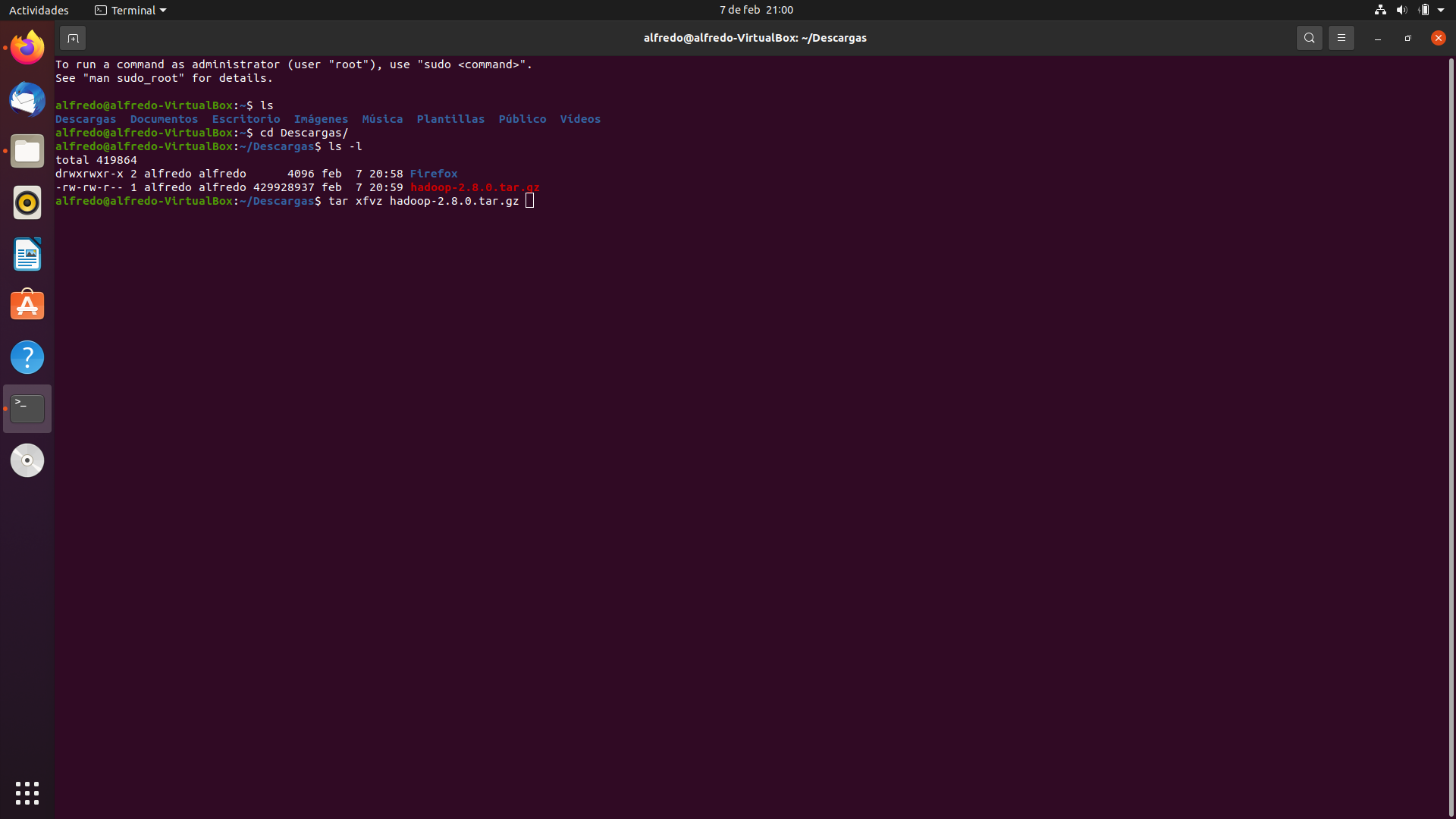
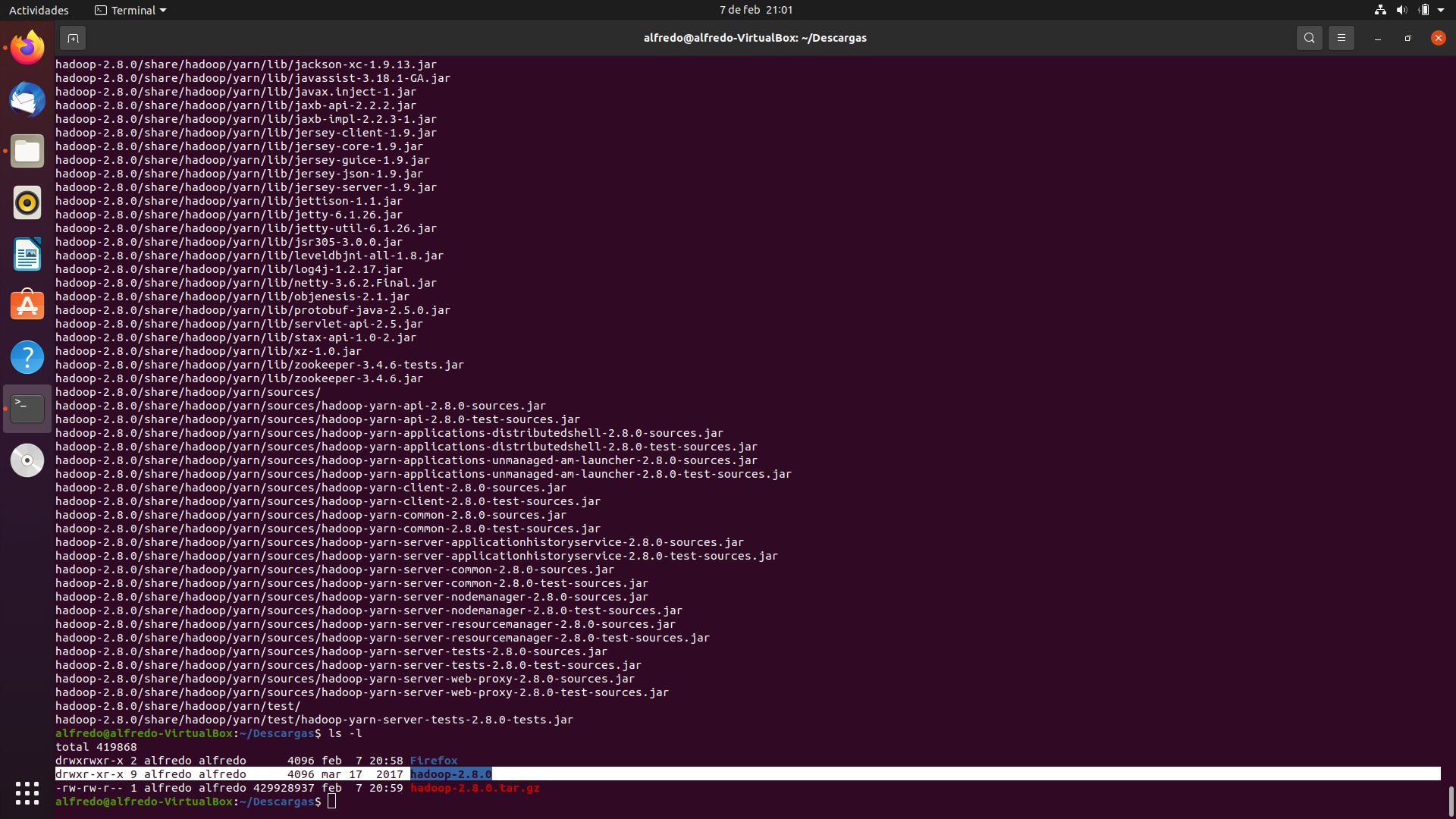
Alfredo Talavera Ramajo

