



---

**Universidad de Valladolid**

---

# Cuaderno de bitácora

---

**Ingeniería Informática  
Tecnologías de la Información**

Daniel Sanabria Salamanqués

24 de noviembre de 2025

---

# Índice

<b>1. Instalación</b>	<b>3</b>
1.1. Clúster de Máquinas Virtuales . . . . .	3
1.2. Configuración de instalación . . . . .	3
1.3. Reconocimiento del entorno . . . . .	3
1.4. Acceso remoto vía ssh . . . . .	4
1.5. Activar cuenta root . . . . .	5
1.6. Administración del disco . . . . .	5
1.6.1. Loop . . . . .	6
1.6.2. LVM . . . . .	7
1.7. Administración de almacenamiento . . . . .	9
1.7.1. RAID 5 . . . . .	9
1.8. Nueva instalación personalizada . . . . .	10
1.9. Trabajo No Presencial . . . . .	11
<b>2. Administración de usuarios y servicios</b>	<b>13</b>
2.1. Gestión de usuarios y grupos . . . . .	13
2.2. Gestión de password . . . . .	15
2.3. Gestión avanzada de grupos . . . . .	15
2.4. Perfiles de usuario . . . . .	16
2.5. Servicios del sistema . . . . .	19
2.6. Permisos de acceso . . . . .	20
2.7. Listas de acceso de control . . . . .	22
2.8. Cuota de disco . . . . .	23
2.9. Autoría /tmp . . . . .	24
2.10. Copia de seguridad . . . . .	24
2.10.1. dump . . . . .	24
2.10.2. tar . . . . .	25
2.10.3. rsync . . . . .	25
<b>3. Autenticación. Seguridad. Control de Servicios.</b>	<b>26</b>
3.1. PAM . . . . .	26
3.1.1. Claves fuertes . . . . .	26
3.1.2. Control de acceso por hora y por terminal . . . . .	27
3.2. Crontabd y atd . . . . .	27
3.2.1. Crontabd . . . . .	27
3.2.2. atd . . . . .	28
3.3. Seguimiento de la ejecución de servicios . . . . .	28
<b>4. Administración del Software. Herramientas de Desarrollo.</b>	<b>29</b>
4.1. Administración del Software . . . . .	29
4.1.1. Instalación de aplicaciones precompiladas . . . . .	29
4.1.2. Reinstalación curl desde su código fuente . . . . .	30
4.1.3. Actualización del sistema . . . . .	32
4.1.4. Nuevas versiones del núcleo Linux . . . . .	32
4.2. Herramientas de Desarrollo . . . . .	33
4.2.1. Shell script avanzados . . . . .	33
4.2.2. Creación de una aplicación con lenguaje de alto nivel . . . . .	34

---

<b>5. Administración avanzada de servicios</b>	<b>37</b>
5.1. Niveles de ejecución (Run-level) . . . . .	37
5.2. Creación de nuevos servicios . . . . .	38
5.3. Servidor Web . . . . .	39
5.4. XWindow . . . . .	41
<b>6. Monitorización</b>	<b>45</b>
6.1. El sistema de archivos virtual /proc . . . . .	45
6.2. Perfilado y traza de un proceso . . . . .	48
6.3. Monitorización . . . . .	50
6.3.1. SAR . . . . .	51

---

## 1. Instalación

### 1.1 Clúster de Máquinas Virtuales

Para comenzar con la instalación, me dirijo a la página [matrix.inf.uva.es](http://matrix.inf.uva.es) e inicio sesión con mi cuenta de laboratorio de la escuela. Una vez hecho, observo que en el **Datacenter** se encuentra mi máquina virtual **vm3803.virtual.lab.inf.uva.es**. Al hacer doble clic, compruebo en la sección de **Hardware** si está en el apartado CD/DVD la imagen de **Ubuntu Server**. Como no aparece, hago clic sobre ese apartado y con la opción **Edit** que aparece en la parte superior, agrego la imagen a ese disco de la máquina.

### 1.2 Configuración de instalación

Tras esto, voy a la sección **Console** para iniciar la máquina virtual y comenzar con la instalación de **Ubuntu Server**. Lo primero es seleccionar el idioma para el sistema; en mi caso, escijo inglés. Después, indico que no quiero realizar la actualización para obtener **Ubuntu Server 25.04**. Luego, para la configuración del teclado, selecciono el teclado español, debido a que mi teclado necesita esa configuración. En la siguiente pantalla, escijo que la instalación base será **Ubuntu Server** por defecto y sin opciones adicionales. En la configuración de red, no modifico ningún valor ni agrego ningún **proxy**. En cuanto al almacenamiento, indico que para la instalación use todo el disco y que no lo monte como un grupo **LVM**. Despues de confirmar la configuración del almacenamiento, relleno en la siguiente pantalla los datos de mi perfil:

- **Nombre:** Daniel
- **Nombre de servidor:** vm3803
- **Username:** dansana

Para la configuración de la conexión SSH, selecciono la opción de que se instale **OpenSSH**. Para terminar, no agrego ninguna **snap** al sistema y después de seleccionar **Done**, dejo que se termine la instalación con la configuración seleccionada. Tras unos minutos, la instalación termina y reinicio el sistema.

Una vez que ha arrancado, inicio sesión con el usuario y la contraseña que he creado y, acto seguido, procedo a purgar ciertas aplicaciones que no son necesarias.

### 1.3 Reconocimiento del entorno

Nos piden realizar un reconocimiento del entorno para conocer acerca del sistema que hemos instalado, además de saber cómo funciona la máquina virtual en la página [matrix.inf.uva.es](http://matrix.inf.uva.es):

- **Version Kernel Linux:** El comando `cat /proc/version`, nos devuelve la información acerca del Linux instalado. En este caso, se trata de un Linux con la versión el kernel 6.8.0-79-generic. El funcionamiento del comando es mostrar lo que contiene el archivo `version` dentro de `proc`, que se trata del sistema de ficheros. Otra opción, es con el comando `uname` que muestra información del sistema dependiendo de la opción que se le pase como argumento.
- **Particiones:** Con el comando `df -h`, se obtiene las particiones montadas. En este caso, tenemos las siguientes particiones:
  - `/dev/sda1`: Montada en el directorio `/boot/efi` y es la encargada de el arranque del sistema.
  - `/dev/sda2`: Montada en el directorio raíz `/`, dedicada al resto de sistema.
- **Espacio libre:** Con el mismo comando que el punto anterior, se puede ver que hay varias columnas dedicadas al almacenamiento de cada partición:
  - `/dev/sda1`: Con 1.1G en total, solo se ha usado el 1 %, es decir, 6.2M se ha utilizado y se encuentran disponibles 1.1G para usar.
  - `/dev/sda2`: Con 58G en total, solo se ha usado el 12 %, es decir, 6.5G se ha utilizado y se encuentran disponibles 49G para usar.

- **Cerrar sesión:** Cuando se ha iniciado sesión y queremos cerrar esa misma sesión, simplemente tenemos que escribir el comando `logout` y el sistema cerrará la sesión.
- **Apagar la máquina:** Desde la consola del sistema, mediante el comando `shutdown -h` se le enviará una señal al sistema para apagar la máquina, deteniendo todos los procesos y sincronizando los discos. Si queremos hacerlo inmediato, añadimos `now` al lado de `shutdown`.
- **Reiniciar la máquina:** Para el reinicio inmediato, se emplea el comando `reboot`, o, para un reinicio programado, se emplea `shutdown -r`.
- **Controles de la consola de la máquina virtual:** Se pide usar los controles que aparecen en la parte superior:
  1. Cuando la máquina esté encendida, nos indican apagar la máquina con `Stop`. Esto obligará a la máquina a hacer un apagado forzado.
  2. Despues de volver a encender, nos piden restear la máquina mediante la opción `Reset`. Funciona igual que escribir el comando `reboot`.
  3. Por ultimo, será apagar de nuevo la máquina pero con la opción `Shutdown` que le envía una señal `ACPI` al sistema operativo.

## 1.4 Acceso remoto vía ssh

Se nos indica que el sistema ya tiene instalado y activado el servicio de conexión segura `sshd` (que previamente hemos configurado en la configuración de la instalación) y para comprobar que funciona correctamente, me conectaré desde `Jair` a esta máquina, usando la red de la UVa. Aquí se muestra una captura del proceso:

```
ssh dansana@jair.lab.inf.uva.es
(dansana@jair.lab.inf.uva.es) Password:
dansana@jair:~ (0.05s)

          _/ \
         / \_ \_ \
        / / \ \ / \
       / / / / / \
      \ \ \ \ / / /
     Servidor de practicas de alumnos

Hello! This is jair.lab.inf.uva.es at 157.88.125.192

dansana@jair ~
ssh -p 38031 virtual.lab.inf.uva.es
dansana@virtual.lab.inf.uva.es's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-79-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Thu Sep 18 10:20:56 AM UTC 2025

 System load:  0.0          Processes:           145
 Usage of /:   11.2% of 57.72GB  Users logged in:   1
 Memory usage: 5%           IPv4 address for enp6s18: 10.0.38.3
 Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
 just raised the bar for easy, resilient and secure K8s cluster deployment.

 https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

149 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Mon Sep 15 16:01:45 2025 from 157.88.125.192
dansana@vm3803:~$
```

## 1.5 Activar cuenta root

Lo siguiente que se indica es activar la cuenta **root** cambiando su contraseña mediante **sudo passwd root** e indicando una clave para ese usuario y así poder acceder a la consola directamente como **root**, ya que por defecto no trae ninguna contraseña y puede ser una brecha de seguridad.

## 1.6 Administración del disco

Se pide obtener información sobre las particiones lógicas y física de nuestra máquina virtual, con ayuda de los comandos que se explican en las transparencias. Y para saber el sistema de ficheros que se está utilizando, tendremos que hacer un **cat** al fichero **/etc/fstab**, que contiene las informaciones que conciernen al montaje de las particiones que hay en el sistema.

```
dansana@vm3803:~$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda     8:0    0   60G  0 disk 
└─sda1   8:1    0    1G  0 part /boot/efi
└─sda2   8:2    0  58.9G 0 part /
sr0    11:0    1  2.6G 0 rom 
dansana@vm3803:~$
```

■ **Dispositivos:** Tal y como se muestra en la imagen, solo tenemos un dispositivo de almacenamiento **sda** con una capacidad de 60G. El otro dispositivo que existe es el CD de instalación de Ubuntu Server que ocupa 2.6G.

```
dansana@vm3803:~$ cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options>      <dump> <pass>
# / was on /dev/sda2 during curtin installation
/dev/disk/by-uuid/6de6d247-712b-4f7c-addb-2e028b04acee / ext4 defaults 0 1
# /boot/efi was on /dev/sda1 during curtin installation
/dev/disk/by-uuid/9996-869C /boot/efi vfat defaults 0 1
/swapp.img none swap sw 0 0
```

■ **Particiones:** Existen dos particiones en el disco **sda**:

- **sda1:** Con un tamaño de 1G y montado en el directorio **/boot/efi**, es la encargada de almacenar las herramientas de arranque del sistema que serán lanzadas por el firmware UEFI. Emplea el sistema de ficheros **vfat**.
- **sda2:** Partición principal, anclada en el directorio **/**, con el tamaño restante del disco para almacenar todas las aplicaciones y ficheros del sistema operativo y del usuario. Emplea el sistema de ficheros **ext4**.

Después, se nos exige investigar el fichero **/proc/filesystems** donde se ubican los sistemas de ficheros que es capaz de entender el sistema.

```
dansana@vm3003:~$ cat /proc/filesystems
nodev sysfs
nodev tmpfs
nodev bdev
nodev proc
nodev cgroup
nodev cgroup2
nodev cpuset
nodev devtmpfs
nodev configfs
nodev debugfs
nodev tracefs
nodev securityfs
nodev socksfs
nodev bpf
nodev pipefs
nodev ramfs
nodev hugetlbfs
nodev devpts
ext3
ext2
ext4
squashfs
vfat
nodev encryptfs
nodev fuseblk
nodev fuse
nodev fusectl
nodev efivarfs
nodev mqueue
nodev pstore
btrfs
nodev autofs
nodev binfmt_misc
```

- Se muestra dos columnas donde en la izquierda se indica si se requiere un dispositivo de bloque asociado al sistema de fichero que se muestra en la columna de la derecha.
- Por ejemplo, para los sistemas de ficheros **ext2**, **ext3** o **ext4** no se indica el valor **nodev**, por lo que es necesario usar un dispositivo físico para usar ese sistema de fichero. Pero, para **tmpfs** o **proc**, no es necesario tener un dispositivo físico.

### 1.6.1 Loop

Después, creamos un sistema de archivos o fichero dentro de un fichero nuevo:

1. Creamos el fichero mediante el comando **dd**, donde se le indica los siguientes parámetros:  
**dd if=/dev/zero of=fichero bs=1 count=4096**

- **if**: Desde que fichero o directorio se van a leer los datos. Como vamos a crear un fichero vacío, haremos uso de **/dev/zero** que se trata de un fichero especial desde el que se obtiene un flujo de cero, con el propósito de inicializar un fichero.
- **of**: Indicamos la ruta con el nombre del fichero creado.
- **bs**: Indicamos el tamaño del bloque que se quiere leer y escribir. Para este caso, se escoge de 1 MB por comodidad.
- **count**: El número de bloques que se van a crear. En este caso 4096M que corresponden a los 4G.

2. El siguiente paso es crear el dispositivo de bloque sobre el fichero que hemos creado con el que trabajaremos para crear el sistema de ficheros, mediante el comando **losetup**:

- Antes de crearlo, tenemos que ver los dispositivos **loop** que están disponibles para asociarlo con el fichero. Para ello, lanzamos el comando **losetup -f** y nos devuelve que el único dispositivo disponible es **/dev/loop0**.
  - Ahora lo único que tenemos que hacer es ejecutar este comando **sudo losetup /dev/loop0 fichero**. Es necesario usar permisos de administrador, por lo que se lanzará el comando con **sudo**.
3. Con el dispositivo de tipo bloque, le asignamos un sistema de fichero cualquiera con **mkfs**. En mi caso, le asigno el mismo que el que tiene la partición principal: **sudo mkfs.ext4 /dev/loop0**.
  4. Lo último es montar ese sistema de fichero nuevo en un directorio (**/mnt** debido a que está dedicado a montar dispositivos).
  5. Para comprobar que lo hemos montado correctamente, usamos el comando **lsblk** para ver todas las particiones montadas.

```
dansana@vm3803:/mnt$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0    7:0    0   4G  0 loop /mnt
sda     8:0    0   60G  0 disk 
└─sda1   8:1    0   1G  0 part /boot/efi
└─sda2   8:2    0 58.9G  0 part /
sr0     11:0   1   2.6G  0 rom
```

Cuando tengamos el dispositivo de disco Loop, nos piden administrar las particiones en ese dispositivo:

- Para crear una partición, haremos uso de la herramienta de **fdisk**. No tendrá ningún valor específico, por lo que se deja todo por defecto.
- Puede ser que el kernel no pueda actualizar automáticamente la tabla de particiones al terminar, por lo que habrá que desanclar y volver anclar el Loop.
- Al igual que hemos hecho con el dispositivo, habrá que formatear esa partición y asignarle un sistema de ficheros. En este caso el mismo que tiene el propio dispositivo.
- Después, se monta la partición con el comando **mount** sobre el directorio **/mnt** y se comprueba lanzando un **df -h**. Tras esto, se desmonta con **umount**.
- Ahora nos piden un resumen acerca de la función de los ficheros **/etc/fstab** y **/etc/mtab**:
  - **/etc/fstab**: Es un fichero de configuración estático que define qué sistemas de archivos hay en el sistema y cómo deben montarse. La máquina lo consulta durante el arranque para montar automáticamente discos, particiones o sistemas de ficheros de red.
  - **/etc/mtab**: Es un fichero dinámico, generado por el sistema, que refleja qué sistemas de ficheros están montados en este momento. En sistemas modernos, muchas veces **/etc/mtab** es un enlace simbólico a **/proc/self/mounts**, que cumple la misma función.
- Por último, nos indican eliminar la partición existente en **loop0** y crear varias particiones primarias y extendidas o lógicas. Además, cada partición tiene que tener un sistema de ficheros independiente. Este sería el esquema resultante:

```
Command (m for help): p
Disk /dev/loop0: 4 GiB, 4294967296 bytes, 8388608 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x657ff1b2

Device      Boot  Start Sectors  Size Id Type
/dev/loop0p1    2048 2099199 2097152  1G 83 Linux
/dev/loop0p2  2099200 4196351 2097152  1G 83 Linux
/dev/loop0p3  4196352 8388607 4192256  2G  5 Extended
/dev/loop0p5  4198400 6295551 2097152  1G 83 Linux
/dev/loop0p6  6297600 8388607 2091008 1021M 83 Linux
```

```
dansana@vm3803:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs          574M   1.2M  573M  1% /run
efivars         56K    27K  25K  52% /sys/firmware/efi/efivars
/dev/sda2       58G   7.1G   48G  13% /
tmpfs          2.9G    0  2.9G  0% /dev/shm
tmpfs          5.0M    0  5.0M  0% /run/lock
/dev/sda1       1.1G   6.2M   1.1G  1% /boot/efi
tmpfs          587M   8.0K  587M  1% /run/user/1000
/dev/loop0p1    974M   24K  987M  1% /mnt/p1
/dev/loop0p2    960M   51M  910M  6% /mnt/p2
/dev/loop0p5   1022M   4.0K 1022M  1% /mnt/p5
/dev/loop0p6    988M   44K  987M  1% /mnt/p6
dansana@vm3803:~$
```

## 1.6.2 LVM

Llegados a este punto, se requiere volver a destruir las particiones existentes para crear y administrar volúmenes lógicos (LVM).

1. Primero, modificamos la etiqueta de cada partición para marcar que es de tipo Linux LVM. Desde la herramienta de **fdisk**, seleccionamos cada partición del dispositivo **loop0** y con la opción **t**, escogemos el número 44 (en este caso que usamos una tabla de particiones de tipo GPT).
2. Despues, creamos el volumen físico en cada partición con **pvcreate**:

```

dansana@vm3803:~$ sudo pvcreate /dev/loop0p1
  Physical volume "/dev/loop0p1" successfully created.
dansana@vm3803:~$ sudo pvcreate /dev/loop0p2
  Physical volume "/dev/loop0p2" successfully created.
dansana@vm3803:~$ sudo pvcreate /dev/loop0p3
  Physical volume "/dev/loop0p3" successfully created.

```

3. Lo siguiente, es crear un grupo de volúmenes físicos, con **vgcreate**, donde agregamos los que hemos creado:

```

dansana@vm3803:~$ sudo vgcreate vg_practica /dev/loop0p1 /dev/loop0p2 /dev/loop0p3
  Volume group "vg_practica" successfully created
dansana@vm3803:~$ sudo vgdisplay
--- Volume group ---
  VG Name           vg_practica
  System ID
  Format            lvm2
  Metadata Areas   3
  Metadata Sequence No  1
  VG Access         read/write
  VG Status          resizable
  MAX LV
  Cur LV
  Open LV
  Max PV
  Cur PV
  Act PV
  VG Size           <3.99 GiB
  PE Size            4.00 MiB
  Total PE          1021
  Alloc PE / Size   0 / 0
  Free PE / Size    1021 / <3.99 GiB
  VG UUID
  a01Q83-5hWC-8UJY-dp01-Q4wb-P7Yw-Ktik5g

```

4. Por último, creamos un par de volúmenes lógicos, con **lvcreate**, sobre ese grupo que hemos creado en el punto anterior para después montarlo en el sistema:

```

dansana@vm3803:~$ sudo lvcreate -n lv_datos -L 500M vg_practica
  Logical volume "lv_datos" created.
dansana@vm3803:~$ sudo lvcreate -n lv_backup -L 300M vg_practica
  Logical volume "lv_backup" created.
dansana@vm3803:~$ ls /dev/vg_practica/
  lv_backup  lv_datos

```

5. Una vez ya tenemos los volúmenes lógicos, los formateamos ambos para asignarles una estructura de directorios y los montamos en el directorio **/mnt** para comprobar que se ha creado de forma correcta:

```

dansana@vm3803:~$ sudo mkdir -p /mnt/lv_datos
dansana@vm3803:~$ sudo mkdir -p /mnt/lv_backup
dansana@vm3803:~$ sudo mount /dev/vg_practica/lv_datos /mnt/lv_datos
dansana@vm3803:~$ sudo mount /dev/vg_practica/lv_backup /mnt/lv_backup
dansana@vm3803:~$ df -h
Filesystem              Size  Used Avail Use% Mounted on
tmpfs                  575M  1.2M  574M  1% /run
efivarsfs                56K  36K  16K  70% /sys/firmware/efi/efivars
/dev/sda2                 58G  7.2G  48G  13% /
tmpfs                   2.5G   0  2.5G  0% /dev/shm
tmpfs                   5.0M   0  5.0M  0% /run/lock
/dev/sda1                 1.1G  6.2M  1.1G  1% /boot/efi
tmpfs                   421M  8.0K  421M  1% /run/user/1000
/dev/mapper/vg_practica-lv_datos  452M  24K  417M  1% /mnt/lv_datos
/dev/mapper/vg_practica-lv_backup  236M  20M  217M  9% /mnt/lv_backup

```

6. Para acabar con este apartado, retornamos el sistema a su estado anterior desmontando y eliminando los volúmenes:

```

dansana@vm3803:~$ sudo lvremove /dev/vg_practica/lv_datos
Do you really want to remove and DISCARD logical volume vg_practica/lv_datos? [y/n]: y
  Logical volume "lv_datos" successfully removed.
dansana@vm3803:~$ sudo lvremove /dev/vg_practica/lv_backup
Do you really want to remove and DISCARD logical volume vg_practica/lv_backup? [y/n]: y
  Logical volume "lv_backup" successfully removed.
dansana@vm3803:~$ sudo vgremove vg_practica
  Volume group "vg_practica" successfully removed
dansana@vm3803:~$ sudo pvmremove /dev/loop0p1 /dev/loop0p2 /dev/loop0p3
  Labels on physical volume "/dev/loop0p1" successfully wiped.
  Labels on physical volume "/dev/loop0p2" successfully wiped.
  Labels on physical volume "/dev/loop0p3" successfully wiped.

```

---

## 1.7 Administración de almacenamiento

### 1.7.1 RAID 5

Lo último que vamos a hacer antes de realizar una nueva instalación es crear y administrar un RAID 5 mediante software:

1. Lo primero es crear 3 nuevos dispositivos loop de la misma manera que lo hemos hecho en el apartado Loop:

```
dansana@dansana:~$ sudo dd if=/dev/zero of=fich1.img bs=1M count=512
[sudo] password for dansana:
512+0 records in
512+0 records out
536870912 bytes (537 MB, 512 MiB) copied, 0,316135 s, 1,7 GB/s
dansana@dansana:~$ sudo dd if=/dev/zero of=fich2.img bs=1M count=512
512+0 records in
512+0 records out
536870912 bytes (537 MB, 512 MiB) copied, 0,335777 s, 1,6 GB/s
dansana@dansana:~$ sudo dd if=/dev/zero of=fich3.img bs=1M count=512
512+0 records in
512+0 records out
536870912 bytes (537 MB, 512 MiB) copied, 1,02255 s, 525 MB/s
dansana@dansana:~$ ls -l
total 1572876
-rw-r--r-- 1 root root 536870912 oct  8 08:11 fich1.img
-rw-r--r-- 1 root root 536870912 oct  8 08:11 fich2.img
-rw-r--r-- 1 root root 536870912 oct  8 08:11 fich3.img
```

Y después asociarlo a 3 dispositivos loop:

```
dansana@dansana:~$ sudo losetup /dev/loop1 fich1.img
dansana@dansana:~$ sudo losetup /dev/loop2 fich2.img
dansana@dansana:~$ sudo losetup /dev/loop3 fich3.img
dansana@dansana:~$ sudo losetup -a
/dev/loop1: [2055]:130827 (/home/dansana/fich1.img)
/dev/loop2: [2055]:130828 (/home/dansana/fich2.img)
/dev/loop3: [2055]:130829 (/home/dansana/fich3.img)
```

2. Consultando el manual para crear el dispositivo RAID, tenemos que seleccionar el modo **Create**, con las opciones de:

- **level**: Indicando el tipo de RAID.
- **raid-devices**: Número de dispositivos que usaremos.

```
dansana@vm3803:~$ sudo mdadm --create /dev/md0 --level=5 --raid-devices=3 /dev/loop1 /dev/loop2 /dev/loop3
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
dansana@vm3803:~$ cat /proc/mdstat
Personalities : [raid0] [raid1] [raid5] [raid4] [raid10]
md0 : active raid5 loop3[3] loop2[1] loop1[0]
      1044480 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]
unused devices: <none>
```

3. Ahora repetimos el mismo procedimiento que con el resto de dispositivos de almacenamientos para formatearlos y darle un sistema de ficheros y montarlo:

```

dansana@vm3803:~$ sudo mkfs.ext4 /dev/md0
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 261120 4k blocks and 65280 inodes
Filesystem UUID: 8a34336f-0feb-43d5-aa1e-4040d30a2160
Superblock backups stored on blocks:
            32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating Journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

dansana@vm3803:~$ sudo mkdir /mnt/raid5
dansana@vm3803:~$ sudo mount /dev/md0 /mnt/raid5
dansana@vm3803:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs          594M   1.2M  593M  1% /run
efivarfs        56K   40K   12K  78% /sys/firmware/efi/efivars
/dev/sda2       58G   8.6G   47G  16% /
tmpfs           2.9G     0   2.9G  0% /dev/shm
tmpfs           5.0M     0   5.0M  0% /run/lock
/dev/sda1       1.1G   6.2M   1.1G  1% /boot/efi
tmpfs          594M   8.0K  594M  1% /run/user/1000
/dev/md0        986M   24K  919M  1% /mnt/raid5

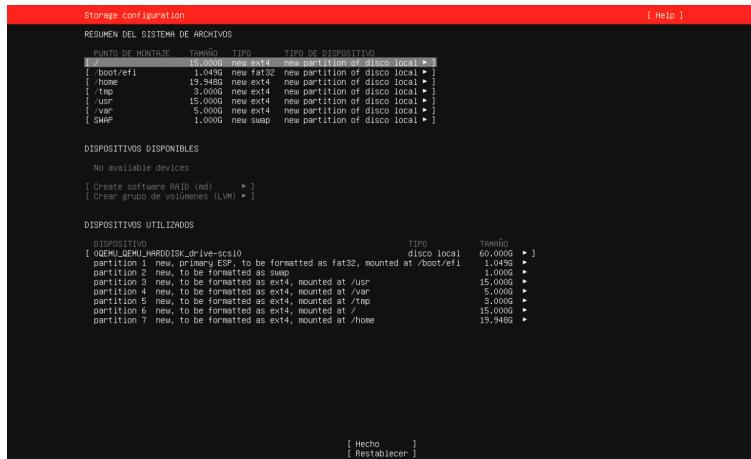
```

Para comprobar que hemos configurado correctamente el disco, ejecutamos el comando `echo \Voy a destruir el disco!!!!" > fichero.img` donde `fichero.img` es uno de los ficheros que da soporte al RAID para hacerlo fallar y ver que sigue funcionando. Al lanzarlo, veo que no se destruye el disco y sigue activo, debido a que el dispositivo RAID 5 está montado y funcionando. Entonces, lo que hay que hacer es reiniciar la máquina para que deje de estar en funcionamiento y lanzar el comando.

## 1.8 Nueva instalación personalizada

Hasta ahora hemos trabajado con una configuración "por defecto" sobre la administración del disco de la máquina, teniendo únicamente dos particiones: `/boot/efi` empleada para el arranque del sistema y `/` para el resto de archivos del equipo. En un entorno real, tenemos más particiones para minimizar posibles fallos y errores, por lo que vamos a realizar una nueva instalación con las siguientes particiones:

- `/boot/efi`: Contiene los ficheros para el arranque del sistema y tiene un espacio de `1.049G`, que es lo que ocupaba inicialmente y no se puede modificar. El formato que tiene es `fat32`, bastante antiguo y limitado, pero compatible con muchos dispositivos.
- `/`: Destinado a todos los ficheros para el sistema operativo y con un espacio de `15G` para que no haya problemas a la hora de agregar elementos al sistema.
- `swap`: Con un espacio de `1G`, solo actúa en caso de que la memoria RAM se quede sin espacio. No es necesario añadir nada más ya que el sistema ya cuenta con `6GB` de RAM.
- `/usr`: Donde se ubica los programas y librerías instaladas, por lo que es necesario `15G` de almacenamiento.
- `/var`: Lugar donde se ubican los ficheros como logs y colas. No es necesario dar mucho tamaño, por lo que se le asignan `3GB`.
- `/tmp`: Contiene archivos temporales y es necesario separarlo para que en caso de que se ocupe por completo, no bloquee el sistema. Solo es necesario `5G`.
- `/home`: Dedicada al espacio personal del usuario que ocupa el resto del espacio restante del disco.



## 1.9 Trabajo No Presencial

### ■ Administración de discos – particiones:

- Los discos duros o dispositivos de bloques, se dividen en unidades lógicas llamadas *particiones*.[1]
- Una partición sirve para organizar y almacenar el sistema operativo, las aplicaciones y los archivos personales. Existen diferentes esquemas para la distribución de particiones en un disco, como MBR o GPT.
- Cada partición se representa como un archivo en el sistema de archivos de Linux y se encuentra ubicada en el directorio `/dev`.

### ■ Sistemas de archivos:

- Es un elemento que controla cómo se almacenan y recuperan los datos. Sin un sistema de archivos, los datos colocados en un medio de almacenamiento serían un gran cuerpo de datos sin manera de saber dónde termina un dato y comienza el siguiente.
- Se organizan en una estructura jerárquica, de tipo árbol. El nivel más alto del sistema de ficheros es `/` o directorio raíz. Todos los demás ficheros y directorios están bajo el directorio raíz.[2]
- Por debajo del directorio raíz (`/`) hay un importante grupo de directorios común a la mayoría de las distribuciones de GNU/Linux: `/bin`, `/boot`, `/etc/`, `/opt`, etc.
- Tipos de sistemas de ficheros más utilizados en la actualidad:
  - EXT: Con sus versiones `ext2`, `ext3` y `ext4` (siendo esta la más usada en sistemas Linux), fue creado para sobreponer las limitaciones de MINIX y consiguió implementar VFS.
  - XFS: Manejo de grandes volúmenes de datos, por lo que es muy usado en servidores y sistemas empresariales.
  - NTFS: Sistema principal de Windows y con cierta compatibilidad con Linux.
  - FAT32: Antiguo y limitado, pero muy compatible con todos los sistemas.
  - exFAT: Evolución del anterior sistema y diseñado especialmente para memorias flash.

### ■ Actualización de un sistema operativo previamente instalado:

- En el caso de nuestra máquina virtual, estamos trabajando con Ubuntu que pertenece al grupo de distribuciones Debian, por lo que para actualizar el sistema operativo una vez instalado se hará uso de la herramienta `apt`.
- `apt` nos proporciona un sistema de gestión de paquetes donde maneja automáticamente las dependencias para la instalación de esos paquetes. Requiere de privilegios administrativos.[3]
- Para las actualizaciones será necesario usar los comandos `sudo apt update` y `sudo apt upgrade`.

---

- **Identificación discos duros y particiones:**

- En Linux, los dispositivos se representan dentro del directorio `/dev` y se identifican como dispositivos de bloques (`sda`, `sdb`, `sdc`, etc. o `nvme0n1`, `nvme0n2`, `nvme0n3`, etc.).
- Además, las particiones, tal y como se mencionaba en el primer apartado, son unidades lógicas de estos dispositivos y se identifican numerándose en orden seguido del nombre del dispositivo (`sda1`, `sda2`, `sda3`, etc. o `nvme0n1p1`, `nvme0n1p2`, `nvme0n1p3`, etc.).<sup>[1]</sup>
- También, cada partición puede tener un UUID único, que no cambia aunque el disco se conecte en distinto orden.

- **RAID:**

- RAID o Redundant Array of Independent Disks hace referencia a un sistema de almacenamiento de datos que utiliza múltiples discos duros, entre las cuales se distribuyen o replican los datos. <sup>[4]</sup>
- Estas son las principales configuraciones de RAID:
  - **RAID 0:** Distribuye los datos equitativamente entre dos o más discos sin información de paridad que proporcione redundancia. No tiene tolerancia a fallos, si falla un disco, lo pierdes todo.
  - **RAID 1:** Crea una copia exacta de un conjunto de datos en dos o más discos. Puede fallar solo un disco para no perder todos los datos.
  - **RAID 5:** División de datos a nivel de bloques que distribuye la información de paridad entre todos los discos miembros del conjunto. Esta variante de RAID ha logrado popularidad gracias a su bajo coste de redundancia. Puede tolerar 1 disco defectuoso; reconstrucción en curso mientras funciona.
  - **RAID 6:** amplía el nivel RAID 5 añadiendo otro bloque de paridad, por lo que divide los datos a nivel de bloques y distribuye los dos bloques de paridad entre todos los miembros del conjunto. Puede tolerar 2 discos defectuosos; más seguro que RAID 5 en entornos con discos grandes.

## 2. Administración de usuarios y servicios

### 2.1 Gestión de usuarios y grupos

Para comprobar si la clave de nuestro usuario y el resto de usuarios está encriptada, tenemos que ver si en el fichero `/etc/passwd`, dedicado a recopilar la información de un usuario, en el segundo campo aparece escrita la clave o se muestra en su lugar el carácter 'x'. Con solo lanzar un `cat` a ese fichero, descubrimos que cada usuario tiene su clave encriptada:

```
dansana@dansana:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/usr/sbin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin:/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin:/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin:/nologin
systemd-network:x:998:98:systemd Network Management:/:/usr/sbin:/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin:/nologin
dhcpcd:x:100:65534:DHCP Client Daemon,,,:/usr/lib/dhcpcd:/bin/false
messagebus:x:101:102:/nonexistent:/usr/sbin:/nologin
systemd-resolve:x:992:992:systemd Resolver:/:/usr/sbin:/nologin
pollinate:x:102:1::/var/cache/pollinate:/bin/false
polkitd:x:991:991:User for polkitd:/:/usr/sbin:/nologin
syslog:x:103:104::/nonexistent:/usr/sbin:/nologin
uuidd:x:104:105:/run/uuidd:/usr/sbin:/nologin
tcpdump:x:105:107:/nonexistent:/usr/sbin:/nologin
tss:x:106:108:TPM software stack,,,:/var/lib/tpm:/bin/false
landscape:x:107:109:/var/lib/landscape:/usr/sbin:/nologin
fwupd-refresh:x:989:989:Firmware update daemon:/var/lib/fwupd:/usr/sbin:/nologin
usbmux:x:108:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin:/nologin
sshd:x:109:65534::/run/sshd:/usr/sbin:/nologin
dansana:x:1000:1000:Daniel:/home/dansana:/bin/bash
```

Con esto concluimos que se está usando el fichero `/etc/shadow` para contener la clave encriptada.

Creamos dos nuevos grupos de usuario con `addgroup` y dos usuarios para cada grupo con `useradd`, para crearlos, y `adduser`, para agregarlo a uno de los grupos:

```
info: Selecting GID from range 1000 to 59999 ...
info: Adding group `grupo1' (GID 1001) ...
dansana@dansana:~$ sudo addgroup grupo2
info: Selecting GID from range 1000 to 59999 ...
info: Adding group `grupo2' (GID 1002) ...
```

```
dansana@dansana:~$ sudo adduser user1 grupo1
info: Adding user `user1' to group `grupo1' ...
dansana@dansana:~$ sudo adduser user2 grupo2
info: Adding user `user2' to group `grupo2' ...
```

Tras la creación de los nuevos usuarios y grupos, volvemos a ver el contenido del fichero `/etc/passwd` y también de `/etc/group`, que contiene información de cada grupo y los usuarios que hay dentro de él:

```
sshd:x:109:65534::/run/sshd:/usr/sbin:/nologin
dansana:x:1000:1000:Daniel:/home/dansana:/bin/bash
user1:x:1001:1003::/home/user1:/bin/sh
user2:x:1002:1004::/home/user2:/bin/sh
```

```
grupo1:x:1001:user1
grupo2:x:1002:user2
user1:x:1003:
user2:x:1004:
```

Y para el `password` de cada usuario, lo recomendable es seguir estos consejos:

- **Longitud de la contraseña:** Debería haber entre 12 y 16 caracteres.

- 
- **Uso de distintos tipos de caracteres:** Mezclar entre símbolos y letras mayúsculas y minúsculas, incluso números.
  - **No utilizar secuencias conocidas.**

La siguiente tarea es tener una estructura de directorios/ficheros tipo en el directorio `/etc/skel`, encargado de otorgar esa jerarquía al directorio `/home/<user>`. Decidí dar una estructura parecida a la que otorga **Ubuntu Desktop** en su instalación, usando `mkdir`:

- Downloads
- Documents
- Images
- Desktop

Al crear otro par de usuarios y agregarlos cada uno a los grupos nuevos, la estructura de ficheros de sus directorios `/home` tendrá esa jerarquía:

```
dansana@dansana:~$ sudo useradd -m user3
dansana@dansana:~$ sudo useradd -m user4
dansana@dansana:~$ sudo passwd user3
New password:
Retype new password:
passwd: password updated successfully
dansana@dansana:~$ sudo passwd user4
New password:
Retype new password:
passwd: password updated successfully
dansana@dansana:~$ sudo adduser user3 grupo1
info: Adding user `user3' to group `grupo1' ...
dansana@dansana:~$ sudo adduser user4 grupo2
info: Adding user `user4' to group `grupo2' ...
dansana@dansana:~$ ls /home/user3
ls: cannot open directory '/home/user3': Permission denied
dansana@dansana:~$ sudo ls /home/user3
Desktop Documents Downloads Images
dansana@dansana:~$ sudo ls -la /home/user3
total 36
drwxr-x--- 6 user3 user3 4096 oct  6 17:17 .
drwxr-xr-x  8 root  root  4096 oct  6 17:17 ..
-rw-r--r--  1 user3 user3  220 mar 31  2024 .bash_logout
-rw-r--r--  1 user3 user3 3771 mar 31  2024 .bashrc
drwxr-xr-x  2 user3 user3 4096 oct  6 17:12 Desktop
drwxr-xr-x  2 user3 user3 4096 oct  6 17:11 Documents
drwxr-xr-x  2 user3 user3 4096 oct  6 17:11 Downloads
drwxr-xr-x  2 user3 user3 4096 oct  6 17:12 Images
-rw-r--r--  1 user3 user3  807 mar 31  2024 .profile
dansana@dansana:~$ sudo ls -la /home/user4
total 36
drwxr-x--- 6 user4 user4 4096 oct  6 17:17 .
drwxr-xr-x  8 root  root  4096 oct  6 17:17 ..
-rw-r--r--  1 user4 user4  220 mar 31  2024 .bash_logout
-rw-r--r--  1 user4 user4 3771 mar 31  2024 .bashrc
drwxr-xr-x  2 user4 user4 4096 oct  6 17:12 Desktop
drwxr-xr-x  2 user4 user4 4096 oct  6 17:11 Documents
drwxr-xr-x  2 user4 user4 4096 oct  6 17:11 Downloads
drwxr-xr-x  2 user4 user4 4096 oct  6 17:12 Images
-rw-r--r--  1 user4 user4  807 mar 31  2024 .profile
```

## 2.2 Gestión de password

Se puede establecer una fecha de caducidad a la contraseña de un usuario y de su cuenta con el comando **chage** que establece dicho atributo. Hay que tener en cuenta que el sistema está 2 horas retrasado de la hora real (se puede comprobar con el comando **date**). Para la contraseña, hay que usar la opción **-M** para indicar el máximo de días que tiene esa contraseña antes de caducar, y la opción **-E** indica la fecha tope de ese usuario:

```
dansana@dansana:~$ sudo chage -M 1 user3
dansana@dansana:~$ sudo chage -E 2025-10-07 user3
dansana@dansana:~$ sudo chage -l user3
Last password change : oct 06, 2025
Password expires      : oct 07, 2025
Password inactive     : never
Account expires        : oct 07, 2025
Minimum number of days between password change : 0
Maximum number of days between password change : 1
Number of days of warning before password expires : 7
```

Para comprobar que funciona, podemos verlo en el fichero **/etc/shadow** ya que al tener la información de las claves de los usuarios, se indica también cuando es su fecha de expiración:

```
user3:$y$J9T$Y38Bi9bIm024zzVgGD6sc.$jToxpM.Ngr3izjHnBaq8P/NsyftZgeSSxiifbJpxM9B:20367:0:1:7::20368:
```

## 2.3 Gestión avanzada de grupos

Nos piden que uno de los nuevos usuarios creados forme parte de uno de los dos grupos. Compruebo viendo el fichero **/etc/group** que cada usuario está asignado a un grupo, por lo que únicamente habrá usar **usermod** con las opciones **-aG** para añadir a ese usuario a un grupo a mayores. En mi caso lo pruebo con el usuario **user4**.

```
grupo1:x:1001:user1,user3,user4
grupo2:x:1002:user2,user4
```

Una de las ventajas de tener un usuario en varios grupos puede darse a la hora de asignar permisos sobre un recurso compartido, ya que un solo usuario puede tener acceso y privilegios a recursos de ambos grupos.

```
dansana@dansana:~$ sudo chown :grupo1 ejemploGrupo1 -R
dansana@dansana:~$ sudo chown :grupo2 ejemploGrupo2 -R
dansana@dansana:~$ sudo chmod 770 ejemploGrupo1 ejemploGrupo2
dansana@dansana:~$ ls -la
total 1048644
drwxr-x--- 6 dansana dansana    4096 oct  8 17:43 .
drwxr-xr-x  8 root   root      4096 oct  6 17:17 ..
-rw-----  1 dansana dansana   1737 oct  8 08:24 .bash_history
-rw-r--r--  1 dansana dansana   220 mar 31 2024 .bash_logout
-rw-r--r--  1 dansana dansana  3771 mar 31 2024 .bashrc
drwx----- 2 dansana dansana   4096 oct  6 14:46 .cache
drwxrwx---  2 dansana grupo1  4096 oct  8 17:41 ejemploGrupo1
drwxrwx---  2 dansana grupo2  4096 oct  8 17:42 ejemploGrupo2
-rw-r--r--  1 root   root      29 oct  8 16:38 fich1.img
-rw-r--r--  1 root   root    536870912 oct  8 16:28 fich2.img
-rw-r--r--  1 root   root    536870912 oct  8 16:28 fich3.img
-rw-----  1 dansana dansana   20 oct  8 17:43 .lessht
-rw-r--r--  1 root   root      9 oct  8 08:41 nuevo
-rw-r--r--  1 dansana dansana   807 mar 31 2024 .profile
drwx----- 2 dansana dansana   4096 oct  6 14:42 .ssh
-rw-rw-r--  1 dansana dansana   40 oct  8 08:25 sudo
-rw-r--r--  1 dansana dansana    0 oct  6 16:10 .sudo_as_admin_successful
-rw-----  1 dansana dansana  1334 oct  8 17:42 .viminfo
```

---

A mayores, tenemos que asignarle una clave a un grupo, con `sudo gpasswd group`. En la captura podemos ver la clave encriptada del grupo1:

```
grupo1:$6$AWd6g00JSHSqE/FN$1MgXSChLQJrMda0A1FNauDzyQGTkQjRc44gwHT1kf8JCNdU4RKJyvrbIYGBdcf8VoNJa0aQ10oox777Bsqe1::user1,user3,user4
grupo2:!:user2,user4
```

Esto puede tener varias ventajas, entre ellas el acceso temporal a los recursos que puede tener ese grupo, de forma que el administrador no tiene que estar agregando o excluyendo usuarios del grupo.

## 2.4 Perfiles de usuario

En este apartado, se pide modificar la configuración del editor de texto `vi`, de forma que para uno de los usuarios creados (por ejemplo `user4`) se realicen dos operaciones automáticamente cuando se abra el programa, por lo que será necesario modificar el archivo con la configuración de este editor.

