

# Usar una Raspberry Pi 4 para Time Machine

En primer lugar actualizaremos

```
sudo apt-get update && sudo apt-get upgrade
```

## Instalar Samba

Primero instalaremos Samba (para SMB), que es un protocolo de intercambio de archivos de código abierto muy popular que cuenta con el respaldo oficial de Time Machine para realizar copias de seguridad de datos a través de una red. El daemon de Avahi se utiliza para el protocolo mDNS “Bonjour” de Apple para que nuestro servidor Raspberry Pi se detecte automáticamente dentro de la red.

```
sudo apt-get install samba avahi-daemon
```

## Conecta el disco duro al puerto USB 3.0

Como la Raspberry Pi 4 viene con 2 puertos USB 3.0, podemos conectar un disco duro externo como nuestro espacio en disco para hacer una copia de seguridad de los datos. tendremos velocidades decentes, si también tenemos una conexión por cable de 1 GBit/s con el Pi 4.

## Creamos una carpeta para montar el disco

## Comprobamos que el disco está conectado

```
root@raspberrypi: /mnt# lsblk
```

NAME	MAJ: MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8: 0	0	1,8T	0	disk	
mmcblk0	179: 0	0	29,7G	0	disk	
└─mmcblk0p1	179: 1	0	256M	0	part	/boot
└─mmcblk0p2	179: 2	0	29,5G	0	part	/

Vemos que tenemos un disco en /dev/sda

## Comprobamos el UUID

```
root@raspberrypi: /mnt# blkid
/dev/mmcblk0p1: LABEL_FATBOOT="boot" LABEL="boot" UUID="5EA1-EC88" BLOCK_SIZE="512"
TYPE="vfat" PARTUUID="c90a195c-01"
/dev/mmcblk0p2: LABEL="rootfs" UUID="f0702b0a-e54b-4c34-bd2d-6e52a561898c" BLOCK_SIZE="4096"
TYPE="ext4" PARTUUID="c90a195c-02"
/dev/sda1: UUID="5e5aa2cf-a809-4139-8096-e3647fa4e116" BLOCK_SIZE="4096" TYPE="ext4"
PARTUUID="cf7da363-01"
```

Vemos que el UUID es "5e5aa2cf-a809-4139-8096-e3647fa4e116"

En nuestro caso, ya está particionado, si no fuera así

```
sudo mkfs -t ext4 /dev/sda
```

## Creamos un usuario para el backup

```
sudo adduser timemachine
```

Te pedirá que establezcas una contraseña para el usuario recién creado.

## Ahora crearemos la contraseña de Samba

```
sudo smbpasswd -a timemachine
```

## Crear punto de montaje

Para montar el disco duro USB, necesitamos crear una carpeta donde Linux pueda montar nuestro dispositivo.

```
sudo mkdir /mnt/timemachine
```

## Cambiar permisos

Para permitir que nuestro usuario de respaldo escriba en el dispositivo, necesitaremos asumir la propiedad del punto de montaje:

```
sudo chown -R timemachine: /mnt/timemachine
```

# Crear montaje

Editaremos el fichero fstab

```
nano /etc/fstab
```

Estará aproximadamente así

```
proc                /proc                proc    defaults                0        0
PARTUUID=c90a195c-01 /boot                vfat    defaults,flush          0        2
PARTUUID=c90a195c-02 /                  ext4    defaults,noatime        0        1
# a swapfile is not a swap partition, no line here
#   use dphys-swapfile swap[on|off] for that
```

Añadimos una linea con el UUID del disco, y el punto de montaje al final del fichero

```
UUID=5e5aa2cf-a809-4139-8096-e3647fa4e116 /mnt/timemachine    ext4
sync,noexec,nodev,noatime,nodiratime    0        0
```

