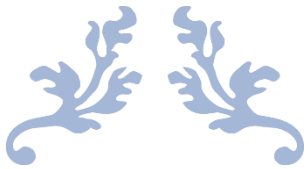
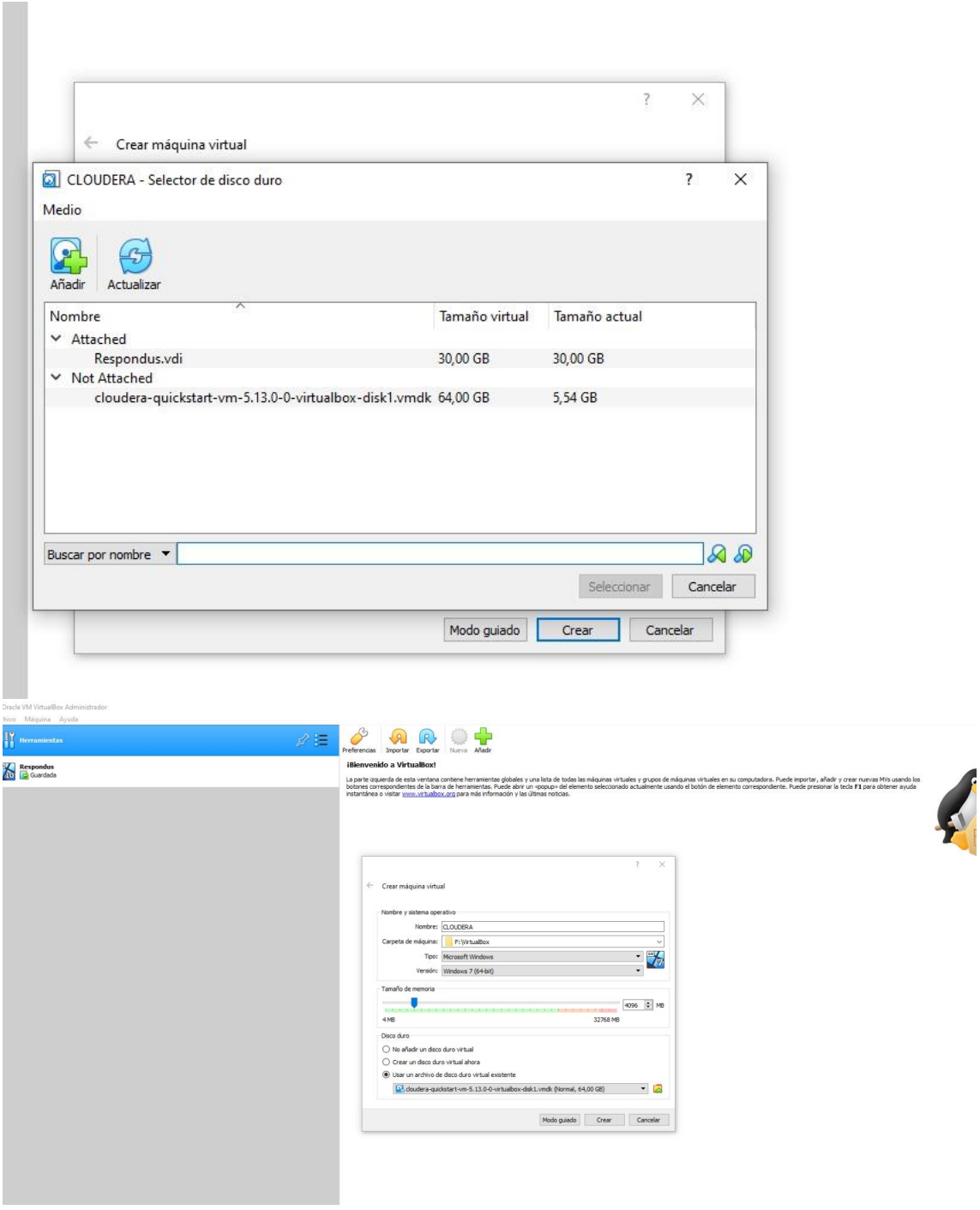


