

TEMA 2: SOFTWARE DE UN SISTEMA INFORMÁTICO

1.	SOFTWARE DE UN SISTEMA INFORMÁTICO.....	2
1.1.	TIPOS DE APLICACIONES INFORMÁTICAS.....	2
1.2.	REQUISITOS E INSTALACIÓN DE UNA APLICACIÓN.	3
1.2.1.	DETERMINACIÓN DEL EQUIPO NECESARIO.....	3
1.2.2.	EJECUCIÓN DEL PROGRAMA DE INSTALACIÓN.	5
1.2.3.	CONFIGURANDO LA APLICACIÓN.	6
1.3.	LICENCIAS SOFTWARE	6
1.3.1.	LICENCIAS DE SOFTWARE CON CÓDIGO ABIERTO.	7
1.4.	LICENCIAS DE SOFTWARE CON CÓDIGO CERRADO.....	8
2.	EL SISTEMA OPERATIVO.	10
2.1.	OBJETIVOS Y EVOLUCIÓN DE LOS SISTEMAS OPERATIVOS.	10
2.2.	TIPOS DE SISTEMAS OPERATIVOS.....	11
2.2.1.	SISTEMAS OPERATIVOS SEGÚN SU ESTRUCTURA INTERNA.....	11
2.2.2.	SISTEMAS OPERATIVOS SEGÚN LOS SERVICIOS QUE OFRECEN.....	14
2.2.3.	SISTEMAS OPERATIVOS SEGÚN LA FORMA EN LA QUE OFRECEN LOS SERVICIOS.....	14
2.3.	FUNCIONES DEL SISTEMA OPERATIVO.....	15
3.	GESTIÓN DE PROCESOS.	16
3.1.	PLANIFICACIÓN DEL PROCESADOR.....	16
3.2.	PLANIFICACIÓN APROPIATIVA Y NO APROPIATIVA.....	17
4.	GESTIÓN DE MEMORIA.	19
4.1.	GESTIÓN DE MEMORIA EN SISTEMAS OPERATIVOS MONOTAREA.....	20
4.2.	GESTIÓN DE MEMORIA EN SISTEMAS OPERATIVOS MULTITAREA.	20
4.2.1.	ASIGNACIÓN DE PARTICIONES FIJAS.	21
4.2.2.	ASIGNACIÓN DE PARTICIONES VARIABLES.....	22
4.2.3.	MEMORIA VIRTUAL.	22
5.	GESTIÓN DE ENTRADA/SALIDA.....	24
5.1.	CONTROLADORES DE DISPOSITIVO.....	24
5.2.	ESTRUCTURA DE DATOS DE LA E/S.....	25
5.3.	TÉCNICAS DE LA E/S.....	25
5.4.	PLANIFICACIÓN DE DISCOS.....	26
6.	GESTIÓN DE SISTEMA DE ARCHIVOS.	27
6.1.	ORGANIZACIÓN LÓGICA Y FÍSICA.	28
6.2.	OPERACIONES SOPORTADAS POR UN SISTEMA DE ARCHIVOS.....	29
6.3.	RUTAS DE ACCESO	29
7.	MECANISMOS DE SEGURIDAD Y PROTECCIÓN.....	30
8.	DOCUMENTACIÓN Y BÚSQUEDA DE INFORMACIÓN TÉCNICA.	31
9.	ANEXO I: EVOLUCIÓN HISTÓRICA DE LOS S.O.	32
10.	ANEXO II. GESTIÓN DE MEMORIA: TÉCNICAS DE MEMORIA VIRTUAL.	34

1. SOFTWARE DE UN SISTEMA INFORMÁTICO

En la unidad anterior definimos el concepto de sistema informático como un conjunto de elementos que hacen posible el tratamiento automatizado de la información. En esta unidad nos vamos a centrar en el **software de un sistema informático**. Éste está formado por **programas, estructuras de datos y documentación** asociada. Así, el software está distribuido en el ordenador, los periféricos y el subsistema de comunicaciones. Podemos distinguir dos tipos de software:

- **Software base:** Es el conjunto de programas necesarios para que el hardware tenga capacidad de trabajar. Recibe el nombre de **sistema operativo**.
- **Software de aplicación:** Son los programas que maneja el usuario (paquetes ofimáticos, compresores, editores de imágenes y un sinnúmero de programas más o menos específicos según el conocimiento y ámbito profesional del usuario).

1.1. TIPOS DE APLICACIONES INFORMÁTICAS

Podemos decir que las aplicaciones informáticas pueden clasificarse en dos **tipos**, en función del **ámbito o la naturaleza de uso**:

1. **Aplicaciones de propósito general:** Se emplean para el desempeño de funciones no específicas (informes, documentos, presentaciones, gráficos, hojas de cálculo, etc.) y algunas de ellas se suelen comercializar en paquetes integrados denominados suites, tales como: Microsoft Office, OpenOffice, StarOffice, Lotus SmartSuite, etc.
 - ✓ **Gestión de texto:**
 - Editores de texto (no permiten formato, como por ejemplo Notepad).
 - Procesadores de texto (Microsoft Word, Writer de OpenOffice).
 - Programas de autoedición, maquetación y diseño: Microsoft Publisher.
 - ✓ **Hoja de cálculo** (Microsoft Excel, Calc de OpenOffice, Lotus 1-2-3).
 - ✓ **Asistente personal:** agenda, calendario, listín telefónico.
 - ✓ **Generador de presentaciones** (Microsoft PowerPoint, Impress de OpenOffice).
 - ✓ **Herramientas de acceso y gestión de bases de datos** (Microsoft Access, Base de OpenOffice).
 - ✓ **Editores de XML y HTML** (Microsoft FrontPage).
 - ✓ **Herramientas para la comunicación:** groupware o trabajo en grupo como gestores de e-mail, servicio de mensajería instantánea, gestión de FAX, etc.
 - ✓ **Utilidades y herramientas:** antivirus, navegadores web, gestores de archivos, compresores de archivos, visores de archivos, etc.
2. **Aplicaciones de propósito específico:** Se utilizan para el desempeño de funciones específicas, científicas, técnicas o de gestión, tales como:
 - ✓ **Administración, contabilidad, facturación, gestión de almacén, RRHH:** ContaPlus.
 - ✓ **Entornos gráficos de desarrollo:** Visual Studio, Borland Builder C++, etc.
 - ✓ **Herramientas de administración de bases de datos:** Oracle, phpMyAdmin, etc.
 - ✓ **Herramientas de gestión de red:** Tivoli, NetView, etc.

TEMA 2: Software de un sistema informático.

