

Expresiones regulares:

- * → Reemplazar todos los caracteres
- ? → Reemplazar un único caracter
- [] → Reemplazar por un rango numérico
- ^ → Inicio
- & → Se añade al final de un comando (ejecutable por ej.) para dejarlo corriendo en segundo plano

Comandos básicos:

```

adduser nombreUsuario
    # Añade un usuario con el nombre que le pongas

awk '{print $1,$NF}' data.txt # Muestra la 1ª y última columna de data.txt
    # Sintaxis general: patrón { acción }
        # Patrón → Cuando aplicar la acción
        # Procesa el texto línea a línea
        # Si el patrón es cierto, se aplica la acción para esa línea
        # Si no se proporciona patrón, se aplica la acción a todas las
líneas
        # Ejemplo: awk 'END { print NR }' data.txt → Cuenta el nº de
líneas de data.txt

badblocks # Detecta y excluye sectores inválidos del disco

bc:
numero_float=$(echo "$cadena" | bc -l) # Pasa un string a float. Si quisieramos
pasarlo a int:
numero_entero=$(expr "$cadena" + 0)

chrt --rr 20 ./miPrograma # Lanza un proceso con una prioridad de tiempo real
(PR=-21) → Calcular: PR = - 1 - prioridad_tiempo_real

cp -a # Para replicar contenido de disco a nivel de fichero (copia de seguridad)

crontab # Programar tareas en base a 5 parametros
    # -e Programar
    # -u Para que usuario
    # -l Listar tareas (ver contenido del crontab)

cut -d ' ' -f2 archivo
    # Corta las líneas del fichero en columnas
    # -d → En base a qué cortar las líneas en cada columna
    # -f → Que valores mostrar (qué columna)

dd if=/dev/urandom of=/tmp/archivo_aleatorio bs=1M count=2048 # Crear ficheros de
contenido X
    # if Tipo de contenido (random en este caso)
    # of Donde se crea
    # bs Tamaño del bloque de datos
    # count Cuántos bloques de datos

```

```
# 1M * 2048 = 2GB

du -sh /home    # Muestra el tamaño total del directorio /home

fsck # Detección y corrección (no siempre) de problemas de corrupción en el
sistema de ficheros

iostat -p /dev/sda # Estadísticas de uso y tasas de transferencia de los
dispositivos de almacenamiento
                # -p → Disco

last # Últimas conexiones al sistema

ln -s    # Crear enlace a directorio (-s simbólico)

ls -hrt
    # h → Tamaño de ficheros en KB/MB
    # r → Reverse order
    # t → Order chronologically

lsof # Lista archivos que están siendo utilizados por procesos en el sistema
    # -i          Información sobre conexiones de red abiertas
    # -p <PID>    Archivos abiertos por un proceso específico
    # -u <usuario>: Archivos abiertos por un usuario específico
    # -c <nombre>: Archivos abiertos por un proceso específico identificado
por su nombre
    # -t:         Números de PID en lugar de información detallada
    # -n:         Evita la resolución de nombres de host y puertos → salida
más rápida
    # -F <formato>: Especifica un formato de salida personalizado

nice -n 10 ./miScript    # Lanzar script o comando con -n prioridad

nohup ./miScriptLargo & # Deja runeando un proceso en segundo plano incluso
cerrando la SHELL

parted /dev/sdX          # Permite copiar, mover, cambiar sistemas de ficheros

passwd nombreUsuario    # Cambia la contraseña del usuario

pgrep stress-ng # Los PID del nombre del proceso si este está en ejecución

ps al    # More simplified list of running processes
ps aux   # Detailed list of all running processes
pstree   # Árbol de procesos

pwd      # Directorio actual

rsync -vazh /home /dev/sdc    # Realiza una copia de seguridad de HOME en
/dev/sdc
    # -a Mantiene usuarios
    # -z Comprime antes de copiar
    # -h Mostrar tasas de transferencia y tamaños (MB/s en vez de bytes/s)
```

```
sed 's/altro/alto/g' archivo.txt > archivo_modificado.txt # Cambia las palabras
altro por alto
    # -i → El fichero de entrada es sobrescrito (no hace falta el >)
    # Para eliminar todas las ocurrencias de una palabra: '/altro/d'
    # Mostrar las líneas 2, 3 de todos los ficheros: sed '2,3 p'

sort -t ":" -k3n /etc/passwd #ordena en base a:
    # dividida por :
    # k3    significa que se usará el tercer campo
    # n     significa que se realizará una ordenación numérica
    # -r    sería orden inverso

tar -czvf carpeta.tgz miCarpeta      # Comprimir
tar -xzvf carpeta.tgz                #Descomprimir

tcpdump -i ens4 -Z unai -w miCaptura # Comenzar a capturar tráfico y guardar en un
fichero
    # tcpdump -i <interfaz> -Z <usuario> -w <ficheroCaptura>

telnet IP puerto # Comprobar si un servicio remoto está a la escucha

useradd nombreUsuario
    # Añade usuario (faltan datos porque no los pone automatico. Ej.: password
(password))

userdel nombreUsuario    # Borrar usuario por nombre

wc -l    # Contar palabras (-l → Líneas en vez de palabras)
```

