

# Gestión de Volúmenes Lógicos (V 2.0)

Administración del Almacenamiento en Disco  
*Administración Avanzada de Sistemas Operativos*

Depto. de Arquitectura de Computadores  
Universidad de Málaga  
© Guillermo Pérez Trabado 2008

*Esta obra está bajo una licencia Reconocimiento-No comercial-Compartir bajo la misma licencia 3.0 España de Creative Commons.*

*Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-nc-sa/3.0/es/> o envíe una carta a Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.*





# Discos, particiones y devices en Linux

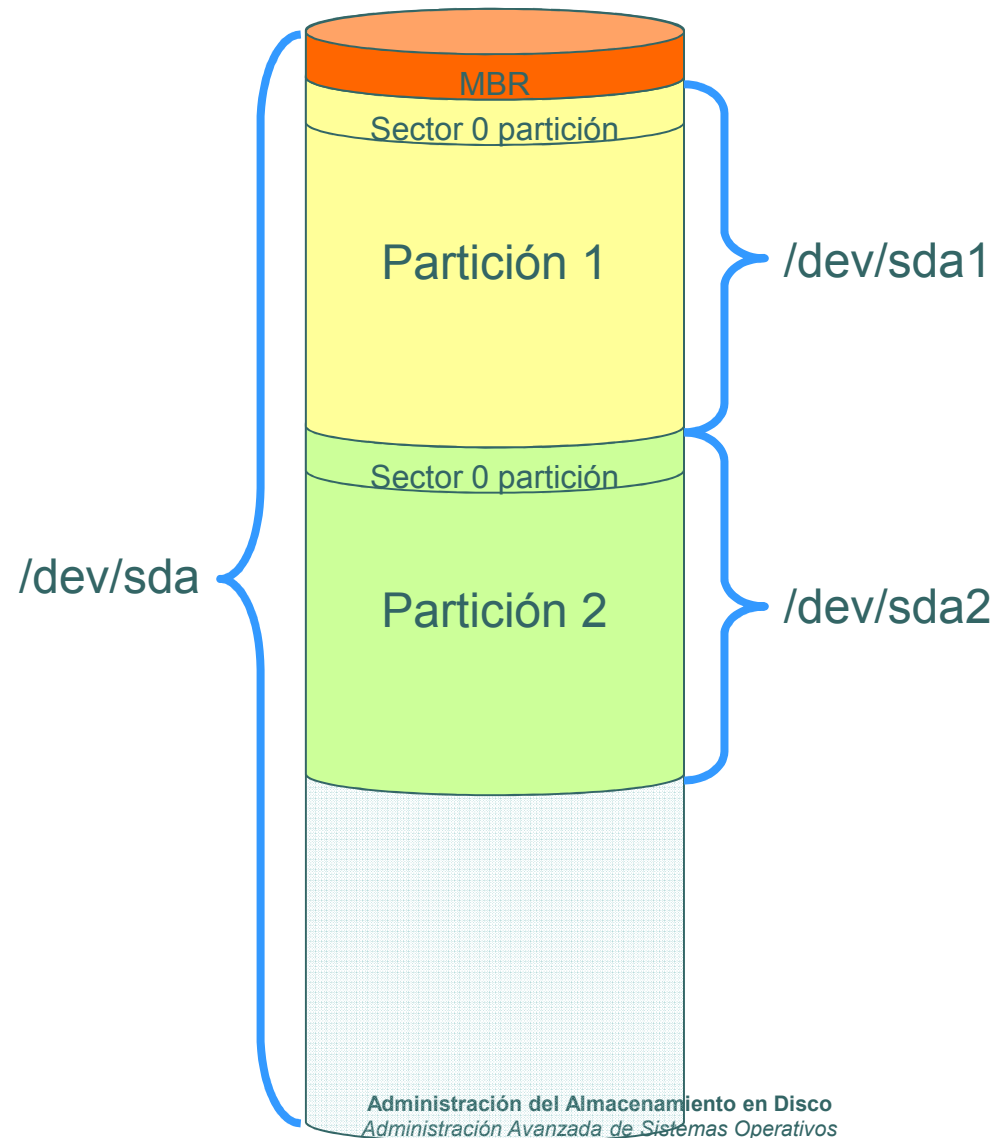
# ● ● ● | Discos y devices

- Cuando los drivers del kernel detectan un dispositivo de almacenamiento se le asigna un nombre simbólico (device) bajo el directorio /dev para poder referirse a él.
  - Esto sucede tanto en el arranque como cuando insertamos un dispositivo “hot plug-in” (como por ejemplo los discos USB externos). Ejemplo:
    - /dev/sda (primer disco SCSI o SATA)
    - /dev/sdb (segundo disco SCSI)
    - /dev/hda (primer disco IDE)
    - Etc...
- Acceder a un device de disco implica acceder a los bloques del disco que representa. Por ejemplo:  
`dd if=/dev/zero of=/dev/sda bs=1M count=1000`
  - Este comando escribe 1000 bloques de 1Mbyte con ceros sobre el primer disco empezando por su sector 0. Si es un sistema en uso, esto representa un desastre ya que borra no solo la tabla de particiones del disco sino también el contenido del primer Gbyte del disco seguramente arrasando la primera partición.

# ● ● ● | Particiones y devices

- Después de la detección de un nuevo device de disco, el kernel trata de obtener su tabla de particiones.
- Si tiene una tabla válida se crea un device para cada partición:
  - /dev/sda1 representa la primera partición del primer disco.
  - /dev/sda2 representa la segunda partición del primer disco
  - Etc...
  - /dev/sdb1 (primera partición segundo disco)
  - /dev/sdb2 (segunda partición segundo disco)
  - Etc...
- Acceder a un device de una partición implica acceder a los bloques de la partición que representa. Por ejemplo:  
`dd if=/dev/zero of=/dev/sda1 bs=1M count=1000`