*Práctica configuración de Hadoop en Ubuntu*

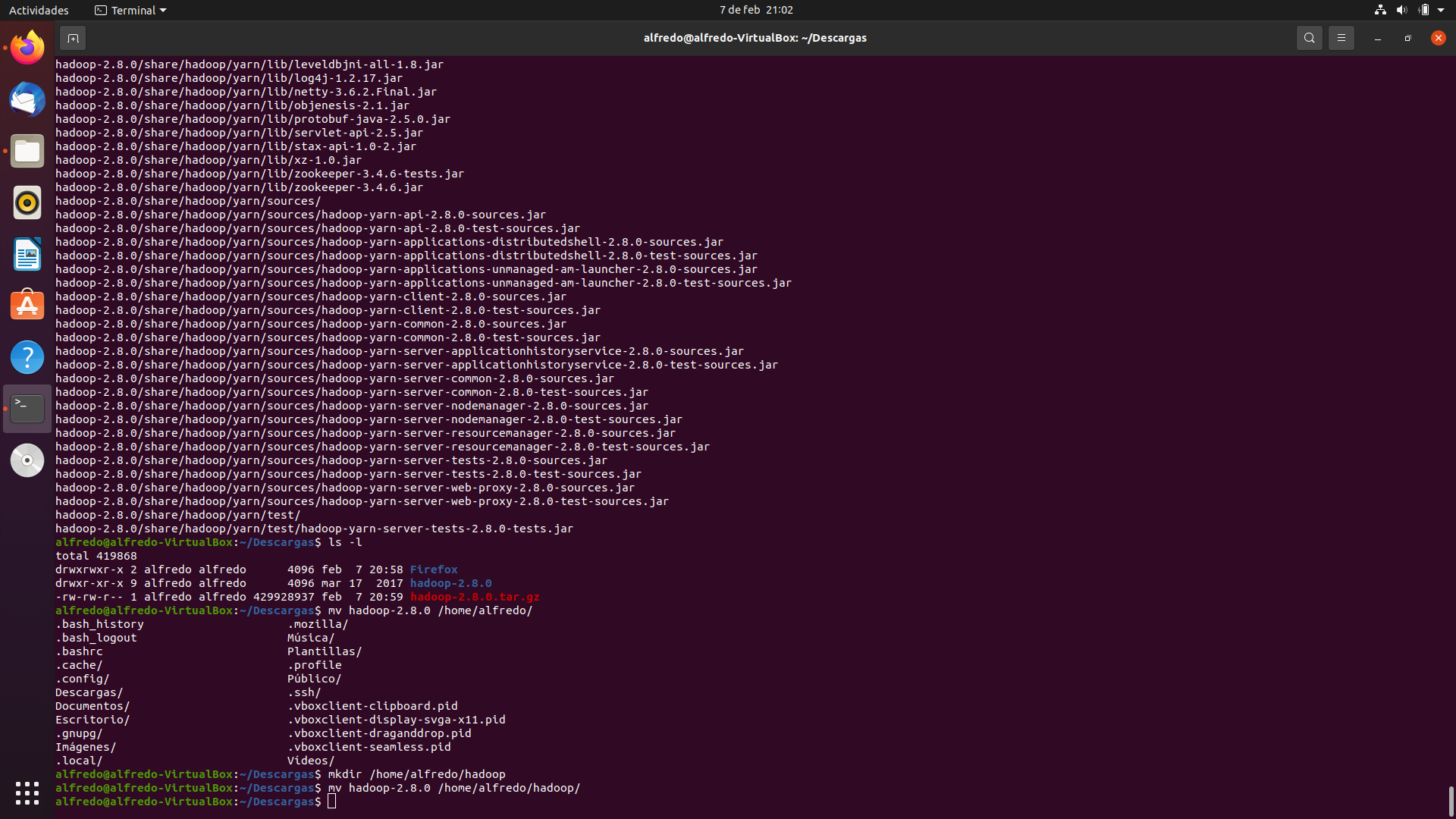
1.- Descomprimimos el archivo hadoop en su versión 2.8.0 con el comando: *tar xfvz*



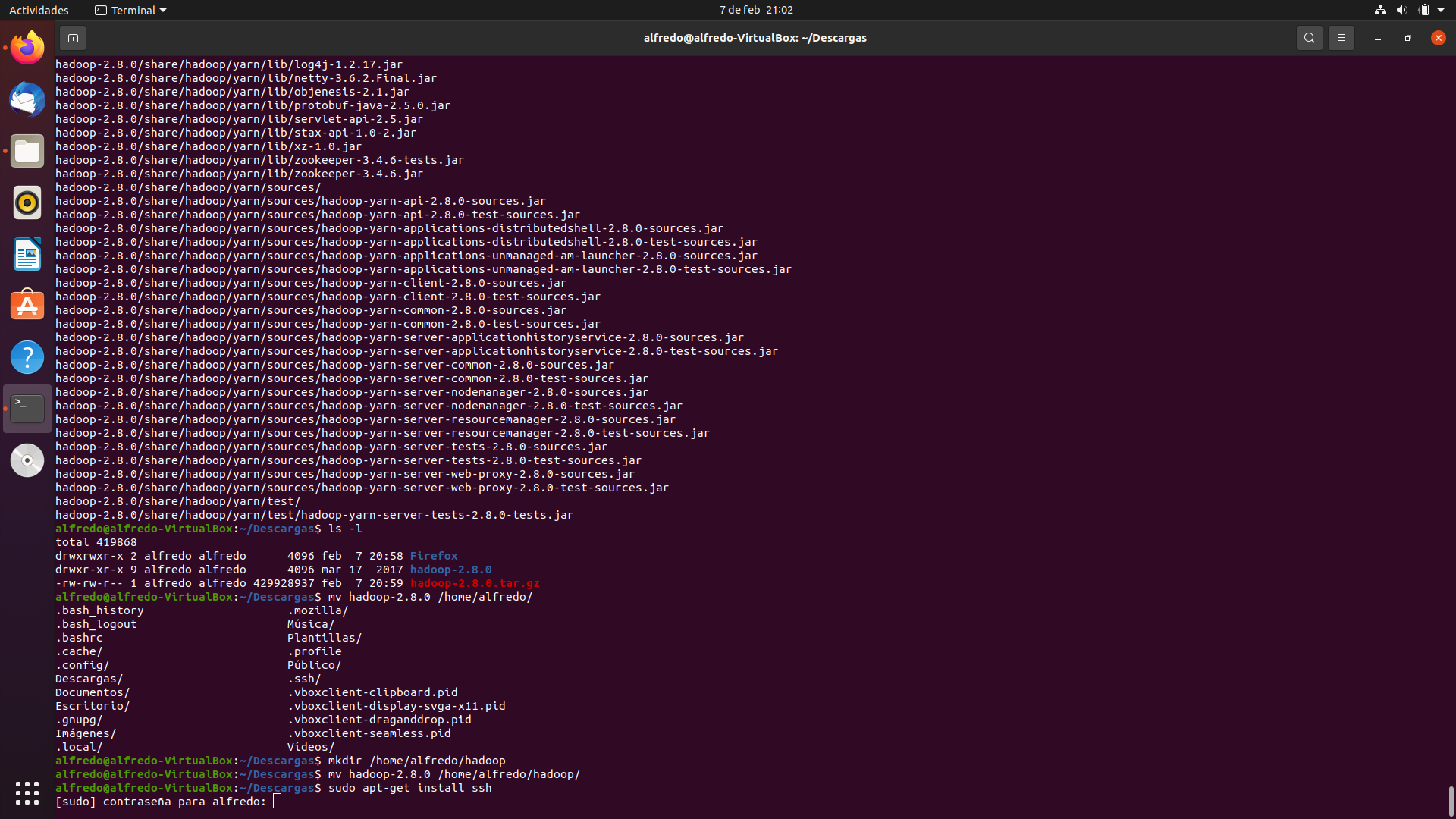


2.- Movemos lo que ha creado al descomprimir a la carpeta hadoop alojado en */home/agora/hadoop* con *mv*

*(Como queremos tener todo el contenido dentro de una carpeta llamada Hadoop creamos previamente, la cual tenemos que tener en cuenta en un futuro, si no la queremos solo tenemos que cambiarle el nombre con un cp hadoop-2.8.0 hadoop, que es copy de la carpeta hadoop-2.8.0 a hadoop, estando en la ruta /home/nombre\_usuario/, si no se tendría que utilizar rutas absolutas y quedaría como: cp /home/nombre\_usuario/hadoop-2.8.0 /home/nombre\_usuario/hadoop)*



