

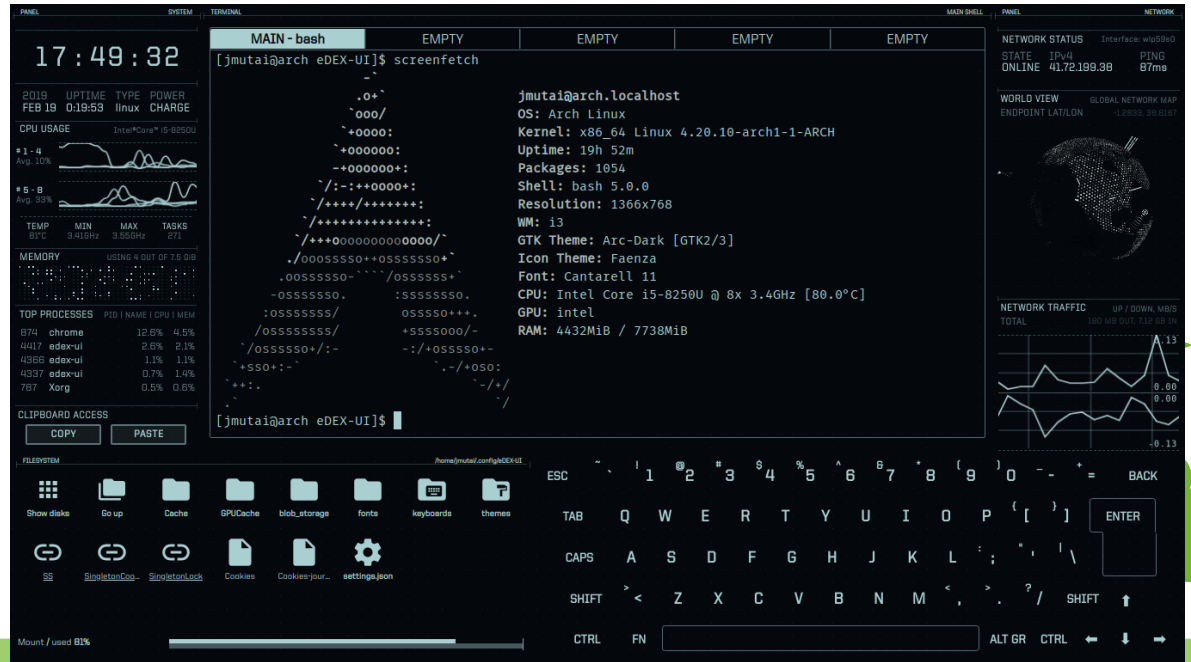
Administración de sistemas Linux

La *shell* – administración de Linux usando comandos



¿Qué es una *shell*?

- Una *shell* es un intérprete de comandos consistente en una interfaz tradicional de los sistemas Unix



¿Qué es una *shell*?

- Una *shell* nos permite interactuar con el sistema operativo a cualquier nivel.
- En Linux, se pueden realizar desde tareas especialmente complejas hasta tareas muy sencillas mediante la consola de comandos.
- Existen muchos tipos de *shell*: `sh`, `ash`, `bash`, `dash`, `ksh`, `zsh`.
- Cada una ofrece funcionalidades/mejoras sobre versiones anteriores.
- `bash` la más usada, `zsh` la más completa.
- En un mismo sistema podemos tener cuantas *shell* queramos.



1. Comandos para manejo de ficheros

```
javinator9889@Tony-MkIII ► ~ ► whatis ls  
ls (1)                - list directory contents
```

Algunas opciones muy útiles:

-a, --all	no oculta las entradas que comienzan con .
-h, --human-readable	with -l and -s, print sizes like 1K 234M 2G etc.
--si	likewise, but use powers of 1000 not 1024
-d, --directory	list directories themselves, not their contents
-i, --inode	print the index number of each file
-l	utiliza un formato de listado largo
-r, --reverse	invierte el orden, en su caso
-R, --recursive	muestra los subdirectorios recursivamente
-s, --size	muestra el tamaño de cada fichero, en bloques
-S	sort by file size, largest first



1. Comandos para manejo de ficheros

```
cd - change directory
```

Algunas opciones muy útiles:

```
cd /directory/to/access  
cd # va a la carpeta HOME  
cd ~ # va a la carpeta HOME
```



1. Comandos para manejo de ficheros

```
javinator9889@Tony-MkIII ▶ ~ ▶ whatis pwd  
pwd (1)                - print name of current/working directory
```

Algunos ejemplos:

```
pwd  # /home/user/test  
cd ../../alfa  
pwd  # /home/alfa
```



1. Comandos para manejo de ficheros

```
javinator9889@Tony-MkIII ► ~ ► whatis mkdir  
mkdir (1)          - make directories  
mkdir (2)          - create a directory
```

Algunas opciones útiles:

```
-m, --mode=MODE    establece los permisos (como en chmod), en lugar  
                   de a=rwx - umask  
-p, --parents      no hay error si existen, crea los directorios padres en  
                   caso necesario
```