- ✓ **Herramientas “ad-hoc” especializadas:** OCR¹/OMR², monitores bursátiles, gestión empresarial ERP³, etc.
- ✓ **Herramientas de diseño gráfico y maquetación:** Corel Draw, Visio, Adobe PhotoShop, PaintShop, etc.
- ✓ **Herramientas de ingeniería y científicas** utilizadas en ámbitos de investigación, en universidades, etc.

1.2. REQUISITOS E INSTALACIÓN DE UNA APLICACIÓN.

¿Qué ocurre cuando **queremos instalar una aplicación** software en el ordenador? En todo **proceso de instalación** se han de seguir unos pasos que si no se realizan adecuadamente podemos encontrarnos con un funcionamiento limitado o erróneo de la aplicación. Los **pasos** serían:

1. **Determinación del equipo necesario.**
2. **Ejecución del programa de instalación.**
3. **Configuración de la aplicación.**

1.2.1. DETERMINACIÓN DEL EQUIPO NECESARIO.

Lo primero que debemos hacer es conocer **qué necesita la aplicación** para que funcione adecuadamente en el ordenador, es decir, qué **características o requisitos** tendrá que tener el sistema informático. Cada desarrollador crea sus aplicaciones enfocadas a plataformas concretas, con unas necesidades de hardware y software necesarias para su funcionamiento. Una aplicación creada para una plataforma no podrá ser instalada en otra distinta. Tampoco podrá ser instalada si nuestro sistema informático no cumple los requisitos mínimos. Antes de proceder a la instalación de una aplicación tendrá que reunir la información sobre el hardware de su ordenador y deberá verificar que su hardware le permite realizar el tipo de instalación que desea efectuar.

Las **características para que la aplicación se ejecute adecuadamente** pueden ser de **naturaleza hardware**:

- ✓ **Plataforma hardware:** PC, Mac, etc.
- ✓ **Procesador:** Fabricante y velocidad (generalmente se indica el inferior posible de la gama con el que la aplicación funciona adecuadamente).
- ✓ **Memoria RAM mínima.**
- ✓ **Espacio mínimo disponible en el soporte de almacenamiento:** En disco duro o en una unidad de almacenamiento externa para aplicaciones portables.
- ✓ **Tarjeta gráfica:** La memoria gráfica necesaria para el buen funcionamiento de la aplicación.
- ✓ **Resolución recomendada del monitor.**

¹ El Reconocimiento Óptico de Caracteres (OCR), así como el reconocimiento de texto, en general son aplicaciones dirigidas a la digitalización de textos. Identifican automáticamente símbolos o caracteres que pertenecen a un determinado alfabeto, a partir de una imagen para almacenarla en forma de datos.

² La lectura óptica de marca (OMR) es una tecnología que permite la recogida automática de gran fiabilidad de los datos a partir de documentos de una o varias páginas. El software OMR permite la creación de documentos, su lectura "inteligente" y el análisis inmediato de los datos.

³ Los sistemas ERP o de planificación de recursos empresariales manejan principalmente la producción, logística, distribución, inventario, envíos, facturas y contabilidad de una empresa.

TEMA 2: Software de un sistema informático.

Y de carácter **software**:

- ✓ **Plataforma software:** Sistema operativo bajo el que funciona la aplicación, como Windows, Linux, etc.
- ✓ **Otros paquetes software adicionales** necesarios, tales como actualizaciones concretas de seguridad para el sistema operativo, la JVM (máquina virtual de Java), el Flash Player, etc. Por ejemplo, para instalar el editor de imágenes de Microsoft te indica que debes tener instalado varios componentes de Microsoft.

Teniendo en cuenta lo visto anteriormente, los fabricantes de aplicaciones informáticas suelen establecer **tres niveles de requisitos** para la instalación de sus aplicaciones:

1. Equipo básico: En las **especificaciones básicas** del fabricante aparecerán los siguientes elementos:

- ✓ Plataforma hardware.
- ✓ Tipo de microprocesador.
- ✓ RAM mínima necesaria.
- ✓ Espacio mínimo necesariamente disponible en el soporte de almacenamiento.
- ✓ Plataforma software.
- ✓ Resolución recomendada del monitor
- ✓ Tarjeta gráfica necesaria.

2. Equipo opcional: Además de los requisitos mínimos, los fabricantes suelen especificar configuraciones de equipos que convengan para conseguir un funcionamiento eficaz de la aplicación. En muchos casos, la configuración básica (requisitos mínimos) no permite funcionar a la aplicación con un rendimiento satisfactorio. Entre las recomendaciones suelen aparecer:

- ✓ Plataforma hardware recomendada.
- ✓ Cantidad de memoria adicional recomendada.
- ✓ Presencia de coprocesador matemático (para aplicaciones que realicen gran cantidad de cálculos).
- ✓ Tipo de impresora.
- ✓ Capacidad recomendada para la tarjeta gráfica.

3. Equipo en red: Actualmente muchos fabricantes diseñan las aplicaciones para su funcionamiento en red. El fabricante indicará, si la aplicación se ha diseñado para trabajar en red, las **configuraciones para el servidor y los clientes** (ordenadores que hacen peticiones al servidor), así como los **sistemas operativos en red** bajo los que puede trabajar. La **configuración de la red** (tarjetas de comunicación y topología) no depende de la aplicación sino del sistema operativo.

El fabricante indicará además si existe alguna limitación en cuanto al **número máximo de clientes** que pueden trabajar con la aplicación. Así, por cada estación en la que se instale la aplicación habría que adquirir una licencia de uso para los ordenadores cliente.

Por ejemplo, en el sistema operativo *Windows Server* se limita el número de ordenadores clientes a los que atiende en función del número de licencias que exista para ello, esta opción de configuración se especifica durante la instalación del servidor.

TEMA 2: Software de un sistema informático.

1.2.2. EJECUCIÓN DEL PROGRAMA DE INSTALACIÓN.

Por la instalación de un programa o aplicación informática entendemos el conjunto de pasos que nos van a permitir copiar los archivos necesarios, configurar, implantar y poner en funcionamiento una aplicación en un sistema informático.

La mayoría de las aplicaciones presentan dos niveles en función de los conocimientos del usuario:

- ✓ **Instalación básica:** Este nivel está diseñado para usuarios con pocos conocimientos informáticos. El programa realizará una instalación en función de los elementos que detecte en el equipo y según unos parámetros básicos establecidos por defecto por el fabricante.
- ✓ **Instalación personalizada o avanzada:** Permite a un usuario experto incluir o eliminar elementos de la aplicación con el fin de optimizar los recursos sistema informático, instalando sólo aquellos elementos de la aplicación que se van a utilizar. Por ejemplo, la instalación personalizada del paquete Microsoft Office permite elegir los programas a instalar (Microsoft Word, Excel, PowerPoint, Frontpage, etc.).