3.- Instalamos ssh con sudo *(permisos de superuser)* *apt-get install ssh*

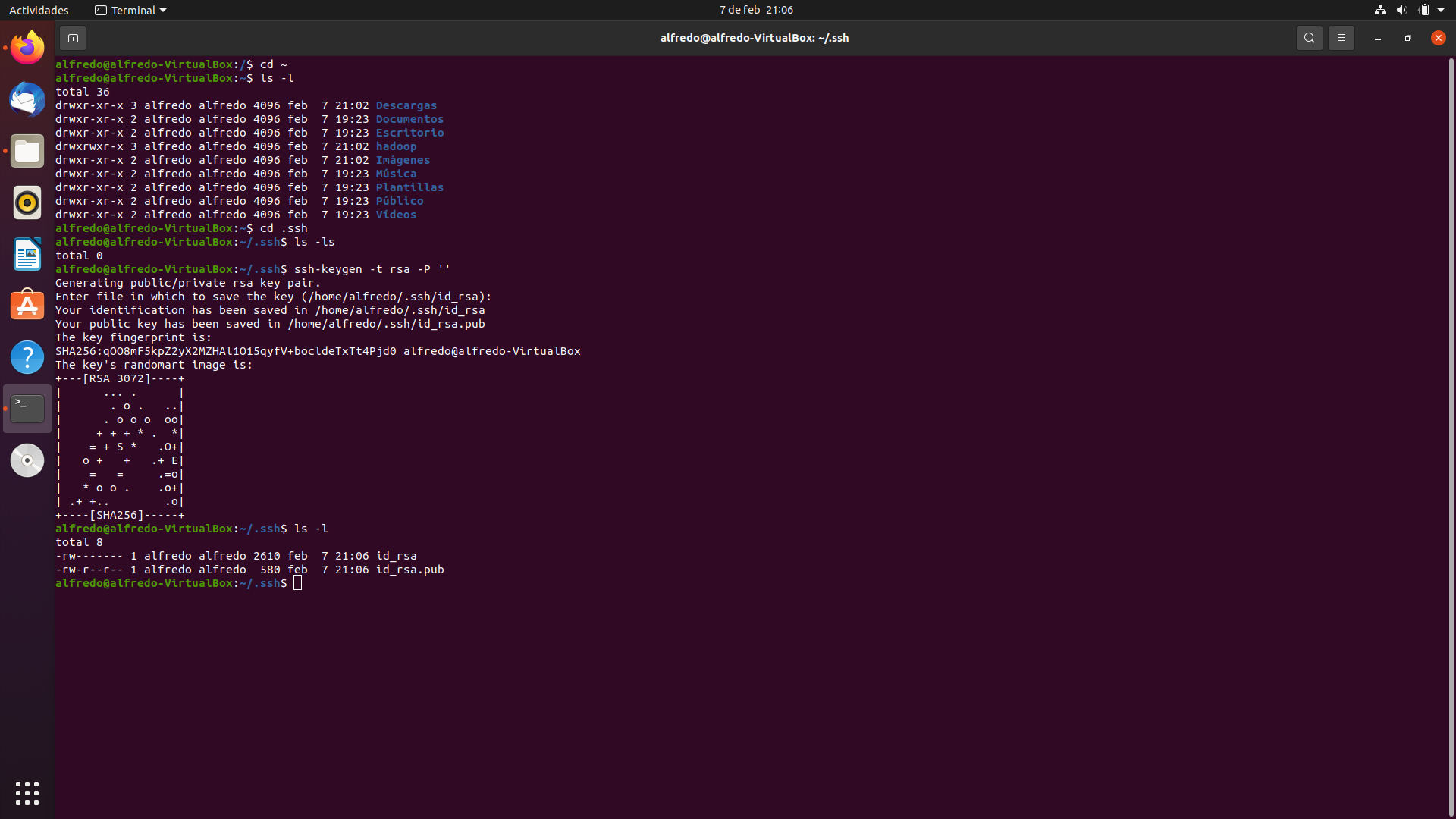


***\* Nos tendremos que mover a la carpeta /.ssh (****cd /.ssh****)***

***\*\* El . delante del nombre de un archivo o una carpeta significa que está oculto, pero con un ls -l podremos visualizarlos***

4.- Creamos una keygen dentro de la carpeta .ssh con el comando: *ssh -keygen -t rsa -P ‘’*

***· Le daremos a enter cuando nos pida el archivo y la identificación (porque asi por defecto lo guardará en ./ssh/id\_rsa) y nos generará la key***

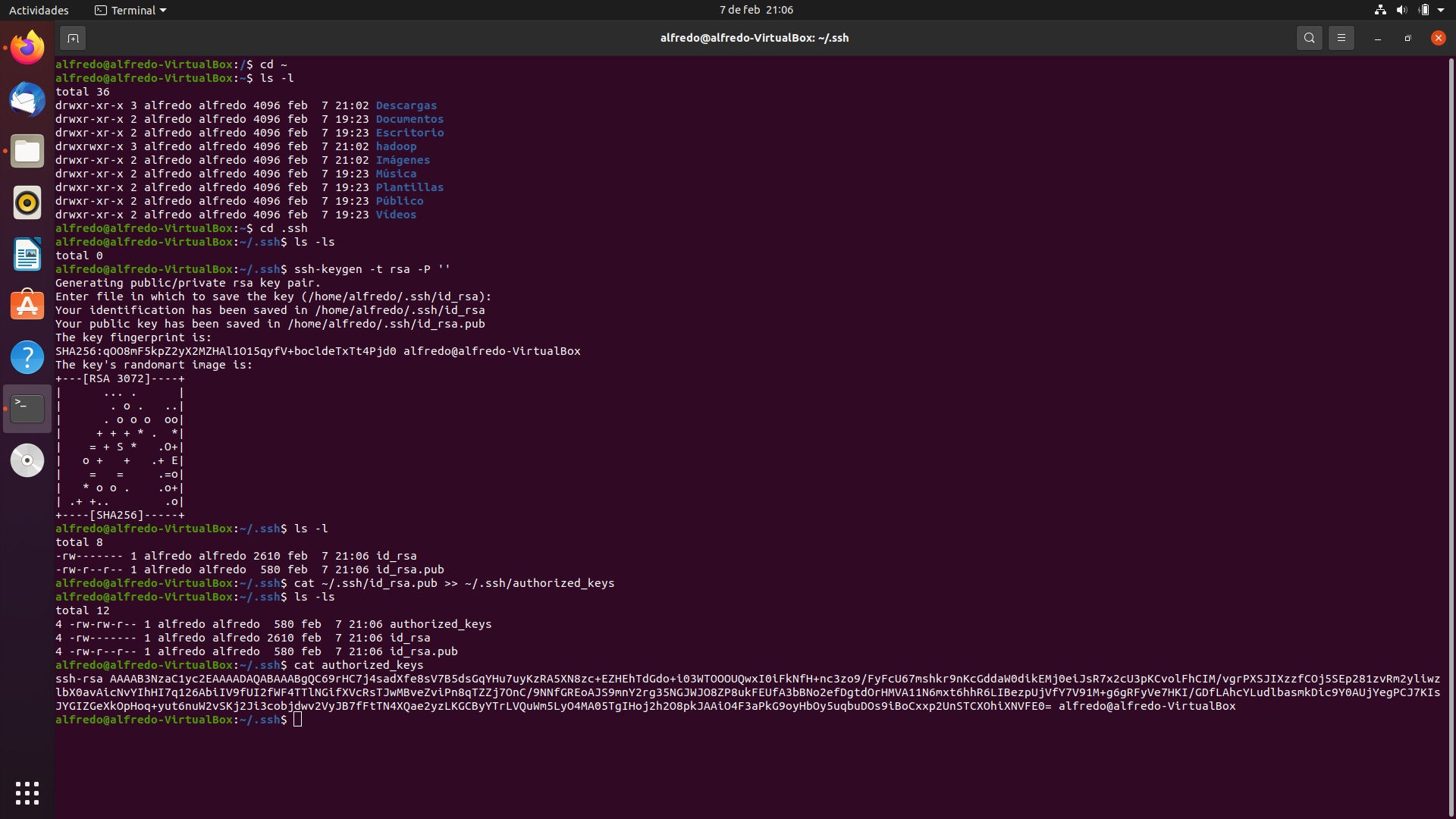


5.- Con el comando cat y el << lo que hacemos el volcar el contenido que tiene el archivo id.rsa.pub alojado en ~/.ssh/ y lo volcamos en el archivo authorized\_keys

*cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys*

***La ~ significa que nos estamos refiriendo al directorio personal del usuario, es decir, a: /home/nombre\_usuario/***

y con cat visualizamos el archivo authorized\_keys para ver que todo esté correcto



6.- Editamos el archivo.bashrc

*sudo nano ~/.bashrc*

y pegamos al final de todo:

**### VARIABLES INICIO HADOOP**

**export JAVA\_HOME=/usr/lib/jvm/nombre\_javajdk\_amd64** *(Hay que cambiarlo dependiendo de la versión de las jdk que se tenga, para comprobarlo solo basta con ir a la carpeta con: cd /usr/lib/jvm/ y hacer un: ls -l y copiamos y pegamos el nombre)*

