



## **TEMA 132**

**VIRTUALIZACIÓN DE SISTEMAS Y DE CENTROS DE DATOS.  
VIRTUALIZACIÓN DE PUESTOS DE TRABAJO. MAQUETAS  
DE TERMINALES WINDOWS Y DE SERVIDORES LINUX.**

<b>Versión</b>	<b>30.1</b>
<b>Fecha de actualización</b>	<b>18/09/2024</b>



## ÍNDICE

ÍNDICE .....	2
1. TEMAS RELACIONADOS .....	3
2. INTRODUCCIÓN .....	3
2.1 CONCEPTOS.....	3
2.2 VIRTUALIZACIÓN DE RECURSOS .....	3
2.3 VENTAJAS DE LA VIRTUALIZACIÓN .....	4
3. VIRTUALIZACIÓN DE SERVIDORES .....	4
3.1 HIPERVISOR .....	4
3.2 TIPOS DE VIRTUALIZACIÓN DE SERVIDORES.....	5
4. VIRTUALIZACIÓN DE ALMACENAMIENTO .....	8
5. VIRTUALIZACIÓN DE ESCRITORIOS Y APLICACIONES.....	8
5.1 COMPONENTES.....	8
5.2 VENTAJAS E INCONVENIENTES .....	8
6. VIRTUALIZACIÓN DE RED .....	9



# 1. Temas relacionados

---

El concepto de Cloud Computing está muy relacionado con la virtualización (ver tema específico Cloud – Tema 55). La mayor parte de la infraestructura utilizada actualmente por los servicios basados en cloud computing se apoya en servicios de alta fiabilidad suministrados a través de centros de datos de nueva generación, contruidos con tecnologías de virtualización informática y de almacenamiento.

Así mismo, este tema también está relacionado con el Tema 96 sobre herramientas de integración continua, donde se desarrollan las herramientas DevOps, que son utilizadas tanto para despliegues en nube como para automatizar la virtualización en CPDs, como por ejemplo Terraform o Ansible.

## 2. Introducción

---

La virtualización es una capa que permite a un conjunto de recursos físicos (**procesadores, memoria, redes, almacenamiento**) funcionar como si fueran múltiples recursos lógicos. Esto se puede aplicar tanto a **servidores, almacenamiento y redes como a escritorios o puestos de trabajo**.

### 2.1 Conceptos

---

- **Anfitrión (host)**, que gestiona los accesos a los recursos físicos por parte de los recursos virtualizados.
- **Huésped o invitado (guest)**, recurso virtualizado.
- **Hipervisor o Virtual Machine Manager (VMM)**. Es el software encargado de realizar la abstracción de los recursos físicos y de gestionar el acceso a los mismos.

### 2.2 Virtualización de recursos

---

- **Procesador**. Es necesario coordinar el acceso al procesador por parte de las distintas máquinas virtuales (teniendo en cuenta que los sistemas x86 no fueron diseñados para soportar múltiples sistemas alojados).
- **Memoria**. Debe ser manejada para que múltiples sistemas alojados la compartan sin colisionar.
- **Red:**
  - Se reparte el ancho de banda disponible en los distintos canales y se aprovecha la infraestructura.
  - **VLAN**. Se puede asignar los equipos conectados a una o N redes virtuales y gestionar los recursos de almacenamiento e impresión.
- **Almacenamiento**. Muestra el almacenamiento como una unidad única (Tema 56):
- **Aplicaciones**. Consiste en ejecutar una aplicación en un entorno virtual que le proporciona todos los componentes que necesita, usando los recursos locales como si estuvieran utilizando directamente los recursos del sistema. El entorno actúa como una capa entre la aplicación y el sistema operativo, eliminando los conflictos entre las aplicaciones y con el sistema operativo.
  - Gran portabilidad porque la app virtualizada puede ser transportada a otros sistemas sin necesidad de instalación.
  - Un ejemplo es la Máquina Virtual de Java (JVM – Java Virtual Machine). La máquina virtual se ejecuta como un proceso en el sistema operativo y soporta sólo un proceso. Cuando se lanza el proceso se ejecuta la máquina virtual automáticamente y cuando el proceso finaliza se para también automáticamente. De esta forma, se le proporciona al proceso un entorno de ejecución independiente de la plataforma hardware y del sistema operativo, permitiéndole que se ejecute sobre cualquier plataforma.



## 2.3 Ventajas de la virtualización

---

- **Maximización o mejor aprovechamiento del hardware.** La creciente capacidad y potencia de los servidores hace más difícil el total aprovechamiento de sus recursos si no es mediante la virtualización. De esta forma se **reduce el consumo de energía y la huella de carbono. Ahorro de costes.**
- **Mayor facilidad de adaptación a la demanda.** Desplegar una nueva máquina virtual es menos costoso que una física.
- **Mejor administración de la infraestructura.**
- **Facilidad de replicar entornos.** Despliegue rápido de entornos de prueba, preproducción y producción. Aislamiento ideal para pruebas.
- **Mayor flexibilidad en la configuración, tanto de recursos como de software.** Se pueden probar nuevos sistemas operativos, actualizaciones, o migraciones de forma independiente a la máquina física. Se puede configurar el entorno virtual con distintas configuraciones (procesadores, memoria, etc.).
- **Mayor seguridad.** Aislamiento entre máquinas virtuales.
- **Mejora la continuidad del negocio y la alta disponibilidad.** Pudiendo mover máquinas virtuales entre servidores físicos sin interrupción del servicio. Encapsulación de máquinas virtuales en ficheros que se pueden copiar e implantar en cualquier servidor físico.
- **Facilidad de realizar backups de la máquina virtual. Snapshots.** Un snapshot es una imagen concreta del sistema en el momento en el que se toma, de forma que se pueda restaurar al momento del snapshot en caso de alguna contingencia.

## 3. Virtualización de servidores

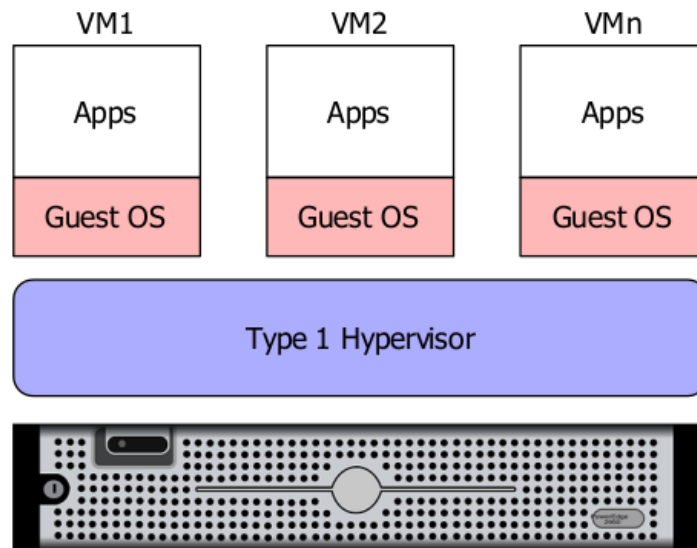
---

### 3.1 Hipervisor

---

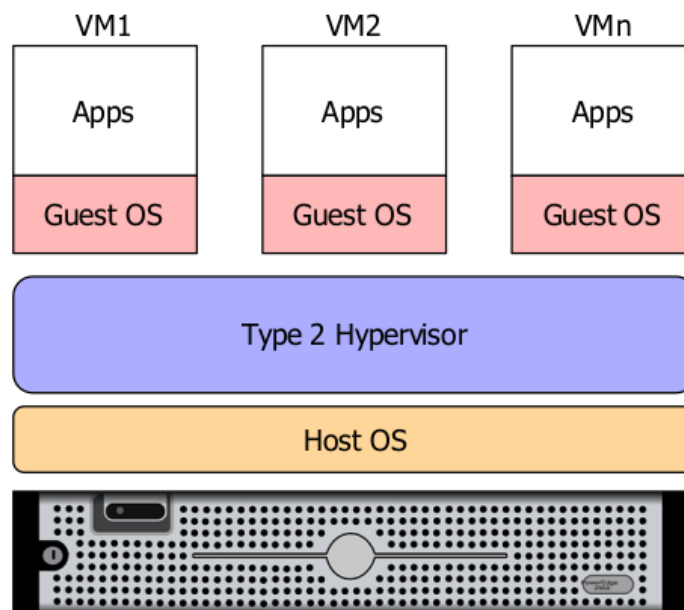
Es como un microkernel que se encarga de monitorizar las máquinas virtuales que ejecuta. Se pueden distinguir dos tipos de hipervisor:

- **Tipo 1, nativo, bare metal o unhosted**
  - Se ejecuta directamente sobre el hardware y permite gestionar los accesos de los sistemas operativos invitados a los recursos hardware.
  - No se requiere contar con un sistema operativo anfitrión, sino que el propio hipervisor será el sistema operativo.
  - Ejemplos: VMware ESXi (de pago), Xen, Microsoft Hyper-V, KVM (KVM convierte al kernel de Linux en un hipervisor de tipo 1, mientras el sistema completo podría ser categorizado de tipo 2 porque el sistema operativo anfitrión sigue siendo completamente funcional y las máquinas virtuales son procesos Linux estándar desde su perspectiva), Proxmox VE (plataforma open source que integra el hipervisor KVM).



- **Tipo 2, alojado o hosted.**

- Se ejecuta en el entorno convencional de un sistema operativo anfitrión instalado en la máquina física.
- Hipervisor realiza emulación hardware para gestionar operaciones privilegiadas.
- Ejemplos: Oracle VM VirtualBox, VMware Workstation, QEMU, Windows Virtual PC.



## 3.2 Tipos de virtualización de servidores

### 3.2.1 Emulación

Software que permite ejecutar programas en una plataforma diferente de aquella para la que fueron escritos originalmente. A diferencia de un simulador, que solo trata de reproducir el comportamiento del programa, un



emulador trata de modelar el dispositivo de manera que funcione como si estuviese siendo usado en la plataforma original. Ejemplos: Qemu, DOSBox, Bochs, MAME, Hercules, MESS.

### 3.2.2 Virtualización completa o de Hardware

---

- Permite al **sistema operativo invitado** ser ejecutado **sin modificaciones**.
- Permite ejecutar múltiples sistemas operativos en distintas máquinas virtuales, siempre que dichos sistemas operativos sean compatibles con la arquitectura hardware subyacente.
- El hipervisor es el que administra los recursos, traduciendo las peticiones de los sistemas invitados a solicitudes al sistema anfitrión, por lo que debe contar con los drivers necesarios para el acceso a los recursos físicos.
- Aunque se logra un rendimiento muy elevado, este mecanismo de traducción genera una sobrecarga considerable, penalizando el funcionamiento global.
- Según la forma de virtualizar el hardware, se puede destacar, por un lado, la virtualización de los recursos hardware por separado, creando un driver específico para cada recurso, y, por otro lado, la creación de un microkernel sobre el que funciona igualmente tanto el sistema operativo como las máquinas virtuales. Esta segunda opción es la implementada por Microsoft Hyper-V y es la tendencia tecnológica en cuanto a rendimiento.
- Ejemplos: VMware Workstation, VMware ESXi, Microsoft Hyper-V, Oracle VM VirtualBox, KVM (Kernel-based Virtual Machine), Windows Virtual PC, etc.

### 3.2.3 Virtualización parcial o paravirtualización

---

- La máquina virtual no necesariamente simula un hardware, sino que ofrece un API especial que solo puede utilizarse en un **sistema operativo invitado modificado**, para que hagan llamadas al hipervisor.
- Las llamadas del sistema operativo invitado al hipervisor se denominan hypercalls.
- Ralentiza la ejecución de las máquinas virtuales para que el hipervisor:
  - Recompile continuamente el sistema operativo invitado para obligar a todas las instrucciones a ejecutarse fuera del anillo 0.
  - Emule de ring 0 completo.
- Solución: virtualización asistida por hardware (Intel VT-x, AMD-V). El rendimiento de las máquinas virtuales es casi idéntico al de sistemas operativos reales, ya que no se requiere recompilar nada ni emular ningún anillo, permitiendo ejecutar los sistemas operativos directamente en el ring 0.
- No todos los sistemas operativos soportan paravirtualización.
- Ejemplos: Xen, L4, XtratuM.
  - Xen utiliza una técnica llamada "ring depriving", donde el sistema operativo es modificado para poder ejecutarse en ring 1 dejando el ring 0 para el Hypervisor.

