

Tarea de Vacaciones

Para esta tarea se deberá responder una serie de preguntas de temas que se han abordado en el curso, el objetivo consiste entonces en reforzar los conocimientos e indagar en otros nuevos que, aunque no forman parte explícita del temario, le servirán al estudiante para incursionar en temas Big Data a plenitud.

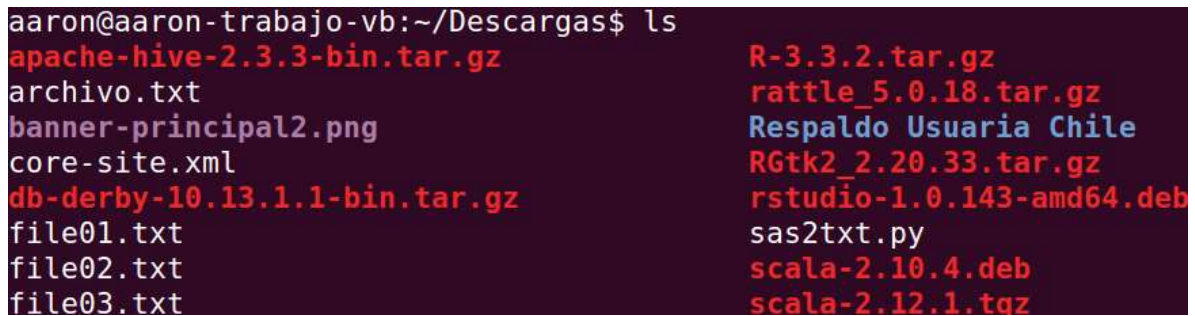
El estudiante debe crear primero que nada un directorio dentro de su directorio local de Git llamado **TareaVacaciones** y dentro de éste crear una copia de esta tarea que lleve por nombre **TareaX y colocarla juntos con los resultados en formato PDF**, donde X es su nombre de usuario empleado en Github. Por ejemplo:

TareaYoNoFui

Con respecto de los ejercicios, a menos que se indique lo contrario, todas las respuestas constarán del código o instrucción resultante acompañada de una captura de pantalla. Ejemplo:

-1.- Indique el comando que se emplea para listar archivos en GNU/Linux de manera simple:

Respuesta: **ls**



```
aaron@aaron-trabajo-vb:~/Descargas$ ls
apache-hive-2.3.3-bin.tar.gz      R-3.3.2.tar.gz
archivo.txt                      rattle_5.0.18.tar.gz
banner-principal2.png           Respaldo Usuaría Chile
core-site.xml                   RGtk2_2.20.33.tar.gz
db-derby-10.13.1.1-bin.tar.gz   rstudio-1.0.143-amd64.deb
file01.txt                     sas2txt.py
file02.txt                     scala-2.10.4.deb
file03.txt                     scala-2.12.1.tgz
```

En este tipo de preguntas de faltar alguno de los elementos señalados se considerará como errónea la respuesta y no se obtendrá el acierto.

Es menester mencionar que hay casos donde las preguntas son de tipo abierto, entonces en esos casos lo único que se pide adjuntar es tanto la respuesta como la(s) fuente(s). Ejemplo:

0.- ¿Cuál es el significado de la vida?

Respuesta: **42**

Fuente: <https://www.independent.co.uk/life-style/history/42-the-answer-to-life-the-universe-and-everything-2205734.html>

De nueva cuenta, si no existe al menos uno de estos dos elementos, la respuesta se considerará como inválida.

La fecha límite de entrega es el **Lunes 25 de Junio a las 15:00:00**, como se había mencionado con anterioridad el flujo de archivos se mantiene única y exclusivamente por Github, para ello se dejan los comandos a emplearse:

- **git pull** (para actualizar el repositorio)
- **git add .** (para indicar todos los elementos que se desean agregar al repositorio)
- **git commit -m "TareaVacaciones nombre_usuario"** (para colocar un mensaje que distinga a esta subida de las de los demás usuarios)
- **git push origin master** (para efectuar los cambios)

Por cierto que en lo que se repara Git en Cloudera puede ocupar Git de Windows y de esta manera, ya que se sugirió la instalación de Guest Additions en VirtualBox, copiar los resultados al primer sistema.

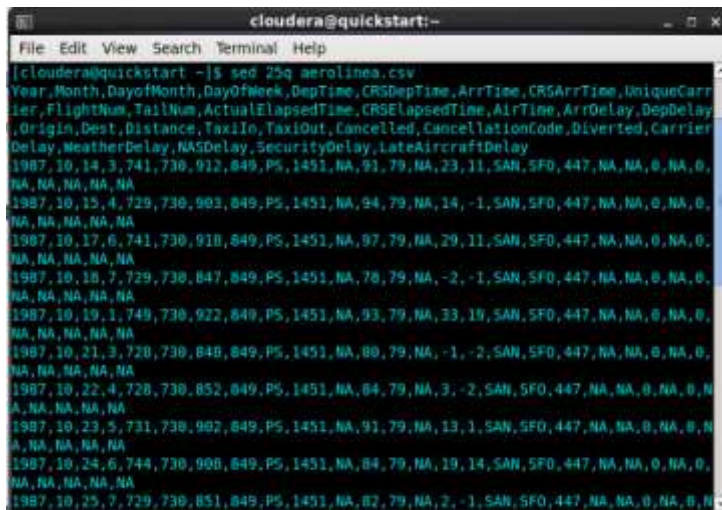
Nuevamente, de no cumplirse al menos uno de los señalamientos anteriores la tarea se considerará no entregada.

Dicho lo anterior se les desea mucho éxito en la travesía, cualquier cosa no duden en preguntar...

SECCION 1. GNU/LINUX

1.- Del archivo **aerolineas.csv** (el archivo descomprimido que todavía debería estar en su local y no el del HDFS) use comandos de GNU/Linux para obtener las 25 primeras líneas (incluyendo encabezado) **SIN** usar el comando head.

sed 25q aerolinea.csv



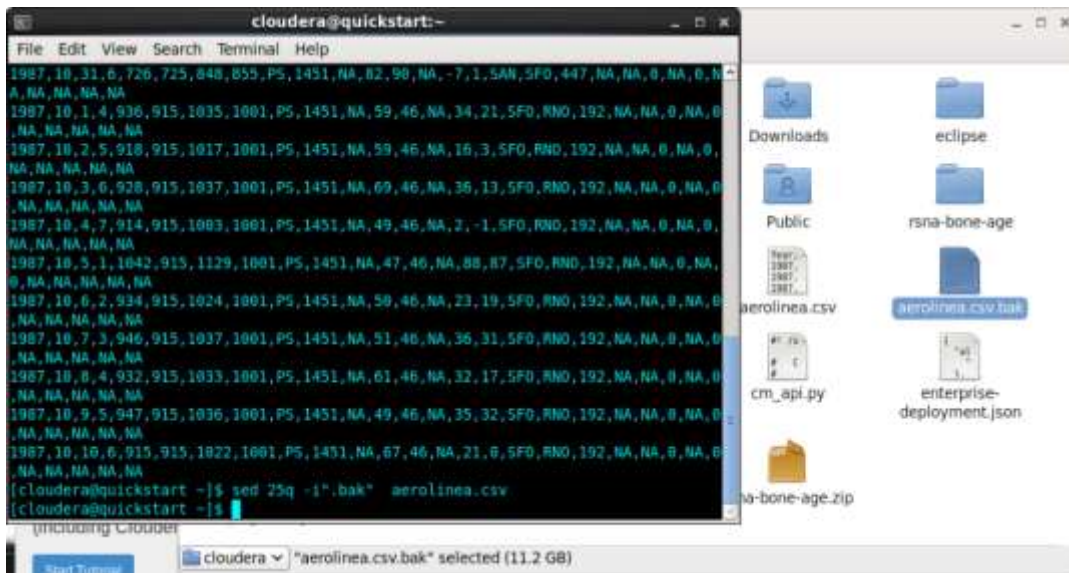
```

cloudera@quickstart:~$ sed 25q aerolinea.csv
Year,Month,DayOfMonth,DayOfWeek,DepTime,CRSDepTime,ArrTime,CRSArrTime,UniqueCarrier,FlightNum,TailNum,ActualElapsedTime,CRSElapsedTime,AirTime,ArrDelay,DepDelay,Origin,Dest,Distance,TaxiIn,TaxiOut,Cancelled,CancellationCode,Diverted,CarrierDelay,WeatherDelay,NASDelay,SecurityDelay,LateAircraftDelay
1987,10,14,3,741,730,912,849,PS,1451,NA,91,79,NA,23,11,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,15,4,729,730,903,849,PS,1451,NA,94,79,NA,14,-1,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,17,6,741,730,910,849,PS,1451,NA,97,79,NA,29,11,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,18,7,729,730,847,849,PS,1451,NA,78,79,NA,-2,-1,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,19,1,749,730,922,849,PS,1451,NA,93,79,NA,33,19,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,21,3,720,730,840,849,PS,1451,NA,80,79,NA,-1,-2,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,22,4,720,730,852,849,PS,1451,NA,84,79,NA,3,-2,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,23,5,731,730,902,849,PS,1451,NA,91,79,NA,13,1,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,24,6,744,730,900,849,PS,1451,NA,84,79,NA,19,14,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,25,7,729,730,851,849,PS,1451,NA,82,79,NA,2,-1,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA

```

Para generar el archivo con 25 líneas y hacer archivo original pasarlo a un backup

sed 25q -i".bak" aerolinea.csv



2.-Como ya se ha visto, utilizar el redireccionamiento destructivo (>) implica almacenar típicamente algún contenido en un archivo (**ej. echo "contenido" > archivo**).

