Primero instalamos los paquetes necesarios:

```
apt-get -y install mdadm rsync initramfs-tools
```

Luego, suponiendo que el primer disco para el RAID es /dev/nvme2n1 y el segundo /dev/nvme3n1, borramos sus particiones y datos, ejecutando como root:

```
wipefs -a /dev/nvme2n1 /dev/nvme3n1
```

ejecutamos como root:

```
mdadm -C /dev/md0 -l raid0 -n 2 /dev/nvme2n1 /dev/nvme3n1
```

Nos advertirá sobre que los discos seleccionados ya tienen una tabla de particiones y que dicha tabla se borrará al crear el RAID. Le decimos que si.

Formateamos la partición, ejecutando como root:

```
mkfs.ext4 /dev/md0
```

Hacemos que no se modifique el nombre del RAID al reiniciar

```
mdadm --detail --scan >> /etc/mdadm/mdadm.conf
update-initramfs -u
```

Lo siguiente es montar el raid. Para ello ejecutamos como root:

```
mkdir /mnt/raid0/
mount -t auto /dev/md0 /mnt/raid0
```

Hacemos el montaje permanente en fstab:

```
echo "/dev/md0 /mnt/raid0 ext4 defaults 0 2" >> /etc/fstab
```

Y ahora le damos permisos a esa carpeta para que pueda usarla el usuario por defecto de la instalación de Debian:

```
chown 1000:1000 /mnt/raid0 -R
```

Y ya tendríamos el RAID 0 funcional en la sesión donde se instaló y disponible después de cada inicio del sistema.

Miramos los detalles de nuestro RAID ejecutando como root:

```
mdadm --detail /dev/md0
```

CASUÍSTICA PARA DISCOS CON DE DIFERENTE TAMAÑO DE SECTOR

Es posible que, si ambos discos tenían tamaños de asignación de sectores diferentes (por ejemplo uno con 4096 y el otro con 512) el RAID no sobreviva al reinicio y se pierdan todos los datos, por lo que no es conveniente meter datos en el RAID, ni bien se crea. Siempre es mejor reiniciar primero, antes de empezar a meter datos.

Es posible crear un RAID con discos de diferente tamaño de asignación de sector. El truco está en crear el RAID usando una partición de cada uno de los discos, en vez de usar el disco entero, siempre que las particiones tengan exactamente el mismo tamaño.

Para ver los tamaños de asignación, hacemos:

```
lsblk -o NAME,MODEL,SIZE,LOG-SEC,PHY-SEC /dev/nvme2n1 /dev/nvme3n1
```

Es posible que ambos discos tengan exactamente el mismo tamaño en espacio, pero cantidad de asignación diferente por sector. Por ejemplo, esta podría ser una salida a ese comando:

```
NAME    MODEL                 SIZE LOG-SEC PHY-SEC
nvme3n1 Seagate FireCuda 530  3,6T 4096    4096
nvme2n1 CT4000P3SSD8          3,6T 512     512
```

Entonces, seguiremos este procedimiento:

1. Creamos un mapa de particiones GPT en cada uno:

```
parted -s /dev/nvme2n1 mklabel gpt
parted -s /dev/nvme3n1 mklabel gpt
```

2. Borramos las posibles particiones que puedan haber en ambos discos:

```
wipefs -a /dev/nvme2n1 /dev/nvme3n1
```

El siguiente paso es crear una partición RAID con exactamente los mismos límites en bytes en ambos. Como los sectores lógicos son distintos, no usaremos números de sector sino MiB:

```
parted -s /dev/nvme2n1 mkpart primary 1MiB 3815446MiB
parted -s /dev/nvme3n1 mkpart primary 1MiB 3815446MiB
```

Marcamos ambas particiones como tipo Linux RAID:

```
parted -s /dev/nvme2n1 set 1 raid on
parted -s /dev/nvme3n1 set 1 raid on
```

Ahora creamos el raid sobre las particiones, no sobre el disco completo, ejecutando:

```
mdadm --create /dev/md0 --level=0 --raid-devices=2 /dev/nvme2n1p1 /dev/nvme3n1p1
```

Ahora ya podemos agregar el array a /etc/mdadm/mdadm.conf: