

## BASH 02.03 - Tuberías - explicación y preguntas

[Información sobre el documento.....1](#)

[1. - Introducción.....2](#)

[2. - Tuberías.....2](#)

[¿Qué es una tubería?.....2](#)

[Ejemplos de uso.....4](#)

[3. - Preguntas y respuestas.....5](#)

### Información sobre el documento

El objetivo de este documento es enseñar, compartir conocimientos para facilitar el aprendizaje. Este documento es mejorable, y será actualizado si es preciso.

Este documento puede ser utilizado para uso personal, no comercial, como se presenta, respetando una serie de **condiciones de uso basadas en Licencia Creative Commons, como aparece en el logotipo, que se describe a continuación**



- El **documento se proporciona como está**, por tanto *no se pueden realizar modificaciones*, ni en el **formato** ni en el **contenido**, ni trabajos derivados, sin la autorización expresa del autor.
- Se debe **mencionar al autor** del mismo, por supuesto sin modificar los enlaces o imágenes introducidas por éste en el documento.
- **No se puede utilizar** este material **con fines lucrativos, comerciales o cualquier uso que pueda proporcionar, directa o indirectamente, un beneficio económico de terceros**, sin la autorización expresa del autor.
- Si se desea publicar el documento en algún sitio web, se debe hacer a través de un **enlace al documento en el sitio del autor** ([www.educatica.es](http://www.educatica.es) o [cursos.educatica.es](http://cursos.educatica.es))

Si se desea publicar como un recurso dentro de un sitio web, sin utilizar un enlace al material en el sitio web del autor, se debe solicitar autorización expresa y, en cualquier caso, referenciar el sitio web del autor ([www.educatica.es](http://www.educatica.es)).

Espero sea de provecho ;)



## 1. - Introducción

El objetivo de este documento es explicar la tuberías y utilizarlas a lo largo de un caso práctico guiado. Para sacar el máximo partido al documento, se recomienda leer el apartado donde se presentan las tuberías, su uso y utilidad, para después tratar de realizar el caso práctico resuelto que está compuesto por una serie de preguntas que se responden a continuación.

Se recomienda que el lector trate de responder a cada pregunta individualmente con la ayuda de las pistas que se proporcionan y consultando, si es necesario, la página de manual del comando o ficheros involucrados.

También se puede utilizar como apoyo el material didáctico disponible en el sitio web de [educatica](http://educatica.es) o de terceros.

## 2. - Tuberías

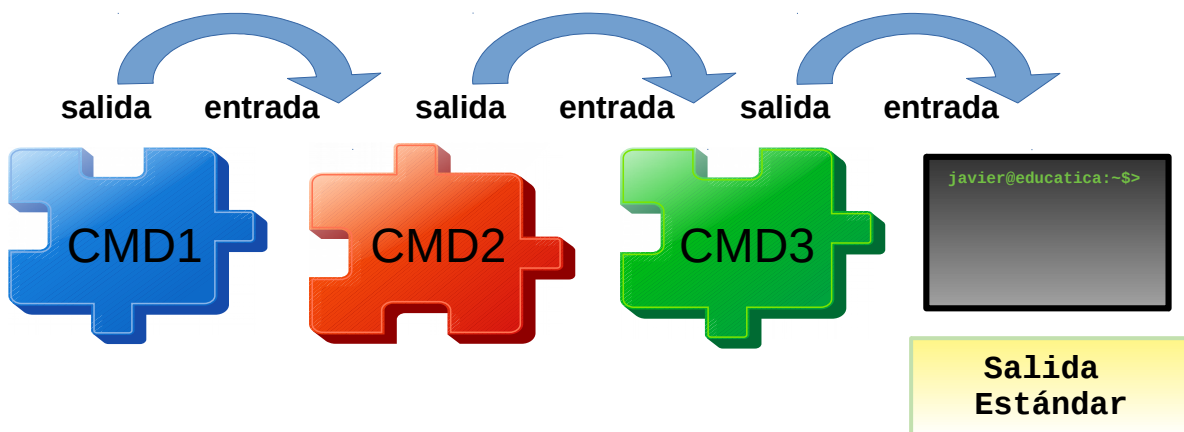
Hemos visto como en GNU/Linux podemos utilizar redirecciones para redirigir la salida de un comando y enviarla a un fichero en lugar de la salida estándar: la pantalla.

También hemos visto, a lo largo de la unidad dedicada a sistemas GNU/Linux, como uno de los principios de los sistemas operativos Unix-like es “matener las cosas simples” (KISS, Keep It Simple Sir). Así pues los comandos y aplicaciones de estos sistemas deben realizar una función muy concreta, simple pero bien hecha.

¿Pero qué sucede si necesitamos realizar una tarea más compleja que no nos proporciona un comando concreto?. Aquí tenemos dos soluciones, utilizar tuberías para conectar varios comandos o crear un shell script que solucione el problema. En este apartado vamos a estudiar la primera solución, las tuberías.

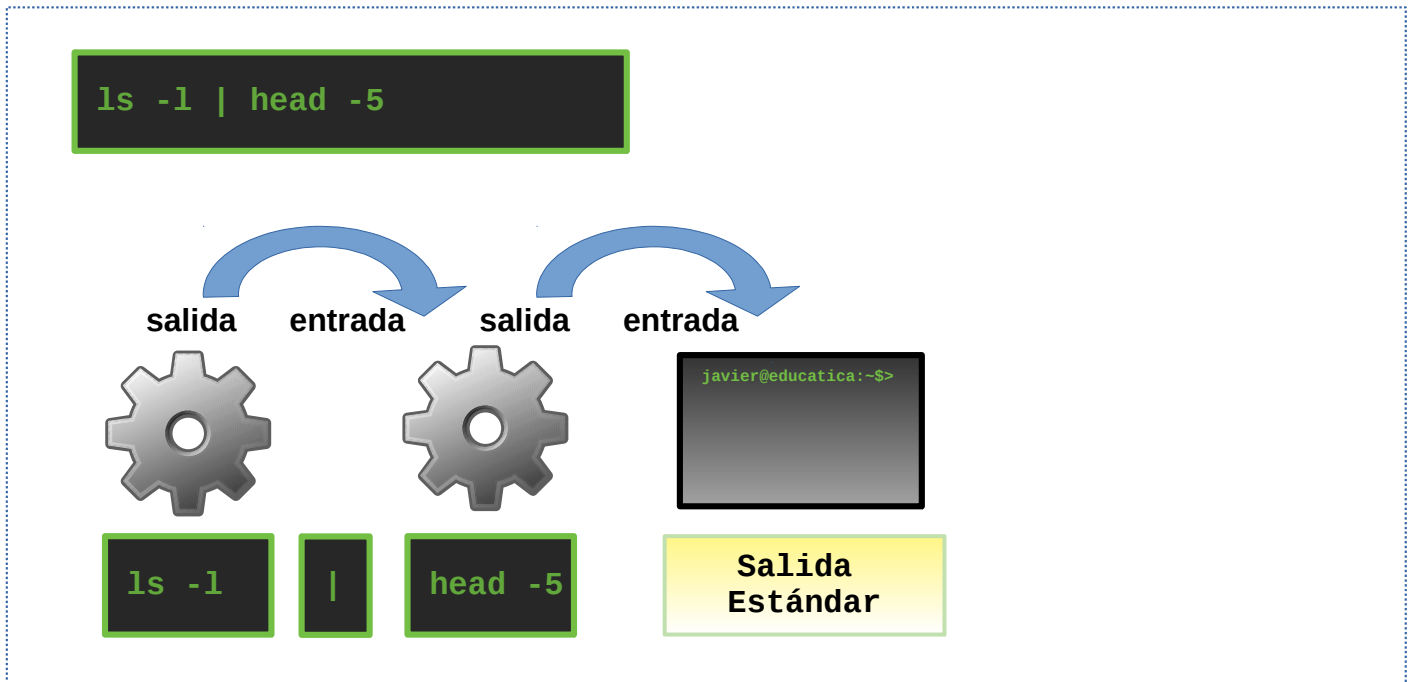
### ¿Qué es una tubería?

Básicamente una tubería es una herramienta que proporciona el interprete de comandos que permite conectar un comando con otro.

