



# FUNDAMENTOS DE HARDWARE

## IMPLANTACIÓN DE SISTEMA RAID USANDO MDADM EN UN SISTEMA OPERATIVO DEBIAN

Ana María Cuenca Hoyo

María Moreno Muñoz

Fernando Muñoz Espejo

1º de Administración de Sistemas Informáticos en Red

## Índice Contenido

1. INTRODUCCIÓN .....	1
1.1. TIPOS DE RAID .....	1
1.1.1. RAID 0.....	1
1.1.2. RAID 1 .....	1
1.1.3. RAID 4.....	2
1.1.4. RAID 5.....	2
1.1.5. RAID 0+1.....	2
1.1.6. RAID 1+0.....	3
1.1.7. RAID 50.....	3
1.2. USO DE MDADM.....	4
1.2.1. INSTALACIÓN DE MDADM.....	4
1.2.2. CREACIÓN DE RAID CON MDADM .....	5
2. CREAR RAID .....	5
2.1. Crear un disco duro.....	5
2.2. RAID 0.....	8
2.3. RAID 1 .....	11
2.4. RAID 4.....	15
2.5. RAID 5.....	17
2.6. RAID 10.....	23
2.6.1. RAID 10 – DURANTE INSTALACIÓN DEBIAN .....	23
2.6.2. RAID 10 –DESPUES DE INSTALAR DEBIAN.....	31
2.7. RAID 01 .....	36
2.8. RAID 50.....	40
3. USO EN DIFERENTES SITUACIONES .....	44
3.1. USO EN ORDENADOR PARTICULAR .....	44
3.2. USO EN SERVIDOR .....	44
4. WEBGRAFÍA.....	45

## Índice Ilustraciones

Ilustración 1: RAID 0.....	1
Ilustración 2: RAID 1.....	1
Ilustración 3: RAID 4.....	2
Ilustración 4: RAID 5.....	2
Ilustración 5: RAID 01.....	2
Ilustración 6: RAID 10.....	3
Ilustración 7: RAID 50.....	3
Ilustración 8: MDADM - Actualización de paquetes.....	4
Ilustración 9: MDADM – Instalación.....	5
Ilustración 10: DISCOS - Inicio para añadir disco.....	5
Ilustración 11: DISCOS - Crear disco.....	6
Ilustración 12: DISCOS - Crear disco en controlador SATA.....	6
Ilustración 13: DISCOS - Tipo de disco duro.....	7
Ilustración 14: DISCOS - Tamaño del disco.....	7
Ilustración 15: DISCOS - Ubicación y tamaño.....	8
Ilustración 16: RAID0 - Comando Creación.....	8
Ilustración 17: RAID0 - lsblk.....	9
Ilustración 18: RAID0 – Estado.....	9
Ilustración 19: RAID0 - Formato ext4.....	9
Ilustración 20: RAID0 - Montar unidad.....	10
Ilustración 21: RAID0 - Estado tras fallo.....	10
Ilustración 22: RAID0 - lsblk -fm tras fallo.....	10
Ilustración 23: RAID1 – Creación.....	11
Ilustración 24: RAID1 - Comprobación.....	11
Ilustración 25: RAID1 - Creación de la partición.....	11
Ilustración 26: RAID1 - Montamos la carpeta.....	12
Ilustración 27: RAID1 - Creamos archivos.....	12
Ilustración 28: RAID1 – Fallo de un disco.....	12
Ilustración 29: RAID1 - Estado de los archivos.....	13
Ilustración 30: RAID1 - Eliminar disco.....	13
Ilustración 31: RAID1 - Comprobación de archivos.....	14
Ilustración 32: RAID1 - Añadir disco.....	14
Ilustración 33: RAID1 – Añadimos nuevo disco.....	14
Ilustración 34: RAID1 – Comprobamos archivos.....	15
Ilustración 35: RAID4 - Comando para crearlo.....	15
Ilustración 36: RAID4 - Estado.....	15
Ilustración 37: RAID4 – Indicar formato.....	16
Ilustración 38: RAID4 - Creación de archivos.....	16
Ilustración 39: RAID4 – Fallo y posterior estado.....	16
Ilustración 40: RAID4 – Fallo de dos discos.....	17
Ilustración 41: RAID5 – Creación del RAID.....	17
Ilustración 42: RAID5 – Comprobación con "lsblk -fm".....	18
Ilustración 43: RAID5 – Creación del RAID.....	18
Ilustración 44: RAID5 – Formato ext4.....	19
Ilustración 45: RAID5 – Montar dispositivo.....	19
Ilustración 46: RAID5 – Fallo en un disco.....	19
Ilustración 47: RAID5 – Mostramos archivos.....	20
Ilustración 48: RAID5 – Eliminamos un disco.....	20
Ilustración 49: RAID5 – Sustituimos el disco.....	21

Ilustración 50: RAID5 – Comprobamos estado con el nuevo disco .....	21
Ilustración 51: RAID5 – Fallo en dos discos .....	22
Ilustración 52: RAID5 – Vemos estado .....	22
Ilustración 53: RAID5 – Pérdida de datos .....	22
Ilustración 54: RAID10 – Partición guiada.....	23
Ilustración 55: RAID10 – Utilizar todo el disco .....	24
Ilustración 56: RAID10 – Elegir disco.....	24
Ilustración 57: RAID10 – Esquema de particionado .....	25
Ilustración 58: RAID10 – Crear RAID por software .....	26
Ilustración 59: RAID10 – Guardar cambios .....	26
Ilustración 60: RAID10 – Crear un dispositivo MD .....	27
Ilustración 61: RAID10 – Elegir tipo RAID .....	27
Ilustración 62: RAID10 – Número de discos activos.....	28
Ilustración 63: RAID10 – dispositivos libres.....	28
Ilustración 64: RAID10 – Discos activos.....	29
Ilustración 65: RAID10 – Disco libre .....	29
Ilustración 66: RAID10 – Guardar cambios .....	30
Ilustración 67: RAID10 – Terminar .....	30
Ilustración 68: RAID10 – Proceso terminado .....	31
Ilustración 69: RAID10 – Creación .....	31
Ilustración 70: RAID10 – Comprobación de estado .....	32
Ilustración 72: RAID10 – Montamos la carpeta .....	32
Ilustración 73: RAID10 – Creamos archivos .....	33
Ilustración 74: RAID10 – eliminamos un disco.....	33
Ilustración 75: RAID10 – Sustituimos disco.....	34
Ilustración 76: RAID10 – Vemos los datos .....	34
Ilustración 77: RAID10 – Fallo de disco .....	35
Ilustración 78: RAID10 – Fallo de dos discos .....	35
Ilustración 79: RAID10 – Estado sin dos discos.....	35
Ilustración 80: RAID10 – Fallo de 3 discos.....	36
Ilustración 82: RAID 01 – Creación del RAID 0.....	36
Ilustración 83: RAID 01 - Comprobación.....	37
Ilustración 84: RAID 01 – Creación del RAID 1 .....	37
Ilustración 85: RAID 01 – Comprobación .....	37
Ilustración 86: RAID 01 – Formato ext4 .....	37
Ilustración 87: RAID 01 – Montar el dispositivo.....	38
Ilustración 88: RAID 01 – Estado de los archivos.....	38
Ilustración 89: RAID 01 – Fallo de un disco .....	38
Ilustración 90: RAID 01 – Estado.....	39
Ilustración 91: RAID 01 – Estado.....	39
Ilustración 92: RAID 01 – Listado datos tras pérdida.....	39
Ilustración 93: RAID 50 – RAID creado .....	40
Ilustración 94: RAID 50 – Estado del RAID .....	41
Ilustración 95: RAID 50 – Formato ext4 .....	41
Ilustración 96: RAID 50 – Montamos la carpeta .....	41
Ilustración 97: RAID 50 – Fallo de un disco .....	42
Ilustración 98: RAID 50 – Fallo de dos discos .....	42
Ilustración 99: RAID 50 – Fallo de 3 discos.....	43
Ilustración 100: RAID 50 – Fallo de 4 discos.....	43





# 1. INTRODUCCIÓN

Un raid es un grupo de discos independientes utilizados como sistema de almacenamiento de datos. Este sistema, dependiendo su configuración, utiliza las múltiples unidades para distribuir o duplicar los datos.

Los tipos de configuraciones se agrupan en niveles. En este caso nos hemos centrado en los raid 0, 1, 4, 5 y 1+0.

Una ventaja de los raid es que todas las implementaciones pueden soportar el uso de uno o más discos de reserva, unidades preinstaladas que pueden usarse tras el fallo de un disco del RAID. Esto reduce el tiempo del período de reparación al acortar el tiempo de reconstrucción del RAID.

Para la realización de las pruebas se usará Oracle VirtualBox versión 5.2.0 donde se virtualizará el sistema operativo Debian 9.4 de 32 bits y estará instalado mdadm versión 3.4.

## 1.1. TIPOS DE RAID

### 1.1.1. RAID 0

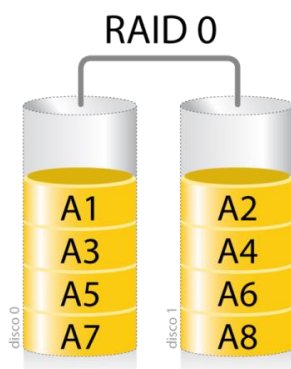


Ilustración 1:  
RAID 0

En el RAID-0, los datos simplemente se particionan y distribuyen a lo largo de los diferentes discos que forman el conjunto, lo que repercute en un alto rendimiento en el manejo de los datos, aunque no ofrece tolerancia a fallos ya que no se almacena ningún tipo de información de paridad, ni de control de errores.

El RAID-0 es adecuado con aplicaciones que requieren una velocidad de acceso muy alta a grandes volúmenes de almacenamiento y no necesitan redundancia de datos.

Se puede ver su implementación [aquí](#).

### 1.1.2. RAID 1

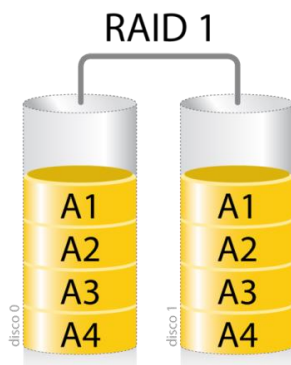


Ilustración 2:  
RAID 1

RAID 1 crea una copia exacta de los datos de los discos. Esto sirve para tener más seguridad, pero desaprovechas capacidad ya que si tenemos dos discos solo podremos utilizar la capacidad de uno de ellos.

Como en el RAID 0, el tiempo medio de lectura se reduce, ya que los sectores a buscar pueden dividirse entre los discos, bajando el tiempo de búsqueda. Al escribir, el conjunto se comporta como un único disco, dado que los datos deben ser escritos en todos los discos del RAID 1. Por tanto, el rendimiento de escritura no mejora.

Se puede ver su implementación [aquí](#).



### 1.1.3. RAID 4

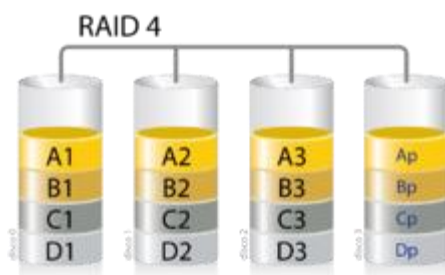


Ilustración 3: RAID 4

Este tipo de Raid se caracteriza por el uso de un disco de paridad dedicado, por esto necesitamos un mínimo de 3 discos físicos para su funcionalidad. Además, permite servir varias peticiones de escritura o lectura simultáneamente.

Es el indicado para el almacenamiento de archivos grandes, donde es necesario asegurar la integridad de la información. Eso porque, en este nivel, cada operación de grabación requiere un nuevo cálculo de paridad, dando mayor confianza al almacenamiento.

Se puede ver su implementación [aquí](#).

### 1.1.4. RAID 5

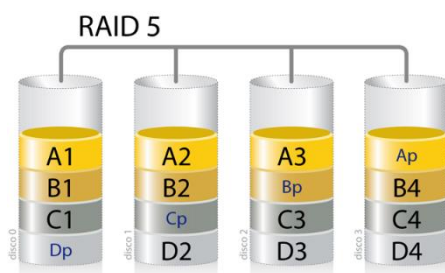


Ilustración 4: RAID 5

Este sistema de almacenamiento se caracteriza por la distribución de la información de paridad entre todos los discos miembros del raid. RAID 5 ha obtenido gran popularidad por no tener tanta redundancia de datos permitiendo mayor capacidad al disco.

Para ser implementado necesita un mínimo de 3 discos. Su funcionamiento se basa en generar un bloque de paridad cada vez que se escriben datos. Los bloques de paridad no se leen durante la búsqueda de datos para evitar una sobrecarga innecesaria, solo se acude a ellos cuando se detecta un error en un sector del disco.

Para recuperar el error los bloques de paridad y los datos dañados se combinan matemáticamente para reconstruir los datos dañados. Aunque el fallo de un segundo disco provoca la pérdida completa de los datos.

Se puede ver su implementación [aquí](#).

### 1.1.5. RAID 0+1

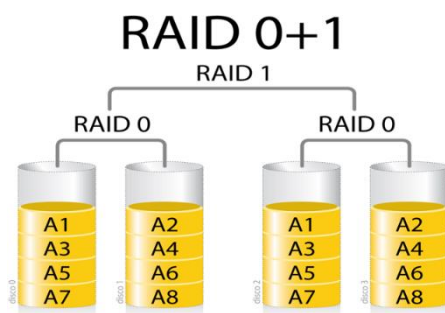


Ilustración 5: RAID 01

RAID usado para replicar y compartir datos entre varios discos. Primero se crean dos conjuntos RAID 0 (dividiendo los datos en discos) y luego, sobre los anteriores, se crea un conjunto RAID 1, el cual realiza un espejo de los RAID 0 anteriores.

La ventaja de un RAID 0+1 es que cuando un disco duro falla, los datos perdidos pueden ser copiados del otro conjunto de nivel 0 para reconstruir el conjunto global. Sin

embargo, añadir un disco duro adicional en una división, es obligatorio añadir otro al de la otra división para equilibrar el tamaño del conjunto.

Este sistema necesita como mínimo 4 discos.

Se puede ver su implementación [aquí](#).

### 1.1.6. RAID 1+0

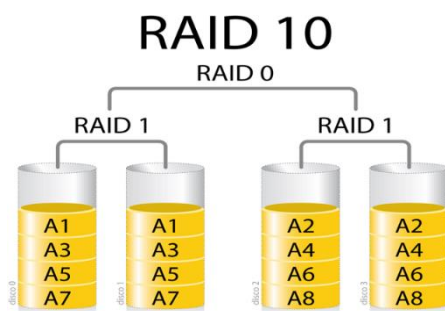


Ilustración 6: RAID 10

La combinación de los niveles de almacenamiento 1 y 0 hacen que RAID 10 sea tanto rápido como resistente. RAID 10 permite que falle un disco de cada par sin perder los datos. Sin embargo, si los discos que han fallado no se reemplazan, el restante pasa a ser un punto único de fallo para todo el conjunto. Si ese disco falla entonces, se perderán todos los datos del conjunto completo.

Para implementarlo necesitaremos al menos cuatro discos duros físicos y es a menudo la mejor elección para bases de datos de altas prestaciones, debido a que la ausencia de cálculos de paridad proporciona mayor velocidad de escritura.

Se puede ver su implementación [aquí](#).

### 1.1.7. RAID 50

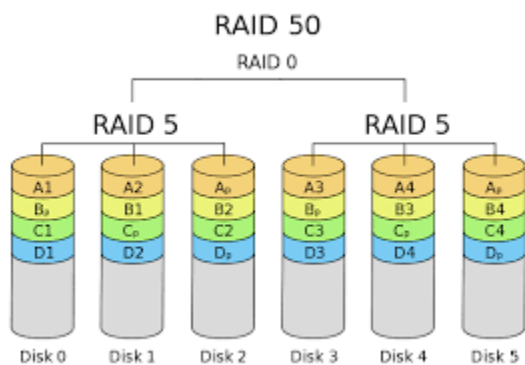


Ilustración 7: RAID 50

Combina la capacidad de dividir los datos por todos los discos del RAID 0 y la paridad de RAID 5. Lo que conlleva la necesidad de 6 discos mínimo y permite que un disco de cada RAID 5 falle sin perder los datos.

El RAID 50 mejora el rendimiento del RAID 5, especialmente en escritura, y proporciona mejor tolerancia a fallos que un nivel RAID único. Este nivel se recomienda para aplicaciones que necesitan gran tolerancia a fallos, capacidad y rendimiento de búsqueda aleatoria.

A medida que el número de unidades del conjunto RAID 50 crece y la capacidad de los discos aumenta, el tiempo de recuperación lo hace también.

Se puede ver su implementación [aquí](#).





## 1.2. USO DE MDADM

Mdam se trata de una herramienta de gestión de RAIDS mediante software.

En nuestro caso hemos usado dicho software porque estamos usando máquinas virtuales para realizar los RAID, y éstas no reconocen los controladores de los discos duros. Además de esto también queríamos mostrar cómo se realizan los RAID mediante la terminal y que no fuese en modo gráfico, lo cual es más intuitivo.

Además de lo anterior también hay que resaltar las ventajas, u opciones que nos permite el RAID por software:

- Bajo coste ya que no necesitaremos utilizar tarjetas RAID que son costosas, y además tienen las restricciones propias de los fabricantes.
- La configuración se basa en el núcleo de sistema.
- Aprovecharemos de manera más eficiente los recursos del sistema, ya que cuando hay que reconstruir un dato se utilizarán los recursos que estén libres.
- Podremos quitar, y añadir unidades en caliente, es decir, sin necesidad de apagar la máquina.
- Detectará automáticamente el número de núcleos que tiene el microprocesador para de esta forma aprovechar mejor los recursos del sistema.

### 1.2.1. INSTALACIÓN DE MDADM

Primero ejecutaremos el comando "apt-get update" para actualizar los paquetes del sistema.

```
root@debian:/home/usuario# apt-get update
Ign:1 cdrom://[Debian GNU/Linux 9.4.0 _Stretch_ - Official i386 DVD Binary-1 201
80310-11:55] stretch InRelease
Ign:2 cdrom://[Debian GNU/Linux 9.4.0 _Stretch_ - Official i386 DVD Binary-1 201
80310-11:55] stretch Release
Obj:3 cdrom://[Debian GNU/Linux 9.4.0 _Stretch_ - Official i386 DVD Binary-1 201
80310-11:55] stretch/contrib i386 Packages
Ign:4 cdrom://[Debian GNU/Linux 9.4.0 _Stretch_ - Official i386 DVD Binary-1 201
80310-11:55] stretch/contrib all Packages
Ign:5 cdrom://[Debian GNU/Linux 9.4.0 _Stretch_ - Official i386 DVD Binary-1 201
80310-11:55] stretch/contrib Translation-es
Obj:6 cdrom://[Debian GNU/Linux 9.4.0 _Stretch_ - Official i386 DVD Binary-1 201
80310-11:55] stretch/contrib Translation-en
Ign:7 cdrom://[Debian GNU/Linux 9.4.0 _Stretch_ - Official i386 DVD Binary-1 201
80310-11:55] stretch/contrib Translation-es_ES
Ign:8 cdrom://[Debian GNU/Linux 9.4.0 _Stretch_ - Official i386 DVD Binary-1 201
80310-11:55] stretch/contrib i386 DEP-11 Metadata
Ign:9 cdrom://[Debian GNU/Linux 9.4.0 _Stretch_ - Official i386 DVD Binary-1 201
80310-11:55] stretch/contrib all DEP-11 Metadata
Ign:10 cdrom://[Debian GNU/Linux 9.4.0 _Stretch_ - Official i386 DVD Binary-1 20
180310-11:55] stretch/contrib DEP-11 64x64 Icons
```

*Ilustración 8: MDADM - Actualización de paquetes.*



Para realizar la descarga de mdadm utilizaremos "apt-get install mdadm".

```
usuario@debian: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
root@debian:/home/usuario# apt-get install mdadm  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
Se instalarán los siguientes paquetes adicionales:  
  exim4-base exim4-config exim4-daemon-light guile-2.0-libs libgsasl7 libkyotocabinet16v5  
  libmailutils5 libmariadbclient18 libntlm0 mailutils mailutils-common mysql-common  
Paquetes sugeridos:  
  eximon4 exim4-doc-html | exim4-doc-info spf-tools-perl swaks mailutils-mh mailutils-doc  
Se instalarán los siguientes paquetes NUEVOS:  
  exim4-base exim4-config exim4-daemon-light guile-2.0-libs libgsasl7 libkyotocabinet16v5  
  libmailutils5 libmariadbclient18 libntlm0 mailutils mailutils-common mdadm mysql-common  
0 actualizados, 13 nuevos se instalarán, 0 para eliminar y 0 no actualizados.  
Se necesita descargar 0 B/8.155 kB de archivos.  
Se utilizarán 27,1 MB de espacio de disco adicional después de esta operación.  
¿Desea continuar? [Y/n]
```

Ilustración 9: MDADM – Instalación

## 1.2.2. CREACIÓN DE RAID CON MDADM

Para crear un raid con mdadm usaremos el mismo comando de dos formas:

`mdadm -Cv /dev/md0 --level=raid0 --raid-devices=2 /dev/sdb /dev/sdc`

`mdadm -Cv /dev/md0 -l0 -n2 /dev/sdb /dev/sdc`

Tanto --level como -l no sirve para indicar el tipo de raid que queremos montar y tanto --raid-devices como -n lo usamos para indicar el número de dispositivos que vamos a hacer.

## 2. CREAR RAID

### 2.1. Crear un disco duro

Lo primero que tenemos que hacer es seleccionar la máquina a la que le queremos añadir el disco, y después hacemos click sobre *Configuración*.

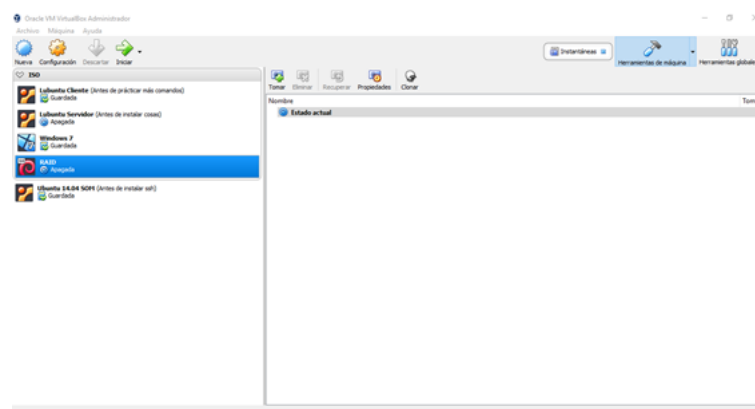
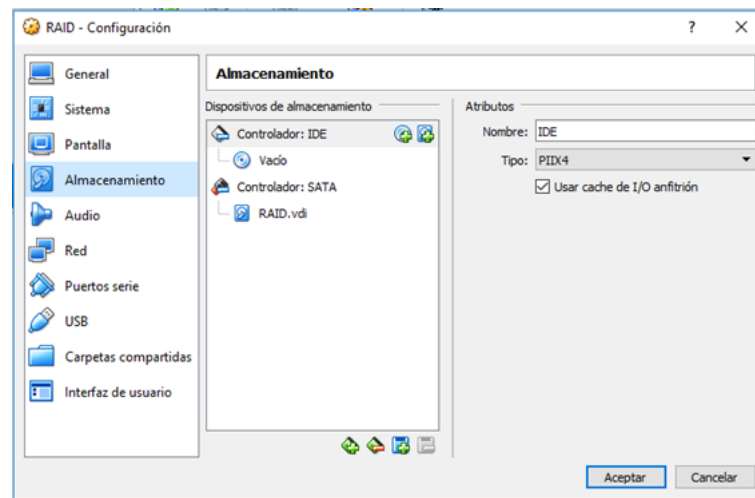


Ilustración 10: DISCOS - Inicio para añadir disco

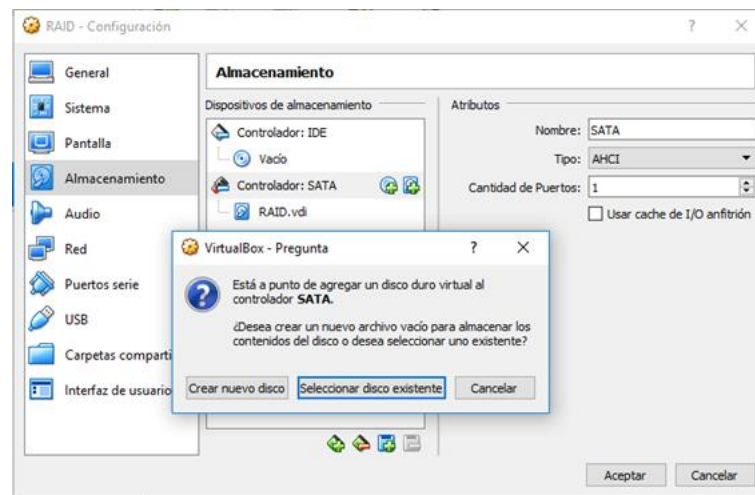


Después seleccionamos el apartado *Almacenamiento*, y hacemos click sobre el icono del disco duro.



*Ilustración 11: DISCOS - Crear disco*

Llegados a este punto podemos elegir entre crear un disco nuevo, o elegir un disco que ya esté creado previamente.



*Ilustración 12: DISCOS - Crear disco en controlador SATA*



Elegimos el tipo de disco que queremos crear.

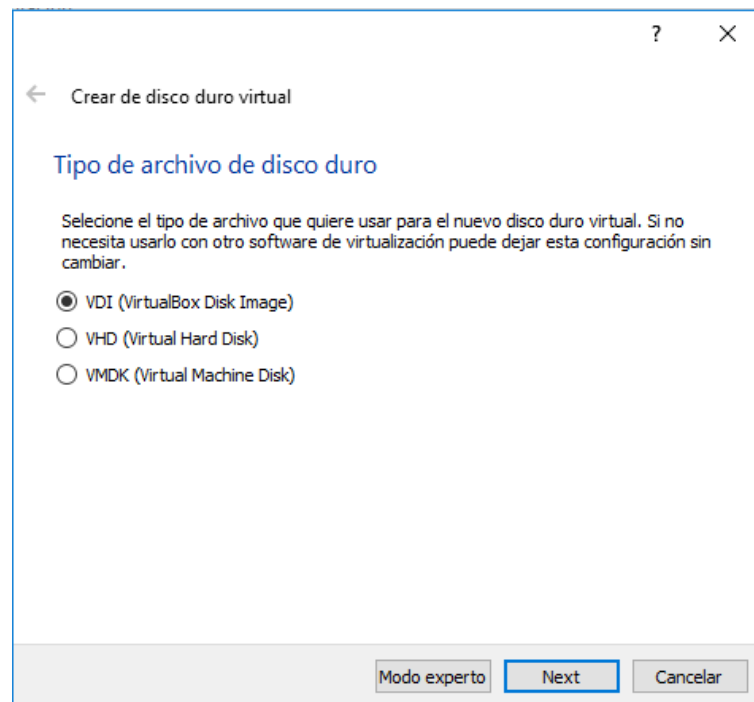


Ilustración 13: DISCOS - Tipo de disco duro

El siguiente paso es elegir si queremos que el tamaño del disco se reserve dinámicamente, y vaya ocupando más espacio conforme se vaya ocupando, o que se reserve automáticamente un tamaño para él, ocupe o no ese espacio.

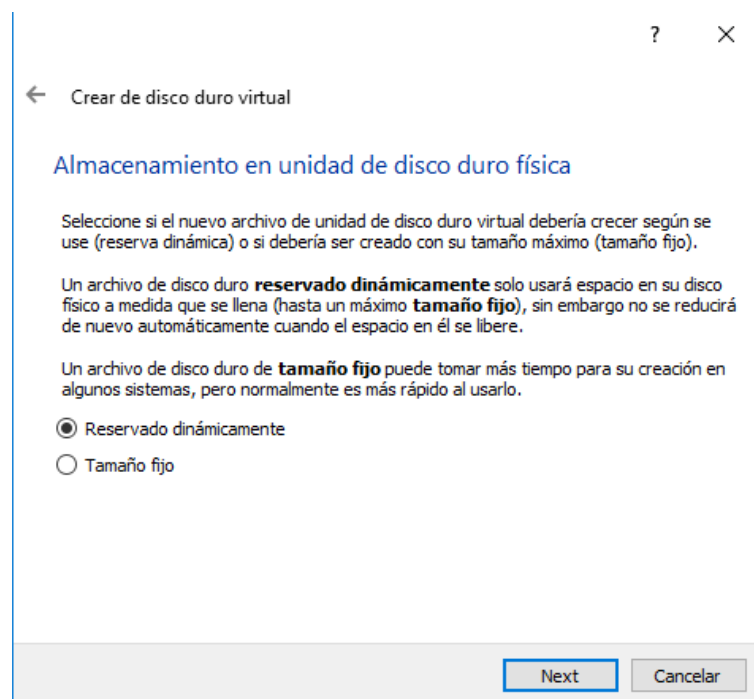


Ilustración 14: DISCOS - Tamaño del disco.



Por último, escribimos el nombre del disco, esto es opcional ya que virtualbox le asignará un nombre por defecto, elegimos el tamaño del mismo, y hacemos click sobre *Crear*.

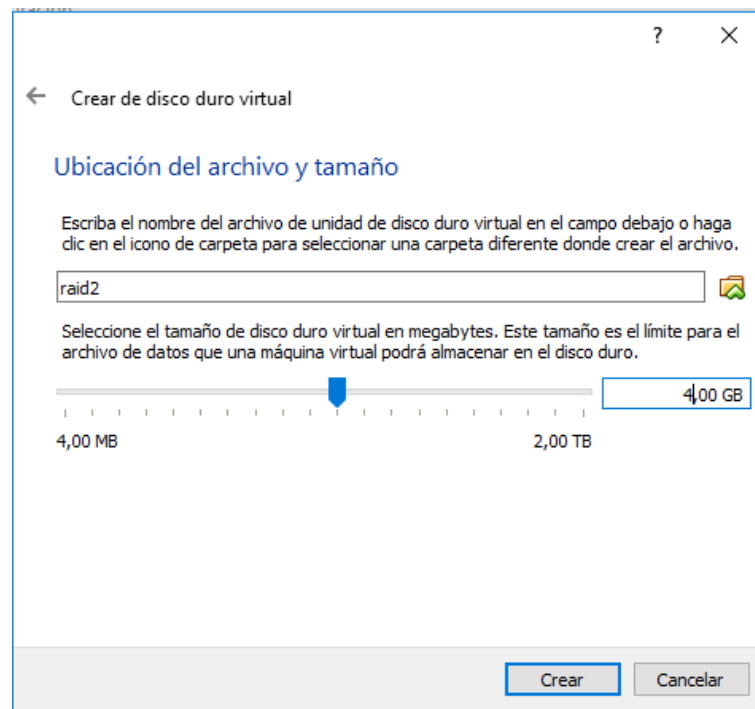


Ilustración 15: DISCOS - Ubicación y tamaño.

## 2.2. RAID 0

Como hemos visto en el anterior apartado con el [RAID 0](#) no se almacena ningún tipo de información de paridad, por lo que añadiremos 2 disco para el crear el conjunto, siguiendo los pasos indicados en el punto [2.1](#), ya que son los mínimos necesarios.

En primer lugar, introduciremos el comando de mdadm para la creación de un raid 0, donde indicaremos el tipo de raid con el parámetro `--level`, el número de dispositivos con `--raid-devices` y los dispositivos que forman el raid.

```
root@debian:/home/usuario# mdadm -Cv /dev/md0 --level=raid0 --raid-devices=2 /dev/sdb /dev/sdc
mdadm: chunk size defaults to 512K
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

Ilustración 16: RAID0 - Comando Creación

Una vez se ha creado el raid usamos el comando `"lsblk -fm"`, para comprobar que se ha creado correctamente y que nos aparece en la tabla de dispositivos. Como podemos ver los dispositivos sdb y sdc son miembros de un raid denominado md0.

```

root@debian:/home/usuario# lsblk -fm
NAME FSTYPE LABEL UUID MOUNTPOINT NAME SIZE OWNER GROUP MODE
sda
├─sda1 ext4 8c22015e-3ab0-43f4-ac73-498ef0b277cd /
├─sda2
├─sda5 swap 3e13e863-5bb5-4aa1-8594-fe84d18bcb84 [SWAP]
├─sdb linux_raid_member debian:0 78230c9b-1f92-afc9-88fa-766df604a0d3
├─md0
├─sdc linux_raid_member debian:0 78230c9b-1f92-afc9-88fa-766df604a0d3
├─md0
└─sr0 iso9660 VBox_GAs_5.2.1 2017-10-17-20-04-56-16 /media/cdrom0 sr0 57,5M root cdrom brw-rw----

```

Ilustración 17: RAID0 - lsblk

También podemos usar el comando “mdadm -D” para comprobar el estado del raid., como podemos ver se ha creado un raid0, formado por dos dispositivos y los dos dispositivos están activos.

```

root@debian:/home/usuario# mdadm -D /dev/md0
/dev/md0:
  Version : 1.2
  Creation Time : Tue Jun  5 12:00:24 2018
  Raid Level : raid0
  Array Size : 16760832 (15.98 GiB 17.16 GB)
  Raid Devices : 2
  Total Devices : 2
  Persistence : Superblock is persistent

  Update Time : Tue Jun  5 12:00:24 2018
  State : clean
  Active Devices : 2
  Working Devices : 2
  Failed Devices : 0
  Spare Devices : 0

  Chunk Size : 512K

  Name : debian:0 (local to host debian)
  UUID : 78230c9b:1f92afc9:88fa766d:f604a0d3
  Events : 0

  Number Major Minor RaidDevice State
    0      8     16        0 active sync  /dev/sdb
    1      8     32        1 active sync  /dev/sdc

```

Ilustración 18: RAID0 – Estado

Una vez que tenemos el raid creado, pasamos a darle formato ext4.

```

root@debian:/home/usuario# mkfs.ext4 /dev/md0
mke2fs 1.43.4 (31-Jan-2017)
Se está creando un sistema de ficheros con 4190208 bloques de 4k y 1048576 nodos-i
UUID del sistema de ficheros: ff514287-d6b8-4da7-beab-f8672c06df37
Respaldo del superbloque guardado en los bloques:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000

Reservando las tablas de grupo: hecho
Escribiendo las tablas de nodos-i: hecho
Creando el fichero de transacciones (16384 bloques): hecho
Escribiendo superbloques y la información contable del sistema de ficheros: hecho

```

Ilustración 19: RAID0 - Formato ext4



Ahora que a tenemos el raid con formato podemos montarlo.

```
root@debian:/home/usuario# mount /dev/md0 /raid0
root@debian:/home/usuario# ls /raid0/
lost+found
```

Ilustración 20: RAID0 - Montar unidad

Ya que lo tenemos montado y hemos visto que funciona, quitaremos uno de los dispositivos (cuando reiniciamos el sistema, en ocasiones el nombre del dispositivo de modifica y se cambia por md127).

```
root@debian:/home/usuario# mdadm -D /dev/md127
/dev/md127:
  Version : 1.2
  Raid Level : raid0
  Total Devices : 1
  Persistence : Superblock is persistent

  State : inactive

  Name : debian:0 (local to host debian)
  UUID : 78230c9b:1f92afc9:88fa766d:f604a0d3
  Events : 0

  Number Major Minor RaidDevice
    -     8     16    -         /dev/sdb
```

Ilustración 21: RAID0 - Estado tras fallo

```
root@debian:/home/usuario# lsblk -fm
NAME FSTYPE LABEL UUID MOUNTPOINT NAME SIZE OWNER GROUP MODE
sda
├─sda1 ext4 8c22015e-3ab0-43f4-ac73-498ef0b277cd /
├─sda2
└─sda5 swap 3e13e863-5bb5-4aa1-8594-fe84d18bcb84 [SWAP]
sdb linux_raid_member debian:0 78230c9b-1f92-afc9-88fa-766df604a0d3 sdb 8G root disk brw-rw----
sr0 iso9660 VBox_GAS_5.2.1 2017-10-17-20-04-56-16 /media/cdrom0 sr0 57,5M root cdrom brw-rw----
```

Ilustración 22: RAID0 - lsblk -fm tras fallo

Al quitar uno de los dispositivos como podemos ver en el estado, el raid se corrompe, con lo que no puede llegar a ser montado y todos los datos que tuviéramos almacenados ha sido perdidos, también podemos ver que al comprobar los dispositivos con “lsblk -fm” el dispositivo se ha perdido.





## 2.3. RAID 1

Para este [RAID 1](#) vamos a usar dos discos. Para crearlos seguiremos los pasos anteriormente indicados en el punto [2.1](#)

El primer paso que vamos a realizar es crear el RAID1 para ello utilizaremos mdadm indicando tipo de RAID que vamos a crear, el número de dispositivos que vamos a utilizar y el nombre de los mismos.

```

usuario@debian: ~
Archivo Editar Ver Buscar Terminal Ayuda

root@debian:~# mdadm -Cv /dev/md0 --level=raid1 --raid-devices=2 /dev/sdb /dev/sdc
mdadm: Note: this array has metadata at the start and
may not be suitable as a boot device. If you plan to
store '/boot' on this device please ensure that
your boot-loader understands md/v1.x metadata, or use
--metadata=0.90
mdadm: size set to 4190208K
Continue creating array? yes
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
root@debian:~#

```

Ilustración 23: RAID1 – Creación

Hecho esto comprobaremos con el comando `lsblk -fm` si se ha creado correctamente.

```

root@debian:~# lsblk -fm
NAME        FSTYPE LABEL  UUID                                  MOUNTPOINT NAME  SIZE OWNER GROUP MODE
sda                                     sda         8G root disk brw-rw----
├─sda1 ext4                                     ┌─sda1      6G root disk brw-rw----
├─sda2                                     ┌─sda2      1K root disk brw-rw----
└─sda5 swap                                     ┌─sda5      2G root disk brw-rw----
sdb linux_raid_member debian:0 81291ced-0893-dd1b-70b4-b64b0446c27d sdb        4G root disk brw-rw----
├─md0                                     ┌─md0        4G root disk brw-rw----
└─sdc linux_raid_member debian:0 81291ced-0893-dd1b-70b4-b64b0446c27d sdc        4G root disk brw-rw----
├─md0                                     ┌─md0        4G root disk brw-rw----
└─sdd                                     ┌─sdd        4G root disk brw-rw----
sr0                                     sr0       1024M root cdrom brw-rw----
root@debian:~#

```

Ilustración 24: RAID1 - Comprobación

El siguiente paso que vamos a realizar es crear una partición que ocupará todo el disco el disco, para ello usaremos `n` para indicar que es una nueva partición, con la opción `p` indicamos que es una partición primaria, seleccionamos el número de partición, donde comenzará y donde terminará dicha partición. Por último, con la opción `w` guardaremos los cambios, y saldremos.

```

root@debian:~# sudo fdisk /dev/sdd

Bienvenido a fdisk (util-linux 2.29.2).
Los cambios solo permanecerán en la memoria, hasta que decida escribirlos.
Tenga cuidado antes de utilizar la orden de escritura.

El dispositivo no contiene una tabla de particiones reconocida.
Se ha creado una nueva etiqueta de disco DOS con el identificador de disco 0xdd220a15.

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)
Selección (valor predeterminado p): p
Número de partición (1-4, valor predeterminado 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.

Orden (m para obtener ayuda): w
Se ha modificado la tabla de particiones.
Llamando a ioctl() para volver a leer la tabla de particiones.
Se están sincronizando los discos.

```

Ilustración 25: RAID1 - Creación de la partición





Hecho esto montamos la carpeta.

```
root@debian: mount /dev/md0 /datos/
root@debian: ls /datos/
lost+found
root@debian: █
```

*Ilustración 26: RAID1 - Montamos la carpeta*

Creamos los archivos que contendrán los datos, y comprobamos que podemos verlos.

```
root@debian:~# cat /datos/a.txt
total 0
1
root@debian:~# cat /datos/b.txt
53
root@debian:~# cat /datos/c.txt
30
root@debian:~# █
```

*Ilustración 27: RAID1 - Creamos archivos*

Ahora vamos a provocar un fallo en uno de los discos. Y comprobamos el estado del RAID.

```
root@debian:~# mdadm /dev/md0 --fail /dev/sdb
mdadm: set /dev/sdb faulty in /dev/md0
root@debian:~# mdadm -D /dev/md0
/dev/md0:
  Version : 1.2
  Creation Time : Fri Jun  8 01:39:19 2018
  Raid Level : raid1
  Array Size : 4190208 (4.00 GiB 4.29 GB)
  Used Dev Size : 4190208 (4.00 GiB 4.29 GB)
  Raid Devices : 2
  Total Devices : 2
  Persistence : Superblock is persistent

  Update Time : Fri Jun  8 01:43:51 2018
  State : clean, degraded
  Active Devices : 1
  Working Devices : 1
  Failed Devices : 1
  Spare Devices : 0

    Name : debian:0 (local to host debian)
    UUID : befcc69e:dd035225:7b601bdb:0050eb99
    Events : 19

   Number Major Minor RaidDevice State
    -     -     -     -     -
    1      8     32      1     active sync  /dev/sdc
    0      8     16      -     faulty   /dev/sdb
root@debian:~# █
```

*Ilustración 28: RAID1 – Fallo de un disco*



Y comprobamos si podemos acceder a los archivos.

```
root@debian:~# cat /datos/a.txt
2
root@debian:~# cat /datos/b.txt
30
root@debian:~# cat /datos/c.txt
35
root@debian:~#
```

*Ilustración 29: RAID1 - Estado de los archivos*

Lo siguiente que vamos a hacer es quitar el disco duro.

```
root@debian:~# mdadm --remove /dev/md0 /dev/sdb
mdadm: hot removed /dev/sdb from /dev/md0
root@debian:~# mdadm -D /dev/md0
/dev/md0:
  Version : 1.2
  Creation Time : Fri Jun  8 01:39:19 2018
  Raid Level : raid1
  Array Size : 4190208 (4.00 GiB 4.29 GB)
  Used Dev Size : 4190208 (4.00 GiB 4.29 GB)
  Raid Devices : 2
  Total Devices : 1
  Persistence : Superblock is persistent

  Update Time : Fri Jun  8 01:45:45 2018
  State : clean, degraded
Active Devices : 1
Working Devices : 1
Failed Devices : 0
Spare Devices : 0

    Name : debian:0 (local to host debian)
    UUID : befcc69e:dd035225:7b601bdb:0050eb99
    Events : 26

   Number Major Minor RaidDevice State
    -     -     -     -     -
    1      8     32        1   active sync  /dev/sdc
root@debian:~#
```

*Ilustración 30: RAID1 - Eliminar disco*



Comprobamos si los archivos se han perdido.

```
root@debian:~# cat /datos/a.txt
2
root@debian:~# cat /datos/b.txt
30
root@debian:~# cat /datos/c.txt
35
root@debian:~# █
```

*Ilustración 31: RAID1 - Comprobación de archivos*

Una vez comprobado que el raid no está activo, y que da fallo, se va añadir un disco duro nuevo en el lugar del que hemos retirado anteriormente.

```
root@debian:~# mdadm --add /dev/md0 /dev/sdd
mdadm: added /dev/sdd
```

*Ilustración 32: RAID1 - Añadir disco*

Una vez que se añade el disco duro nuevo, comienza el proceso de reconstrucción como se puede ver en la [Ilustración 49](#), hemos usado esa debido a que el RAID 5 es el más lento y era más fácil sacar la captura, pero el proceso es el mismo.

```
root@debian:~# mdadm -D /dev/md0
/dev/md0:
  Version : 1.2
  Creation Time : Fri Jun  8 01:39:19 2018
  Raid Level : raid1
  Array Size : 4190208 (4.00 GiB 4.29 GB)
  Used Dev Size : 4190208 (4.00 GiB 4.29 GB)
  Raid Devices : 2
  Total Devices : 2
  Persistence : Superblock is persistent

  Update Time : Fri Jun  8 01:47:53 2018
  State : clean
Active Devices : 2
Working Devices : 2
Failed Devices : 0
Spare Devices : 0

    Name : debian:0 (local to host debian)
    UUID : befcc69e:dd035225:7b601bdb:0050eb99
    Events : 45

   Number Major Minor RaidDevice State
     2       8      48         0  active sync  /dev/sdd
     1       8      32         1  active sync  /dev/sdc
root@debian:~# █
```

*Ilustración 33: RAID1 – Añadimos nuevo disco*



Y volvemos a comprobar si los archivos siguen siendo accesibles.

