

# Programación con Shell Scripting: Sesión 4

Máster Universitario en Bioinformática



**Universidad**  
Internacional  
de Valencia

Dra. Paula Soler Vila  
[paula.solerv@professor.universidadviu.com](mailto:paula.solerv@professor.universidadviu.com)

De:  
 Planeta Formación y Universidades

## Aspectos a tratar

1

Teoría y Práctica:

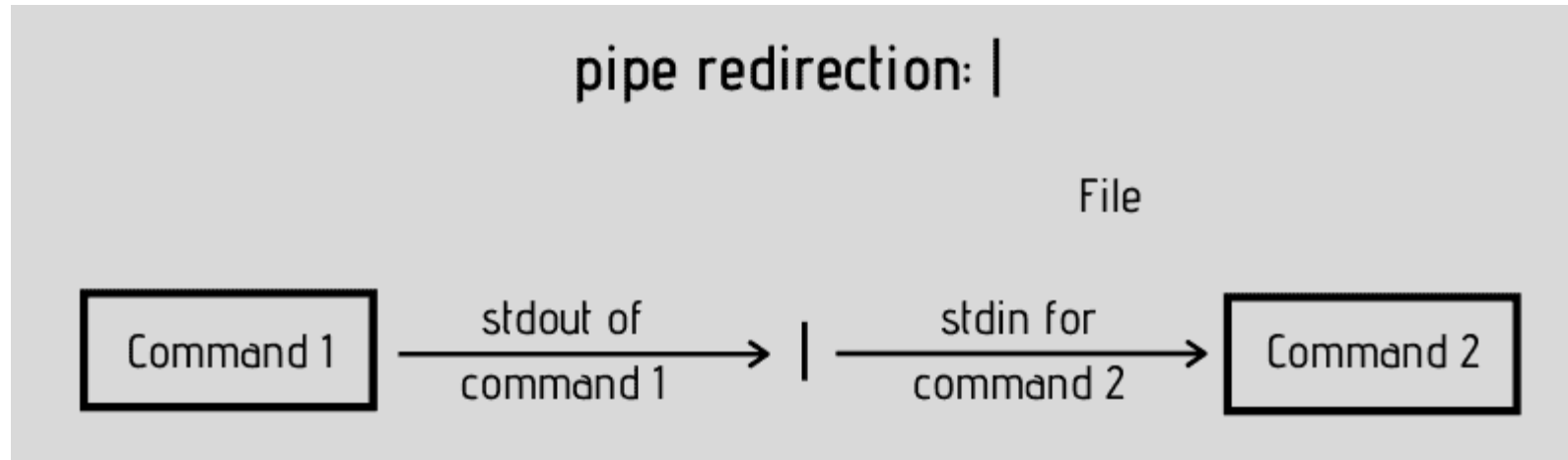
- Ejecuciones concatenadas (|)
  - **Comando tee**
- Ejecuciones secuenciales (;)
- Ejecuciones condicionales (&& , ||)

2

Comandos para el procesamiento de archivos

- *Cat*
- *Paste*
- *Join*
- *Diff*
- *Cut*
- *Sort*
- *Uniq*
- *Tr...*

## Ejecuciones Concatenadas -> Tubería o pipe



Para concatenar instrucciones

```
$ cat /etc/passwd | less
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
:
```

## Tubería o pipe

```
$ cat /etc/passwd | sort | head -n 5
```



Visualizamos (**cat**) el contenido del archivo **passwd** que se encuentra en el directorio **etc** que se encuentra en el directorio raíz (/)