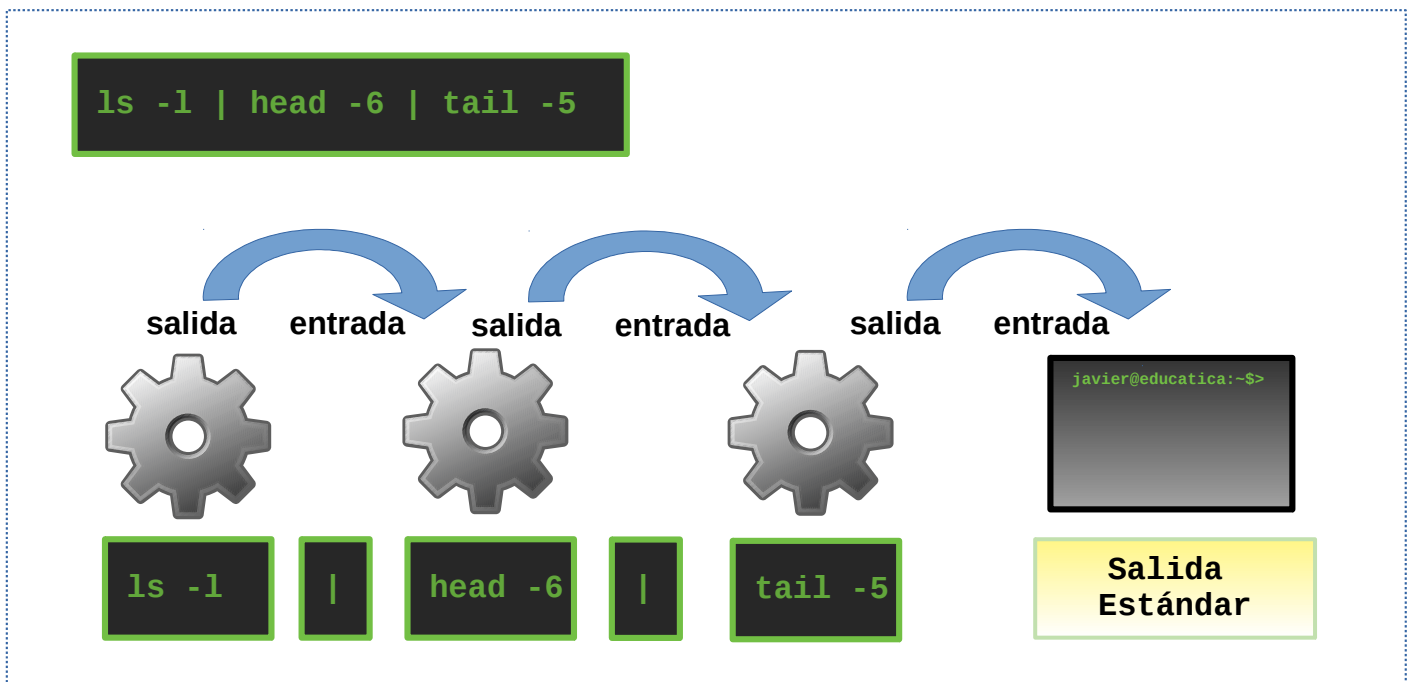


Cuando usamos una tubería para comunicar dos comandos, indicamos al intérprete de comandos o shell que la salida de un comando, en lugar de mostrarse por pantalla, se envíe como entrada al comando siguiente.

La tubería se expresa con el carácter especial “|” entre dos comandos. Podemos conectar varios comandos en secuencia, uno tras otro con tuberías, para hacer que la salida del anterior pase al siguiente, así hasta el final de la cadena.



De esta forma, podemos conseguir obtener una funcionalidad compleja o resolver un problema complejo compuesto por una serie de comandos que colaboran pasándose resultados en secuencia.



La mejor forma de comprender las tuberías es trabajar con ellas, por eso se proporcionan a continuación una serie de ejemplos simples y después una serie de preguntas a resolver utilizando comandos y tuberías que los conecten.

## Ejemplos de uso

Con el comando `ls -l` mostramos un listado de ficheros de nuestro directorio actual. Si el listado es extenso no podremos ver los primeros ficheros del listado, salvo que nos movamos por la salida estándar gracias a que ejecutamos la shell en una ventana.

Si solo queremos mostrar, por ejemplo, los cinco primeros ficheros podemos utilizar tuberías para comunicar la salida del comando `ls -l` con un comando que nos muestre solo las cinco primeras líneas. Este comando es `head` con la opción `-5`.

```
javier@educatica:~$ ls -l | head -5
total 80
drwxr-xr-x 3 alumno alumno 4096 ene 28 14:05 backup
drwxrwxrwx 4 alumno alumno 4096 ene 25 12:00 castelar
-rw-rw-r-- 1 alumno alumno 6741 feb  4 13:39 CMDs-190204.txt
drwxr-xr-x 2 alumno alumno 4096 ene 23 12:18 Cosas
```

El problema de estos comandos es que como solo muestra las cinco primeras líneas, está mostrando una línea con el total de ficheros seguida de solo cuatro líneas de información de ficheros. Nosotros queremos mostrar información de los cinco primeros ficheros.

Una opción es ampliar a seis las líneas a mostrar.

```
javier@educatica:~$ ls -l | head -6
total 80
drwxr-xr-x 3 alumno alumno 4096 ene 28 14:05 backup
drwxrwxrwx 4 alumno alumno 4096 ene 25 12:00 castelar
-rw-rw-r-- 1 alumno alumno 6741 feb  4 13:39 CMDs-190204.txt
drwxr-xr-x 2 alumno alumno 4096 ene 23 12:18 Cosas
drwxr-xr-x 2 alumno alumno 4096 ene 16 12:14 Descargas
```

No obstante, se sigue mostrando una primera línea con información del total que no nos interesa ni se corresponde con los ficheros mostrados.

La solución al problema está en utilizar una nueva tubería que de esta última salida, muestre solo las últimas cinco líneas. Para ello utilizamos el comando `tail` con la opción `-5`.

```
javier@educatica:~$ ls -l | head -6 | tail -5
drwxr-xr-x 3 alumno alumno 4096 ene 28 14:05 backup
drwxrwxrwx 4 alumno alumno 4096 ene 25 12:00 castelar
-rw-rw-r-- 1 alumno alumno 6741 feb  4 13:39 CMDs-190204.txt
drwxr-xr-x 2 alumno alumno 4096 ene 23 12:18 Cosas
drwxr-xr-x 2 alumno alumno 4096 ene 16 12:14 Descargas
```

De esta forma, con el primer comando mostraríamos todos los ficheros de directorio actual. Con la salida de ese comando, se pasa como entrada del siguiente, que mostrará solo las primeras 6 líneas de esa salida. La salida de ese comando, las primeras 6 líneas del listado de ficheros en el directorio actual, la pasamos con otra tubería al siguiente comando, el cuál mostrará las últimas cinco líneas pasadas por la entrada del comando.

La salida de un comando pasa como entrada al siguiente, así podemos ir trabajando con la salida anterior para obtener el resultado deseado o irlo modificando, comando tras comando gracias a las tuberías, hasta que obtenemos el resultado final.

## 3. -Preguntas y respuestas

### 1. Muestra los procesos en ejecución (comando ps)

Podemos mostrar los procesos en ejecución en el sistema con el comando **ps**. Para más información de este comando podemos consultar la página de ayuda.

```
man ps
```

Según la ayuda del comando, éste muestra un instantánea de información de los procesos en ejecución en un momento dado.