Pero lo cierto es que con este comando no se apreciará en pantalla lo que se desea almacenar en dicho archivo, por ello es que se necesita que, con base en el comando resultado del ejercicio 1 y con la investigación del comando **tee**, por un lado el contenido se introduzca en el archivo **ejercicio_2.txt** y por el otro se muestre en pantalla la operación.

Caber mencionar que todo se debe registrar como una sola instrucción, es decir, no se puede ejecutar el resultado por partes, para ello tal vez quiera leer esta liga:

<http://www.linfo.org/pipes.html>

sed 25q aerolinea.csv | tee ejercicio_2.txt

```
cloudera@quickstart:~$ sed 25q aerolinea.csv | tee ejercicio_2.txt
Year,Month,DayOfMonth,DayOfWeek,DepTime,CRSDepTime,ArrTime,CRSArrTime,UniqueCarrier,FlightNum,TailNum,ActualElapsedTime,CRSElapsedTime,AirTime,ArrDelay,DepDelay,Origin,Dest,Distance,TaxiIn,TaxiOut,Cancelled,CancellationCode,Diverted,CarrierDelay,WeatherDelay,NASDelay,SecurityDelay,LateAircraftDelay
1987,10,14,3,741,730,912,849,PS,1451,NA,91,79,NA,23,11,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA,NA
1987,10,15,4,729,730,903,849,PS,1451,NA,94,79,NA,14,-1,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA,NA
1987,10,17,6,741,730,918,849,PS,1451,NA,97,79,NA,29,11,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA,NA
1987,10,18,7,729,730,847,849,PS,1451,NA,78,79,NA,-2,-1,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA,NA
1987,10,19,1,749,730,922,849,PS,1451,NA,93,79,NA,33,19,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA,NA
1987,10,21,3,728,730,848,849,PS,1451,NA,80,79,NA,-1,-2,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA,NA
1987,10,22,4,728,730,852,849,PS,1451,NA,84,79,NA,3,-2,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA,NA
1987,10,23,5,731,730,902,849,PS,1451,NA,91,79,NA,13,1,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA,NA
1987,10,24,6,744,730,906,849,PS,1451,NA,84,79,NA,19,14,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA,NA
1987,10,25,7,729,730,851,849,PS,1451,NA,82,79,NA,2,-1,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA,NA
1987,10,26,1,735,730,984,849,PS,1451,NA,89,79,NA,15,5,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA,NA
1987,10,28,3,741,725,919,855,PS,1451,NA,90,96,NA,24,10,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA,NA
1987,10,29,4,742,725,906,855,PS,1451,NA,84,90,NA,11,17,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA,NA
1987,10,31,6,726,725,848,855,PS,1451,NA,82,90,NA,-7,1,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA,NA
1987,10,1,4,936,915,1835,1801,PS,1451,NA,59,46,NA,34,21,SFO,RND,192,NA,NA,0,NA,0,NA,NA,NA,NA,NA,NA
1987,10,2,5,918,915,1817,1801,PS,1451,NA,59,46,NA,16,3,SFO,RND,192,NA,NA,0,NA,0,NA,NA,NA,NA,NA,NA
1987,10,3,6,928,915,1837,1801,PS,1451,NA,69,46,NA,36,13,SFO,RND,192,NA,NA,0,NA,0,NA,NA,NA,NA,NA,NA
1987,10,4,7,914,915,1803,1801,PS,1451,NA,49,46,NA,2,-1,SFO,RND,192,NA,NA,0,NA,0,NA,NA,NA,NA,NA,NA
1987,10,5,1,1042,915,1129,1801,PS,1451,NA,47,46,NA,88,87,SFO,RND,192,NA,NA,0,NA,0,NA,NA,NA,NA,NA,NA
1987,10,6,2,934,915,1824,1801,PS,1451,NA,58,46,NA,23,19,SFO,RND,192,NA,NA,0,NA,0,NA,NA,NA,NA,NA,NA
1987,10,7,3,946,915,1837,1801,PS,1451,NA,51,46,NA,36,31,SFO,RND,192,NA,NA,0,NA,0,NA,NA,NA,NA,NA,NA
1987,10,8,4,932,915,1833,1801,PS,1451,NA,61,46,NA,32,17,SFO,RND,192,NA,NA,0,NA,0,NA,NA,NA,NA,NA,NA
1987,10,9,5,947,915,1836,1801,PS,1451,NA,49,46,NA,35,32,SFO,RND,192,NA,NA,0,NA,0,NA,NA,NA,NA,NA,NA
1987,10,10,6,915,915,1822,1801,PS,1451,NA,67,46,NA,21,0,SFO,RND,192,NA,NA,0,NA,0,NA,NA,NA,NA,NA,NA
[cloudera@quickstart ~]$ ls ejercicio*
ejercicio_2.txt  ejercicio_2.txt-
[cloudera@quickstart ~]$
```

3.- Cambie el nombre del archivo **ejercicio_2.txt** a **ejercicio_3.txt** SIN usar el comando **rename**
cat ejercicio_2.txt > ejercicio_3.txt | rm -f ejercicio_2.*

```
cloudera@quickstart:~$ cat ejercicio_2.txt > ejercicio_3.txt | rm -f ejercicio_2.*
[cloudera@quickstart ~]$ ls
aerolinea      aerolinea.csv.bz2  Documents      enterprise-deployment.json  Music      rana-bone-age      workspace
aerolinea1.csv cloudera-manager  Downloads      express-deployment.json    parcels    rana-bone-age.zip
aerolinea2.csv  ca api.py          eclipse        kerberos                  Pictures    Templates
aerolinea.csv   Desktop            ejercicio_3.txt lib                  Public      Videos
[cloudera@quickstart ~]$
```

4.- Con algún comando en GNU/Linux tome las 25 últimas líneas del archivo **aerolínea.csv** SIN emplear el comando **tail** y guárdelo como **ejercicio_4.txt**

sed -e :a -e '\$q;N;26,\$D;ba' aerolinea.csv > ejercicio_4.txt

```
cloudera@quickstart:~$ sed -e :a -e '$q;N;26,$D;ba' aerolinea.csv > ejercicio_4.txt
[cloudera@quickstart ~]$ wc -l ejercicio_4.txt
25 ejercicio_4.txt
[cloudera@quickstart ~]$
```

5.- Concatene los archivos **ejercicio_3.txt** y **ejercicio_4.txt** en un archivo **ejercicio_5.txt** y en esa misma pantalla resultado muestre el contenido de **ejercicio_5.txt**

cat ejercicio_3.txt ejercicio_4.txt | tee ejercicio_5.txt


```
File Edit View Search Terminal Help
[cloudera@quickstart ~]$ cat ejercicio_3.txt ejercicio_4.txt | tee ejercicio_5.txt
Year,Month,DayOfMonth,DayOfWeek,DepTime,CRSDepTime,ArrTime,CRSArrTime,UniqueCarrier,FlightNum,TailNum,ActualElapsedTime,CRSElapsedTime,AirTime,ArrDelay,DepDelay,Origin,Dest,Distance,TaxiIn,TaxiOut,Cancelled,CancellationCode,Diverted,CarrierDelay,WeatherDelay,NASDelay,SecurityDelay,LateAircraftDelay
1987,10,14,3,741,730,912,849,PS,1451,NA,91,79,NA,23,11,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,15,4,729,730,903,849,PS,1451,NA,94,79,NA,14,-1,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,17,6,741,730,918,849,PS,1451,NA,97,79,NA,29,11,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,18,7,729,730,847,849,PS,1451,NA,78,79,NA,-2,-1,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,19,1,749,730,922,849,PS,1451,NA,93,79,NA,33,19,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,21,3,728,730,848,849,PS,1451,NA,80,79,NA,-1,-2,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,22,4,728,730,852,849,PS,1451,NA,84,79,NA,3,-2,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,23,5,731,730,902,849,PS,1451,NA,91,79,NA,13,1,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,24,6,744,730,908,849,PS,1451,NA,84,79,NA,19,14,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,25,7,729,730,851,849,PS,1451,NA,82,79,NA,2,-1,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,26,1,735,730,904,849,PS,1451,NA,89,79,NA,15,5,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,28,3,741,725,919,855,PS,1451,NA,90,90,NA,24,16,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,29,4,742,725,906,855,PS,1451,NA,84,90,NA,11,17,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,31,6,728,725,848,855,PS,1451,NA,82,90,NA,-7,1,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,1,4,936,915,1035,1001,PS,1451,NA,59,46,NA,34,21,SFO,RNO,192,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,2,5,910,915,1017,1001,PS,1451,NA,59,46,NA,16,3,SFO,RNO,192,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,3,6,928,915,1037,1001,PS,1451,NA,69,46,NA,36,13,SFO,RNO,192,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,4,7,914,915,1003,1001,PS,1451,NA,49,46,NA,2,-1,SFO,RNO,192,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,5,1,1042,915,1129,1001,PS,1451,NA,47,46,NA,80,87,SFO,RNO,192,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,6,2,934,915,1024,1001,PS,1451,NA,50,46,NA,23,19,SFO,RNO,192,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,7,3,946,915,1037,1001,PS,1451,NA,51,46,NA,36,31,SFO,RNO,192,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,8,4,932,915,1033,1001,PS,1451,NA,61,46,NA,32,17,SFO,RNO,192,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,9,5,947,915,1036,1001,PS,1451,NA,49,46,NA,35,32,SFO,RNO,192,NA,NA,0,NA,0,NA,NA,NA,NA,NA
1987,10,10,6,915,915,1022,1001,PS,1451,NA,67,46,NA,21,0,SFO,RNO,192,NA,NA,0,NA,0,NA,NA,NA,NA,NA
2000,12,13,6,1910,1910,2017,2016,DL,1612,N927DA,67,60,38,1,0,ATL,CHS,259,5,24,0,0,0,NA,NA,NA,NA,NA
2000,12,13,6,1441,1445,1604,1622,DL,1613,N973DL,83,97,65,-10,-4,IND,ATL,432,8,10,0,0,0,NA,NA,NA,NA,NA
2000,12,13,6,921,830,1112,1000,DL,1616,N907DE,111,90,82,64,51,ATL,PBI,545,8,21,0,0,0,51,0,13,0,0
2000,12,13,6,1435,1440,1701,1704,DL,1618,N914DL,86,84,56,-3,-5,MSY,ATL,425,20,10,0,0,0,NA,NA,NA,NA,NA
2000,12,13,6,1750,1755,2010,2015,DL,1618,N914DL,140,140,113,-5,-5,ATL,BOL,859,7,20,0,0,0,NA,NA,NA,NA,NA
2000,12,13,6,700,710,850,837,DL,1619,N949DL,104,87,49,13,-4,LEX,ATL,303,23,32,0,0,0,NA,NA,NA,NA,NA
2000,12,13,6,1552,1520,1735,1710,DL,1620,N905DE,43,58,27,17,32,HSV,ATL,151,9,7,0,0,0,0,0,0,17
```

