SISTEMAS INFORMÁTICOS UD-2

INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS Y LAS MÁQUINAS VIRTUALES

ÍNDICE

[CONCEPTOS GENERALES 3](#_Toc118482829)

[A. MÁQUINA VIRTUAL 3](#_Toc118482830)

[B. *HOST* O SISTEMA ANFITRIÓN 3](#_Toc118482831)

[C. *GUEST* O SISTEMA HUESPED 3](#_Toc118482832)

[FUNCIONAMIENTO DE UNA MÁQUINA VIRTUAL (MV) 3](#_Toc118482833)

[VENTAJAS DEL USO DE MÁQUINAS VIRTUALES 5](#_Toc118482834)

[REQUISITOS REALES 6](#_Toc118482835)

[A. MEMORIA (RAM) 6](#_Toc118482836)

[B. ALMACENAMIENTO (HDD) 6](#_Toc118482837)

[VIRTUALBOX 6](#_Toc118482838)

[TERMINOLOGÍA BÁSICA 6](#_Toc118482839)

[MODOS DE RED 7](#_Toc118482840)

[A. NO CONECTADO 7](#_Toc118482841)

[B. NAT 7](#_Toc118482842)

[C. RED NAT 8](#_Toc118482843)

[D. ADAPTADOR PUENTE 8](#_Toc118482844)

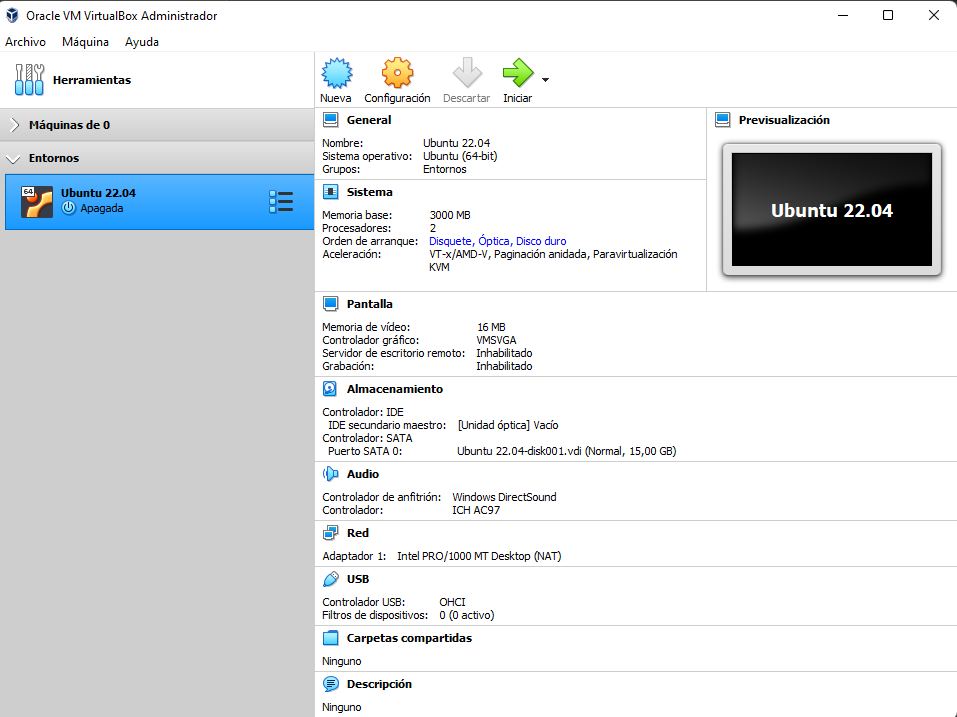
[E. RED INTERNA 8](#_Toc118482845)

[F. SOLO ANFITRIÓN (HOST ONLY) 9](#_Toc118482846)

[TIPOS DE DISCOS 9](#_Toc118482847)

# CONCEPTOS GENERALES

## MÁQUINA VIRTUAL

Es un software que se va a instalar en un ordenador (por ejemplo, VirtualBox o VMWare) que va a permitir un entorno virtual que emula el hardware de un ordenador.

Se pueden crear tantos ordenadores virtuales como se necesiten. Estos ordenadores virtuales usan los recursos que tiene el ordenador anfitrión.

Cada sistema virtual ejecutará su propio sistema operativo y las aplicaciones que instalemos.

## *HOST* O SISTEMA ANFITRIÓN

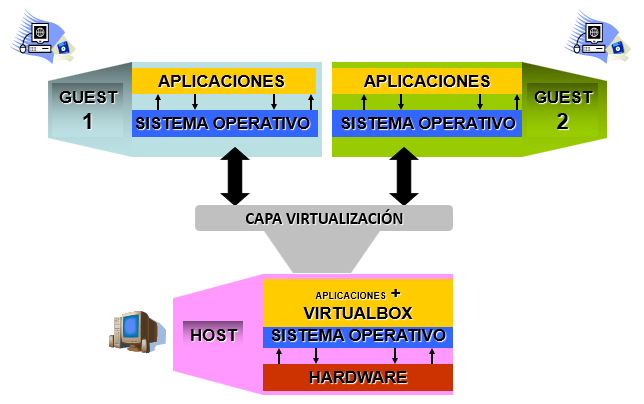
Es el ordenador físico donde se instala la aplicación de virtualización. Por tanto, sólo habrá un *host*.

## *GUEST* O SISTEMA HUESPED

Será cada ordenador virtual que se cree mediante la aplicación. Por tanto, tendremos tantos *guests* como se quieran tener (siempre que los recursos de memoria de almacenamiento lo permitan).

# FUNCIONAMIENTO DE UNA MÁQUINA VIRTUAL (MV)

Cada *guest* comparte los recursos hardware con el host:

* RAM.
* HDD.
* Microprocesador.
* Tarjeta gráfica.
* Tarjeta de red.
* Otros dispositivos: DVD, CD-ROM, disquetera…

El funcionamiento es muy simple:

1. El SO del *host* ejecuta la aplicación de virtualización y utiliza los recursos de este.
2. La aplicación genera una capa de virtualización a partir de la cual se creará una conexión *host-guest.*
3. En cada uno de los *guest* tendremos un SO ejecutándose y unas aplicaciones ejecutándose sobre ese SO.

Por tanto, vamos a tener varios SOs *guest* ejecutándose sobre el SO *host.*

Tenemos dos modelos de uso de máquinas virtuales, el **modelo de hipervisor alojado** y el **modelo de hipervisor nativo**.

Normalmente se usa el modelo de hipervisor alojado. Es el descrito más arriba en el funcionamiento. Es decir, vamos a tener un *host* con un hardware real, un SO y un software de virtualización. Y una serie de *guests* que van a tener un hardware virtual, un SO y unas aplicaciones moviéndose sobre ese SO.

Es el más sencillo de administrar tanto para un administrador como para el equipo.

Ejemplos de software virtualización serían: Oracle VirtualBox, VMWare Workstation, Parallel y QEMU.

El modelo hipervisor nativo no tiene un SO como tal en lo que sería el *host* sino un hipervisor que nos permitirá la creación de MVs. Se usa para ordenadores que utilizan servidores clientes.

Ejemplos de software de virtualización serían: VMWare ESXi, KVM, Citrix XenServer, Microsoft Hyper-V y Oracle VM Server para x86.

# VENTAJAS DEL USO DE MÁQUINAS VIRTUALES

* Una MV no depende del hardware ni del SO del *host*.
* Mejor aprovechamiento del *host*. El hardware de los equipos actuales es muy potente.
* Reducción de coste energético. Se necesitan muchas menos máquinas físicas para obtener el mismo resultado, lo cual se traduce en un menor coste energético y de refrigeración.
* Recuperación ante desastres: alta disponibilidad. Se puede levantar una máquina similar a la estropeada en muy poco tiempo.
* Menos conflictos con aplicaciones y servicios. Aunque el hardware del *host* sea distinto, se pueden implementar aplicaciones y servicios en equipos virtuales idénticos.

Una máquina virtual se puede encender (*power on*), apagar (*power off*) y suspender (*suspends*).

Cuando se suspende se guarda un fichero con el contenido que tiene la memoria RAM en ese momento y apaga el ordenador.

Cuando se vuelva a iniciar, recupera del fichero el contenido de la memoria y continuamos trabajando en el mismo estado en el que nos encontrábamos.

Se puede almacenar el estado de la MV en un momento dado para poder volver a él más adelante (copia de seguridad), esto se puede hacer:

* Mediante **snapshot**. Guarda el estado actual. El problema es que ocupan demasiado espacio, pero se pierde sólo el tiempo entre inicio e instantánea.
* Mediante **revert**. Permite eliminar todos los cambios. Es la vuelta a un estado anterior.

# REQUISITOS REALES

## MEMORIA (RAM)

Cada MV que instalemos consume memoria del *host.* La RAM mínima que será necesaria será la suma del mínimo de memoria recomendada para cada SO (incluido el del *host*).

Evidentemente los requisitos mínimos de memoria varían en función de las aplicaciones que vamos a instalar.

## ALMACENAMIENTO (HDD)

Cada MV que instalemos consume HDD del *host*. Esto es así porque cada MV crea un fichero que utilizará como HDD. Ese fichero lo definimos con un tamaño máximo y nos permite configurarlo para que vaya creciendo según vamos necesitando espacio.

Al igual que un ordenador físico, el espacio ocupado serán igual al espacio necesario para instalar el SO, las apliaciones y nuestros archivos.

# VIRTUALBOX

Es un software de virtualización para arquitecturas x86/amd64. Actualmente es desarrollado por *Oracle Corporation* como parte de su familia de productos de virtualización. Por medio de esta aplicación es posible instalar sistemas operativos adicionales, conocidos como «sistemas invitados», dentro de otro sistema operativo «anfitrión», cada uno con su propio ambiente virtual.

Los SOs soportados por VirtualBox son:

* **Anfitrión**: GNU/Linux, MacOS X, OS/2 Warp, Microsoft Windows y Solaris / OpenSolaris.
* **Huésped**: FreeBSD, GNU/Linux, OpenBSD, OS/2 Warp, Windows, Solaris, MS-DOS y muchos otros.

# TERMINOLOGÍA BÁSICA

* **Sistema Operativo Anfitrión (*host OS*)**, el sistema operativo del ordenador físico sobre el que se ha instalado VirtualBox.
* **Sistema Operativo Invitado (*guest OS*)**, el sistema operativo que se instala y ejecuta en la máquina virtual.
* **Máquina Virtual (*VM*)**, entorno especial que VirtualBox crea para el sistema operativo invitado mientras se ejecuta.
* ***Guest Additions***, software especial que se instala en la máquina virtual para mejorar el rendimiento del sistema operativo invitado y añadir características extra.

# MODOS DE RED

Cada tarjeta de red puede configurarse por separado para operar en uno de los siguientes modos:

## NO CONECTADO

En este modo la tarjeta de red está presente pero no se encuentra conectada. Es como si el cable de red no estuviera conectado a la tarjeta de red.

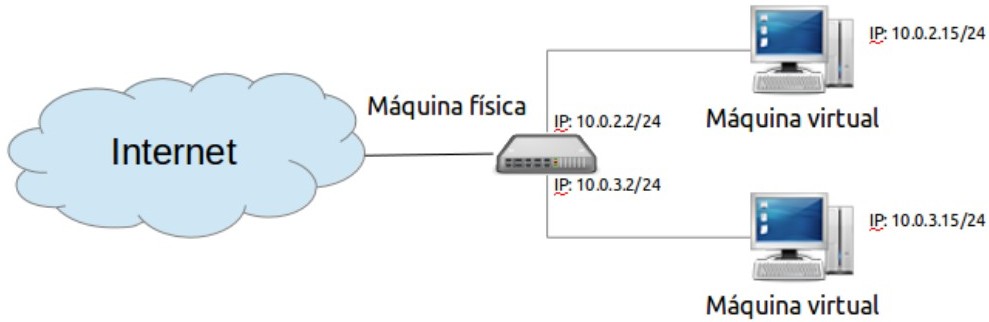
## NAT

Para entender como funcionan los tipos de red, es necesario entender cómo funcionan las IPs. Por ejemplo, teniendo en cuenta la IP 10.0.2.15 / 24 hay que tener en cuenta que:

* + Todos los números anteriores a la barra son de 8 bits, es decir, pueden tener valores de 0 a 255.
  + El número posterior a la barra indica el número de dígitos binarios que identifican la red, en este caso 10.0.2 (8 bits x 3 = 24 bits). Esos valores identifican de forma única a una red.
  + El último valor anterior a la barra identifica a la máquina dentro de la red.

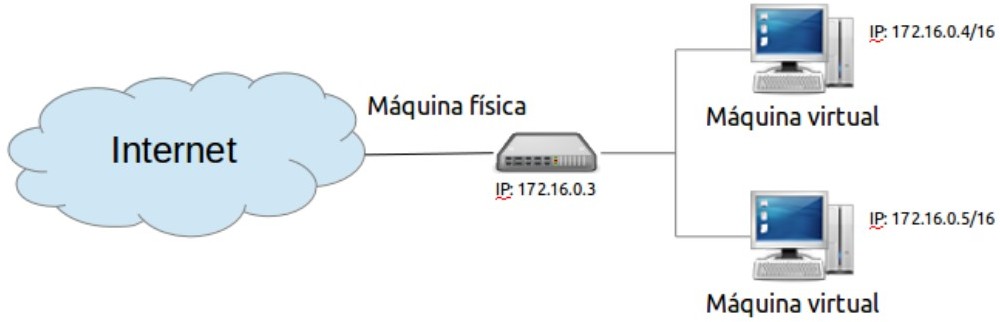
En el caso del modo de red NAT, el *host* va a actuar como un dispositivo NAT que permita a la MV salir a Internet.

Las diferentes MVs van a estar aisladas del anfitrión (tendrán otra red) y no podrán comunicarse con ninguna otra MV.

Por tanto, el *host* se comporta como un router con N interfaces, una para cada máquina virtual.

## RED NAT

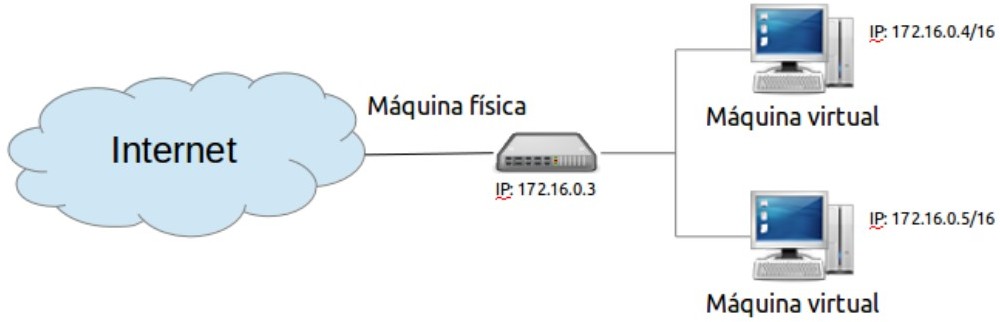
En la red NAT, el *host* vuelve a funcionar como un dispositivo NAT pero la MV no está aislada de la red del anfitrión y, además, puede comunicarse con otras MVs.

En este caso, el *host* actúa como un router doméstico dando salida a internet a todos los ordenadores de la red que a su vez se pueden comunicar entre sí.

## ADAPTADOR PUENTE

El *host* actúa como un puente y la MV pertenece a la red física a la que está conectada el anfitrión.

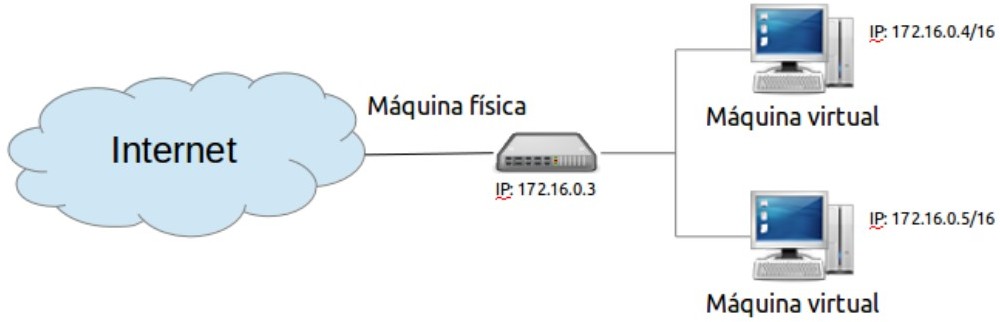
Las MVs pueden comunicarse entre sí y con todos los PCs de la red física. A diferencia de la Red NAT, todas las MVs están conectadas en la misma red física.



## RED INTERNA

En este modo de red lo que se genera es una red de MVs interna de modo que todas las MVs de este modo que tengan el mismo nombre de red virtual formarán una red virtual que les permitirá comunicarse entre sí.

## SOLO ANFITRIÓN (*HOST ONLY*)

Las MVs se comunican con el anfitrión y entre sí. No pueden salir a Internet salvo que la conexión de red de la máquina física que conecta a Internet se comparta.

# TIPOS DE DISCOS

Lo normal es que los discos del *guest* sean ficheros del *host*. En VirtualBox tenemos:

* **VDI**. Formato nativo de VirtualBox.
* **VMDK**. Formato abierto propio de VMWare.
* **VHD**. Formato creado por Microsoft para su abandonado programa de virtualización Virtual PC.
* **RAW**. Es un fichero que contiene byte a byte el contenido del disco virtual.