

# Repaso comandos GNU/Linux - Shell bash

## ESCENARIO

### Máquinas virtuais:

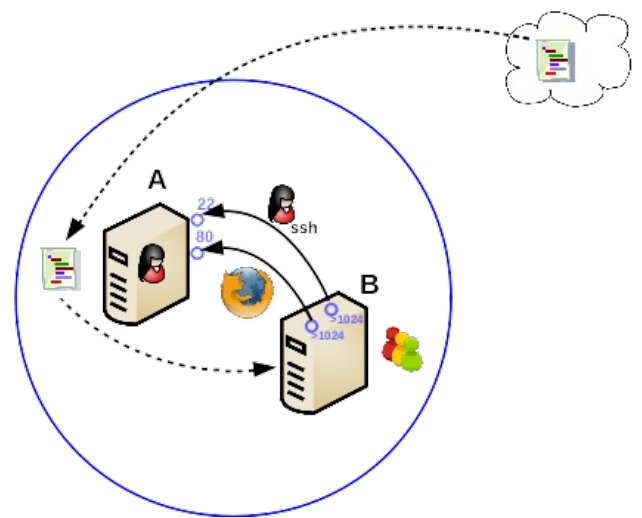
RAM ≤ 2048MB CPU ≤ 2 PAE/NX habilitado  
Rede: 192.168.120.0

### Máquina virtual A:

Rede Interna e NAT  
Servidor SSH  
Servidor Web: Apache  
ISO: Debian Live amd64 xfce  
IP/MS: 192.168.120.100/24

### Máquina virtual B:

Rede Interna  
Cliente SSH  
Cliente Web: Navegador  
ISO: Kali Live amd64  
IP/MS: 192.168.120.101/24



## NOTAS:

- **Conexión remota a través de cliente/servidor SSH:** Instalaremos na Máquina Virtual A o servidor SSH, de tal xeito que a maioría da práctica coa súa execución de comandos farase conectando co cliente SSH dende a Máquina Virtual B.
- **usuarios.csv:** Ficheiro plantilla a empregar na práctica. Descargar da Aula Virtual no cartafol \$HOME co usuario co que vas a realizar a práctica.
- **cp -pv \$HOME/usuarios.csv /tmp:** Copiar o ficheiro en /tmp en modo verbose (detallado) e preservando permisos e datas. Este comando pode ser necesario empregalo máis dunha vez, aínda que non se explique explicitamente na práctica, para voltar a ter copia do ficheiro orixinal en /tmp, por exemplo antes de executar as OPCIÓN 2 que se amosan no contido desta práctica.

## Práctica

### Máquina virtual A: Debian amd64

1. Cambiar contrasinais. Así na contorna gráfica (shell xfce) abrir un terminal e executar:  
user@debian:~\$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.  
user@debian:~\$ passwd #Crear (ou modificar non caso que tivera contrasinal) o contrasinal para o usuario de nome user, xa que a non pór nome de usuario logo do comando passwd interprétase que se quere cambiar o contrasinal do usuario que está a executar o comando. Débese escribir 2 veces e non se amosará ningún carácter namentras se escriba por seguridade. Pór como contrasinal **abc123**. (Olo que o contrasinal ten un carácter punto final).  
user@debian:~\$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)  
root@debian:~\$ passwd #Crear (ou modificar non caso que tivera contrasinal) o contrasinal para o usuario de nome root, xa que a non pór nome de usuario logo do comando passwd interprétase que se quere cambiar o contrasinal do usuario que está a executar o comando. Débese escribir 2 veces e non se amosará ningún carácter namentras se escriba por seguridade. Pór como contrasinal **abc123**. (Olo que o contrasinal ten un carácter punto final).
2. Instalar Servidor SSH:  
root@debian:~# apt update #Actualizar o listado de paquetes dos repositorios (/etc/apt/sources.list, /etc/apt/sources.list.d/)  
root@debian:~# apt search ssh #Buscar calquera paquete que coincida co patrón de búsqueda openssh-server  
root@debian:~# apt -y install openssh-server #Instalar o paquete openssh-server, é dicir, instalar o servidor SSH openssh-server. Co parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na instalación do paquete.
3. Configuración da rede:  
root@debian:~# /etc/init.d/avahi-daemon stop #Parar o demo avahi-daemon(control resolución de nomes) para poder configurar de forma manual a configuración de rede e non ter conflito con este demo.  
root@debian:~# /etc/init.d/network-manager stop #Parar o demo network-manager(xestor de rede) para poder configurar de forma manual a configuración de rede e non ter conflito con este xestor.  
root@debian:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina A, as tarxetas de redes: loopback(lo), NAT(enps03) e interna(enps08).  
root@debian:~# ip addr add 192.168.120.100/24 dev enp0s3 #Configurar a tarxeta de rede interna enp0s8, coa IP: 192.168.120.100 e máscara de subrede: 255.255.255.0.

root@debian:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina A, as tarxetas de redes: loopback(lo), NAT(enps03) e interna(enps08).

root@debian:~# ping -c4 192.168.120.100 #Comprobar mediante o comando ping a conectividade coa interface de rede local enp0s8

4. Usuario alumno para poder acceder dende Kali (Máquina virtual B). De momento, seguimos executando os comandos dende a Máquina Virtual A. Así:

root@debian:~# useradd -m -d /home/alumno -s /bin/bash alumno #Crear o usuario *alumno* co comando *useradd*, onde:

- d /home/alumno → Xera a casa do usuario, é dicir, o directorio de traballo do usuario, no cartafol /home/alumno
- m → Copia na casa do usuario o que exista no cartafol /etc/skel
- s /bin/bash → Establece como shell de traballo para o usuario a shell bash
- alumno → Establece como nome de autenticación de usuario o nome alumno

root@debian:~# id alumno #Amosa información de usuario e grupo do usuario de nome alumno

root@debian:~# groups alumno #Amosa os grupos aos cales pertence o usuario de nome alumno

root@debian:~# grep alumno /etc/passwd /etc/shadow #Buscar o patrón de caracteres alumno dentro dos ficheiros /etc/passwd e /etc/shadow. Debe amosar que existen entradas, pero como aínda non xeramos o contrasinal do usuario alumno non amosará en /etc/shadow un contrasinal cifrado, senón o carácter admiración final '!', o que significa que o usuario está bloqueado, non podendo acceder ao sistema.

root@debian:~# passwd alumno #Crear (ou modificar non caso que tivera contrasinal) o contrasinal para o usuario de nome *alumno*. Débese escribir 2 veces e non se amosará ningún carácter namentras se escriba por seguridade. Pór como contrasinal **abc123**. (Ollo que o contrasinal ten un carácter punto final).

root@debian:~# grep alumno /etc/passwd /etc/shadow #Buscar o patrón de caracteres alumno dentro dos ficheiros /etc/passwd e /etc/shadow. Debe amosar que existen entradas, e agora como xa xeramos o contrasinal do usuario alumno amosará en /etc/shadow un contrasinal cifrado, podendo éste acceder ao sistema.

root@debian:~# echo -e "abc123.\nabc123." | passwd alumno #Modificar o contrasinal para o usuario de nome *alumno*. O comando passwd recolle o contrasinal repetido *abc123*. a través do comando anterior *echo*, tal que mediante o parámetro -e interprétase \n como salto de liña. Ollo!: O contrasinal con este comando vai en texto claro e ademais o comando é recordado polo historial de comandos (~/.bash\_history).

root@debian:~# userdel -r alumno #Eliminar o usuario alumno e coa opción -r eliminar tamén o seu cartafol de usuario (/home/alumno) e se existe o seu cartafol de correo (var/mail/alumno ou /var/spool/mail/alumno; depende da varialbe MAIL\_DIR definida no ficheiro /etc/login.defs)

root@debian:~# id alumno #Amosa erro, porque quere amosar información de usuario e grupo do usuario de nome alumno, pero o usuario non existe.

root@debian:~# groups alumno #Amosa erro, porque quere amosar os grupos aos cales pertence o usuario de nome alumno, pero o usuario non existe

root@debian:~# useradd -m -d /home/alumno -s /bin/bash -p \$(mkpasswd abc123.) alumno

#Crear o usuario *alumno* co comando *useradd*, onde:

- d /home/alumno → Xera a casa do usuario, é dicir, o directorio de traballo do usuario, no cartafol /home/alumno
- m → Copia na casa do usuario o que exista no cartafol /etc/skel
- s /bin/bash → Establece como shell de traballo para o usuario a shell bash
- p \$(mkpasswd abc123.) → Establece como contrasinal o valor *abc123*. e mediante o comando *mkpasswd* cifra o texto anterior. Así o texto cifrado é o que se incorpora no ficheiro /etc/shadow
- alumno → Establece como nome de autenticación de usuario o nome alumno

Ollo!: O contrasinal con este comando vai en texto claro e ademais o comando é recordado polo historial de comandos (~/.bash\_history).

root@debian:~# id alumno #Amosa información de usuario e grupo do usuario de nome alumno

root@debian:~# groups alumno #Amosa os grupos aos cales pertence o usuario de nome alumno

root@debian:~# usermod -L alumno #Bloquea o acceso ao usuario alumno

root@debian:~# grep alumno /etc/passwd /etc/shadow #Buscar o patrón de caracteres alumno dentro dos ficheiros /etc/passwd e /etc/shadow. Comprobar en /etc/shadow como aparece o carácter admiración final '!' antes do contrasinal cifrado, co cal o usuario está bloqueado e non pode acceder ao sistema.

root@debian:~# usermod -U alumno #Desbloquea o acceso ao usuario alumno

root@debian:~# grep alumno /etc/passwd /etc/shadow #Buscar o patrón de caracteres alumno dentro dos ficheiros /etc/passwd e /etc/shadow. Comprobar en /etc/shadow como non aparece o carácter admiración final '!' antes do contrasinal cifrado, co cal o usuario non está bloqueado e pode acceder ao sistema.

root@debian:~# passwd -l alumno #Bloquea o acceso ao usuario alumno

root@debian:~# grep alumno /etc/passwd /etc/shadow #Buscar o patrón de caracteres alumno dentro dos ficheiros /etc/passwd e /etc/shadow. Comprobar en /etc/shadow como aparece o carácter admiración final '!' antes do contrasinal cifrado, co cal o usuario está bloqueado e non pode acceder ao sistema.

root@debian:~# passwd -u alumno #Desbloquea o acceso ao usuario alumno

root@debian:~# grep alumno /etc/passwd /etc/shadow #Buscar o patrón de caracteres alumno dentro dos ficheiros /etc/passwd e /etc/shadow. Comprobar en /etc/shadow como non aparece o carácter admiración final '!' antes do contrasinal cifrado, co cal o usuario non está bloqueado e pode acceder ao sistema.