Como ya se ha comentado en la instalación, estamos usando `Ubuntu Server 24` en nuestras máquinas virtuales y, consultando la información en foros [5], en sistemas modernos, utiliza el mismo fichero de configuración que `vim` que es `.vimrc`, porque el comando `vi` es un enlace simbólico al editor `vim`.

```
dansana@dansana:~$ ls -l /usr/bin/vi
lrwxrwxrwx 1 root root 20 ago 27 2024 /usr/bin/vi -> /etc/alternatives/vi
dansana@dansana:~$ ls -l /etc/alternatives/vi
lrwxrwxrwx 1 root root 18 ago 27 2024 /etc/alternatives/vi -> /usr/bin/vim.basic
```

En `/home/user4`, no existe ese fichero de configuración, por lo que habrá que crearlo y agregar las opciones requeridas.

- Para la auto-sangría, tenemos que activar la opción `autoindent`.
- Y para un máximo de 75 caracteres con un salto de línea, la opción `textwidth=75 linebreak`.

```
dansana@dansana:~$ sudo cat /home/user4/.vimrc
set autoindent

set textwidth=75 linebreak

syntax on
set number
```

Las dos últimas líneas son configuración adicional que he añadido para agregar color al texto y que aparezcan los números de línea por pantalla.

```
1 Esto es un fichero de prueba con la configuración agregada al fichero
2 .vimrc
```

---

Finalizado esta primera personalización, ahora nos piden que cada vez que el usuario inicie sesión, se muestre el contenido de `$HOME/docs/Agenda.txt`. Haciendo uso de **Shell scripting**, modificamos el fichero `.profile` porque es el que se ejecuta cuando se inicia sesión en un usuario:

```
cat .profile
# ~/.profile: executed by the command interpreter for login shells.
# This file is not read by bash(1), if ~/.bash_profile or ~/.bash_login
# exists.
# see /usr/share/doc/bash/examples/startup-files for examples.
# the files are located in the bash-doc package.

# the default umask is set in /etc/profile; for setting the umask
# for ssh logins, install and configure the libpam-umask package.
#umask 022

# if running bash
if [ -n "$BASH_VERSION" ]; then
    # include .bashrc if it exists
    if [ -f "$HOME/.bashrc" ]; then
        . "$HOME/.bashrc"
    fi
fi

# set PATH so it includes user's private bin if it exists
if [ -d "$HOME/bin" ] ; then
    PATH="$HOME/bin:$PATH"
fi

# set PATH so it includes user's private bin if it exists
if [ -d "$HOME/.local/bin" ] ; then
    PATH="$HOME/.local/bin:$PATH"
fi

#Aparece el contenido de Agenda.txt
if [ -f "$HOME/Documents/Agenda.txt" ]; then
    cat "$HOME/Documents/Agenda.txt"
fi
```

Reiniciando, veremos lo que hay escrito en `Agenda.txt` ubicado en la última línea de todo el contenido que se muestra tras iniciar sesión.

```
Ubuntu 24.04.1 LTS dansana tty1
 dansana login: dansana
 Password:
 Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-85-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

 System information as of jue 09 oct 2025 16:42:50 UTC

 System load:  0.0          Processes:           162
 Usage of /home: 5.1% of 19.47GB   Users logged in:      0
 Memory usage:  5%          IPv4 address for enp6s18: 10.0.38.3
 Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
 just raised the bar for easy, resilient and secure K8s cluster deployment.

 https://ubuntu.com/engage/secure-kubernetes-at-the-edge

El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 129 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

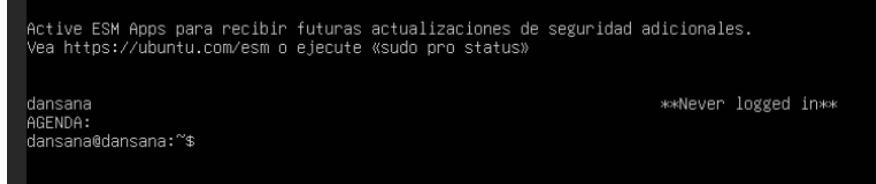
AGENDA:
dansana@dansana:~$ _
```

---

También, se requiere que cuando el usuario abre un **shell**, se muestre la fecha y hora del último login. En este caso, el fichero **.bashrc** es el que contiene la configuración de cualquier **shell**. Al final del fichero, agregamos **lastlog -u "\$USER"** con **tail -n 1**, donde **lastlog** es el comando que muestra el login más reciente de un usuario y **tail** es para mostrar solo la última línea del registro.

```
109 # this, if it's already enabled in /etc/bash.bashrc and /etc/profile
110 # sources /etc/bash.bashrc).
111 if ! shopt -oq posix; then
112   if [ -f /usr/share/bash-completion/bash_completion ]; then
113     . /usr/share/bash-completion/bash_completion
114   elif [ -f /etc/bash_completion ]; then
115     . /etc/bash_completion
116   fi
117 fi
118
119 # Muestra la fecha y hora del último login
120 lastlog -u "$USER" | tail -n 1
```

Al estar una máquina virtual, no estamos interactuando con una **tty** real”, por lo que nunca se registrará el inicio de sesión.



A screenshot of a terminal window. At the top, there is a message from ESM Apps: "Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales. Vea https://ubuntu.com/esm o ejecute «sudo pro status»". Below this, the terminal shows a login session for the user "dansana". The prompt "AGENDA:" is followed by "dansana@dansana:~\$". To the right of the prompt, the text "Never logged in\*\*" is displayed. The background of the terminal window is dark grey.

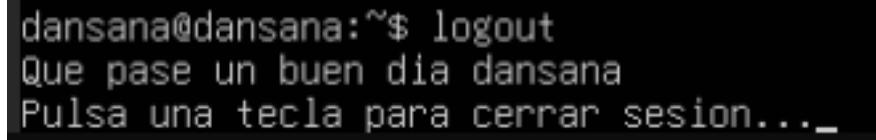
Y, finalmente, la última personalización es que cada vez que se cierre sesión, se muestre un mensaje de despedida y solo se cierre sesión tras haber respondido al mensaje. Al hacer **logout**, el sistema ejecuta el fichero **.bash\_logout** con todas las instrucciones que contiene. Por ello, modificamos dicho archivo para que mediante un **echo** muestre el mensaje y se quede esperando con un **read**:

```
dansana@dansana:~$ cat .bash_logout
# ~/.bash_logout: executed by bash(1) when login shell exits.

echo "Que pase un buen dia $USER"
read -p "Pulsa una tecla para cerrar sesion..."

# when leaving the console clear the screen to increase privacy

if [ "$SHLVL" = 1 ]; then
  [ -x /usr/bin/clear_console ] && /usr/bin/clear_console -q
fi
```



A screenshot of a terminal window showing the output of the **logout** command. The text "dansana@dansana:~\$ logout" is at the top. Below it, the message "Que pase un buen dia dansana" is displayed, followed by "Pulsa una tecla para cerrar sesion....". The background of the terminal window is dark grey.

## 2.5 Servicios del sistema

En este apartado, se trabajará acerca de los servicios que hay en el sistema, destacando el de conexión segura o `ssh`. Para comprobar los servicios activos, lanzamos el comando `systemctl list-units --type=service --state=running` donde señalamos que en el listado de servicios nos interesa aquellos que están activos:

```
dansana@dansana:~$ systemctl list-units --type=service --state=running
UNIT           LOAD  ACTIVE SUB   DESCRIPTION
cron.service    loaded active running Regular background program processing daemon
dbus.service     loaded active running D-Bus System Message Bus
getty@tty1.service loaded active running Getty on tty1
getty@tty6.service loaded active running Getty on tty6
ModemManager.service loaded active running Modem Manager
multipathd.service loaded active running Device-Mapper Multipath Device Controller
polkit.service   loaded active running Authorization Manager
rsyslog.service  loaded active running System Logging Service
systemd-journald.service loaded active running Journal Service
systemd-logind.service loaded active running User Login Management
systemd-networkd.service loaded active running Network Configuration
systemd-resolved.service loaded active running Network Name Resolution
systemd-timesyncd.service loaded active running Network Time Synchronization
systemd-udevd.service loaded active running Rule-based Manager for Device Events and Files
udisks2.service  loaded active running Disk Manager
unattended-upgrades.service loaded active running Unattended Upgrades Shutdown
upower.service   loaded active running Daemon for power management
user@1000.service loaded active running User Manager for UID 1000

Legend: LOAD + Reflects whether the unit definition was properly loaded.
        ACTIVE + The high-level unit activation state, i.e. generalization of SUB.
        SUB  + The low-level unit activation state, values depend on unit type.

18 loaded units listed.
```

Cada proceso nos indica el nombre (**UNIT**), si está cargado en el sistema (**LOAD**), si está activo (**ACTIVE**), su estado (**SUB**) y una descripción (**DESCRIPTION**). Observando la lista, vemos que no aparece el servicio `ssh` porque está inactivo:

```
dansana@dansana:~$ systemctl status ssh
  └─ ssh.service - OpenBSD Secure Shell server
      Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
      Active: inactive (dead)
        TriggeredBy: ● ssh.socket
          Docs: man:sshd(8)
                  man:sshd_config(5)
```

Vemos que está desactivado, por lo que habrá que activarlo con `sudo systemctl enable ssh`, después se para con el mismo comando pero usando `stop` en vez de `enable`, y se iniciará con `start`.

```
dansana@dansana:~$ systemctl stop ssh
==== AUTHENTICATING FOR org.freedesktop.systemd.manage-units ====
Authentication is required to stop 'ssh.service'.
Authenticating as: Daniel (dansana)
Password:
==== AUTHENTICATION COMPLETE ====
Stopping 'ssh.service', but its triggering units are still active:
ssh.socket
dansana@dansana:~$ systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
==== AUTHENTICATING FOR org.freedesktop.systemd.reload-daemon ====
Authentication is required to reload the systemd state.
Authenticating as: Daniel (dansana)
Password:
==== AUTHENTICATION COMPLETE ====
==== AUTHENTICATING FOR org.freedesktop.systemd.reload-daemon ====
Authentication is required to reload the systemd state.
Authenticating as: Daniel (dansana)
Password:
==== AUTHENTICATION COMPLETE ====
==== AUTHENTICATING FOR org.freedesktop.systemd.manage-unit-files ====
Authentication is required to manage system service or unit files.
Authenticating as: Daniel (dansana)
Password:
==== AUTHENTICATION COMPLETE ====
dansana@dansana:~$ systemctl start ssh
==== AUTHENTICATING FOR org.freedesktop.systemd.manage-units ====
Authentication is required to start 'ssh.service'.
Authenticating as: Daniel (dansana)
Password:
==== AUTHENTICATION COMPLETE ====

```

Es esencial tener un equipo con los servicios esenciales activos y desactivar aquellos que no tienen mucha relevancia, ya que consumen recursos innecesarios y pueden llegar a ser un riesgo en la seguridad. Investi-

---

gando el listado de servicios que hay activos en mi máquina, hay ciertos servicios que se pueden quitar como `ModemManager.service` debido a que solo se emplea para la gestión de módems 3G/4G.

Y para terminar con esta sección, nos piden que el sistema desde el arranque vaya al nivel de ejecución 3 (`runlevel3`), es decir sin entorno gráfico y solo en modo texto con red. Dependiendo de cada `runlevel` entre 1 y 5, Linux se ejecutará de una forma distinta. Para arrancar como ese nivel de ejecución `sudo systemctl set-default multi-user.target`

```
dansana@dansana:~$ sudo systemctl set-default multi-user.target
Created symlink /etc/systemd/system/default.target → /usr/lib/systemd/system/multi-user.target.
dansana@dansana:~$
```

## 2.6 Permisos de acceso

Vamos a aprender acerca de cómo funcionan los permisos en sistemas Linux [6]. En los sistemas operativos tipo POSIX cada elemento del sistema de archivos tiene la característica de poseer permisos que lo ubican dentro del mismo. Éstos sirven como uno más de los niveles de seguridad del sistema operativo al impedir que cualquier usuario pueda leer, escribir, ejecutar o acceder a dichos archivos y directorios de manera arbitraria. Estos permisos vistos de manera básica son: lectura (r, `read`), escritura (w, `write`) y ejecución (x, `execution`) y se agrupan en bloques (`rwx`) para 3 diferentes clases (usuario, grupo y otros).

Por defecto, cuando se crea un elemento en el sistema de fichero, se le asignará unos permisos dependiendo del objeto creado. `umask` es una función que establece los permisos predeterminados para los nuevos archivos y directorios creados en el sistema. El valor de la máscara de usuario, que se asigna ejecutando `umask`, corresponde a los bits contrarios del permiso predeterminado que se quiera asignar. Es decir, si por ejemplo se quiere asignar una máscara de usuario equivalente a 0775 (`rwxrwxr-x`), el valor de la máscara de usuario corresponderá a 0002 (el resultado de la operación 775 menos 775), que será lo mismo que definir `u=rwx,g=rwx,o=rx`. En este caso, para lograr que cuando se cree un fichero tenga los permisos 550, tenemos que saber que los permisos por defecto son 664 y realizar el siguiente cálculo:  $550 = 664 - \text{umask}$ . El resultado de `umask` es 114 en octal. Para aplicarlo a todos los usuarios, tenemos que agregar al final del fichero `/etc/profile` esta línea `umask 114`.

```
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).

if [ "$PS1-#" ]; then
    if [ "$BASH-#" ] && [ "$#BASH" != "/bin/sh" ]; then
        # The file bash.bashrc already sets the default PS1.
        # PS1='\$:\w\$'
        if [ -f /etc/bash.bashrc ]; then
            . /etc/bash.bashrc
        fi
    else
        if [ "$id -u" -eq 0 ]; then
            PS1='# '
        else
            PS1='\$ '
        fi
    fi
fi

if [ -d /etc/profile.d ]; then
    for i in /etc/profile.d/*.sh; do
        if [ -r $i ]; then
            . $i
        fi
    done
done
unset i
fi

umask 114
~

dansana@dansana:~$ ls -la
total 1048689
drwxr-x--- 10 dansana dansana 4096 oct 10 09:48 .
drwxr-xp-x 8 root root 4096 oct 6 17:17 ..
-rw----- 1 dansana dansana 4113 oct 9 19:29 .bash_history
-rw-r--r-- 1 dansana dansana 383 oct 9 17:20 .bash_logout
-rw-r--r-- 1 dansana dansana 3849 oct 9 17:13 .bashrc
drwx----- 2 dansana dansana 4096 oct 6 14:46 .cache
drwxrwxr-x 2 dansana dansana 4096 oct 9 16:33 Desktop
drwxrwxr-x 2 dansana dansana 4096 oct 9 16:34 Documents
drwxrwxr-x 2 dansana dansana 4096 oct 9 16:33 Downloads
drwxrwxr-- 2 dansana grupo1 4096 oct 8 17:41 ejemploGrupo1
drwxrwxr-- 2 dansana grupo2 4096 oct 8 17:42 ejemploGrupo2
-rw-r--r-- 1 root root 29 oct 8 16:38 fich1.img
-rw-r--r-- 1 root root 536876912 oct 8 16:28 fich2.img
-rw-r--r-- 1 root root 536876912 oct 8 16:28 fich3.img
drwxrwxr-x 2 dansana dansana 4096 oct 9 16:33 Images
-rw----- 1 dansana dansana 28 oct 10 09:42 lessht
-rw-r--r-- 1 root root 9 oct 8 09:41 nuevo
-rw-r--r-- 1 dansana dansana 926 oct 9 16:21 .profile
drwx----- 2 dansana dansana 4096 oct 6 14:42 .ssh
-rw-rw-r-- 1 dansana dansana 40 oct 8 09:25 sudo
-rw-rw-r-- 1 dansana dansana 0 oct 6 16:18 sudo_as_admin_successful
-rw-rw-r-- 1 dansana dansana 28 oct 10 09:48 umaskPrueba.txt
-rw----- 1 dansana dansana 8218 oct 10 09:48 .viminfo
-rw-r--r-- 1 root root 21 oct 9 16:46 .vimrc
```

---

Acerca del bit **SetUID**, es un permiso especial que hace que cuando se ha establecido ejecución, el proceso resultante asumirá la identidad del usuario dado en la clase de usuario (propietario del elemento). Corresponde al primer dígito del conjunto octal de permisos de los cuatro que hay. Un ejemplo para ver cómo funciona sería creando un programa en C que me indique cuál es el valor del UID efectivo (permisos con los que ejecuta el usuario) y real (permisos de ejecución reales de ese usuario).

```
1 #include <stdio.h>
2 #include <unistd.h>
3
4 int main() {
5     printf("Efectivo: %d\n", geteuid());
6     printf("Real: %d\n", getuid());
7     return 0;
8 }
```

Ahora, le otorgamos permisos a **root** para comprobar cómo varía el valor del UID:

```
dansana@dansana:~$ ./muestraUID
Efectivo: 1000
Real: 1000
dansana@dansana:~$ sudo chown root:root muestraUID
dansana@dansana:~$ sudo chmod u+s muestraUID
sudo: chmod: command not found
dansana@dansana:~$ sudo chmod u+s muestraUID
dansana@dansana:~$ ls -la
total 1048700
drwxr-x--- 10 dansana dansana    4096 oct 10 10:28 .
drwxr-xr-x  8 root    root      4096 oct  6 17:17 ..
-rw-----  1 dansana dansana   4113 oct  9 19:29 .bash_history
-rw-r--r--  1 dansana dansana   303 oct  9 17:20 .bash_logout
-rw-r--r--  1 dansana dansana  3849 oct  9 17:13 .bashrc
drwx----- 2 dansana dansana   4096 oct  6 14:46 .cache
drwxrwxr-x  2 dansana dansana   4096 oct  9 16:33 Desktop
drwxrwxr-x  2 dansana dansana   4096 oct  9 16:34 Documents
drwxrwxr-x  2 dansana dansana   4096 oct  9 16:33 Downloads
drwxrwx---  2 dansana grupo1   4096 oct  8 17:41 ejemploGrupo1
drwxrwx---  2 dansana grupo2   4096 oct  8 17:42 ejemploGrupo2
-rw-r--r--  1 root    root      29 oct  8 16:38 fich1.img
-rw-r--r--  1 root    root  536870912 oct  8 16:28 fich2.img
-rw-r--r--  1 root    root  536870912 oct  8 16:28 fich3.img
drwxrwxr-x  2 dansana dansana   4096 oct  9 16:33 Images
-rw-----  1 dansana dansana   20 oct 10 09:42 .lesshist
-rwsrwxr-x  1 root    root  16056 oct 10 10:26 muestraUID
-rw-rw-r--  1 dansana dansana   137 oct 10 10:26 muestraUID.c
-rw-r--r--  1 root    root    9 oct  8 08:41 nuevo
-rw-r--r--  1 dansana dansana   926 oct  9 16:21 .profile
drwx----- 2 dansana dansana   4096 oct  6 14:42 .ssh
-rw-rw-r--  1 dansana dansana   40 oct  8 08:25 sudo
-rw-r--r--  1 dansana dansana    0 oct  6 16:10 .sudo_as_admin_suc
-rwSrwr--  1 dansana dansana   28 oct 10 09:48 umaskPrueba.txt
-rw-----  1 dansana dansana  8228 oct 10 10:28 .viminfo
-rw-r--r--  1 root    root    21 oct  9 16:46 .vimrc
dansana@dansana:~$ ./muestraUID
Efectivo: 0
Real: 1000
```

---

## 2.7 Listas de acceso de control

Volvemos a crear dos usuarios nuevos y lo asignamos cada uno a un grupo nuevo distinto, de la misma manera que se ha hecho al comienzo de esta sección. Iniciando sesión en el **userA**, creamos el directorio con solo permisos de lectura y acceso, el cuál va a contener un fichero con solo permiso de lectura para el usuario **root**.

Ahora, mediante el uso de la utilidad ACL, damos permisos de lectura únicamente al usuario **userB** con el comando **sudo setfacl -m u:userB:r /home/userA/dirNuevo/fichUserA.txt** donde indicamos con la opción **-m** que se quiere modificar la ACL del fichero y señalamos el usuario que obtiene esos permisos:

```
dansana@dansana:~$ sudo ls -la /home/userA/
total 48
drwxr-x---  8 userA userA 4096 oct 10 16:42 .
drwxr-xr-x 10 root  root 4096 oct 10 10:39 ..
-rw-r--r--  1 userA userA 220 mar 31 2024 .bash_logout
-rw-r--r--  1 userA userA 3771 mar 31 2024 .bashrc
drwx----- 2 userA userA 4096 oct 10 10:44 .cache
drwxr-xr-x  2 userA userA 4096 oct  6 17:12 Desktop
dr-xr-xr-x  2 userA userA 4096 oct 10 16:42 dirNuevo
drwxr-xr-x  2 userA userA 4096 oct  6 17:11 Documents
drwxr-xr-x  2 userA userA 4096 oct  6 17:11 Downloads
drwxr-xr-x  2 userA userA 4096 oct  6 17:12 Images
-rw-r--r--  1 userA userA 807 mar 31 2024 .profile
-rw-----  1 userA userA 1239 oct 10 16:42 .viminfo
dansana@dansana:~$ sudo ls -la /home/userA/dirNuevo
total 12
dr-xr-xr-x  2 userA userA 4096 oct 10 16:42 .
drwxr-x---  8 userA userA 4096 oct 10 16:42 ..
-r-----  1 userA userA   11 oct 10 16:42 fichUserA.txt
```

```
dansana@dansana:~$ sudo setfacl -m u:userB:r /home/userA/dirNuevo/fichUserA.txt
dansana@dansana:~$ sudo ls -la /home/userA/dirNuevo
total 12
dr-xr-xr-x  2 userA userA 4096 oct 10 16:42 .
drwxr-x---  8 userA userA 4096 oct 10 16:42 ..
-r-----+ 1 userA userA   11 oct 10 16:42 fichUserA.txt
dansana@dansana:~$ sudo getfacl /home/userA/dirNuevo/fichUserA.txt
getfacl: Removing leading '/' from absolute path names
# file: home/userA/dirNuevo/fichUserA.txt
# owner: userA
# group: userA
user::r--
user:userB:r--
group::r--
mask::r--
other::r--
```

**NOTA:** La utilidad no viene instalada en la máquina, hay que instalarla con **sudo apt install acl**.

## 2.8 Cuota de disco

Se trata de un límite establecido por el administrador del sistema que restringe ciertos aspectos del uso del sistema de archivos en los sistemas operativos modernos. El objetivo de la utilización de las cuotas de disco es limitar la asignación de espacio en el disco duro de una manera razonable [7]. Para activarla, tenemos que seguir estos pasos[8]:

1. Instalamos la herramienta con el comando `sudo apt install quota`.
2. Una vez instalada, tenemos que editar el fichero `/etc/fstab`, para activar las opciones de cuota en la partición `/home`, tanto del usuario como para el grupo.

