



Fuente: Bing Chat AI

LABORATORIO "0A". CREANDO LA MÁQUINA VIRTUAL PARA LA ASIGNATURA

DEPARTAMENTO DE INFORMÁTICA. UNIVERSIDAD DE OVIEDO

Seguridad de Sistemas Informáticos | 2023 – 2024 (v4.0 "S-81 Isaac Peral")



Contenido

	Bloque 1: Antes de empezar	3
	¿Por qué todo esto?.....	3
	Soporte de virtualización en la BIOS del host.....	4
	Usar GUI o no	4
	<i>Uso de varios terminales.....</i>	5
	<i>TMux: Terminales y paneles.....</i>	5
	<i>Midnight Commander: Explorador de archivos.....</i>	7
	<i>SSH a la máquina virtual.....</i>	7
	<i>Edición de archivos de texto</i>	7
	Conocimiento de Linux	8
	Bloque 2: Opciones de creación de la máquina virtual de la asignatura	11
	Opción 1: Importación de la imagen .ova suministrada (dificultad fácil)	11
	¿Por qué elegir este método?	11
	Usando el fichero .ova	11
	Resolución de problemas.....	13
	Opción 2: Usar Vagrant para construir automáticamente una máquina virtual (dificultad fácil-media)	15
	¿Por qué elegir este método?	15
	Cómo usar los archivos Vagrant.....	15
	Resolución de problemas.....	21
	Opción 3: Creación de tu propia máquina virtual (dificultad media, no se recomienda).....	23
	¿Por qué elegir este método?	23
	Configuración inicial de la plataforma de virtualización para instalar un servidor Linux	23
	Instalar un servidor Ubuntu Linux.....	31
	Primera actualización.....	39
	¿Instalar un GUI?	41
	Configuración de la máquina para mejorar la virtualización	42
	Software para instalar en la máquina virtual personalizada	43
	Resolución de problemas general (cualquier opción de creación de máquinas)	45
	Cuando arranco la MV, me da el error VERR_INTNET_FLT_IF_NOT_FOUND	45
	Si intento arrancar Firefox desde un cliente SSH me da un error	45



Bloque 3: Operaciones de gestión de MV generales	46
Clonación de máquinas	46
¿Qué es la clonación enlazada?.....	48
Creación de instantáneas de máquinas	48
Tomar, restaurar y eliminar instantáneas	49
Carpetas compartidas con el host	55
Redirección de puertos	58
Unattended-upgrades	59

BACK

1

2

3



Bloque 1: Antes de empezar

¿Por qué todo esto?

La crisis sanitaria del COVID-19 nos puso en una situación en la que la mayoría de nuestras actividades docentes tenían que realizarse online o en casa. Esto causó muchos problemas, y uno de los más importantes fue que debías usar el *hardware* de tu casa para ejecutar el *software* requerido para esta asignatura. Ahora que las clases ya son presenciales, sabemos que aún persiste un problema que fue de capital importancia durante la pandemia: **vuestras capacidades *hardware* pueden ser muy diferentes.**

Por tanto, hemos mantenido el rediseño "pandémico" de todos los laboratorios de la asignatura para disminuir drásticamente los requisitos de *hardware*: solo necesitarás usar **una sola máquina virtual** para ejecutarlos todos. Ahora usamos tecnologías de construcción y aprovisionamiento de infraestructuras modernas, aunque estudiarlas está fuera del alcance de esta asignatura. Para contrarrestar esto, **te proporcionamos *scripts* que te permiten trabajar con ellas sin necesidad de entender la tecnología subyacente** (a menos que tengas curiosidad, ya que todo está comentado en caso de que quieras aprender 😊)

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*. Eso sí, necesitas un mínimo para hacerlo:

- Cualquier CPU de 64 bits con extensiones de virtualización por *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 viable, 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 (1Gb podría ser viable para la mayoría de los laboratorios, pero no se ha probado a fondo). **Recomendamos encarecidamente no utilizar una GUI con 1Gb solamente.**

Si tu máquina tiene 8Gb o más, no debería ser problema.

- Un máximo de **80 Gb para el disco duro** de la máquina virtual

El disco de la máquina virtual es dinámico, solo ocupará espacio si es necesario. Alcanzar los 80 Gb es muy poco probable

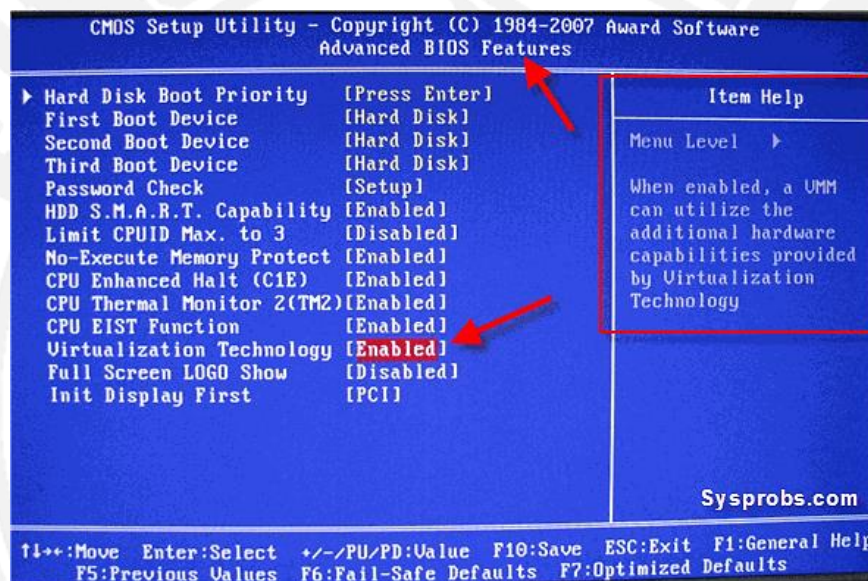
Te proporcionaremos **tres opciones** para crear la máquina, cada una con ventajas y desventajas: por favor, elige la que más te beneficie.

Soporte de virtualización en la BIOS del host

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 textos como "Intel Virtualization Technology", "Virtualization Technology", "Enable Virtualization Technology", "Enable VT-X" o "Enable AMD-V".

Necesitamos 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 2012 implementan esta tecnología.



Ejemplo de soporte en BIOS de tecnología de virtualización por hardware (Fuente:

<https://support.bluestacks.com/hc/es/articles/115003910391--C%C3%B3mo-puedo-habilitar-la-virtualizaci%C3%B3n-VT-en-mi-PC->)

Usar GUI o no

La máquina virtual que suministramos o la que construirás tiene un GUI ligero para facilitar tu trabajo en la asignatura. **Esta GUI es accesible escribiendo `startx` en la línea de comandos** (no arranca con GUI por defecto).

La decisión de usar una GUI o no es tuya: ahorraras tiempo, pero con un coste de recursos. Aunque hoy en día es raro no tener un hardware que pueda asumir el coste del GUI que recomendamos instalar

Si prefieres no utilizar una GUI (o tu hardware no te permite tener una) te damos una opción diferente: **usar funciones avanzadas de terminal**. Vamos a detallarlas más aquí, porque se pueden utilizar sin importar la opción que elijas, y la mayoría de estas opciones también **están presentes en los contenedores Docker que suministramos para cada laboratorio** (documento [Laboratorio 0B](#))

Uso de varios terminales

Incluso con servidores sin 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 concreta, 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 ssiuser 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, con integración de ratón (módulo **gpm**) y soporte de colores de terminal, en la máquina *Vagrant* (Opción 2), en la **.ova** (Opción 1) y en cada contenedor interactivo de los laboratorios.

Podría ser una buena idea usarlo si quieres más agilidad y productividad sin instalar o ir a 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.

<pre># 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*</pre>	<pre>[pungki@dev-machine ~]\$ ls backup-2013-12-06.tar.gz ntop Desktop ntop~ display.date ntopng Documents ntopng~ Downloads Pictures dump.rdb Public Mono and Gnome Do Templates Music Videos [pungki@dev-machine ~]\$ _ [pungki@dev-machine ~]\$</pre>
---	--

"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, los archivos de la máquina virtual **Vagrant** tienen 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**)

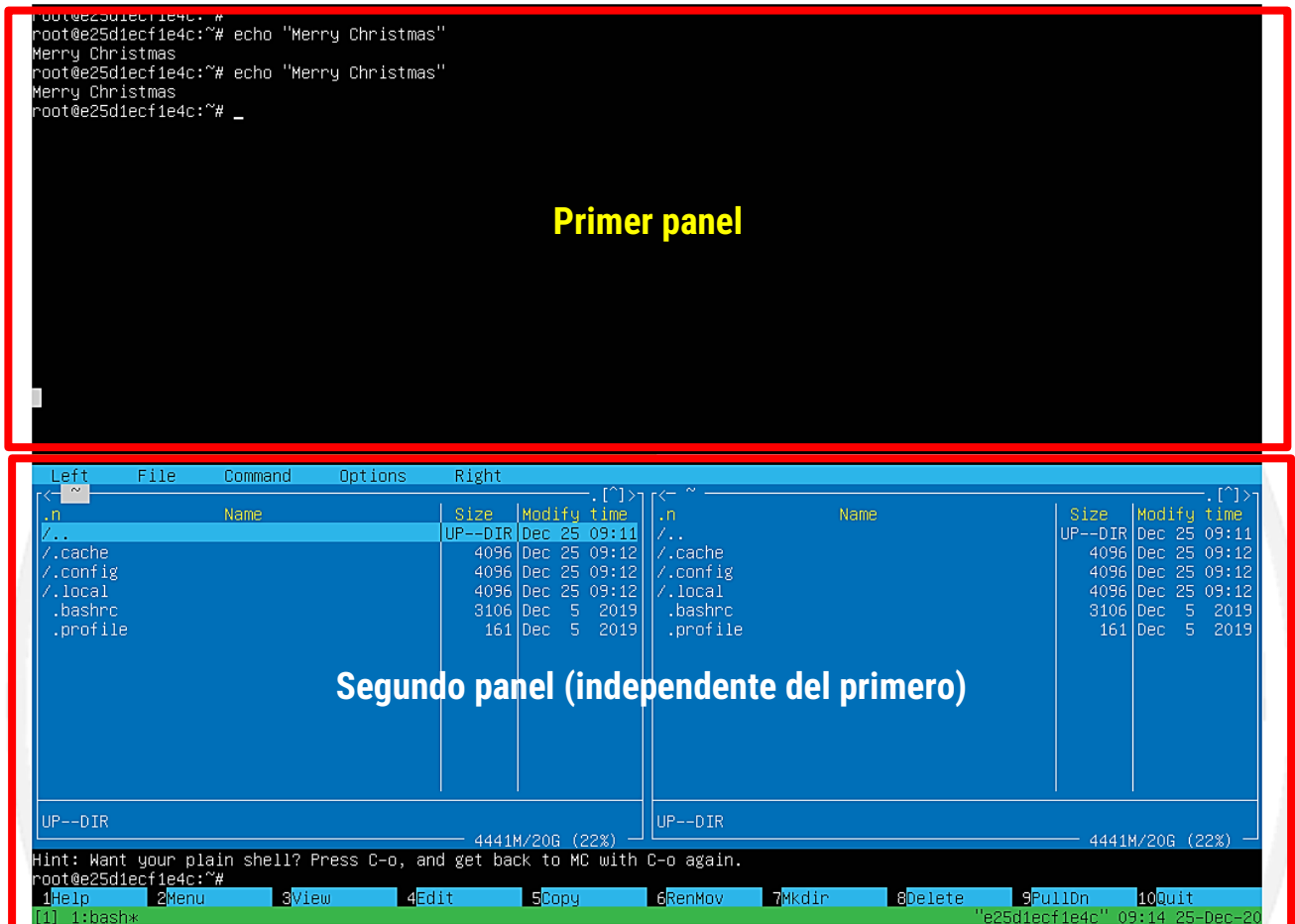
TMux 3.1 Cheatsheet (ingenieriainformatica.uniovi.es)	
Console enhancer and multi-pane support tool	
https://github.com/tmux/tmux/wiki	
GENERAL USAGE	
tmux	
NOTES	
Press Ctrl + B, release them, and then press the corresponding key exit on the last pane exits tmux	
OPTIONS	
Window handling	
New window: 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	
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 esta:

[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			
[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		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	

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 en el disco duro. Hemos incluido este **navegador de archivos basado en consola** muy popular para darte esta capacidad y aumentar tu productividad. Como muestra esta imagen, se puede integrar con **tmux** y tiene soporte para ratón. Esto **también se instala en todos los contenedores interactivos de Docker** que vamos a utilizar en la asignatura.



SSH a la máquina virtual

Si utilizas una GUI con nuestra máquina *Vagrant* o la **.ova**, 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. La otra opción es la **redirección de puertos**. Para saber cómo hacer ambas, consulta la [opción 3 de crear máquinas virtuales](#) y la [sección correspondiente](#) del bloque 3.

Las máquinas virtuales obtenidas de las opciones 1 y 2 asignan el puerto 2222 de tu PC al puerto ssh de la máquina virtual, por lo que si tu MobaXTerm / Putty / cualquier cliente SSH se conecta a 127.0.0.1:2222 se conectará realmente a la máquina virtual sin requerir más pasos.

Edición de archivos de texto

Deberás editar archivos en consola muchas veces, salvo si decides usar la GUI. Para facilitar el trabajo, **hemos 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 o como parte de los archivos *Vagrant*, si quieres usarla en otros cursos o para su uso personal. Las máquinas de las opciones 1 y 2 incluyen 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 esta asignatura. Sin embargo, no más de los que ya has adquirido en la asignatura de *Sistemas Operativos*. Sin embargo, sabemos que es posible que hayas olvidado algunos. No te preocupes, ya lo imaginábamos y hemos implementado un inicio lento (pero progresivo) en este tema. Para ayudarte te recomendamos lo siguiente:

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

- 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_mkhomedir (8) - PAM module to create users home directory
update-info-dir (8) - update or create index file from all installed info fi...
```

Otra forma de buscar el comando que necesitas es usar *cheatsheets*. Este es uno de los más completos que he encontrado en Internet (**Fuente:** <https://twitter.com/linuxopsys/status/1722438906157646127>):

LINUX COMMANDS CHEAT SHEET

System

uname	→	Displays system information: kernel version, machine type, and more.
uname -r	→	Displays the running Linux kernel's release version.
uptime	→	Shows current time, system uptime, users, and load averages.
hostname	→	Shows the system hostname
hostname -i	→	Displays the IP address of the current host.
last reboot	→	Shows last reboot times and durations in logs.
date	→	Displays the current date and time information.
timedatectl	→	Displays detailed system clock and time zone information.
cal	→	Displays a simple calendar of the current month.
w	→	Shows who is logged on and their activity.
whoami	→	Displays the username of the current user.
finger username	→	Displays information about a user named 'username'.

Hardware

dmesg	→	Displays messages from the kernel's ring buffer.
cat /proc/cpuinfo	→	Displays detailed information about the CPU.
cat /proc/meminfo	→	Displays detailed system memory usage information.
lshw	→	Lists detailed hardware configuration of the system.
lsblk	→	Lists information about all available block devices.
free -m	→	Shows system memory usage in megabytes.
lspci -tv	→	Displays PCI devices in tree format, verbosely.
lsusb -tv	→	Shows USB devices as a tree, verbosely.
dmidecode	→	Displays hardware information from system BIOS
hdparm -i /dev/sda	→	Displays information of disk /dev/sda.
badblocks -s /dev/sda	→	Checks /dev/sda for bad blocks, showing progress.

User Management

id	→	Displays the user's UID, GID, and groups.
last	→	Shows list of last logged-in users.
who	→	Displays who is currently logged in.
groupadd admin	→	Creates a new user group named admin.
adduser Sam	→	Creates a new user account named Sam.
userdel Sam	→	Deletes the user account named Sam.
usermod	→	Modifies properties of an existing user account.

File Permission

chmod 644 /data/test.c	→	Sets the permissions of the file /data/test.c to be read/write for the owner, and read-only for the group and others.
chmod 755 /dir1	→	Assigns read, write, and execute permissions to the owner, and read and execute permissions to the group and others for the directory /dir1.
chown bob:devops filename	→	Changes file 'filename' ownership to 'bob' and 'devops'.
chown ownername:groupname directory	→	Change owner and group of the directory.

Network

ip addr show	→	Displays all network interfaces and their information.
ip address add 192.168.0.1/24 dev eth0	→	Assigns IP address 192.168.0.1 to interface eth0.
ifconfig	→	Shows network interfaces and their configuration.
ping host	→	Sends ICMP packets, measures round-trip time to "host".
whois domain	→	Retrieves and displays domain's registration information.
dig domain	→	Queries DNS, provides domain's DNS information.
dig -x host	→	Resolves IP address to hostname, shows DNS information.
host google.com	→	Performs an IP lookup for the domain name
wget file_path	→	Downloads file from specified path.
netstat	→	Displays various network-related information and statistics.

Compression / Archives

tar -cf backup.tar /home/ubuntu	→	Creates a tar archive of /home/ubuntu directory.
tar -xf backup.tar	→	Extracts files from "backup.tar" archive.
tar -czvf backup.tar.gz /home/ubuntu	→	Creates compressed "backup.tar.gz" archive of "/home/ubuntu"
gzip file1	→	Compresses "file1" into "file1.gz", original is removed.

Install Packages

rpm -i pkg_name.rpm	→	Installs the package "pkg_name.rpm" using RPM Package Manager.
rpm -e pkg_name	→	Uninstalls the specified RPM package.
dnf install pkg_name	→	Installs the specified package using DNF.
pacman -S pkg_name	→	Installs the specified package using Pacman.

Directory Traverse

cd ..	→	Navigate to the parent directory.
cd	→	Changes the current directory to the user's home.
cd /mnt	→	Changes the current directory to "/mnt".

File Commands

ls -al	→	Lists all files, detailed information, in long format.
pwd	→	Displays the present working directory's path.
mkdir dir1	→	Creates a new directory named dir1.
rm file1	→	Deletes the file named file1.
rm -f file2	→	Forcefully deletes the file named file2.
rm -r dir1	→	Recursively removes directory dir1 and its contents.
rm -rf dir1	→	Forcefully deletes directory dir1 and its contents.
cp file1 file2	→	Copies file1, creating or overwriting file2.
cp -r dir1 dir2	→	Copies dir1 to dir2, including subdirectories.
mv file1 file2	→	Renames or moves file1 to file2.
ln -s /path/to/file_name link_name	→	Creates symbolic link named link_name to file_name.
touch file1	→	Creates an empty file named file1.
cat > file1	→	Creates/overwrites file1, awaiting standard input.
more file1	→	Displays file1 content, paginating through output.
head file1	→	Displays the first ten lines of file1.
tail file1	→	Displays the last ten lines of file1.
gpg -c file1	→	Encrypts file1 with symmetric cipher using passphrase.
gpg file2.gpg	→	Decrypts file2.gpg, prompting for the passphrase.
wc	→	Counts words, lines, and characters in files.
xargs	→	Executes commands with piped or file-provided arguments.

Process Related

ps	→	Displays a snapshot of current processes.
ps aux grep telnet	→	Displays running telnet processes with details.
mpmap	→	Shows memory map of a process.
top	→	Displays dynamic real-time view of running tasks.
kill 1234	→	Terminates the process with PID 1234.
killall proc	→	Kills all processes named 'proc'.
pkill process-name	→	Terminates processes with the name.
bg	→	Resumes suspended jobs in the background
fg	→	Brings a suspended job to foreground
fg n	→	Brings job number 'n' to foreground.
lsdf	→	Lists all open files and processes.
renice 19 PID	→	Changes priority of process with given PID.
pgrep firefox	→	Displays Process ID(s) for firefox processes.
pstree	→	Displays a tree of running processes.

Install Source (Compilation)

./configure	→	Checks system compatibility and generates makefile for software installation.
make	→	Compiles code by following instructions in the Makefile.
make install	→	Installs compiled code into specified system locations.

Search

grep pattern file	→	Search for a given pattern within the file.
grep -r pattern dir1	→	Recursively searches for the specified "pattern" within the "dir1" directory and its subdirectories
locate file	→	Finds files named "file" using prebuilt database.
find /home -name index	→	Searches "/home" directory for files named "index" recursively.
find /home -size +10000k	→	Finds files over 10000k size in /home directory.

Login

ssh user@hostname	→	Initiates SSH connection to specified hostname.
ssh -p port_number user@hostname	→	Initiates SSH connection using specific port.
Connect to the host via telnet default port 23	→	Securely connect to the system via SSH default port 22
telnet host	→	Connect to the host via telnet default port 23.

File Transfer

scp file.txt remoteuser@remote_host:/remote_directory	→	Copies file.txt to remote host's specified directory.
rsync -a /home/ubuntu /backup/	→	Synchronizes content from source directory to destination directory, preserving attributes.
rsync -a /var/www/web/user@remote_host:/backup/web_backup/	→	Synchronizes local directory to remote, preserving attributes.

Disk Usage

df -h	→	Displays human-readable disk space usage for all mounted filesystems.
df -i	→	Displays inode usage information for all mounted filesystems.
fdisk -l	→	Lists all partitions and their information on all drives.
du -sh /dir1	→	Displays summary of total disk usage size of /dir1, human-readable.
findmnt	→	Displays a list of all mounted filesystems and their properties.
mount device-path mount-point	→	Mounts the device at the specified filesystem mount point.

 Si necesitas refrescar algo más que comandos de Linux y o te acuerdas de determinados conceptos básicos, tienes una presentación adjunta al Laboratorio 1 que puedes usar para "resucitar" 😊 conceptos del curso de Sistemas Operativos





Bloque 2: Opciones de creación de la máquina virtual de la asignatura

↔ ON! Opción 1: Importación de la imagen .ova suministrada (dificultad fácil)

¿Por qué elegir este método?

Recomendado si: Tienes instalado *VirtualBox*, una conexión a *Internet* estable y no quieres probar todavía tecnologías *Infrastructure as Code* para ver lo que pueden hacer por ti en el futuro.

Ventajas

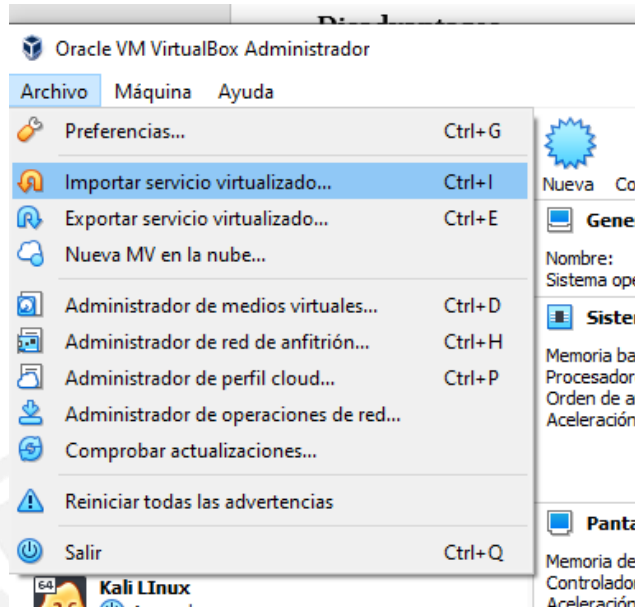
- MV preconfigurada lista para usar, con todos los paquetes instalados para hacer los laboratorios de la asignatura.
- Las infraestructuras de los laboratorios **ya están preparadas**. Todos tienen el `prepare_labXX.sh` hecho, por lo que solo tienes que ejecutar el `build_labXX.sh` y seguir los pasos para empezar a trabajar, como se indica en el documento **00B**.
- Configuración de la MV en un paso (**usuario:** `vagrant`, **contraseña:** `test123...` (tres puntos al final)).

Desventajas

- **Descarga grande** que puede no ser conveniente en ciertas situaciones.
- **Tienes que crear una carpeta compartida con el host manualmente** (ver [la última sección](#) de este documento)
- **Preparada 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:
 - **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).

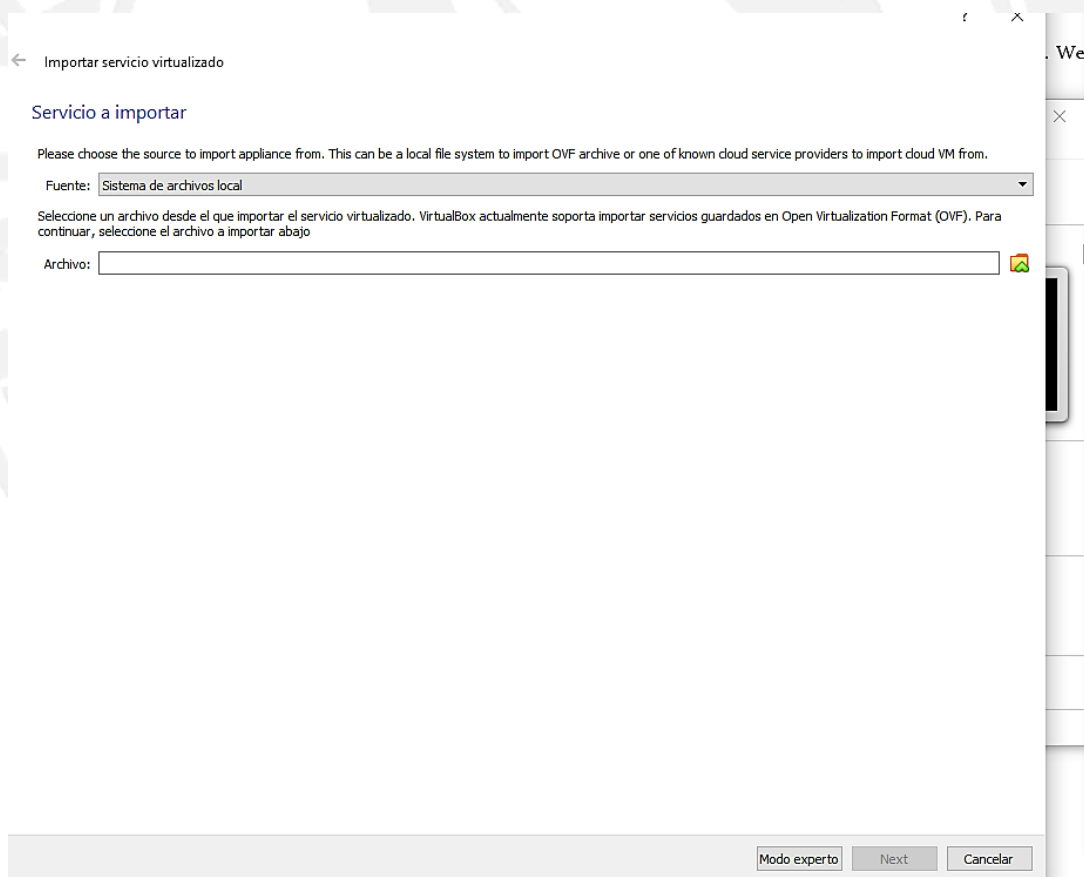
Usando el fichero .ova

Para facilitar tu trabajo durante la asignatura, te proporcionamos **una máquina virtual preconstruida**. Puedes descargarla desde una URL que te proporcionará tu profesor. La máquina es un **Ubuntu Server 22.04 LTS** basado en consola (con GUI **opcional**). 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. Tardará un tiempo. **Se recomienda usar un SSD** para reducir drásticamente los tiempos de importación, de arranque y la capacidad de respuesta general, pero no es algo necesario.



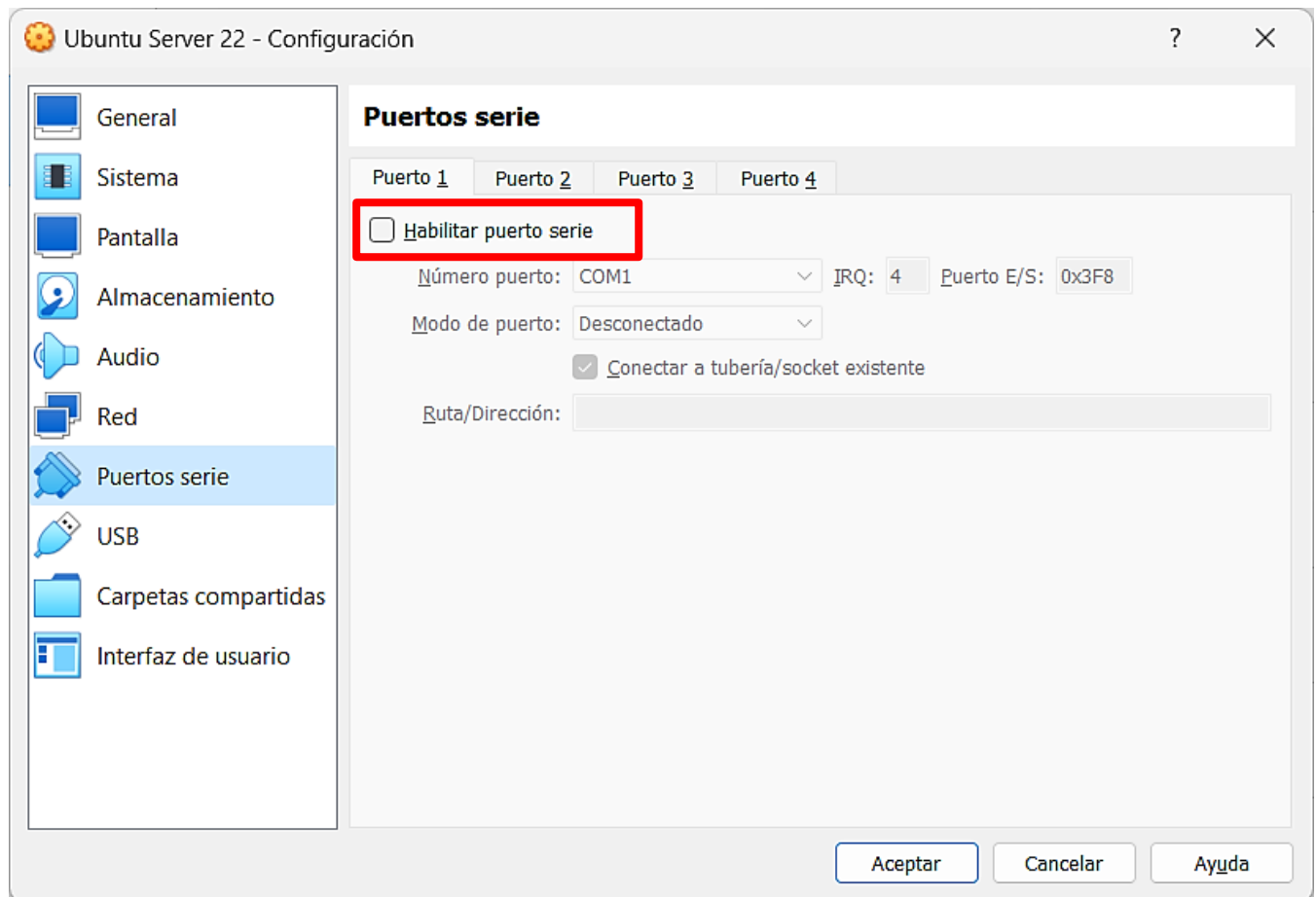
Selección del filtro de archivos .ova

Otras opciones para administrar la máquina virtual importada se detallarán en la **Opción 3**. Esta máquina es casi idéntica a la proporcionada a través de *Vagrant*.

Resolución de problemas

El .ova me da un error al cargarlo una vez importado

Se ha detectado que en algunas configuraciones de VirtualBox por alguna razón la instalación del puerto serie no funciona correctamente y da un error al arrancar la MV. Si es tu caso, asegúrate de **no marcar la opción de "Habilitar puerto serie"**:



No puedo conectarme vía SSH con el usuario y password correcto

En las últimas versiones de *Ubuntu* el login remoto vía SSH con contraseña viene desactivado por defecto, y solo se admite hacer login con un certificado ([Laboratorio 4](#)). Si quieres activar el login con usuario y password tradicional, asegúrate de descomentar esta línea y reiniciar el servicio SSH (o la MV):

```
GNU nano 6.2 /etc/ssh/sshd_config
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
KbdInteractiveAuthentication no

# Kerberos options
#KerberosAuthentication no

^G Ayuda      ^O Guardar    ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación
^X Salir      ^R Leer fich. ^\ Reemplazar  ^U Pegar      ^J Justificar ^_ Ir a línea
```

NOTA: Algunos de vosotros tenéis dentro de la carpeta `/etc/ssh/sshd_config.d` un fichero de configuración de SSH adicionales. Este habitualmente contiene una directiva que de nuevo anula la autenticación con passwords (hace referencia a las máquinas que se configuran en entornos de nube, que no la permiten). Borra este fichero, porque si no sobrescribirá este cambio y el sistema seguirá sin aceptar usuarios y passwords para hacer login por ssh.

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

Recomendado si: Tienes instalado *VirtualBox*, *Hyper-V* o *VMware Desktop*, quieres probar tecnologías *Infrastructure as Code* para ver lo que pueden hacer por ti en el futuro.

¿Por qué elegir este método?

Ventajas

- La descarga, configuración, creación y configuración de la máquina está **totalmente automatizada**, no se necesita interacción.
- Se ejecuta en cualquier SO compatible con *Vagrant* (pero requiere *VirtualBox*). *Vagrant* (y *VirtualBox*) están disponibles en todas las versiones modernas de *Windows*, *Linux* y *MacOS*.
- La configuración de la máquina se puede distribuir (e implementar en diferentes sistemas) en **archivos de texto** muy pequeños, siendo siempre la misma y pudiendo actualizarse fácilmente.
- Se crea automáticamente una carpeta compartida con el host en el mismo directorio donde se construye la máquina.
- Puedes usarla como base para hacer tus propias versiones de la MV 😊

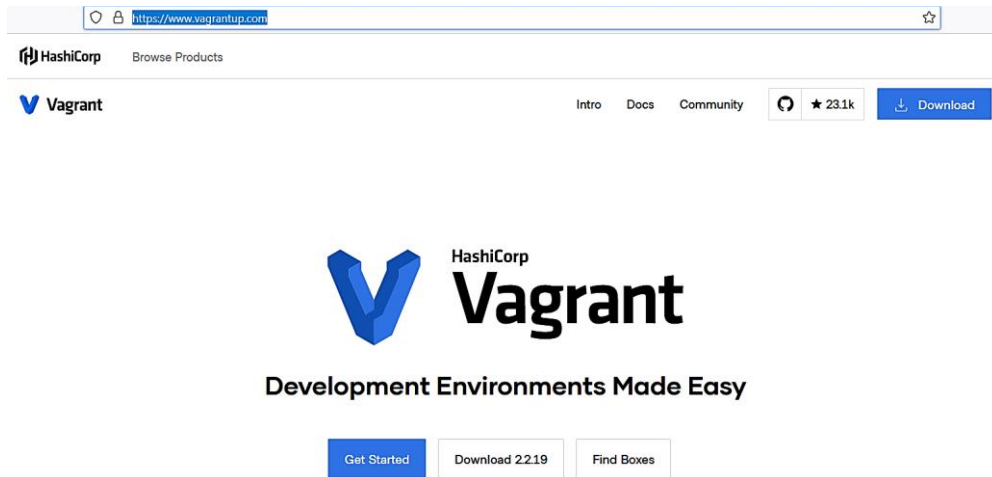
Desventajas

- **Es necesario usar *VirtualBox***, ya que es el sistema de virtualización soportado por la base *box Vagrant* que usamos.
- Necesita un **tiempo de inicialización grande** (aunque solo en su primera ejecución)
- **Requiere que instales software adicional en tu PC (*Vagrant*)**. Los PC de los laboratorios de nuestra escuela tienen *Vagrant* instalado.
- Los laboratorios se deben **descargar** (archivo .zip dentro del campus) **y preparar individualmente** dentro de la máquina posteriormente. No obstante, así solo ocuparás lo que realmente vayas usando (la opción 1 tiene todo preparado ya, pero por ese motivo ocupa tanto).
- Probablemente haya que **instalar un navegador**, salvo que desde *Ubuntu* se solucione el problema a la hora de instalar *Firefox* que estaba presente al menos en Enero de 2024. Se recomienda usar **Chrome** o **Brave** si la instalación de *Firefox* diese problemas.

Cómo usar los archivos *Vagrant*

Preparar la construcción de la MV

- Descarga el archivo **.zip** del campus virtual con los ficheros de *Vagrant*
- Asegúrate de que la plataforma de virtualización que vas a utilizar está correctamente instalada
- Instala la versión más moderna de *Vagrant* para tu SO:
https://developer.hashicorp.com/vagrant/install?product_intent=vagrant
 - Para *Ubuntu*, **usa el paquete descargado de la página web oficial**; los repositorios de Ubuntu tienen una versión vieja de *Vagrant* que podría dar algún error con los archivos proporcionados.



- Descomprime el archivo y copia su contenido a una carpeta que crees (por ejemplo, **ssiubuntu**). Estos contenidos son:

.vagrant	08/12/2021 12:49	Carpeta de archivos	
customization_files	28/12/2021 7:33	Carpeta de archivos	
customization_scripts	10/12/2021 9:55	Carpeta de archivos	
provisioners	08/12/2021 12:41	Carpeta de archivos	
shared	21/12/2021 13:41	Carpeta de archivos	
buildvm	13/11/2021 10:10	Archivo por lotes ...	1 KB
buildvm	13/11/2021 10:10	Archivo SH	1 KB
destroyvm	13/11/2021 10:10	Archivo por lotes ...	1 KB
destroyvm	13/11/2021 10:10	Archivo SH	1 KB
runvm	25/10/2021 9:54	Archivo por lotes ...	1 KB
runvm	25/10/2021 9:54	Archivo SH	1 KB
updatevm	13/11/2021 10:11	Archivo por lotes ...	1 KB
updatevm	13/11/2021 10:11	Archivo SH	1 KB
Vagrantfile	28/12/2021 7:41	Archivo	8 KB

- La **Vagrantfile**: Con toda la configuración de la VM que estás a punto de crear.
- Scripts de shell para administrar Vagrant** en forma de archivos *bash* estándar de *Linux*. También se proporciona una versión **.bat** compatible con *Windows*.
 - buildvm.sh**: Compila la máquina virtual a partir del archivo **Vagrantfile**. Ten en cuenta que **esto solo debe hacerse una vez**.
 - runvm.sh**: Ejecuta la máquina virtual construida por el *script* anterior desde la línea de comandos, si por alguna razón no se puede arrancar desde la GUI de *VirtualBox*.
 - updatevm.sh**: Si realizas cambios en la **Vagrantfile** (por ejemplo, instalas más paquetes por defecto, aumentas la RAM disponible, el número de núcleos...) este *script* actualizará la configuración de la VM sin realizar una reconstrucción completa (mucho más rápido). Ten en cuenta que no necesitas hacer eso en la asignatura, solo se da por si acaso.
 - destroyvm.sh**: Elimina una máquina virtual previamente compilada, liberando todo el espacio y perdiendo todos los cambios y los resultados de sus operaciones. Si deseas usar la máquina virtual otra vez, debes compilarla (**buildvm.sh**) de nuevo.
 - El resto de los archivos y carpetas son utilizados por los anteriores y no necesitas entenderlos para seguir esta asignatura (aunque si tienes curiosidad... están comentados 😊)

- Una vez que hayas descomprimido todos los archivos, vete a la subcarpeta **customization_files** y localiza el archivo **your_username.txt**. Se creará una cuenta para este usuario (contraseña **test123...**) en la máquina virtual. Existirá un usuario adicional **vagrant** (también con contraseña **test123...**) en la máquina virtual que se creará.
- Ahora puedes continuar con la compilación de la máquina virtual. Sin embargo, si tu PC tiene suficiente potencia, **puedes aumentar los recursos de la máquina virtual** para que funcione de manera más eficiente. Abre el **Vagrantfile** y localiza esta sección en el archivo. 2 núcleos y 2Gb de RAM son los requisitos mínimos, pero si puedes darle al menos 4Gb de RAM notarás la diferencia.

```
## Machine Characteristics:
## -----

# In Mb, if you can afford it, more memory is always welcome :). Be careful not to push this too hard, and turn yourself into a
Chrome browser :P
RAM = 2048

# If you can afford it, more cores are always welcome :). Be careful not to push this too hard, and let your host breathe :P
CORES = 2

# Size of the box disk
DISK_SIZE = "80GB"

# Amount of available VRAM. Consider if your VM is going to have a GUI or not to choose this.
VRAM = 128
```

Si te da por explorar la **Vagrantfile**, hay una serie de características avanzadas que vienen comentadas, porque son incompatibles con la instalación de Hyper-V que necesitáis para otra asignatura. Puedes decidir activarlas, ya que suponen una ganancia de rendimiento considerable y es posible que en tu caso pudieran funcionar si no tienes que cursar nada que use Hyper-V.

No obstante, si lo haces, que sepas que una de las características avanzadas de esta **Vagrantfile**, **PageFusion**, **puede no ser compatible con los hosts MAC**. Por favor, si utilizas un MAC para crear la máquina virtual con Vagrant y decides usar estas características, abre el **Vagrantfile** y comenta o elimina la línea que tiene la opción **modifyvm** con el parámetro **--pagefusion**. No debería ser necesario ningún otro cambio.

Vagrantfile: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
# Use VBoxManage to customize the VM for high-performance options and other things
vb.customize ["modifyvm", :id, "--acpi", "on"]
vb.customize ["modifyvm", :id, "--accelerate3d", VIDEO_3D]
vb.customize ["modifyvm", :id, "--apic", "on"]
vb.customize ["modifyvm", :id, "--biosapic", "x2apic"]
vb.customize ["modifyvm", :id, "--cpu-profile", "host"]
vb.customize ["modifyvm", :id, "--graphicscontroller", "vmsvga"]
vb.customize ["modifyvm", :id, "--hpet", "on"]
vb.customize ["modifyvm", :id, "--hwvrtex", "on"]
vb.customize ["modifyvm", :id, "--largepages", "on"]
vb.customize ["modifyvm", :id, "--longmode", "on"]
vb.customize ["modifyvm", :id, "--nestedpaging", "on"]
vb.customize ["modifyvm", :id, "--pae", "on"]
# If you have a MAC host and get an error, comment this line, as this feature is not currently supported on MAC (yet)
vb.customize ["modifyvm", :id, "--pagefusion", "on"]
vb.customize ["modifyvm", :id, "--rtcuseutc", "on"]
vb.customize ["modifyvm", :id, "--spec-ctrl", "off"]
vb.customize ["modifyvm", :id, "--x2apic", "on"]
vb.customize ["modifyvm", :id, "--vram", VRAM]
vb.customize ["modifyvm", :id, "--vtxux", "on"]
vb.customize ["modifyvm", :id, "--vtxvpid", "on"]
vb.customize ["modifyvm", :id, "--vrde", VRDE]
vb.customize ["storagectl", :id, "--name", HD_CONTROLLER, "--hostiocache", "on"]

# Shared folder (host path, guest path)
config.vm.synced_folder "./shared", "/host_data", create: true, automount: true
end
```

Proceso de creación de VM

Una vez que se hayan completado estos preparativos, solo necesitas ejecutar **buildvm.bat** (o **buildvm.sh** si no estás en un sistema operativo *Windows*) y esperar a que se cree la máquina virtual.

Simbolo del sistema - buildvm.bat

```
C:\Users\Redondo\Documents\SSI_VM_2022>buildvm.bat

C:\Users\Redondo\Documents\SSI_VM_2022>vagrant up
==> vagrant: You have requested to enabled the experimental flag with the following features:
==> vagrant:
==> vagrant: Features: disks
==> vagrant:
==> vagrant: Please use with caution, as some of the features may not be fully
==> vagrant: functional yet.
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'hashicorp/bionic64'...
```

Asegúrate de tener conexión de red durante todo el proceso. ¡No abras la GUI de VirtualBox mientras esto está funcionando!

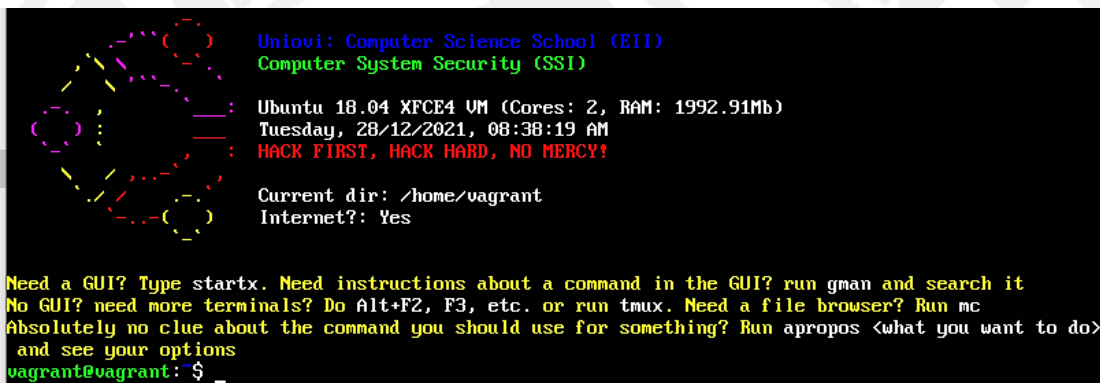
La máquina virtual se arrancará casi de inmediato, **pero espera a que finalice el script y no inicies sesión aún** (la máquina virtual se reiniciará automáticamente cuando termine del todo). Ya podrías usar la máquina virtual en cuanto apareciese, pero carece de todas las características que necesitas. **La máquina virtual se actualiza en segundo plano**, instalando todo el *software* y las actualizaciones necesarias mientras la usas, pero como necesita reiniciarse al final, puedes perder lo que hayas hecho. Además, *Docker*, *Docker-Compose* y otro *software* necesario no estarán disponibles a menos que dejes que finalice el proceso de construcción.

```
Simbolo del sistema - buildvm.bat
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'hashicorp/bionic64'...
==> default: Matching MAC address for NAT networking...
==> default: Setting the name of the VM: Ubuntu SSI Image 2022e
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
default: Adapter 1: nat
default: Adapter 2: hostonly
==> default: Forwarding ports...
default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Configuring storage mediums...
default: Disk 'vagrant_primary' needs to be resized. Resizing disk...
==> default: Running 'pre-boot' VM customizations...
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
default: SSH address: 127.0.0.1:2222
default: SSH username: vagrant
default: SSH auth method: private key
default: Vagrant insecure key detected. Vagrant will automatically replace
default: this with a newly generated keypair for better security.
default:
default: Inserting generated public key within guest...
default: Removing insecure key from the guest if it's present...
default: Key inserted! Disconnecting and reconnecting using new SSH key...
==> default: Machine booted and ready!
==> default: Configuring and enabling network interfaces...
default: Mounting shared folders...
default: /host_data => C:/Users/Redondo/Documents/SSI_VM_2022/shared

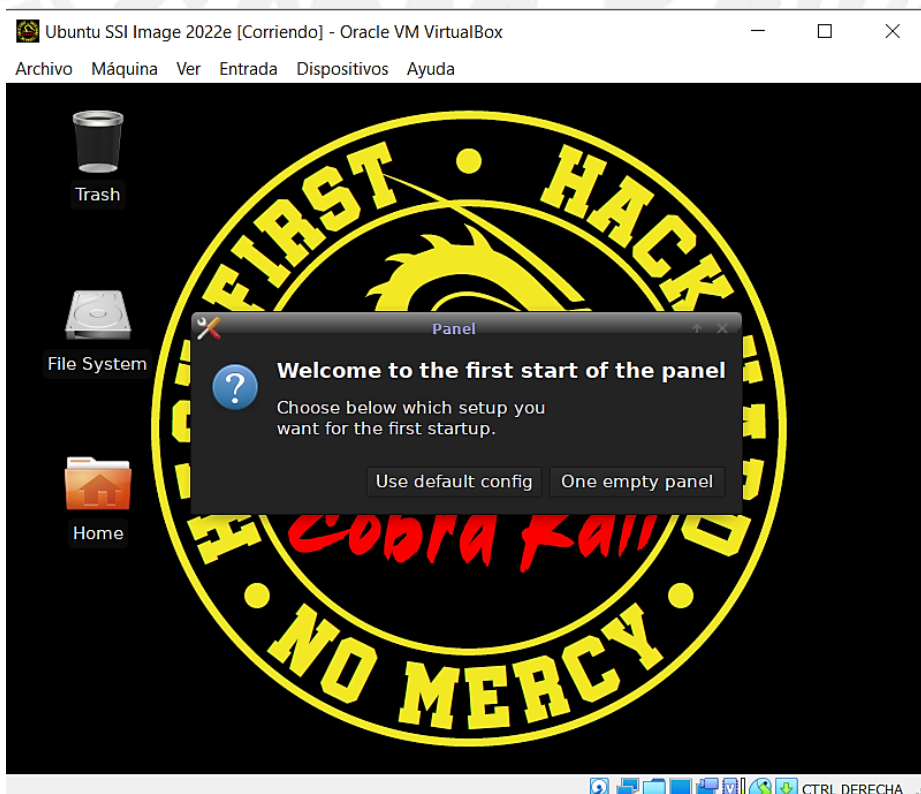
Ubuntu SSI Image 2022e [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Ubuntu 18.04.3 LTS vagrant tty1
vagrant login:
```

Una vez que finaliza el *script*, la máquina se reinicia y verás la pantalla de inicio de sesión. Ahora puedes usar la máquina virtual con tu usuario (**ssiuser** si no lo has cambiado) o con el usuario **vagrant** (ambos tienen la misma contraseña **test123...**).

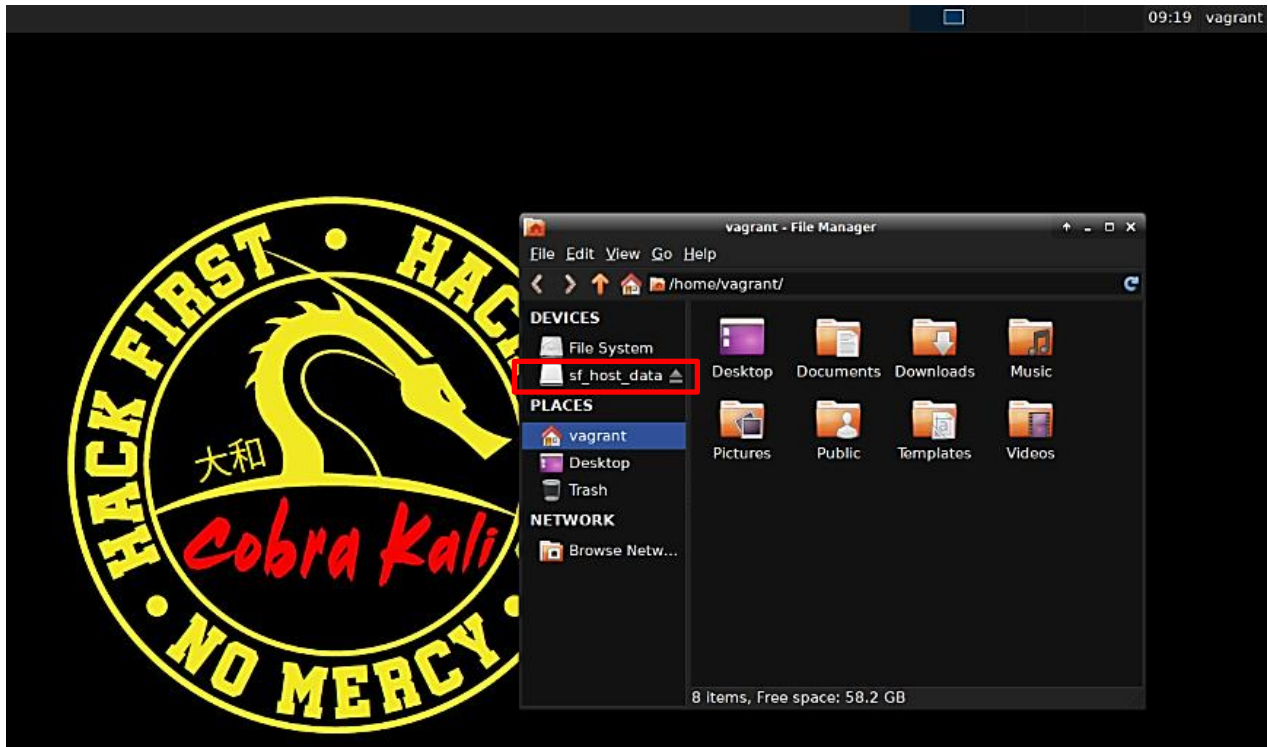
Si se inicia sesión con éxito verás esta pantalla de bienvenida con instrucciones básicas (un recordatorio de lo que dijimos en la primera sección de este documento).



Si quieres utilizar una GUI, ejecuta **startx** como se indica en la pantalla de bienvenida y **acepta la configuración predeterminada** (asegúrate de marcar **"Use default config"**). Ahora puedes maximizar la pantalla para usar la máquina virtual más fácilmente:



Si ahora abres un explorador de archivos, verás que hay una carpeta especial llamada **sf_host_data**:



Todo lo que escribas aquí se escribirá en esta subcarpeta de la ruta donde descomprimiste los archivos de la VM en tu PC (¡y viceversa!). ¡Esto hace que la transferencia de archivos entre la máquina virtual y tu PC sea muy fácil!

Este equipo > Documentos > SSI_VM_2022 > **shared**

Nombre	Fecha de modificación
ssi_labs	21/12/2021 13:38

Como verás, los materiales de los laboratorios ya están presentes allí, y debes seguir las instrucciones del próximo documento "**Lab 00B. Infraestructuras Automatizadas de SSI**" para saber cómo usarlos. Además, ten en cuenta para arrancar la máquina virtual de nuevo no necesitas seguir este proceso otra vez. ¡La máquina virtual aparecerá en la GUI de tu sistema de virtualización como una "normal"! También puedes tener acceso SSH desde el host a la máquina virtual en ejecución con **ssh vagrant@127.0.0.1 -p 2222**.



Resolución de problemas

Error en el arranque de Vagrant

Si en el arranque inicial de la máquina obtienes este error:

The VirtualBox VM was created with a user that doesn't match the current user running Vagrant. VirtualBox requires that the same user be used to manage the VM that was created. Please re-run Vagrant with that user. This is not a Vagrant issue.

The UID used to create the VM was: X Your UID is: Y

Borra el directorio **.vagrant** que aparece en el zip descomprimido y vuelve a lanzar el proceso¹. Si ya habéis intentado desplegar la máquina con esa carpeta creada, además tenéis que asegurarnos que tampoco exista la carpeta **C:/Users/<NombreUsuario>.vagrant.d/** donde **<NombreUsuario>** es el nombre de la cuenta que usáis en vuestro equipo.

Estás ejecutando VirtualBox, pero el Vagrantfile proporcionado da error durante la compilación de la máquina

Si tu sistema no puede ejecutar la **Vagrantfile** debido a errores relacionados con la configuración de **VirtualBox**, puedes hacer estos pasos:

- Asegúrate de tener instalada la última versión de **Vagrant**.
- **Actualiza tu VirtualBox** a la versión más moderna. El año pasado esto resolvió casi el 100% de los problemas 😊.
- Si el error muestra la opción **VirtualBox** que da el error, puedes abrir la **Vagrantfile** y comentar solo esta opción, intentando compilar la máquina virtual nuevamente.
- Si esto tampoco funciona, te recomendamos que sigas las otras opciones para la creación de la máquina virtual.

¹ Gracias Raúl Mínguez Rodríguez por informar de este problema y aportar la solución

- Hay una incompatibilidad conocida entre el **antivirus Kaspersky** y una de las últimas versiones de *Vagrant*, ya que la función de monitorización de conexiones SSL del antivirus interfiere con la conexión a la *Vagrant Cloud* para extraer las imágenes de la MV base y da un error de validación de certificados. La única solución conocida es crear una excepción en el antivirus para la dirección de *Vagrant* que muestra el error, o usar la opción 1. Este error se espera que se solucione en las siguientes versiones de *Vagrant*.
- Algunas instalaciones de *VirtualBox* que sean actualizaciones de versiones anteriores pueden dar problemas configurando interfaces de red *host-only*. En casos raros, puedes ver este error en *Vagrant*:

```
There was an error while executing `VBoxManage`, a CLI used by Vagrant
for controlling VirtualBox. The command and stderr is shown below.
```

```
Command: ["startvm", "d04f893a-4165-426f-b94c-3599a1471dc6", "--type", "gui"]
```

```
Stderr: VBoxManage.exe: error: Failed to open/create the internal network
'HostInterfaceNetworking-VirtualBox Host-Only Ethernet Adapter'
(VERR_INTNET_FLT_IF_NOT_FOUND).
VBoxManage.exe: error: Failed to attach the network LUN (VERR_INTNET_FLT_IF_NOT_FOUND)
VBoxManage.exe: error: Details: code E_FAIL (0x80004005), component ConsoleWrap,
interface IConsole
```

La única solución conocida es abrir la **Vagrantfile** y quitar esta línea:

```
config.vm.network "private_network", ip: "192.168.56.10", adapter: 2
```

Aunque pierdas la interfaz de red *host-only* al hacer esto, recuerda que puedes conectarte a la MV usando SSH de todas formas haciendo **ssh localhost:2222**.

No puedo usar VirtualBox

Para otros sistemas de virtualización, como *VMWare*, *Parallels*, *libvirt*, etc. o MAC con tecnología ARM, tiene estas opciones:

- Busca un **.ova** adecuado aquí: <https://cloud-images.ubuntu.com/jammy/current/> y ejecútalo para tener una instalación base compatible. Puedes instalar los paquetes necesarios más adelante, siguiendo [estas instrucciones](#). Los tutoriales de imágenes y la documentación de *Ubuntu Cloud* se pueden encontrar aquí: <https://ubuntu.com/server/docs/tutorials>
- Busque en **Vagrant Cloud** una *Vagrant base box* de **Ubuntu 22** (*Ubuntu Jammy*) que sea compatible con tu hipervisor. Por favor, **usa solo las oficiales**, no confíes en boxes creadas por personas desconocidas, ya que puede contener *malware*
- Cree la máquina virtual tú mismo con la [opción 3](#)

El usuario por defecto no me funciona

Entre los ficheros distribuidos con *Vagrant*, hay un fichero **user.txt** donde aparece el nombre del usuario que se puede usar con la máquina creada con *Vagrant*. Por error se ha puesto un nombre distinto al indicado, **ckuser**. La contraseña sigue siendo la indicada en el documento.

Opción 3: Creación de tu propia máquina virtual (dificultad media, 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** y 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: **ssuser**, 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 inicial de la plataforma de virtualización para instalar un servidor Linux

Información general sobre el hardware virtual

La máquina virtual preconstruída 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

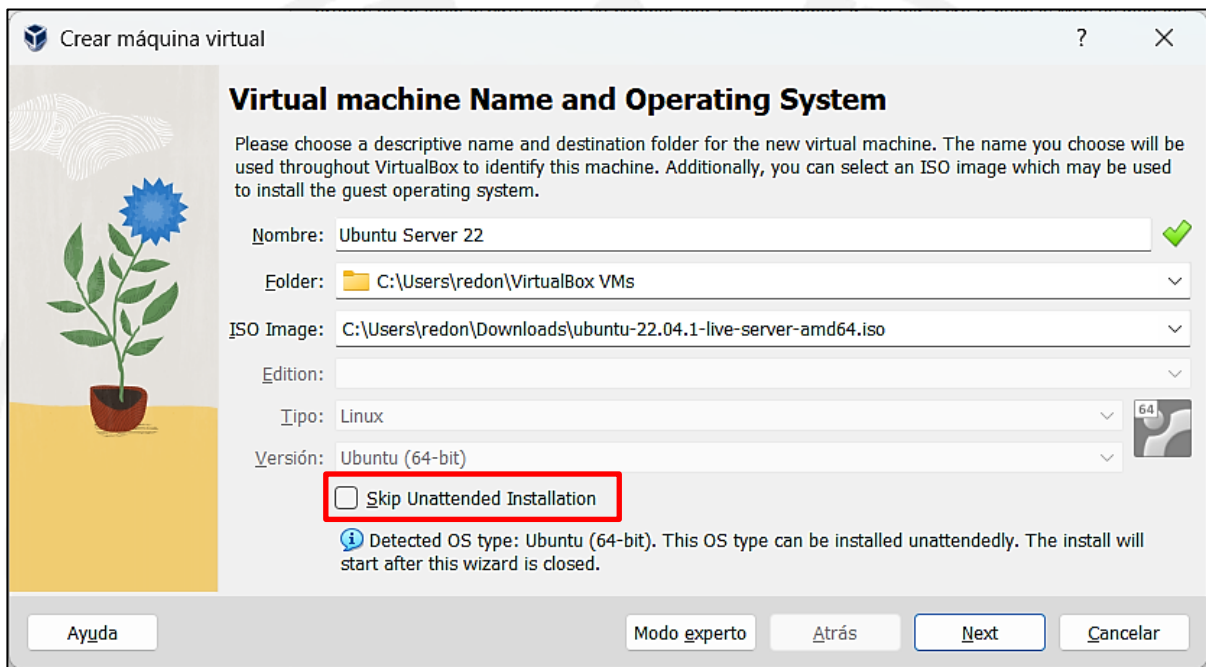
- **2 Gb de RAM** (o más si puedes permitirte 😊)
- **2 núcleos de CPU** (1 también podría ser posible, pero a un coste de rendimiento grande)
- **64-128Mb de memoria de vídeo** (máximo) con aceleración 3D habilitada si vas a usar GUI. Si no, basta con 16Mb.
- **80Gb de HD** con almacenamiento dinámico. 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 [en esta sección](#) 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 22.04 LTS de 64 bits** que puedes descargar desde aquí: <https://ubuntu.com/download/server>



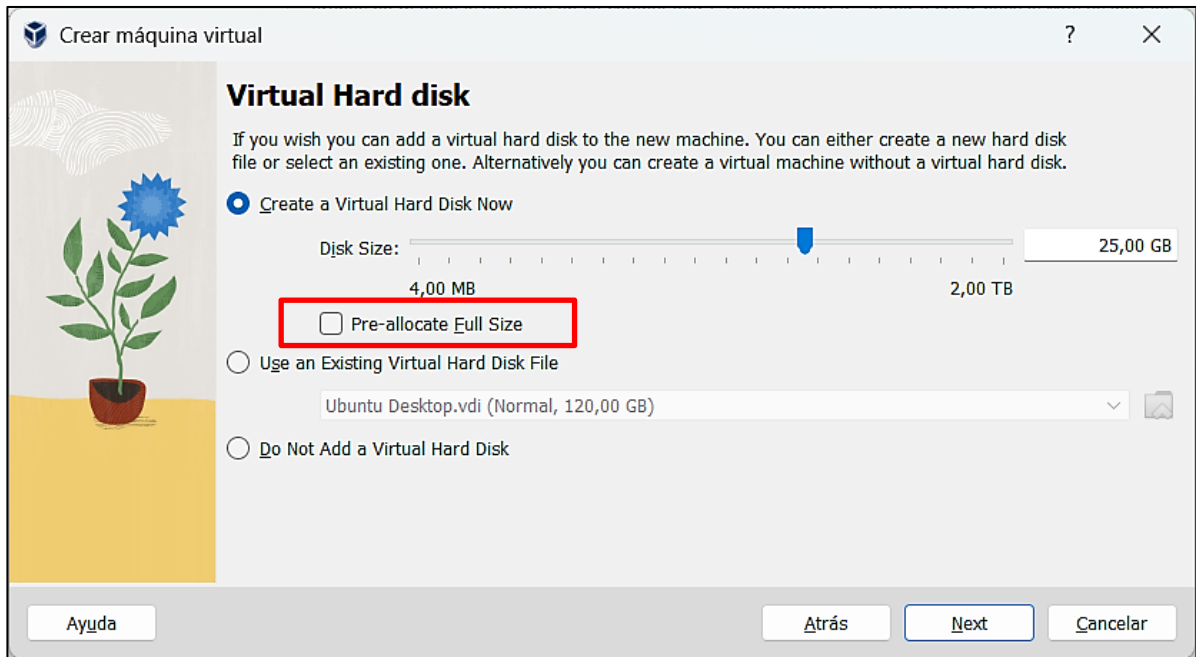
Tipo de máquina virtual a crear y RAM

En este punto vemos como aparece la opción **"Skip unattended installation"**. Desde la versión 7.0 VirtualBox permite la instalación completamente automatizada de determinados tipos de sistemas operativos, siendo Ubuntu Linux uno de ellos.

Dejar esta opción como está es muy recomendable si no tienes conocimientos técnicos, puesto que la máquina se instalará todo con opciones por defecto y no tendrás que configurar casi nada. El principal problema que podría tener es que puede que alguna configuración por defecto, como el idioma del teclado, no sea la que más nos conviene. Obviamente, esta opción no nos interesa en este apartado.

Le damos a "Next" y procedemos ahora a darle la RAM, la CPU y el espacio en el HD recomendados anteriormente.

Como hemos dicho, para ser más eficiente lo crearemos dinámico (no marcamos "Preallocate full size")

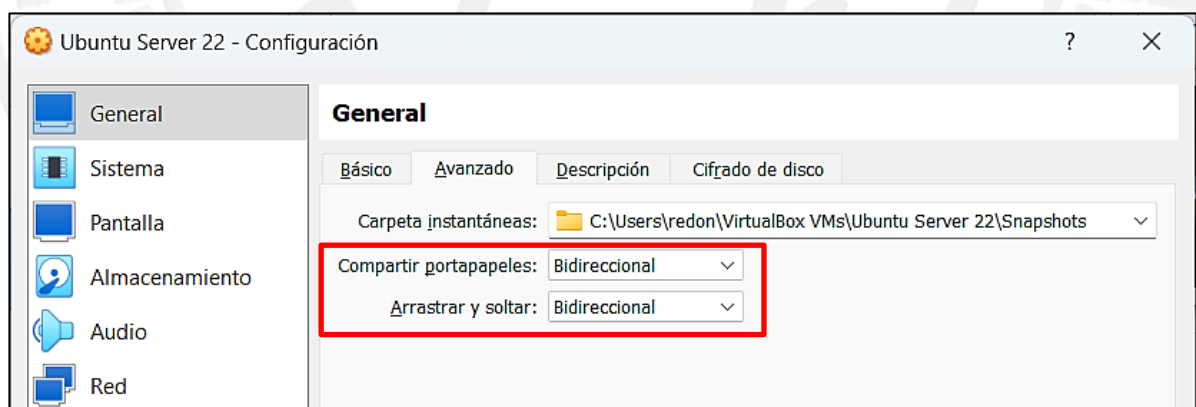


Disco duro de la máquina virtual

Una vez hecho esto, y pulsando en **"Terminar"** cuando se muestre la **pantalla de Resumen**, se creará la nueva plantilla para la máquina, y podremos mejorar su configuración editándola luego seleccionándola en la lista de máquinas y pulsando en **"Configuración"**.

Personalizar la configuración de la máquina virtual creada

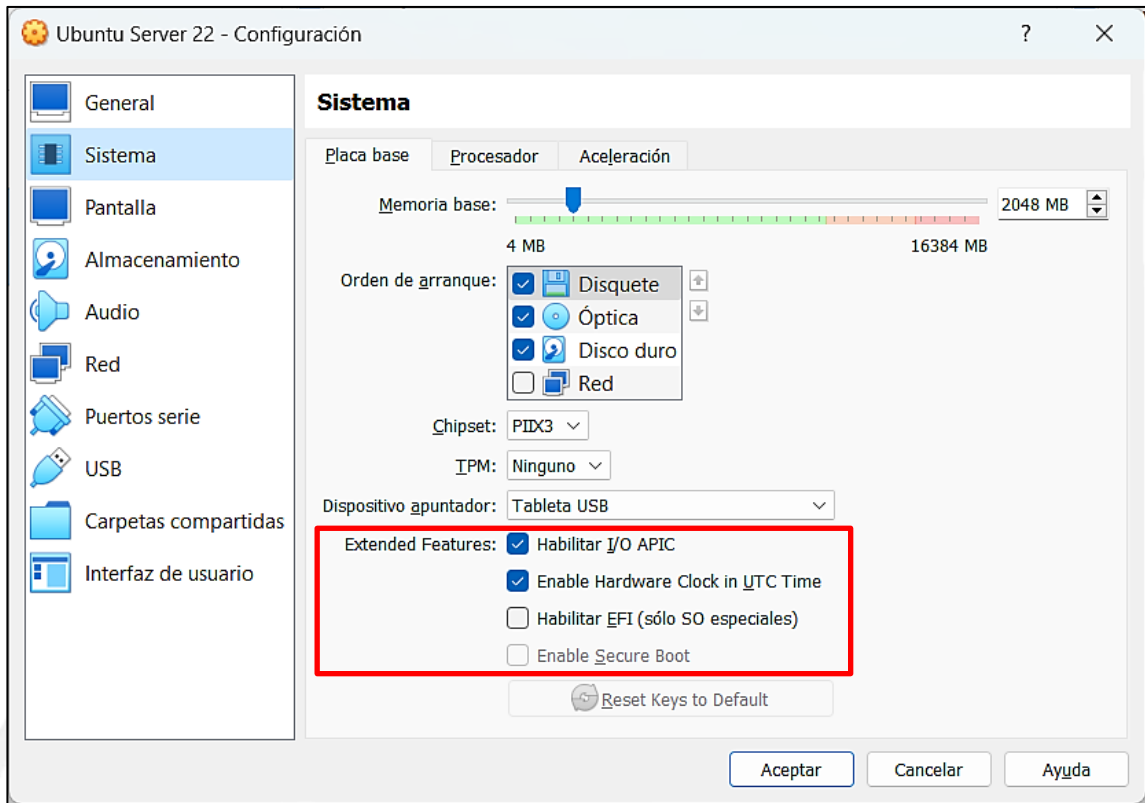
El primer paso es ir a la categoría **General**, donde podemos habilitar el **Portapapeles** y la capacidad de arrastrar y soltar entre el host y la máquina virtual y viceversa (**Bidireccional**). Sin embargo, **esto solo será efectivo si tenemos una GUI**.



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.

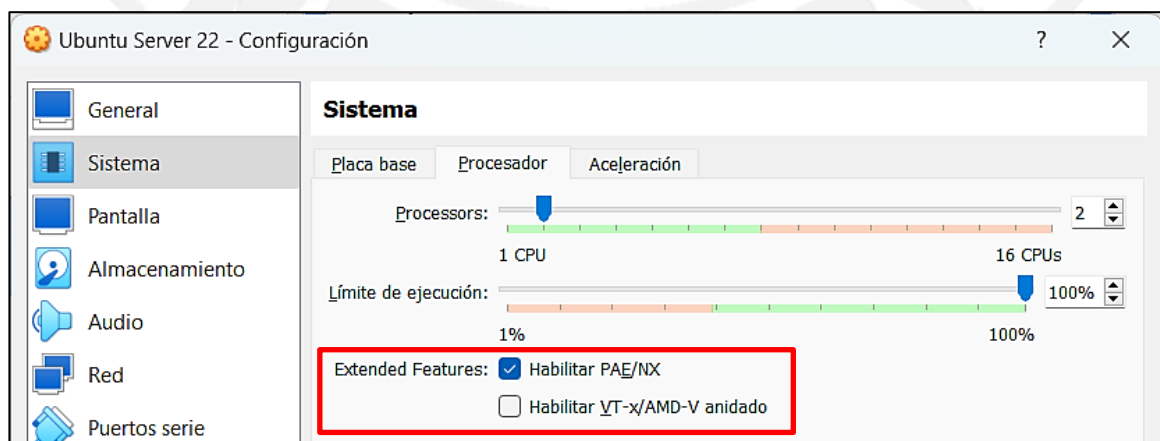


Categoría "Sistema" de una máquina virtual VirtualBox

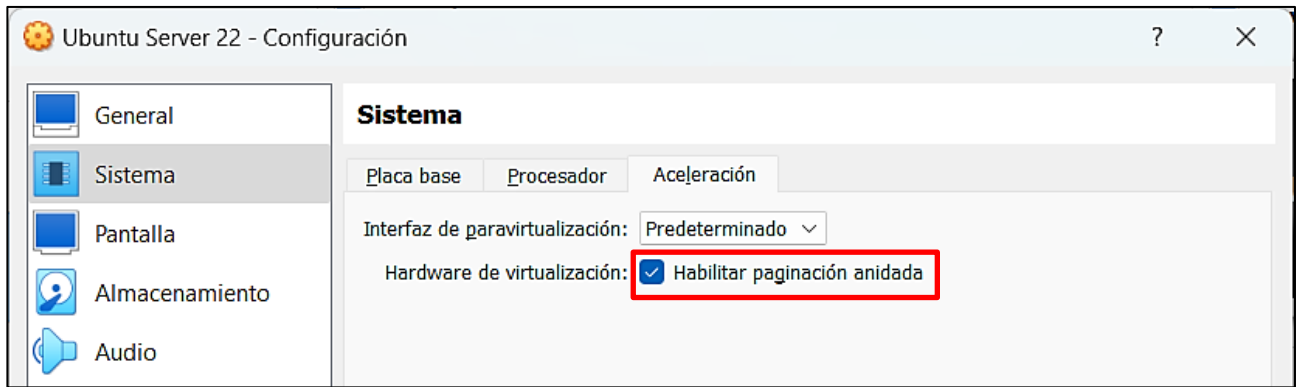
Luego asignaremos **2 CPUs**, procederemos a **activar PAE/NX** (pestaña "Procesador"), los conjuntos de instrucciones de la CPU **VT-X o AMD-V** y las funciones de **paginación anidadas** (si están disponibles en nuestro *hardware* base) para un mejor rendimiento de la virtualización (pestaña "**Aceleración**").

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):

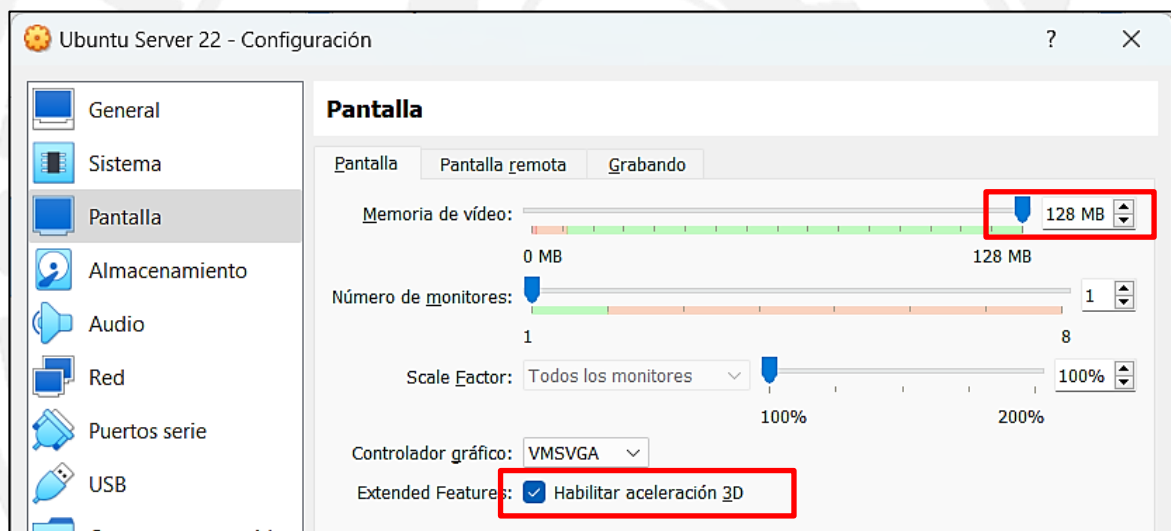


N.º de CPU de la máquina a crear



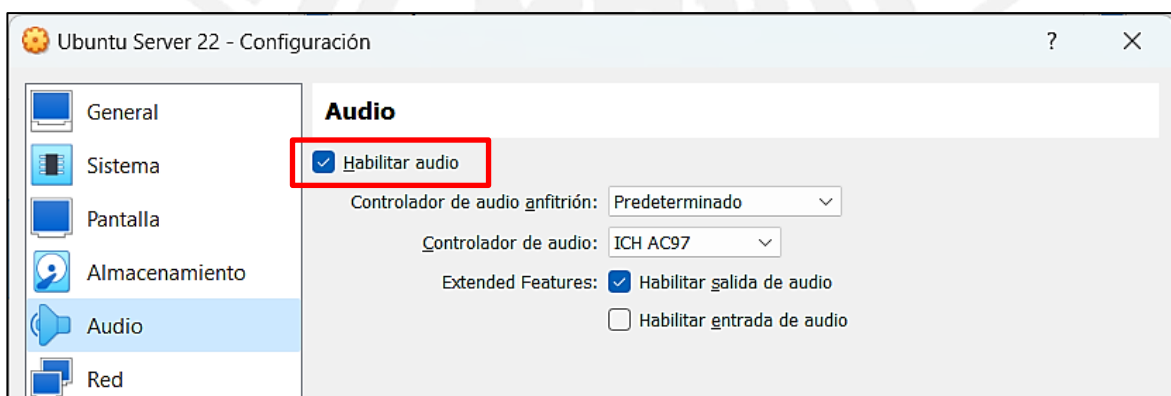
Aceleración para la máquina virtual

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 podemos optar por deshabilitarlo y, por lo tanto, no instalar controladores o servicios adicionales, o dejarlo como está (no es importante).



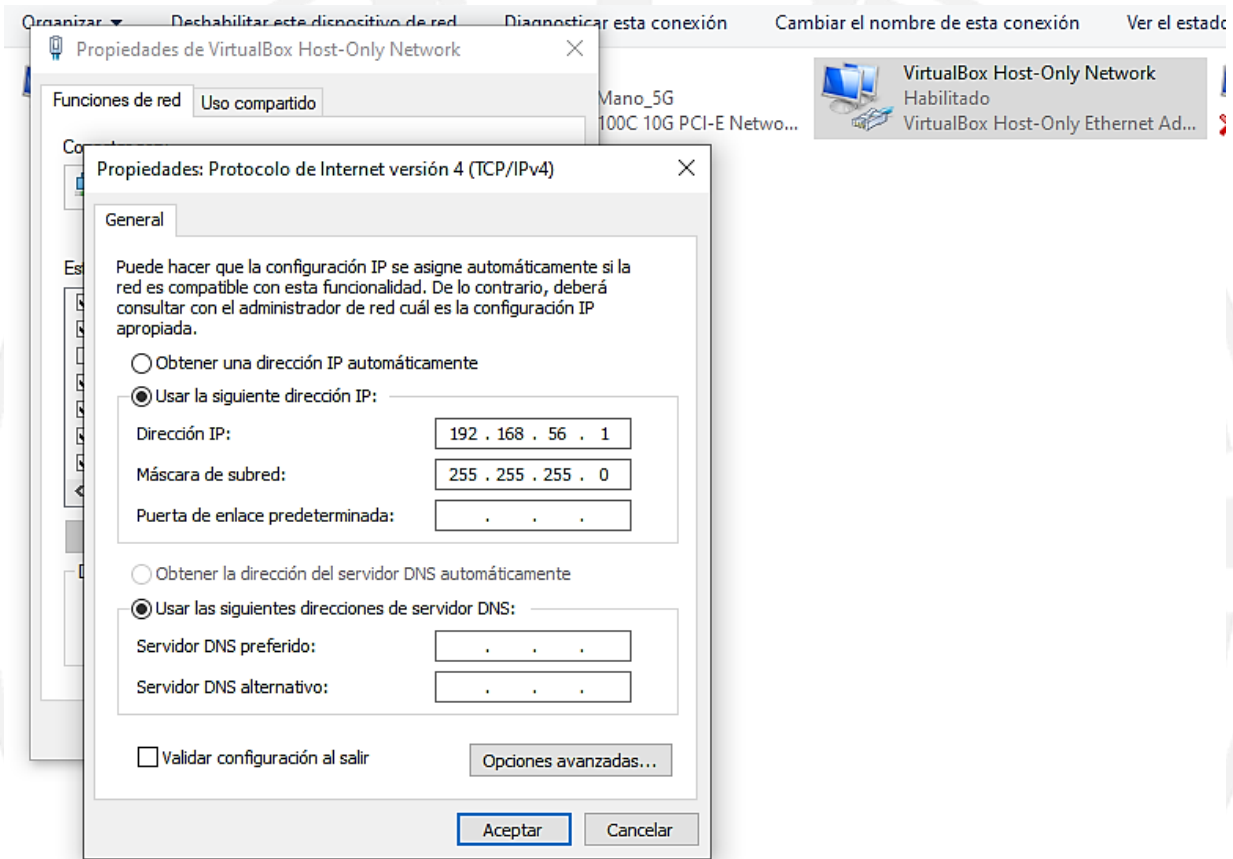
Configuración de audio

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:

VirtualBox 7 ha incorporado algunos tipos más, pero no son interesantes para este curso

- **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**, como si fuera un PC físico más conectado a la red. 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.

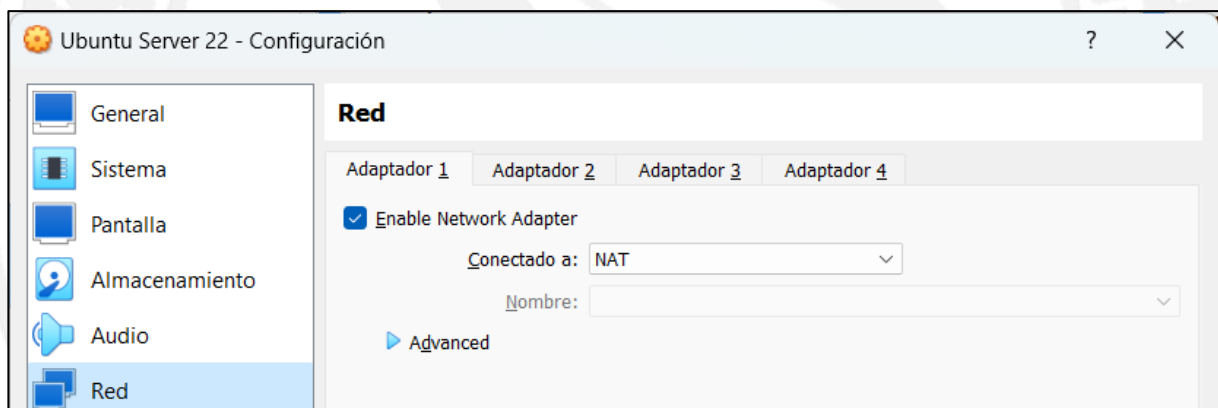
El uso de este modo en laboratorios de la escuela puede causar problemas, ya que el grupo de IPs disponible por DHCP para un laboratorio puede agotarse si todos usasen una red puente. Si no se usa DHCP, habría que asignar manualmente a esta interfaz una IP en la misma red que el host.

- **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.

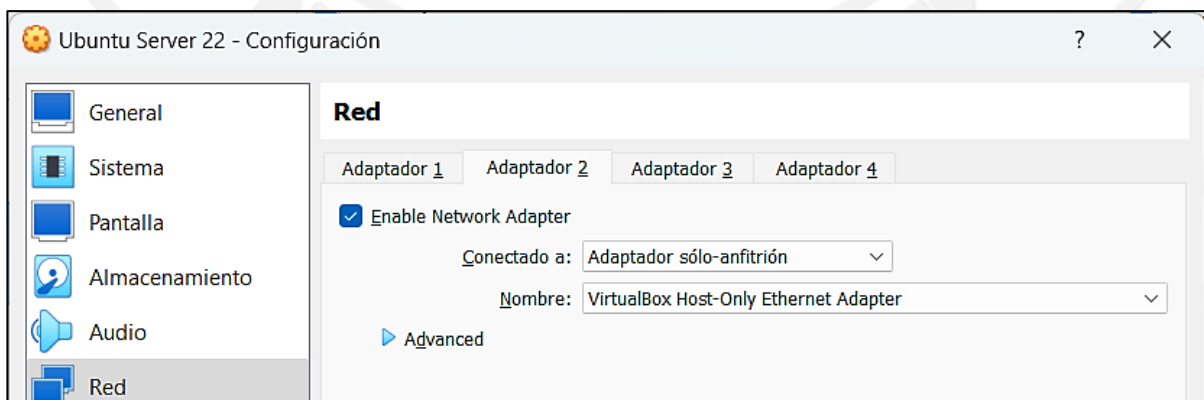
La máquina *Linux* a crear debería tener dos tarjetas de red:

- **Adaptador 1: 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

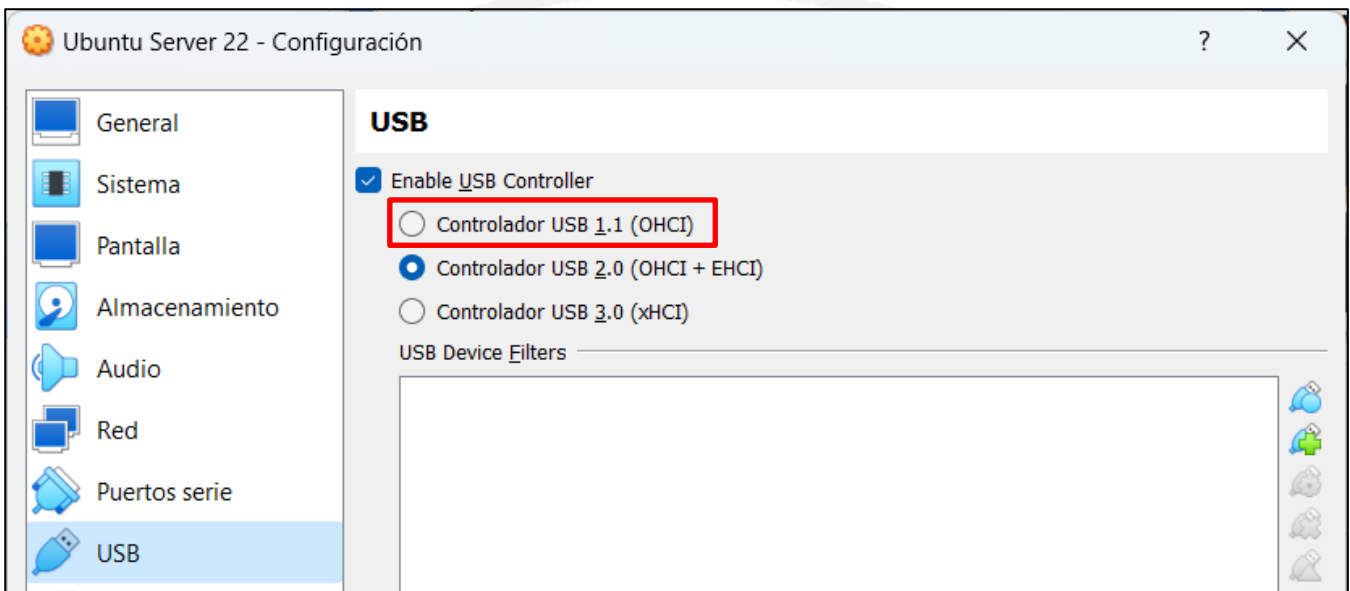


Adaptador de red 2

USB

Solo debe activarse la compatibilidad con USB 1.1 en las MVs 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

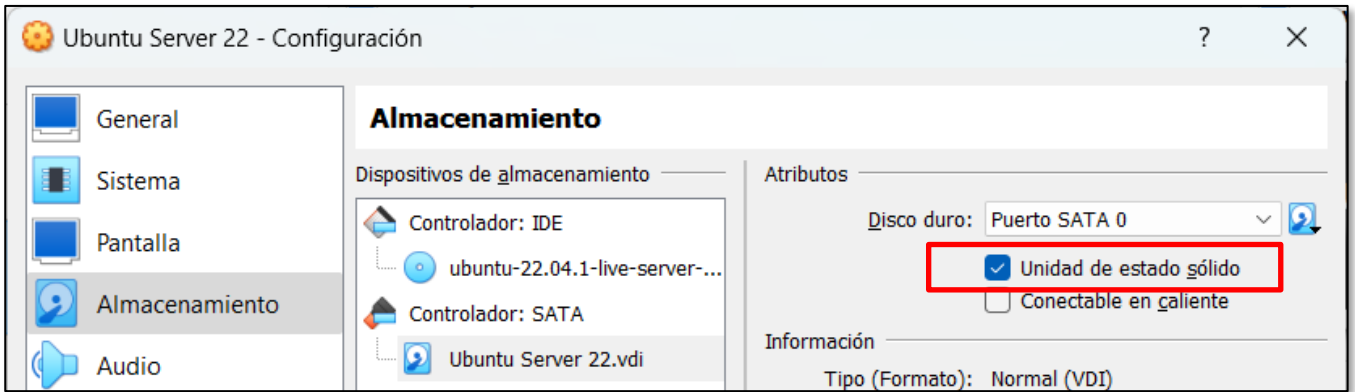
Almacenamiento y selección de la ISO del sistema operativo

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



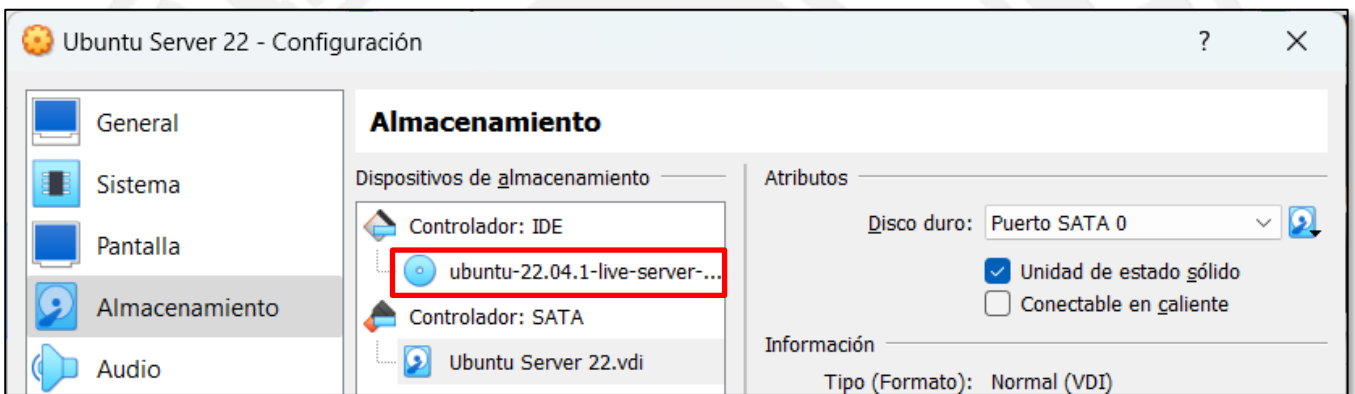
Configuración del controlador SATA

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

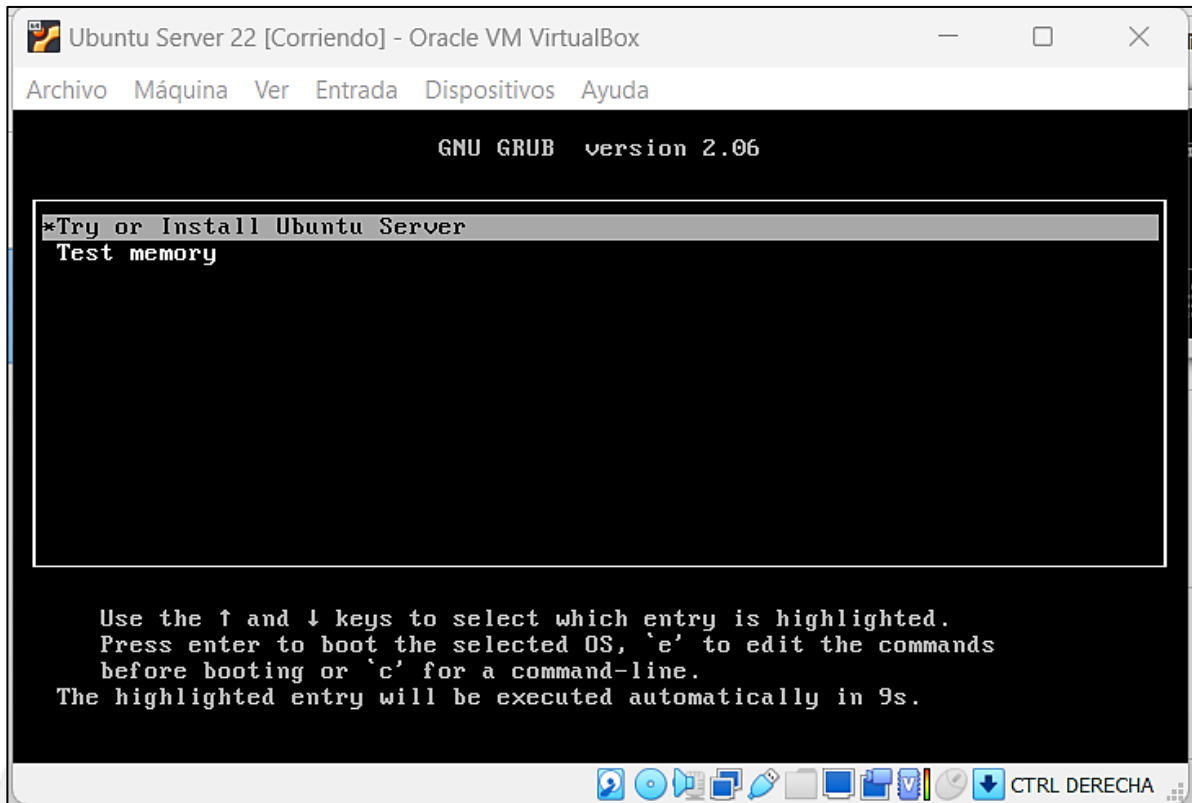
Ahora podemos iniciar la máquina y la instalación

Instalar un servidor *Ubuntu Linux*

En todo el proceso de instalación hay dos puntos muy importantes: **particionar el disco** y **crear un usuario inicial** para el SO. Hemos creado un apartado independiente para cada uno.

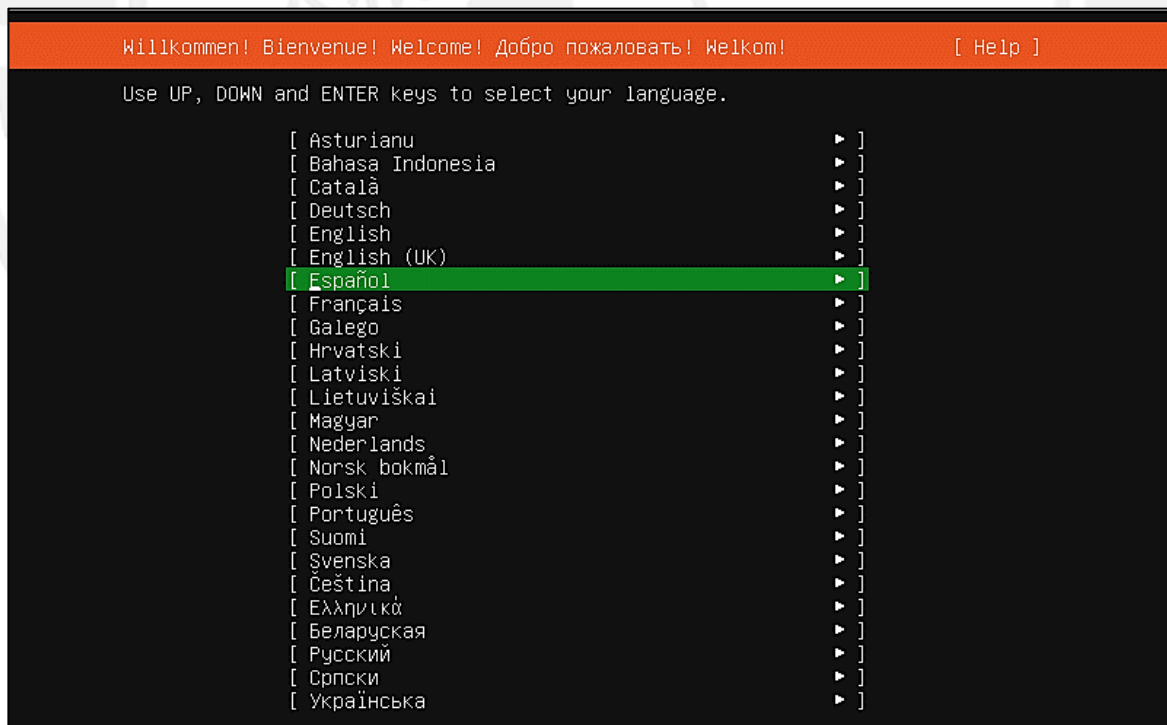
Comenzar la instalación del sistema operativo

Una vez configurada la máquina virtual, es el momento de instalar el *servidor Ubuntu*. Las pantallas de instalación **pueden haber cambiado ligeramente en versiones posteriores**, pero son esencialmente las mismas que las vistas aquí. En cualquier caso, debemos iniciar la instalación utilizando la opción de instalación simple de *Ubuntu*:



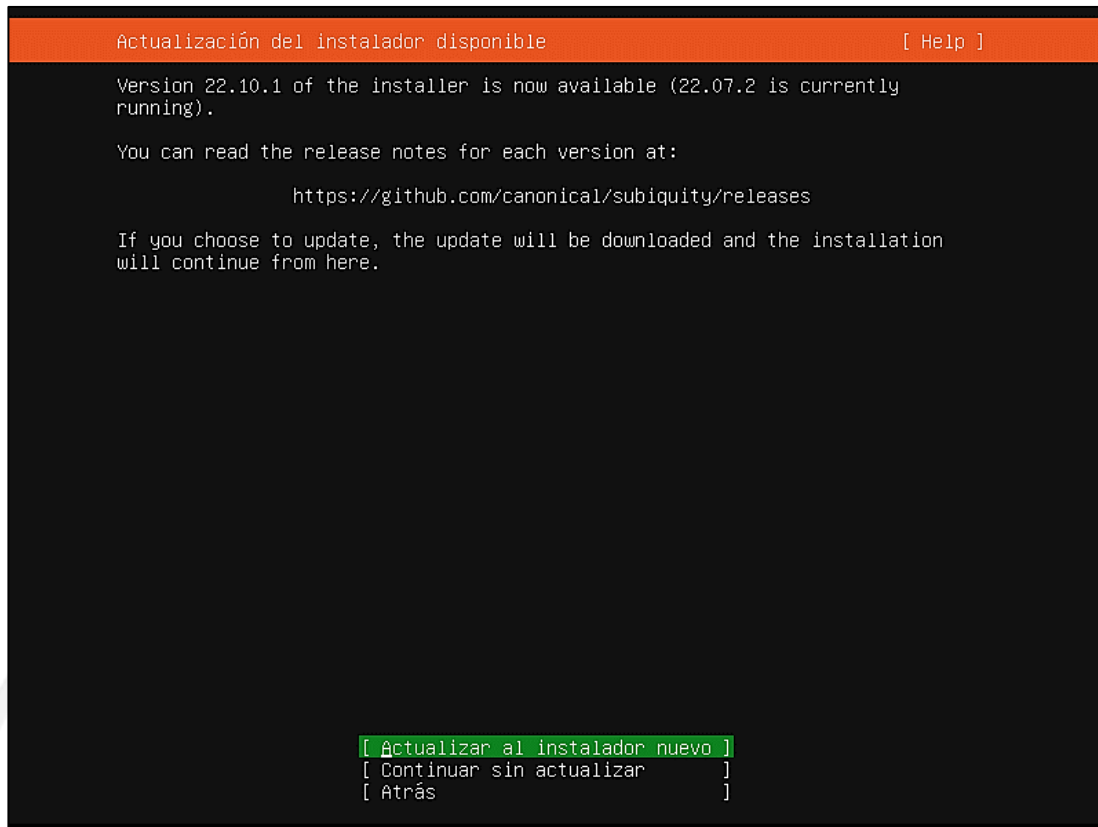
Inicio de la instalación de *Ubuntu*

Hecho esto llegamos a la pantalla de selección del idioma del sistema operativo, eligiendo el que más nos apetezca. Ten en cuenta que el idioma del teclado se selecciona posteriormente.



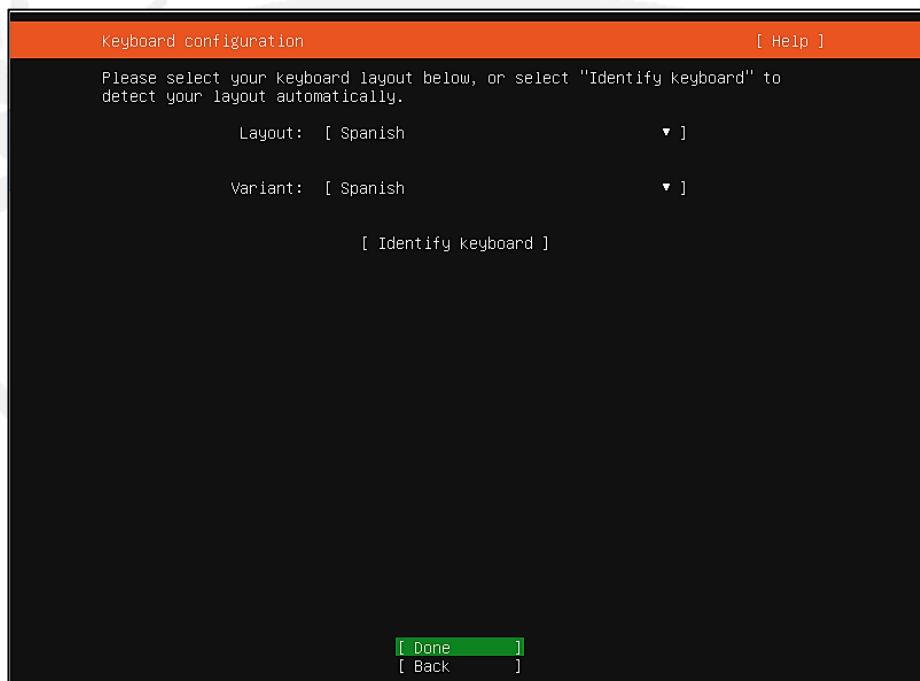
Idioma del sistema operativo

Es probable que aunque nos hayamos descargado la versión más moderna del sistema operativo el instalador en sí no lo sea, y **nos solicite ahora actualizarlo**. Lo haremos para evitar problemas en la instalación.



Actualización del instalador de Ubuntu

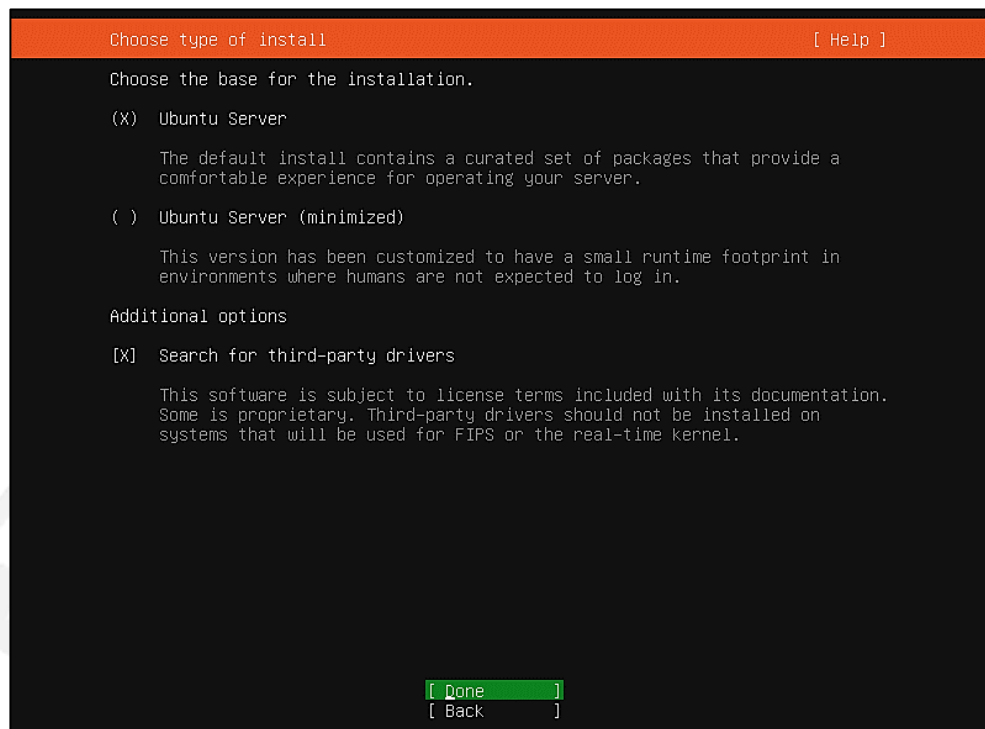
Ahora es cuando la instalación permite **seleccionar el idioma del teclado**, que por defecto es el mismo que hemos seleccionado para el sistema operativo.



Idioma del teclado

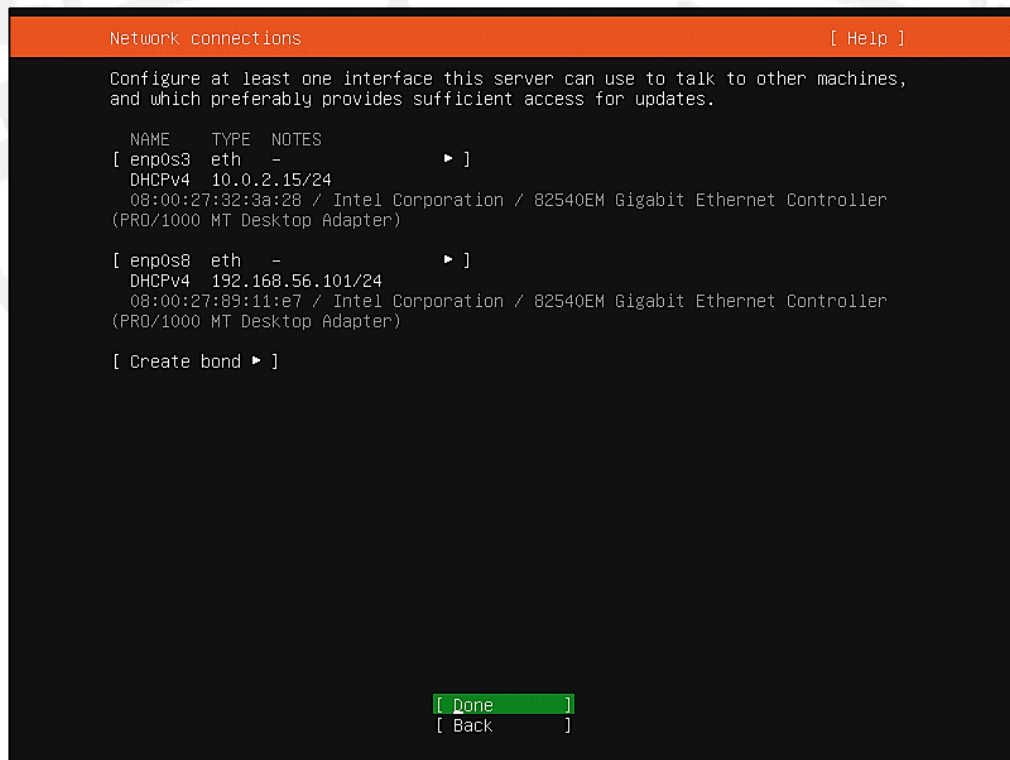
En adelante para seleccionar las opciones adecuadas **tenemos que usar la tecla *Tabulador*** para cambiar entre los elementos de la pantalla, porque no tenemos ratón, y pulsar INTRO para seleccionar la que queramos. En esta pantalla nos pregunta si queremos instalar controladores adicionales de terceros para

nuestro *hardware*, algo que puede ser conveniente en máquinas reales pero **no tendrá efecto en máquinas virtuales**, así que da igual lo que seleccionemos en este caso.



Instalación de controladores de terceros

En el siguiente paso las tarjetas de red se deberían configurar automáticamente en una instalación normal, adquiriendo una IP correcta (en la imagen se ve la NAT para salir a Internet y una *host-only*). Si es tu caso, simplemente continúa.



Redes de la máquina virtual configuradas automáticamente

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, ya que la red que vamos a usar no usa uno:

Configure proxy [Help]

If this system requires a proxy to connect to the internet, enter its details here.

Proxy address:

If you need to use a HTTP proxy to access the outside world, enter the proxy information here. Otherwise, leave this blank.

The proxy information should be given in the standard form of "http://[[user] [:pass]@]host[:port]/".

[Done]
[Back]

Configuración de un proxy

El siguiente paso permite cambiar de que máquina se descargarán los paquetes que forman parte de la instalación del sistema operativo. **Aceptamos la de por defecto** y continuamos.

Configure Ubuntu archive mirror [Help]

If you use an alternative mirror for Ubuntu, enter its details here.

Mirror address:

You may provide an archive mirror that will be used instead of the default.

[Done]
[Back]

Máquina de la que se descargará Ubuntu

Partición de discos

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 la siguiente dirección:**
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 para que 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 cifrado en el siguiente enlace: <http://diegosamuel.blogspot.com.es/2014/01/particiones-con-lvm-y-cifrado.html>.

 **En lo que respecta a nuestra instalación, simplemente aceptamos las opciones por defecto y continuamos.**

Guided storage configuration
[Help]

Configure a guided storage layout, or create a custom one:

(X) Use an entire disk
 [VBOX_HARDDISK_VBa936ece0-f52c4c2f local disk 25.000G ▾]

[X] Set up this disk as an LVM group
 [] Encrypt the LVM group with LUKS
 Passphrase:
 Confirm passphrase:

() Custom storage layout

[Done]
[Back]

Particionamiento de disco

Llegados a este punto el sistema operativo está listo para instalarse y antes de dar la orden de hacerlo nos mostrará un resumen de las opciones que hemos seleccionado en las pantallas anteriores. Como no tenemos nada que cambiar, simplemente **damos a "Done"** y permitimos que la instalación continúe.

Storage configuration
[Help]

FILE SYSTEM SUMMARY

MOUNT POINT	SIZE	TYPE	DEVICE TYPE
[/	11.496G	new ext4	new LVM logical volume ▶]
[/boot	2.000G	new ext4	new partition of local disk ▶]

AVAILABLE DEVICES

DEVICE	TYPE	SIZE
[ubuntu-vg (new)	LVM volume group	22.996G ▶]
free space		11.500G ▶]

[Create software RAID (md) ▶]
 [Create volume group (LVM) ▶]

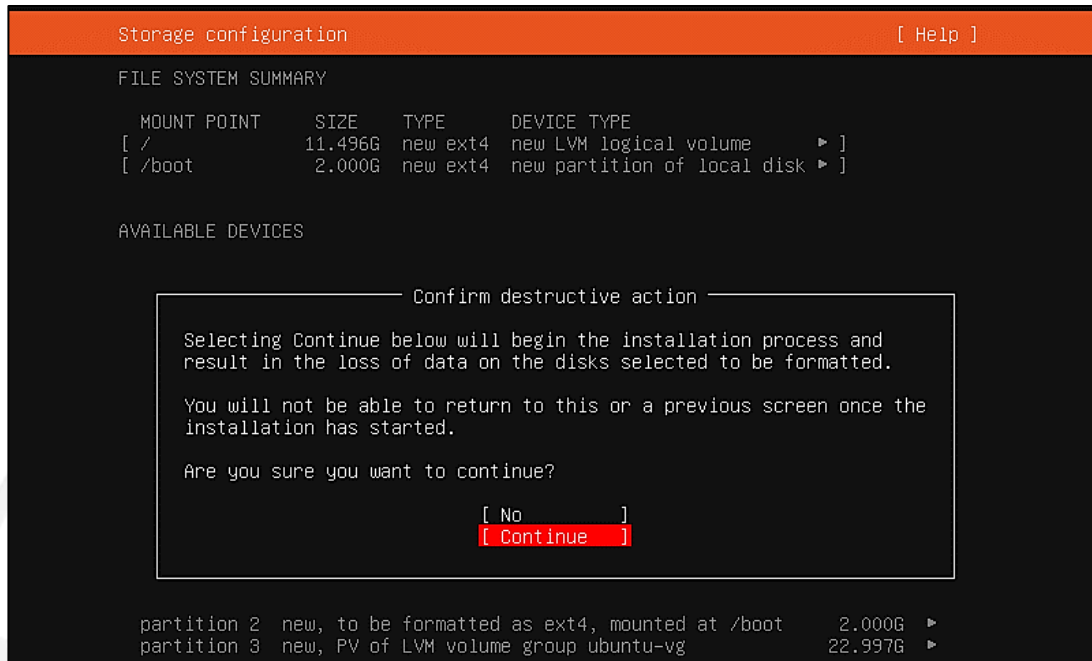
USED DEVICES

DEVICE	TYPE	SIZE
[ubuntu-vg (new)	LVM volume group	22.996G ▶]
ubuntu-lv	new, to be formatted as ext4, mounted at /	11.496G ▶]
[VBOX_HARDDISK_VBa936ece0-f52c4c2f	local disk	25.000G ▶]
partition 1	new, BIOS grub spacer	1.000M ▶]
partition 2	new, to be formatted as ext4, mounted at /boot	2.000G ▶]
partition 3	new, PV of LVM volume group ubuntu-vg	22.997G ▶]

[Done]
[Reset]
[Back]

Resumen de las opciones de instalación

Dado que esta operación **destruye todos los datos del disco duro** seleccionado para que se instale el SO, el instalador solicita confirmación antes de continuar. En nuestro caso no es problema: la máquina virtual tiene el disco vacío y no importa. No obstante, en una instalación real tenemos que estar bien seguros de que vamos a usar el disco correcto para evitar disgustos.



Confirmación de la instalación

Creación de un usuario inicial

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

- **No debe hacerse 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.

Limita el uso de la cuenta `root` solo a operaciones de administración

- Como obviamente necesitamos un usuario para trabajar, la instalación pide crear uno, que en este ejemplo se llamará `ssiuser`.
- 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 el SO 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 la teoría (**Tema 2**).

Cifrado de la carpeta personal de los usuarios

Primera actualización

Aunque hayamos instalado la versión más moderna de *Ubuntu*, siempre es posible que haya actualizaciones pendientes cuando la máquina está recién instalada. Para comprobarlo usamos la orden **sudo apt update** para buscar los paquetes más modernos.

```
redondo@servidor2204:~$ sudo apt update
[sudo] password for redondo:
Obj:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease
Obj:3 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
Obj:4 http://es.archive.ubuntu.com/ubuntu jammy-security InRelease
Des:5 http://es.archive.ubuntu.com/ubuntu jammy/main Translation-es [332 kB]
Des:6 http://es.archive.ubuntu.com/ubuntu jammy/restricted Translation-es [964 B]
Des:7 http://es.archive.ubuntu.com/ubuntu jammy/universe Translation-es [1.356 kB]
Des:8 http://es.archive.ubuntu.com/ubuntu jammy/multiverse Translation-es [68,2 kB]
Descargados 1.758 kB en 4s (435 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Se pueden actualizar 45 paquetes. Ejecute «apt list --upgradable» para verlos.
redondo@servidor2204:~$
```

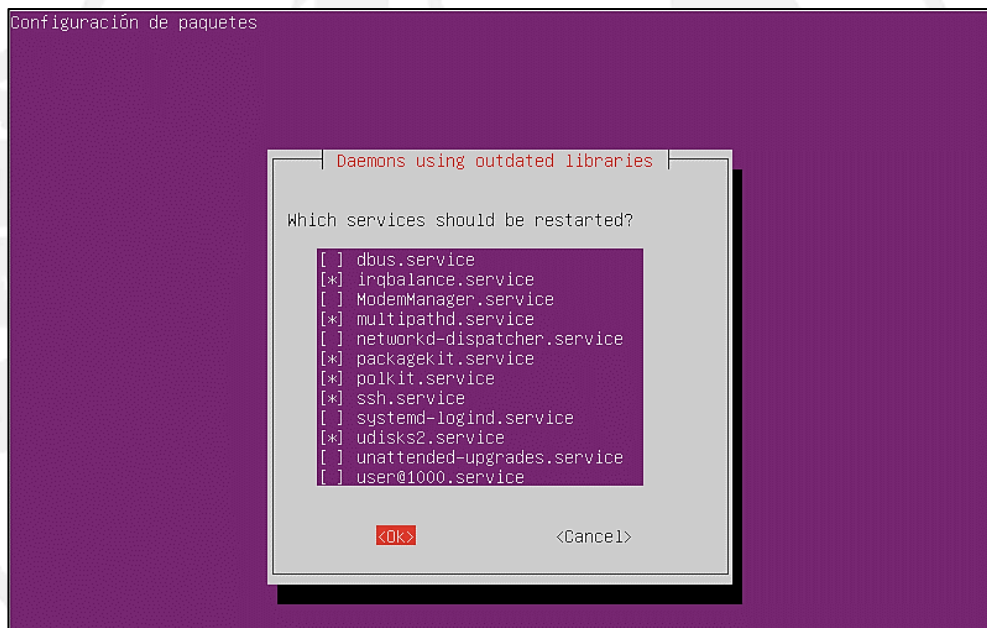
Actualización de las versiones de los paquetes

Y finalmente **sudo apt full-upgrade** para proceder a actualizarlos.

```
redondo@servidor2204:~$ sudo apt full-upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
 libflashrom1 libftdi1-2
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes NUEVOS:
 python3-magic systemd-hwe-hwdb
Se actualizarán los siguientes paquetes:
 apt apt-utils cloud-init cryptsetup cryptsetup-bin cryptsetup-initramfs dmidecode fwupd gzip
 libapt-pkg6.0 libcryptsetup12 libfwupd2 libfwupdplugin5 libldap-2.5-0 libldap-common
 libnftables1 libnss-systemd libpam-systemd libpython3-stdlib libpython3.10 libpython3.10-minimal
 libpython3.10-stdlib libsystemd0 libudev1 nftables python3 python3-distupgrade python3-distutils
 python3-gdbm python3-lib2to3 python3-minimal python3-software-properties python3.10
 python3.10-minimal snapd software-properties-common sosreport sudo systemd systemd-sysv
 systemd-timesyncd tzdata ubuntu-advantage-tools ubuntu-release-upgrader-core udev
45 actualizados, 2 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 47,3 MB de archivos.
Se utilizarán 5.360 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s_
```

Actualización de los paquetes instalados

Al finalizar la actualización es posible que el sistema operativo nos pida **confirmación para reiniciar ciertos servicios** que se han actualizado, **algo que debemos hacer** si no queremos encontrarnos con problemas de funcionamiento sobrevenidos.



Reinicio de servicios tras actualización

Algunos de los paquetes de las versiones más modernas de *Ubuntu* no vienen con el gestor de paquetes clásico **apt**, sino con otro complementario llamado **snap**. Este gestor requiere que sus paquetes se actualicen independientemente, si no queremos depender de los procesos que hacen estas actualizaciones automáticas periódicamente. Para ello hay que hacer **sudo snap refresh**, preferiblemente con ninguna aplicación de GUI abierta (si lo tuviéramos).

```
redondo@servidor2204:~$ sudo snap refresh
[sudo] password for redondo:
2022-10-30T21:40:57Z INFO Waiting for automatic snapd restart...
core20 20220826 from Canonical refreshed
lxd (5.0/stable) 5.0.1-9dcf35b from Canonical refreshed
snapd 2.57.4 from Canonical refreshed
redondo@servidor2204:~$ _
```

Actualización de paquetes snap

Conviene hacer este proceso de actualización de ambos gestores periódicamente por si acaso o automatizar las actualizaciones (descrito en [esta sección](#)).

¿Instalar un GUI?

Si hubiésemos elegido la instalación de un *Ubuntu Desktop*, tendríamos un GUI a nuestra disposición por defecto sin necesidad de instalar nada adicional. En cambio, un **Ubuntu Server** viene sin él para ahorrar recursos. Si bien es posible instalarle el mismo que su versión *Desktop*, tiene más sentido **instalar uno ligero** que se puede activar solo si es necesario. En este caso uno de los más adecuados es el GUI Xfce4, instalable con `sudo apt-get -y install -y xfce4`. Como vemos en la siguiente imagen, ocupa bastante espacio.

```
libnss-mdns libogg0 libopengl0 libopenjp2-7 libopus0 liborc-0.4-0 libpango-1.0-0
libpangocairo-1.0-0 libpangoft2-1.0-0 libpangomm-1.4-1v5 libpangoxft-1.0-0 libpciaccess0
libpipewire-0.3-0 libpipewire-0.3-common libpipewire-0.3-modules libpixman-1-0 libpoppler-glib8
libpoppler118 libproxy1v5 libpulse-mainloop-glib0 libpulse0 libpulsedsp libraw1394-11 librsync2-2
librsync2-common libsamplerate0 libsane-common libsane1 libsecret-1-0 libsecret-common
libsensors-config libsensors5 libshout3 libsigc++-2.0-0v5 libsm6 libsnapd-glib1 libsndfile1
libsnmp-base libsnmp40 libsoup2.4-1 libsoup2.4-common libsoxr0 libspa-0.2-modules libspeex1
libspeexdsp1 libstartup-notification0 libtag1v5 libtag1v5-vanilla libtdb1 libthai-data libthai0
libtheora0 libthunarx-3-0 libtiff5 libtumbler-1-0 libtwolame0 liburi-perl libv4l-0
libv4lconvert0 libvisual-0.4-0 libvorbis0a libvorbisenc2 libvorbisfile3 libvpx7 libvte-2.91-0
libvte-2.91-common libvulkan1 libwacom-bin libwacom-common libwacom9 libwavpack1
libwayland-client0 libwayland-cursor0 libwayland-egl1 libwayland-server0 libwebkit2gtk-4.0-37
libwebp7 libwebpdemux2 libwebp7c audio-processing1 libwnck-3-0 libwnck-3-common libwoff1
libx11-xcb1 libxatracker2 libxaw7 libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0
libxcb-randr0 libxcb-render0 libxcb-shape0 libxcb-shm0 libxcb-sync1 libxcb-util1 libxcb-xfixes0
libxcomposite1 libxcursor1 libxcvt0 libxdamage1 libxfce4panel-2.0-4 libxfce4ui-2-0
libxfce4ui-common libxfce4ui-utils libxfce4util-bin libxfce4util-common libxfce4util17
libxfconf-0-3 libxfconf3 libxfconf2 libxft2 libxi6 libxinerama1 libxkbcommon0 libxkbfile1
libxklavier16 libxmu6 libxpm4 libxpresent1 libxrandr2 libxrender1 libxres1 libxshmfence1 libxss1
libxt6 libxtst6 libxv1 libxvmc1 libxxf86dga1 libxxf86vm1 libyelp0 mesa-vulkan-drivers
nautilus-extension-gnome-terminal pavucontrol pipewire pipewire-bin pipewire-media-session
plymouth-label policykit-1-gnome poppler-data pulseaudio pulseaudio-utils rtkit sane-airscan
sane-utils session-migration sgml-base sgml-data sound-theme-freedesktop tango-icon-theme thunar
thunar-data thunar-volman tumbler tumbler-common ubuntu-mono update-inetd x11-apps x11-common
x11-session-utils x11-utils x11-xkb-utils x11-xserver-utils xbitmaps xcv1 xdg-dbus-proxy
xdg-desktop-portal xdg-desktop-portal-gtk xfce4 xfce4-appfinder xfce4-helpers xfce4-notifd
xfce4-panel xfce4-pulseaudio-plugin xfce4-screensaver xfce4-session xfce4-settings xfconf
xfdesktop4 xfdesktop4-data xfonts-base xfonts-encodings xfonts-scalable xfonts-utils xfw4 xiccd
xinit xinput xml-core xorg xorg-docs-core xserver-common xserver-xorg xserver-xorg-core
xserver-xorg-input-all xserver-xorg-input-libinput xserver-xorg-input-wacom xserver-xorg-legacy
xserver-xorg-video-all xserver-xorg-video-amdgpu xserver-xorg-video-ati xserver-xorg-video-fbdev
xserver-xorg-video-intel xserver-xorg-video-nouveau xserver-xorg-video-qxl
xserver-xorg-video-radeon xserver-xorg-video-vesa xserver-xorg-video-vmware yelp yelp-xsl
0 actualizados, 380 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 161 MB de archivos.
Se utilizarán 632 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Instalación de XFCE4

Es necesario reiniciar la máquina para poder usar el GUI. No obstante, conviene hacer el siguiente paso antes de tratar de arrancarlo para conseguir la mejor experiencia de usuario.

Configuración de la máquina para mejorar la virtualización

VirtualBox viene con una serie de **extensiones que permiten mejorar el hardware** de la máquina virtual, de manera que, entre otras cosas, podamos disfrutar de mayor resolución de pantalla, redimensionamiento de esta a voluntad, capacidad de copiar y pegar desde / hacia la máquina virtual y otras ventajas.

No obstante, **estas extensiones realmente solo se les puede sacar partido si tenemos un GUI**, por lo que no se instalan por defecto. Ahora que lo tenemos, conviene acceder a la terminal y teclear **`sudo apt install virtualbox-guest-x11`** para hacer dicha instalación.

```
redondo@servidor2204:~$ sudo apt install virtualbox-guest-x11
[sudo] password for redondo:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  libflashrom1 libftdi1-2
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  virtualbox-guest-utils
Se instalarán los siguientes paquetes NUEVOS:
  virtualbox-guest-utils virtualbox-guest-x11
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 1.673 kB de archivos.
Se utilizarán 8.257 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s_
```

Instalación de las extensiones de virtualización

Para seguir trabajando con la máquina **es necesario hacer un reinicio**. Ahora ya podremos teclear **`startx`** desde la consola para disponer de un GUI funcional multiusos con las herramientas básicas para trabajar, que además se adaptará a la resolución de nuestras pantallas sin problema cuando redimensionemos la ventana e irá más ágil.

```
Ubuntu 22.04.1 LTS servidor2204 tty1
servidor2204 login: redondo
Password:
Welcome to Ubuntu 22.04.1 LTS (GNU/Linux 5.15.0-52-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

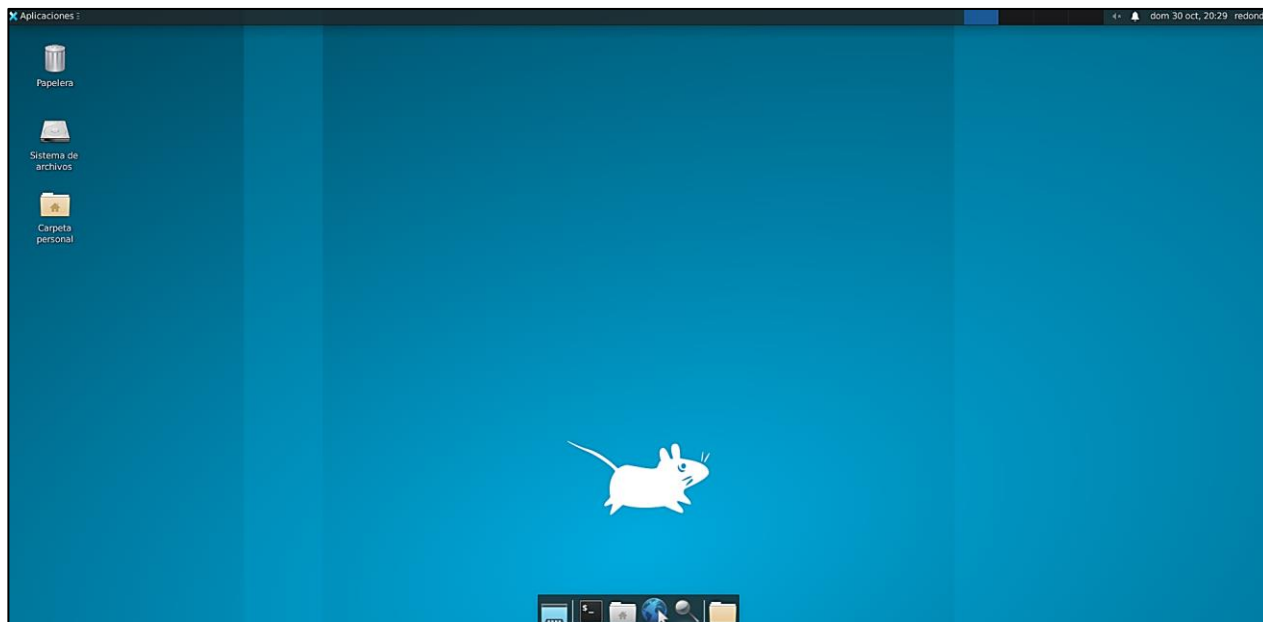
System information as of dom 30 oct 2022 20:28:24 UTC

System load:  0.0               Processes:            119
Usage of /:   47.2% of 11.21GB   Users logged in:     0
Memory usage: 10%              IPv4 address for enp0s3: 10.0.2.15
Swap usage:   0%               IPv4 address for enp0s8: 192.168.56.101

0 updates can be applied immediately.

Last login: Sun Oct 30 20:26:40 UTC 2022 on tty1
redondo@servidor2204:~$ startx
```

Arranque del GUI ligero XFCE4 desde consola



GUI XFCE4

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.

- 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
```

- Agregar el **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
```

- **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 a la hora de configurar la máquina para mejorar la virtualización. Hay que cambiar lo dicho en este apartado por esto para asegurar 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
```

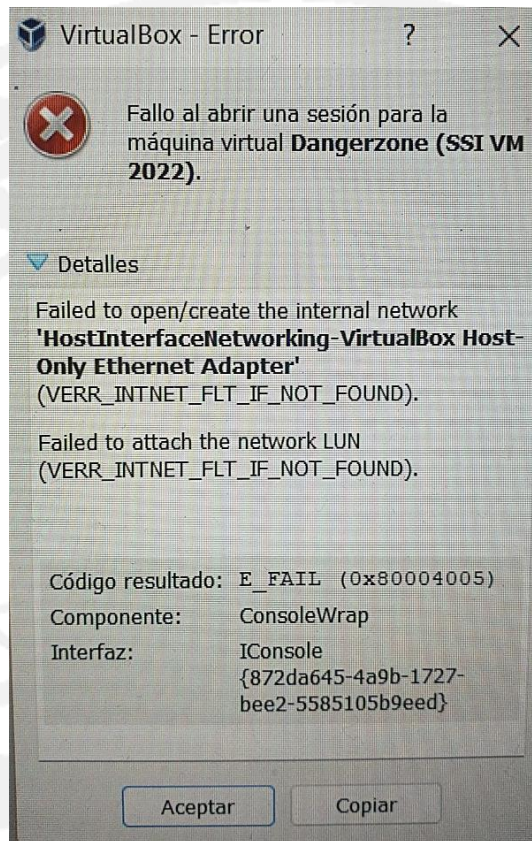
- Actualizar `initramfs`

```
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.



Si intento arrancar *Firefox* desde un cliente SSH me da un error

Concretamente nos referimos a este:

```
MoTTY X11 proxy: Unsupported authorisation protocol
Error: cannot open display: localhost:11.0
```

La solución es ejecutar `XAUTHORITY=$HOME/.Xauthority /snap/bin/firefox` o bien seguir alguna de estas otras formas de solucionar el problema: <https://askubuntu.com/questions/1181046/x-snaps-not-starting>



Bloque 3: Operaciones de gestión de MV generales



Clonación 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. Veamos cómo realizar el proceso de clonación.



Otra opción es trabajar con instantáneas de la máquina base, algo que haremos más adelante.

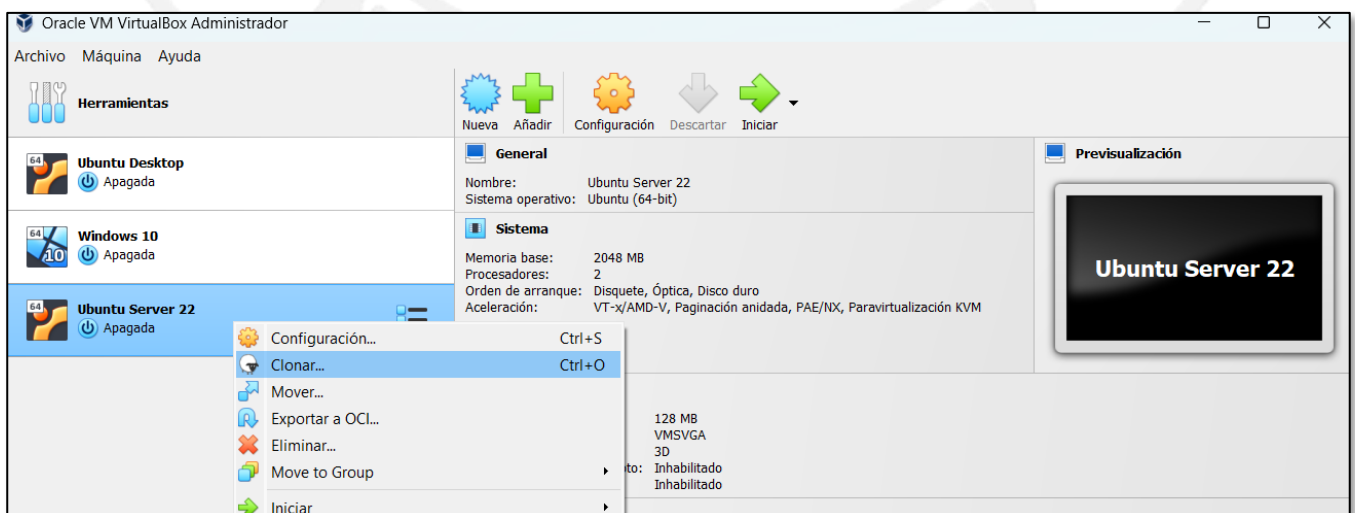
Lo primero que hay que hacer antes de clonar una MV es estudiar si merece la pena hacer una **compactación de HD**. Este es un proceso que examina el disco duro de la máquina virtual, y lo optimiza para que el archivo que representa ese disco duro en el *host* ocupe lo menos posible, pero sin romper 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.

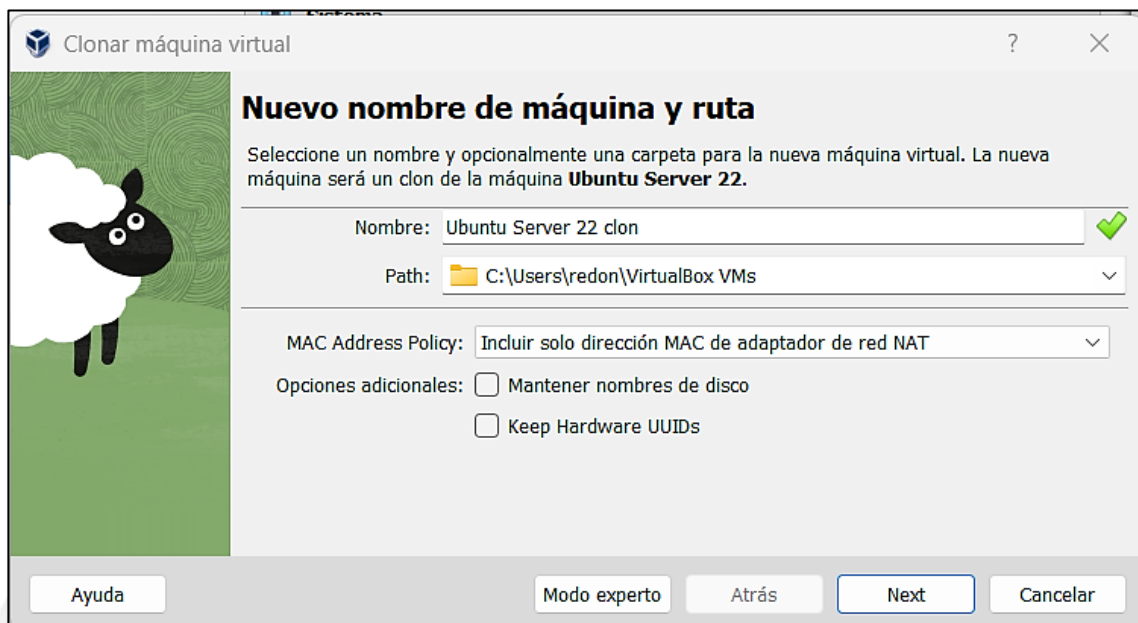
Sin embargo, en nuestro caso, al ser una máquina recién instalada no merece la pena. Las instrucciones sobre cómo realizar este proceso se pueden encontrar aquí: <https://www.virtualbox.org/manual/ch08.html> o en: <https://www.howtogeek.com/312883/how-to-shrink-a-virtualbox-virtual-machine-and-free-up-disk-space/>. 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

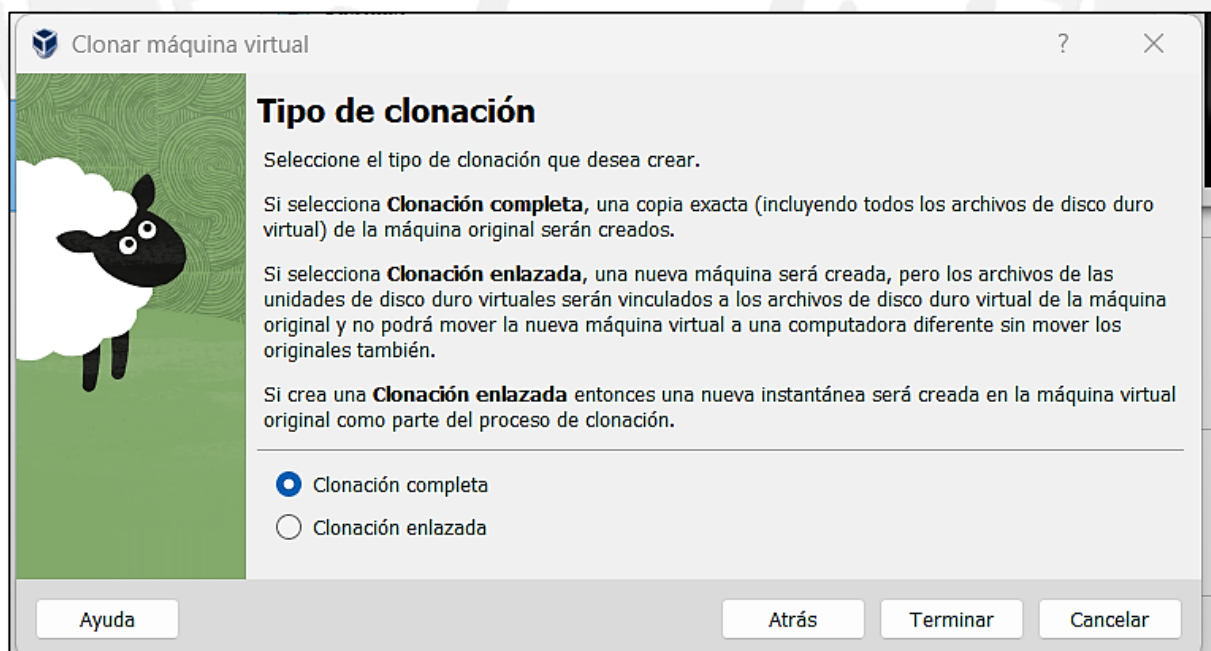
La máquina clonada aparecerá en el inventario de *VirtualBox*, por lo que necesitamos darle un **nombre nuevo que no exista en el mismo**:



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 otros clones, y queremos evitar errores de red 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:



Selección del tipo de clonación que se va a realizar

Ahora tenemos dos máquinas diferentes, 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`:

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

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.

¿Qué es la clonación enlazada?

La clonación enlazada en *VirtualBox* es un proceso que crea una instantánea del disco duro virtual de la máquina original (como las que explicaremos en la siguiente sección), que luego se utiliza como origen para el disco de la máquina clonada. De esta manera, ambos discos permanecen relacionados, lo que significa que **no puedes eliminar la máquina original mientras necesites la clonada**.

Este método es muy útil para ahorrar espacio y crear clones de la MV de forma muy rápida, ya que no se crean archivos completos, sino enlaces a la máquina original.

 **Sin embargo, debes tener en cuenta que cualquier cambio que hagas en la máquina clonada también afectará a la máquina original, debido a esta relación.**

Creación de instantáneas de máquinas

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. Puedes verlas como las partidas guardadas de un videojuego 😊**

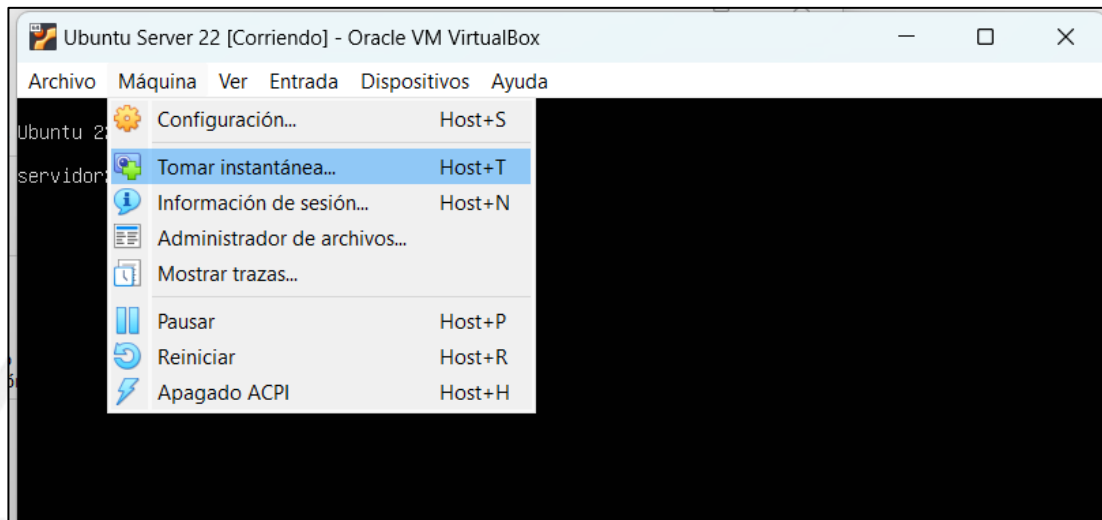
Con las instantáneas puedes guardar un estado determinado de una máquina virtual para su uso posterior. En cualquier momento puedes volver a cualquiera de esos estados, incluso si la máquina virtual ha cambiado considerablemente desde entonces. Por lo tanto, **una instantánea de una máquina virtual es 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

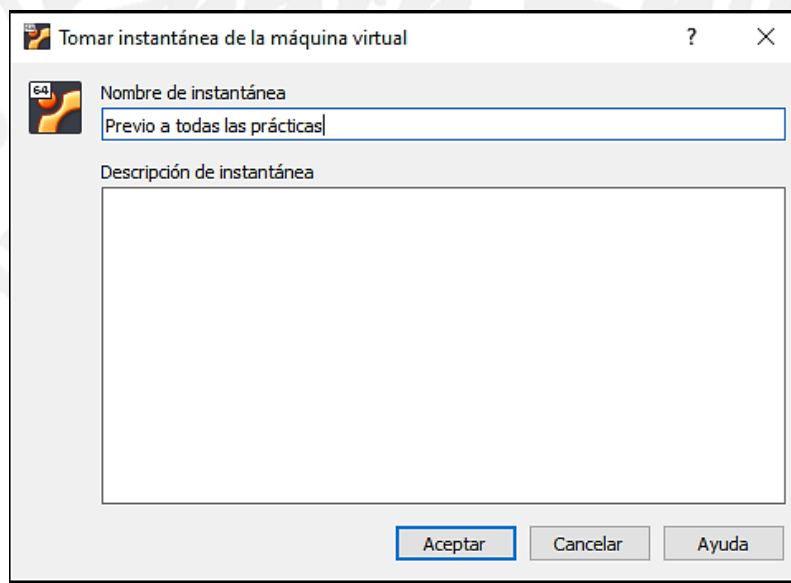
Toma una instantánea

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.



Tome una instantánea de una máquina en funcionamiento

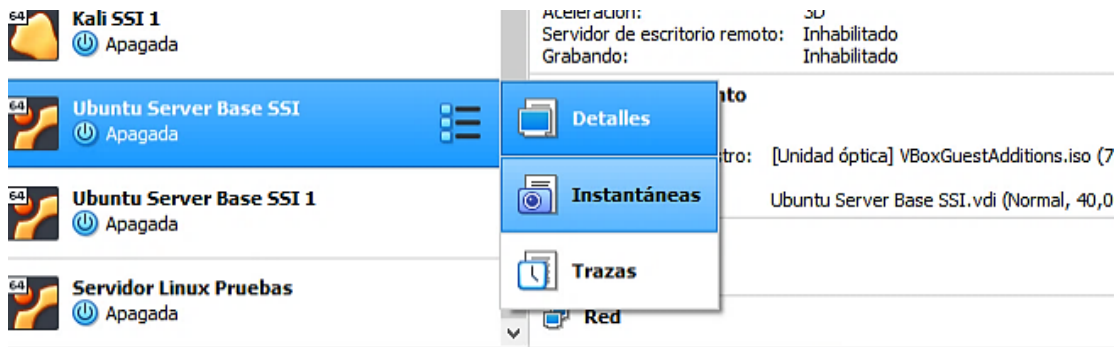
Esto muestra una ventana que solicita un nombre de instantánea. Este nombre es puramente para referenciarla, y ayuda a recordar el estado de la instantánea. **En nuestra asignatura tiene sentido generar estados para cada laboratorio**, de manera que puedas volver a cada uno de ellos de forma independiente 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

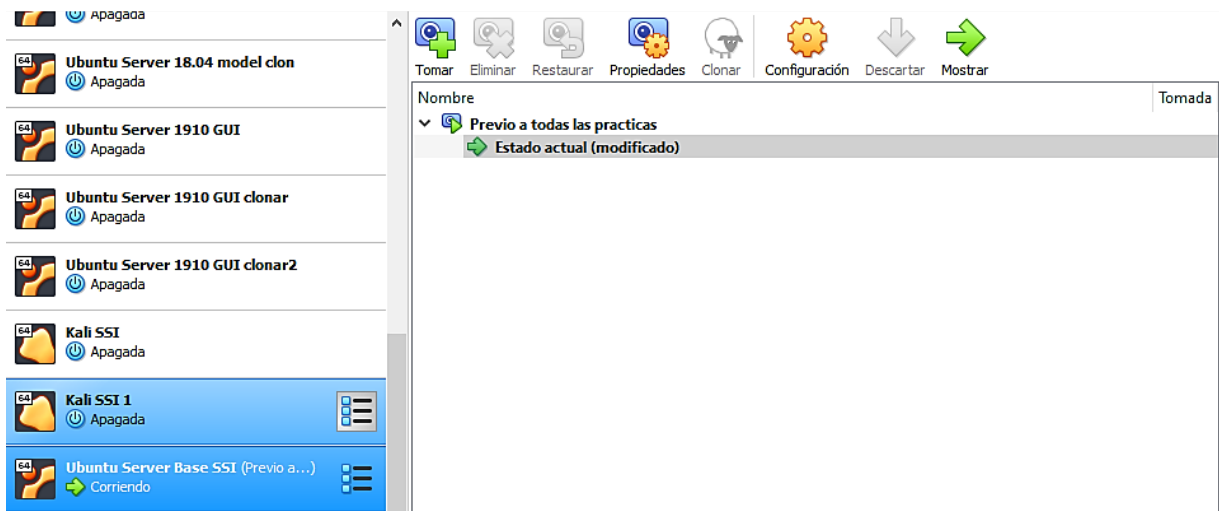
Espacio ocupado por la MV *Ubuntu* o uno de sus clones

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

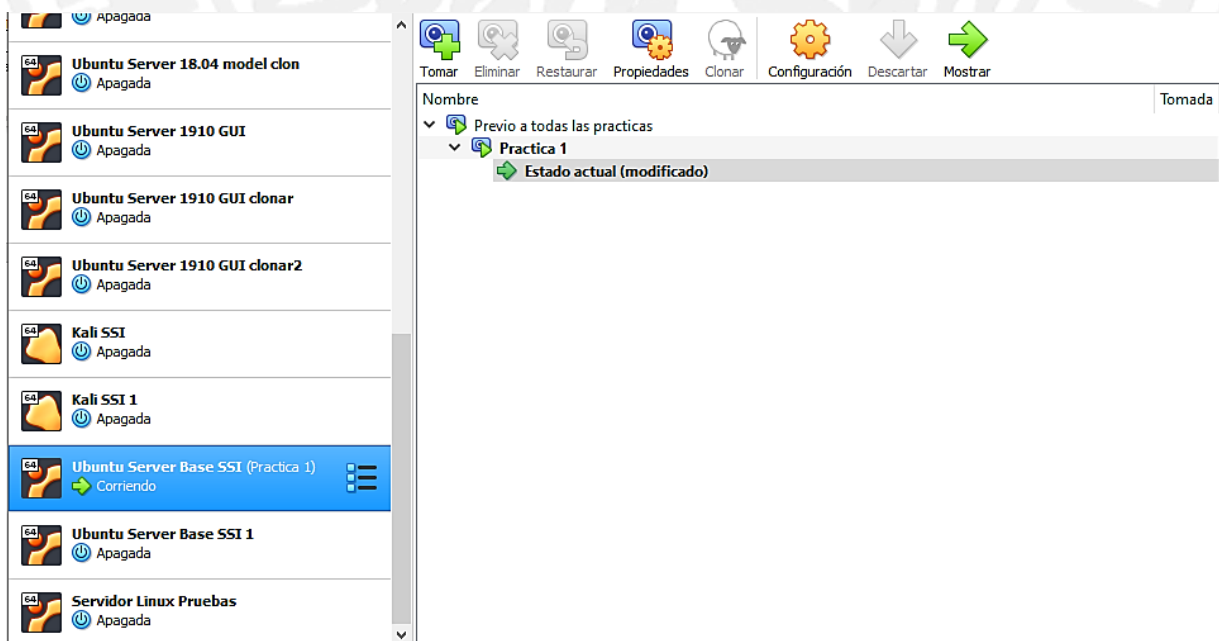


Lista de instantáneas

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:~$ _
```

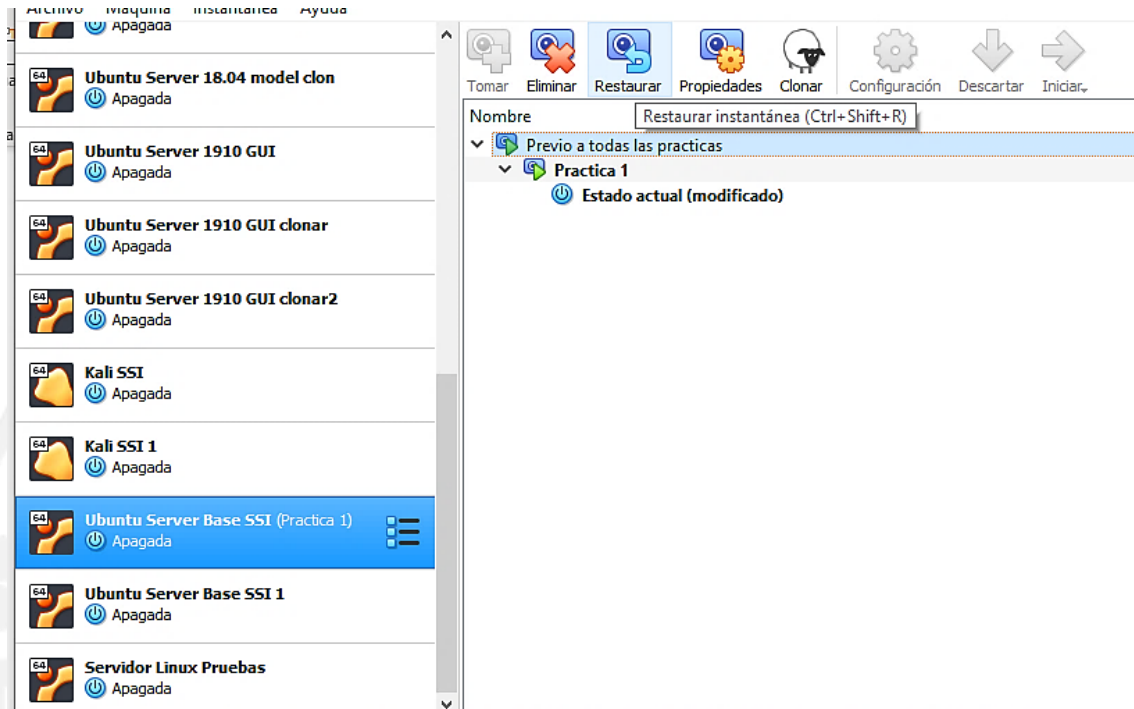
Vemos que, al tomar la instantánea, el estado actual aparece debajo de ella.



Nueva instantánea en la lista de instantáneas

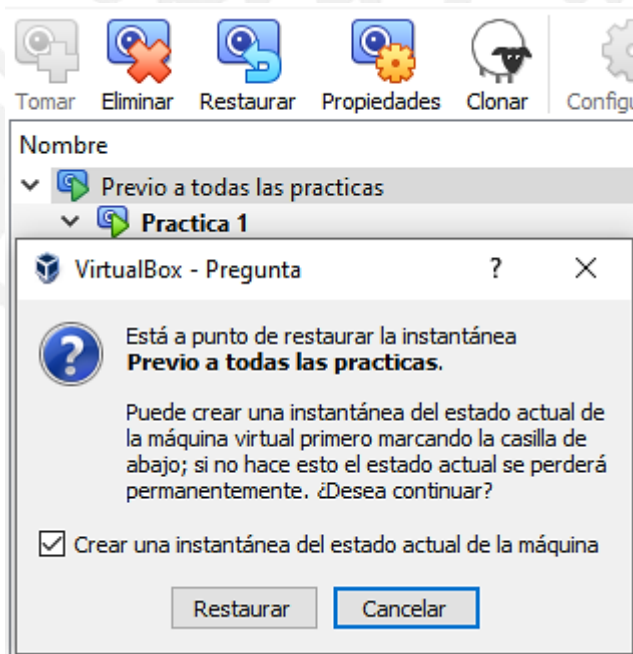
Restaurar una instantánea

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.



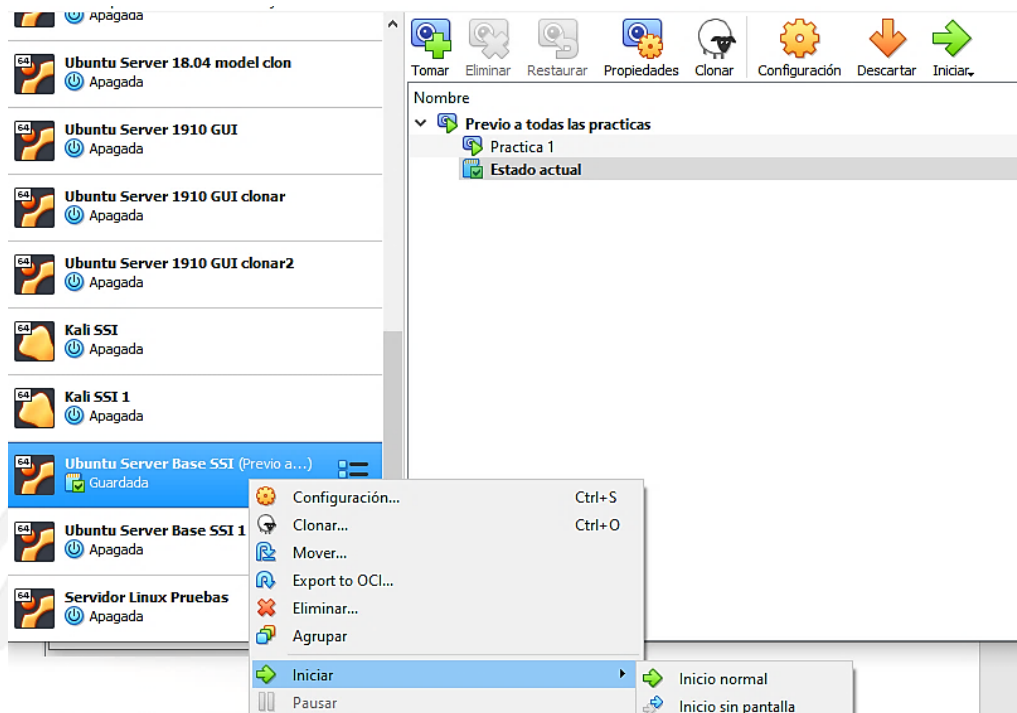
Restauración de una instantánea

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.

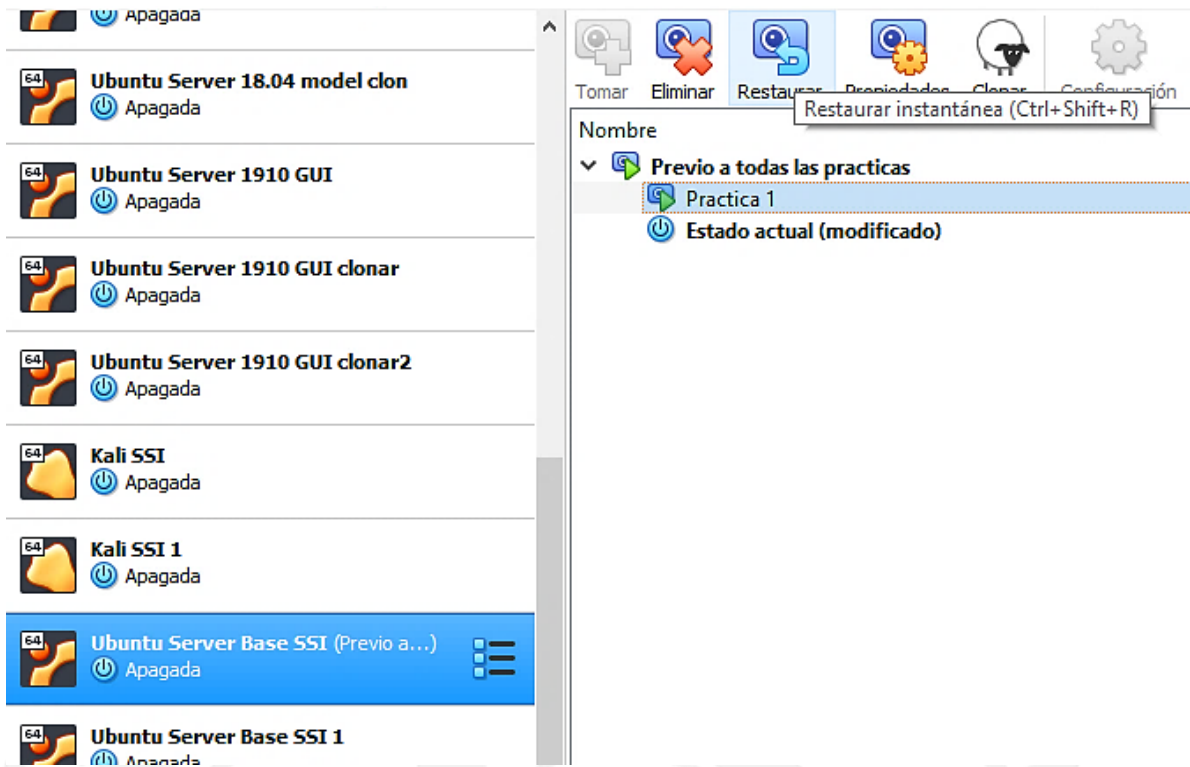
```

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----- 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
drwx----- 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:~$ _

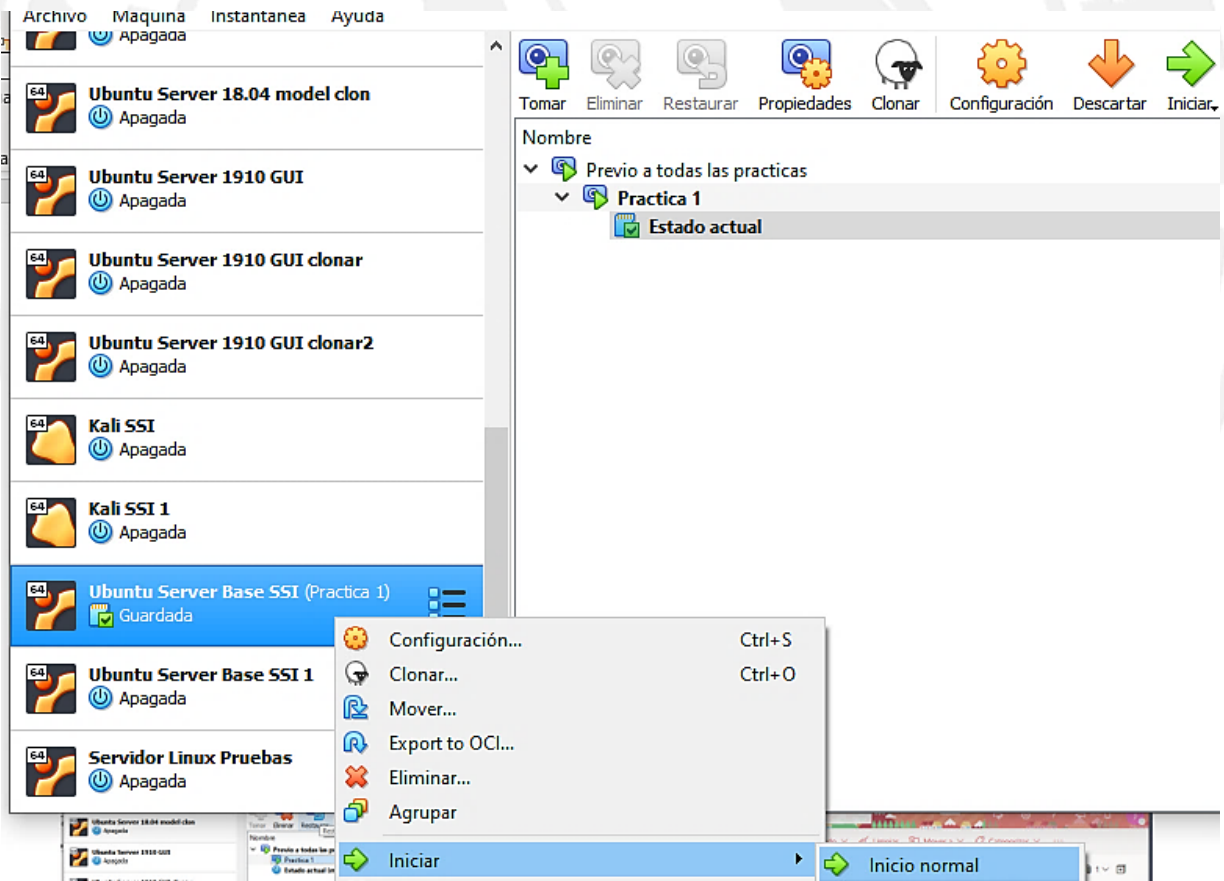
```

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

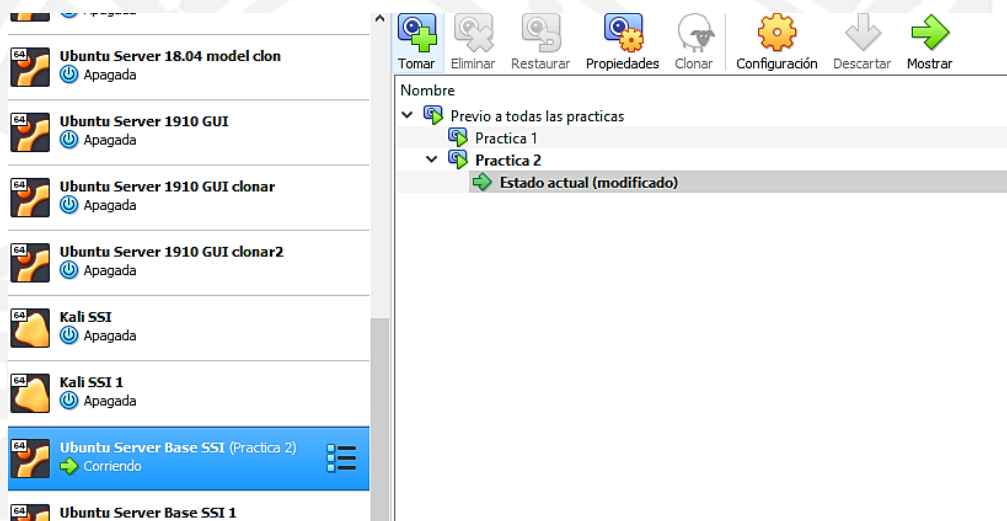


Lugar para restaurar un estado posterior


```
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 con el host

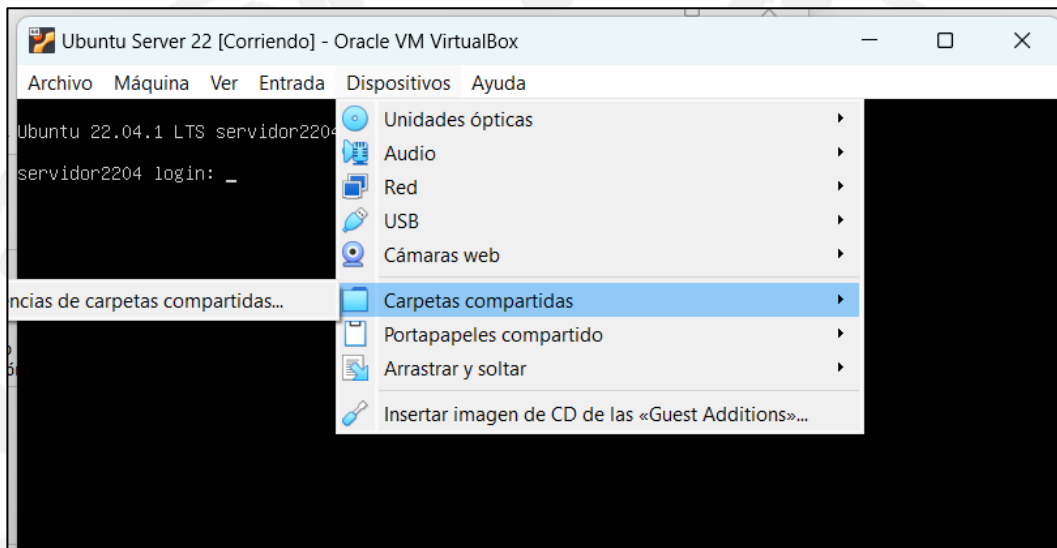
NOTA: Esto no es necesario si creaste la máquina virtual mediante la opción 2, que ya tiene carpetas compartidas configuradas.

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*.

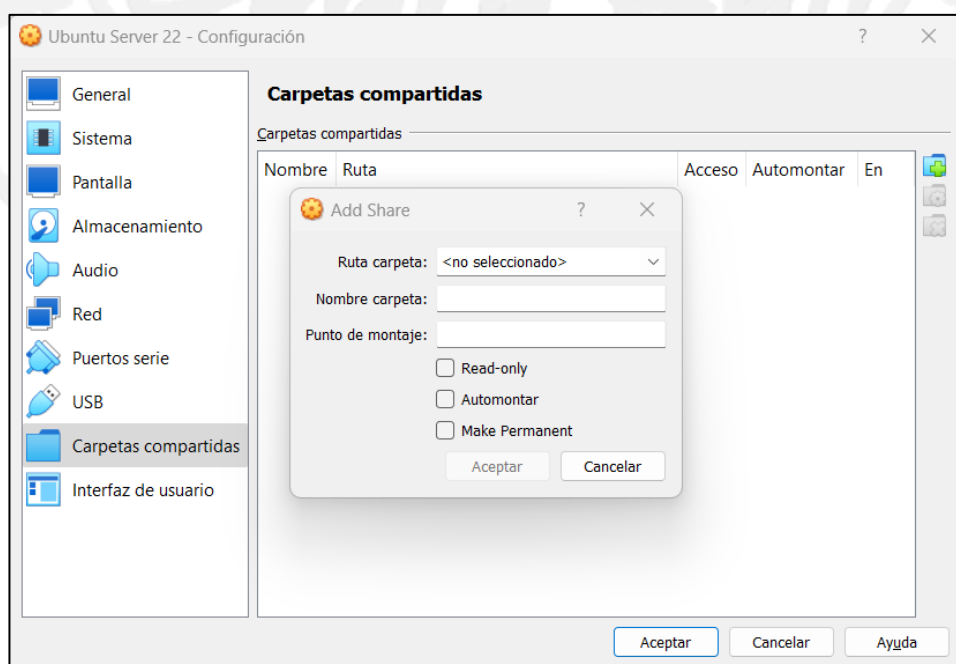
Por lo tanto, este es un mecanismo de transferencia de archivos bidireccional adecuado para muchos usos. Solo tiene dos condiciones: **tener previamente instaladas las Guest Additions y crear y asignar las carpetas compartidas adecuadamente**. Esto último depende de la máquina virtual y de lo que haya 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). Esto muestra una pantalla que nos pide que creamos una carpeta de máquina (potencialmente permanente) o una transitoria (no permanente). Elegimos la primera opción



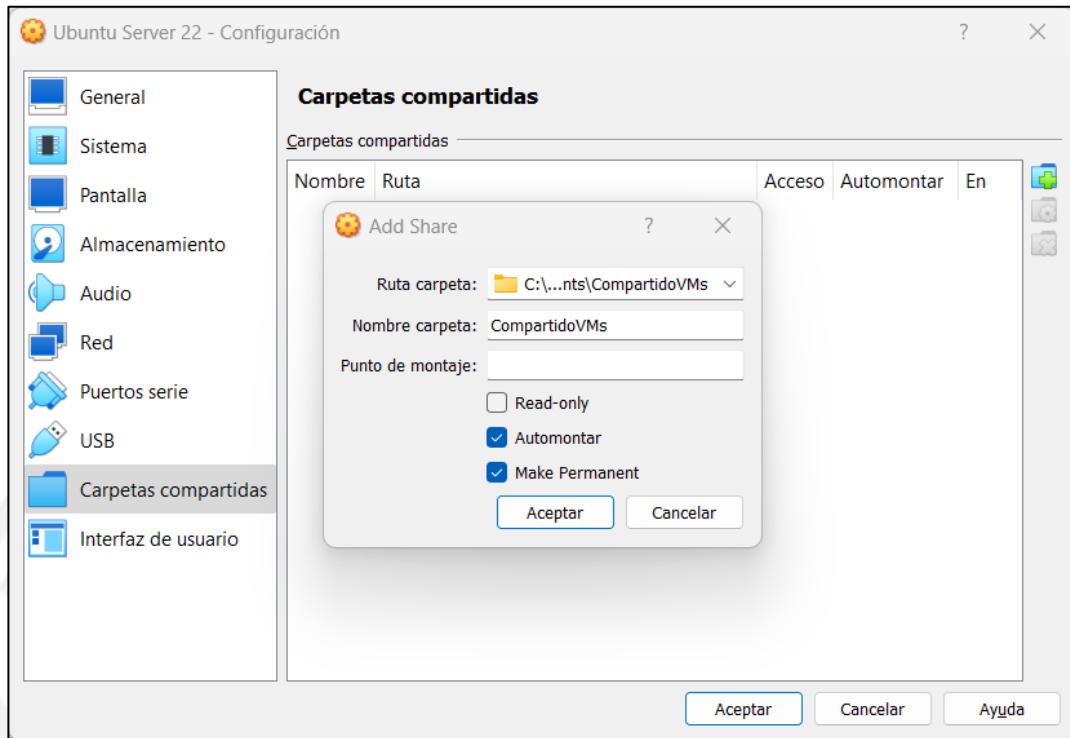
Carpetas compartidas actualmente en una máquina VirtualBox

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. 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.

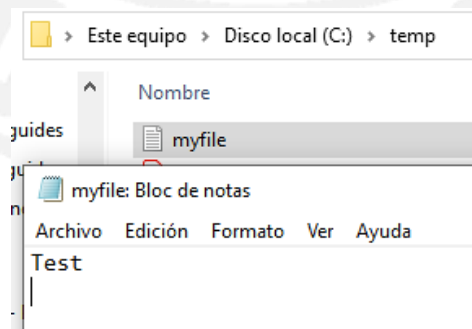


Configuración completa de una carpeta compartida de *VirtualBox*

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@ubuntuss:~$ 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](https://gist.github.com/estorgio/1d679f962e8209f8a9232f7593683265) para hacerla persistente:

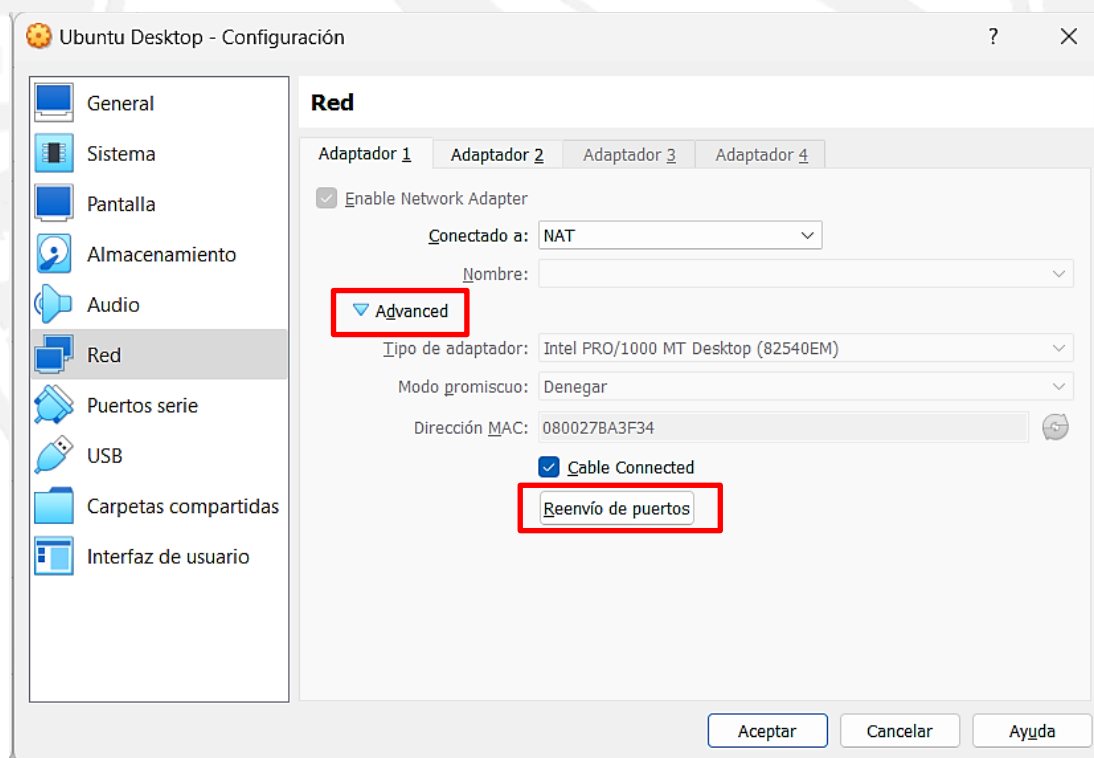
Redirección de puertos

La redirección de puertos en *VirtualBox* es un proceso que **permite acceder a servicios específicos de una máquina virtual desde tu ordenador host**. Cuando creamos una máquina virtual en *VirtualBox*, hemos visto que, por defecto se le asigna una red de tipo NAT. La dirección IP de esta máquina será privada y asignada por el propio *VirtualBox* que actúa como servidor DHCP.

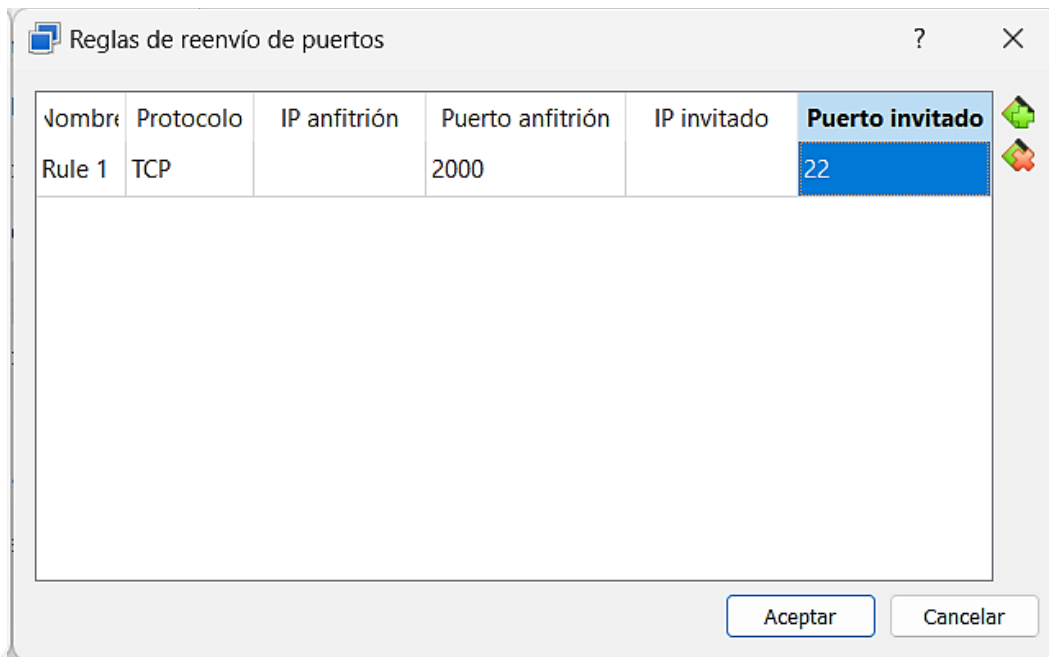
El problema viene cuando necesitamos acceder a determinados servicios internos de esa máquina desde nuestro ordenador (host). Para solucionar esto, podemos recurrir a **crear una tarjeta *host-only***, como explicamos [anteriormente](#), o podemos **hacer una redirección de puertos**. Esto significa que un puerto específico en el ordenador host se "redirige" a un puerto en la máquina virtual.

Por ejemplo, podríamos redirigir el puerto 2000 del anfitrión al puerto 22 (puerto SSH por defecto) en la máquina virtual. De esta manera, cuando te conectas al puerto 2000 en el anfitrión, en realidad te estás conectando al puerto 22 en la máquina virtual.

Este proceso es útil cuando **quieres ejecutar software que hace de servidor** dentro de una máquina virtual y necesitas acceder a él desde el ordenador host y no quieres (o no puedes, como pasa habitualmente en las aulas de los laboratorios) crear una tarjeta *host-only* para ello. Para usar la redirección de puertos, tienes que ir a esta opción de la tarjeta de red que uses para conectarte vía NAT en la MV en la que quieras hacer dicha redirección. Una vez allí, haz clic en "**Advanced**" y en "**Reenvío de puertos**":



Ahora debes **crear las reglas** que controlarán la redirección de puertos para esa máquina virtual concreta. Como ves en la imagen siguiente, puedes crear varias (una por servicio, por ejemplo), e incluso vincularlas a IPs concretas del host y del invitado (útil cuando alguno de ellos tiene más de una tarjeta de red). En este caso no es así y queremos hacer una redirección sencilla de manera que cuando nos conectemos a nosotros mismos a través del puerto 2000 / TCP, vayamos al puerto 22 (SSH) de la MV.



Si estamos en *Windows*, al crear la regla **nos saldrá un aviso del firewall**, ya que va a crearse una regla en el mismo para permitir esta redirección automáticamente. Si le decimos que sí, entonces podremos hacer la operación que queremos:

```
C:\Users\redon>ssh 127.0.0.1 -p 2000
The authenticity of host '[127.0.0.1]:2000 ([127.0.0.1]:2000)' can't be established.
ED25519 key fingerprint is SHA256:NUhYICnt4hQ9S647/McyHqVLj4Pi8cgOM3mmihCskGE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[127.0.0.1]:2000' (ED25519) to the list of known hosts.
redon@127.0.0.1's password:
```

Si recuerdas lo que hablamos en la **Opción 2**, podrías conectarte a una VM creada con Vagrant usando *ssh* sobre el puerto 2222 del host. Vagrant usa exactamente este mecanismo para permitirlo.

Unattended-upgrades

Si lo que dijimos **antes** de comprobar periódicamente las actualizaciones de tu *Ubuntu* no es lo tuyo, el software **unattended-upgrades** de *Ubuntu* es la solución. Se trata de un paquete que permite la **instalación automática de actualizaciones de seguridad**. Esto significa que el SO instala las actualizaciones críticas sin intervención de ningún usuario, lo que reduce el riesgo de brechas de seguridad y mantiene tu sistema seguro.

Ten en cuenta que, gracias a él, ya no tendrás problemas porque olvides actualizar el sistema periódicamente, lo cual es algo muy interesante, especialmente si es un sistema que está en producción.

Este paquete es la forma estándar de **aplicar automáticamente importantes correcciones de errores y parches de seguridad en Ubuntu**. Una vez instalado el paquete con `sudo apt install unattended-upgrades`, puedes activar las actualizaciones automáticas ejecutando el siguiente comando: `sudo dpkg-reconfigure -plow unattended-upgrades`

El fichero de configuración de las actualizaciones desatendidas o automáticas es `/etc/apt/apt.conf.d/50unattended-upgrades`. En *Debian* y en *Ubuntu*, la configuración estándar de `unattended-upgrades` solo aplicará las actualizaciones de seguridad. Si quieres modificar este comportamiento, deberás editar el archivo y modificar algunos parámetros. **Más información:**

- <https://geekland.eu/actualizaciones-automaticas-en-nuestro-equipo-con-unattended-upgrades/>
- <https://dev.to/bearlike/why-and-how-to-use-unattended-upgrades-on-a-ubuntu-server-4hoc>

 **Es importante destacar que, si bien este paquete puede ser muy útil para mantener tu sistema actualizado, también debes tener cuidado, ya que algunas actualizaciones pueden causar problemas de estabilidad del SO o romper dependencias. No obstante, en este curso eso no es un problema serio, por lo que puedes activarlas sin miedo si quieres.**