

Artix Linux
Guía de Usuario

1. Introducción

Este documento pretende ser una guía para que termines de configurar tu instalación de Artix Linux. Hay cosas que no son razonables/posibles auto-configurar y esta guía contiene instrucciones sobre como configurar servicios como cronie, libvirt, etc.

También contiene instrucciones sobre como usar el entorno de trabajo del instalador.

Nota

Cualquier texto que veas coloreado en azul es un hiper-vínculo y si le haces doble clic, este te llevará a la página web a la que hace referencia o que explica de que se esta hablando.

2. Atajos de teclado

[Dwm](#) es el programa que se encarga de repartir el espacio en pantalla entre nuestras ventanas (*aplicaciones gráficas*). Hay una cantidad enorme de [videos sobre dwm](#), como funciona y sus ventajas. Aquí veremos simplemente los atajos de teclado configurados:

Atajos de teclado	
Abrir este documento	Alt Izq. + Ctrl + H
Abrir lanzador de comandos	Alt Izq. + P
Abrir lanzador de aplicaciones	Alt Izq. + Shift + P
Abrir dmenu (calculadora)	Ctrl + Alt Izq. + P
Abrir dmenu (contraseñas)	Ctrl + P
Abrir terminal	Alt Izq. + Shift + Intro
Aumentar el espacio entre ventanas	Alt Izq. + Ctrl + ,
Disminuir el espacio entre ventanas	Alt Izq. + Ctrl + .
Configurar pantallas	Alt Izq. + F1 / Alt + Shift + F1
Abrir Navegador (<i>Firefox</i>)	Alt Izq. + F2
Abrir Explorador de archivos	Alt Izq. + F3
Explorador de Archivos (/run/media)	Alt Izq. + Shift + F3
Abrir el reproductor de música	Alt Izq. + F4
Montar dispositivo Android	Alt Izq. + F5
Desmontar dispositivo Android	Alt Izq. + Shift + F5
Abrir menú de apagado	Alt Izq. + F11
Reiniciar dwm	Alt Izq. + Shift + F11
Abrir mezclador de sonido	Alt Izq. + F12
Canción anterior	Alt Izq. + Z
Canción siguiente	Alt Izq. + X
Pausar/reanudar la reproducción	Alt Izq. + Shift + Z/X
Bajar Volumen	Alt Izq. + N
Subir Volumen	Alt Izq. + M
Establecer volumen al 50 %	Alt Izq. + Shift + N
Establecer volumen al 100 %	Alt Izq. + Shift + M
Silenciar/activar el sonido	Alt Izq. + Ctrl + N/M
Captura de pantalla al portapapeles	Alt Izq. + O
Guardar captura de pantalla	Alt Izq. + Ctrl + O
Captura al portapapeles (Área selec.)	Alt Izq. + Shift + O
Guardar captura (Área seleccionada)	Alt Izq. + Ctrl + Shift + O
Ocultar/Mostrar barra de tareas	Alt Izq. + B

Atajos de teclado	
Mover el foco a la posición anterior	Alt Izq. + ,
Mover el foco a la posición siguiente	Alt Izq. + .
Mover la ventana a la pos. anterior	Alt Izq. + Shift + ,
Mover la ventana a la pos. siguiente	Alt Izq. + Shift + .
Moverse al espacio anterior	Alt Izq. + Q
Moverse al espacio siguiente	Alt Izq. + W
Moverse al espacio 1-9	Alt Izq. + 1-9
Mover ventana al espacio 1-9	Alt Izq. + Shift + 1-9
Cerrar ventana	Alt Izq. + Shift + Q
Destruir ventana	Alt Izq. + Shift + C
Mover el foco al monitor anterior	Alt Izq. + {
Mover el foco al monitor siguiente	Alt Izq. + }
Mover ventana al monitor anterior	Alt Izq. + Shift + {
Mover ventana al monitor siguiente	Alt Izq. + Shift + }
Abrir <i>Terminal Scratchpad</i>	Alt Izq. + F
Mostrar/Ocultar <i>scratchpads</i>	Alt Izq. + S
Alternar ventana como <i>scratchpad</i>	Alt Izq. + Shift + S
Alternar ventana como <i>sticky</i>	Alt Izq. + Ctrl + S
Mover ventana	Alt Izq. + Ctrl + Clic Izq.
Re-dimensionar ventana	Alt Izq. + Ctrl + Clic Der.

3. Compartir audio en Discord

Si has instalado discord te darás cuenta que en Linux, no puedes compartir audio cuando compartes tu escritorio o una ventana. Para solucionar este problema, cuando inicias sesión se crea un *micrófono virtual* en el que puedes juntar el audio de tu micrófono real y el audio de las aplicaciones que desees.

Para que la gente pueda escuchar el audio de las aplicaciones tendrás que elegir en discord el micrófono a usar como *my-virtualmic*, y, recomendable; desactivar la reducción de ruido. Pues esta detectará el audio de las aplicaciones como ruido y hará que estas no se escuchen bien.

Para añadir tu micrófono, haz clic en una parte vacía del escritorio y en la pestaña *Compar. audio* haz clic en *Ajustes Mic..* Aparecerá un menú para añadir o quitar dispositivos de audio del micrófono virtual.

También verás que en la pestaña *Compar. audio* puedes hacer clic en las diferentes opciones que hay para compartir o dejar de compartir el audio de las aplicaciones.

4. Cronie y scripts útiles

El repositorio incluye una variedad de scripts para automatizar ciertas tareas. Por defecto no se usan, pero podemos hacer que se ejecuten en fechas concretas de forma automática.

