

## 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:

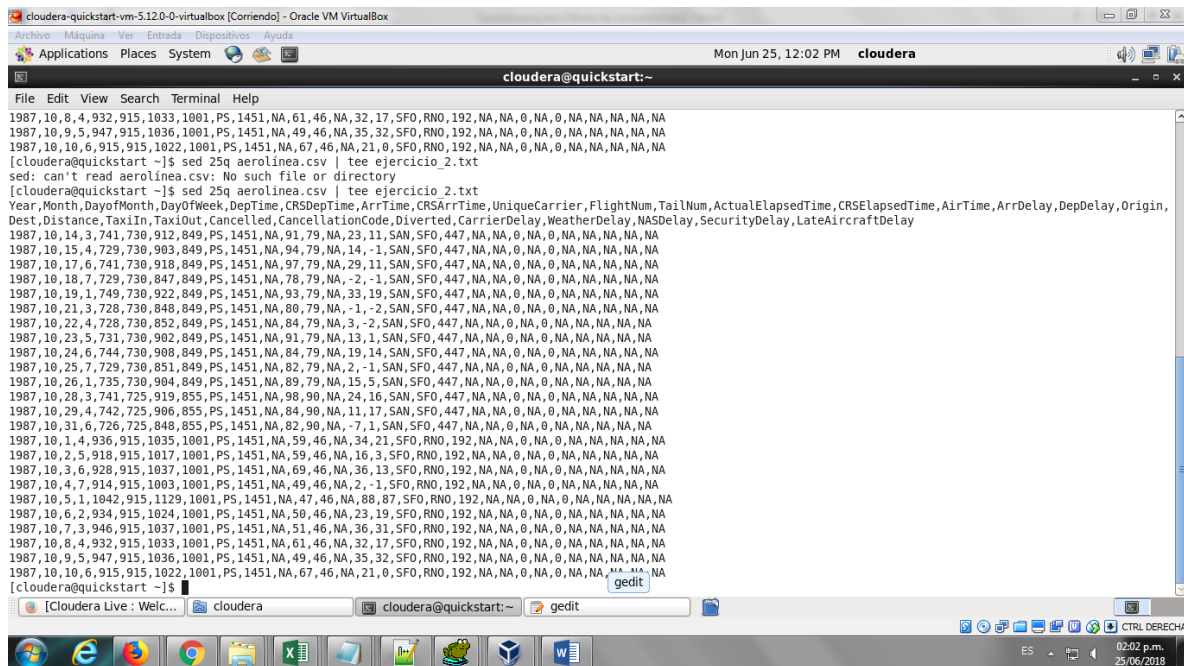
2.-Como ya se ha visto, utilizar el redireccionamiento destructivo (>) implica almacenar típicamente algún contenido en un archivo (ej. **echo "contenido" > archivo**).

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>

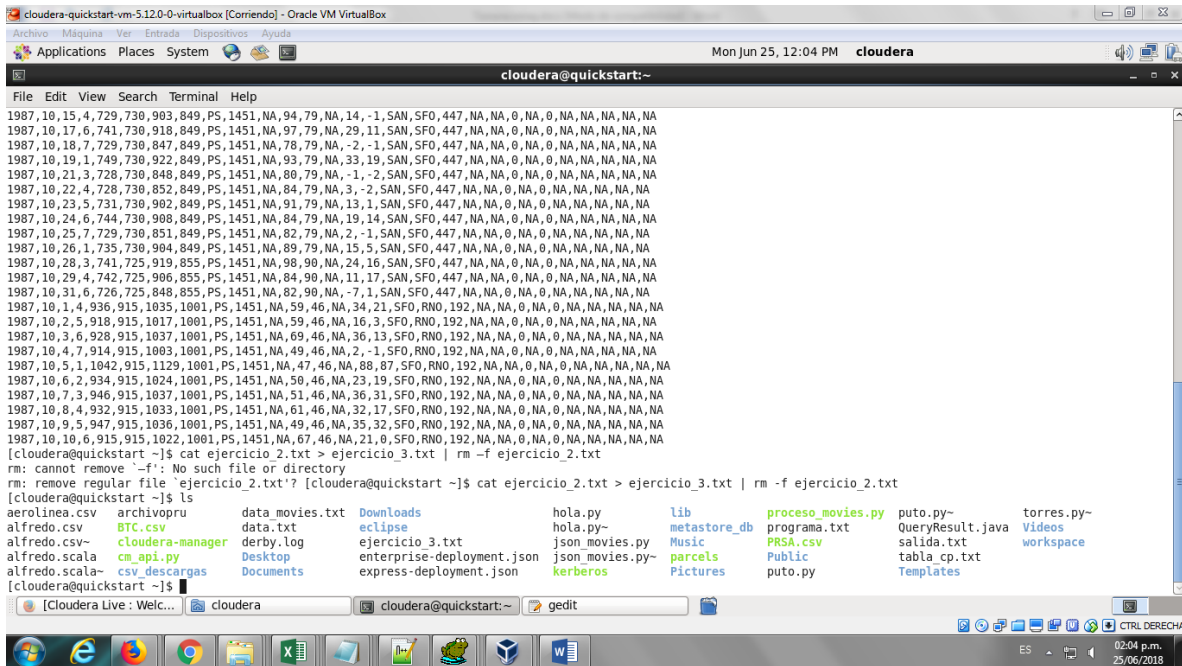
sed 25q aerolínea.csv | tee ejercicio\_2.txt



```
cloudera-quickstart-vm-5.12.0-0-virtualbox [Corriendo] - Oracle VM VirtualBox
Archivo  Maquina  Ver  Entrada  Dispositivos  Ayuda
Applications  Places  System
cloudera@quickstart:~
File Edit View Search Terminal Help
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
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
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
[cloudera@quickstart ~]$ sed 25q aerolinea.csv | tee ejercicio_2.txt
sed: can't read aerolinea.csv: No such file or directory
[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
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,98,90,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
1987,10,31,6,726,725,848,855,PS,1451,NA,82,90,NA,-7,1,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
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
1987,10,2,5,918,915,1017,1001,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,88,87,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
[cloudera@quickstart ~]$
```