6.- Usando el comando **ls** y sus opciones, verifique el peso de **ejercicio_5.txt**, señalando en la captura de pantalla dónde se encuentra éste.

ls -lah ejercicio_5.txt

```
cloudera@quickstart:~$ ls -lah ejercicio_5.txt
-rw-rw-r-- 1 cloudera cloudera 5.0K Jun 24 12:55 ejercicio_5.txt
```

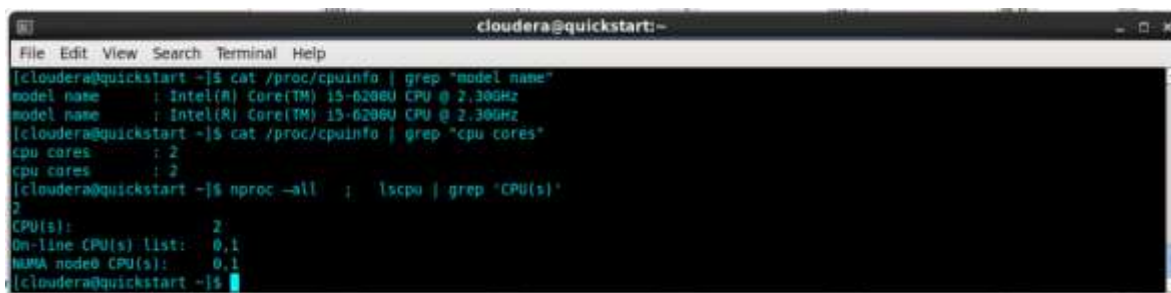
7.- Modifique la fecha de acceso de **ejercicio_5.txt** al 25 de Agosto del 2018 y muestre en pantalla dónde se puede apreciar ese resultado.

touch -t 1808250000 ejercicio_5.txt

```
cloudera@quickstart:~$ touch -t 1808250000 ejercicio_5.txt
cloudera@quickstart:~$ ls -lat ejercicio_5.txt
-rw-rw-r-- 1 cloudera cloudera 5.0K Aug 25 2018 ejercicio_5.txt
```

8.- ¿Con cuál comando se puede averiguar el número de núcleos en un sistema GNU/Linux? Investigue y coloque el resultado, haciendo énfasis en el lugar donde se puede apreciar esa información.

`nproc -all ; lscpu | grep 'CPU(s)'`



```
cloudera@quickstart:~$ cat /proc/cpuinfo | grep "model name"
model name      : Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz
model name      : Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz
cloudera@quickstart:~$ cat /proc/cpuinfo | grep "cpu cores"
cpu cores       : 2
cpu cores       : 2
cloudera@quickstart:~$ nproc -all ; lscpu | grep 'CPU(s)'
2
CPU(s):         2
On-line CPU(s) list:  0,1
NUMA node0 CPU(s):   0,1
cloudera@quickstart:~$
```

9.- Investigue en qué consiste awk y por medio de esa herramienta imprima en pantalla sólo la tercera y quinta columnas (de izquierda a derecha) del archivo **ejercicio_5.txt**. He aquí un ejemplo de cómo se ve el resultado con otro archivo que no tiene que ver con el curso:



```
aaron@aaron-trabajo-vb:~/Descargas$ cat muestra.txt
a,c,b,d,e,f
g,h,i,j,k,l
m,n,o,p,q,r
aaron@aaron-trabajo-vb:~/Descargas$ awk '{print $3,$5}' muestra.txt
be
ik
oq
aaron@aaron-trabajo-vb:~/Descargas$
```

AWK es una herramienta de procesamiento de patrones en líneas de texto. Su utilización estándar es la de filtrar ficheros o salida de comandos de UNIX, tratando las líneas para, por ejemplo, mostrar una determinada información sobre las mismas.

Por ejemplo:

Mostrar sólo los nombres y los tamaños de los ficheros:
`ls -l | awk '{ print $8 ":" $5 }'`

Mostrar sólo los nombres y tamaños de ficheros .txt:
`ls -l | awk '$8 ~ /\.txt/ { print $8 ":" $5 }'`

Imprimir las líneas que tengan más de 4 campos/columnas:
`awk 'NF > 4 {print}' fichero`

Formato de uso:

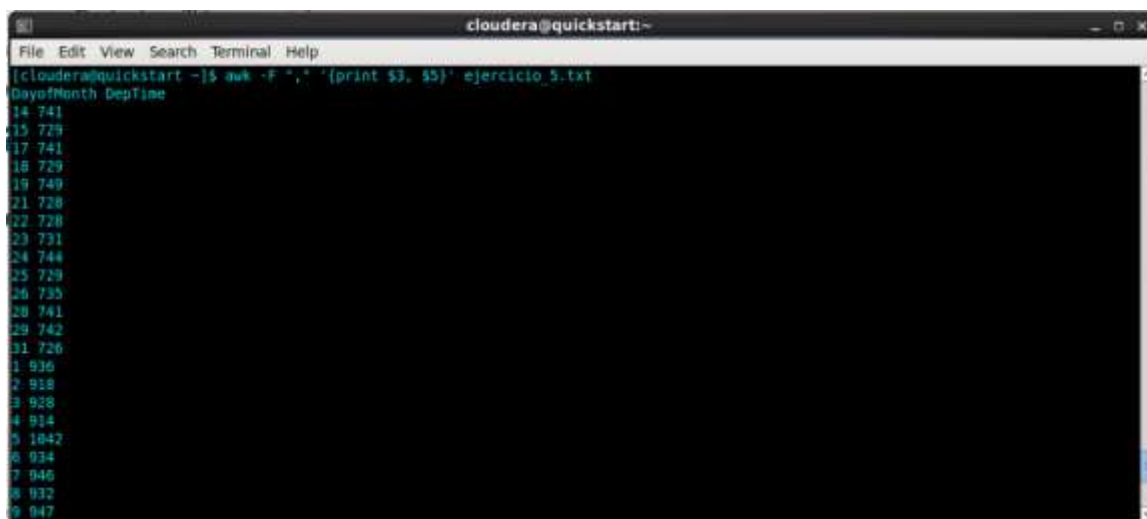
El uso básico de AWK es:

```
awk [condicion] { comandos }
```

Donde:

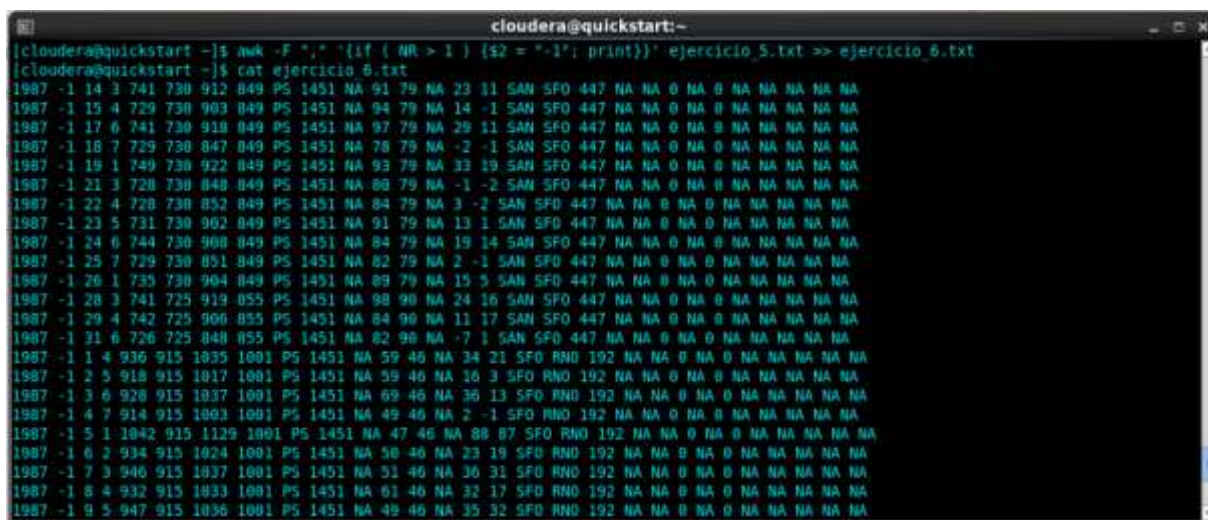
- **[condicion]** representa una condición de matcheo de líneas o parámetros.
- **comandos** : una serie de comandos a ejecutar

awk -F "," '{print \$3, \$5}' ejercicio_5.txt



10.- Sin usar vim, nano o editor de texto alguno use comandos de Linux para reemplazar TODOS los elementos de la segunda columna por -1, guárdelo como **archivo_6.txt** y hágale un cat a ese mismo archivo.

awk -F "," '{if (NR > 1) {\$2 = "-1"; print}}' ejercicio_5.txt >> ejercicio_6.txt



SECCION 2. HDFS Y HIVE

11.- Se está tratando de hacer la siguiente operación:

hdfs dfs -head /raw/aerolínea.csv

Con una captura muestre qué es lo que pasa y por medio de argumentos sólidos (una captura de pantalla con la evidencia, una fuente de consulta) por qué sucede esto.

```
cloudera@quickstart:~  
File Edit View Search Terminal Help  
[cloudera@quickstart ~]$ hdfs dfs -head /raw/aerolinea.csv  
-head: Unknown command  
[cloudera@quickstart ~]$ hdfs dfs  
Usage: hadoop fs [generic options]  
    [-appendToFile <localsrc> ... <dst>]  
    [-cat [-ignoreCrc] <src> ...]  
    [-checksum <src> ...]  
    [-chgrp [-R] GROUP PATH...]  
    [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]  
    [-chown [-R] [OWNER][:[GROUP]] PATH...]  
    [-copyFromLocal [-f] [-p] [-l] <localsrc> ... <dst>]  
    [-copyToLocal [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]  
    [-count [-q] [-h] [-v] [-x] <path> ...]  
    [-cp [-f] [-p | -p[topax]] <src> ... <dst>]  
    [-createSnapshot <snapshotDir> [<snapshotName>]]  
    [-deleteSnapshot <snapshotDir> <snapshotName>]  
    [-df [-h] [<path> ...]]  
    [-du [-s] [-h] [-x] <path> ...]  
    [-expunge]  
    [-find <path> ... <expression> ...]  
    [-get [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]  
    [-getfacl [-R] <path>]  
    [-getfattr [-R] {-n name | -d} [-e en] <path>]  
    [-getmerge [-nl] <src> <localdst>]
```

El comando **head** no forma parte del listado de opciones del Sistema de Archivos de Hadoop, solo se encuentra el comando **tail**.

12.- Cuente cuántas líneas tiene el archivo **aerolínea.csv** que está en el **HDFS**. Recuerde el carácter pipe (|) empleado en ejercicios anteriores.

hdfs dfs -cat /raw/aerolinea.csv | wc -l

13.- Indague en la instrucción de HDFS para averiguar el factor de réplica del archivo **aerolínea.csv** y colóquelo aquí junto con captura del resultado.

```
[cloudera@quickstart ~]$ hdfs dfs -stat /raw/aerolinea.csv  
2018-06-13 23:34:31  
[cloudera@quickstart ~]$ hdfs dfs -stat %r /raw/aerolinea.csv  
1  
[cloudera@quickstart ~]$
```

14.- Tome como base el archivo **aerolínea.csv** del HDFS y almacene en el sistema local un archivo **ejercicio_14.txt** que contenga las primeras 15 líneas sin usar el comando **-tail** del HDFS. Muestre ese contenido también.

hdfs dfs -cat /raw/aerolinea.csv | sed 15q | tee ejercicio_14.txt

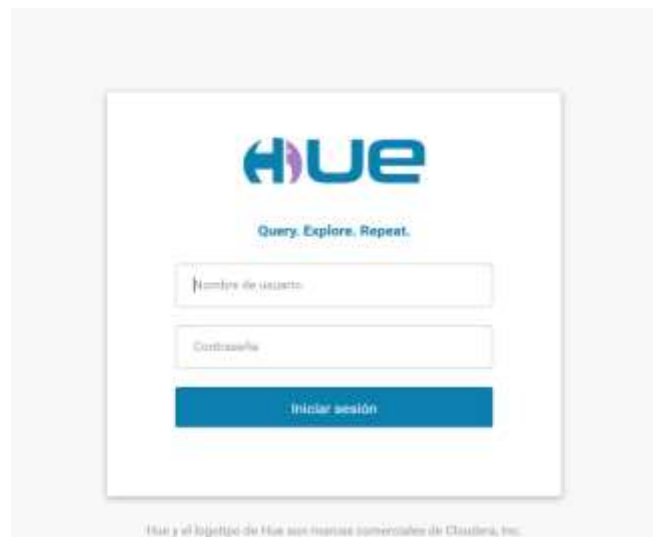
15.- Cree los directorios **master** y **stagin** en el directorio raíz del HDFS y además al archivo **aerolínea.csv** que está en **raw** cámbiele los permisos de tal manera que el propietario tenga todas las facilidades sobre él, el grupo sólo pueda leer y escribir y cualquier otro no tenga ningún permiso. Coloque las capturas de ambos ejercicios por separado.


```
[cloudera@quickstart ~]$ sudo -u hdfs hdfs dfs -mkdir /master
[cloudera@quickstart ~]$ sudo -u hdfs hdfs dfs -mkdir /staging
[cloudera@quickstart ~]$ hdfs dfs -ls /
Found 9 items
drwxrwxrwx - hdfs supergroup 0 2017-07-19 05:34 /benchmarks
drwxr-xr-x - hbase supergroup 0 2018-06-12 15:52 /hbase
drwxr-xr-x - hdfs supergroup 0 2018-06-24 20:52 /master
drwxr-xr-x - cloudera supergroup 0 2018-06-13 16:34 /raw
drwxr-xr-x - solr solr 0 2017-07-19 05:37 /solr
drwxr-xr-x - hdfs supergroup 0 2018-06-24 20:52 /staging
drwxrwxrwt - hdfs supergroup 0 2018-06-11 13:51 /tmp
drwxr-xr-x - hdfs supergroup 0 2017-07-19 05:36 /user
drwxr-xr-x - hdfs supergroup 0 2017-07-19 05:36 /var
[cloudera@quickstart ~]$
```

```
[cloudera@quickstart ~]$ hdfs dfs -chmod 760 /raw/aerolinea.csv
[cloudera@quickstart ~]$ hdfs dfs -ls /raw
Found 1 items
-rwxrw---- 1 cloudera supergroup 12029208594 2018-06-13 16:34 /raw/aerolinea.csv
[cloudera@quickstart ~]$
```

16.- Para los siguientes ejercicios puede hacer uso del servicio Hue (si no ha activado los servicios en Cloudera Manager tiene que hacerlo antes, para entrar a Hue en el mismo navegador se encuentra esta opción).

Aparecerá una ventana como ésta:



Recuerde que tanto el usuario como la contraseña es **cloudera**:



Entonces tome el siguiente código y cree una tabla en Hive:

```
CREATE EXTERNAL TABLE tabla_aerolinea(
```

```
Year STRING,  
Month STRING,  
DayofMonth STRING,  
DayOfWeek STRING,  
DepTime STRING,  
CRSDepTime STRING,  
ArrTime STRING,  
CRSArrTime STRING,  
UniqueCarrier STRING,  
FlightNum STRING,  
TailNum STRING,  
ActualElapsedTime STRING,  
CRSElapsedTime STRING,  
AirTime STRING,  
ArrDelay STRING,  
DepDelay STRING,  
Origin STRING,  
Dest STRING,  
Distance STRING,  
TaxiIn STRING,  
TaxiOut STRING,  
Cancelled STRING,  
CancellationCode STRING,  
Diverted STRING,  
CarrierDelay STRING,  
WeatherDelay STRING,  
NASDelay STRING,
```

```
SecurityDelay STRING,  
LateAircraftDelay STRING)
```

```
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY ','  
STORED AS TEXTFILE  
location '/raw';
```

En el código anterior **NO** existe una forma de omitir los encabezados por lo que es su deber encontrar esa manera, incluirla en el código y crear la tabla.
Para acreditar el ejercicio debe mostrar la sentencia que requirió para la parte de los encabezados y hacer un SELECT de los 10 primeros elementos de la tabla.

17.- Borre la tabla anterior y vuélvala a crear pero ahora el tipo de dato Origin debe ser INT, entonces vuelva a ejecutar la consulta y especifique qué ha pasado y con una captura muéstrela.