Con tu instalación base deberías de tener activado el servicio [cron](#). Este servicio se encarga de ejecutar comandos de forma automática. El archivo de configuración se encuentra en `/etc/crontab` y tal como se ha configurado, debería verse como:

```
SHELL=/bin/bash
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

@hourly root reflector --verbose --latest 10 --sort rate --save /etc/pacman.d/mirrorlist-arch
@reboot usuario ionice -c3 syncthing --no-browser --no-default-folder
```

- “@hourly” es la parte de donde definimos cuando se ejecutará el comando, si quieres investigar hay un montón de guías en youtube o cualquier otra plataforma sobre cuál es la sintaxis para configurar la ejecución de los comandos. Puedes también comprobar si tu sintaxis es correcta en crontab.guru.
- “root” es el usuario que ejecuta el comando, y *reflector -verbose -latest 10 -sort rate -save /etc/pacman.d/mirrorlist-arch*, el comando.

4.1. convert-2m4a y convert-2mp3

Este script convierte toda la música del directorio que damos como primer argumento a formato *.m4a* o *.mp3* en un mirror que replica al estructura de los archivos en el directorio original.

Si ejecutamos:

```
convert-2mp3 /musica/biblioteca /musica/mp3
```

Esto convertirá toda la música de */musica/biblioteca* a la carpeta */musica/mp3*, en formato mp3. Lo que puede resultarnos útil porque los archivos mp3 ocupan menos espacio que otros formatos sin pérdida de calidad.

De esta manera podemos tener una copia de nuestra biblioteca de música más liviana. Para, por ejemplo, guardarla en un dispositivo con menos almacenamiento como un dispositivo móvil.

Podemos automatizar este proceso, para que el usuario “usuario” convierta a mp3 su música todos los días, añadiendo esta línea a */etc/crontab*:

```
@daily usuario convert-2mp3 /musica/biblioteca /musica/mp3
```

- El script puede usarse con el flag *-p* para convertir varios archivos de forma paralela, lo que reduce el tiempo necesario para la conversión de archivos, pero consume más recursos y probablemente ocupe todo el tiempo de CPU hasta que el script se termine de ejecutar.
- El script también puede usarse con el flag *-l* lo que hará que además de convertir las canciones a otro formato, se busque la letra de la canción en Internet y se incluya dentro del archivo de audio.

4.2. corruption-check

Este script comprueba que no haya archivos corruptos en nuestra biblioteca de música, corrige falsos positivos de corrupción y nos escribe una lista con los archivos que realmente están corruptos y no se pueden reproducir correctamente en */tmp/corruption.log*. Solo necesita como argumento el directorio cuyos archivos de audio queremos analizar.

Podemos automatizar esta tarea añadiendo el comando a */etc/crontab*, aquí un ejemplo:

```
@daily usuario corruption-check /musica/biblioteca
```

4.3. exif-remove

Este script necesita borrar toda la información EXIF de las imágenes que contiene el directorio que se le da como argumento.

Los metadatos EXIF sirven para identificar el usuario que tomó una fotografía, captura de pantalla, etc. Por ejemplo, si tomas una fotografía con tu teléfono, el teléfono guarda como información EXIF datos como; el teléfono desde el cual la fotografía fue tomada, o desde que coordenadas geográficas fue tomada la foto.

Para borrar automáticamente los metadatos de una carpeta, puedes añadir a `/etc/crontab` una línea parecida a:

```
@hourly usuario exif-remove /fotografias
```

4.4. wake

Este script comprueba si hay alguna máquina virtual en ejecución, y si no encuentra ninguna, suspende nuestro equipo y lo reanuda a las 7 de la mañana del día siguiente. Útil para ahorrar energía y no tener que preocuparte por suspender tu equipo, ni de encenderlo por las mañanas.

El script te avisa de que el sistema se va suspender y a los 10 minutos lo suspende. Así que tienes que ejecutarlo 10 minutos antes de la hora a la que quieres que el ordenador se suspenda (*También puedes hacer que no avise al usuario y el ordenador se suspenda inmediatamente pasándole el argumento "now" al script, de forma que quede así: "wake now"*). Para suspender tu equipo automáticamente todos los días a las 11:00 puedes añadir a tu crontab:

```
50 22 * * * root wake
```

4.5. wakeme

Este script es un despertador, hace sonar el archivo de audio especificado hasta que le damos al un botón que apaga nuestra alarma. Si queremos que el despertador suene todas las mañanas a las 7:30, podemos añadir una línea parecida a esta a `/etc/crontab`:

```
30 7 * * * usuario wakeme ~/musica/alarma.mp3
```

4.6. Mantenimiento btrfs

Si elegiste particionar tu disco con BTRFS es recomendable que se ejecuten tareas de mantenimiento cada *X* tiempo. Añadir esto a tu `/etc/crontab` bastará:

```
@weekly root btrfs balance start -dusage=95 -musage=95 /
```

Esto realizará diversas tareas de mantenimiento en tu disco principal una vez por semana. Si tienes un disco diferente para tu partición home, también formateado en BTRFS, puede interesarte hacer lo mismo para tu disco `/home`

```
@weekly root btrfs balance start -dusage=95 -musage=95 /home
```

4.7. Limpiar cache periódicamente

Puedes limpiar periódicamente la caché y archivos de registro antiguos de forma periódica añadiendo estas líneas a `/etc/crontab` sustituyendo *usuario* por el nombre de tu usuario:

```
@daily usuario find ~/.cache -mtime +2 -delete
@hourly usuario find ~/ -name "*.log" -mtime +4 -delete
```

4.8. Reorganizar plugins

Si elegiste instalar herramientas de producción musical, se instaló [yabridge](#), una aplicación para usar plugins [VST](#) de Windows en Linux. Es recomendable configurar el escaneo de nuevos plugins y el mantenimiento de tu librería de plugins para que se haga automáticamente.

```
@hourly usuario yabridgectl sync --prune
```

4.9. compressed-backup

Este script crea un fichero comprimido *tar.gz* con una copia de seguridad del directorio que se le da por primer argumento en el directorio que se le da por segundo argumento. Además se encarga de borrar las copias de seguridad que tienen mas de un mes automáticamente.

