

## U.T. 2.-SISTEMAS OPERATIVOS Y USO DE MÁQUINAS VIRTUALES

Objetivos:

- Conocer la estructura de un sistema operativo.
- Distinguir la arquitectura de un sistema operativo.
- Conocer las funciones de un SO.
- Ver los distintos tipos de SSOO.
- Distinguir los tipos de aplicaciones y los tipos de licencia que se pueden utilizar.
- Conocer qué es la virtualización y sus elementos básicos.
- Conocer las herramientas más habituales de virtualización.
- Construir máquinas virtuales.
- Utilizar máquinas virtuales.

### **2.1.- EL SISTEMA OPERATIVO**

El SO es un **grupo de programas que gestiona y optimiza los recursos del ordenador y coordina la ejecución de los programas**.

También mantiene las relaciones adecuadas entre la CPU, las memorias, los dispositivos, los periféricos, y entre estos y los programas de aplicación.

Además, se encarga de proporcionar al usuario un interfaz amigable para gestionar el sistema informático.

El SO es en si mismo un programa especial. Cuando se conecta un ordenador se carga parte del SO en la memoria y se ejecuta.

Inmediatamente que el microprocesador recibe corriente, envía una orden al chip de la memoria ROM del BIOS (*Basic Input/Output System*), donde se encuentran grabadas las rutinas del POST (*Power-On Self-Test*) A continuación el SO toma el control del ordenador.

Es muy importante también la función en la que el SO proporciona una interfaz al usuario, haciendo transparente para él el uso de HW. Es decir, un usuario simplemente abre los ficheros que grabó en un disco, sin preocuparse de como están almacenados y gestionados los bits en el medio físico, el tiempo de espera o la posición del cabezal, etc.



## **2.2.- ESTRUCTURA DE UN SISTEMA OPERATIVO**

### **2.2.1.- ESTRUCTURA MONOLÍTICA.**

Es la estructura de los primeros SO constituidos fundamentalmente por un único programa compuesto de un conjunto de rutinas entrelazadas que pueden llamarse entre sí cada vez que sea necesario. Las características fundamentales de este tipo de SSOO son:

- Tienen una estructura interna indefinida y sus niveles de funcionalidad no están bien separados.
- Tienen una difícil configuración y actualización, así como falta de protecciones y privilegios ya que se puede acceder a rutinas que manejan diferentes aspectos de la computadora.
- Disponen de una buena definición de los parámetros de enlace entre las distintas rutinas existentes,
- Suelen ser eficientes y rápidos, pero carecen de flexibilidad para soportar distintos ambientes de trabajo.

### **2.2.2.- ESTRUCTURA POR CAPAS.**

A medida que fueron creciendo las necesidades de los usuarios y se perfeccionaron los sistemas, se hizo necesaria una mayor organización del software, donde una parte del sistema contenía subpartes y estaba organizado en forma de niveles.

Se dividió el sistema en módulos, de manera que cada uno de ellos estuviera perfectamente definido y con una clara interfaz con el resto del sistema.

Actualmente la gran mayoría de los SSOO usan una estructura en cinco niveles o capas. Los dos primeros serían **dependientes de HW**, el resto pertenecen a la parte del SO **independiente del HW**.

Cada uno de los niveles se comunica con el inmediatamente inferior y superior, coordinando sus funciones.

**Nivel 1: Gestión del procesador:** En este nivel se encuentra el Kernel o núcleo que es la parte del SO encargada de gestionar la CPU. Cuando arranca el ordenador, se carga en memoria y permanece allí realizando sus funciones básicas.

En este nivel se lleva la cuenta de los procesos activos, trasladando el control a la CPU de un proceso a otro y almacenando el estado del sistema (contexto) en estructuras de datos.

El planificador que trabaja en este nivel es el responsable de asignar la CPU a cada uno de los procesos. La comunicación entre procesos se realiza a base de semáforos o señales.

También en este nivel se realiza el control de interrupciones y manejo de condiciones de error.

**Nivel 2: Gestión de memoria:** En este nivel se reparte la memoria disponible entre los procesos. Se realizan funciones de asignación, liberación de memoria y control de la violación de acceso a zonas de memoria no permitidas.

**Nivel 3: Gestión de procesos:** En este nivel se crean y destruyen los procesos y los intercambios de mensajes.

**Nivel 4: Gestión de dispositivos:** Aquí se hace la gestión de las entradas/salidas (E/S) en función de los dispositivos existentes. Se encarga de las funciones de creación de procesos de E/S, asignación a los procesos de dispositivos de E/S y planificación de la E/S.

**Nivel 5: Gestión de la información:** El objetivo de este nivel es el de gestionar el espacio de nombres lógicos, utilizados para simplificar el acceso a los recursos. Otro de sus cometidos es la protección de la información realizando funciones de creación y destrucción de ficheros, directorios, apertura y cierre de ficheros, lectura y escritura de ficheros y protección de acceso.

Estos 5 niveles presentan las siguientes características:

- Cada nivel realiza un subconjunto de funciones.
- Cada nivel utiliza las funciones que le brinda su nivel inferior que es más cercano a la máquina en general.
- Se dispone de interfaces bien definidas, de forma que se puede modificar un nivel sin afectar al resto de niveles.

Es importante destacar que un mismo SO puede trabajar en diferentes plataformas HW, por lo que debe poder adaptarse a cada una de ellas.

Pensemos en un programador que realiza una aplicación de gestión de archivos, teóricamente debería conocer las peculiaridades del HW donde correrá su aplicación, pero gracias al SO el programador puede abstraerse de las peculiaridades de la máquina y su aplicación funcionará correctamente independientemente del HW que tenga por debajo.

### **2.2.3.- SSOO CLIENTE/SERVIDOR**

Estos sistemas diferencian dos clases de procesos: los procesos servidores, cada uno de los cuales proporciona cierto servicio, y los procesos clientes, que utilizan estos servicios.