Con la opción **-A** muestra todos los procesos. Podemos usarla para mostrar todos los procesos en ejecución en el sistema.

```
javier@educatica:~$ ps -A
  PID TTY          TIME CMD
    1 ?            00:00:14 systemd
    2 ?            00:00:00 kthreadd
    4 ?            00:00:00 kworker/0:0H
    6 ?            00:00:00 mm_percpu_wq
    7 ?            00:00:02 ksoftirqd/0
...
21986 ?            00:00:00 kworker/u2:1
22082 ?            00:00:00 cupsd
22084 ?            00:00:00 cups-browsed
22163 ?            00:00:00 kworker/u2:0
22206 pts/0          00:00:00 bash
22222 pts/0          00:00:00 man
22232 pts/0          00:00:00 pager
22246 pts/1          00:00:00 ps
```

Si ejecutamos este comando, podemos comprobar que es demasiada información la que se muestra en pantalla como para poder analizarla de un vistazo.

### 2. Calcula el número de líneas que muestra el comando ps -A al ejecutarlo (comando wc opción -l)

Utilizando el comando **wc** (contador de palabras, bytes o líneas) con la opción **-l** (contar solo las líneas) podemos contar la cantidad de procesos que hay en ejecución en el sistema.

```
javier@educatica:~$ ps -A | wc -l
238
```

### 3. Muestra información de los procesos que se estén ejecutando en alguna terminal del sistema

Ahora vamos a utilizar el comando **grep** junto con el comando **ps -A** para mostrar información de procesos que están ejecutándose en una terminal.

El comando **grep** es un comando muy útil que merece un estudio aparte. No obstante vamos a empezar a utilizarlo ahora mismo. Su función es filtrar las líneas de texto de un fichero o que le llegan por entrada: solo mostrará las líneas de texto que tengan el patrón de texto que se pasa por parámetro a **grep**. Podemos consultar la ayuda de **grep** para echar un vistazo a su sintaxis y algunas de las opciones disponibles

```
man grep
```



Por ejemplo, si queremos mostrar las líneas del fichero saludo.txt que contengan la cadena de texto javier usaremos el comando grep de la siguiente forma:

```
grep javier saludo.txt
```

Sin embargo, en esta pregunta, queremos utilizar grep no para filtrar el contenido de un fichero, sino para filtrar la salida de un comando, así que utilizamos una tubería que conecte ambos comandos.

Primero usamos ps -A para mostrar todos los procesos en ejecución y pasamos esa lista a grep para que solo muestre las líneas que contengan la cadena de texto tty.

```
javier@educatica:~$ ps -A | grep tty
 718 tty1      00:00:00 gdm-wayland-ses
 722 tty1      00:00:00 gnome-session-b
 756 tty1      00:00:55 gnome-shell
 790 tty1      00:00:00 Xwayland
 946 tty1      00:00:00 ibus-daemon
 983 tty1      00:00:00 ibus-dconf
 986 tty1      00:00:00 ibus-x11
1007 tty1      00:00:00 gsd-xsettings
1011 tty1      00:00:00 gsd-a11y-settin
1013 tty1      00:00:00 gsd-clipboard
1021 tty1      00:00:05 gsd-color
1025 tty1      00:00:00 gsd-datetime
1028 tty1      00:00:00 gsd-housekeepin
1030 tty1      00:00:00 gsd-keyboard
1038 tty1      00:00:00 gsd-media-keys
1039 tty1      00:00:00 gsd-mouse
1040 tty1      00:00:00 gsd-power
1043 tty1      00:00:00 gsd-print-notif
1045 tty1      00:00:00 gsd-rfkill
1051 tty1      00:00:00 gsd-screensaver
1058 tty1      00:00:00 gsd-sharing
1062 tty1      00:00:00 gsd-smartcard
1066 tty1      00:00:00 gsd-sound
1074 tty1      00:00:00 gsd-wacom
1122 tty1      00:00:00 ibus-engine-sim
2331 tty2      00:00:00 gdm-x-session
2333 tty2      00:06:12 Xorg
2343 tty2      00:00:01 gnome-session-b
2491 tty2      00:45:56 gnome-shell
2529 tty2      00:00:29 ibus-daemon
2549 tty2      00:00:00 ibus-dconf
2551 tty2      00:00:00 ibus-x11
2620 tty2      00:00:01 gsd-power
```



```
2621 tty2    00:00:00 gsd-print-notif
2623 tty2    00:00:00 gsd-rfkill
2624 tty2    00:00:00 gsd-screensaver
2625 tty2    00:00:00 gsd-sharing
2627 tty2    00:00:00 gsd-smartcard
2630 tty2    00:00:00 gsd-sound
2631 tty2    00:00:01 gsd-wacom
2633 tty2    00:00:01 gsd-xsettings
2645 tty2    00:00:00 gsd-a11y-settin
2648 tty2    00:00:00 gsd-clipboard
2649 tty2    00:00:00 gsd-datetime
2651 tty2    00:00:06 gsd-color
2653 tty2    00:00:01 gsd-keyboard
2656 tty2    00:00:02 gsd-housekeepin
2658 tty2    00:00:00 gsd-mouse
2659 tty2    00:00:01 gsd-media-keys
2670 tty2    00:00:00 gsd-printer
2719 tty2    00:01:06 gnome-software
2724 tty2    00:00:00 gsd-disk-utilit
2726 tty2    00:00:24 nautilus-deskto
2776 tty2    00:00:07 ibus-engine-sim
2926 tty2    00:00:02 update-notifier
3014 tty2    00:00:00 deja-dup-monito
19838 tty4    00:00:00 agetty
19839 tty5    00:00:00 login
19964 tty6    00:00:00 agetty
20488 tty5    00:00:00 bash
21395 tty2    00:00:03 gedit
21807 tty2    00:00:26 update-manager
javier@educatica:~$
```