Por ejemplo, añadir esta línea a nuestro `/etc/crontab`:

```
@daily usuario compressed-backup ~/Música /Resguardos/Música
```

Nos crearía todos los días una copia de seguridad de la carpeta `~/Música` en un archivo comprimido en `/Resguardos/Música`

4.10. crypt-backup

Este script crea una copia de seguridad de nuestros archivos más importantes, necesita como argumento el directorio donde esta montado el disco que albergará nuestra copia de seguridad.

El script solo se ejecutará si el disco que elegimos para las copias de seguridad es un disco encriptado. *El script no se ejecutará si el disco no esta motado o no esta encriptado, pues en este se harán copias de la llave de encriptado de nuestra instalación.* Para hacer una copia periódicamente en `/mnt/backup` debemos añadir una línea parecida a nuestro *crontab*:

```
@daily root crypt-backup /mnt/backup
```

5. VFIO GPU passthrough

VFIO (Virtual Function I/O) GPU passthrough es una forma de asignar una tarjeta gráfica física a una máquina virtual. Esto nos permite tener gráficos acelerados dentro de una máquina virtual, por ejemplo, para jugar a videojuegos.

Si a la hora de usar el instalador eligió usar máquinas virtuales, *virt-manager* debería haberse instalado y configurado para usar máquinas virtuales con QEMU y KVM.

Sin embargo, crear una máquina virtual y auto-configurar el uso de una gráfica dedicada a través de VFIO es un proceso más complejo, que prefiero dejar en manos del usuario.

A continuación te acompañare en el proceso:

5.1. Pasos Iniciales

- Antes de usar máquinas virtuales necesitas tener [VT-d o AMD-v](#) activado *dependiendo de si tienes un procesador Intel o AMD*, que es la tecnología que nos permite la ejecución de máquinas virtuales.
- Debes de tener [IOMMU \(Unidad de administración de memoria de entrada/salida\)](#) activado, lo mas probable es que en tu placa base, activar VT-d o AMD-v active también IOMMU. Incluyo este requisito para los raros casos en los que activarlo es un ajuste por separado.
- Debes tener desactivado [CSM \(Compatibility Support Module\)](#). Esto normalmente se hace en los ajustes de arranque de tu placa base.

5.2. Pre-configurar el gestor de arranque

El gestor de arranque que estamos usando es [GNU GRUB](#), es el programa que se encarga de cargar el kernel de nuestro sistema operativo.

Para poder usar VFIO necesitamos configurar como arranca nuestro sistema operativo. Las opciones de configuración globales de GRUB están en `/etc/default/grub`

En este archivo, tendremos que localizar la línea donde están las opciones de arranque, la podemos identificar por `GRUB_CMDLINE_LINUX` y el archivo es algo parecido a:

```
GRUB_CMDLINE_LINUX_DEFAULT="loglevel=3 quiet"
GRUB_CMDLINE_LINUX=""
```

Tendremos que añadir las siguientes opciones:

- `iommu=pt`
- `amd_iommu=on` o `intel_iommu=on`, dependiendo de si tenemos una cpu Intel o AMD.
- `video=efifb:off`

Al finalizar, nuestro archivo debería verse así:

```
GRUB_CMDLINE_LINUX_DEFAULT="loglevel=3 quiet"
GRUB_CMDLINE_LINUX="amd_iommu=on iommu=pt video=efifb:off"
```

Una vez editado el archivo, debemos actualizar nuestra configuración de GRUB, ejecutando el siguiente comando con permisos de administrador, y después de esto vamos a reiniciar nuestro ordenador para avanzar al siguiente paso:

```
grub-mkconfig -o /boot/grub/grub.cfg
```

5.3. Identificar el ID de nuestra gráfica y los grupos IOMMU

Después de esto ya podemos asignar el driver VFIO a nuestra tarjeta gráfica para poder usarla en nuestra máquina virtual.

Necesitamos conocer el identificador de nuestra gráfica, que es una cadena de caracteres que identifica inequívocamente un dispositivo PCI (*en nuestro caso nuestra tarjeta gráfica*). Para poder obtener esta información, deberás ejecutar esta función con BASH.

```
shopt -s nullglob
for g in /sys/kernel/iommu_groups/*; do
    echo "IOMMU Group ${g##*/}:"
    for d in $g/devices/*; do
        echo -e "\t$(lspci -nns ${d##*/})"
    done;
done;
```

Una vez ejecutes este bloque de comandos, deberás obtener una salida parecida a la siguiente:

```
IOMMU Group 15:
08:00.0 VGA compatible controller [0300]: NVIDIA TU116 [GeForce GTX 1660] [10de:21c4] (rev a1)
08:00.1 Audio device [0403]: NVIDIA TU116 High Definition Audio Controller [10de:1aeb] (rev a1)
08:00.2 USB controller [0c03]: NVIDIA Device [10de:1aec] (rev a1)
08:00.3 Serial bus controller [0c80]: NVIDIA TU116 [GeForce GTX 1650] [10de:1aed] (rev a1)
```

Debes fijarte en que grupo se encuentra tu tarjeta gráfica, en este caso, se trata de una MSI GTX 1660.

Ahora que tienes identificado en que grupo esta tu tarjeta gráfica, nos tenemos que fijar en las ID, que están contenidas por corchetes, para nuestro ejemplo serían:

- [10de:21c4] Tarjeta de video (GPU) compatible con VGA, se correspondería con el procesador gráfico.
- [10de:1aeb] Es el dispositivo de audio de nuestra gráfica, se encarga por ejemplo, de darnos audio através de HDMI
- [10de:1aec] Este es el controlador USB de nuestra gráfica
- [10de:1aed] Este es el controlador de bus serie de nuestra gráfica

Ahora debemos añadir cada uno de estos ID (*sólo el texto entre corchetes*, separados por comas a la opción de GRUB; "vfio-pci.ids=". Si añadimos esta opción a nuestro archivo de configuración de GRUB, este quedaría como:

```
GRUB_CMDLINE_LINUX_DEFAULT="loglevel=3 quiet"
GRUB_CMDLINE_LINUX="amd_iommu=on iommu=pt video=efifb:off
vfio-pci.ids=10de:21c4,10de:1aeb,10de:1aec,10de:1aed"
```

Ejecutamos de nuevo (Con permisos de administrador):

```
grub-mkconfig -o /boot/grub/grub.cfg
```

Esto lo que hace es decirle al kernel que utilice el driver VFIO para todos esos dispositivos PCI, por el contrario, si no hiciésemos esto, el kernel le asignaría el driver *nvidia* o el correspondiente para un uso normal.