3.- Cambie el nombre del archivo **ejercicio\_2.txt** a **ejercicio\_3.txt** SIN usar el comando rename

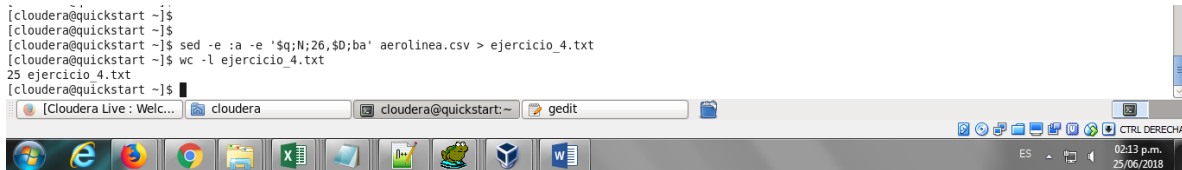
cat ejercicio\_2.txt > ejercicio\_3.txt | rm -f ejercicio\_2.txt



4.- Con algún comando en GNU/Linux tome las 25 últimas líneas del archivo **aerolinea.csv** SIN emplear el comando **tail** y guárdelo como **ejercicio\_4.txt**

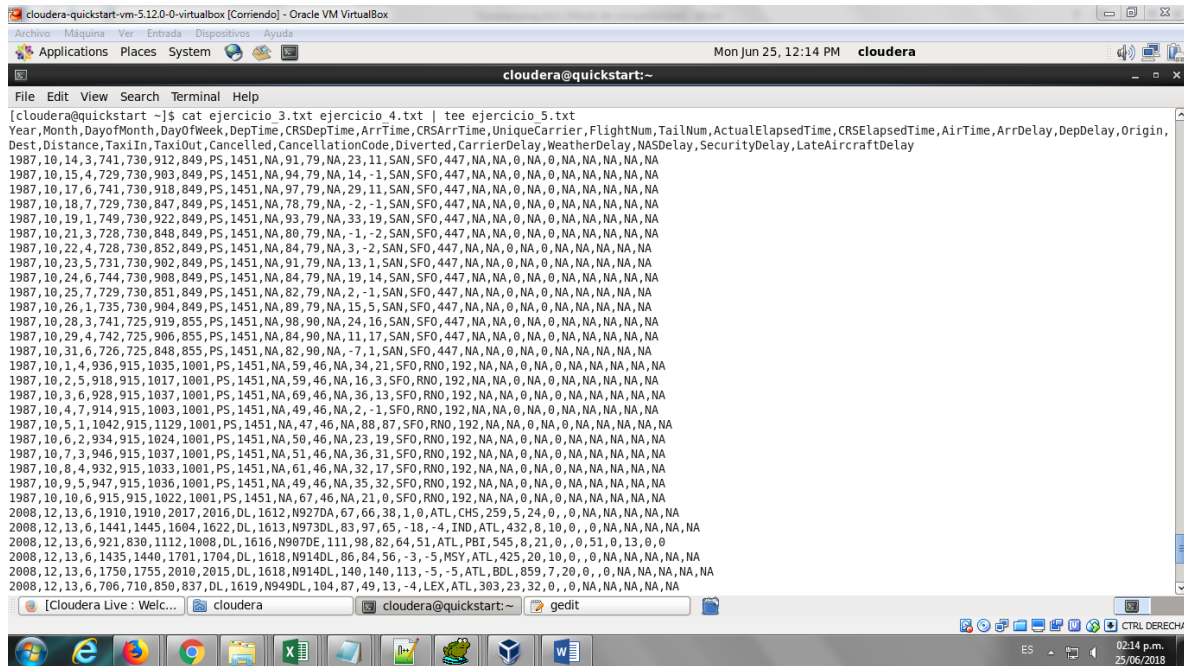
sed -e 'a -e '\$q;N;26,\$D;ba' aerolinea.csv > ejercicio\_4.txt

```
:a      # define label a
${      # match the last line
  P      # print the first line of the pattern space
  q      # quit
}
N      # match all lines: append the next line to the pattern
26,${  # match the range of lines 26 to the end of the file
  D      # delete the first line of the pattern space
}
ba     # match all lines: jump back to label a
```