El fichero quedará así

```
proc                /proc                proc    defaults                0        0
PARTUUID=c90a195c-01 /boot                vfat    defaults,flush          0        2
PARTUUID=c90a195c-02 /                  ext4    defaults,noatime        0        1
# a swapfile is not a swap partition, no line here
#   use dphys-swapfile swap[on|off] for that
UUID=5e5aa2cf-a809-4139-8096-e3647fa4e116 /mnt/timemachine    ext4
sync,noexec,nodev,noatime,nodiratime    0        0
```

Los modificadores significan:

**sync:** todas las escrituras en el disco deben hacerse sincrónicamente.

**noexec:** no permite la ejecución directa de ningún binario en el sistema de archivos montado, ya que solo se utilizará para el almacenamiento de copia de seguridad.

**nodev:** no interpreta caracteres ni bloquea dispositivos especiales en el sistema de archivos.

**noatime:** no almacena tiempos de acceso en este sistema de archivos.

**nodirtime:** no almacena tiempos de acceso al directorio en este sistema de archivos.

**0:** ignora este sistema de archivos cuando lo use dump

**0:** no verifica el fsck al reiniciar.

Salvamos y comprobamos que funciona con un mount -a

```
mount -a
```

Si nada falla, tendremos el disco montado en la carpeta /etc/timemachine

```
root@raspberrypi: /mnt# ls -la /mnt/timemachine/
total 40
drwxr-xr-x  5 timemachine timemachine 4096 ago 10 22:52 .
drwxr-xr-x  3 root         root         4096 ago 12 18:51 ..
drwx----- 2 timemachine timemachine 16384 jul 28 19:24 lost+found
```

## Configurar Samba

Edita el archivo de configuración de Samba y agrega lo siguiente al final:

```
[backups]
    comment = Backups
    path = /mnt/timemachine
    valid users = timemachine
    read only = no
    vfs objects = catia fruit streams_xattr
    fruit:time machine = yes
```

Comprobamos la configuración de samba

```
sudo testparm -s
Load smb config files from /etc/samba/smb.conf
Loaded services file OK.
Weak crypto is allowed
WARNING: some services use vfs_fruit, others don't. Mounting them in conjunction on OS X
clients results in undefined behaviour.

Server role: ROLE_STANDALONE

# Global parameters
[global]
    log file = /var/log/samba/log.%m
    logging = file
    map to guest = Bad User
    max log size = 1000
    obey pam restrictions = Yes
    pam password change = Yes
    panic action = /usr/share/samba/panic-action %d
```

```

[passwd chat = *Enter\snew\s*\spassword: * %n\n *Retype\snew\s*\spassword: * %n\n
*password\supdated\ssuccessfully* .
[passwd program = /usr/bin/passwd %u
[server role = standalone server
[unix password sync = Yes
[usershare allow guests = Yes
[idmap config * : backend = tdb


[homes]
[browseable = No
[comment = Home Directories
[create mask = 0700
[directory mask = 0700
[valid users = %S


[printers]
[browseable = No
[comment = All Printers
[create mask = 0700
[path = /var/spool/samba
[printable = Yes


[print$]
[comment = Printer Drivers
[path = /var/lib/samba/printers


[backups]
[comment = Backups
[path = /mnt/timemachine
[read only = No
[valid users = timemachine
[vfs objects = catia fruit streams_xattr
[fruit:time machine = yes

```

## Reiniciamos Samba

```
sudo service smb reload
```

Y lo habilitamos en el inicio

```
systemctl enable smb
```

## Configurar el demonio de Avahi

Para permitir que MacOS detecte automáticamente nuestra nueva Time Machine, necesitaremos configurar Avahi. Para eso edita el siguiente archivo:

```
sudo nano /etc/avahi/services/samba.service
```