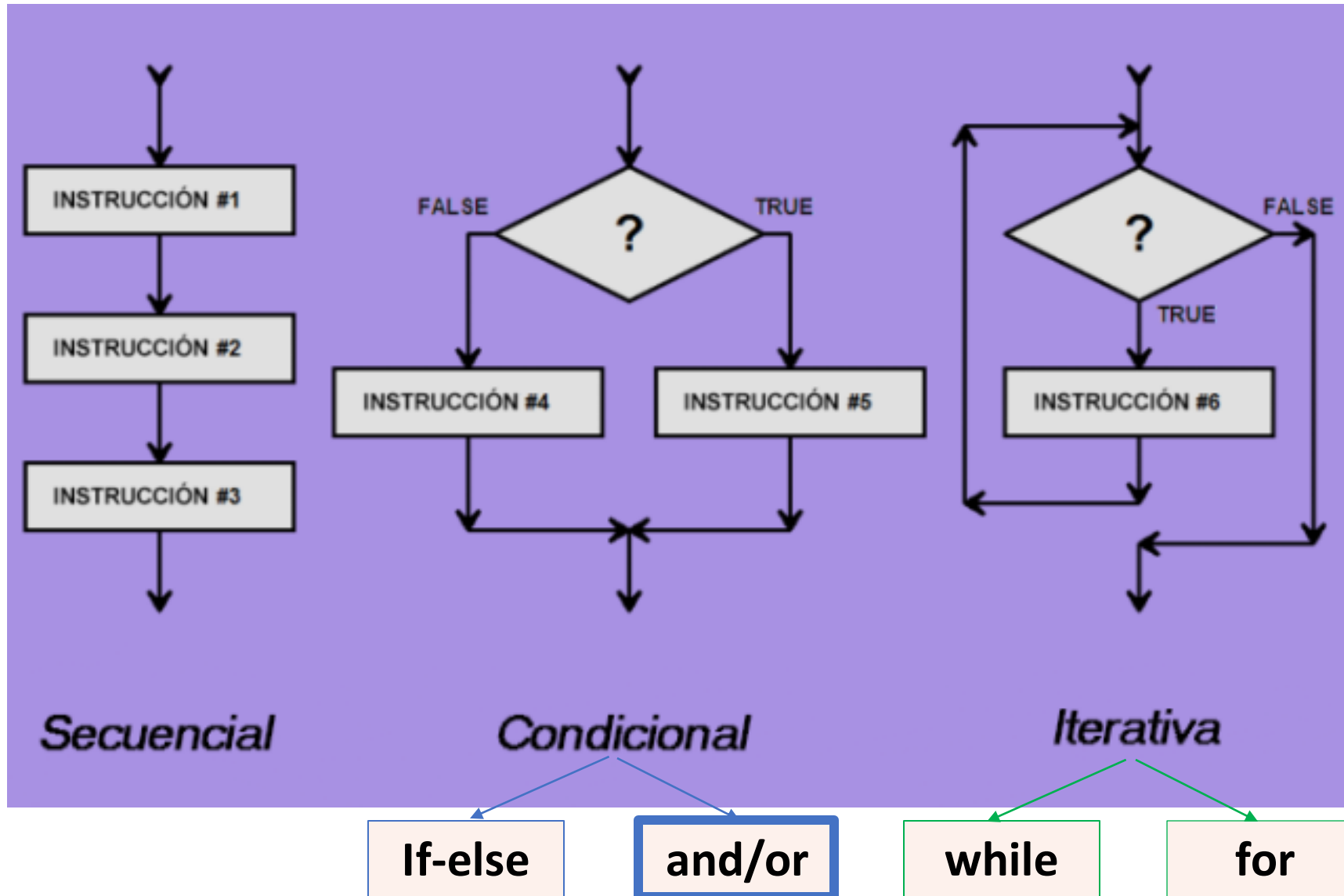


El resultado del comando **cat** se usa como entrada del comando **sort**  
Se ordena la información alfabéticamente por el primer campo del archivo passwd