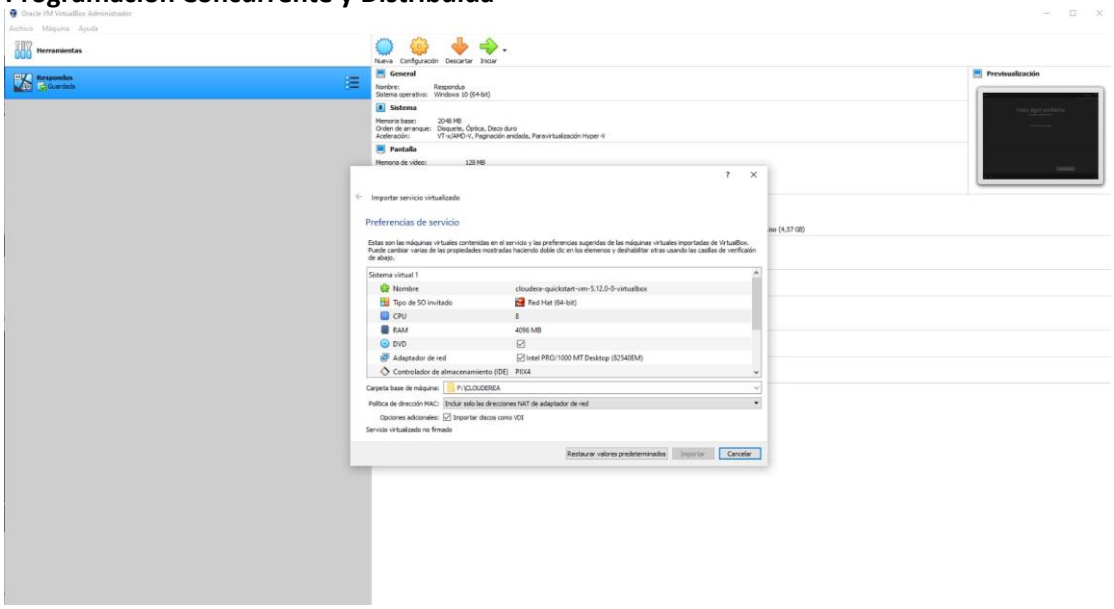
PRÁCTICA HADOOP

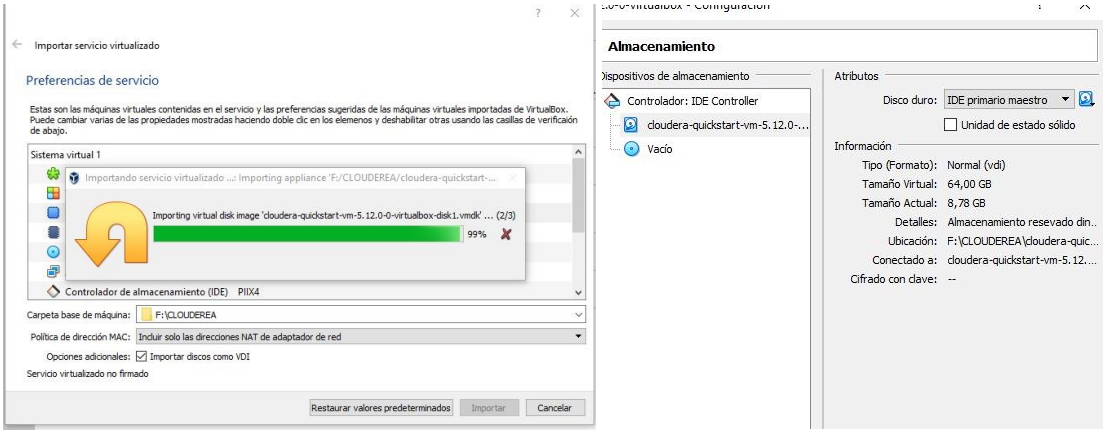
PROGRAMACIÓN CONCURRENTE Y DISTRIBUIDA

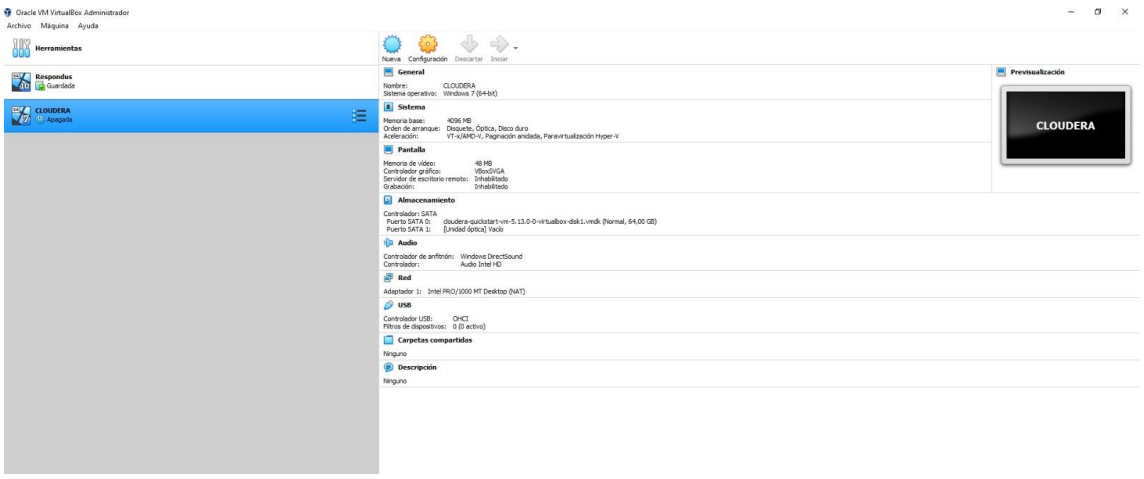
MARCOS ELADIO SOMOZA CORRAL
21711787

El primer paso es instalar una máquina virtual, en mi caso, instalé Orecle VM VirtualBox. Una vez instalada, y teniendo el archivo vmdk de Cloudera procedemos a instalarlo en la máquina virtual. (https://downloads.cloudera.com/demo_vm/virtualbox/cloudera-quickstart-vm-5.12.0-0-virtualbox.zip)

