




Sistemas Informáticos

UNIDAD DIDÁCTICA 3
INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS Y
VIRTUALIZACIÓN

CREADO POR: VANESA ESPÍN
1º DAW BILINGÜE
IES Politécnico HLANZ



Índice de Contenidos

Contenido

1.	Planificación de la Instalación de Sistemas Operativos	2
1.1.	Tipos de Instalaciones	2
1.2.	Planificación de la Instalación	2
	REGLAS ORDEN DE INSTALACIÓN	3
2.	Máquinas Virtuales.....	5
2.1.	¿Qué son las Máquinas Virtuales?.....	5
1.1.	Virtualización Intel (VT-x).	6
1.2.	Virtualización AMD (AMD-V).	7
2.	Tipos de máquinas virtuales.....	8
2.1.	Máquinas virtuales de sistema.	8
2.2.	Máquinas virtuales de proceso.....	8
3.	Técnicas de virtualización.	9
2.2.	Ventajas de la Virtualización	7
2.3.	Máquinas virtuales vs contenedores	7
2.4.	Software de Virtualización.....	11
	VMware.....	11
	Windows Server Hyper-V de Microsoft	¡Error! Marcador no definido.
	Oracle VM VirtualBox	12
3.	Instalación de SO.....	30
3.1.	INSTALACIÓN DE WINDOWS	30
	CONCEPTOS BÁSICOS SOBRE MSDOS (USO EN REPARACIÓN)	30
3.2.	INSTALACIÓN DE LINUX.....	31
	PUNTOS DE MONTAJE	32
	PARTICIÓN SWAP	33
4.	Restauración y Actualización de SO	33

1. Planificación de la Instalación de Sistemas Operativos

1.1. Tipos de Instalaciones

Cuando pensamos en instalar un sistema operativo, solemos pensar en una situación ideal: un ordenador nuevo, un sistema operativo en CD o DVD, un disco duro que vamos a ocupar en su totalidad con la nueva instalación y un asistente de instalación en el que pulsaremos el botón Siguiente en cada una de sus etapas. Actuaremos del mismo modo cuando el ordenador ya tenga contenido, pero no nos interese preservarlo. Sin embargo, aunque no vamos a entrar en demasiados detalles, hay muchas más posibilidades, que vamos a mencionar a continuación:

1. Una de las situaciones más frecuentes es que el ordenador donde realizaremos la instalación disponga ya de un sistema operativo anterior. En ese caso, podremos optar por dos vías:
 - i. Instalar una nueva versión del sistema operativo que ya tenemos, preservando los datos y los programas instalados. En este caso, estaremos realizando una *actualización* del sistema.
 - ii. Sustituir completamente el sistema existente por otro, que puede ser del mismo fabricante o no, recuperando los datos desde la copia de seguridad una vez concluida la instalación. Este proceso recibe el nombre de *migración*.
2. Cuando trabajamos con ordenadores que forman parte de una red local, en lugar de utilizar un CD o un DVD para hacer la instalación, el origen puede ser una carpeta, una imagen ISO o una unidad de almacenamiento compartido en red. Una variante de esto es el protocolo PXE (del inglés, Preboot eXecution Environment). Si un ordenador tiene habilitada esta función, durante los primeros pasos del arranque, buscará un ordenador desde el que obtener el software que le permita arrancar (llamado NBP, de network bootstrap program). De esta forma, un ordenador puede arrancar sin recurrir a un dispositivo de almacenamiento local, lo que le podría permitir varias cosas:
 - i. Arrancar un sistema operativo desde la red local.
 - ii. Iniciar un asistente de instalación.
 - iii. Ejecutar un programa que permita recuperar una copia de la instalación local.

Un problema al que no suelen enfrentarse los usuarios domésticos, pero sí los administradores, es la instalación de un mismo sistema operativo en multitud de ordenadores. En estos casos, para evitar la necesidad de permanecer delante del ordenador mientras se completa la instalación, es muy útil recurrir a un archivo de respuesta que contenga toda la información que necesita el sistema operativo para instalarse. Esta técnica se denomina ***instalación desatendida***.

En los últimos años ha empezado a trabajarse también con asiduidad con las *máquinas virtuales*. Por su especial importancia, dedicaremos un apartado completo a hablar detalladamente de este tipo de instalaciones.

1.2. Planificación de la Instalación

Indistintamente de que realicemos una instalación desde cero o estemos actualizando una instalación que ya se encuentra en funcionamiento, debemos planificar tanto los objetivos como las necesidades que se derivan de ellos, dedicando especial atención a:

1. Los recursos que deben estar disponibles (dispositivos de almacenamiento externo, impresoras, conexiones de red, ...).
2. Relación de elementos que reutilizaremos de la instalación anterior, si existen.
3. Detalle de elementos que deben adquirirse, incluida su valoración.
4. Número de ordenadores implicados y la relación entre ellos.
5. Estudio de la compatibilidad entre los diferentes componentes que participan en la instalación, tanto el hardware como el software, prestando especial atención a la disponibilidad de controladores de dispositivo para todos los elementos implicados y a la compatibilidad entre las diferentes aplicaciones, y de éstas con el hardware que emplearemos. Recuerda que el procedimiento será más viable cuanto más contenga sus gastos sin menoscabar los resultados. En realidad, ese es el principal objetivo de todo buen proyecto.
6. La disponibilidad de licencias de aquellos componentes que las necesiten.
7. Lugar en el que se ubicará cada elemento de la instalación.
8. Configuración de los distintos dispositivos.
9. Las funciones que desempeñará cada elemento.
10. Por último, deberá realizarse una estimación de futuro, para intentar vislumbrar si el sistema deberá evolucionar en tamaño o en funcionalidad a corto o medio plazo. De esta forma, trataremos de evitar, en la medida de lo posible volver a modificar a corto o medio plazo la instalación que estamos realizando ahora.

Algunos de los datos de esta planificación se incluirán también en la **documentación** sobre la instalación que recogeremos más adelante. Es importante dedicar a la fase de elaboración de la práctica todo el tiempo necesario, para no precipitarnos en las decisiones prestando especial atención a la organización y documentando cada aspecto de la tarea realizada. Al terminar, realizaremos una comprobación detallada de la instalación asegurándonos que cada elemento cumple sus funciones de forma satisfactoria y documentaremos los resultados obtenidos.

REGLAS DE ORDEN DE INSTALACIÓN

En el caso de querer hacer una instalación sencilla habrá que preparar la partición o particiones dependiendo del SSOO en cuestión. Además, en el caso de ser múltiple hemos de tener en cuenta el orden de instalación siguiendo varias reglas (además de todas las mencionadas en apartados anteriores):

- 1º. En el caso de instalar Microsoft y Linux, este último se ha de dejar para el final ya que Windows no respeta el arranque de Linux y lo elimina literalmente (al reescribir su MBR).
- 2º. En el caso de querer instalar varias versiones de Microsoft lo haremos en el orden de aparición de los mismos o como mucho por familias (por ejemplo, a pesar de tener que 7 es más antiguo que 10 se podría instalar en orden indistinto ya que ambos tienen el gestor de arranque BOOTMGR)
- 3º. El gestor de arranque se instalará en la partición **activa** del disco duro. Si es el primer sistema que se instala pondrá automáticamente activa dicha partición. En cualquier momento puedo cambiar la partición activa pero esto puede hacer que los sistemas arranquen o no. En concreto el funcionamiento de un gestor de arranque a la hora de instalarse es el siguiente:
 - a. Mira qué hay en la partición activa. Si entiende lo que hay lo tiene en cuenta (si es de la misma familia o anterior), si no, instala su propio gestor de arranque en el sector de arranque: lo que hace es decirle al sector de arranque que apunte al fichero ejecutable (BOOTSECT.DOS, NTLDR o BOOTMGR).

- b. Cuando se dice que un gestor de arranque “machaca” a otro no quiere decir que elimine ningún fichero asociado al anterior gestor, lo único que hace es “instalarse” en el sector de arranque, es decir: el sector de arranque apuntará a la dirección donde está el ejecutable del gestor de arranque que se acaba de instalar. Este fichero ha de estar en la partición en cuestión, si no he de copiarlo desde un CD/DVD del propio SSOO.
 - c. Podemos tener el caso de que a pesar de no entender todo lo que hay en el gestor de arranque actual reconozca instalaciones anteriores. Ej: instalando un NTLDR sobre BOOTMGR, pondría el NTLDR en el primer sector (“machacaría” el BOOTMGR, aunque realmente no elimina el fichero), pero si había un boot.ini anterior lo tendrá en cuenta y lo añadirá al nuevo boot.ini que va a crear. Lo mismo ocurriría si seguimos con el caso anterior y ahora volvemos a instalar un BOOTMGR, seguiría estando la BCD antigua del anterior sistema BOOTMGR (y por tanto tendría en cuenta lo que hubiera en dicha BCD).
- 4º. Si se instala MSDOS, WIN311 o Windows 9X se instalarán **obligatoriamente** en la primera partición C:, por lo que solo podremos tener uno de ellos simultáneamente (aunque MSDOS se podría tener junto con los otros dos compartiendo partición)

Para entender esto mejor vamos a hacer la Práctica 3.1. Ejercicios de orden de instalación (Ejercicios Orden Instalac.pdf)

2. Máquinas Virtuales

2.1. ¿Qué son las Máquinas Virtuales?

Al igual que un videojuego es un programa que permite desplegar un mundo virtual interactivo, una máquina virtual utiliza los sistemas operativos para poder interactuar con ella. Pero ¿cómo se crea una máquina virtual con un sistema operativo?

Existen básicamente dos métodos:

- A. Generar una máquina virtual desde cero.
- B. Importar una máquina virtual previamente creada y exportada.

Para construir la **máquina virtual** tenemos que asignar determinados recursos de hardware, como son espacio en disco duro, memoria RAM, número de procesadores, etc. que el anfitrión cederá o compartirá con el invitado.

Cuando tengamos nuestra máquina virtual el siguiente paso que debemos hacer consistirá en instalar un sistema operativo, ya sea Windows o Linux y funcionará con las mismas reglas que lo hace en un ordenador normal, actualizaciones, licencias, instalación de software adicional, etc.

Imaginemos un ordenador en el que tenemos instalado un Windows 10, si en dicho ordenador instalamos un software de virtualización y creamos una máquina virtual que corra Windows 11 por ejemplo, dicho ordenador estará realmente ejecutando dos sistemas operativos al mismo tiempo (el host y el guest), y todos sus recursos se estarán repartiendo entre ambos. Si dicho ordenador tiene por ejemplo 8 GB de RAM, podemos darle 4 GB a la máquina virtual, y nuestro ordenador seguirá funcionando con 4 GB, cosa totalmente aceptable. Sin embargo, si nuestro ordenador tuviera 1 GB de RAM únicamente tendríamos problemas, ya que 512 MB son muy pocos para trabajar con un sistema operativo de una forma correcta.

Para que la virtualización funcione aceptablemente bien se necesitarán ordenadores modernos y potentes, que puedan ceder recursos a sus sistemas invitados para que luego funcionen bien.

Es conveniente como mínimo contar con 2 GB de RAM, suficiente espacio en disco duro, y lo más importante, un microprocesador potente que pueda dividir su tiempo de proceso entre los dos SO.

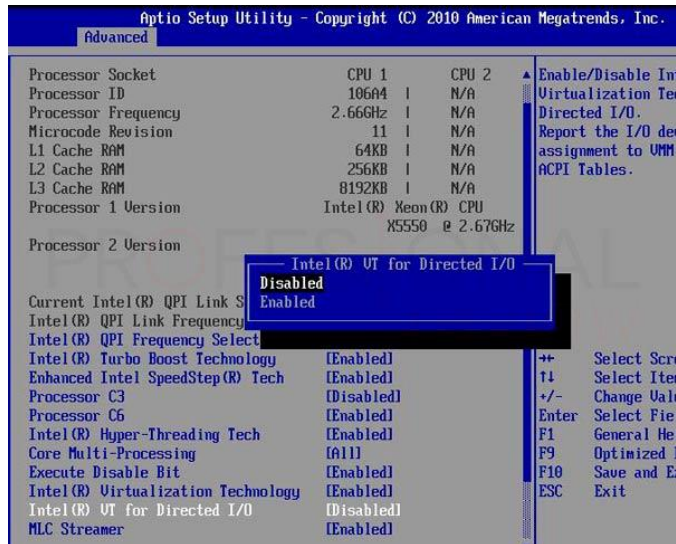
Realizar una virtualización completa mediante software es un proceso engorroso y lento, por eso los fabricantes de microprocesadores han incorporado en algunos de sus productos soluciones hardware especializadas en virtualización que han conseguido que la velocidad de las soluciones virtuales aumente considerablemente, siendo esta una de las principales causas del gran empuje que ha recibido la virtualización últimamente.