4. Muestra los procesos que se estén ejecutando en una terminal virtual y cuyo comando contenga la cadena gnome.

Como ya sabemos, se pueden encadenar varios comandos con tuberías para que realizar una función compuesta por la función de la secuencia de dichos comandos. Hay que tener claro el orden en que deben ejecutarse esos comandos a la hora de armar el comando final a ejecutar.

En este caso, se quiere mostrar solo aquellos comandos que se estén ejecutando en una terminal y además que el comando tenga como parte de su nombre la cadena gnome.

Lo primero que tenemos que obtener es el listado de los procesos en ejecución. Después filtramos dicha lista. En este caso, nos da igual filtrar solo los que se ejecutan en una terminal y luego los que ejecuten un comando que contenga la cadena gnome. Así que, primero filtramos las líneas que contengan la cadena tty y después filtramos los procesos que contengan la cadena gnome.

```
javier@educatica:~$ ps -A | grep tty | grep gnome
722 tty1    00:00:00 gnome-session-b
```

```

756 tty1      00:00:55 gnome-shell
2343 tty2      00:00:01 gnome-session-b
2491 tty2      00:46:01 gnome-shell
2719 tty2      00:01:06 gnome-software
javier@educatica:~$ ps -A | grep tty | grep gnome | tail -1
2719 tty2      00:01:06 gnome-software
javier@educatica:~$ ps -A | grep tty | grep gnome | head -1
722 tty1       00:00:00 gnome-session-b
javier@educatica:~$ ps -A | grep tty | grep gnome | head -1 > primero.txt
javier@educatica:~$ cat primero.txt
722 tty1       00:00:00 gnome-session-b
javier@educatica:~$ cat /etc/passwd | grep alfredoff
alfredoff:x:1001:1001:Alfredo Flores Fuentes,,,:/home/alfredoff:/bin/bash
javier@educatica:~$

```

5. Muestra las líneas de configuración de cuentas de usuario que utilicen el shell `/bin/false`.

En este caso podemos utilizar solo el comando `grep` o bien una combinación de `cat` y `grep`. Vamos a utilizar la última opción por practicar con tuberías.

Con el comando `cat` mostramos el contenido de un fichero en pantalla, sin esperar ninguna interacción con el usuario.

Con tuberías no podríamos utilizar ni `less` ni `more`, puesto que estos comandos esperan interacción con el usuario para mostrar la información en pantalla. La tubería espera la salida de un comando para pasarla como entrada al siguiente.

Con `cat /etc/passwd` podemos mostrar todo el contenido del fichero de cuentas de usuario. Ahora basta con filtrar las líneas de este fichero para solo mostrar aquellas que contengan la cadena `/bin/false`

```

javier@educatica:~$ cat /etc/passwd | grep /bin/false
speech-dispatcher:x:111:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
whoopsie:x:112:117::/nonexistent:/bin/false
hplip:x:118:7:HPLIP system user,,,:/var/run/hplip:/bin/false
gnome-initial-setup:x:120:65534::/run/gnome-initial-setup:/bin/false
gdm:x:121:125:Gnome Display Manager:/var/lib/gdm3:/bin/false

```

Como podemos intuir, las cuentas de usuario que tienen como shell `/bin/false` son cuentas que no van a iniciar sesión en el sistema. Se trata de cuentas especiales de servicios o daemons en terminología Unix-like. Aunque hay más servicios con cuentas en el sistema.

6. Muestra las líneas de configuración de cuentas de usuario que utilicen el shell `/bin/bash`.

Es bastante similar a la pregunta anterior, solo que ahora solo mostraremos las cuentas que utilizan este shell. En este caso, como veremos, se trata de las cuentas de usuario del sistema, tanto `root` como las creadas en otros casos prácticos anteriores.

```

javier@educatica:~$ cat /etc/passwd | grep /bin/bash
root:x:0:0:root:/root:/bin/bash
alumno:x:1000:1000:alumno,,,:/home/alumno:/bin/bash
alfredoff:x:1001:1001:Alfredo Flores Fuentes,,,:/home/alfredoff:/bin/bash

```



```

marinapg:x:1002:1002:Marina Perez Gomez,,,:/home/marinapg:/bin/bash
ramonam:x:1003:1003:Ramon Andrade Marín,,,:/home/ramonam:/bin/bash
jesusrp:x:1004:1007:Jesus Rodriguez,14,04545,555-123,No hay más
informacion:/home/jesusrp:/bin/bash

```

7. Muestra las líneas de configuración de cuentas de usuario del sistema que no utilicen el shell `/bin/false`.

En este caso tenemos que usar `grep` para filtrar. Sin embargo, nos piden filtrar o mostrar las líneas que no cumplan con un determinado patrón. Debemos echar un vistazo a la ayuda del comando `grep` y buscar alguna opción que invierta la búsqueda. En este caso es la opción `-v`.

