

Computación en Física

Harvey Rodriguez Gil

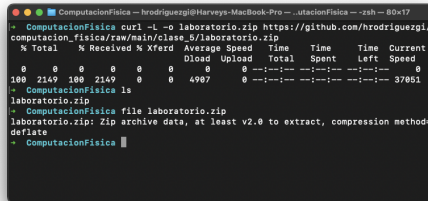
Universidad EIA

25 de Julio de 2024

cURL

cURL es un programa que permite transferir archivos a través de la línea de comandos utilizando una URL.

Para descargar un archivo lo podemos realizar de la siguiente forma: `curl -L -o nombre_archivo https://url_archivo`



```
ComputacionFisica — hrodriguezgi@Harveys-MacBook-Pro — ..utacionFisica — -zsh — 80x17
+ ComputacionFisica curl -L -o laboratorio.zip https://github.com/hrodriguezgi/
computacion_fisica/raw/main/clase_5/laboratorio.zip
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
  0     0    0     0    0     0      0      0  --:--:-- --:--:-- --:--:--    0
100 2149 100 2149    0     0  4907    0  --:--:-- --:--:-- --:--:-- 37051
+ ComputacionFisica ls
laboratorio.zip
+ ComputacionFisica file laboratorio.zip
laboratorio.zip: Zip archive data, at least v2.0 to extract, compression method=
deflate
+ ComputacionFisica
```

wget

wget es otro programa que nos permite realizar la descarga de ficheros desde internet a través de línea de comandos.

Para descargar un archivo lo podemos realizar de la siguiente forma: `wget https://url_archivo -O nombre_archivo`

```
ComputacionFisica — hrodriguezgi@Harveys-MacBook-Pro — ..utacionFisica — -zsh — 80x24
+ ComputacionFisica wget https://github.com/hrodriguezgi/computacion_fisica/raw
/main/clase_5/laboratorio.zip -O laboratorio.zip
--2024-07-24 18:58:42-- https://github.com/hrodriguezgi/computacion_fisica/raw
/main/clase_5/laboratorio.zip
Resolving github.com (github.com)... 140.82.114.4
Connecting to github.com (github.com)[140.82.114.4]:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/hrodriguezgi/computacion_fisica/main
/clase_5/laboratorio.zip [following]
--2024-07-24 18:58:43-- https://raw.githubusercontent.com/hrodriguezgi/computac
ion_fisica/main/clase_5/laboratorio.zip
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.109.1
33, 185.199.110.133, 185.199.111.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)[185.199.109.
133]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2149 (2.1K) [application/zip]
Saving to: 'laboratorio.zip'

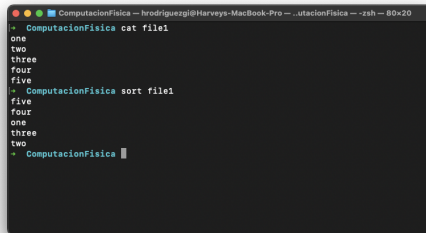
laboratorio.zip 100%[=====>>>] 2.10K --KB/s in 0.001s

2024-07-24 18:58:43 (3.47 MB/s) - 'laboratorio.zip' saved [2149/2149]

+ ComputacionFisica █
```

sort

sort Es un comando que nos permite ordenar la información. Por defecto el ordenamiento es a través de las reglas del lenguaje

A terminal window titled 'ComputacionFisica' with a user prompt 'hrodriguezgi@Harveys-MacBook-Pro'. The window shows the execution of 'cat file1' which outputs 'one', 'two', 'three', 'four', 'five'. Then 'sort file1' is executed, resulting in the output 'five', 'four', 'one', 'three', 'two'.

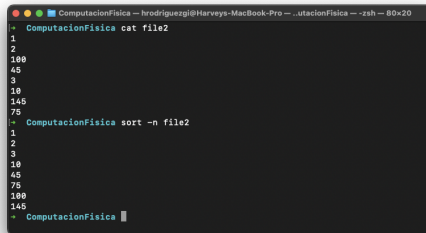
```
ComputacionFisica -- hrodriguezgi@Harveys-MacBook-Pro -- -zsh -- 80x20
+ ComputacionFisica cat file1
one
two
three
four
five
+ ComputacionFisica sort file1
five
four
one
three
two
+ ComputacionFisica
```

sort

```
ComputacionFisica — hrodriguezgi@Harveys-MacBook-Pro — ..utacionFisica — zsh — 80x20
+ ComputacionFisica cat file2
1
2
100
45
3
10
145
75
+ ComputacionFisica sort file2
1
10
100
145
2
3
45
75
+ ComputacionFisica
```

sort

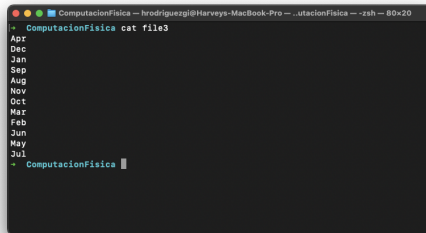
Cuando el contenido del archivo contiene números, debemos utilizar el flag `-n` para que el ordenamiento sea numérico.

A terminal window titled 'ComputacionFisica' showing the execution of the 'sort' command. The first command is 'cat file2', which displays a list of numbers: 1, 2, 100, 45, 3, 10, 145, 75. The second command is 'sort -n file2', which displays the same numbers sorted numerically: 1, 2, 3, 10, 45, 75, 100, 145.

```
ComputacionFisica -- hrodriguezgi@Harveys-MacBook-Pro -- -zsh -- 80x20
+ ComputacionFisica cat file2
1
2
100
45
3
10
145
75
+ ComputacionFisica sort -n file2
1
2
3
10
45
75
100
145
+ ComputacionFisica
```

sort

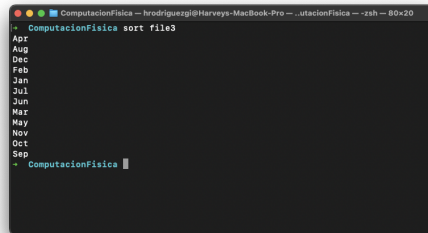
En este otro ejemplo, vemos las iniciales de los 12 meses del año sin ningún orden:



```
ComputacionFisica -- hrodriguezgi@Harveys-MacBook-Pro -- .utacionFisica -- zsh -- 80x20
+ ComputacionFisica cat file3
Apr
Dec
Jan
Sep
Aug
Nov
Oct
Mar
Feb
Jun
May
Jul
+ ComputacionFisica
```

sort

Al aplicarle el ordenamiento, este se hace alfabeticamente, lo cual no es la salida esperada:



```
ComputacionFisica -- hrodriguezgi@Harveys-MacBook-Pro -- ..utacionFisica -- -zsh -- 80x20
+ ComputacionFisica sort file3
Apr
Aug
Dec
Feb
Jan
Jul
Jun
Mar
May
Nov
Oct
Sep
+ ComputacionFisica
```


sort

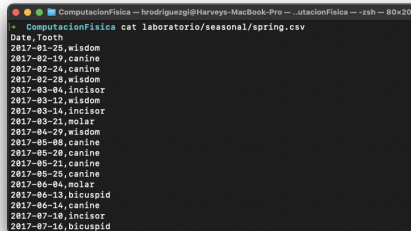
Cuando el contenido del archivo contiene los meses del año, podemos utilizar el flag `-M` para que el ordenamiento sea el adecuado.



```
ComputacionFisica -- hrodriguezgi@Harveys-MacBook-Pro -- .utacionFisica -- zsh -- 80x20
+ ComputacionFisica sort -M file3
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
+ ComputacionFisica
```