La comunicación entre clientes y servidores se lleva a cabo mediante el paso de mensajes. Para obtener un servicio, un proceso cliente construye un mensaje indicando lo que desea y lo envía al servidor que ofrece el servicio que necesita. Después el servidor hace el trabajo y envía de vuelta la respuesta al proceso cliente.

Hay un conjunto de llamadas al sistema mínimas, para implementar servicios básicos como manejo de interrupciones, comunicación entre procesos y planificación básica, que se ejecutarán en modo supervisor.

El resto de los servicios (gestión de memoria, sistema de archivos, operaciones de E/S, etc.), que en general son provistos por el núcleo, se ejecutan como procesos servidores en espacio de usuario y no tienen acceso directo al hardware.

## **2.3.- FUNCIONES DEL SISTEMA OPERATIVO**

Las funciones principales que realiza todo SO son:

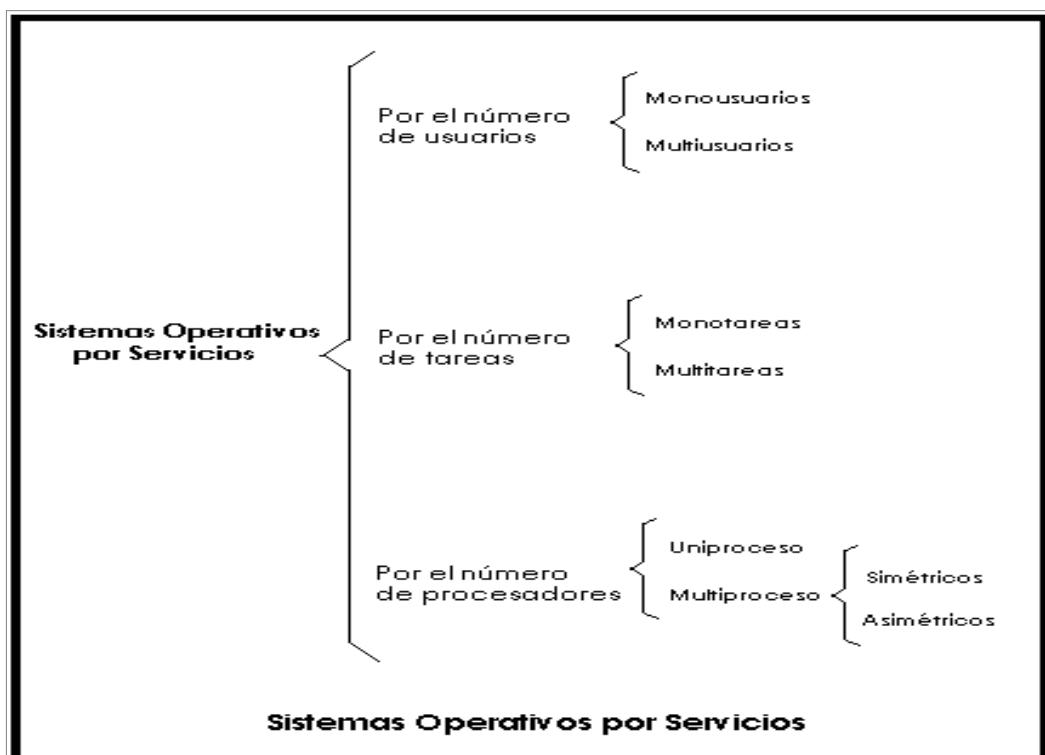
- **Control de la ejecución de los programas:** Para ello, acepta los trabajos, administra la manera en que se realizan, les asigna los recursos y los conserva hasta su finalización.
- **Administración de periféricos:** Coordinando y manipulando los dispositivos conectados al ordenador.
- **Gestión de permisos y de usuarios:** Adjudica los permisos de acceso a los usuarios y evita que las acciones de uno afecten al trabajo de otro.
- **Control de concurrencia:** Establece prioridades cuando diferentes procesos solicitan el mismo recurso.
- **Control de errores:** Gestiona los errores HW y la perdida de datos.
- **Administración de la memoria:** Asigna memoria a los procesos y gestiona su uso.
- **Control de seguridad:** Debe proporcionar seguridad tanto para los usuarios como para el SW y la información almacenada en el sistema.
- **Ofrecer una interfaz amigable al usuario:** Ofrece una forma fácil e intuitiva para que el usuario interacciones con el equipo y pueda obtener el máximo rendimiento de su trabajo

## **2.4.- CLASIFICACIONES DE SISTEMAS OPERATIVOS**

Según el punto de vista con el que veamos los SSOO pueden realizarse múltiples clasificaciones. Veamos algunas de ellas:

### **2.4.1.- POR LOS SERVICIOS OFRECIDOS**

Dependiendo de los servicios ofrecidos podemos hacer la siguiente clasificación:



#### **2.4.1.1.- CLASIFICACIÓN SEGÚN EL NÚMERO DE USUARIOS CAPACES DE ATENDER**

**Sistema Operativo Monousuario:** Los sistemas monousuario son aquellos que únicamente pueden atender a un solo usuario en cada momento debido a las limitaciones del hardware, los programas o el tipo de aplicación que se este ejecutando.

Son tipos de SSOO simples. Hay un único usuario que trabaja con el ordenador y tiene disponibles todos los recursos del sistema para él. Si otro usuario quiere utilizar el sistema ha de esperar a que termine el primero para utilizarlo.

Ejemplos de monousuarios serían MS-DOS, Windows 3.0, Windows 95 o Windows 98.

**Sistema Operativo Multiusuario:** En esta categoría se encuentran todos los SSOO que cumplen **simultáneamente** las necesidades de dos o más usuarios, que comparten los mismos recursos. Este tipo de sistemas se emplean especialmente en redes. Usan el fraccionamiento del tiempo o timesharing para poder atender a varios usuarios “simultáneamente”.

Varios usuarios pueden ejecutar procesos distintos sobre la misma CPU y compartir el uso de los recursos existentes en el sistema de forma simultánea.

Un ejemplo de SSOO multiusuario podrían ser los SSOO GNU/Linux

#### **2.4.1.2.- CLASIFICACIÓN SEGÚN EL NÚMERO DE TAREAS CAPACES DE ATENDER**

**Monoprogramación o monotarea:** Un SSOO monotarea es aquel que solo es capaz de ejecutar una única tarea en cada momento, de forma que hasta que no se acaba dicha tarea no se comienza la siguiente.

Suelen ser SSOO antiguos que no preveían una posible compartición de recursos entre programas o entre usuarios.

Estos SSOO asignan una parte de los recursos, aunque no los use, que quedan retenidos hasta que se acabe el proceso.

Podría verse como ventaja que la asignación de recursos a un usuario y a una tarea permite la concentración sobre un determinado trabajo y una mayor velocidad de éste.

La desventaja es que el rendimiento del equipo baja dado que se establece una lista de espera de usuarios o de tareas, permitiendo la entrada de uno u otro según las prioridades que tengan asignadas en el sistema.

**Multiprogramación o multitarea:** Son aquellos SSOO capaces de atender a la realización de más de una tarea en cada momento, bien sea por el uso de la multiprogramación o por el uso de múltiples procesadores.

La técnica más generalizada para tratar estos aspectos en los SSOO es la de la multiprogramación, que permite usar de manera más rápida y eficiente el microprocesador, aporta diversos marcos de memoria para distintos programas y posibilita alternar y simultanear trabajos realizados por diferentes programas.

Gestiona mejor la memoria dejando más memoria para la ejecución de los programas y aporta mecanismos de protección de memoria.

En general, los sistemas de multiprogramación se caracterizan por tener múltiples programas activos compitiendo por los recursos del sistema: procesador, memoria, dispositivos periféricos.

Esto puede presentar problemas, ya que el acceso de varios programas a un mismo recurso o a un mismo módulo y/o posición de memoria puede llegar a producir interbloqueos.

El uso de multiprogramación puede tener algún otro tipo de problema. Por ejemplo, si antes se producía un error solamente afectaba al programa ejecutado en ese instante, pero en cambio, con multiprogramación, puede afectar a todos los programas en proceso.

Para que esto no suceda y el núcleo detecte los errores de programa existe tres tipos de protección: protección de E/S, protección de memoria y protección de CPU.

#### **2.4.1.3.- CLASIFICACIÓN SEGÚN EL NÚMERO DE PROCESADORES DEL SISTEMA.**

**Monoprocesador:** Son los SSOO que se usan en los ordenadores que solo tienen un único procesador. Todos los trabajos para realizar pasan por este procesador.

**Multiprocesador:** Son los SSOO que se usan en los ordenadores que tienen varios procesadores o un procesador con varios núcleos. De esta manera el SO puede aprovechar las ventajas de tener varios procesadores.

También suelen soportar el multihilo (multithreading), es decir, subprocesos que comparten espacio de memoria. (Solaris y Windows 2000).

La forma de usar los diferentes procesadores puede hacerse de dos formas:

**Multiproceso Simétrico (SMP):** El sistema operativo utiliza la potencia por igual todos los procesadores.

**Multiproceso Asimétrico (AMP):** El SO reparte las tareas entre los diferentes procesadores.

#### **2.4.1.4.- CLASIFICACIÓN POR TIEMPO DE RESPUESTA:**

Esta clasificación se hace atendiendo al tiempo que tarda el usuario en recibir los resultados después de lanzar un programa.

**Tiempo real:** La respuesta es inmediata o casi inmediata, tras lanzar un proceso. Deben responder a unos requisitos temporales estrictos. El tiempo ocupado por el procesamiento y respuesta tras una entrada de datos supone un condicionante ineludible.

Reservados habitualmente al ámbito militar o industrial. QNX basado en Unix es un SO. de tiempo real.

**Tiempo compartido:** Cada proceso utiliza ciclos de CPU hasta que finaliza. La principal prioridad no es acabar el proceso inmediatamente.

#### **2.4.1.5.- POR LA FORMA DE OFRECER LOS SERVICIOS**

**Sistemas centralizados:** Hasta que no se popularizó y abarató el uso de la informática la mayoría de los sistemas usaban el procesamiento centralizado. Los computadores Mainframe se encargaban de todo el procesamiento y los usuarios únicamente disponían de terminales habitualmente con casi ningún recurso.

Actualmente se siguen usando estos sistemas centralizados pero los terminales ahora suelen ofrecer muchos más servicios.

**Sistemas Operativos Distribuidos:** Los que se ejecutan en varias máquinas a la vez. Permiten distribuir los trabajos entre un conjunto de procesadores. El usuario no tiene porqué saber donde se está ejecutando la aplicación que está usando, o donde se guarda la información, si en su disco local, o en cualquier otro remoto. (Amoeba, solarix-MC Sprite)

#### **2.4.1.6.- POR SU DISPONIBILIDAD**

Una **licencia** de SW es una especie de **contrato** que el desarrollador o propietario del software ofrece al usuario. En este contrato se especifican las **condiciones de uso, así como los derechos y obligaciones de ambas partes.**

Según la ley el **desarrollador está en posesión de los derechos de autor y la propiedad intelectual** por lo que suya es la potestad de decidir bajo que condiciones se utiliza su producto.

La situación más general es que el **usuario adquiera el SW pagando un precio que le da derecho a usarlo y disponer de ciertos servicios ( soporte, actualizaciones, etc. )** Esto es lo que se conoce como **SW propietario o privativo**, siempre con licencia comercial y **protegido por el copyright de forma que no puede ser modificado ni redistribuido.**

En contraposición al SW propietario existe el concepto de **SW libre ( free software)**. Existen muchos tipos de software libre, cada uno con sus características y condiciones de uso.

**Sistemas Operativos propietarios:** Son aquellos SSOO protegidos por **el Copyright de forma que no puede ser modificado ni redistribuido.** Esto implica que se necesita pagar sus licencias de uso para que el usuario ejecute el software.

No se tiene acceso a su código fuente, o aún teniendo acceso a él, no se puede modificar ni distribuir. En este grupo se encuentran la gran mayoría de los productos de Microsoft.

**Sistemas operativos libres:** Son los que garantizan las cuatro libertades del software ( según Richard Stallman)

- **La libertad de usar el programa** con cualquier propósito.
- **La libertad de estudiar cómo funciona el programa y modificarlo adaptándolo** a las necesidades que tuviera el usuario.
- **La libertad de distribuir copias del programa**, con lo que se puede ayudar a otros usuarios.
- **La libertad de mejorar el programa y hacer públicas dichas mejoras** , de forma que toda la comunidad se beneficie de ello.

La 2<sup>a</sup> y la 4<sup>a</sup> libertad requiere tener acceso al código fuente para poder estudiar y modificar el software, por lo que el **SW libre es también de código abierto.**

El SW libre suele estar disponible gratuitamente o a precio de coste de su distribución a través de otros medios; sin embargo, no es obligatorio que sea así, ya que conservando su carácter de libre podría ser distribuido comercialmente, es decir el **SW libre no necesariamente es gratuito.**

De la misma forma el SW gratuito puede incluir el código fuente , pero eso no quiere decir que sea SW libre a no ser que se garanticen los derechos de modificación y redistribución de las versiones modificadas del SW.

Algunos tipos de software libre:

**Copyleft:** El SW bajo esta licencia obliga a los usuarios a que en caso de modificar o redistribuir el SW, el producto resultante siga siendo libre. **Impide la conversión en SW propietario.**

**SW de código abierto: (open source)** Son aplicaciones que se **distribuyen de manera gratuita normalmente , incluyendo su código fuente y con la posibilidad de modificarlo, adaptarlo o incluso distribuirlo.**

**Licencia GPL:** Tipo especial de licencia de SW libre y código abierto bajo copyleft. El proyecto GNU trabaja generalmente bajo esta licencia.

**Licencia LGPL:** Tipo especial de licencia de SW libre y código abierto que permite enlazar con módulos no libres. GNU permite bajo ciertas circunstancias productos con licencias LGPL.

Otros tipos no relacionados con el free SW:

**Dominio público:** No está protegido por derechos de autor, se distribuye sin ningún tipo de licencia pues sus derechos de explotación pertenecen a todos por igual y cualquiera puede hacer uso de él, siempre con fines legales.

**Freeware:** Se distribuye normalmente de forma gratuita y con posibilidad de redistribución . No incluye el código fuente y por tanto no se puede modificar , por lo que no se incluye en la categoría de free SW.

**Shareware:** Se puede distribuir y redistribuir de forma gratuita para su evaluación, pero para poder disponer de todas las funcionalidades el usuario debe pagar una cantidad y registrarse. En ocasiones tiene un tiempo de funcionamiento limitado si no se adquiere o registra pasado un tiempo.

#### **2.4.2.- TIPOS DE LICENCIA COMERCIALES.**

En función de las licencias de distribución comercial podemos clasificar el SW comercial en:

- **OEM:** Está sujeta a la compra de algún equipo nuevo, estando prohibido venderlo si no es bajo esta condición. Aunque la compra de SW bajo esta licencia implica la propiedad del software por parte del comprador, los fabricantes pueden poner ciertas limitaciones a su uso, como el número máximo de veces que se puede reinstalar.
- **Retail:** Son las versiones de venta de software en las que el programa es completamente del comprador, pudiendo este venderlo o cederlo a terceros.
- **Licencias por volumen:** Destinada a grandes usuarios, en condiciones similares a las OEM, aunque sin estar supeditadas a la compra de equipos nuevos. Se trata de estipular un determinado número de equipos que pueden usar el mismo código de licencia. El fabricante está autorizado a comprobar que las licencias se estén usando correctamente. Se venden en paquetes de X licencias ( 25 por ejemplo). Este tipo de licencias no se puede ceder a terceros.

## 2.5.- ¿QUÉ ES UNA MÁQUINA VIRTUAL?

Es un programa que emula a un ordenador permitiendo usar e instalar más de un sistema operativo sobre la misma máquina física.

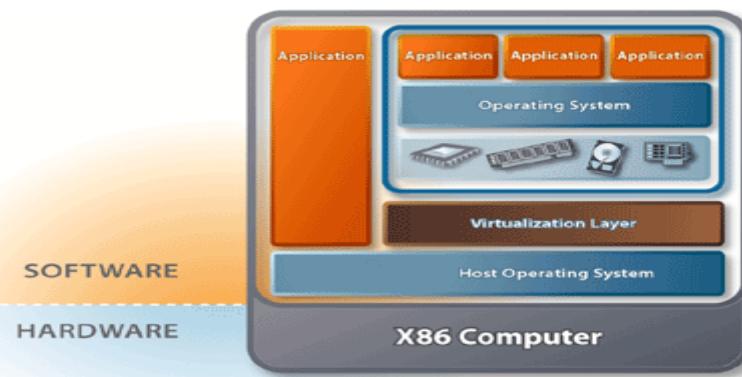
Permite ejecutar programas como si fuese un ordenador completo e independiente del equipo sobre el que se ejecuta.

A la máquina virtual se le asigna una serie de recursos cuando la creamos (RAM, dispositivos E/S). **Estos recursos son un subconjunto de la máquina física real.**

El software de virtualización permite que las instrucciones de la máquina virtual se ejecuten en el procesador de la máquina real, excepto aquellas que puedan interferir en el funcionamiento de la máquina real. Por ejemplo, cuando la máquina virtual ejecuta instrucciones que escriben información en el disco duro, el programa de virtualización toma el control de las instrucciones y realiza la escritura en un disco virtual que, generalmente, es un archivo.

Para realizar el proceso de virtualización, se utiliza dos tipos de técnicas: técnicas por software y técnicas por hardware.

**Virtualización por Software:** Las técnicas de virtualización por software posibilitan la virtualización en todo tipo de PC, antiguos y nuevos, pero las máquinas virtuales son lentas y necesita de más tiempo de microprocesador.



**Virtualización por Hardware:** Las técnicas de virtualización por hardware aceleran la ejecución del huésped, invitado o guest, pero sólo funcionan con microprocesadores modernos. En el caso de microprocesadores Intel, la virtualización por hardware se denomina VT-x y en el caso de AMD, se denomina AMD-V. En la mayoría de los ordenadores, la virtualización por hardware se debe habilitar en la BIOS.



En la virtualización por software, se coloca una capa de abstracción por encima del SO de la máquina física, esta capa suele ser una aplicación que gestiona la virtualización que se conoce como hipervisor.

En la virtualización por hardware, directamente por encima de este se coloca el hipervisor que hace las funciones de SO, aunque no es tal, sino que es un gestor de la virtualización de manera que el SO anfitrión realmente es este hipervisor.

## **2.6.- SISTEMAS OPERATIVOS EN UNA MÁQUINA VIRTUAL**

### **2.6.1.- SO ANFITRIÓN O HOST**

Es el sistema operativo que ésta instalado en nuestra máquina real. Sobre este sistema operativo host se instala el software de virtualización que permite que se instalen otros sistemas operativos (maquinas virtuales).

### **2.6.2.- SO INVITADO, HUÉSPED O GUEST**

Es el sistema operativo que se instala a través del software de virtualización. Sobre un mismo SO host, podemos tener varios sistemas operativos de este tipo (guest)

### **2.6.3.- HIPERVISOR**

Es el software que permite aplicar diversas técnicas de control de virtualización para utilizar, al mismo tiempo, diferentes sistemas operativos, en la misma máquina física.

## **2.7.- APLICACIONES DE LAS MÁQUINAS VIRTUALES**

Sin duda alguna las ventajas de la virtualización están en las aplicaciones que este software ofrece.

Varios **sistemas operativos distintos** pueden **coexistir** sobre la misma computadora, en sólido aislamiento el uno del otro, por ejemplo, para probar un sistema operativo nuevo sin necesidad de instalarlo directamente.

La máquina virtual puede proporcionar una **arquitectura de instrucciones** (ISA) que sea diferente de la verdadera máquina. Es decir, podemos simular hardware.

Varias máquinas virtuales (cada una con su propio sistema operativo llamado sistema operativo "invitado" o "guest"), pueden ser utilizadas para **consolidar servidores**. Esto permite que servicios que normalmente se tengan que ejecutar en computadoras distintas para evitar interferencias, se puedan ejecutar en la misma máquina de manera completamente aislada y compartiendo los recursos de una única computadora. La consolidación de servidores a menudo contribuye a reducir **el coste total de las instalaciones necesarias para mantener los servicios, dado que permiten ahorrar en hardware**.

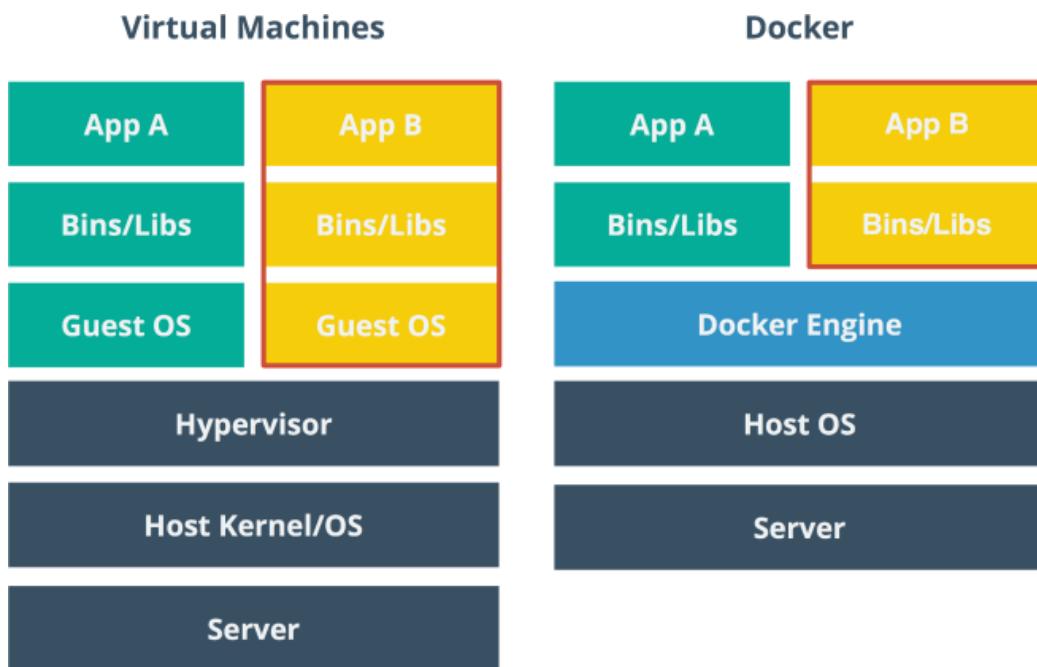
La virtualización es una excelente opción hoy día, ya que las máquinas actuales (Laptops, desktops, servidores) en la mayoría de los casos están siendo "sub-utilizadas" (gran capacidad de disco duro, memoria RAM, etc.), llegando a un uso de entre 30% a 60% de su capacidad. **Al virtualizar, la utilización de nuevas máquinas en una ya existente permite un ahorro considerable en energía, mantenimiento, espacio, etc.**

## **2.8.- VIRTUALIZACIÓN DE APLICACIONES Y SERVICIOS. DOCKERS**

En la nube cada vez es más importante garantizar el correcto funcionamiento. Es necesario utilizar una plataforma que optimice los recursos, sea lo más escalable posible para poder ampliar sus características de forma sencilla cuando sea necesario.

Hoy día la nube es sinónimo de virtualización. Ejecutar un SO invitado por cada instancia de una aplicación es un proceso pesado y lento. Para optimizar este proceso se ha creado el concepto de contenedores, capaz de optimizar recursos creando pequeños espacios virtuales de las aplicaciones necesarias. Solo se carga el núcleo de la aplicación y sus dependencias y funcionan siempre sobre un único kernel, o SO.

**Docker** es uno de estos proyectos. Lejos de ser un sistema operativo como tal, es una plataforma de **código abierto** que hace uso de las funciones de asignación de recursos del kernel de Linux para generar contenedores independientes en los que se ejecutará una única aplicación con sus respectivas dependencias y que funciona siempre en un único kernel, el de la máquina real, en lugar de virtualizar uno por cada contenedor o máquina virtual.



Usar Docker ofrece una serie de ventajas respecto a la virtualización convencional:

- **Portabilidad.** Los contenedores son portables, sin necesidad de reconfiguración.
- **Ligereza.** No virtualizan un sistema completo, solo lo necesario. El ahorro de recursos es de alrededor de un 80%.
- **Autosuficiencia.** Los contenedores tan solo deben tener lo necesario para que la aplicación funcione, por ejemplo, aquellas librerías, archivos y configuraciones necesarias para poder realizar su función.

## **2.9.- INCONVENIENTES DE LA VIRTUALIZACIÓN**

No obstante, no todo son ventajas en la virtualización, también existen algunas desventajas:

- Las máquinas virtuales agregan gran complejidad al sistema en tiempo de ejecución. Esto tiene como efecto la ralentización del sistema, es decir, el programa no alcanzará la misma velocidad de ejecución que si se instalase directamente en el sistema operativo "anfitrión" (host) o directamente sobre la plataforma de hardware. Sin embargo, a menudo la flexibilidad que ofrecen compensa esta pérdida de eficiencia.
- Rendimiento inferior. Un sistema operativo virtualizado nunca alcanzará las mismas cotas de rendimiento que si estuviera directamente instalado en el hierro. Dado que el software de virtualización introduce una capa intermedia en la gestión del hardware para gestionar las peticiones de acceso y la concurrencia al mismo, el rendimiento de la máquina virtual se ve afectado irremediablemente.
- No todas las soluciones de virtualización obtienen el mismo rendimiento en las mismas operaciones.
- No es posible utilizar hardware que no esté gestionado o soportado por el hipervisor.
- Además, el software de virtualización nos impondrá una serie de dispositivos virtuales como tarjetas de vídeo y red de las que no podremos salir.
- Hasta hace poco no se disponía de aceleración de vídeo por hardware, por lo que aplicaciones con efectos 3D y juegos que utilizaban las librerías OpenGL o DirectX no funcionaban en la máquina virtual, o al menos no lo hacían con todas sus prestaciones.
- La avería del servidor anfitrión de virtualización afecta a todas las máquinas virtuales alojadas en él.
- La portabilidad entre plataformas está condicionada a la solución de virtualización adoptada. Elegir GNU/Linux, Mac OS X, Windows o Solaris como anfitrión es una decisión importante en entornos corporativos.
- Disminuye el número de ventas de hardware. Aunque el hardware adquirido para alojar máquinas virtuales es más potente, el número de máquinas vendidas será inferior ¿Nos pondrán los fabricantes de hardware un canon por la Virtualización?
- El sistema operativo anfitrión se vuelve de rol crítico. Hasta ahora estábamos acostumbrados a utilizar un sistema para todas nuestras necesidades. Pero, cuando varios servidores o entornos de producción virtualizados dependen de la estabilidad de su anfitrión, nos pensaremos mucho antes de aplicar actualizaciones y parches. Será necesario reforzar la seguridad y estabilidad, así que habrá que tener mucho cuidado con operaciones como las actualizaciones que requieran reiniciar.

## **2.10.- SOFTWARE DE VIRTUALIZACIÓN**

Existen diferentes programas para usar virtualización, nosotros usaremos solamente uno de ellos VMware Workstation. Pero es conveniente que conozcamos, aunque sea superficialmente algún otro software para virtualizar.

### **2.10.1.- WINDOWS VIRTUAL PC**

Es un software gestor de virtualización desarrollado por Connectix y comprado por Microsoft para crear *equipos virtuales*. Emula mediante virtualización, un hardware sobre el que funciona un determinado sistema operativo. La instalación de sistemas Linux como huésped en Virtual PC es posible, pero no está oficialmente reconocida y puede traernos problemas de funcionamiento.

### **2.10.2.- VMWARE.**

Es un sistema de virtualización por software que simula un sistema físico con un hardware determinado. VMware es un software propietario, es decir se vende bajo una licencia comercial.

Cuando se ejecuta el programa (**simulador**), proporciona un ambiente de ejecución similar a todos los efectos a un computador físico, con CPU, BIOS, tarjeta gráfica, memoria RAM, tarjeta de red, sistema de sonido, conexión USB, disco duro, etc.

Entre sus versiones se encuentran:

- **VMware Workstation:** Versión de pago de las más usadas. Permite emular tantas máquinas virtuales como los recursos HW permitan.
- **VMware Server:** Esta versión que en un principio era de pago, hoy en día se puede descargar y utilizar de forma gratuita. Es más potente que Workstation. Permite ejecutar de forma concurrente más máquinas virtuales soportando servidores y ofreciendo además administración remota.
- **VMware Player:** Es un producto gratuito que permite correr máquinas virtuales creadas con otros productos VMware, pero no permite crearlas.

### **2.10.3.- ORACLE VM VIRTUALBOX**

Es un software de virtualización, creado originalmente por la empresa alemana innotek GmbH. Actualmente es desarrollado por Oracle Corporation como parte de su familia de productos de virtualización.

La aplicación fue inicialmente ofrecida bajo una licencia de software privativo, pero en 2007, surgió VirtualBox OSE bajo la licencia GPL 2.

Actualmente existe la versión privativa Oracle VM VirtualBox, que es gratuita únicamente bajo uso personal o de evaluación, y la versión Open Source, VirtualBox OSE, que es software libre, sujeta a la licencia GPL.

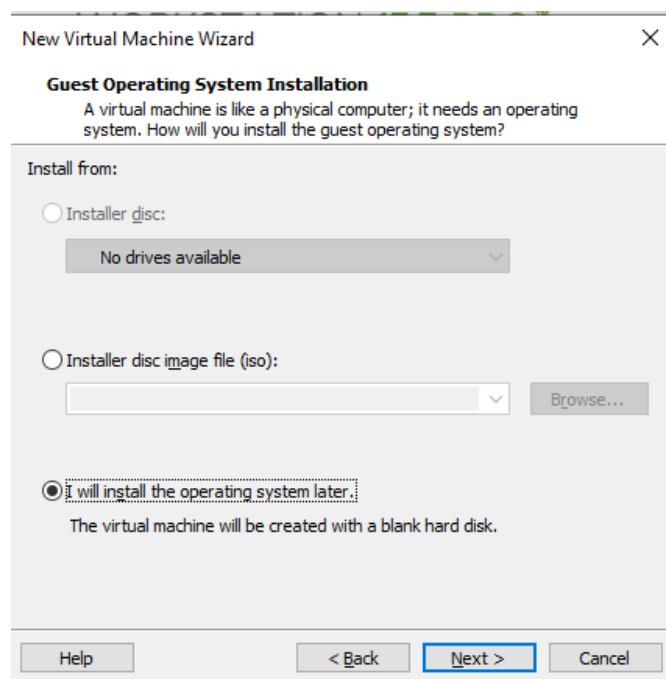
## **2.10.4.- CREAR UNA NUEVA MÁQUINA VIRTUAL EN VMWARE**

Para crear una nueva máquina virtual iniciamos VMware Workstation, haremos clic en el icono **Create a new virtual Machine** y obtendremos la siguiente ventana:

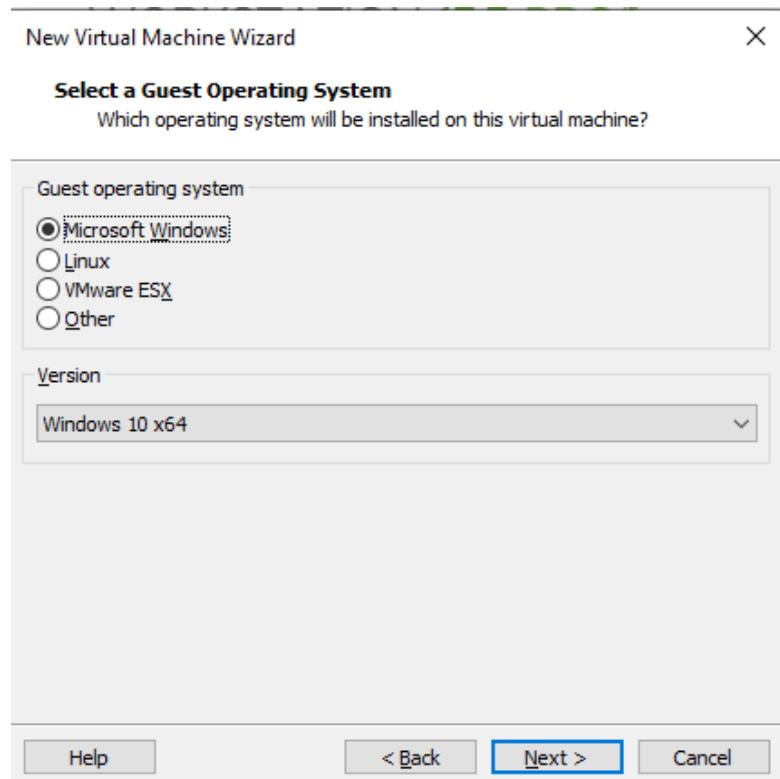


En esta ventana elegiremos si queremos configurar la máquina de forma predeterminada(Typical) o personalizada (Custom).

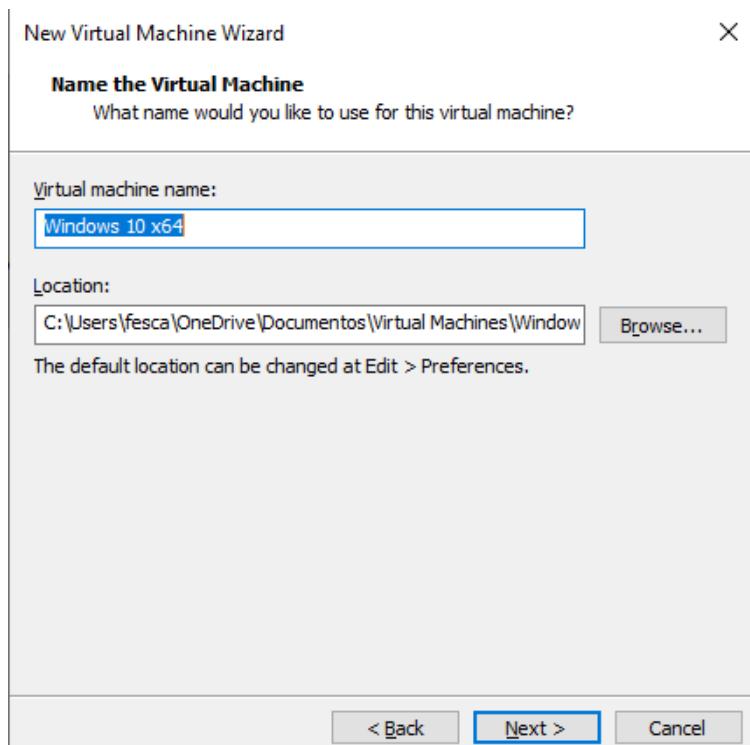
Elegimos Typical y hacemos clic en siguiente. Y nos aparece la siguiente ventana, en esta ventana elegimos instalar el SO más tarde.



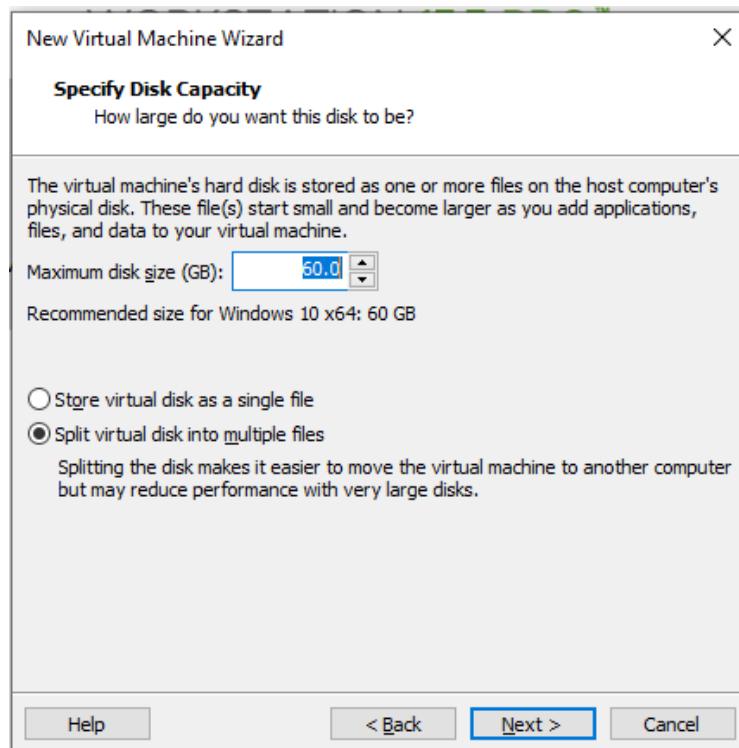
En la siguiente ventana indicaremos que SO vamos a instalar después de crear la máquina virtual.



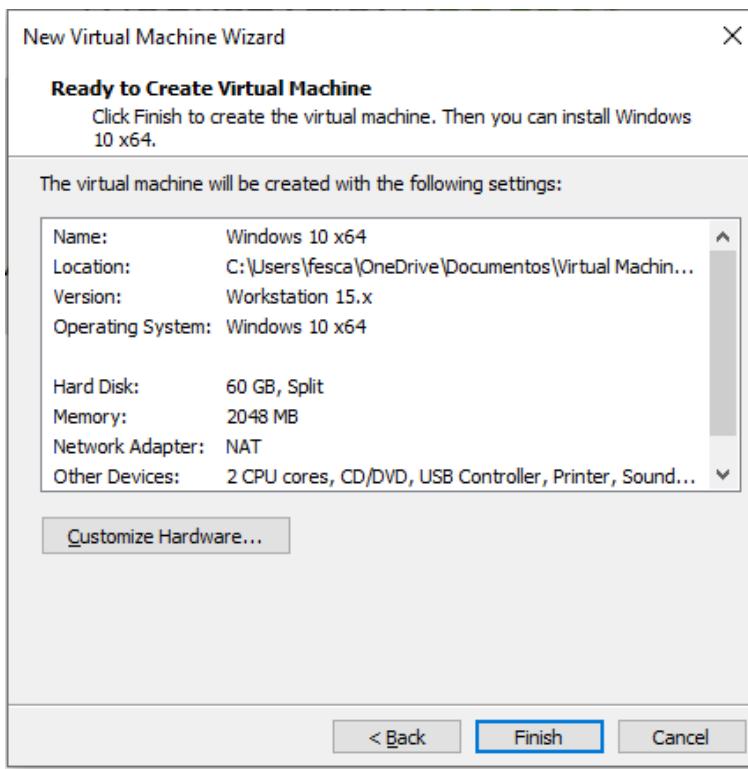
A continuación, se nos solicita el nombre que le vamos a dar a nuestra máquina virtual en VMware y donde vamos a guardar la máquina.



Tenemos que especificar ahora el tamaño del disco que queremos usar. Es importante elegir la opción de que el disco crezca gradualmente, guardándose en múltiples ficheros, según se va ocupando, porque de esta manera nuestras máquinas serán más ligeras y manejables.



Finalmente, se nos dará un informe con las características que hemos elegido y las que vienen por defecto.



Desde la entrada del menú **Edit Virtual Machine Settings** se puede personalizar las características del HW que hemos virtualizado, elegir el modo de la tarjeta de red o la memoria RAM que queremos que tenga la máquina, agregar nuevos dispositivos, discos, USB, impresoras, monitores, etcétera.