  - Este comando escribe 1000 bloques de 1Mbyte con ceros sobre la primera partición del primer disco empezando por su sector 0. Borrará el contenido del primer Gbyte del disco seguramente arrasando la primera partición. Si la primera partición tuviera menos de 1Gbyte, el comando pararía antes de tocar la segunda partición ya que /dev/sda1 solo representa los bloques de la primera partición.

# ● ● ● | Particiones y devices



# ● ● ● | Más sobre particiones

- Existen innumerables formatos de tablas de particiones resultado de la historia de los sistemas operativos.
  - El más extendido todavía en los PCs es el usado por la BIOS (llamado esquema de particiones MBR o PC BIOS).
  - En los servidores UNIX ha habido decenas de formatos. Actualmente en los servidores Linux se usa por defecto el MBR.
  - Los servidores con Itanium usan el GUID de Intel (parte del estándar EFI que reemplaza la BIOS).
- Soporte:
  - Windows soporta el esquema PC-BIOS.
  - Linux soporta PC-BIOS, GUID y es capaz de leer casi una decena de formatos antiguos de otros UNIX. También es capaz de formatear con los principales que quedan en uso (IRIX, Sun Disklabel, UFS).

# ● ● ● | Sistemas de ficheros

- Aunque las particiones tienen tipos asignados, el formato del sistema de ficheros es independiente del particionado. Linux soporta no solo su propio formato de sistema de ficheros (EXT3) sino también decenas de otros por compatibilidad (XFS, Sun, UFS, Minix, FAT16, FAT32, NTFS).
  - *Nota: Reiser rivalizó con EXT3, pero está prácticamente en vía muerta. No se aconseja usarlo ya que no hay soporte.*
- Una vez creada una partición se le da formato con el comando correspondiente (mkfs.ext3, mkfs.xfs, mkfs.ufs, mkfs.reiserfs, mkfs.vfat, etc).

```
mkfs.ext3 /dev/sdb1
```

  - Crea un sistema de ficheros vacío en la partición /dev/sdb1 usando el formato EXT3.
- El acceso a los ficheros requiere asociar el device de la partición con un punto de montaje y un tipo de sistema de ficheros (comando mount).

```
mount -t ext3 /dev/sdb1 /usuarios
```

  - Indica que la partición /dev/sdb1 debe contener un sistema de ficheros en formato EXT3 y que su árbol de directorios será accesible debajo del directorio /usuarios.
- Las particiones de swap también requieren formato previo:

```
mkswap /dev/sdb2  
swapon /dev/sdb2
```
- Los formatos se identifican por números únicos de forma que no es posible confundir al kernel y malinterpretar una partición con un sistema de ficheros equivocado.