**export HADOOP\_HOME=/home/nombre\_usuario/hadoop/hadoop-2.8.0** *(Lo que dijimos al principio)*

**export HADOOP\_INSTALL=$HADOOP\_HOME**

**export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME**

**export HADOOP\_COMMON\_HOME=$HADOOP\_HOME**

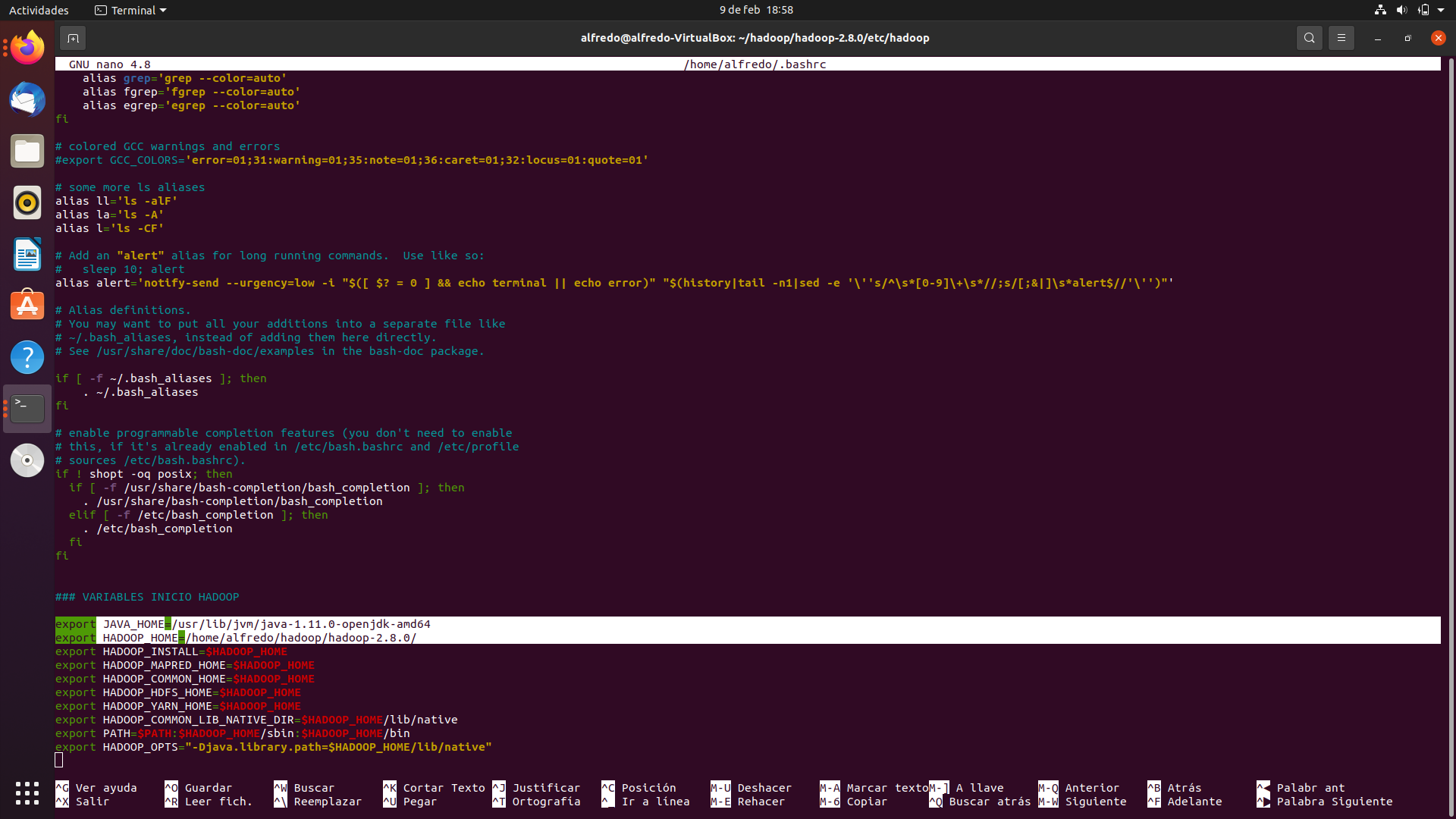
**export HADOOP\_HDFS\_HOME=$HADOOP\_HOME**

**export HADOOP\_YARN\_HOME=$HADOOP\_HOME**

**export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native**

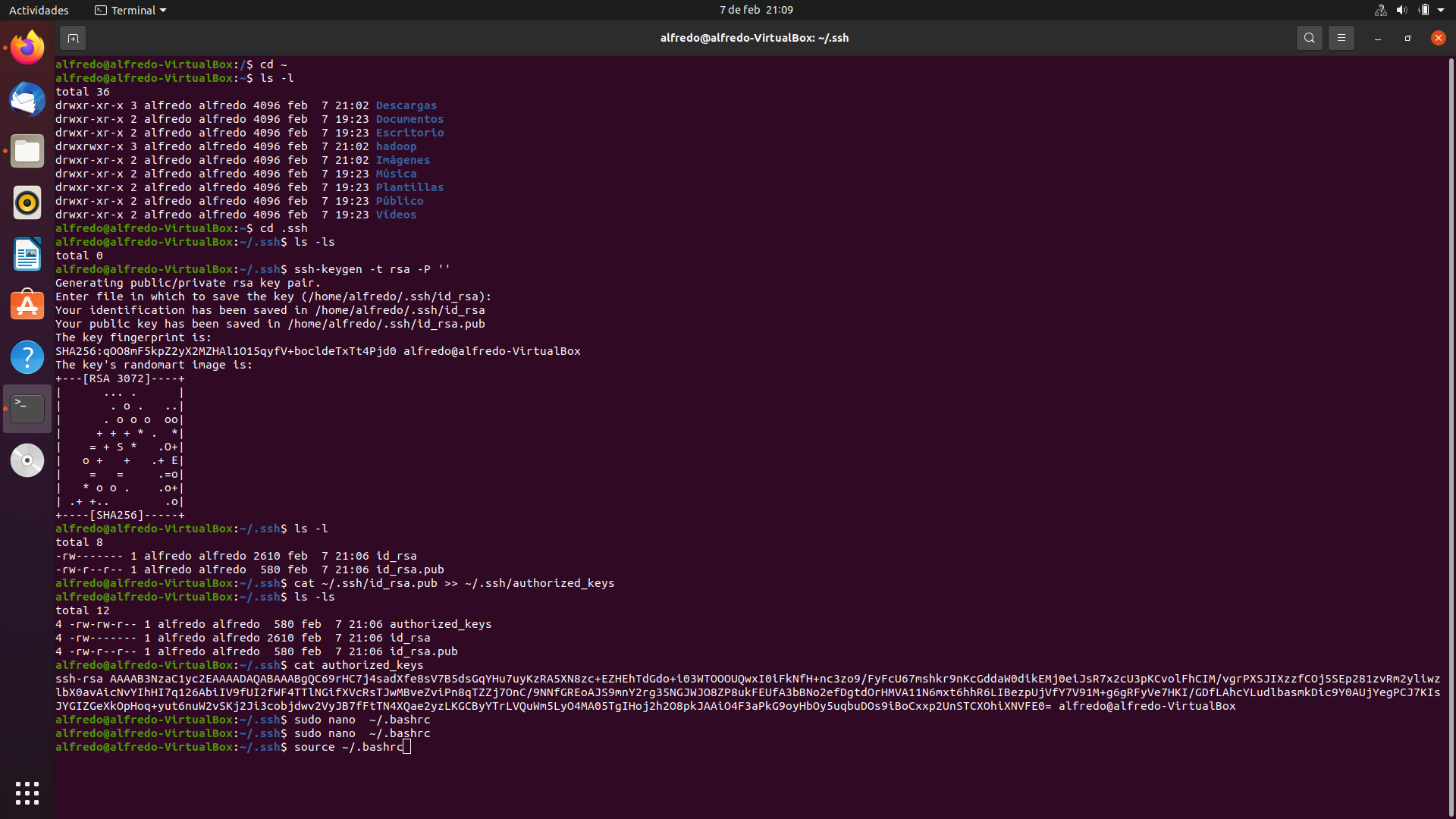
**export PATH=$PATH:$HADOOP\_HOME/sbin:$HADOOP\_HOME/bin**