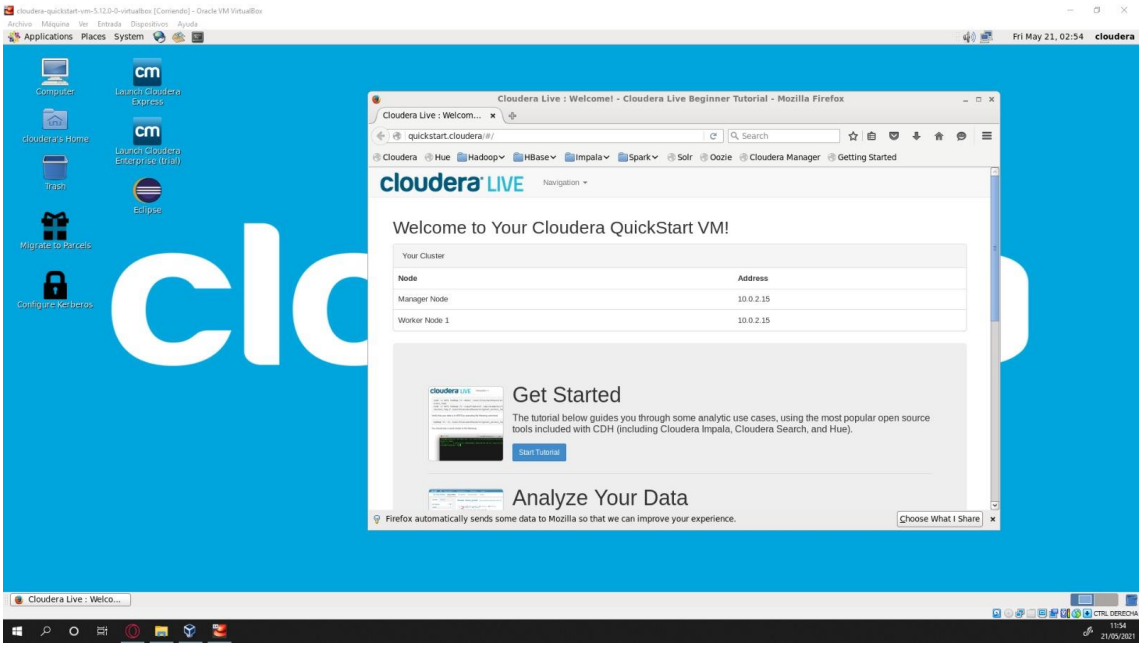



Importar servicio virtualizado
Preferencias de servicio


Importar servicio virtualizado
Preferencias de servicio
Estas son las máquinas virtuales contenidas en el servicio y las preferencias sugeridas de las máquinas virtuales importadas de VirtualBox. Puede cambiar varias de las propiedades mostradas haciendo doble clic en los elemenos y deshabilitar otras usando las casillas de verificación de abajo.
Sistema virtual 1
Importando servicio virtualizado ...: Importing appliance 'F:/CLOUDEREA/cloudera-quickstart-...
Importing virtual disk image 'cloudera-quickstart-vm-5.12.0-0-virtualbox-disk1.vmdk' ... (2/3)
99%
Controlador de almacenamiento (IDE) PIIX4
Carpeta base de máquina: F:\CLOUDEREA
Política de dirección MAC: Incluir solo las direcciones NAT de adaptador de red
Opciones adicionales: Importar discos como VDI
Servicio virtualizado no firmado
Restaurar valores predeterminados
Importar
Cancelar
Almacenamiento
Dispositivos de almacenamiento
Controlador: IDE Controller
cloudera-quickstart-vm-5.12.0-...
Vacío
Atributos
Disco duro: IDE primario maestro
Unidad de estado sólido
Información
Tipo (Formato): Normal (vdi)
Tamaño Virtual: 64,00 GB
Tamaño Actual: 8,78 GB
Detalles: Almacenamiento reservado din..
Ubicación: F:\CLOUDEREA\cloudera-quic...
Conectado a: cloudera-quickstart-vm-5.12....
Cifrado con clave: --

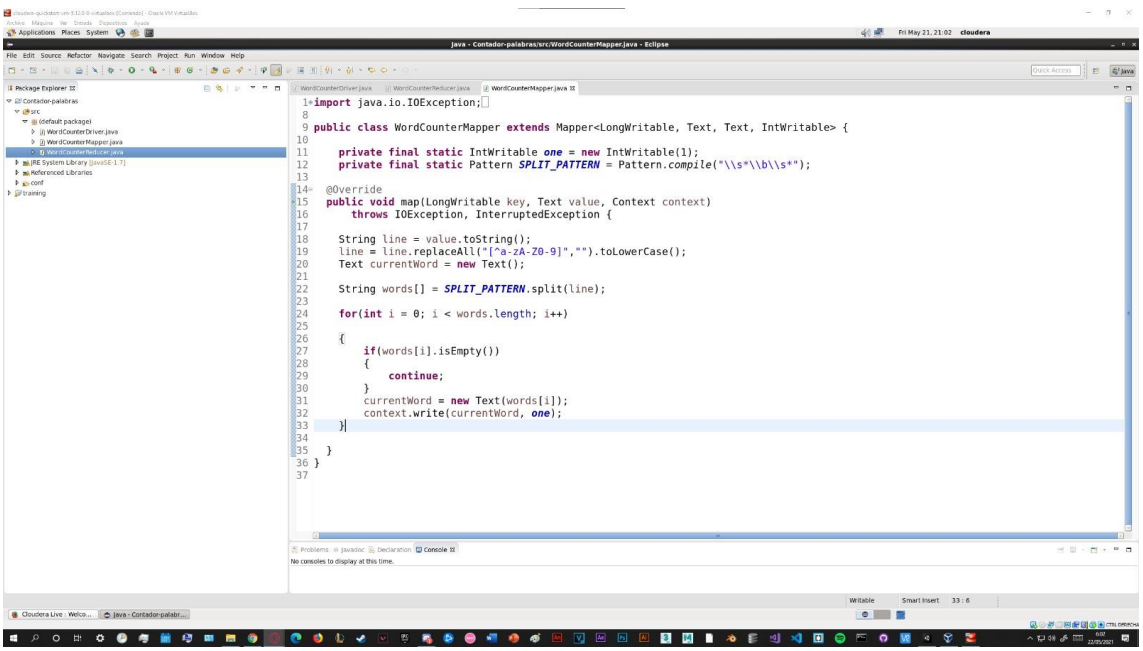

Oracle VM VirtualBox Administrador
Archivo Máquina Ayuda
Herramientas
Respondus
Guardada
CLOUDERA
Apagada
Nueva
Configuración
Descartar
Iniciar
General
Sistema
Pantalla
Almacenamiento
Audio
Red
USB
Carpetas compartidas
Descripción
Previsualización
CLOUDERA

Una vez instalado, iniciamos el sistema de Cloudera.