5. Arrancar servidor SSH. Seguimos executando os comandos dende a Máquina Virtual A. Así:

```
root@debian:~# /etc/init.d/ssh status #Comprobar o estado do servidor SSH, por defecto non debe estar
arrancado.
root@debian:~# /etc/init.d/ssh start #Arrancar o servidor SSH.
root@debian:~# /etc/init.d/ssh status #Comprobar o estado do servidor SSH, agora debe estar arrancado.
root@debian:~# nc -vz localhost 22 #Mediante o comando nc(netcat) comprobar que o porto 22 do servidor SSH
está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar
información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o
escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.
root@debian:~# ss -natp | grep 22 #Mediante o comando ss comprobar que o porto 22 do servidor SSH está en
estado escoita(listen), esperando conexións. A opción -n permite non resolver nomes amosando así soamente as IPs e o
comando ser máis rápido na execución. A opción -a equivale á opción all o que permite amosar todos os sockets
(conectores) á escoita no servidor. A opción -t equivale a tcp o que permite buscar soamente información sobre o protocolo
TCP. A opción -p equivale a program e amosa o PID e nome do programa ao cal pertence o socket.
root@debian:~# apt update #Actualizar o listado de paquetes dos repositorios (/etc/apt/sources.list,
/etc/apt/sources.list.d/)
root@debian:~# apt -y install net-tools #Instalar o paquete net-tools, é dicir, instalar o paquete que integra os
comandos ifconfig, route e netstat entre outros. Co parámetro -y automaticamente asumimos yes a calquera pregunta que
ocorra na instalación do paquete.
root@debian:~# netstat -natp | grep 22 #Mediante o comando netstat comprobar que o porto 22 do servidor
SSH está en estado escoita(listen), esperando conexións. A opción -n permite non resolver nomes amosando así soamente
as IPs e o comando ser máis rápido na execución. A opción -a equivale á opción all o que permite amosar todos os sockets
(conectores) á escoita no servidor. A opción -t equivale a tcp o que permite buscar soamente información sobre o protocolo
TCP. A opción -p equivale a program e amosa o PID e nome do programa ao cal pertence o socket.
root@debian:~# ssh -v localhost #Comprobar se o servidor SSH está activo e podemos conectarnos a el dende
localhost. Como é a primeira vez que nos conectamos o servidor avísanos se estamos de acordo coa autenticación.
Respostamos yes e pulsamos Enter. Dá un erro porque o usuario root non pode acceder por ssh na configuración por
defecto. Entón imos acceder como usuario alumno a través de ssh.
root@debian:~# ssh -v alumno@localhost #Comprobar se o servidor SSH está activo e podemos conectarnos a
el dende localhost. Agora accedemos como o usuario alumno a través da conexión cifrada SSH.
root@debian:~# exit #Saír da consola do usuario root, para voltar á consola do usuario user sen permisos de root
user@debian:~$ exit #Saír da consola do usuario user, é dicir, pechar o terminal.
```

6. usuarios.csv: Ficheiro plantilla a empregar na práctica. Descargar da Aula Virtual no cartafol \$HOME do usuario alumno (/home/alumno).

7. Agora minimizamos a máquina virtual A xa que imos traballar de continuo coa máquina virtual B.

## Máquina virtual B: Kali amd64

8. Configuración da rede. Así na contorna gráfica abrir un terminal e executar:

```
kali@kali:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
```

```
kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

```
root@kali:~# /etc/init.d/avahi-daemon stop #Parar o demo avahi-daemon(control resolución de nomes) para poder configurar de forma manual a configuración de rede e non ter conflito con este demo.
```

```
root@kali:~# /etc/init.d/network-manager stop #Parar o demo network-manager(xestor de rede) para poder configurar de forma manual a configuración de rede e non ter conflito con este xestor.
```

```
root@kali:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina B as tarxetas de redes: loopback(lo) e interna(eth0).
```

```
root@kali:~# ip addr add 192.168.120.101/24 dev eth0 #Configurar a tarxeta de rede interna eth0, coa IP: 192.168.120.101 e máscara de subrede: 255.255.255.0.
```

```
root@kali:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina B as tarxetas de redes: loopback(lo) e interna(eth0).
```

```
root@kali:~# ping -c4 192.168.120.101 #Comprobar mediante o comando ping a conectividade coa interface de rede local eth0
```

```
root@kali:~# ping -c4 192.168.120.100 #Comprobar mediante o comando ping a conectividade coa interface de rede da máquina virtual A
```

```
root@kali:~# echo '192.168.120.100 debian' >> /etc/hosts #Engadir no ficheiro /etc/hosts, é dicir, na táboa estática de búsqueda para nomes de host (DNS) o nome debian, para que atenda á IP 192.168.120.100
```

```
root@kali:~# ping -c4 debian #Comprobar mediante o comando ping a conectividade coa interface de rede da máquina virtual A
```

9. Acceder mediante a SSH á máquina virtual A dende a máquina virtual B. A partir de agora executaremos sempre os comandos dende a máquina virtual B, a través da consola SSH:

```
root@kali:~# nc -vz 192.168.100.100 22 #Mediante o comando nc(netcat) comprobar que o porto 22 do servidor SSH está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.
```

```
root@kali:~# nc -vz debian 22 #Mediante o comando nc(netcat) comprobar que o porto 22 do servidor SSH está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.
```

```
root@kali:~# ssh -v alumno@192.168.100.100 #Comprobar se o servidor SSH está activo e podemos conectarnos a el. Agora accedemos como o usuario alumno a través da conexión cifrada SSH.
```

```
alumno@debian:~$ exit #Saír da consola remota ssh, do usuario alumno, á que acabamos de acceder da máquina virtual A, para voltar á consola local de root da máquina virtual B.
```

```
root@kali:~# ssh -v alumno@debian #Comprobar se o servidor SSH está activo e podemos conectarnos a el. Agora accedemos como o usuario alumno a través da conexión cifrada SSH.
```

```
alumno@debian:~$
```

10. Instalar Servidor Web Apache:

```
alumno@debian:~$ sudo su - #Problema: Non se pode acceder co usuario alumno á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

```
alumno@debian:~$ su - user #Acceder como usuario user (como se fixera login nunha consola ttyX). Escribir o contrasinal de user: abc123. (Olo que o contrasinal ten un caracter punto final)
```

```
user@debian:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

```
root@debian:~# apt update #Actualizar o listado de paquetes dos repositorios (/etc/apt/sources.list, /etc/apt/sources.list.d/)
```

```
root@debian:~# apt search apache2 #Buscar calquera paquete que coincida co patrón de búsqueda apache2
```

```
root@debian:~# apt -y install apache2 #Instalar o paquete apache2, é dicir, instalar o servidor HTTP apache2. Co parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na instalación do paquete.
```

```
root@debian:~# nc -vz 192.168.100.100 80 #Mediante o comando nc(netcat) comprobar que o porto 80 do servidor web Apache está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 80 é o porto TCP a escanear.
```

11. Lanzar na máquina virtual B (Kali) un navegador e visitar a IP 192.168.100.100 ou a URL <http://192.168.100.100>

## 12. Permisos apache:

```
root@debian:~# cp -pv /home/alumno/usuarios.csv /var/www/html/ #Copiar o ficheiro en
/var/www/html (DocumentRoot Apache) en modo verbose (detallado) e preservando permisos e datas.
root@debian:~# ls -l /home/alumno/usuarios.csv /var/www/html/usuarios.csv #Listar de forma
extendida os ficheiros /home/alumno/usuarios.csv e /var/www/html/usuarios.csv
root@debian:~# chown www-data. /var/www/html/usuarios.csv #Cambiar usuario propietario
www-data e grupo propietario www-data ao ficheiro usuarios.csv situado en /var/www/html
root@debian:~# chmod 444 /var/www/html/usuarios.csv #Cambiar a só lectura os permisos ugo do
ficheiro usuarios.csv situado en /var/www/html, é dicir, establecer os permisos r--r-- (soamente lectura para o
usuario propietario, o grupo propietario e o resto do mundo)
root@debian:~# /etc/init.d/apache2 status #Comprobar o estado do servidor web Apache.
root@debian:~# /etc/init.d/apache2 restart #Reiniciar o servidor web Apache.
root@debian:~# /etc/init.d/apache2 status #Comprobar o estado do servidor web Apache.
```

## 13. Lanzar na máquina virtual B (Kali) un navegador, visitar a IP 192.168.100.100 ou a URL <http://192.168.100.100/usuarios.csv> e descargar o ficheiro usuarios.csv en \$HOME do usuario kali, é dicir, descargar en /home/kali

## 14. Estudar o arquivo. Así:

### a. Listar o arquivo.

```
kali@kali:~$ cp -pv $HOME/usuarios.csv /tmp: Copiar o ficheiro en /tmp en modo verbose (detallado) e
preservando permisos e datas.
kali@kali:~$ cat -n /tmp/usuarios.csv #Listar numeradas todas as liñas do ficheiro /tmp/usuarios.csv
kali@kali:~$ cat -n /tmp/usuarios.csv | less #Listar numeradas todas as liñas do ficheiro /tmp/usuarios.csv.
Filtrar a saída anterior co comando less para poder subir, baixar, moverse a esquerda e dereita mediante as teclas frechas.
Saír do comando less premendo a tecla q
kali@kali:~$ less /tmp/usuarios.csv #Listar todas as liñas do ficheiro /tmp/usuarios.csv movéndose hacia abaixo,
arriba, dereita, esquerda mediante as frechas do teclado grazas ao comando less. Saír do comando less premendo a tecla q
```

### b. Listar as primeiras liñas.

```
kali@kali:~$ head /tmp/usuarios.csv #Listar as primeiras 10 liñas do ficheiro /tmp/usuarios.csv
kali@kali:~$ head -2 /tmp/usuarios.csv #Listar as primeiras 2 liñas do ficheiro /tmp/usuarios.csv
kali@kali:~$ head -n 2 /tmp/usuarios.csv #Listar as primeiras 2 liñas do ficheiro /tmp/usuarios.csv
```

### c. Listar as últimas liñas.

```
kali@kali:~$ tail /tmp/usuarios.csv #Listar as últimas 10 liñas do ficheiro /tmp/usuarios.csv
kali@kali:~$ tail -2 /tmp/usuarios.csv #Listar as últimas 2 liñas do ficheiro /tmp/usuarios.csv
kali@kali:~$ tail -n 2 /tmp/usuarios.csv #Listar as últimas 2 liñas do ficheiro /tmp/usuarios.csv
kali@kali:~$ tail -n +2 /tmp/usuarios.csv #Listar todas as liñas a partir da segunda liña do ficheiro
/tmp/usuarios.csv, é dicir, non listar a cabeceira do ficheiro, pero si o resto de liñas.
```

### d. Buscar no arquivo.

```
kali@kali:~$ less /tmp/usuarios.csv #Listar todas as liñas do ficheiro /tmp/usuarios.csv grazas ao comando less o
que nos permite baixar, subir, moverse á esquerda e dereita do contido do ficheiro, a través das frechas do teclado, así
como buscar o patrón que queiramos. Para buscar premer Shift+7, é dicir, a barra / e escribir o patrón a buscar.
```

Imos buscar varios patróns. Así:

- **::** → Buscamos o patrón dos caracteres punto e coma. Se existe aparece o patrón buscado. Pode ser que teñamos máis dunha concurrencia do patrón buscado, polo que se queremos ver a seguinte referencia atopada premer a tecla **n** e para ver a anterior referencia buscada premer a tecla **N**
- **^\$** → Buscamos o patrón liñas en branco (^ → simboliza comezo de liña, \$ → simboliza final de liña, como no medio non existe nada pois ^\$ simboliza liñas en branco). Se existe aparece o patrón buscado. Pode ser que teñamos máis dunha concurrencia do patrón buscado, polo que se queremos ver a seguinte referencia atopada premer a tecla **n** e para ver a anterior referencia buscada premer a tecla **N**
- **"** → Buscamos o patrón do carácter comiñas dobres. Se existe aparece o patrón buscado. Pode ser que teñamos máis dunha concurrencia do patrón buscado, polo que se queremos ver a seguinte referencia atopada premer a tecla **n** e para ver a anterior referencia buscada premer a tecla **N**
- **'** → Buscamos o patrón do carácter comiña simple. Se existe aparece o patrón buscado. Pode ser que teñamos máis dunha concurrencia do patrón buscado, polo que se queremos ver a seguinte referencia atopada premer a tecla **n** e para ver a anterior referencia buscada premer a tecla **N**

Para saír da pantalla capturada polo comando **less** premer a tecla **q**