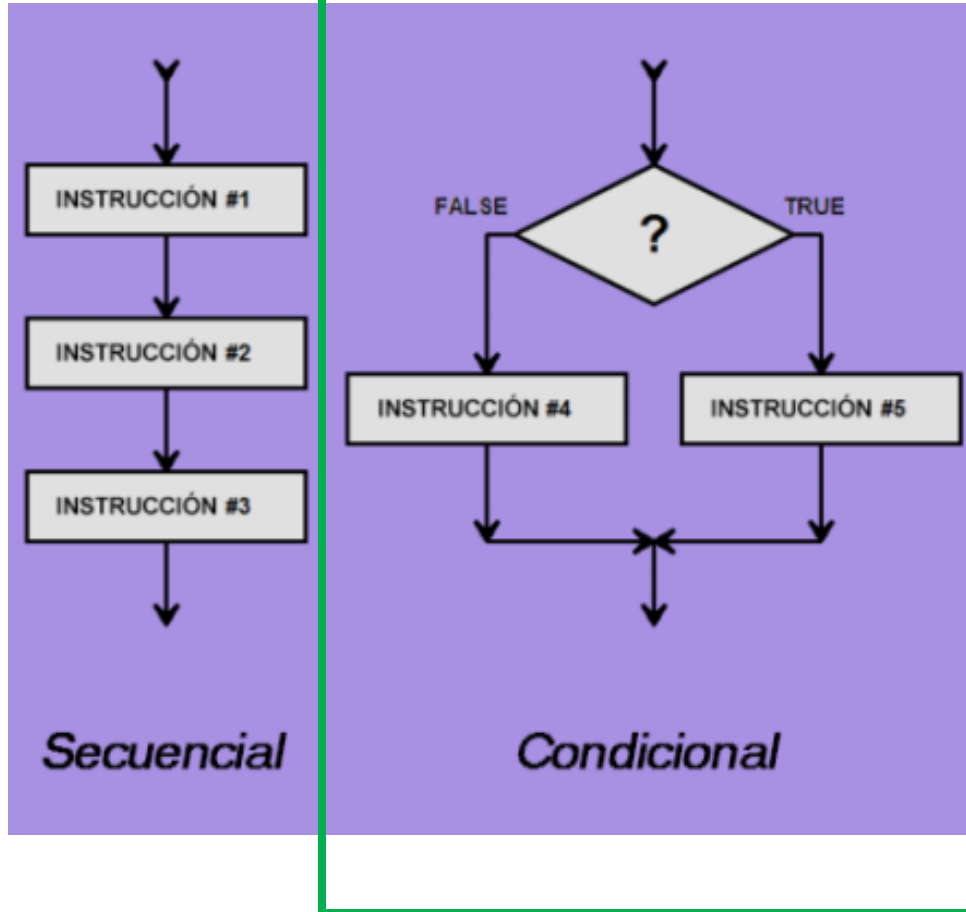


La salida ordenada de **sort** se usa como entrada en el comando **head** que nos mostrará solo las primeras 5 líneas

# Ejecuciones de comandos: Estructuras de control



# Ejecuciones de comandos



## 1. Ejecución secuencial

**comando1 ; comando2 ; comando3**

- La ejecución de un comando **no** está condicionada por el resultado o código de retorno del anterior.

## 2. Ejecución condicional

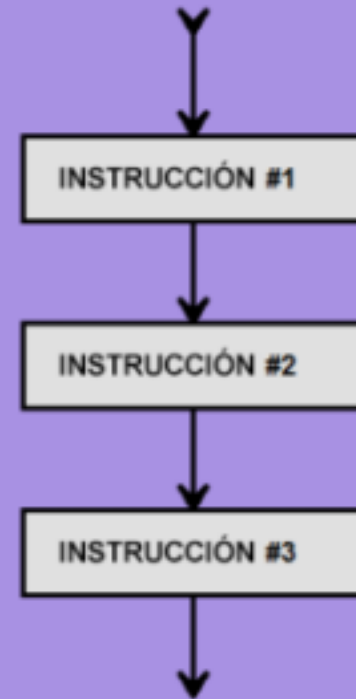
- Operador lógico Y (&&):** El comando anterior ha de finalizar exitosamente

**\$ comando1 && comando2**

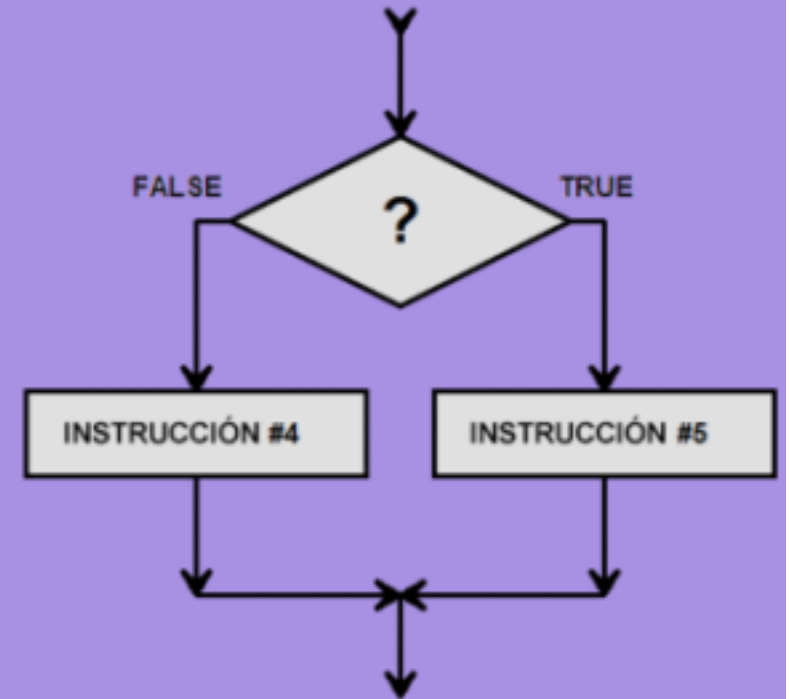
- Operador lógico O (||):** El comando anterior **no** ha de finalizar exitosamente