Cuando se adquiere una aplicación informática, nos encontramos con un grupo de manuales y de DVD o CD. La aplicación se encuentra normalmente en formato comprimido. El traspaso del programa al soporte de almacenamiento de nuestro ordenador, normalmente el disco duro, se realiza a través del **programa de instalación** (su nombre puede ser setup, install, instalar, etc.), y es el encargado de **extraer los bloques de la aplicación** de los discos, descomprimiéndolos si es necesario; **crear la estructura de directorios necesaria, ubicar los archivos** de la aplicación donde corresponda, y, si fuera necesario, **modificar el registro del sistema**.

En la actualidad, la mayoría de los fabricantes distribuyen también sus aplicaciones en formato DVD, CD o con posibilidad de descarga de los archivos de instalación o en imágenes ISO⁴ (por ejemplo: muchas distribuciones de Linux pueden descargarse en este formato).

En las versiones actuales de Windows disponemos de la herramienta *Microsoft Store* para la instalación de software y en Linux la forma más común para la instalación de software es a través de repositorios oficiales con ayuda de la herramienta *Ubuntu Software* o aplicaciones como *Synaptic* que es un sistema de gestor de paquetes. Estas herramientas se conectan a un repositorio como puede ser el de la distribución Ubuntu en <http://archive.ubuntu.com/> que contienen binarios o paquetes precompilados por ejemplo .deb en distribuciones Debian/Ubuntu y se encargan de realizar la instalación de la aplicación en nuestro sistema automáticamente descargando cualquier paquete del que dependa. Actualmente existen paquetes universales que funcionan en cualquier distribución Linux sin necesidad de ser instalados en el sistema como **AppImage** o **Snap**.

Conoce las posibilidades que te ofrecen las aplicaciones portables. Puedes llevarlas en tu memoria USB y utilizarlas donde y cuando quieras, sin necesidad de instalación. Para ello, visita este enlace:

[Aplicaciones portables](#)

⁴ Es un archivo donde se almacena una copia o imagen exacta de un sistema de ficheros, normalmente un disco óptico. Se rige por el estándar ISO 9660 que le da nombre. Algunos de los usos más comunes incluyen la distribución de sistemas operativos, tales como sistemas Linux, BSD o Live CD.

1.2.3. CONFIGURANDO LA APLICACIÓN.

Una vez realizado correctamente el proceso de instalación sólo queda configurar las opciones de la aplicación, a veces también del sistema operativo, y configurar el entorno de trabajo. En la configuración se pueden modificar los parámetros establecidos por defecto para la aplicación. Algunas aplicaciones pueden generar una serie de archivos de configuración con los datos introducidos por los usuarios. El usuario debe realizar una última tarea antes de comenzar a utilizar la aplicación: **configurar el entorno de trabajo**. Esto consiste en definir una serie de parámetros de funcionamiento que adecuen el funcionamiento de la aplicación a las exigencias del usuario. Este proceso se realizará en el caso que no sea satisfactoria la configuración establecida por defecto por el programa. Entre estos **parámetros** aparecen:

- ✓ **Ajuste y distribución de la pantalla** (tamaños de las ventanas, colores, tipos de letras, cambio de resolución, etc.).
- ✓ **Definición de directorios de trabajo** (directorios para los archivos, proyectos, plantillas, etc.).

Como ejemplo de configuración dentro de las opciones de la parte servidora de la aplicación de control remoto⁵ UltraVNC, se nos permite cambiar la contraseña de administrador, cambiar los puertos por defecto, etc. Por otro lado, algunas aplicaciones web requieren la activación de cookies⁶ y la modificación de la configuración de seguridad de nuestro navegador.

Tras este último paso de configuración la aplicación ya está lista para empezar a funcionar adecuadamente.

1.3. LICENCIAS SOFTWARE

Comenzaremos definiendo algunos **conceptos clave** para entender gran parte de lo que rodea a las **licencias software**.

En primer lugar, las **licencias software** nos sirven para establecer un **contrato entre el autor** de una aplicación software (sometido a propiedad intelectual y a derechos de autor) y el **usuario**. En el contrato se definen con precisión los **derechos y deberes de ambas partes**, es decir, los “**actos de explotación legales**”.

Por otra parte, entendemos por **derecho de autor o copyright** la **forma de protección proporcionada por las leyes vigentes** en la mayoría de los países **para los autores de obras originales** incluyendo obras literarias, dramáticas, musicales, artísticas e intelectuales, tanto publicadas como pendientes de publicar.

Pueden existir tantas licencias software como acuerdos concretos se den entre el autor y el usuario, pero en general podemos clasificarlas en los siguientes **tipos**: licencias de software con código abierto y con código cerrado.

Software de dominio público (sin licencia): Existe también la posibilidad de tener software publicado sin licencia cuyos derechos son de explotación para toda la humanidad. Como es lógico, este tipo de software puede utilizarse, modificarse, redistribuirse o licenciarse sin ningún tipo de limitaciones, con o sin fines de lucro.

Esto ocurre cuando el autor lo dona a la humanidad o si los derechos de autor han expirado (en un plazo contado desde la muerte del autor, generalmente 70 años). En caso de que el autor condicione el uso de su software bajo una licencia, por muy débil que sea, ya no se consideraría software de dominio público.

⁶ Es un fragmento de información que se almacena en el disco duro del visitante de una página web por medio de su navegador, bajo petición del servidor de la página. Esta información puede ser recuperada más tarde por el servidor en posteriores visitas. Entre sus funciones están la de llevar el control de usuarios y conseguir información de hábitos de navegación del usuario.

TEMA 2: Software de un sistema informático.

1.3.1. LICENCIAS DE SOFTWARE CON CÓDIGO ABIERTO.

Las **licencias de software con código abierto** ponen a disposición de los usuarios el **código fuente** con el que está construido el software. Encontramos dos subtipos:

- ✓ **Licencias permisivas (con permisos).** El software puede modificarse o dar lugar a otro nuevo (redistribuirse), sin que el resultado tenga que mantener la misma licencia del software original. Ejemplos: Apache Software License, PHP License, Perl License, Python License, W3C Software Notice and License, BSD License, MIT License, Open LDAP License, Perl License, VMS License.
- ✓ **Licencias no permisivas (con restricciones),** también llamadas **copyleft**. Pueden ser:
 - **Fuertes o con copyleft fuerte:** El software puede modificarse o dar lugar a otro nuevo, pero el resultado deberá mantener la misma licencia del software original. Este tipo de licencias suelen estar unidas al concepto de **software libre**. Ejemplos: Common Public License, GNU General Public License, Eclipse Public License, Sleepycat Software Product License, Affero License, OpenSSL License.
 - **Débiles o con copyleft débil:** Las modificaciones que se realicen del software original deberán mantener la misma licencia, pero si da lugar a otro software nuevo, éste sí podría tener distinta licencia del software original. Ejemplos: GNU Lesser General Public License, Mozilla Public License, Open Source License, Apple Source License, CDDL, EUPL.

