

Fuente: Bing Chat AI



MV PARA LA ASIGNATURA



MÁSTER EN INGENIERÍA WEB 2023 – 2024

Administración de Sistemas Operativos v1.2 ("L-62 Princesa de Asturias")

© José Manuel Redondo López. Universidad de Oviedo



Contenido

Antes de empezar	4
¿Por qué todo esto?	4
Soporte de virtualización en la BIOS del host	4
GUI y terminales	5
Uso de terminales	5
TMux: Terminales y paneles	5
Midnight Commander: Explorador de archivos	8
SSH a la máquina virtual	8
Edición de archivos de texto	9
Conocimiento de Linux	9
Redondo, ¿Por qué entregas una máquina optimizada y una que no está optimizada?	10
Opciones de creación de la máquina virtual para el curso	12
Opción 1: Importación de la imagen .ova suministrada (fácil)	12
¿Por qué elegir este método?	12
Opción 2: Usar Vagrant para construir automáticamente una máquina virtual (fácil-medio)	14
Opción 3: Creación de tu propia máquina virtual (difícil, no se recomienda)	15
¿Por qué elegir este método?	15
Configuración de una plataforma de virtualización para instalar un servidor Linux	15
USB	22
Almacenamiento y selección de la ISO del sistema operativo	22
Instalar un servidor Ubuntu Linux	24
Configuración de la máquina para mejorar la virtualización	29
Software para instalar en la máquina virtual personalizada	31
Resolución de problemas general (cualquier opción de creación de máquinas)	33
Cuando arranco la MV, me da el error VERR_INTNET_FLT_IF_NOT_FOUND	33
Mi MV arranca, pero solo veo una pantalla en negro y ninguna salida	33
Necesito Hyper-V para otras asignaturas / mi trabajo, y desactivarlo cada vez que arranco la VM es muy pesado	34
Operaciones de gestión de MV generales	36
Resolviendo incompatibilidades entre VirtualBox e Hyper-V	36
Clonación y configuración de máquinas	37
Creación de instantáneas de máquinas en funcionamiento	39
Tomar, restaurar y eliminar instantáneas	40
Carpetas compartidas	47



1

2

3



Antes de empezar

¿Por qué todo esto?

Este documento solo tiene un objetivo: darte opciones para **implementar la máquina virtual que necesitas para seguir la asignatura en casa**, sin importar tu sistema de virtualización, sistema operativo o capacidades de *hardware*. El motivo de hacer este documento es que, **aunque la creación de una máquina virtual puede hacerse perfectamente usando los conocimientos del módulo de Vagrant**, este módulo es de realización opcional, por lo que ésta es la alternativa para que siga siendo así. Eso sí, necesitas un mínimo para hacerlo:

- Cualquier CPU de 64 bits con extensiones de virtualización de hardware (VT-X, AMD-V) (las CPUs lanzadas a partir del 2012 deberían cumplir este requisito)
- Posibilidad de asignar 2 núcleos a la máquina virtual (1 también debería ser posible, pero con un coste de rendimiento sustancial). Recomendamos no utilizar una GUI con 1 núcleo solamente.
- Posibilidad de asignar 2Gb de RAM a la máquina virtual (4Gb sería lo ideal).
- Un máximo de 80-120 Gb para el disco duro de la máquina virtual, que será dinámico.

Te proporcionaremos tres opciones para crear la máquina, cada una con ventajas y desventajas: por favor, elige la que más te beneficie, pero te **recomiendo usar la máquina que entrego prefabricada** para usar tu tiempo en hacer solamente actividades.

Soporte de virtualización en la BIOS del host

Hoy día, para poder virtualizar máquinas es necesario habilitar el soporte de **virtualización por hardware** en la BIOS de tu ordenador. Esta opción se encuentra en diferentes lugares **dependiendo del fabricante de la BIOS**, pero generalmente está en el grupo de opciones avanzadas y está descrita con texto como "Intel Virtualization Technology", "Virtualization Technology", "Enable Virtualization Technology", "Enable VT-X" o "Enable AMD-V". **Es necesario** que te **asegures de que esta opción está activa**, sobre todo porque **muchos fabricantes la desactivan por defecto en la BIOS** y, si no la has usado anteriormente, podría estar deshabilitada. Sin esta opción **no puedes ejecutar la mayoría del software de virtualización moderno**. Afortunadamente, todas las CPU Intel y AMD a partir de 2010 implementan esta tecnología.



Ejemplo de soporte en BIOS de tecnología de virtualización por hardware

GUI y terminales

La máquina virtual que suministramos o la que construirás tiene un GUI ligero (XFCE4) para facilitar tu trabajo en el curso. Esta GUI es accesible escribiendo **startx** en la línea de comandos, porque **no arranca automáticamente** para ahorrar recursos. La decisión de usar una GUI o no es tuya: ahorrarás tiempo, pero a un costo de recursos. Si prefieres no utilizar una GUI (o tu *hardware* no te permite tener una) te doy una opción diferente: utilizar funciones avanzadas de terminal. Voy a detallarlas más aquí, porque se pueden utilizar sin importar si eliges o no un GUI.

Uso de terminales

Incluso si no arrancamos el GUI para minimizar el uso de recursos podemos hacer múltiples tareas al mismo tiempo. Un usuario, una vez que ha iniciado sesión en el sistema, **puede abrir varios terminales de consola** simplemente con Alt-F1, Alt-F2, etc. Si es la primera vez que se abres un terminal asociado a una tecla de función, se te volverá a preguntar el usuario y contraseña. Los próximos cambios de terminal continuarán con la actividad que se estaba realizando en ella, sin necesidad de autenticarse otra vez. Esto es muy útil, ya que permite realizar tareas en paralelo, por ejemplo, editar y guardar una configuración en un terminal dejándolo abierta y tener otro para hacer reinicios del servicio asociado a la misma.

```
Ubuntu 18.04.5 LTS ssiserver tty1
ssiserver login: ssiuser_
```

Nuevo terminal, resultante de la secuencia de teclas Alt-F2

TMux: Terminales y paneles

tmux es un paquete que te permite tener múltiples ventanas de terminal en la misma pantalla y dividir una ventana en diferentes paneles que pueden funcionar simultáneamente. **tmux** está preinstalado en la



máquina virtual prefabricada, con integración de ratón (módulo **gpm**) y soporte de colores de terminal. Podría ser una buena idea si quieres más agilidad y productividad sin usar una GUI, ya que te brinda parte de las características de una GUI con mucho menos consumo de recursos. También puede ser útil en otros cursos que te obligan a usar mucho el terminal.

```
# vnStat 1.11 config file
##

# default interface
Interface "eth0"

# location of the database directory
DatabaseDir "/var/lib/vnstat"

# locale (LC_ALL) ("-" = use system locale)
Locale "-"

# on which day should months change
MonthRotate 1

# date output formats for -d, -m, -t and -w
# see 'man date' for control codes
DayFormat "%x"
MonthFormat "%b '%y"
TopFormat "%x"

<tc/vnstat.conf" [readonly] 129L, 2889C
[0] 0: bash* "pungki@dev-machine:~" 11:24 12-Jan-14
```

Este *cheatsheet* te muestra las opciones más típicas de **tmux**. Si decides crear la máquina virtual tú mismo, la máquina virtual prefabricada tiene un archivo **tmux.conf** con una buena configuración inicial que puedes copiar en **/etc/tmux.conf** (también proporciona una configuración **nano** recomendada que puedes copiar a **<tu directorio home>/nanorc**)

TMux 3.1 Cheatsheet (ingenieriainformatica.uniovi.es) Console enhancer and multi-pane support tool https://github.com/tmux/tmux/wiki	
GENERAL USAGE	
tmux	
NOTES	
Press Ctrl + B, release them, and then press the corresponding key exit on the last pane exits tmux	
OPTIONS	
Window handling	
New window: CTRL+b, release and then c	
Next window: CTRL+b, release and then n	
List all windows: CTRL+b, release and then w	
Change window name: CTRL+b, release and then ,	
Panel management	
Vertical: CTRL+b, release and then %	
Horizontal: CTRL+b, release and then " (press Uppercase key)	
Hide panel: CTRL+b, release and then x	
Show panel number: CTRL+b, release and then q	
Change panel: CTRL+b, release and then the corresponding arrow key	
Close current panel: exit	
Copy and paste text	
- CTRL+b, release and then [: Enter "copy" mode.	
- Move to start/end of text to highlight.	
- CTRL + space and start highlighting text. Selected text change color. Move to opposite end of text to copy.	
- ALT + w: Copies selected text into tmux clipboard. (On Mac use Esc+w.)	
- Go to the destination pane and windows and put the cursor where you want to paste the text.	
- CTRL+b, release and then]: Paste copied text from tmux clipboard	

```
00:00 UTC; path=/;";document.cookie=" gat=; expires=Thu, 01 Jan 1970 00:00:00 U
TC; path=/;";window.cookieconsent.initialise({palette:{popup:{background:"#edeff
5",text:"#838391"},button:{background:"#4b81e8"}},type:"opt-in",content:{message
:"Nuestra web utiliza cookies propias y de terceros para analizar sus hbitos de
navegaci n y elaborar estad sticas sobre su interacci n con nuestro sitio web,
as como mostrar botones de compartici n de contenido en redes sociales. Puede o
btener m s informaci n en nuestra Pol tica de Cookies.",dismiss:"Descartar",allo
w:"Aceptar",deny:"Rechazar",link:"M s informaci n",href:"/politicacookies",polic
y:"Pol tica de Cookies"}}));/*]]>*/</script> </body> </html> root@75b0892bfc
1:/#
root@75b0892bfc1:/#
```

```
64 bytes from 172.217.17.4 (172.217.17.4): icmp_seq=50 ttl=113 time=18.4 ms
64 bytes from 172.217.17.4 (172.217.17.4): icmp_seq=51 ttl=113 time=18.9 ms
64 bytes from 172.217.17.4 (172.217.17.4): icmp_seq=52 ttl=113 time=18.6 ms
64 bytes from 172.217.17.4 (172.217.17.4): icmp_seq=53 ttl=113 time=19.0 ms
64 bytes from 172.217.17.4 (172.217.17.4): icmp_seq=54 ttl=113 time=18.2 ms
64 bytes from 172.217.17.4 (172.217.17.4): icmp_seq=55 ttl=113 time=19.0 ms
64 bytes from 172.217.17.4 (172.217.17.4): icmp_seq=56 ttl=113 time=18.6 ms
64 bytes from 172.217.17.4 (172.217.17.4): icmp_seq=57 ttl=113 time=18.8 ms
64 bytes from 172.217.17.4 (172.217.17.4): icmp_seq=58 ttl=113 time=18.1 ms
64 bytes from 172.217.17.4 (172.217.17.4): icmp_seq=59 ttl=113 time=18.3 ms
```

```
[0] 0:bash: 1:bash* e75b0892bfc1 10:43 23-Dec-20
```

MORE INFORMATION
<http://www.sromero.org/wiki/linux/aplicaciones/tmux>
<https://www.hostinger.es/tutoriales/usar-tmux-cheat-sheet/>

Nice default configuration kindly provided by Alejandro Saez Morollón (put this in /etc/tmux.conf to apply it globally):

```
# Start windows and panes at 1, not 0
set -g base-index 1
setw -g pane-base-index 1
# Tmux fix for terminal colours
set -g default-terminal "screen-256color"
# Open new pane or window in the current directory
bind "" split-window -c "#{pane_current_path}"
bind % split-window -h -c "#{pane_current_path}"
bind c new-window -c "#{pane_current_path}"
# Avoid rename
set -g allow-rename off
# if run as "tmux attach", create a session if one does not already exist
new-session -A -s main
```

Otro cheatsheet de **tmux** es este:

[tmux sessions] linuxacademy.local		[tmux windows] linuxacademy.local	
_ new sessions tmux tmux new tmux new-session tmux new -s sessionname		_ windows are like tabs in a browser. Windows exist in sessions and occupy the space of a session screen.	
_ attach sessions tmux a tmux att tmux attach tmux attach-session tmux a -t sessionname		_ key bindings Ctrl + B C create window Ctrl + B N move to next window Ctrl + B P move to previous window Ctrl + B L move to window last used	
_ remove sessions tmux kill-ses tmux kill-session -t sessionname		Ctrl + B 0 .. 9 select window by number Ctrl + B ' select window by name Ctrl + B . change window number Ctrl + B , rename window Ctrl + B F search windows Ctrl + B & kill window	
_ key bindings Ctrl + B \$ rename session Ctrl + B D detach session Ctrl + B) next session Ctrl + B (previous session		_ key bindings G go to bottom h move cursor left j move cursor down k move cursor up l move cursor right / search # list paste buffers q quit	
[tmux panes] linuxacademy.local		[tmux copy mode] linuxacademy.com	
_ panes are sections of windows that have been split into different screens - just like the panes of a real window!		_ key bindings Ctrl + B [enter copy mode Ctrl + B] paste from buffer	
_ key bindings Ctrl + B % vertical split Ctrl + B " horizontal split Ctrl + B → move to pane to the right Ctrl + B ← move to pane to the left		_ copy mode commands space start selection enter copy selection Esc clear selection g go to top	
Ctrl + B ↑ move up to pane Ctrl + B ↓ move down to pane Ctrl + B O go to next pane Ctrl + B ; go to last active pane Ctrl + B } move pane right Ctrl + B { move pane left Ctrl + B ! convert pane to window Ctrl + B X kill pane			



Midnight Commander: Explorador de archivos

Una de las opciones que más se echan en falta cuando no se tiene una GUI es poder **navegar y ver fácilmente los archivos de los discos**. He incluido este navegador de archivos muy popular basado en consola para darte esta capacidad y aumentar tu productividad. Como muestra esta imagen, se puede integrar con **tmux** y tiene soporte para ratón.

```
root@e25d1ecf1e4c:~#
root@e25d1ecf1e4c:~# echo "Merry Christmas"
Merry Christmas
root@e25d1ecf1e4c:~# echo "Merry Christmas"
Merry Christmas
root@e25d1ecf1e4c:~# _
```

Left	File	Command	Options	Right			
<- ~	Name	Size	Modify time	<- ~	Name	Size	Modify time
..	UP--DIR	4096	Dec 25 09:11	..	UP--DIR	4096	Dec 25 09:11
./cache		4096	Dec 25 09:12	./cache		4096	Dec 25 09:12
./config		4096	Dec 25 09:12	./config		4096	Dec 25 09:12
./local		4096	Dec 25 09:12	./local		4096	Dec 25 09:12
./bashrc		3106	Dec 5 2019	./bashrc		3106	Dec 5 2019
./profile		161	Dec 5 2019	./profile		161	Dec 5 2019

UP--DIR 4441M/20G (22%)

Hint: Want your plain shell? Press C-o, and get back to MC with C-o again.

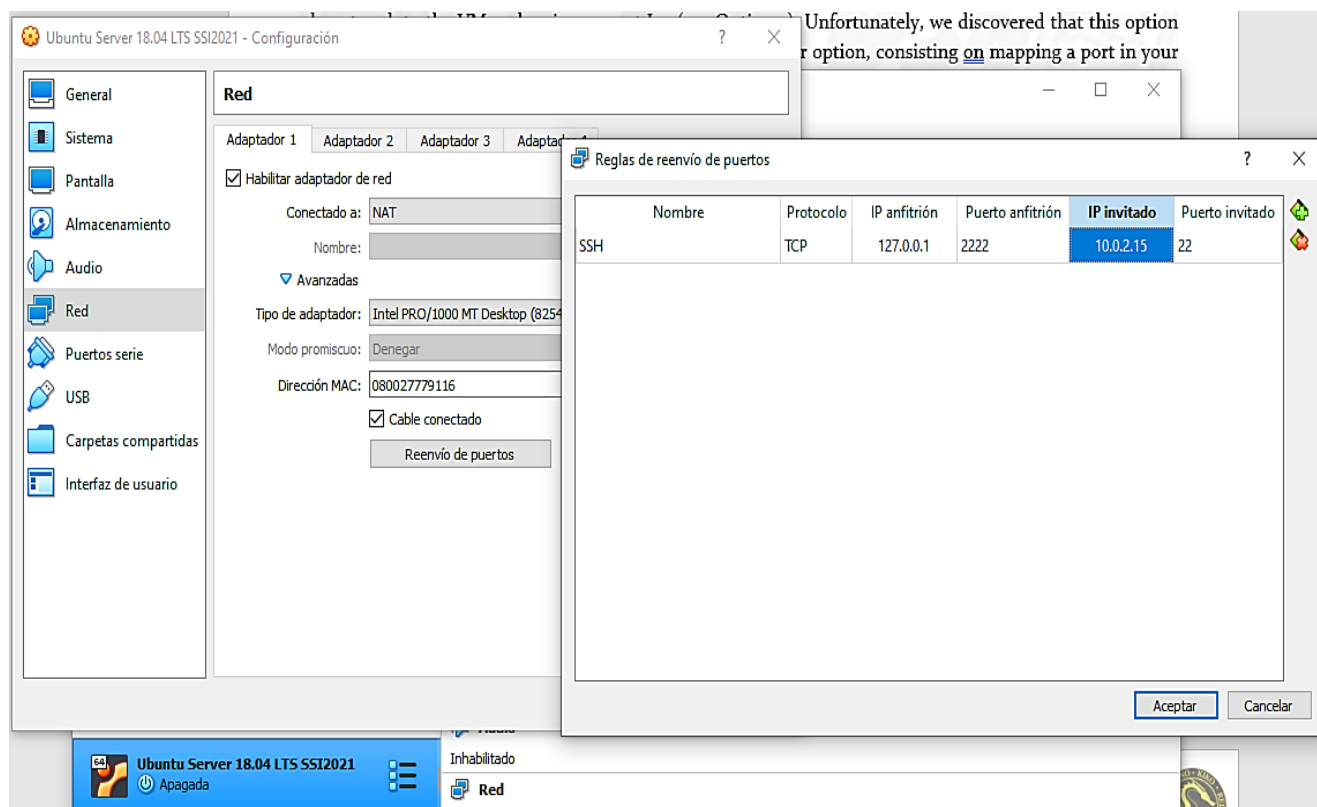
root@e25d1ecf1e4c:~#

1Help 2Menu 3View 4Edit 5Copy 6RenMov 7Mkdir 8Delete 9PullDn 10Quit

[1] 1:bash* "e25d1ecf1e4c" 09:14 25-Dec-20

SSH a la máquina virtual

Si utilizas un GUI generalmente no es necesario tener una conexión SSH, ya que la máquina virtual es totalmente compatible con copiar y pegar texto hacia / desde tu PC. Sin embargo, una forma directa de conectarse a tu máquina con un cliente SSH avanzado como *MobaXTerm* es crear una red *host-only* a la máquina virtual y asignar las IP correctas (consulta la opción 2 de crear máquinas virtuales). Si no quieres crear una red de este estilo, es posible acceder vía redirección de puertos yendo a la configuración de la máquina virtual, *Red*, elegir la interfaz NAT, hacer clic en *Avanzado*, redirección de puertos y configurar los ajustes que se ven en la imagen. Si haces esto para varias máquinas virtuales, utiliza diferentes puertos del *host*.



Edición de archivos de texto

Deberás editar archivos en consola muchas veces, salvo si decides usar la GUI. Para facilitar el trabajo, he creado una configuración **nano** especial que facilita esto, convirtiendo tabuladores en espacios en blanco, utilizando mejores colores, mostrando la posición de cursor, números de línea y realizando copias de seguridad automáticas de los archivos que edites. Puedes encontrar esta configuración en el archivo **.nanorc** del perfil de usuario predeterminado de la máquina virtual preconstruida, si quieres usarla en otros cursos o para su uso personal. La máquina prefabricada incluye los editores **mousepad** y **Visual Studio Code** (opción recomendada) para editar archivos en GUI.

Conocimiento de *Linux*

Sí, necesitas conocimientos de la línea de comandos de *Linux* para hacer este curso. Sin embargo, sé que es posible que hayas olvidado algunos. No te preocupes, ya lo imaginaba y para ayudarte te recomiendo lo siguiente:

- Utiliza este *cheatsheet* como guía para recordar los comandos típicos que necesitarás. No olvides que el comando **ifconfig** o el comando **ip** a te da los nombres e IPs de tus tarjetas de red actuales:

Some Basic Linux Commands

by @SecurityGuill

FILE COMMANDS

- `ls` = directory listing
- `ls -al` = formatted listing with hidden files
- `cd dir` = change directory to dir
- `pwd` = show current directory
- `mkdir dir` = create directory dir
- `rm file` = delete file
- `rm -r dir` = delete directory dir
- `rm -f file` = force remove file
- `rm -rf dir` = force remove directory
- `cp file1 file2` = copy file1 to file2
- `mv file1 file2` = rename file1 to file2
- `ln -s file link` = create symbolic link 'link' to file
- `touch file` = create or update file
- `cat > file` = place standard input into file
- `more file` = output the contents of the file
- `less file` = output the contents of the file
- `head file` = output first 10 lines of file
- `tail file` = output last 10 lines of file
- `tail -f file` = output contents of file as it grows

NETWORK

- `ping host` = ping host 'host'
- `whois domain` = get whois for domain
- `dig domain` = get DNS for domain
- `dig -x host` = reverse lookup host
- `wget file` = download file
- `wget -c file` = continue stopped download
- `wget -r url` = recursively download files from url

SSH

- `ssh user@host` = connect to host
- `ssh -p port user@host` = connect using port p
- `ssh -D port user@host` = connect & use bind port

SYSTEM INFO

- `date` = show current date/time
- `cal` = show this month's calendar
- `uptime` = show uptime
- `w` = display who is online
- `whoami` = who are you logged in as
- `uname -a` = show kernel config
- `cat /proc/cpuinfo` = cpu info
- `cat /proc/meminfo` = memory info
- `man command` = show manual for command
- `df` = show disk usage
- `du` = show directory space usage
- `du -sh` = human readable size in GB
- `free` = show memory & swap usage
- `whereis app` = show possible locations of app
- `which app` = show which app will be run by default

PROCESS MANAGEMENT

- `ps` = display currently active processes
- `ps aux` = ps with a lot of detail
- `kill pid` = kill process with pid 'pid'
- `killall proc` = kill all processes named proc
- `bg` = lists stopped/background jobs
- `fg` = bring most recent job to foreground
- `fg n` = brings job n to foreground

FILE PERMISSIONS

- `chmod octal file` = change permission of file (4:read / 2:write / 1:execute)
 Order: owner/group/world
 Eg: `chmod 755 file` = read write for owner, read execute for group/world

SEARCHING

- `grep pattern files` = search for pattern in files
- `grep -r pattern dir` = search recursively for pattern in dir
- `command | grep pattern` = search for pattern in the output of command
- `locate file` = find all instances of file

COMPRESSION

- `tar cf file.tar files` = tar files into file.tar
- `tar xf file.tar` = untar into current directory
- `tar tf file.tar` = show contents of archive

Follow @SecurityGuill on Twitter for more about Infosec / Cybersecurity

- Si no tienes absolutamente ni idea sobre el comando que debes usar, ejecuta `apropos -a <texto a buscar>` para dejar que sea el sistema el que te sugiera comandos que puedan hacer lo que estas buscando. Por ejemplo:

```
redondo@redondo-VirtualBox:~$ apropos -a create directory
docker-plugin-create (1) - Create a plugin from a rootfs and configuration. P...
hpmkdir (1) - create a directory on an HFS+ volume
mkdir (2) - create a directory
mkdirat (2) - create a directory
mkdtemp (3) - create a unique temporary directory
mkfontdir (1) - create an index of X font files in a directory
mklost+found (8) - create a lost+found directory on a mounted Linux secon...
mktemp (1) - create a temporary file or directory
pam_mkhome (8) - PAM module to create users home directory
update-info-dir (8) - update or create index file from all installed info fi...
```

Redondo, ¿Por qué entregas una máquina optimizada y una que no está optimizada?

En mi trabajo con sistemas *laC* (especialmente si interactúan con *VirtualBox*) he necesitado consultar la documentación de los proveedores de virtualización muy a fondo. Esto me ha permitido usar opciones "ocultas" (no disponibles en el GUI de *VirtualBox*) que dotan a las máquinas generadas de más rendimiento (entre un 15-25% de una creada vía GUI) y menor uso de RAM, especialmente si usar varias muy similares, al compartirse páginas de memoria entre máquinas muy parecidas (algo llamado *Page Fusion*).

Desgraciadamente, estas características producen problemas en plataformas MAC (*Page Fusion* no está soportado) y si también tienes instalado *Hyper-V* en *Windows* (la máquina se bloqueará al arrancar). Si es tu caso, la máquina no optimizada es idéntica, pero sin esas características especiales para no provocar incompatibilidades (a cambio, obviamente, de perder algo de rendimiento y eficiencia). Esto aumenta mucho las posibilidades que la máquina virtual de ejemplo pueda desplegarse en cualquier SO que soporte *VirtualBox*. **Te recomiendo usar la optimizada si te es posible.**



NOTA MUY IMPORTANTE: Si la máquina te arranca, pero la pantalla se queda en negro y sin respuesta, por favor usa la versión no optimizada.





Opciones de creación de la máquina virtual para el curso

Opción 1: Importación de la imagen .ova suministrada (fácil)

¿Por qué elegir este método?

Recomendado si: Tienes instalado *VirtualBox*, una conexión a *Internet* estable, no quieres hacer el módulo de *Vagrant*.

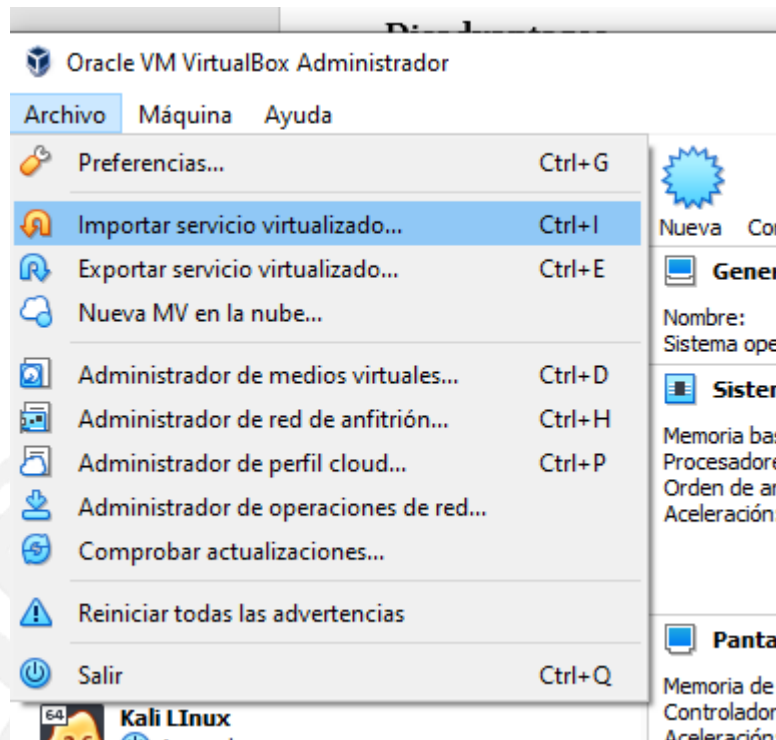
Ventajas

- MV preconfigurada lista para usar con todos los paquetes básicos instalados para empezar a hacer las actividades del curso de inmediato.
- Configuración de la MV en un solo paso (usuario: **r11user**, contraseña: **test123...** (tres puntos al final)).

Desventajas

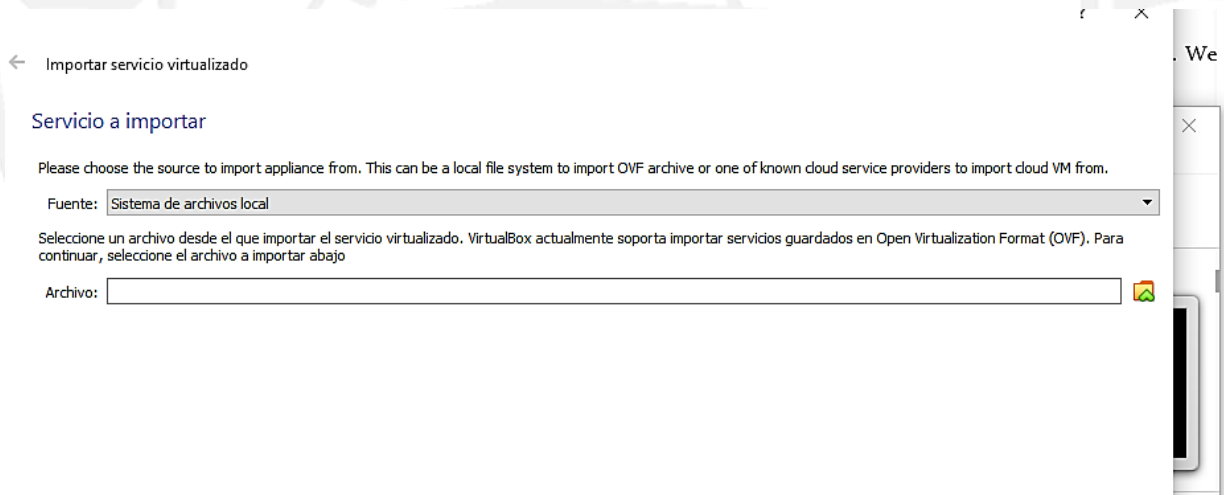
- Descarga grande (1,1 Gb) que puede no ser posible en ciertas situaciones. La máquina ha sido tratada para minimizar el espacio que ocupa no obstante.
- Tienes que crear una carpeta compartida con el *host* manualmente (ver la última sección de este documento). **Por favor, entra en la opción correspondiente antes de arrancar la máquina tras importar el .ova y asegúrate de que la carpeta compartida está en una ruta reconocida en tu disco.**
- Preparado para ejecutarse solo con *VirtualBox*. Es posible que otros sistemas de virtualización no puedan usar archivos **.ova**. Aquí te damos dos ejemplos de importación a otros sistemas, pero no se garantiza que funcionen porque estos procesos tienen fama de ser inestable:
 - **Importación de un archivo .ova con VMware:** <https://docs.vmware.com/es/VMware-Fusion/11/com.vmware.fusion.using.doc/GUID-275EF202-CF74-43BF-A9E9-351488E16030.html>
 - **Importar un .ova con Hyper-V:** <https://superuser.com/questions/1133256/convert-ova-to-vhd-for-usage-in-hyper-v> (vete a la Opción 3 para saber las características de la máquina virtual que debes crear).

Esta **máquina virtual preconstruida** puedes descargarla desde una URL que habrá en el campus virtual. La máquina es un **Ubuntu Server 18.04 LTS** basado en consola (con GUI XFCE4 **opcional**), con todo lo mencionado en la primera sección de este documento. Para importarla, descarga el archivo **.ova** y usa esta opción de *VirtualBox*:



Importación de una opción de archivo .ova

Elije el archivo **.ova** descargado, haz clic en *Siguiente* y sigue las instrucciones para importar la máquina virtual, que tardará un tiempo. **Se recomienda usar un SSD** para reducir drásticamente los tiempos de importación, el arranque y la capacidad de respuesta general, pero no es algo necesario.



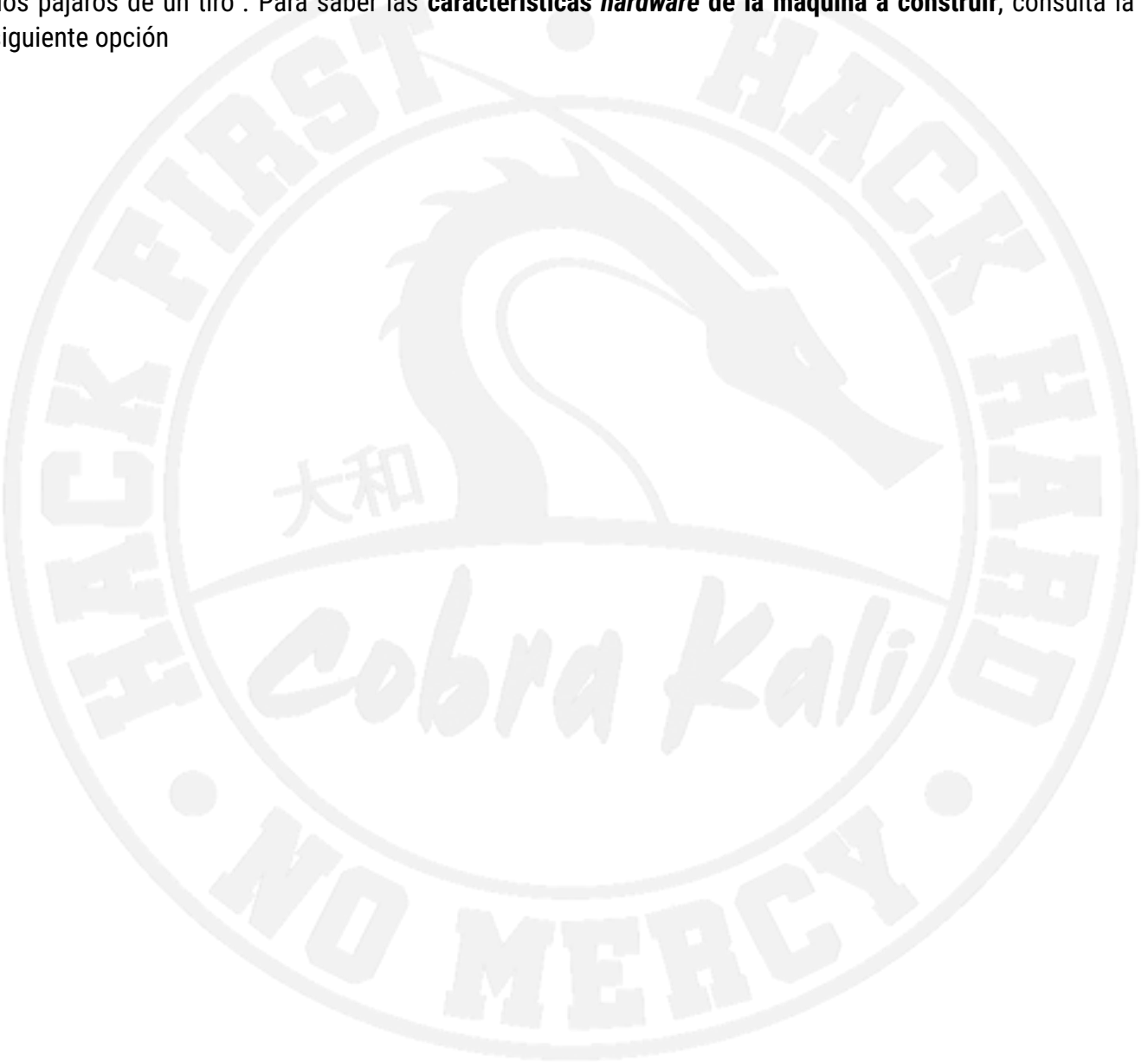
Selección del filtro de archivos .ova

Otras opciones para administrar la máquina virtual importada se detallarán en la siguiente sección.



Opción 2: Usar *Vagrant* para construir automáticamente una máquina virtual (fácil-medio)

Recomendado si: Tienes instalado *VirtualBox*, *Hyper-V* o *VMware Desktop* y **quieres hacer el módulo de *Vagrant***. Es posible hacer los ejercicios del bloque de *Vagrant* pensando en hacerse una máquina virtual con la que trabajar en otros módulos. Las características de la máquina (optimizaciones, carpetas compartidas...) que podrás ir metiendo serán más avanzadas cuantos más ejercicios te plantees hacer. De esta forma tu trabajo con *Vagrant* puede servirte de utilidad para el resto de los módulos, “matando dos pájaros de un tiro”. Para saber las **características hardware de la máquina a construir**, consulta la siguiente opción





Opción 3: Creación de tu propia máquina virtual (difícil, no se recomienda)

Recomendado si: Tienes instalado cualquier otro hipervisor de los mencionados, quieres tener el control total del proceso de creación de la MV, tienes experiencia manejando MVs.

¿Por qué elegir este método?

Ventajas

- Tienes el control de todo lo que creas
- Puedes utilizar cualquier sistema de virtualización en cualquier sistema operativo. Es recomendable (pero no obligatorio) utilizar las mismas credenciales que la VM preconfigurada (usuario: **r11user**, contraseña: **test123...**) para evitar confusiones.

Desventajas

- Lleva mucho más tiempo completar la configuración.
- Requiere conocimientos técnicos y debes seguir los pasos de esta sección. No es automatizable.

Configuración de una plataforma de virtualización para instalar un servidor Linux

Información general sobre el hardware virtual

La máquina virtual preconstruida tiene las siguientes características que deberías poder reproducir en tu sistema de virtualización y sistema operativo (**asegúrate de tener suficiente espacio disponible**). Damos aquí las características de la máquina virtual con *VirtualBox* como ejemplo. Debes buscar las características equivalentes en tu sistema de virtualización

- **2 Gb de RAM** (o más si puedes permitirte 😊)
- **2 núcleos de CPU** (1 también podría ser posible, pero a un coste de rendimiento grande)
- **64-128Mb de memoria de vídeo** (máximo) con aceleración 3D habilitada si vas a usar GUI. Si no, basta con 16Mb.
- **80Gb de HD** con almacenamiento dinámico. Usar un SSD, si está disponible, es recomendable pero no obligatorio. Se recomienda marcar la opción "*Usar la caché de E/S del host*".
- No se requiere **audio** (tampoco pasa nada porque lo dejes activo).
- Una tarjeta de **red NAT** (la primera), configurada para obtener su IP a través de DHCP
- Una tarjeta de **red host-only** (la segunda), con la IP estática **192.168.56.33** (esto es **opcional**, y solo si planeas usar clientes SSH como *MobaXTerm* para interactuar con la máquina y no quieres usar la redirección de puertos como se explica al principio del documento por alguna razón).

El resto de la configuración se puede dejar en sus valores predeterminados. Estas opciones se explicarán en las siguientes secciones utilizando *VirtualBox* <https://www.virtualbox.org/> como software de virtualización. Ten en cuenta que *VirtualBox* está disponible para los principales sistemas operativos de forma gratuita, por lo que no deberías tener ningún problema para trabajar con él. También ten en cuenta que si tienes *Hyper-V* habilitado en una máquina *Windows*, *VirtualBox* no podrá funcionar correctamente,



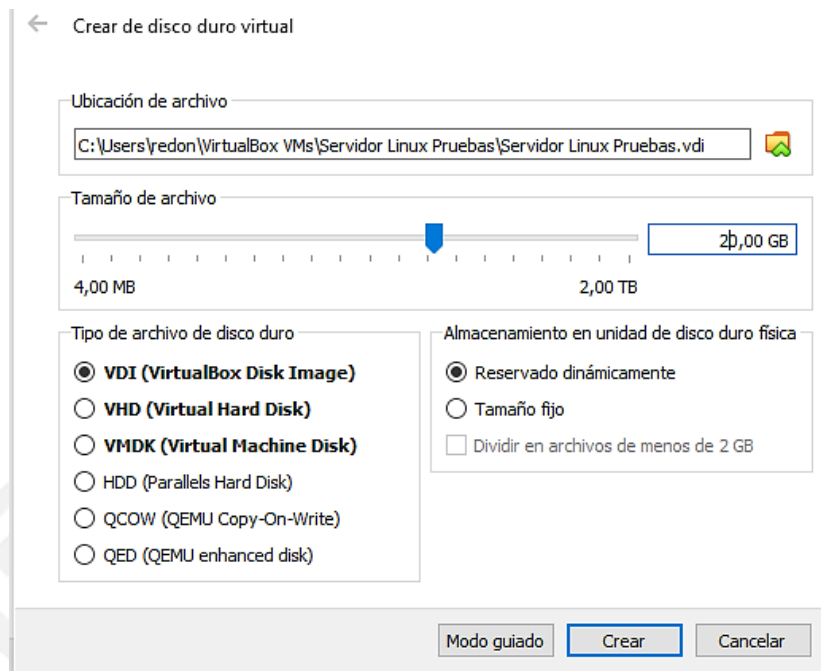
ya que ambos presentan problemas de compatibilidad si están instalados en un mismo PC. Es posible arreglar dichos problemas, como veremos en la sección de solución de problemas posterior.

Creación de máquinas virtuales

El primer paso es crear un nuevo perfil de máquina adecuado para el sistema operativo que vamos a instalar, que en nuestro caso será un servidor **Ubuntu 18.04 LTS de 64 bits** que puedes descargar desde aquí: <https://ubuntu.com/download/server>

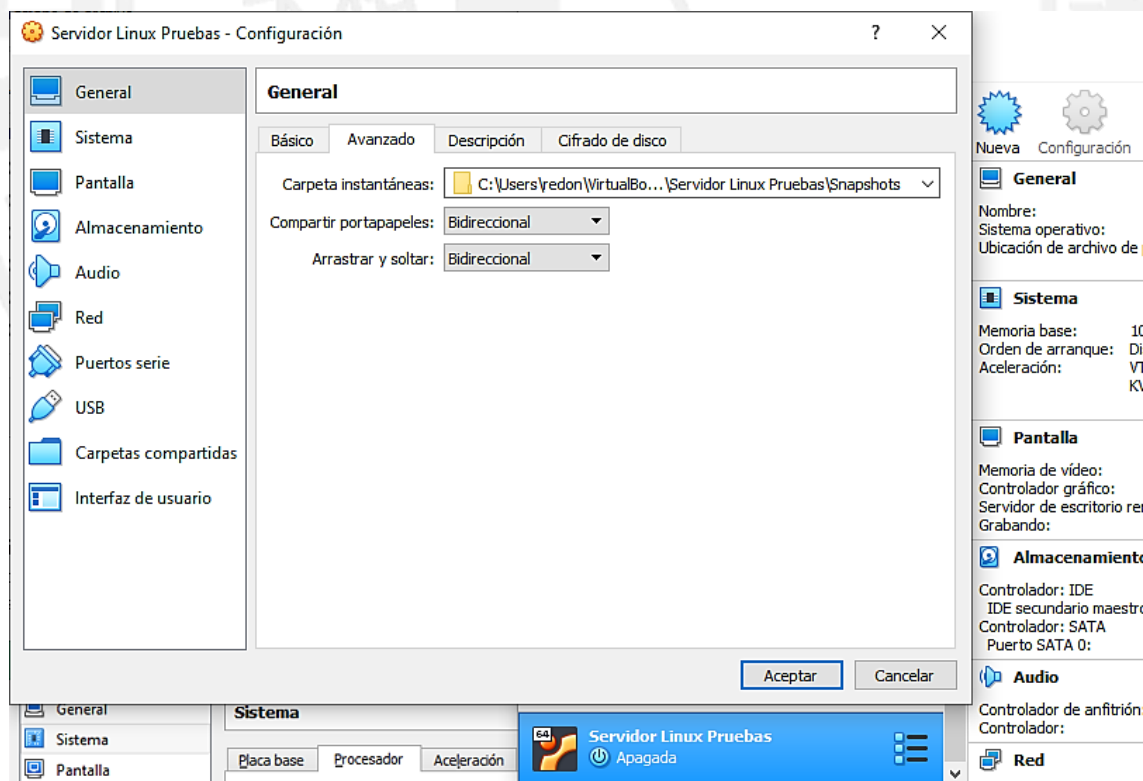
Tipo de máquina virtual a crear y RAM

Procedemos ahora a darle la RAM y espacio en el HD recomendados anteriormente. El formato puede ser *nativo de VirtualBox* (VDI) o *VMDK* para portabilidad con *VMWare*. Como hemos dicho, para ser más eficiente lo crearemos **dinámico**.



Disco duro de la máquina virtual

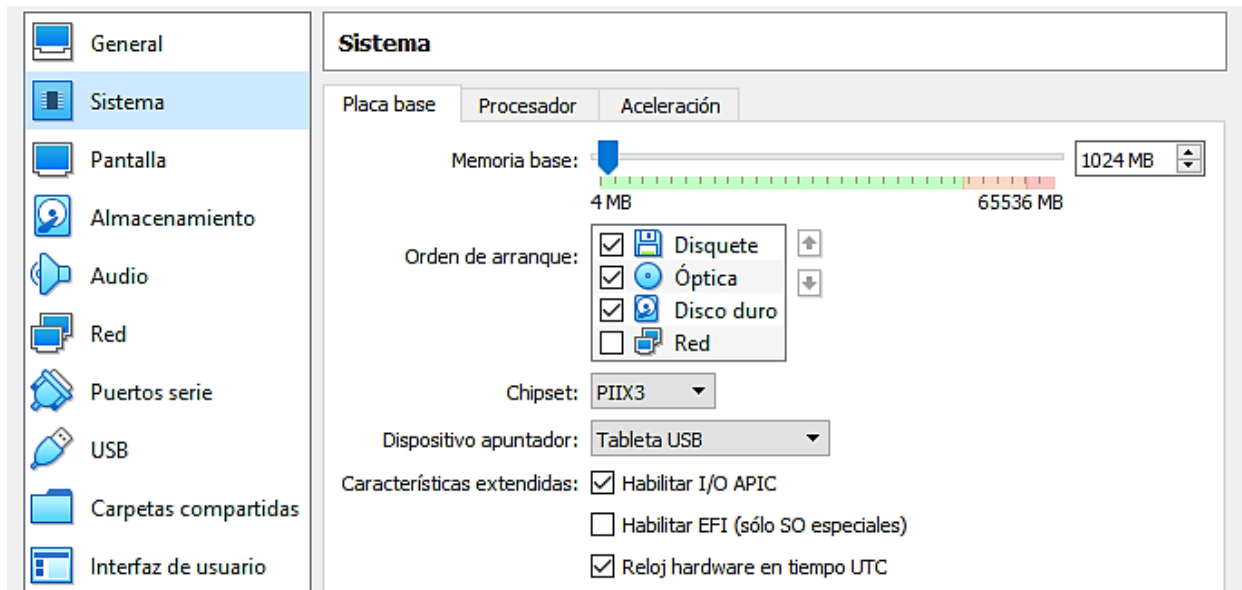
Una vez hecho esto, se creará la nueva plantilla para la máquina, y podremos mejorar su configuración editándola luego. El primer paso es ir a la categoría *General*, donde podemos habilitar el *Portapapeles* y la capacidad de arrastrar y soltar entre el host y la máquina virtual y viceversa (**Bidireccional**). Sin embargo, **esto solo será efectivo si tenemos una GUI**.



Categoría general de un VirtualBox

Más tarde, en la categoría *Sistema* habilitamos las opciones de reloj UTC y IO/APIC. Si bien EFI es conveniente habilitarlo para los sistemas operativos *Windows* o *Linux* modernos, hemos detectado que si

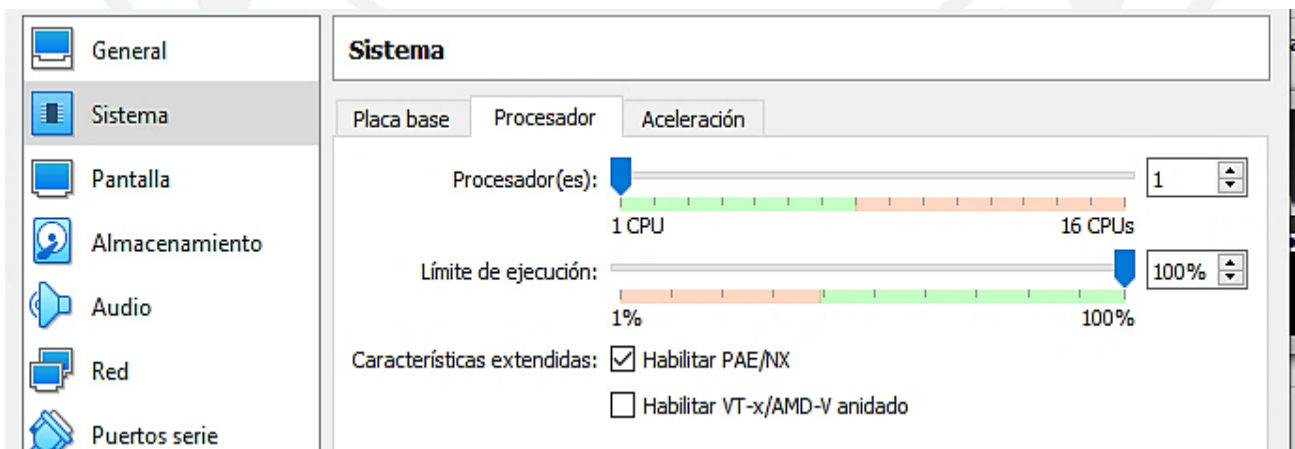
la máquina se exporta a **.ova**, durante la reimportación puede sufrir problemas de arranque, por lo que elegimos no habilitarla para evitarlos. Cabe señalar que, una vez instalado el sistema, esta opción no se puede cambiar: **un sistema instalado sin modo UEFI no arrancará si está activado y viceversa**.



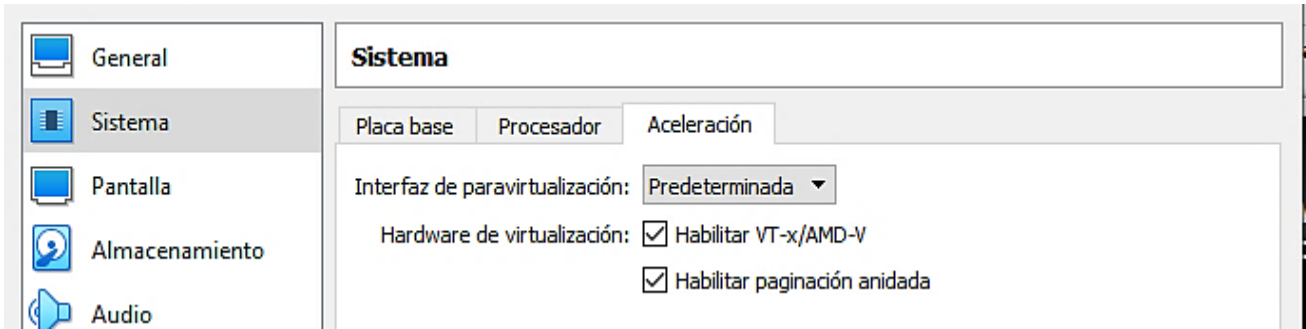
Sistema de categorías de una máquina VirtualBox

Luego asignaremos 1 o 2 CPUs y procederemos a activar PAE/NX (pestaña "Procesador") y los conjuntos de instrucciones de la CPU VT-X o AMD-V y las funciones de paginación anidadas (si están disponibles en nuestro hardware base) para un mejor rendimiento de la virtualización (pestaña "Aceleración"). Es posible que VT-x /AMD-V anidado no esté disponible dependiendo de la CPU, pero no es necesario para esta asignatura. Esto permitiría que haya máquinas virtuales ejecutándose dentro de nuestra máquina virtual (con el coste de rendimiento no trivial que esto conlleva 😊 😊).

La interfaz de paravirtualización no necesita cambiarse, a menos que tengamos *Hyper-V* instalado en un host de MV Windows. **VirtualBox normalmente no puede funcionar si está instalado Hyper-V en la misma máquina**, pero cambiando el interfaz de paravirtualización a *Hyper-V* debería arreglarlo (con un coste de rendimiento sustancial):

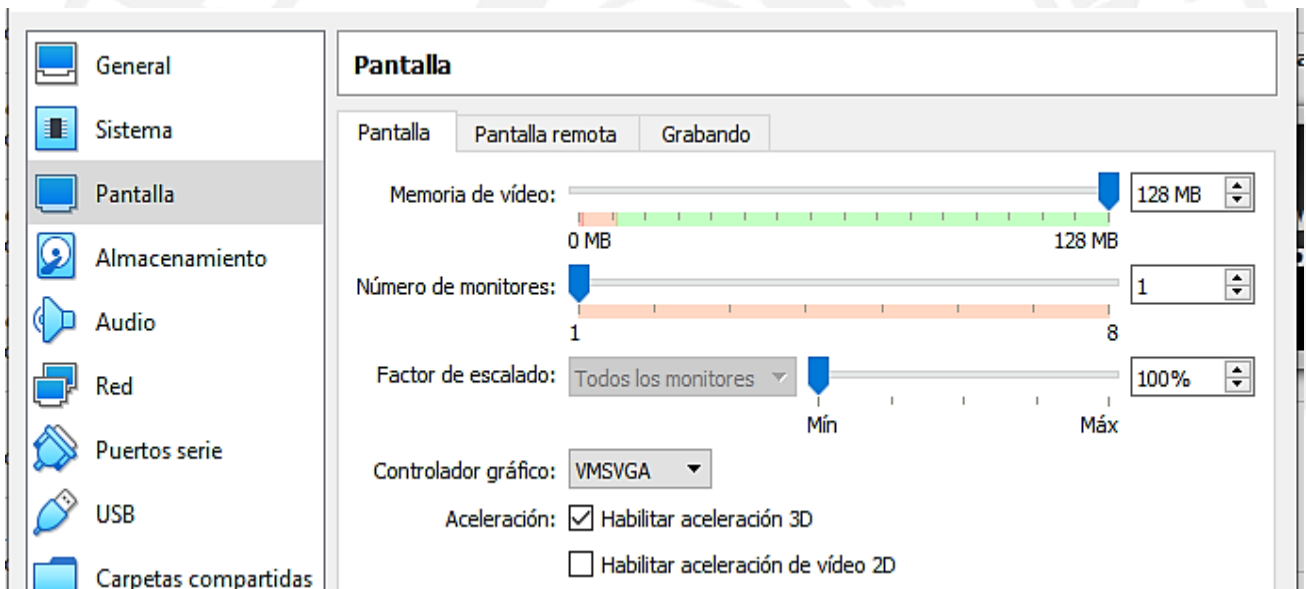


N.º de CPU de la máquina a crear



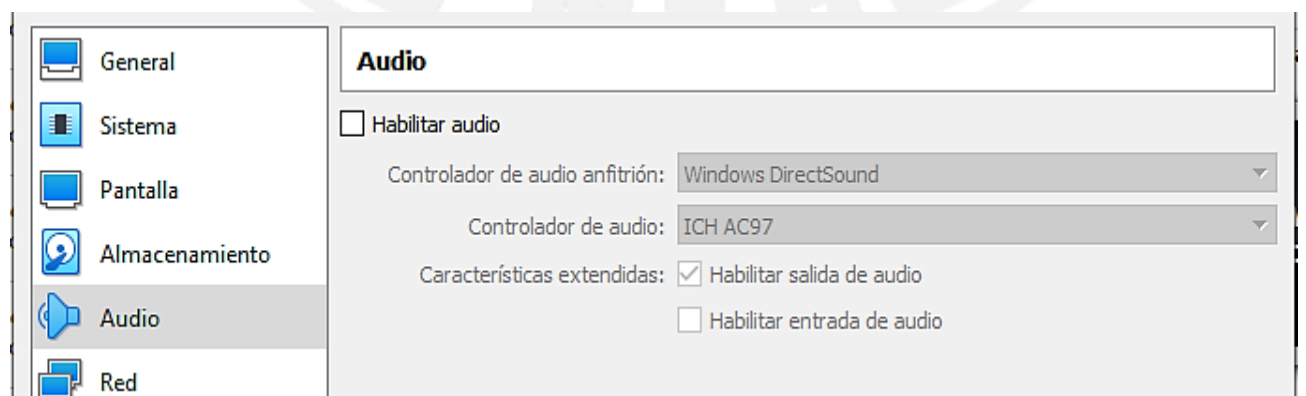
Aceleración para la máquina virtual

En cuanto a la selección de memoria de video y capacidades de aceleración 2D y 3D, todo depende de si la máquina a instalar tendrá una GUI o no. En un escenario real, al ser un servidor orientado a usar la menor cantidad de recursos posible, no debería tenerlo, pero en nuestra asignatura sí contaremos con uno. Cabe señalar que actualmente *VirtualBox* no permite la aceleración 2D en plataformas *Linux* (se admite 3D). Sin embargo, como planeamos instalar una GUI más adelante, elegimos la memoria de video máxima disponible (128Mb). Como controlador gráfico dejamos la configuración por defecto:



Memoria de vídeo asignada a la máquina

El perfil de la máquina creado no usará *Audio*, por lo que optamos por deshabilitarlo y, por lo tanto, no instalar controladores o servicios adicionales.

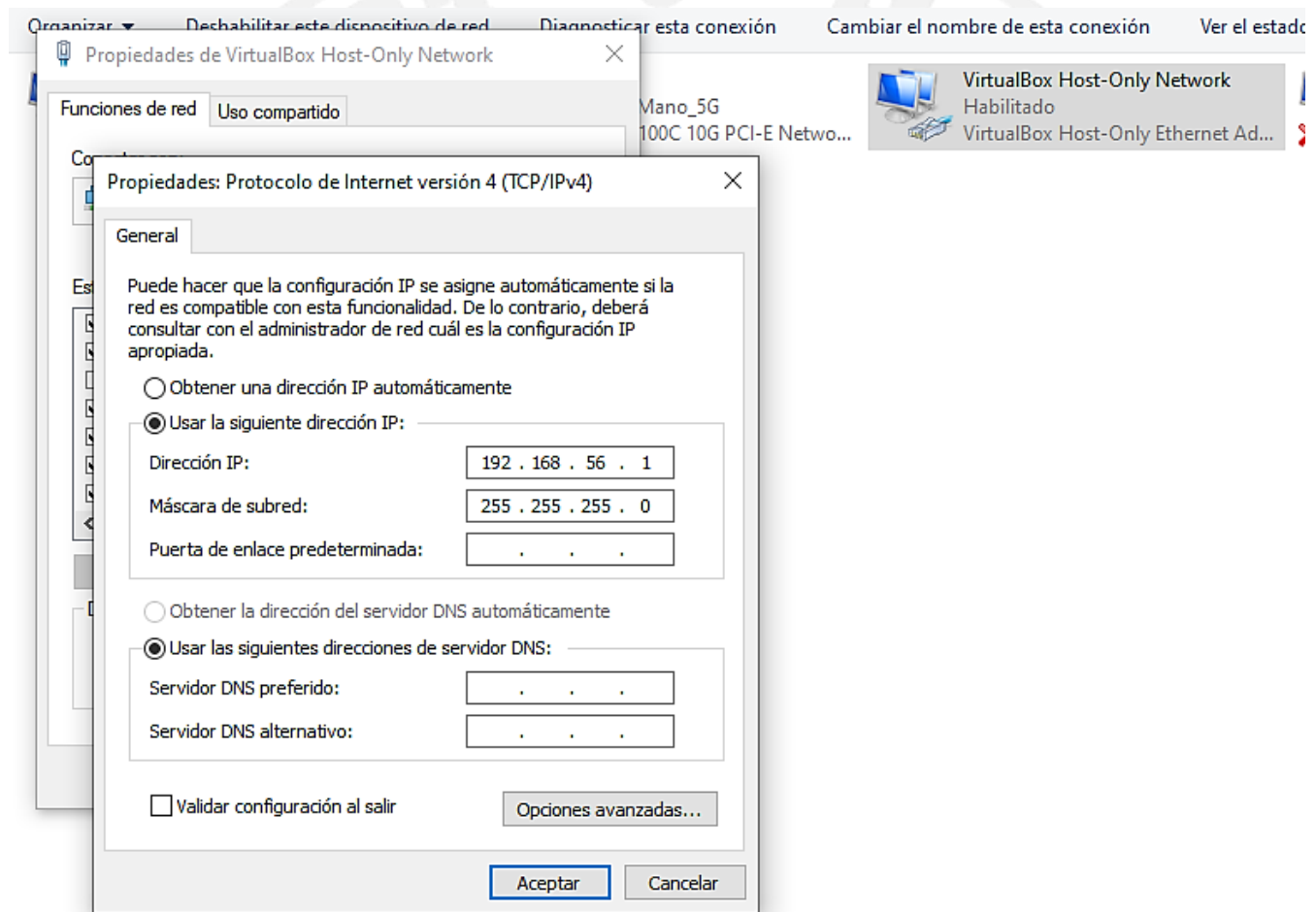


Configuración de audio

Redes VirtualBox

Las redes de máquinas virtuales son más complejas y requieren atención adicional. Los principales tipos de redes que puede tener una máquina *VirtualBox* son:

- **Host-only:** Permite que nuestra máquina *host* se comunique directamente con la máquina virtual utilizando una red local creada para este propósito. En este caso, tu PC debe tener una IP estática asignada a un adaptador virtual de red que se crea al instalar *VirtualBox*. Suele ser la que se ve en esta imagen:



Por lo tanto, bastaría con configurar el adaptador *host-only* en la máquina virtual, de modo que tuviera una IP fija en esa subred (56 o la utilizada en esta instalación) para permitir la conectividad. Este es el método que suele dar menos problemas: solo puedes ver tus máquinas locales y la IP nunca cambia. En los laboratorios de la escuela, **esto puede no ser factible** porque *VirtualBox* podría haberse instalado sin una tarjeta de red que admita esta función, y debería crearse por un administrador local en cada máquina.

- **"Bridge":** Esto suele usarse si tu PC obtiene su IP a través de DHCP. El efecto es que la máquina virtual creada también lo hará, es decir, se le asignará automáticamente una IP en la misma subred que el *host*, siendo accesible desde el *host*, y desde fuera del mismo, utilizándola. La desventaja es que no es una IP fija y la máquina virtual es visible para los agentes externos. Sin embargo, es una opción muy viable en instalaciones típicas de redes domésticas, donde un *router* o módem por



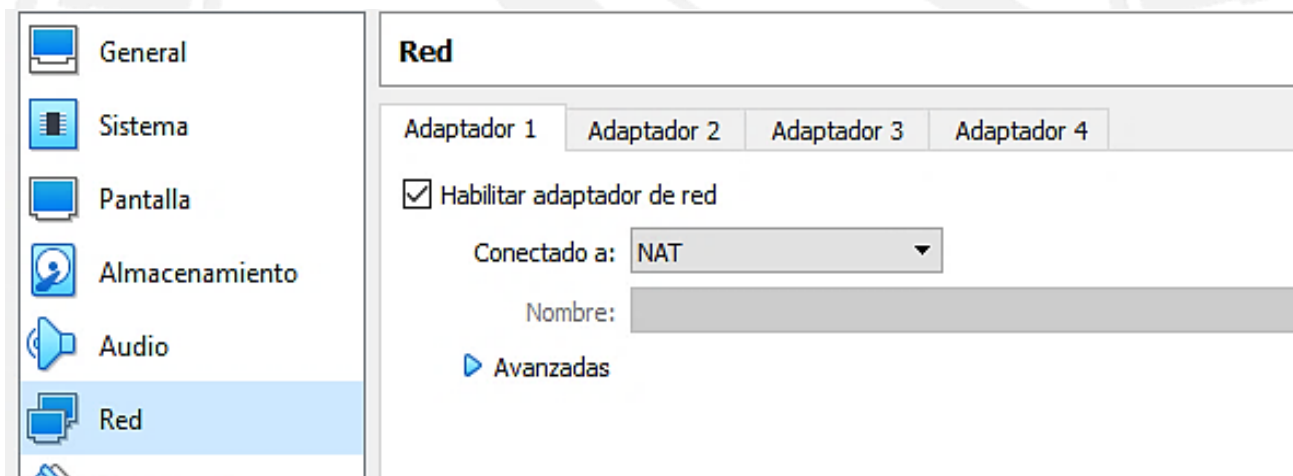
cable sirve a máquinas que usan DHCP y enruta todo el tráfico por una sola IP externa. **El uso de este modo en laboratorios de la escuela puede causar problemas**, ya que el grupo de IPs disponible por DHCP para un laboratorio puede agotarse si todos usasen una red puente. Si no se usa DHCP, habría que asignar manualmente a esta interfaz una IP en la misma red que el host.

- **Red interna privada:** Si tienes dos máquinas virtuales, puedes hacer que una sea el cliente de la otra asignando a ambas una nueva interfaz de red interna **con el mismo nombre**. Hecho esto, solo debes asignar a cada interfaz una IP fija diferente en la misma subred con la misma máscara de red para que puedan comunicarse entre sí. La desventaja es que necesita ambas máquinas virtuales arrancadas para hacer pruebas, lo que significa un mayor consumo de recursos.
- **NAT:** Esta interfaz se encarga de proporcionar **salida a Internet** a través de la conexión del host de forma automática. Debe estar activo en la primera tarjeta para evitar problemas y no preocuparse más de ella.

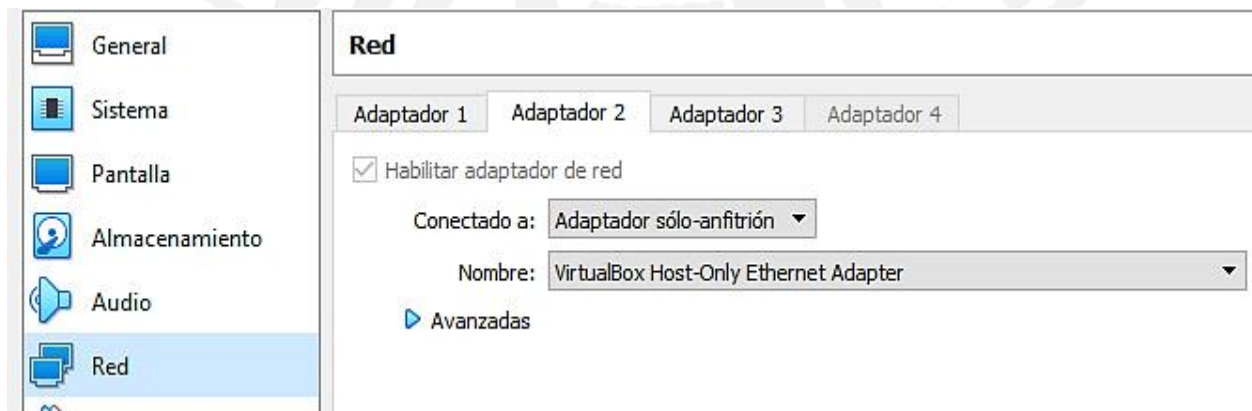
La máquina *Linux* a crear debería tener dos tarjetas de red:

- **Adaptador 1:** Por *NAT*. Permite la conexión a Internet a través del host y será el utilizado para las actualizaciones y descargas de software desde repositorios de *Ubuntu* y servidores de terceros.
- **Adaptador 2:** *host-only*. El propósito de esta red es que el *host* pueda ser un cliente de la máquina creada (**opcional**).

En las siguientes imágenes vemos la configuración de los dos adaptadores:



Adaptador de red 1



Adaptador de red 2

USB

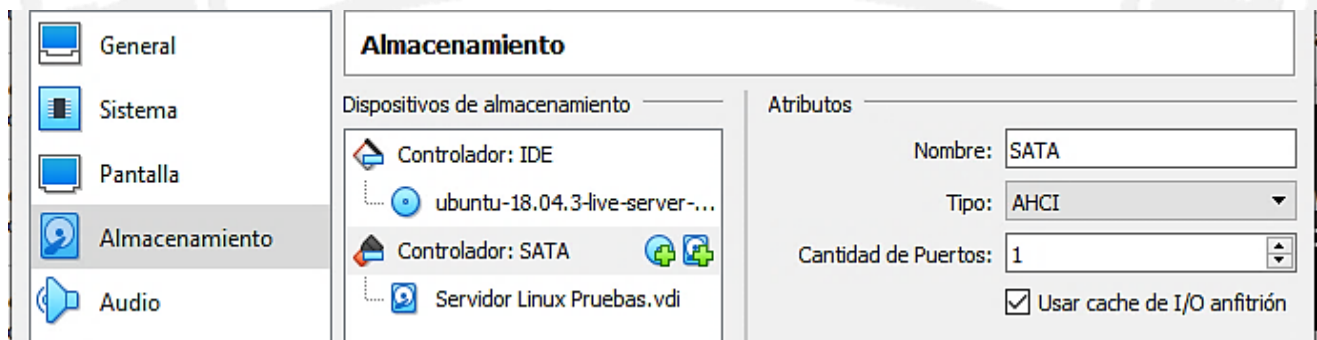
Solo debe activarse la compatibilidad con USB 1.1 en las máquinas porque cualquier revisión posterior de USB requiere la instalación del *VirtualBox Extension Pack*, que puede dar problemas de cara a reimportar la máquina en otro PC. Esto limita el uso de USB al teclado y al ratón básicamente, ya que la conexión de *pendrives* o discos duros a la máquina virtual sería muy lenta. Si ejecutas esto en casa o no usas *VirtualBox*, puedes habilitar el mejor soporte USB que permita tu sistema de virtualización.



Compatibilidad con máquinas virtuales USB

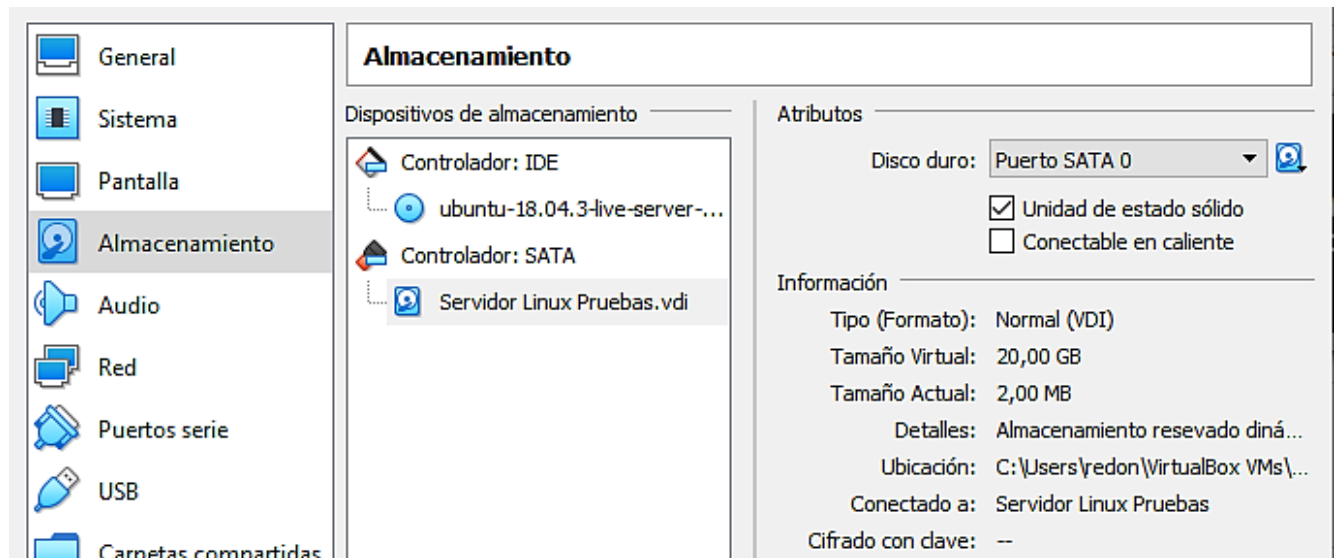
Almacenamiento y selección de la ISO del sistema operativo

En lo relativo a almacenamiento, para una virtualización óptima podemos habilitar la caché de E/S del host para el controlador de disco duro SATA



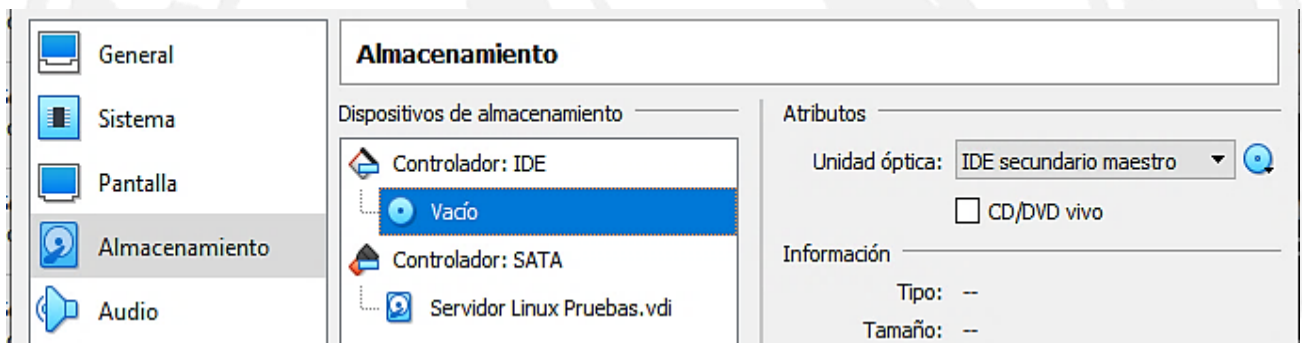
Configuración del controlador SATA

Además, si la máquina se guarda en un SSD, podemos decirle a *VirtualBox* que realice las optimizaciones necesarias. No debes activar que la unidad sea conectable en caliente a menos que realmente lo sea (es decir, es un disco duro secundario que está en un lápiz USB o algo similar):

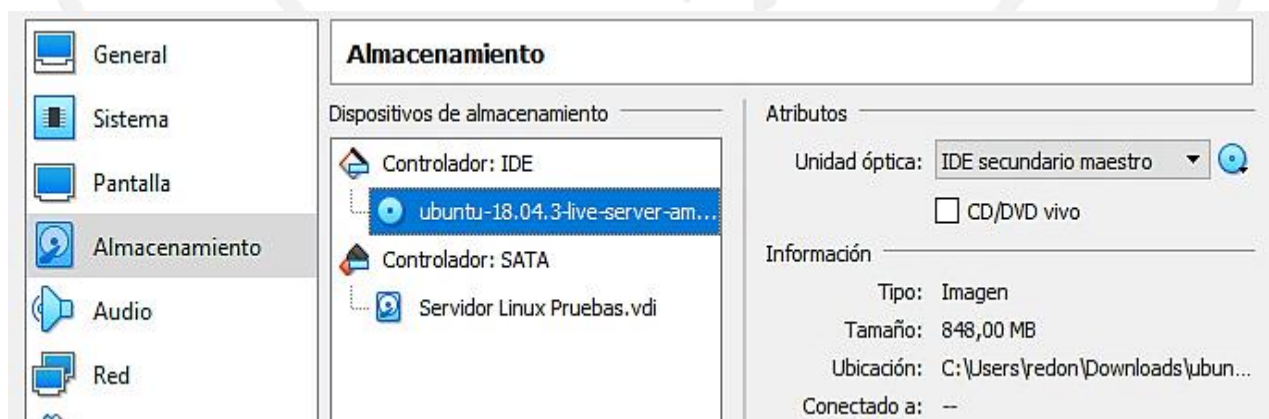


Configuración del disco

Por último, ya hemos configurado la máquina para iniciar la instalación, y ahora solo debemos seleccionar la ISO del sistema operativo que hemos descargado. Procedemos a asignarla a la unidad de CD/DVD correspondiente.



Asignación de la ISO de instalación (I)



Asignación de la instalación ISO (II)

Ahora podemos iniciar la máquina y la instalación

Instalar un servidor Ubuntu Linux

Instalación del sistema operativo

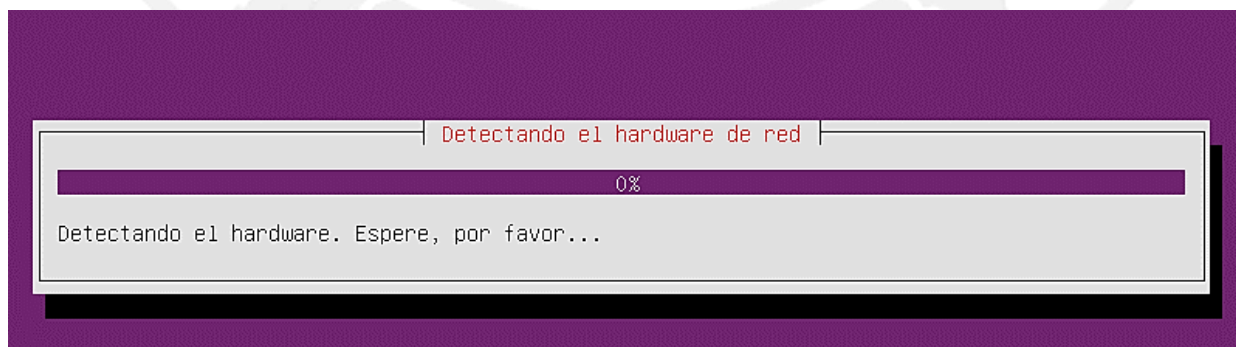
Una vez configurada la máquina virtual, es el momento de instalar el *servidor Ubuntu*. Las pantallas de instalación **pueden haber cambiado ligeramente en versiones sucesivas**, pero son esencialmente las mismas que las enumeradas aquí. **Si necesitas una guía más reciente, puedes ver la instalación de una versión *Ubuntu 22* sobre *VirtualBox 7* en esta publicación:** https://www.researchgate.net/publication/366445532_Guia_Paso_a_Paso_para_la_Creacion_de_Maquinas_Virtuales_Caso_de_Ejemplo_con_Ubuntu_Linux_2204_y_VirtualBox_7

En cualquier caso, debemos iniciar la instalación utilizando la opción de instalación simple de *Ubuntu*:



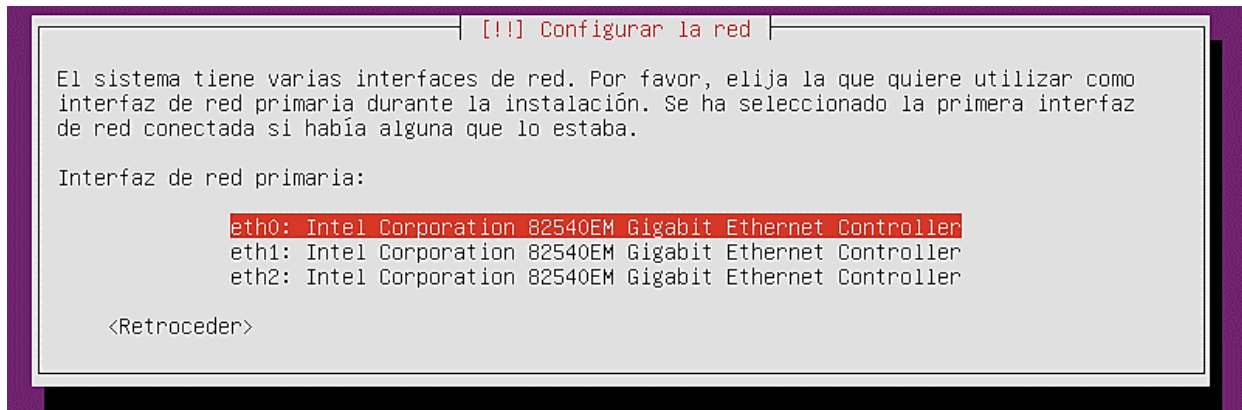
Inicio de la instalación de Ubuntu

El descubrimiento del hardware es un proceso que puede tardar mucho tiempo, pero con el hardware virtual de *VirtualBox* no debería ser un problema:



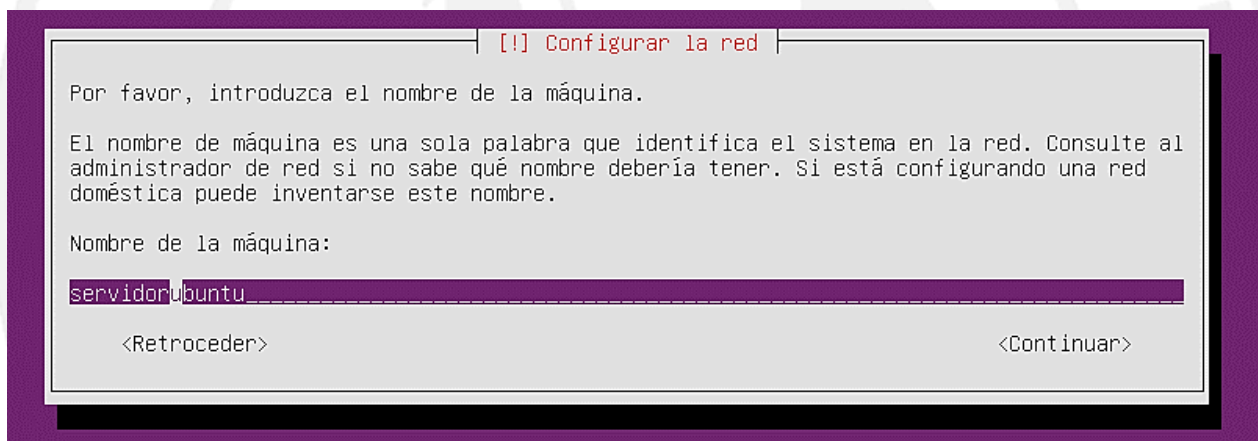
Detección de hardware virtual

Seleccionar la tarjeta de red principal es importante porque determinará la que el sistema utiliza para conectarse a Internet para instalaciones / actualizaciones. En nuestro caso es la primera, como vimos anteriormente:



Configuración de red

Dado que esta máquina va a ser clonada, el **nombre** que le demos es un poco irrelevante en este punto, porque cada clon que hagamos debe tener el suyo propio.



Asignación de un nombre a la máquina

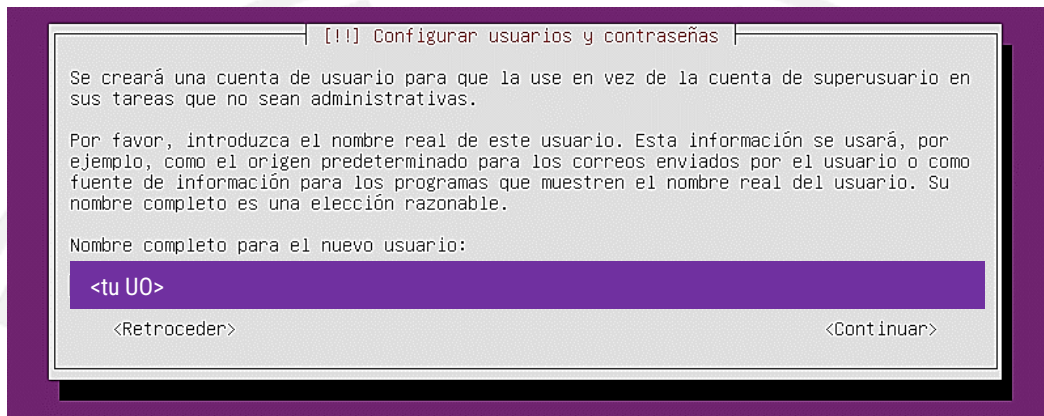
Crear un usuario inicial es un punto crucial en la instalación de un sistema operativo. Hay varias razones:

- No deben hacerse operaciones de administración y uso "normal" de la máquina desde la cuenta **root**. Esto se debe a que esta cuenta tiene muchos privilegios y puede hacer prácticamente cualquier cosa. Cualquier acción que hagamos como **root** puede ser muy perjudicial si no lo hacemos correctamente, por lo que *Ubuntu* no nos deja acceder directamente con este usuario a través del login estándar.
- Como obviamente necesitamos un usuario para trabajar, la instalación pide crear uno, que en este ejemplo se llamará "**ssiuser**", pero en tu caso usa tu UO.
- Este usuario es "especial", ya que pertenece al grupo de los *sudoers*. ¿Qué significa esto? que aunque no accedamos como **root** directamente, seguimos necesitando privilegios elevados para realizar instalaciones, actualizaciones, etc. según sea necesario. Esto no se puede hacer con un usuario "normal" (como el que creamos), pero si ese usuario es *sudoer*, puede usar el comando **sudo <comando que queremos ejecutar>** para ejecutar, después de escribir la propia contraseña



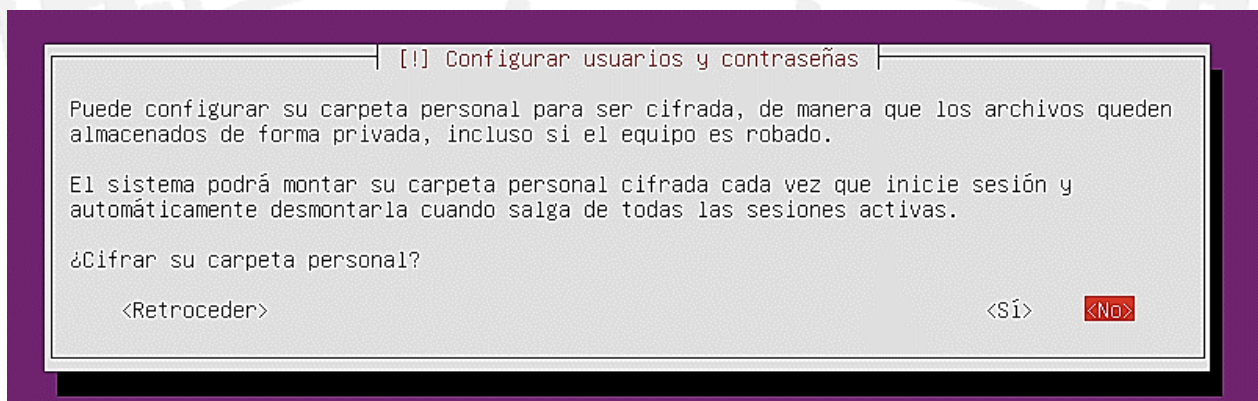
del usuario `sudoer`, ese comando con privilegios de `root` y por lo tanto puede hacer la administración / instalación / etc. El usuario creado en la instalación pertenece por defecto a ese grupo y es el que usaremos para gestionar en lugar de `root`.

- Los futuros usuarios que se pueden crear en el sistema no son `sudoers` por defecto.
- De esta manera, solo un usuario tiene privilegios elevados y los usa cuando invoca explícitamente un comando de forma interactiva. Esto evita que el software potencialmente dañino haga mucho daño al sistema, ya que los privilegios de un usuario normal están lejos de los de `root`. Esto no sería válido, eso sí, si ejecutamos un software dañino con `sudo`...



Creación de un usuario predeterminado

El uso del cifrado de archivos personales es opcional, pero conveniente por razones de seguridad física. Sin embargo, en esta asignatura no lo usaremos, pero es bueno saber que se puede activar aquí para cuando hablemos de esta opción en teoría.



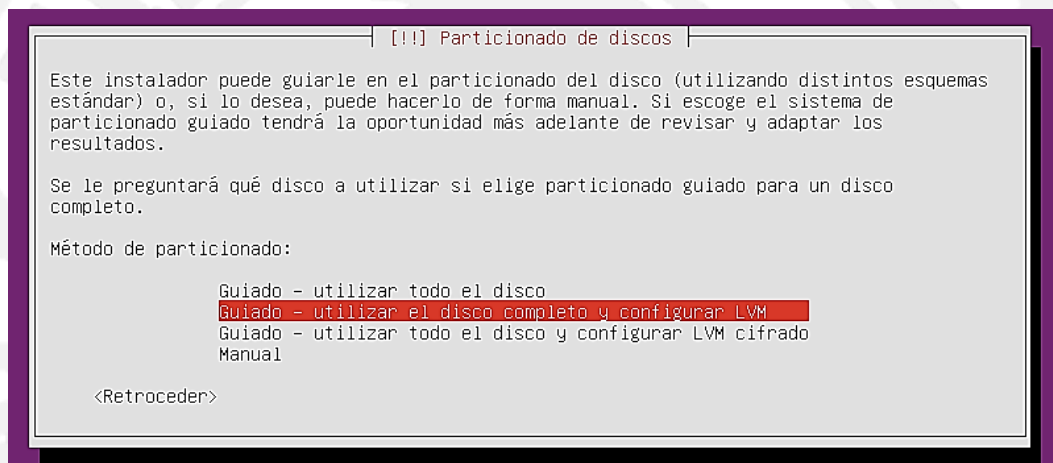
Cifrado de la carpeta personal de los usuarios

La partición de discos es un tema delicado que incluso puede requerir planificación previa, especialmente en sistemas multidisco. El particionamiento debe adaptarse a las necesidades del servidor. Si disponemos de varios discos de diferentes tecnologías y velocidades, podemos decidir iniciar el sistema de archivos para que cada disco quede ocupado por una parte concreta del sistema de almacenamiento (archivos de usuario, archivos web, *subidas*...). También es común tener una partición `/home` con archivos de usuario independientes de los otros archivos. En cualquier caso, no insistiremos en este aspecto y utilizaremos el modelo sencillo propuesto por el instalador **"Usar el disco completo"**. Otras opciones de partición se



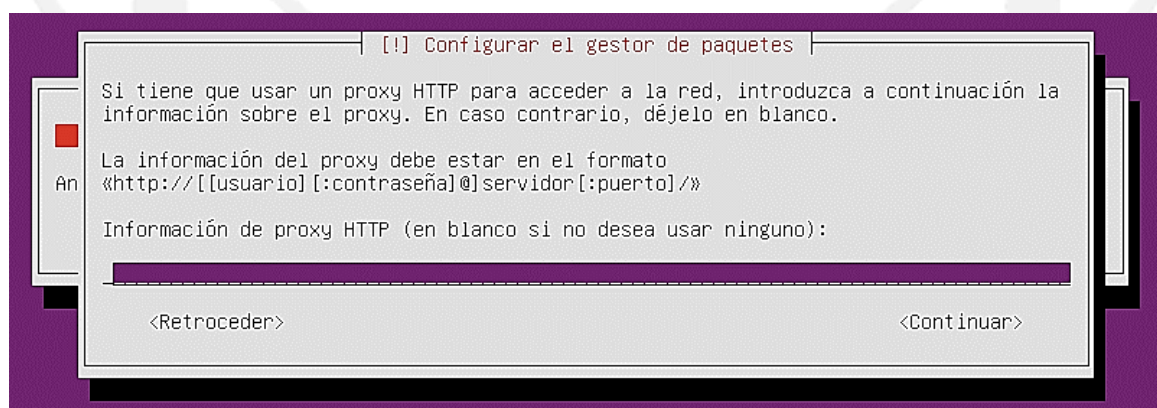
pueden encontrar en: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/s2-diskpartrecommend-x86.html

Sin embargo, es bueno hablar de la otra opción: **"Usar el disco completo y configurar LVM"**. LVM significa *Logical Volume Manager* y es una utilidad que permite crear volúmenes lógicos a partir de discos duros físicos. Por ejemplo, si tenemos dos discos duros físicos y queremos que el sistema operativo los vea como una sola partición, LVM se puede utilizar para "unir" lógicamente ambos, y así parezca tener instalado un único disco duro cuyo tamaño es la suma de ambos. Por lo tanto, es una forma muy adecuada de administrar los discos de un PC agrupando *volúmenes físicos* (PV) en grupos virtuales de discos (*grupos de volúmenes*, VG) para que luego puedas crear particiones o *volúmenes lógicos* (LV). A efectos prácticos, si tienes varios discos duros físicos que quieres unir en un solo espacio, es conveniente utilizar LVM. Para un solo disco físico no es necesario, pero si es conveniente si el espacio del que disponemos resulta ser muy pequeño y queremos ampliarlo con nuevos discos duros físicos "en caliente". Podemos encontrar más información sobre este último aspecto y el uso de LVM encriptado en el siguiente enlace: <http://diegosamuel.blogspot.com.es/2014/01/particiones-con-lvm-y-cifrado.html>.



Particionamiento de disco

Después de configurar el particionamiento del disco duro, debemos especificar si queremos utilizar un **proxy HTTP** para conectarnos a Internet. En nuestro caso no será necesario:

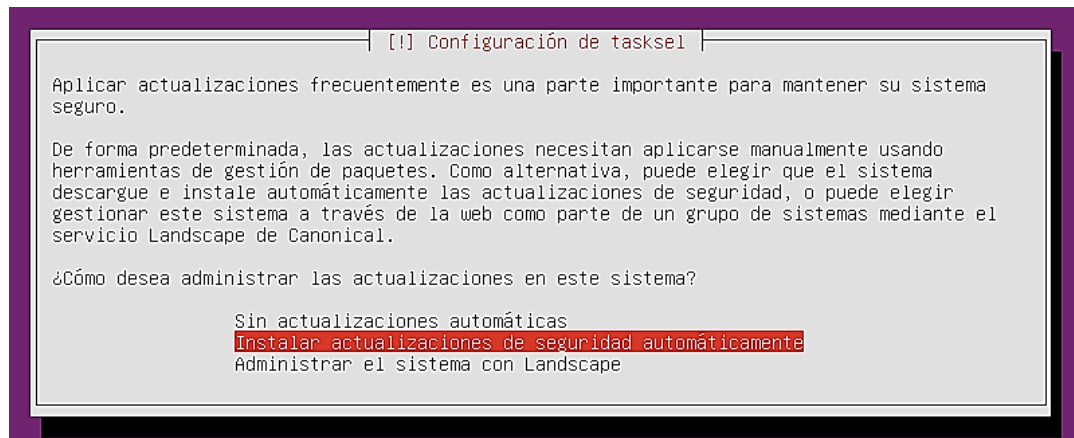


Asignación de proxy de red

Cuando se trata de **actualizaciones**, mantener el sistema operativo actualizado es muy importante para la seguridad de la máquina, por lo que habilitaremos la instalación automática de actualizaciones de

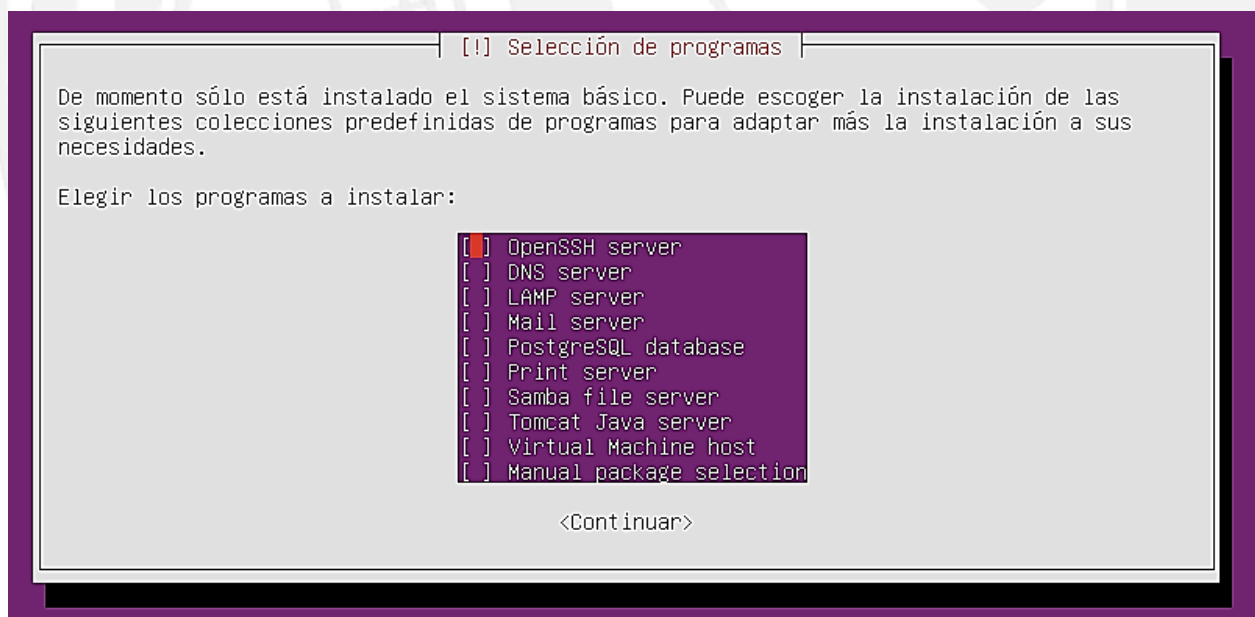


seguridad. Sin embargo, es recomendable acostumbrarse a actualizar el sistema operativo manualmente nosotros mismos de forma regular por si acaso.



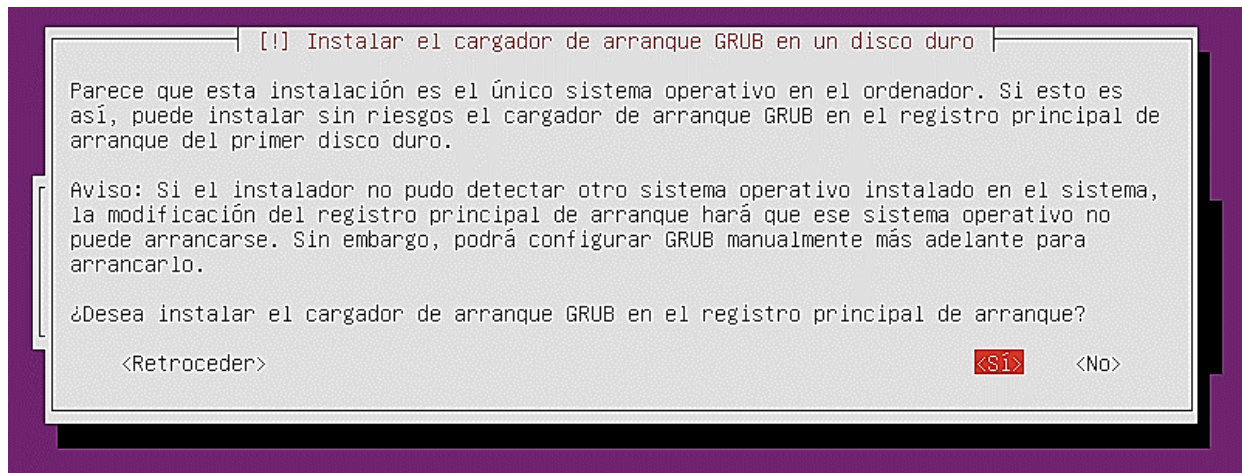
Administración de actualizaciones de seguridad

Cuando casi ha finalizado la instalación, el instalador nos pregunta si queremos realizar **instalaciones de grupos de paquetes** que configuren el servidor con un perfil de uso determinado (servidor DNS, servidor SSH, etc.). Dado que estamos instalando una máquina base, en este caso lo mejor es **no instalar nada por defecto** y dejar el sistema lo más simple posible (**NOTA:** Esta pantalla suele mostrar *Docker* como una opción de instalación. Esta versión no es la última pero debería ser suficiente para la asignatura. Puedes instalar *Docker* ahora y omitir su instalación más tarde).



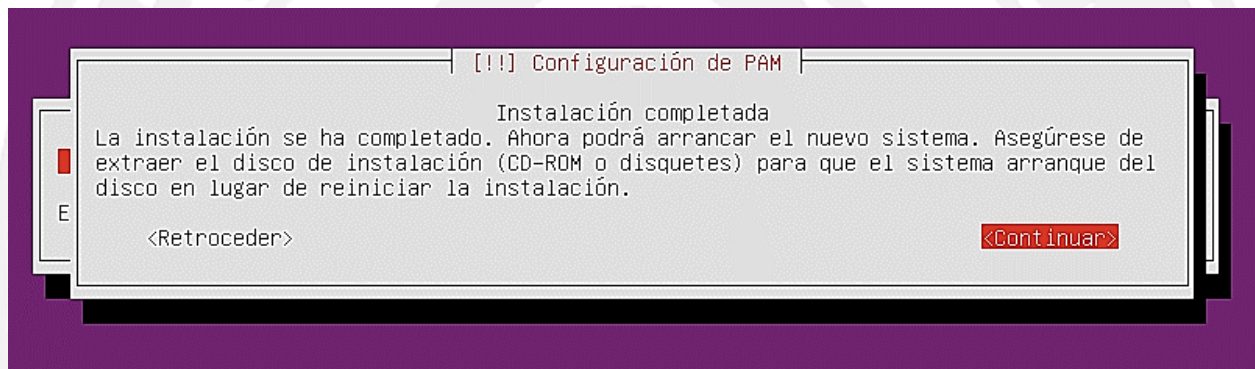
Selección de un perfil de instalación de paquetes

Tras la elección de las opciones de instalación, como este va a ser el único sistema operativo de la máquina y solo tenemos un HD, podemos **instalar GRUB** sin problemas.



Fin del proceso de configuración de la instalación

Luego procedemos a la instalación automática hasta que llegamos al mensaje de finalización y reinicio. **Debes quitar la ISO del sistema operativo de la unidad virtual de la máquina antes de reiniciar.**



Fin de la instalación y reinicio

Una vez hecho esto, ya podemos empezar a trabajar con el *Linux* instalado, haciendo uso del único usuario que hemos creado para él:

```
Ubuntu 18.04.5 LTS ssiuser tty1
ssiserver login: ssiuser_
```

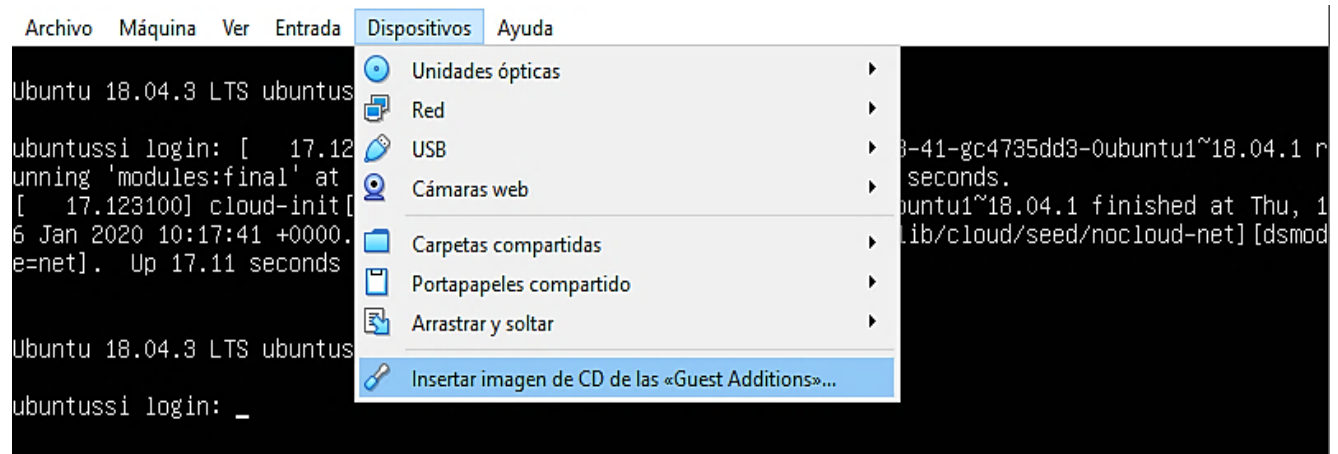
Primer inicio de sesión

Configuración de la máquina para mejorar la virtualización

Una vez que la máquina está operativa, y antes de dejarla lista para trabajar, una de las cosas que debemos hacer es mejorar su instalación aprovechando las "**Guest Additions**" de la plataforma de virtualización utilizada (muchos productos de virtualización tienen módulos similares de este tipo). El uso de estas *guest additions* es conveniente, ya que **mejorará el rendimiento general de la máquina** instalando controladores más adecuados. En caso de que se trate de *VirtualBox*, las instrucciones para instalarlos se pueden encontrar en este enlace: <https://www.tecmint.com/install-virtualbox-guest-additions-in-ubuntu/>. Sin embargo, los reproduciremos a continuación.



Ten en cuenta que *Guest Additions* necesitan que instalemos un conjunto de paquetes de *Linux* no instalados como estándar, como **dkms**, **build-essential** y **linux-headers-generic** para su correcta instalación. También es conveniente instalar las cabeceras del *kernel* de la arquitectura específica agregando el paquete **linux-headers-\$(uname -r)**. Con estos paquetes instalados podemos insertar la imagen en CD de las *Guest Additions* y proceder a la ejecución de su instalador:



Instalación de Guest Additions

Para ello primero montamos el CD, una vez insertado en el directorio **/media/cdrom**

```
operario@servidorubuntu:~$ sudo mount /dev/sr0 /media/cdrom
mount: dispositivo de bloques /dev/sr0 está protegido contra escritura; se monta
como sólo lectura
operario@servidorubuntu:~$
```

Acceso a archivos de Guest Additions

Accedemos a su contenido y ejecutamos **./VBoxLinuxAdditions.run** para seleccionar el ejecutable de las *Guest Additions* más adecuadas para la máquina que tenemos y proceder a su instalación:

```
operario@servidorubuntu:/media/cdrom$ dir
32Bit      cert      VBoxSolarisAdditions.pkg
64Bit      OS2       VBoxWindowsAdditions-amd64.exe
AUTORUN.INF runasroot.sh VBoxWindowsAdditions.exe
autorun.sh VBoxLinuxAdditions.run VBoxWindowsAdditions-x86.exe
operario@servidorubuntu:/media/cdrom$ sh ./VBoxLinuxAdditions.run
Verifying archive integrity... All good.
Uncompressing VirtualBox 4.3.28 Guest Additions for Linux.....
This program must be run with administrator privileges. Aborting
operario@servidorubuntu:/media/cdrom$ sudo sh ./VBoxLinuxAdditions.run
```

Ejecución del instalador de adiciones invitadas

Para seguir trabajando con la máquina es necesario hacer un reinicio.

En este punto es **MUY RECOMENDABLE** guardar la máquina tal cual, y trabajar con clones de esta a partir de ahora. De esta forma, si cometemos un error en la configuración de la máquina en alguna actividad y su solución no es trivial, podremos volver a la máquina de arranque sin perder demasiado tiempo. Las opciones 1 y 2 permiten reconstruir la máquina muy fácilmente, pero tampoco es mala idea hacerlo.



Software para instalar en la máquina virtual personalizada

Después de configurar las especificaciones de hardware correspondientes y el sistema operativo instalado, debes asegurarte de instalar el siguiente software en la máquina virtual para poder seguir correctamente los laboratorios.

- Agregar repositorio de Docker y actualizar `apt`

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable"
sudo apt-get -y update
```

- Actualización de paquetes

```
sudo apt-get -y full-upgrade
```

- Instalación de los programas necesarios

```
sudo apt-get install -y --no-install-recommends apt-utils apt-transport-https ca-
certificates gnupg-agent software-properties-common unzip curl wget tmux preload gpm mc
virt-what nano grep firefox iputils-ping net-tools less openssl bash-completion
```

- Instalación de la GUI XCFE4

```
sudo apt-get -y install -y xfce4
```

- Instalación de las *VirtualBox Additions* (solo si usas *VirtualBox* como plataforma de virtualización y no lo has hecho introduciendo el CD de instalación como vimos antes)

```
sudo apt-get -y install -y virtualbox-guest-dkms virtualbox-guest-utils virtualbox-guest-
x11
sudo VBoxClient --clipboard
sudo VBoxClient --draganddrop
sudo VBoxClient --vmsvga
sudo VBoxClient --checkhostversion
sudo VBoxClient --seamless
sudo adduser ssiuser vboxsf
```

- Instalar Docker y convertir a tu usuario actual en usuario de Docker sin necesidad de `sudo`

```
sudo apt-get -y install docker-ce docker-ce-cli containerd.io
sudo groupadd docker
sudo usermod -aG docker ssiuser
sudo systemctl enable docker
```

- Instalación de Docker Compose

```
sudo apt-get -y install docker-compose
```

- Cambiar la distribución del teclado al español (no es necesario si eliges español durante la instalación del sistema operativo)



```
loadkeys es
sudo dpkg-reconfigure console-setup
sudo sed -i 's/XKBLAYOUT=\\"w*"/XKBLAYOUT=\\"es\\"/g' /etc/default/keyboard
```

- Instalar **tmux** para tener varias ventanas y paneles de terminal (opcional). Puede consultar la *cheatsheet* de **tmux** en una sección anterior tener una buena configuración inicial. Además, los archivos para trabajar con *Vagrant* contienen un archivo **tmux.conf** con otra buena configuración inicial que puedes copiar a **/etc/tmux.conf**

```
sudo apt-get -y install tmux
```

- Limpiar la instalación de paquetes

```
sudo apt-get -y autoremove
sudo apt-get -y autoclean
```

- Reiniciar la máquina

```
sudo reboot
```

También recomendamos instalar ahora todos los programas mencionados en la carpeta **customization_scripts\software** de los archivos de máquina virtual de *Vagrant* que puedes descargar del campus virtual. Solo necesitas copiar y pegar los *scripts* proporcionados.

Pasos especiales en las plataformas de virtualización de Hyper-V

Si en lugar de *VirtualBox* estás utilizando *Hyper-V*, el procedimiento de instalación de software recomendado cambia únicamente en un solo paso. Cambia el paso "*Instalación de las VirtualBox Additions*" por este, asegurando así el mejor rendimiento posible de la VM en esta plataforma de virtualización:

- Habilitar *Hyper-V Linux Integration Services* (LIS)

```
sudo sed -i '$a hv_vmbus' /etc/initramfs-tools/modules
sudo sed -i '$a hv_storvsc' /etc/initramfs-tools/modules
sudo sed -i '$a hv_blkvsc' /etc/initramfs-tools/modules
sudo sed -i '$a hv_netvsc' /etc/initramfs-tools/modules
```

- Reemplazar el *kernel* predeterminado con **linux-virtual** para tener mejor experiencia de uso con LIS

```
sudo apt-get -y install linux-virtual linux-cloud-tools-virtual linux-tools-virtual
```

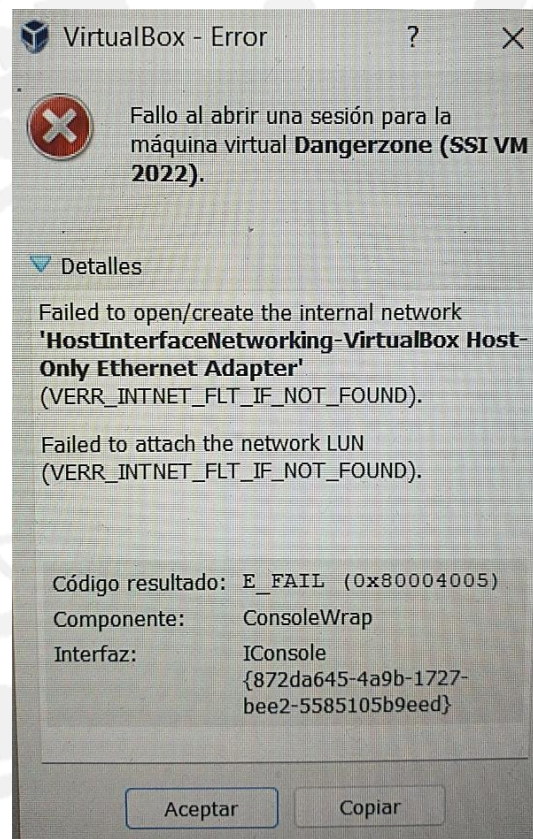
- Actualizar **initramfs**

```
sudo update-initramfs -u
```

Resolución de problemas general (cualquier opción de creación de máquinas)

Cuando arranco la MV, me da el error VERR_INTNET_FLT_IF_NOT_FOUND

Este error (ver imagen) ocurre cuando la instalación actual de *VirtualBox* falla y no configura un adaptador *host-only* y, por lo tanto, cada máquina virtual que use uno fallará en el arranque. La solución más rápida es simple, desactivar dicho adaptador en la pestaña de configuración "Red" de la máquina que da este error.



Mi MV arranca, pero solo veo una pantalla en negro y ninguna salida

Si tu MV no arranca y simplemente se congela con una pantalla en blanco y sin salida, la causa suele ser que tienes **Hyper-V** habilitado en tu host *Windows* y *VirtualBox* no puede funcionar. La solución es

- Usa la máquina no optimizada, que es idéntica en manejo a esta (recomendado)
- Desinstala *Hyper-V* (si realmente no lo necesitas, pero ten cuidado porque cosas como *Docker Desktop* y otros programas lo piden, así que verifica primero si realmente no lo necesitas para otras asignaturas)
- Desinstala *VirtualBox*, y deja que *Vagrant* detecte *Hyper-V* e intente instalar la máquina virtual en él (si es que vas a hacer este módulo). Puedes encontrar información adicional sobre el problema aquí: <https://forums.virtualbox.org/viewtopic.php?f=6&t=92673>

Por supuesto, ¡HAZ UNA COPIA DE SEGURIDAD DE CUALQUIER COSA VALIOSA ANTES DE DESINSTALAR NADA! 😊



Necesito *Hyper-V* para otras asignaturas / mi trabajo, y desactivarlo cada vez que arranco la VM es muy pesado

Algunos de vosotros nos habéis dicho que el software utilizado en otras asignaturas interfiere con la máquina virtual en algunas ocasiones. Específicamente, uno de los principales "culpables" es *Docker Desktop*, que usa *Hyper-V* por debajo. Si bien una solución es deshabilitarlo cada vez que nuestra MV vaya a arrancar, esto puede ser engorroso.

Por esa razón, hemos vuelto a examinar nuestra máquina virtual y hemos encontrado que la razón de esta incompatibilidad (icono de "tortuga verde", pantalla negra sin ninguna salida) está en una de las optimizaciones que hemos realizado para garantizar que la máquina vaya lo más rápido posible y ocupe la menor cantidad de RAM, de forma que funcione mejor en cualquier host.

Por ese motivo, **hemos creado una nueva versión de la máquina virtual** (Opciones 1 y 2) **que no utiliza estas optimizaciones**, y que hemos comprobado que funciona con los siguientes servicios activos, **que son todos los que deben desactivarse si no queremos usar *Hyper-V*** (no basta con desinstalar solo el primero, ver siguiente sección):

1. *Hyper-V*
2. *Virtual Machine Platform*
3. *Windows Hypervisor Platform*
4. *Windows Subsystem for Linux*



Activar o desactivar las características de Windows

Para activar una característica, active la casilla correspondiente. Para desactivarla, desactive la casilla. Una casilla rellena indica que solo está activada una parte de la característica.

- ☒ .NET Framework 4.8 Advanced Services
- ☐ Active Directory Lightweight Directory Services
- ☐ Bloqueo de dispositivo
- ☒ Características multimedia
- ☒ Cliente de Carpetas de trabajo
- ☐ Cliente Telnet
- ☐ Cliente TFTP
- ☐ Compatibilidad con el protocolo para compartir archivos SMB 1.0
- ☒ Compatibilidad de API para compresión diferencial remota
- ☐ Componentes heredados
- ☐ Contenedores
- ☒ Escritor de documentos XPS de Microsoft
- ☐ Espacio aislado de Windows
- ☐ Host protegido
- ☒ Hyper-V
- ☒ Imprimir en PDF de Microsoft
- ☒ Internet Explorer 11
- ☐ Internet Information Services
- ☐ MultiPoint Connector
- ☐ Núcleo de web hospedable de Internet Information Services
- ☒ Plataforma de máquina virtual
- ☒ Plataforma del hipervisor de Windows
- ☐ Protección de aplicaciones de Microsoft Defender
- ☐ Protocolo de puente del centro de datos
- ☐ Servicio WAS (Windows Process Activation Service)
- ☒ Servicios de impresión y documentos
- ☐ Servicios de TCPIP simple (por ej; echo, daytime etc)
- ☐ Servicios para NFS
- ☐ Servidor de cola de mensajes Microsoft Message Queue (MSMQ)
- ☒ SMB directo
- ☒ Subsistema de Windows para Linux
- ☐ Windows Identity Foundation 3.5
- ☒ Windows PowerShell 2.0
- ☐ Windows Projected File System
- ☐ Windows TIFF IFilter

Aceptar

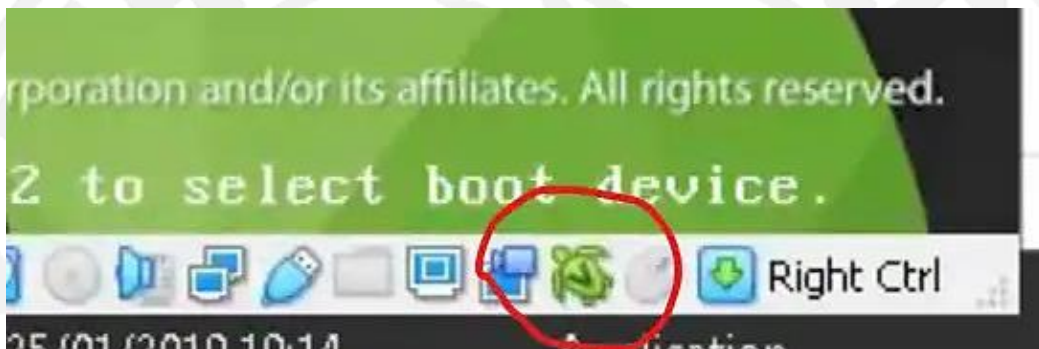
Cancelar

No obstante, es posible que la activación de los servicios 2 y 3 sea necesaria para un buen funcionamiento de esta nueva versión, que puedes encontrar justo debajo de la primera (la optimizada) en el campus virtual. La nueva versión de la MV **no cambia nada de la anterior en términos de software y características**. Hemos comprobado que pierde alrededor del 15% de rendimiento, pero funciona en un entorno donde hemos reproducido el escenario problemático que algunos de vosotros habéis encontrado, en el que el original no era capaz de arrancar. Ten en cuenta que aunque aparece el icono de "tortuga verde", esta vez la máquina puede arrancar y cumplir su misión.

Operaciones de gestión de MV generales

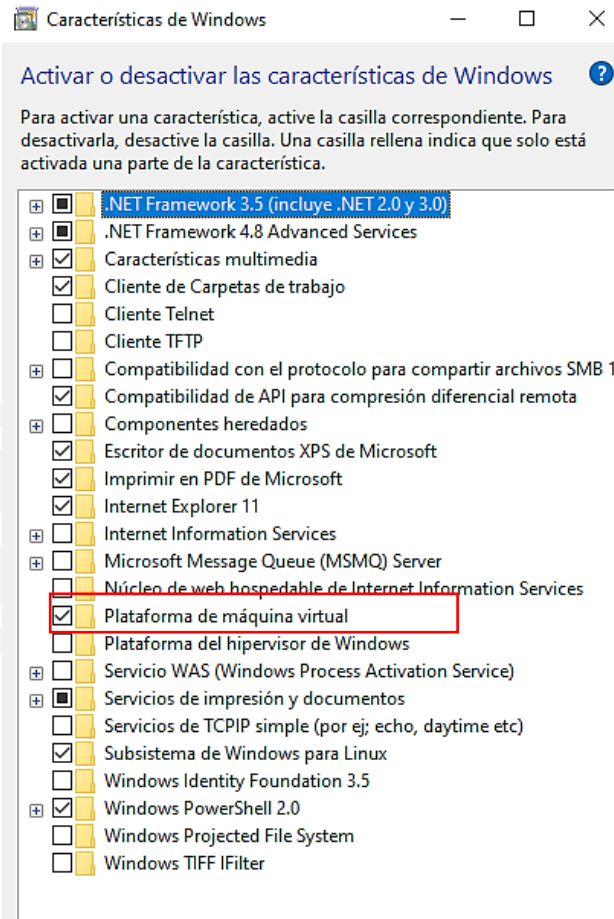
Resolviendo incompatibilidades entre VirtualBox e Hyper-V

Algunos estudiantes han experimentado el terrible "problema de la tortuga verde", lo que significa que *VirtualBox* se está ejecutando lentamente debido a algo que bloquea la aceleración de la virtualización por *hardware* (esto se indica con un icono de tortuga verde en la barra inferior de la ventana de la pantalla de la máquina virtual). El principal indicador de sufrir este problema **es una pantalla negra sin salida alguna**.



Por lo general, esto se debe a que *Hyper-V* bloquea esta característica y por eso hemos creado una MV especial no optimizada que puede funcionar en estas circunstancias (ver sección anterior). No obstante, algunos de vosotros habéis optado por desinstalar *Hyper-V* para poder usar la maquina optimizada, ya que no vais a usarlo más. El problema es que algunas veces **la desinstalación de *Hyper-V* no arregla el problema**. El motivo es que aún tenéis algo que usa la virtualización y es más difícil de localizar, lo que le da el problema de lentitud (la tortuga). Alguien puso en los foros de *VirtualBox* la lista de software problemático que causa este problema: <https://forums.virtualbox.org/viewtopic.php?f=6&t=92673>

Sin embargo, descubrimos que la mayoría de vosotros tenéis el problema debido a un culpable, aparte de *Hyper-V*: la función de Windows "*Plataforma de máquina virtual*" habilitada.



Si deshabilitas esto, tu *VirtualBox* comenzará a ejecutarse a velocidad normal de nuevo (desaparece la tortuga verde y aparece un icono con una V azul), con lo que también vuestra máquina dejará de sufrir ralentizaciones. Esto pasa sin importar la opción que uses para crear la VM, es un problema entre *Microsoft* y *VirtualBox*.

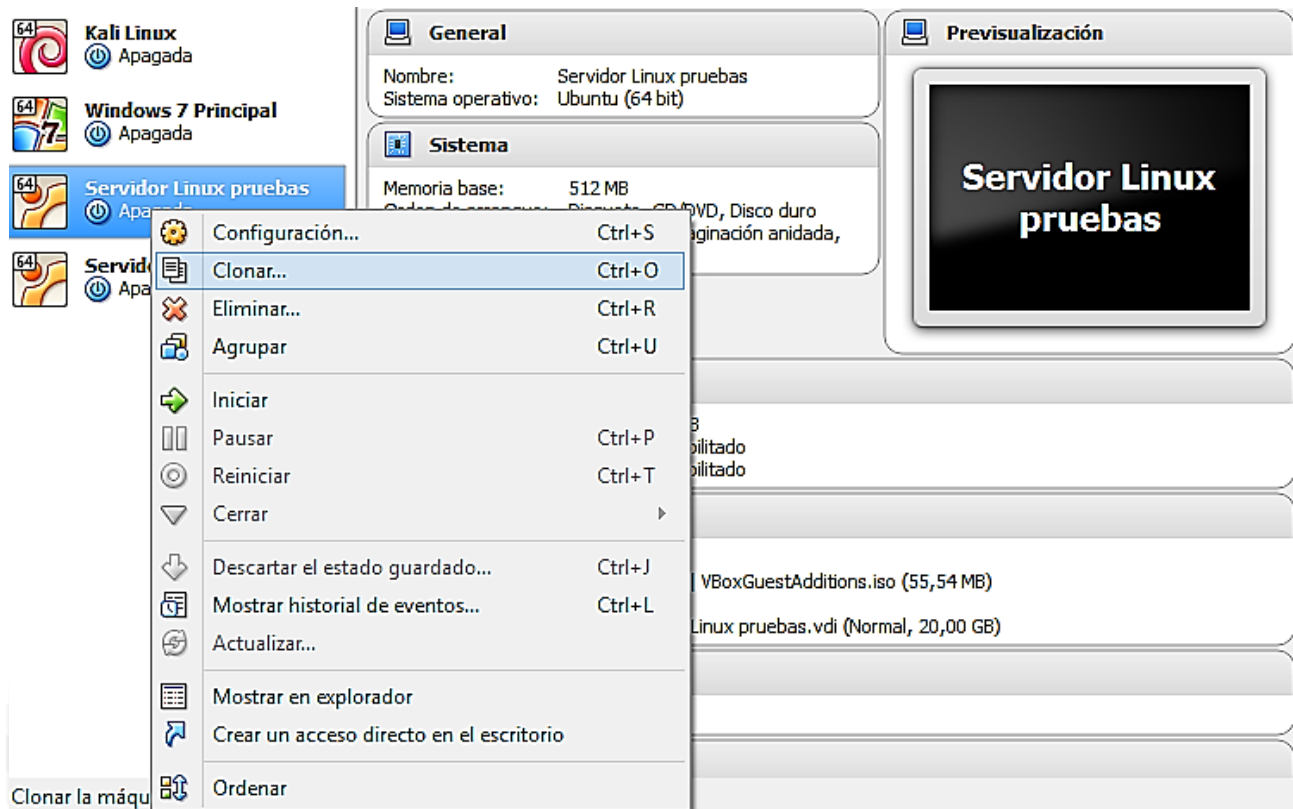
Clonación y configuración de máquinas

Si debemos instalar N máquinas *Linux*, repetir un proceso de instalación para cada una de ellas es tedioso, propenso a errores (especialmente si no lo automatizamos) y, por lo tanto, inadecuado. Normalmente lo que se hace en estos casos es **preparar una máquina "modelo"**, como la que hemos construido, y una vez optimizada, clonarla tantas veces como sea necesario para crear una base para máquinas para diferentes propósitos, o tener más copias de ella disponibles. **Otra opción es trabajar con instantáneas** de la máquina base, algo que haremos más adelante. Veamos cómo realizar el proceso de clonación.

Lo primero que hay que hacer antes de clonar una MV es estudiar si merece la pena hacer una **compactación de HD**. Este es un proceso que examina el disco duro de la máquina virtual y lo optimiza para que el archivo que representa ese disco duro en el *host* ocupe lo menos posible, sin romper la máquina 😊. Para una máquina que ha estado funcionando durante mucho tiempo y / o ha instalado o desinstalado múltiples aplicaciones es un proceso interesante, ya que **puede disminuir en gran medida el tamaño que ocupa**. Sin embargo, en nuestro caso, al ser una máquina recién instalada, no lo haremos ya que no merece la pena. Sin embargo, las instrucciones sobre cómo realizar este proceso se pueden encontrar aquí: <https://www.virtualbox.org/manual/ch08.html> o en: <https://www.howtogeek.com/312883/how-to-shrink->

[a-virtualbox-virtual-machine-and-free-up-disk-space/](#). Esto facilita la descarga y distribución de la máquina.

Una vez que todo esté listo, podemos elegir cualquier máquina y proceder a clonarla con la opción *VirtualBox* asociada:



Clonación de una máquina en VirtualBox

La máquina clonada aparecerá en el inventario de *VirtualBox*, por lo que necesitamos darle un **nombre nuevo que no exista en el mismo**:

Nuevo nombre de máquina

Seleccione un nombre para la nueva máquina virtual. La nueva máquina será un clon de la máquina **Servidor Linux pruebas**.

☒ Reinicializar la dirección MAC de todas las tarjetas de red

Nombrar un nuevo clon de una máquina

Es importante **cambiar los MACs de las tarjetas de red** porque es probable que las máquinas clonadas estén conectadas vía red con su "original" y entre ellas, y queremos evitar errores de comunicación que se producirán si intentan comunicar dos máquinas con tarjetas que tienen la misma MAC.

Una vez hecho esto, seleccionamos el tipo de clonación que queremos, que en este caso será **completa** para poder moverla sin restricciones a donde queramos:



Tipo de clonación

Seleccione el tipo de clonación que desea crear.

Si selecciona **Clonación completa**, una copia exacta (incluyendo todos los archivos de unidad de disco duro virtual) de la máquina original serán creados.

Si selecciona **Clonación enlazada**, una nueva máquina será creada, pero los archivos de las unidades de disco duro virtuales serán vinculados a los archivos de unidad de disco duro virtual de la máquina original y no podrá mover la nueva máquina virtual a una computadora diferente sin mover los originales también.

Si crea una **Clonación enlazada** entonces una nueva instantánea será creada en la máquina virtual original como parte del proceso de clonación.

- ☒ Clonación completa
☐ Clonación enlazada

Clonar

Cancelar

Selección del tipo de clonación que se va a realizar

Ahora tenemos dos máquinas diferentes, la máquina cliente, y el servidor, pero ambas tienen el mismo nombre interno o *hostname* (**ubuntuserver**, el que le dimos en la instalación). Debemos cambiárselo al menos a una de ellas para evitar posibles problemas de red si ambos funcionan al mismo tiempo. Para ello editamos el archivo **/etc/hostname:**

```
ssuser@ssiserver:~$ sudo nano /etc/hostname  
[sudo] password for ssuser:
```

Cambiar el nombre de la máquina clon (I)

Y también debes editar el archivo **/etc/hosts** para cambiar el nombre de la máquina que aparece ahí.

Creación de instantáneas de máquinas en funcionamiento

Los *Snapshots* o instantáneas son una forma "barata" de tener una máquina en la que podemos hacer operaciones y **volver a su estado anterior si algo sale mal**. Clonar una máquina te permite crear una máquina separada, pero idéntica a la que ya tienes. Esto es perfectamente válido, **pero consume muchos recursos** porque duplica todos sus contenidos. En ciertos contextos, como nuestros laboratorios de la asignatura, puede ser más conveniente utilizar instantáneas, porque el espacio que consumen es mucho menor, y por lo tanto se crean mucho más rápidamente.

Con las instantáneas puedes guardar un estado determinado de una máquina virtual para su uso posterior. En cualquier momento puedes volver a cualquiera de esos estados, incluso si la máquina virtual ha cambiado considerablemente desde entonces. Por lo tanto, **una instantánea de una máquina virtual es como una máquina en el estado Guardado**, pero puede haber muchos de ellos y estos estados guardados se conservan.



Para ver las instantáneas de una máquina virtual, haz clic en el nombre de la máquina en el GUI de *VirtualBox*. A continuación, haz clic en el icono de lista situado junto al nombre de la máquina y selecciona **Instantáneas**. Hasta que se haga una instantánea de la máquina la lista de instantáneas estará vacía, excepto el elemento **Estado actual**, que representa la máquina virtual tal y como está ahora.

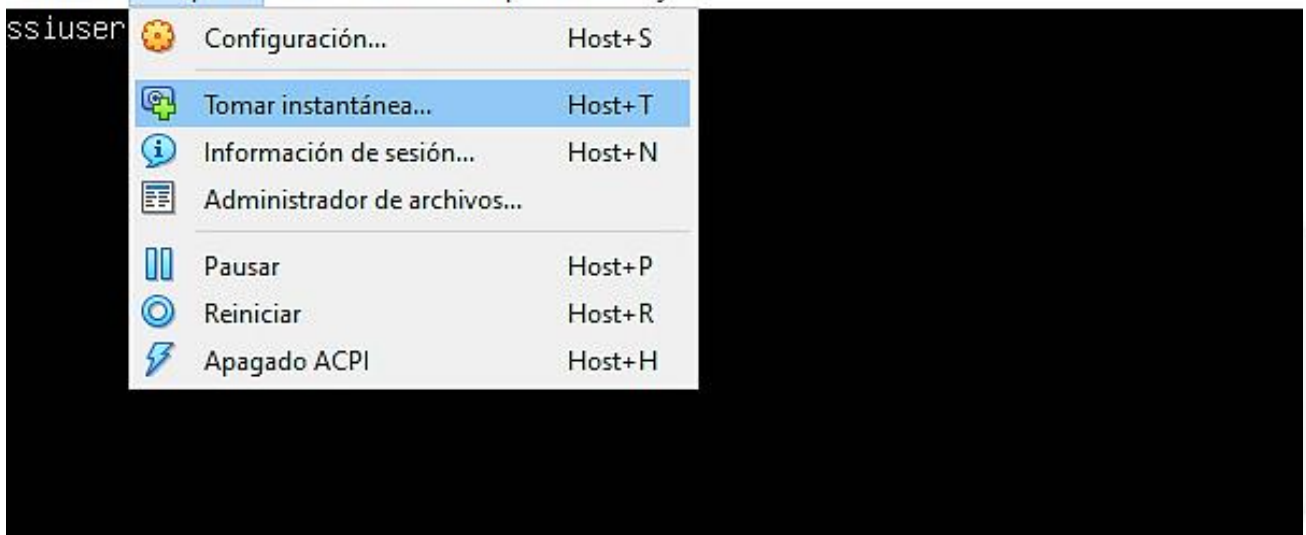
Tomar, restaurar y eliminar instantáneas

Toma una instantánea

Esto hace una copia del estado actual de la máquina al que se puede volver en cualquier momento posterior. **Si la máquina virtual** se está ejecutando, selecciona *Tomar instantánea* en el menú desplegable *Máquina* en la ventana máquina virtual.

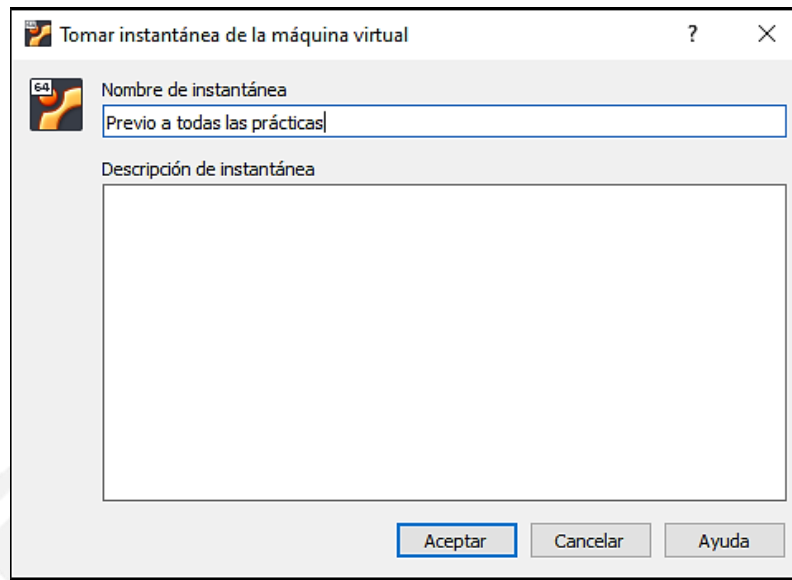
Ubuntu Server Base SSI [Corriendo] - Oracle VM VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda



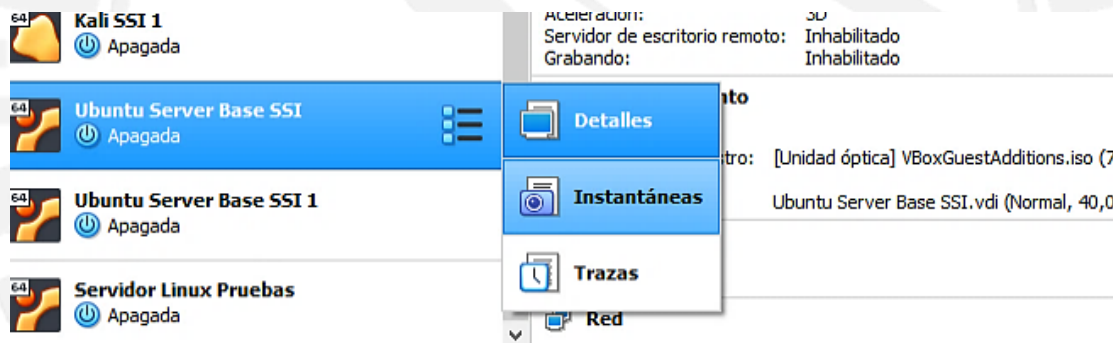
Tome una instantánea de una máquina en funcionamiento

Esto muestra una ventana que solicita un nombre de instantánea. Este nombre es puramente para referenciarla, y ayuda a recordar el estado de la instantánea. **En nuestra asignatura tiene sentido generar estados para cada laboratorio**, de manera que puedas volver a cada uno de ellos de forma independiente y poder partir de las operaciones realizadas en cada uno si algo sale mal. También deberías crear un estado base de la máquina actual para volver al "**estado cero**" si es necesario. Puedes poner un texto más largo en el campo *Descripción*.



Nombre de una instantánea

Si la máquina virtual está en el estado **Guardado o Desactivado**, en el nombre de la máquina virtual en la ventana principal de *VirtualBox* se puede hacer clic en el icono Lista junto al nombre de la máquina y podemos seleccionar *Instantáneas*. Esto muestra la ventana de instantáneas donde puede tomar una instantánea del estado actual de la máquina o volver a una de las guardadas anteriormente.



Tomar una instantánea de una máquina detenida o guardada

En cualquier caso, aparecerá una lista de instantáneas. Si debajo de una instantánea aparece un elemento llamado *Estado actual*, significa que el estado actual de la máquina virtual es una variación de esa instantánea. Si se toma otra instantánea más tarde, se muestran secuencialmente, y cada instantánea posterior se deriva de una instantánea anterior.

Lista de instantáneas para una máquina virtual

VirtualBox no tiene un límite en el número de instantáneas que puede tomar, más allá del espacio en disco del host. Cada instantánea almacena el estado de la máquina virtual y, por lo tanto, ocupa espacio en disco. Sin embargo, es mucho menos que una máquina virtual completa (o uno de sus clones). Por ejemplo, la base de *Ubuntu* de la asignatura podría ocupar unos 4Gb



Logs	16/01/2020 11:55	Carpeta de archivos	
Snapshots	16/01/2020 12:03	Carpeta de archivos	
Ubuntu Server Base SSI	16/01/2020 12:03	VirtualBox Machin...	22 KB
Ubuntu Server Base SSI.vbox-prev	16/01/2020 12:03	Archivo VBOX-PREV	22 KB
Ubuntu Server Base SSI	16/01/2020 11:59	Virtual Disk Image	3.784.704 KB

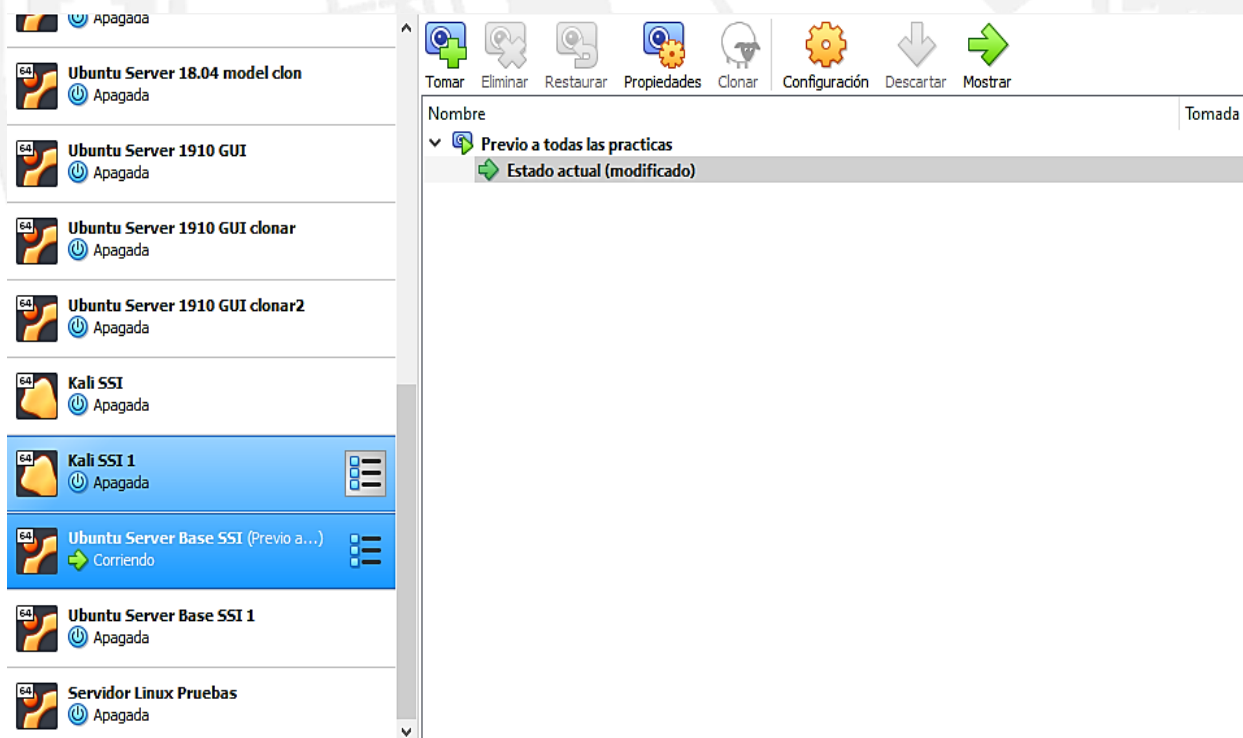
Espacio ocupado por la MV Ubuntu o uno de sus clones

Sin embargo, cada instantánea de la máquina que tomamos (que se encuentra en la carpeta *Instantáneas* en la ruta donde se guarda la máquina) ocupa mucho menos espacio.

Nombre	Fecha de modificación	Tipo	Tamaño
{2b77b0ae-93b3-4952-82ae-3cd7a7abefaf}	16/01/2020 12:03	Virtual Disk Image	27.648 KB
{3be97ea4-94ba-4854-aedf-5f1bd4ac4cf9}	16/01/2020 12:05	Virtual Disk Image	19.456 KB
2020-01-16T10-59-52-226259100Z.sav	16/01/2020 11:59	Archivo SAV	588.905 KB
2020-01-16T11-03-25-769435400Z.sav	16/01/2020 12:03	Archivo SAV	588.969 KB

Espacio ocupado por las instantáneas de la VM Ubuntu

La máquina que se ejecuta desde el estado base aparecerá así en la lista de instantáneas



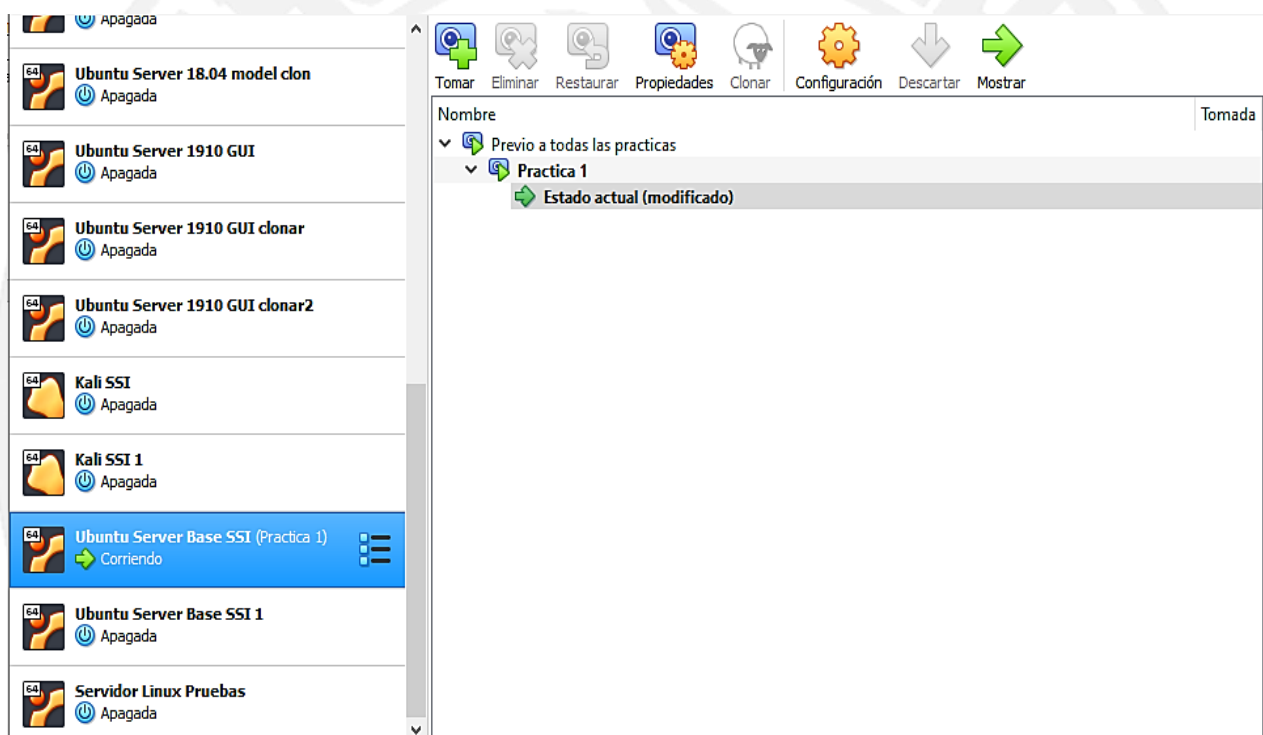
Lista de instantáneas

Si ahora escribimos un archivo de texto en el disco duro y tomamos una instantánea, el nuevo estado de la máquina contendrá este archivo, mientras que el estado base no lo hará.



```
ssiuser@ubuntussi:~$ ls
ficheroprac1.txt
ssiuser@ubuntussi:~$ ls -la
total 40
drwxr-xr-x 5 ssiuser ssiuser 4096 Jan 16 11:02 .
drwxr-xr-x 3 root    root    4096 Jan 13 19:29 ..
-rw-r--r-- 1 ssiuser ssiuser   94 Jan 13 19:55 .bash_history
-rw-r--r-- 1 ssiuser ssiuser  220 Apr  4 2018 .bash_logout
-rw-r--r-- 1 ssiuser ssiuser 3771 Apr  4 2018 .bashrc
drwx----- 2 ssiuser ssiuser 4096 Jan 13 19:34 .cache
-rw-rw-r-- 1 ssiuser ssiuser   10 Jan 16 11:02 ficheroprac1.txt
drwx----- 3 ssiuser ssiuser 4096 Jan 13 19:34 .gnupg
drwxrwxr-x 3 ssiuser ssiuser 4096 Jan 16 11:02 .local
-rw-r--r-- 1 ssiuser ssiuser  807 Apr  4 2018 .profile
-rw-r--r-- 1 ssiuser ssiuser    0 Jan 13 19:35 .sudo_as_admin_successful
ssiuser@ubuntussi:~$ _
```

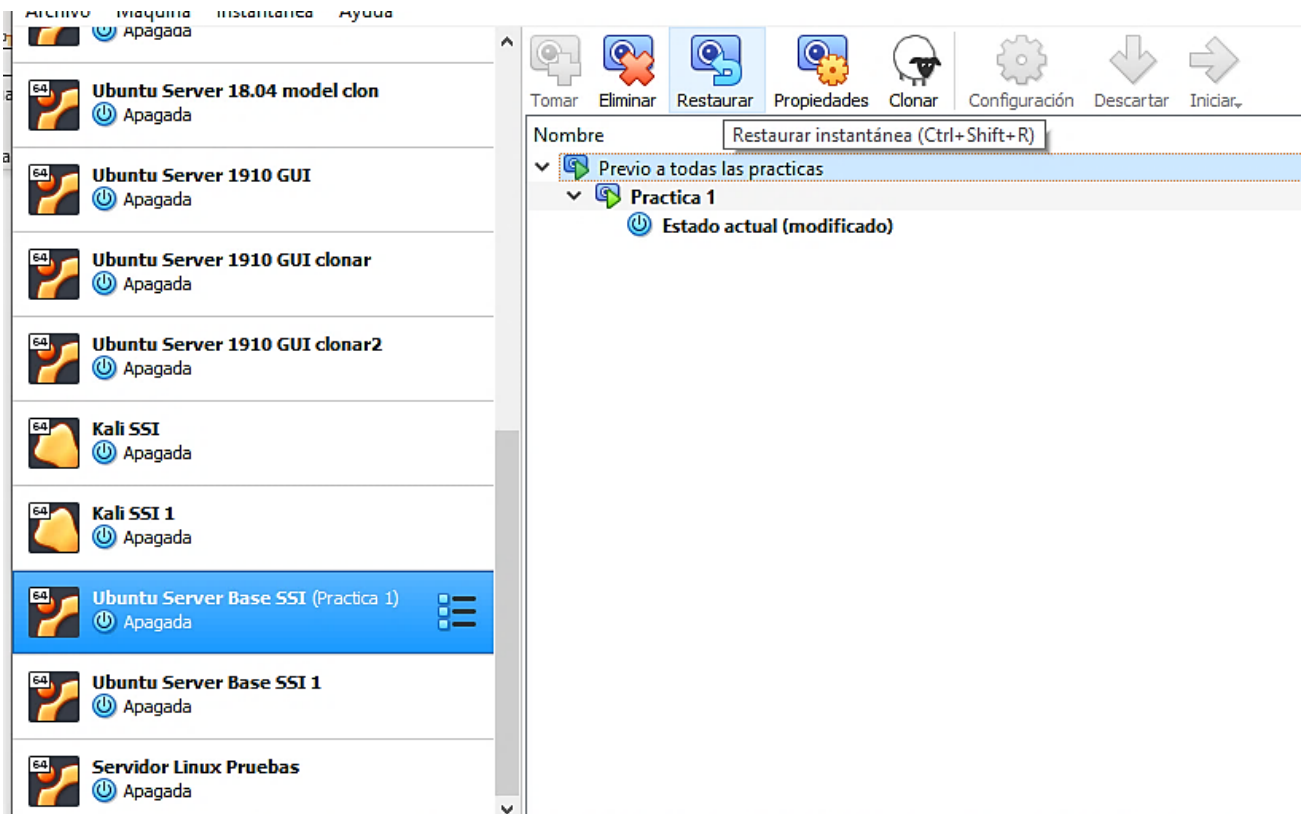
Vemos que al tomar la instantánea, el estado actual aparece debajo de ella.



Nueva instantánea en la lista de instantáneas

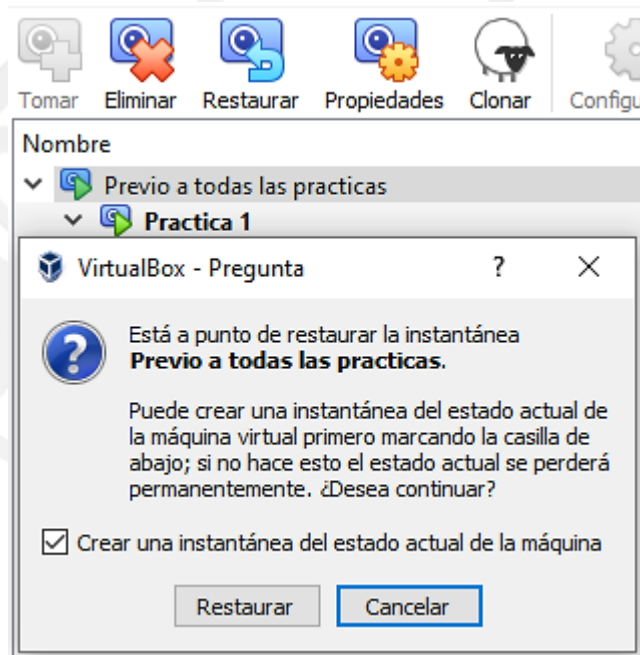
Restaurar una instantánea

Con la máquina guardada o apagada podemos ir a la lista de instantáneas, pulsar con el botón derecho sobre cualquier instantánea y seleccionar *Restaurar*. Cuando se restaura una instantánea, se retrocede o avanza en el tiempo en función de cuándo se haya tomado en relación con el estado actual. Se pierde el estado actual de la máquina y la máquina se restaura al estado exacto en el que se encontraba cuando se tomó la instantánea.



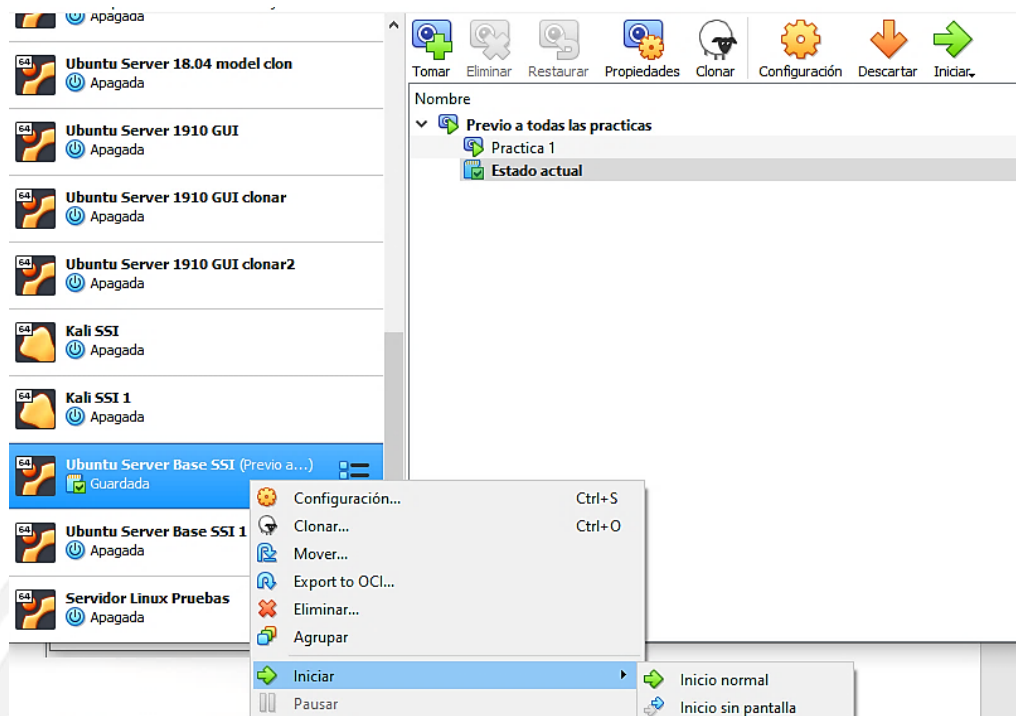
Restauración de una instantánea

Esta pérdida del estado actual de la máquina puede ser un problema en algunos contextos, y por lo tanto *VirtualBox* pregunta primero si deseas tomar una instantánea del estado actual antes de restaurar el que hemos seleccionado. En este caso diremos que no.



Aviso de pérdida del estado actual de la máquina por restauración

En este ejemplo hemos vuelto al estado inicial (aparece en negrita) y una vez hecho esto, podemos volver a poner en marcha la máquina.



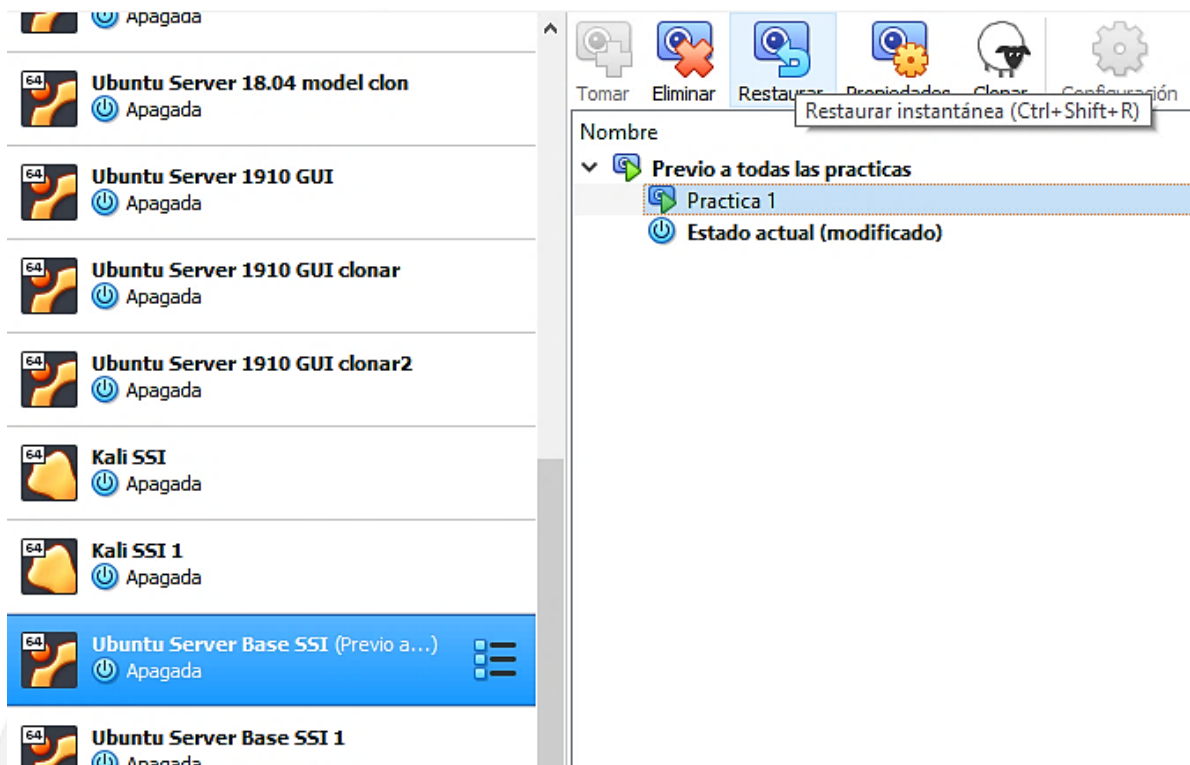
Inicio de la máquina en la instantánea base

Cuando lo iniciamos, vemos que la máquina ya no tiene el archivo que habíamos creado, porque eso solo se hizo en el estado Laboratorio 1.

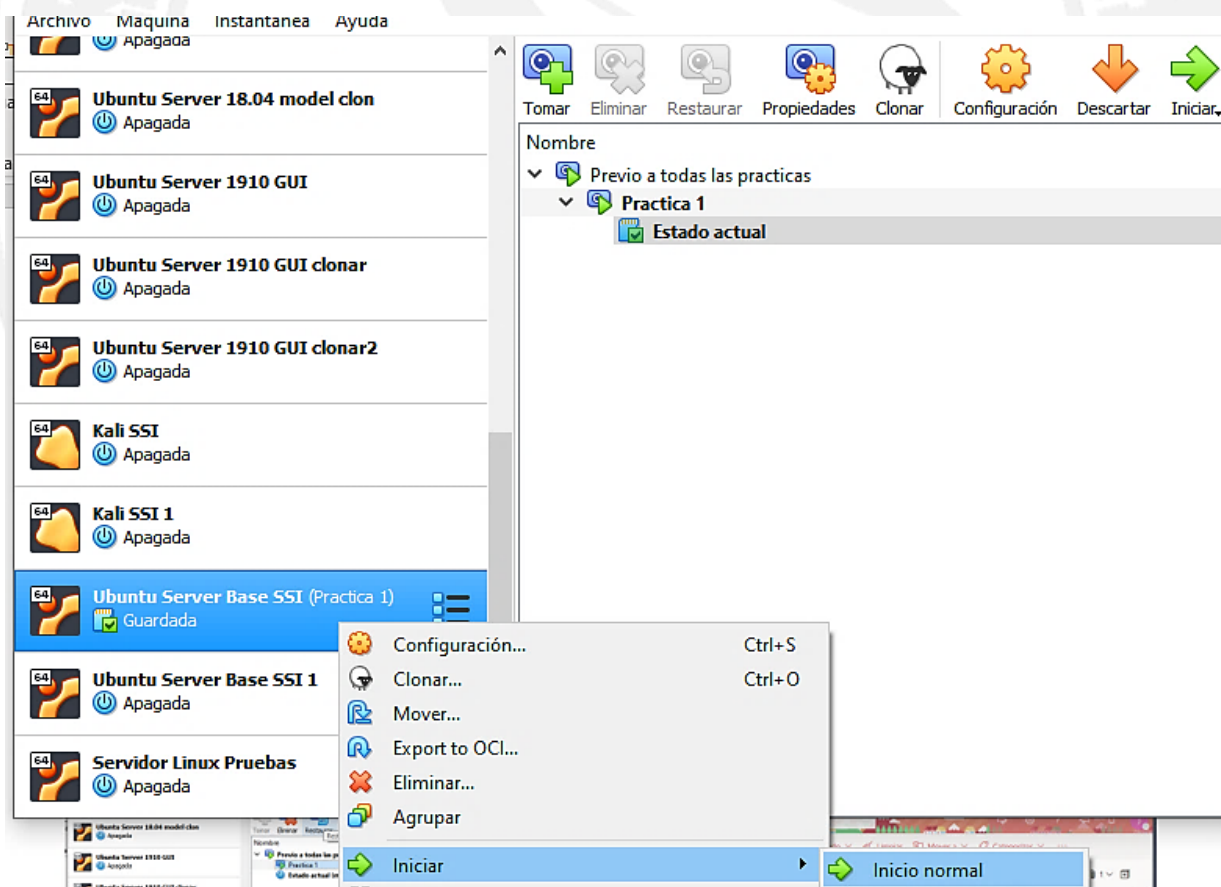
```
Ubuntu Server Base SSI (Previo a todas las practicas) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
ssiuser@ubuntussi:~$ ls -la
total 32
drwxr-xr-x 4 ssiuser ssiuser 4096 Jan 13 19:39 .
drwxr-xr-x 3 root    root    4096 Jan 13 19:29 ..
-rw-r--r-- 1 ssiuser ssiuser  94 Jan 13 19:55 .bash_history
-rw-r--r-- 1 ssiuser ssiuser 220 Apr  4 2018 .bash_logout
-rw-r--r-- 1 ssiuser ssiuser 3771 Apr  4 2018 .bashrc
drwxr-xr-x 2 ssiuser ssiuser 4096 Jan 13 19:34 .cache
drwxr-xr-x 3 ssiuser ssiuser 4096 Jan 13 19:34 .gnupg
-rw-r--r-- 1 ssiuser ssiuser 807 Apr  4 2018 .profile
-rw-r--r-- 1 ssiuser ssiuser  0 Jan 13 19:35 .sudo_as_admin_successful
ssiuser@ubuntussi:~$ _
```

Máquina en su estado original

Asimismo, podemos volver a restaurar el laboratorio 1 repitiendo el proceso y recuperando los cambios de estado (en este caso nuestro archivo)



Cambiar a un estado posterior



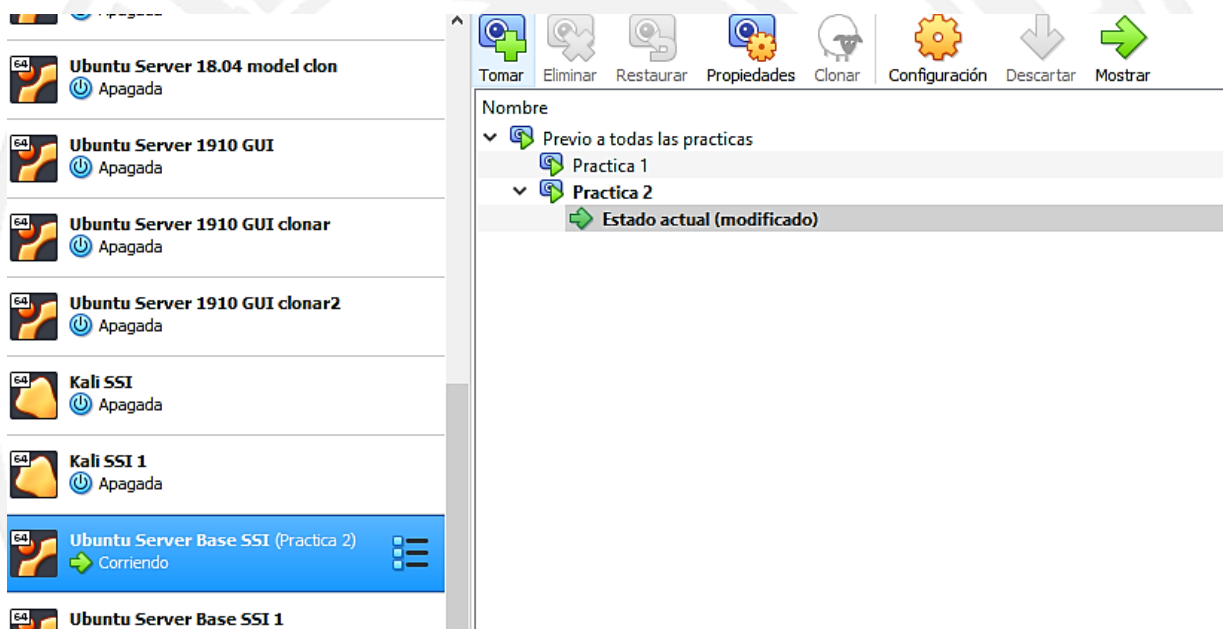
Lugar para restaurar un estado posterior



```
ssiuser@ubuntussi:~$ ls -la
total 40
drwxr-xr-x 5 ssiuser ssiuser 4096 Jan 16 11:02 .
drwxr-xr-x 3 root    root    4096 Jan 13 19:29 ..
-rw-r----- 1 ssiuser ssiuser  94 Jan 13 19:55 .bash_history
-rw-r--r-- 1 ssiuser ssiuser 220 Apr  4 2018 .bash_logout
-rw-r--r-- 1 ssiuser ssiuser 3771 Apr  4 2018 .bashrc
drwx----- 2 ssiuser ssiuser 4096 Jan 13 19:34 .cache
-rw-rw-r-- 1 ssiuser ssiuser  10 Jan 16 11:02 ficheroprac1.txt
drwx----- 3 ssiuser ssiuser 4096 Jan 13 19:34 .gnupg
drwxrwxr-x 3 ssiuser ssiuser 4096 Jan 16 11:02 .local
-rw-r--r-- 1 ssiuser ssiuser 807 Apr  4 2018 .profile
-rw-r--r-- 1 ssiuser ssiuser   0 Jan 13 19:35 .sudo_as_admin_successful
ssiuser@ubuntussi:~$ _
```

Estado posterior con los cambios realizados

Ahora podríamos repetir el mismo proceso para el resto de los laboratorios y proceder de la misma manera, haciendo una lista de *instantáneas* que nos ayuden a seguir la asignatura.



Lista de instantáneas por laboratorio

Para obtener más información sobre el contenido de una *instantánea* y otros detalles, recomendamos consultar: <https://www.virtualbox.org/manual/ch01.html#snapshots>

Carpetas compartidas

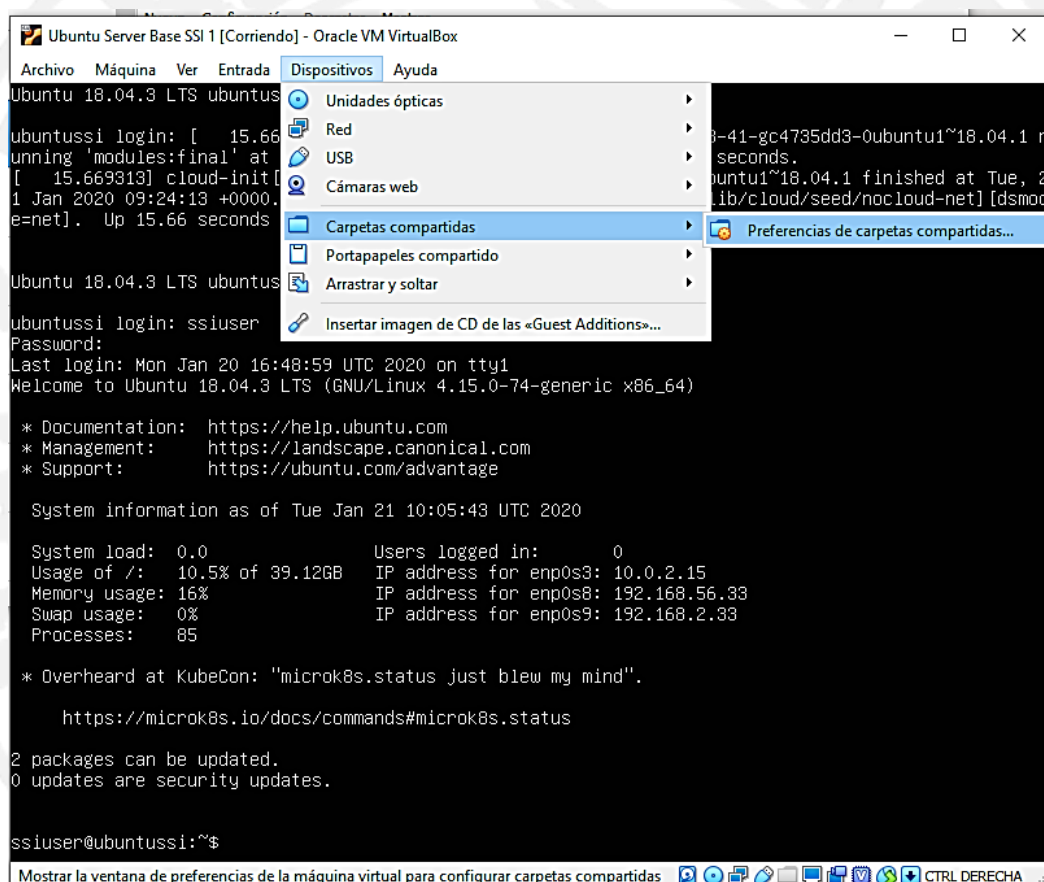
Nota: esto no es necesario si creaste la máquina virtual mediante la opción 1, que ya tiene carpetas compartidas configuradas.

Una de las formas más típicas de **compartir información entre una máquina virtual y su host** es mediante la creación de carpetas compartidas. Este mecanismo permite la creación de una carpeta en la máquina virtual, cuyo contenido se comparte con el host:



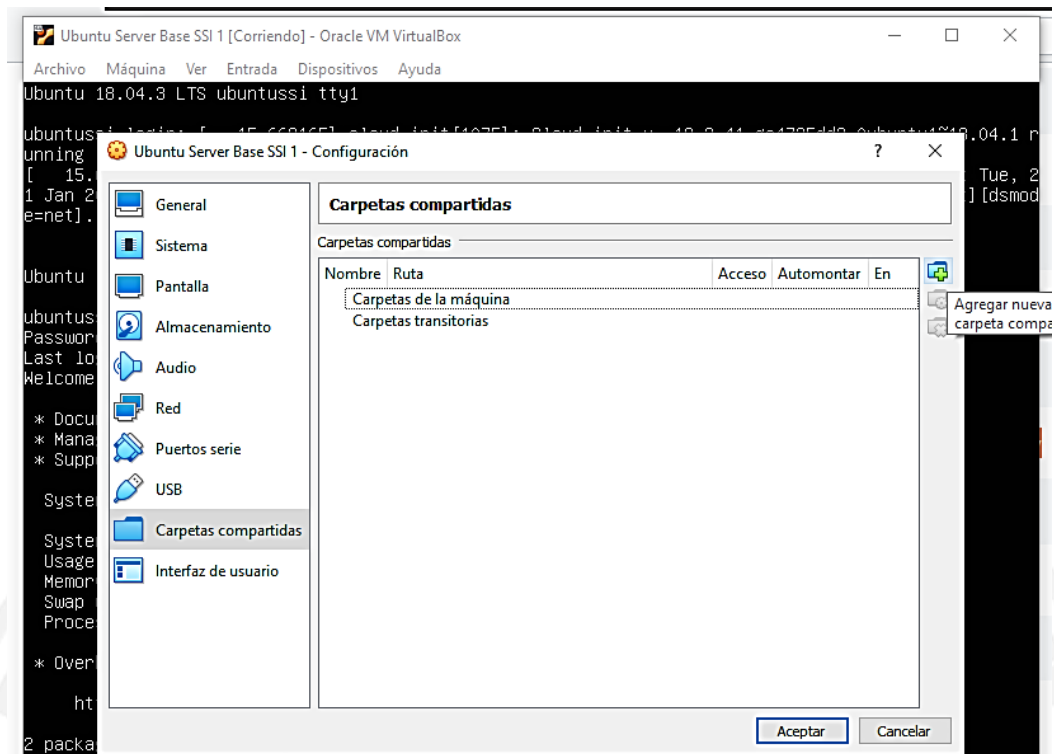
- Todo lo que se haya copiado o creado en esa carpeta en el host será legible, editable y copiable desde la máquina virtual.
- En consecuencia, todo lo que copiemos o creemos en la máquina virtual en esa carpeta será legible, editable y copiable desde el *host*.

Por lo tanto, este es un mecanismo de transferencia de archivos bidireccional adecuado para muchos usos. Solo tiene dos condiciones: **tener previamente instaladas las *Guest Additions* y crear y asignar las carpetas compartidas adecuadamente**. Esto último depende de la máquina virtual y de lo que haya instalado. Si la máquina tiene GUI, en cuanto creamos la carpeta, podremos acceder a ella desde su explorador de archivos y trabajar con ella como si fuera una carpeta local. Sin embargo, si no disponemos de GUI debemos hacer varios pasos adicionales para conseguir acceder a la misma. En cualquier caso, lo primero es crear la carpeta compartida, y para ello iremos a "*Dispositivos – Carpetas compartidas – Preferencias de carpetas compartidas*" (no importa si la máquina está encendida o no).



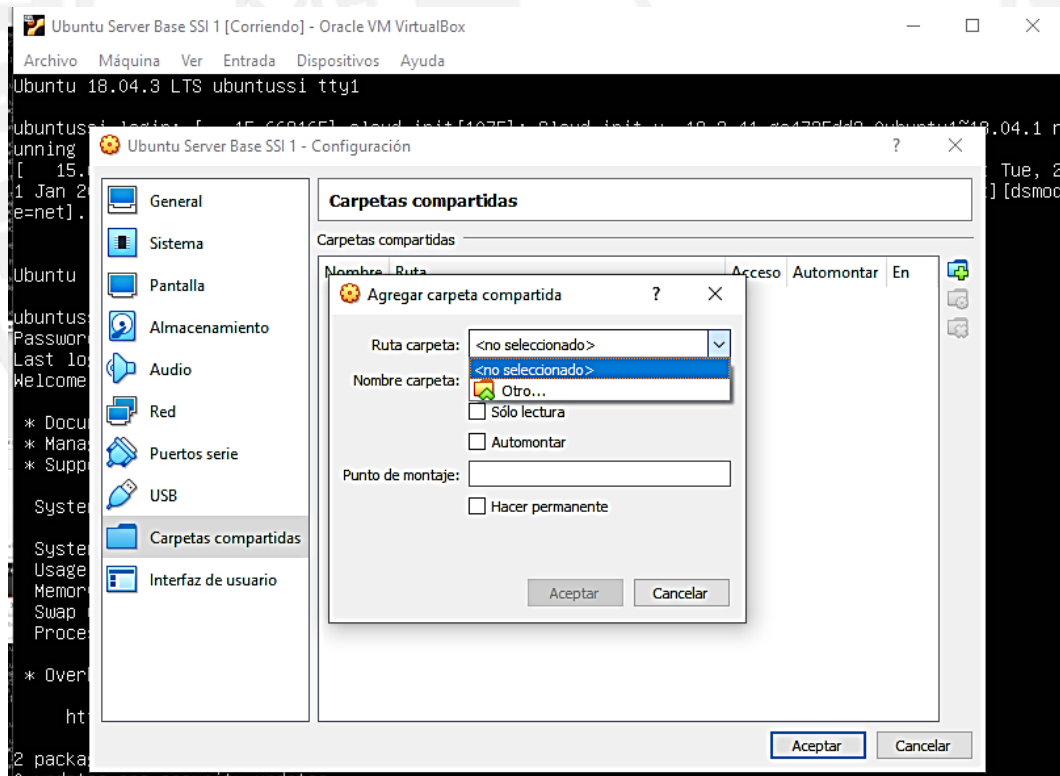
Acceso a la interfaz de carpetas compartidas de una máquina *VirtualBox*

Esto muestra una pantalla que nos pide que creamos una carpeta de máquina (potencialmente permanente) o una transitoria (no permanente). Elegimos la primera opción



Carpetas compartidas actualmente en una máquina VirtualBox

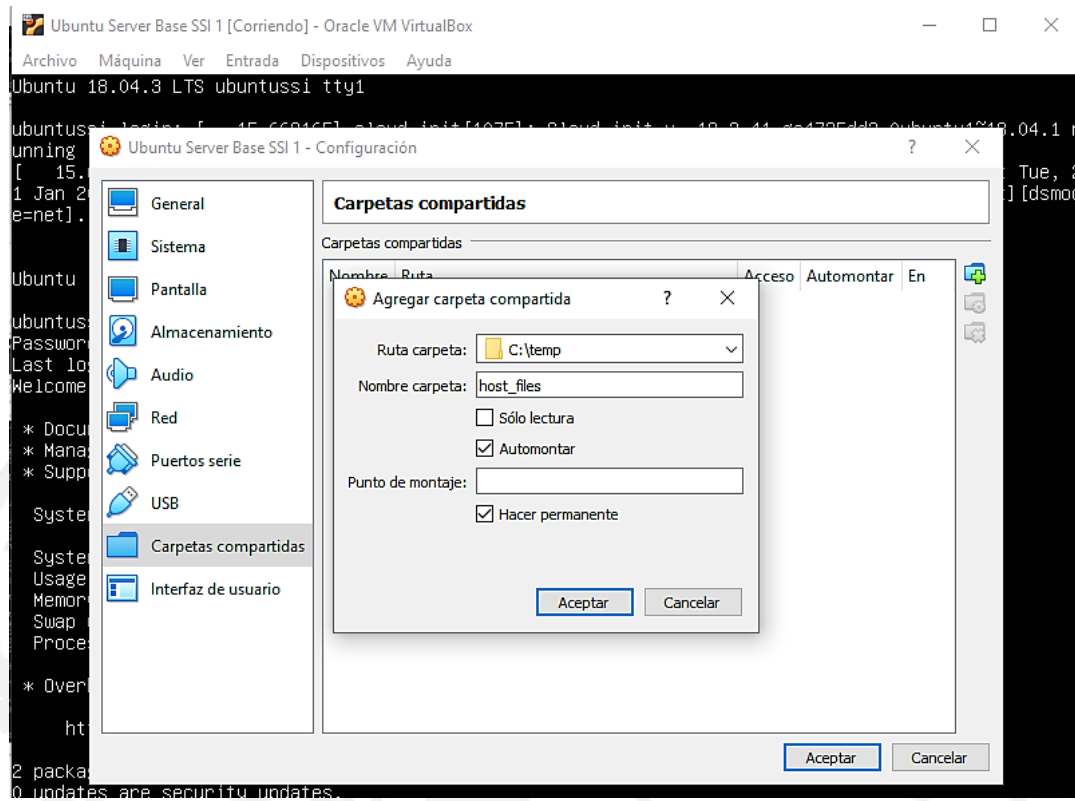
Ruta carpeta es la carpeta del *host* que queremos utilizar para escribir o leer contenido, es decir, qué lugar de nuestro disco duro utilizaremos para los archivos que vayan hacia o desde la máquina.



Selección de una carpeta de host



Seleccionamos "Otro" y una carpeta existente (**c:/temp** en el ejemplo). Nombramos la carpeta compartida (**host_files**, nombre que se utilizará en el lado de la máquina virtual) y marcamos las opciones "**Automontar**" y "**Hacer permanente**" para facilitar su uso. Una vez hecho esto, aceptamos.

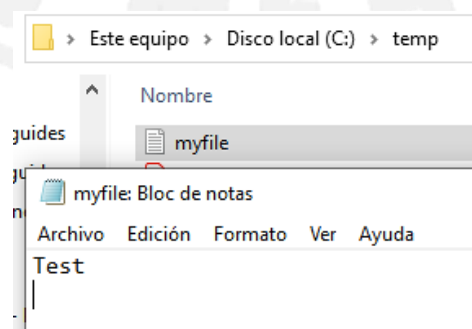


Configuración completa de una carpeta compartida de VirtualBox

El siguiente paso es crear un directorio en la máquina virtual donde se verá el contenido del *host*. Creamos el directorio "**shared**" en el **home** del usuario actual con **mkdir ~/shared** y luego montamos la carpeta compartida creada anteriormente sobre el directorio que acabamos de crear en el MV (aquí usamos el nombre especificado en "Nombre carpeta" en la interfaz *VirtualBox*: **sudo mount -t vboxsf host_files ~/shared** (**host_data** si usamos la máquina *Vagrant*)

```
ssiuser@ubuntussi:~$ sudo mount -t vboxsf host_files ~/shared
```

Ahora la carpeta del *host* será totalmente accesible desde la máquina virtual, y puedes copiar o extraer los archivos con ella. Por ejemplo, si editamos un archivo de texto en la máquina virtual con **nano myfile.txt** dentro de esa carpeta y ponemos el texto "**Test**", el *host* de *Windows* también lo mostrará.



Carpeta compartida vista desde un host de Windows



En caso de que la carpeta no esté montada permanentemente en la máquina, podemos utilizar la técnica explicada aquí para hacerla persistente:

<https://gist.github.com/estorgio/1d679f962e8209f8a9232f7593683265>