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



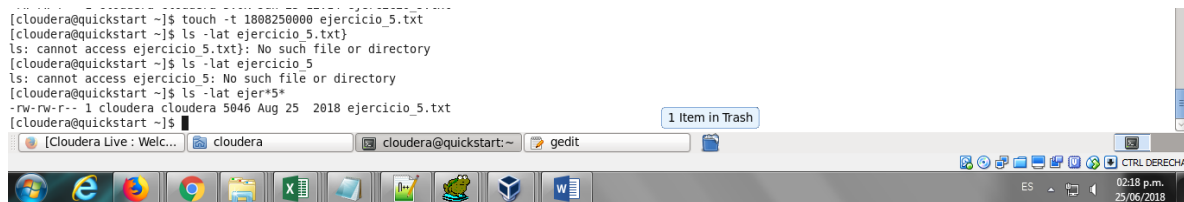
6.- Usando el comando `ls` y sus opciones, verifique el peso de **ejercicio\_5.txt**, señalando en la captura de pantalla dónde se encuentra éste.

`ls -lah ejercicio_5.txt`



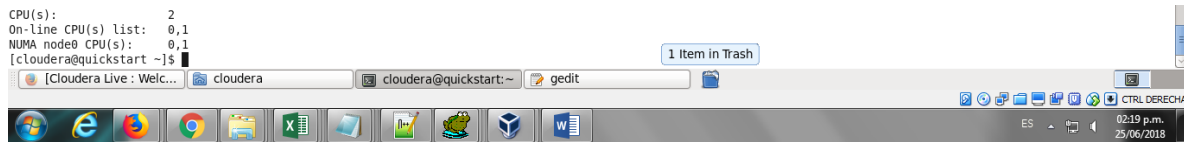
7.- Modifique la fecha de acceso de **ejercicio\_5.txt** al 25 de Agosto del 2018 y muestre en pantalla dónde se puede apreciar ese resultado.

`touch -t 1808250000 ejercicio_5.txt`



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)'`



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
be
ik
oq
aaron@aaron-trabajo-vb:~/Descargas$
```

AWK: Lee la entrada un renglón a la vez, cada renglón se compara con cada patrón en orden; para cada patrón que concuerde con el renglón se efectúa la acción correspondiente. Si se omite la acción, la acción por defecto consiste en imprimir los renglones que concordaron con el patrón y si se omite el patrón, la parte de la acción se hace en cada renglón de entrada. awk divide cada renglón de entrada en campos, (por defecto) cada campo estará separado por espacios, llama a los campos \$1, \$2, ..\$NF donde NF es una variable cuyo valor es igual al número de campos. Los patrones deben ir rodeados por caracteres / y puede contener dos patrones separados por una coma, en cuyo caso la acción se realizará para aquellas líneas comprendidas entre la primera aparición del primer patrón y la siguiente aparición del segundo patrón.

NR variable igual número de línea actual.

FILENAME nombre del archivo de la entrada.

-F especificamos que carácter queremos que tome como separador de campos.

BEGIN realiza acciones antes de procesar entrada por ejemplo: awk 'BEGIN {FS=":"}' el carácter separador será dos puntos :

awk '{print \$1","}' pp agrega un coma (,) al final del primer campo.

```
# cat pp
3 hola
2 pepe
2 mama
1 www
1 si
1 no
# awk '{print $1","}' pp
3,
2,
2,
```

1,

1,

1,

awk '{print \$0","}' pp agrega un coma (,) al final de cada línea.

# awk '{print \$0","}' pp

3 hola,

2 pepe,

2 mama,

1 www,

1 si,

1 no,

awk '\$1 ~ /1/' pp muestra todas las líneas cuyo primer campo contenga la cadena 1.

# awk '\$1 ~ /1/' pp

1 www

1 si

1 no

awk '\$2 ~ /a/' pp muestra todas las líneas cuyo segundo campo contenga la letra a

# awk '\$2 ~ /a/' pp

3 hola

2 mama

awk '\$2 !~ /a/' pp busca todas las líneas cuyo segundo campo no contenga la letra a

# awk '\$2 !~ /a/' pp

2 pepe

1 www

1 si

1 no

awk '/pepe/' ejemplo.txt busca todas las líneas que contenga la cadena pepe es equivalente a ejecutar grep pepe ejemplo.txt

# cat ejemplo.txt

mama

pepe

hola

Hola

pepe

Hola

si

no

Mama

www

# awk '/pepe/' ejemplo.txt

pepe

pepe

# grep pepe ejemplo.txt

pepe

pepe

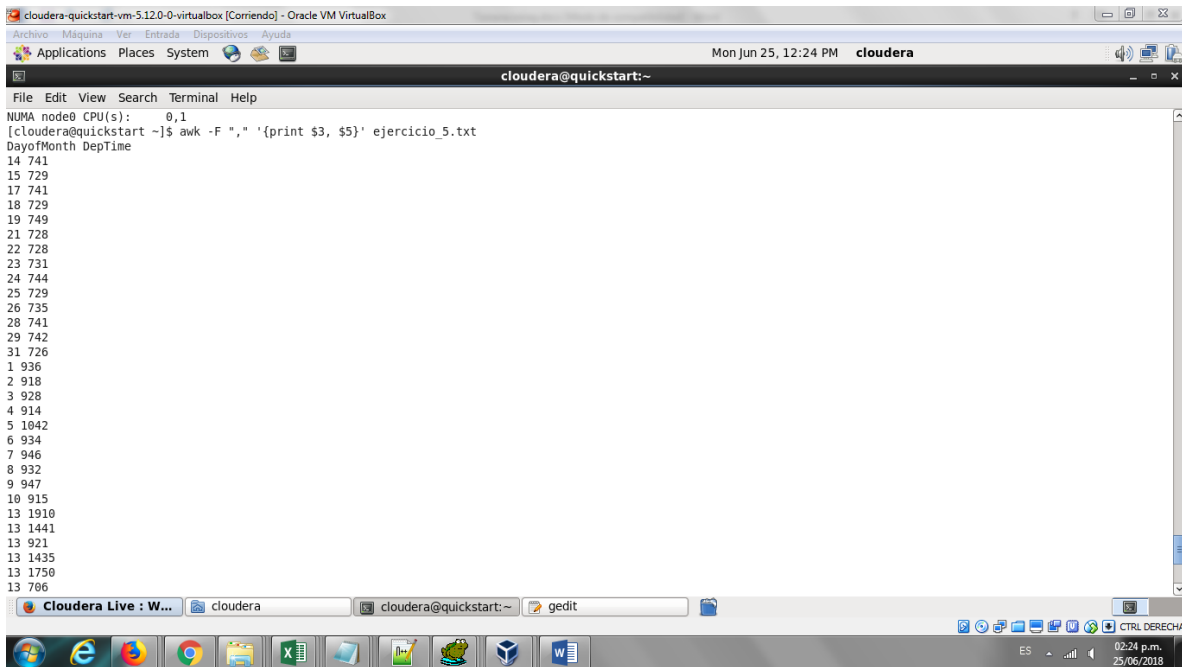
awk '/pepe/,/si/' ejemplo.txt busca todas las líneas existentes entre los patrones pepe y si

# awk '/pepe/,/si/' ejemplo.txt

pepe

hola  
Hola  
pepe  
Hola  
si

awk -F "," '{print \$3, \$5}' ejercicio\_5.txt

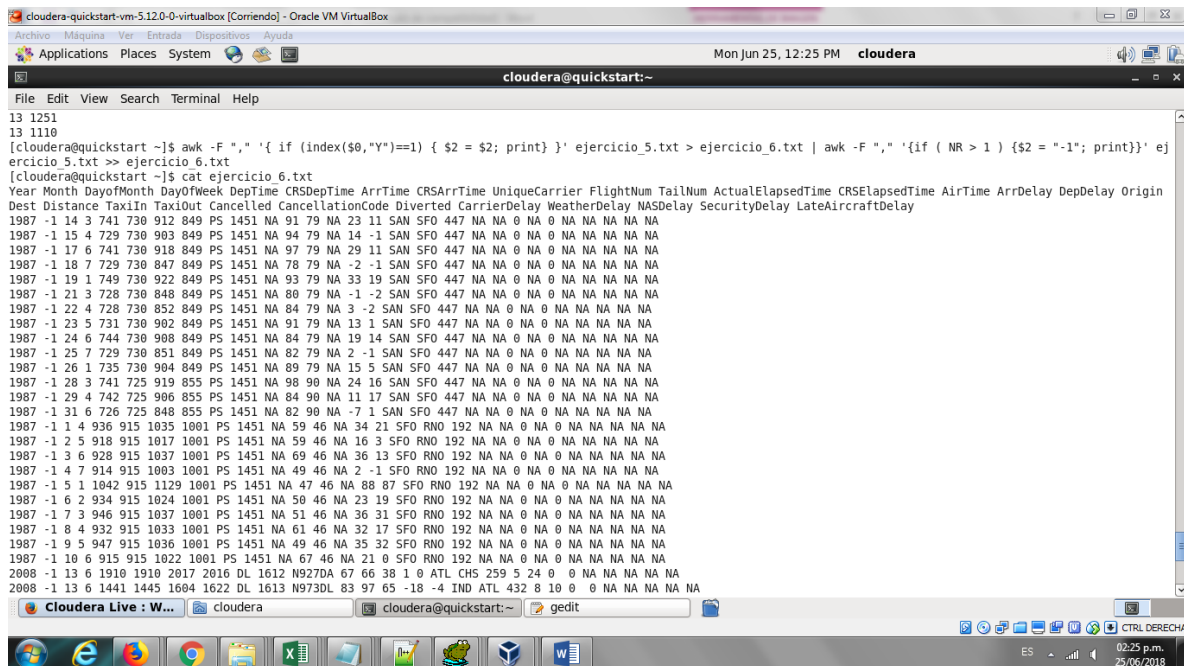


```
cloudera-quickstart-vm-5.12.0-0-virtualbox [Corriendo] - Oracle VM VirtualBox
Archivo Maquina Ver Entrada Dispositivos Ayuda
Applications Places System
Mon Jun 25, 12:24 PM cloudera
cloudera@quickstart:~
File Edit View Search Terminal Help
NUMA node0 CPU(s): 0,1
[cloudera@quickstart ~]$ awk -F "," '{print $3, $5}' ejercicio_5.txt
DayOfMonth DepTime
14 741
15 729
17 741
18 729
19 749
21 728
22 728
23 731
24 744
25 729
26 735
28 741
29 742
31 726
1 936
2 918
3 928
4 914
5 1042
6 934
7 946
8 932
9 947
10 915
13 1910
13 1441
13 921
13 1435
13 1750
13 706
```

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 (index(\$0,"Y")==1) { \$2 = \$2; print } }' ejercicio\_5.txt > ejercicio\_6.txt | awk -F "," '{if ( NR > 1 ) { \$2 = "-1"; print } }' ejercicio\_5.txt >> ejercicio\_6.txt





```
cloudera-quickstart-vm-5.12.0-0-virtualbox [Comando] - Oracle VM VirtualBox
Applications Places System
cloudera@quickstart:~
File Edit View Search Terminal Help
13 1251
13 1110
[cloudera@quickstart ~]$ awk -F "," '{ if (index($0,"Y")==1) { $2 = $2; print; } }' ejercicio_5.txt > ejercicio_6.txt | awk -F "," '{if ( NR > 1 ) { $2 = "-1"; print; } }' ej
ercicio_5.txt >> ejercicio_6.txt
[cloudera@quickstart ~]$ cat ejercicio_6.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 -1 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
1987 -1 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
1987 -1 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
1987 -1 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
1987 -1 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
1987 -1 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
1987 -1 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
1987 -1 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
1987 -1 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
1987 -1 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
1987 -1 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
1987 -1 28 3 741 725 919 855 PS 1451 NA 98 90 NA 24 16 SAN SFO 447 NA NA 0 NA 0 NA NA NA NA
1987 -1 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
1987 -1 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
1987 -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
1987 -1 2 5 918 915 1017 1001 PS 1451 NA 59 46 NA 16 3 SFO RNO 192 NA NA 0 NA 0 NA NA NA NA
1987 -1 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
1987 -1 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
1987 -1 5 1 1042 915 1129 1001 PS 1451 NA 47 46 NA 88 87 SFO RNO 192 NA NA 0 NA 0 NA NA NA NA
1987 -1 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
1987 -1 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
1987 -1 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
1987 -1 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
1987 -1 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
2008 -1 13 6 1910 1910 2017 2016 DL 1612 N927DA 67 66 38 1 0 ATL CHS 259 5 24 0 0 NA NA NA NA NA
2008 -1 13 6 1441 1445 1604 1622 DL 1613 N973DL 83 97 65 -18 -4 IND ATL 432 8 10 0 0 NA NA NA NA NA
```

## SECCION 2. HDFS Y HIVE

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

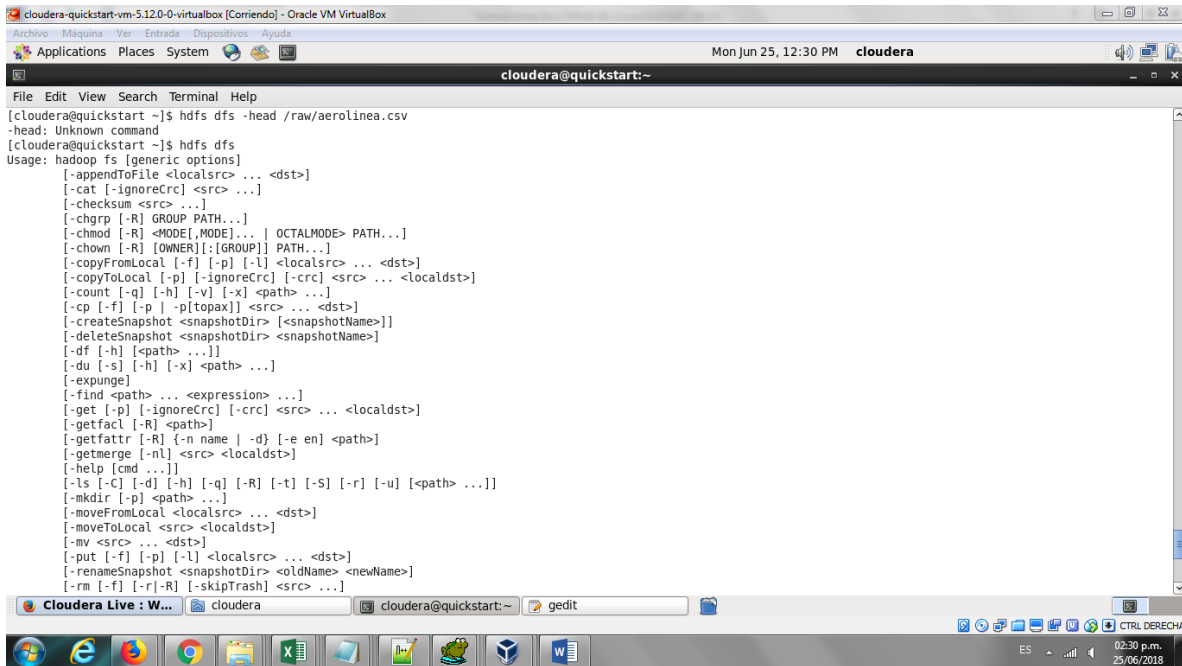
`hdfs dfs -head /raw/aerolinea.csv`

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 ~]$ hdfs dfs -head /raw/aerolinea.csv
-head: Unknown command
[cloudera@quickstart ~]$
```