El **software libre** proporciona al usuario las **cuatro libertades** siguientes:

- ✓ **Libertad 0. Utilizar el programa,** para cualquier propósito.
- ✓ **Libertad 1. Estudiar cómo funciona el programa** y adaptarlo a tus necesidades, debe proporcionarse las fuentes, directa o indirectamente, pero siempre de forma fácil y asequible.
- ✓ **Libertad 2. Distribuir copias exactas del programa.**
- ✓ **Libertad 3. Modificar el programa y hacer públicas las modificaciones a los demás.**

Todo programa que no incorpore alguna de estas libertades se considera no libre o **semilibre**.

La mayor parte de las licencias de software libre surgen de la FSF⁷.

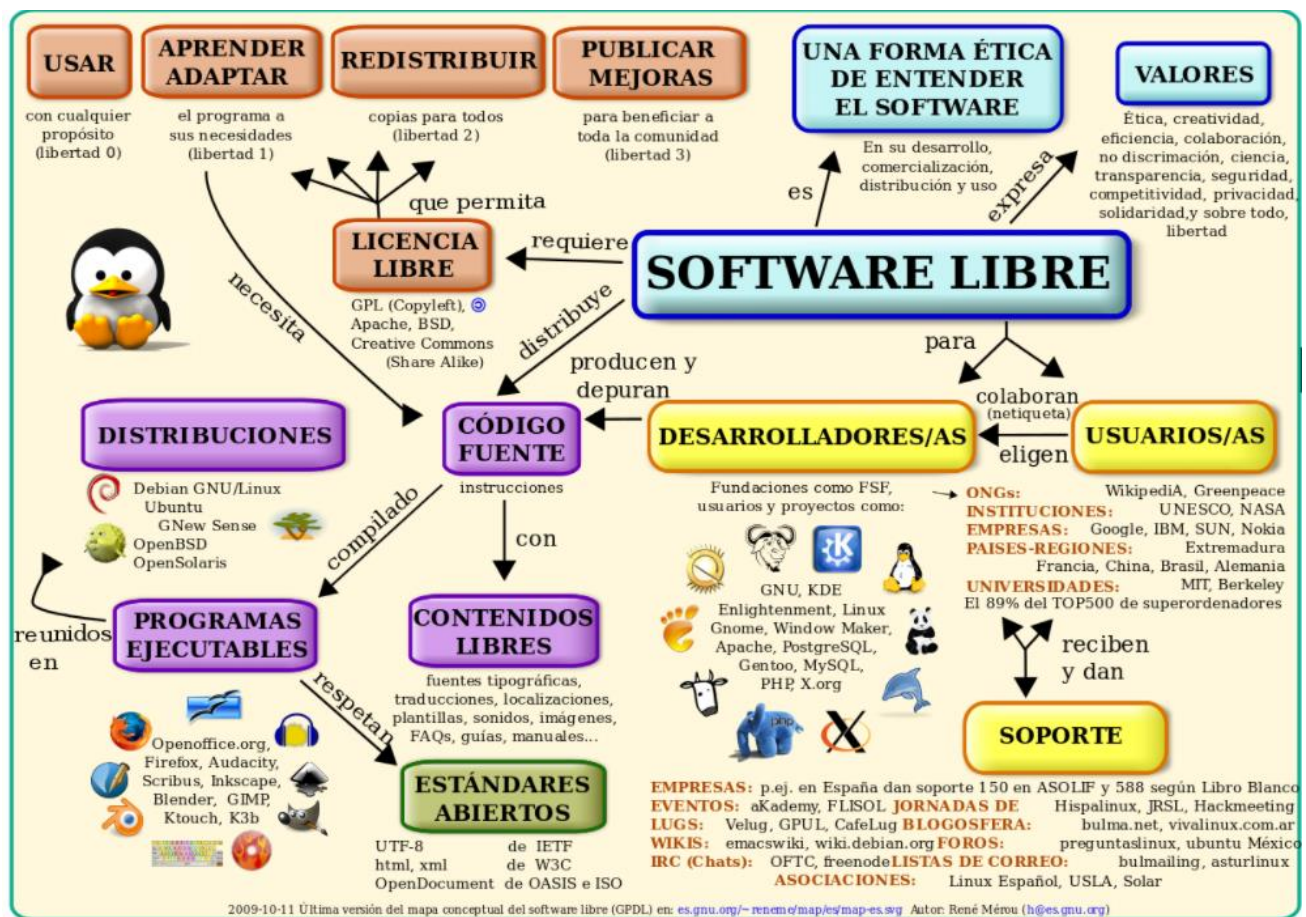
El software libre suele estar disponible gratuitamente, o al precio de costo de la distribución a través de otros medios, pero no es obligatorio que sea así, por lo que **no hay que asociar software libre a "software gratuito"** (denominado usualmente **freeware**), ya que, conservando su carácter de libre, puede ser distribuido comercialmente.

Veamos algunos **ejemplos** de aplicaciones de **software libre**:

- ✓ Sistemas Operativos: Debian GNU/Linux, Ubuntu, Linex, Guadalinex, MAX, etc.
- ✓ Entornos de escritorio: GNOME, KDE, etc.
- ✓ Aplicaciones de oficina: OpenOffice, KOffice, LATEX, etc.
- ✓ Navegación web: FireFox, Konqueror, etc.
- ✓ Aplicaciones para Internet: Apache, Zope, etc.

⁷ Free Software Foundation) o Fundación para el software libre es una organización creada por Richard Stallman y otros seguidores del software libre para promocionar el desarrollo y uso del software libre en todas las áreas de la computación.

TEMA 2: Software de un sistema informático.



Conoce más a fondo la protección legislativa de los programas de ordenador en España:

R.D 1/1996, de Propiedad Intelectual

1.4. LICENCIAS DE SOFTWARE CON CÓDIGO CERRADO

Las **licencias de software con código cerrado** no permiten la distribución del **código fuente** con el que está construido el software. También suelen llamarse **software propietario** o **privativo**.

En estas licencias los propietarios limitan las posibilidades que tienen los usuarios para utilizar, copiar, modificar, redistribuir o ceder el software. Si no se cumpliera lo que hay estipulado en los términos y condiciones de la licencia estaríamos hablando de **piratería de Software**.

Además, suelen ofrecer servicios de soporte técnico y actualizaciones durante el tiempo de vida del producto otorgado en la licencia.

TEMA 2: Software de un sistema informático.