**export HADOOP\_OPTS="-Djava.library.path=$HADOOP\_HOME/lib/native"**



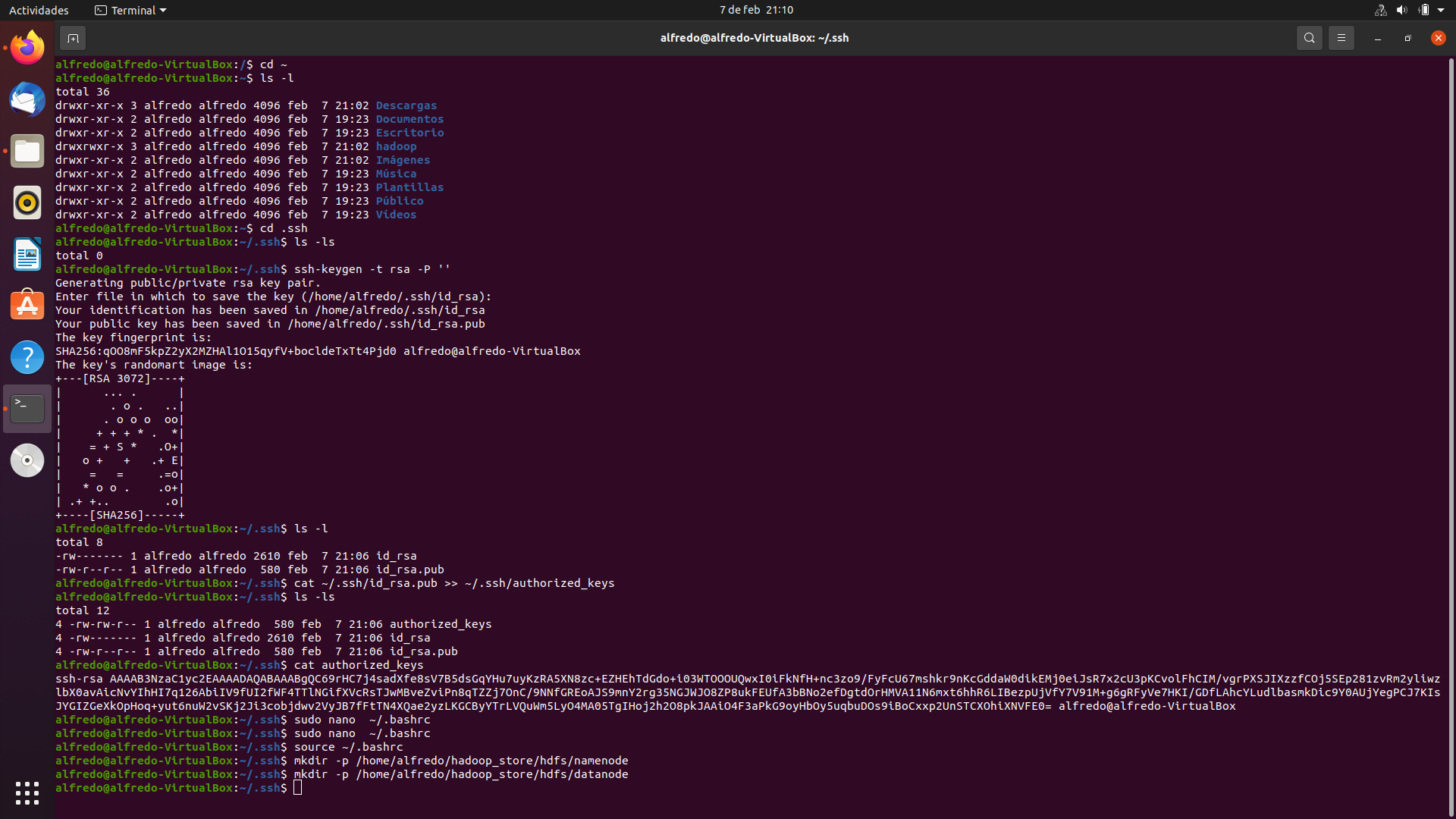
7.- Cargamos los cambios del archivo basrc con source

*source ~/.bashrc*

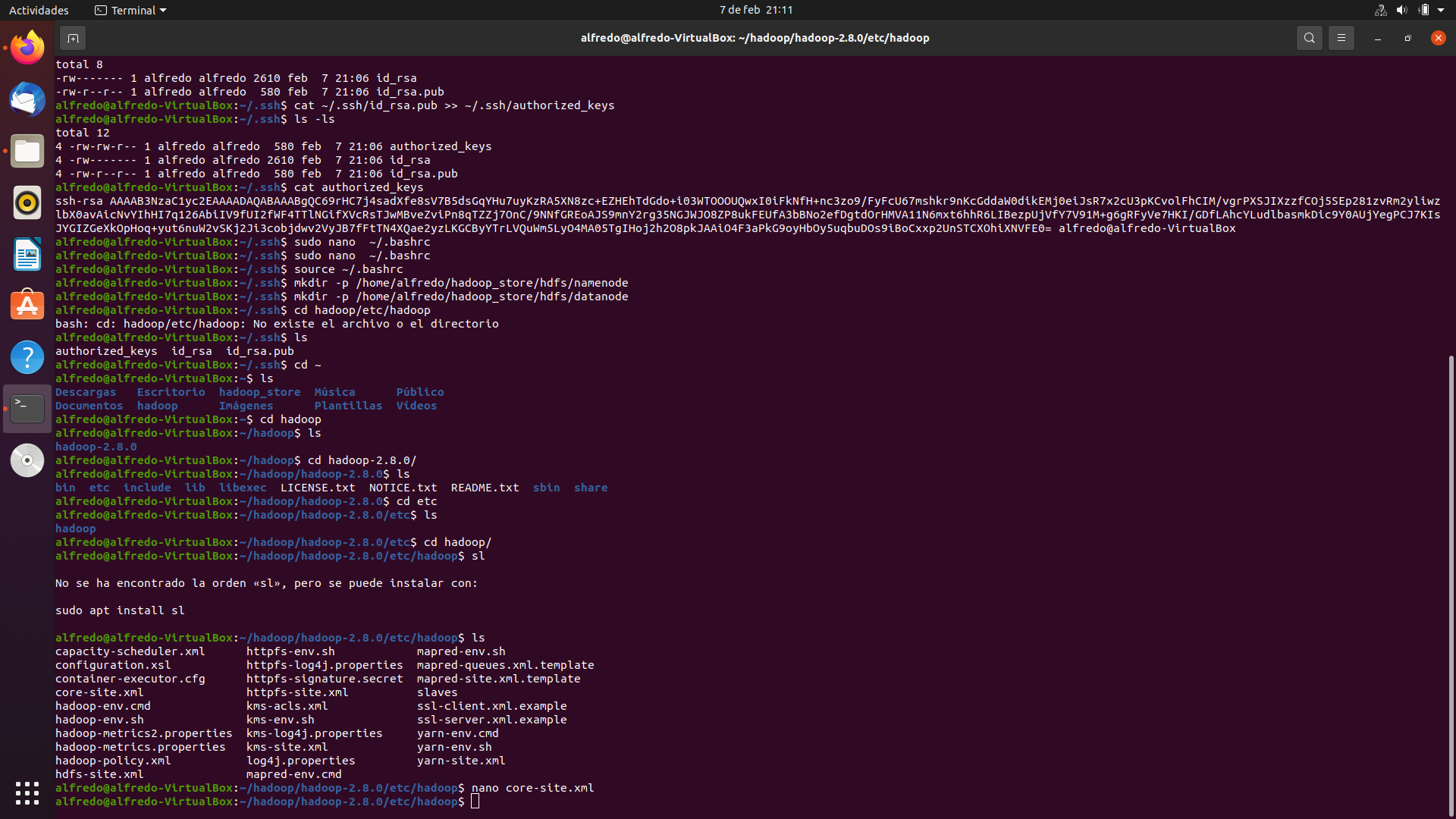


8.- Creamos las carpetas namenode y datanode con el comando: *mkdir -p /home/nombre\_usuario/hadoop\_store/namenode*  y *datanode*

***\* \* El -p es por si no tenemos creadas algunas carpetas hijos las crea todas en la misma línea de comando***



9.- Nos dirijimos a la carpeta */hadoop/hadoop-2.8.0/etc/hadoop* y la listamos para ver los archivos que tiene



10.- Editamos el archivo *core-site.xml* con *nano* e introducimos:

<configuration>

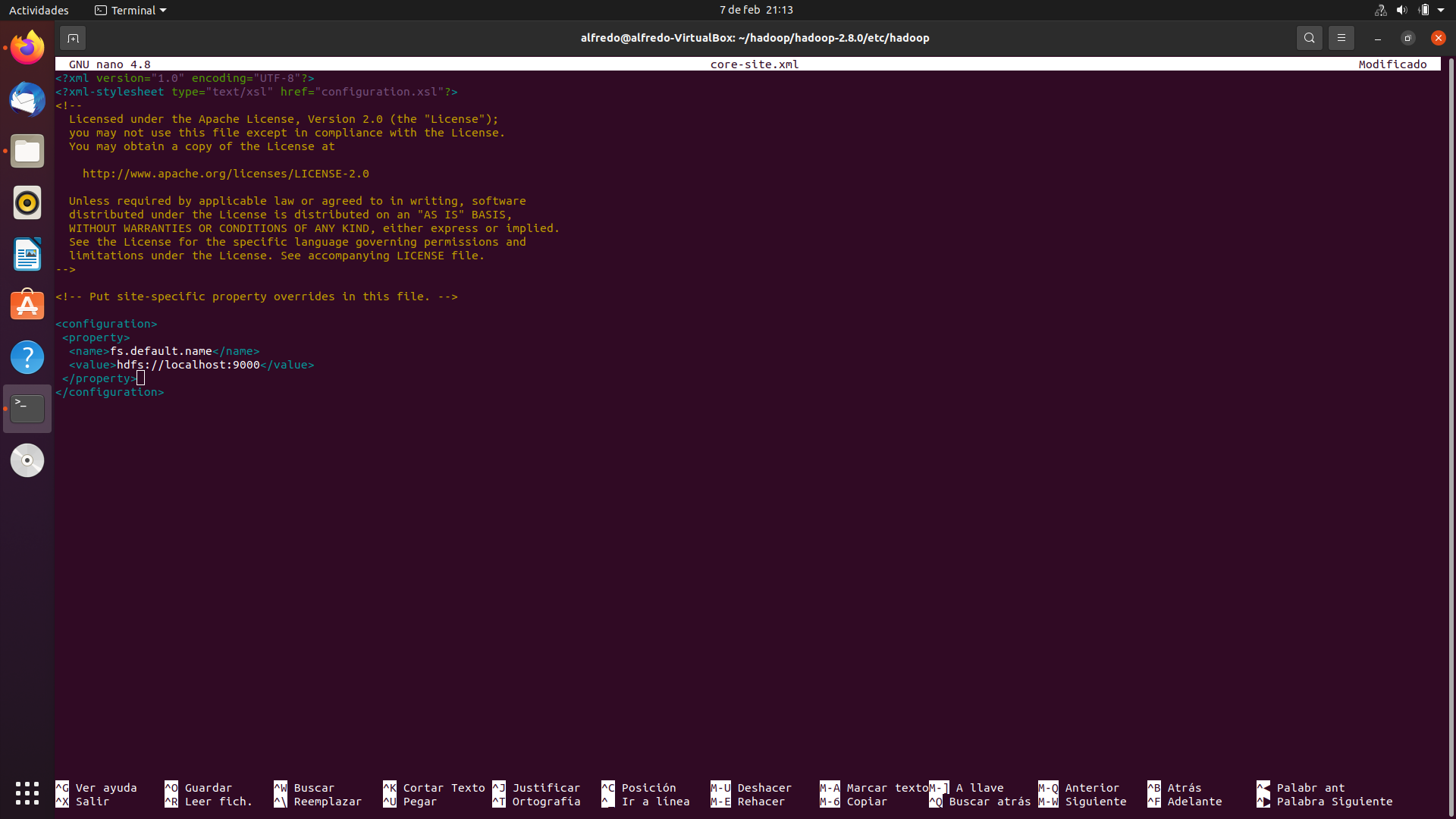
<property>

<name>fs.default.name</name>

<value>hdfs://localhost:9000</value>

</property>

</configuration>



11.- Editamos el archivo *hdfs-site.xml* con *nano* e introducimos:

<configuration>

<property>

<name>dfs.replication</name>

<value>1</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>file:/home/nombre\_usuario/hadoop\_store/hdfs/namenode</value>

***Introducimos la ruta de la carpeta namenode***

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>file:/home/nombre\_usuario/hadoop\_store/hdfs/datanode</value> ***Introducimos la ruta de la carpeta datanode***

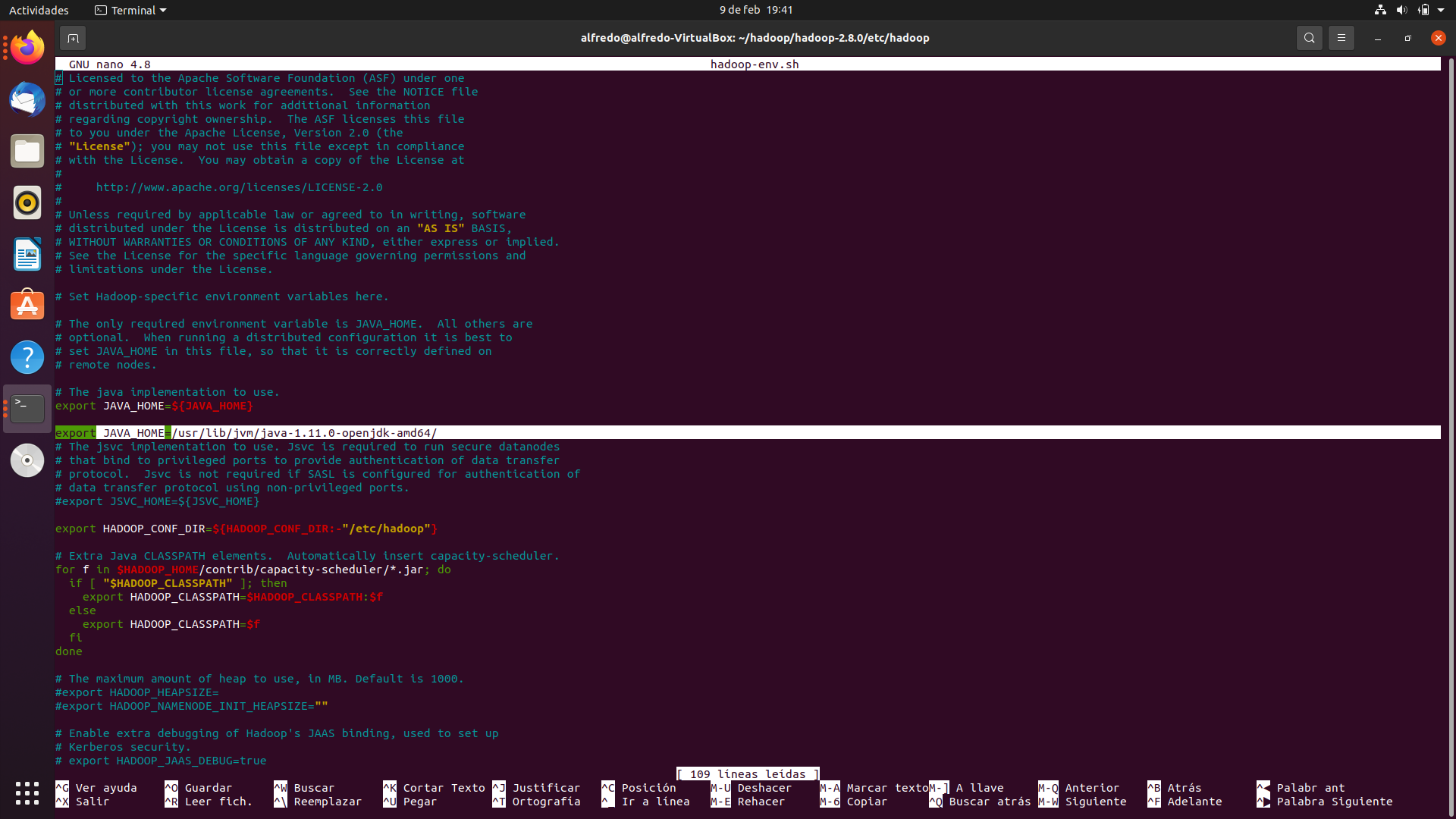
</property>

</configuration>



12.- Abrimos el archivo *hadoop-env.sh* e introducimos la siguiente sentencia:

*export JAVA\_HOME=/usr/lib/jvm/java-1.11.0-openjdk-amd64/ (Especificamos las jdk que tengamos nosotros en concreto)*



13.- Abrimos el archivo *mapred-site*.xml e introducimos:

<configuration>

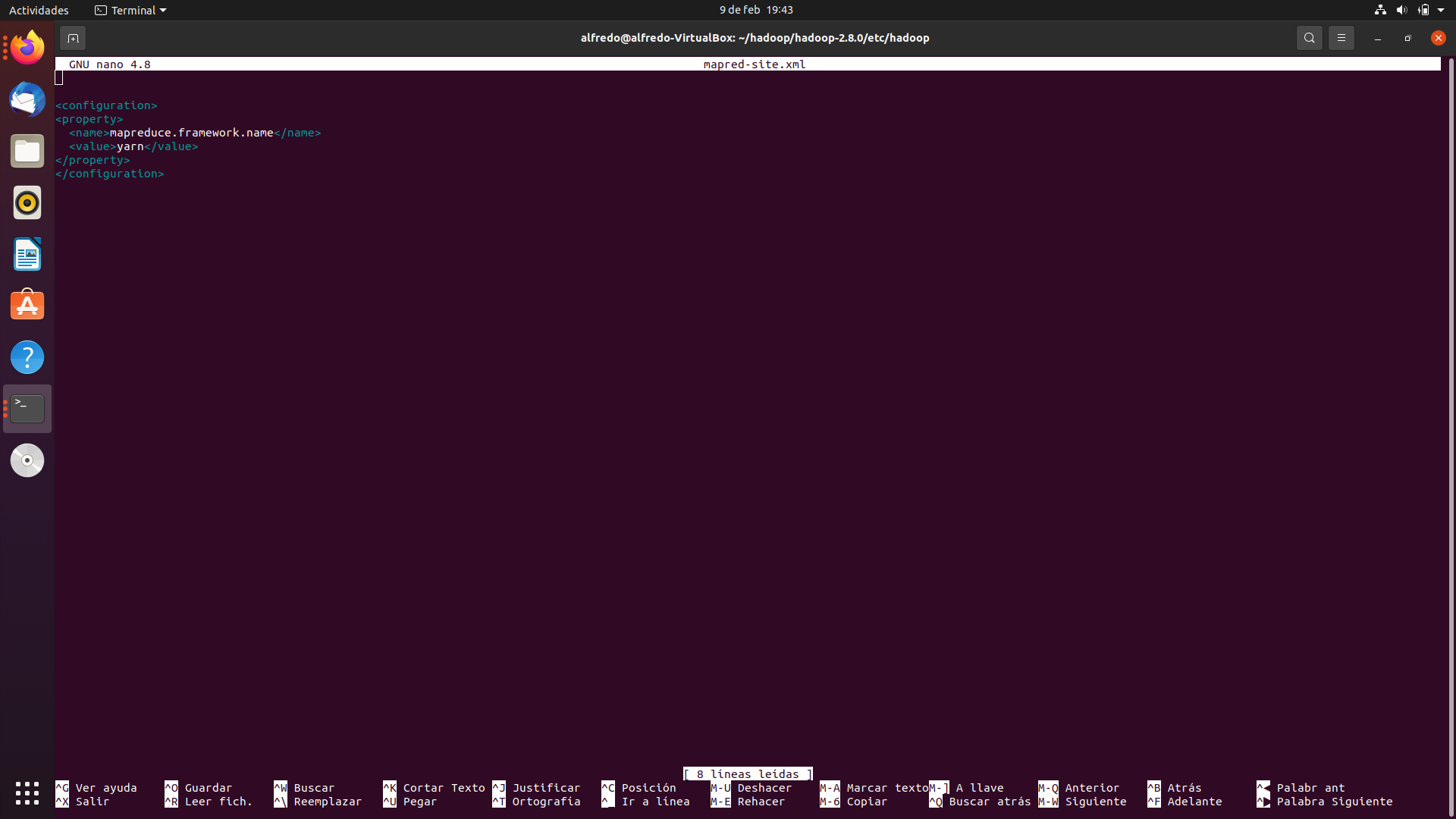
<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

