



Unidad 01 - Introducción y Virtualización

Angel Berlanas Vicente

September 21, 2020

Contents

1 Virtualización	2
1.1 Guest y Host	3
2 Configuración de la red en los entornos virtuales	4
2.1 Hardware de Red Virtualizado	4
2.1.1 VirtualBox y los JumboFrames	5
2.2 No conectado	6
2.3 NAT	6
2.4 Red NAT	8
2.5 Adaptador puente	9
2.6 Red interna	10
2.7 Solo anfitrión	11
2.8 Adaptador Genérico	12
2.9 Enlaces sobre la red	13
3 Recursos compartidos	14
3.1 Permisos y vboxusers	14
3.2 Audio	15
3.3 Puertos Serie	15
3.4 USB	15
3.5 Carpetas compartidas	16
4 Tarea 06 : Compartir Recursos 1	17
4.1 Instalación de las Guest Additions	18
4.2 Instalando desde el terminal	19
4.3 Instalando paquetes requeridos	22
4.4 Continuamos la instalación	24
4.5 Portapapeles	25

5 Tarea 07 [CLIL] : Why is Virtualization Useful?

27

1 Virtualización

La virtualización es la tecnología que nos permite trabajar con el *Hardware* como si fuera *Software*.

Vamos a ver esto con un pequeño ejemplo.

Cuando el programa que ejecutamos en un ordenador es relativamente simple, muchas veces consiste en un único archivo ejecutable que cuando se lanza, realiza la función que debe hacer y luego termina. Si deseamos actualizarlo, desinstalarlo, copiarlo, o lo que se necesite, es sencillo: *basta con trabajar con el fichero*.

Sin embargo, incluso en los programas más simples, existen una serie de *dependencias*, que tienen que ver con los diferentes Sistemas Operativos, las versiones de las librerías gráficas, las de acceso a disco, la gestión de los USB, etc.

A medida que los programas van complicándose y haciéndose más y más complejos comienzan a tener subsistemas y ficheros que son cargados dinámicamente para realizar determinadas tareas (.dll).

Otros ficheros que son utilizados por parte del Software en ejecución son:

- Ficheros de datos.
- Ficheros de configuración.
- Imágenes: iconos, colores, etc.
- ...

Al final la instalación de un Software medio incluye una serie de ficheros ejecutables, librerías, recursos y configuraciones que deben ser *copiados* al Sistema Operativo destino para que se pueda ejecutar.

El proceso no es sencillo y muchas veces no es posible realizar una *vuelta al estado inicial*.

Si para el Software es complicado, para los Sistemas Operativos, que son piezas de Software que acaban muy vinculadas al Hardware, la instalación, desinstalación y configuración es al menos igual de complicada, *si no lo es mucho más*.

Para los Administradores de Sistemas (*aka.* vosotros :-), la realización de pruebas de instalación es algo que haréis muy habitualmente.



En cursos posteriores veréis no solo que se debe de instalar y adaptar el Sistema Operativo si no también todas las aplicaciones y servicios que forman parte de la solución.

Utilizando los Entornos de Virtualización, conseguimos emular el *Hardware* para tratarlo en estas pruebas (o incluso en la entrega final), como *Software*. Nos permitirá, sobre un único soporte *físico*, ejecutar diferentes entornos completos virtualizando:

- Discos Duros.
- RAM.
- Dispositivos de Entrada/Salida.
- Configuraciones de conectividad (*Networking*).

Iremos viendo más detalles de la Virtualización a lo largo del curso, ya que se trata de una de las herramientas más potentes para los Administradores de Sistemas.

1.1 Guest y Host

A medida que vayamos leyendo documentación veremos muchas maneras de referirse a las máquinas virtualizadas y a los hypervisores. Una de las nomenclaturas que más puede aparecer son las palabras Guest y Host :

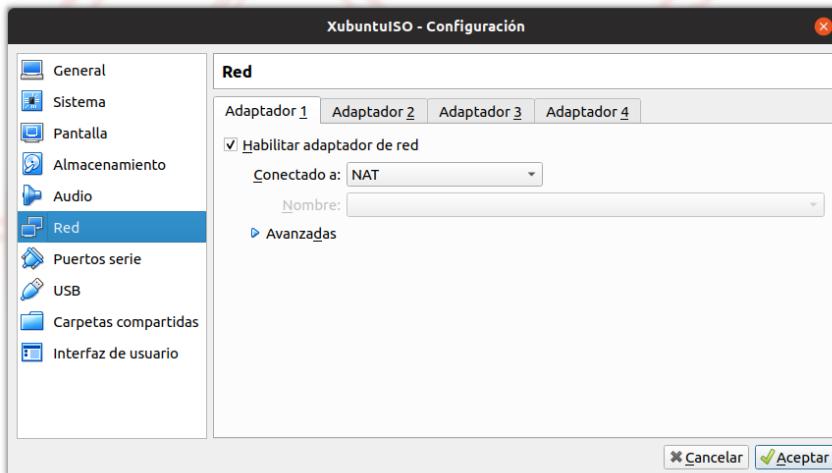
Guest (*invitado*) es la máquina virtualizada, la que vive "dentro" de la máquina real. **Host** (*anfitrión*) es la máquina real, la que dispone de hardware real y virtualiza las otras.

2 Configuración de la red en los entornos virtuales

En los sistemas de virtualización, sean cuales sean, uno de los recursos más importantes que tendremos que administrar es la red. En los sistemas de información, los servidores y los recursos compartidos entre las máquinas se comunican a través de la red en la mayoría de los casos. Para una correcta configuración de los Sistemas Operativos, es necesario que adquiramos las destrezas y conocimientos que nos permitirán ajustar la configuración de la red de nuestras máquinas. Ya sean Virtuales o físicas.

Las posibilidades que veremos a continuación son aquellas que nos permite VirtualBox, sin embargo estos mecanismos los podremos aplicar a casi todos los sistemas de Virtualización (*Hyper-V, Proxmox, Docker, ...*). La manera de configurarlos será distinta pero los conceptos son muy parecidos.

Cuando preparamos una máquina virtual, uno de los menús que nos muestra Virtualbox es el de la red.



En este menú podemos configurar cual va a ser el modo en el que esa máquina virtual podrá interactuar con la red a la que pertenece el Hypervisor. Tal y como veremos a continuación existen varios nodos y múltiples opciones, y deberemos seleccionar el que más se adecue a cada uno de los casos en los que nos podremos encontrar.

2.1 Hardware de Red Virtualizado

Para cada una de las tarjetas de red que podemos conectar a nuestras MVs, podemos seleccionar cual es *modelo* de tarjeta que utilizaremos.



Los modelos disponibles son estos:

- AMD PCNet PCI II (Am79C970A);
- AMD PCNet FAST III (Am79C973);
- Intel PRO/1000 MT Desktop (82540EM) [/Por defecto/];
- Intel PRO/1000 T Server (82543GC);
- Intel PRO/1000 MT Server (82545EM);
- Paravirtualized network adapter (virtio-net)

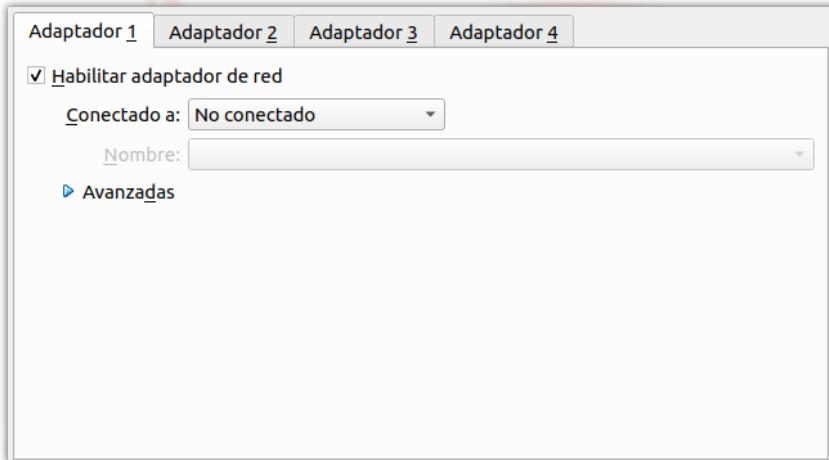
El modelo *Paravirtualized network adapter* es especial ya que requiere de un software adicional para la gestión de la red. Dejaremos el que viene por defecto en la mayoría de los casos de este módulo.

2.1.1 VirtualBox y los JumboFrames

Si estamos utilizando *Jumbo Frames* en nuestra red, es necesario que utilicemos los modelos basados en Intel, ya que no tenemos soporte para los mismos en el hardware Virtualizado de AMD. Sin embargo, esto lo veremos más adelante en el tema de compartición de recursos en la red (**NFS y SAMBA**) y estableceremos algunos parámetros que nos permitirán trabajar con los recursos de la manera deseada.

2.2 No conectado

En este modo VirtualBox le dirá al SO Virtualizado (*Guest*) que tiene una tarjeta de red pero esta no se encuentra **CONECTADA**, es decir es como si no tuviera cable conectado. Esto será útil para simular que la red se ha desconectado, nuestro Switch se ha caído, o diferentes ejercicios de simulación que podremos realizar.

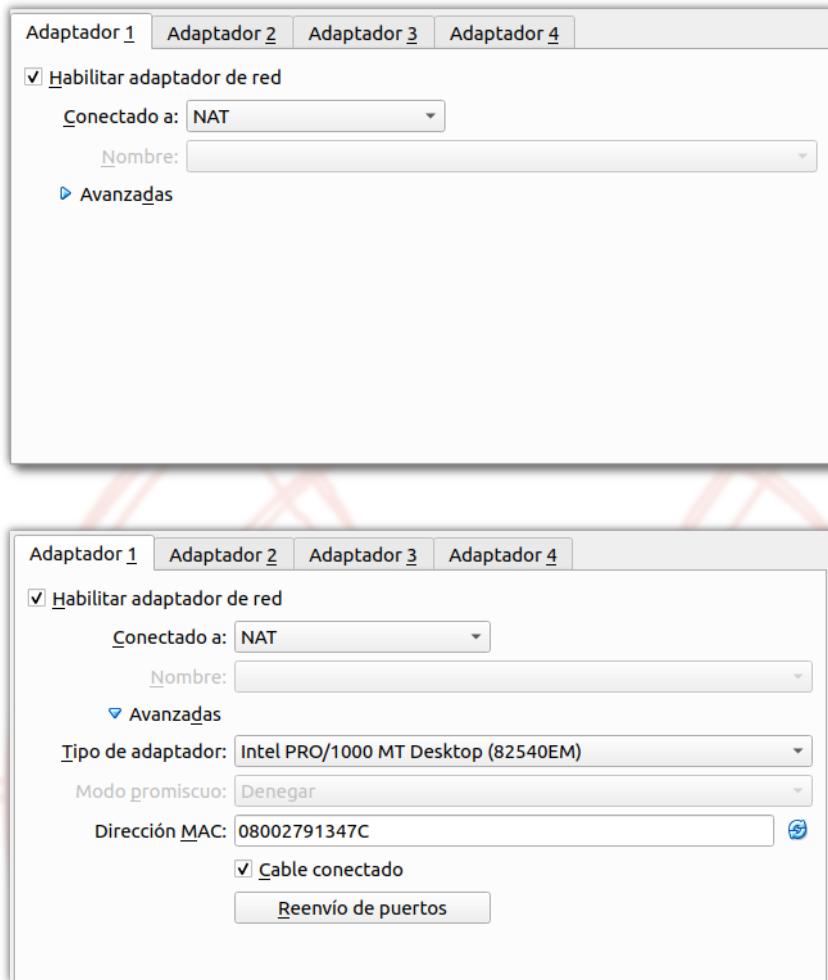


2.3 NAT

Network Address Translation (**NAT**) es el modo de conexión que viene por defecto cuando creamos una máquina virtual. Si estamos configurando una máquina para instalar cierto software, comprobar que todo funciona, o para realizar pruebas que no tienen que ver con servicios de la red, esta es la configuración más sencilla de utilizar.

En este modo, la IP asignada por defecto a la máquina es la :

10.0.2.15



Debemos tener en cuenta que en este modo tenemos algunas limitaciones que pueden afectar a nuestras máquinas/pruebas. Podemos comprobarlo en la página oficial del proyecto:

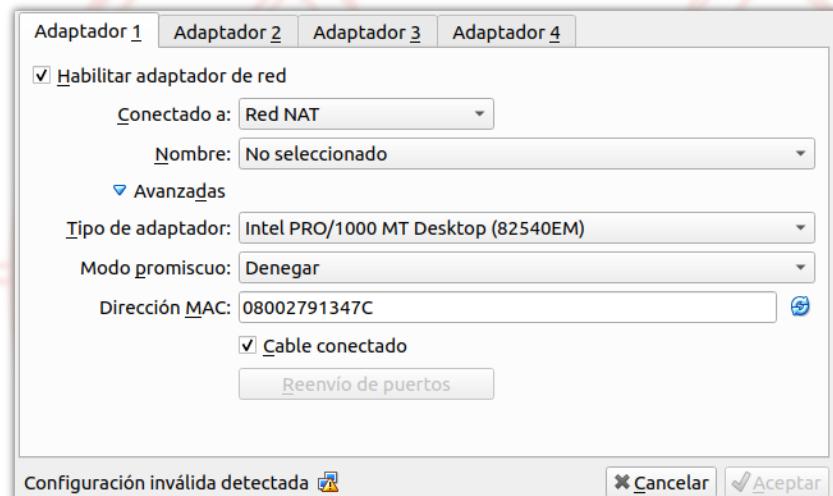
- [[<https://www.virtualbox.org/manual/ch06.html#nat-limitations>][VirtualBox Limitaciones de NAT]]

2.4 Red NAT

Este servicio funciona de una manera similar a como funciona nuestro Router en casa o en una pequeña empresa. Se agrupan los sistemas en una red y se impide que los sistemas ajenos a ella (más allá del router) puedan acceder directamente a ellos. Los sistemas que se encuentran conectados mediante esta red pueden utilizar TCP y UDP sobre IPv4 e IPv6 para comunicarse.

Todas las máquinas que conectemos de esta manera serán capaces de comunicarse entre ellas tal y como acabamos de comentar. La configuración de esta red se realiza cuando se crea por primera vez.

- [[https://www.virtualbox.org/manual/ch06.html#network_nat_service][VirtualBox Red NAT]]



2.5 Adaptador puente

La configuración de adaptador puente nos permite interactuar con el entorno de red del Hipervisor como si la Máquina Virtual se encontrara en la misma Red. Genera una nueva interfaz de red por software, accediendo a la red de manera física.

Si en nuestra red del Hipervisor tenemos un servidor de DHCP este le asignará IPs a las máquinas que se encuentren conectadas de este modo, permitiendo trabajar de manera conjunta a todas las máquinas, estén virtualizadas o no, ya que para la red se encontrarán todas al mismo nivel.

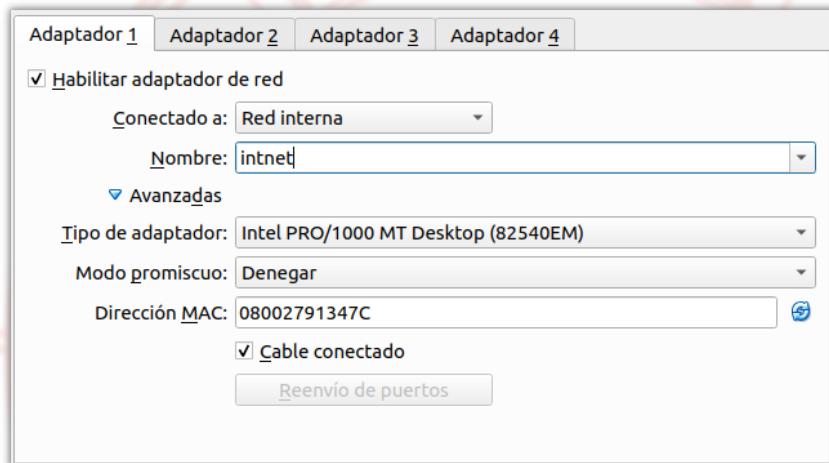


2.6 Red interna

Cuando configuramos las máquinas virtuales en este modo, lo que generamos es un *switch* virtual al cual podemos ir conectando máquinas, pero no podemos interactuar desde fuera del Hipervisor con esta red. Servicios como DHCP no están en este modo por defecto, teniendo que configurar todos los servicios de la red desde 0.

Podemos generar varias redes internas, que se crean automáticamente cuando les cambiamos el nombre en el menú de configuración. Por defecto el nombre de la red interna es **intnet**.

Este será el modo de conexión cuando tengamos que configurar los diferentes servicios que van asociados al dominio.



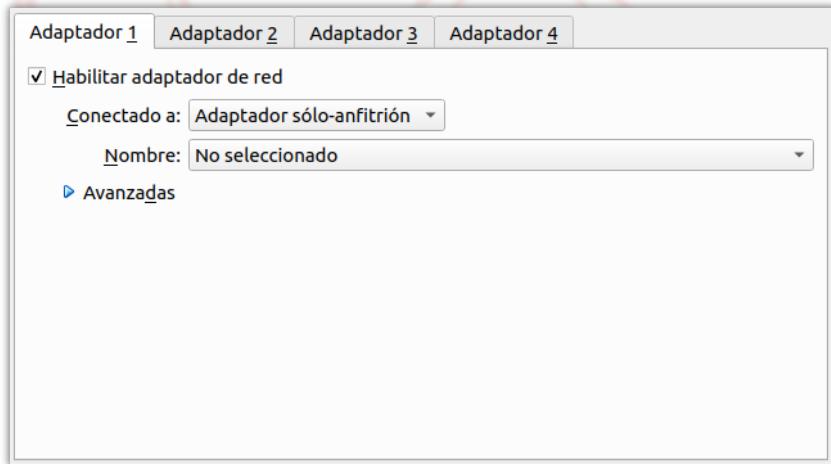
2.7 Solo anfitrion

Se trata de una configuración de red que es un híbrido entre el modo de *Adaptador puente* y *Red Interna*:

- Como *Adaptador puente* las máquinas virtuales pueden hablar entre ellas y con el hipervisor como si estuvieran conectadas mediante el mismo cable físico.
- Como *Red Interna*, la interfaz de red interna no existe fuera del hipervisor y no es posible contactar con las máquinas virtuales desde fuera.

Cuando se utiliza este modo, VirtualBox crea una interfaz de red mediante Software, que parece estar al lado de la tarjeta de red real del hipervisor. Se trata de una dirección de red al estilo de **loopback**.

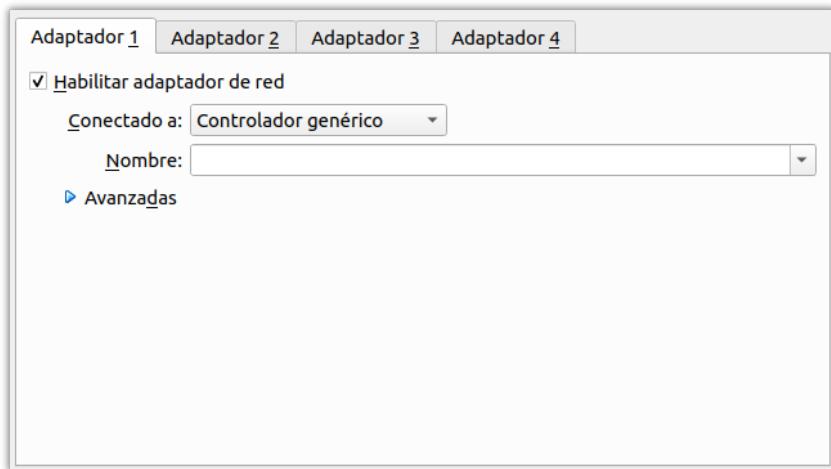
Este modo es particularmente útil cuando estamos preconfigurando sistemas que contienen un servicio o varios. Por ejemplo, una máquina virtual podría contener un Servidor Web y otra un servidor de Base de Datos, entre ambas deben poder comunicarse, el despliegue puede incluir órdenes para el VirtualBox para que genere una red de *Solo anfitrión* que comunique ambas máquinas. Una segunda conexión de red mediante *Adaptador Puente* conectada a la máquina que tiene el Servidor Web, nos permitirá acceder a esta desde cualquier máquina de la red del mundo exterior al hipervisor, sin embargo no podremos acceder a la máquina con el servidor de la Base de Datos.





2.8 Adaptador Genérico

Se trata de un modo similar al *Adaptador Puente* pero permite al usuario seleccionar el Driver que debe ser utilizado para emular esta tarjeta.





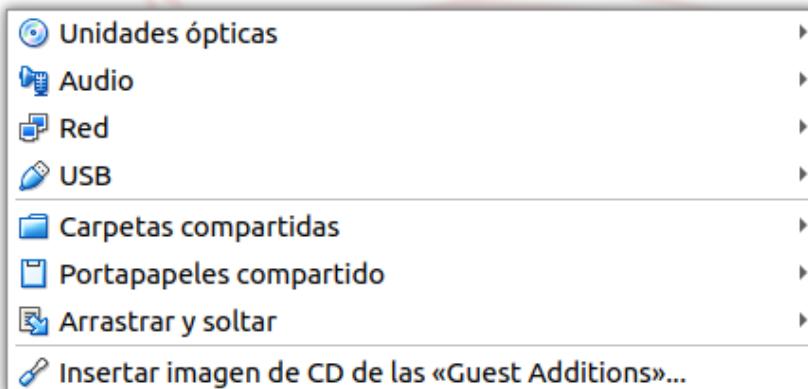
2.9 Enlaces sobre la red

- [Manual de VirtualBox](#)



3 Recursos compartidos

Hemos visto como se gestiona la red y los recursos de red en los sistemas de virtualización. Además de la red, existen otros como discos duros USB, Carpetas Compartidas entre máquinas, audio de los sistemas virtualizados, portapapeles, etc. que debemos saber configurar ya que nos permitirán trabajar mejor y establecer mejores soluciones entre sistemas.



3.1 Permisos y vboxusers

En GNU/Linux el acceso a los diferentes dispositivos hardware por parte de los usuarios (ya veremos más adelante todo esto en profundidad) está gestionado mediante pertenencia a grupos de usuarios.

Cuando instalamos VirtualBox en GNU/Linux, este añade un grupo de usuarios que es **vboxusers**, todos los usuarios que vayan a utilizar el VirtualBox, si queremos tener acceso a todas las prestaciones y operaciones posibles, debemos añadir ese grupo a los usuarios. Para ello, bastará con que ejecutemos en el terminal:

```
sudo adduser NOMBRE_USUARIO vboxusers
```

donde **NOMBRE_USUARIO** es el *login* del usuario que va a utilizar el VirtualBox.

Una vez realizada esta operación es necesario reiniciar, ya que los grupos y permisos son comprobados y establecidos durante el arranque de la máquina.

3.2 Audio

VirtualBox nos permite configurar el sistema de audio de la Máquina Virtual para que suene a través del sistema de sonido del anfitrión (Hipervisor).

3.3 Puertos Serie

Los puertos serie también pueden ser emulados dentro de las máquinas virtuales, esto puede ser usado para hacer *debug* o interactuar con algunos sistemas cuyo mecanismo de comunicación no es el habitual.

3.4 USB

Muchas veces cuando estemos trabajando con las máquinas virtuales necesitaremos copiar algo *dentro* o *desde* una memoria USB que conectamos al hipervisor, así como configurar ciertos dispositivos hardware dentro del sistema virtualizado.

En el menú de USB de la configuración de la MV podremos habilitar la compatibilidad con los dispositivos USB. Por defecto el modo de compatibilidad es 1.1, y este modo no nos dará ningún problema ya que viene por defecto en todos los VirtualBox, sin embargo si deseamos habilitar el USB 2.0 o el 3.0, se nos indica que debemos instalar el *Extension Pack* de VirtualBox para permitir el uso de este tipo de dispositivos.



En caso de que no lo tengamos instalado en el hipervisor e intentemos arrancar una máquina que si que tiene estas características habilitadas, lo

que ocurrirá es que no arrancará y nos mostrará un mensaje de error acerca del USB.



Los dispositivos USB que conectemos al VirtualBox, se dejarán de ver en el hipervisor, ya que el módulo del VirtualBox se *hará cargo* de los dispositivos y no le permitirá al hipervisor administrarlos. Esto lo tendremos que indicar en cada arranque de la máquina virtual, o establecer un filtro. Esto lo veremos una práctica.

3.5 Carpetas compartidas

Las carpetas compartidas es uno de los recursos más útiles que encontraremos en el VirtualBox, ya que nos permitirá el paso de ficheros cómodamente entre el hipervisor y el sistema virtualizado.

Veremos en una práctica como configurarlas, así como los permisos necesarios para poder hacer uso de las carpetas en los sistemas virtualizados.

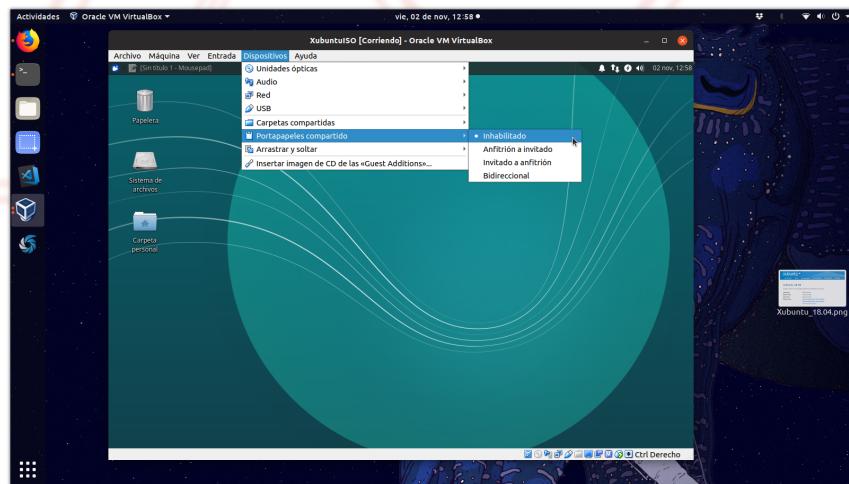
Estas carpetas pueden estar configuradas para permitir tan solo operaciones de lectura en los sistemas virtualizados, o configuradas en modo lectura-escritura (bidireccional) de tal manera que nos permiten pasar ficheros entre ambos sistemas, o más en el caso de que tengamos la carpeta compartida entre varias máquinas virtuales en el mismo hipervisor.

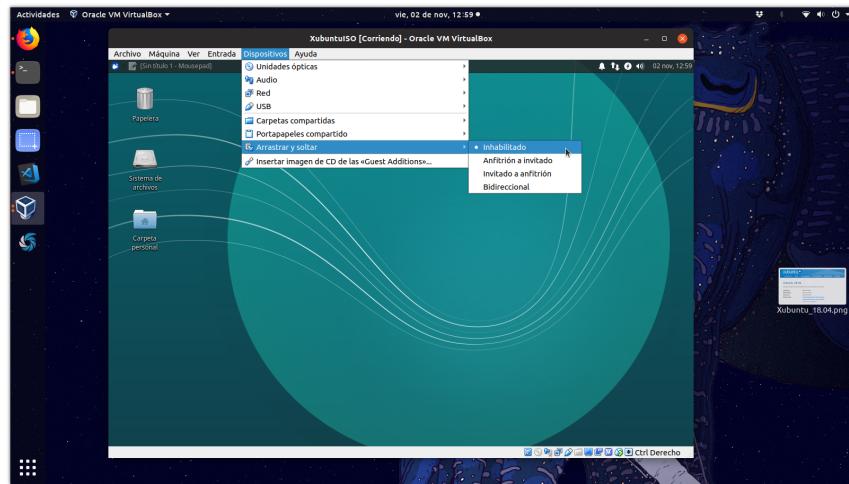
4 Tarea 06 : Compartir Recursos 1

Vamos a realizar una tarea en la que configuraremos una máquina Xubuntu para que podamos compartir una serie de recursos entre el Hipervisor y la máquina virtual.

Hemos visto en teoría que existen una serie de herramientas que nos permiten mecanismos de comunicación entre la máquina virtual y el hipervisor. No solo a través de la configuración de la red, sino también herramientas útiles para el trabajo diario con estas máquinas virtuales:

- Establecer carpetas compartidas entre el *Hipervisor* y la MV
- La posibilidad de añadir texto/enlaces/... en el hipervisor y pegar ese contenido en el sistema virtualizado (y *viceversa*).
- Permitir el uso de *arrastrar y soltar* ficheros entre el Hipervisor y la MV.





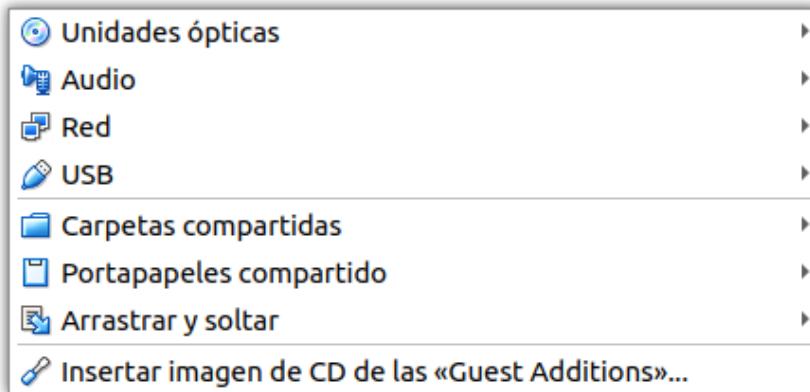
Para poder utilizar todas las posibilidades que nos ofrece VirtualBox para trabajar con las máquinas virtuales, utilizar memorias USBs en nuestras máquinas, pasarnos ficheros, copiar y pegar del portapapeles entre máquinas, etc. Necesitamos instalar en el sistema virtualizado una serie de programas que le indican al sistema operativo virtualizado como debe interactuar con esos recursos que le van a ser ofrecidos desde el hipervisor.

Estos programas se denominan las *Guest Additions* y a continuación veremos una serie de pasos que se deben seguir para su instalación en los sistemas GNU/Linux. En el caso de los sistemas Microsoft Windows, basta con que ejecutemos el *autorun* que nos aparecerá cuando insertemos el CD en nuestro sistema virtualizado.

4.1 Instalación de las Guest Additions

En el menú *Dispositivos* del VirtualBox y con la máquina virtual encendida y con la sesión iniciada, activamos la opción:

Insertar imagen de CD de las <>Guest Additions>>...



Cuando lo hacemos en el sistema virtualizado nos aparece en el Escritorio el ícono del CD que acabamos de insertar (*montar*), esto es un comportamiento habitual en los Escritorios de GNU/LinuX.

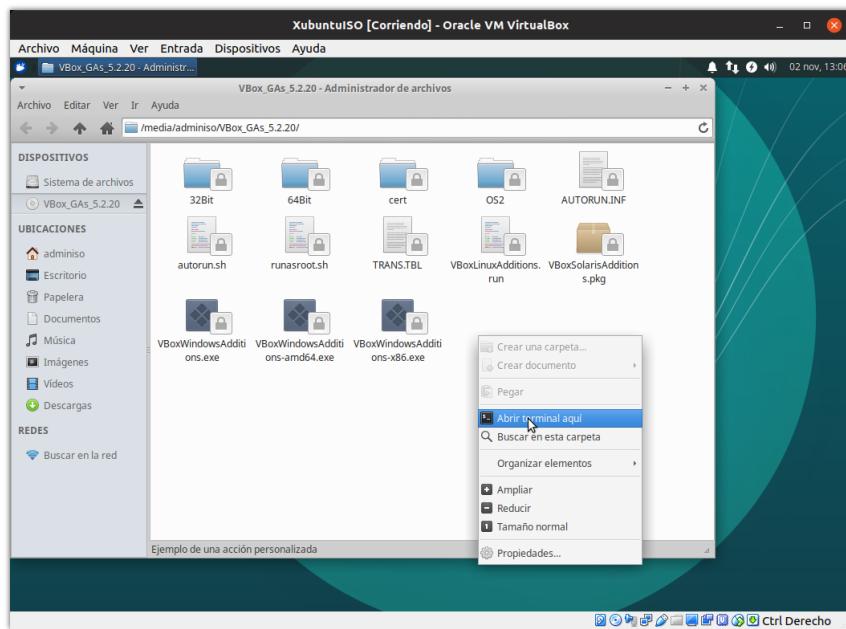
Abrimos el CD y vemos el contenido de ese CD, podemos observar que hay distintos ficheros, cada uno preparado para ser instalado en los diferentes sistemas y arquitecturas habituales que se virtualizan:

- Windows (**32 bits**)
- Windows (**64 bits**)
- GNU/LinuX
- Solaris

Lo que haremos será abrir una terminal aquí y ejecutaremos el instalador para GNU/LinuX. Como funciona la terminal lo veremos más adelante en el módulo, así como una explicación más detallada de las acciones que vamos a realizar ahora, sin embargo son un buen punto de partida para establecer un entorno de trabajo y pruebas más eficiente y cómodo.

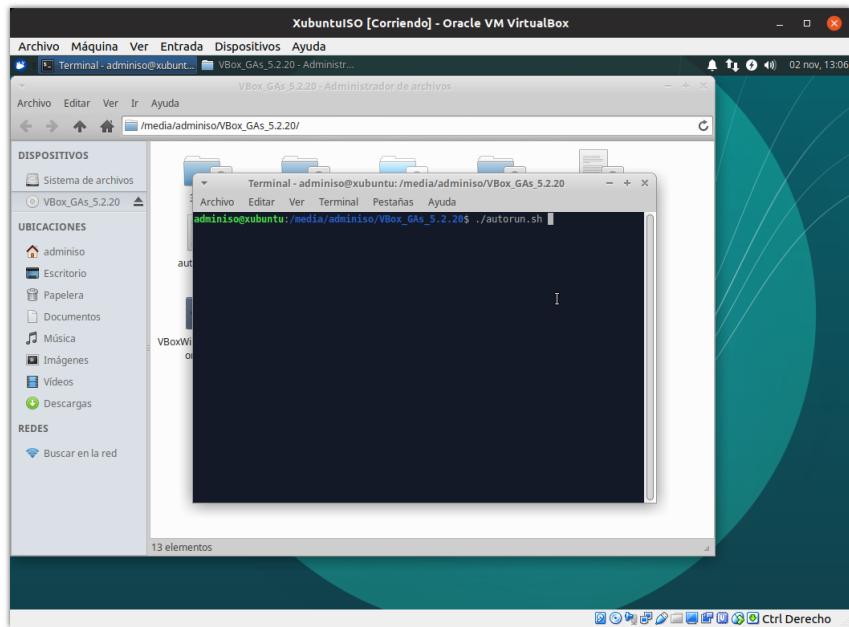
4.2 Instalando desde el terminal

Para abrir un terminal, pulsaremos botón derecho en el explorador de ficheros y seleccionaremos la opción: **Abrir un terminal aquí**.



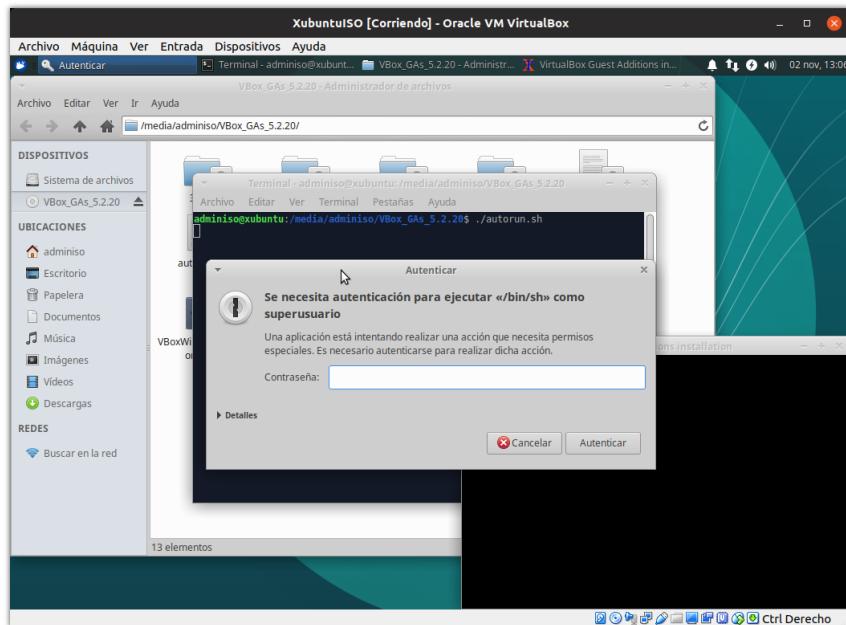
Una vez abierto ejecutaremos el script de *autorun.sh* que se encuentra en la carpeta, para ello escribiremos en la terminal:

`./autorun.sh`



Esta ejecución requiere de privilegios de administración, no sin motivo, ya que en realidad estaremos configurando una serie de drivers que tienen que ver directamente con nuestro Kernel (*núcleo*) y con como van a interactuar con ese hardware.

Introduciremos la contraseña de administrador.



Al ejecutarlo, nos aparece una pequeña terminal que nos muestra una serie de mensajes acerca del progreso de la instalación, si leemos los mensajes (y sí... toca leer este tipo de cosas, más vale que nos vayamos acostumbrando ya...), nos daremos cuenta de que nos pide *Por favor* que instalemos : *gcc*, *make* y *perl* para nuestra distribución.

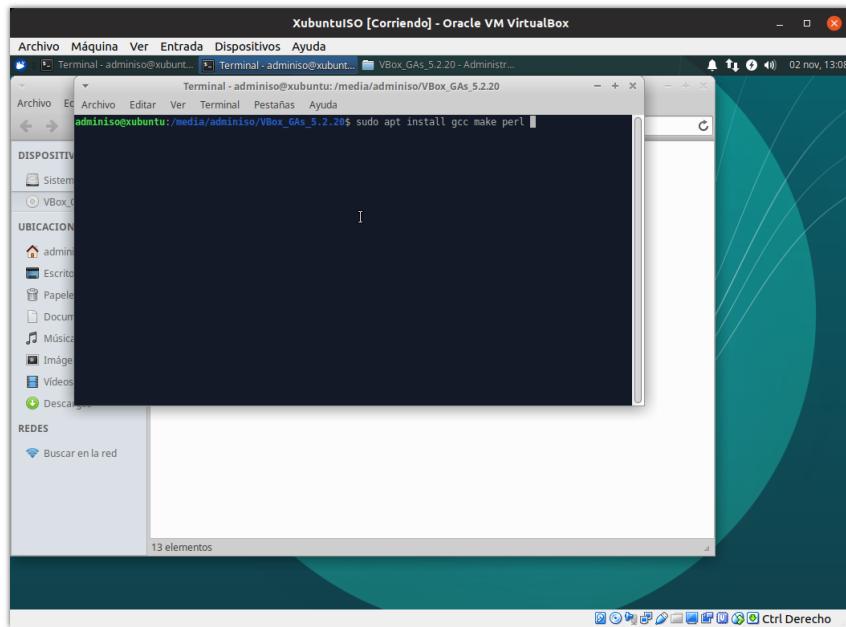
4.3 Instalando paquetes requeridos

Por ahora lo que haremos será ejecutar una orden en la terminal que nos descarga los paquetes desde los repositorios y nos los instala. Esto lo veremos con mucho más detalle en temas futuros, pero lo vamos a necesitar ahora.

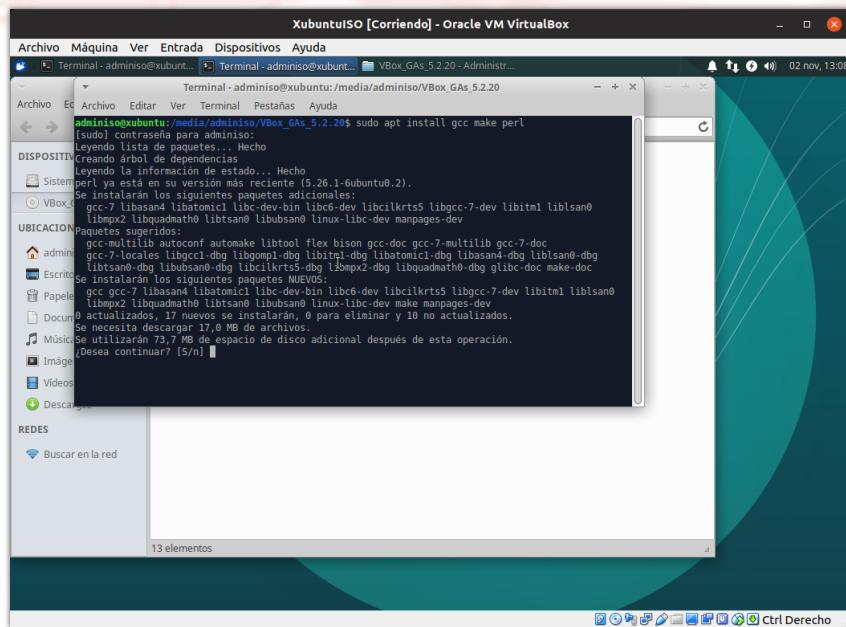
```
sudo apt install gcc make perl
```

Unidad 01: Sistemas Operativos]

1-SMX



A continuación nos muestra que se van a instalar una serie de paquetes y nos pide confirmación, tal y como vemos en la imagen.

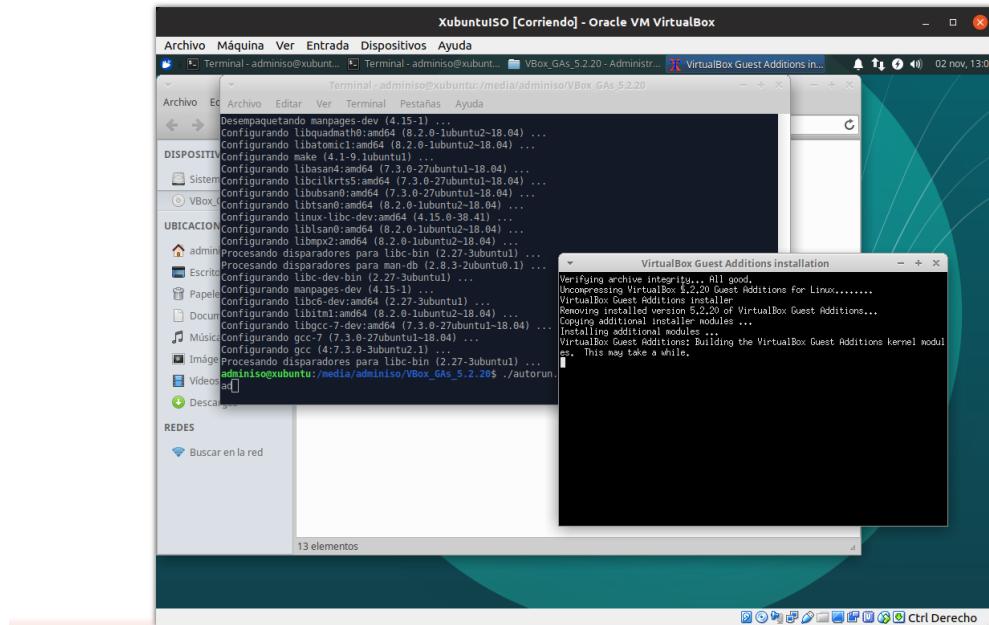


Unidad 01: Sistemas Operativos]

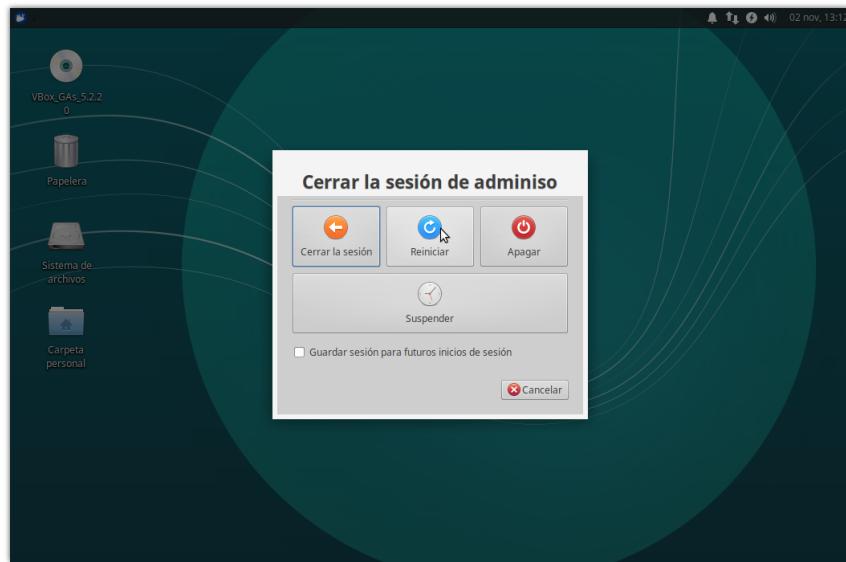
1-SMX

4.4 Continuamos la instalación

Ahora si volvemos a ejecutar el script de *autorun.sh* no nos dará ningún mensaje de error.



Una vez realizado este paso, es conveniente reiniciar el sistema virtualizado.

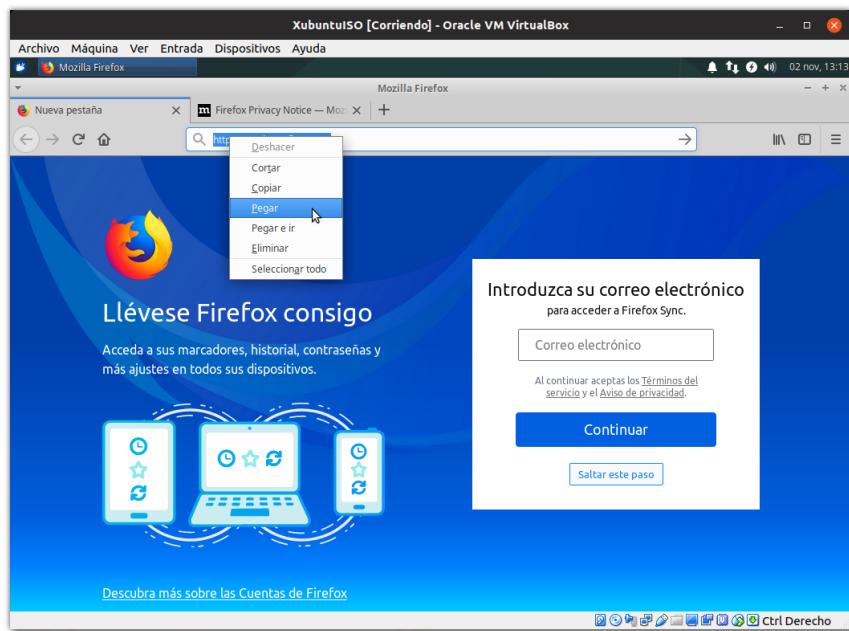


4.5 Portapapeles

Ahora ya tenemos el portapapeles compartido habilitado entre nuestro hipervisor y la máquina virtual, podemos decidir, la direccionalidad de este portapapeles.

- Inhabilitado
- Hipervisor -> MV
- MV -> Hipervisor
- MV <-> Hipervisor

Esto nos será muy útil para seguir manuales, buscar errores en Internet cuando la máquina virtualizada no tenga conexión o cuando no tengamos el navegador abierto, para la elaboración de documentación, etc.



USTOM_ID: ejercicio :EN

Realiza todas las configuraciones que hemos visto en la tarea y avisa al profesor para que compruebe que todo funciona.



5 Tarea 07 [CLIL] : Why is Virtualization Useful?

Lee el texto que aparece a continuación y contexta a las preguntas:

El texto ha sido extraido del Manual de VirtualBox: <https://www.virtualbox.org/manual/ch01.html#virt-why-useful>



— The techniques and features that Oracle VM VirtualBox provides are useful in the following scenarios:

- **Running multiple operating systems simultaneously** Oracle VM VirtualBox enables you to run more than one OS at a time. This way, you can run software written for one OS on another, such as Windows software on Linux or a Mac, without having to reboot to use it. Since you can configure what kinds of virtual hardware should be presented to each such OS, you can install an old OS such as DOS or OS/2 even if your real computer's hardware is no longer supported by that OS.
- **Easier software installations** Software vendors can use virtual machines to ship entire software configurations. For example, installing a complete mail server solution on a real machine can be a tedious task. With Oracle VM VirtualBox, such a complex setup, often called an appliance, can be packed into a virtual machine. Installing and running a mail server becomes as easy as importing such an appliance into Oracle VM VirtualBox.
- **Testing and disaster recovery** Once installed, a virtual machine and its virtual hard disks can be considered a container that can be arbitrarily frozen, woken up, copied, backed up, and transported between hosts. On top of that, with the use of another Oracle VM VirtualBox feature called snapshots, one can save a particular state of a virtual machine and revert back to that state, if necessary. This way, one can freely experiment with a computing environment. If something goes wrong, such as problems after installing software or infecting the guest with a virus, you can easily switch back to a previous snapshot and avoid the need of frequent backups and restores. Any number of snapshots can be created, allowing you to travel back and forward in virtual machine time. You can delete snapshots while a VM is running to reclaim disk space.
- **Infrastructure consolidation** Virtualization can significantly reduce hardware and electricity costs. Most of the time, computers today only use a fraction of their potential power and run with low average system loads. A lot of hardware resources as well as electricity is thereby wasted. So, instead of running many such physical computers that are only partially used, one can pack many virtual machines onto a few powerful hosts and balance the loads between them.



1. Realiza un listado de las palabras que no conoces su significado.
2. ¿Qué significa la expresión : *ship entire software configurations.*?
3. Explica la frase: *a virtual machine and its virtual hard disks can be considered a container that can be arbitrarily frozen, woken up, copied, backed up, and transported between hosts.*
4. ¿Qué nos aportan los *snapshots*? .¿Qué recursos del *anfitrión* utilizan?
5. En el último punto (*Infrastructure consolidation*), explica en castellano porqué las Herramientas de Virtualización pueden reducir el uso de los recursos energéticos.