Como ejemplos de estas licencias tenemos:

- ✓ **Licencias ALUF**, en algunos países llamadas **CLUF** o en inglés **EULA**: El uso del software sólo está permitido para un único usuario (el comprador).
- ✓ **Freeware**: Se aplica a software que se distribuye gratuitamente y por tiempo indefinido. Es posible que requiera que nos registremos, pero siempre de forma gratuita. Normalmente está permitida su redistribución, pero no la modificación. No suele incluir el código fuente.
- ✓ **Shareware**: Permite que el software se evalúe, aunque puede limitar el tiempo de uso o algunas de sus funcionalidades. Para disponer del producto sin limitaciones habría que pagarlo.

Las *licencias propietarias o privativas* pueden adquirirse a través de diferentes vías:

- ✓ **Retail**, también llamada **FPP**: Esta es la forma habitual en la que un usuario compraría un software en un establecimiento. Normalmente, está dirigida a usuarios que no necesitan más de cinco licencias y pueden encontrarse dos variantes:
 - **Producto completo**: No requiere una versión previa del software para su instalación en el ordenador del usuario.
 - **Actualización (Upgrade)**: Parte de una versión previa del software para la que ya tenemos licencia. Normalmente, este tipo de licencias tienen un coste menor.
- ✓ **OEM**: Es la licencia del software que viene preinstalado cuando se adquiere un equipo nuevo, estando prohibida explícitamente su venta si no forma parte de un todo. El software no podrá utilizarse ni tan siquiera en el caso de hacer una sustitución del equipo por otro, salvo que ésta sea por motivos de garantía. Lo que sí suele estar permitido es ceder el equipo completo (*hardware y software*) a un usuario diferente.
- ✓ **Licencias por volumen**: Los fabricantes de software suelen tener contratos dirigidos a entidades o empresas, de diferentes tamaños, que necesiten un número de licencias mayor que un usuario normal. El contrato puede incluir derechos específicos, como, por ejemplo, los derechos de transferencia a determinados usuarios que cumplan con unas características concretas. En este tipo de licencias también encontramos las opciones de producto completo o actualización. Además, es frecuente que dispongamos de derechos de **downgrade**, para dar soporte a sistemas más antiguos.

2. EL SISTEMA OPERATIVO.

En este apartado introduciremos gran parte de la teoría en la que están basados los sistemas operativos actuales. El sistema operativo (SO) es un conjunto de programas que se encarga de gestionar los recursos hardware y software del ordenador, por lo que actúa como una interfaz entre los programas de aplicación del usuario y el hardware puro.

2.1. OBJETIVOS Y EVOLUCIÓN DE LOS SISTEMAS OPERATIVOS.

Los principales **objetivos de los sistemas operativos** son:

- ✓ **Abstraer al usuario de la complejidad del hardware:** El sistema operativo hace que el ordenador sea más fácil de utilizar.
- ✓ **Eficiencia:** Permite que los recursos del ordenador se utilicen de la forma más eficiente posible. Por ejemplo, se deben **optimizar** los accesos a disco para acelerar las operaciones de entrada y salida.
- ✓ **Permitir la ejecución de programas:** Cuando un usuario quiere ejecutar un programa, el sistema operativo realiza todas las tareas necesarias para ello, tales como cargar las instrucciones y datos del programa en memoria, iniciar dispositivos de entrada/salida y preparar otros recursos.
- ✓ **Acceder a los dispositivos entrada/salida:** El sistema operativo suministra una interfaz homogénea para los dispositivos de entrada/salida para que el usuario pueda utilizar de forma más sencilla los mismos.
- ✓ **Proporcionar una estructura y conjunto de operaciones** para el **sistema de archivos**.
- ✓ **Controlar el acceso al sistema y los recursos:** En el caso de sistemas compartidos, proporcionando protección a los recursos y los datos frente a usuarios no autorizados.
- ✓ **Detección y respuesta ante errores:** El sistema operativo debe **prever** todas las posibles **situaciones críticas y resolverlas**, si es que se producen.
- ✓ **Capacidad de adaptación:** Un sistema operativo debe ser construido de manera que pueda evolucionar a la vez que surgen actualizaciones hardware y software.
- ✓ **Gestionar las comunicaciones en red:** El sistema operativo debe permitir al usuario manejar con facilidad todo lo referente a la instalación y uso de las redes de ordenadores.
- ✓ **Permitir a los usuarios compartir recursos y datos:** Este aspecto está muy relacionado con el anterior y daría al sistema operativo el papel de gestor de los recursos de una red.



¿Sabes cómo han ido cambiando los sistemas operativos desde sus inicios? Resulta interesante conocer la evolución histórica que han sufrido los sistemas operativos para comprender mejor las características que explicaremos más adelante. Lee este interesante documento: Ver Anexo I.- Evolución histórica de los sistemas operativos



Evolución histórica del PC.

2.2. TIPOS DE SISTEMAS OPERATIVOS.

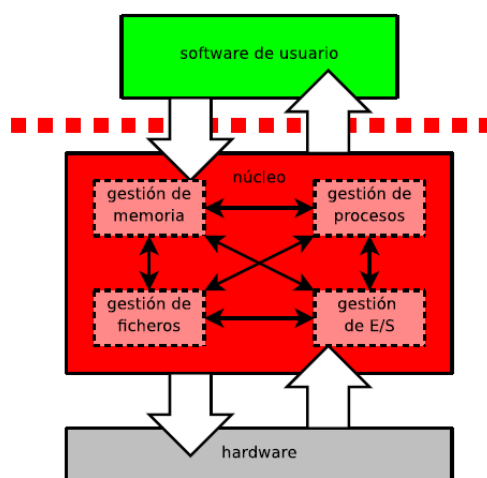
Ahora vamos a clasificar los sistemas operativos en base a su estructura interna, los servicios que ofrecen y la forma en la que ofrecen los servicios.