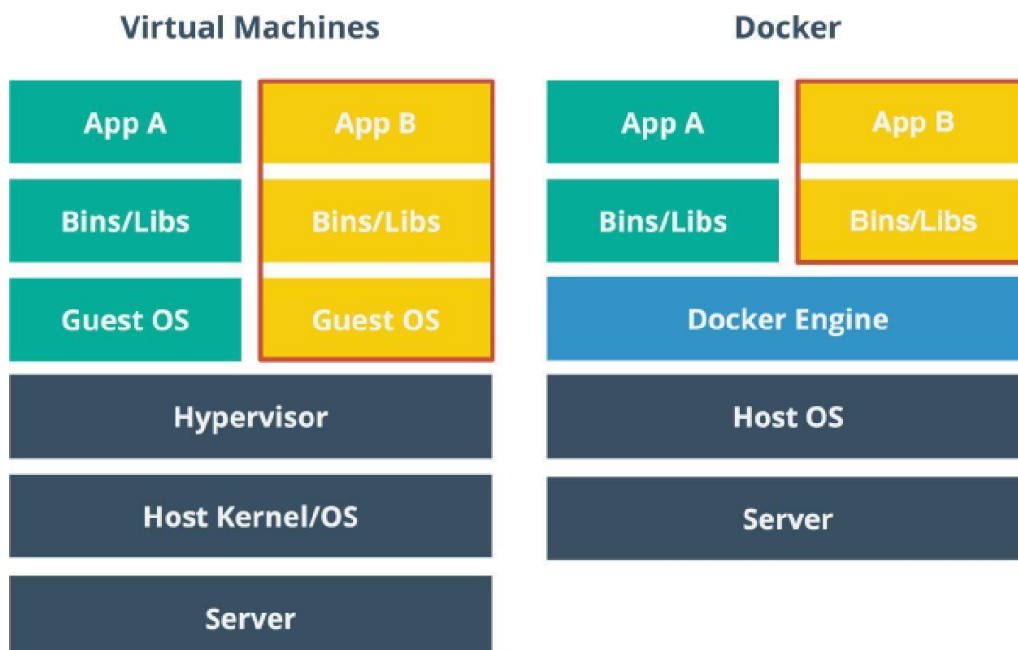
### 3.2.4 Virtualización a nivel de sistema operativo

---

- El sistema operativo anfitrión virtualiza el hardware a nivel de sistema operativo, permitiendo que varios sistemas operativos virtuales se ejecuten de forma aislada en un mismo servidor físico.
- Aunque cada MV puede seguir funcionando de forma independiente, todas deben utilizar el mismo sistema operativo que el servidor anfitrión.
- Se crean contenedores o entornos virtuales (EV) aislados en un único servidor físico y una única instancia de sistema operativo (sin necesidad de contar con un hipervisor).
- Como debilidad de este tipo de virtualización destaca que el SO anfitrión supone un único punto de fallo.
- Ejemplos: Virtuozzo Containers, FreeBSD Jails, Oracle Solaris Containers, OpenVZ, LXC (Linux Containers), Proxmox VE (plataforma open source que integra contenedores LXC), etc.