18.- Borre la tabla anterior, vuélvala a crear (con Origin STRING) pero ahora añada una columna después de LateAircraftDelay llamada **Adicional** con tipo de dato **STRING**, ejecute la creación, indique qué ha sucedido y coloque captura del resultado.

19.- En esta tabla anterior inserte un renglón a la tabla con todos los valores iguales a “NA” (tiene que investigar cómo añadir elementos a la tabla), y luego después de la inserción del elemento indague en qué parte del HDFS se ha guardado ese nuevo elemento.

SECCIÓN 3. PREGUNTAS ABIERTAS

20.- ¿Qué es el Sticky Bit? Ejemplifíquelo con el archivo **ejercicio_5.txt** adjuntando una captura de pantalla.

<http://rm-rf.es/permisos-especiales-setuid-setgid-sticky-bit/>

*Este bit suele asignarse en directorios a los que todos los usuarios tienen acceso, y **permite evitar que un usuario pueda borrar ficheros/directorios de otro usuario dentro de ese directorio, ya que todos tienen permiso de escritura.***

```
[cloudera@quickstart ~]$ ls -lat ejercicio_5.txt
-rw-rw-r-- 1 cloudera cloudera 5046 Aug 25 2018 ejercicio_5.txt
[cloudera@quickstart ~]$ chmod o+t ejercicio_5.txt
[cloudera@quickstart ~]$ ls -lat ejercicio_5.txt
-rw-rw-r-T 1 cloudera cloudera 5046 Aug 25 2018 ejercicio_5.txt
[cloudera@quickstart ~]$
```

21.- ¿A qué se le conoce como NoSQL?, ¿considera que Hive e Impala son representantes? Justifique la respuesta.

<https://blogs.oracle.com/spain/qu-es-una-base-de-datos-nosql>

Las características comunes entre todas las implementaciones de bases de datos NoSQL suelen ser las siguientes:

- **Consistencia Eventual:** *A diferencia de las bases de datos relacionales tradicionales, en la mayoría de sistemas NoSQL, no se implementan mecanismos rígidos de consistencia que garanticen que cualquier cambio llevado a cabo en el sistema distribuido sea visto, al mismo tiempo, por todos los nodos y asegurando, también, la no violación de posibles restricciones de integridad de los datos u otras reglas definidas. En su lugar y para obtener un mayor rendimiento, se ofrece el concepto de “consistencia eventual”, en el que los cambios realizados “con el tiempo” serán propagados a todos los nodos por lo que, una consulta podría no devolver los últimos datos disponibles o proporcionar datos inexactos, problema conocido como lecturas sucias u obsoletas.*

Asimismo, en algunos sistemas NoSQL se pueden presentar pérdidas de datos en escritura. Esto se conoce también como BASE (Basically Available Soft-state Eventual Consistency), en contraposición a ACID (Atomicity, Consistency, Isolation, Durability), su analogía en las bases de datos relacionales.

- **Flexibilidad en el esquema:** *En la mayoría de base de datos NoSQL, los esquemas de datos son dinámicos; es decir, a diferencia de las bases de datos relacionales en las que, la escritura de los datos debe adaptarse a unas estructuras(o tablas, compuestas a su vez por filas y columnas) y tipos de datos pre-definidos, en los sistemas NoSQL, cada registro (o documento, como se les suele llamar en estos casos) puede contener una información con diferente forma cada vez, pudiendo así almacenar sólo los atributos que*

interesen en cada uno de ellos, facilitando el polimorfismo de datos bajo una misma colección de información. También se pueden almacenar estructuras complejas de datos en un sólo documento, como por ejemplo almacenar la información sobre una publicación de un blog (título, cuerpo de texto, autor, etc) junto a los comentarios y etiquetas vertidos sobre el mismo, todo en un único registro.

- **Escalabilidad horizontal:** Por escalabilidad horizontal se entiende la posibilidad de incrementar el rendimiento del sistema añadiendo, simplemente, más nodos (servidores) e indicando al sistema cuáles son los nodos disponibles.
- **Estructura distribuida:** Generalmente los datos se distribuyen, entre los diferentes nodos que componen el sistema. Hay dos estilos de distribución de datos:
 - **Particionado (ó Sharding):** El particionado distribuye los datos entre múltiples servidores de forma que, cada servidor, actúe como única fuente de un subconjunto de datos. Normalmente, a la hora de realizar esta distribución, se utilizan mecanismos de tablas de hash distribuidas (DHT).
 - **Réplica:** La réplica copia los datos entre múltiples servidores, de forma que cada bit de datos pueda ser encontrado en múltiples lugares. Esta réplica puede realizarse de dos maneras:
 - **Réplica maestro-esclavo** en la que un servidor gestiona la escritura de la copia autorizada mientras que los esclavos se sincronizan con este servidor maestro y sólo gestionan las lecturas.
 - **Réplica peer-to-peer** en la que se permiten escrituras a cualquier nodo y ellos se coordinan entre sí para sincronizar sus copias de los datos
- **Tolerancia a fallos y Redundancia:** Pese a lo que cualquiera pueda pensar cuando se habla de NoSQL, no todas las tecnologías existentes bajo este paraguas usan el mismo modelo de datos ya que, al ser sistemas altamente especializados, la idoneidad particular de una base de datos NoSQL dependerá del problema a resolver. Así a todo, podemos

agrupar los diferentes modelos de datos usados en sistemas NoSQL en cuatro grandes categorías:

1. **Base de datos de Documentos:** Este tipo de base de datos almacena la información como un documento, usando para habitualmente para ello una estructura simple como JSON, BSON o XML y donde se utiliza una clave única para cada registro. Este tipo de implementación permite, además de realizar búsquedas por clave-valor, realizar consultas más avanzadas sobre el contenido del documento. Son las bases de datos NoSQL más versátiles.
2. **Almacenamiento Clave-Valor:** Son el modelo de base de datos NoSQL más popular, además de ser la más sencilla en cuanto a funcionalidad. En este tipo de sistema, cada elemento está identificado por una clave única, lo que permite la recuperación de la información de forma muy rápida, información que suele almacenarse como un objeto binario. Se caracterizan por ser muy eficientes tanto para las lecturas como para las escrituras.
3. **Bases de datos de grafos:** Usadas para aquellos datos cuyas relaciones se pueden representar adecuadamente mediante un grafo. Los datos se almacenan en estructuras grafo con nodos (entidades), propiedades (información entre entidades) y líneas (conexiones entre las entidades).
4. **Base de datos Columnar (o Columna ancha):** En vez de "tablas", en las bases de datos de columna tenemos familias de columnas que, son los contenedores de las filas. A diferencia de los RDBMS, no necesita conocer de antemano todas las columnas, cada fila no tiene por qué tener el mismo número de columnas. Este tipo de bases de datos se adecuan mejor a operaciones analíticas sobre grandes conjuntos de datos.

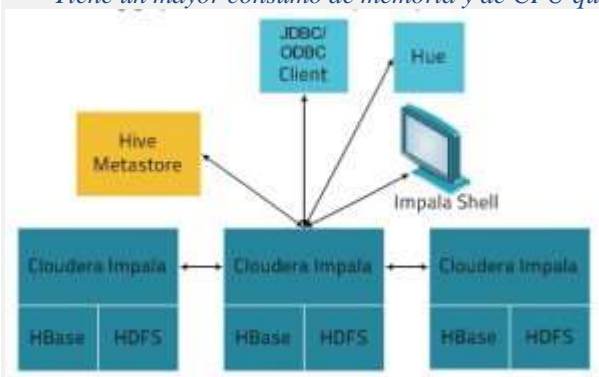
Pese a todas las opciones proporcionadas por el auge de las bases de datos NoSQL, esto no significa la desaparición de las bases de datos de RDBMS ya que son tecnologías complementarias. Estamos entrando en una era de persistencia políglota, una técnica que utiliza diferentes tecnologías de almacenamiento de datos para manejar las diversas necesidades de almacenamiento de datos.

<https://unpocodejava.com/2013/09/12/impala-y-hive-no-tan-parecidos/>

Dos de los proyectos más usados para realizar consultas sobre el ecosistema Hadoop son Impala y Hive. Pero aunque a simple vista pueden parecer muy similares no lo son tanto.

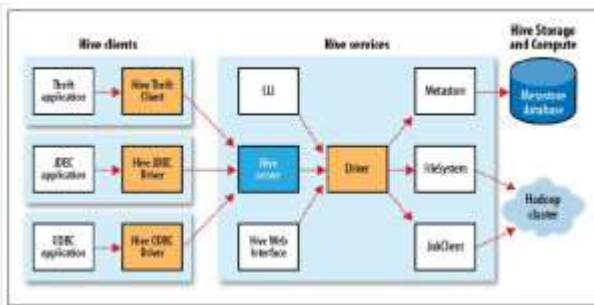
Cloudera Impala es un motor distribuido de consultas sobre HDFS o HBase

- *Su arquitectura es MPP (Massively Parallel Processing)*
- *No usa el Map/Reduce de Hadoop sino que utiliza procesos que se ejecutan en los nodos y que consultan directamente sobre HDFS o HBase*
- *Las consultas con Impala tienen una latencia menor que con Hive*
- *Tiene un mayor consumo de memoria y de CPU que Hive*



Hive es un sistema de data warehouse sobre Hadoop

- *Su arquitectura se basa en el Map/Reduce de Hadoop*
- *Es tolerante a fallos*
- *Consultas más lentas que Impala pero más robustas*



Los ámbitos idóneos de aplicación de cada uno serían los siguientes:

Cloudera Impala aplicaría mejor en consultas más ligeras donde prime la velocidad sobre la fiabilidad, cualquier fallo en algún nodo o proceso obligaría a relanzar la consulta.