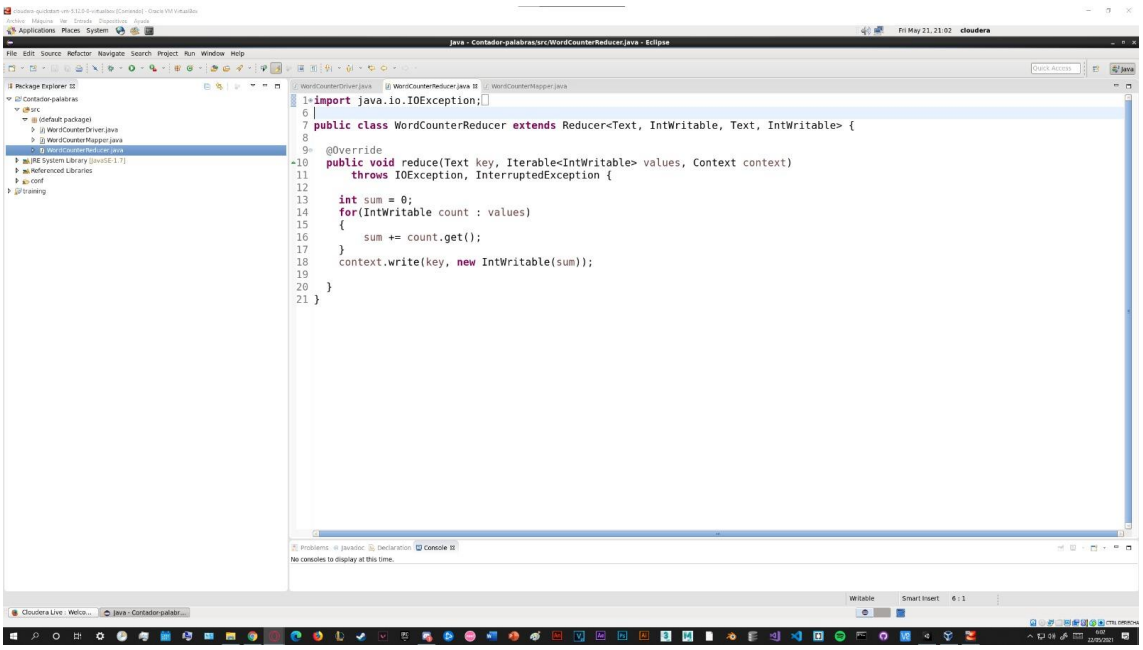


Ya en Cloudera, abriremos eclipse y crearemos los scripts necesarios para realizar el map reduce en hadoop.

Primero crearemos el script que hará de mapper (WordCounterMapper.java):

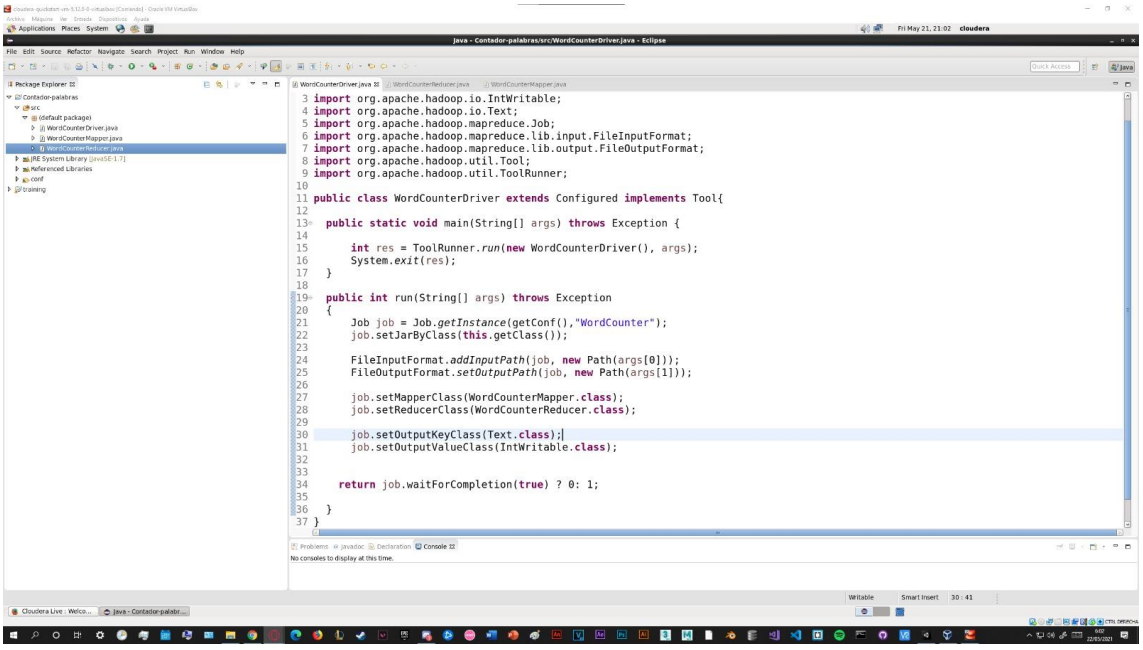


Después el script que hará de reducer (WordCounterReducer):



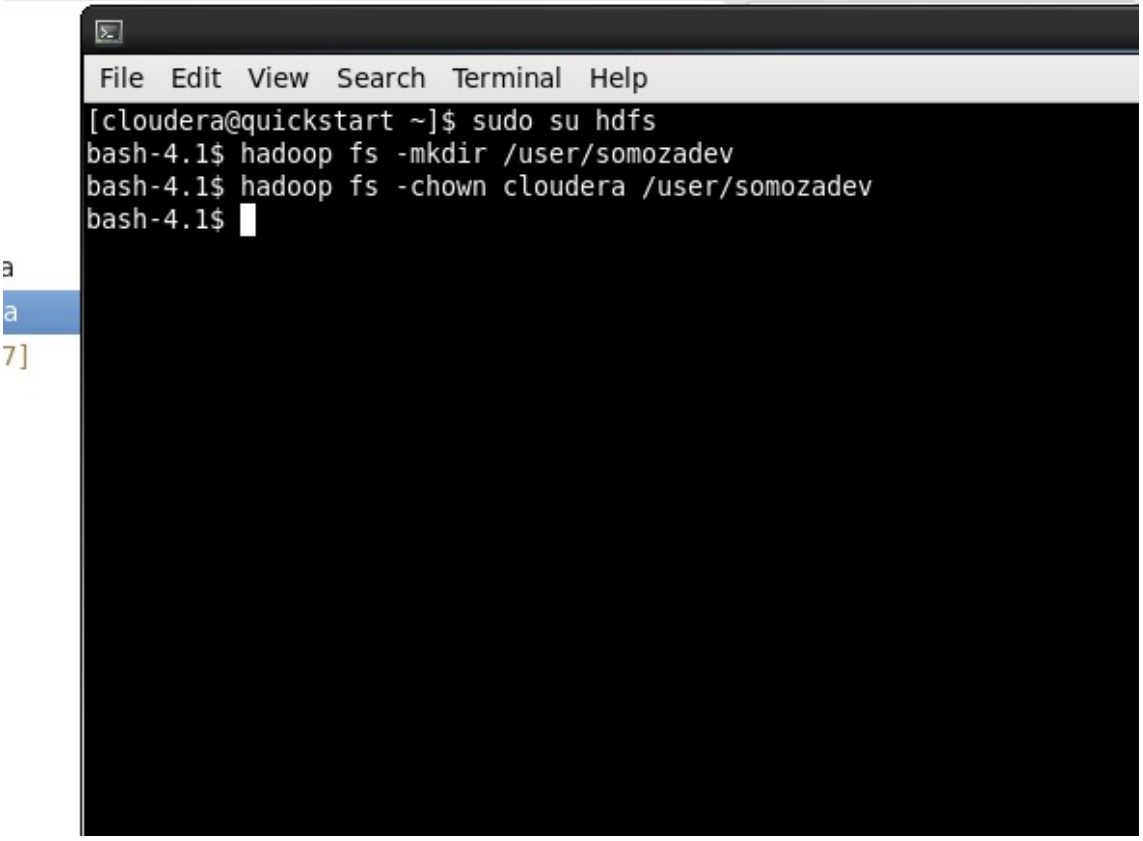
```
1  import java.io.IOException;
6
7  public class WordCounterReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
8
9    @Override
10   public void reduce(Text key, Iterable<IntWritable> values, Context context)
11       throws IOException, InterruptedException {
12
13     int sum = 0;
14     for(IntWritable count : values)
15     {
16         sum += count.get();
17     }
18     context.write(key, new IntWritable(sum));
19
20   }
21 }
```

Y por último el driver donde estará ubicado el main(), en este controlaremos el Job correspondiente al mapreduce (WordCounterDriver):

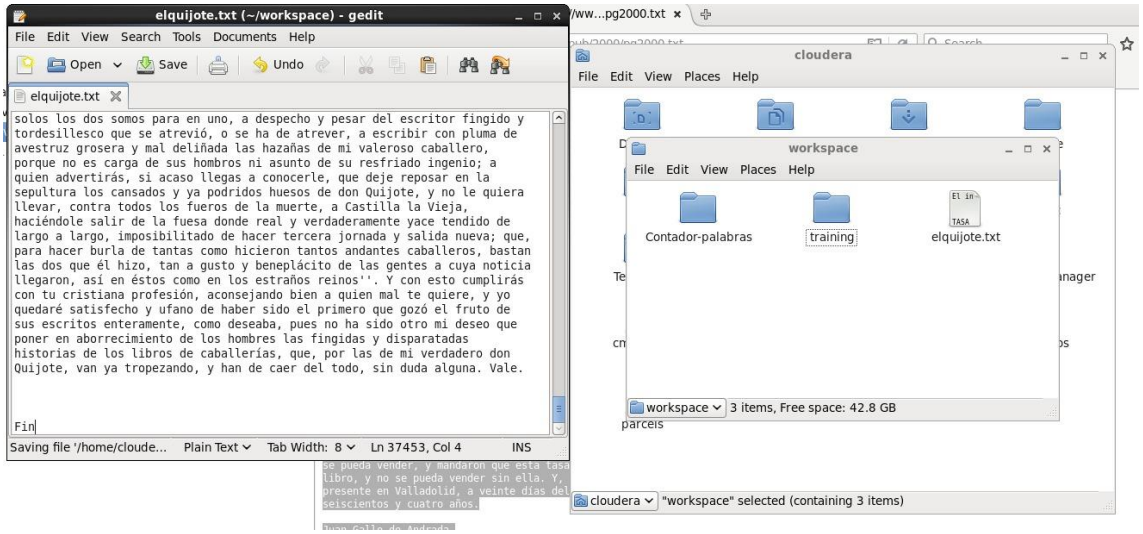


```
3  import org.apache.hadoop.io.IntWritable;
4  import org.apache.hadoop.io.Text;
5  import org.apache.hadoop.mapreduce.Job;
6  import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
7  import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
8  import org.apache.hadoop.util.Tool;
9  import org.apache.hadoop.util.ToolRunner;
10
11 public class WordCounterDriver extends Configured implements Tool{
12
13   public static void main(String[] args) throws Exception {
14
15       int res = ToolRunner.run(new WordCounterDriver(), args);
16       System.exit(res);
17   }
18
19   public int run(String[] args) throws Exception
20   {
21       Job job = Job.getInstance(getConf(),"WordCounter");
22       job.setJarByClass(this.getClass());
23
24       FileInputFormat.addInputPath(job, new Path(args[0]));
25       FileOutputFormat.setOutputPath(job, new Path(args[1]));
26
27       job.setMapperClass(WordCounterMapper.class);
28       job.setReducerClass(WordCounterReducer.class);
29
30       job.setOutputKeyClass(Text.class);
31       job.setOutputValueClass(IntWritable.class);
32
33
34     return job.waitForCompletion(true) ? 0: 1;
35
36   }
37 }
```

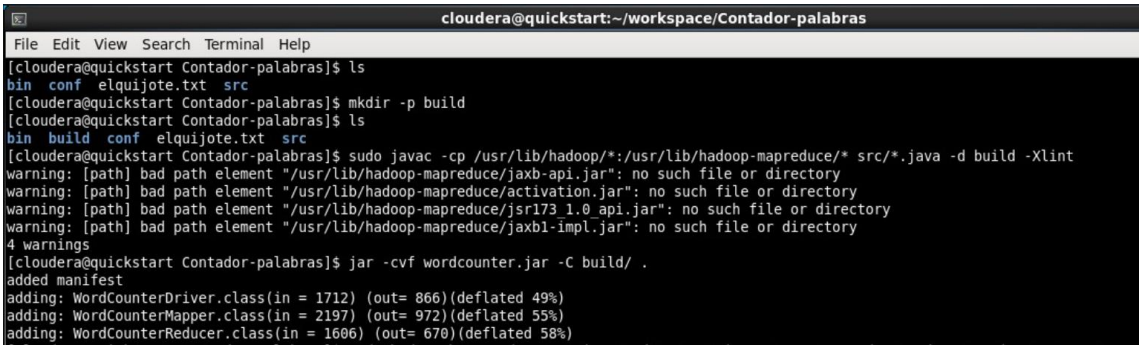
Una vez creados los scripts, deberemos abrir la consola de comandos para configurar hadoop. Empezaremos por crear una carpeta con nuestro usuario y darle permisos con -chow.



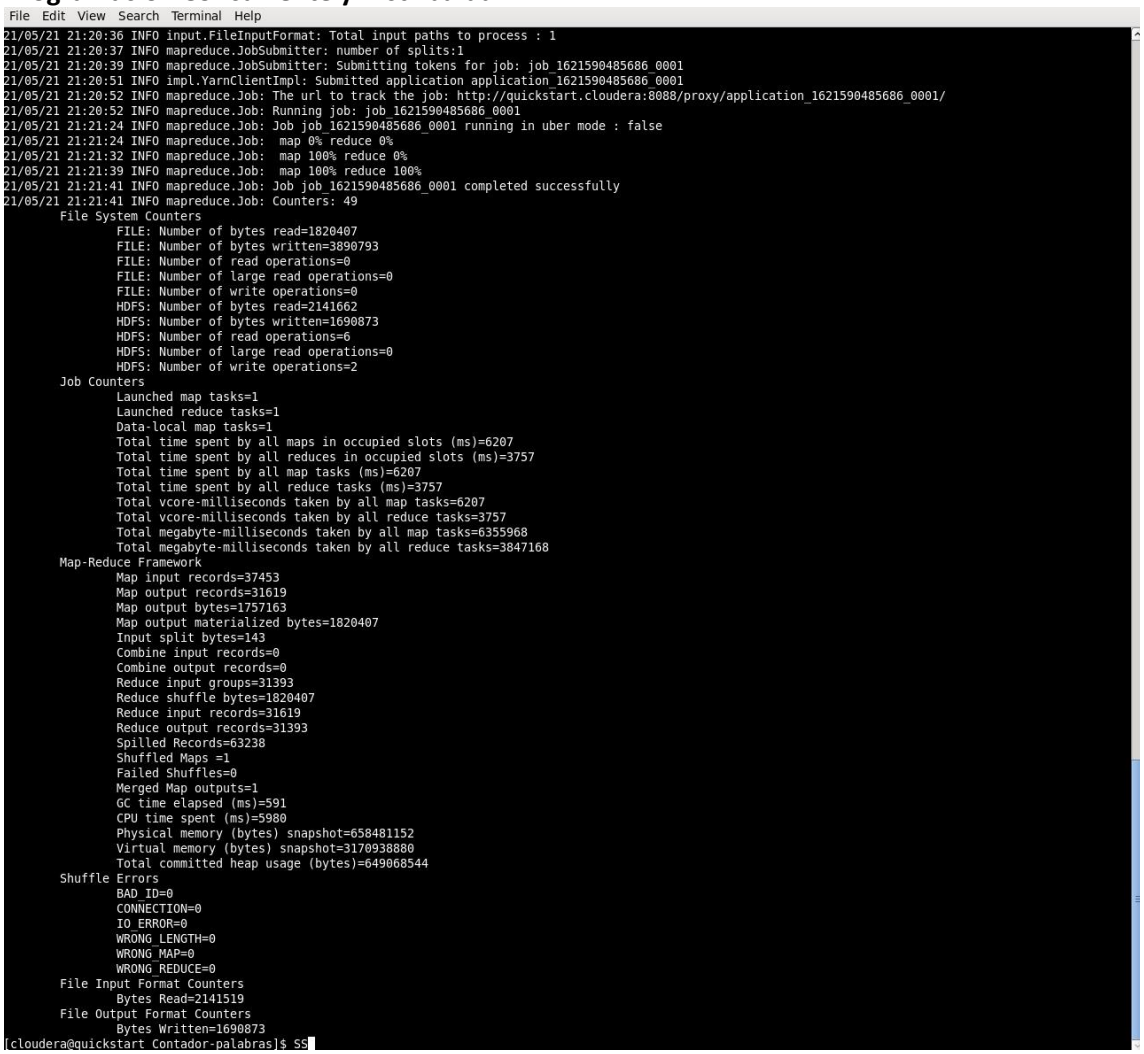
Después, buscaremos el texto al queremos aplicar map reduce (el quijote en este caso) y lo guardaremos en una carpeta local.



Una vez guardado, iremos al directorio donde se encuentre nuestro proyecto de ecplise desde la consola y haremos una build del proyecto llamada “wordcounter.jar.

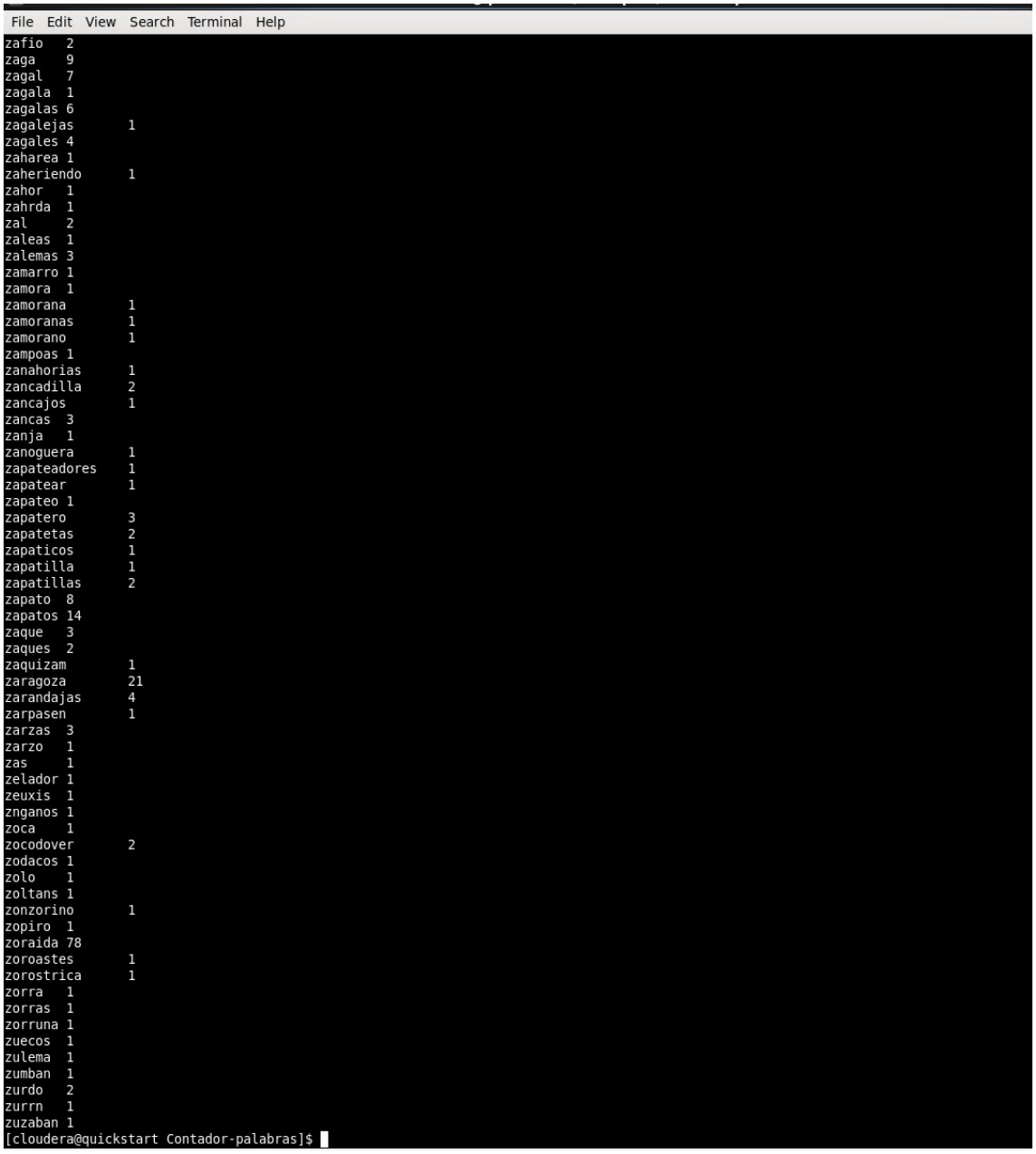


Una vez creada la build, la ejecutaremos desde hadoop, pasando como parámetros el texto del quijote (previamente guardado en /user/somozadev/input) y el directorio destino donde queremos guardar el resultado del mapreduce (/user/somozadev/wordcounter/output).