```
kali@kali:~$ grep ';;' /tmp/usuarios.csv #Buscar o patrón de caracteres 2 puntos e coma no ficheiro
/tmp/usuarios.csv, é dicir, listar todas as liñas do ficheiro /tmp/usuarios.csv que posúen o patrón de búsqueda ';;'
```

```
kali@kali:~$ grep -v ';;' /tmp/usuarios.csv #Listar todas as liñas do ficheiro /tmp/usuarios.csv que non posúen o
patrón de búsqueda ';;'.
```

```
kali@kali:~$ cat -n /tmp/usuarios.csv | grep ';;' #Listar numeradas todas as liñas do ficheiro /tmp/usuarios.csv
que posúan o patrón buscado ';;'. Para iso primeiro amósase todas as liñas numeradas mediante o comando cat; logo filtrase
esa saída e búscase o patrón ';;' co comando grep. Deste xeito averiguamos o/s número/s de liña/s onde existe o patrón
buscado.
```



15. Depurar o arquivo. Así:

a. Eliminar liñas en branco.

**OPCIÓN 1**

```
kali@kali:~$ cat -n /tmp/usuarios.csv #Listar numeradas todas as liñas do ficheiro /tmp/usuarios.csv
kali@kali:~$ sed -i '/^$/d' /tmp/usuarios.csv #Eliminar as liñas en branco do ficheiro /tmp/usuarios.csv
```

**OPCIÓN 2**

```
kali@kali:~$ nl -b 'a' /tmp/usuarios.csv #Listar numeradas todas as liñas do ficheiro /tmp/usuarios.csv
kali@kali:~$ nl /tmp/usuarios.csv > /tmp/usuarios2.csv #Listar numeradas as liñas do ficheiro
/tmp/usuarios.csv, pero non lista as liñas en branco. A saída redireccionada non á consola senón a un novo ficheiro de nome
/tmp/usuarios2.csv que non posúe ningunha liña en branco.
kali@kali:~$ mv /tmp/usuarios2.csv /tmp/usuarios.csv #Mover o ficheiro /tmp/usuarios2.csv a
/tmp/usuarios.csv
```

b. Eliminar liñas sen datos de usuario.

**OPCIÓN 1**

```
kali@kali:~$ cat -n /tmp/usuarios.csv #Listar numeradas todas as liñas do ficheiro /tmp/usuarios.csv
kali@kali:~$ grep -v ';;;' /tmp/usuarios.csv > /tmp/usuarios2.csv #Listar todas as liñas do ficheiro
/tmp/usuarios.csv que non posúen o patrón de búsqueda ';;;'. A saída redireccionada non á consola senón a un novo ficheiro
de nome /tmp/usuarios2.csv que non posúe liñas sen datos de usuario.
kali@kali:~$ mv /tmp/usuarios2.csv /tmp/usuarios.csv #Mover o ficheiro /tmp/usuarios2.csv a
/tmp/usuarios.csv
```

**OPCIÓN 2**

```
kali@kali:~$ cat -n /tmp/usuarios.csv | grep ';;;' | awk '{print $1}' | sort -rn | xargs -I VAR sed -i
"VARd" /tmp/usuarios.csv #Listar numeradas todas as liñas do ficheiro /tmp/usuarios.csv que non posúan o patrón
buscado ';;;'. Para iso primeiro amósase todas as liñas numeradas mediante o comando cat; logo filtrase esa saída e
búscase o patrón ';;;' co comando grep; logo fíltrase e quedamos coa primeira columna con datos mediante o comando awk,
que son os números das liñas onde aparece o patrón anterior; logo filtrase ordeando de forma numérica e inversa mediante
o comando sort, para que aparezan as liñas numeradas de maior a menor; logo filtrase e bórrase esa liña mediante o
comando sed quedando o ficheiro /tmp/usuarios.csv sen as liñas que non posuían datos de usuario.
```

c. Eliminar caracteres comiñas dobres.

**OPCIÓN 1**

```
kali@kali:~$ cat -n /tmp/usuarios.csv #Listar numeradas todas as liñas do ficheiro /tmp/usuarios.csv
kali@kali:~$ sed -i 's|\"|g' /tmp/usuarios.csv #Eliminar todos os caracteres comiñas dobres do ficheiro
/tmp/usuarios.csv
```

**OPCIÓN 2**

```
kali@kali:~$ cat -n /tmp/usuarios.csv #Listar numeradas todas as liñas do ficheiro /tmp/usuarios.csv
kali@kali:~$ cat /tmp/usuarios.csv | tr -d '"' > /tmp/usuarios2.csv #Eliminar todos os caracteres comiñas
dobres do ficheiro /tmp/usuarios.csv, redireccionando a saída non á consola senón ao ficheiro /tmp/usuarios2.csv, o cal
créase se non existe e elimínase e créase se existe.
kali@kali:~$ mv /tmp/usuarios2.csv /tmp/usuarios.csv #Mover o ficheiro /tmp/usuarios2.csv a
/tmp/usuarios.csv
```

d. Substituír varios espazos por soamente un espazo.

**OPCIÓN 1**

```
kali@kali:~$ cat -n /tmp/usuarios.csv | tr -s ' ' > /tmp/usuarios2.csv #Listar numeradas todas as liñas
do ficheiro /tmp/usuarios.csv e substituír 2 espazos por 1, redireccionando a saída non á consola senón ao ficheiro
/tmp/usuarios2.csv, o cal créase se non existe e elimínase e créase se existe.
kali@kali:~$ mv /tmp/usuarios2.csv /tmp/usuarios.csv #Mover o ficheiro /tmp/usuarios2.csv a
/tmp/usuarios.csv
```

**OPCIÓN 2**

```
kali@kali:~$ tr -s ' ' < /tmp/usuarios.csv > /tmp/usuarios2.csv #Substituír no ficheiro /tmp/usuarios.csv
2 espazos por 1, redireccionando a saída non á consola senón ao ficheiro /tmp/usuarios2.csv, o cal créase se non existe e
elimínase e créase se existe.
kali@kali:~$ mv /tmp/usuarios2.csv /tmp/usuarios.csv #Mover o ficheiro /tmp/usuarios2.csv a
/tmp/usuarios.csv
```

e. Substituír caracteres ';' por ','

**OPCIÓN 1**

```
kali@kali:~$ tr -s ';' < /tmp/usuarios.csv > /tmp/usuarios2.csv #Substituír no ficheiro /tmp/usuarios.csv
todos os caracteres punto e coma seguidos por un espazo, por soamente o carácter punto e coma, redireccionando a saída
non á consola senón ao ficheiro /tmp/usuarios2.csv, o cal créase se non existe e elimínase e créase se existe.
kali@kali:~$ mv /tmp/usuarios2.csv /tmp/usuarios.csv #Mover o ficheiro /tmp/usuarios2.csv a
/tmp/usuarios.csv
```

**OPCIÓN 2**

```
kali@kali:~$ sed -i 's|;|,|g' /tmp/usuarios.csv #Substituír no ficheiro /tmp/usuarios.csv todos os caracteres
punto e coma seguidos por un espazo por soamente o carácter punto e coma
```

16. Ordear. Así:

a. Ordear por Provincia.

**OPCIÓN 1**

kali@kali:~\$ tail +2 /tmp/usuarios.csv | sort -t';' -k5 > /tmp/usuarios2.csv #Ordear a partir da liña 2 pola columna 5, tendo en conta que as columnas son aquelas onde o campo separador é o carácter punto e coma. A saída é redireccionada non á consola senón a un novo ficheiro de nome /tmp/usuarios2.csv

kali@kali:~\$ mv /tmp/usuarios2.csv /tmp/usuarios.csv #Mover o ficheiro /tmp/usuarios2.csv a /tmp/usuarios.csv

**OPCIÓN 2**

kali@kali:~\$ sort -t';' -k5 /tmp/usuarios.csv | grep -v Provincia > /tmp/usuarios2.csv #Ordear pola columna 5, tendo en conta que as columnas son aquelas onde o campo separador é o carácter punto e coma. A saída é redireccionada non á consola senón a un novo ficheiro de nome /tmp/usuarios2.csv que non posúe a cabeceira porque co comando grep evitamos que apareza a liña co patrón de carecteres 'Provincia'

kali@kali:~\$ mv /tmp/usuarios2.csv /tmp/usuarios.csv #Mover o ficheiro /tmp/usuarios2.csv a /tmp/usuarios.csv

b. Ordear por Provincia e o primeiro Apelido.

**OPCIÓN 1**

kali@kali:~\$ tail +2 /tmp/usuarios.csv | sort -t';' -k5,1 -k2,1 > /tmp/usuarios2.csv #Ordear a partir da liña 2 pola columna 5 e o seu primeiro carácter, e logo ordear pola columna 2 e o seu primeiro carácter, tendo en conta que as columnas son aquelas onde o campo separador é o carácter punto e coma. A saída é redireccionada non á consola senón a un novo ficheiro de nome /tmp/usuarios2.csv

kali@kali:~\$ mv /tmp/usuarios2.csv /tmp/usuarios.csv #Mover o ficheiro /tmp/usuarios2.csv a /tmp/usuarios.csv

**OPCIÓN 2**

kali@kali:~\$ sort -t';' -k5,1 -k2,1 /tmp/usuarios.csv | grep -v Provincia > /tmp/usuarios2.csv #Ordear pola columna 5 e o seu primeiro carácter, e logo ordear pola columna 2 e o seu primeiro carácter, tendo en conta que as columnas son aquelas onde o campo separador é o carácter punto e coma. A saída é redireccionada non á consola senón a un novo ficheiro de nome /tmp/usuarios2.csv que non posúe a cabeceira porque co comando grep evitamos que apareza a liña co patrón de carecteres 'Provincia'

kali@kali:~\$ mv /tmp/usuarios2.csv /tmp/usuarios.csv #Mover o ficheiro /tmp/usuarios2.csv a /tmp/usuarios.csv

c. Ordear por Provincia Pontevedra, primeiro Apelido, segundo Apelido e Nome.

**OPCIÓN 1**

kali@kali:~\$ tail +2 /tmp/usuarios.csv | grep -i Pontevedra | sort -t';' -k2,1 -k3,1 -k1,1 > /tmp/usuarios2.csv #Ordear a partir da liña 2 soamente onde se atope o texto pontevedra (sen ter en conta maiúsculas e minúsculas) pola columna 2 e o seu primeiro carácter, e logo ordear pola columna 3 e o seu primeiro carácter, e logo ordear pola columna 1 e o seu primeiro carácter, tendo en conta que as columnas son aquelas onde o campo separador é o carácter punto e coma. A saída é redireccionada non á consola senón a un novo ficheiro de nome /tmp/usuarios2.csv

kali@kali:~\$ mv /tmp/usuarios2.csv /tmp/usuarios.csv #Mover o ficheiro /tmp/usuarios2.csv a /tmp/usuarios.csv

**OPCIÓN 2**

kali@kali:~\$ sort -t';' -k2,1 -k3,1 -k1,1 /tmp/usuarios.csv | grep -v Provincia | grep -i pontevedra > /tmp/usuarios2.csv #Ordear pola columna 2 e o seu primeiro carácter, e logo ordear pola columna 3 e o seu primeiro carácter, e logo ordear pola columna 1 e o seu primeiro carácter, tendo en conta que as columnas son aquelas onde o campo separador é o carácter punto e coma. A saída é redireccionada non á consola senón a un novo ficheiro de nome /tmp/usuarios2.csv que non posúe a cabeceira porque co comando grep evitamos que apareza a liña co patrón de carecteres 'Provincia' e que soamente contén á provincia de pontevedra (sen ter en conta maiúsculas e minúsculas)

kali@kali:~\$ mv /tmp/usuarios2.csv /tmp/usuarios.csv #Mover o ficheiro /tmp/usuarios2.csv a /tmp/usuarios.csv

## 17. Crear variables e usuarios. Así:

- a. Crear soamente os usuarios coa consola bash tendo en conta nome, apelidos e contrasinais, de tal xeito que o nome de usuario será creado con caracteres minúsculas combinando o nome e os 3 primeiros caracteres dos apelidos:

```
while read LINE
do
NOME=$(echo ${LINE} | grep -i bash | cut -d ';' -f1 | tr '[:upper:]' '[:lower:]')
APELIDO1=$(echo ${LINE} | grep -i bash | cut -d ';' -f2 | cut -c1-3 | tr '[:upper:]' '[:lower:]')
APELIDO2=$(echo ${LINE} | grep -i bash | cut -d ';' -f3 | cut -c1-3 | tr '[:upper:]' '[:lower:]')
CONTRASINAL=$(echo ${LINE} | grep -i bash | cut -d ';' -f6)
useradd -m -d /home/${NOME} -s /bin/bash ${NOME}${APELIDO1}${APELIDO2}
echo -e "${CONTRASINAL}\n${CONTRASINAL}" | passwd ${NOME}${APELIDO1}${APELIDO2}
done</tmp/usuarios.csv
```

#Percorrer a partir da liña 2 cada liña do ficheiro /tmp/usuarios.csv. Por cada liña créanse variables que logo empréganse para xerar os usuarios coas súas contrasinais. Así, por cada liña créanse as variables:

- NOME → Filtrar a liña onde aparece bash independentemente de maiúsculas e minúsculas, quedarse coa columna 1 sendo o campo separador de columnas o carácter punto e coma, e transformar todos os caracteres en minúsculas.
- APELIDO1 → Filtrar a liña onde aparece bash independentemente de maiúsculas e minúsculas, quedarse coa columna 2 sendo o campo separador de columnas o carácter punto e coma, quedarse soamente cos 3 primeiros caracteres e transformar todos os caracteres en minúsculas.
- APELIDO2 → Filtrar a liña onde aparece bash independentemente de maiúsculas e minúsculas, quedarse coa columna 3 sendo o campo separador de columnas o carácter punto e coma, quedarse soamente cos 3 primeiros caracteres e transformar todos os caracteres en minúsculas.
- CONTRASINAL → Filtrar a liña onde aparece bash independentemente de maiúsculas e minúsculas, quedarse coa columna 6 sendo o campo separador de columnas o carácter punto e coma.  
Por cada liña coas variables anteriores créase un usuario co comando *useradd*, onde:
  - -d /home/\${NOME} → Xera a casa do usuario, é dicir, o directorio de traballo do usuario, no cartafol /home/{NOME}
  - -m → Copia na casa do usuario o que exista no cartafol /etc/skel
  - -s /bin/bash → Establece como shell de traballo para o usuario a shell bash
  - \${NOME}\${APELIDO1}\${APELIDO2} → Establece como nome de autenticación de usuario o nome e os 3 primeiros caracteres dos apelidosTamén por cada liña xérase o contrasinal do usuario anterior a través da variable \${CONTRASINAL}



## 18. Enlaces duros/simbólicos. Así:

### a. Crear/eliminar enlaces simbólicos.

```
kali@kali:~$ mkdir /tmp/enlacesSimb ; cd /tmp/enlacesSimb #Crear directorio /tmp/enlacesSimb e unha vez executado o comando mkdir independentemente que tivera éxito ou sexa errónea a execución, acceder ao directorio /tmp/enlacesSimb
kali@kali:~$ ln -s /tmp/usuarios.csv #Crear o enlace simbólico do ficheiro /tmp/usuarios.csv dentro de /tmp/enlacesSimb co nome usuarios.csv debido a que non escribimos explicitamente o nome do enlace.
kali@kali:~$ ln -s /tmp/usuarios.csv linksimbolic1 #Crear o enlace simbólico do ficheiro /tmp/usuarios.csv dentro de /tmp/enlacesSimb co nome linksimbolic1 debido a que escribimos explicitamente o nome do enlace.
kali@kali:~$ ls -l #Listar de forma extendida o contido do directorio onde estamos situados /tmp/enlacesSimb
kali@kali:~$ ls -li #Listar de forma extendida amosando os inodos do contido do directorio onde estamos situados /tmp/enlacesDuros. Podemos observar o número 3 logo da máscara de permisos, o cal determina que existen 3 ficheiros apuntando ao mesmo inodo.
kali@kali:~$ rm /tmp/enlacesSimb/linksimbolic1 #Eliminar o enlace simbólico linksimbolic1 do cartafol /tmp/enlacesSimb
kali@kali:~$ ls -l #Listar de forma extendida o contido do directorio onde estamos situados /tmp/enlacesSimb
kali@kali:~$ rm /tmp/usuarios.csv #Eliminar o ficheiro /tmp/usuarios.csv. Todos os enlaces simbólicos que existan deste ficheiro non se borran, e ao listalos de forma extendida pode que amosen en cores a perda de conexión do enlace simbólico co ficheiro ao que apuntan.
kali@kali:~$ ls -l #Listar de forma extendida o contido do directorio onde estamos situados /tmp/enlacesSimb
```

### b. Crear/eliminar enlaces duros.