Hive es mejor para trabajos pesados de tipo ETL (Extract, Transform and Load) donde no nos interesa tanto la velocidad como la robustez de la ejecución, ya que la alta tolerancia a fallos que presenta evita la necesidad de relanzamientos al fallar algún nodo.

22.- Investigue el uso del comando nohup en GNU/Linux y con base en esto responda: ¿cómo puede ser aplicado dicho comando en un sistema distribuido?

<http://rm-rf.es/nohup-mantiene-ejecucion-comando-pese-salir-terminal/>

El comando **nohup** permite mantener la ejecución de un comando (el cual le pasamos como un argumento) pese a salir de la terminal (logout), ya que hace que se ejecute de forma independiente a la sesión.

Básicamente, lo que hace es ignorar la señal HUP (señal que se envía a un proceso cuando la terminal que lo controla se cierra), esto implica que aunque cerremos la terminal, el proceso se siga ejecutando.

Por defecto, la salida del comando, que normalmente aparecería directamente en la terminal, será procesada a un fichero llamado nohup.out que aparecerá en la ruta donde nos encontremos al ejecutar el comando.

Este comando servirá para dejar ejecuciones en segundo plano en ambientes distribuidos.

23.- Se quiere averiguar la memoria RAM disponible con base en la siguiente imagen:


```
aaron@aaron-trabajo-vb:~/Descargas$ free -h
              total        usado        libre      compart.     búffers     almac.
Memoria:      3.9G         2.7G         1.2G         16M         66M         1.9G
-/+ buffers/cache:  700M         3.2G
Swap:         4.0G         176K         4.0G
```

Indique el o los valores adecuados y por qué.

La línea **Mem** nos indica como esta repartiendo la memoria del sistema:

- **Total** de memoria disponible
- **Memoria usado**, generalmente lo encontraremos muy próximo al total
- **Memoria libre**, generalmente veremos poca
- **Memoria compartida**
- **Buffers**, incluye memoria que se use para acceso a disco, red, sistemas de ficheros temporales...
- **Cache**, datos que se mantienen en memoria para posteriores accesos

24.- Se tiene el siguiente escenario: personal ajeno a su área de sistemas desea tener acceso al sistema, en particular para ver algunos datos del archivo **objetivo.txt**

Por otra parte se sabe de manera extraoficial que la meta de ellos consiste en “ensuciar” el archivo para que el área no tenga tanto repunte como la nuestra.

Por cuestiones burocráticas la creación de algún usuario nuevo no es plausible no obstante debido a asuntos políticos es prácticamente un hecho que se le tiene que dar permiso, por ello es que se optó por prestarles un usuario (**usuario_nuestro**) cuyo grupo es **grupo_nuestro**.

Con base en estas características y limitando el escenario únicamente a comandos **chmod** (y si lo desea **chown** y **chgrp**), ¿cuál sería la configuración que usted propondría para garantizar el acceso al archivo pero al mismo tiempo protegerlo de las circunstancias mencionadas y sin afectar al mismo tiempo a los demás miembros de **grupo_nuestro**?

25.- ¿Cuál es la diferencia entre Hadoop y Cloudera?

<https://www.zylk.net/en/web-2-0/blog/-/blogs/hortonworks-vs-cloude-1>

Las principales diferencias entre las distribuciones son las siguientes:

- *Cloudera ha anunciado que su objetivo a largo plazo es convertirse en una empresa data hub disminuyendo la necesidad de tener un almacén de datos. Hortonworks sin embargo, sigue*

siendo un proveedor de Hadoop distro, y se ha asociado con la empresa de almacenamiento de datos Teradata.

- *Mientras Cloudera puede ejecutarse en un servidor Windows, Hortonworks es disponible en el servidor Windows de forma nativa. Un cluster basado Windows-Hadoop puede ser desplegado en Windows Azure.*
- *Cloudera tiene software de gestión propietario llamado Cloudera Manager, un motor de consultas SQL llamado Impala, también Cloudera Search para búsquedas fáciles y acceso a los productos en tiempo real. Hortonworks no tiene software propietario, usa Ambari para la gestión, Stinger para manejar consultas y Apache Solr para búsquedas de datos.*
- *Cloudera tiene una licencia comercial, mientras que Hortonworks tiene una licencia de código abierto. Cloudera también permite el uso de sus proyectos open-source gratuitamente, pero el paquete no incluye la suite de gestión Cloudera Manager o cualquier otro software propietario.*
- *Cloudera tiene un trial gratuito de 60 días, Hortonworks es completamente gratis.*
- *A la hora de configurarlos inicialmente aparte de descargar la maquina virtual que trabajará con el software de virtualización favorito que se tenga, Cloudera ofrece una imagen Docker y Hortonworks te da la oportunidad de ejecutarlo desde la nube. Hortonworks en la nube es bastante fácil de configurar. Tan solo tienes que registrarte en Microsoft, puesto que la plataforma corre en Azure, y elegir una configuración de las predefinidas.*
- *Como herramientas de administración, Horton utiliza Ambari y Cloudera usa Cloudera Manager. A la hora de escalarlo Cloudera Manager cuenta con una serie de características (previo pago) como son: gestión multi-cluster, Actualizaciones continuas, integraciones extensibles con servicios de partner, backup y restauración ante desastres.*

26.- ¿Cuáles son los tipos de archivos existentes en GNU/Linux y Windows?

<https://ldc.usb.ve/~jose/OldPage/Colo2.html#SECTION00031000000000000000>

Éstos se refieren a aquellos que utilizan un solo disco físico o aquellos que se utilizan en un conjunto de los mismos que se comporta como si fuera uno solo.

Generalmente cada sistema operativo tiene un sistema de archivos predefinido. Por ejemplo: ext para Linux, NTFS para Windows y ZFS para Solaris. Entre los principales sistemas de archivos de disco tenemos*

1. *ADFS (Acorn's Advanced Disc Filing System), sucesor de DFS.*
2. *AdvFS (Advanced File System), diseñado por Digital Equipment Corporation (DEC) para su SO UNIX (Tru 64).*
3. *AFS (Ami File Safe), un sistema de archivo comercial para Amiga.*
4. *BFS (Be File System) usado en BeOS, también llamado BeFS.*
5. *Btrfs, un sistema de archivo implementado por Oracle en 2007 bajo GPL que tiene como punto a favor copia - escritura.*
6. *ext (Extended File System): diseñado para sistemas Linux.*
7. *ext2: Continuación de ext.*
8. *ext3: Continuación de ext2.*
9. *ext4: Continuación de ext3.*
10. *FAT (File Allocation Table): usado en Microsoft Windows y DOS.*
11. *HFS (Hierarchical File System): usado en sistemas MAC. Sucesor de MFS y predecesor de HFS+.*
12. *HPFS (High Performance File System): usado en OS/2.*
13. *ISO 9660: usado en CD-ROM y DVD-ROM.*
14. *JFS (IBM Journaling File System): provisto en AIX, Linux y sus extensiones.*
15. *MFS (Macintosh File System): usado en el SO MAC antiguos.*
16. *Minix file system: usado en sistemas Minix.*
17. *NTFS (NT File System): usado en sistemas operativos basados en Windows.*
18. *ReiserFS: utilizado en investigaciones.*
19. *STL (STandard Language file system): un Sistema de archivo desarrollado por IBM.*
20. *UFS (Unix File System): usado en Solaris y viejos sistemas BSD.*
21. *UFS2 (Unix File System): usado en sistemas BSD actuales.*
22. *XFS: usado en SGI IRIX y sistemas Linux.*
23. *ZFS (Zettabyte File System): Una especificación realizada por Sun Microsystem para los sistemas Z.*

27.- ¿Qué es el SerDe y cuál es su relación con Hive e Impala?

<https://unpocodejava.com/2013/01/24/apache-hive-y-serde/>

El interfaz SerDe permite indicarle a Hive como debe procesar un registro. SerDe es una combinación de Serializer y Deserializer.

• **Deserializer toma una representación string o binaria y lo convierte a un objeto Java que Hive puede manipular.**

• **Serializer: toma un objeto Java y lo convierte en algo que Hive puede escribir a HDFS.**

Para usar un SerDe a la hora de crear la tabla debo indicar que SerDe usar:

```

ADD JAR /tmp/hive-serdes-1.0-SNAPSHOT.jar

CREATE EXTERNAL TABLE tweets (
  ...
  retweeted_status STRUCT<
    text:STRING,
    user:STRUCT<screen_name:STRING,name:STRING>>,
  entities STRUCT<
    urls:ARRAY<STRUCT<expanded_url:STRING>>,
    user_mentions:ARRAY<STRUCT<screen_name:STRING,name:STRING>>>,
    hashtags:ARRAY<STRUCT<text:STRING>>>,
    text STRING,
  ...
)
PARTITIONED BY (datehour INT)
ROW FORMAT SERDE 'com.cloudera.hive.serde.JSONSerDe'
LOCATION '/user/flume/tweets';

```

28.- ¿A qué se le conoce como Big Table y Big Query?