Estructura interna	Servicios que ofrecen	Forma de ofrecer los servicios
Monolítico	Monousuario	Centralizado
Jerárquico	Multiusuario	
Máquina Virtual	Monotarea	En red
Microkernel o Cliente-Servidor	Multitarea	
Híbrido	Monoprocesador	Distribuido
	Multiprocesador	

2.2.1. SISTEMAS OPERATIVOS SEGÚN SU ESTRUCTURA INTERNA.

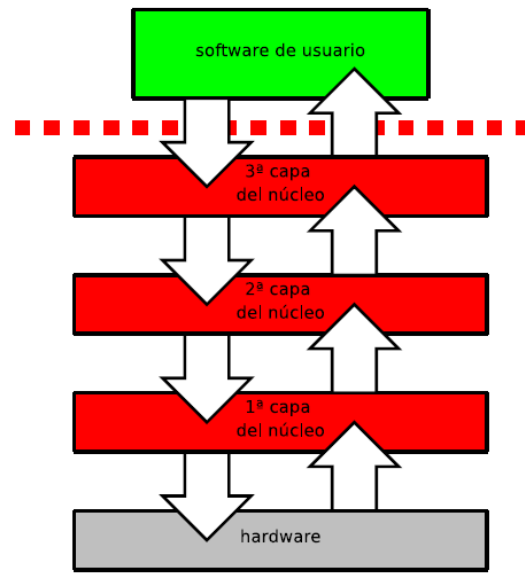
Esta clasificación se basa en cómo se diseñan los sistemas a la hora de ser creados. Hay que tener en cuenta que, en la mayoría de los casos, estas concepciones de diseño no se aplican aisladas, sino que puede haber interrelación entre ellas:

- ✓ **Monolítico:** Es la estructura utilizada en los primeros sistemas operativos en la que todas las funciones se implementaban en el núcleo. El sistema operativo consistía en un único programa desarrollado con rutinas entrelazadas que podían llamarse entre sí. Por lo general, eran sistemas operativos hechos a medida, pero difíciles de mantener.

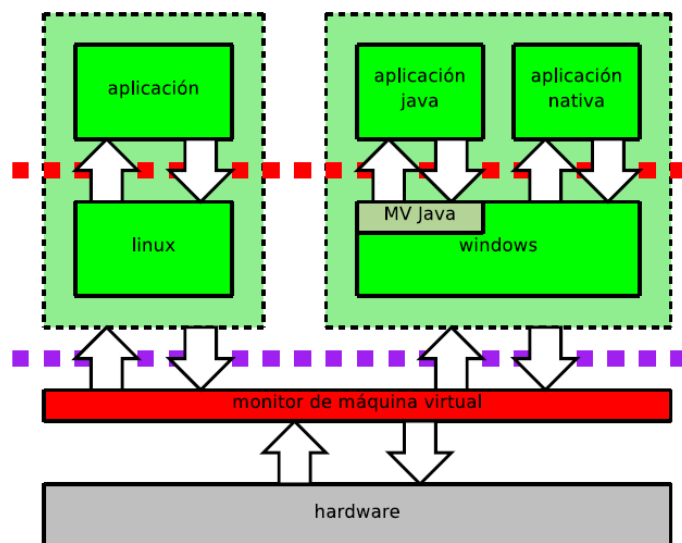


TEMA 2: Software de un sistema informático.

- ✓ **Jerárquico:** Conforme las necesidades de los usuarios aumentaron, los sistemas operativos fueron creciendo en complejidad y funciones. Esto llevó a que se hiciera necesaria una mayor organización del software del sistema operativo, dividiéndose en partes más pequeñas (llamados niveles), diferenciadas por funciones y con una interfaz clara para interoperar con los demás elementos. Cada uno de los niveles se comunica con el nivel inmediatamente inferior y superior de tal forma que todos ellos están coordinados y consiguen el objetivo del sistema operativo. Un ejemplo de este tipo de sistemas operativos fue MULTICS.

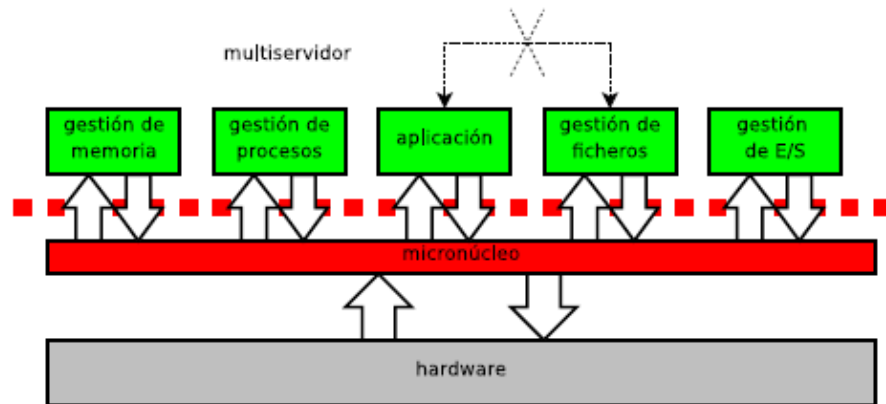


- ✓ **Máquina Virtual:** El objetivo de los sistemas operativos es el de integrar distintos sistemas operativos dando la sensación de ser varias máquinas diferentes. Presentan una interfaz a cada proceso, mostrando una máquina que parece idéntica a la máquina real subyacente. Estas máquinas no son máquinas extendidas, son una réplica de la máquina real, de manera que en cada una de ellas se pueda ejecutar un sistema operativo diferente, que será el que ofrezca la máquina extendida al usuario. VMware y VM/CMS son ejemplos de este tipo de sistemas operativos.

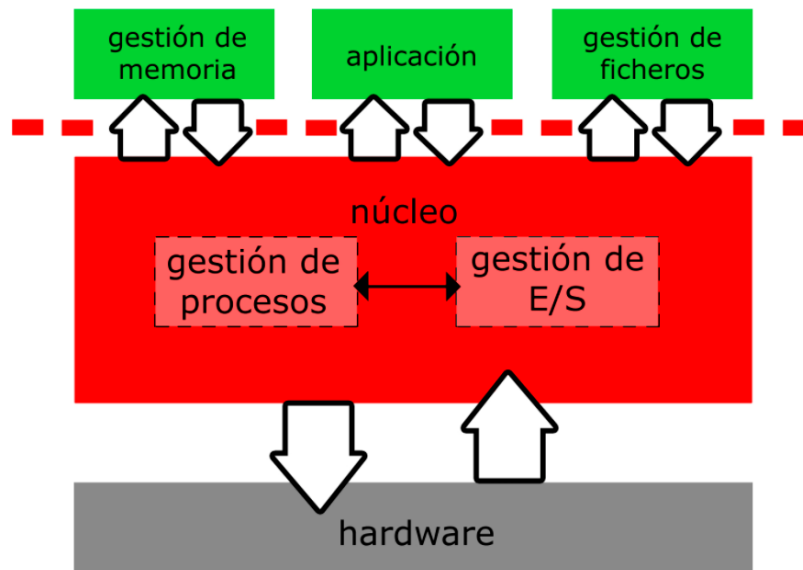


TEMA 2: Software de un sistema informático.

- ✓ **Microkernel o Cliente-Servidor:** El modelo del núcleo de estos sistemas operativos distribuye las diferentes tareas en porciones de código modulares y sencillas. El objetivo es aislar del sistema, su núcleo, las operaciones de entrada/salida, gestión de memoria, del sistema de archivos, etc. Esto incrementa la tolerancia a fallos, la seguridad y la portabilidad entre plataformas de hardware. Algunos ejemplos son Mach, MINIX 3 o AIX.



- ✓ **Híbrido:** Esta estructura es en realidad una combinación entre la monolítica y la microkernel con la idea de incluir en el núcleo ciertas funciones para que se ejecuten más rápido de lo que se haría si estuviera en espacio de usuario.



TEMA 2: Software de un sistema informático.

2.2.2. SISTEMAS OPERATIVOS SEGÚN LOS SERVICIOS QUE OFRECEN.

Esta clasificación podemos dividirla a su vez en:

✓ **Según el número de usuarios:**

- **Monousuario:** son aquellos que soportan a un usuario a la vez, sin importar el número de procesos o tareas que el usuario pueda ejecutar en un mismo instante de tiempo. Ejemplos de sistemas operativos de este tipo son MS-DOS, Microsoft Windows 9x y ME, MAC OS, entre otros.
- **Multiusuario:** son capaces de dar servicio a más de un usuario a la vez, ya sea por medio de varias terminales conectadas al ordenador o por medio de sesiones remotas en una red de comunicaciones. No importa el número de procesadores en la máquina ni el número de procesos que puede ejecutar cada usuario simultáneamente. Algunos ejemplos serán UNIX, GNU/Linux, Microsoft Windows Server o MAC OS X.

✓ **Según el número de procesos o tareas:**

- **Monotarea:** sólo permiten una tarea a la vez por usuario. Se puede dar el caso de un sistema multiusuario y monotarea, en el cual se admiten varios usuarios simultáneamente pero cada uno de ellos puede ejecutar sólo una tarea en un instante dado. Ejemplos de sistemas monotarea son MS-DOS, Microsoft Windows 3.x y 95 (estos últimos sólo simulan la multitarea).
- **Multitarea:** permite al usuario realizar varias tareas al mismo tiempo. Algunos ejemplos son MAC OS, UNIX, Linux, Microsoft Windows 98, 2000, XP, Vista y 7.

✓ **Según el número de procesadores:**

- **Monoprocesador:** es aquel capaz de manejar sólo un procesador, de manera que si el ordenador tuviese más de uno le sería inútil. MS-DOS y MAC OS son ejemplos de este tipo de sistemas operativos.
- **Multiprocesador:** es capaz de manejar varios procesadores del sistema para distribuir su carga de trabajo e incrementar así el poder de procesamiento. Estos sistemas trabajan de dos formas: **simétricamente** (los procesos son enviados indistintamente a cualquiera de los procesadores disponibles) y **asimétricamente** (uno de los procesadores actúa como maestro o servidor y distribuye la carga de procesos a los demás).

2.2.3. SISTEMAS OPERATIVOS SEGÚN LA FORMA EN LA QUE OFRECEN LOS SERVICIOS.

Por la forma de ofrecer los servicios, los sistemas operativos pueden ser:

- ✓ **Sistemas operativos centralizados:** Permite utilizar los recursos de un solo ordenador, es decir, su memoria, CPU, disco y periféricos. Se denomina así porque el ordenador es único y no necesita trabajar en paralelo con ningún otro. En estos sistemas hay un uso ocasional de la red, como para transferir ficheros o logins remotos, pero son utilidades o funciones agregadas que permite realizar el sistema operativo centralizado, sin llegar a ser lo que buscaba como objetivo principal el sistema al ser diseñado. Hasta que los ordenadores personales no tuvieron un precio accesible y suficiente potencia, la mayoría de los sistemas utilizaban el sistema de proceso centralizado. Hoy en día son muy conocidos los sistemas centralizados con los que contamos, basta con empezar por los que tenemos instalados en nuestros ordenadores: Windows, Linux, Mac OS, Unix, etc.

TEMA 2: Software de un sistema informático.

- ✓ **Sistemas operativos en red:** Estos sistemas tienen la capacidad de interactuar con los sistemas operativos de otras máquinas a través de la red, con el objeto de intercambiar información, transferir archivos, etc. La clave de estos sistemas es que el usuario debe conocer la ubicación de los recursos en red a los que desee acceder. Los sistemas operativos modernos más comunes pueden considerarse sistemas en red, por ejemplo: Novell, Windows Server, Linux, etc.
- ✓ **Sistemas operativos distribuidos:** Abarcan los servicios de red, las funciones se distribuyen entre diferentes ordenadores, logrando integrar recursos (impresoras, unidades de respaldo, memoria, procesos, etc.) en una sola máquina virtual que es a la que el usuario accede de forma transparente. En este caso, el usuario no necesita saber la ubicación de los recursos, sino que los referencia por su nombre y los utiliza como si fueran locales a su lugar de trabajo habitual. MOSIX es un ejemplo de estos sistemas operativos.

Comparativa entre Windows 10 y Ubuntu 16.04

2.3. FUNCIONES DEL SISTEMA OPERATIVO.

El sistema operativo necesita administrar los recursos para tener control sobre las funciones básicas del ordenador. Pero, **¿cuáles son los recursos que gestiona el sistema operativo?** Los principales recursos que administra el sistema operativo son:

- ✓ **El procesador.**
- ✓ **La memoria.**
- ✓ **Los dispositivos de entrada/salida.**
- ✓ **El sistema de archivos.**

Para gestionar todos estos recursos, existe **una parte muy importante** del sistema operativo, **el núcleo o kernel**. El núcleo normalmente **representa sólo una pequeña parte de todo lo que es el sistema operativo**, pero es una de las partes que más se utiliza. Por esta razón, el **núcleo reside** por lo general **en la memoria principal**, mientras que otras partes del sistema operativo son cargadas en la memoria principal sólo cuando se necesitan.

Resumiendo, el **núcleo** supone la **parte principal** del **código de un sistema operativo** y se encarga de **controlar y administrar los servicios y peticiones de recursos**. Para ello se divide en distintos **niveles**:

- ✓ **Gestión de procesos.**
- ✓ **Gestión de memoria.**
- ✓ **Gestión de la entrada/salida (E/S).**
- ✓ **Gestión del sistema de archivos.**
- ✓ **Mecanismos de seguridad y protección.**

GESTIÓN DE PROCESOS

GESTIÓN DE MEMORIA

GESTIÓN DE E/S

GESTIÓN DEL SISTEMA
DE ARCHIVOS

3. GESTIÓN DE PROCESOS.

Entre las principales tareas del sistema operativo está la de **administrar los procesos del sistema**.

¿A qué nos referimos cuando hablamos de procesos? **Un proceso es un programa en ejecución**. Un proceso simple tiene un **hilo de ejecución** (o subproceso). En ocasiones, un proceso puede dividirse en varios subprocesos. Un hilo es básicamente una tarea que puede ser ejecutada en paralelo con otra tarea. Por lo que los hilos de ejecución permiten a un programa realizar varias tareas a la vez.