LABORATORIO 1

1

```
maitane@as2-maitane:~$ cal # Visualizar calendario (mes)

# Listar los 5 ficheros más nuevos de la carpeta /etc (que no sean carpetas):
maitane@as2-maitane:~$ ls -ltr /etc | grep -v '^d' | tail -n 5 # grep -v, --
invert-match    Invert the sense of matching, to select non-matching lines.

# Cambiar los permisos para que sólo el usuario propietario tenga capacidad de
leer y escribir:
maitane@as2-maitane:~/lab1$ chmod 700 calendario.txt
maitane@as2-maitane:~/lab1$ chmod u-x calendario.txt

# Cual es el nuestro Shell de inicio:
maitane@as2-maitane:~/lab1$ echo $SHELL

# Buscar todos los archivos que sean propiedad de tu usuario en el sistema y
listarlos en forma extendida (con toda la información):
maitane@as2-maitane:~/lab1$ sudo find / -user root -exec ls -l {} \;
    # -exec → Le indica a find que ejecute un comando en cada archivo o
```

```
directorio que coincida con los criterios de búsqueda.  
# ls -l → El comando que se ejecuta en cada archivo.  
# {} → Marcador que find reemplaza con la ruta completa de cada archivo  
antes de ejecutar el comando.  
# \; → Esta parte indica el final del comando -exec para que find sepa  
dónde termina
```

2

1. Crear un script llamado lsdirs.sh que muestre los directorios contenidos en el directorio actual.

```
#!/bin/bash  
for archivo in `ls`; do  
    echo $archivo;  
done
```

2. Crear un script llamado see.sh que reciba un nombre de fichero/directorio como parámetro. Si el nombre corresponde a un fichero, el script muestra el contenido del fichero con more, sino muestra el contenido del directorio con ls.

```
#!/bin/bash  
if [ -d $1 ]; then # Comprueba si es un directorio  
    ls $1  
elif [ -f $1 ]; then # Comprueba si es un fichero  
    more $1  
else  
    echo "prueba de nuevo"  
fi
```

3. Crear un script que pida al usuario que teclee una palabra y escriba por pantalla el número de caracteres de esa palabra.

```
#!/bin/bash  
echo "Mete una palabra:"  
read palabra  
echo "La palabra $palabra tiene una longitud de ${#palabra}" # Longitud
```

4. Crear un script que pida al usuario que teclee una palabra y compruebe si es un comando del sistema o no.

```
#!/bin/bash  
echo "Introduce una palabra:"  
read palabra
```

```
if command -v "$palabra" >/dev/null 2>&1 ; then    # Dice si la palabra es
un comando
    echo "$palabra es un comando del sistema."
else
    echo "$palabra NO es un comando del sistema."
fi
```

5. Crear un script que cree una carpeta llamada cosas y después cree 100 ficheros vacíos llamados fich.txt dentro de ella, donde es un número entre 0 y 99.

```
#!/bin/bash
if [ ! -d cosas ]; then # Si no existe el directorio "cosas"
    mkdir cosas
fi
cd cosas || exit # De ocurrir un error al abrir el directorio se sale del
script
for i in {0..99}; do
    touch "fich$i.txt"
done
```