**\$ comando1 || comando2**

# PRACTIQUEMOS



*Secuencial*



*Condicional*

# Entrada estándar – Salida estándar – Error estándar

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cat /etc/passwd
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cat < /etc/passwd
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ wc -l << final
> uno
> dos
> tres
> final
3
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ head -n 5 /etc/passwd > head_passwd.
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ head -n 1 /etc/passwd > head_passwd.
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ head -n 2 /etc/passwd >> head_passwd.
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cat f20
cat: f20: No such file or directory
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cat f20 2> error.log
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cat f20 > salida.txt 2> error.log
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cat salida.txt
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cat error.log
cat: f20: No such file or directory
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cat head_passwd2 > salida.txt 2> error.log
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cat salida.txt
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cat error.log
```

es **cat /etc/passwd** (el cat separado de la barra)

comando **cat > fichero.txt** para poder crear y escribir texto en un archivo y control D para salir y guardar el archivo de texto

**cat >> fichero.txt** para escribir más en ese archivo sin que se sobrescriba lo otro



# Comando tee

```
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ head -n 5 /etc/passwd | tee head_passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
```

```
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ ping www.google.es | tee ping.txt
PING www.google.es (209.85.203.94) 56(84) bytes of data.
64 bytes from dh-in-f94.1e100.net (209.85.203.94): icmp_seq=1 ttl=104 time=1.36 ms
64 bytes from dh-in-f94.1e100.net (209.85.203.94): icmp_seq=2 ttl=104 time=1.40 ms
64 bytes from dh-in-f94.1e100.net (209.85.203.94): icmp_seq=3 ttl=104 time=2.01 ms
64 bytes from dh-in-f94.1e100.net (209.85.203.94): icmp_seq=4 ttl=104 time=1.39 ms
^C
```

```
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cat ping.txt
PING www.google.es (209.85.203.94) 56(84) bytes of data.
64 bytes from dh-in-f94.1e100.net (209.85.203.94): icmp_seq=1 ttl=104 time=1.36 ms
64 bytes from dh-in-f94.1e100.net (209.85.203.94): icmp_seq=2 ttl=104 time=1.40 ms
64 bytes from dh-in-f94.1e100.net (209.85.203.94): icmp_seq=3 ttl=104 time=2.01 ms
64 bytes from dh-in-f94.1e100.net (209.85.203.94): icmp_seq=4 ttl=104 time=1.39 ms
```

Con ping es para mirar la conexión a internet que tienes en una página web, por ejemplo google

Con tee le estás diciendo que lo guarde en un archivo

Para que termine de ejecutar una orden se le da control C

El comando TEE se utiliza para redirigir la salida de un comando a un archivo y, al mismo tiempo, mostrar esa salida en la terminal

# Ejecución secuencial (I)

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ pwd  
/home/paula.soler
```

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ date  
Thu May 4 11:36:17 CEST 2023
```

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ echo "Este ha sido el resultado"  
Este ha sido el resultado
```

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ pwd ; date ; echo "Este ha sido el resultado"  
/home/paula.soler  
Thu May 4 11:36:17 CEST 2023 Este ha sido el resultado
```

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ date ; pwd ; echo "Este ha sido el resultado"  
Thu May 4 11:36:17 CEST 2023  
/home/paula.soler  
Este ha sido el resultado
```

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ date - ; pwd ; echo "Este ha sido el resultado"  
date: invalid date '-'  
/home/paula.soler  
Este ha sido el resultado
```

# Ejecución secuencial (II)

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ date ; pwd ; ls > estado.txt
```

```
Mon Oct 24 16:58:05 CEST 2022
```

```
/home/paula.soler
```

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cat estado.txt
```

```
cat.txt
```

```
comandos
```

```
comandos.txt
```

```
core.31938
```

```
core.4554
```

```
Desktop
```

```
Documents
```

```
Downloads
```

```
error.txt
```

```
....
```

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ (date ; pwd ; ls) > estado.txt
```

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cat estado.txt
```

```
Mon Oct 24 16:58:21 CEST 2022
```

```
/home/paula.soler
```

```
cat.txt
```

```
comandos
```

```
comandos.txt
```

```
core.31938
```

```
core.4554
```

```
Desktop
```

```
....
```

# Ejecución condicional

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ head /etc/passwd && echo "Puedo abrir este archivo"
```

```
root:x:0:0:root:/root:/bin/bash
```

```
bin:x:1:1:bin:/bin:/sbin/nologin
```

```
daemon:x:2:2:daemon:/sbin:/sbin/nologin
```

```
adm:x:3:4:adm:/var/adm:/sbin/nologin
```

```
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
```

```
sync:x:5:0:sync:/sbin:/bin/sync
```

```
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
```

```
halt:x:7:0:halt:/sbin:/sbin/halt
```

```
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
```

```
operator:x:11:0:operator:/root:/sbin/nologin
```

```
Puedo abrir este archivo
```

```
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ head /etc/shadow && echo "Puedo abrir este archivo"
```

```
head: cannot open '/etc/shadow' for reading: Permission denied
```

```
(base) [UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ head /etc/shadow || echo "No puedo abrir este archivo"
```

```
head: cannot open '/etc/shadow' for reading: Permission denied
```

```
No puedo abrir este archivo
```

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cat /etc/shadow 2>error.txt || echo "NO puedo abrir el archivo"
```

```
NO puedo abrir el archivo
```

## Aspectos a tratar

1

Teoría y Practica:

- Ejecuciones concatenadas (|)
  - Comando tee
- Ejecuciones secuenciales (;)
- Ejecuciones condicionales (&& , ||)

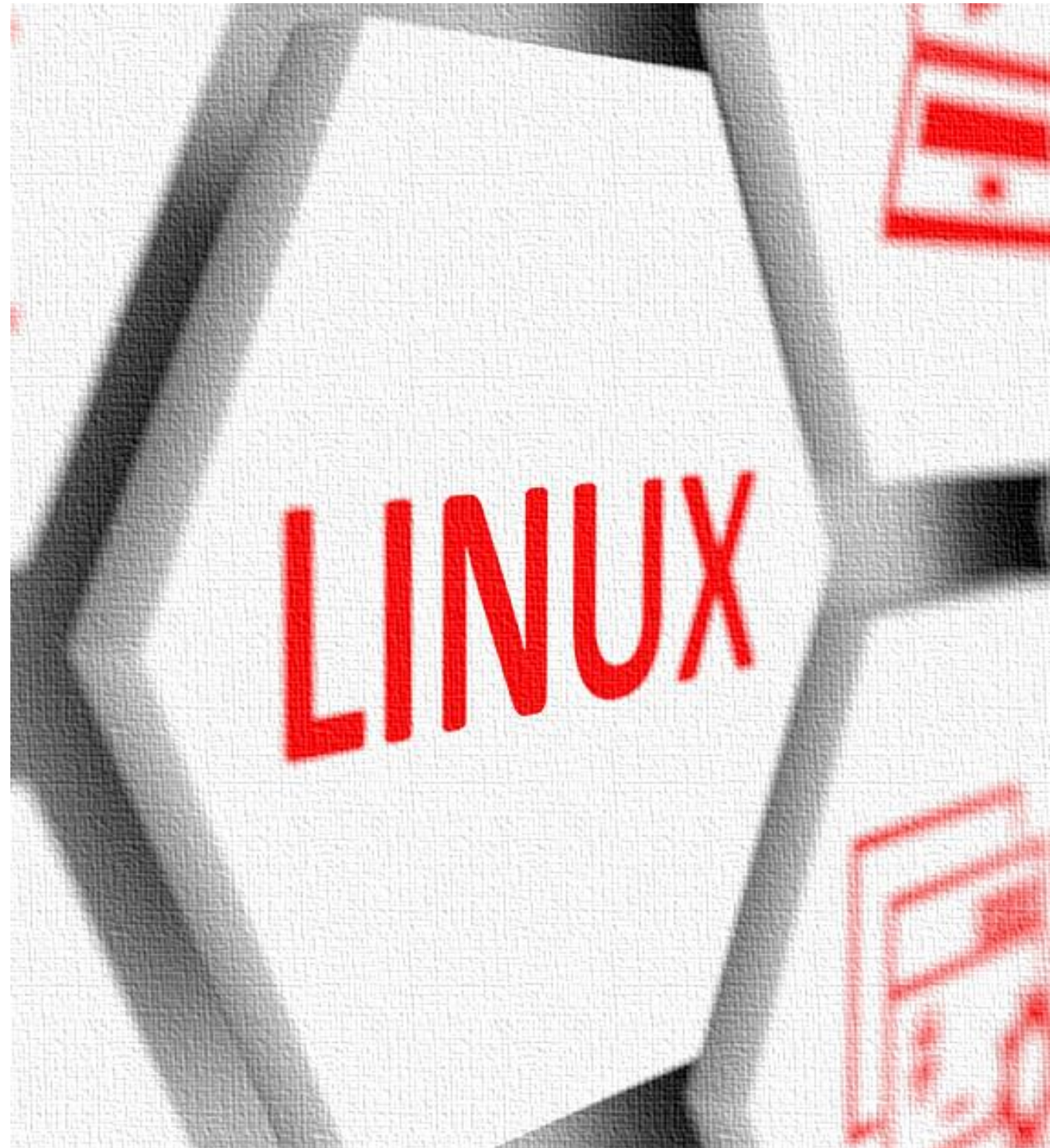
2

**Comandos para el procesamiento de archivos**

- *Cat*
- *Paste*
- *Join*
- *Diff*
- *Cut*
- *Sort*
- *Uniq*
- *Tr...*



# Comandos para el **procesamiento** **básico** de archivo



# Comando *cat*

## Diseño básico del comando

```
cat <opciones> <fichero>
```

Crear un archivo → `cat > file.txt`

Ver el contenido de un archivo → `cat file.txt`

Redirigir contenido → `cat file.txt > destino.txt`

Concatenar archivos → `cat file1.txt file2.txt > destino.txt`

Mostrar números de línea → `cat -n file.txt`      **te enumera los archivos**

# Comando *paste*

Diseño básico del comando

```
paste <opciones> <fichero>
```

Las coordenadas genómicas se delimitan con :-

Sintaxis	Descripción
paste	Fusiona líneas de archivos
	Escribe líneas de texto en la salida estándar. Estas líneas se generan a partir de la fusión de cada una de las líneas de los archivos separados por un tabulador (por defecto)
	-s Fusiona un archivo por vez de forma serial
	-d Separa la unión de los archivos por la lista de separadores que se especifica con la opción



# Comando *join*

Diseño básico del comando

```
join <opciones> <fichero1> <fichero2>
```

**prueba1: DNI y nombre**

11321345,Mario Neta  
20324151,Aquiles Bailo  
12415132,Elsa Blazo  
32412512,Elena Nito  
13241351,César Noso

**prueba2: DNI y sueldo**

11321345,10000  
20324151,8700  
12415132,10500  
32412512,6500  
13241351,9000

**DNI, Nombre, Sueldo**


# Comando *join*

## Diseño básico del comando

```
join <opciones> <fichero1> <fichero2>
```


### prueba1: DNI y nombre

11321345,Mario Neta  
20324151,Aquiles Bailo  
12415132,Elsa Blazo  
32412512,Elena Nito  
13241351,César Noso



### prueba2: DNI y sueldo

11321345,10000  
20324151,8700  
12415132,10500  
32412512,6500  
13241351,9000



### join -t "," prueba1 prueba2

11321345,Mario Neta,10000  
20324151,Aquiles Bailo,8700  
12415132,Elsa Blazo,10500  
32412512,Elena Nito,6500  
13241351,César Noso,9000

# Comando *join*

## ¿Qué ocurre si cambia el orden de las *columnas*?

```
join <opciones> <fichero1> <fichero2>
```

Join solo trabaja con archivo de DOS EN DOS,

**prueba3: nombre y DNI**

Mario Neta, 11321345  
Aquiles Bailo, 20324151  
Elsa Blazo, 12415132  
Elena Nito, 32412512  
César Noso, 13241351

**prueba2: DNI y sueldo**

11321345,10000  
20324151,8700  
12415132,10500  
32412512,6500  
13241351,9000

-1 2 indica que el archivo 1 esté en la columna 2, y -2 1 que el archivo 2 esté en la columna 1

**join -1 2 -2 1 -t “,”prueba3 prueba2**

11321345,Mario Neta,10000  
20324151,Aquiles Bailo,8700  
12415132,Elsa Blazo,10500  
32412512,Elena Nito,6500  
13241351,César Noso,9000

# Comando *join*

## ¿Qué ocurre si cambia el orden de las *filas*?

```
join <opciones> <fichero1> <fichero2>
```

**prueba1: DNI y nombre**

11321345,Mario Neta  
20324151,Aquiles Bailo  
32412512,Elena Nito  
13241351,César Noso  
**12415132,Elsa Blazo**

**prueba2: DNI y sueldo**

11321345,10000  
20324151,8700  
**12415132,10500**  
32412512,6500  
13241351,90000

**\$ join -t “,” prueba1 prueba2**

join: prueba2:3: **is not sorted:** 12415132,10500  
join: prueba1:4: **is not sorted:** 13241351,César Noso

# Comando *diff*

## Diseño básico del comando

```
diff <opciones> <fichero1> <fichero2>
```

	Compara línea por línea dos archivos	
	Algunas opciones:	
diff	-q reporta solo si los archivos difieren	Únicamente reporta si los archivos son distintos, si son iguales no reporta nada, tampoco sale error
	-s reporta si los archivos son idénticos	
	-y salida en dos columnas	

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ diff -q file1.txt file2.txt
Files file1.txt and file2.txt differ
```

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ diff -s file1.txt file2.txt
Files file1.txt and file2.txt are identical
```

# Comando *diff*

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cat distribuciones1.txt
```

CentOS

Red Hat

Ubuntu Debian

Linux Mint

Fedora

OpenSUSE

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cat distribuciones2.txt
```

CentOS

Red Hat

Ubuntu Debian

Linux Mint

OpenSUSE

Fedora



**¿Son iguales?**

# Comando *diff*

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cat distribuciones1.txt
```

CentOS

Red Hat

Ubuntu Debian

Linux Mint

Fedora

OpenSUSE

```
[UNIVERSIDADVIU\paula.soler@a-3edhijmqygwxr ~]$ cat distribuciones2.txt
```

CentOS

Red Hat

Ubuntu Debian

Linux Mint

OpenSUSE

Fedora

**NO** para el  
comando diff  
**(SORT)**

# Comando *diff*

## Diseño básico del comando

```
diff <opciones> <fichero1> <fichero2>
```

La salida del comando *diff* puede tener el siguiente formato:

- Normal (predeterminado)
- Contexto (-c)
- Unificado (-u)
- En columnas (-y)



# Comando *diff* normal

## distribuciones1.txt

CentOS

**Slax**

Red Hat

Debian

Linux Mint

Fedora

**OpenSUSE**

## distribuciones2.txt

CentOS

Red Hat

Debian

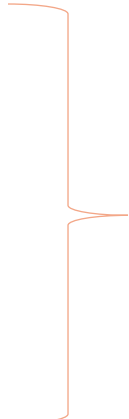
Linux Mint

Fedora

**Xubuntu**

## diff distribuciones1.txt distribuciones2.txt

```
2d1
< Slax
7c6
< OpenSUSE
---
> Xubuntu
```



1. Los números de línea correspondientes al primer fichero
2. Una letra [ “a” (*add*) para añadir, “c” (*change*) para cambiar, y “d” (*delete*) para borrar]
3. Los números de línea correspondientes al segundo fichero
4. Símbolos especiales en los cambios a realizar:
  - el símbolo < son líneas del primer fichero
  - el símbolo > son líneas del segundo fichero

# Comando *diff -y* por columnas

## distribuciones1.txt

CentOS

**Slax**

Red Hat

Debian

Linux Mint

Fedora

**OpenSUSE**

## distribuciones2.txt

CentOS

Red Hat

Debian

Linux Mint

Fedora

**Xubuntu**

**diff -y distribuciones1.txt distribuciones2.txt**

CentOS

Slax

Red Hat

Debian

Linux Mint

Fedora

OpenSUSE

<

CentOS

Red Hat

Debian

Linux Mint

Fedora

|

Xubuntu

# Comando *diff -y* por columnas

## distribuciones1.txt

CentOS

**Slax**

Red Hat

Debian

Linux Mint

Fedora

**OpenSUSE**

## distribuciones2.txt

CentOS

Red Hat

Debian

Linux Mint

Fedora

**Xubuntu**

**diff -y distribuciones1.txt distribuciones2.txt --suppress-common-lines**

Hay que usar esta opción para eliminar los genes que sí son comunes, porque cuando salgan listas muy largas de genes sería difícil ver las diferencias

Slax  
OpenSUSE

<  
| Xubuntu

# Comando *diff -c* en modo contextual

```
diff -c distribuciones1.txt distribuciones2.txt
```

```
*** distribuciones1.txt      2023-05-04 15:03:23.053374252 +0200
```

```
--- distribuciones2.txt      2023-05-04 15:03:41.723395284 +0200
```

```
*****
```

```
*** 1,7 ***
```

```
CentOS
```

```
- Slax
```

```
Red Hat
```

```
Debian
```

```
Linux Mint
```

```
Fedora
```

```
! OpenSUSE
```

```
--- 1,6 ----
```

```
CentOS
```

```
Red Hat
```

```
Debian
```

```
Linux Mint
```

```
Fedora
```

```
! Xubuntu
```

# Comando *cut*

Permite **seleccionar** o **cortar** secciones de **cada línea** de un archivo y escribir el resultado en la salida estándar.

Se utiliza para seleccionar **partes de una línea** por posición de **byte**, **carácter** o **delimitador**.

Se puede usar para extraer datos de archivos:

- **csv** (comma separated-values)
- **tsv** (tab separated-values)

# Comando *cut*

```
$ cat prueba_cut.txt
```

la utilidad cut

es fácil de usar y es útil

```
$ cut -c 4 prueba_cut.txt
```

u

f

ES MEJOR TRABAJAR POR CHARACTER que por BYTE porque no sabemos muy bien que caracteres especiales está contando

```
$ cut -c 1,2 prueba_cut.txt
```

la

es

```
$ cut -c 2-6 prueba_cut.txt
```

a uti

s fác

El espacio en blanco y el *tab* es un caracter

```
$ cut -b 2-6 prueba_cut.txt
```

a uti

s fá

Caracteres especiales pueden estar representados por más de un byte

## Comando **cut -d**

Podemos seleccionar secciones del flujo de datos usando un **delimitador** con la opción **-d**

**Por defecto** el delimitador que asume el comando es el **TAB (\t)**.

Esta opción **se usa** normalmente **junto** con la opción **-f** para especificar **el campo o los campos** a seleccionar

## Comando *cut* -> Ejemplo

```
John,Smith,34,London  
Arthur,Evans,21,Newport  
George,Jones,32,Truro
```

names.csv

El delimitador se puede indicar utilizando la opción **-d** junto al delimitador de interés entre comillas

**-d ','**

```
>cut -d ',' -f 1 names.csv
```

John

Arthur

George



## Comando *cut* -> Ejemplo

```
John,Smith,34,London  
Arthur,Evans,21,Newport  
George,Jones,32,Truro
```

names.csv

El delimitador se puede indicar utilizando la opción **-d** junto al delimitador de interés entre comillas

**-d ','**

```
> cut -d ',' -f 1,4 names.csv  
John,London  
Arthur,Newport  
George,Truro
```

## Comando *cut* -> Ejemplo

```
John,Smith,34,London  
Arthur,Evans,21,Newport  
George,Jones,32,Truro
```

names.csv

El delimitador se puede indicar utilizando la opción **-d** junto al delimitador de interés entre comillas

**-d ','**

```
>cut -d ',' -f 1 names.csv
```

John

Arthur

George

**-f 5 ¿Qué nos reportaría?**

# Comando *cut* -> Ejemplo

```
John,Smith,34,London  
Arthur,Evans,21,Newport  
George,Jones,32,Truro
```

names.csv

El delimitador se puede indicar utilizando la opción **-d** junto al delimitador de interés entre comillas

**-d ','**

```
>cut -d ',' -f 1 names.csv
```

```
John  
Arthur  
George
```

¿Qué nos reportaría?

```
John,Smith,34,London  
Arthur,Evans,21,Newport  
George,Jones,32,Truro
```



viu

**Universidad**  
Internacional  
de Valencia

[universidadviu.com](http://universidadviu.com)

De:  
 Planeta Formación y Universidades