Y pega este contenido

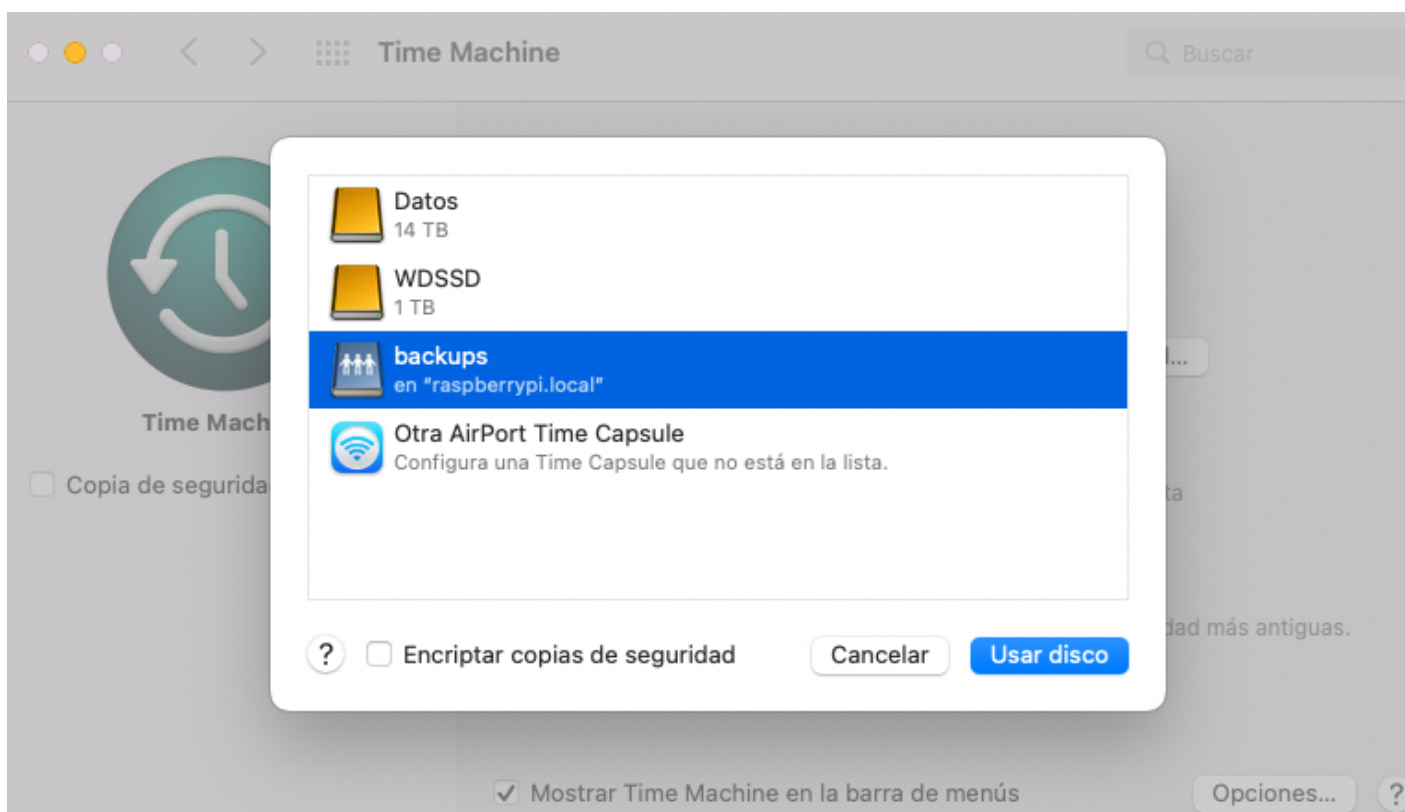
```
<?xml version="1.0" standalone='no'?><!--*-nxml-*-->
<!DOCTYPE service-group SYSTEM "avahi-service.dtd">
<service-group>
  <name replace-wildcards="yes">%h</name>
  <service>
    <type>_smb._tcp</type>
    <port>445</port>
  </service>
  <service>
    <type>_device-info._tcp</type>
    <port>9</port>
    <txt-record>model=TimeCapsule8,119</txt-record>
  </service>
  <service>
    <type>_adisk._tcp</type>
    <port>9</port>
    <txt-record>dk0=adVN=backups,adVF=0x82</txt-record>
    <txt-record>sys=adVF=0x100</txt-record>
  </service>
</service-group>
```