Mediante el comando `hdfs dfs`, podemos verificar que comando podemos utilizar en el FileSystem de hdfs:



Como podemos observar solo existe en comando tail, el comando head no funciona en FS.

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

```
[cloudera@quickstart ~]$ hdfs dfs -cat raw/aerolineas.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.

```
hdfs getconf -confKey dfs.replication
```

```
[cloudera@quickstart ~]$ hdfs getconf -confKey dfs.replication
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
```

```
[cloudera@quickstart aerolineas]$ hdfs dfs -cat /raw/aerolinea.csv | sed 15q | tee ejercicio_14.txt
Year,Month,DayOfMonth,DayOfWeek,DepTime,CRSDepTime,ArrTime,CRSArrTime,UniqueCarrier,FlightNum,TailNum,ActualElapsedTime,Distance,TaxiIn,TaxiOut,Canceled,CancellationCode,Diverted,CarrierDelay,WeatherDelay,NASDelay,SecurityDelay
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,98,90,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
1987,10,31,6,726,725,848,855,PS,1451,NA,82,90,NA,-7,1,SAN,SFO,447,NA,NA,0,NA,0,NA,NA,NA,NA,NA
```

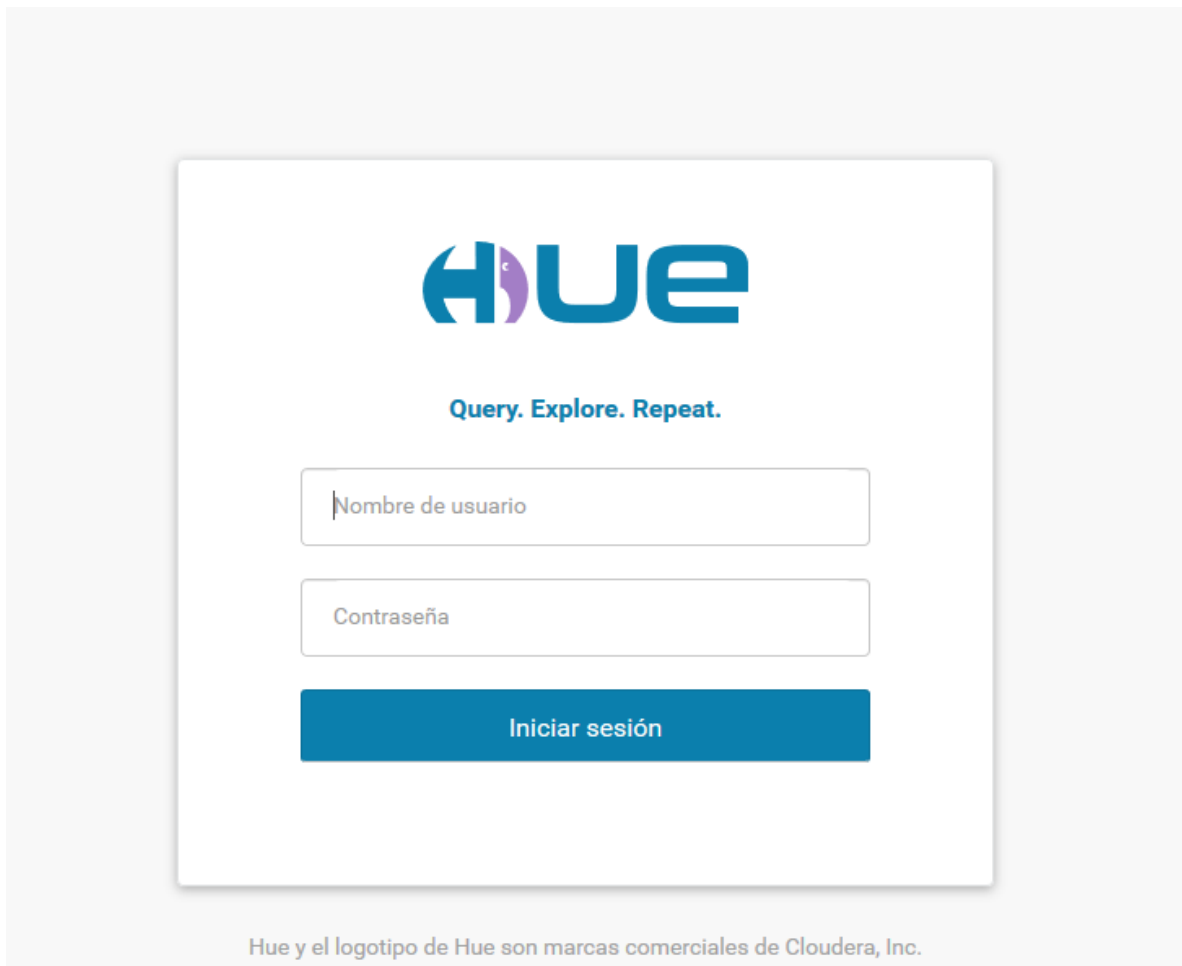
15.- Cree los directorios **master** y **stagin** en el directorio raíz del HDFS y además al archivo aerolínea.csv que está en raw cámbiele los permisos de tal manera que el propietario tenga todas las facilidades sobre él, el grupo sólo pueda leer y escribir y cualquier otro no tenga ningún permiso.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