1. Comandos para manejo de ficheros

Trucos:

```
cd /home/$USER/Desktop  
mkdir ISO  
mkdir ISO/etc  
mkdir ISO/lib  
mkdir ISO/usr  
mkdir ISO/bin  
mkdir ISO/tmp  
mkdir ISO/opt
```

```
mkdir -p /home/$USER/Desktop/ISO/{etc/x
```

```
mkdir ISO/etc/x1  
mkdir ISO/usr/x2  
mkdir ISO/usr/x3  
mkdir ISO/tmp/Y1  
mkdir ISO/tmp/Y2  
mkdir ISO/tmp/Y3  
mkdir ISO/tmp/Y3/z
```

```
javinator9889@Tony-MkIII ► ~ ► tree Desktop/ISO  
Desktop/ISO
```

```
├── bin  
├── etc  
│   └── x1  
├── lib  
├── opt  
├── tmp  
│   ├── Y1  
│   ├── Y2  
│   └── Y3  
│       └── z  
├── usr  
│   ├── x2  
│   └── x3  
└── var
```

```
14 directories, 0 files
```


1. Comandos para manejo de ficheros

```
javinator9889@Tony-MkIII ► ~ ► whatis mv  
mv (1)                - move (rename) files
```

Algunas opciones útiles:

<code>--backup[=CONTROL]</code>	crea una copia de seguridad antes de borrar
<code>-f, --force</code>	no pregunta nunca antes de sobrescribir
<code>-i, --interactive</code>	pide confirmación antes de sobrescribir
<code>-u, --update</code>	mueve solamente cuando el fichero ORIGEN es más moderno que el fichero de destino, o cuando falta el fichero de destino



1. Comandos para manejo de ficheros

Ejemplos:

```
# Renombrar un fichero / directorio
```

```
mv file.txt file.old.txt
```

```
# Mover de sitio ficheros del mismo tipo a otro directorio de forma interactiva
```

```
mv -i *.txt /tmp/txtfiles
```

```
# Si el fichero es más moderno, actualiza la versión antigua
```

```
mv -u main.cpp src/
```

```
# Si ya hay un fichero con el mismo nombre, crea una copia de seguridad del mismo
```

```
mv --backup=existing file.dat /usr/files/
```

```
# Opciones de copia
```

```
javinator9889@Tony-MkIII ▶ ~ ▶ mv --backup=
```

```
existing  nil      -- numbered backups, if they already exist
```

```
none      off      -- never make backups
```

```
numbered  t        -- make numbered backups
```

```
simple     never    -- always make simple backups
```



1. Comandos para manejo de ficheros

```
javinator9889@Tony-MkIII ► ~ ► whatis ln  
ln (1)                - make links between files
```

Algunas opciones útiles:

<code>-d, -F, --directory</code>	permite al superusuario intentar crear un enlace duro entre directorios (note: puede que falle por restricciones del sistema, incluso para el superusuario)
<code>-f, --force</code>	borra los ficheros destino que ya existan
<code>-r, --relative</code>	crea enlaces simbólicos relativos al lugar del enlace
<code>-s, --symbolic</code>	crea enlaces simbólicos en vez de enlaces duros



1. Comandos para manejo de ficheros

```
/tmp ► ls -li | grep test
5374043 drwxrwxr-x 2 javinator9889 javinator9889 4096 feb  4 12:51 test
5374044 drwxrwxr-x 2 javinator9889 javinator9889 4096 feb  4 12:51 test2
/tmp ► echo "El contenido de un nuevo fichero" > test/data.txt
/tmp ► cat test/data.txt
El contenido de un nuevo fichero
/tmp ► ls -li test/data.txt
5374045 -rw-rw-r-- 1 javinator9889 javinator9889 33 feb  4 12:53 test/data.txt
/tmp ► cd test2
/tmp/test2 ► ln -s ../test/data.txt
/tmp/test2 ► ls -li data.txt
5374046 lrwxrwxrwx 1 javinator9889 javinator9889 16 feb  4 12:54 data.txt → ../test/data.txt
/tmp/test2 ► echo "Añado una nueva línea al final" >> data.txt
/tmp/test2 ► cat data.txt
El contenido de un nuevo fichero
Añado una nueva línea al final
/tmp/test2 ► cat /tmp/test/data.txt
El contenido de un nuevo fichero
Añado una nueva línea al final
/tmp/test2 ► rm /tmp/test/data.txt
/tmp/test2 ► ls -li data.txt
5374046 lrwxrwxrwx 1 javinator9889 javinator9889 16 feb  4 12:54 data.txt → ../test/data.txt
/tmp/test2 ► cat data.txt
cat: data.txt: No existe el archivo o el directorio
```