Reiniciamos el demonio de Avahi

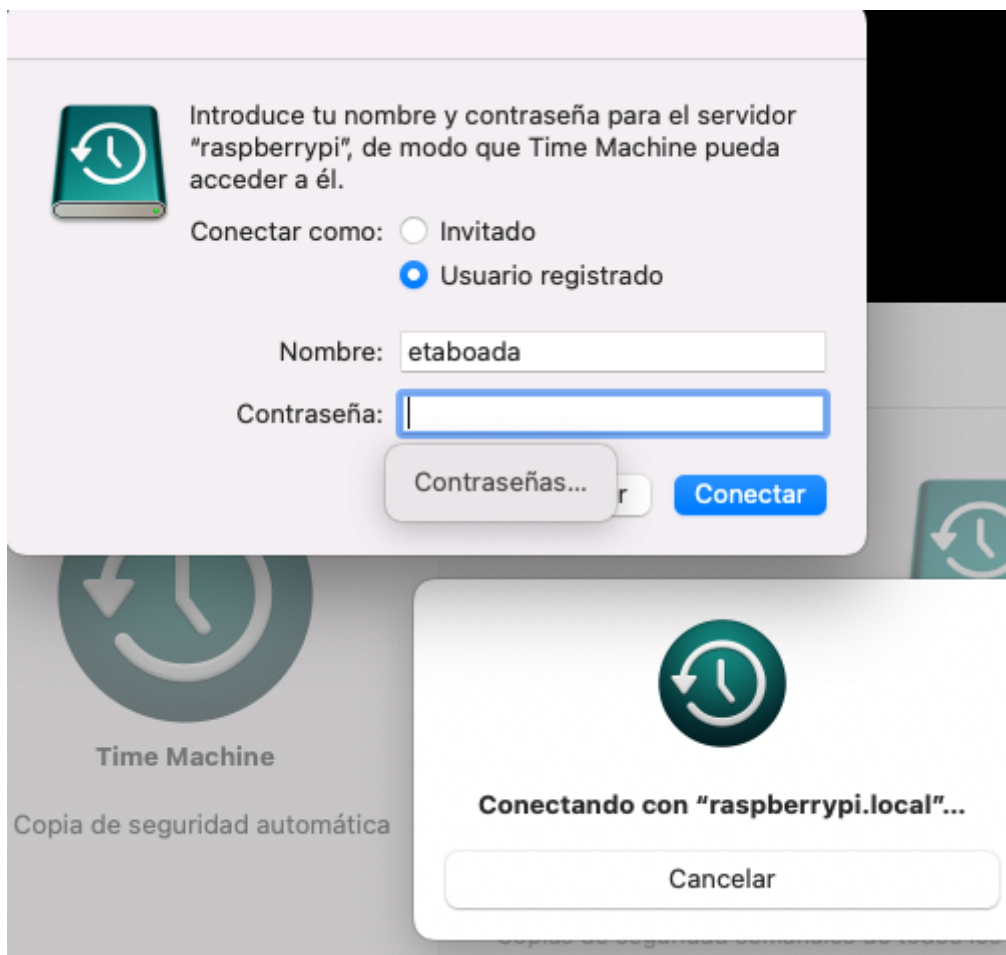
```
sudo service avahi-daemon restart
sudo systemctl enable avahi-daemon
```