sort

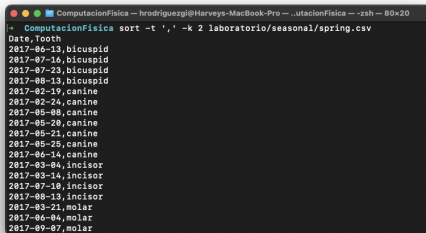
Otras opciones que tiene sort es el uso de `-t` para indicarle el caracter separador, y `-k` para indicarle por cuál campo ordenar la salida



```
ComputacionFisica -- hrodriguezgi@Harveys-MacBook-Pro -- .utacionFisica -- zsh -- 80x20
ComputacionFisica cat laboratorio/seasonal/spring.csv
Date,Tooth
2017-01-25,wisdom
2017-02-19,canine
2017-02-24,canine
2017-02-28,wisdom
2017-03-04,incisor
2017-03-12,wisdom
2017-03-14,incisor
2017-03-21,molar
2017-04-29,wisdom
2017-05-08,canine
2017-05-20,canine
2017-05-21,canine
2017-05-25,canine
2017-06-04,molar
2017-06-13,bicuspid
2017-06-14,canine
2017-07-10,incisor
2017-07-16,bicuspid
```

sort

En este ejemplo, vemos como la segunda columna está ordenada correctamente: bicuspid, canine, incisor, molar



```
ComputacionFisica -- hrodriguezgi@Harveys-MacBook-Pro -- ..utacionFisica -- -zsh -- 80x20
+ ComputacionFisica sort -t ',' -k 2 laboratorio/seasonal/spring.csv
Date,Tooth
2017-06-13,bicuspid
2017-07-16,bicuspid
2017-07-23,bicuspid
2017-08-13,bicuspid
2017-02-19,canine
2017-02-24,canine
2017-05-08,canine
2017-05-20,canine
2017-05-21,canine
2017-05-25,canine
2017-06-14,canine
2017-03-04,incisor
2017-03-14,incisor
2017-07-10,incisor
2017-08-13,incisor
2017-03-21,molar
2017-06-04,molar
2017-09-07,molar
```

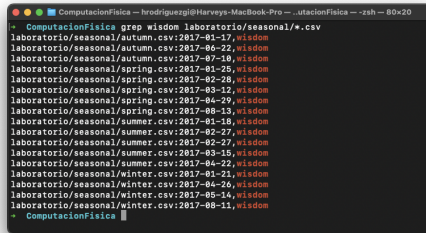
sort

Otras opciones del comando `sort` son:

- ▶ `-f` No discrimina entre mayúsculas y minúsculas. Por defecto mayúsculas siempre van de primero
- ▶ `-k` Define el campo por el cuál se realiza el ordenamiento
- ▶ `-M` Ordenamiento de los meses (tres primeras letras)
- ▶ `-n` Ordenamiento numérico
- ▶ `-r` Ordenamiento inverso
- ▶ `-t` Caracter separador de campos

grep

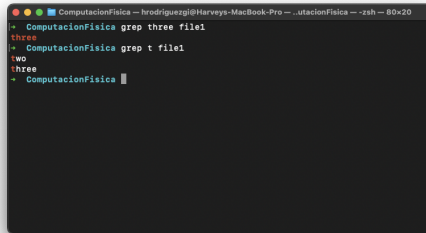
grep Es un comando que nos permite ubicar texto en archivos. En lugar de abrir el archivo y buscar manualmente un texto en específico, podemos validar con este comando si existe y en qué líneas se encuentra el texto.



```
ComputacionFisica — hrodriguezgi@Harveys-MacBook-Pro — .utacionFisica — zsh — 80x20
+ ComputacionFisica grep wisdom laboratorio/seasonal/*.csv
laboratorio/seasonal/autumn.csv:2017-01-17,wisdom
laboratorio/seasonal/autumn.csv:2017-06-22,wisdom
laboratorio/seasonal/autumn.csv:2017-07-10,wisdom
laboratorio/seasonal/spring.csv:2017-01-25,wisdom
laboratorio/seasonal/spring.csv:2017-02-28,wisdom
laboratorio/seasonal/spring.csv:2017-03-12,wisdom
laboratorio/seasonal/spring.csv:2017-04-29,wisdom
laboratorio/seasonal/spring.csv:2017-08-13,wisdom
laboratorio/seasonal/summer.csv:2017-01-18,wisdom
laboratorio/seasonal/summer.csv:2017-02-27,wisdom
laboratorio/seasonal/summer.csv:2017-02-27,wisdom
laboratorio/seasonal/summer.csv:2017-03-15,wisdom
laboratorio/seasonal/summer.csv:2017-04-22,wisdom
laboratorio/seasonal/winter.csv:2017-01-21,wisdom
laboratorio/seasonal/winter.csv:2017-04-26,wisdom
laboratorio/seasonal/winter.csv:2017-05-14,wisdom
laboratorio/seasonal/winter.csv:2017-08-11,wisdom
+ ComputacionFisica
```

grep

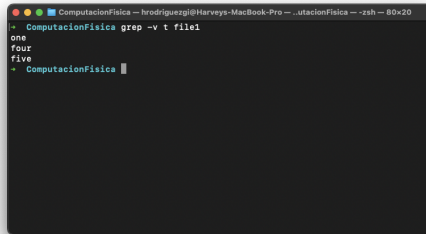
grep nos devuelve las ocurrencias del texto buscado, sea que lo encuentre completamente o como parte de otro texto (palabra)

A terminal window titled 'ComputacionFisica' with a user prompt 'hrodriguezgi@Harveys-MacBook-Pro'. It shows two commands: 'grep three file1' which outputs 'three' (highlighted in red), and 'grep t file1' which outputs 'two' and 'three' (both highlighted in red). The prompt is currently at 'ComputacionFisica' with a cursor.

```
ComputacionFisica — hrodriguezgi@Harveys-MacBook-Pro — .utacionFisica — -zsh — 80x20
+ ComputacionFisica grep three file1
three
+ ComputacionFisica grep t file1
two
three
+ ComputacionFisica
```

grep

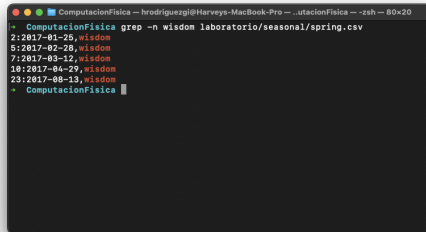
grep nos permite hacer búsquedas inversas, esto quiere decir que la salida sea donde el texto buscado no se encuentre, esto con ayuda del flag `-v`



```
ComputacionFisica -- hrodriguezgi@Harveys-MacBook-Pro -- ..utacionFisica -- -zsh -- 80x20
+ ComputacionFisica grep -v t file1
one
four
five
+ ComputacionFisica
```

grep

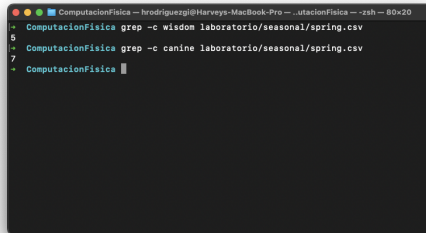
Si queremos saber los números de línea donde nuestro texto aparece, basta con agregar el flag `-n`



```
ComputacionFisica -- hrodriguezgi@Harveys-MacBook-Pro -- ..utacionFisica -- -zsh -- 80x20
+ ComputacionFisica grep -n wisdom laboratorio/seasonal/spring.csv
2:2017-01-25,wisdom
5:2017-02-28,wisdom
7:2017-03-12,wisdom
10:2017-04-29,wisdom
23:2017-08-13,wisdom
+ ComputacionFisica
```


grep

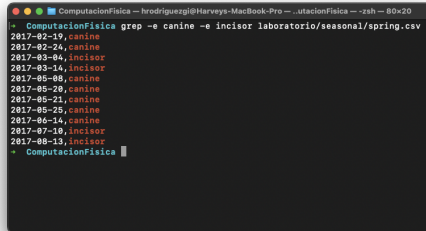
El flag `-c` nos permite contar la cantidad de ocurrencias del texto buscado



```
ComputacionFisica — hrodriguezgi@Harveys-MacBook-Pro — .utacionFisica — zsh — 80x20
+ ComputacionFisica grep -c wisdom laboratorio/seasonal/spring.csv
5
+ ComputacionFisica grep -c canine laboratorio/seasonal/spring.csv
7
+ ComputacionFisica
```