</configuration>



14.- Abrimos y editamos el archivo yarn-site.xml e introducimos:

<configuration>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

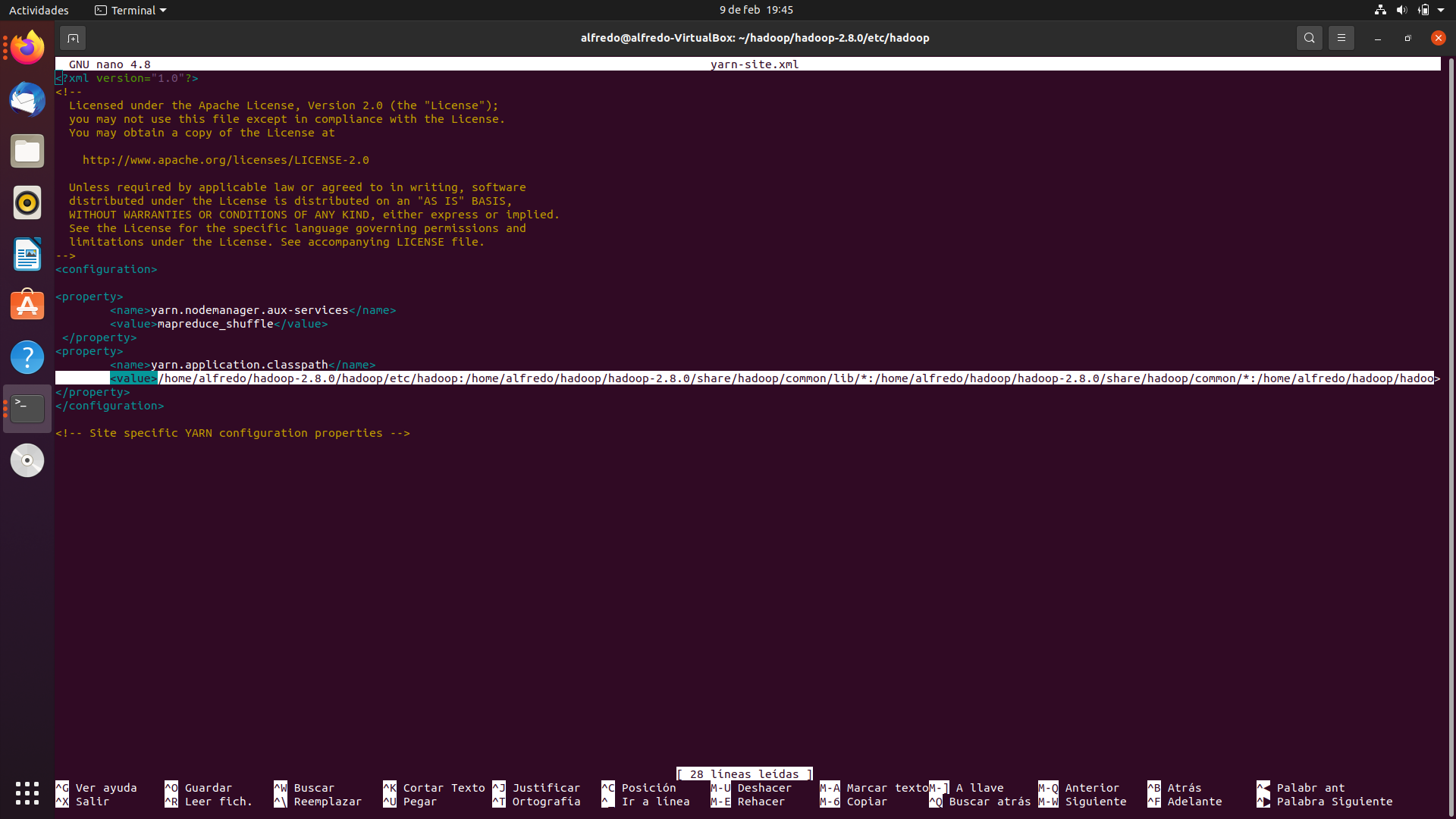
<property>

<name>yarn.application.classpath</name>

<value>/home/alfredo/hadoop-2.8.0/hadoop/etc/hadoop:/home/nombre\_usuario/hadoop/hadoop-2.8.0/share/hadoop/common/lib/\*:/home/nombre\_usuario/hadoop/hadoop-2.8.0/share/hadoop/common/\*:/home/nombre\_usuario/hadoop/hadoop-2.8.0/share/hadoop/hdfs:/home/nombre\_usuario/hadoop/hadoop-2.8.0/share</value> – ***Tenemos que cambiar todas las rutas y ponerlas correctamente***