```
root@debian:~# cat /datos/a.txt
2
root@debian:~# cat /datos/b.txt
30
root@debian:~# cat /datos/c.txt
35
root@debian:~#
```

Ilustración 34: RAID1 – Comprobamos archivos

## 2.4. RAID 4

Para crear el [RAID 4](#) utilizaremos 3 discos duros, añadidos como se explica en el paso [2.1](#). Primero utilizamos el comando mdadm indicándole el dispositivo a montar /dev/md0, l4 que indica que es raid 4 y n3 que indica que tiene 3 discos, los indicados después /dev/sdb1, /dev/sdc1 y /dev/sdd1.

```
root@debian:/home/usuario# mdadm -Cv /dev/md0 -l4 -n3 /dev/sdb1 /dev/sdc1 /dev/sdd1
mdadm: chunk size defaults to 512K
mdadm: /dev/sdd1 appears to contain an ext2fs file system
       size=8387584K  mtime=Thu Jan  1 01:00:00 1970
mdadm: size set to 8383488K
Continue creating array? yes
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
root@debian:/home/usuario#
```

Ilustración 35: RAID4 - Comando para crearlo

Para comprobar que se ha montado introducimos el comando "mdadm -D" y vemos el estado del disco y de las 3 unidades que lo forman.

```
root@debian:/home/usuario# mdadm -D /dev/md0
/dev/md0:
  Version : 1.2
  Creation Time : Fri Jun  1 13:08:27 2018
   Raid Level : raid4
   Array Size : 16766976 (15.99 GiB 17.17 GB)
  Used Dev Size : 8383488 (8.00 GiB 8.58 GB)
   Raid Devices : 3
  Total Devices : 3
 Persistence : Superblock is persistent

 Update Time : Fri Jun  1 13:09:28 2018
   State : clean
 Active Devices : 3
 Working Devices : 3
 Failed Devices : 0
 Spare Devices : 0

   Chunk Size : 512K

   Name : debian:0 (local to host debian)
  UUID : 4ed20b9b:028db827:e30b24c3:13df54ef
   Events : 18

   Number Major Minor RaidDevice State
    0         8    17        0 active sync /dev/sdb1
    1         8    33        1 active sync /dev/sdc1
    3         8    49        2 active sync /dev/sdd1
root@debian:/home/usuario#
```

Ilustración 36: RAID4 - Estado



Una vez creado el sistema de almacenamiento debemos darle formato, en este caso será ext3.

```
root@debian:/home/usuario# mke2fs -t ext3 /dev/md0
mke2fs 1.43.4 (31-Jan-2017)
/dev/md0 contiene un sistema de ficheros ext3
última fecha de montaje de /datos Fri Jun 1 11:55:21 2018
Proceed anyway? (y,N) y
```

Ilustración 37: RAID4 – Indicar formato

Creamos archivos con contenido para asegurarnos de que funciona correctamente.

```
root@debian:/# cd datos/
root@debian:/datos# find /etc/init.d | wc -l >> aa.txt
root@debian:/datos# find /etc | wc -l > bb.txt
root@debian:/datos# fdisk -l /dev/md0 > cc.txt
```

Ilustración 38: RAID4 - Creación de archivos

Ahora vamos a provocarle fallo a uno de los discos y ver después el estado del dispositivo. Podemos ver que esta degradado, pero continúa funcionando y para ello vemos que aún están los datos.

```
root@debian:/home/usuario# mdadm /dev/md0 --fail /dev/sdc1
mdadm: set /dev/sdc1 faulty in /dev/md0
root@debian:/home/usuario# mdadm -D /dev/md0
/dev/md0:
    Version : 1.2
  Creation Time : Fri Jun 1 14:08:37 2018
    Raid Level : raid4
    Array Size : 16766976 (15.99 GiB 17.17 GB)
  Used Dev Size : 8383488 (8.00 GiB 8.58 GB)
    Raid Devices : 3
  Total Devices : 3
 Persistence : Superblock is persistent

    Update Time : Fri Jun 1 14:13:44 2018
      State : clean, degraded
Active Devices : 2
Working Devices : 2
Failed Devices : 1
Spare Devices : 0

    Chunk Size : 512K

    Name : debian:0 (local to host debian)
   UUID : 9d368de6:aa6ce9f5:51e3e500:93f38fd7
  Events : 20

   Number Major Minor RaidDevice State
    0       8      17        0 active sync /dev/sdb1
    -       0       0        1 removed
    3       8      49        2 active sync /dev/sdd1
    1       8      33        - faulty /dev/sdc1
root@debian:/home/usuario# ls /datos/
aa.txt bb.txt cc.txt lost+found
```

Ilustración 39: RAID4 – Fallo y posterior estado.



Ahora que hemos provocado un fallo en uno de los dispositivos podemos proceder a borrar la unidad que está fallando como se muestra en la [Ilustración 30](#), para después añadir una unidad nueva que solucione el fallo como podemos ver en la [Ilustración 32](#) perteneciente al raid 1, una vez que hemos añadido la unidad nueva comenzaría el proceso de reconstrucción del raid, como mostramos en la [Ilustración 49](#) del raid 5.

Para seguir comprobando el funcionamiento del RAID 4, quitaremos otro disco y al ver su estado indica FAILED, es decir, que detecta el fallo y perdemos todos los datos.

```
root@debian:/datos# mdadm /dev/md0 --fail /dev/sdc1
mdadm: set /dev/sdc1 faulty in /dev/md0
root@debian:/datos# mdadm -D /dev/md0
/dev/md0:
  Version : 1.2
  Creation Time : Fri Jun 1 13:08:27 2018
  Raid Level : raid4
  Array Size : 16766976 (15.99 GiB 17.17 GB)
  Used Dev Size : 8383488 (8.00 GiB 8.58 GB)
  Raid Devices : 3
  Total Devices : 3
  Persistence : Superblock is persistent

  Update Time : Fri Jun 1 13:37:24 2018
  State : clean, FAILED
Active Devices : 1
Working Devices : 1
Failed Devices : 2
Spare Devices : 0

  Chunk Size : 512K

   Name : debian:0 (local to host debian)
  UUID : 4ed20b9b:028db827:e30b24c3:13df54ef
  Events : 22

Number   Major   Minor   RaidDevice State   /dev/sdb1
  0         8       17         0   active sync
  -         0         0         1   removed
  -         0         0         2   removed

  1         8       33         -   faulty   /dev/sdc1
  3         8       49         -   faulty   /dev/sdd1
```

Ilustración 40: RAID4 – Fallo de dos discos.

## 2.5. RAID 5

Para crear un [RAID 5](#) con mdadm, hemos añadido 5 dispositivos a la máquina virtual, siguiendo los pasos que se explican en el punto [2.1](#), 4 de ellos formarán parte del raid desde el primer momento y otro se usará para sustituir el dispositivo en el que provocamos el fallo. Para empezar, indicamos el dispositivo que creamos /dev/md0, indicamos el tipo de raid con -l5, el número de dispositivos con -n4 y por último indicamos los dispositivos que queremos añadir al raid, en este caso sdb, sdc, sdd, sde.

```
root@debian:/home/usuario# mdadm -Cv /dev/md0 -l5 -n4 /dev/sdb /dev/sdc /dev/sdd /dev/sde
mdadm: layout defaults to left-symmetric
mdadm: layout defaults to left-symmetric
mdadm: chunk size defaults to 512K
mdadm: size set to 8380416K
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

Ilustración 41: RAID5 – Creación del RAID





Para comprobar que se ha creado correctamente podemos usar el comando “lsblk -fm”, con el que podemos observar como los dispositivos sdb, sdc, sdd, sde son miembros de un raid y el raid es md0 como indicamos en el comando de creación.

```
root@debian:/home/usuario# lsblk -fm
NAME        FSTYPE     LABEL        UUID                                 MOUNTPOINT  NAME  SIZE  OWNER  GROUP  MODE
├─sda
│ ├─sda1 ext4              8c22015e-3ab0-43f4-ac73-498ef0b277cd /
│ ├─sda2
│ └─sda5 swap              3e13e863-5bb5-4aa1-8594-fe84d18bcb84 [SWAP]
├─sdb  linux_raid_member  debian:0      cdec7ca3-e2ff-f8d7-dab8-bf71c186628e
├─md0
├─sdc  linux_raid_member  debian:0      cdec7ca3-e2ff-f8d7-dab8-bf71c186628e
├─md0
├─sdd  linux_raid_member  debian:0      cdec7ca3-e2ff-f8d7-dab8-bf71c186628e
├─md0
├─sde  linux_raid_member  debian:0      cdec7ca3-e2ff-f8d7-dab8-bf71c186628e
├─md0
└─sr0  iso9660          VBox_GAs_5.2.1 2017-10-17-20-04-56-16          /media/cdrom0 sr0    57,5M root   cdrom  brw-rw----
```

Ilustración 42: RAID5 – Comprobación con "lsblk -fm"

También podemos comprobar el estado del raid usando el comando “mdadm -D” y el nombre de dispositivos que indicamos cuando se creó, como podemos ver el raid está funcionando y forman parte de él todos los dispositivos que indicamos.

```
root@debian:/home/usuario# mdadm -D /dev/md0
/dev/md0:
  Version : 1.2
  Creation Time : Wed Jun  6 19:12:29 2018
    Raid Level : raid5
    Array Size : 25141248 (23.98 GiB 25.74 GB)
  Used Dev Size : 8380416 (7.99 GiB 8.58 GB)
  Raid Devices : 4
  Total Devices : 4
 Persistence : Superblock is persistent

 Update Time : Wed Jun  6 19:13:52 2018
   State : clean
 Active Devices : 4
 Working Devices : 4
 Failed Devices : 0
 Spare Devices : 0

 Layout : left-symmetric
 Chunk Size : 512K

 Name : debian:0 (local to host debian)
  UUID : 624b5255:5556dd82:1259d603:c95998a3
 Events : 18

Number Major Minor RaidDevice State
 0         8      16           0 active sync /dev/sdb
 1         8      32           1 active sync /dev/sdc
 2         8      48           2 active sync /dev/sdd
 4         8      64           3 active sync /dev/sde
```

Ilustración 43: RAID5 – Creación del RAID



Al ser un raid 5, reparten los datos y la paridad entre todos los discos duros, por lo que su creación también es un poco más lenta, que los demás.

Una vez que está creado y lo hemos comprobado le damos formato al dispositivo.

```
root@debian:/home/usuario# mkfs.ext4 /dev/md0
mke2fs 1.43.4 (31-Jan-2017)
Se está creando un sistema de ficheros con 6285312 bloques de 4k y 1572864 nodos-i
UUID del sistema de ficheros: 3946d72e-d870-4f59-86fa-b07523d08b42
Respaldo del superbloque guardado en los bloques:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000

Reservando las tablas de grupo: hecho
Escribiendo las tablas de nodos-i: hecho
Creando el fichero de transacciones (32768 bloques): hecho
Escribiendo superbloques y la información contable del sistema de ficheros: hecho
```

Ilustración 44: RAID5 – Formato ext4

Ya que le hemos dado formato, podemos pasar a montarlo para usarlo.

```
root@debian:/home/usuario# mount /dev/md0 /raid5/
root@debian:/home/usuario# ls /raid5
aa.txt bb.txt cc.txt dd.txt ee.txt lost+found
```

Ilustración 45: RAID5 – Montar dispositivo

Ahora se provocará un fallo en uno de los dispositivos, con el comando “mdadm --fail”, para ver que no se pierden datos y que sigue funcionando con la posibilidad de cambiar el dispositivo por uno nuevo y que vuelva a funcionar correctamente.

Como podemos ver tras provocar un fallo en uno de los dispositivos, el estado del raid es degradado, pero sigue funcionando.

```
root@debian:/raid5# mdadm --fail /dev/md0 /dev/sde
mdadm: set /dev/sde faulty in /dev/md0
root@debian:/raid5# mdadm -D /dev/md0
/dev/md0:
Version : 1.2
Creation Time : Wed Jun  6 19:12:29 2018
Raid Level : raid5
Array Size : 25141248 (23.98 GiB 25.74 GB)
Used Dev Size : 8380416 (7.99 GiB 8.58 GB)
Raid Devices : 4
Total Devices : 4
Persistence : Superblock is persistent

Update Time : Wed Jun  6 19:52:07 2018
State : clean, degraded
Active Devices : 3
Working Devices : 3
Failed Devices : 1
Spare Devices : 0

Layout : left-symmetric
Chunk Size : 512K

Name : debian:0 (local to host debian)
UUID : 624b5255:5556dd82:1259d603:c95998a3
Events : 20

Number Major Minor RaidDevice State
0        8       16        0 active sync /dev/sdb
1        8       32        1 active sync /dev/sdc
2        8       48        2 active sync /dev/sdd
-         0         0        3 removed
4        8       64        - faulty /dev/sde
```

Ilustración 46: RAID5 – Fallo en un disco





Ahora que hemos provocado un fallo comprobaremos si podemos ver el contenido de los archivos para ver se ha perdido algo.

```
root@debian:/raid5# cat aa.txt
77
root@debian:/raid5# cat bb.txt
77
root@debian:/raid5# cat cc.txt
77
root@debian:/raid5# cat dd.txt
77
root@debian:/raid5# cat ee.txt
77
```

*Ilustración 47: RAID5 – Mostramos archivos*

Como podemos ver no hemos perdido el contenido de ningún archivo.

Ahora eliminaremos el disco duro que tiene el fallo y lo sustituiremos por otro para que el funcionamiento vuelva ser el adecuado.

En primer lugar, tenemos que eliminar el disco duro que está fallando, para esto usaremos el comando “mdadm --remove” e indicaremos primero a que dispositivos queremos borrarle una unidad y luego la unidad que queremos eliminar.

```
root@debian:/raid5# mdadm --remove /dev/md0 /dev/sde
mdadm: hot removed /dev/sde from /dev/md0
root@debian:/raid5# mdadm -D /dev/md0
/dev/md0:
  Version : 1.2
  Creation Time : Wed Jun  6 19:12:29 2018
  Raid Level : raid5
  Array Size : 25141248 (23.98 GiB 25.74 GB)
  Used Dev Size : 8380416 (7.99 GiB 8.58 GB)
  Raid Devices : 4
  Total Devices : 3
  Persistence : Superblock is persistent

  Update Time : Wed Jun  6 19:57:20 2018
  State : clean, degraded
  Active Devices : 3
  Working Devices : 3
  Failed Devices : 0
  Spare Devices : 0

  Layout : left-symmetric
  Chunk Size : 512K

  Name : debian:0 (local to host debian)
  UUID : 624b5255:5556dd82:1259d603:c95998a3
  Events : 21
```

Number	Major	Minor	RaidDevice	State	
0	8	16	0	active sync	/dev/sdb
1	8	32	1	active sync	/dev/sdc
2	8	48	2	active sync	/dev/sdd
-	0	0	3	removed	

*Ilustración 48: RAID5 – Eliminamos un disco*



Como podemos ver en la captura anterior el dispositivo sdf ya no pertenece al raid, por lo que ahora tenemos que añadir uno nuevo raid.