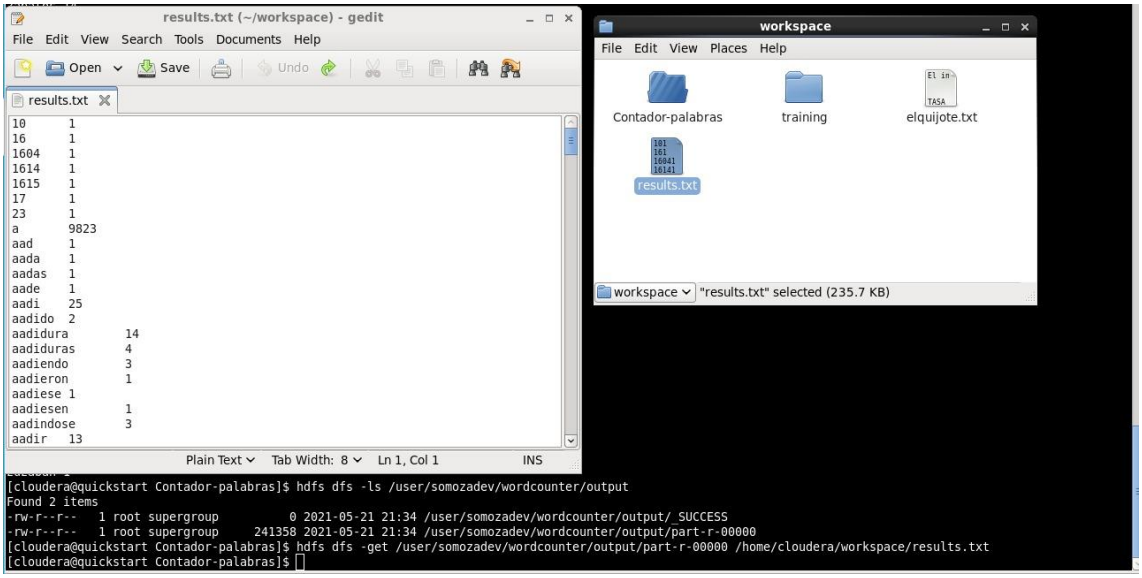


Y como se puede ver en la captura anterior, se ha realizado el mapreduce de forma correcta. Se conecta al resourcesManager, ejecuta el trabajo y empiezan los procesos de map y reduce. Dado que solo tenemos un único modo, los trabajos tienen que hacerse secuencialmente (se ve en la captura que primero se realizan el 0% de las tareas map y 0% de las tareas reduce. Luego pasa a 100% de tareas map y 0% de tareas reduce y finalmente 100% de cada). Es decir, se trata de una pseudo distribución.

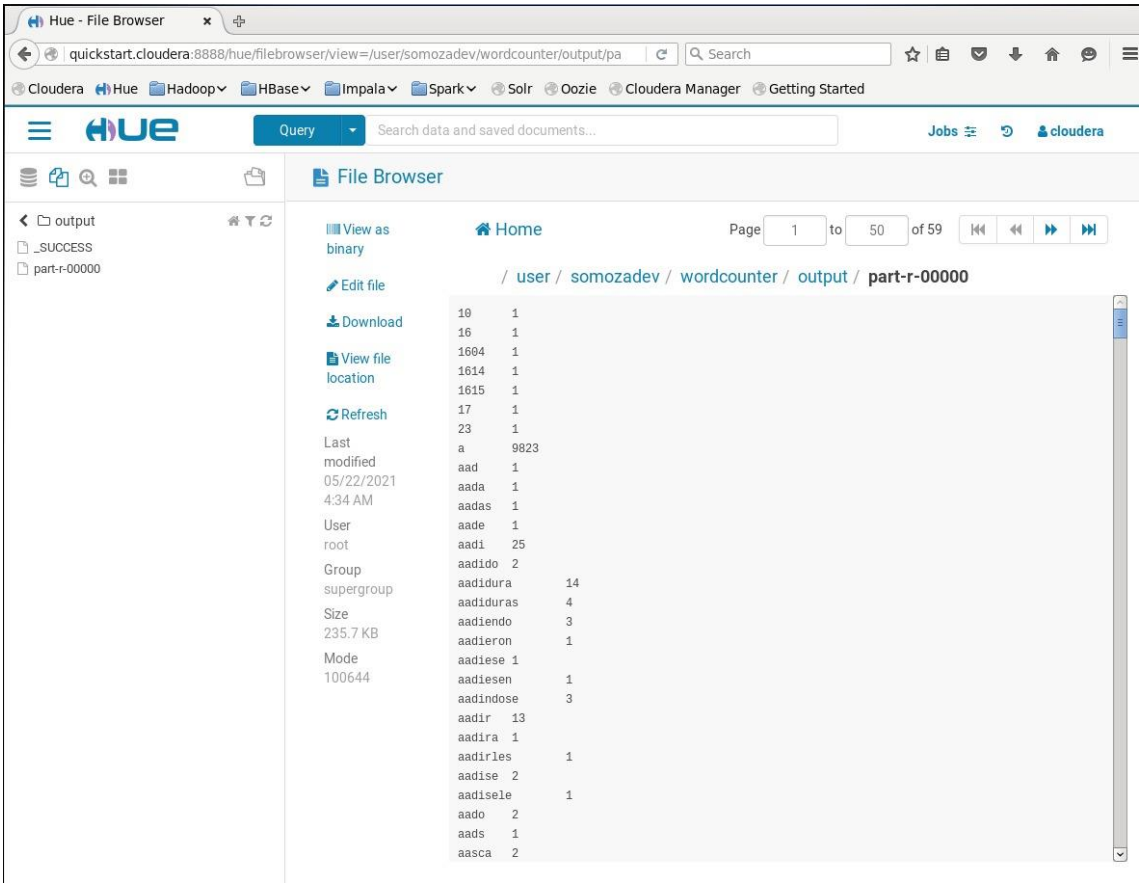
Finalmente, para ver el output podemos usar el comando hadoop fs -cat con el directorio para ver el contenido.



De todas formas, este output no se encuentra de manera local en nuestro equipo, para poder traerlo a nuestra máquina debemos ubicarlo en el servidor hadoop y usar hdfs dfs -get [directorio del output] [directorio donde se desee guardar en local].



O si se prefiere, acceder al hue en nuestro navegador (iniciando sesión con usr: cloudera y pass: cloudera) donde podemos acceder a los directorios hdfs y encontraremos de una manera visual el directorio de salida del fichero.



Finalmente, si se desearía volver a realizar la ejecución de la aplicación, se deberá eliminar el directorio de salida para que no haya problemas a la hora de sobreescribir en él.

```
[cloudera@quickstart ~]$ sudo hadoop fs -rm -r -f /user/somozadev/wordcounter/ou
tput
Deleted /user/somozadev/wordcounter/output
[cloudera@quickstart ~]$
```