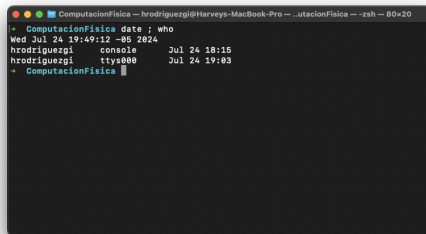
grep

El flag `-e` nos permite definir múltiples textos a buscar

A terminal window titled 'ComputacionFisica' with the prompt 'hrodriguezgi@Harveys-MacBook-Pro ~.utacionFisica - zsh - 80x20'. The command 'ComputacionFisica grep -e canine -e incisor laboratorio/seasonal/spring.csv' is entered. The output lists dates and animal types: 2017-02-19,canine; 2017-02-24,canine; 2017-03-04,incisor; 2017-03-14,incisor; 2017-05-08,canine; 2017-05-20,canine; 2017-05-21,canine; 2017-05-25,canine; 2017-06-14,canine; 2017-07-10,incisor; 2017-08-13,incisor. The prompt returns to 'ComputacionFisica'.

Usando múltiples comandos

Linux nos permite ejecutar múltiples comandos en una sola línea de código, obteniendo como resultado la salida de cada uno:



```
ComputacionFisica -- hrodriguezgi@Harveys-MacBook-Pro -- .utacionFisica -- -zsh -- 80x20
+ ComputacionFisica date ; who
Wed Jul 24 19:49:12 -05 2024
hrodriguezgi      console      Jul 24 18:15
hrodriguezgi      ttys000      Jul 24 19:03
+ ComputacionFisica
```

Bash Script

Si tomamos todos los comandos que hemos visto hasta ahora, y quisieramos reunirlos en un lugar, sería en un script de bash. El cual tiene la siguiente estructura con los dos comandos anteriores:

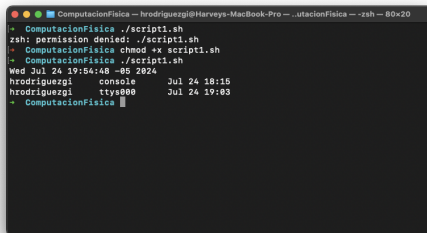
A screenshot of a macOS terminal window titled "ComputacionFisica -- vim script1.sh -- vim -- vim script1.sh -- 80x20". The terminal shows the following sequence of actions:

- The prompt is "#!/bin/bash".
- A comment "# Este script muestra la fecha actual y quien esta logueado en el sistema" is present.
- The command "date" is entered, resulting in the output "Sat Aug 17 19:06:05 CDT 2019".
- The command "who" is entered, resulting in the output "root tty1 0 0.00 0.00 root /dev/tty1 0.00 0.00".

The cursor is at the end of the "who" command line. At the bottom of the screen, it says "-- INSERT --".

Bash Script

Si lo queremos ejecutar, debemos garantizar que tenga permisos de ejecución (`chmod +x nombre_script`)



```
ComputacionFisica — hrodriguezgi@Harveys-MacBook-Pro — .utacionFisica — -zsh — 80x20
+ ComputacionFisica ./script1.sh
zsh: permission denied: ./script1.sh
+ ComputacionFisica chmod +x script1.sh
+ ComputacionFisica ./script1.sh
Wed Jul 24 19:54:48 -05 2024
hrodriguezgi      console          Jul 24 18:15
hrodriguezgi      ttys000          Jul 24 19:03
+ ComputacionFisica
```

Vamos a practicar

Crear un script de bash en el que ejecute las siguientes acciones:

- ▶ Descargue el archivo `data.txt` que se encuentra en el repositorio de Github en la carpeta `clase_6/laboratorio/`, ya sea utilizando `wget` o `curl`.
- ▶ Ordene por la columna 1 de forma numérica y luego por la columna 2 el archivo `data.txt`

Vamos a practicar

Crear un script de bash en el que ejecute las siguientes acciones:

- ▶ Descargue el archivo `log.txt` que se encuentra en el repositorio de Github en la carpeta `clase_6/laboratorio/`, ya sea utilizando `wget` o `curl`.
- ▶ Indicar el total de líneas que contienen la palabra `error` ignorando las mayúsculas y minúsculas.
- ▶ Imprimir en pantalla el número y la línea en las que aparece la palabra `error` ignorando las mayúsculas y minúsculas.