Con la opción `-v` lo que hacemos es pedir que se aplique el filtro pero a la inversa, es decir se seleccionarán las líneas que no cumplan con el patrón textual que se pase.

```

javier@educatica:~$ cat /etc/passwd | grep -v /bin/false
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106:./home/syslog:/usr/sbin/nologin
messagebus:x:103:107:./nonexistent:/usr/sbin/nologin
_apt:x:104:65534:./nonexistent:/usr/sbin/nologin
uidd:x:105:111:./run/uidd:/usr/sbin/nologin
avahi-autoipd:x:106:112:Avahi autoip
daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:107:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
dnsmasq:x:108:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
rtkit:x:109:114:RealtimeKit,,,:/proc:/usr/sbin/nologin
cups-pk-helper:x:110:116:user for cups-pk-helper
service,,,:/home/cups-pk-helper:/usr/sbin/nologin

```



```
kernoops:x:113:65534:Kernel Oops Tracking Daemon,,,:/usr/sbin/nologin
saned:x:114:119:./var/lib/saned:/usr/sbin/nologin
pulse:x:115:120:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
avahi:x:116:122:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
colord:x:117:123:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:119:124:./var/lib/geoclue:/usr/sbin/nologin
alumno:x:1000:1000:alumno,,,:/home/alumno:/bin/bash
alfredoff:x:1001:1001:Alfredo Flores Fuentes,,,:/home/alfredoff:/bin/bash
marinapg:x:1002:1002:Marina Perez Gomez,,,:/home/marinapg:/bin/bash
ramonam:x:1003:1003:Ramon Andrade Marín,,,:/home/ramonam:/bin/bash
jesusrp:x:1004:1007:Jesus Rodriguez,14,04545,555-123,No hay más
informacion:/home/jesusrp:/bin/bash
```

Como se puede observar, hay más cuentas que no usan /bin/false ni /bin/bash, por ejemplo tenemos muchas cuentas que usan /usr/sbin/nologin. Si quieres más información puedes consultar ayuda sobre ese fichero.

#### 8. Muestra la línea de configuración de la cuenta de alfredoff

De nuevo usaremos grep en nuestra tubería, pero en este caso con el nombre de la cuenta de usuario para la que queremos obtener la información.

```
javier@educatica:~$ cat /etc/passwd | grep alfredoff
alfredoff:x:1001:1001:Alfredo Flores Fuentes,,,:/home/alfredoff:/bin/bash
```

#### 9. Muestra solo el nombre de la cuenta de usuario que contenga la cadena de texto alfredoff. Necesitarás utilizar el comando cut.

Es similar al comando anterior, solo que necesitamos modificar la salida que obteníamos. En este caso nos piden que solo mostremos el nombre del usuario. Es decir, no hay que mostrar más información que el nombre del usuario.

Necesitamos cortar la salida obtenida de los comandos anteriores. Para ello utilizaremos el comando cut. Para obtener información del comando hay que consultar la ayuda de este comando.

```
man cut
```

Este comando es muy útil, puesto que nos permite seleccionar partes concretas de líneas de ficheros o de líneas que se obtienen de la entrada del comando.

En este caso, queremos cortar o seleccionar un campo determinado, en concreto el primer campo de la línea que se obtiene. Para seleccionar el primer campo utilizamos la opción -f1 (-f indica field y 1 es el primer campo). También debemos indicar que carácter se utiliza para delimitar los campos de información de la línea. En este caso, el delimitador de los campos del fichero /etc/passwd es el carácter ":".

```
javier@educatica:~$ cat /etc/passwd | grep alfredoff | cut -d":" -f1
alfredoff
```

#### 10. Muestra solo el UID de la cuenta de usuario que contenga la cadena de texto alfredoff. Necesitarás utilizar el comando cut.

El comando es similar al anterior, con la diferencia que el campo que se desea mostrar no es el primero, sino el tercero.

Si deseamos consultar la ayuda del fichero /etc/passwd para consultar los campos que tiene cada línea y el número concreto podemos consultar la sección 5 de la página de manual de passwd.

```
man 5 passwd
```

En este caso, el UID es el tercer campo de la línea de configuración de la cuenta de usuario, así que tendremos que seleccionar ese campo con el comando cut.

Por tanto, primero mostramos todo el contenido del fichero de cuentas de usuario y filtramos solo las líneas que contengan la cadena alfredoff en cualquier parte. La salida de ese filtro se la pasamos al comando cut (otro filtro) que seleccionará solo el tercer campo de cada línea, utilizando como delimitador el carácter ":"

```
javier@educatica:~$ cat /etc/passwd | grep alfredoff | cut -d":" -f3
1001
```

11. Muestra solo la ruta del directorio personal del usuario que contenga la cadena de texto alfredoff. Necesitarás utilizar el comando cut.

Similar a la anterior, pero solo mostraremos el campo en cuestión. En este caso, el campo número 6

```
javier@educatica:~$ cat /etc/passwd | grep alfredoff | cut -d":" -f6
/home/alfredoff
```

12. Muestra solo el nombre de todas las cuentas de usuario del sistema.

Ahora nos piden que mostremos solo el nombre de todas las cuentas de usuario del sistema. Para mostrar toda la información de cuentas de usuario utilizamos cat /etc/passwd.

De esa salida, filtramos solo mostrando el primer campo, que es el que corresponde con el nombre de la cuenta de usuario.

```
javier@educatica:~$ cat /etc/passwd | cut -d":" -f1
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
list
irc
gnats
nobody
systemd-network
systemd-resolve
syslog
messagebus
```

```
_apt
uuuid
avahi-autoipd
usbmux
dnsmasq
rtkit
cups-pk-helper
speech-dispatcher
whoopsie
kernoops
saned
pulse
avahi
colord
hplip
geoclue
gnome-initial-setup
gdm
alumno
alfredoff
marinapg
ramonam
jesusrp
javier@educatica:~$
javier@educatica:~$ cat /etc/passwd | grep /bin/bash | cut -d":" -f1
root
alumno
alfredoff
marinapg
ramonam
jesusrp
```

13. Almacena en un fichero llamado usuarios.txt el listado de los nombres de usuario del sistema que utilicen el shell /bin/bash.

En este caso tendremos que utilizar redirecciones de salida para almacenar la salida en un fichero y no mostrarla en la salida estándar.

Ahora vamos a pensar en como mostraríamos la información que nos piden almacenar en el fichero. Para ello, primero mostraríamos todo el fichero y seleccionaríamos aquellas líneas que contengan la cadena /bin/bash, que es el shell que nos piden. De esta forma, terminaremos teniendo solo las líneas de configuración de usuarios con ese shell.

Ahora, debemos quedarnos solo con el nombre de la cuenta de usuario. Como ya sabemos este es el campo número 1. Utilizamos cut como en los casos anteriores, indicando el delimitador y el campo que queremos utilizar.

No se nos puede olvidar utilizar la redirección para guardar los datos en el fichero saludo.txt



```
javier@educatica:~$ cat /etc/passwd | grep /bin/bash | cut -d":" -f1 > usuarios.txt
```

Si queremos mostrar el contenido del fichero, podemos utilizar cat seguido de la ruta del fichero a mostrar.

```
javier@educatica:~$ cat usuarios.txt
root
alumno
alfredoff
marinapg
ramonam
jesusrp
```

En este caso concreto el orden de los comandos no se puede cambiar. Si usamos primero cut, nos quedaremos con los nombres de cuentas de usuario del sistema y nada más. Ya no podemos utilizar con éxito grep para filtrar las líneas de texto que contengan el patrón /bin/false, puesto que hemos perdido esa información

14. Muestra solo los permisos para el propietario del fichero usuarios.txt creado en la pregunta anterior. Deberás usar el comando cut con tuberías para mostrar ciertos caracteres.

Sabemos que para mostrar información de permisos de un fichero tenemos que usar ls con la opción -l y pasar como parámetro la ruta del fichero a consultar.

```
javier@educatica:~$ ls -l usuarios.txt
-rw-r--r-- 1 alumno alumno 47 feb  6 12:09 usuarios.txt
```

En la pregunta nos piden que mostremos solo y exclusivamente los permisos del propietario. Si nos fijamos en la línea de salida, los permisos serían los caracteres del 2 al 4 (2,3 y 4) de la línea de salida.

Podemos utilizar cut con la opción -c para seleccionar caracteres de cada línea de un fichero o de la entrada estándar. Puedes consultar la ayuda del comando cut y echar un vistazo a la opción -c y a la opción LIST.

En este caso, mostramos toda la información de permisos y la salida se la pasamos a cut con la opción -c seleccionando los caracteres del 2 al 4.

```
javier@educatica:~$ ls -l usuarios.txt | cut -c2-4
rw-
```

15. Muestra solo los permisos para el grupo del fichero usuarios.txt creado en la pregunta anterior. Deberás usar el comando cut con tuberías para mostrar ciertos caracteres.

La solución es similar a la de la pregunta anterior, solo que ahora vamos a seleccionar los caracteres del 5 al 7.

```
javier@educatica:~$ ls -l usuarios.txt | cut -c5-7
r--
```

16. Muestra solo los permisos para el resto de usuario del fichero usuarios.txt creado en la pregunta anterior. Deberás usar el comando cut con tuberías para mostrar ciertos caracteres.

```
javier@educatica:~$ ls -l usuarios.txt | cut -c8-10
r--
```

17. Muestra solo el nombre del propietario del fichero usuarios.txt creado en la pregunta anterior. Deberás usar el comando cut con tuberías para mostrar ciertos caracteres.

En este caso ya no podemos utilizar `c` para seleccionar permisos, puesto que el nombre del propietario no va de un carácter concreto a otro de un fichero a otro. Podríamos hacerlo para un fichero concreto con un propietario determinado, pero esto no tendría mucho sentido.

Tenemos que volver a pensar en campos. Si leemos la ayuda del comando `cut`, con la opción `-f`, utiliza como delimitador por defecto el tabulador. Sin embargo, el carácter separador si analizamos la salida del comando `ls -l` es el espacio, así que tendremos que determinar que campo es el del propietario teniendo en cuenta que el separador a utilizar es un espacio.

El campo es el número 3

```
javier@educatica:~$ ls -l usuarios.txt
-rw-r--r-- 1 alumno alumno 47 feb  6 12:09 usuarios.txt
```

Así que, primero mostramos toda la información extendida del fichero `saludo.txt` y la salida se la pasamos al comando `cut` para que seleccione solo el campo número tres, teniendo como delimitador el espacio.

```
javier@educatica:~$ ls -l usuarios.txt | cut -d" " -f3
alumno
```

18. Muestra solo el nombre del grupo del fichero `usuarios.txt` creado en la pregunta anterior. Deberás usar el comando `cut` con tuberías para mostrar ciertos caracteres.

La solución es similar a la anterior, pero seleccionando el campo número 4.

```
javier@educatica:~$ ls -l usuarios.txt | cut -d" " -f4
alumno
```

Para que se vea con claridad, vamos a cambiar el grupo del fichero con el comando `chgrp` y volvemos a ejecutar el comando.

```
javier@educatica:~$ sudo chgrp contable usuarios.txt
javier@educatica:~$ ls -l usuarios.txt | cut -d" " -f4
contable
```

19. Muestra solo los permisos, el nombre del propietario y del grupo del fichero `usuarios.txt` creado en la pregunta anterior. Deberás usar el comando `cut` con tuberías para mostrar ciertos caracteres.

Ahora nos piden tres campos. Podemos consultar la ayuda de `cut` y mirar la sección donde se explican las listas o `LIST`. Podemos seleccionar segmentos de caracteres, bytes o campos de cada línea, utilizando comas y segmentos con el guión.

De esta forma, de la salida del comando `ls -l saludo.txt` queremos seleccionar solo el primer campo, el tercero y el cuarto.

```
javier@educatica:~$ ls -l usuarios.txt
-rw-r--r-- 1 alumno contable 47 feb  6 12:09 usuarios.txt
javier@educatica:~$ ls -l usuarios.txt | cut -d" " -f1,3-4
-rw-r--r-- alumno contable
```

20. Muestra solo los permisos, el nombre del propietario, del grupo y el nombre del propio fichero `usuarios.txt`. Deberás usar el comando `cut` y el comando `tr` con la opción `-s`.

Lo primero que se nos ocurre es seleccionar los campos anteriores, más el campo noveno. Si contamos los campos, el campo noveno será el nombre del fichero.

```
javier@educatica:~$ ls -l usuarios.txt | cut -d" " -f1,3-4,9
-rw-r--r-- alumno contable 12:09
```



Sin embargo, como podemos ver, no ha funcionado como esperabamos. Esto es así, puesto que el separador de campos es un espacio. Si nos encontramos con más de un espacio seguido, cut los interpreta como campos vacíos, pero a fin de cuentas, campos.

```
javier@educatica:~$ ls -l usuarios.txt
-rw-r--r-- 1 alumno contable 47 feb 6 12:09 usuarios.txt
```

Si nos fijamos en la salida, después del mes (feb) hay más de un espacio para separar el nombre del mes del día.

Lo que tenemos que hacer es, antes de cortar los campos, eliminar la repetición de espacios vacíos. De esta forma, si aparecen dos o más espacios seguidos, deberíamos convertirlos o traducirlos en un solo espacio.

Para ello, usaremos el comando tr con la opción -s. Podemos obtener más información de este comando consultando su página de manual.

man tr

Este comando traduce, modificando la entrada que recibe o las líneas de un fichero. En este caso, con la opción -s lo que va a hacer es quitar repeticiones del patrón que se pase, dejando un solo elemento.

Por ejemplo, si ejecutamos hola que tal, y separamos cada palabra con más de un espacio seguido, al pasarle el filtro o comando tr con la opción -s y parámetro " ", el espacio, este comando sustituirá todos los espacios seguidos por uno solo.

```
javier@educatica:~$ echo hola que tal estas | tr -s " "
hola que tal estas
```

Aplicando esto a lo que necesitamos, primero mostramos toda la información de permisos con ls -l. Ahora, utilizamos tr con la opción -s para sustituir varios espacios seguidos por uno solo.

```
javier@educatica:~$ ls -l usuarios.txt | tr -s " "
-rw-r--r-- 1 alumno contable 47 feb 6 12:09 usuarios.txt
```

Ahora si, podemos utilizar cut con el delimitador espacio " " para separar campos.

```
javier@educatica:~$ ls -l usuarios.txt | tr -s " " | cut -d" " -f1,3-4,9
-rw-r--r-- alumno contable usuarios.txt
```

21.Muestra el nombre de cuenta de usuario y ruta del directorio personal de todas las cuentas de usuarios del sistema que utilicen el shell /bin/bash.

Similar a las anteriores preguntas, al menos en cuanto al uso de cut. Primero mostramos todo el fichero y con la ayuda de grep seleccionamos las líneas que contengan la cadena /bin/bash, que es el shell que buscamos.

Ahora, debemos seleccionar de esas líneas el campo 1 y el 6 utilizando el comando cut, especificando que el separador de campos es el carácter ":"

```
javier@educatica:~$ cat /etc/passwd | grep /bin/bash | cut -d":" -f1,6
root:/root
alumno:/home/alumno
alfredoff:/home/alfredoff
marinapg:/home/marinapg
ramonam:/home/ramonam
jesusrp:/home/jesusrp
```