

**ext4** es el sistema de ficheros nativo de Linux en las ultimas versiones, utiliza un sistema de **i-nodos** (indice de nodos), y bloques parecido a los clusters en otros tipos de sistemas.

```
mkfs.ext4 -L pruebas -m 1 -b 4096 -O^large_file -i 64K /dev/sdb2
```

**-m 1**

Reserva el 1% para el usuario Root, por defecto el 5%

**-b 4096**

Tamaño del bloque, puede ser 1024,2048,4096, mientras mas grande mejor por temas de rendimiento y velocidad, pero se rompe el estándar, así que 4096, que es lo máximo que permite, si trabajamos con ficheros muy pequeños pues optar por 1024 es una muy buena opción aunque ganaremos en lentitud. Se puede especificar un tamaño diferente pero debe estar marcada la opción `large_file`.

**-O^large\_file**

Archivos de fichero muy grandes, permite trabajar con particiones muy grandes, por ejemplo de Petabytes= 1000Terabytes

**-i bytes-per-inode**

Similar al tamaño del cluster pero referido al i-nodo, lo normal 64K. Por temas de rendimiento mientras mas grande mejor, puedes poner 4K,8K,16K,32K,64K,128K... siempre en potencia de dos. Mientras mas inodos (tamaño mas pequeño) mas lentitud y espacio mejor aprovechado. A menos inodos se pierde espacio pero se gana velocidad, en la actualidad con discos duros de 1000Gb que pierdas 1Gb de datos de todo el almacenamiento para tener mejor rendimiento, se puede permitir.

Existen muchas mas opciones pero con esto de sobra puedes ir trasteando la que mejor te convenga.

**/etc/fstab**

Es el fichero que permite las opciones de montaje. Comentaré las mas usadas.

`discard`

Si tenemos un Disco SSD esta opción la tenemos que habilitar en `fstab`. También es válido para eMMC o tarjetas de memoria.

`noatime`

Permite no escribir la fecha de acceso cada vez que accede a un fichero o directorio, ahorra escrituras en el disco con lo que se alarga la vida de los SSD. Sólo cambia la fecha de acceso cuando el fichero se modifica.

`relatime`

No escribe la fecha si se ha accedido en menos de 24 horas o bien si se modifica el fichero por escritura pues cambia la fecha, lo cual reduce las escrituras considerablemente sobre todo en servidores.

`commit=60`

Escritura diferida, no se escribe directamente los cambios en el disco, en el ejemplo 60 segundos, por ejemplo en sistemas que no son críticos y no nos importa perder datos que hacemos en 60 segundos.

`data=ordered`

`data=` especifica el tipo de Journal que se utiliza, *ordered* es por defecto, primero se escribe en disco y luego la transacción de escritura en el sistema de ficheros. *writeback* permite la escritura después de la transacción, solo se guardan los metadatos, no es lo idóneo porque se pierden datos en caso de cortes de luz, pero es muy rápido.

Algunos ejemplos de `fstab`

```
#monta /tmp en ram (tmpfs) con 1500M, pero se puede poner por ejemplo
#porcentaje, 3%
tmpfs /tmp tmpfs nodev,nosuid,noexec,relatime,mode=1777,size=1500M 0 0
```

```
# Ejemplo para una memoria SSD para el sistema raíz
# El ultimo 1 es para que se cheque antes de todo, 0 no chequea.
/dev/sdb1 /root ext4 defaults,discard,noatime,data=ordered,commit=30 0 1
```

```
# Ejemplo para un disco duro que no nos importa perder datos
```

```
# El ultimo 2 es para que se cheque después del root o raiz.  
/dev/sdc2 /home ext4 defaults,noatime,data=writeback,commit=60 0 2
```