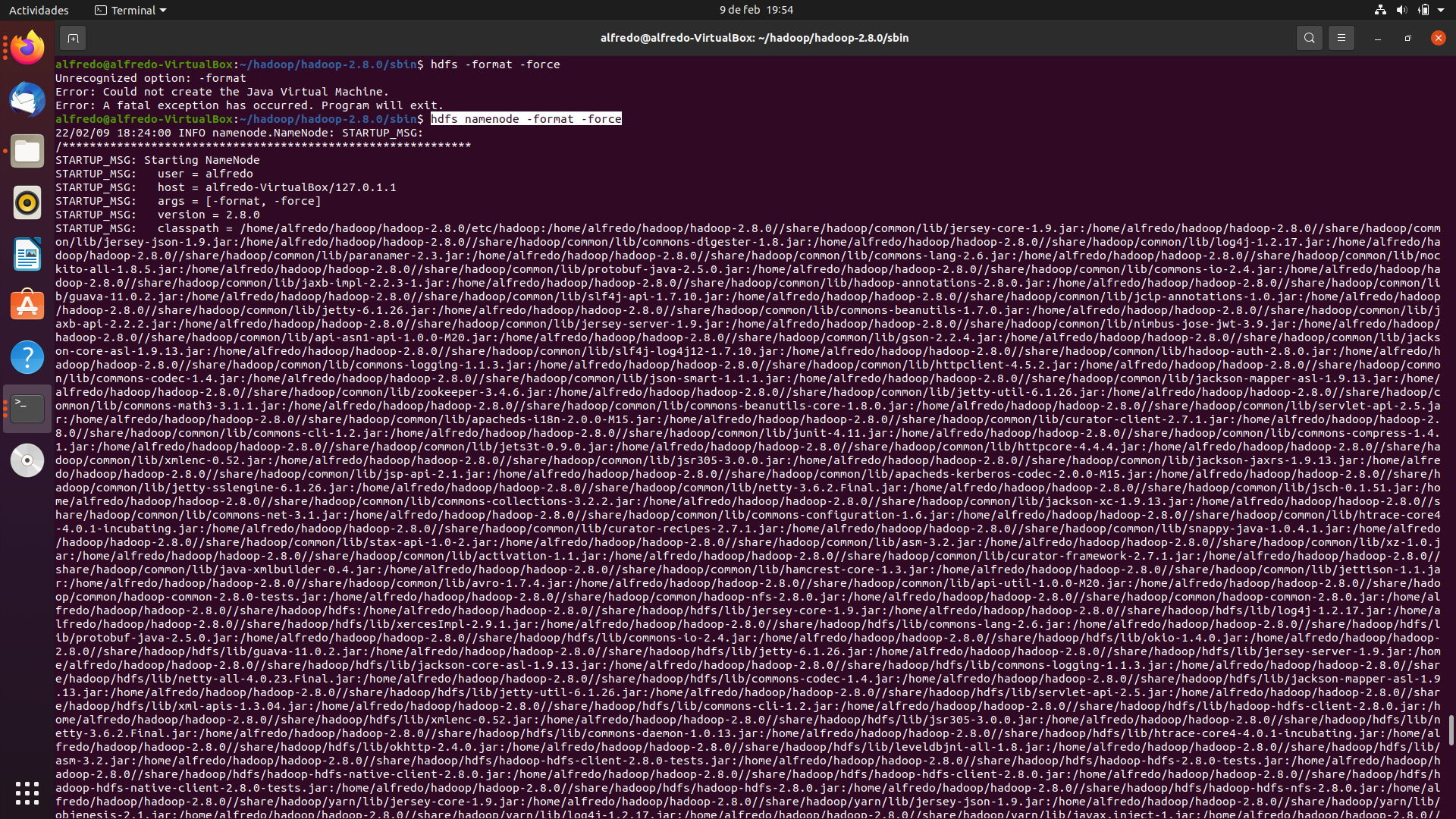
</property>

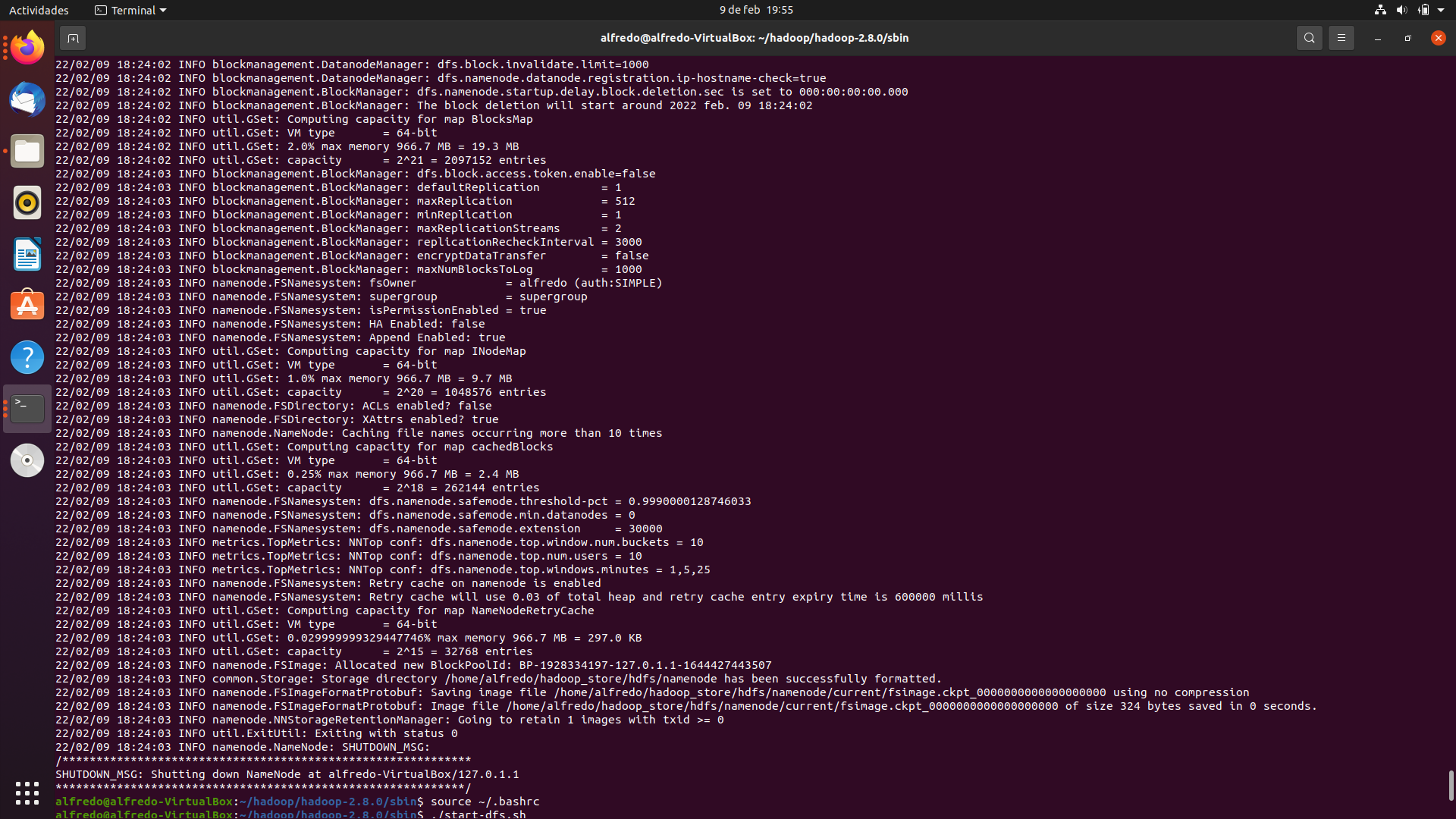
</configuration>



15.- Nos dirigimos a la carpeta *~/hadoop/hadoop-2.8.0/sbin* y ejecutamos el comando:

***hdfs namenode -format -force*** para recargar los cambios hechos en los archivos

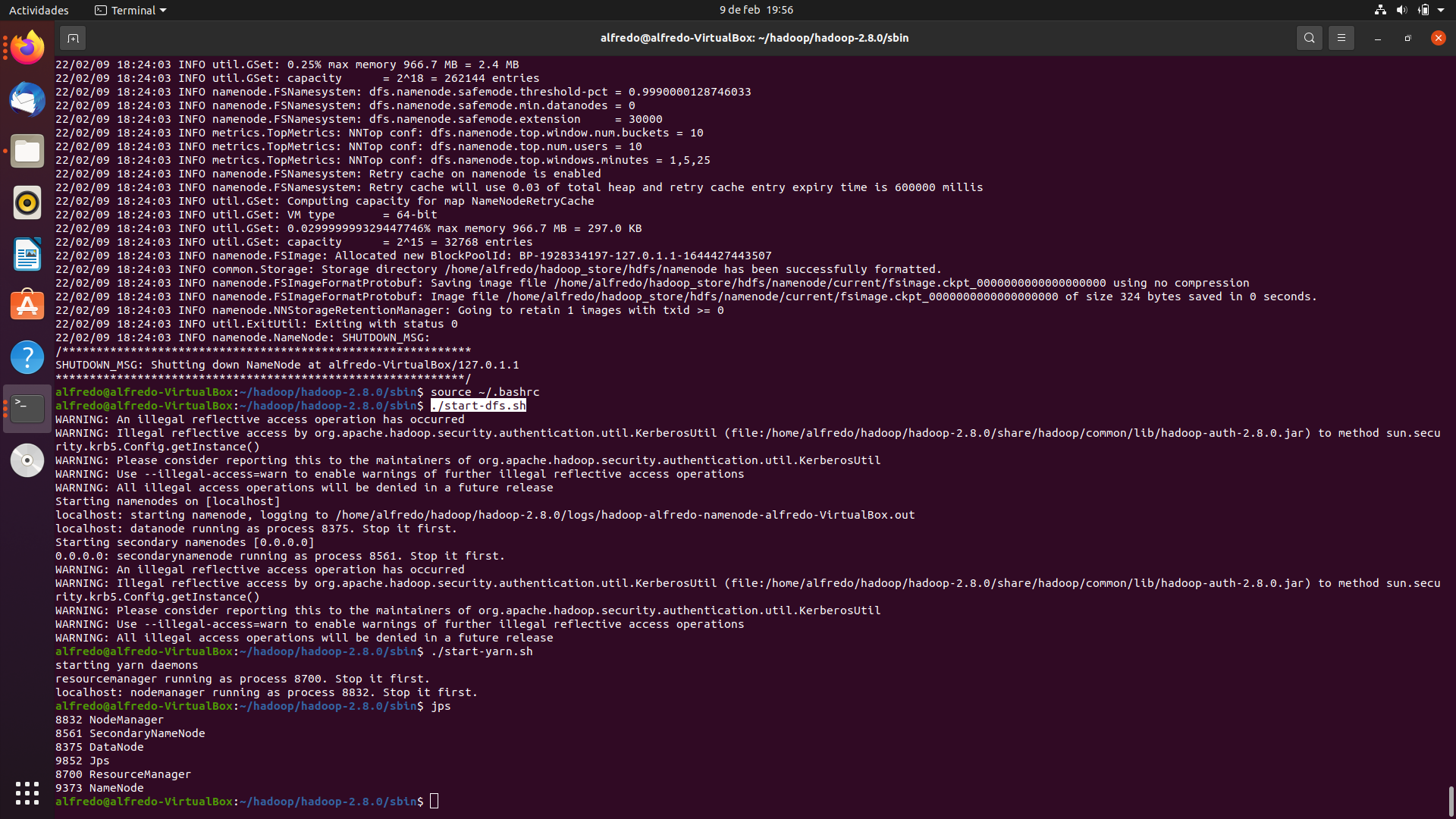




16.- Ya en la carpeta sbin tenemos que inciar los demonios y para ello tendremos que usar:

***./start-dfs.sh*** y ***./start-yarn.sh***

Para comprobar que todo es correcto solamente tenemos que utilizar: *jps*

**

Sabremos que todo es correcto cuando al ejecutar *jps* nos sale: *NodeManager, SecondaryNameNode, DataNode, Jps, ResourceManager y NameNode*