1. Comandos para manejo de ficheros

```
/tmp ▶ echo "Un nuevo fichero" > test/file.dat
/tmp ▶ cat test/file.dat
Un nuevo fichero
/tmp ▶ ls -li test/file.dat
5374045 -rw-rw-r-- 1 javinator9889 javinator9889 17 feb  4 13:07 test/file.dat
/tmp ▶ cd test2
/tmp/test2 ▶ ln ../test/file.dat
/tmp/test2 ▶ ls -li file.dat
5374045 -rw-rw-r-- 2 javinator9889 javinator9889 17 feb  4 13:07 file.dat
/tmp/test2 ▶ echo "Nueva línea" >> file.dat
/tmp/test2 ▶ cat file.dat
Un nuevo fichero
Nueva línea
/tmp/test2 ▶ cat ../test/file.dat
Un nuevo fichero
Nueva línea
/tmp/test2 ▶ rm ../test/file.dat
/tmp/test2 ▶ ls -li file.dat
5374045 -rw-rw-r-- 1 javinator9889 javinator9889 30 feb  4 13:10 file.dat
/tmp/test2 ▶ cat file.dat
Un nuevo fichero
Nueva línea
```

1. Comandos para manejo de ficheros

```
javinator9889@Tony-MkIII ▶ /tmp ▶ whatis cp  
cp (1)                - copy files and directories
```

Algunas opciones útiles:

<code>--backup[=CONTROL]</code>	crea una copia de seguridad de cada fichero de destino que exista
<code>-f, --force</code>	si un fichero de destino no se puede abrir, lo borra y lo intenta de nuevo (no se tiene en cuenta si se utiliza también la opción <code>-n</code>)
<code>-i, --interactive</code>	pide confirmación antes de sobrescribir
<code>-l, --link</code>	crea enlaces duros de los ficheros en vez de copiarlos
<code>-R, -r, --recursive</code>	copia directorios recursivamente
<code>-s, --symbolic-link</code>	crea enlaces simbólicos en lugar de copiarlos
<code>-u, --update</code>	copia solamente cuando el fichero ORIGEN es más moderno que el fichero de destino, o cuando falta el fichero de destino

1. Comandos para manejo de ficheros

Ejemplos:

```
# Hacer una copia de un fichero / directorio
```

```
cp file.txt file.old.txt
```

```
# Copiar ficheros del mismo tipo a otro directorio de forma interactiva
```

```
cp -i *.txt /tmp/txtfiles
```

```
# Si el fichero es más moderno, actualiza la versión antigua
```

```
cp -u main.cpp src/
```

```
# Si ya hay un fichero con el mismo nombre, crea una copia de seguridad del mismo
```

```
cp --backup=existing file.dat /usr/files/
```

```
# Opciones de copia
```

```
javinator9889@Tony-MkIII ▶ ~ ▶ cp --backup=
```

```
existing  nil      -- numbered backups, if they already exist
```

```
none      off      -- never make backups
```

```
numbered  t        -- make numbered backups
```

```
simple     never    -- always make simple backups
```



1. Comandos para manejo de ficheros

```
javinator9889@Tony-MkIII ▶ /tmp ▶ whatis cat  
cat (1)                - concatenate files and print on the standard output
```

Algunas opciones útiles:

-E, --show-ends	display \$ at end of each line
-n, --number	number all output lines



1. Comandos para manejo de ficheros

Ejemplos:

```
javinator9889@Tony-MkIII ▶ /tmp ▶ cat > file
hola
mundo
estoy escribiendo
^C
javinator9889@Tony-MkIII ▶ /tmp ▶ cat file
hola
mundo
estoy escribiendo
javinator9889@Tony-MkIII ▶ /tmp ▶ cat < file > otro_fichero
javinator9889@Tony-MkIII ▶ /tmp ▶ cat otro_fichero
hola
mundo
estoy escribiendo
javinator9889@Tony-MkIII ▶ /tmp ▶ cat file otro_fichero
hola
mundo
estoy escribiendo
hola
mundo
estoy escribiendo
```

1. Comandos para manejo de ficheros

```
javinator9889@Tony-MkIII ▶ /tmp ▶ whatis nano
nano (1)                - Nano's ANOther editor, inspired by Pico
```

Algunas opciones útiles:

Opción	Opción larga	Significado
-B	--backup	Guardar respaldo de los ficheros existentes
-E	--tabstospaces	Convertir tabulaciones a espacios al escribir
-Y <nombre>	--syntax=<nombre>	Definición de sintaxis a usar para coloreado
-i	--autoindent	Sangrar automáticamente nuevas líneas
-l	--linenumbers	Mostrar los números de línea delante del texto
-m	--mouse	Habilitar el uso del ratón
-t	--tempfile	Guardar automáticamente al salir, sin preguntar



1. Comandos para manejo de ficheros

Ejemplos:

```
javinator9889@Tony-MkIII ▶ /tmp ▶ nano test.py -l -i
```

```
GNU nano 4.8
1 print("Hello world from Python inside nano")
2 exit(0)|
3
```

Opciones dentro del editor nano:

^G Ver ayuda	^O Guardar	^W Buscar	^K Cortar Texto	^J Justificar	^C Posición	M-U Deshacer	M-A Marcar texto	M-] A llave	M-Q Anterior	^B Atrás	^◀ Palabr ant
^X Salir	^R Leer fich.	^\ Reemplazar	^U Pegar	^T Ortografía	^_ Ir a línea	M-E Rehacer	M-6 Copiar	^Q Buscar atrás	M-W Siguiente	^F Adelante	^▶ Palabra Siguiente

<https://www.nano-editor.org/dist/latest/cheatsheet.html>



1. Comandos para manejo de ficheros

```
javinator9889@Tony-MkIII ▶ /tmp ▶ whatis less  
less (1)                - opposite of more
```

Algunas opciones útiles:

```
man less
```

- Pulsar la tecla 'q' para salir.
- Enter para bajar página.
- Espacio para bajar más páginas.
- También sirven Av. Pag, Re. Pag, y las flechas del teclado

```
less filename  
cat filename | less
```



2. Comandos para búsqueda en ficheros

```
javinator9889@Tony-MkIII ► ~ ► whatis grep
grep (1)                - print lines that match patterns
```

Algunas opciones útiles:

-E, --extended-regexp	PATTERNS are extended regular expressions
-i, --ignore-case	ignore case distinctions in patterns and data
-c, --count	imprimir solo un número de líneas seleccionadas por ARCHIVO
-H, --with-filename	imprimir el nombre de archivo con las líneas de salida
-l, --files-with-matches	imprimir solo nombres de ARCHIVOS con líneas seleccionadas
-r, --recursive	equivalente a -directories=recurse
-v, --invert-match	selecciona las líneas que no coinciden
-n, --line-number	imprimir el número de línea con las líneas de salida



2. Comandos para búsqueda en ficheros

Ejemplos:

```
javinator9889@Tony-MkIII ► ~ ► grep root /etc/passwd  
root:x:0:0:root:/root:/bin/bash
```

```
javinator9889@Tony-MkIII ► ~ ► grep -n error /var/log/syslog
```

```
javinator9889@Tony-MkIII ► ~ ► ls -l | grep -E "*.mp4"  
-rw-rw-r-- 1 javinator9889 javinator9889 39974143 ago 27 14:00 wmihai-mkii.mp4  
-rw-rw-r-- 1 javinator9889 javinator9889 38054330 ago 27 13:52 wmihai.mp4
```



2. Comandos para búsqueda en ficheros

```
javinator9889@Tony-MkIII ► ~ ► whatis find  
find (1)                - search for files in a directory hierarchy
```

Algunas opciones útiles:

```
-type c  
    b      block (buffered) special  
    c      character (unbuffered) special  
    d      directory  
    p      named pipe (FIFO)  
    f      regular file  
    l      symbolic link;  
    s      socket  
    D      door (Solaris)  
-name pattern  
-exec command ;  
-size +/-n[cwbkMG]
```

2. Comandos para búsqueda en ficheros

Ejemplos:

```
javinator9889@Tony-MkIII ► ~ ► find . -size +10240k -size -20480k -exec rm -i {} \;
```

Busca todos los archivos mayores de 10MB y menores de 20MB y los borra de forma interactiva.

```
javinator9889@Tony-MkIII ► ~ ► find . -type d -exec chmod 0750 {} \;
```

Busca todas las carpetas en el directorio actual y cambia sus permisos a 0750

```
javinator9889@Tony-MkIII ► ~ ► find /etc -name '*.conf'
```

Busca todos los ficheros de configuración (.conf) en la ruta /etc



3. Comandos para filtrar ficheros

```
javinator9889@Tony-MkIII ► ~ ► whatis sort  
sort (1)                - sort lines of text files
```

Algunas opciones útiles:

<code>-r, --reverse</code>	reverse the result of comparisons
<code>-f, --ignore-case</code>	convierte las minúsculas en mayúsculas
<code>-g, --general-numeric-sort</code>	compara de acuerdo con el valor numérico general

Ejemplos:

```
javinator9889@Tony-MkIII ► ~ ► sort -f /etc/passwd  
Ordena alfabéticamente el fichero /etc/passwd
```



3. Comandos para filtrar ficheros

```
javinator9889@Tony-MkIII ► ~ ► whatis uniq  
uniq (1)                - report or omit repeated lines
```

Algunas opciones útiles:

-c, --count	precede a las líneas con el número de ocurrencias
-d, --repeated	muestra sólo las líneas duplicadas, una por cada grupo
-i, --ignore-case	pasa por alto las diferencias entra mayúsculas y minúsculas

Ejemplos:

```
javinator9889@Tony-MkIII ► /tmp ► echo "hola" > hola.txt  
javinator9889@Tony-MkIII ► /tmp ► echo "hola" >> hola.txt  
javinator9889@Tony-MkIII ► /tmp ► uniq -c hola.txt  
      2 hola
```



3. Comandos para filtrar ficheros

```
javinator9889@Tony-MkIII ► ~ ► whatis head tail
head (1)                - output the first part of files
tail (1)                 - output the last part of files
```

Algunas opciones útiles:

```
-f, --follow[={name|descriptor}] muestra a medida que el fichero crece;
                                sin opción significa 'descriptor'
-n, --lines=[+]NUM             output the last NUM lines, instead of the last 10;
                                or use -n +NUM to output starting with line NUM
```

Ejemplos:

```
javinator9889@Tony-MkIII ► /tmp ► tail -f /var/log/syslog
    Actualizaciones en tiempo real sobre los eventos que suceden en el sistema
```



3. Comandos para filtrar ficheros

```
javinator9889@Tony-MkIII ► ~ ► whatis cut  
cut (1)                - remove sections from each line of files
```

Algunas opciones útiles:

```
-c, --characters=LISTA  selecciona solamente estos caracteres  
-d, --delimiter=DELIM  usa DELIM en vez de caracteres de tabulación para delimitar los  
campos  
-f, --fields=LISTA      selecciona solamente estos campos; también muestra  
                        cualquier línea que no tenga un carácter  
                        delimitador, a menos que se especifique la  
                        opción -s
```

Ejemplos:

```
javinator9889@Tony-MkIII ► /tmp ► cut -d':' -f4 /etc/passwd  
Muestra los UIDs de los usuarios en el fichero "passwd"
```

3. Comandos para filtrar ficheros

Una combinación de todos ellos

```
javinator9889@Tony-MkIII ► /tmp ► cut -d ' ' -f1 el_quijote.txt | sort | uniq -c | sort
```

```
23 Con  
25 -Y  
26 de  
29 A  
31 -Así  
35 El  
35 que  
38 -Pues  
42 No  
53 Capítulo  
56 En  
79 -No  
95 Y
```

3. Comandos para filtrar ficheros

```
javinator9889@Tony-MkIII ► ~ ► whatis sed  
sed (1)                - stream editor for filtering and transforming text
```

Algunas opciones útiles:

```
-n, --quiet, --silent  
                        suprime la muestra automática del espacio de patrones  
-e guión, --expression=guión  
                        agrega el guión a la lista de órdenes para ejecutar  
-f fichero-guión, --file=fichero-guión  
                        agrega el contenido del fichero guión a la lista de órdenes  
                        para ejecutar
```



3. Comandos para filtrar ficheros

Los comandos `sed` cuentan con una estructura definida y marcada:

`[inicio[,fin]] función [argumentos]`

donde:

- `[inicio[,fin]]` hacen referencia a las líneas que se van a tratar.
- `función` hace referencia a la acción que se va a realizar.
- `[argumentos]` indican cómo se va a aplicar la acción.

Los comandos más habituales que se van a encontrar son:

- `a\`: añade al final de línea
- `c\`: cambia el contenido del patrón de texto
- `d`: borra las líneas indicadas
- `s`: realiza sustituciones
- `g`: realiza sustituciones generales
- `i\`: inserta líneas
- `p`: imprime líneas
- `q`: abandona `sed`
- `r arch`: lee el fichero y lo añade a la salida
- `w arch`: copia las líneas de un fichero
- `=`: da el número de línea
- `/^tex/`: realiza un `grep` simple buscando “`tex`”

3. Comandos para filtrar ficheros

Ejemplos:

```
javinator9889@Tony-MkIII ▶ ~ ▶ sed "3d" canciones
```

Muestra por pantalla el contenido de canciones pero borra la tercera línea

```
javinator9889@Tony-MkIII ▶ ~ ▶ sed "a\Hola" canciones
```

Produce una salida donde, después de cada línea, se escribe "hola"

```
javinator9889@Tony-MkIII ▶ ~ ▶ sed "i\Hola" canciones
```

Produce una salida donde, antes de cada línea, se escribe "hola"

```
javinator9889@Tony-MkIII ▶ ~ ▶ sed -n 2p canciones
```

Muestra la segunda línea de canciones

```
javinator9889@Tony-MkIII ▶ ~ ▶ sed -n '3,$p' canciones
```

Muestra desde la tercera línea al final. Se usan comillas simples para evitar que se interprete "\$p" como una variable

3. Comandos para filtrar ficheros

Ejemplos:

```
javinator9889@Tony-MkIII ▶ ~ ▶ sed -n /^ho/ canciones
```

Muestra solo las líneas que comienzan por "ho"

```
javinator9889@Tony-MkIII ▶ ~ ▶ sed -n '/^t/, $p' canciones
```

Muestra desde la primera línea que comienza por 't' hasta el final

```
javinator9889@Tony-MkIII ▶ ~ ▶ sed 's/antes/después/g' canciones
```

Sustituye todos los "antes" por "después" en el fichero canciones

```
javinator9889@Tony-MkIII ▶ ~ ▶ sed 's/antes/después/g' canciones
```

Sustituye todos los "antes" por "después" en el fichero canciones

```
javinator9889@Tony-MkIII ▶ ~ ▶ sed '1,6s/antes/después/g' canciones
```

Sustituye todos los "antes" por "después" en el fichero canciones pero solo las líneas 1, 6

3. Comandos para filtrar ficheros

Ejemplos:

```
javinator9889@Tony-MkIII ▶ ~ ▶ sed '/^en/s/jose/pepe/g' canciones
```

Sustituye todos los "jose" por "pepe" pero solo aquellas líneas que comiencen por "en"

```
javinator9889@Tony-MkIII ▶ ~ ▶ sed 'y/[123]/[456]' canciones
```

Sustituye '1' por '4', '2' por '5' y '3' por '6'

```
javinator9889@Tony-MkIII ▶ ~ ▶ sed 's/juana/#&' canciones
```

Sustituye "Juana" por "#Juana" (el '&' representa el texto buscado)

```
javinator9889@Tony-MkIII ▶ ~ ▶ sed '/pirata/p' canciones
```

Solo muestra las líneas que contengan "pirata"

sed se apoya completamente en las expresiones regulares las cuales permiten definir patrones de texto complejo para obtener los resultados buscados



4. Comandos para comprimir ficheros

```
javinator9889@Tony-MkIII ▶ ~ ▶ whatis gzip
gzip (1)                - compress or expand files
```

Algunas opciones útiles:

```
-d, --decompress  decompress
-l, --list        list compressed file contents
-r, --recursive  operate recursively on directories
-1, --fast        compress faster
-9, --best        compress better
-S, --suffix=SUF use suffix SUF on compressed files
-k, --keep        keep (don't delete) input files
```

Ejemplos:

```
javinator9889@Tony-MkIII ▶ /tmp ▶ ls -lh el_quijote.txt
-rw-rw-r-- 1 javinator9889 javinator9889 1,1M feb 23 10:14 el_quijote.txt
javinator9889@Tony-MkIII ▶ /tmp ▶ gzip -9 el_quijote.txt
javinator9889@Tony-MkIII ▶ /tmp ▶ ls -lh el_quijote.txt.gz
-rw-rw-r-- 1 javinator9889 javinator9889 353K feb 23 10:14 el_quijote.txt.gz
```

4. Comandos para comprimir ficheros

```
javinator9889@Tony-MkIII ► ~ ► whatis tar
tar (1)                - an archiving utility
```

Algunas opciones útiles:

-c, --create	crea un nuevo archivo
-r, --append	añade ficheros al final del archivo
-u, --update	solamente añade ficheros más recientes que la copia del archivo
-x, --extract, --get	extrae ficheros de un archivo
-a, --auto-compress	utiliza el sufijo del archivo para determinar el programa de compresión
-j, --bzip2	filtra el archivo a través de bzip2
-J, --xz	filtra el archivo a través de xz
--lzip	filtra el archivo a través de lzip
--lzma	filtra el archivo a través de xz
--lzop	filtra el archivo a través de lzop
--no-auto-compress	no utiliza el sufijo del archivo para determinar el programa de compresión
-z, --gzip, --gunzip, --ungzip	filtra el archivo a través de gzip
--zstd	filtra el archivo a través de zstd
-f, --file=ARCHIVO	utiliza un archivo o el dispositivo ARCHIVO

4. Comandos para comprimir ficheros

Ejemplos:

```
javinator9889@Tony-MkIII ▶ ~ ▶ tar cvzf grande *
```

Crea un fichero "grande" donde se guardarán todos los contenidos del directorio actual comprimidos

```
javinator9889@Tony-MkIII ▶ ~ ▶ tar xvzf grande
```

Extrae el contenido del fichero "grande" y lo descomprime

```
javinator9889@Tony-MkIII ▶ ~ ▶ tar cf uconf.tar /etc/passwd /etc/shadow /etc/group
```

Crea el fichero "uconf.tar" que contiene los ficheros /etc/{passwd,shadow,group}

```
javinator9889@Tony-MkIII ▶ ~ ▶ tar xf uconf.tar
```

Desagrupa el fichero "uconf.tar" en el directorio actual

```
javinator9889@Tony-MkIII ▶ ~ ▶ tar cJvf fichero.tar.xz *
```

Crea un fichero "fichero.tar.xz" con el contenido del directorio actual usando "xz" como método de compresión



5. Comandos para desconectar el sistema

```
javinator9889@Tony-MkIII ▶ ~ ▶ whatis exit  
exit (2)                - terminate the calling process
```

```
javinator9889@Tony-MkIII ▶ ~ ▶ whatis logout  
logout (3)              - write utmp and wtmp entries
```

```
javinator9889@Tony-MkIII ▶ ~ ▶ whatis shutdown  
shutdown (8)            - Halt, power-off or reboot the machine
```



5. Comandos para desconectar el sistema

```
javinator9889@Tony-MkIII ► ~ ► shutdown --help  
shutdown [OPTIONS ...] [TIME] [WALL ...]
```

Shut down the system.

Options:

--help	Show this help
-H --halt	Halt the machine
-P --poweroff	Power-off the machine
-r --reboot	Reboot the machine
-h	Equivalent to --poweroff, overridden by --halt
-k	Don't halt/power-off/reboot, just send warnings
--no-wall	Don't send wall message before halt/power-off/reboot
-c	Cancel a pending shutdown

```
shutdown -h now      # Apaga el equipo inmediatamente  
shutdown -h 20:00    # Apaga el equipo a las 20:00  
shutdown -h +4       # Apaga el equipo dentro de 4 minutos
```



6. Comandos para gestión de redes

```
javinator9889@Tony-MkIII ► ~ ► whatis tracepath  
tracepath (8)          - traces path to a network host discovering MTU along this path
```

Algunas opciones útiles:

```
-4          use IPv4  
-6          use IPv6  
-b          print both name and ip  
-n          no dns name resolution  
-p <port>   use destination <port>
```

```
javinator9889@Tony-MkIII ► ~ ► tracepath google.com  
Muestra la ruta entre el host y el destino que indicamos, mostrando los pasos intermedios
```



6. Comandos para gestión de redes

```
javinator9889@Tony-MkIII ▶ ~ ▶ whatis write
write (1)                - send a message to another user
javinator9889@Tony-MkIII ▶ ~ ▶ whatis write
wall (1)                 - write a message to all users
```

```
javinator9889@Tony-MkIII ▶ ~ ▶ write javinator9889 tty1
javinator9889@Tony-MkIII ▶ ~ ▶ mesg y  # habilito mensajes
javinator9889@Tony-MkIII ▶ ~ ▶ mesg n  # deshabilito mensajes
javinator9889@Tony-MkIII ▶ ~ ▶ wall
Este es un mensaje
Que recibirán todos
Los usuarios
Ctrl - d
javinator9889@Tony-MkIII ▶ ~ ▶
```



6. Comandos para gestión de redes

```
javinator9889@Tony-MkIII ► ~ ► whatis netstat
netstat (8)          - Print network connections, routing tables, interface statistics,
masquerade connections, and multicast memberships
```

Algunas opciones útiles:

```
-i, --interfaces      muestra la tabla de interfaz
-s, --statistics      muestra las estadísticas de la red (como SNMP)
-l, --listening       muestra los sockets a la escucha de servidores
-p, --programs        muestra el nombre PID/Programa para los sockets
-o, --timers          muestra temporizadores
-c, --continuous      listado continuo
<Socket> ={-t|--tcp} {-u|--udp} {-U|--udplite} {-S|--sctp} {-w|--raw}
           {-x|--unix} --ax25 --ipx --netrom
```



6. Comandos para gestión de redes

Ejemplos:

```
javinator9889@Tony-MkIII ▶ ~ ▶ netstat -i
```

Muestra las interfaces de red y sus características

```
javinator9889@Tony-MkIII ▶ ~ ▶ netstat -lt
```

Aplicaciones que están escuchando ("listening") en un puerto TCP

```
javinator9889@Tony-MkIII ▶ ~ ▶ netstat -lu
```

Aplicaciones que están escuchando ("listening") en un puerto UDP

```
javinator9889@Tony-MkIII ▶ ~ ▶ sudo netstat -tulpn
```

Aplicaciones que están usando puertos TCP o UDP, que están escuchando conexiones ("listening"), que muestre su PID y que muestre las direcciones en formato numérico (que no se resuelvan a nombres de host)



6. Comandos para gestión de redes

```
javinator9889@Tony-MkIII ► ~ ► whatis wget  
wget (1) - The non-interactive network downloader.
```

Algunas opciones útiles:

-b,	--background	se va a segundo plano después de empezar
-o,	--output-file=FICHERO	registra mensajes en FICHERO
-O,	--output-document=FILE	escribe documentos al fichero FILE
-q,	--quiet	silencioso (sin texto de salida)
-c,	--continue	continua una descarga parcial de un fichero
	--user=USUARIO	pone el usuario de ambos ftp y http a USUARIO
	--password=CONTRASEÑA	pone la contraseña de ambos ftp y http a CONTRASEÑA
-r,	--recursive	especifica descarga recursiva

Ejemplos:

```
javinator9889@Tony-MkIII ► ~ ► wget -O sizes.min.js https://afarkas.github.io/lazysizes/lazysizes.min.js  
Descarga el fichero "lazysizes.min.js" en la ruta "sizes.min.js"
```



6. Comandos para gestión de redes

```
javinator9889@Tony-MkIII ► ~ ► whatis route  
route (8)                - show / manipulate the IP routing table
```

Ejemplos:

```
javinator9889@Tony-MkIII ► ~ ► route
```

Tabla de rutas IP del núcleo

Destino	Pasarela	Genmask	Indic	Métric	Ref	Uso	Interfaz
default	192.168.1.1	0.0.0.0	UG	600	0	0	wlo1
link-local	0.0.0.0	255.255.0.0	U	1000	0	0	virbr0
192.168.1.0	0.0.0.0	255.255.255.0	U	600	0	0	wlo1
192.168.122.0	0.0.0.0	255.255.255.0	U	0	0	0	virbr0