Intel y AMD han desarrollado independientemente extensiones de virtualización a la arquitectura x86. No son directamente compatibles entre sí, pero proporcionan las mismas funciones.

Las máquinas virtuales suelen ser bastante exigentes en cuanto a consumo de recursos, ralentizando o incluso impidiendo la ejecución de procesos. Sin embargo, existen ciertos aspectos, desde el *hardware* hasta la misma máquina virtual que, correctamente configurados, pueden mejorar su rendimiento.

Prácticamente todos los *procesadores* actuales, tanto de Intel (VT-x) como AMD (AMD-V), poseen extensiones que soportan la virtualización. Sin embargo, es posible que vengan desactivadas, por lo que es necesario acceder a la BIOS o UEFI, pulsando alguna de estas teclas al iniciar el equipo: 'F1', 'F2', 'F10', 'DEL' o 'Supr', y posteriormente, activar la opción que haga referencia a alguna de las extensiones antes mencionadas, que seguramente se encuentre en los menús de 'Processor',

'Chipset', 'Advanced Chipset Control' o 'Advanced CPU Configuration', dependiendo de la placa base y del fabricante.



Para terminar, se salvan los cambios y se sale para que el equipo arranque con la nueva configuración. Además, es necesario activar las extensiones desde el software de virtualización.

Virtualización Intel (VT-x).

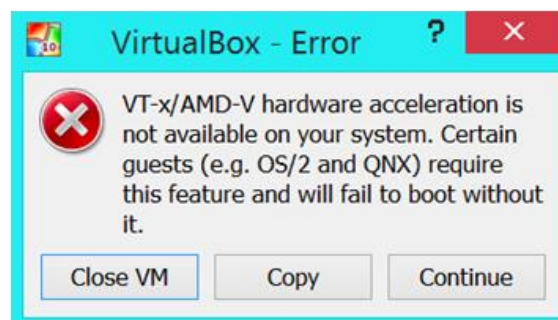
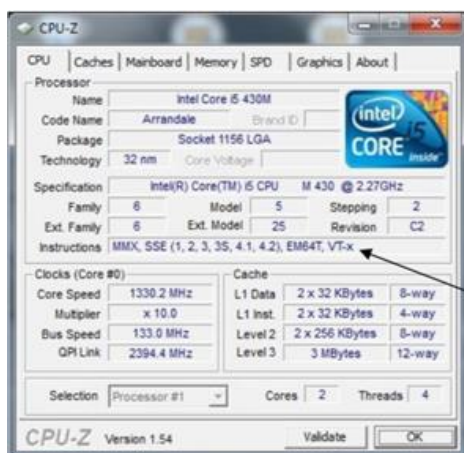
La extensión de Intel para virtualización de la arquitectura de 32 y 64 bits se llama IVT (Intel Virtualization Technology). Está disponible para todos los procesadores de última generación de Intel (Quad Core, I3, I5, I7) y en algunos modelos de los procesadores anteriores. Si queremos ver si nuestro procesador cuenta con estas extensiones VT que nos permiten trabajar sin problemas con la virtualización, podemos o bien instalar un software en Windows que nos informe de ello o bien mirar en la página de Intel donde viene la lista de procesadores con VT incorporado.

Software que nos da información sobre el micro:

- <http://www.cpubid.com/software/cpu-z.html>

Página de información de Intel:

- <http://ark.intel.com/VTList.aspx>



Virtualización AMD (AMD-V).

La extensión de virtualización AMD para la arquitectura de 64 bits x86 se llama AMD Virtualization (abreviada AMD-V). Los procesadores AMD que usan Socket AM3, Socket AM2, Socket S1 y Socket F incluyen AMD-V.

Evidentemente todos los micros AMD de nueva generación también incluyen estas extensiones.

El mismo software que nos daba información sobre micros Intel evidentemente también funciona para micros AMD.

2.2. Ventajas de la Virtualización

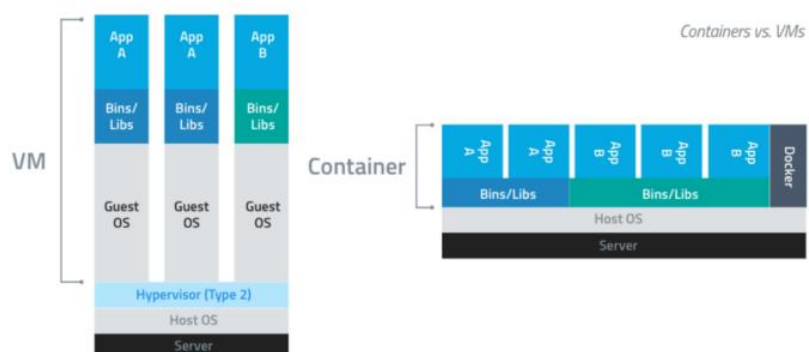
- ◆ Multiplica los recursos y aumenta la cantidad de servicios que se pueden ofrecer, además de agilizar y mejorar la gestión de estos recursos.
- ◆ Permite ahorrar dinero en ordenadores, servidores, etc, al implementarlos de forma virtual en el equipo físico ya disponible.
- ◆ Reduce el consumo de energía, ya que no hacen falta tantos equipos o dispositivos, con el ahorro energético que ello supone.
- ◆ Disminuye los recursos necesarios para optimizar los equipos, por ejemplo si tenemos varios servidores también sería necesario contar con equipos de refrigeración para ellos.
- ◆ Minimiza las tareas de mantenimiento ya que existen menos equipos físicos, lo que también supone un ahorro de tiempo.
- ◆ Aumenta la seguridad, ya que si una de las máquinas virtuales falla, el resto de particiones seguirán funcionando de forma independiente.

2.3. Máquinas virtuales vs contenedores

Recientemente han aparecido tecnologías que pretenden mejorar ciertos problemas que tiene la virtualización convencional.

- ◆ **Virtual Machines (VM):** En el caso que conocemos tenemos Apps que utilizan librerías y corren sobre sistemas operativos virtualizados por completo. Para que eso funcione es necesario un componente llamado Hypervisor que “muestra” el HW de la máquina para que sea virtualizado.
- ◆ **Contenedores (Docker):** En este caso tanto las librerías como el Sistema Operativo Host se comparten para ejecutar las aplicaciones. Es el Hypervisor de Docker (Hyper-V) el encargado de hacer esto.

¿Cuál es mejor?



2.4. Tipos de máquinas virtuales.

Existen dos tipos principales de máquinas virtuales:

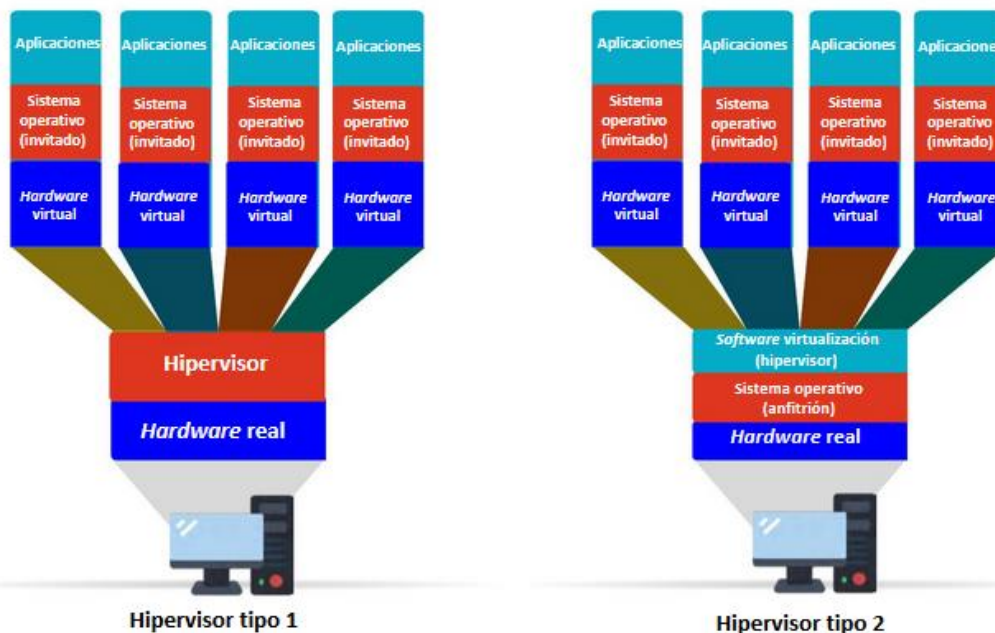
- Máquinas virtuales de sistema.
- Máquinas virtuales de proceso.

Máquinas virtuales de sistema.

Las máquinas virtuales de sistema, también llamadas máquinas virtuales de hardware, permiten a la máquina física dividirse entre varias máquinas virtuales, cada una ejecutando su propio sistema operativo. Este tipo de máquinas es de la que hemos hablado hasta ahora.

A la capa de software que permite la virtualización se la llama **monitor de máquina virtual, hypermonitor o hipervisor**, y pueden ser de dos tipos:

- *De tipo 1.* El hipervisor o monitor corre directamente sobre nuestro hardware y nos permite crear máquinas virtuales, por lo tanto, desaparece la necesidad de contar con un sistema operativo anfitrión, solo tendremos sistemas huéspedes, y el anfitrión será directamente nuestro monitor o hipervisor.
- *De tipo 2.* Es el que hemos visto anteriormente, en el cual un sistema operativo corre sobre el hardware del sistema, montamos un monitor o hipervisor sobre dicho sistema operativo anfitrión, y este monitor crea los sistemas operativos invitados.



Máquinas virtuales de proceso.

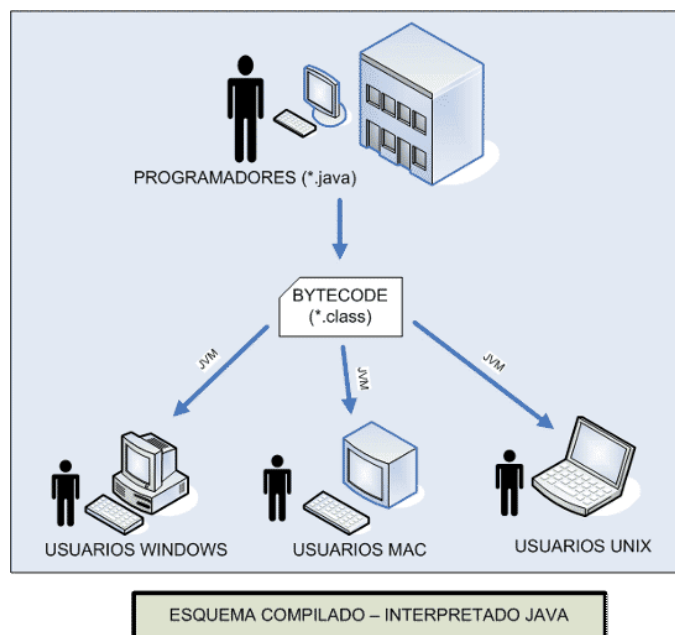
Una máquina virtual de proceso, a veces llamada "*máquina virtual de software*", se ejecuta como un proceso normal dentro de un sistema operativo y soporta un solo proceso. La máquina se inicia automáticamente cuando se lanza el proceso que se desea ejecutar y se detiene cuando éste finaliza.

Una máquina virtual de proceso es menos ambiciosa que una de sistema. En vez de emular un PC por completo, ejecuta un proceso concreto, como una aplicación, en su entorno de ejecución. Suena algo esotérico, pero lo usas cada vez que ejecutas una aplicación basada en Java o basada en .NET Framework (librería de Microsoft para ejecutar programas C++ y en C#).

Esto es de utilidad a la hora de desarrollar aplicaciones para varias plataformas, pues en vez de tener que programar específicamente para cada sistema, el entorno de ejecución (es decir, la máquina virtual) es el que se encarga de lidiar con el sistema operativo.

Su objetivo es el de proporcionar un entorno de ejecución independiente de la plataforma de hardware y del sistema operativo, que oculte los detalles de la plataforma subyacente y permita que un programa se ejecute siempre de la misma forma sobre cualquier plataforma.

El ejemplo más conocido actualmente de este tipo de máquina virtual es la **máquina virtual de Java**.



Cuando vamos a ejecutar un programa en Java el compilador genera un paquete o *Bytecode* que es interpretado y ejecutado por la máquina virtual de java siendo independiente del SSOO donde nos encontremos.

2.5. Técnicas de virtualización.

Hoy en día se usan 3 técnicas de virtualización distintas:

- Virtualización nativa.
- Virtualización no nativa.
- Paravirtualización.

Virtualización nativa.

Cada máquina virtual puede ejecutar cualquier sistema operativo soportado por el hardware real del sistema. Así los usuarios pueden ejecutar dos o más sistemas operativos distintos simultáneamente

en computadoras “privadas” virtuales, pero siempre que dichos sistemas operativos puedan funcionar en la arquitectura hardware de la máquina física.

Así, por ejemplo, en un PC (arquitectura x86) podemos instalar máquinas virtuales para correr Windows, Linux, MacOS, etc. Sin embargo, no podríamos correr sistemas operativos que no puedan funcionar en arquitecturas x86 como sería por ejemplo un IBM AIX o el sistema operativo de una XBOX.

Esta técnica de virtualización es la más usada, y la única que vamos a tratar nosotros en este curso.

Virtualización no nativa.

Las máquinas virtuales también pueden actuar como emuladores de hardware, permitiendo que aplicaciones y sistemas operativos concebidos para otras arquitecturas de procesador se puedan ejecutar sobre un hardware que en teoría no soportan.

Por ejemplo, si ejecutamos en un PC un emulador de la consola de juegos Nintendo Wii, estamos creando una máquina virtual que emula la arquitectura de dicha consola sobre la arquitectura x86 de nuestro PC.

Algunas máquinas virtuales que usan esta técnica, aprovechan para emular un hardware que sólo existe como una especificación, es decir, un hardware común teórico. Esta técnica la utiliza por ejemplo la máquina virtual de Java, permitiendo así que el mismo programa se pueda ejecutar en un PC, en un móvil o en una consola de juegos. La misma técnica utiliza el motor de la máquina virtual de .NET.

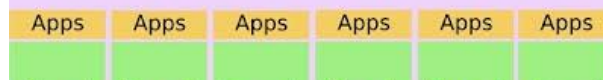
Paravirtualización.

Es una variante de la virtualización nativa. Consiste en permitir que los sistemas operativos que corren en las máquinas virtuales accedan en algunos casos directamente al hardware del sistema, de modo que las instrucciones lleguen directamente al hardware de nuestra máquina real o anfitrión, sin tener que ser traducidas y gestionadas por el hipervisor. Esto conlleva que esta *paravirtualización* es más rápida y eficiente.

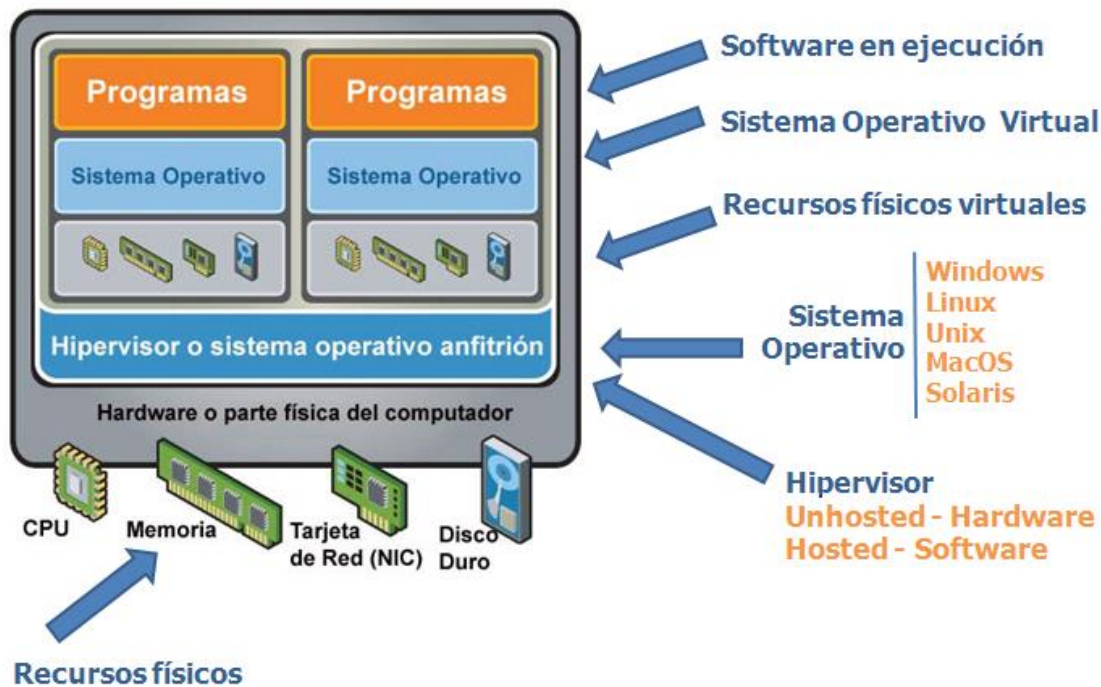
El gran problema de la *paravirtualización* es que los sistemas operativos no están preparados para funcionar dentro de una máquina virtual, de modo que el sistema operativo guest o invitado tiene que ser modificado íntegramente para poder ser utilizado. Es decir, no vale con instalar dentro de una máquina virtual un SO normal, sino que hay que tocar las fuentes de dicho SO, modificarlos y volver a compilar el SO antes de que pueda ser usado.

Evidentemente esto lo podemos hacer si el SSOO tiene una licencia de software abierto, como GNU/Linux, por ejemplo, pero es imposible realizarlo con SSOO de licencias cerradas como Windows o Mac OS, lo que implica que en las máquinas virtuales que se monten con este sistema de *paravirtualización* no podremos emular ningún sistema operativo comercial cerrado. En la actualidad los SSOO que han sido modificados son NetBSD, GNU/Linux y FreeBSD entre otros.

Paravirtualization



Máquinas virtuales - Arquitectura



2.6. Software de Virtualización

Cuando hablamos de software de virtualización, disponemos de un amplio abanico de posibilidades entre las que podemos encontrar opciones muy diferentes, que se adaptarán en mayor o menor medida al uso que pensemos hacer de ellas. Por este motivo, a continuación vamos a hacer un repaso de las herramientas más frecuentes, indicando sus características más destacadas:

VMware

VMware Inc. es, probablemente la empresa de referencia en el mundo de la virtualización. Dispone de dos líneas de producto diferentes:



1. **Las versiones de escritorio:** Pueden utilizar como sistema operativo anfitrión a cualquiera de los tres sistemas operativos de escritorio más extendidos: Microsoft Windows, GNU/Linux y Apple Mac OS X. En las máquinas virtuales que creamos, podremos instalar cualquier sistema operativo que sea compatible con la arquitectura x86 o x86-64. VMware comercializa tres productos diferentes en este segmento:
 - ◆ *VMware Workstation*, que es la versión comercial, para sistemas de escritorio que se instala sobre Microsoft Windows y GNU/Linux.
 - ◆ *VMware Fusion*, que ofrece una funcionalidad similar en sistemas de Apple que se ejecuten sobre procesadores Intel.
 - ◆ *VMware Player*, es la versión gratuita, con menos funcionalidad, orientada a un uso no comercial.

2. **Las versiones empresariales:** Están orientadas a implantaciones de mayor envergadura. Dos de las tres soluciones que propone la empresa se ejecutan directamente sobre el ordenador, ya que disponen de su propio sistema operativo, embebido dentro del producto, lo que deriva en un aumento del rendimiento. Se comercializan tres soluciones distintas:

- ◆ VMware ESX, que se instala directamente sobre el hardware, ofrece un alto rendimiento y facilita el uso, más o menos directo, del hardware por parte de los sistemas virtuales.
- ◆ VMware ESXi es un producto similar a VMware ESX, pero se ha eliminado la Consola de Servicio, ahorrando una considerable cantidad de almacenamiento en disco y en memoria RAM, pero esto hace que la administración de los hosts deba realizarse a través de un servidor de VirtualCenter. Además, desde el verano de 2008, existe una versión de pago y otra con licencia freeware. Se diseñó para ejecutarse desde dispositivos SSD (Solid-State Drive, en español, Dispositivos de Estado Sólido), pero puede instalarse en cualquier disco duro.

Oracle VM VirtualBox

Según diversas encuestas, VirtualBox es la opción preferida por la mayoría de los usuarios de sistemas de escritorio. VirtualBox se instala como cualquier aplicación sobre un sistema operativo anfitrión, que puede ser cualquier versión actual de escritorio de Microsoft Windows (a partir de XP), cualquier distribución de GNU/Linux, Apple Mac OS X, Solaris (incluido OpenSolaris) e incluso FreeBSD. Una vez instalada la aplicación, permite la ejecución, sobre ella, de casi cualquier sistema operativo compatible con la arquitectura x86.



Como todas las máquinas virtuales usan controladores de dispositivos virtuales, son independientes del hardware del anfitrión, por lo que son fáciles de transportar a un host diferente cuando sea necesario.

Manuales de VirtualBox en *somebooks.es*:

<http://somebooks.es/capitulo-2-virtualizacion-oracle-virtualbox/2/>

Los manuales de *somebooks.es* serán útiles para las tareas y las prácticas.

VirtualPC.

Microsoft también tiene una apuesta de software para facilitar la virtualización en los entornos de escritorio, que hoy en día puede parecer un poco desfasado puesto que la última versión es del 2007 pero que cumple perfectamente su función.



La virtualización con Virtual Pc es la propuesta de Microsoft, y está disponible para los distintos sistemas operativos de Microsoft y para Mac de forma gratuita.

Virtual PC está totalmente integrado en el sistema operativo anfitrión de modo que cuando se instala nos crea una carpeta en Mis Documentos llamada Mis Equipos Virtuales, donde por defecto irán los archivos de las máquinas virtuales que creamos. A la hora de crear una nueva máquina virtual nos aparecerá un Asistente para la creación de un nuevo equipo virtual, que nos guiará para elegir las opciones básicas a la hora de crear una nueva máquina virtual. Todas estas opciones las podremos modificar posteriormente desde la consola de Virtual PC.

Este hipervisor de tipo 2 es el más limitado de los que hemos visto hasta ahora, aunque cumple con su cometido de permitir instalar una máquina virtual Windows de forma fácil.

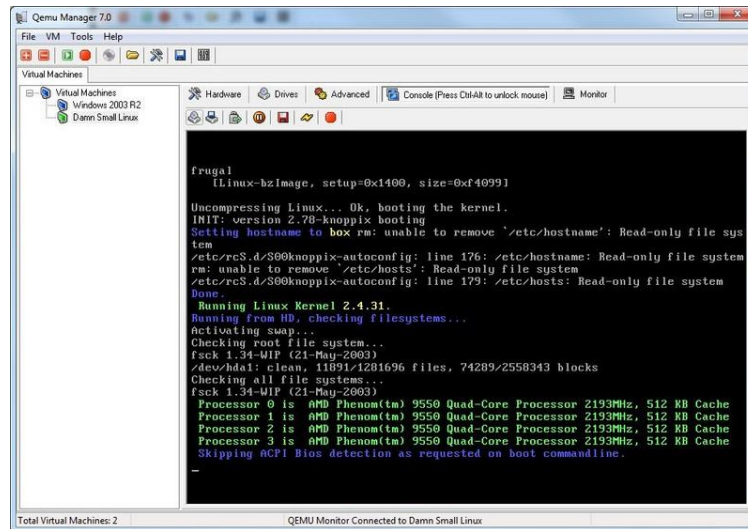
KVM.

KVM se encuentra integrado en el núcleo de Linux a partir de la versión 2.6.20 del Kernel.



Esta aplicación necesita soporte de hardware para ejecutar la virtualización, ya sea con procesadores de Intel o de AMD. En caso de que nuestro procesador no soporte dichas tecnologías de virtualización será inútil intentar instalarlo.

KVM es un hipervisor de tipo 2 de máquina completa que utiliza **Virt-Manager** como administrador de máquinas virtuales y **Qemu** como hipervisor.



Parallels Desktop.

Parallels Desktop es un software que proporciona virtualización de hardware para ordenadores Macintosh con procesadores Intel.



3. Virtual Box

Virtual Box será el software que usemos para virtualizar en clase.

3.1. Creación de una máquina virtual desde cero.

Para crear una máquina virtual desde cero es necesario, al menos:

1. *Asignar un nombre:* preferiblemente, descriptivo del sistema invitado.
2. *Elegir el tipo de sistema operativo que se va a instalar:* para que, automáticamente, el *software* de virtualización habilite/deshabilite ciertas configuraciones y asigne, por defecto, los recursos *hardware* dependiendo de los requisitos del sistema operativo.
3. *Indicar el hardware básico necesario:* como la memoria y el disco duro. Es importante tener presente que la memoria RAM asignada al sistema operativo invitado no podrá ser usada por el anfitrión

Al igual que un ordenador, la máquina virtual necesita un sistema operativo para poder funcionar. Para ello, se puede utilizar una versión *live* de un sistema operativo o instalarlo mediante el CD/DVD físico o desde un archivo de imagen de disco *.iso*, que es una copia exacta de un sistema de archivos, como un CD/DVD.

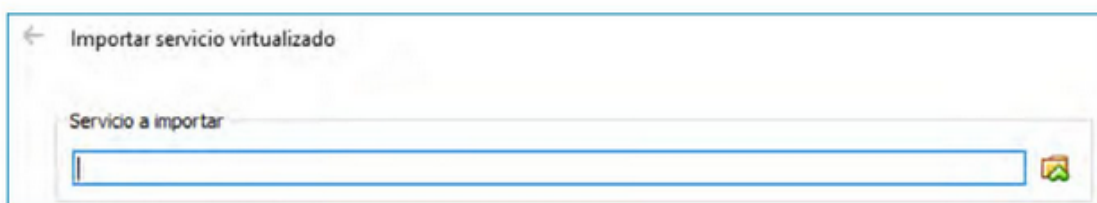
El proceso de creación desde cero de una máquina virtual difiere según el *software* de virtualización utilizado.

3.2. Importación de una máquina virtual existente.

La utilización de una máquina virtual creada se hace a través de su importación. Se pueden descargar máquinas previamente exportadas para Windows (<https://developer.microsoft.com/en-us/microsoft-edge/tools/vms>) o Linux (<https://descargarmaquinasvirtuales.com/>).

En *VirtualBox*. Para importar, debe seleccionarse ‘Importar servicio virtualizado’ e indicarse el correspondiente archivo *.ova* o *.ovf*.

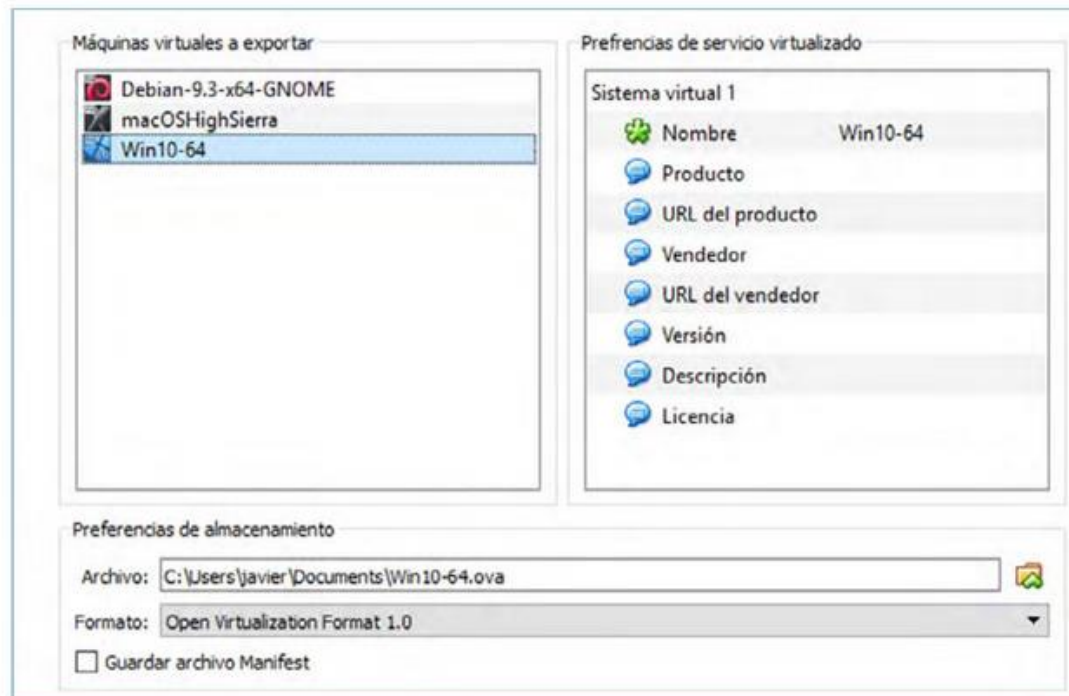
La exportación se realiza desde ‘Exportar servicio virtualizado’, indicando la máquina.



Importar máquina virtual ('Archivo → Importar servicio virtualizado').

RECUERDA

- ✓ Los archivos *.ova* (*Open Virtual Appliance*) son archivos comprimidos con Tar que contienen los archivos de disco virtual (*.vmdk* o *.vhd*), ficheros *.mf* (*manifest file*, opcional) y un *.ovf* con la definición de la máquina virtual.

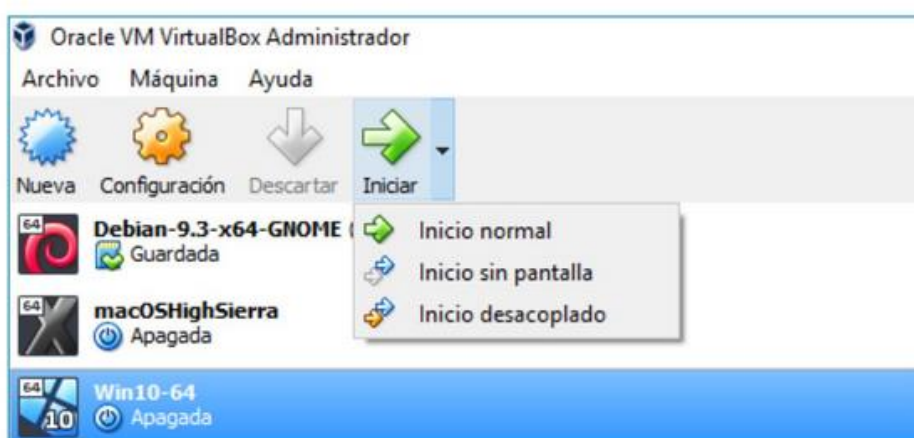


3.3. Primeros pasos con una máquina virtual.

Los procedimientos básicos para *trabajar* con una máquina virtual consisten en:

- *Iniciarla*: para encender la máquina virtual y arrancar el sistema operativo instalado.
- *Devolver el control al sistema anfitrión*: cuando se está operando en el sistema invitado.
- *Eliminarla*: cuando no se necesite.

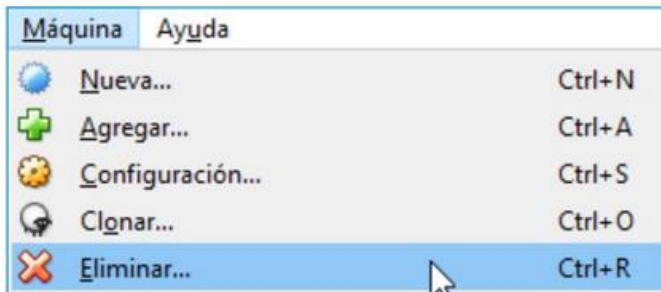
En *VirtualBox*. Para iniciar una máquina virtual con su interfaz (también se pueden iniciar en segundo plano, para utilizarla como servidor, por ejemplo) se selecciona y se hace clic en la flecha verde ('Iniciar').



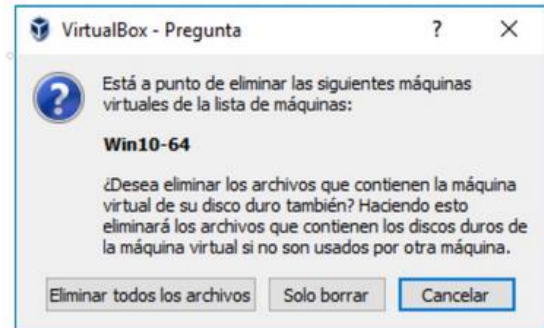
Arrancar una máquina virtual (flecha verde 'Iniciar').

Una vez dentro de la máquina virtual, la tecla ‘CTRL derecha’ permite devolver el control del teclado y del ratón a la máquina física.

El borrado se hace desde ‘Eliminar’. Esta opción permite ‘Eliminar todos los archivos’, para no dejar ningún rastro, o ‘Solo borrar’, eliminándola del administrador pero manteniéndola en la carpeta donde estuviera almacenada, con lo que, ejecutando el fichero de la máquina virtual con extensión *.vbox* o desde ‘Máquina → Agregar’, se vuelve a cargar.



Eliminar máquina virtual ('Máquina → Eliminar').

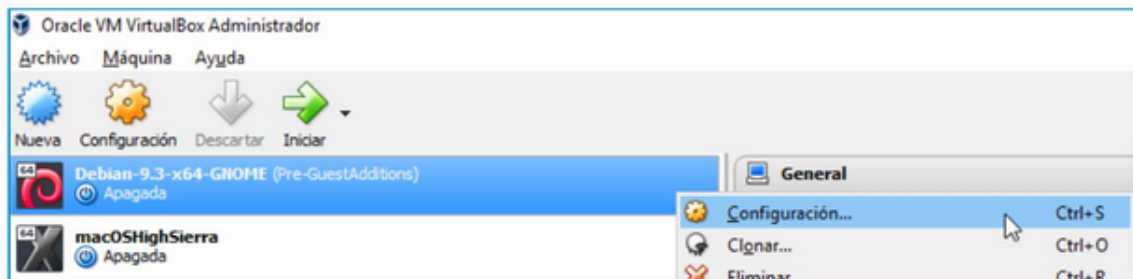


Seleccionar modo de borrado.

3.4. Configuración.

La configuración de las características del *hardware*, el sistema operativo anfitrión, el *software* de virtualización y de las propias máquinas virtuales permite optimizar su usabilidad, rendimiento y seguridad. Además, existen configuraciones específicas dependiendo del entorno de trabajo.

En VirtualBox, la configuración se realiza desde 'Máquina → Configuración' o haciendo clic derecho en la propia máquina y en 'Configuración'.



3.5. Usabilidad

Usar el sistema operativo invitado desde el anfitrión puede no ser muy intuitivo, *a priori*. Sin embargo, existen opciones que permiten mejorar la usabilidad para optimizar la experiencia del usuario trabajando con máquinas virtuales.

A) Instalar software de utilidades adicional.

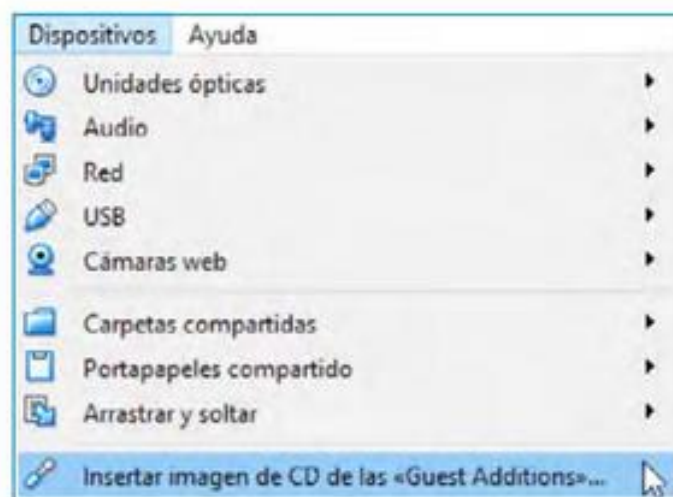
Es un *software* especial que se instala en el sistema operativo invitado para mejorar su usabilidad, rendimiento y añadir nuevas funciones:

- a) Permite a los usuarios mover el puntero del ratón sin problemas entre el anfitrión y el invitado.
- b) Mejora la capacidad gráfica de la máquina virtual (ajustando la resolución de pantalla de forma automática y permitiendo la aceleración de gráficos 3D).
- c) Habilita el portapapeles compartido bidireccional entre el sistema anfitrión y el invitado.
- d) Permite que el sistema operativo invitado tenga acceso de lectura y escritura a los discos duros del sistema operativo anfitrión.

Para instalar este *software* hay que tener en cuenta que:

- El sistema operativo invitado debe estar funcionando.
- El proceso de instalación depende del sistema invitado. Por ejemplo, puede requerir privilegios de administrador.
- Este *software* se instala desde un archivo *.iso*, por lo que hace falta una unidad de CD/DVD virtual. Además, si la ejecución automática del CD/DVD no está habilitada en el sistema invitado, es necesario hacerlo manualmente para ejecutar el archivo de instalación.
- Es posible que, durante la instalación, el ratón no responda adecuadamente, por lo que se recomienda utilizar atajos de teclado.

El *software de utilidades adicional* en VirtualBox se llama ‘*Guest Additions*’, y se hace desde ‘Dispositivos → Insertar imagen de CD de las “Guest Addition”’.

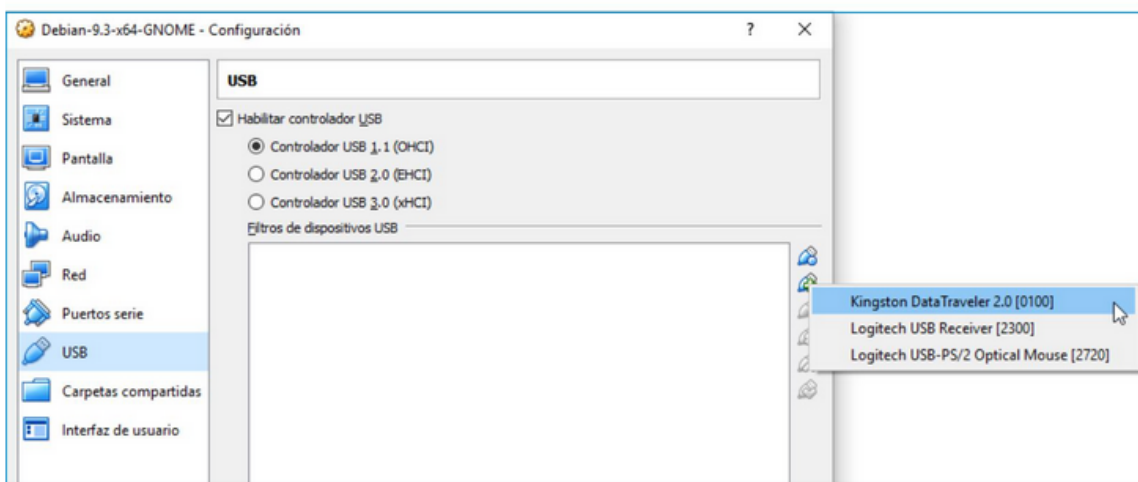


B) Comunicar anfitrión e invitado.

En entornos de trabajo con máquinas virtuales es fundamental que los sistemas anfitrión e invitado puedan *comunicarse* para poder utilizar dispositivos físicos como *pendrives*, discos duros externos o impresoras USB, o simplemente para compartir ficheros o texto. Existen principalmente tres métodos de comunicación directa entre anfitrión e invitado:

1. *Dispositivos USB*: permiten utilizar dispositivos USB conectados a la máquina física.

En *VirtualBox*. Es necesario habilitar el controlador USB que se necesite, según la máquina física. Para utilizar un dispositivo USB de forma “permanente”, es aconsejable conectarlo y seleccionarlo para crear un filtro con los campos establecidos. Si se crea un filtro con todos los campos vacíos, al encender la máquina, filtrará cualquier dispositivo USB conectado, con lo que la máquina física pierde el control de sus dispositivos USB (como el ratón o el teclado).



Configuración de dispositivos USB ('Máquina → Configuración → USB').

El principal inconveniente de los dispositivos USB compartidos es que, cuando el control pasa al sistema invitado, el dispositivo se desconecta del anfitrión, pudiendo ocasionar pérdida de datos. Además, es importante recordar que en distribuciones Linux es necesario añadir el usuario al grupo *vboxusers*. Para trabajar con dispositivos USB, hay que tener instalado el paquete de extensiones.

2. *Carpetas compartidas*: permiten compartir ficheros sin necesidad de una conexión de red. Pueden ser:
 - *Fijas* (Always enabled o *permanentes*): se pueden crear con la máquina virtual encendida o apagada.
 - *Temporales* (Enabled until next power off or suspend o *transitorias*): solo se pueden crear cuando la máquina virtual está encendida y desaparecen cuando se apaga.

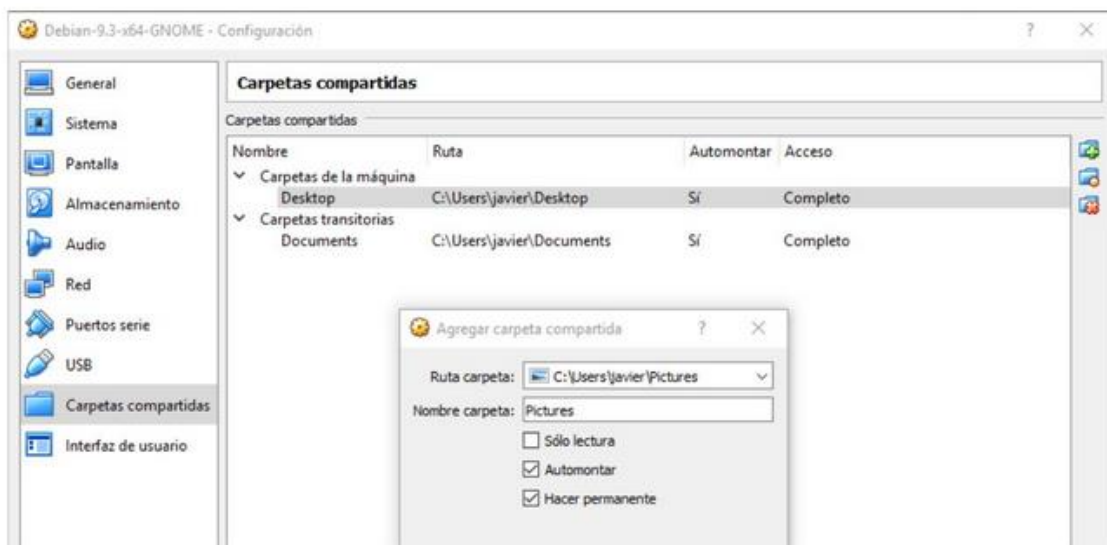
Su creación consiste en asignarle un nombre, seleccionar la ruta a la carpeta de la máquina física que se quiere compartir y habilitar las características deseadas:

- *Solo lectura* (Read-only): la máquina virtual no puede escribir en la carpeta compartida.
- *Automontar* (Enable this share): la carpeta se configura automáticamente al arrancar el sistema invitado.

El acceso a las carpetas compartidas difiere según el sistema operativo invitado:

- Unidad de red en Windows (es necesario tener activada la 'Detección de redes').
- Carpeta en `/mnt` o `/media`, en Debian.

En Virtual Box



Configuración de carpetas compartidas
(‘Máquina → Configuración →
Carpetas compartidas’).

En Windows, se accede desde Red y se selecciona `VBOXSRV` (nombre del servidor virtual). En Debian, la carpeta compartida se encuentra en `/media/sf_nombre-carpeta`.

Para usar las carpetas compartidas, es necesario que el usuario pertenezca al grupo `vboxsf` (en Debian), tener instaladas las Guest Additions y reiniciar la máquina:

```
# usermod -a -G vboxsf nombre_usuario
```

PARA SABER MÁS

Se pueden crear acceso directos a las carpetas compartidas para facilitar su uso:

- ✓ En Windows: desde el 'Explorador de archivos', hacer clic en el botón derecho en la 'carpeta compartida → Enviar a → Escritorio (crear acceso directo)'.
- ✓ En Debian: creando un enlace simbólico a la carpeta compartida:

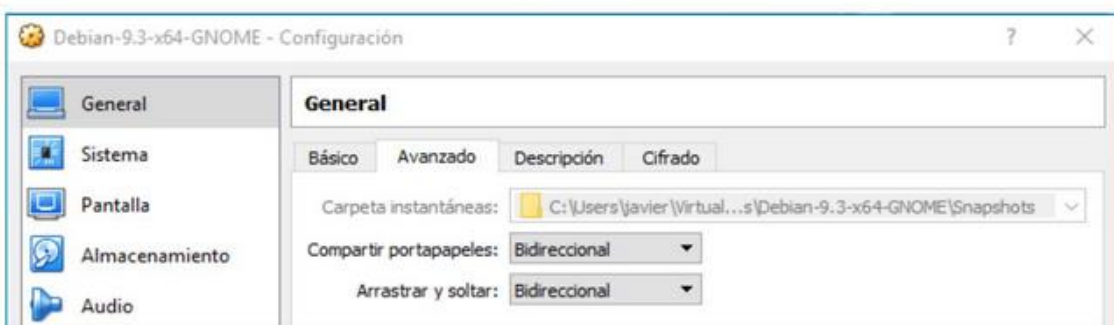
```
# ln -s /media/nombre-carpeta /home/nombre_usuario/Escritorio
```

3. Portapapeles compartido y arrastrar y soltar: ambas características permiten compartir información:

- *Portapapeles compartido* (copy and paste): mediante las acciones de copiar/cortar y pegar.
- *Arrastrar y soltar* (drag and drop): seleccionando y arrastrando un fichero o carpeta.

En *VirtualBox*. El portapapeles compartido está limitado a información de texto, es decir, no permite pasar archivos o carpetas. En el caso de arrastrar y soltar, actualmente, solo está disponible para sistemas Windows tanto invitado como anfitrión.

Para ambas funcionalidades se suele indicar el sentido de funcionamiento bidireccional. Es necesario tener instaladas las Guest Additions para usar estas características.



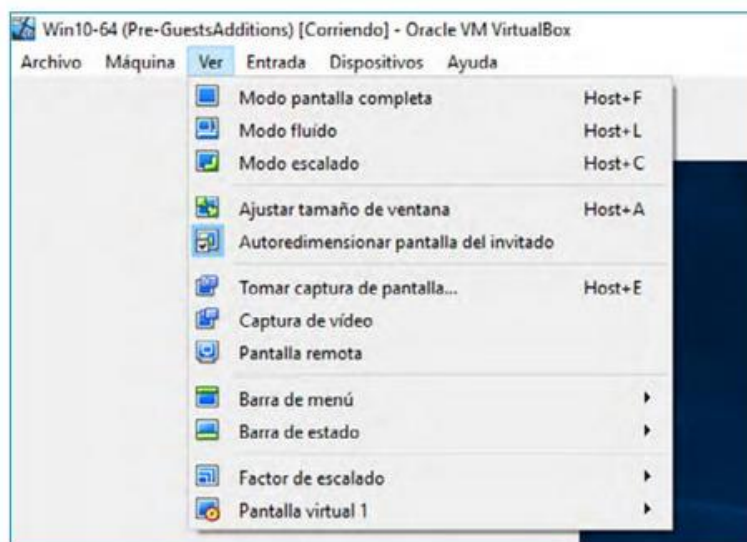
Configuración de portapapeles compartido y arrastrar y soltar ('Máquina → Configuración → General → Avanzado').

C) Visualizar la máquina virtual

A través de las *vistas*, el usuario puede determinar cómo ver las máquinas virtuales, adaptándolas a sus necesidades y su entorno de trabajo.

En *VirtualBox*. Existen diferentes formas de visualizar la máquina virtual, entre las que destacan:

- *Modo pantalla completa*: permite ver la máquina virtual ocupando toda la pantalla. Se habilita desde ‘Ver → Modo pantalla completa’.
- *Modo escalado*: visualiza la máquina virtual como una ventana más del sistema anfitrión. Se habilita desde ‘Ver → Modo escalado’.



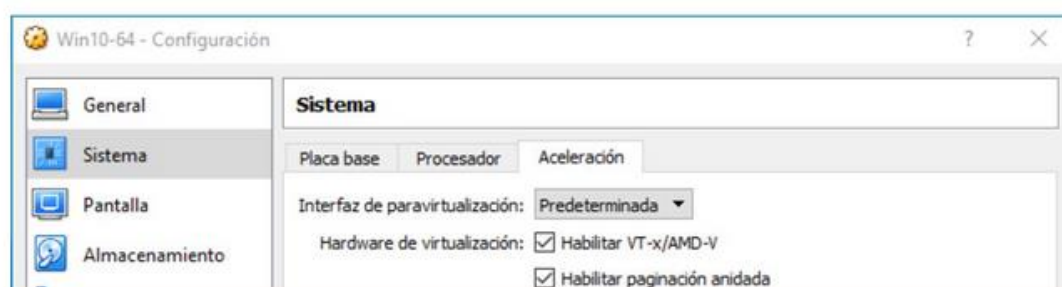
Modos de visualización ('Ver').

3.6. Rendimiento

Las máquinas virtuales suelen ser bastante exigentes en cuanto a consumo de recursos, ralentizando o incluso impidiendo la ejecución de procesos. Sin embargo, existen ciertos aspectos, desde el *hardware* hasta la misma máquina virtual que, correctamente configurados, pueden mejorar su rendimiento.

A) Activar hardware de virtualización.

En *VirtualBox*

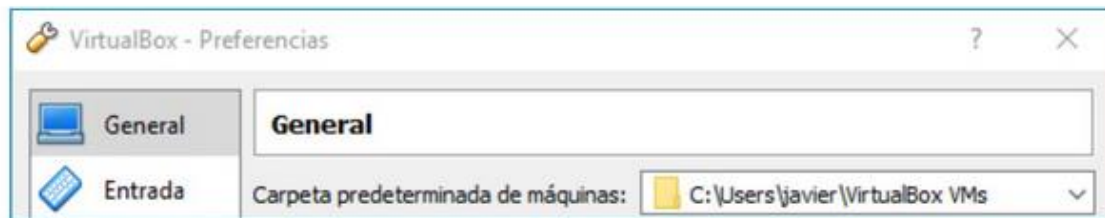


Activar *hardware* de virtualización
(‘Máquina → Configuración → Sistema → Aceleración’).

B) Guardar las máquinas virtuales en un disco SSD.

Los discos SSD proporcionan una velocidad muy superior a los discos duros convencionales, por lo que su uso permite mejorar el rendimiento de las máquinas virtuales instaladas en ellos. Por la misma razón, debe evitarse ejecutar máquinas virtuales desde medios extraíbles, ya que son bastante más lentos.

En *VirtualBox*. Por defecto, las máquinas virtuales se guardan, en Windows, en el directorio `C:\Users\nombre_usuario\VirtualBox VMs`, y en Debian, en `/home/nombre_usuario/VirtualBox VMs`. También se puede conocer su ubicación mediante la opción ‘Mostrar en explorador’ o ver todos los archivos de discos virtuales desde ‘Archivo → Administrador de medios virtuales’. Esta ubicación por defecto se puede modificar:



Ubicación por defecto de las máquinas virtuales
(‘Archivo → Preferencias → General’).

C) Crear discos de tamaño fijo.

Por defecto, al crear una máquina virtual, se utilizan discos reservados dinámicamente, puesto que el espacio en disco no suele ser un aspecto crítico. Sin embargo, los discos de tamaño fijo ofrecen un mayor rendimiento.

Esta configuración se realiza en el proceso de creación de un disco duro virtual.



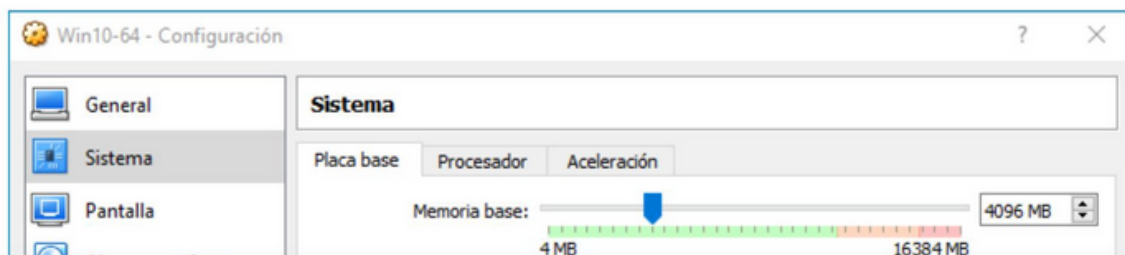
Creación de disco de tamaño fijo
(‘Máquina → Configuración → Almacenamiento →
Agrega una nueva conexión de almacenamiento’).

D) Asignar toda la memoria posible.

Una máquina física ejecuta, a la vez, el sistema operativo anfitrión y una máquina virtual con el sistema operativo invitado, por lo que tiene que repartir la memoria RAM entre los dos sistemas. Dependiendo del sistema operativo invitado, este requerirá más o menos memoria, sin embargo, debe tenerse en cuenta que la máquina física necesita un mínimo para funcionar. Como la memoria RAM asignada a una máquina virtual puede cambiarse, siempre y cuando esté apagada, se pueden probar diferentes configuraciones hasta que funcione de forma óptima.

En función del sistema operativo invitado y de la cantidad de memoria instalada en la máquina real, el asistente ofrece información sobre diferentes configuraciones de la memoria virtual.

En *VirtualBox*

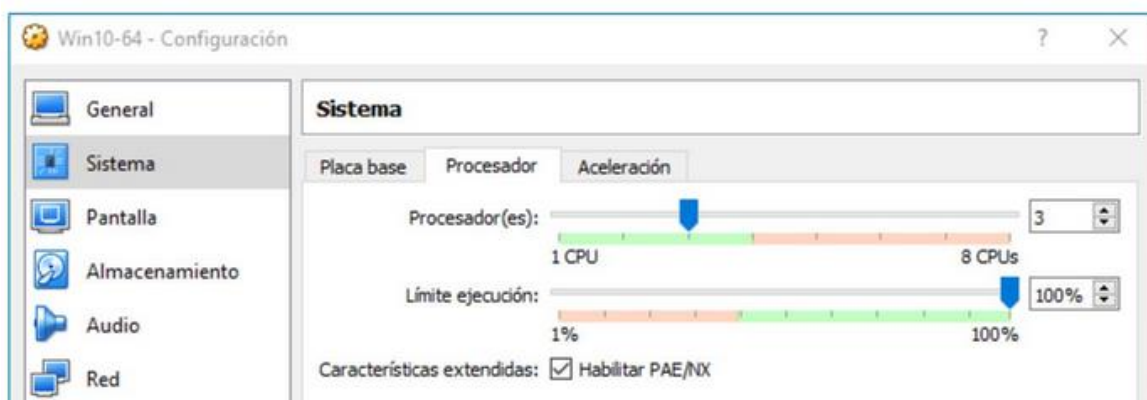


Configuración de memoria ('Máquina → Configuración → Sistema → Placa Base').

E) Asignar más núcleos del procesador.

Al igual que la memoria, el procesador es fundamental en el rendimiento de las máquinas virtuales. En el caso de procesadores multinúcleo, es posible asignar varios núcleos para la máquina virtual. La mejor técnica es probar varias configuraciones hasta conseguir el equilibrio entre el rendimiento de la máquina virtual y la física.

En *VirtualBox*

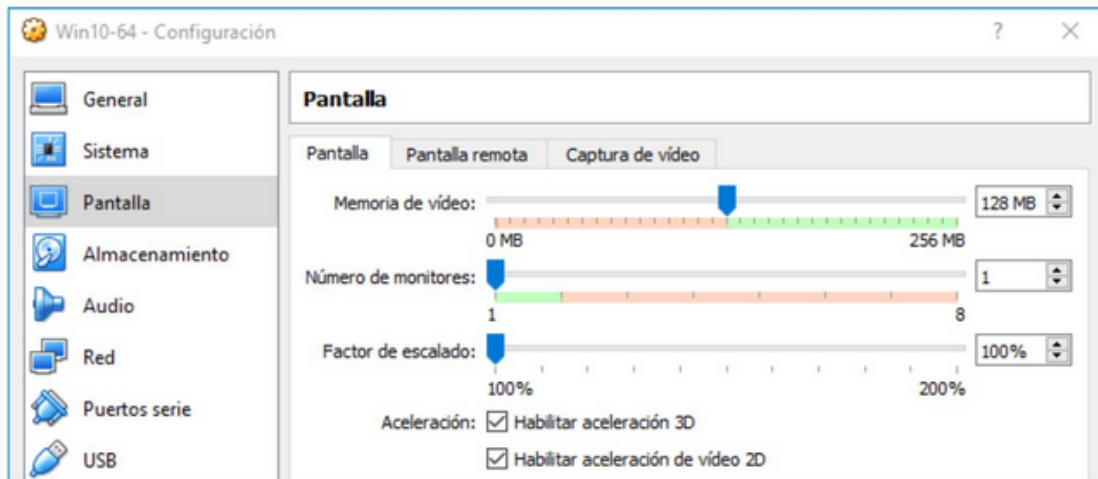


Configuración del procesador
('Máquina → Configuración → Sistema → Procesador').

F) Adaptar la configuración de vídeo.

Modificar algunas de las configuraciones de vídeo también puede mejorar la velocidad de las máquinas virtuales. Por ejemplo, la función de aceleración 2D en VirtualBox mejora la reproducción de vídeo, mientras que la aceleración 3D mejora la velocidad de algunas aplicaciones 3D, al igual que aumenta la memoria de vídeo. Sin embargo, estas modificaciones consumen más recursos de la máquina real.

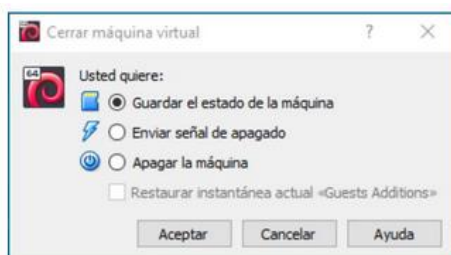
En *VirtualBox*



Adaptación de la configuración de vídeo
(‘Máquina → Configuración → Pantalla → Pantalla’).

G) Dejar la máquina en suspensión.

De forma similar a la hibernación o suspensión de un sistema operativo, es posible suspender la ejecución de una máquina virtual en lugar de apagarla. De esta forma se guarda el estado actual de la máquina virtual en el archivo de disco y, cuando se reanuda, lo hará en el instante en el que se suspendió.



Suspender una máquina virtual
en ejecución (‘Archivo → Cerrar’).



Reanudar una máquina virtual
en ejecución.

3.7. Seguridad.

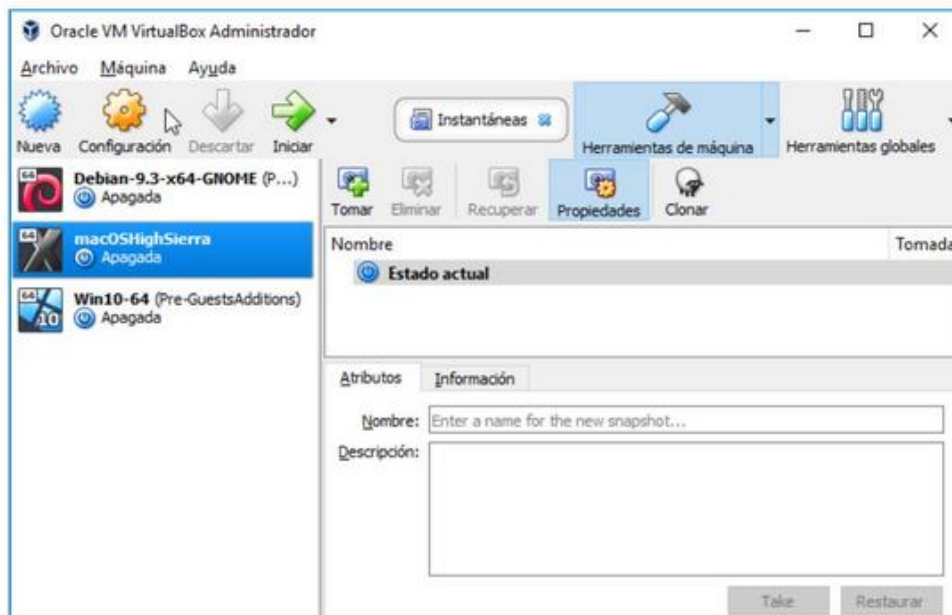
La configuración de la seguridad a nivel de máquina virtual permite conseguir un sistema seguro y fiable, gestionando la integridad, la confidencialidad y la buena disponibilidad de las mismas.

A) Restaurar el sistema.

Es una medida de seguridad que permite devolver al sistema invitado a un momento anterior. Esos momentos anteriores se denominan *instantáneas* (*snapshot*) e incluyen absolutamente todo el estado del sistema invitado: programas, configuraciones y datos. Resulta útil cuando se ha producido algún fallo en el sistema, se ha eliminado información valiosa o simplemente se quieren realizar diferentes configuraciones sobre un mismo estado del sistema.

Las instantáneas permiten el anidamiento, pudiendo organizarse como los capítulos de un libro. Además, se pueden crear independientemente del estado del sistema virtual (iniciado o apagado). El proceso para volver a un momento anterior se denomina *restaurar*, *recuperar* o *revertir*.

En *VirtualBox*



Gestionar instantáneas
(‘Herramientas de máquina → Instantáneas → Propiedades’).

También se pueden hacer desde ‘Máquina → Tomar instantánea’, cuando la máquina está iniciada.

B) Realizar copias de seguridad.

Frente a los problemas derivados de la eliminación de máquinas virtuales o máquinas virtuales corruptas a causa de virus, despistes, fallos físicos u otras causas, las copias de seguridad son una de las soluciones más económicas y eficientes de implementar.