<https://eduarea.wordpress.com/2013/01/11/que-es-google-bigtable/>

Big Table

Google BigTable es un mecanismo no relacional, un almacenamiento de datos distribuida y secuencia-replicasmultidimensional basado en las tecnologías de almacenamiento de propiedad de Google para la mayoría de aplicaciones en línea y back-end de la empresa / productos. Proporciona arquitectura de datos escalable para infraestructuras de bases de datos muy grandes. BigTable se utiliza principalmente en los productos de propiedad de Google, aunque algunos disponible en Internet en el Google App Engine y aplicaciones de otros fabricantes de bases de datos.

BigTable es un mapa continuo y ordenado. Cada cadena en el mapa consta de una fila, las columnas (varios tipos) y un valor de marca de tiempo que se utiliza para la indexación. Por ejemplo, una serie de datos de un sitio web que le ahorra:

- *La dirección URL invertido como el nombre de la fila (com.google.www).*
- *La columna de contenido almacena el contenido de las páginas web.*
- *El contenido ancla guarda cualquier texto de anclaje o referencia a la página de contenido.*
- *Un sello de tiempo proporciona la hora exacta en que se almacenaron los datos y se utiliza para ordenar múltiples instancias de una página.*

BigTable se construye en la parte superior de las tecnologías como Google File System y SSTable. Es utilizado por más de 60 aplicaciones de Google a partir de 2012, como Google Finance, Google Reader, Google Maps, Google Analytics y la indexación Web.

Está construido sobre GFS (Google File System), Chubby Lock Service, y algunos otros servicios y programas de Google, y funciona sobre 'commodity hardware' (sencillos y baratos PCs con procesadores Intel).

BigTable comenzó a ser desarrollado a principios de 2004.

BigTable almacena la información en tablas multidimensionales cuyas celdas están, en su mayoría, sin utilizar. Además, estas celdas disponen de versiones temporales de sus valores, con lo que se puede hacer un seguimiento de los valores que han tomado históricamente.

Para poder manejar la información, las tablas se dividen por columnas, y son almacenadas como 'tabletas' de unos 100-200 Mbytes cada una. Cada máquina almacena 100 tabletas, mediante el sistema 'Google File System'. La disposición permite un sistema de balanceo de carga (si una tableta está recibiendo un montón de peticiones, la máquina puede desprenderse del resto de las tabletas o trasladar la tableta en cuestión a otra máquina) y una rápida recomposición del sistema si una máquina 'se cae'.

BigTable es un mapa multidimensional ordenado, disperso, distribuido y persistente.

Google creó BigTable porque los sistemas de bases de datos tradicionales no tenían ni tienen, la capacidad de crear sistemas lo suficientemente grandes. Además, estos sistemas de bases de datos relacionales, como SQL Server, Oracle o MySQL fueron pensados y diseñados para que se ejecutasen en un solo servidor con mucha potencia. Por ello, no encajarían en las estructuras distribuidas de miles de servidores.

El paper de este sistema de almacenamiento, denominado BigTable, se encuentra en la página de Google.

<https://cloud.google.com/bigquery/?hl=es>

Big Query

Almacén de datos empresariales rápido, económico, escalable y totalmente administrado para analizar datos a cualquier escala.

Sin servidor

El almacenamiento de datos sin servidor proporciona los recursos que se necesitan y cuando se necesitan. Gracias a BigQuery podrás centrarte en los datos y análisis en lugar de en las operaciones y el tamaño de los recursos informáticos.

Análisis en tiempo real

La API de inserción de transmisión de alta velocidad de BigQuery es una potente base para los análisis en tiempo real. BigQuery permite analizar lo que sucede ahora mismo porque tienes los últimos datos empresariales disponibles de inmediato.

Alta disponibilidad automática

La replicación gratuita de los datos y los recursos informáticos en diversas áreas geográficas garantiza la disponibilidad de los datos, aun en el caso de fallos críticos. BigQuery proporciona de forma transparente y automática almacenamiento replicado y duradero y alta disponibilidad sin cargos ni configuraciones adicionales.

SQL estándar

BigQuery es compatible con un dialecto de SQL estándar que cumple con ANSI:2011, de modo que se reduce la necesidad de reescribir el código y se pueden aprovechar las funciones avanzadas de SQL. BigQuery proporciona controladores ODBC y JDBC gratuitos para asegurar que las aplicaciones actuales puedan interactuar con el potente motor de BigQuery.

Consultas federadas y almacenamiento de datos lógico

BigQuery separa los almacenes de datos para que puedas analizar todos los activos de datos desde un solo lugar. A través de la potente consulta federada, BigQuery procesa datos en almacenamiento de datos (Cloud Storage), en bases de datos de transacciones (Cloud Bigtable) o en hojas de cálculo en Google Drive sin tener que duplicar datos. Con una sola herramienta puedes consultar todas las fuentes de datos.

Separación de almacenamiento y recursos informáticos

BigQuery permite realizar un control exhaustivo de costes y accesos. Con la separación del almacenamiento y los recursos informáticos de BigQuery, solo pagas por los recursos que utilizas. Tienes la opción de elegir las soluciones de almacenamiento y procesamiento más adecuadas para tu empresa y controlar el acceso a cada una de ellas.

Copia de seguridad automática y restauración sencilla

BigQuery replica de forma automática los datos y conserva el historial de los cambios durante siete días con el fin de evitar sorpresas por cambios inesperados. De esta forma, puedes restaurar datos con facilidad y compararlos con los datos de otros momentos.

Data Transfer Service

Con BigQuery es fácil comenzar a usar el almacenamiento de datos, aunque estos se encuentren en una aplicación SaaS. BigQuery Data Transfer Service transfiere automáticamente los datos de fuentes de datos externas como DoubleClick, AdWords y YouTube a Google BigQuery de forma programada y totalmente administrada.

Integración del ecosistema de Big Data

Con Cloud Dataproc y Cloud Dataflow, BigQuery se integra en el ecosistema de Big Data Apache, lo que permite a las cargas de trabajo actuales de Hadoop/Spark y Beam leer o escribir datos directamente desde BigQuery. BigQuery permite sacar el máximo partido a los datos estructurados al facilitar el análisis en SQL y la integración con tus trabajos de Big Data actuales para que no tengas que deshacerte del trabajo que ya has hecho.

Escalado a petabyte

BigQuery es rápido y fácil de utilizar en datos de cualquier tamaño. Gracias a BigQuery disfrutarás de un magnífico rendimiento en tus datos sabiendo que puedes escalar sin problemas para almacenar y analizar más petabytes sin necesidad de comprar más capacidad.

Modelos de precios flexibles

BigQuery permite elegir el modelo de precios más idóneo. El modelo de precios bajo demanda significa que solo pagas por el almacenamiento y los recursos informáticos que utilices. En cambio, los precios fijos permiten a los usuarios o empresas con un gran volumen de análisis pagar un coste mensual fijo por estos análisis. Para obtener más información, consulta los precios de BigQuery.

Seguridad y encriptado de datos

Dispones de un control absoluto sobre quién tiene acceso a los datos almacenados en Google BigQuery. Con BigQuery es fácil conservar una seguridad elevada con un control detallado de la gestión de identidades y accesos y accesos con Google Cloud IAM. Además, los datos siempre están encriptados, tanto en reposo como en tránsito.

Localización de datos

Tienes la opción de almacenar tus datos de BigQuery en instalaciones europeas o estadounidenses a la vez que sigues beneficiándote de un servicio totalmente administrado. BigQuery ofrece la opción del control geográfico de datos, sin los quebraderos de cabeza que conlleva configurar y administrar los clústeres y otros recursos informáticos en la región.

Fundamento de la IA

BigQuery constituye un fundamento potente y flexible de aprendizaje automático e inteligencia artificial. BigQuery se integra con CloudML Engine y TensorFlow para formar modelos potentes sobre datos estructurados. Además, la capacidad de BigQuery de transformar y analizar datos ayuda a dar forma a los datos para el aprendizaje automático.

Ingestión flexible de datos

Carga los datos desde Google Cloud Storage o Google Cloud Datastore o bien transmítelos a BigQuery a miles de filas por segundo para analizarlos en tiempo real. Utiliza herramientas de integración de datos conocidas como Informatica, Talend y otras que se ofrecen preparadas para utilizar.

Control de datos

BigQuery controla de forma exhaustiva el acceso a datos y realiza controles basados en funciones en API a través de la integración con Google Cloud IAM. Con BigQuery y Cloud IAM, puedes estar seguro de que no habrá accesos no autorizados a tus datos.

Interacción mediante programación

BigQuery proporciona una API REST para facilitar el acceso mediante programación y la integración de aplicaciones. Para dar cabida a programadores de todo tipo, BigQuery ofrece bibliotecas de clientes en Java, Python, Node.js, C#, Go, Ruby y PHP. Los usuarios empresariales pueden utilizar Google AppScript para acceder a BigQuery desde Hojas de cálculo de Google.

Supervisión y almacenamiento de registros avanzados con Stackdriver

BigQuery cuenta con supervisión, almacenamiento de registros y alertas avanzados a través de registros de auditoría de Stackdriver. Los recursos de BigQuery se pueden supervisar de un solo vistazo y BigQuery puede servir como repositorio para los registros de cualquier aplicación o servicio que utilice el almacenamiento de registros de Stackdriver.