5.4. Módulos del kernel

Ahora cada vez que iniciemos nuestro ordenador el kernel intentará asignarle a esos dispositivos PCI el driver VFIO, *pero hay un problema*. El driver VFIO por defecto, no se carga en memoria a la hora del arranque del sistema (*cosa que tiene sentido, poca es la gente que utiliza esta función del sistema operativo y cargarla en memoria por defecto sería un desperdicio de memoria*).

Debemos entonces configurar que cuando arranquemos el ordenador, se nos cargue el driver VFIO, para poder asignárselo a nuestra gráfica.

Para esto debemos editar `/etc/mkinitcpio.conf` y añadir los módulos del kernel que vamos a usar. El archivo se vería algo así:

```
MODULES=()
BINARIES=()
FILES=()
HOOKS=(base udev autodetect modconf block filesystems keyboard fsck)
```

Esto es una simplificación, y el archivo real tendría un montón de líneas encabezadas por `#` (comentarios), pero nos sirve para hacernos una idea.

Debemos localizar la línea con `MODULES=()` y añadir los módulos que queremos cargar con el arranque del ordenador. Si antes los paréntesis no contenían nada, ahora la línea pasaría a verse así:

```
MODULES=( vfio_pci vfio vfio_iommu_type1 )
```

Una vez editado nuestro archivo debemos regenerar el [initramfs \(Initial RAM File System\)](#) ejecutando con permisos de administrador el comando:

```
mkinitcpio -P
```

Vamos a comprobar que los cambios se han realizado correctamente. Para esto vamos a reiniciar nuestro ordenador y cuando arranque el ordenador de nuevo, vamos a ejecutar:

```
dmesg | grep -i vfio
```

Si tenemos una salida parecida a:

```
[3.416692] vfio_pci: add [10de:21c4[ffffffff:ffffffff]] class 0x000000/00000000
[3.433353] vfio_pci: add [10de:1aeb[ffffffff:ffffffff]] class 0x000000/00000000
[3.450019] vfio_pci: add [10de:1aec[ffffffff:ffffffff]] class 0x000000/00000000
[3.466953] vfio_pci: add [10de:1aed[ffffffff:ffffffff]] class 0x000000/00000000
```

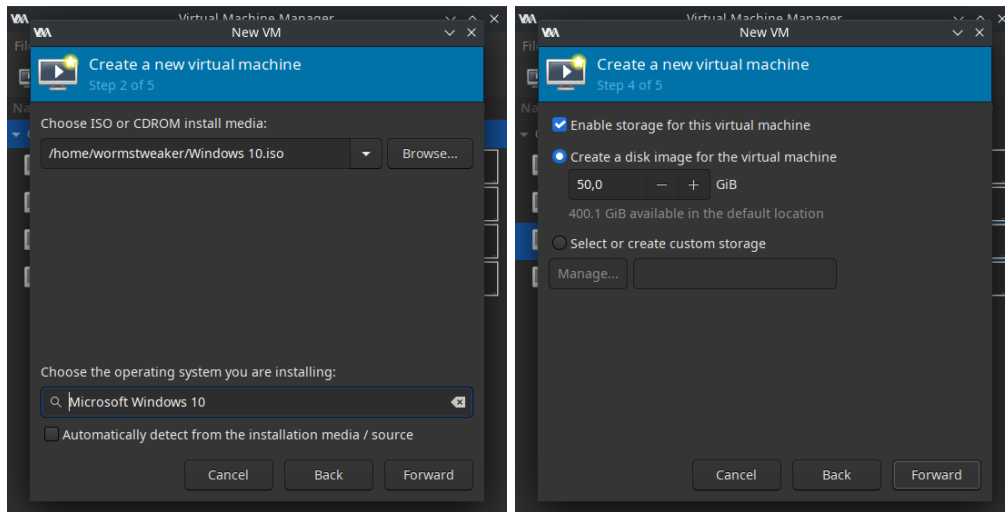
donde podemos ver que el driver `vfio_pci` se ha cargado correctamente para nuestros dispositivos, entonces hemos realizado correctamente todos los pasos.

5.5. Instalación del sistema operativo

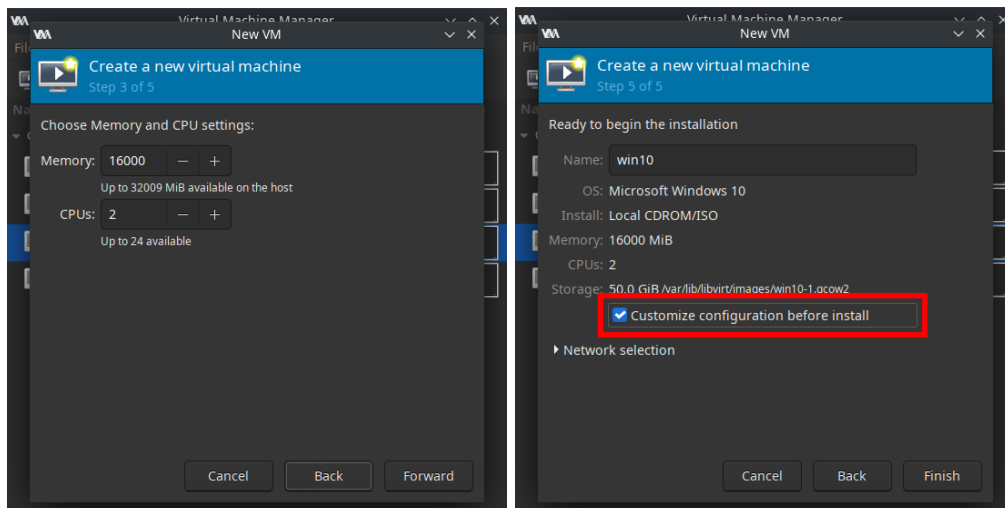
Ya tenemos lista nuestra gráfica para ser usada por nuestra máquina virtual, queda instalar nuestra máquina virtual y configurarla para usarla cómodamente.

Primero vamos a descargar la ISO de Windows desde <https://www.microsoft.com/en-us/software-download/windows10ISO>, y el ISO con los drivers estables de virtio para Windows.

Una vez descargada la ISO, abriremos *virt-manager* y crearemos una máquina virtual, vamos a seleccionar nuestra ISO y el tamaño que queremos asignarle al disco:



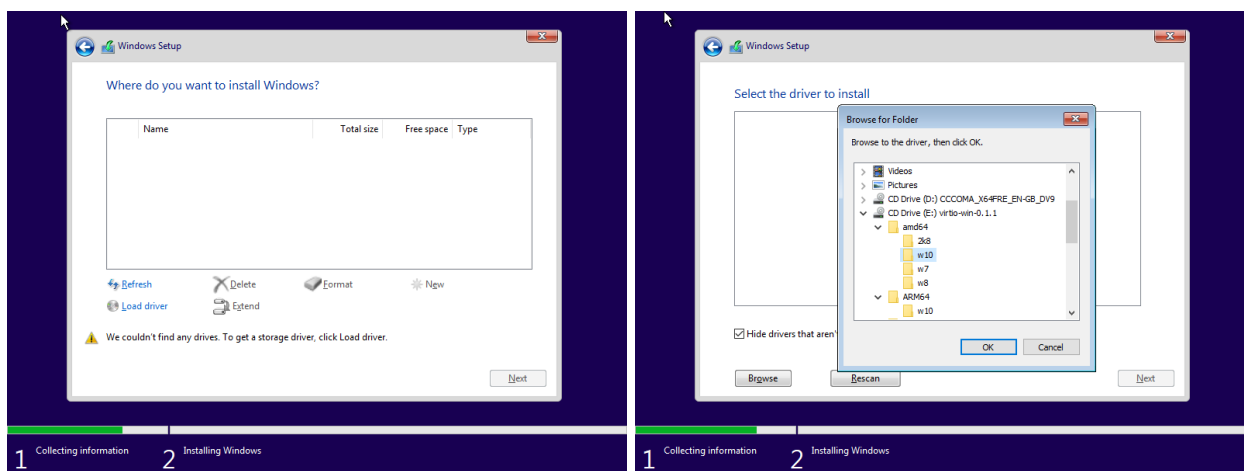
Vamos a asignarle la memoria RAM (*mi recomendación es asignar, como mínimo, 8GB*) a nuestra máquina virtual. Y en el último paso, vamos a elegir el *customizar nuestra máquina virtual antes de la instalación* (Señalado en la imagen).



Ahora deberás hacer las siguientes modificaciones:

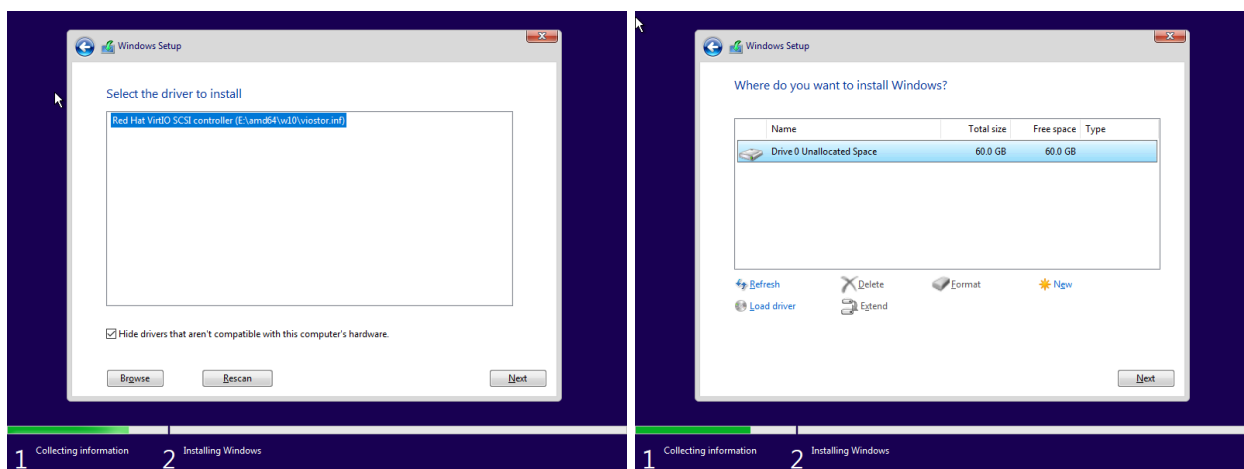
- Cambiar el chipset a Q35
- Elegir el firmware UEFI:x86_64:/usr/share/edk2/x64/OVMF_CODE.fd
- En la pestaña *CPUs* activa el recuadro *Copiar configuración de la CPU del anfitrión*, despliega el menú *Topología* y ajusta la topología de la forma siguiente:
 - 1 *Socket*, Tantos *Centros* como núcleos tenga tu procesador y tantos *Hilos* como hilos tenga tu procesador por núcleo. Por ejemplo para un procesador con 8 núcleos y 16 hilos, elegiríamos: 1 *Socket*, 8 *Centros*, 2 *Hilos*. Esto es, si queremos asignar todos los núcleos a nuestra máquina virtual.
- Nos dirigimos a *SATA Disk 1* y cambiamos el bus a *VirtIO*, de forma que ahora aparezca *VirtIO Disk 1*. Abrimos las opciones avanzadas, y cambiamos el modo de caché a *writeback*.
- En los ajustes del *NIC* cambiamos el modelo de dispositivo a *virtio*

- Por último dale a *Añadir hardware* y añade un almacenamiento de tipo de dispositivo CDROM y escribe al lado de *Administrar...* la localización de el ISO con los drivers virtio, por ejemplo: `/home/usuario/Downloads/virtio-win.iso`.



Dale a *Iniciar la instalación* y cuando veas la selección de disco del instalador de windows verás que está vacío. Primero debes darle a *Cargar driver* o *Load driver*, en inglés.

Selecciona el CD con los drivers virtio y la carpeta que contiene los drivers para Windows 10. Entonces aparecerá tu disco duro virtual y podrás continuar la instalación de Windows. Cuando esta finalice, apaga la máquina virtual.



5.6. Looking Glass en el Host

Para poder usar nuestra máquina virtual de forma cómoda vamos a configurar [Looking Glass](#). Una aplicación que nos permite ver la pantalla conectada a nuestra gráfica a través de una ventana y poder controlar nuestra máquina virtual sin necesidad de conectar un teclado y ratón sólo para nuestra máquina virtual. El único problema es que para usar looking glass sin necesidad de dos monitores, necesitamos comprar un [display dummy](#). Por lo menos, su precio suele rondar los 5-7€.

Looking Glass nos muestra una imagen de lo que esta haciendo nuestra gráfica con una latencia muy baja, suficiente para poder jugar a juegos o editar vídeo (y otros usos posibles) sin problema en nuestra máquina virtual. Logra esto mandando la información a través de un archivo compartido entre máquina virtual y host.

Primero necesitamos configurar **tmpfilesd** para que cada vez que iniciamos el sistema, cree el archivo que compartirán máquina virtual y host para enviarse información. Tenemos que ejecutar el siguiente comando (*sustituyendo usuario por el nombre de tu usuario*):

```
echo "f /dev/shm/looking-glass 0660 usuario kvm -" | \
doas tee -a /etc/tmpfiles.d/looking-glass.conf
```

Tenemos que instalar el servicio que se encarga de crear estos archivos con:

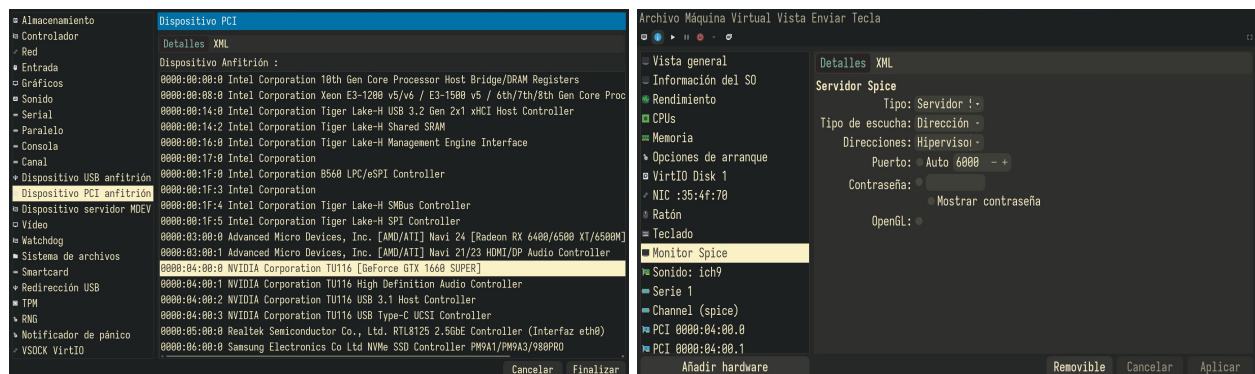
```
doas pacman -Sy --noconfirm etmpfiles
```

5.7. Añadir Gráfica y Looking Glass en el Guest

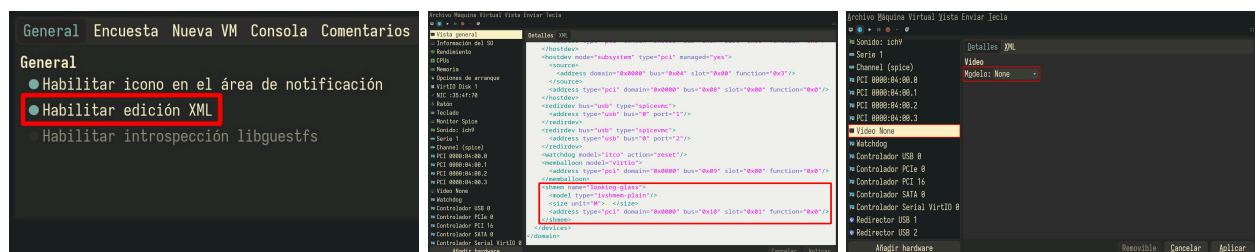
Ahora ya tenemos casi todo listo, queda instalar los drivers de vídeo e instalar Looking Glass en nuestra máquina virtual (*a la hora de la instalación debió de haberse instalado ya en nuestro linux*).

Primero tenemos que añadir nuestra tarjeta gráfica a nuestra máquina virtual. Para ello tenemos que ir a los ajustes de nuestra máquina virtual y en “Añadir Dispositivo”, busca la pestaña para añadir dispositivos PCI y añade tu gráfica.

Después debemos de configurar el servidor Spice para que use el puerto 6000 en vez de asignar un puerto automáticamente



Ahora tenemos que activar la edición de XML en Virt-Manager (Editar → Preferencias) y editar el código XML de nuestra máquina virtual. Añadiremos estas líneas de código al final para que el archivo acabe viéndose así:



Iniciamos nuestra máquina virtual e instalamos nuestros drivers de **NVIDIA** o **AMD** y **Looking Glass Host**.

Finalmente cambiamos el driver virtual de video de QXL a *None* y borramos el dispositivo de tableta táctil. La próxima vez que iniciemos la máquina virtual solo tenemos que iniciarla en Virt-Manager e interactuaremos con nuestra máquina virtual abriendo Looking Glass (*Podemos abrirlo seleccionando “Looking Glass” en el menú que aparece cuando hacemos clic derecho en alguna parte vacía del Escritorio*).

Para que esto funcione debemos reiniciar nuestro ordenador o ejecutar el comando.

```
doas install -g kvm -o $(whoami) -m 0660 /dev/null /dev/shm/looking-glass
```

Nota

Para más información sobre como configurar un VFIO passthrough puede consultar:

- <https://qqq.ninja/blog/post/vfio-on-arch/>
- <https://gitlab.com/risingprismtv/single-gpu-passthrough/-/wikis/home>
- https://wiki.archlinux.org/title/PCI_passthrough_via_OVMF

6. VirtioFS

Podemos usar virtiofs para compartir directorios del host con nuestra máquina virtual. Para ello, necesitamos añadir los directorios deseados desde Virt-Manager, seleccionando: Añadir hardware, Sistema de archivos, eligiendo la Ruta de origen (la carpeta que queremos compartir) y la Ruta objetivo (el nombre con el que aparecerá en nuestra máquina invitada).

También necesitamos instalar los controladores de VirtIO en el host. Si todavía tenemos montado el disco que contiene estos controladores, encontraremos en el CD un instalador (virtio-win-gt-x64.msi) para los controladores, así como otro instalador *opcional* para añadir la integración con el host (virtio-win-guest-tools.exe), que permite funciones como compartir el portapapeles, entre otras.

Después de instalar los controladores, debemos instalar [WinFSP](#) para poder montar sistemas de archivos VirtioFS.

Una vez que tenemos todo instalado, podemos crear un script '.bat' que monte nuestras carpetas en la máquina invitada. Por ejemplo, si tenemos dos carpetas cuyas rutas objetivo son "documentos" y "videos", y queremos montarlas en "Y:" y "Z:", respectivamente, podemos crear un script para montar ambas de la siguiente manera:

```
"C:\Program Files (x86)\WinFsp\bin\launchctl-x64.exe" start virtiofs documentos documentos Y:
"C:\Program Files (x86)\WinFsp\bin\launchctl-x64.exe" start virtiofs videos videos Z:
```

7. SSH

Si queremos conectarnos a un equipo remoto a través de internet debemos configurar [OpenSSH](#). Por defecto incluyo un script que configura SSH de forma muy básica, configurando SSH para poder acceder a tu equipo remoto con una contraseña. Para activar SSH simplemente ejecuta:

```
ssh-configure
```

Mi recomendación es no conformarse con esta configuración básica. Desactiva el login con el usuario root, desactiva el login por contraseña y usa claves públicas para conectarte. Si no sabes de lo que estoy hablando, es mejor que no uses SSH y no te expongas a abrir la puerta a tu ordenador al internet, aunque sea con candado.

7.1. VNC a través de SSH

Podemos usar SSH para acceder a nuestro entorno gráfico de forma remota y usar nuestro ordenador sin estar necesariamente delante de él físicamente. Por defecto dwm inicia un servidor VNC para que puedas conectarte remotamente a una interfaz gráfica. Para poder hacer uso del servidor VNC tienes que usar [tunneling](#). Para conectarte y poder usar VNC, ejecuta:

```
ssh usuario@255.255.255.255 -L 5900:localhost:5900
```

Sustituye *usuario* por el usuario de tu máquina, y *255.255.255.255* por la dirección IP de tu máquina. Para conectarte al equipo remotamente debes tener instalado algún cliente VNC, mi recomendación es [remmina](#).

8. Firewall

Instalar y configurar un firewall no es posible dentro de un chroot. Por eso configurar el firewall debe hacerse una vez reiniciemos y hallamos arrancado nuestro sistema operativo. Para instalar el firewall ejecuta:

```
doas sh -c 'pacman -Sy ufw-openrc; rc-update add ufw default; ufw enable'
```

Ahora, la próxima vez que inicies tu equipo tendrás un firewall ya instalado. Para configurarlo puedes ejecutar los siguientes comandos, que establecen unas reglas de filtrado de paquetes bastante permisivas.

```
doas ufw limit 22/tcp; doas ufw allow 80/tcp
doas ufw allow 443/tcp; doas ufw allow syncthing
doas ufw default deny incoming; doas ufw default allow outgoing
```

Nota

Si ejecutaste *ssh-configure* el firewall ya se configuró automáticamente

9. Syncthing

[Syncthing](#) es un software para sincronizar archivos entre dispositivos, es lo que uso para poder trabajar en todos mis equipos sin necesidad de estar intercambiando pen-drives como un troglodita.

Viene instalado por defecto y se ejecuta cuando inicias tu ordenador. Para configurarlo tiene una [interfaz gráfica](#) a la que puedes acceder desde tu navegador.