```
[cloudera@quickstart aerolineas]$ sudo -u hdfs hdfs dfs -mkdir /master
[cloudera@quickstart aerolineas]$ sudo -u hdfs hdfs dfs -mkdir /stagin
[cloudera@quickstart aerolineas]$ sudo -u hdfs hdfs dfs -chown cloudera /master
[cloudera@quickstart aerolineas]$ sudo -u hdfs hdfs dfs -chown cloudera /stagin
[cloudera@quickstart aerolineas]$ hdfs dfs -ls /
Found 9 items
drwxrwxrwx - hdfs supergroup 0 2017-07-19 05:34 /benchmarks
drwxr-xr-x - hbase supergroup 0 2018-06-19 08:53 /hbase
drwxr-xr-x - cloudera supergroup 0 2018-06-19 15:46 /master
drwxr-xr-x - cloudera supergroup 0 2018-06-13 16:38 /raw
drwxr-xr-x - solr solr 0 2017-07-19 05:37 /solr
drwxr-xr-x - cloudera supergroup 0 2018-06-19 15:46 /stagin
drwxrwxrwt - hdfs supergroup 0 2018-06-13 16:42 /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 aerolineas]$
```

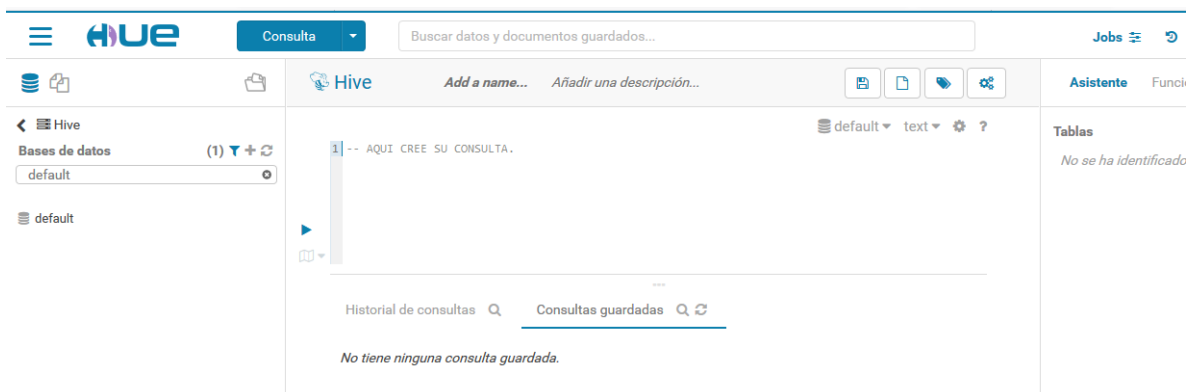
```
[cloudera@quickstart aerolineas]$ hdfs dfs -chmod 760 /raw/aerolinea.csv
[cloudera@quickstart aerolineas]$ hdfs dfs -ls /raw
Found 1 items
-rwxrw---- 1 cloudera supergroup 12029208594 2018-06-13 16:38 /raw/aerolinea.csv
[cloudera@quickstart aerolineas]$
```

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'
```

```
tblproperties ("skip.header.line.count"="1");
```