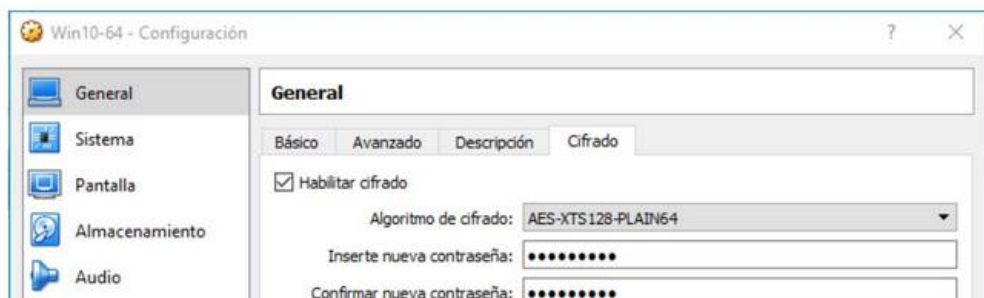
El procedimiento para hacer copias de seguridad de máquinas virtuales se denomina *clonación* y consiste en hacer una copia exacta de una máquina virtual existente, aunque con una identidad distinta, por lo que también sirve para disponer de forma rápida de un conjunto de máquinas virtuales similares operativas.

Existen dos tipos de clonación:

- *Clonación completa*: es una copia exacta e independiente de la máquina virtual.
- *Clonación vinculada o enlazada*: se crea una instantánea del disco duro virtual en la máquina original para utilizarla como disco de la máquina clonada. De esta forma, la máquina clonada depende de la original.

C) Cifrar una máquina virtual.

El cifrado permite encriptar y restringir el acceso a las máquinas virtuales mediante una contraseña.



Configurar cifrado

(‘Máquina → Configuración → General → Cifrado → Habilitar cifrado’).

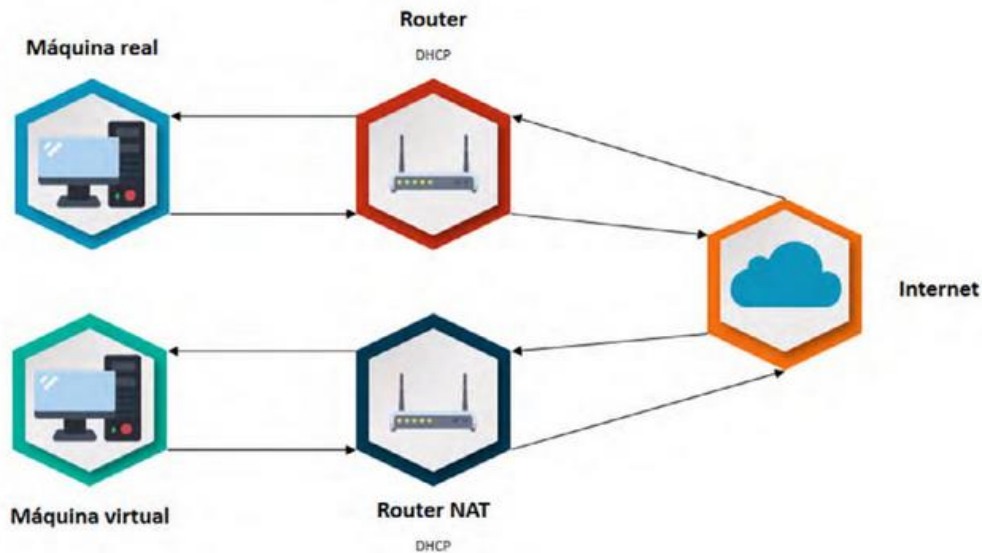
3.8. Configuración de red

En ocasiones, las necesidades del entorno de trabajo o del problema que se quiere resolver requieren configuraciones muy concretas, que se adapten para encontrar una solución más eficiente.

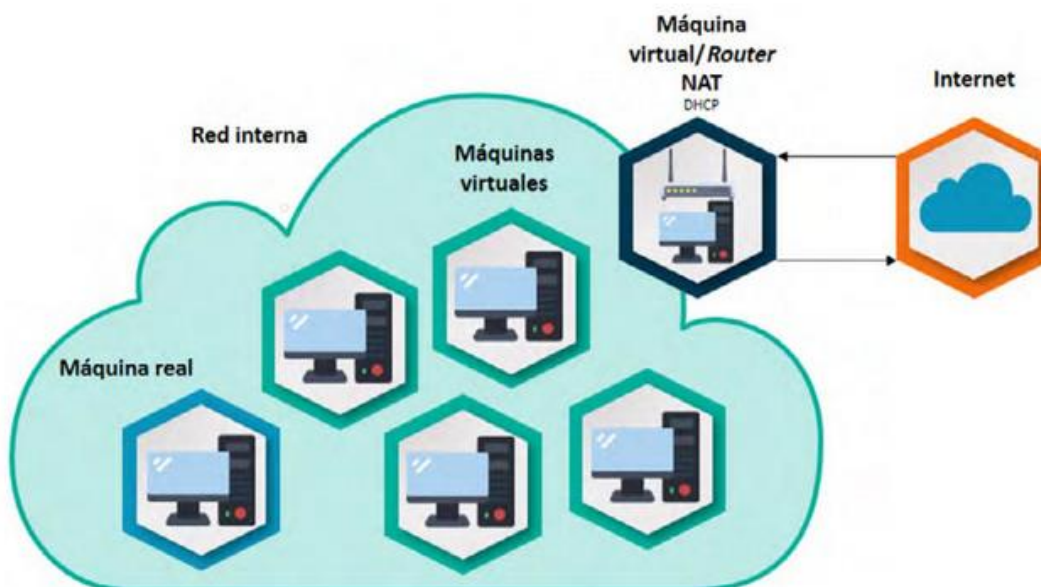
Las redes de ordenadores son un conjunto de equipos informáticos conectados entre sí que permiten compartir información y recursos, además de ofrecer determinados servicios. Para crear diferentes tipos de redes es necesario configurar la tarjeta de red del equipo. Los tipos de adaptador de red más comunes son:

- *NAT*: es el tipo por defecto y sirve para conectarse a Internet sin complicaciones, ya que no requiere ningún tipo de configuración de red. En este modo, la máquina virtual posee su propio *router*, que mapea el tráfico desde y hacia la máquina virtual de forma transparente, por lo que las máquinas con esta configuración no pueden “verse” entre sí.

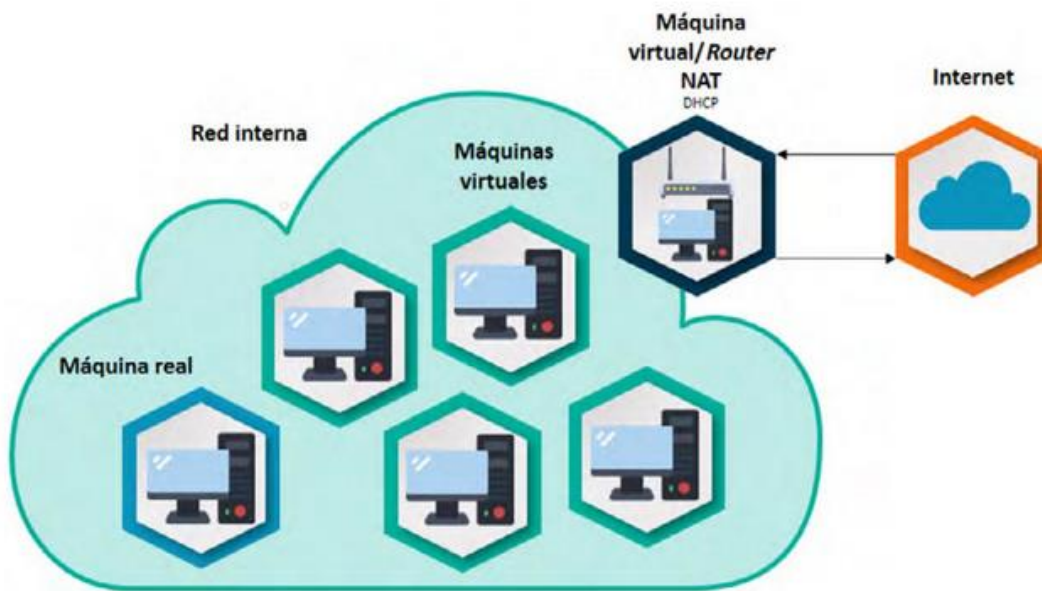
A pesar de que, *a priori*, con esta configuración sería imposible acceder a un servidor instalado en la máquina virtual, se puede conseguir mediante el reenvío de puertos a través del anfitrión.



- *Adaptador puente* (bridged): se simula que la máquina virtual está conectada al mismo dispositivo de red que la máquina física y, por lo tanto, ambas se encuentran en la misma red física.



- *Adaptador solo-anfitrión* (host-only): añade el equipo anfitrión, mediante una tarjeta de red virtual, a una *red interna* en la que solo existe comunicación entre las máquinas virtuales que pertenezcan a dicha red. Para que la red interna tenga salida al exterior, alguna de las máquinas debe hacer de *router*, por lo que debe tener una de sus tarjetas de red configurada como NAT o adaptador puente. Se añade a la red interna.

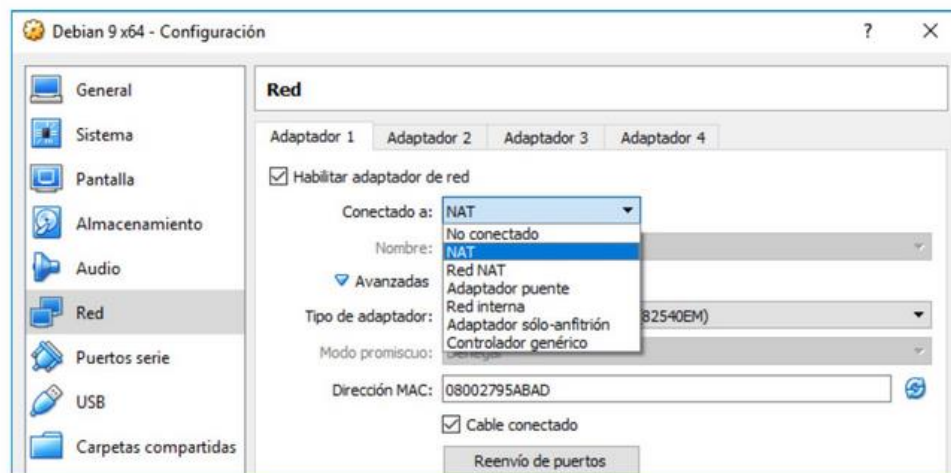


Además, se pueden crear varios adaptadores de red para una misma máquina virtual y modificar sus direcciones MAC.

En *VirtualBox*. En la opción 'Red' es posible configurar hasta cuatro tarjetas de red por máquina virtual. El modo por defecto es NAT, aunque existen otras posibilidades:

- *No conectado*. Como si el cable de red no estuviera conectado. Sirve, principalmente, para obligar al sistema operativo a reconfigurar la tarjeta de red. Otra forma de conseguir esto es desactivando la característica 'Cable conectado' de las opciones avanzadas.
- *Adaptador puente*.
- *Adaptador solo-anfitrión*. Para administrar las interfaces de red solo-anfitrión se accede mediante 'Archivo → Administración de red anfitrión'.
- *Red NAT*. Es similar al modo NAT, pero creando una red NAT, donde todas las máquinas que pertenecen a la misma red pueden comunicarse entre sí. Para configurar este modo, primero es necesario crear una red NAT desde 'Archivo → Preferencias → Red'. Una vez creada, solo es necesario unirse a dicha red.
- *Red interna*. Como el modo "Adaptador solo-anfitrión", pero sin la máquina real dentro de la red interna.
- *Controlador genérico*. En el caso de que la máquina virtual no reconozca el adaptador de la red física, se establece uno genérico.

Desde la opción 'Avanzadas', se pueden administrar ciertas características del modo seleccionado, como cambiar la dirección MAC de una tarjeta de red.



Configurar adaptador de red ('Máquina → Configuración → Red → Adaptador 1').

PARA SABER MÁS

La dirección MAC o dirección física es un número formado por seis bloques de dos dígitos hexadecimales (por ejemplo: 6C-FD-B0-28-0E-11), que identifica de forma única una tarjeta o dispositivo de red. En ocasiones puede ser necesario modificarla mediante una técnica conocida como *MAC spoofing*, para ocultar un dispositivo en una red o hacerse pasar por otro dispositivo.