6. Extender el script anterior para que cada fichero contenga la N-sima línea del manual de ls (man ls). El fichero fich0.txt tendrá la línea 0 del manual, fich1.txt tendrá la línea 1 del manual, ...

```
#!/bin/bash
if [ ! -d cosas2 ]; then
    mkdir cosas2
fi

i=0

cd cosas2 || exit # De ocurrir un error al abrir el directorio se sale del
script

man ls | while IFS= read -r linea; do # Leer cada línea del manual y
almacenarla en la variable linea
    # IFS= → evitar que se eliminen los espacios en blanco
    # iniciales y finales de cada línea
    # -r → evitar que se interpreten caracteres especiales de
    manera especial
    touch "fich$i.txt"
    echo $linea > "fich$i.txt"
    ((i++))
done
```

7. Crear un script que modifique la extensión de todos los ficheros .txt de un directorio a .t.

```
#!/bin/bash

for fichero in *.txt; do
    if [ -f "$fichero" ]; then
        nombre="${fichero%.txt}" # % Elimina la segunda parte
después del %
        mv "$fichero" "${nombre}.t"
    fi
done
```

8. Crear un script borra.sh que reciba un número indefinido de parámetros (de 0 a 9) y borre el fichero correspondiente a la suma del valor de los parámetros que reciba. Por ejemplo, borra.sh 1 4 5 9, borraría el fichero fich19.txt ($1 + 4 + 5 + 9 = 19$).

```
#!/bin/bash
suma=0
for i in "$@"; do # Itera todos los parámetros que le han pasando
    suma=$((suma+i))
done
rm "fich$suma.txt"
```

9. Crear un script orden.sh que muestre el contenido del fichero /etc/passwd ordenado por nombre de usuario, UID o GID. El script recibirá un parámetro que indique cuál de los 3 utilizar como clave de ordenación.

```
#!/bin/bash
echo "Qué clave de ordenación deseas usar: nombre de usuario (1), UID (2) o GID (3):"
read clave

case $clave in
    1)
        echo "Con usuario"
        sort -t ":" -k1 /etc/passwd
        ;;
    2)
        echo "Con UID"
        sort -t ":" -k3n /etc/passwd
        ;;
    3)
        echo "Con GID"
        sort -t ":" -k4n /etc/passwd
        ;;
    *)
        echo "Opción no válida, prueba de nuevo (elige 1, 2 o 3)"
        ;;
esac
```

10. Imagina que quieres mandar un e-mail con el mismo cuerpo a varios destinatarios, pero personalizando la primera línea con el nombre de cada uno. Crear un fichero cuerpo.txt con el texto del cuerpo del e-mail y la palabra NOMBRE en cada lugar donde querías poner el nombre, y un fichero nombres.txt con varios nombres personales. Crear un script que genere varios ficheros con el cuerpo del e-mail personalizado para cada destinatario.

```
#!/bin/bash
reemplazar="NOMBRE"
if [[ -f "nombres.txt" && -f "cuerpo.txt" ]]; then
    while read -r nombre; do # -r → evitar que el comando
read interprete barras invertidas como caracteres de escape.
        echo "PRIMER EMAIL PARA: $nombre"
        # Buscar el contenido de la variable $reemplazar en el archivo y
reemplazarlo por el contenido de la variable $nombre.
        # El g al final del patrón significa "global" y asegura que todas
las coincidencias en una línea se reemplacen, no solo la primera
        sed "s/$reemplazar/$nombre/g" "cuerpo.txt" > "carta_$nombre.txt"
    done < "nombres.txt"
fi
```

LABORATORIO 2

1

Gestión de particiones:

```
maitane@as2-maitane:~$ lsblk -e7 # Muestra las particiones y dónde están montadas
maitane@as2-maitane:~$ df -h # Muestra los sistemas de ficheros

maitane@as2-maitane:~$ sudo cfdisk /dev/sdb # Gestionar particiones

maitane@as2-maitane:~$ sudo mount -t ext4 /dev/sdb1 /mnt/disks # Montar partición
# -r → Montar solo en lectura
# -l → Listar
maitane@as2-maitane:~$ sudo umount /mnt/disks/disc2 # Desmontar partición
```