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
/dev/disk/by-uuid/b7924c12-88fc-4477-ab50-ebc9346eb3f3 none swap sw 0 0
# / was on /dev/sda5 during curtin installation
/dev/disk/by-uuid/bf4ae9ff-8ca6-4d69-ad1a-2eeb46ba55c8 / ext4 defaults 0 1
# /home was on /dev/sda7 during curtin installation
/dev/disk/by-uuid/fc0a346f-23a9-445c-ad91-a521c1067e4a /home ext4 defaults,usrquota,grpquota 0 1
# /boot/efi was on /dev/sda1 during curtin installation
/dev/disk/by-uuid/4024-Fa21 /boot/efi vfat defaults 0 1
# /usr was on /dev/sda3 during curtin installation
/dev/disk/by-uuid/2c65be95-ef46-49c3-a504-31ca6f9c8b6 /usr ext4 defaults 0 1
# /var was on /dev/sda4 during curtin installation
/dev/disk/by-uuid/4f9cb4f5-cf8c-4598-83ee-b56381f6c636 /var ext4 defaults 0 1
# /tmp was on /dev/sda5 during curtin installation
/dev/disk/by-uuid/67c8089c-7398-41ff-bc1d-8ed1e4d09fee /tmp ext4 defaults 0 1
~
~
```

3. Remontamos la partición con `mount -o remount /home` y reiniciamos los servicios demonios con `sudo systemctl daemon-reload`.
4. Creamos el fichero con el índice de cuota con `quotacheck -cum /home` y lo activamos con `quotaon -v /home`.

```
dansana@dansana:~$ sudo quotacheck -cum /home
dansana@dansana:~$ sudo quotaon -v /home
quotao: Your kernel probably supports ext4 quota feature but you are using external quota files. Please switch your filesystem to use ext4 quota feature as ext
ernal quota files on ext4 are deprecated.
quotao: cannot find /home/quota.group on /dev/sda7 [/home]
/dev/sda7 [/home]: user quotas turned on
```

Aparece un mensaje de aviso sobre el tipo de cuotas que puede ser ignorado ya que funciona sin problemas.

5. Revisamos la configuración de las cuotas de cada usuario con `edquota`, en el que aparece las columnas `soft limit` o límite blando que se puede sobrepasar temporalmente, y `hard limit` o límite duro que indica que no se puede sobrepasar ese límite:

```
GNU nano 7.2
Disk quotas for user dansana (uid 1000):
  Filesystem        blocks      soft      hard      inodes      soft      hard
    /dev/sda7           88        0        0       24        0        0
```

Figura 1: Cuota dansana

Un ejemplo donde se puede ver perfectamente su funcionamiento es modificando uno de los ficheros anteriores de los usuarios y estableciendo un límite de cuota tanto duro como blando, y ocupando el espacio disponible hasta ver el mensaje de `Disk quota exceeded`. Para el límite blando, se nos otorga un periodo de 7 días hasta bloquear la escritura en disco.

## 2.9 Autoría /tmp

La auditoría no proporciona seguridad adicional a su sistema; más bien, puede utilizarse para descubrir violaciones de las políticas de seguridad utilizadas en su sistema. Por ello, para realizar este tipo de operaciones, la herramienta `audit` que nos muestra esta información acerca de cualquier evento:

- Instalamos el servicio con `sudo apt install auditd` y lo arrancamos tal y como hemos visto en el apartado de **Servicios del sistema**.
- Después, creamos una nueva regla de auditoría para ese directorio con el comando `sudo auditctl -w /tmp -p rwx`, donde se indica la carpeta a auditar y las operaciones que se auditán. Comprobamos que se ha agregado correctamente con `sudo auditctl -l`.

```
dansana@dansana:~$ sudo systemctl start auditd
dansana@dansana:~$ sudo auditctl -w /tmp -p rwx
[sudo] password for dansana:
dansana@dansana:~$ sudo auditctl -l
-w /tmp -p rwx
```

- Por último, realizaremos una prueba para ver cuál es contenido del reporte tras crear un fichero en ese directorio y ver su contenido:

```
dansana@dansana:~$ echo "prueba de tmp" > /tmp/test.txt
dansana@dansana:~$ cat /tmp/test.txt
prueba de tmp
dansana@dansana:~$ sudo aureport -f
File Report
=====
# date time file syscall success exe auid event
=====
1. 10/10/25 19:59:23 /tmp/systemd-private-4ec7d0e4c3d843dc8dec8a17c14143-fuupd.service-4WTJNq 257 yes /usr/lib/systemd/systemd -1 536
2. 10/10/25 19:59:23 tmp 257 yes /usr/lib/systemd/systemd -1 537
3. 10/10/25 19:59:23 systemd-private-4ec7d0e4c3d843dc8dec8a17c14143-fuupd.service-4WTJNq 257 yes /usr/lib/systemd/systemd -1 538
4. 10/10/25 19:59:23 /proc/self/fd/5 257 yes /usr/lib/systemd/systemd -1 539
5. 10/10/25 19:59:23 /proc/self/fd/5 257 yes /usr/lib/systemd/systemd -1 540
6. 10/10/25 19:59:23 /tmp/systemd-private-4ec7d0e4c3d843dc8dec8a17c14143-fuupd.service-4WTJNq 257 yes /usr/lib/systemd/systemd -1 541
7. 10/10/25 19:59:23 tmp 257 yes /usr/lib/systemd/systemd -1 542
8. 10/10/25 19:59:23 / 263 yes /usr/lib/systemd/systemd -1 543
9. 10/10/25 19:59:23 /tmp/ 263 yes /usr/lib/systemd/systemd -1 544
10. 10/10/25 20:02:31 /tmp/test.txt 257 yes /usr/bin/bash 1006 559
11. 10/10/25 20:02:32 /tmp/test.txt 257 yes /usr/bin/cat 1006 560
12. 10/10/25 20:02:47 /tmp/test.txt 257 yes /usr/bin/cat 1006 561
```

El resultado se puede ver en la última línea, que aparece repetida debido a una comprobación anterior del reporte.

## 2.10 Copia de seguridad

### 2.10.1 dump

Consultando el manual, esta herramienta trabaja con sistemas de ficheros `ext2`, `ext3`, `ext4` examinando su contenido y determinando qué archivos son necesarios hacer `backup`. Se pueden almacenar tanto en el propio disco como en un medio externo. Después, para recuperar el `backup`, la herramienta que se debe utilizar es `restore` pasando como argumento el fichero `backup`.

Existen diferentes opciones a tener en cuenta para realizar la copia de seguridad:

- **-level**: Nivel de la copia, siendo 0 un copia completa del sistema de ficheros.
- **-f**: El fichero donde queremos tener la copia de seguridad.
- **-u**: Actualiza el historial de copias `/etc/dumpdates`.
- **-z**: Activación del modo compresión de la copia.

```
dansana@dansana:~$ sudo dump -0uz -f /mnt/backups/home.dump /home
DUMP: Date of this level 0 dump: Sat Oct 11 15:49:45 2025
DUMP: Dumping /dev/sda7 (/home) to /mnt/backups/home.dump
DUMP: Label: none
DUMP: Writing 10 Kilobyte records
DUMP: Compressing output at transformation level 2 (zlib)
DUMP: mapping (Pass I) [regular files]
DUMP: mapping (Pass II) [directories]
DUMP: estimated 1051319 blocks.
DUMP: Volume 1 started with block 1 at: Sat Oct 11 15:49:45 2025
DUMP: dumping (Pass III) [directories]
DUMP: dumping (Pass IV) [regular files]
DUMP: Closing /mnt/backups/home.dump
DUMP: Volume 1 completed at: Sat Oct 11 15:49:48 2025
DUMP: Volume 1 took 0:00:03
DUMP: Volume 1 transfer rate: 2594 kB/s
DUMP: Volume 1 1051270kB uncompressed, 7782kB compressed, 135.090:1
DUMP: 1051270 blocks (1026.63MB) on 1 volume(s)
DUMP: finished in 3 seconds, throughput 350423 kBBytes/sec
DUMP: Date of this level 0 dump: Sat Oct 11 15:49:45 2025
DUMP: Date this dump completed: Sat Oct 11 15:49:48 2025
DUMP: Average transfer rate: 2594 kB/s
DUMP: Wrote 1051270kB uncompressed, 7782kB compressed, 135.090:1
DUMP: DUMP IS DONE
```

### 2.10.2 tar

Conocida herramienta para empaquetar y comprimir archivos o directorios en un fichero. Añadir que con el mismo comando podemos también desempaquetar y restaurar los archivos en los discos. Con **tar** podemos preservar los metadatos de los archivos originales, permisos, propiedad o enlaces simbólicos. Esto es muy importante a la hora de realizar copias de seguridad o permitir restaurar los contenidos respetando sus propiedades originales.

Las **flags** más utilizadas son las siguientes:

- **-c**: Indica la creación de un archivo.
- **-v**: Muestra todo lo que se está empaquetando.
- **-f**: Especifica el nombre del archivo comprimido.
- **-x**: Indica que se quiere extraer.

```
dansana@dansana:~$ sudo tar -cvf /mnt/backups/home.tar /home
```

### 2.10.3 rsync

Es una herramienta de sincronización de archivos local y remota, de forma que podemos transferir ficheros de manera eficiente. Utiliza el algoritmo de transferencia delta que minimiza la transferencia de datos al copiar solo las secciones de un archivo que se han actualizado. Admite la copia de enlaces, dispositivos, propietarios, grupos y permisos.

Los parámetros más comunes son:

- **-t**: Indicamos que es de forma recursiva.
- **-a**: Para copiar recursivamente en un directorio, con el añadido de mantener privilegios, permisos y fechas de los ficheros y directorios.
- **-z**: Copiado de forma comprimida.
- **-t**: Preservamos el tiempo de modificación de los archivos.

```
dansana@dansana:~$ sudo rsync -a /home/ /mnt/backups/home_backup/
```

### 3. Autenticación. Seguridad. Control de Servicios.

#### 3.1 PAM

En Linux, la autenticación de los usuarios en el sistema está estandarizado mediante el uso de PAM. Proporciona un mecanismo para añadir autenticación a los programas mediante el uso de llamadas a las bibliotecas de funciones PAM.

El uso de diferentes módulos para cada servicio facilita que cada uno de ellos no tenga que implementar el mecanismo de acceso, sino simplemente pasar las credenciales a PAM y este se encargue de indicar si el usuario tiene acceso a no.

##### 3.1.1 Claves fuertes

El comando `passwd` es uno de los que sí usan PAM a la hora de crear las contraseñas, ya que en vez comunicarse con los ficheros `/etc/passwd` y `/etc/shadow`, hace una llamada al sistema PAM y recibe las instrucciones del fichero `/etc/pam.d/password`, donde realmente tiene su configuración en `/etc/pam.d/common-password`. Comprendido esto, modificaremos este fichero para obligar a los nuevos usuarios a que la nueva contraseña que asignen tenga estos criterios:

- Permitir que, en caso de fallo, sólo reintente 2 veces.
- Obligatoriedad de contener al menos 1 carácter numérico, 1 carácter mayúscula y 1 carácter minúscula.
- Que no se utilicen palabras claves ya usadas, o que estén en un diccionario.

Antes de comenzar, **IMPORTANTE** hacer copia del fichero `/etc/pam.d/password` porque si se cometen errores se puede bloquear los cambios de clave y habrá que formatear la máquina. También, es conveniente tener una sesión `root` abierta en caso de problemas.

Este sería el fichero `/etc/pam.d/common-password` actualizado para conseguir cumplir esos criterios:

```
# /etc/pam.d/common-password - password-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define the services to be
# used to change user passwords. The default is pam_unix.
#
# Explanation of pam_unix options:
# The "yescrypt" option enables
# hashed passwords using the yescrypt algorithm, introduced in Debian
# #11. Without this option, the default is Unix crypt. Prior releases
# used the option "sha512"; if a shadow password hash will be shared
# between Debian 11 and older releases replace "yescrypt" with "sha512"
# #for compatibility. The "obscure" option replaces the old
# # OBSCURE_CHECKS_ENAB' option in login.defs. See the pam_unix manpage
# #for other options.

# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules. See
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
password requisite pam_pwquality.so retry=2 minlen=12 ucredit=-1 lcredit=-1 dcredit=-1
password required pam_pwhistory.so remember=5 use_authtok
password [success=1 default=ignore] pam_unix.so obscure use_authtok try_first_pass yescrypt
# here's the fallback if no module succeeds
password requisite pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the module above will each just jump around
password requisite pam_permit.so
# and here are more per-package modules (the "Additional" block)
# end of pam-auth-update config
~
```

- En el primer módulo, `libpam-pwquality`, indicamos el número de reintentos que tiene el usuario al cambiar la clave y cómo tiene que ser esa clave.
- En el segundo módulo, `libpam-pwhistory`, conseguimos que se almacene las 5 últimas contraseñas.

**NOTA:** En el caso de perder o dañar dicho archivo y sí tener `backup`, habrá que reiniciar la máquina virtual e ir al CD de **Ubuntu Server** y desde el shell recuperar el fichero.

### 3.1.2 Control de acceso por hora y por terminal

Siguiendo con el servicio PAM, procedemos a cambiar el login de nuestra máquina para agregar las siguientes restricciones, a través del fichero `/etc/security/time.conf` que configura las reglas del módulo `pam_time` encargado de la autenticación de un usuario en el sistema:

- Un usuario cualquiera (por ejemplo `userA`) no pueda trabajar de 11:30h a 11:40h.
- Que ese mismo usuario no se pueda conectar a la consola.
- Y además, pueda acceder al resto de consolas que no sea la `tty1`.

En el fichero, cada línea tiene que tener esta estructura: `services;ttys;users;times`.

**IMPORTANTE:** Al igual que antes, hay que hacer copia de los ficheros que vayamos a cambiar en caso de fallo.

```
# Se bloquea el acceso al sistema del usuario A en el horario de
# 11:30 a 11:40 de la mañana
login;*;userA;!Al1130-1140
sshd;*;userA;!Al1130-1140

# No tiene permitido el usuario A acceder a la consola tty1
login;tty1;userA;!
```

## 3.2 Crontab y atd

### 3.2.1 Crontab

Se trata de un servicio demonio que permite programar tareas automáticas para que se ejecuten de forma periódica. Para asignar una nueva tarea, lanzamos el comando `crontab -e` donde indicamos primero cada cuanto se va a lanzar el script.

```
dansana@dansana:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
* * * * * /home/dansana/script.sh
```

---

### 3.2.2 atd

La orden **at** ejecuta un programa en un momento específico en el futuro. Toma el tiempo y la fecha deseados como parámetros de línea de comandos, y el comando a ejecutar en su entrada estándar. Ejecutará el programa como si hubiese ingresado en la consola.

```
dansana@dansana:~$ at now + 4 minutes
warning: commands will be executed using /bin/sh
at Sun Oct 12 12:12:00 2025
at> /home/dansana/script.sh
at> <EOT>
job 2 at Sun Oct 12 12:12:00 2025
dansana@dansana:~$ atq
2      Sun Oct 12 12:12:00 2025 a dansana
```

## 3.3 Seguimiento de la ejecución de servicios

Para ver los resultados el servicio demonio **crond** en el registro **rsyslogd**, hay que activar el registro y editar su configuración añadiendo la regla de que los **logs** del servicio aparezcan en el registro. La línea que hay que añadir es **cron.\* /var/log/cron.log** y reiniciar tanto el servicio **cron** y el **rsyslog**.

---

## 4. Administración del Software. Herramientas de Desarrollo.

### 4.1 Administración del Software

#### 4.1.1 Instalación de aplicaciones precompiladas

Nos piden instalar la herramienta `apt-rdepends` (con `sudo apt install apt-rdepends`) para obtener las dependencias de las aplicaciones que necesitemos utilizar, como es el caso del compilador `gcc`, para programas escritos en C. Mirando el manual, el comando para ver las dependencias del compilador es: `apt-depends gcc`. Como la salida recibida aparece demasiada información, lo que haremos es redirigir la salida a un fichero, desde el que podremos ver todos los paquetes se necesitan:

```
1  gcc
2   Depends: cpp (>= 4:13.2.0-7ubuntu1)
3   Depends: cpp-x86-64-linux-gnu (= 4:13.2.0-7ubuntu1)
4   Depends: gcc-13 (>= 13.2.0-11~)
5   Depends: gcc-x86-64-linux-gnu (>= 4:13.2.0-7ubuntu1)
6  cpp
7   Depends: cpp-13 (>= 13.2.0-11~)
8   Depends: cpp-x86-64-linux-gnu (= 4:13.2.0-7ubuntu1)
9  cpp-13
10  Depends: cpp-13-x86-64-linux-gnu (= 13.3.0-6ubuntu2~24.04)
11  Depends: gcc-13-base (= 13.3.0-6ubuntu2~24.04)
12  cpp-13-x86-64-linux-gnu
13  Depends: gcc-13-base (= 13.3.0-6ubuntu2~24.04)
14  Depends: libc6 (>= 2.38)
15  Depends: libgmp10 (>= 2:6.3.0+dfsg)
16  Depends: libis123 (>= 0.15)
17  Depends: libmc3 (>= 1.1.0)
18  Depends: libmofr6 (>= 3.1.3)
19  Depends: libzstd1 (>= 1.5.5)
20  Depends: zlib1g (>= 1:1.1.4)
21  gcc-13-base
22  libc6
23  Depends: libgcc-s1
24  libgcc-s1
25  Depends: gcc-14-base (= 14.2.0-4ubuntu2~24.04)
26  Depends: libc6 (>= 2.35)
27  gcc-14-base
28  libgmp10
29  Depends: libc6 (>= 2.38)
30  libis123
31  Depends: libc6 (>= 2.14)
32  Depends: libgmp10 (>= 2:6.3.0+dfsg)
33  libmpc3
34  Depends: libc6 (>= 2.4)
35  Depends: libgmp10 (>= 2:6.3.0+dfsg)
36  Depends: libmpfr6 (>= 4.0.0)
37  libmpfr6
38  Depends: libc6 (>= 2.14)
39  Depends: libgmp10 (>= 2:6.3.0+dfsg)
40  libzstd1
41  Depends: libc6 (>= 2.34)
42  zlib1g
43  Depends: libc6 (>= 2.14)
44  cpp-x86-64-linux-gnu
45  Depends: cpp-13-x86-64-linux-gnu (>= 13.2.0-11~)
46  gcc-13
47  Depends: binutils (>= 2.39)
48  Depends: cpp-13 (= 13.3.0-6ubuntu2~24.04)
49  Depends: gcc-13-base (= 13.3.0-6ubuntu2~24.04)
```

- En las primeras filas, se nos muestra las dependencias de la aplicación indicada, `gcc`, con su versión más reciente.
- Y para el resto de filas, las dependencias de esos paquetes y, a su vez, las dependencias de esas dependencias.

Otra forma rápida de comprobar si tenemos los paquetes necesarios para instalar cualquier aplicación, es realización directamente la instalación de esa aplicación primero, ya que `apt` se encarga de comprobar y gestionar las dependencias. Si falta alguna de ellas, no nos dejará instalarlo y se nos mostrará la(s) dependencia(s) que faltan por instalar.

Después de haberlo instalado correctamente con `sudo apt install gcc`, descargamos `git`, gestor de versiones, siguiendo la documentación oficial de la herramienta [9] y comprobando sus dependencias.

#### 4.1.2 Reinstalación curl desde su código fuente

Cuando hemos instalado en el bloque anterior **Ubuntu Server**, se instaló junto a él **curl**, empleado para transferir información desde o hacia un servidor usando una URL. Como vamos a aprender a instalar aplicaciones desde su código fuente, lo que haremos primero es desinstalarla con **apt-get**:

- Antes de eliminar cualquier herramienta, hay que asegurarse si vamos a eliminarla pero mantiene los ficheros de configuración (**remove**), o por el contrario, queremos una eliminación completa (**purge**). Como vamos a reinstalarlo en el sistema, habría que usar la primera opción:

```
dansana@dansana:~$ sudo apt-get remove curl
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  libcurl4t64
Utilice «sudo apt autoremove» para eliminarlo.
Los siguientes paquetes se ELIMINARÁN:
  curl pollinate ubuntu-server ubuntu-server-minimal
0 actualizados, 0 nuevos se instalarán, 4 para eliminar y 1 no actualizados.
Se liberarán 614 kB después de esta operación.
¿Desea continuar? [S/n] S
(Leyendo la base de datos ... 126307 ficheros o directorios instalados actualmente.)
Desinstalando ubuntu-server (1.539.2) ...
Desinstalando ubuntu-server-minimal (1.539.2) ...
Desinstalando pollinate (4.33-3.1ubuntu1.1) ...
Desinstalando curl (8.5.0-2ubuntu10.6) ...
Procesando disparadores para man-db (2.12.0-4build2) ...
```

- Despues, clonamos el repositorio con toda la información acerca de **curl**:

```
dansana@dansana:~$ sudo git clone https://github.com/curl/curl.git
Cloning into 'curl'...
remote: Enumerating objects: 253250, done.
remote: Counting objects: 100% (97/97), done.
remote: Compressing objects: 100% (54/54), done.
remote: Total 253250 (delta 58), reused 48 (delta 43), pack-reused 253153 (from 3)
Receiving objects: 100% (253250/253250), 92.77 MiB | 28.65 MiB/s, done.
Resolving deltas: 100% (204484/204484), done.
```

- Las instrucciones se ubican en **curl/README.md**, que leyendo el contenido del fichero, vemos que hay un apartado que indica acerca de la instalación de **git**. En él, se indica el apartado de la documentación donde explican cómo instalar **curl** desde el repositorio **git**, leyendo **GIT-INFO.md**:

```
dansana@dansana:~/curl$ sudo autoreconf -fi
libtoolize: putting auxiliary files in '..'.
libtoolize: copying file './ltmain.sh'.
libtoolize: putting macros in AC_CONFIG_MACRO_DIRS, 'm4'.
libtoolize: copying file 'm4/libtool.m4'.
libtoolize: copying file 'm4/ltoptions.m4'.
libtoolize: copying file 'm4/ltconfig.m4'.
libtoolize: copying file 'm4/ltversion.m4'.
libtoolize: copying file 'm4/lt~obsolete.m4'.
libtoolize: Remember to add 'LT_INIT' to configure.ac.
configure.ac:128: installing './compile'
configure.ac:455: installing './config.guess'
configure.ac:455: installing './config.sub'
configure.ac:128: installing './install-sh'
configure.ac:131: installing './missing'
docs/examples/Makefile.am: installing './depcomp'
```

- **autoreconf**: Es una utilidad del sistema **Autotools** que regenera los **scripts** del sistema de compilación de un proyecto de software, ejecutando varias herramientas en orden de **Autotools** para conseguirlo. Su objetivo es construir el fichero **configure**, que su función es comprobar el sistema si existen las dependencias necesarias y verifica las herramientas que se necesitan en la compilación. Por defecto, no está instalado en el sistema, por lo que habrá que lanzar el comando **sudo apt install autoreconf**.

- `./configure --with-openssl`: Por una parte, tenemos la ejecución del script creado en el paso anterior para crear el **Makefile** a partir de la búsqueda para encontrar ciertas utilidades en el sistema, como el compilador C, la ubicación de **make**, etc. Por otra parte, se agrega la opción que aparece al lado para que en la compilación de **curl** haya soporte con OpenSSL para permitir una comunicación segura.

Al terminar, se nos muestra la información que aparece en la figura:

```
curl version: 8.17.0-DEV
SSL: enabled (OpenSSL v3+)
SSH: no (-with-{libssh,libssh2})
ZLIB: no (-with-zlib)
Brotli: no (-with-brotli)
ZSTD: no (-with-zstd)
GSS-API: no (-with-gssapi)
GSASL: no (-with-gsasl)
TLS-SRP: enabled
Resolver: POSIX threaded
IPv6: enabled
Unix sockets: enabled
IDN: no (-with-{libidn2,winidn})
Build docs: enabled (-disable-docs)
Build libcurl: Shared=yes, Static=yes
Built-in manual: enabled
--libcurl option: enabled (--disable-libcurl-option)
Verbose errors: enabled (-disable-verbose)
Code coverage: disabled
SSPI: no (-enable-sspi)
ca native: no
ca cert bundle: /etc/ssl/certs/ca-certificates.crt
ca cert path: /etc/ssl/certs
ca cert embed: no
ca fallback: no
LDAP: no (-enable-ldap / --with-ldap-lib / --with-lber-lib)
LDAPS: no (-enable-ldaps)
IFPS/IPNS: enabled
RTSP: enabled
RTMP: no (-with-librtmp)
PSL: enabled
Alt-svc: enabled (--disable-alt-svc)
Headers API: enabled (--disable-headers-api)
HSTS: enabled (--disable-hsts)
HTTP1: enabled (internal)
HTTP2: no (-with-nghttp2)
HTTP3: no (-with-ngtcp2 --with-nghttp3, --with-quiche, --with-openssl-quic)
ECH: no (-enable-ech)
HTTPS RR: no (-enable-httpsrr)
SSL5-EXPORT: no (-enable-ssls-export)
Protocols: dict file ftp ftps gopher gophers http https imap imaps ipfs ipns mhttppop3 pop3s rtsp smb smbs smtp smtps telnet tftp ws wss
Features: alt-svc AsynchDNS HSTS HTTPS-proxy IPv6 Largefile PSL SSL threadsafe TLS-SRP UnixSockets
```

- **make**: Por último, solo tendremos que lanzar el comando **make** que se encargará de construir el programa haciendo uso del **Makefile**. Es posible que no venga instalada la herramienta y haya que lanzar el comando `sudo apt install make`:

```
dansana@dansana:~/curl$ sudo apt install make
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  libcurl4t64
Utilice «sudo apt autoremove» para eliminarlo.
Paquetes sugeridos:
  make-doc
Se instalarán los siguientes paquetes NUEVOS:
  make
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 1 no actualizados.
Se necesita descargar 180 kB de archivos.
Se utilizarán 414 kB de espacio de disco adicional después de esta operación.
Des:1 http://archive.ubuntu.com/ubuntu/noble/main amd64 make amd64 4.3-4.1build2 [180 kB]
Descargados 180 kB en 0s (960 kB/s)
Seleccionando el paquete make previamente no seleccionado.
(Leyendo la base de datos ... 126651 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../make_4.3-4.1build2_amd64.deb ...
Desempaquetando make (4.3-4.1build2) ...
Configurando make (4.3-4.1build2) ...
Procesando disparadores para man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Pending kernel upgrade!
Running kernel version:
  6.8.0-85-generic
Diagnostics:
  The currently running kernel version is not the expected kernel version 6.8.0-86-generic.

Restarting the system to load the new kernel will not be handled automatically, so you should consider rebooting.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

---

Para verificar si está bien instalada la aplicación, vamos a lanzar el programa que se encuentra en `curl/src(curl` pasándole una dirección web:

```
dansana@dansana:~/curl$ ./src/curl https://gui.inf.uva.es
<!DOCTYPE html>
<html lang="">
<head>
<meta charset="UTF-8">
<link rel="icon" href="/assets/favicon-CARYXMx0.ico">
<link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600&family=IBM+Plex+Mono:wght@400;600&display=swap" rel="stylesheet">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>GUI</title>
<script type="module" crossorigin src="/assets/index-CN0agsWQ.js"></script>
<link rel="stylesheet" crossorigin href="/assets/index-C_gVLav_.css">
</head>
<body>
<div id="app"></div>
</body>
</html>
```

Como vemos, nos devuelve el contenido de la página `gui.inf.uva.es`, verificando que funciona correctamente.

#### 4.1.3 Actualización del sistema

Una de las primeras tareas a realizar siempre cuando instalamos un sistema operativo es actualizar con los nuevos parches y versiones que existen desde que se publicó el medio de instalación usado.

Simplemente, usamos `sudo apt-get update` para obtener las actualizaciones y `sudo apt-get upgrade` para descargarlas e instalarlas. Evidentemente, si solo se hace `upgrade` sin previamente haber hecho `update`, no se instalarán las actualizaciones más recientes.

```
dansana@dansana:~$ sudo apt-get update
[sudo] password for dansana:
Leyendo lista de paquetes... Hecho
Obj:1 http://archive.ubuntu.com/ubuntu noble InRelease
Obj:2 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Obj:3 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Obj:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Leyendo lista de paquetes... Hecho
```

```
dansana@dansana:~$ sudo apt-get upgrade
Leyendo lista de paquetes... Hecho
Creado árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  libcurl4-openssl4
Utilice sudo apt autoremove para eliminarlo.
Los siguientes paquetes se han retido:
  linux-generic linux-headers-generic linux-image-generic
Se actualizarán los siguientes paquetes:
  libssh-4 libxml2 libxml2-dev linux-tools-common sosreport
5 actualizados, 0 nuevos se instalarán, 0 para eliminar y 3 no actualizados.
Se necesita descargar 3.075 kB/4.027 kB de archivos.
Se utilizarán 45.1 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] -
```

#### 4.1.4 Nuevas versiones del núcleo Linux

El último punto de la sección, se aborda el concepto de los **Tarballs**. Se tratan de unos archivos que contiene múltiples ficheros y directorios en un solo archivo. Son especialmente útiles para realizar copias de seguridad, distribuir software o archivos. Además, son una forma común de distribuir software en Linux [10].

Para llevar a cabo la tarea de determinar el nombre del paquete que contiene el núcleo, tenemos que averiguar primero su versión con `cat /proc/version` o `uname -r`:

```
dansana@dansana:~$ cat /proc/version
Linux version 6.8.0-86-generic (buildd@lcy02-amd64-031) (x86_64-linux-gnu-gcc-13 (Ubuntu 13.3.0-6ubuntu2~24.04) 13.3.0, GNU ld (GNU Binutils for Ubuntu) 2.42) #87-Ubuntu SMP PREEMPT_DYNAMIC Mon Sep 22 18:03:36 UTC 2025
```

Nos indica que se está empleando la **6.8.0-86-generic**, por lo que iremos al directorio `/boot` que es donde se encontrará los ficheros del arranque. Ahora, descargamos con `curl` que hemos descargado y compilado el código fuente de esa misma versión en forma de **tarball**. El repositorio usado se localiza en <https://cdn.kernel.org/pub/linux/kernel/v6.x/linux-6.8.tar.xz>:

```
dansana@dansana:~$ ./curl/src/curl -O https://cdn.kernel.org/pub/linux/kernel/v6.x/linux-6.8.tar.xz
% Total    % Received % Xferd  Average Speed   Time     Time   Current
          Dload  Upload Total   Spent    Left   Speed
100 135.9M 100 135.9M  0     0 35513k   0:00:03  0:00:03  ---:--- 35509k
dansana@dansana:~$ ls
curl      dependenciasGCC.txt  Documents  ejemploGrupo1  fich1.img  fich3.img      Images
dependenciasCURDL.txt  Desktop       Downloads  ejemploGrupo2  fich2.img  helpConfigure.txt  linux-6.8.tar.xz
```

*NOTA:* La razón de usar la extensión .xz en vez de .gz se basa en el tamaño del núcleo de Linux.

Por último, vamos a comprobar el contenido del fichero comprimido, descomprimiéndolo primero con el comando `tar -xvf linux-6.8.tar.xz`. Cuando haya terminado, el contenido que se nos mostrará será el siguiente:

```
dansana:dansana:"$ ls -la linux-6.8
total 1112
drwxrwxr-x  26 root      root      4096 mar 10 2024 .
drwxr-x---  13 dansana  dansana  4096 nov 20 12:21 ..
drwxrwxr-x  23 root      root      4096 mar 10 2024 arch
drwxrwxr-x  3 root      root      4096 mar 10 2024 block
drwxrwxr-x  2 root      root      4096 mar 10 2024 certs
-rw-rw-r--  1 root      root     22187 mar 10 2024 clang-format
-rw-rw-r--  1 root      root      59 mar 10 2024 coccinelle
-rw-rw-r--  1 root      root      496 mar 10 2024 COPYING
-rw-rw-r--  1 root      root 104031 mar 10 2024 CREDITS
drwxrwxr-x  4 root      root      4096 mar 10 2024 crypto
drwxrwxr-x  76 root      root      4096 mar 10 2024 Documentation
drwxrwxr-x 143 root      root      4096 mar 10 2024 drivers
-rw-rw-r--  1 root      root      672 mar 10 2024 editorconfig
drwxrwxr-x  81 root      root      4096 mar 10 2024 fs
-rw-rw-r--  1 root      root      151 mar 10 2024 .get_maintainer.ignore
-rw-rw-r--  1 root      root      105 mar 10 2024 .gitattributes
-rw-rw-r--  1 root      root     2089 mar 10 2024 .gitignore
drwxrwxr-x  31 root      root      4096 mar 10 2024 include
drwxrwxr-x  2 root      root      4096 mar 10 2024 init
drwxrwxr-x  2 root      root      4096 mar 10 2024 ioURING
drwxrwxr-x  2 root      root      4096 mar 10 2024 ipc
drwxrwxr-x  1 root      root     2573 mar 10 2024 Kbuild
-rw-rw-r--  1 root      root      555 mar 10 2024 Kconfig
drwxrwxr-x  22 root      root      4096 mar 10 2024 kernel
drwxrwxr-x  21 root      root    12288 mar 10 2024 lib
drwxrwxr-x  6 root      root      4096 mar 10 2024 LICENSES
-rw-rw-r--  1 root      root    38618 mar 10 2024 .mailmap
-rw-rw-r--  1 root      root 741084 mar 10 2024 MAINTAINERS
-rw-rw-r--  1 root      root    67710 mar 10 2024 Makefile
drwxrwxr-x  6 root      root      4096 mar 10 2024 mm
drwxrwxr-x  72 root      root      4096 mar 10 2024 net
-rw-rw-r--  1 root      root     727 mar 10 2024 README
drwxrwxr-x  7 root      root      4096 mar 10 2024 rust
-rw-rw-r--  1 root      root     369 mar 10 2024 .rustfmt.toml
drwxrwxr-x  42 root      root      4096 mar 10 2024 samples
drwxrwxr-x  17 root      root      4096 mar 10 2024 scripts
drwxrwxr-x  14 root      root      4096 mar 10 2024 security
drwxrwxr-x  27 root      root      4096 mar 10 2024 sound
drwxrwxr-x  43 root      root      4096 mar 10 2024 tools
drwxrwxr-x  4 root      root      4096 mar 10 2024 usr
drwxrwxr-x  4 root      root      4096 mar 10 2024 virt
```

## 4.2 Herramientas de Desarrollo

#### 4.2.1 Shell script avanzados

Un **shell script** es un programa creado con instrucciones que son ejecutadas por un **shell** de Linux o Unix. Necesita un programa que entienda los comandos y estructuras que contiene y esto se suele poner en la primera línea del programa (`#!/bin/bash`) [11].

Una vez definido el concepto, procedemos a escribir el programa que nos devuelva los datos pedidos:

- Lógica central: arch, init, kernel y mm
- Hardware y Controladores: drivers, block y sound
- Conectividad y Almacenamiento: net y fs
- Seguridad y Características Avanzadas: security, crypto, virt, ipc y io\_uring
- Herramientas y Soporte: include, lib, scripts, tools, Documentation, samples, rust
- Ficheros importantes: Makefile, Kconfig, COPYING, MAINTAINERS y CREDITS

- Ficheros importantes: Makefile, Kconfig, COPYING, MAINTAINERS y CREDITS

ucciones que son ejecutadas por un shell de Linux o

s y estructuras que contiene y esto se suele poner en

el programa que nos devuelva los datos pedidos:

```

  disk = os.path.join(working_dir, "disk")
  name = "disk"
  count=1
  totalSize += size;
  print("")

  if (min_size <= i) & (size < min_size) | min_size < size: f.write(name + "\n")
  if (max_size > size): max_size = size; f.write(name + "\n")
  if (time < runTime): runTime = time; f.write(name + "\n")
  if (time > maxTime): maxTime = time; f.write(name + "\n")
  if (count > maxCount): maxCount = count; f.write(name + "\n")
  if (bytes > maxBytes): maxBytes = bytes; f.write(name + "\n")
  if (bytes < minBytes): minBytes = bytes; f.write(name + "\n")
  count+=1
  print("Número de objetos: " + count)
  print("Objeto más grande: " + name + " (" + str(maxBytes) + " bytes)")
  print("Objeto más pequeño: " + name + " (" + str(minBytes) + " bytes)")
  print("Tiempo más tarde: " + name + " (" + str(maxTime) + " bytes)")
  print("Tiempo más temprano: " + name + " (" + str(minTime) + " bytes)")
  print("Objeto más nuevo: " + name)
  print("Objeto más antiguo: " + name)

  print("")

  # Recorremos el total de bytes y de usuario para mostrarlo en la lista
  totalSize = 0
  for user in users:
    totalSize += len(users[user])
    echo("Total bytes Usuario: " + str(totalSize) + " " + user)
    echo("")

  # Verificamos los archivos maestros a 10000 en /home
  if (len(grandes) > 10000):
    grandes.sort(key=lambda x: x['size'], reverse=True)
    print("Archivos grandes: " + str(len(grandes)))
    if (len(grandes) > 10000):
      echo("Archivos grandes: " + str(len(grandes)))
      echo("Archivos grandes: " + str(len(grandes)))
    else:
      echo("Archivos grandes: " + str(len(grandes)))
  else:
    echo("Archivos grandes: " + str(len(grandes)))

  done = </etc/passwd>
  print("")

  # Una vez terminada la primera parte, imprimimos las listas ordenadas
  echo "<-- LÍSTAS USUARIOS ORDENADOS POR ESPACIO CONSUMIDO -->"
  if (len(users) > 0):
    users = sorted(users.items(), key=lambda x: len(x[1]), reverse=True)
    for user in users:
      sort = "n"
      if (sort == "n"):
        print("User: " + user[0] + " " + str(len(user[1])) + " bytes")
      else:
        print("User: " + user[0] + " " + str(len(user[1])) + " bytes")
  else:
    echo("No hay usuarios en el sistema")

  echo "<-- LÍSTAS USUARIOS CON FICHEROS EN /home MAYORES DE 10KB -->"
  if (len(grandes) > 0):
    for user in grandes:
      cat = "cat " + user[0] + ".txt"
      sort = "n"
      if (sort == "n"):
        print(cat)
      else:
        print(cat)
  else:
    echo("No hay usuarios con ficheros en /home mayores de 10KB")

  # Minimizamos los ficheros temporales
  rm -f *.txt
  rm -f *.log

```

Comprobamos su funcionamiento, después otorgarle permisos de ejecución con `sudo chmod +x usuarios.sh`, y estos son los resultados que nos imprime (redirigiendo la salida a un fichero de texto, ya que sino no es posible verlo al completo):

```
1 -Usuario: root--
2   Número de objetos: 230951
3   Total ocupado: 8711915873 bytes
4   Objeto más pequeño: /root/.ssh/authorized_keys (0 bytes)
5   Objeto más grande: /mnt/backups/home.tar (1073879040 bytes)
6   Objeto más antiguo: /boot/efi
7   Objeto más nuevo: /tmp
8 --Usuario: daemon--
9   Número de objetos: 4
10  Total ocupado: 63822 bytes
11  Objeto más pequeño: /var/spool/cron/atjobs/.SEQ (6 bytes)
12  Objeto más grande: /usr/bin/at (55624 bytes)
13  Objeto más antiguo: /usr/bin/at
14  Objeto más nuevo: /var/spool/cron/atjobs
15 --Usuario: bin--
16   Número de objetos: 1
17   Total ocupado: 0 bytes
18   Objeto más pequeño: ( bytes)
19   Objeto más grande: (0 bytes)
20   Objeto más antiguo:
21   Objeto más nuevo:
22 --Usuario: sys--
23   Número de objetos: 1
24   Total ocupado: 0 bytes
25   Objeto más pequeño: ( bytes)
26   Objeto más grande: (0 bytes)
27   Objeto más antiguo:
28   Objeto más nuevo:
29 --Usuario: sync--
30   Número de objetos: 1
31   Total ocupado: 0 bytes
32   Objeto más pequeño: ( bytes)
33   Objeto más grande: (0 bytes)
34   Objeto más antiguo:
35   Objeto más nuevo:
36 --Usuario: games--
37   Número de objetos: 1
38   Total ocupado: 0 bytes
39   Objeto más pequeño: ( bytes)
40   Objeto más grande: (0 bytes)
41   Objeto más antiguo:
42   Objeto más nuevo:
43 --Usuario: man--
44   Número de objetos: 166
45   Total ocupado: 2028364 bytes
46   Objeto más pequeño: /var/cache/man/ro/CACHEDIR.TAG (190 bytes)
47   Objeto más grande: /var/cache/man/index.db (1032192 bytes)
48   Objeto más antiguo: /var/cache/man/cat6
49   Objeto más nuevo: /var/cache/man/ja
"salida.txt" 331L, 10285B
```

```
281 --- LISTA USUARIOS ORDENADOS POR ESPACIO CONSUMIDO ---
282 Usuario: backup , Ocupa: 0 bytes
283 Usuario: bin , Ocupa: 0 bytes
284 Usuario: dhcpcd , Ocupa: 0 bytes
285 Usuario: games , Ocupa: 0 bytes
286 Usuario: irc , Ocupa: 0 bytes
287 Usuario: list , Ocupa: 0 bytes
288 Usuario: lp , Ocupa: 0 bytes
289 Usuario: mail , Ocupa: 0 bytes
290 Usuario: messagebus , Ocupa: 0 bytes
291 Usuario: news , Ocupa: 0 bytes
292 Usuario: nobody , Ocupa: 0 bytes
293 Usuario: polkitd , Ocupa: 0 bytes
294 Usuario: proxy , Ocupa: 0 bytes
295 Usuario: sshd , Ocupa: 0 bytes
296 Usuario: sync , Ocupa: 0 bytes
297 Usuario: sys , Ocupa: 0 bytes
298 Usuario: systemd-network , Ocupa: 0 bytes
299 Usuario: systemd-resolve , Ocupa: 0 bytes
300 Usuario: tcpdump , Ocupa: 0 bytes
301 Usuario: usbmux , Ocupa: 0 bytes
302 Usuario: uucp , Ocupa: 0 bytes
303 Usuario: www-data , Ocupa: 0 bytes
304 Usuario: fwupd-refresh , Ocupa: 4096 bytes
305 Usuario: pollinate , Ocupa: 4096 bytes
306 Usuario: systemd-timesync , Ocupa: 4096 bytes
307 Usuario: tss , Ocupa: 4096 bytes
308 Usuario: uuid , Ocupa: 4096 bytes
309 Usuario: landscape , Ocupa: 8475 bytes
310 Usuario: user1 , Ocupa: 17788 bytes
311 Usuario: user2 , Ocupa: 17788 bytes
312 Usuario: _apt , Ocupa: 32768 bytes
313 Usuario: user3 , Ocupa: 50556 bytes
314 Usuario: userB , Ocupa: 58748 bytes
315 Usuario: daemon , Ocupa: 63822 bytes
316 Usuario: userA , Ocupa: 69440 bytes
317 Usuario: user4 , Ocupa: 70520 bytes
318 Usuario: man , Ocupa: 2028364 bytes
319 Usuario: syslog , Ocupa: 11384752 bytes
320 Usuario: dansana , Ocupa: 271520933 bytes
321 Usuario: root , Ocupa: 8711915873 bytes
322 --- LISTA USUARIOS CON FICHEROS EN /home MAYORES DE 100K
323 demon
324 man
325 backup
326 systemd-network
327 systemd-timesync
328 systemd-resolve
329 polkitd
```

#### 4.2.2 Creación de una aplicación con lenguaje de alto nivel

Para terminar con esta sección, vamos a profundizar en un programa que se encargue de leer los datos de un fichero de log que contiene información procedente de Apache Web Server y de Internet Information Server. Además, crearemos un **Makefile** que se encargue de la actualización del ejecutable de la aplicación. Un **Makefile** contiene un conjunto de instrucciones que se usan para ayudar a decidir qué partes de un programa grande necesitan ser recompiladas [12].