Información

Aquí puedes encontrar documentación sobre como usar syncthing:
<https://docs.syncthing.net/>

10. Monitores de altas tasas de refresco

Por defecto dwm actualiza el movimiento de las ventanas a 60 FPS. Si tienes un monitor de más de 60 Hz esto puede hacer la experiencia de usar dwm insatisfactoria. Para cambiar este comportamiento, debes editar el archivo *dwm.c*. En él encontrarás varias funciones con esta misma línea de código:

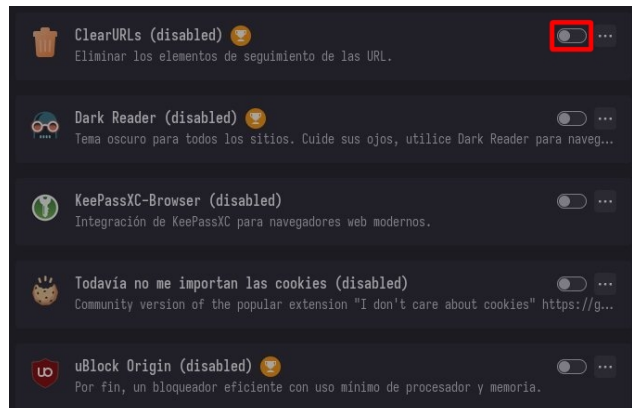
```
if ((ev.xmotion.time - lasttime) ≤ (1000 / 60))
```

Deberás cambiar el valor *60* por la tasa de refresco de tu monitor en cada aparición de esta línea. Por ejemplo, para un monitor de *144 Hz*, la línea a cambiar deberían verse así: `if((ev.xmotion.time-lasttime)<=(1000/144))`

11. Firefox

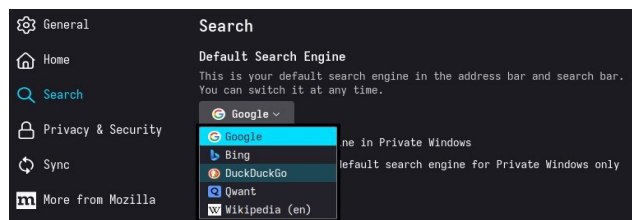
11.1. Activar extensiones

Nuestro Firefox viene instalado con varias extensiones útiles que vienen desactivadas por defecto. Para activarlas, pulsa en el icono con forma de pieza de puzle, que está situado a la derecha de la barra de búsqueda. Te encontrarás con una lista como la de la imagen de abajo. Simplemente activa cada extensión haciendo clic en el deslizador señalado y dale a aceptar en los menús flotantes que aparezcan pidiendo confirmación.



11.2. Cambiar buscador por defecto

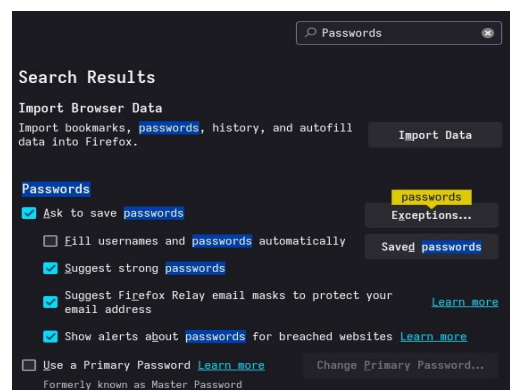
Pese a usar al configuración reforzada de Firefox, [arkenfox](#), esta no cambia el que el buscador por defecto de Firefox es Google. Mi recomendación es cambiar Google por DuckDuckGo.



11.3. Desactivar baúl de contraseñas

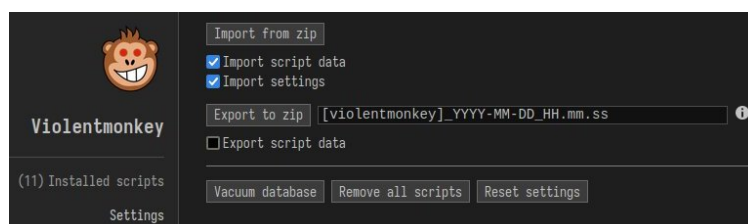
Por defecto Firefox intentará manejar nuestras contraseñas, si ya estamos usando la extensión de Keepass que viene incluida con nuestra instalación, las ventanas de Firefox que aparecen preguntando si queremos guardar nuestra contraseña pueden ser bastante molestas.

Para desactivar esta función, simplemente busca *Contraseña* en el menu de configuración de Firefox y desactiva el *Ask to save passwords* o *Preguntar para guardar contraseñas*, en español.



11.4. Violentmonkey

Violentmonkey es una extensión para cargar pequeños scripts de javascript justo después de cargar la página web, para automatizar ciertas tareas o añadir funcionalidades. Si vas a los ajustes de la extensión, verás que hay una pestaña donde puedes importar tus ajustes. Podemos importar nuestros ajustes que están en el archivo:



~/dotfiles/assets/violentmonkey.zip

11.5. Añadir buscadores útiles

Aquí tienes una lista de buscadores útiles, para añadirlos a tu buscador simplemente accede a cada página haciendo doble clic, y cuando estés en ella haz clic derecho en la barra de navegación y selecciona *Añadir "Buscador"*.

- [Arch Wiki](#)
- [Alpine Wiki](#)
- [Piped](#)
- [GitHub](#)
- [Nyaa](#)
- [Gentoo Wiki](#)
- [YouTube](#)
- [Invidious](#)
- [Yandex](#)
- [Stack Overflow](#)

Las Wikis de Gentoo y Alpine nos resultarán de utilidad, pues ambas distribuciones utilizan OpenRC en vez de SystemD. De hecho, si no encuentras un script de OpenRC para algún servicio en los repositorios de Artix, puedes copiar el script que utilizan Alpine o Gentoo.

12. Miscelánea

Configuraciones locales

Nuestra configuración de zsh tiene varios *alias* configurados en:

```
~/.config/zsh/aliasrc
```

Si deseamos añadir alias de forma no global, sin tener que añadir archivos al repositorio y de manera exclusiva para nuestro equipo, podemos crear un archivo con más funciones y abreviaciones en:

```
~/.config/useraliases
```

Aviso de Uso

Este proyecto es para uso personal. Se comparte con la intención de que pueda ser útil para otros, pero se proporciona tal cual, sin ninguna garantía. No se asume responsabilidad por ninguna pérdida de datos o problemas que puedan surgir del uso de este proyecto.

No obstante, se aceptan propuestas de mejora razonables. Si alguien desea contribuir a hacer este setup más extensible y robusto, puede realizar un merge request sin inconveniente.