También se puede con:

```
fdisk /dev/sda
# Algunas opciones:
# p Ver tabla de particiones de disco
# n Nueva partición
# w Escribir nueva tabla de particiones
# q Salir
```

Para mantener montada una partición con reinicio de la máquina virtual:

```

maitane@as2-maitane:~$ sudo cp /etc/fstab /etc/fstab.backup

maitane@as2-maitane:~$ sudo blkid /dev/sdb1
/dev/sdb1: UUID="91cbda65-64be-4ca2-a4c6-68e82a6662ff" SEC_TYPE="ext2"
TYPE="ext3" PARTUUID="32fd36ac-7458-f24c-9fd9-069466eb3e18"

maitane@as2-maitane:~$ sudo nano /etc/fstab # EDITAR CONFIGURACION DE MONTAJE
maitane@as2-maitane:~$ cat /etc/fstab
LABEL=cloudimg-rootfs / ext4 defaults 0 1
LABEL=UEFI /boot/efi vfat umask=0077 0 1
UUID=91cbda65-64be-4ca2-a4c6-68e82a6662ff /mnt/disks/disc1 ext3
discard,defaults,nofail 0 2

```

Tamaños particiones:

```

maitane@as2-maitane:~$ sudo resize2fs -M /dev/sdb2 # Aumentar al máximo
maitane@as2-maitane:~$ cfdisk /dev/sdb # → TAMBIEN SE PUEDE HACER EL RESIZE AQUI

```

2: RENDIMIENTO

Fio:

```

# rwmixread= PORCENTAJE DE LECTURA
# size= TAMAÑO
# --filename= PATH HASTA FICHERO
sudo fio --randrepeat=1 --ioengine=libaio --direct=1 --gtod_reduce=1 --name=test -
-bs=4k --iodepth=64 --readwrite=randrw --rwmixread=75 --size=1G --
filename=/discoBalanceado/testFile

```

3:

LVM

Creación:

```

# Volúmenes físicos
maitane@as2-maitane:~$ sudo pvcreate -ff /dev/sdb1 /dev/sdb2 /dev/sdc1 # -ff
forzará la creación del etiquetado físico

# Grupo de volúmenes físicos
maitane@as2-maitane:~$ sudo vgcreate grupoVolúmenes /dev/sdb1 /dev/sdb2 /dev/sdc1

# Volumen lógico unificado
maitane@as2-maitane:~$ sudo lvcreate grupoVolúmenes -l 100%FREE -n volumenlab2

# Formatear volumen a sistema de ficheros:

```



```
maitane@as2-maitane:~$ sudo mkfs.ext4 /dev/grupoVolumenes/volumenlab2

# Montar volumen:
maitane@as2-maitane:~$ sudo mount -t ext4 /dev/grupoVolumenes/volumenlab2
```

Visualización:

```
maitane@as2-maitane:~$ sudo pvdisplay

maitane@as2-maitane:~$ sudo vgdisplay

maitane@as2-maitane:~$ sudo lvdisplay
```

Añadir volumen físico:

```
# Extender el grupo:
maitane@as2-maitane:~$ sudo vgextend grupoVolumenes /dev/sdc2

# Extender el volumen lógico
maitane@as2-maitane:~$ sudo lvextend -l +100%FREE /dev/grupoVolumenes/volumenlab2

# Para hacer que ocupe el total disponible del volumen:
maitane@as2-maitane:~$ sudo resize2fs -p /dev/mapper/grupoVolumenes-volumenlab2
```

Reducir tamaño de volúmenes:

```
# Reduce el volumen del grupo de volúmenes
maitane@as2-maitane:~$ sudo vgreduce grupoVolumenes /dev/sdc2

# Reducir el tamaño de un volumen lógico
maitane@as2-maitane:~$ sudo lvreduce -L 8G -t mi_volumen
# -L → Nuevo tamaño deseado del volumen lógico
# -t → Información detallada sobre el proceso de reducción.
```

Borrar volumen lógico:

```
maitane@as2-maitane:~$ sudo umount /labo2_lvm

maitane@as2-maitane:~$ sudo lvremove /dev/grupoVolumenes/volumenlab2
```

RAID

Crear:

```
# Crear RAID de nivel 5 con 3 particiones, llamado md0
maitane@as2-maitane:~$ sudo mdadm --create /dev/md0 -v --level=5 --raid-devices=3
/dev/sdb1 /dev/sdb2 /dev/sdc1

# Formatearlo:
maitane@as2-maitane:~$ sudo mkfs.ext4 /dev/md0

# Montarlo:
maitane@as2-maitane:~$ sudo mount -t ext4 /dev/md0 /lab2 RAID5
```

Visualizar:

```
maitane@as2-maitane:~$ sudo mdadm --detail /dev/md0
```

Monitorizar:

```
maitane@as2-maitane:~$ sudo mdadm --monitor [opciones] /dev/md0
# Ejemplo de [opciones]: --scan --daemonise --delay=1800
```

Generar fallo y solucionarlo:

```
# Generamos el fallo:
maitane@as2-maitane:~$ sudo mdadm /dev/md0 -f /dev/sdc1

# Para arreglarlo:
# 1. Borrarnos la partición que haya fallado de nuestro sistema RAID:
maitane@as2-maitane:~$ sudo mdadm /dev/md0 --remove /dev/sdc1

# 2. Volvemos a añadirla y que se recupere automáticamente el contenido perdido:
maitane@as2-maitane:~$ sudo mdadm --add /dev/md0 /dev/sdc1

# Monitorizar el proceso de reconstrucción:
maitane@as2-maitane:~$ cat /proc/mdstat
```

Desmontar y eliminar el dispositivo RAID:

```
# Desmontar:
maitane@as2-maitane:~$ sudo umount /lab2 RAID5

# Parar:
maitane@as2-maitane:~$ sudo mdadm --stop /dev/md0
```

```
# Limpia la información existente de un dispositivo RAID parado
maitane@as2-maitane:~$ mdadm --zero-superblock /dev/sdX

# Borrar carpeta si necesario
```

4

Configuración de rsnapshot:

```
maitane@as2-maitane:~$ sudo nano /etc/rsnapshot.conf

# Se le da permisos de escritura a otros procesos sobre la carpeta que añadamos en
rsnapshot para escritura:
maitane@as2-maitane:~$ sudo chmod o+w /backups

# Después de escribir:
maitane@as2-maitane:~$ rsnapshot configtest

# Para probar la configuración:
maitane@as2-maitane:~$ sudo rsnapshot hourly
```

Dentro del fichero entre descomentar y añadir filas debería aparecer algo como esto según la configuración:

```
snapshot_root          /backups/

...

cmd_du                  /usr/bin/du
cmd_rsnapshot_diff      /usr/bin/rsnapshot-diff
cmd_ssh                  /usr/bin/ssh

...

#####
# BACKUP LEVELS / INTERVALS #
# Must be unique and in ascending order #
# e.g. alpha, beta, gamma, etc. #
#####
retain hourly 24
retain daily 7
retain weekly 4

...

#####
### BACKUP POINTS / SCRIPTS ###
#####
# LOCALHOST
```

```
backup /home/ localhost/  
backup /etc/ localhost/  
backup /var/log localhost/
```

Para comprobar la diferencia entre una copia de seguridad y la siguiente:

```
maitane@as2-maitane:~$ sudo rsnapshot hourly  
  
# REALIZAR CAMBIOS VARIOS  
  
maitane@as2-maitane:~$ sudo rsnapshot hourly  
  
# Diferencias entre los dos ficheros creados:  
  
maitane@as2-maitane:~$ rsnapshot-diff -v /backups/hourly.0 /backups/hourly.1
```

