

Cuando queremos preparar un hackintosh tenemos que disponer de la imagen de instalación de OS X o MacOS para ser quemada en un pendrive. Esto puede suponer un problema si no tenemos un ordenador Mac previo a la creación de dicho pendrive. Por suerte existe un script que nos soluciona la papeleta desde Debian y nos permite descargar todas las imágenes de instalación de OS X y MacOS que existen hasta la fecha. Este script se llama **gibMacOS**.

INSTALACIÓN

Nos aseguramos primero de tener los paquetes necesarios para clonar el repo y para lanzar el script:

```
sudo apt-get -y update
sudo apt-get -y install git
sudo apt-get -y install python3
```

Una vez instalados los paquetes, clonamos el repo con:

```
git clone https://github.com/corpnewt/gibMacOS
```

USO

Hay diferentes formas de usar el script. La más normal es simplemente lanzarlo con:

```
cd gibMacOS
python3 gibMacOS.py
```

...y seguir las instrucciones. Pero también acepta parámetros con los que se pueden hacer diferentes cosas. Por ejemplo:

Descargar los archivos de instalación de una versión específica para la que conocemos el número de versión y el número de compilación:

```
python3 gibMacOS.py -v 15.6 -b 24G84
```

Último instalador disponible:

```
python3 gibMacOS.py -l
python3 gibMacOS.py -l -i          # solo mostrar URLs
python3 gibMacOS.py -l -j          # solo JSON de metadatos
python3 gibMacOS.py -l -d          # solo .dmg
python3 gibMacOS.py -l -D VMM-x86_64 # filtrar por Device ID (ej: VMs x86_64)
python3 gibMacOS.py -l -D MacBookPro17,1 # filtrar por un modelo concreto
python3 gibMacOS.py -l -o /datos/macOS # cambiar carpeta de descarga
python3 gibMacOS.py -l -c developer  # usar catálogo developer
python3 gibMacOS.py -l -c public     # beta pública
python3 gibMacOS.py -l -c customer  # customer seed
python3 gibMacOS.py -l -m 15         # limitar URL a máximo macOS 15
```

Por versión (número o nombre):

```
python3 gibMacOS.py -v 10.15      # Catalina
python3 gibMacOS.py -v 11         # Big Sur
python3 gibMacOS.py -v 12         # Monterey
python3 gibMacOS.py -v 13         # Ventura
python3 gibMacOS.py -v 14         # Sonoma
python3 gibMacOS.py -v 15         # Sequoia
python3 gibMacOS.py -v Sequoia
python3 gibMacOS.py -v Sonoma -d  # solo .dmg de Sonoma
python3 gibMacOS.py -v Monterey -i # solo URLs de Monterey
python3 gibMacOS.py -v Ventura -j # JSON de Ventura
```

```
python3 gibMacOS.py -v Sequoia -D VMM-x86_64
python3 gibMacOS.py -v Sequoia -o "/mnt/raid/Apple Installers"
```

Versión + build exacta (Recuerda: -build requiere -version):

```
python3 gibMacOS.py -v 10.15 -b 19H2026
python3 gibMacOS.py -v Sonoma -b 23G93
python3 gibMacOS.py -v Sequoia -b 24G90 -d
```

Por Product ID (PID de catálogo):

```
python3 gibMacOS.py -p 012-08543
python3 gibMacOS.py -p 012-08543 -d
python3 gibMacOS.py -p 012-08543 -i
```

Sólo paquetes de Recovery:

```
python3 gibMacOS.py -r          # busca RecoveryHDDUpdate.pkg/MetaDmg
python3 gibMacOS.py -r -c developer
python3 gibMacOS.py -r -v Monterey
python3 gibMacOS.py -r -l      # último con paquetes de recovery
```

Catálogo local (offline/forzado):

```
python3 gibMacOS.py -s          # usar sucatalog.plist local si existe
python3 gibMacOS.py -g /ruta/a/sucatalog.plist -n      # descargar y sobrescribir catálogo local
python3 gibMacOS.py -g /ruta/a/sucatalog.plist -v Sequoia
python3 gibMacOS.py -s -l -i    # usar local y listar URLs del último
```

Cambiar catálogo de Apple (fuente de productos):

```
python3 gibMacOS.py -c publicrelease -l
python3 gibMacOS.py -c public -l
python3 gibMacOS.py -c customer -l
python3 gibMacOS.py -c developer -l
python3 gibMacOS.py -c developer -v Sequoia -d
```

Limitar construcción de URL por macOS:

```
python3 gibMacOS.py -m 10.14 -i          # catálogo construido hasta 10.14
python3 gibMacOS.py -m 12 -c public -l   # hasta Monterey, beta pública
python3 gibMacOS.py -m 15 -c developer -v Sequoia
```

Combinaciones útiles:

```
python3 gibMacOS.py -l -D VMM-x86_64 -i          # último compatible con VMs x86_64, solo URLs
python3 gibMacOS.py -v Sequoia -D MacBookPro17,1 -d      # Sequoia solo DMGs y filtrado por modelo
python3 gibMacOS.py -v Ventura -b 22G120 -o /srv/macOS
python3 gibMacOS.py -c developer -m 15 -l -d
python3 gibMacOS.py -s -c publicrelease -v 10.15 -i      # usar catálogo local y listar URLs de Catalina
```

Modo no interactivo explícito (Se activa solo al usar -p o -v, pero puedes forzarlo):

```
python3 gibMacOS.py --no-interactive -l -i
```

```
python3 gibMacOS.py --no-interactive -v Sequoia -d -o /datos/macOS
```

Salida solo URLs / solo JSON (sin descargar):

```
python3 gibMacOS.py -l -i
python3 gibMacOS.py -v Sonoma -i
python3 gibMacOS.py -p 012-08543 -i
python3 gibMacOS.py -l -j
python3 gibMacOS.py -v Sequoia -j
```

Directorio de descarga personalizado:

```
python3 gibMacOS.py -l -o /var/tmp/macOS
python3 gibMacOS.py -v 11 -o "/mnt/almacen/Apple Installers"
```

Casos “menos comunes”, pero válidos:

```
python3 gibMacOS.py -v Big\ Sur -d # nombre con espacio
python3 gibMacOS.py -v Tahoe # el script mapea "Tahoe" internamente
python3 gibMacOS.py -c public -m 12 -v Monterey # catálogo beta pública, URL hasta 12
```

