

# sesion2.pdf



**Blancabril**



**Ingeniería de Servidores**



**3º Grado en Ingeniería Informática**

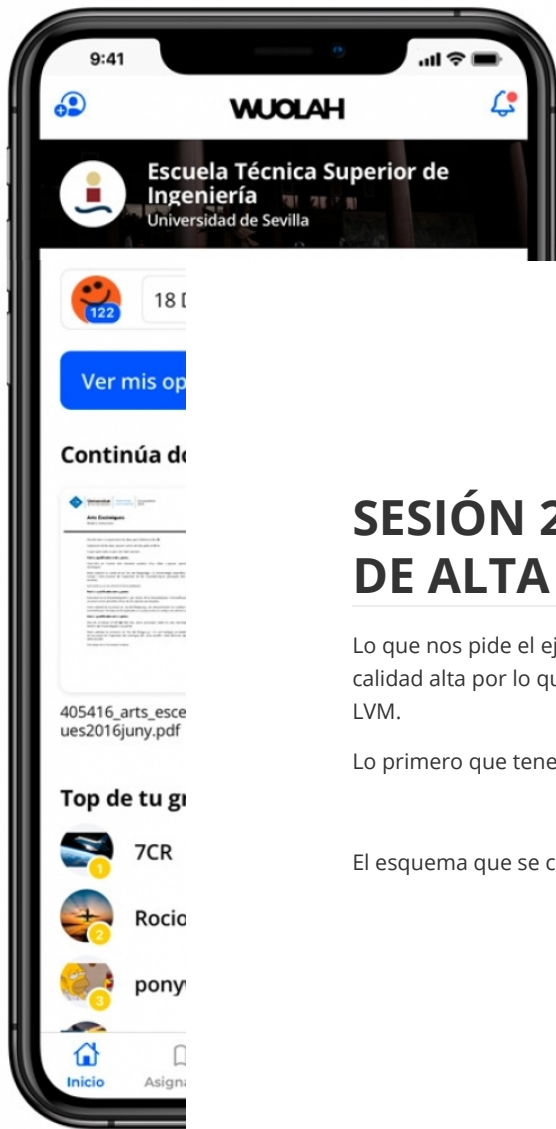


**Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación**  
**Universidad de Granada**



**Descarga la APP de Wuolah.**  
Ya disponible para el móvil y la tablet.





**Descarga la APP de Wuolah.**  
Ya disponible para el móvil y la tablet.

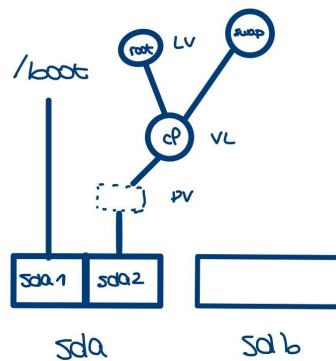


## SESIÓN 2: MAQUINA CON INFORMACIÓN DE ALTA CALIDAD

Lo que nos pide el ejercicio es crear una máquina CentOS que va a alojar vídeos grandes y de calidad alta por lo que /var va a necesitar más espacio y asignarle un LV por lo que crearemos un LVM.

Lo primero que tenemos que hacer es instalar por defecto de CentOS.

El esquema que se crea inicialmente es el siguiente:



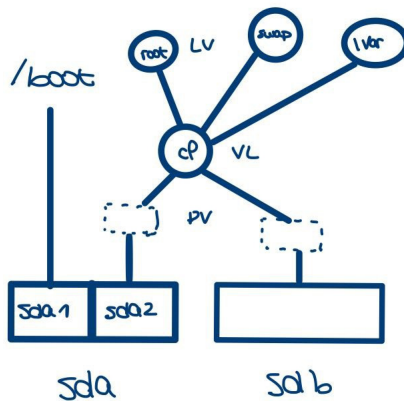
Donde vemos que el sistema operativo nos ha creado directamente la partición en el sda donde ha reservado 1GB para el /boot y el resto para VLM denominado cl. Donde podemos ver un LV para la raíz y otro para swap.

Por otro lado, vemos como sdb no tiene ninguna partición.

Comando para ver los discos **lsblk**.

```
[BlancaAG@localhost ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0   8G  0 disk
├─sda1       8:1    0    1G  0 part /boot
├─sda2       8:2    0    7G  0 part
│   └─cl-root 253:0    0   6.2G  0 lvm /
│       └─cl-swap 253:1    0   820M  0 lvm [SWAP]
sdb          8:16    0    8G  0 disk
sr0         11:0    1 1024M  0 rom
```

Lo que queremos hacer es asociado sdb al VG y crear LV para /var.



## Crear la partición del sdb

Lo que hacemos ahora es **formatear** el disco sdb. Para ello:

```
sudo fdisk /dev/sdb
```

Y tenemos los comandos básicos:

- **m:** para mostrar todas las opciones.
- **p:** para ver todas las particiones.
- **n:** para crear una partición.

Le damos a **n** y después a **p** para crear una **partición primaria**, luego le asignamos el número 1 a la partición y enter a todo. ¡

```
Orden (m para obtener ayuda): n
Tipo de partición
  p  primaria (0 primaria(s), 0 extendida(s), 4 libre(s))
  e  extendida (contenedor para particiones lógicas)
Seleccionar (valor predeterminado p): p
Número de partición (1-4, valor predeterminado 1): 1
Primer sector (2048-16777215, valor predeterminado 2048):
Último sector, +sectores o +tamaño{K,M,G,T,P} (2048-16777215, valor predeterminado 16777215):

Crea una nueva partición 1 de tipo 'Linux' y de tamaño 8 GiB.
```

Comprobamos con **p** que se han creado las particiones correctamente.

Por último, para guardar los cambios: **w**.

```
[BlancaAG@localhost ~]$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0   8G  0 disk
├─sda1       8:1    0   1G  0 part /boot
└─sda2       8:2    0   7G  0 part
   └─cl-root 253:0   0  6.2G  0 lvm  /
      └─cl-swap 253:1   0  820M  0 lvm  [SWAP]
sdb          8:16   0   8G  0 disk
└─sdb1       8:17   0   8G  0 part
sr0         11:0    1 1024M  0 rom
```



**KEEP  
CALM  
AND  
ESTUDIA  
UN POQUITO**

## Creamos el PV

Para ver los volúmenes físicos que hay creados lo hacemos con:

```
sudo pvdisplay
```

Y como podemos ver en la siguiente captura, hay uno creado en el sda2:

```
[BlancaAG@localhost ~]$ sudo pvdisplay
[sudo] password for BlancaAG:
--- Physical volume ---
PU Name                /dev/sda2
UG Name                cl
PU Size                <7,00 GiB / not usable 3,00 MiB
Allocatable            yes (but full)
PE Size                4,00 MiB
Total PE               1791
Free PE                0
Allocated PE           1791
PU UUID                aPQWRv-Ix5k-dNPd-ABBR-yR1A-Tg0c-zhCpbz
```

Para crear un volumen físico para el sdb1 y así unirlo a grupo de volúmenes, tenemos que ejecutar el siguiente comando:

```
sudo pvcreate /dev/sdb1
```

Y comprobamos que esté creado con

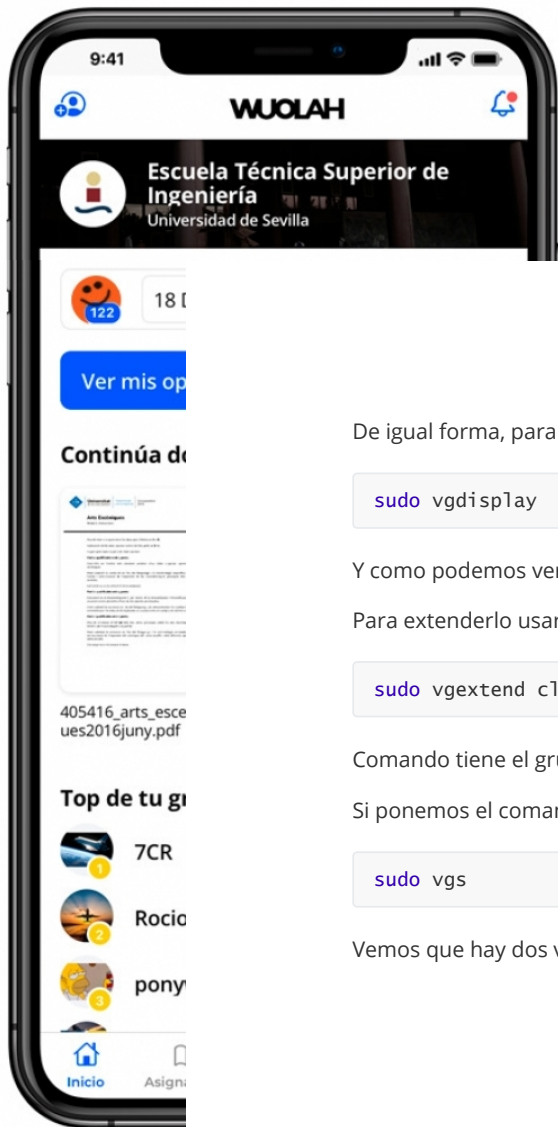
```
sudo pvdisplay
```

```
[BlancaAG@localhost ~]$ sudo pvdisplay
--- Physical volume ---
PU Name                /dev/sda2
UG Name                cl
PU Size                <7,00 GiB / not usable 3,00 MiB
Allocatable            yes (but full)
PE Size                4,00 MiB
Total PE               1791
Free PE                0
Allocated PE           1791
PU UUID                aPQWRv-Ix5k-dNPd-ABBR-yR1A-Tg0c-zhCpbz

"/dev/sdb1" is a new physical volume of "<8,00 GiB"
--- NEW Physical volume ---
PU Name                /dev/sdb1
UG Name
PU Size                <8,00 GiB
Allocatable            NO
PE Size                0
Total PE               0
Free PE                0
Allocated PE           0
PU UUID                y0j508-yR6P-a33E-sKLZ-wNzs-jw5f-ioZRQ6
```

