



Crear un pendrive booteable con el instalador de macOS a partir del archivo InstallAssistant.pkg

Si ya disponemos de un archivo **InstallAssistant.pkg** correspondiente a alguna versión de macOS, hay dos caminos para crear un pendrive booteable:

USANDO MACOS

Con seguir [este hack](#), ya lo tenemos. Esto es así porque una vez que dispongamos de ese .pkg, podríamos meterlo en el Mac, ejecutarlo y, al finalizar la instalación, quemarlo en un pendrive siguiendo los pasos descritos en el hack. Esto es posible porque la descarga y ejecución de ese .pkg es justamente lo que hace en background la AppStore de un Mac original cuando le pedimos que descargue una versión de macOS desde ella. Esto nos evitaría todo el proceso de tener que preparar el pendrive instalador desde Debian.

USANDO DEBIAN

Si no disponemos de un Mac en el que quemar el instalador, tendremos que hacerlo desde Debian. Entonces, en Debian, para extraer el contenido del archivo InstallAssistant.pkg, hacemos:

```
sudo apt-get -y update
sudo apt-get -y install libarchive-tools
mkdir -p "$HOME/macOSInstaller/Descomprimido"
mv -v "$HOME/Descargas/InstallAssistant.pkg" "$HOME/macOSInstaller/"
cd "$HOME/macOSInstaller/"
bsdtar -xvf InstallAssistant.pkg -C Descomprimido
```

Esos comandos extraerán todo el contenido del pkg en la carpeta **~/macOSInstaller/Descomprimido**. De esos archivos extraídos, nos interesa **SharedSupport.dmg**, pero para trabajar con él, deberemos convertirlo previamente a .img. Lo hacemos de esta forma:

```
sudo apt-get -y update
sudo apt-get -y install dmg2img
cd "$HOME/macOSInstaller/Descomprimido/"
dmg2img SharedSupport.dmg SharedSupport.img
```

Una vez convertido vamos a mapear sus particiones con kpartx:

```
sudo kpartx -av SharedSupport.img
```

Nos mostrará algo como esto:

```
add map loop1p1 (252:0): 0 409600 linear 7:1 40
add map loop1p2 (252:1): 0 32905880 linear 7:1 409640
```

La primera es la EFI, la segunda la HFS+. Entonces, para montar la HFS+:

```
sudo modprobe hfsplus
sudo mkdir -p /mnt/PartHFS
sudo mount -t hfsplus -o ro /dev/mapper/loop1p2 /mnt/PartHFS
```

Una vez montada, dentro de la carpeta **/mnt/PartHFS/com_apple_MobileAsset_MacSoftwareUpdate/** encontraremos un archivo .zip de gran tamaño.

s

Finalmente, para extraer el archivo InstallESD.dmg de dentro del .img, primero montamos el .img:



Crear un pendrive booteable con el instalador de macOS a partir del archivo InstallAssistant.pkg

```
sudo mkdir /mnt/sequoia  
sudo mount -o loop SharedSupport.img /mnt/sequoia
```

¿el contenido de SharedSupport.dmg de esta forma:

```
p7zip-full cpio xz-utils gzip dmg2img
```

Extrae el .pkg:

```
mkdir ~/SequoiaInstaller  
cd ~/SequoiaInstaller
```

```
dmg2img «Install macOS Sequoia.app/Contents/SharedSupport/InstallESD.dmg» Sequoia.img
```

```
sudo dd if=Sequoia.img of=/dev/sdX bs=4M status=progress oflag=sync
```

4. Añadir OpenCore al USB

El instalador no tiene bootloader. Ahora:

Monta la partición EFI del USB:

```
sudo mkdir /mnt/efiusb  
sudo mount /dev/sdX1 /mnt/efiusb
```

Copia tu carpeta EFI (la misma que ya te funciona):

```
sudo cp -R /ruta/a/tu/EFI /mnt/efiusb/
```

Desmonta:

```
sudo umount /mnt/efiusb
```



Crear un pendrive booteable con el instalador de macOS a partir del archivo InstallAssistant.pkg

Para meter el instalador en un pendrive USB, primero lo descargaremos con la herramienta gibMacOS. Entonces, nos descargamos el repo, con:

```
python3 gibMacOS.py -r
```