- Desde 2014, el uso de contenedores vuelve a estar en auge. Prueba de ello, es el proyecto de código abierto Docker, que permite la fácil creación de contenedores para la ejecución de aplicaciones.
  - La idea central es crear un contenedor (simplemente es un proceso para el sistema operativo) con aquellas librerías, archivos y configuraciones necesarias para la aplicación concreta, sin tener que instalar un sistema operativo completo de forma subyacente.
  - Se diferencia de una máquina virtual en que ésta necesita contener todo el sistema operativo, mientras que un contenedor aprovecha el sistema operativo sobre el que se ejecuta.
  - Las ventajas del uso de contenedores son las siguientes:
    - **Portabilidad:** un contenedor Docker se puede desplegar en cualquier otro sistema que soporte esta tecnología (que tenga instalado Docker).
    - **Flexibilidad:** el arranque de un contenedor supone el mismo tiempo que arrancar un proceso en el sistema, mientras que arrancar una máquina virtual implica más tiempo.
    - **Ligereza:** el contenedor, al contener únicamente las dependencias necesarias para la ejecución de la aplicación, ocupa mucho menos espacio, por lo que el tamaño de una imagen o plantilla de un contenedor es muy reducido, en comparación con el de una máquina virtual
  - Para la gestión de los contenedores y su despliegue existen herramientas muy potentes como Docker Compose y Kubernetes. También existen otras herramientas como: Docker Swarm, Moby, Red Hat OpenShift (plataforma de aplicaciones en contenedores open source basada en el orquestador Kubernetes).
  - Para poder crear una imagen Docker es necesario disponer de un fichero llamado Dockerfile que incluye los contenidos de dicha imagen. Dicha imagen se construirá capa a capa y llevará a cabo cada operación especificada sobre la capa anterior.
  - Para poder ejecutar varios contenedores de manera coordinada, se recomienda utilizar Docker Compose para lo que es necesario disponer de un fichero docker-compose.yaml, que parametriza el lanzamiento de los contenedores e indica cómo interactúan los diferentes contenedores entre sí. Por ejemplo, una aplicación ejecutada en el contenedor A se conecta a una Base de Datos en el contenedor B.





## 4. Virtualización de almacenamiento

---

La virtualización de las redes de almacenamiento permite:

- Aplicar un nivel adicional de seguridad mediante autorizaciones de acceso.
- Cambiar y hacer crecer el tamaño de los volúmenes sobre la marcha para responder a demandas de servidores o aplicaciones que lo requieren.

Existen dos tipos principales de virtualización de almacenamiento.

- El acceso a nivel de bloque utilizado en las redes SAN (Storage Area Network).
- El acceso a nivel de fichero, utilizado en una NAS (Network Attached Storage).

## 5. Virtualización de escritorios y aplicaciones

---

Infraestructura de escritorio virtual (Virtual Desktops Infrastructure, VDI). Consiste básicamente en implementar el escritorio como servicio. Todas las aplicaciones, procesos y datos del usuario se almacenan de forma centralizada en el servidor y el usuario podrá utilizar para ejecutar dicho escritorio una plataforma hardware y software totalmente distinta de la ejecutada en el servidor central, pudiendo acceder al escritorio desde cualquier dispositivo y cualquier lugar, siempre que cuente con el cliente adecuado para conectar con el servidor.

El escritorio se puede ejecutar remotamente en el servidor, o en local sincronizándose con el servidor. Este segundo método permite ejecución offline pero requiere mayor capacidad del equipo local.

Ejemplos de software comercial: VMware Horizon, XEN Desktop, etc.

### 5.1 Componentes

---

Los componentes principales son los siguientes:

- **Servidor de escritorios y aplicaciones.** Se ocupará de la ejecución de los distintos escritorios y/o aplicaciones de los usuarios, garantizando seguridad y aislamiento para cada uno de ellos.
- **Equipo de acceso del usuario.** Se ocupará de conectar con el servidor de escritorios y de comunicar cualquier cambio de pantalla que realice el usuario, a través del teclado y/o ratón.
- **Red de comunicaciones.** Para comunicar los equipos de acceso del usuario y los servidores de escritorio.
- **Protocolos de comunicación.** Principales: RDP, X Windows, ICA, NX o VNC.

### 5.2 Ventajas e inconvenientes

---

Ventajas:

- Reducción de costes de administración, al tener los clientes centralizados en el servidor.
- Mayor seguridad.
- Reducción de costes en los equipos de los usuarios.
- Mayor facilidad de despliegue de aplicaciones.
- Facilidad de creación y configuración de escritorios nuevos.
- Menor sobrecarga de red (sólo los cambios de pantalla).
- Posibilidad de conexión desde cualquier ubicación. Teletrabajo.

Desventajas:

- Dependencia de la infraestructura y comunicaciones que dan servicio a los escritorios remotos.
- Rendimiento medio-bajo de aplicaciones con alto uso multimedia





## 6.Virtualización de red

Desvinculación de los recursos de red del hardware subyacente, de manera que todos los elementos de la red pasan a controlarse de forma centralizada y pueden entenderse unos con otros independientemente del fabricante. Para transformar redes estándar en redes virtuales se hace uso de las tecnologías “Network Function Virtualization” (NFV) y “Software Defined Network” (SDN).