Ahora nos conectamos

Vamos a las preferencias de timemachine en el mac

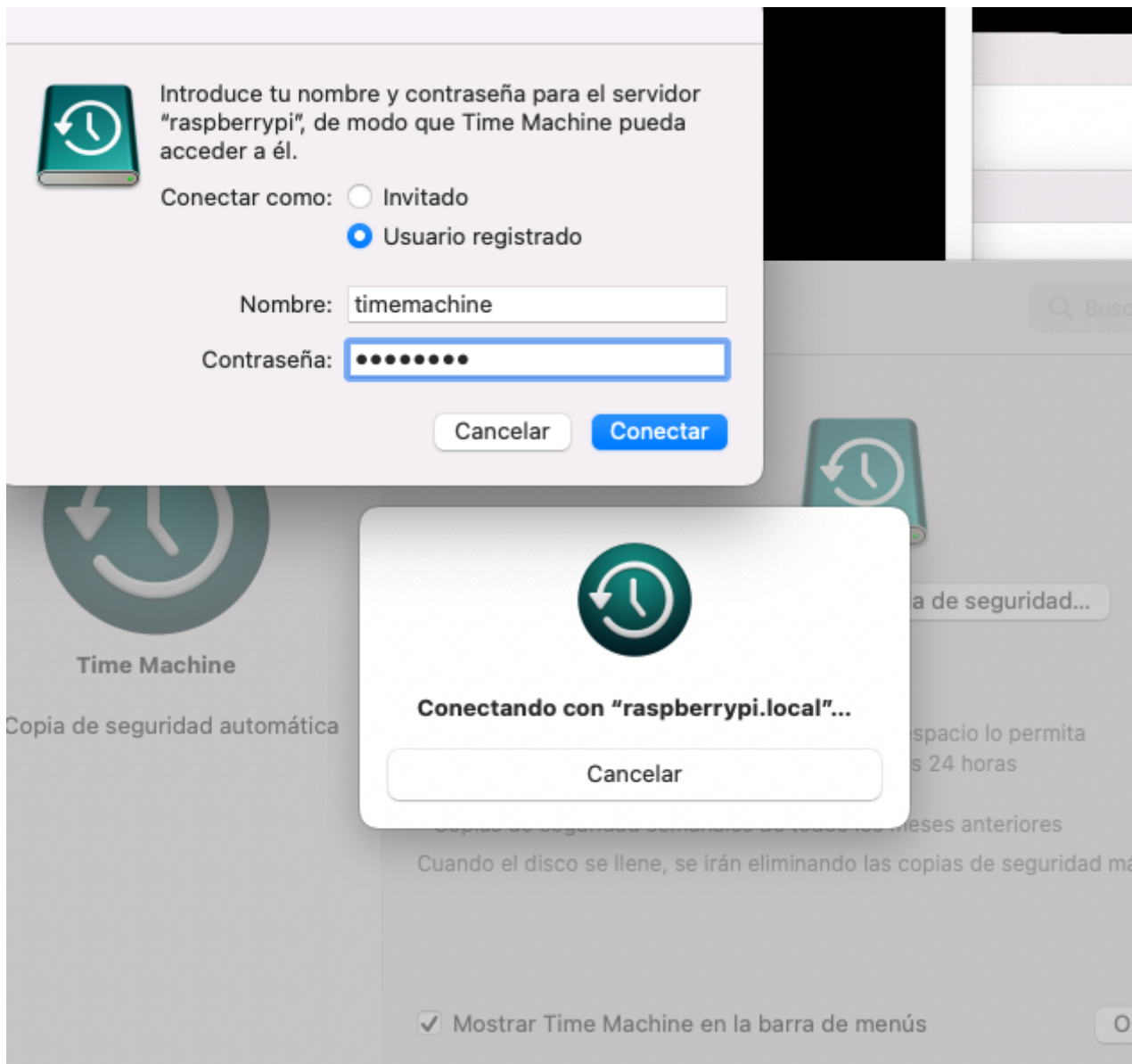


Conectamos con el disco



Debemos de poner nuestro usuario y la contraseña que hemos elegido





Y ya está.

---

Revision #6

Created 12 August 2022 17:01:41 by Admin

Updated 12 August 2022 17:30:06 by Admin