6. Comandos para gestión de redes

```
javinator9889@Tony-MkIII ► ~ ► whatis who
who (1)                - show who is logged on
```

Algunas opciones útiles:

-a, --all	lo mismo que -b -d --login -p -r -t -T -u
-b, --boot	tiempo del último inicio del sistema
-d, --dead	muestra los procesos muertos
-H, --heading	muestra la línea de encabezados de columnas
-l, --login	muestra los procesos de login del sistema
-m	sólo el nombre del 'host' y de usuario asociado con la entrada estándar
-p, --process	muestra los procesos activos lanzados por init
-q, --count	todos los nombres de entrada y número de usuarios conectados
-t, --time	muestra el último cambio en el reloj del sistema
-T, -w, --mesg	añade el estado de mensajes del usuario como +, - ó ?

6. Comandos para gestión de redes

Ejemplos:

```
javinator9889@Tony-MkIII ▶ ~ ▶ who  
javinator9889 :1          2021-03-02 09:24 (:1)
```

```
javinator9889@Tony-MkIII ▶ ~ ▶ who -H  
NOMBRE  LÍNEA      TIEMPO          COMENTARIO  
javinator9889 :1          2021-03-02 09:24 (:1)
```

```
javinator9889@Tony-MkIII ▶ ~ ▶ who -q  
javinator9889  
N° de usuarios=1
```

```
javinator9889@Tony-MkIII ▶ ~ ▶ who -a  
          arranque del sistema 2021-03-02 09:22  
          `run-level' 5 2021-03-02 09:22  
javinator9889 ? :1          2021-03-02 09:24    ?          35379 (:1)
```



6. Comandos para gestión de redes

```
javinator9889@Tony-MkIII ► ~ ► whatis w
w (1)                - Show who is logged on and what they are doing.
```

Algunas opciones útiles:

-h, --no-header	no mostrar la cabecera
-u, --no-current	ignorar nombre de usuario de proceso actual
-s, --short	formato corto
-f, --from	mostrar campo de nombre de equipo remoto

```
javinator9889@Tony-MkIII ► ~ ► w
11:51:26 up 2:29, 1 user, load average: 0,86, 1,42, 1,26
USUARIO  TTY      DE              LOGIN@  IDLE   JCPU   PCPU   WHAT
javinato :1        :1              09:24   ?xdm?  1:21m  0.01s /usr/lib/gdm3/gdm-x-session --run-script
env GNOME_SHELL_SESSION_MODE=ubuntu /usr/bin/gnome-session --systemd --session=ubuntu
```



6. Comandos para gestión de redes

```
javinator9889@Tony-MkIII ► ~ ► whatis ping
ping (8)                - send ICMP ECHO_REQUEST to network hosts
```

Algunas opciones útiles:

```
-c <count>          stop after <count> replies
-i <interval>       seconds between sending each packet
-s <size>           use <size> as number of data bytes to be sent
-q                 quiet output
```

```
javinator9889@Tony-MkIII ► ~ ► ping -c 3 javinator9889.com
PING javinator9889.com (80.29.175.241) 56(84) bytes of data.
64 bytes from 241.red-80-29-175.dynamicip.rima-tde.net (80.29.175.241): icmp_seq=1 ttl=64
time=3.16 ms
64 bytes from 241.red-80-29-175.dynamicip.rima-tde.net (80.29.175.241): icmp_seq=2 ttl=64
time=2.76 ms
64 bytes from 241.red-80-29-175.dynamicip.rima-tde.net (80.29.175.241): icmp_seq=3 ttl=64
time=24.1 ms

--- javinator9889.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 2.755/9.997/24.081/9.959 ms
```

7. SSH

```
javinator9889@Tony-MkIII ► ~ ► whatis ssh  
ssh (1)                - OpenSSH remote login client
```

ssh es un cliente que nos permite conectarnos de manera remota y completamente segura a una máquina en la cual tengamos credenciales.

La sintaxis de funcionamiento es de este estilo:

ssh [user]@host

Aunque se admiten muchas opciones de personalización y configuración:

- C solicita comprimir todos los datos que se envían (útil en redes lentas, contraproducente en redes veloces)
- i file utiliza el fichero especificado como clave privada (por defecto, ubicado en el directorio .ssh)
- p port se conecta al host remoto al puerto en cuestión (por defecto, el 22)
- X habilita el envío y reenvío de X11
- x deshabilita el envío y reenvío de X11



7. SSH

ssh se puede ejecutar tanto en modo cliente como en modo servidor. Para este último, existe un fichero de configuración con múltiples opciones que permiten habilitar, ajustar o modificar características del sistema. Dicho fichero se encuentra en la ruta
`/etc/ssh/sshd_config`

Algunos ajustes interesantes son:

Port	22
SyslogFacility	AUTH
LogLevel	INFO
PermitRootLogin	yes
LoginGraceTime	120
X11Forwarding	yes
PasswordAuthentication	no
PubkeyAuthentication	yes
Match User	username
ChrootDirectory	/directory

```
sudo systemctl reload sshd
```