```
kali@kali:~$ mkdir /tmp/enlacesDuros && cd /tmp/enlacesDuros #Crear directorio /tmp/enlacesDuros e unha vez executado o comando mkdir se soamente tivera éxito a execución, acceder ao directorio /tmp/enlacesDuros
kali@kali:~$ ln /tmp/usuarios.csv #Crear o enlace duro do ficheiro /tmp/usuarios.csv dentro de /tmp/enlacesDuros co nome usuarios.csv debido a que non escribimos explicitamente o nome do enlace.
kali@kali:~$ ln /tmp/usuarios.csv linkDuro1 #Crear o enlace duro do ficheiro /tmp/usuarios.csv dentro de /tmp/enlacesDuros co nome linkDuro1 debido a que escribimos explicitamente o nome do enlace.
kali@kali:~$ ls -l #Listar de forma extendida o contido do directorio onde estamos situados /tmp/enlacesDuros. Podemos observar o número 3 logo da máscara de permisos, o cal determina que existen 3 ficheiros apuntando ao mesmo inodo.
kali@kali:~$ ls -li #Listar de forma extendida amosando os inodos do contido do directorio onde estamos situados /tmp/enlacesDuros. Podemos observar o número 3 logo da máscara de permisos, o cal determina que existen 3 ficheiros apuntando ao mesmo inodo.
kali@kali:~$ ls -li . /tmp/usuarios.csv #Listar de forma extendida amosando os inodos do contido do directorio actual, onde estamos situados /tmp/enlacesDuros e o ficheiro /tmp/usuarios.csv. Podemos observar o número 3 logo da máscara de permisos, o cal determina que existen 3 ficheiros apuntando ao mesmo inodo.
kali@kali:~$ rm /tmp/enlacesDuros/usuarios.csv #Eliminar o enlace duro usuarios.csv do cartafol /tmp/enlacesDuros
kali@kali:~$ ls -li #Listar de forma extendida o contido do directorio onde estamos situados /tmp/enlacesDuros amosando os inodos. Podemos observar o número 2 logo da máscara de permisos, o cal determina que existen 2 ficheiros apuntando ao mesmo inodo.
kali@kali:~$ rm /tmp/usuarios.csv #Eliminar o ficheiro /tmp/usuarios.csv. Todos os enlaces duros que existan deste ficheiro non se borran, xa que non estamos a borrar o inodo senón o ficheiro que apunta ao inodo.
kali@kali:~$ ls -li #Listar de forma extendida o contido do directorio onde estamos situados /tmp/enlacesDuros amosando os inodos. Podemos observar o número 1 logo da máscara de permisos, o cal determina que existe soamente 1 ficheiro apuntando ao mesmo inodo.
```

19. Buscar. Así:

a. Buscar enlaces simbólicos.

```
kali@kali:~$ find /tmp/enlacesSimb -type l #Buscar dentro da ruta /tmp/enlacesSimb calquer enlace simbólico
```

b. Buscar enlaces duros.

```
kali@kali:~$ ls -li . /tmp/usuarios.csv #Listar de forma extendida amosando os inodos do contido do directorio actual, onde estamos situados /tmp/enlacesDuros e o ficheiro /tmp/usuarios.csv. Podemos observar o número 3 logo da máscara de permisos, o cal determina que existen 3 ficheiros apuntando ao mesmo inodo.
```

```
kali@kali:~$ find /tmp -inum $(ls -li /tmp/enlacesDuros/linkDuro1 | cut -d ' ' -f1) #Buscar dentro da ruta /tmp calquera cousa dentro do sistema de arquivos que posúa o número de inodo que escribimos logo da opción -inum. Este número foi averiguado tras realizar o comando ls -li /tmp/enlacesDuros/linkDuro1 | cut -d ' ' -f1, o cal lista de forma extendida e sobre esa saída quédase co valor da primeira columna, sendo o campo separador de columnas empregado o caracter espazo.
```

```
kali@kali:~$ find /tmp -links 3 #Buscar dentro da ruta /tmp calquera cousa dentro do sistema de arquivos que posúa 3 enlaces duros.
```

```
kali@kali:~$ find /tmp -links 1 #Buscar dentro da ruta /tmp calquera cousa dentro do sistema de arquivos que posúa 1 enlace duro.
```

c. Buscar ficheiros.

```
kali@kali:~$ find /tmp -type f -name usuarios.csv #Buscar dentro da ruta /tmp calquera ficheiro que posúa exactamente o nome usuarios.csv, é dicir, todo en minúsculas
```

```
kali@kali:~$ find /tmp -type f -iname usuarios.csv #Buscar dentro da ruta /tmp calquera ficheiro que posúa o nome usuarios.csv, independentemente das maiúsculas e minúsculas
```

d. Buscar dentro de ficheiros.

```
kali@kali:~$ find /tmp -type f -iname usuarios.csv -exec grep -Hi lugo {} \; #Buscar dentro da ruta /tmp calquera ficheiro que posúa o nome usuarios.csv, independentemente das maiúsculas e minúsculas, e dentro dos ficheiros atopados buscar o patrón de caracteres lugo, independentemente das maiúsculas e minúsculas (opción -i). Ademais, amosa antes do patrón buscado o nome do ficheiro (opción -H)
```

Ricardo Feijoo Costa



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)