LABORATORIO 3

1

Estresar al sistema:

```
maitane@as2-maitane:~$ stress-ng -c 1 -v --timeout 20s  
# -c          Número de CPUs utilizadas (nosotros tenemos dos: lscpu)  
# --timeout   Cuando tiene que parar el proceso
```

Ver los procesos y cuánto consumen:

```
maitane@as2-maitane:~$ top  
# Shift f para cambiar el orden prioritario
```

Para enseñar los parámetros PID, nice, priority y CMD de los procesos ordenados en base a la prioridad:

```
maitane@as2-maitane:~$ ps -eo pid,nice,priority,cmd --sort=priority  
PID  NI PRI CMD  
15   - -100 [migration/0]
```

Cambiar la prioridad de un proceso:

```
maitane@as2-maitane:~$ sudo renice -n 19 -p 1211 # Se calcula: PR = 20 + NI
```

Cambiar limite de uso de las CPU:

```
maitane@as2-maitane:~$ ulimit -a
maitane@as2-maitane:~$ ulimit -t 300
maitane@as2-maitane:~$ ulimit -a

# Limitar el uso de CPU para un proceso
maitane@as2-maitane:~$ cpulimit --pid 12345 --limit 50
```

NETCAT:

```
# Abrir conexión a la escucha del puerto 3000
maitane@as2-maitane:~$ nc -l -p 3000 > ArchivoParaRedirigirEntrada.txt
# -l    Modo escucha (listen)
# -p    Puerto
# Se puede hacer sin redirigir

# Para saber si esta escuchando
maitane@as2-maitane:~$ telnet localhost 3000

# Conexión al puerto en otra IP
maitane@as2-maitane:~$ dd if=/dev/urandom of=/home/user/random_data bs=1M
count=2048
maitane@as2-maitane:~$ nc 34.173.208.253 3000 < /home/user/random_data
# Se puede hacer sin mandar los datos así (escribiendo a mano tras la
conexión)

# Monitorear el uso de ancho de banda de procesos individuales:
maitane@as2-maitane:~$ sudo nethogs

# Mandar datos aleatorios a la IP
maitane@as2-maitane:~$ dd if=/dev/urandom | nc 34.171.168.81 3000
```

2

Logger:

```
maitane@as2-maitane:~$ sudo logger "Hola"
# -p    Priority type
```

Enviar todos los mensajes de nivel “debug” generados por el servicio sshd al fichero /var/log/ssh.log:

```
# Creamos el fichero que se nos pide
maitane@as2-maitane:~$ sudo touch /var/log/ssh.log

# Abirmos el fichero de configuración para indicar las acciones
maitane@as2-maitane:~$ sudo nano /etc/rsyslog.d/50-default.conf
```

Añadimos en el fichero:

```
auth.debug                                -/var/log/ssh.log
```

Cambiamos los permisos y los grupos para que coincidan con los de syslog (el fichero por defecto para escribir los log)

```
maitane@as2-maitane:~$ ls -l /var/log
maitane@as2-maitane:~$ sudo chown syslog:adm /var/log/ssh.log
```

Reiniciamos para que se apliquen cambios y CONFIGURAMOS EL MODO DEBUG:

```
# Reiniciamos el servicio rsyslog
maitane@as2-maitane:~$ sudo service rsyslog restart

# CONFIGURAMOS EL MODO DEBUG
# Editamos el archivo de configuración ssh
maitane@as2-maitane:~$ sudo vim /etc/ssh/sshd_config
# Buscamos la línea 'LogLevel INFO' y la sustituimos por 'LogLevel DEBUG'

# Reiniciamos el sistema sshd
maitane@as2-maitane:~$ sudo systemctl restart sshd

# Comprobamos que se esté escribiendo el fichero
maitane@as2-maitane:~$ cat /var/log/ssh.log
```

Configurar la rotación de logs (fichero /var/log/syslog):

Archivo /etc/rotate.d/rsyslog:

```
/var/log/syslog
{
    rotate 12
    monthly
    compress
    olddir /var/log/syslog.old
    createolddir 0640 syslog adm # En teoria no hace falta porque deberia
```

```
estar creado
}
```

FIRESTARTER:

Descargar el benchmark desde la web oficial: <https://tu-dresden.de/zih/firestarter>