PRÁCTICA 2 – Introducción a VirtualBox (en Moodle)

4. Instalación de SO

A la hora de instalar cualquier sistema operativo hay que tener en cuenta ciertos aspectos, los cuales cambian dependiendo del SSOO en cuestión:

- ♦ **REQUISITOS, VERSIONES Y LICENCIAS:** los requisitos de disco y RAM determinan que un sistema funcione correctamente o no (esta información suele venir en la web). La versión de un sistema (32bits o 64bits) también es importante ya determina sobre qué tipo de Hardware puede funcionar. Las licencias también determinan la forma de reinstalación y uso.
- ♦ **CONTROLADORES DE DISPOSITIVOS:** la instalación de los DRIVERS cambia mucho de un sistema operativo a otro, siendo la tendencia el instalarse de forma automática. Lo que sí es importante es asegurarse de que el sistema ha detectado todo el HW presente.
- ♦ **PROCEDIMIENTOS DE ARRANQUE DE SISTEMAS OPERATIVOS:** es MUY importante conocer el proceso de arranque ya que esto nos permite resolver problemas aparentemente graves de una forma muy sencilla.
- ♦ **INSTALACIONES DESATENDIDAS:** hay ciertos SSOO que permiten una instalación casi sin intervención, este tipo es interesante cuando necesitamos tener un SSOO instalado de forma rápida y con una configuración general.
- ♦ **DOCUMENTACIÓN DE LOS PROCESOS REALIZADOS:** en cualquier actividad es muy importante documentar todo aquello que se va haciendo y aquí no es menos. En principio puede parecer algo inservible pero cuando aparece algún problema, saber qué se instaló y de qué forma puede ayudar a resolverlo.

Vamos a ver los pasos de instalación de cada sistema operativo

4.1. INSTALACIÓN DE WINDOWS

La instalación de cualquier Windows ha sido similar desde los tiempos de Windows 95 siguiendo siempre los mismos pasos:

- 1) Arrancar la maquina con el CD/USB/PXE de Windows.
- 2) Aparece un menú indicando si queremos instalar/reparar
- 3) Si optamos por la instalación nos permite particionar de forma personalizada o directamente toma todo el disco duro para su uso.
- 4) Comienza la copia de archivos, la detección del Hw, la instalación de drivers y se reinicia en varias ocasiones.

Lo instalaremos en una máquina virtual para poder practicar.

CONCEPTOS BÁSICOS SOBRE MSDOS (USO EN REPARACIÓN)

MSDOS fue el primer SSOO creado por Microsoft, todos los SSOO creados hasta llegar al Windows actual se basan en él. Actualmente MSDOS es un emulador de WINDOWS que permite hacer CUALQUIER COSA que permita la interfaz gráfica (crear usuarios, particionar, gestionar servicios, gestionar el registro del sistema, etc..). Podemos acceder a una terminal para los comandos del sistema mediante el comando `cmd` en el buscador de la interfaz gráfica de Windows. PowerShell es una versión mejorada del emulador de MSDOS, pero orientada a objetos.

Tenemos que conocer algunos comandos básicos para poder realizar acciones relacionadas con la gestión de los SSOO Microsoft (consola de recuperación, gestión de cuentas de usuario, etc..)

COMANDO CD (CHANGE DIRECTORY)

Para utilizar este comando debemos saber algunos conceptos previos:

- ◆ **ruta absoluta:** una ruta absoluta siempre se da desde la raíz de la unidad a la que queremos acceder hasta el elemento en cuestión.
- ◆ **ruta relativa** se da desde el sitio desde donde estoy en este momento.
- ◆ El **direccionamiento absoluto o relativo** es muy importante ya que cualquier comando puede utilizarlos para usar ficheros o directorios.
- ◆ **Directorio .** : siempre se refiere al directorio en el cual estoy en este momento.
- ◆ **Directorio ..** : siempre se refiere al directorio padre del que estoy en este momento.
- ◆ Para meterme dentro de una carpeta siempre tengo que mirar donde estoy, para de esta forma cambiarme mediante ruta absoluta o relativa. Comando **echo %cd%**
- ◆ Si me encuentro en el nivel superior de un directorio, he de poner **cd nombredirec** para poder meterme en dicho directorio.
- ◆ Para subir por los niveles de carpetas: **cd ..** (estoy cambiando al directorio padre)
- ◆ Puedo enlazar los directorios mediante la barra: **/**. Por ejemplo, si quiero salirme de la carpeta padre y a su vez de su carpeta padre hago lo siguiente: **cd ../..**
- ◆ Puedo salirme directamente al directorio raíz mediante el comando **cd**

OTROS COMANDOS

Otros comandos básicos son:

- ◆ **HELP:** muestra ayuda de los comandos, si ponernos **HELP** comando nos da información ampliada sobre un comando concreto.
- ◆ **DIR:** lista el directorio actual
- ◆ **MKDIR/RMDIR** directorio: crea/borra un directorio
- ◆ **COPY** origen destino: copia fichero/s de un origen a un destino.
- ◆ **MOVE** origen destino: mueve fichero/s de un origen a un destino.
- ◆ **DEL** fichero/s: borra fichero/s
- ◆ **“UNIDAD:”** cambia a la unidad en cuestión. Por ejemplo, si el CD tiene asociada la letra D, para cambiar a esta unidad pondremos D:

1.

4.2. INSTALACIÓN DE LINUX

La instalación de Linux es muy parecida a la que hemos visto para otros sistemas operativos. Sin embargo, no existe una “instalación normal” de Linux. Cada distribución de Linux dispone de su instalador propio, siendo distintos los instaladores de SuSe y de Red Hat, por poner un ejemplo. Incluso varían los instaladores de una versión a otra de la misma distribución. Aunque los conceptos básicos de la instalación siempre serán los mismos, la forma de trabajar varía. En general, veremos que los pasos básicos de la instalación serán "más o menos" estos:

- 1) Arrancar la maquina con el CD/USB/PXE de Linux.
- 2) Detección del hardware básico para la instalación (CD, HD, teclado, pantalla, ratón, tarjeta de red, etc.)
- 3) Elegir el tipo de instalación (personalizada, manual, experta).

- 4) Crear una partición para instalar en ella Linux, y seleccionar dicha partición.
- 5) Montar la partición creada en un punto de montaje.
- 6) Crear una partición para utilizarla como swap.
- 7) Escoger los paquetes que queremos instalar.
- 8) Copiar archivos e instalar un gestor de arranque
- 9) Reiniciar la máquina para ejecutar ya nuestro Linux

Otro punto que hay que tener en cuenta es el caso de las **instalaciones múltiples** (esto ya lo sabemos). Si por cualquier razón, deseo tener un SSOO Microsoft y un Linux he de seguir cierto orden para su instalación: Linux ha de ser el último en instalarse ya que éste sí que tiene en cuenta instalaciones Microsoft, pero no al revés. Es decir, si instalo Windows sobre Linux, lo más normal es que no se pueda arrancar el sistema Linux o que incluso ni arranque Windows, esto es debido a que Microsoft no respeta el arranque GRUB que suelen instalar los Linux (ya que sobrescribe el MBR/GPT con algo que NO es un MBR/GPT, sino un salto a otro sitio, al gestor de arranque de Linux).

*Hay distribuciones de Linux que si no detectan nada en el disco duro se instalan sin preguntar nada, para evitar esto siempre podremos entrar y formatear el disco por ejemplo a un sistema de archivos NTFS. Al introducir el disco de Linux y detectar otro sistema de archivos (Linux es generalmente ext4) nos dará la posibilidad de personalizar la instalación.

PUNTOS DE MONTAJE

Estamos acostumbrados a la forma en que los sistemas operativos de Microsoft denominan a los medios de almacenamiento secundarios, asignando una letra a cada volumen, de modo que la disquetera es la A: la primera partición del primer disco duro es la C:, la siguiente la D:, etc. En Linux todo esto cambia, un punto de montaje es la carpeta del disco duro donde se va a situar ("montar") parte o la totalidad del sistema de ficheros. Es obligatorio montar el sistema de ficheros / en una partición para poder instalar linux. Hay algunas carpetas como /home o /usr que se pueden montar en otras particiones (porque son los más grandes y los que más pueden crecer).

En primer lugar, veamos como Linux referencia a los propios discos duros. Así, el primer disco duro de nuestro maquina en Linux se conoce como /dev/hda (si es paralelo) o /dev/sda (si es serie).

- ◆ / :indica el root o raíz del árbol de Linux (En Linux solo existe un árbol)
- ◆ dev :nos indica el directorio donde se almacenan todos los dispositivos (devices)
- ◆ /hda :nos indica que nos referimos al Hard Disk (hd paralelo) con la letra a (el primero).

Algunos ejemplos:

- ◆ /dev/hda – Dispositivo maestro en la IDE 1.
- ◆ /dev/hdb – Dispositivo esclavo en la IDE 1.
- ◆ /dev/hdc – Dispositivo maestro en la IDE 2.
- ◆ /dev/hdd – Dispositivo esclavo en la IDE 2.
- ◆ /dev/sda - Dispositivo serie en el bus 1.
- ◆ /dev/sdd – Dispositivo serie en el bus 4.

Cuando referenciamos particiones, se utiliza un número a continuación del nombre del disco duro. Este número representa la partición. Así, `/dev/hda2` nos indica que nos referimos a la segunda partición del disco duro maestro de IDE 1. Como en un disco duro no pueden existir más de cuatro particiones primarias (siempre en el caso de un disco MBR), estas reciben números del 1 al 4. Si creamos una partición extendida, esta no recibe ningún número (igual que en Windows no se le asigna una letra) y a las unidades lógicas que se crean dentro de dicha partición extendida se le asignan números a partir del 5. Veamos algunos ejemplos.

- ☐ `/dev/hdb1` - Primera partición primaria del disco duro 2 (esclavo en el 1º canal IDE).
- ☐ `/dev/hda5` - Primera unidad lógica del disco duro 1 (maestro en el 1º canal IDE).
- ☐ `/dev/sdc7` - Tercera unidad lógica del disco duro 3 (SATA en el bus 3).

En Linux, cada dispositivo de almacenamiento (partición, usb, CD) que utilicemos debe ser montado en un punto de montaje: hace que su espacio de almacenamiento se encuentre disponible accediendo al directorio especificado (conocido como punto de montaje). Dicho de otra forma, accedemos al dispositivo mediante una carpeta del sistema.

PARTICIÓN SWAP

En los sistemas operativos existe una técnica que nos permite ofrecer a los programas más memoria de la que existe físicamente en la máquina (MEMORIA VIRTUAL), se llama técnica de "paginación". En Windows esta técnica utiliza un archivo de intercambio que es gestionado normalmente por Windows, se suele llamar `pagefile.sys` y tiene un tamaño alrededor de 1.5 veces el tamaño de la RAM (si arrancamos **DISKMGMT.MSC**, para ver las particiones, una de ellas tiene que tener la etiqueta "Archivo de Paginación", en la raíz de esa partición se guarda el fichero de paginación para las máquinas virtuales).

Linux puede usar una utiliza una partición entera para este fin, conocida como partición Swap. Simplemente tenemos que crearla en la instalación de Linux, y el sistema se encarga de usarla, sin tener nosotros que montarla ni nada por el estilo. El tamaño que se le suele dar a una partición Swap, es el doble de la memoria RAM que tengamos instalado en nuestro sistema. Actualmente, sin embargo, es común en algunas distribuciones, como Ubuntu, el uso un archivo de intercambio Swap.

En el caso de que necesites averiguar si tu instalación de Ubuntu está usando una partición swap o un archivo basta con abrir una nueva ventana de terminal y escribir lo siguiente: **cat /proc/swaps**

5. Restauración y Actualización de SO

Restaurar sistema es una herramienta de Microsoft Windows diseñada para proteger y reparar el software del equipo. Restaurar sistema realiza una "instantánea" de algunos archivos del sistema y del registro de Windows y los guarda como Puntos de restauración. Cuando se produce un error de instalación o daños en los datos, la restauración del sistema puede devolver un sistema a una condición de trabajo sin tener que reinstalar el sistema operativo. Repara el entorno de Windows revirtiendo a los archivos y la configuración que se guardaron en el punto de restauración.

Nota: no afecta a los archivos de datos personales del equipo.

Los Sistemas Operativos requieren de actualizaciones periódicas, por varios motivos, ya sea por actualizaciones de HW, actualizaciones de los programas o nuevas funcionalidades. Cada SSOO tiene su propio método de actualización:

- Windows: se puede realizar desde Windows Update
- MAC: desde el programa incorporado llamado Software Update
- Linux: dependiendo de la distribución se hará de forma gráfica o mediante el Shell utilizando los comandos **apt upgrade** y **apt dist-upgrade**.