Una vez hecho esto, ya tenemos el volumen físico creado.

## Unimos el volumen físico al grupo de volúmenes



# Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



De igual forma, para ver el grupo de volúmenes, usamos el comando:

```
sudo vgdisplay
```

Y como podemos ver hay un grupo de volúmenes denominado **cl**.

Para extenderlo usaremos:

```
sudo vgextend cl /dev/sdb1
```

Comando tiene el grupo de volúmenes donde queremos añadir el volumen físico.

Si ponemos el comando

```
sudo vgs
```

Vemos que hay dos volúmenes físicos.

```
[BlancaAG@localhost ~]$ sudo vgs
VG #PU #LV #SN Attr   USize  UFree
cl  2   2   0 wz--n- 14,99g <8,00g
[BlancaAG@localhost ~]$
```

## Crear volumen lógico

Ahora, para terminar la configuración, vamos a crear el volumen lógico para el /var.

Para ello con el siguiente comando podemos ver los volúmenes lógicos que hay:

```
sudo lvdisplay
```

```
[BlancaAG@localhost ~]$ sudo lvdisplay
--- Logical volume ---
LV Path                /dev/cl/swap
LV Name                 swap
VG Name                 cl
LV UUID                P9DPXE-rDTj-0ZgE-UZYD-ykyw-RCii-JZ2hMy
LV Write Access         read/write
LV Creation host, time localhost, 2021-10-11 11:49:17 -0400
LV Status                available
# open                  2
LV Size                 820,00 MiB
Current LE              205
Segments                1
Allocation              inherit
Read ahead sectors      auto
- currently set to     8192
Block device            253:1

--- Logical volume ---
LV Path                /dev/cl/root
LV Name                 root
VG Name                 cl
LV UUID                C5kNbv-cZ9Q-3tA3-zsBs-TEjy-d7XZ-lYAavz
LV Write Access         read/write
LV Creation host, time localhost, 2021-10-11 11:49:18 -0400
LV Status                available
# open                  1
LV Size                 <6,20 GiB
Current LE              1586
Segments                1
Allocation              inherit
Read ahead sectors      auto
- currently set to     8192
Block device            253:0
```

Como podemos ver, hay dos, el swap y raíz.

Y ahora con el siguiente comando le indicamos el nombre con -n, la longitud con -L y el grupo de volúmenes al que queremos que se asigne.

```
sudo lvcreate -n new_var -L 3G cl
```

```
--- Logical volume ---
LV Path                /dev/cl/new_var
LV Name                new_var
VG Name                cl
LV UUID                eACli0-7xxQ-ZSC1-hRBj-luhH-LOiQ-5E1efw
LV Write Access        read/write
LV Creation host, time localhost.localdomain, 2021-10-11 17:05:35 -0400
LV Status              available
# open                 0
LV Size                3,00 GiB
Current LE             768
Segments               1
Allocation             inherit
Read ahead sectors     auto
- currently set to    8192
Block device           253:2
```

La parte de configuración del LVM ya está configurada.

Pasamos a:

1. Montar y darle formato
2. Copiar la información del /var al LV de forma atómica
3. Indicar al sistema operativo donde está /var
4. Liberar espacio

## Montar y darle formato

Lo primero que hacemos es **crear un punto de montaje** para /var a la que llamaremos **new\_var**.

```
sudo mkdir /new_var
```

Ahora creamos un **sistema de archivos** dentro del volumen lógico de new\_var y con -t le decimos el tipo y posteriormente ponemos el lugar donde lo vamos a poner, en nuestro caso donde se sitúa el LV.

```
sudo mkfs -t ext4 /dev/cl/new_var
```

Una vez está creado el sistema de archivos, vamos a montarlo y de esta manera se hace accesible al sistema operativo.

```
sudo mount /dev/cl/new_var /new_var/
```

Y comprobamos con el comando **mount** que está montado.

```
mount
```

```
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,devices)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,memory)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,seclabel,blkio)
configfs on /sys/kernel/config type configfs (rw,relatime)
/dev/mapper/cl-root on / type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
selinuxfs on /sys/fs/selinux type selinuxfs (rw,relatime)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=34,pgrp=1,timeout=0,minproto=5,max
proto=5,direct,pipe_ino=17870)
debugfs on /sys/kernel/debug type debugfs (rw,relatime,seclabel)
mqueue on /dev/mqueue type mqueue (rw,relatime,seclabel)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,seclabel,pagesize=2M)
/dev/sda1 on /boot type ext4 (rw,relatime,seclabel)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=83844k,mode=700,uid=1000,
gid=1000)
/dev/mapper/cl-new_var on /new_var type ext4 (rw,relatime,seclabel)
```

## Copiar datos

Como queremos que sea atómica, es que se copia todo sin ninguna modificación. Para ello, evitamos a los usuarios acceder. Para hacer esto tendremos que entrar en modo mantenimiento.

```
sudo systemctl isolate rescue
```

Lo que ocurre es que se produce un bug y nos saca, así que **iniciamos sesión con root** y volvemos a poner el comando.

```
sudo systemctl isolate rescue
```

Y si hacemos:

```
systemctl status
```

Vemos que estamos en modo mantenimiento.

```
[root@localhost ~]# systemctl status
● localhost.localdomain
  State: maintenance
    Jobs: 0 queued
  Failed: 0 units
   Since: Tue 2021-10-12 05:01:55 EDT; 13min ago
   CGroup: /
          └─init.scope
              └─1 /usr/lib/systemd/systemd --switched-root --system --deserialize 16
                 system.slice
                     └─rngd.service
                         └─841 /sbin/rngd -f --fill-watermark=0
                            systemd-udevd.service
                                └─698 /usr/lib/systemd/systemd-udevd
                                   systemd-journald.service
                                       └─672 /usr/lib/systemd/systemd-journald
                                          rescue.service
                                              └─2294 /usr/lib/systemd/systemd-sulogin-shell rescue
                                                 └─2295 bash
                                                    └─2313 systemctl status
                                                       └─2314 (pager)
[root@localhost ~]# _
```

Ahora pasamos a copiar los datos, para preservar todos los archivos se usa la extensión **-a**.

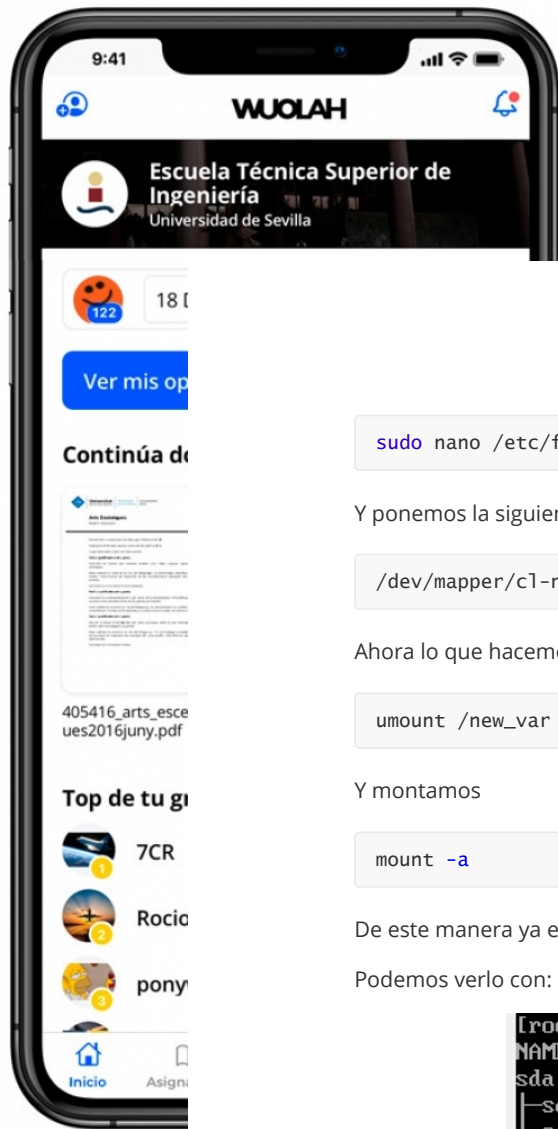
```
cp -a /var/. /new_var/
```

Una vez que hemos copiado la información, asignamos el punto de montaje a /new\_var para /var

## Punto de montaje de /var

Para ello accedemos al archivo **/etc/fstab**.





# Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



```
sudo nano /etc/fstab
```

Y ponemos la siguiente línea en el archivo:

```
/dev/mapper/cl-new_var /var ext4 defaults 0 0
```

Ahora lo que hacemos es

```
umount /new_var
```

Y montamos

```
mount -a
```

De este manera ya el /var estará montado según el archivo de configuración.

Podemos verlo con:

```
[root@localhost ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0   8G  0 disk
├─sda1       8:1    0   1G  0 part /boot
├─sda2       8:2    0   7G  0 part
├─┬─cl-root 253:0    0  6,2G  0 lvm /
├─┬─cl-swap 253:1    0  820M  0 lvm [SWAP]
└─sdb        8:16    0   8G  0 disk
   └─sdb1     8:17    0   8G  0 part
      └─cl-new_var 253:2    0   3G  0 lvm /var
sr0         11:0    1 1024M  0 rom
```

## Liberar espacio

Para ello, primero tenemos que ir marcha atrás porque el /var original está ocupando espacio.

Para ello entramos en fstab y comentamos la línea anteriormente puesta y hacemos

```
mount -a
```

```
[root@localhost ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0   8G  0 disk
├─sda1       8:1    0   1G  0 part /boot
├─sda2       8:2    0   7G  0 part
├─┬─cl-root 253:0    0  6,2G  0 lvm /
├─┬─cl-swap 253:1    0  820M  0 lvm [SWAP]
└─sdb        8:16    0   8G  0 disk
   └─sdb1     8:17    0   8G  0 part
      └─cl-new_var 253:2    0   3G  0 lvm /var
sr0         11:0    1 1024M  0 rom
```

Como podemos ver, aún sigue montado así que habría que desmontarlo manualmente.

```
umount /dev/cl/new_var
```

```
[root@localhost ~]# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                  8:0    0   8G  0 disk
├─sda1                8:1    0   1G  0 part /boot
├─sda2                8:2    0   7G  0 part
│   └─cl-root         253:0    0  6,2G  0 lvm  /
│       └─cl-swap      253:1    0  820M  0 lvm  [SWAP]
sdb                  8:16    0   8G  0 disk
├─sdb1                8:17    0   8G  0 part
│   └─cl-new_var      253:2    0   3G  0 lvm
sr0                  11:0    1 1024M  0 rom
```

Una vez hecho esto, vamos a mover /var a otra carpeta, /var\_old. De esta manera, si hay algún problema, podremos recuperar la información.

```
mv /var /var_old
```

Y si llamáramos a mount -a con la línea descomentada, nos diría que no existe /var así que cremamos la carpeta.

```
mkdir /var
```

Y para que /var sea el nuevo, tenemos que hacer

```
restorecon /var
```

Y ahora montamos de nuevo con:

```
mount -a
```

```
[root@localhost ~]# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                  8:0    0   8G  0 disk
├─sda1                8:1    0   1G  0 part /boot
├─sda2                8:2    0   7G  0 part
│   └─cl-root         253:0    0  6,2G  0 lvm  /
│       └─cl-swap      253:1    0  820M  0 lvm  [SWAP]
sdb                  8:16    0   8G  0 disk
├─sdb1                8:17    0   8G  0 part
│   └─cl-new_var      253:2    0   3G  0 lvm  /var
sr0                  11:0    1 1024M  0 rom
[root@localhost ~]#
```