Tiene un readme.

```
#!/bin/bash
max=0.0
valor=0.0

list=$(./FIRESTARTER --list-instruction-groups | grep -A1 "available instruction
groups for payload AVX512:" | awk 'NR==2')

IFS=',' read -ra instructions <<< "$list"

for instruction in "${instructions[@]}"; do

    svalor=$(./FIRESTARTER --run-instruction-groups $instruction:1 -r -t 30 |
grep "estimated floating point performance" | cut -d ' ' -f5)

    valor=$( echo "$svalor" | bc -l)
    resultado=$(echo "$valor > $max" | bc -l)

    if [ "$resultado" -eq 1 ]; then
        max="$valor"
    fi
done

echo "El valor máximo de GFLOPS es: $max"
```

LABORATORIO 4

MOSQUITTO

Instalación:

```
maitane@as2-maitane:~$ sudo apt install mosquitto mosquitto-clients
maitane@as2-maitane:~$ service mosquitto status
```

Comandos para Mosquitto:

```
# Publicar en un Broker información:
maitane@as2-maitane:~$ mosquitto_pub -h 34.173.208.253 -t "EUR/GBP" -m $valor
# -h      Host
# -t      Tema del foro
# -m      Mensaje a publicar
# -r      El último comentario la aparecera a los subscriptores no
conectados en el momento una vez se conecten

# Suscribirse a los mensajes del topic datosBroker y reenviarlos al fichero del
servidor NFS (ESTE ES EL BROKER)
maitane@as2-maitane:~$ sudo mosquitto_sub -h localhost -t "EUR/GBP" >
/tmp/carpetaRemota/HistoricoEurGbp.txt
# -h      Host
# -t      Tema del foro
```

Configuración Mosquitto:

```
sudo nano /etc/mosquitto/mosquitto.conf
## FICHERO /etc/mosquitto/mosquitto.conf ##
pid_file /var/run/mosquitto.pid
persistence true
persistence_location /var/lib/mosquitto/
log_dest file /var/log/mosquitto/mosquitto.log
include_dir /etc/mosquitto/conf.d

# Estas dos son las líneas que habría que añadir:
# Escuchando en localhost, puerto 1883
listener 1883 0.0.0.0
# Se permite que todo el mundo pueda entrar
allow_anonymous true
# También se pueden configurar usuarios
####
```

NFS

Instalación:

```
# Cliente:
maitane@as2-maitane:~$ sudo apt install nfs-common

# Servidor:
maitane@as2-maitane:~$ sudo apt install nfs-kernel-server nfs-common
```

Preparación y configuración de servicio NFS:


```
# Crear la carpeta donde se almacenaran los datos
maitane@as2-maitane:~$ sudo mkdir /datosNfs

# Editar el fichero de configuracion
maitane@as2-maitane:~$ sudo nano /etc/exports

## AÑADIR LAS SIGUIENTES LINEAS ##
# Le da permisos a una IP sobre una carpeta
# (esa carpeta también debe tener los permisos necesarios mediante chmod)
# rw    Read/Write
# ro    Read only
/datosNfs      34.173.208.253(rw)
/datosNfs      34.172.134.50(ro)
#####

# Comprobar que el archivo de configuracion es correcto
maitane@as2-maitane:~$ exportfs -ra

# Aplicar la configuracion de NFS
maitane@as2-maitane:~$ sudo exportfs -v

# Reiniciar el servicio
maitane@as2-maitane:~$ sudo service nfs-kernel-server restart

# Crear la carpeta para los datos del Broker
maitane@as2-maitane:~$ sudo touch datosBroker

# Permitir a otros usuarios la escritura en el archivo
maitane@as2-maitane:~$ sudo chmod o+w datosBroker
```

Ahora para que un servidor web recoja esos datos, solo tiene que montar la carpeta del servidor NFS donde le convenga (tiene que tener su IP con los permisos que hemos especificado en el servidor NFS):

```
maitane@as2-maitane:~$ sudo mount -t nfs 35.224.232.153:/datosNfs
/tmp/carpetaRemota
```