 estrellas, ya que el mayor número de evaluadores (2550 y 2969, respectivamente) han dado estas puntuaciones. Además, se puede ver que la cantidad de evaluadores que han dado una puntuación de 1 o 2 estrellas es menor en comparación con los que han dado una puntuación de 3, 4 o 5 estrellas.

Estos resultados indican que en general, los evaluadores tienen una buena opinión del hotel. Sin embargo, sería necesario realizar un análisis más detallado para comprender mejor las razones detrás de estas puntuaciones y ver si hay algún aspecto específico del hotel que esté influyendo en las evaluaciones.

Uso de hadoop

El uso de Hadoop en Big Data se ha convertido en una solución popular debido a su capacidad para procesar grandes volúmenes de datos de manera eficiente y escalable, lo que significa que se puede agregar más nodos al clúster para procesar más datos. Esto lo hace ideal para procesar grandes cantidades de datos. En resumen, Hadoop es una solución efectiva para el procesamiento de Big Data, que proporciona escalabilidad, procesamiento distribuido, tolerancia a fallos, bajo costo y versatilidad.