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.

```
1 SELECT *
2 FROM tabla_aerolinea
3 LIMIT 10;
4
```

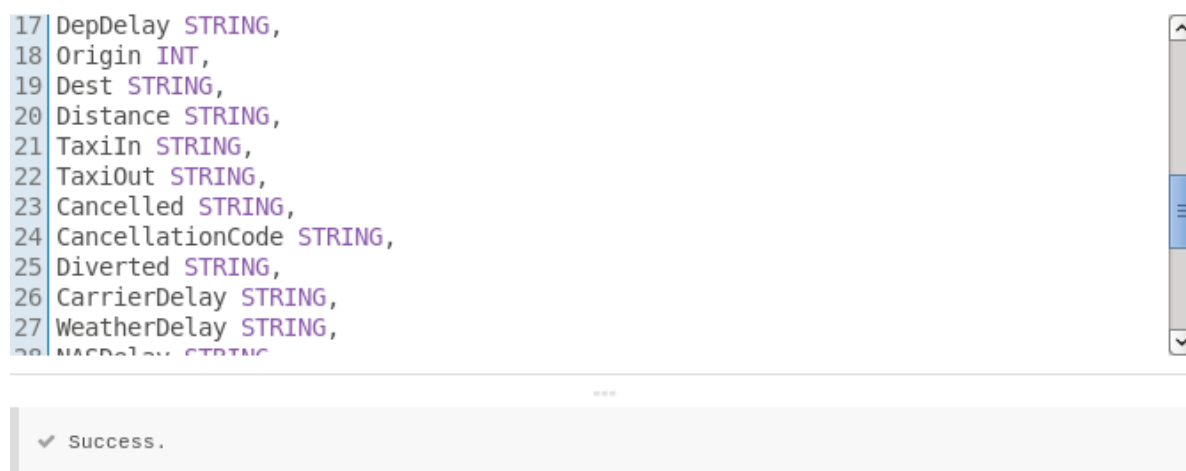


The screenshot shows a database interface with a query history bar at the top containing 'Query History', 'Saved Queries', and 'Results (10)'. Below this is a table with the following structure:

	tabla_aerolinea.year	tabla_aerolinea.month	tabla_aerolinea.dayofmonth	tabl
1	1987	10	14	3
2	1987	10	15	4
3	1987	10	17	6
4	1987	10	18	7

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.

La tabla se creo:



The screenshot shows a database interface with a query editor containing the following SQL statement:

```
17 DepDelay STRING,
18 Origin INT,
19 Dest STRING,
20 Distance STRING,
21 TaxiIn STRING,
22 TaxiOut STRING,
23 Cancelled STRING,
24 CancellationCode STRING,
25 Diverted STRING,
26 CarrierDelay STRING,
27 WeatherDelay STRING,
28 NASDelay STRING
```

Below the query editor, a success message is displayed: "✓ Success."

El dato de Origin es de tipo alfanumérico, por ello todos los datos de la columna Origin se reportan como NULL:

```

1 SELECT ORIGIN
2 FROM tabla_aerolinea
3 LIMIT 10;

```

Query History 🔍 📄 Saved Queries 🔍 ↺ Results (10) 🔍 ↗

	origin
1	NULL
2	NULL
3	NULL

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.

La tabla se creo sin errores:

```

28 NASDelay STRING,
29 SecurityDelay STRING,
30 LateAircraftDelay STRING,
31 Adicional STRING)
32 ROW FORMAT DELIMITED
33 FIELDS TERMINATED BY ','
34 STORED AS TEXTFILE
35 location '/raw'
36 tblproperties ("skip.header.line.count"="1");
37

```

✓ Success.

Sin embargo, el campo adicional tiene valores nulos:

```
1 SELECT LateAircraftDelay, ADICIONAL
2 FROM tabla_aerolinea
3 LIMIT 10;
4
```

lateaircraftdelay		adicional
1	NA	NULL
2	NA	NULL
3	NA	NULL

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.

[illegible]