# Gestores de Volúmenes Lógicos

- Los sistemas de ficheros ubicados en particiones tienen muchas limitaciones cuando se llenan de datos:
  - Todos los bloques deben estar contiguos físicamente en el disco.
  - No pueden extenderse a más de un disco.
  - Normalmente un cambio de tamaño requiere borrar esta u otras particiones copiando los datos en otro sitio mientras.
- La respuesta es usar una capa que abstraiga el sistema de ficheros de las particiones físicas en las que está ubicado.
  - Las particiones generadas se denominan “Volúmenes Lógicos”
  - La capa se denomina “Gestor de Volúmenes Lógicos”.



# ● ● ● | Conceptos básicos

- Cada elemento de almacenamiento gestionado se denomina un “Volumen Físico” o “Physical Volume” (PV).
  - Pueden ser discos completos o particiones de discos indistintamente.
- Los PV se agrupan en “Grupos de Volúmenes” o “Volume Groups” (VG), que actúan como bolsas flexibles de recursos de almacenamiento. Los PV se pueden añadir o retirar de los VG cuando haga falta.
- Los VG se particionan en “Volúmenes Lógicos” o “Logical Volumes” (LV) cuyo tamaño tan solo está limitado por el espacio no asignado que está disponible en un VG.
- El S.O. asigna un device al LV para acceder a él de forma idéntica al device de una partición. Ejemplo:

```
mkfs.ext3 /dev/VolGroup_DatosUsuarios/LogVol_Usuarios1
mount /dev/VolGroup_DatosUsuarios/LogVol_Usuarios1
      /usuarios1
```
- Un VG agrupa varios PVs y puede tener definidos uno o más LV.
- Si un LV necesita crecer, se pueden añadir PVs al VG y para añadir espacio libre y redimensionar el LV sin perder su contenido.

# ● ● ● | LVM en Linux

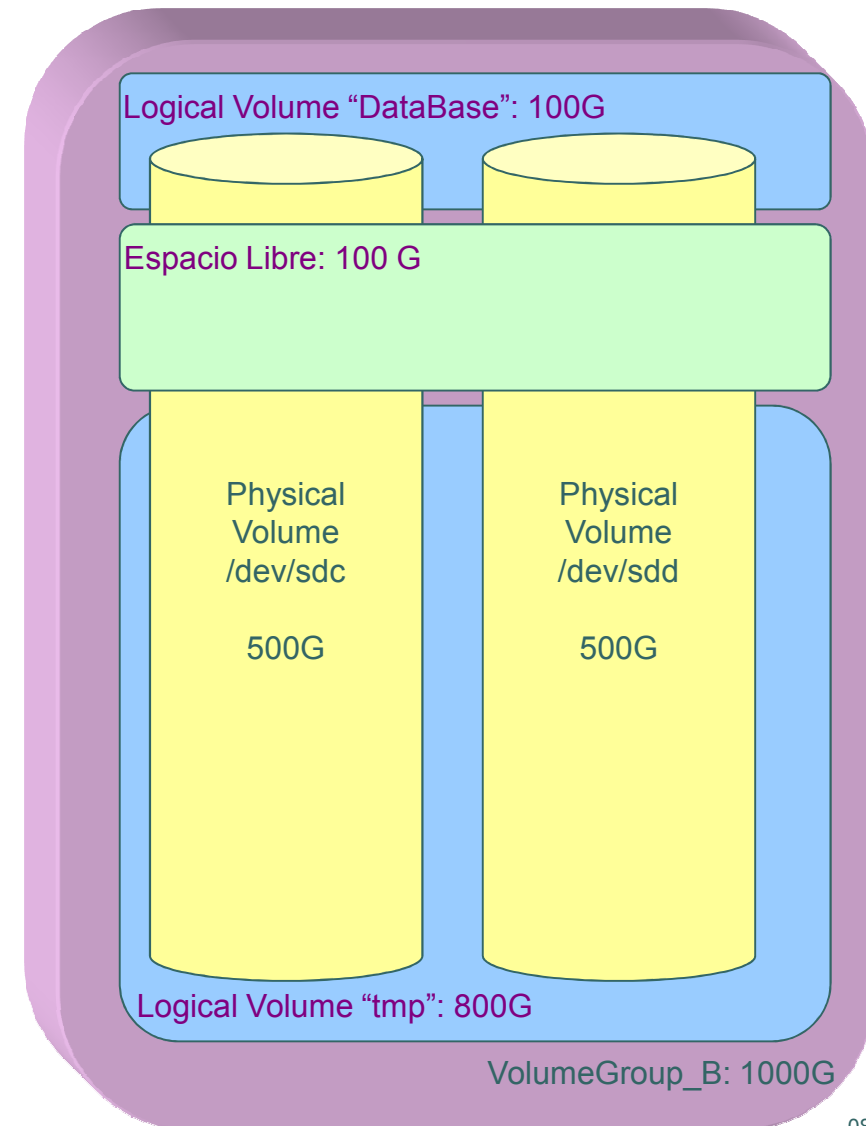
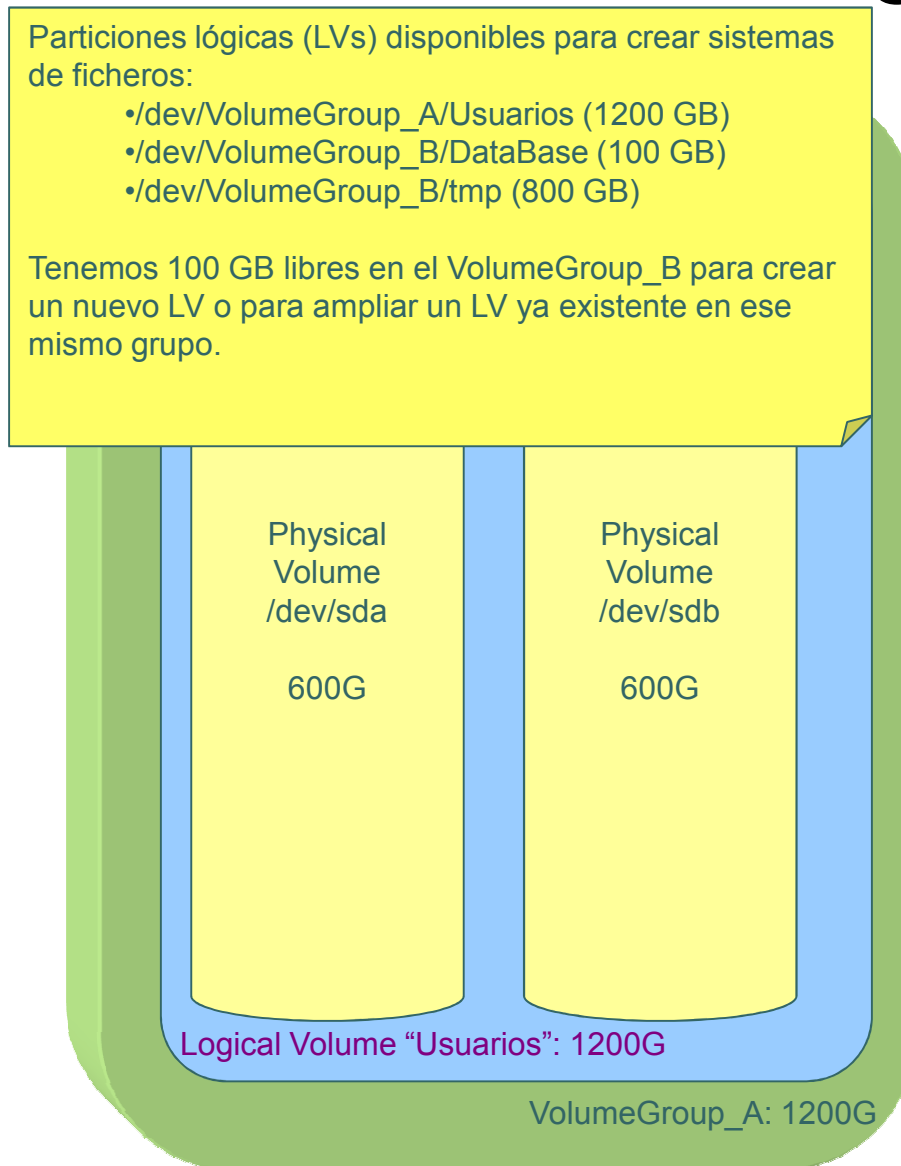
- LVM (Logical Volume Manager) es una implementación de un Gestor de Volúmenes Lógicos para Linux.
  - Implementa los conceptos de PV, VG y LV.
  - Consta de un módulo en el kernel que gestiona los volúmenes y de un conjunto de comandos para la administración.
- La configuración se almacena en los propios volúmenes físicos.
  - No se guardan ficheros de configuración.
  - Al conectarlos a otro sistema, se reconocen y ensamblan en Grupos y se detectan los volúmenes lógicos.
  - Se generan automáticamente los devices con el nombre `/dev/<grupo>/<volumen lógico>`.

# ● ● ● | Gestión Lógica del Espacio

Particiones lógicas (LVs) disponibles para crear sistemas de ficheros:

- /dev/VolumeGroup\_A/Usuarios (1200 GB)
- /dev/VolumeGroup\_B/DataBase (100 GB)
- /dev/VolumeGroup\_B/tmp (800 GB)

Tenemos 100 GB libres en el VolumeGroup\_B para crear un nuevo LV o para ampliar un LV ya existente en ese mismo grupo.



# ● ● ● | Implementación en LVM

## 1. Añadir Physical Volumes.

```
# pvcreate /dev/sda /dev/sdb /dev/sdc /dev/sdd
```

## 2. Ensamblaje de Volume Groups

```
# vgcreate VolumeGroup_A /dev/sda /dev/sdb
```

```
# vgcreate VolumeGroup_B /dev/sdc /dev/sdd
```

## 3. Creación de Logical Volumes

```
# lvcreate --name Usuarios --size 1200G VolGroup_A
```

```
# lvcreate --name tmp --size 800G VolGroup_B
```

```
# lvcreate --name DataBase --size 100G VolGroup_B
```

## 4. Ejemplo de uso de los Logical Volumes

```
# mkfs.ext3 /dev/VolGroup_A/Usuarios
```

```
# mkfs.xfs /dev/VolGroup_B/tmp
```

```
# mkfs.ext3 /dev/VolGroup_B/DataBase
```

```
# mount /dev/VolGroup_A/Usuarios /mnt/usuarios
```

```
# mount /dev/VolGroup_B/tmp /mnt/tmp
```

```
# mount /dev/VolGroup_B/Database /mnt/database
```

# ● ● ● | Operaciones Avanzadas

- Características

- Las referencias a volúmenes lógicos y grupos de volúmenes permanecen inmutables.
- Las operaciones se pueden realizar sin interrumpir el acceso al sistema de ficheros (“on-line”).

- Operaciones destacadas por útiles:

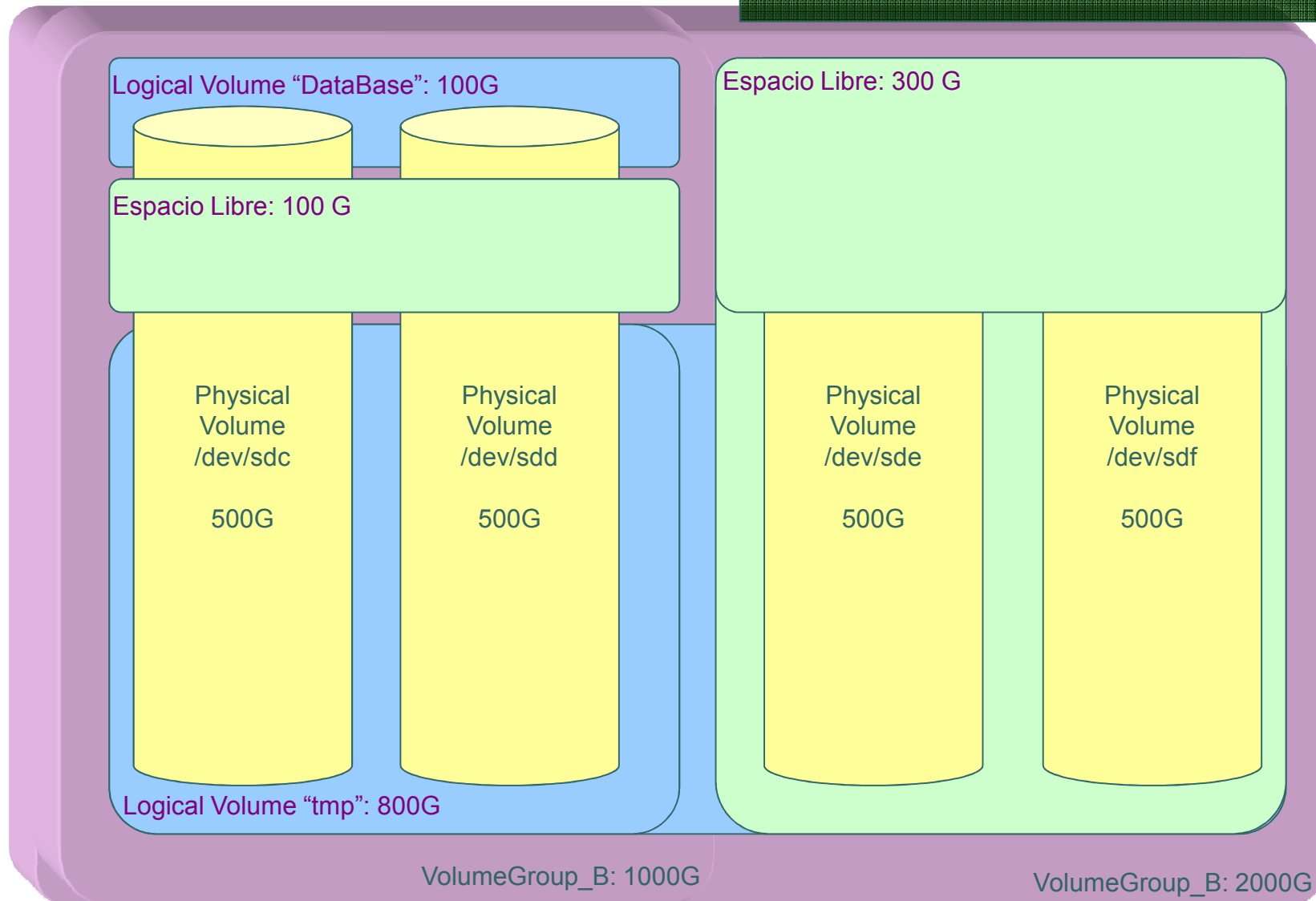
- Ampliar el espacio disponible en un sistema de ficheros
- Migrar un sistema de ficheros de un volumen físico a otro

# ● ● ● | Ampliación del espacio

- Operaciones
  - Se amplía el Volume Group (opcional)
  - Se expande el Logical Volume
  - Se expande el sistema de ficheros para que ocupe todo el volumen lógico
- Restricciones
  - Las operaciones son casi instantáneas.
  - Con XFS todo esto se puede hacer con el sistema de ficheros montado. No se interfiere el acceso a ficheros.
  - Con EXT2/EXT3 hay que desmontar el sistema de ficheros. La expansión es un poco más lenta porque inicializa los nuevos i-nodes.
  - Un Logical Volume se puede encoger, pero no el sistema de ficheros que contiene.
- Secuencia
  1. Añadir más Physical Volumes.  
`# pvcreate /dev/sde /dev/sdf`
  2. Expansión de Volume Groups  
`# vgextend VolumeGroup_B /dev/sde /dev/sdf`
  3. Expansión de Logical Volumes  
`# lvextend -size +700G /dev/VolGroup_B/tmp`
  4. Redimensión del sistema de ficheros  
`# xfs_growfs /dev/VolGroup_B/tmp`

# Ampliación del

```
# pvcreate /dev/sde /dev/sdf
# vgextend VolumeGroup_B /dev/sde /dev/sdf
# lvextend -size +700G /dev/VolGroup_B/tmp
# xfs_growfs /dev/VolGroup_B/tmp
```



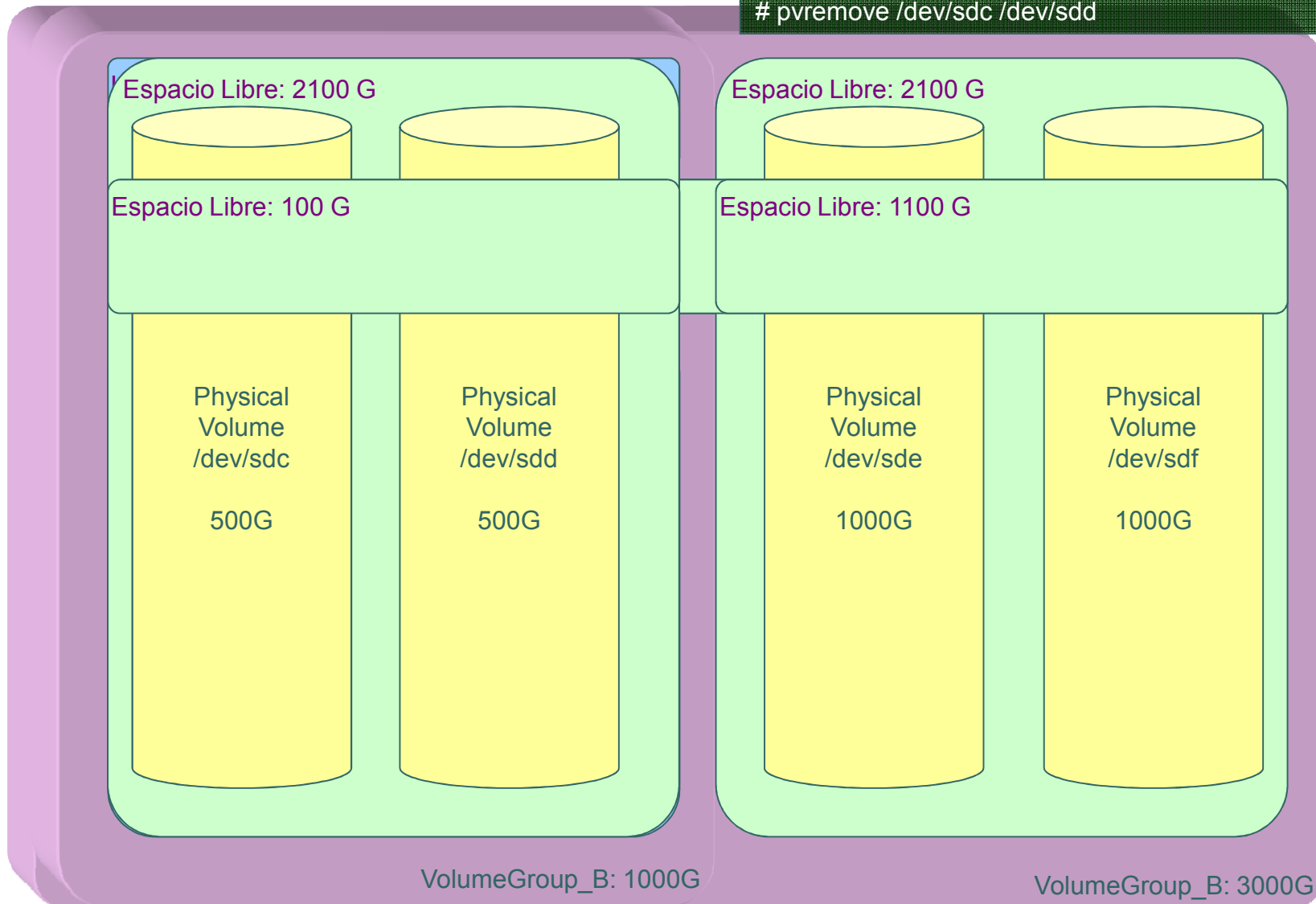
# ● ● ● | Migración del espacio

- Operaciones
  - Se amplía el Volume Group con nuevos Physical Volumes
  - Se mueve todo el contenido de un Physical Volume a otro distinto
  - Se eliminan los Physical Volumes vacíos del Volume Group
- Restricciones
  - Las operaciones no interrumpen el acceso al contenido de los volúmenes a mover.
  - Las operaciones son casi instantáneas excepto mover el contenido
  - La operación de movimiento es atómica. Hasta que no termina se puede deshacer sin consecuencias.
  - Es independiente del formato del volumen (XFS, EXT3, SWAP, etc).
- Secuencia
  1. Añadir más Physical Volumes.  
`# pvcreate /dev/sde /dev/sdf`
  2. Expansión de Volume Groups  
`# vgextend VolumeGroup_B /dev/sde /dev/sdf`
  3. Movimiento de Contenidos  
`# pvmove /dev/sdc /dev/sde; pvmove /dev/sdd /dev/sdf`
  4. Retirada de Physical Volumes  
`# vgreduce VolumeGroup_B /dev/sdc /dev/sdd`  
`# pvremove /dev/sdc /dev/sdd`



# Migración del E

```
# pvcreate /dev/sde /dev/sdf
# vgextend VolumeGroup_B /dev/sde /dev/sdf
# pvmove /dev/sdc /dev/sde; pvmove /dev/sdd /dev/sdf
# vgreduce VolumeGroup_B /dev/sdc /dev/sdd
# pvremove /dev/sdc /dev/sdd
```



# ● ● ● | Consulta del estado

- Volúmenes físicos
  - pvscan
  - pvs
- Grupos de Volúmenes
  - vgscan
  - vgs
- Volúmenes Lógicos
  - lvscan
  - lvs
- Operaciones de Migración
  - pvmove: Indica el % actual de las operaciones de migración en curso.