Lo primero que nos piden es analizar el formato de los ficheros de log de d100.log y da-11-16.ipntld.log:

- 
- Según su estructura, los ficheros siguen el formato estándar NCSA Combined Log Format, que es el formato por defecto y más utilizado por Apache Web Server.
  - Tomemos la primera línea del fichero d100.log para diseccionarla. Cada espacio separa un campo diferente:
    - 136.199.199.88: IP del Cliente
    - -: Identidad (RFC 1413)
    - -: Usuario
    - [17/Nov/2012:03:00:27 +0100]: Fecha y hora
    - GET /index.php HTTP/1.1: Petición
    - 200: Código de estado
    - 4882: Tamaño
    - "-": Referer
    - "Mozilla/5.0 ...": User Agent

Podemos concluir que el fichero d100.log muestra la actividad normal de un servidor web:

1. Vemos tráfico de 3 sistemas operativos distintos (según su dirección IP)
2. Vemos un cliente antiguo o bot (IP 213.60...) buscando archivos de configuración estándar (favicon, robots.txt) que no existen, generando errores 404.
3. Vemos un usuario recurrente (IP 83.33...) que aprovecha bien la caché del navegador (muchos códigos 304), lo que reduce la carga del servidor.

Después de este análisis, construimos un programa que lea el fichero log y determine la dirección que más consultas ha hecho al servidor, la cantidad de bytes que se han descargado tanto por cliente conectado como por hora. Cuando lo tengamos, crearemos el fichero Makefile para la aplicación:

```
dansana@dansana:~$ cat Makefile
CC = gcc
CFLAGS = -Wall -pg # -pg activa gprof
TARGET = analizador
SRC = analizador_log.c

# Regla por defecto
all: $(TARGET)

# Compilación
$(TARGET): $(SRC)
    $(CC) $(CFLAGS) -o $(TARGET) $(SRC)

# Perfilado automático (Compila, ejecuta y genera informe)
perfilado: all
    ./$(TARGET)
    gprof $(TARGET) gmon.out > analisis_rendimiento.txt
    @echo "Informe generado en analisis_rendimiento.txt"

# Limpieza
clean:
    rm -f $(TARGET) gmon.out analisis_rendimiento.txt
```

Como se puede observar, hemos creado una regla que realiza un perfilado automático y lo guarda en un fichero de texto con todo el análisis. Hay que usar el fichero log da-11-16.ipntld.log:

```
gcc -Wall -pg -o analizador analizador_log.c
./analizador
Procesando fichero da-11-16.ipntld.log ...
Analisis completado. Clientes unicos: 2486
gprof analizador gmon.out > analisis_rendimiento.txt
Informe generado en analisis_rendimiento.txt
dansana@dansana:~/Documents$ cat analisis_rendimiento.txt
Flat profile:

Each sample counts as 0.01 seconds.
% cumulative   self      self      total
time    seconds   seconds   calls  Ts/call  Ts/call  name
71.43      0.05     0.05          1        0.05        0.05  main
28.57      0.07     0.02          1        0.07        0.07  __init

%           the percentage of the total running time of the
time          program used by this function.

cumulative a running sum of the number of seconds accounted
seconds   for by this function and those listed above it.

self      the number of seconds accounted for by this
seconds   function alone. This is the major sort for this
           listing.

calls     the number of times this function was invoked, if
           this function is profiled, else blank.

self      the average number of milliseconds spent in this
ms/call   function per call, if this function is profiled,
           else blank.

total      the average number of milliseconds spent in this
ms/call   function and its descendants per call, if this
           function is profiled, else blank.

name      the name of the function. This is the minor sort
           for this listing. The index shows the location of
           the function in the gprof listing. If the index is
           in parenthesis it shows where it would appear in
           the gprof listing if it were to be printed.
```

Copyright (C) 2012-2024 Free Software Foundation, Inc.

Copying and distribution of this file, with or without modification,  
are permitted in any medium without royalty provided the copyright  
notice and this notice are preserved.

## 5. Administración avanzada de servicios

### 5.1 Niveles de ejecución (Run-level)

En entornos UNIX/LINUX, los niveles de ejecución son aquellos estados que definen qué servicios del sistema están funcionando y se identifican por números [13].

Como bien se menciona, en la actualidad, se han sustituido los **run-level** por **targets** (realizamos esta tarea en [2.5](#)). En nuestro sistema, investigando los targets que existen en nuestra máquina, lanzamos el comando `ls -l /etc/systemd/system`:

```
dansana@dansana:~$ ls -l /etc/systemd/system
total 100
drwxr-xr-x 2 root root 4096 ago 27 2024 cloud-final.service.wants
drwxr-xr-x 2 root root 4096 ago 27 2024 cloud-init.target.wants
lrwxrwxrwx 1 root root   48 ago 27 2024 dbus-org.freedesktop.resolve1.service -> /usr/lib/systemd/system/systemd-resolved.service
lrwxrwxrwx 1 root root   40 oct  6 14:38 dbus-org.freedesktop.thermald.service -> /usr/lib/systemd/system/thermald.service
lrwxrwxrwx 1 root root   49 ago 27 2024 dbus-org.freedesktop.timesync1.service -> /usr/lib/systemd/system/systemd-timesyncd.service
lrwxrwxrwx 1 root root   41 oct  9 19:28 default.target -> /usr/lib/systemd/system/multi-user.target
drwxr-xr-x 2 root root 4096 ago 27 2024 display-manager.service.wants
drwxr-xr-x 2 root root 4096 oct  6 14:37 emergency.target.wants
drwxr-xr-x 2 root root 4096 ago 27 2024 final.target.wants
drwxr-xr-x 2 root root 4096 ago 27 2024 getty.target.wants
drwxr-xr-x 2 root root 4096 ago 27 2024 graphical.target.wants
drwxr-xr-x 2 root root 4096 oct 14:37 hibernate.target.wants
drwxr-xr-x 2 root root 4096 oct  6 14:37 hybrid-sleep.target.wants
lrwxrwxrwx 1 root root   42 ago 27 2024 iscsi.service -> /usr/lib/systemd/system/open-iscsi.service
drwxr-xr-x 2 root root 4096 ago 27 2024 mdmonitor.service.wants
drwxr-xr-x 2 root root 4096 oct 12 12:07 multi-user.target.wants
drwxr-xr-x 2 root root 4096 ago 27 2024 network-online.target.wants
drwxr-xr-x 2 root root 4096 ago 27 2024 oem-config.service.wants
drwxr-xr-x 2 root root 4096 ago 27 2024 open-vm-tools.service.requires
drwxr-xr-x 2 root root 4096 ago 27 2024 paths.target.wants
drwxr-xr-x 2 root root 4096 oct  6 14:37 rescue.target.wants
drwxr-xr-x 2 root root 4096 oct  6 14:37 sleep.target.wants
drwxr-xr-x 2 root root 4096 oct  6 14:39 sockets.target.wants
drwxr-xr-x 2 root root 4096 ago 27 2024 sshd-keygen@.service.d
lrwxrwxrwx 1 root root   35 oct  9 18:28 sshd.service -> /usr/lib/systemd/system/sshd.service
drwxr-xr-x 2 root root 4096 oct  6 14:39 ssh.service.requires
drwxr-xr-x 2 root root 4096 oct  6 14:37 suspend.target.wants
drwxr-xr-x 2 root root 4096 oct  6 14:37 suspend-then-hibernate.target.wants
drwxr-xr-x 2 root root 4096 ago 27 2024 susinit.target.wants
lrwxrwxrwx 1 root root   39 ago 27 2024 syslog.service -> /usr/lib/systemd/system/rsyslog.service
drwxr-xr-x 2 root root 4096 ago 27 2024 sysstat.service.wants
drwxr-xr-x 2 root root 4096 ago 27 2024 timers.target.wants
lrwxrwxrwx 1 root root   45 ago 27 2024 vmtoolsd.service -> /usr/lib/systemd/system/open-vm-tools.service
```

Ahora, con ayuda del manual de Linux (`man runlevel`, `man systemd` y `man systemd.target`) ponemos en práctica los niveles de ejecución del sistema:

- Para determinar el **target** por defecto que se encuentra, usamos el comando `systemctl get-default`:

```
dansana@dansana:~$ systemctl get-default
multi-user.target
```

El que aparece por defecto, es el **multi-user.target**, que es el que cambiamos en [2.5](#).

- Ahora, pasamos al nivel **monousuario** o el antiguo nivel 1, que corresponde a **rescue.target**. En este modo, el sistema cerrará los servicios de red y gráficos, dejándonos solos en una consola de texto:

```
You are in rescue mode. After logging in, type "journalctl -xb" to view
system logs, "systemctl reboot" to reboot, or "exit"
to continue bootup.
Press Enter for maintenance
(or press Control-D to continue):
```

- Volvemos al modo anterior, con el mismo comando cambiando el **target**.

## 5.2 Creación de nuevos servicios

Una vez que conocemos los distintos **targets** y servicios que existen en el sistema (Bloque 1), vamos a crear nuestro propio servicio que se encargue de montar un disco sobre un dispositivo **loop** cada vez que se inicie el sistema.

Primero, creamos el disco sobre un fichero (como en el Bloque 1) y montar un sistema de ficheros dentro de él:

```
dansana@dansana:~$ sudo dd if=/dev/zero of=/media/disk bs=1M count=100
[sudo] password for dansana:
100+0 records in
100+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 0,0638091 s, 1,6 GB/s
dansana@dansana:~$ ls -l /media/
total 102400
-rw-r--r-- 1 root root 104857600 nov 22 17:17 disk
```

```
dansana@dansana:~$ sudo mkfs.ext4 /media/disk
mke2fs 1.47.0 (5-Feb-2023)
Discarding device blocks: done
Creating filesystem with 25600 4k blocks and 25600 inodes

Allocating group tables: done
Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done
```

Creamos el fichero en `/root/.scripts/inicia_disco.sh` y le otorgamos permisos de ejecución con el comando `chmod +x inicia_disco.sh`. Despues, creamos el fichero con el servicio en `/etc/systemd/system/inicia_mi_vol.service` y lo habilitamos con `systemctl enable inicia_mi_vol.service`:

```
root@dansana:~/.scripts# systemctl enable inicia_mi_vol.service
Created symlink /etc/systemd/system/multi-user.target.wants/inicia_mi_vol.service → /etc/systemd/system/inicia_mi_vol.service.
```

```
dansana@dansana:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs          594M   1,2M  593M   1% /run
efivarfs        56K   26K   26K  50% /sys/firmware/efi/efivars
/dev/sda6       15G   2,3G   12G  16% /
/dev/sda3       15G   2,7G   12G  20% /usr
tmpfs          2,9G     0  2,9G   0% /dev/shm
tmpfs          5,0M     0  5,0M   0% /run/lock
/dev/sda1       1,1G   6,2M   1,1G   1% /boot/efi
/dev/sda7       20G   2,9G   16G  16% /home
/dev/sda5       2,9G   52K   2,8G   1% /tmp
/dev/sda4       4,9G  741M   3,9G  16% /var
/dev/loop40      90M   24K   83M   1% /mnt
tmpfs          594M   12K  594M   1% /run/user/1000
```

### 5.3 Servidor Web

Continuando con esta sección, vamos a instalar y configurar un servidor web en nuestra máquina, concretamente el paquete **nginx**. Para ello, primero comprobamos si existe alguna actualización y instalamos dicho paquete:

```
dansana@dansana:~$ sudo apt install nginx
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  nginx-common
Paquetes sugeridos:
  fcgiwrap nginx-doc ssl-cert
Se instalarán los siguientes paquetes NUEVOS:
  nginx nginx-common
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 1 no actualizados.
Se necesita descargar 564 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx-common all 1.24.0-2ubuntu7.5 [43,4 kB]
Des:2 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx amd64 1.24.0-2ubuntu7.5 [520 kB]
Descargado 564 kB en 0s (2,100 kB/s)
Preconfigurando paquetes ...
Seleccionando el paquete nginx-common previamente no seleccionado.
(Leyendo la base de datos ... 128405 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../nginx-common_1.24.0-2ubuntu7.5_all.deb ...
Desempaquetando nginx-common (1.24.0-2ubuntu7.5) ...
Seleccionando el paquete nginx previamente no seleccionado.
Preparando para desempaquetar .../nginx_1.24.0-2ubuntu7.5_amd64.deb ...
Desempaquetando nginx (1.24.0-2ubuntu7.5) ...
Configurando nginx-common (1.24.0-2ubuntu7.5) ...
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
Configurando nginx (1.24.0-2ubuntu7.5) ...
[*] Upgrading binary nginx
Procesando disparadores para man-db (2.12.0-4build2) ...
Procesando disparadores para ufw (0.36.2-6) ...
Scanning processes...
Scanning linux images...

Pending kernel upgrade!
Running kernel version:
  6.8.0-87-generic
Diagnostics:
  The currently running kernel version is not the expected kernel version 6.8.0-88-generic.
  Restarting the system to load the new kernel will not be handled automatically, so you should consider rebooting.

No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.

[ OK ]
```

Ahora, comprobamos con la herramienta **curl**, si funciona correctamente el funcionamiento local (**localhost**):

```
dansana@dansana:~$ curl /src/curl localhost
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

Como se puede apreciar en el código HTML, se puede ver la página de bienvenida de **nginx**, confirmando que el servicio funciona correctamente.

---

Siguiendo la documentación oficial de `nginx` [14] para saber cómo configurar el servicio web, vemos que se nos menciona el fichero de configuración principal `/etc/nginx/nginx.conf` que importa distintos módulos para desempeñar distintas funciones. Además, en el directorio `/var/log/nginx`, se ubican los distintos logs o registros que son necesarios tener en cuenta para auditar su funcionamiento.

Lo siguiente, es agregar en el directorio `/var/www/html` un par de páginas HTML con enlaces a ficheros para descargar con imágenes que visualizar. Cuando estén, accedemos a esos ficheros HTML con `curl` y vemos como han cambiado los fichero de registro anteriores:

```
dansana@dansana:~$ sudo cat /var/log/nginx/access.log
[sudo] password for dansana:
::1 - - [22/Nov/2025:18:32:45 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/8.17.0-DEV"
::1 - - [22/Nov/2025:19:28:25 +0000] "GET / HTTP/1.1" 200 615 "-" "curl/8.17.0-DEV"
::1 - - [22/Nov/2025:19:28:35 +0000] "GET /index.html HTTP/1.1" 404 162 "-" "curl/8.17.0-DEV"
::1 - - [22/Nov/2025:19:28:46 +0000] "GET /prueba.html HTTP/1.1" 404 162 "-" "curl/8.17.0-DEV"
```

Se muestra en las dos últimas líneas el acceso a las páginas HTML indicadas.

Por último, vamos poner a prueba la seguridad de nuestro servicio web. Le he pedido a un compañero que acceda a mi servicio web lanzando el comando `curl 10.0.38.3`, que es mi dirección IP:

```
roisuar@usuario:~$ curl 10.0.38.3
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
roisuar@usuario:~$
```

Como bien se aprecia, se puede demostrar lo que nos indica el enunciado de que por defecto, Ubuntu no trae activados los cortafuegos, así que mi compañero ha podido conectarse a mi servicio web.

Para activar el `firewall`, utilizamos el comando `sudo ufw enable` que interactuará con `iptables` de nuestra máquina:

```
dansana@dansana:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

---

La política por defecto es que rechazar todas las conexiones entrantes y se puede comprobar si desde otra máquina intentamos conectarnos a la máquina virtual, por ejemplo mediante `ssh`.

Para permitir que se puedan realizar conexiones SSH, tenemos que poner el siguiente comando:

```
sudo iptables -I INPUT -p tcp --dport 22 -j ACCEPT;
```

- `-I INPUT`: Insertar al principio de la cadena de entrada
- `-p tcp`: Protocolo TCP
- `--dport 22`: Puerto destino 22
- `-j ACCEPT`: Aceptar paquete

Y para el servicio HTTP, el comando será: `sudo iptables -I INPUT -p tcp --dport 80 -j ACCEPT`

## 5.4 XWindow

El **X Window System**, también conocido como **X Windows**, es un sistema de ventanas que proporciona una infraestructura para la creación y manipulación de interfaces gráficas de usuario en sistemas **UNIX** [15]. Se realizará una instalación mínima con los siguientes paquetes:

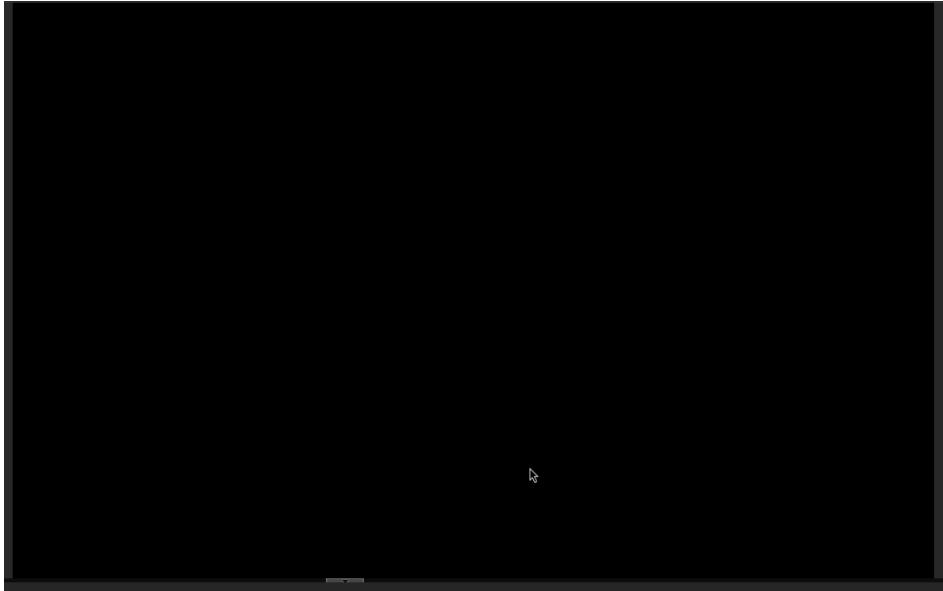
- `xserver-xorg-core`: El corazón del sistema gráfico (el servidor)
- `openbox`: Un gestor de ventanas ligero (para poder moverlas, cerrarlas, etc.)
- `xinit`: Aplicación para iniciar el entorno gráfico XWindow
- `xterm`: La terminal básica en modo gráfico
- `slim`: Un gestor de inicio de sesión. Al instalar `slim`, el sistema cambiará su comportamiento de arranque por defecto al `runlevel 5` (modo gráfico).

Ahora, iniciamos sesión como `root` para arrancar el servidor, usando el comando `X`. Como estamos en una máquina virtual, no hay manera de salir, por lo que habrá que iniciar sesión desde otra máquina mediante `ssh` y matar el proceso. Al finalizarlo, nos aparecerá esto:

```
root@dansana:~# X
X.Org X Server 1.21.1.11
X Protocol Version 11, Revision 0
Current Operating System: Linux dansana 6.8.0-87-generic #88-Ubuntu SMP PREEMPT_DYNAMIC Sat Oct 11 09:28:41 UTC 2025 x86_64
Kernel command line: BOOT_IMAGE=/boot/vmlinuz-6.8.0-87-generic root=UUID=bf4ae9ff-8ca6-4d69-ad1a-2eeb46ba55c8 ro
xorg-server 2:21.1.12-1ubuntu1.5 (For technical support please see http://www.ubuntu.com/support)
Current version of pixman: 0.42.2
Before reporting problems, check http://wiki.x.org
to make sure that you have the latest version.
Markers: (--) probed, (**) from config file, (==) default setting,
(++) from command line, (!!) notice, (II) informational,
(WW) warning, (EE) error, (NI) not implemented, (??) unknown.
(==) Log file: "/var/log/Xorg.0.log", Time: Sat Nov 22 20:58:54 2025
(==) Using system config directory "/usr/share/X11/xorg.conf.d"
pci id for fd 16: 1234:1111, driver (null)
pci id for fd 17: 1234:1111, driver (null)
kmsro: driver missing
The XKEYBOARD keymap compiler (xkbcomp) reports:
> Warning: Could not resolve keysym XF86CameraAccessEnable
> Warning: Could not resolve keysym XF86CameraAccessDisable
> Warning: Could not resolve keysym XF86CameraAccessToggle
> Warning: Could not resolve keysym XF86NextElement
> Warning: Could not resolve keysym XF86PreviousElement
> Warning: Could not resolve keysym XF86AutopilotEngageToggle
> Warning: Could not resolve keysym XF86MarkWaypoint
> Warning: Could not resolve keysym XF86Sos
> Warning: Could not resolve keysym XF86NavChart
> Warning: Could not resolve keysym XF86FishingChart
> Warning: Could not resolve keysym XF86SingleRangeRadar
> Warning: Could not resolve keysym XF86DualRangeRadar
> Warning: Could not resolve keysym XF86RadarOverlay
> Warning: Could not resolve keysym XF86TraditionalSonar
> Warning: Could not resolve keysym XF86ClearvuSonar
> Warning: Could not resolve keysym XF86SidevuSonar
> Warning: Could not resolve keysym XF86NavInfo
Errors from xkbcomp are not fatal to the X server
(II) Server terminated successfully (0). Closing log file.
```

---

Por otro lado, se puede iniciar el gestor mediante el comando `xinit` y lanzar un primer cliente que es normalmente `xterm`. Leyendo la documentación acerca de ese comando [16], primero hay que configurar el fichero `/root/.xinitrc` y agregar una línea con `exec openbox-session`. Al ejecutar el servicio, se iniciará el gestor de ventanas Openbox:



Otros clientes del servicio se encuentran dentro del paquete `x11-apps`, como `xclock` (Un reloj analógico), `xeyes` (Unos ojos que siguen el cursor) y `xcalc` (Una calculadora científica).

```
root@danasa:~# apt install x11-apps
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libxcb-damage0
Paquetes sugeridos:
  mesa-utils
Se instalarán los siguientes paquetes NUEVOS:
  libxcb-damage0 x11-apps
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 1 no actualizados.
Se necesita descargar 714 kB de archivos.
Se utilizarán 2.523 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://archive.ubuntu.com/ubuntu noble/main amd64 libxcb-damage0 amd64 1.15-1ubuntu2 [4.980 B]
Des:2 http://archive.ubuntu.com/ubuntu noble/main amd64 x11-apps amd64 7.7+1build3 [709 kB]
Descargados 714 kB en 0s (1.903 kB/s)
Seleccionando el paquete libxcb-damage0:amd64 previamente no seleccionado.
(Leyendo la base de datos ... 144945 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../libxcb-damage0_1.15-1ubuntu2_amd64.deb ...
Desempaquetando libxcb-damage0:amd64 (1.15-1ubuntu2) ...
Seleccionando el paquete x11-apps previamente no seleccionado.
Preparando para desempaquetar .../x11-apps_7.7+1build3_amd64.deb ...
Desempaquetando x11-apps (7.7+1build3) ...
Configurando libxcb-damage0:amd64 (1.15-1ubuntu2) ...
Configurando x11-apps (7.7+1build3) ...
Procesando disparadores para libc-bin (2.39-0ubuntu8.6) ...
Procesando disparadores para man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Pending kernel upgrade!
Running kernel version:
  6.8.0-87-generic
Diagnostics:
  The currently running kernel version is not the expected kernel version 6.8.0-88-generic.

Restarting the system to load the new kernel will not be handled automatically, so you should consider rebooting.

No services need to be restarted.

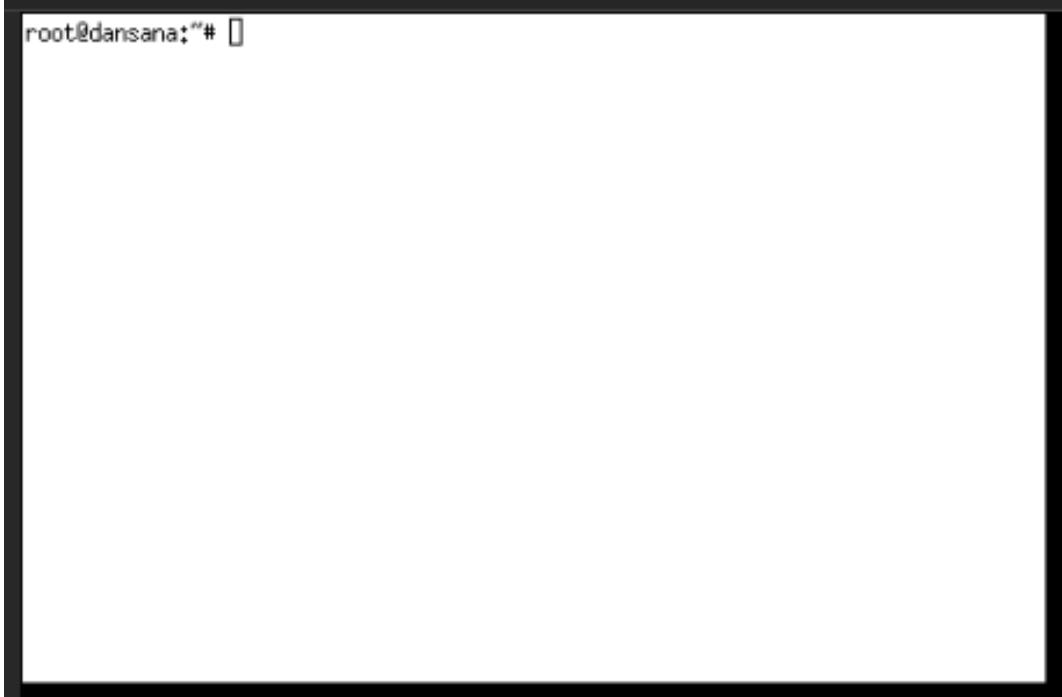
No containers need to be restarted.

No user sessions are running outdated binaries.

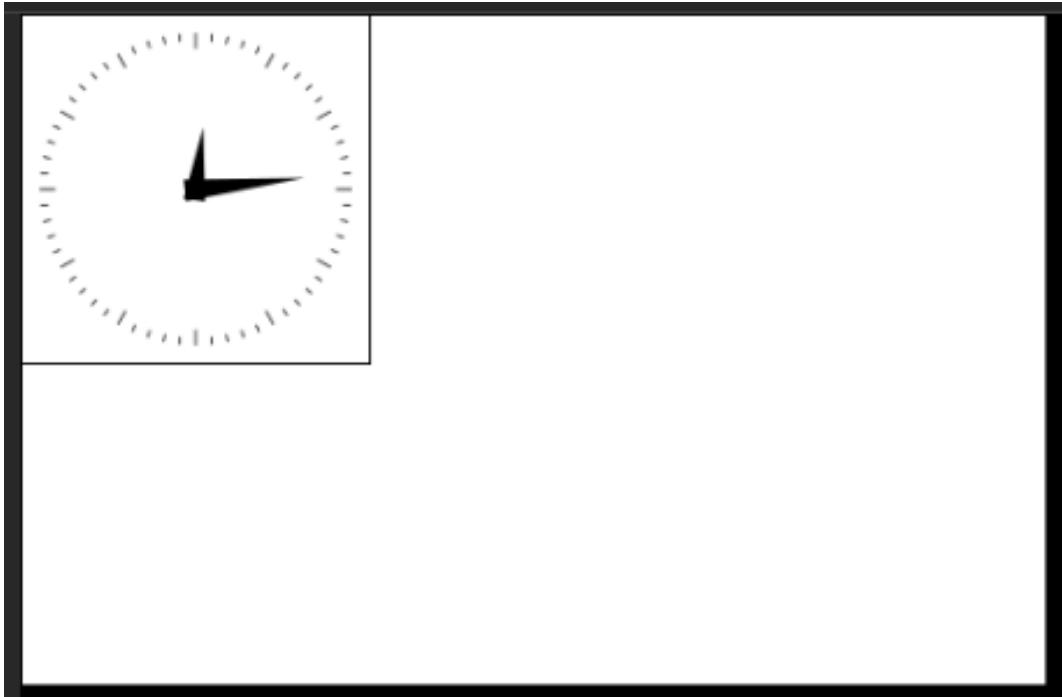
No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

---

Como queremos hacer la prueba de ejecutar los clientes desde el servidor, tenemos que iniciar el servicio **xinit** con la terminal **xterm**, modificando el **.xinitrc**:



Y desde ahí, lanzamos el cliente **xclock**:



Para iniciarlos automáticamente, modificamos el fichero anterior y agregamos **xclock -geometry 150x150+10+10 &**, antes de lanzar el cliente **xterm**. En el caso que queramos arrancar XWindow con el script **startx**, se escribe en la línea de comando **startx** y lanza el fichero **.xinitrc**.

Cuando estamos en un entorno gráfico, tenemos que saber cómo funciona la lógica del servicio. Por una parte, tenemos el servidor que se encarga de mostrar los gráficos y capturar el teclado y el ratón; y por otra parte, tenemos el cliente que es la propia aplicación. Para saber dónde se están dibujando esos gráficos, imprimimos el valor de la variable \$DISPLAY. El resultado que obtenemos es :0, que sigue el formato `hostname:display_number.screen_number`. En este caso, significa que es en la máquina local y en el primer display físico.

```
root@dansana:~# echo $DISPLAY
:0
root@dansana:~# xterm
root@dansana:~#
```

Para terminar con esta parte, vamos a ejecutar el cliente X desde mi máquina local Linux. Para ello, desde mi terminal, se lanza el comando `ssh -X -p 38031 dansana@virtual.lab.inf.uva.es` para conectarme a la máquina virtual, activando el **X11 Forwarding** para que aparezca en mi pantalla. Una vez dentro, se ejecuta el servidor y se comprueba el valor \$DISPLAY, ya que nos indicará que el túnel gráfico funciona correctamente:



```
dani@dani-ASUS:~$ ssh -X -p 38031 dansana@virtual.lab.inf.uva.es
dansana@virtual.lab.inf.uva.es's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-88-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of dom 23 nov 2025 13:59:49 UTC

System load:  0.02      Processes:          161
Usage of /home: 14.5% of 19.47GB  Users logged in:        1
Memory usage:   4%
Swap usage:    0%
IPv4 address for enp6s18: 10.0.38.3

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

El mantenimiento de seguridad expandido para Applications está desactivado
Se pueden aplicar 0 actualizaciones de forma inmediata.

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

Last login: Sun Nov 23 13:58:16 2025 from 157.88.19.40
dansana      pts/1      157.88.19.40          dom nov 23 13:59:50 +0000 2025
AGENDA:
dansana@dansana: ~ xterm
```

## 6. Monitorización

### 6.1 El sistema de archivos virtual /proc

Se trata de pseudo sistema de ficheros que proporciona una interfaz con estructuras de datos del núcleo. Se monta comúnmente en `/proc`, por lo general, se monta automáticamente por el sistema, pero también se puede montar manualmente utilizando un comando como: `mount -t proc proc /proc`. La mayoría de los archivos en el sistema de archivos `proc` son de solo lectura, aunque hay ficheros que permiten escribir, por lo que se puede cambiar las variables del kernel [17]. Este es un listado de todos los directorios y ficheros que contiene `proc`:

	1	1842	18	199	27	397	36	4919	49	68	69	77	9	asound	driver	keys	modules	stat	vmstat
1001	1244	184	2	2703	308	374	4644	5	602	639	78	910	bootconfig	dynamic_debug	key-users	mounts	swaps		zoneinfo
1002	1245	189	20	2709	32	3781	4849	50	683	7	88	912	buddyinfo	execdomains	kmsg	mtrr	sgs		
1003	1255	19	200	2714	3227	38	41	51	605	78	81	916	bus	fb	kpagegroup	net	sysrq-trigger		
1004	13	190	201	28	33	3806	42	53	606	785	82	923	cgroups	filesystems	kpagecount	pagetypeinfo	sysvshm		
1005	139	192	21	29	34	3871	426	54	61	706	85	94	cndline	fs	kpageflags	partitions	thread-self		
1009	14	193	22	295	3433	3888	43	55	613	71	86	96	consoles	interrupts	latency_stats	pressure	timer_list		
1017	15	194	23	296	3435	3912	44	56	62	72	868	97	cpuinfo	iomem	loadavg	schedstat	ttu		
1018	16	195	24	2976	35	3921	444	57	63	73	861	98	crypto	loports	locks	scsi	uptime		
103	1645	196	244	3	3508	4	447	596	64	74	87	988	devices	irq	mdstat	self	version		
114	168	197	26	30	3558	400	45	57	66	75	872	991	diskstats	kallsyms	meminfo	slabinfo	version_signature		
12	17	198	2696	3801	3555	481	47	6	67	76	891	acpi	dma	kcore	misc	softirqs	vmallocinfo		

Una vez que hemos determinado la función que desempeña `/proc`, vamos a ver la información que nos proporciona los distintos ficheros y directorios que contiene:

- `/proc/cpuinfo`: Obtenemos información acerca de la CPU:

1	processor	:	0
2	vendor_id	:	GenuineIntel
3	cpu family	:	6
4	model	:	106
5	model name	:	Intel(R) Xeon(R) Gold 5318Y CPU @ 2.10GHz
6	stepping	:	6
7	microcode	:	0xd000410
8	cpu MHz	:	2095.076
9	cache size	:	16384 KB
10	physical id	:	0
11	siblings	:	2
12	core id	:	0
13	cpu cores	:	2
14	apicid	:	0
15	initial apicid	:	0
16	fpu	:	yes
17	fpu_exception	:	yes
18	cpuid level	:	27
19	wp	:	yes
20	flags	:	fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs bts rep_good nopl xtTopology cpuid tsc_known_freq pn1 pcimulodt dtes64 vmx ssse3 fma cx16 pdcm pcid ssse4_1 ssse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch cpuid_fault ssbd ibrs ibpb stibp ibrs_enhanced tpr_shadow flexpriority ept vpid ept_ad fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid avx512f avx512dq rdseed adx smap avx512l1fa clflushopt club avx512cd sha_ni avx512b w avx512vl xsaveopt xsaves xsavec xgetbv1 xsaves unknwnv arat vnni avx512vbmi umip pkru ospke avx512_vbmi2 gfnl vaes vpclmulqdq avx512_vnni avx512_bitalg avx512_vp opcnfdq la57 rdpid frm md_clear flush_llid arch_capabilities
21	vmx flags	:	vnmi preemption_timer posted_intr invvpid ept_x_only ept_ad ept_1gb flexpriority apicv tsc_offset vtpr mtf vapid ept vpid unrestricted_gue st vapid_reg vid shadow_vnics pml tsc_scaling
22	bugs	:	spectre_v1 spectre_v2 spec_store_bypass swapgs taa mmio_stale_data elibrp_pbrsb gds bhi lbp8_no_ret
23	bogomips	:	4190.15
24	clflush size	:	64
25	cache_alignment	:	64
26	address sizes	:	46 bits physical, 57 bits virtual
27	power management:	:	
28	processor	:	1
30	vendor_id	:	GenuineIntel
31	cpu family	:	6
32	model	:	106
33	model name	:	Intel(R) Xeon(R) Gold 5318Y CPU @ 2.10GHz
34	stepping	:	6
35	microcode	:	0xd000410
36	cpu MHz	:	2095.076
37	cache size	:	16384 KB
38	physical id	:	0
39	siblings	:	2
40	core id	:	1
41	cpu cores	:	2
42	apicid	:	1
43	initial apicid	:	1

1,1 Comienzo

Lo primero que se imprime, es el modelo del procesador y la cantidad de núcleos que están asignados a esta máquina. Además, podemos destacar la cantidad de `flags` que se muestran, entre ellas, la flag `hypervisor` que indica que estamos operando en un entorno virtualizado.

- `/proc/filesystems`: Lista de sistemas soportados por el núcleo actual. Ya se explicó este fichero en 1.6:

```
dansana@dansana:~$ cat /proc/filesystems
nodev sysfs
nodev tmpfs
nodev bdev
nodev proc
nodev cgroup
nodev cgroup2
nodev cpuset
nodev devtmpfs
nodev configfs
nodev debugfs
nodev tracefs
nodev securityfs
nodev sockfs
nodev bpf
nodev pipefs
nodev ramfs
nodev hugetlbfs
nodev devpts
    ext3
    ext2
    ext4
    squashfs
    vfat
nodev encryptfs
    fuseblk
nodev fuse
nodev fusectl
nodev efivarfs
nodev mqueue
nodev pstore
    btrfs
nodev autofs
nodev binfmt_misc
```

- **/proc/interrupts:** Estadísticas actuales de las interrupciones:

	CPU0	CPU1	CPU2	CPU3			
1:	0	1662	0	0	IO-APIC 1-edge	i8042	
8:	0	0	0	0	IO-APIC 8-edge	rtc0	
9:	0	0	0	0	IO-APIC 9-fasteoi	acpi	
12:	1203	0	0	0	IO-APIC 12-edge	i8042	
16:	0	0	0	0	IO-APIC 16-fasteoi	uhci_hcd:usb2, uhci_hcd:usb6, i801_smbus	
17:	0	0	0	0	IO-APIC 17-fasteoi	uhci_hcd:usb4, uhci_hcd:usb7	
18:	0	0	0	0	IO-APIC 18-fasteoi	uhci_hcd:usb5, uhci_hcd:usb8	
19:	0	0	0	30	IO-APIC 19-fasteoi	ehci_hcd:usb1, ehci_hcd:usb3	
24:	0	0	0	0	PCI-MSIX-0000:00:1c.0 0-edge	PCIe PME, aerdrv	
25:	0	0	0	0	PCI-MSIX-0000:00:1c.1 0-edge	PCIe PME, aerdrv	
26:	0	0	0	0	PCI-MSIX-0000:00:1c.2 0-edge	PCIe PME, aerdrv	
27:	0	0	0	0	PCI-MSIX-0000:00:1c.3 0-edge	PCIe PME, aerdrv	
28:	0	0	0	0	PCI-MSIX-0000:06:03.0 0-edge	virtio0-config	
29:	0	0	8199	0	PCI-MSIX-0000:06:03.0 1-edge	virtio0-virtqueues	
30:	0	0	0	0	PCI-MSIX-0000:06:08.0 0-edge	virtio1-config	
31:	21	0	0	0	PCI-MSIX-0000:06:08.0 1-edge	virtio1-virtqueues	
32:	0	0	0	0	PCI-MSIX-0000:09:01.0 0-edge	virtio3-config	
33:	0	0	0	0	PCI-MSIX-0000:09:01.0 1-edge	virtio3-control	
34:	0	0	0	0	PCI-MSIX-0000:09:01.0 2-edge	virtio3-event	
35:	0	0	5175	0	PCI-MSIX-0000:09:01.0 3-edge	virtio3-request	
36:	0	0	0	6151	PCI-MSIX-0000:09:01.0 4-edge	virtio3-request	
37:	4482	0	0	0	PCI-MSIX-0000:09:01.0 5-edge	virtio3-request	
38:	0	4125	0	0	PCI-MSIX-0000:09:01.0 6-edge	virtio3-request	
39:	0	0	0	0	PCI-MSIX-0000:06:12.0 0-edge	virtio2-config	
40:	0	0	0	43936	PCI-MSIX-0000:06:12.0 1-edge	virtio2-input.0	
41:	0	0	9062	0	PCI-MSIX-0000:06:12.0 2-edge	virtio2-output.0	
42:	0	0	0	8295	PCI-MSI-0000:00:1f.2 0-edge	ahci[0000:00:1b.2]	
43:	0	0	0	0	PCI-MSI-0000:00:1b.0 0-edge	snd_hda_intel:card0	
NMI:	0	0	0	0	Non-maskable interrupts		
LOC:	126841	95838	119991	243288	Local timer interrupts		
SPU:	0	0	0	0	Spurious interrupts		
PMI:	0	0	0	0	Performance monitoring interrupts		
IWI:	0	0	1268	20	IRQ work interrupts		
RTR:	0	0	0	0	APIC ICR read retries		
RES:	7473	3788	4666	2241	Rescheduling interrupts		
CAL:	41999	35472	58351	28762	Function call interrupts		
TLB:	206	199	810	161	TLB shootdowns		
TRM:	0	0	0	0	Thermal event interrupts		
THR:	0	0	0	0	Threshold APIC interrupts		
DFR:	0	0	0	0	Deferred Error APIC interrupts		
MCE:	0	0	0	0	Machine check exceptions		
MCP:	50	50	50	50	Machine check polls		
HYP:	1	1	1	1	Hypervisor callback interrupts		
ERR:	0	0	0	0			
MIS:	0	0	0	0	Posted-interrupt notification event		
PIN:	0	0	0	0	Nested posted-interrupt event		
NPI:	0	0	0	0	Posted-interrupt wakeup event		
PIW:	0	0	0	0			

---

Se muestra el uso extensivo de **drivers** paravirtualizados como **virtio** con soporte MSI-X, lo que permite distribuir la carga de interrupciones de E/S (Red/Disco) eficientemente entre los 4 núcleos lógicos.

- **/proc/meminfo:** Información detallada sobre el uso de memoria:

1	MemTotal:	6074240 kB
2	MemFree:	5450692 kB
3	MemAvailable:	5573152 kB
4	Buffers:	36448 kB
5	Cached:	346812 kB
6	SwapCached:	0 kB
7	Active:	345060 kB
8	Inactive:	78812 kB
9	Active(anon):	50636 kB
10	Inactive(anon):	0 kB
11	Active(file):	294424 kB
12	Inactive(file):	78812 kB
13	Unevictable:	27444 kB
14	Mlocked:	27444 kB
15	SwapTotal:	1048572 kB
16	SwapFree:	1048572 kB
17	Zswap:	0 kB
18	Zswapped:	0 kB
19	Dirty:	36 kB
20	Writeback:	0 kB
21	AnonPages:	68108 kB
22	Mapped:	76116 kB
23	Shmem:	1256 kB
24	KReclaimable:	25212 kB
25	Slab:	87436 kB
26	SReclaimable:	25212 kB
27	SUnreclaim:	62224 kB
28	KernelStack:	2960 kB
29	PageTables:	2372 kB
30	SecPageTables:	0 kB
31	NFS_Unstable:	0 kB
32	Bounce:	0 kB
33	WritebackTmp:	0 kB
34	CommitLimit:	4085692 kB
35	Committed_AS:	315476 kB
36	VmallocTotal:	13743895347199 kB
37	VmallocUsed:	41504 kB
38	VmallocChunk:	0 kB
39	Percpu:	2576 kB
40	HardwareCorrupted:	0 kB
41	AnonHugePages:	0 kB
42	ShmemHugePages:	0 kB
43	ShmemPmdMapped:	0 kB
44	FileHugePages:	0 kB
45	FilePmdMapped:	0 kB
46	Unaccepted:	0 kB
47	HugePages_Total:	0
48	HugePages_Free:	0
49	HugePages_Rsvd:	0

Como se puede observar, se cuenta con 6 GB de RAM física que ahora mismo está en un estado de muy baja carga (aprox. 91 % de memoria disponible). No existe presión de memoria, evidenciado por el nulo uso de la partición Swap (0 kB usados). La caché de disco es relativamente baja (340 MB), indicando poca actividad reciente de E/S de ficheros.

## 6.2 Perfilado y traza de un proceso

Para ver y determinar acerca de la ejecución de un programa en nuestro sistema, creamos un fichero en C que se encargue de leer un fichero de entrada, la transforme (en este caso convirtiendo todos sus caracteres en mayúsculas) y escriba el resultado. El programa en C, es el siguiente:

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<unistd.h>
4 #include<ctype.h>
5
6 void lectura_fich(const char *entrada, const char *salida, FILE **f_in, FILE **f_out);
7 void procesa(char *buffer, size_t tamano);
8 char transformar(char c);
9 void tarea(void);
10
11 int main(int argc, char *argv[]){
12     if(argc != 3){
13         fprintf(stderr, "Comando: %s. No se ha proporcionado correctamente el fichero de entrada y de salida\n", argv[0]);
14         return 1;
15     }
16     FILE *f_in = NULL;
17     FILE *f_out = NULL;
18     char buffer[1024];
19     size_t bytes_leidos;
20
21     lectura_fich(argv[1], argv[2], &f_in, &f_out);
22
23     while((bytes_leidos = fread(buffer, 1, 1024, f_in)) > 0){
24         procesa(buffer, bytes_leidos);
25         fwrite(buffer, 1, bytes_leidos, f_out);
26     }
27
28     fclose(f_in);
29     fclose(f_out);
30
31     return 0;
32 }
33
34 void lectura_fich(const char *entrada, const char *salida, FILE **f_in, FILE **f_out){
35     *f_in = fopen(entrada, "r");
36     if (!*f_in){
37         perror("Error al abrir el fichero de entrada");
38         exit(1);
39     }
40     *f_out = fopen(salida, "w");
41     if (!*f_out){
42         perror("Error al abrir el fichero de salida");
43         exit(1);
44     }
45 }
46
47 void procesa(char *buffer, size_t tamano){
48     for(size_t i = 0; i < tamano; i++){
49         buffer[i] = transformar(buffer[i]);
50     }
51 }
52
53 char transformar(char c){
54     tarea();
55
56     if (islower(c)){
57         return toupper(c);
58     }
59     return c;
60 }
```

Ahora, para comprobar su funcionamiento correcto, habría que crear un fichero lo suficientemente grande como para ver el tiempo que tarda en ejecutarse, realizar un perfilado del programa, hacer una traza de llamadas de las llamadas al sistema que hace y determinar las bibliotecas dinámicas que se usan.

- **time:** La ejecución del programa muestra un comportamiento limitado por procesador. De los 28.58 segundos totales de ejecución, 28.54 segundos se gastaron en modo usuario (cálculo puro), mientras que apenas 0.03 segundos se dedicaron a llamadas al sistema (E/S). La eficiencia del uso de CPU fue del 99 %, indicando que el proceso no sufrió bloqueos significativos por espera de disco (I/O Wait), lo cual es consistente con el bajo número de fallos de página (76 minor, 0 major).

```
dansana@dansana:~$ sudo time ./programa entradaGrande.txt salidaEsp.txt
[sudo] password for dansana:
Sorry, try again.
[sudo] password for dansana:
28.54user 0.03system 0:28.58elapsed 99%CPU (0avgtext+0avgdata 1536maxresident)k
0inputs+19544outputs (0major+76minor)pagefaults 0swaps
```

- **gprof**: El análisis de perfilado con **gprof** identifica inequívocamente a la función `tarea` como el punto crítico de rendimiento, consumiendo el 99.58 % del tiempo de CPU (28.13 segundos).

```

1 Flat profile:
2
3 Each sample counts as 0.01 seconds.
4 % cumulative self          self      total
5 time   seconds  seconds  calls  ms/call  ms/call  name
6 99.58    28.13   28.13 10000000    0.00     0.00  tarea
7  0.25    28.20    0.07 10000000    0.00     0.00 transformar
8  0.11    28.23    0.03    9766    0.00     2.89 procesa
9  0.07    28.25    0.02           1    0.00     0.00  _init
10 0.00    28.25    0.00           1    0.00     0.00 lectura_fich
11

```

- **strace**: Se registraron 2,444 llamadas al sistema `read` y 2,443 a `write`. Además, `openat` y `close` reflejan no solo la apertura de los ficheros de datos del usuario, sino también la carga transparente de las bibliotecas dinámicas necesarias.

Las operaciones de escritura consumieron la mayor parte del tiempo de sistema (30.07 %), seguidas por la apertura de ficheros, lo cual es consistente con un programa intensivo en E/S secuencial.

```

dansana@dansana:~$ sudo strace -c ./programa entradaGrande.txt salidaEsp.txt
% time  seconds  usecs/call  calls  errors syscall
----- -----
 32,90  0,001963    392       5  openat
30,07  0,001794      0    2443  write
16,73  0,000998    199       5  close
 9,80  0,000585      0    2444  read
 5,48  0,000327    327       1  execve
 1,59  0,000095     11       8  mmap
 1,11  0,000066      0   2958  rt_sigreturn
 0,59  0,000035     11       3  mprotect
 0,39  0,000023      5       4  fstat
 0,34  0,000020     20       1  munmap
 0,25  0,000015      5       3  brk
 0,13  0,000008      4       2  pread64
 0,12  0,000007      7       1  1 access
 0,08  0,000005      2       2  setitimer
 0,08  0,000005      5       1  rseq
 0,07  0,000004      4       1  set_robust_list
 0,07  0,000004      4       1  prlimit64
 0,07  0,000004      4       1  getrandom
 0,05  0,000003      1       2  rt_sigaction
 0,05  0,000003      3       1  arch_prctl
 0,05  0,000003      3       1  set_tid_address
 0,00  0,000000      0       2  writev
----- -----
100,00  0,005967      0    7890      1 total

```

- **ldd**: El análisis de dependencias dinámicas mediante **ldd** confirma que el binario no es estático, sino que depende de las bibliotecas compartidas del sistema. Para `libc.so.6`, es la biblioteca estándar de C, necesaria para las funciones de E/S (`stdio.h`) y manipulación de caracteres (`ctype.h`). Para `ld-linux-x86-64.so.2`, es el enlazador dinámico responsable de cargar las dependencias en tiempo de ejecución. Para `linux-vdso.so.1` es la biblioteca virtual del kernel para optimizar llamadas al sistema (vDSO) sin cambio de contexto completo.

```

dansana@dansana:~$ sudo ldd ./programa
        linux-vdso.so.1 (0x00007ffc52763000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007721cea00000)
        /lib64/ld-linux-x86-64.so.2 (0x00007721ceceee000)

```

### 6.3 Monitorización

Para terminar con esta parte, vamos a crear un script que actúe como un monitor que guarde información del sistema cada 2 minutos. El script es el siguiente:

```
1#!/bin/bash
2
3# Guardaremos el log
4DIRECTORIO_LOG="/home/dansana/logs_monitorizacion"
5ARCHIVO_LOG="$DIRECTORIO_LOG/monitor_sistema.csv"
6
7# Crear directorio si no existe
8mkdir -p "$DIRECTORIO_LOG"
9
10# Crear cabecera si el archivo es nuevo
11if [ ! -f "$ARCHIVO_LOG" ]; then
12    echo "FECHA,HORA,CPU_USER(%),CPU_IDLE(%),MEM_USADA(MB),MEM_LIBRE(MB),DISCO_USADO(%),CONEXIONES_RED" > "$ARCHIVO_LOG"
13fi
14
15# FECHA y HORA actual
16FECHA=$(date +%Y-%m-%d)
17HORA=$(date +%H:%M:%S)
18
19# CPU: Usamos 'top' en modo batch (-b) una sola vez (-n1).
20# Filtramos la linea 'Cpu(s)', extraemos el % de usuario (us) y % ocioso (id).
21DATOS_CPU=$(top -b1 | grep "Cpu(s)")
22CPU_USER=${DATOS_CPU} | awk -F',' '{print $1}' | awk '{print $2}'
23CPU_IDLE=${DATOS_CPU} | awk -F',' '{print $4}' | awk '{print $1}'
24
25# MEMORIA: Usamos 'free' en Megabytes (-m).
26# Extraemos la fila 'Mem' y sacamos usada ($3) y libre ($4).
27MEM_USADA=$(free -m | grep "Mem:" | awk '{print $3}')
28MEM_LIBRE=$(free -m | grep "Mem:" | awk '{print $4}')
29
30# DISCO: Usamos 'df' para la raíz (/).
31# Extraemos el porcentaje de uso (columna 5).
32DISCO_USO=$(df -h / | tail -1 | awk '{print $5}' | tr -d '%')
33
34# RED: Contamos las conexiones establecidas TCP usando 'ss' o 'netstat'.
35# 'ss -t' (tcp) '-a' (all). Filtramos por ESTAB (establecidas).
36CONEXIONES=$(ss -ta | grep ESTAB | wc -l)
37
38# 3. Escritura en el Log
39# -----
40echo "$FECHA,$HORA,$CPU_USER,$CPU_IDLE,$MEM_USADA,$MEM_LIBRE,$DISCO_USO,$CONEXIONES" >> "$ARCHIVO_LOG"
~
```

Le otorgamos permisos de ejecución como hemos hecho con el resto y como vimos en [3.2.1](#), creamos una tarea que se irá ejecutando cada 2 minutos:

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
4 0 * * * ./home/dansana/script.sh
*/2 * * * * ./home/dansana/monitor.sh
~
```

### 6.3.1 SAR

Se trata de una herramienta clásica de los sistemas UNIX y GNU/Linux, que se utiliza para recopilar, informar y guardar datos de la CPU, memoria y lectura/escritura del disco. El comando **SAR** genera los informes sobre la marcha, además también se puede configurar para guardar dichos informes en ficheros de registro [18].

En mi caso, el comando **SAR** ya venía instalado en mi máquina virtual, por lo que solo es necesario configurarlo para que se ejecute de la forma que piden:

- Lo primero es modificar el fichero de configuración de **SAR** que se encuentra en `/etc/default/sysstat` y activar la opción `ENABLE` que habilita la recolección de datos:

```
#  
# Default settings for /etc/init.d/sysstat, /etc/cron.d/sysstat  
# and /etc/cron.daily/sysstat files  
  
#  
  
# Should sadc collect system activity informations? Valid values  
# are "true" and "false". Please do not put other values, they  
# will be overwritten by debconf!  
ENABLED="true"
```

Una vez guardados los cambios, reiniciamos el servicio:

```
dansana@dansana:~$ sudo systemctl restart sysstat  
dansana@dansana:~$ sudo systemctl enable sysstat  
Synchronizing state of sysstat.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.  
Executing: /usr/lib/systemd/systemd-sysv-install enable sysstat  
dansana@dansana:~$ sudo systemctl status sysstat  
● sysstat.service - Resets System Activity Logs  
    Loaded: loaded (/usr/lib/systemd/system/sysstat.service; enabled; preset: enabled)  
    Active: active (exited) since Mon 2025-11-24 09:40:18 UTC; 15s ago  
      Docs: man:sai(8)  
            man:sadc(8)  
            man:sar(1)  
    Main PID: 24425 (code=exited, status=0/SUCCESS)  
       CPU: 3ms  
  
nov 24 09:40:18 dansana systemd[1]: Starting sysstat.service - Resets System Activity Logs...  
nov 24 09:40:18 dansana systemd[1]: Finished sysstat.service - Resets System Activity Logs.
```

- Despues, ejecutamos el comando `sar -o datos.data 2 300` y esperamos 10 minutos. Para que se vea un poco de actividad, desde otra terminal se puede iniciar sesión y realizar tareas para que queden registradas:

```
dansana@dansana:~$ sar -o datos.data 2 300  
Linux 6.8.0-88-generic (dansana) 24/11/25 _x86_64_ (4 CPU)  
09:44:03 CPU %user %nice %system %iowait %steal %idle  
09:44:05 all 0,13 0,00 0,00 0,00 0,00 99,87  
09:44:07 all 0,00 0,00 0,13 0,13 0,00 99,75  
09:44:09 all 0,00 0,00 0,12 0,25 0,00 99,63  
09:44:11 all 0,00 0,00 0,00 0,00 0,00 100,00  
09:44:13 all 0,00 0,00 0,13 0,00 0,00 99,87  
09:44:15 all 0,00 0,00 0,00 0,00 0,00 100,00  
09:44:17 all 0,00 0,00 0,12 0,00 0,00 99,88  
09:44:19 all 0,00 0,00 0,12 0,00 0,00 99,88  
09:44:21 all 0,00 0,00 0,00 0,00 0,00 100,00  
09:44:23 all 0,00 0,00 0,00 0,00 0,00 100,00  
09:44:25 all 0,12 0,00 0,25 0,00 0,00 99,63  
09:44:27 all 0,38 0,00 0,00 0,00 0,00 99,62  
09:44:29 all 0,00 0,00 0,12 0,00 0,00 99,88  
09:44:31 all 0,00 0,00 0,12 0,00 0,00 99,88  
09:44:33 all 0,50 0,00 0,25 0,12 0,00 99,12  
09:44:35 all 0,00 0,00 0,13 0,00 0,00 99,87  
09:44:37 all 0,00 0,00 0,00 0,00 0,00 100,00  
09:44:39 all 0,00 0,00 0,12 0,12 0,00 99,75  
09:44:41 all 0,00 0,00 0,00 0,00 0,00 100,00  
09:44:43 all 0,00 0,00 0,12 0,00 0,00 99,88
```

- Cuando haya terminado, veremos los resultados con **sar -f** ya que se trata de un archivo binario y no se puede leer con un editor de texto.

- **Uso de CPU:** Podemos ver cómo el sistema principalmente está en estado de reposo, ya que hay cerca de un 100 % de los recursos disponibles (**idle**)

dansana@dansana:~\$ sar -u -f datos.data   head -n 20							
09:44:03	CPU	%user	%nice	%system	%iowait	%steal	%idle
09:44:05	all	0,13	0,00	0,00	0,00	0,00	99,87
09:44:07	all	0,00	0,00	0,13	0,13	0,00	99,75
09:44:09	all	0,00	0,00	0,12	0,25	0,00	99,63
09:44:11	all	0,00	0,00	0,00	0,00	0,00	100,00
09:44:13	all	0,00	0,00	0,13	0,00	0,00	99,87
09:44:15	all	0,00	0,00	0,00	0,00	0,00	100,00
09:44:17	all	0,00	0,00	0,12	0,00	0,00	99,88
09:44:19	all	0,00	0,00	0,12	0,00	0,00	99,88
09:44:21	all	0,00	0,00	0,00	0,00	0,00	100,00
09:44:23	all	0,00	0,00	0,00	0,00	0,00	100,00
09:44:25	all	0,12	0,00	0,25	0,00	0,00	99,63
09:44:27	all	0,38	0,00	0,00	0,00	0,00	99,62
09:44:29	all	0,00	0,00	0,12	0,00	0,00	99,88
09:44:31	all	0,00	0,00	0,12	0,00	0,00	99,88
09:44:33	all	0,50	0,00	0,25	0,12	0,00	99,12
09:44:35	all	0,00	0,00	0,13	0,00	0,00	99,87
09:44:37	all	0,00	0,00	0,00	0,00	0,00	100,00

- **Longitud de la cola de procesos:** Relacionado con el punto anterior, el sistema se encuentra en un estado de inactividad. Esto se puede ver en la carga nula del sistema en ese momento (**ldavg**):

dansana@dansana:~\$ sar -q -f datos.data   head -n 20							
09:44:03	rung-sz	plist-sz	ldavg-1	ldavg-5	ldavg-15	blocked	
09:44:05	0	191	0,00	0,00	0,00	0	
09:44:07	0	191	0,00	0,00	0,00	0	
09:44:09	0	191	0,00	0,00	0,00	0	
09:44:11	1	191	0,00	0,00	0,00	0	
09:44:13	0	191	0,00	0,00	0,00	0	
09:44:15	1	191	0,00	0,00	0,00	0	
09:44:17	1	191	0,00	0,00	0,00	0	
09:44:19	0	191	0,00	0,00	0,00	0	
09:44:21	1	191	0,00	0,00	0,00	0	
09:44:23	1	191	0,00	0,00	0,00	0	
09:44:25	0	191	0,00	0,00	0,00	0	
09:44:27	0	191	0,00	0,00	0,00	0	
09:44:29	0	191	0,00	0,00	0,00	0	
09:44:31	0	191	0,00	0,00	0,00	0	
09:44:33	1	191	0,00	0,00	0,00	0	
09:44:35	0	191	0,00	0,00	0,00	0	
09:44:37	0	191	0,00	0,00	0,00	0	

- **Bloques leídos/escritos en disco:** En este caso, de pueden ver que no hay actividad de lectura en el disco y que hay escritura periódica en forma de ráfagas, coincidiendo con los ciclos de vaciado de buffers del sistema de archivos.

dansana@dansana:~\$ sar -b -f datos.data   head -n 20							
09:44:03	tps	rtpS	wtpS	dtpS	bread/s	bwrtn/s	bdscd/s
09:44:05	0,00	0,00	0,00	0,00	0,00	0,00	0,00
09:44:07	32,50	0,00	32,50	0,00	0,00	344,00	0,00
09:44:09	2,50	0,00	2,50	0,00	0,00	56,00	0,00
09:44:11	0,00	0,00	0,00	0,00	0,00	0,00	0,00
09:44:13	7,50	0,00	7,50	0,00	0,00	80,00	0,00
09:44:15	1,50	0,00	1,50	0,00	0,00	40,00	0,00
09:44:17	0,00	0,00	0,00	0,00	0,00	0,00	0,00
09:44:19	0,00	0,00	0,00	0,00	0,00	0,00	0,00
09:44:21	2,50	0,00	2,50	0,00	0,00	56,00	0,00
09:44:23	0,00	0,00	0,00	0,00	0,00	0,00	0,00
09:44:25	0,50	0,00	0,50	0,00	0,00	4,00	0,00
09:44:27	1,50	0,00	1,50	0,00	0,00	48,00	0,00
09:44:29	0,00	0,00	0,00	0,00	0,00	0,00	0,00
09:44:31	0,00	0,00	0,00	0,00	0,00	0,00	0,00
09:44:33	22,00	0,00	22,00	0,00	0,00	276,00	0,00
09:44:35	0,00	0,00	0,00	0,00	0,00	0,00	0,00
09:44:37	0,00	0,00	0,00	0,00	0,00	0,00	0,00

- **Actividad de Red:** Era de suponer que el sistema al estar inactivo, no hay apenas actividad de red

dansana@dansana:~\$ sar -n DEV -f datos.data   head -n 20								
09:44:03	IFACE	rxpck/s	txpck/s	rxKB/s	txKB/s	rxcmp/s	txcmp/s	rxmlst/s %ifutil
09:44:05	lo	0,00	0,00	0,00	0,00	0,00	0,00	0,00
09:44:05	enp6s18	0,50	0,00	0,03	0,00	0,00	0,00	0,00
09:44:07	lo	0,00	0,00	0,00	0,00	0,00	0,00	0,00
09:44:07	enp6s18	1,00	0,00	0,06	0,00	0,00	0,00	0,00
09:44:09	lo	0,00	0,00	0,00	0,00	0,00	0,00	0,00
09:44:09	enp6s18	2,00	0,00	0,11	0,00	0,00	0,00	0,00
09:44:11	lo	0,00	0,00	0,00	0,00	0,00	0,00	0,00
09:44:11	enp6s18	1,00	0,50	0,19	0,16	0,00	0,00	0,00
09:44:13	lo	0,00	0,00	0,00	0,00	0,00	0,00	0,00
09:44:13	enp6s18	0,00	0,00	0,00	0,00	0,00	0,00	0,00
09:44:15	lo	0,00	0,00	0,00	0,00	0,00	0,00	0,00
09:44:15	enp6s18	1,50	0,50	0,08	0,02	0,00	0,00	0,00
09:44:17	lo	0,00	0,00	0,00	0,00	0,00	0,00	0,00
09:44:17	enp6s18	1,00	0,00	0,14	0,00	0,00	0,00	0,00
09:44:19	lo	0,00	0,00	0,00	0,00	0,00	0,00	0,00
09:44:19	enp6s18	1,00	0,00	0,05	0,00	0,00	0,00	0,00
09:44:21	lo	0,00	0,00	0,00	0,00	0,00	0,00	0,00

---

## Referencias

- [1] Wikipedia. Gnu/linux. [https://es.wikipedia.org/wiki/GNU/Linux#Discos,\\_particiones\\_y\\_sistemas\\_de\\_archivos](https://es.wikipedia.org/wiki/GNU/Linux#Discos,_particiones_y_sistemas_de_archivos), 2025.
- [2] Ubuntu. Guía de escritorio de kubuntu. <https://help.ubuntu.com/kubuntu/desktopguide/es/directories-file-systems.html>, 2006.
- [3] Ubuntu. Guía de escritorio de kubuntu. <https://help.ubuntu.com/kubuntu/desktopguide/es/apt-get.html>, 2006.
- [4] Wikipedia. Raid. <https://es.wikipedia.org/wiki/RAID#>, 2011.
- [5] Unix StackExchange. Where is the rc file for vi editor. <https://unix.stackexchange.com/questions/180710/where-is-the-rc-file-for-vi-editor#:~:text=IEEE%20Std%201003.1%2D2001%20does,the%20root%20partition%20is%20present.>, 2022.
- [6] Joel Barrios Dueñas. Permisos del sistema de archivos en gnu/linux. <https://blog.alcancelibre.org/staticpages/index.php/permisos-sistema-de-archivos>, 2016.
- [7] Wikipedia. Cuota de disco. [https://es.wikipedia.org/wiki/Cuota\\_de\\_disco](https://es.wikipedia.org/wiki/Cuota_de_disco), 2024.
- [8] ArchWiki. Disk quota. [https://wiki.archlinux.org/title/Disk\\_quota](https://wiki.archlinux.org/title/Disk_quota), 2022.
- [9] Git. Git install. <https://git-scm.com/install/linux>, 2025.
- [10] TecnoBits. Qué es tarballs en linux y cómo puedo usar archivos tarballs. <https://tecnobits.com/que-es-tarballs-en-linux-y-como-puedo-usar-archivos-tarballs/>.
- [11] Juan Sánchez Corbalán. ¿Qué es un Shell Script? <https://sanchezcorbalan.es/que-es-un-shell-script/>, 2025.
- [12] Chase Lambert. Makefile tutorial by example. <https://makefiletutorial.com/>.
- [13] GeeksforGeeks. Run levels in linux. <https://www.geeksforgeeks.org/linux-unix/run-levels-linux/>, 2024.
- [14] NGINX, Inc. Nginx documentation. <https://nginx.org/en/docs/>.
- [15] TecnoBloggers. X window: Historia y funciones en sistemas unix. <https://tecnobloggers.com/x-window-historia-y-funciones-en-sistemas-unix/>.
- [16] ArchWiki. Xinit (Español). [https://wiki.archlinux.org/title/Xinit\\_\(Espa%C3%B1ol\)](https://wiki.archlinux.org/title/Xinit_(Espa%C3%B1ol)).
- [17] man7.org. proc(5) — Linux manual page. <https://www.man7.org/linux/man-pages/man5/proc.5.html>, 2024.
- [18] Ocho Bits Hacen Un Byte. sar: informes de cpu, memoria y disco en gnu/linux. <https://www.ochobitshacenunbyte.com/2019/03/12/sar-informes-de-cpu-memoria-y-disco-en-gnu-linux/>, mar 2019.