Cuando añadimos el nuevo disco comienza el proceso de reconstrucción.

```
root@debian:/raid5# mdadm --add /dev/md0 /dev/sdf
mdadm: added /dev/sdf
root@debian:/raid5# mdadm -D /dev/md0
/dev/md0:
  Version : 1.2
  Creation Time : Wed Jun  6 19:12:29 2018
  Raid Level : raid5
  Array Size : 25141248 (23.98 GiB 25.74 GB)
  Used Dev Size : 8380416 (7.99 GiB 8.58 GB)
  Raid Devices : 4
  Total Devices : 4
  Persistence : Superblock is persistent

  Update Time : Wed Jun  6 19:58:44 2018
  State : clean, degraded, recovering
  Active Devices : 3
  Working Devices : 4
  Failed Devices : 0
  Spare Devices : 1

  Layout : left-symmetric
  Chunk Size : 512K

  Rebuild Status : 3% complete

  Name : debian:0 (local to host debian)
  UUID : 624b5255:5556dd82:1259d603:c95998a3
  Events : 23

    Number Major Minor RaidDevice State
       0       8      16           0 active sync /dev/sdb
       1       8      32           1 active sync /dev/sdc
       2       8      48           2 active sync /dev/sdd
       4       8      80           3 spare rebuilding /dev/sdf
```

*Ilustración 49: RAID5 – Sustituimos el disco*

Cuando termina el proceso de reconstrucción, el raid vuelve a funcionar correctamente.

```
root@debian:/raid5# mdadm -D /dev/md0
/dev/md0:
  Version : 1.2
  Creation Time : Wed Jun  6 19:12:29 2018
  Raid Level : raid5
  Array Size : 25141248 (23.98 GiB 25.74 GB)
  Used Dev Size : 8380416 (7.99 GiB 8.58 GB)
  Raid Devices : 4
  Total Devices : 4
  Persistence : Superblock is persistent

  Update Time : Wed Jun  6 19:59:51 2018
  State : clean
  Active Devices : 4
  Working Devices : 4
  Failed Devices : 0
  Spare Devices : 0

  Layout : left-symmetric
  Chunk Size : 512K

  Name : debian:0 (local to host debian)
  UUID : 624b5255:5556dd82:1259d603:c95998a3
  Events : 40

    Number Major Minor RaidDevice State
       0       8      16           0 active sync /dev/sdb
       1       8      32           1 active sync /dev/sdc
       2       8      48           2 active sync /dev/sdd
       4       8      80           3 active sync /dev/sdf
```

*Ilustración 50: RAID5 – Comprobamos estado con el nuevo disco*



Ahora provocaremos un fallo dos dispositivos.

```
root@debian:/raid5# mdadm --fail /dev/md0 /dev/sdb
mdadm: set /dev/sdb faulty in /dev/md0
root@debian:/raid5# mdadm --fail /dev/md0 /dev/sdf
mdadm: set /dev/sdf faulty in /dev/md0
```

*Ilustración 51: RAID5 – Fallo en dos discos*

Como podemos ver el estado del raid ahora está en “FAILED” cuando en un raid 5 fallan dos dispositivos los datos son irre recuperables.

```
root@debian:/raid5# mdadm --fail /dev/md0 /dev/sdf
mdadm: set /dev/sdf faulty in /dev/md0
root@debian:/raid5# mdadm -D /dev/md0
/dev/md0:
  Version : 1.2
  Creation Time : Wed Jun  6 19:12:29 2018
    Raid Level : raid5
    Array Size : 25141248 (23.98 GiB 25.74 GB)
  Used Dev Size : 8380416 (7.99 GiB 8.58 GB)
    Raid Devices : 4
  Total Devices : 4
 Persistence : Superblock is persistent

 Update Time : Wed Jun  6 20:13:55 2018
   State : clean, FAILED
Active Devices : 2
Working Devices : 2
Failed Devices : 2
Spare Devices : 0

 Layout : left-symmetric
Chunk Size : 512K

 Name : debian:0 (local to host debian)
  UUID : 624b5255:5556dd82:1259d603:c95998a3
 Events : 66

Number   Major   Minor   RaidDevice State
-         0         0         0      removed
 1         8        32         1  active sync  /dev/sdc
 2         8        48         2  active sync  /dev/sdd
-         0         0         3      removed

 4         8        80         -    faulty   /dev/sdf
 5         8        16         -    faulty   /dev/sdb
```

*Ilustración 52: RAID5 – Vemos estado*

Como podemos ver los datos de los archivos se han perdido.

```
root@debian:/raid5# cat aa.txt
cat: aa.txt: Error de entrada/salida
root@debian:/raid5# cat bb.txt
cat: bb.txt: Error de entrada/salida
root@debian:/raid5# cat cc.txt
cat: cc.txt: Error de entrada/salida
root@debian:/raid5# cat dd.txt
cat: dd.txt: Error de entrada/salida
root@debian:/raid5# cat ee.txt
cat: ee.txt: Error de entrada/salida
root@debian:/raid5#
```

*Ilustración 53: RAID5 – Perdida de datos*



## 2.6. RAID 10

En el caso del [RAID 10](#) vamos a realizarlo de dos formas diferentes.

Primero durante la instalación del sistema operativo Debian 9.4, ya que cuando lo instalas te permite la opción de crear los raid 0, 1, 5, 6 y 10 de forma gráfica.

La segunda opción es la forma que hemos utilizado anteriormente, mediante comandos con el paquete mdadm.

### 2.6.1. RAID 10 – DURANTE INSTALACIÓN DEBIAN

Durante la instalación debe salir esta pantalla y lo que vamos a hacer es seleccionar *Particionado guiado*.

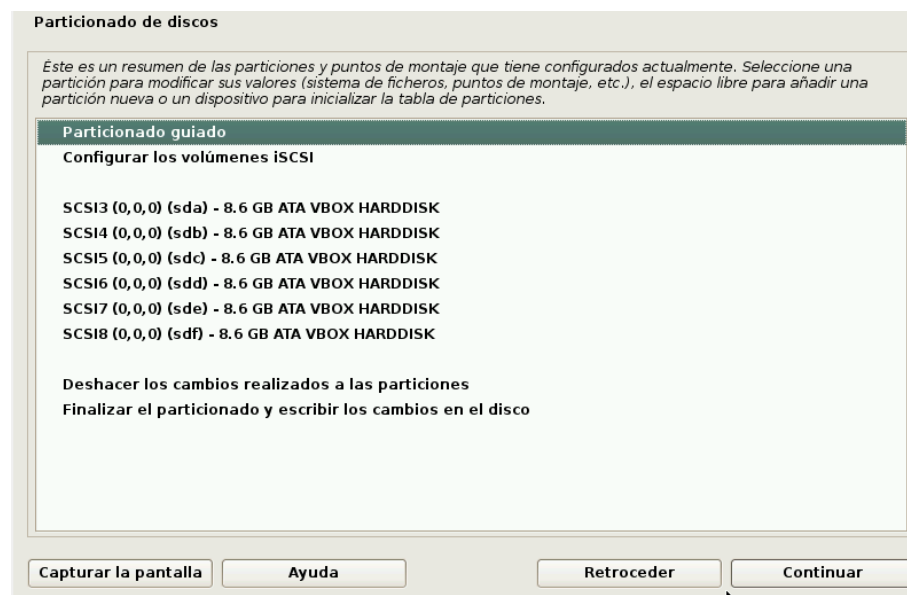


Ilustración 54: RAID10 – Partición guiada



El siguiente paso es seleccionar *Utilizar todo el disco* en el caso de que queramos usar el disco completo.



Ilustración 55: RAID10 – Utilizar todo el disco

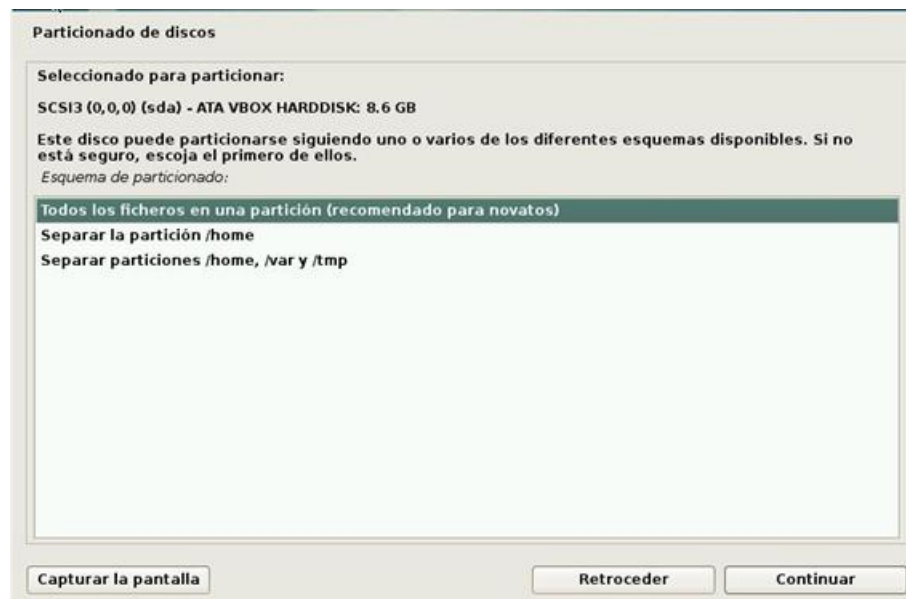
Elegimos el disco que tendrá almacenado el sistema operativo en este caso el disco sda.



Ilustración 56: RAID10 – Elegir disco

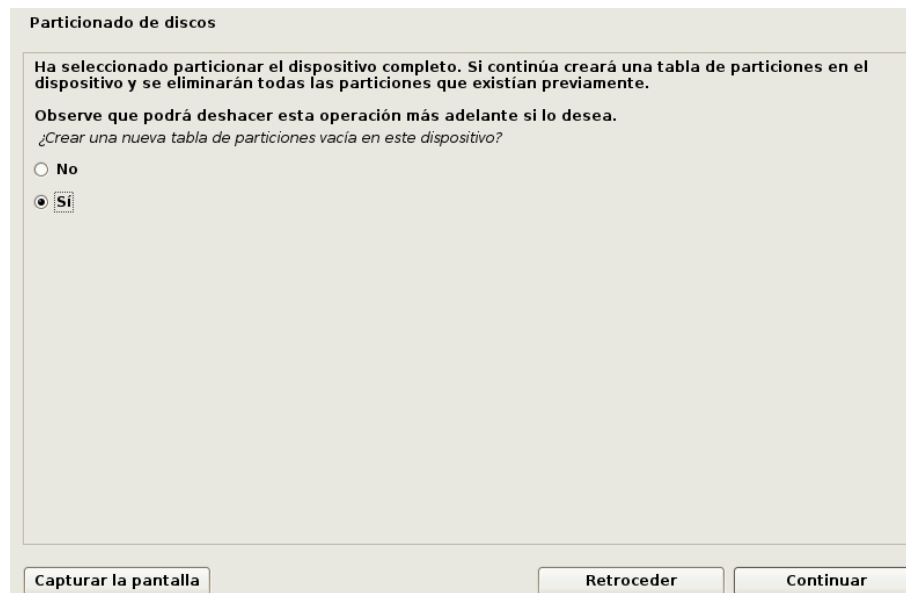


Seleccionamos la primera opción en el caso de que no queramos separar los distintos directorios en particiones.



*Ilustración 57: RAID10 – Esquema de particionado*

Debemos particionar los discos que utilizaremos para el RAID, para ello pulsaremos sobre cada uno de ellos (sdb, sdc, sde, sdd y sdf) y pulsaremos en sí para guardar los cambios.



*Ilustración 62: RAID10 – Guardar particiones de disco*



Hecho lo anterior, debemos confirmar que los discos se encuentran en el mismo estado que la fotografía y ya podremos pulsar en *configurar RAID por software*.



Ilustración 58: RAID10 – Crear RAID por software

Elegimos *Sí* para que se formateen los discos, y configuremos el RAID.

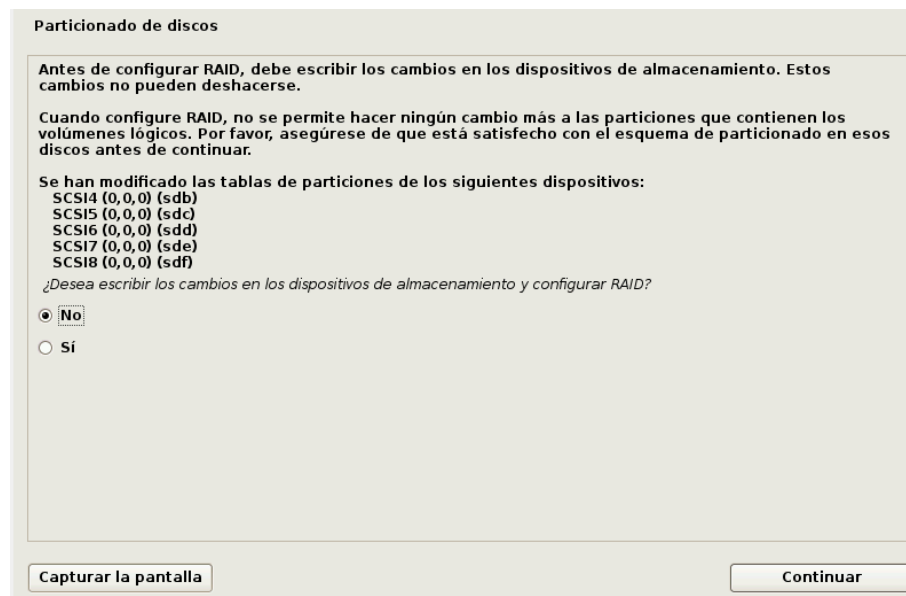
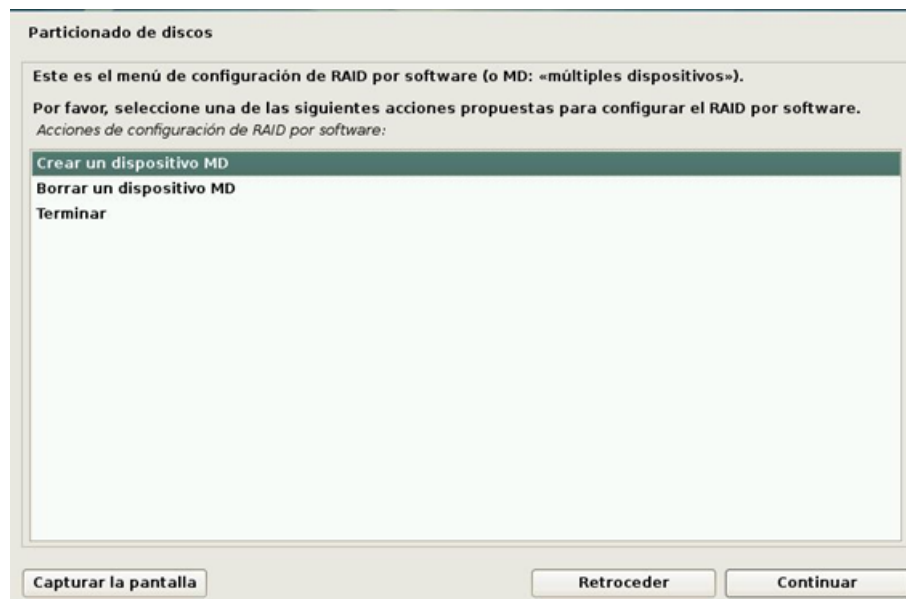


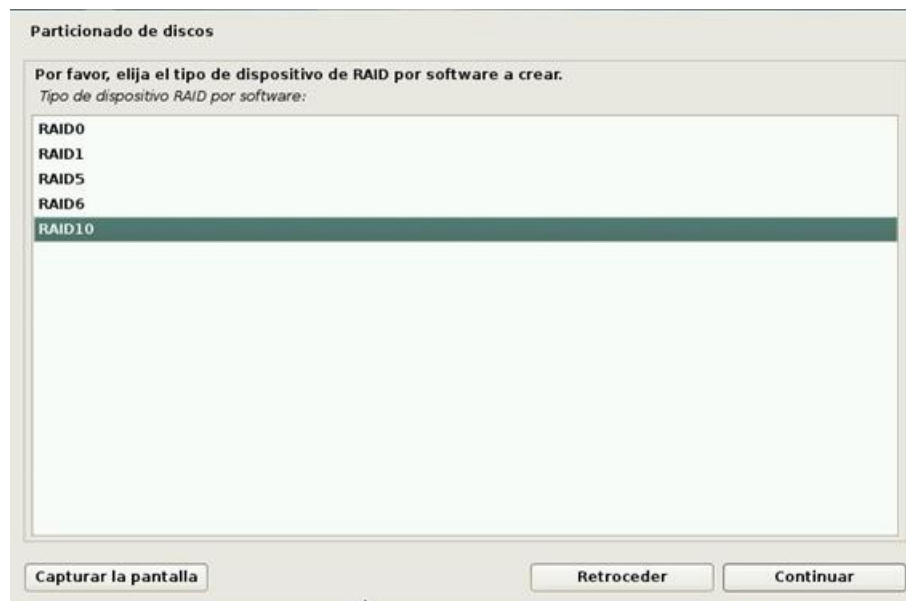
Ilustración 59: RAID10 – Guardar cambios

Pasamos a configurar el RAID, y para ello lo primero que tenemos que hacer es crear el dispositivo.



*Ilustración 60: RAID10 – Crear un dispositivo MD*

Elegimos el tipo de RAID que queremos crear, que en este caso será el RAID10.



*Ilustración 61: RAID10 – Elegir tipo RAID*





Seleccionamos el número de dispositivos que vamos a utilizar.

**Particionado de discos**

El array RAID10 estará formado tanto por particiones activas como libres. Las particiones activas son las que están siendo utilizadas, mientras que los dispositivos libres sólo se utilizarán si falla uno o más de los dispositivos activos. Se necesitan al menos 2 dispositivos activos.

**NOTA:** este valor no puede cambiarse más adelante.  
*Número de dispositivos activos para el array RAID10:*

Capturar la pantalla

RetrocederContinuar

*Ilustración 62: RAID10 – Numero de discos activos*

Y el número de dispositivos que estarán libres.

**Particionado de discos**

Número de dispositivos libres para el array RAID10:

1|

[Capturar la pantalla] [Retroceder] [Continuar]

*Ilustración 63: RAID10 – dispositivos libres*



Seleccionamos los dispositivos que van a estar activos.

Particionado de discos

Ha elegido crear un array RAID10 con 4 dispositivos activos.

Por favor, escoja qué particiones son dispositivos activos. Debe seleccionar 4 particiones.

Dispositivos activos para el array RAID10:

<input type="checkbox"/>	/dev/sda1	(7515MB; ext4)
<input type="checkbox"/>	/dev/sda5	(1071MB; intercambio)
<input checked="" type="checkbox"/>	/dev/sdb free #1	(8589MB; ESPACIO LIBRE)
<input checked="" type="checkbox"/>	/dev/sdc free #1	(8589MB; ESPACIO LIBRE)
<input checked="" type="checkbox"/>	/dev/sdd free #1	(8589MB; ESPACIO LIBRE)
<input checked="" type="checkbox"/>	/dev/sde free #1	(8589MB; ESPACIO LIBRE)
<input type="checkbox"/>	/dev/sdf free #1	(8589MB; ESPACIO LIBRE)

Capturar la pantalla Retroceder Continuar

Ilustración 64: RAID10 – Discos activos

Y también seleccionamos los dispositivos que van a estar libres, es decir, que podrá ser utilizado en caso de que se estropee o falle uno de los discos.

Particionado de discos

Ha elegido crear un array RAID10 con 1 dispositivos libres.

Por favor, escoja las particiones que se utilizarán como particiones libres. Puede elegir hasta 1 particiones. Si escoge menos de 1 dispositivos, las demás particiones se añadirán al array como «no disponibles» y podrán añadirse más adelante.

Dispositivos libres para el array RAID10:

<input type="checkbox"/>	/dev/sda1	(7515MB; ext4)
<input type="checkbox"/>	/dev/sda5	(1071MB; intercambio)
<input checked="" type="checkbox"/>	/dev/sdf free #1	(8589MB; ESPACIO LIBRE)

Capturar la pantalla Retroceder Continuar

Ilustración 65: RAID10 – Disco libre



Hecho lo anterior volvemos escribir los cambios que se han realizado en los distintos dispositivos. Pulsamos *Sí*.

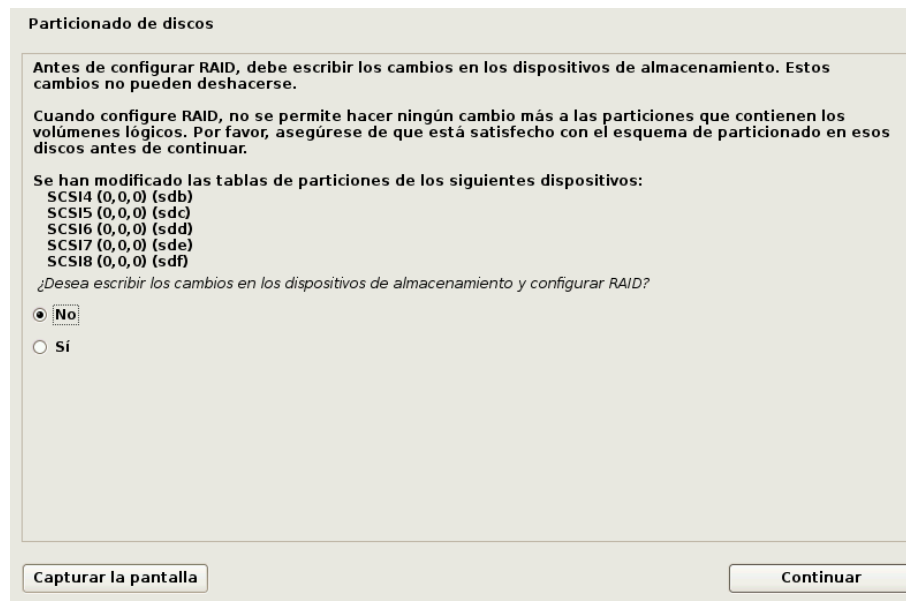


Ilustración 66: RAID10 – Guardar cambios

Una vez que hemos realizado todo lo anterior, seleccionamos *Terminar*.

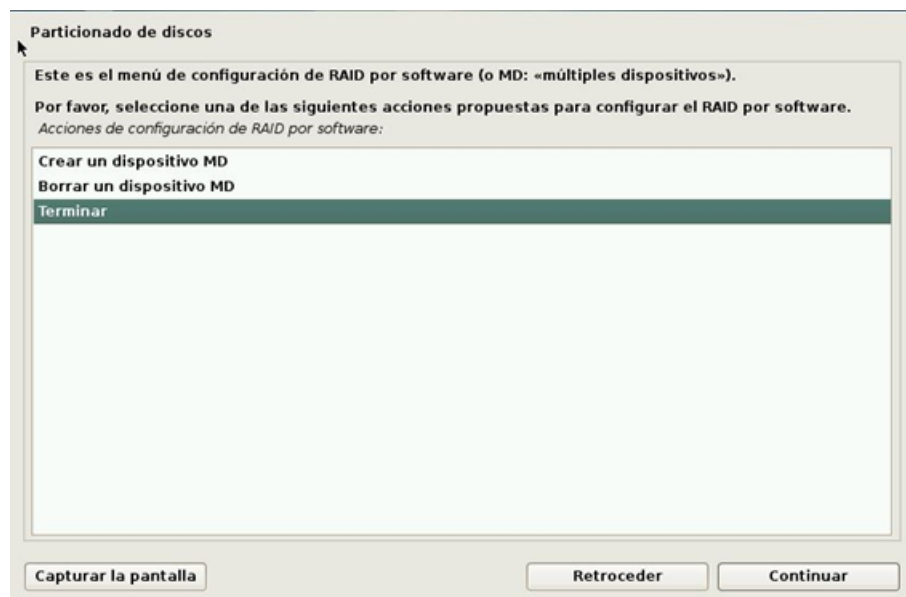


Ilustración 67: RAID10 – Terminar

Antes de finalizar nos aparecerá un resumen con todos los dispositivos y cómo va a estar configurado cada dispositivo. Si todo está correcto seleccionamos *Finalizar el particionado y escribir los cambios en el disco*.

Y ya tendríamos creado nuestro RAID.

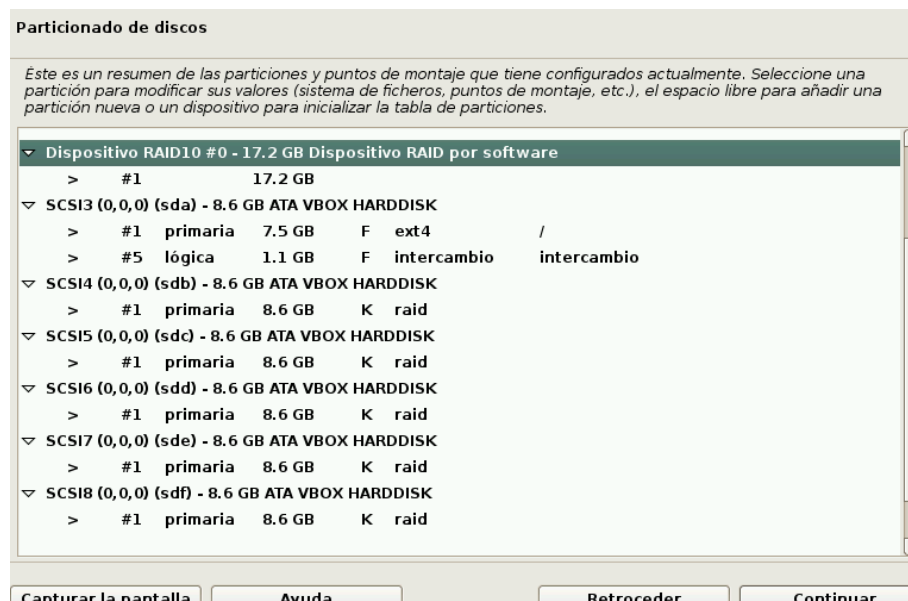


Ilustración 68: RAID10 – Proceso terminado

## 2.6.2. RAID 10 –DESPUES DE INSTALAR DEBIAN

\*Como aclaración inicial, al apagar el ordenador cambia el nombre del dispositivo de md0 a md127.

Para el montaje de un raid 10 se utilizarán 4 unidades de disco, las cuales se añaden como se explica en el punto [2.1](#). Una vez creadas las particiones como hemos hecho en los anteriores casos, procedemos a crear el raid 10 con el comando "mdadm -Cv /dev/md0 --level=raid10 --raid-devices =4 /dev/sdb1 /dev/sdc1 /dev/sde1 /dev/sdf1".

```
root@debian:/# mdadm -Cv /dev/md0 --level=raid10 --raid-devices=4 /dev/sdb1 /dev/sdc1 /dev/sde1 /dev/sdf1
mdadm: layout defaults to n2
mdadm: layout defaults to n2
mdadm: chunk size defaults to 512K
mdadm: size set to 8382976K
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

Ilustración 69: RAID10 – Creación



Comprobamos que se ha creado correctamente con "mdadm -d /dev/md127"

```

root@debian:/# mdadm -D /dev/md127
/dev/md127:
  Version : 1.2
  Creation Time : Thu Jun  7 00:22:32 2018
  Raid Level : raid10
  Array Size : 16765952 (15.99 GiB 17.17 GB)
  Used Dev Size : 8382976 (7.99 GiB 8.58 GB)
  Raid Devices : 4
  Total Devices : 4
  Persistence : Superblock is persistent

  Update Time : Thu Jun  7 00:29:46 2018
  State : clean, resyncing (PENDING)
  Active Devices : 4
  Working Devices : 4
  Failed Devices : 0
  Spare Devices : 0

  Layout : near=2
  Chunk Size : 512K

  Name : debian:0 (local to host debian)
  UUID : ee597a99:108d4542:4809b289:0cbc343e
  Events : 18

   Number   Major   Minor   RaidDevice State
    0         8       17         0     active sync set-A  /dev/sdb1
    1         8       33         1     active sync set-B  /dev/sdc1
    2         8       49         2     active sync set-A  /dev/sdd1
    3         8       65         3     active sync set-B  /dev/sde1
root@debian:/# █

```

*Ilustración 70: RAID10 – Comprobación de estado*

En esta segunda carpeta vemos que unos discos forman parte del set-A y otros de set-B. ¿Qué quiere decir esto? Como vemos en la imagen siguiente los discos se agrupan de 2 en 2, cada uno de estos grupos es un set.

La imagen del raid 10 será importante para entenderlo por lo que podemos verla en el apartado [1.1.6](#).

Primero creamos la carpeta donde vamos a montar el dispositivo y después lo montamos.

```

root@debian:~# mount /dev/md0 datos

```

*Ilustración 71: RAID10 – Montamos la carpeta*



Creamos archivos dentro de la carpeta datos para comprobar que no se dañan al fallar uno de los discos y muestro su contenido.

```
root@debian:/datos# cat hola.txt
24
root@debian:/datos# cat raid.txt
Disco /dev/md127: 16 GiB, 17168334848 bytes, 33531904 sectores
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 524288 bytes / 1048576 bytes
root@debian:/datos# cat a.txt
bin
boot
datos
dev
etc
home
initrd.img
initrd.img.old
lib
lost+found
media
mnt
opt
proc
root
run
sbin
srv
sys
```

Ilustración 72: RAID10 – Creamos archivos

Primero vamos a eliminar un disco sdb1 y sustituirlo por otro sdf1.

```
root@debian:/# mdadm /dev/md127 --fail /dev/sdb1
mdadm: set /dev/sdb1 faulty in /dev/md127
root@debian:/# mdadm /dev/md127 --remove /dev/sdb1
mdadm: hot removed /dev/sdb1 from /dev/md127
root@debian:/# mdadm -D /dev/md127
/dev/md127:
  Version : 1.2
  Creation Time : Thu Jun 7 00:22:32 2018
  Raid Level : raid10
  Array Size : 16765952 (15.99 GiB 17.17 GB)
  Used Dev Size : 8382976 (7.99 GiB 8.58 GB)
  Raid Devices : 4
  Total Devices : 3
  Persistence : Superblock is persistent

  Update Time : Thu Jun 7 11:00:28 2018
  State : clean, degraded
  Active Devices : 3
  Working Devices : 3
  Failed Devices : 0
  Spare Devices : 0

  Layout : near=2
  Chunk Size : 512K

  Name : debian:0 (local to host debian)
  UUID : ee597a99:108d4542:4809b289:0cbc343e
  Events : 40

  Number Major Minor RaidDevice State
    -   -   -   -   -
    1     8    33     1   active sync set-B /dev/sdc1
    2     8    49     2   active sync set-A /dev/sdd1
    3     8    65     3   active sync set-B /dev/sdf1
root@debian:/#
```

Ilustración 73: RAID10 – eliminamos un disco



Para añadirlo utilizamos el comando "mdadm /dev/md0 --add /dev/sdf1".

```

Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@debian:/# mdadm -D /dev/md127
/dev/md127:
  Version : 1.2
  Creation Time : Thu Jun  7 00:22:32 2018
  Raid Level : raid10
  Array Size : 16765952 (15.99 GiB 17.17 GB)
  Used Dev Size : 8382976 (7.99 GiB 8.58 GB)
  Raid Devices : 4
  Total Devices : 4
  Persistence : Superblock is persistent

  Update Time : Thu Jun  7 11:12:55 2018
  State : clean
  Active Devices : 4
  Working Devices : 4
  Failed Devices : 0
  Spare Devices : 0

  Layout : near=2
  Chunk Size : 512K

  Name : debian:0 (local to host debian)
  UUID : ee597a99:108d4542:4809b289:0cbc343e
  Events : 60

  Number  Major  Minor  RaidDevice State
    4         8      81         0  active sync set-A  /dev/sdf1
    1         8      33         1  active sync set-B  /dev/sdc1
    2         8      49         2  active sync set-A  /dev/sdd1
    3         8      65         3  active sync set-B  /dev/sde1
root@debian:/#

```

Ilustración 74: RAID10 – Sustituimos disco

Una vez que hemos añadido comienza el proceso de reconstrucción del raid, podemos ver la [Ilustración 49](#) para ver cual sería el estado durante el proceso.

Volvemos a mirar los archivos para ver si se han dañado con el comando "cat nombre\_archivo" y comprobamos que no.

```

root@debian:/datos# cat hola.txt
24
root@debian:/datos# cat raid.txt
Disco /dev/md127: 16 GiB, 17168334848 bytes, 33531904 sectores
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 524288 bytes / 1048576 bytes
root@debian:/datos# cat a.txt
bin
boot
datos
dev
etc
home
initrd.img
initrd.img.old
lib
lost+found
media
mnt
opt
proc
root
run
sbin
srv
sys

```

Ilustración 75: RAID10 – Vemos los datos



Vamos a probar ahora a ver los datos sin uno de los discos.

```
usuario@debian: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
root@debian:/datos# mdadm /dev/md127 --fail /dev/sdc1  
mdadm: set /dev/sdc1 faulty in /dev/md127  
root@debian:/datos# cat hola.txt  
24  
root@debian:/datos# cat raid.txt  
Disco /dev/md127: 16 GiB, 17168334848 bytes, 33531904 sectores  
Unidades: sectores de 1 * 512 = 512 bytes  
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes  
Tamaño de E/S (mínimo/óptimo): 524288 bytes / 1048576 bytes  
root@debian:/datos# cat a.txt  
bin  
boot  
datos  
dev  
etc  
home  
initrd.img  
initrd.img.old  
lib  
lost+found  
media  
mnt  
opt  
proc  
root  
run  
sbin  
srv  
sys  
tmp  
usr  
var
```

Ilustración 76: RAID10 – Fallo de disco

Vemos que los archivos no están dañados así que quitamos otro disco.

```
root@debian:/datos# mdadm /dev/md127 --fail /dev/sdd1  
mdadm: set /dev/sdd1 faulty in /dev/md127  
root@debian:/datos# cat hola.txt  
24  
root@debian:/datos# █
```

Ilustración 77: RAID10 – Fallo de dos discos

Eliminamos sdd1 y vemos los datos de a.txt que se mantienen. Pasamos a ver el estado del RAID.

```
root@debian:/datos# cat hola.txt  
24  
root@debian:/datos# mdadm -D /dev/md127  
/dev/md127:  
Version : 1.2  
Creation Time : Thu Jun 7 00:22:32 2018  
Raid Level : raid10  
Array Size : 16765952 (15.99 GiB 17.17 GB)  
Used Dev Size : 8382976 (7.99 GiB 8.58 GB)  
Raid Devices : 4  
Total Devices : 4  
Persistence : Superblock is persistent  
  
Update Time : Thu Jun 7 12:03:25 2018  
State : clean, degraded  
Active Devices : 2  
Working Devices : 2  
Failed Devices : 2  
Spare Devices : 0  
  
Layout : near=2  
Chunk Size : 512K  
  
Name : debian:0 (local to host debian)  
UUID : ee597a99:108d4542:4809b289:0cbc343e  
Events : 70  
  
Number Major Minor RaidDevice State  
4 8 81 0 active sync set-A /dev/sdf1  
- 0 0 1 removed  
- 0 0 2 removed  
3 8 65 3 active sync set-B /dev/sde1  
  
1 8 33 - faulty /dev/sdc1  
2 8 49 - faulty /dev/sdd1
```

Ilustración 78: RAID10 – Estado sin dos discos





Vemos que el RAID está degradado, pero mantiene los archivos. Ahora al intentar borrar otro disco no nos lo permite porque ya solo queda un disco del set-A y otro disco del set-B. Si borrásemos uno de los discos no puede haber un raid 10 y por eso no te lo permite.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@debian:/datos# mdadm /dev/md127 --fail /dev/sde1
mdadm: set device faulty failed for /dev/sde1: Device or resource busy
root@debian:/datos# mdadm /dev/md127 --remove /dev/sde1
mdadm: hot remove failed for /dev/sde1: Device or resource busy
root@debian:/datos# █
```

*Ilustración 79: RAID10 – Fallo de 3 discos*

Queda comprobado que un raid 10 puede funcionar fallando hasta dos discos y mantiene todos los datos.

## 2.7. RAID 01

En este apartado vamos a ver cómo crear un [RAID01](#), y para ello emplearemos cuatro discos, los cuales se añaden como indicamos en el punto anterior [2.1](#), de esas cuatro dos se usaran para cada RAID0, y posteriormente con cada uno de los RAID0 formaremos el RAID1. Tendremos que realizarlo de esta forma porque mdadm no nos permite crear un RAID01 directamente, y por lo tanto tendremos que utilizar RAID anidados.

Para entender más fácilmente este raid puede ser útil ver la imagen que se encuentra en el apartado [1.1.5](#).

Lo primero que vamos a realizar es crear los dos RAID0 para ello utilizaremos mdadm indicando tipo de RAID que vamos a crear, el número de dispositivos que vamos a utilizar y el nombre de los mismos.

```
root@debian:~# mdadm -Cv --create /dev/md0 --level=0 --raid-devices=2 /dev/sdb /dev/sdc
mdadm: chunk size defaults to 512K
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
root@debian:~# mdadm -Cv --create /dev/md1 --level=0 --raid-devices=2 /dev/sdd /dev/sde
mdadm: chunk size defaults to 512K
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md1 started.
root@debian:~# █
```

*Ilustración 80: RAID 01 – Creación del RAID 0*

Después de esto, comprobamos con el comando `lsblk -fm` si se han creado correctamente.



```

usuario@debian: ~
Archivo Editar Ver Buscar Terminal Ayuda
root@debian:~# lsblk -fm
NAME        FSTYPE LABEL        UUID                                MOUNTPOINT NAME        SIZE OWNER GROUP MODE
├─sda
│ ├─sda1 ext4              e545b97c-1d62-471c-8271-2a04eac119be /
│ ├─sda2
│ └─sda5 swap              36254647-728b-4e00-a4c5-bac7115f4e98 [SWAP]
├─sdb linux_raid_member debian:0 582d8aec-a09d-efbf-901e-fc7b56b31027
├─md0
├─sdc linux_raid_member debian:0 582d8aec-a09d-efbf-901e-fc7b56b31027
├─md0
├─sdd linux_raid_member debian:1 e9f380a1-489e-5601-8802-7266fedb2e68
├─md1
├─sde linux_raid_member debian:1 e9f380a1-489e-5601-8802-7266fedb2e68
├─md1
├─sdf
└─sro 1024M root  cdrom brw-rw----
root@debian:~#

```

Ilustración 81: RAID 01 - Comprobación

Y ahora pasamos a crear el RAID1 tal y como lo hemos hecho anteriormente con los RAID0.

```

root@debian:~# mdadm -Cv --create /dev/md2 --level=1 --raid-devices=2 /dev/md0 /dev/md1
mdadm: Note: this array has metadata at the start and
may not be suitable as a boot device.  If you plan to
store '/boot' on this device please ensure that
your boot-loader understands md/v1.x metadata, or use
--metadata=0.90
mdadm: size set to 8376320K
Continue creating array? yes
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md2 started.
root@debian:~#

```

Ilustración 82: RAID 01 - Creación del RAID 1

Y volvemos a comprobar si se ha creado correctamente.

```

root@debian:~# lsblk -fm
NAME        FSTYPE LABEL        UUID                                MOUNTPOINT NAME        SIZE OWNER GROUP MODE
├─sda
│ ├─sda1 ext4              e545b97c-1d62-471c-8271-2a04eac119be /
│ ├─sda2
│ └─sda5 swap              36254647-728b-4e00-a4c5-bac7115f4e98 [SWAP]
├─sdb linux_raid_member debian:0 582d8aec-a09d-efbf-901e-fc7b56b31027
├─md0 linux_raid_member debian:2 f686df61-2401-8ff0-061f-af72f776552e
├─md2
├─sdc linux_raid_member debian:0 582d8aec-a09d-efbf-901e-fc7b56b31027
├─md0 linux_raid_member debian:2 f686df61-2401-8ff0-061f-af72f776552e
├─md2
├─sdd linux_raid_member debian:1 e9f380a1-489e-5601-8802-7266fedb2e68
├─md1 linux_raid_member debian:2 f686df61-2401-8ff0-061f-af72f776552e
├─md2
├─sde linux_raid_member debian:1 e9f380a1-489e-5601-8802-7266fedb2e68
├─md1 linux_raid_member debian:2 f686df61-2401-8ff0-061f-af72f776552e
├─md2
├─sdf
└─sro 1024M root  cdrom brw-rw----
root@debian:~#

```

Ilustración 83: RAID 01 - Comprobación

Le damos formato al dispositivo md2.

```

root@debian:~# mkfs.ext4 /dev/md2
mke2fs 1.43.4 (31-Jan-2017)
Se está creando un sistema de ficheros con 2094080 bloques de 4k y 524288 nodos-i
UUID del sistema de ficheros: ca959b8f-f535-4c67-9237-5327fde3e87c
Respaldo del superbloque guardado en los bloques:
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632
Reservando las tablas de grupo: hecho
Escribiendo las tablas de nodos-i: hecho
Creando el fichero de transacciones (16384 bloques): hecho
Escribiendo superbloques y la información contable del sistema de ficheros:
hecho

```

Ilustración 84: RAID 01 - Formato ext4



Y montamos el dispositivo en la carpeta seleccionada.

```

Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@debian:~# mount /dev/md2 datos
root@debian:~# ls datos/
lost+found
root@debian:~# █

```

*Ilustración 85: RAID 01 – Montar el dispositivo*

Comprobamos que podemos ver el contenido de los ficheros.

```

root@debian:~# cat /datos/a.txt
1
root@debian:~# cat /datos/b.txt
61
root@debian:~# cat /datos/c.txt
30
root@debian:~# █

```

*Ilustración 86: RAID 01 – Estado de los archivos*

Ahora vamos a provocar un fallo en uno de los dos discos, pero como vemos no nos permite provocar el fallo mediante comandos puesto que el disco duro el cual estamos intentando que falle, forma parte de un RAID0 y en ese momento se está usando. Por lo que vamos a quitar el disco duro de forma manual mediante VirtualBox.

```

root@debian:~# mdadm /dev/md0 --fail /dev/sdb
mdadm: set device faulty failed for /dev/sdb: Device or resource busy
root@debian:~# mdadm -D /dev/md0
/dev/md0:
  Version : 1.2
  Creation Time : Thu Jun  7 23:32:46 2018
    Raid Level : raid0
    Array Size : 8388616 (7.99 GiB 8.58 GB)
    Raid Devices : 2
    Total Devices : 2
    Persistence : Superblock is persistent

    Update Time : Thu Jun  7 23:32:46 2018
      State : clean
    Active Devices : 2
    Working Devices : 2
    Failed Devices : 0
    Spare Devices : 0

    Chunk Size : 512K

    Name : debian:0 (local to host debian)
    UUID : 0c9af094:5ddf55b8:3bf819cc:4a962535
    Events : 0

  Number Major Minor RaidDevice State
     0      8     16          0 active sync  /dev/sdb
     1      8     32          1 active sync  /dev/sdc
root@debian:~# █

```

*Ilustración 87: RAID 01 – Fallo de un disco*

Una vez quitado el disco duro, vamos a ver el estado de los RAID que hemos creado. Como vemos el RAID0 que habíamos creado se han quedado con un solo disco.



```

root@debian:~# mdadm -D /dev/md125
/dev/md125:
  Version : 1.2
  Raid Level : raid0
  Total Devices : 1
  Persistence : Superblock is persistent

  State : inactive

  Name : debian:2 (local to host debian)
  UUID : 2f73915d:bbb0e30d:d891a103:5aa16310
  Events : 19

  Number Major Minor RaidDevice
    -     9    127      -         /dev/md/debian:1
root@debian:~# mdadm -D /dev/md126
/dev/md126:
  Version : 1.2
  Raid Level : raid0
  Total Devices : 1
  Persistence : Superblock is persistent

  State : inactive

  Name : debian:0 (local to host debian)
  UUID : 42da678e:e8758094:72840cc4:3b689db6
  Events : 0

  Number Major Minor RaidDevice
    -     8     16      -         /dev/sdb

```

Ilustración 88: RAID 01 – Estado

Y el RAID1, aunque sigue teniendo dos dispositivos ha cambiado a RAID0 automáticamente.

```

root@debian:~# mdadm -D /dev/md127
/dev/md127:
  Version : 1.2
  Creation Time : Fri Jun  8 09:11:58 2018
  Raid Level : raid0
  Array Size : 8380416 (7.99 GiB 8.58 GB)
  Raid Devices : 2
  Total Devices : 2
  Persistence : Superblock is persistent

  Update Time : Fri Jun  8 09:11:58 2018
  State : clean
  Active Devices : 2
  Working Devices : 2
  Failed Devices : 0
  Spare Devices : 0

  Chunk Size : 512K

  Name : debian:1 (local to host debian)
  UUID : 0e38d49e:b04e49ed:33f3e300:03c9691c
  Events : 0

  Number Major Minor RaidDevice State
    0         8     32         0 active sync /dev/sdc
    1         8     48         1 active sync /dev/sdd

```

Ilustración 89: RAID 01 – Estado

Por último, vamos a ver el estado de los datos, y como podemos ver los datos se han perdido y es imposible recuperarlos.

```

root@debian:~# ls /datos
root@debian:~# █

```

Ilustración 90: RAID 01 – Listado datos tras perdida

## 2.8. RAID 50

Para realizar un [RAID 50](#) con mdadm debemos hacer uso de raid anidados debido que no tiene implementado este raid para realizarlo de manera directa, esto puede resultar un problema debido a que no trabaje de manera correcta cuando se produzcan fallos.