Controles de costes

BigQuery proporciona mecanismos de control de costes que permiten limitar los costes diarios. Haz clic [aquí](#) para obtener más información sobre los controles de costes.

29.- ¿A qué se le denomina Data Lake y Data Warehouse?

<http://errequeerre.es/data-warehouse-y-data-lake>

El término Data Warehouse fue acuñado en un artículo de diario de Sistemas de IBM de 1988 en el que analizaba las necesidades de las organizaciones de mantener sus bases de datos operacionales así como la necesidad de proporcionar capacidades de acceso y análisis a los usuarios finales.

Generalmente los departamentos de Business Intelligence (BI) creaban subconjuntos de la información contenida en el DW para hacerla accesible a ciertas partes del negocio en respuesta a las necesidades específicas de información de sus procesos de negocio. A esta estructura de datos ad hoc se la denomina Data Mart. Es habitual que una empresa cuente con diferentes Data Marts para diferentes unidades de negocio con distintas estructuras de datos. De esta

manera se hacia accesible en un formato más eficiente la explotación de los datos de interés para cada unidad de negocio.

La forma habitual de hacer llegar la información a un DW es a través de procesos ETL, es decir realizar una extracción de los datos (Extract) de bases de datos heterogéneas (CRM, ERP...), Transformar los datos (Transform), un proceso en el que se aplican una serie de reglas a los datos para adaptarlos a la estructura de datos de la base de datos de destino, y finalmente Cargarlos (Load) físicamente en el Data Warehouse. Estos procesos son requeridos precisamente por la necesidad que tienen los DW de mantener sus datos conforme a una estructura definida.

Data Lakes son inmensos repositorios de datos que son capaces de almacenar información tal y cómo llega sin preocuparse de si los datos son estructurados, desestructurados o semi-estructurados. El término Data Lake es atribuido a Peter Dixon CTO de Pentaho.

En primer lugar la información llega al Data Lake tal y como viene de la fuente original (raw data), sin procesos intermedios de transformación. Esta filosofía implica la segunda característica del Data Lake, su capacidad para recoger los datos de diversas fuentes sin preocuparse de la estructura o la ausencia de estructura del dato que le llega, se lo traga todo por decirlo de alguna manera.

Otra característica de los Data Lakes es su flexibilidad ya que los datos están en formato raw mientras que un Data Warehouse ha realizado un proceso de transformación y adaptación (ETL) a una determinada estructura antes de guardar los datos. Este es un punto fundamental en la diferencia entre los modelos de un DW y un Data Lake.

Tradicionalmente los Data Warehouse han trabajado en una arquitectura denominada schema-on-write, el fundamento detrás del modelo ETL de la carga de datos en un Data Warehouse. Este modelo obliga a la empresa a definir un modelo de datos y crear un marco analítico previo a la carga de ningún dato, es decir, necesitamos definir que vamos a querer hacer con los datos antes de cargarlos en la base de datos. Evidentemente la definición no es inamovible pero el esfuerzo en tiempo y dinero para cambiar el esquema de un Data Warehouse es mucho mayor.

La filosofía de la ingesta de datos de un Data Lake se basa en otro modelo de arquitectura denominado schema-on-read. En esta alternativa se sigue otra secuencia diferente a la anterior, es decir, en lugar de marcar la estructura de los datos en la entrada a la base de datos es cuando se quieren usar los datos cuando se aplica el proceso de transformación de los datos.

Precisamente al estar el dato en formato bruto (raw data) tenemos la posibilidad de poder adaptarnos para prácticamente cualquier proceso analítico. De esta manera podemos dar respuesta a las necesidades de un usuario típico de negocio a la vez que le damos solución a las mucho más complejas y exigentes necesidades de un científico de datos (Data Scientist).

30.- ¿Existe algún otro tipo de sistemas de archivos distribuidos que NO sea HDFS? si es así, ¿de cuáles se trata?

<https://searchdatacenter.techtarget.com/es/cronica/La-tecnologia-de-codigo-abierto-promete-alterar-el-almacenamiento-empresarial>

- *Lustre es un sistema de archivos paralelo utilizado principalmente para los requisitos informáticos de alto rendimiento. Está licenciado bajo GPL, versión 2, administrado por Open Scalable File Systems y diseñado para ejecutarse en Linux. Hasta mayo de 2017, Intel soportaba comercialmente las implementaciones de software Lustre, pero parece haber suspendido el soporte. Esto ha dejado a compañías como DataDirect Networks para proporcionar soporte como parte de los paquetes de hardware.*

- *FreeNAS es un dispositivo de almacenamiento de fuente abierta que tiene más de 10 años. Su software se basa en el Zettabyte File System (ZFS) altamente escalable y de código abierto. IXsystems proporciona soporte comercial para FreeNAS con un dispositivo de hardware llamado TrueNAS.*

- *GlusterFS, o Gluster File System, es un sistema de archivos de escala que también está disponible en Red Hat como plataforma de almacenamiento comercial. La empresa, Gluster, desarrolló y apoyó originalmente GlusterFS hasta que fue adquirida por Red Hat en 2011. El software tiene licencia bajo GPL, versión 3. GlusterFS consolida los recursos de almacenamiento de múltiples servidores o nodos en un solo sistema de archivos paralelo. Los servidores contribuyentes pueden ser proveedores de almacenamiento, llamados ladrillos de almacenamiento o consumidores de almacenamiento. Como producto de almacenamiento, GlusterFS es simple de implementar. Utiliza una arquitectura de metadatos distribuidos, por lo que es especialmente adecuada para archivos de gran escala. Almacenamiento de archivos*

- *Lustre es un sistema de archivos paralelo utilizado principalmente para los requisitos informáticos de alto rendimiento. Está licenciado bajo GPL, versión 2, administrado por Open Scalable File Systems y diseñado para ejecutarse en Linux. Hasta mayo de 2017, Intel soportaba comercialmente las implementaciones de software Lustre, pero parece haber suspendido el soporte. Esto ha dejado a compañías como DataDirect Networks para proporcionar soporte como parte de los paquetes de hardware.*

- *FreeNAS es un dispositivo de almacenamiento de fuente abierta que tiene más de 10 años. Su software se basa en el Zettabyte File System (ZFS) altamente escalable y de código abierto. IXsystems proporciona soporte comercial para FreeNAS con un dispositivo de hardware llamado TrueNAS.*

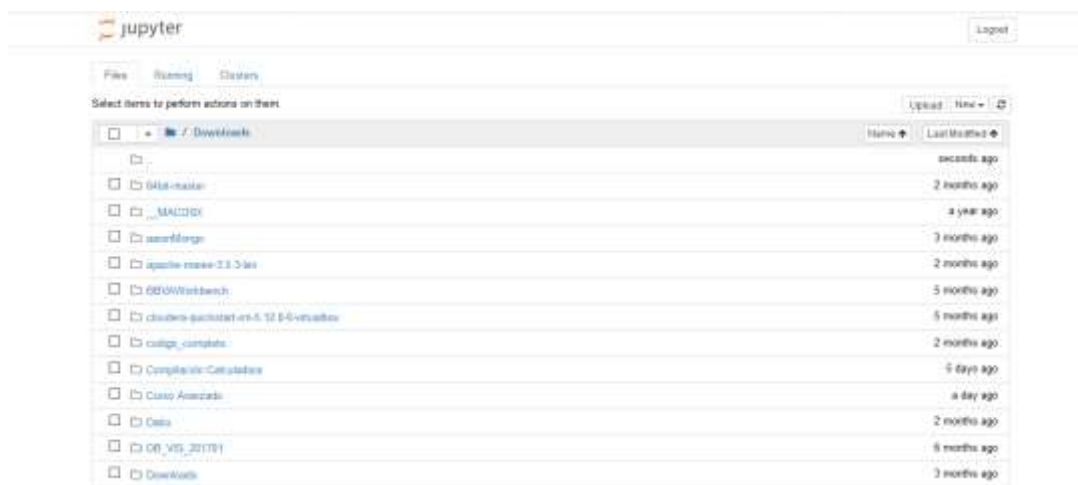
- *GlusterFS, o Gluster File System, es un sistema de archivos de escala que también está disponible en Red Hat como plataforma de almacenamiento comercial. La empresa, Gluster, desarrolló y apoyó originalmente GlusterFS hasta que fue adquirida por Red Hat en 2011. El software tiene licencia bajo GPL, versión 3. GlusterFS consolida los recursos de almacenamiento de múltiples servidores o nodos en un solo sistema de archivos paralelo. Los servidores contribuyentes pueden ser proveedores de almacenamiento, llamados ladrillos de almacenamiento o consumidores de almacenamiento. Como producto de almacenamiento, GlusterFS es simple de implementar. Utiliza una arquitectura de metadatos distribuidos, por lo que es especialmente adecuada para archivos de gran escala.*

SECCIÓN 4. ESPECIAL

31.- Instale Jupyter en Cloudera, para ello puede basarse en la siguiente liga:

<https://medium.com/@vando/install-jupyter-notebook-on-centos-7-1d596abf08da>

Es importante señalar que para continuar el curso es imprescindible esta herramienta y no existirán pausas para su instalación durante las sesiones, motivo por el cual es menester llevar a cabo esta operación aunque solamente valga 1 crédito. Para validar este ejercicio se requiere una captura de pantalla del menú principal, algo así:



Resultado:

