

[FORT] Práctica 2: Reforzando el sistema Operativo

Notas

Establece Cuotas en los directorios /Home de los usuarios

Estas cuotas deben mantenerse activas tras un reinicio.

- Ningún usuario puede usar más de 20Mb por 10 días
- Ningún usuario puede usar más de 23Mb
- user018 y user019 juntos no pueden tener más de 50 archivos

Para establecer una cuota de 20Mb cada 10 días para todos los usuario primero editamos el fichero “/etc/fstab” con:

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

Debemos editar la línea donde se encuentra /home añadiendo “usrquota” y “grpquota” después de defaults:

```
GNU nano 7.2 /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).
#
# systemd generates mount units based on this file, see systemd.mount(5).
# Please run 'systemctl daemon-reload' after making changes here.
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
UUID=c8f90164-6a53-4c62-849f-2f06ac8b9f9c / ext4 errors=remount-ro 0 1
# /home was on /dev/sda3 during installation
UUID=ea229802-cc4d-450a-8a85-4864b28f767f /home ext4 defaults,usrquota,grpquota 0 2
# swap was on /dev/sda2 during installation
UUID=e3e37fb5-b23a-4049-b22d-75be2d487aaa none swap sw 0 0
/dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0

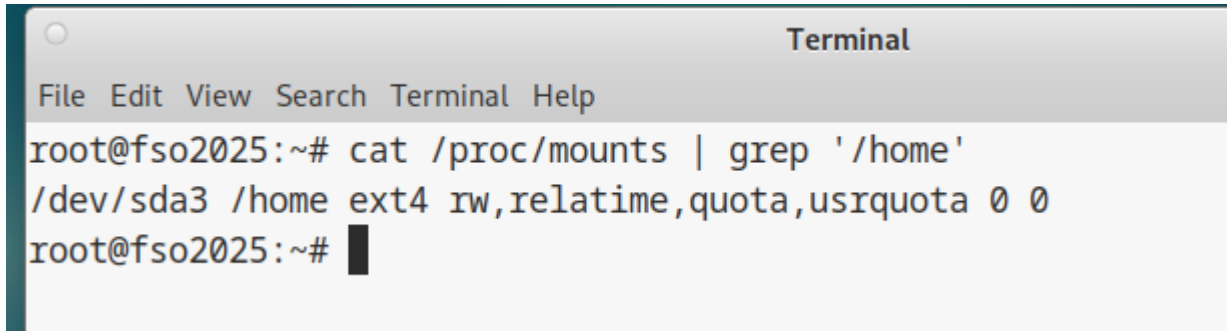
[ Read 17 lines ]
^G Help ^O Write Out ^W Where Is ^K Cut [ Read 17 lines ] ^C Location M-U Undo M-A Set Mark M-] To Bracket
^X Exit ^R Read File ^L Replace ^U Paste ^T Execute ^_ Go To Line M-E Redo M-G Copy ^Q Where Was
```

Una vez hecho eso debemos volver a montar el sistema de ficheros y reiniciamos la máquina:

```
systemctl daemon-reload
sudo mount -o remount /home
reboot
```

Tras el reinicio revisamos que se hayan actualizado los cambios correctamente con:

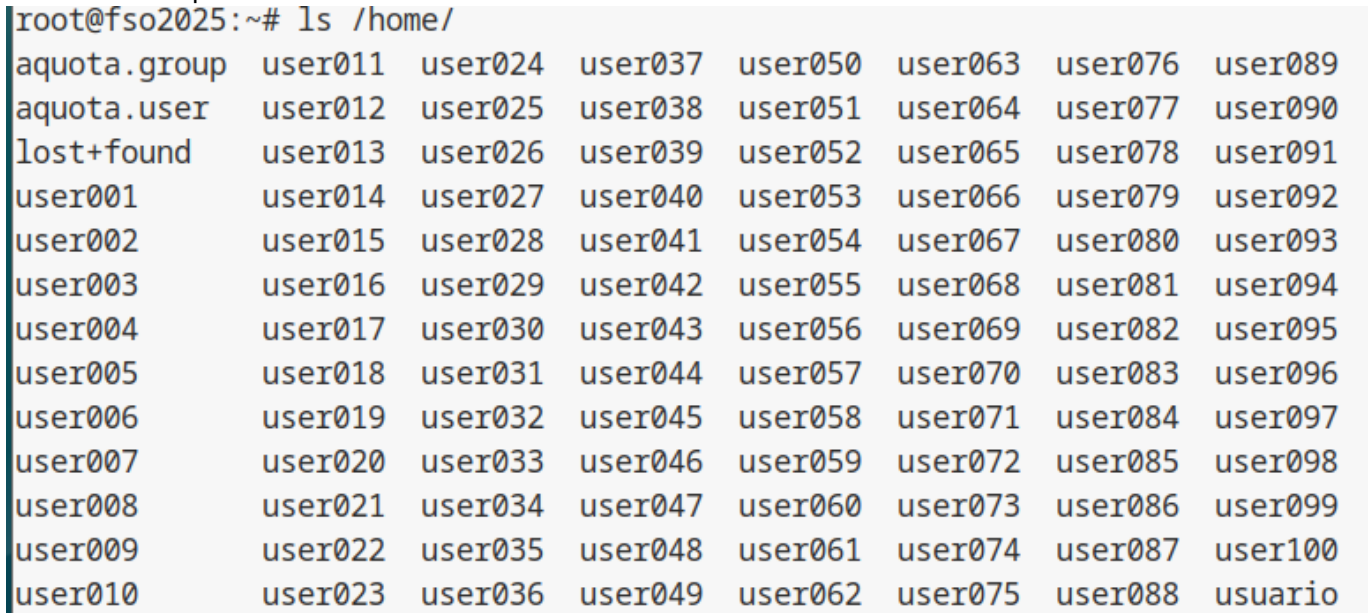
```
cat /proc/mounts | grep '/home'
```



Tras eso procedemos a crear las cuotas mediante el uso de quotacheck, para comenzar ejecutamos el siguiente comando:

```
sudo quotacheck -ugm /home
```

Con este comando se deberían crear los archivos aquota.user y aquota.group en /home, estos ficheros son los que contienen los datos sobre los límites que se aplican al sistema de ficheros. Podemos comprobarlos con un ls a /home



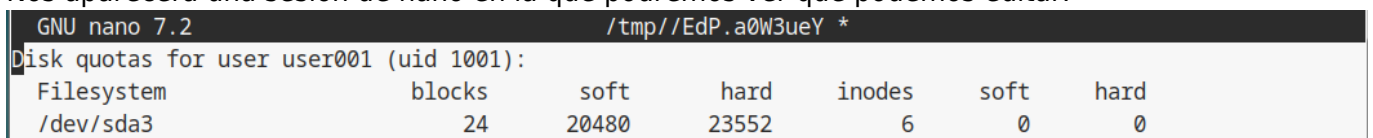
Para establecer la cuota de cada usuario se debe usar el comando setquota:

```
sudo setquota -u user001 20M 23M 0 0 /home
```

También puede ser editada manualmente con el comando:

```
sudo edquota -u user001
```

Nos aparecerá una sesión de nano en la que podremos ver que podemos editar:



Para establecer el límite de 10 días usamos el comando setquota con el flag t:

```
sudo setquota -t 864000 864000 /home
```

Podemos comprobar si esta configuración se ha realizado correctamente con el comando repquota:

```
sudo repquota -s /home
```

```
root@fso2025:~# repquota -s /home
*** Report for user quotas on device /dev/sda3
Block grace time: 10days; Inode grace time: 10days
```

Finalmente, para limitar cuántos archivos tienen los usuarios user018 y user019 tenemos que modificarlos con:

```
sudo edquota -u user018
```

Y modificando la parte del inodes para indicar cuántos archivos pueden tener máximos:

```
GNU nano 7.2 /tmp//EdP.aIPvUiv *
Disk quotas for user user018 (uid 1018):
  Filesystem            blocks      soft      hard    inodes     soft     hard
  /dev/sda3              24        20480    23552      50         50      50
```

Usando ACLs permite a user001 y user002 leer y modificar /boot/grub/grub.cfg

- Permitir a miembros del grupo password modificar /etc/shadow.

Para comenzar vamos a configurar las ACL para ello primero debemos revisar si las ACL están habilitadas con el comando:

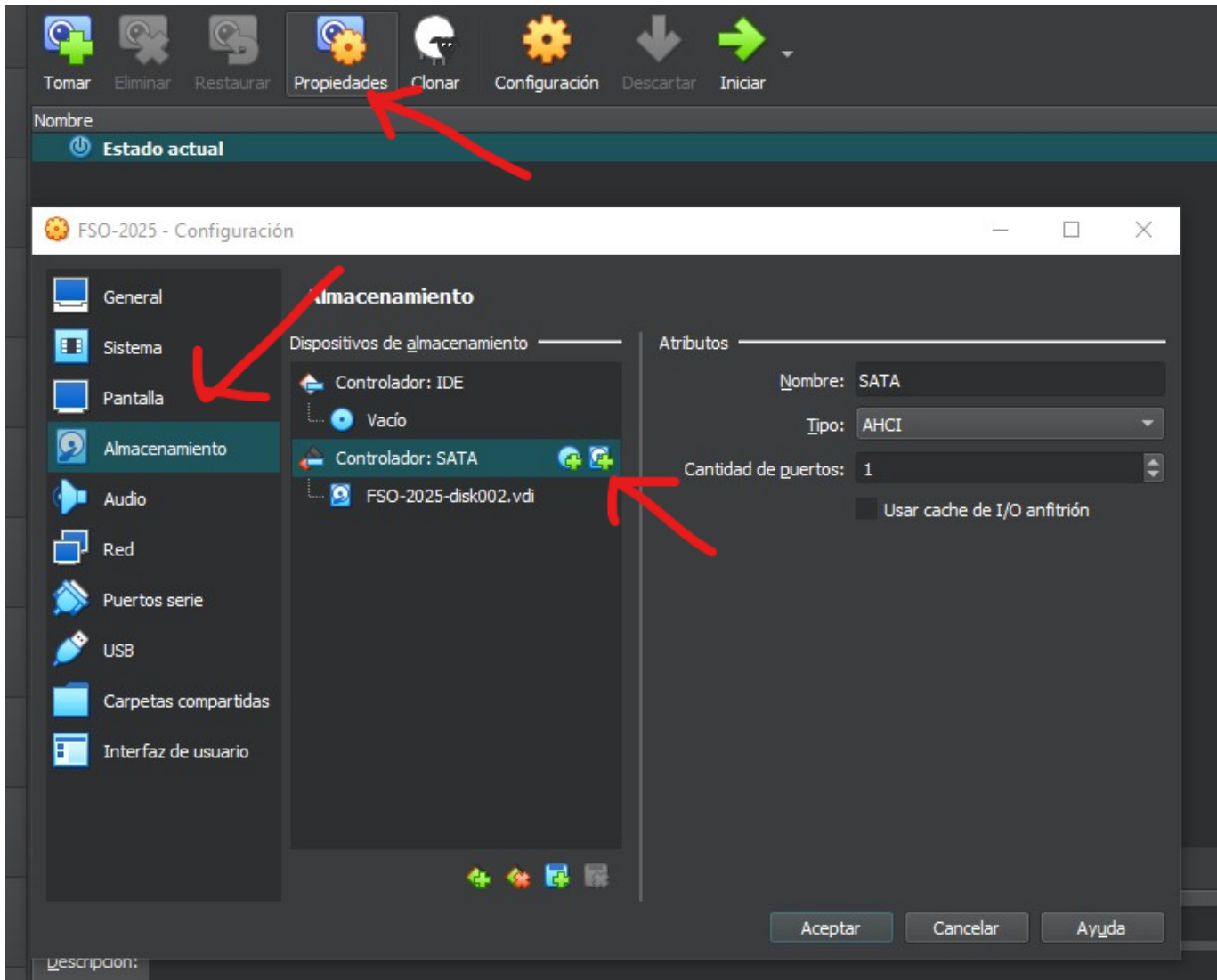
```
sudo tune2fs -l /boot/grub/grub.cfg | grep "Default mount options"
```

Si las ACL están habilitadas podemos proceder a configurarlas, para dar los permisos necesarios a estos usuarios usamos los comandos:

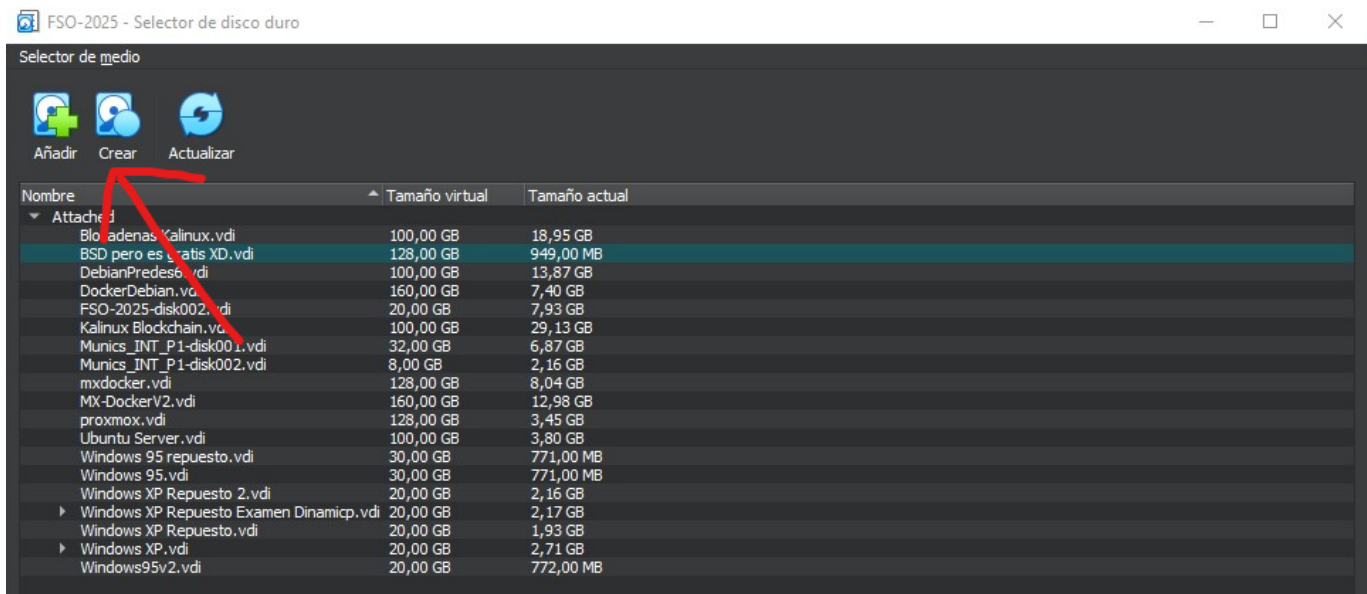
```
sudo setfacl -m u:user001:rw /boot/grub/grub.cfg
sudo setfacl -m u:user002:rw /boot/grub/grub.cfg
```

Añadir un segundo disco a la máquina y particionarlo usando GPT

Para esto, primero procederemos a añadir un disco duro virtual a Virtual Box, primero iremos a configuración de la máquina virtual/almacenamiento y presionaremos en el botón +:

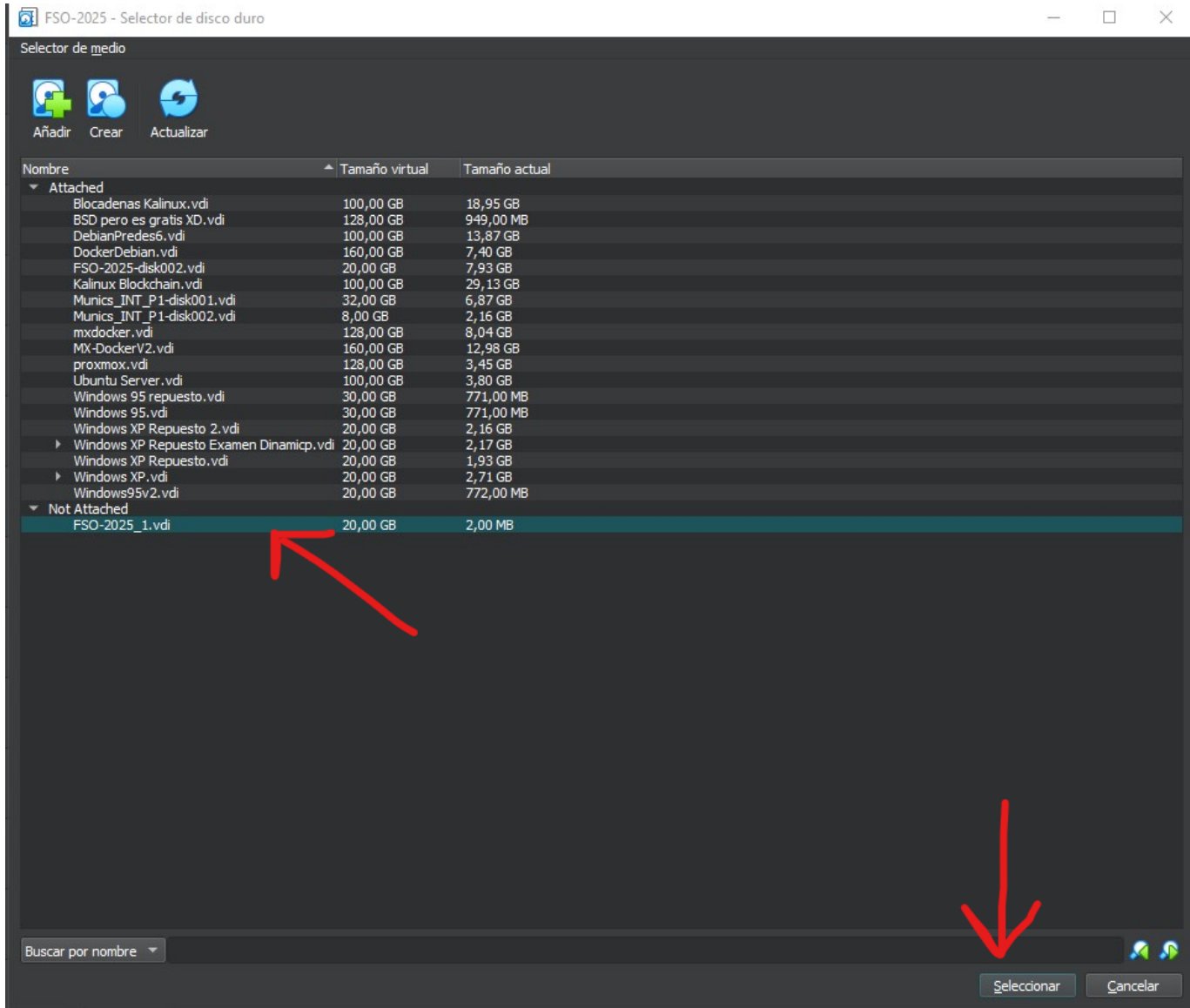


En la ventana que saldrá pulsaremos en crear:





Una vez creado el nuevo disco duro virtual lo seleccionamos u lo creamos:



Tras eso volvemos a arrancar la máquina y abrimos una terminal donde escribiremos el comando "lsblk" para ver los medios de almacenamiento disponibles:

```
lsblk
```

```
root@fso2025:~# lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda        8:0    0   20G  0 disk
├─sda1     8:1    0   9.3G  0 part /
├─sda2     8:2    0   1.9G  0 part [SWAP]
└─sda3     8:3    0   6.7G  0 part /home
sdb        8:16   0   20G  0 disk
sr0       11:0    1 1024M  0 rom
```

En este caso podemos ver que el disco que hemos introducido aparece como "sdb". Ahora podemos proceder a particionarlo con el comando parted:

```
parted /dev/sdb mklabel gpt
parted -a opt /dev/sdb mkpart primary ext4 0% 100%
```

```
root@fso2025:~# parted /dev/sdb mklabel gpt
Information: You may need to update /etc/fstab.
```

```
root@fso2025:~# parted -a opt /dev/sdb mkpart primary ext4 0% 100%
Information: You may need to update /etc/fstab.
```

Finalmente creamos el sistema de archivos con los siguientes comandos:

```
mkfs.ext4 -L datapartition /dev/sdb1
```

Finalmente procedemos a montarlo con el comando:

```
mkdir -p /mnt/data
mount -o defaults /dev/sdb1 /mnt/data
```

Como resultado nuestro equipo quedará de la siguiente forma:

```
root@fso2025:~# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    0   20G  0 disk
├─sda1       8:1    0   9.3G  0 part /
├─sda2       8:2    0   1.9G  0 part [SWAP]
└─sda3       8:3    0   6.7G  0 part /home
sdb          8:16   0   20G  0 disk
└─sdb1       8:17   0   20G  0 part
sr0         11:0    1 1024M  0 rom
```

Crea dos particiones con el espacio restante disponible en el segundo disco

Basándonos en el tamaño de sda1, sda2 y sda3, quedan aproximadamente 2 GB libres, por lo que se crearán dos particiones de 1GB cada una, a las que llamaremos sda4 y sda4. Para crear las particiones usaremos el comando:

```
sudo fdisk /dev/sda
```

Tras ejecutar el comando nos aparecerá la utilidad fdisk:

```
root@fso2025:~# sudo fdisk /dev/sda

Welcome to fdisk (util-linux 2.38.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

This disk is currently in use - repartitioning is probably a bad idea.
It's recommended to umount all file systems, and swapoff all swap
partitions on this disk.

Command (m for help): █
```

Para ver que espacio está disponible presionamos F y le damos a enter:

```
Command (m for help): F

Unpartitioned space /dev/sda: 2.15 GiB, 2306867200 bytes, 4505600 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes

   Start      End Sectors  Size
37437440 41943039 4505600  2.1G

Command (m for help): █
```

From:
<https://knoppia.net/> - **Knoppia**

Permanent link:
https://knoppia.net/doku.php?id=master_cs:fortificacion:p2&rev=1739290197

Last update: **2025/02/11 16:09**