En primer lugar, para crear un raid 50 añadiremos 8 unidades de disco a la máquina virtual como lo se ha hecho en los anteriores casos y se explica en el apartado [2.1](#), de los cuales 4 formarán un raid 5, los otros 4 otro raid 5 y estos dos raid 5 juntos formarán un raid 0.

En primero lugar crearemos los dos raid 5 y una vez que los tenemos creados crearemos un raid 0 con estos dos dispositivos.

Para entender el raid 50 más fácilmente puede ser de gran ayuda ver la imagen que encontramos en el apartado [1.1.7](#).

Como podemos ver en la siguiente imagen los dispositivos sd[bcd] forman el raid denominado md0, los dispositivos sd[fghi] forman el raid denominado md1 y a su vez estos forman el raid md2.

NAME	FSTYPE	LABEL	UUID	MOUNTPPOINT	NAME	SIZE	OWNER	GROUP	MODE
sda					sda	8G	root	disk	brw-rw----
└sda1	ext4		8c22015e-3ab0-43f4-ac73-498ef0b277cd	/	└sda1	7G	root	disk	brw-rw----
└sda2					└sda2	1K	root	disk	brw-rw----
└sda5	swap		3e13e863-5bb5-4aa1-8594-fe84d18bcb84	[SWAP]	└sda5	1022M	root	disk	brw-rw----
sdb	linux_raid_member	debian:0	dc690204-87cb-27d1-f72b-eab9300c688c		sdb	8G	root	disk	brw-rw----
└md0	linux_raid_member	debian:2	ac0f1199-ac4f-2940-d005-6aeabe68cded		└md0	24G	root	disk	brw-rw----
└└md2					└└md2	47,9G	root	disk	brw-rw----
sdc	linux_raid_member	debian:0	dc690204-87cb-27d1-f72b-eab9300c688c		sdc	8G	root	disk	brw-rw----
└md0	linux_raid_member	debian:2	ac0f1199-ac4f-2940-d005-6aeabe68cded		└md0	24G	root	disk	brw-rw----
└└md2					└└md2	47,9G	root	disk	brw-rw----
sdd	linux_raid_member	debian:0	dc690204-87cb-27d1-f72b-eab9300c688c		sdd	8G	root	disk	brw-rw----
└md0	linux_raid_member	debian:2	ac0f1199-ac4f-2940-d005-6aeabe68cded		└md0	24G	root	disk	brw-rw----
└└md2					└└md2	47,9G	root	disk	brw-rw----
sde	linux_raid_member	debian:0	dc690204-87cb-27d1-f72b-eab9300c688c		sde	8G	root	disk	brw-rw----
└md0	linux_raid_member	debian:2	ac0f1199-ac4f-2940-d005-6aeabe68cded		└md0	24G	root	disk	brw-rw----
└└md2					└└md2	47,9G	root	disk	brw-rw----
sdf	linux_raid_member	debian:1	991523a6-04ea-0a2a-5f89-8d73993d1e14		sdf	8G	root	disk	brw-rw----
└md1	linux_raid_member	debian:2	ac0f1199-ac4f-2940-d005-6aeabe68cded		└md1	24G	root	disk	brw-rw----
└└md2					└└md2	47,9G	root	disk	brw-rw----
sdg	linux_raid_member	debian:1	991523a6-04ea-0a2a-5f89-8d73993d1e14		sdg	8G	root	disk	brw-rw----
└md1	linux_raid_member	debian:2	ac0f1199-ac4f-2940-d005-6aeabe68cded		└md1	24G	root	disk	brw-rw----
└└md2					└└md2	47,9G	root	disk	brw-rw----
sdh	linux_raid_member	debian:1	991523a6-04ea-0a2a-5f89-8d73993d1e14		sdh	8G	root	disk	brw-rw----
└md1	linux_raid_member	debian:2	ac0f1199-ac4f-2940-d005-6aeabe68cded		└md1	24G	root	disk	brw-rw----
└└md2					└└md2	47,9G	root	disk	brw-rw----
sdi	linux_raid_member	debian:1	991523a6-04ea-0a2a-5f89-8d73993d1e14		sdi	8G	root	disk	brw-rw----
└md1	linux_raid_member	debian:2	ac0f1199-ac4f-2940-d005-6aeabe68cded		└md1	24G	root	disk	brw-rw----
└└md2					└└md2	47,9G	root	disk	brw-rw----
sdj					sdj	8G	root	disk	brw-rw----
sdk					sdk	8G	root	disk	brw-rw----
sr0	iso9660	VBox_GAs_5.2.1	2017-10-17-20-04-56-16	/media/cdrom0	sr0	57,5M	root	cdrom	brw-rw----

Ilustración 91: RAID 50 – RAID creado





Podemos comprobar el raid con el comando “mdadm -D” y vemos que su funcionamiento es correcto.

```
root@debian:/home/usuario# mdadm -D /dev/md2
/dev/md2:
  Version : 1.2
  Creation Time : Sat Jun  9 21:20:28 2018
  Raid Level : raid0
  Array Size : 50249728 (47.92 GiB 51.46 GB)
  Raid Devices : 2
  Total Devices : 2
  Persistence : Superblock is persistent

  Update Time : Sat Jun  9 21:20:28 2018
  State : clean
  Active Devices : 2
  Working Devices : 2
  Failed Devices : 0
  Spare Devices : 0

  Chunk Size : 512K

  Name : debian:2 (local to host debian)
  UUID : ac0f1199:ac4f2940:d0056aea:be68cded
  Events : 0

   Number   Major   Minor   RaidDevice State
    0         9        0         0      active sync  /dev/md0
    1         9        1         1      active sync  /dev/md1
```

Ilustración 92: RAID 50 – Estado del RAID

Una vez que hemos creado y los raid y comprobado que todo está correctamente pasamos a darle formato.

```
root@debian:/home/usuario# mkfs.ext4 /dev/md2
mke2fs 1.43.4 (31-Jan-2017)
Se está creando un sistema de ficheros con 12562432 bloques de 4k y 3145728 nodos-i
UUID del sistema de ficheros: 26ee1633-7646-4fa9-b8c8-bd07086f4dc2
Respaldo del superbloque guardado en los bloques:
 32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
4096000, 7962624, 11239424

Reservando las tablas de grupo: hecho
Escribiendo las tablas de nodos-i: hecho
Creando el fichero de transacciones (65536 bloques): hecho
Escribiendo superbloques y la información contable del sistema de ficheros:
hecho
```

Ilustración 93: RAID 50 – Formato ext4

Una vez que hemos visto que funciona correctamente y que le hemos dado formato, podemos montarlo para trabajar con él.

```
root@debian:/home/usuario# mount /dev/md2 /raid50
root@debian:/home/usuario# ls /raid50
aa.txt bb.txt cc.txt dd.txt ee.txt lost+found
```

Ilustración 94: RAID 50 – Montamos la carpeta



Ahora provocaremos un fallo en unos de los dispositivos del primer raid que se creó, como podemos ver esto no afecta al raid0 y sigue funcionando correctamente.

```
root@debian:/home/usuario# mdadm --fail /dev/md0 /dev/sde
mdadm: set /dev/sde faulty in /dev/md0
root@debian:/home/usuario# mdadm -D /dev/md2
/dev/md2:
  Version : 1.2
  Creation Time : Sat Jun  9 21:20:28 2018
  Raid Level : raid0
  Array Size : 50249728 (47.92 GiB 51.46 GB)
  Raid Devices : 2
  Total Devices : 2
  Persistence : Superblock is persistent

  Update Time : Sat Jun  9 21:20:28 2018
  State : clean
  Active Devices : 2
  Working Devices : 2
  Failed Devices : 0
  Spare Devices : 0

  Chunk Size : 512K

         Name : debian:2 (local to host debian)
        UUID : ac0f1199:ac4f2940:d0056aea:be68cded
        Events : 0

   Number  Major   Minor  RaidDevice State
    -----
      0         9       0           0  active sync  /dev/md0
      1         9       1           1  active sync  /dev/md1
```

*Ilustración 95: RAID 50 – Fallo de un disco*

Ahora se provocará un fallo en el segundo raid que se creó, como podemos ver esto tampoco afecta al estado del raid0 y sigue funcionando.

```
root@debian:/raid50# mdadm --fail /dev/md1 /dev/sdi
mdadm: set /dev/sdi faulty in /dev/md1
root@debian:/raid50# mdadm -D /dev/md2
/dev/md2:
  Version : 1.2
  Creation Time : Sat Jun  9 21:20:28 2018
  Raid Level : raid0
  Array Size : 50249728 (47.92 GiB 51.46 GB)
  Raid Devices : 2
  Total Devices : 2
  Persistence : Superblock is persistent

  Update Time : Sat Jun  9 21:20:28 2018
  State : clean
  Active Devices : 2
  Working Devices : 2
  Failed Devices : 0
  Spare Devices : 0

  Chunk Size : 512K

         Name : debian:2 (local to host debian)
        UUID : ac0f1199:ac4f2940:d0056aea:be68cded
        Events : 0

   Number  Major   Minor  RaidDevice State
    -----
      0         9       0           0  active sync  /dev/md0
      1         9       1           1  active sync  /dev/md1
```

*Ilustración 96: RAID 50 – Fallo de dos discos*



También se provocará un fallo en dos dispositivos más una de cada raid.

```

root@debian:/raid50# mdadm --fail /dev/md1 /dev/sdh
mdadm: set /dev/sdh faulty in /dev/md1
root@debian:/raid50# mdadm -D /dev/md2
/dev/md2:
    Version : 1.2
  Creation Time : Sat Jun  9 21:20:28 2018
    Raid Level : raid0
    Array Size : 50249728 (47.92 GiB 51.46 GB)
    Raid Devices : 2
    Total Devices : 2
 Persistence : Superblock is persistent

    Update Time : Sat Jun  9 21:20:28 2018
      State : clean
    Active Devices : 2
    Working Devices : 2
    Failed Devices : 0
    Spare Devices : 0

    Chunk Size : 512K

    Name : debian:2 (local to host debian)
    UUID : ac0f1199:ac4f2940:d0056aea:be68cded
    Events : 0

   Number   Major   Minor   RaidDevice State
    -----
     0         9        0         0      active sync  /dev/md0
     1         9        1         1      active sync  /dev/md1

```

Ilustración 97: RAID 50 – Fallo de 3 discos

```

root@debian:/raid50# mdadm --fail /dev/md0 /dev/sdd
mdadm: set /dev/sdd faulty in /dev/md0
root@debian:/raid50# mdadm -D /dev/md2
/dev/md2:
    Version : 1.2
  Creation Time : Sat Jun  9 21:20:28 2018
    Raid Level : raid0
    Array Size : 50249728 (47.92 GiB 51.46 GB)
    Raid Devices : 2
    Total Devices : 2
 Persistence : Superblock is persistent

    Update Time : Sat Jun  9 21:20:28 2018
      State : clean
    Active Devices : 2
    Working Devices : 2
    Failed Devices : 0
    Spare Devices : 0

    Chunk Size : 512K

    Name : debian:2 (local to host debian)
    UUID : ac0f1199:ac4f2940:d0056aea:be68cded
    Events : 0

   Number   Major   Minor   RaidDevice State
    -----
     0         9        0         0      active sync  /dev/md0
     1         9        1         1      active sync  /dev/md1

```

Ilustración 98: RAID 50 – Fallo de 4 discos

Como podemos ver se ha provocado que dos dispositivos más fallen y el estado del raid 0 sigue siendo el correcto, pero esto es algo imposible ya que los dos raid 5 están fallando, debido a que como explicamos anteriormente, cuando dos dispositivos de un raid 5 fallan el raid deja de funcionar, por lo que aunque es posible implementar este tipo de raid con mdadm debemos hacerlo de forma anidada y





los fallos que puedan surgir son más difíciles de controlar debido a que se tienen que controlar todos los raid inferiores que se creen.

### 3. USO EN DIFERENTES SITUACIONES

#### 3.1. USO EN ORDENADOR PARTICULAR

Para este caso, el más conocido y utilizado es el RAID 0, porque distribuye los datos por las unidades que lo forman obteniendo mayor rendimiento de trabajo. El fallo de este sistema es que no permite la recuperación de datos en caso de que un disco falle, junto con el RAID 1 conocido como “modo espejo” o “mirroring”, que duplica los datos en cada unidad de forma sincronizada, por lo que si se produce un fallo o avería en alguna de las unidades no perderemos ningún dato.

Si tuviésemos más recursos, la segunda mejor opción sería el RAID 5 porque guarda información de paridad para recuperar datos en caso de que uno de los discos que forman el raid se dañase, pero este sistema es más utilizado en empresas que no pueden permitirse perder datos y tienen ordenadores con más recursos.

#### 3.2. USO EN SERVIDOR

Para uso en un servidor del sector empresarial el uso más común que podemos encontrar es el RAID 5, ya que ofrece un rendimiento similar a un RAID 0, pero con una redundancia más económica que los RAID 1.

Como podemos ver en la imagen que se encuentra en el punto [1.1.7](#), si el equipo quiere leer el dato A1 el encargado de proporcionarlo es el disco 0, si también quisiera leer el dato B1 el equipo tendría que esperar, ya que el disco está sirviendo el dato A1, pero si quisiera el dato B2 éste sí podría servirlo a la vez, ya que se encargaría otro disco, el disco 1.

De esta forma podemos realizar operaciones de lectura y escritura de forma simultánea, lo que redundará en un mayor rendimiento y una mayor eficiencia.

Para proteger los datos la clave reside en los bloques de paridad, que son los utilizados para reconstruir los datos si una unidad del grupo RAID falla.

Por lo que cada vez que se escribe un bloque de datos se genera un bloque de paridad dentro de la misma división, que como decimos puede ser recuperado y recalculado junto con los demás bloques de datos en caso de fallo, es decir, cuando se produce un fallo en un disco busca cada pequeña porción de paridad que contienen los otros discos para recuperar el disco que ha fallado. Esto nos ayuda a reducir los costes de redundancia y los costes económicos.



## 4. WEBGRAFÍA

[Man de Linux, correspondiente a mdadm](#)

[Raid setup en wiki.kernel.org](#)

[Gestión de mdadm en elpuig.xeil.net](#)

[Gestión de mdadm en alcancelibre.org](#)

[Eliminar raid en tranquilidadtecnologia.com](#)