En los sistemas operativos modernos los procesos pueden tener diferentes estados, según el momento de creación, si están en ejecución, si se encuentran a la espera de algún recurso, etc. Pero podemos hacer una simplificación, y un proceso, en un instante dado, puede estar en uno de los tres estados siguientes:

- ✓ **Listo o preparado.** Puede pasar a estado de ejecución si el planificador del sistema operativo lo selecciona, esto es, cuando llegue su turno (según el orden de llegada o prioridad).
- ✓ **En ejecución.** Se está ejecutando en el procesador en un momento dado.
- ✓ **Bloqueado.** Está esperando la respuesta de algún otro proceso para poder continuar con su ejecución, por ejemplo, una operación de entrada/salida.

El sistema operativo sigue la pista de en qué estado se encuentran los procesos, decide qué procesos pasan a ejecución, cuáles quedan bloqueados, en definitiva, **gestiona los cambios de estado de los procesos**.

Los **procesos pueden comunicarse entre sí o ser independientes**. En el primer caso, los procesos necesitarán sincronizarse y establecer una serie de mecanismos para la comunicación; por ejemplo, los procesos que pertenecen a una misma aplicación y necesitan intercambiar información. En el caso de procesos independientes, estos, por lo general, no interactúan y un proceso no requiere información de otros.

3.1. PLANIFICACIÓN DEL PROCESADOR.

En la planificación del procesador **se decide cuánto tiempo de ejecución se le asigna a cada proceso del sistema y en qué momento**. Si el sistema operativo es monousuario y monotarea no habrá que decidir, pero en el resto de los sistemas multitarea esta decisión es fundamental para el buen funcionamiento del sistema, ya que determinará la correcta ejecución de los distintos programas de aplicación que se estén ejecutando.

El sistema operativo almacena en una tabla denominada **tabla de control de procesos**, la información relativa a cada proceso que se está ejecutando en el procesador. Ésta es:

- ✓ Identificación del **proceso**.
- ✓ Identificación del **proceso padre**.
- ✓ Información sobre el **usuario y grupo** que lo han lanzado.
- ✓ **Estado del procesador:** El contenido de los registros internos, contador de programa, etc. Es decir, el entorno volátil del proceso.
- ✓ Información de **control de proceso**.
- ✓ Información del **planificador**.

TEMA 2: Software de un sistema informático.

- ✓ **Segmentos de memoria asignados.**
- ✓ **Recursos asignados.**

Una **estrategia de planificación** debe buscar que los procesos obtengan sus turnos de ejecución de forma apropiada (**momento en que se le asigna el uso de la CPU**), junto con un buen rendimiento y minimización de la sobrecarga (overhead) del planificador mismo. En general, se buscan **cinco objetivos principales**:

- ✓ **Equidad:** Todos los procesos en algún momento obtienen su turno de ejecución o intervalos de tiempo de ejecución hasta su terminación con éxito.
- ✓ **Rendimiento:** El sistema debe finalizar el mayor número de procesos por unidad tiempo.
- ✓ **Tiempo de respuesta:** El usuario no percibirá tiempos de espera demasiado largos.
- ✓ **Tiempo de retorno:** Evitar el aplazamiento indefinido, los procesos deben terminar en un plazo finito de tiempo. Esto es, el usuario no debe percibir que su programa se ha parado o “colgado”.
- ✓ **Eficacia:** El procesador debe estar ocupado el 100% del tiempo.

La **carga de trabajo de un sistema informático a otro puede variar** considerablemente, esto depende de las **características de los procesos**. Nos podemos encontrar:

- ✓ **Procesos** que hacen un **uso intensivo de la CPU**.
- ✓ **Procesos** que realizan una gran cantidad de **operaciones de Entrada/Salida**.
- ✓ **Procesos por lotes, procesos interactivos, procesos en tiempo real.**
- ✓ **Procesos de menor o mayor duración.**

En función de cómo sean la mayoría de los procesos habrá algoritmos de planificación que den un mejor o peor rendimiento al sistema.

3.2. PLANIFICACIÓN APROPIATIVA Y NO APROPIATIVA.

La **planificación no apropiativa** (en inglés, no preemptive) es aquella en la que, cuando a un proceso le toca su turno de ejecución, **ya no puede ser suspendido**; es decir, no se le puede arrebatar el uso de la CPU, hasta que el proceso no lo determina no se podrá ejecutar otro proceso. Este esquema **tiene sus problemas**, puesto que, si el proceso contiene **ciclos infinitos**, el resto de los procesos pueden quedar aplazados indefinidamente. Otro caso puede ser el de los procesos largos que penalizarían a los cortos si entran en primer lugar. La **planificación apropiativa** (en inglés, preemptive) supone que el sistema operativo **puede arrebatar el uso de la CPU** a un proceso que esté ejecutándose. En la planificación apropiativa existe un reloj que lanza interrupciones periódicas en las cuales el planificador toma el control y se decide si el mismo proceso seguirá ejecutándose o se le da su turno a otro proceso.

En ambos enfoques de planificación se pueden establecer distintos algoritmos de planificación de ejecución de procesos. Algunos de los **algoritmos para decidir el orden de ejecución de los procesos en el sistema son**:

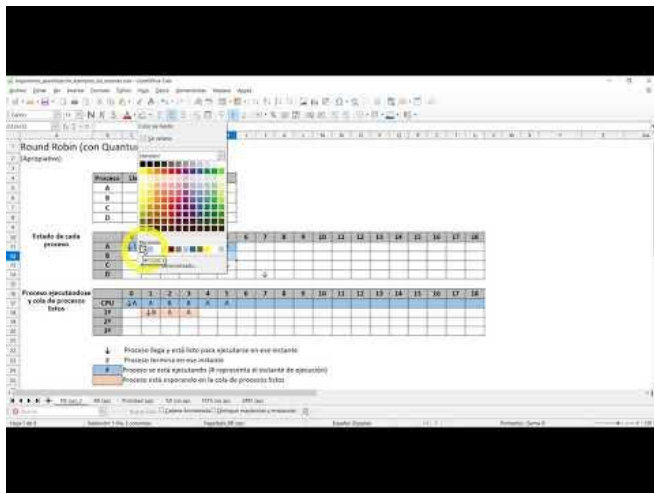
A continuación, puedes ver una serie de vídeos con ejemplos simples explicados en los que se muestra el funcionamiento básico de estos algoritmos de planificación. Puedes visualizar los vídeos directamente en YouTube para verlos a mayor tamaño y con mejor resolución:

Algoritmos de planificación de procesos

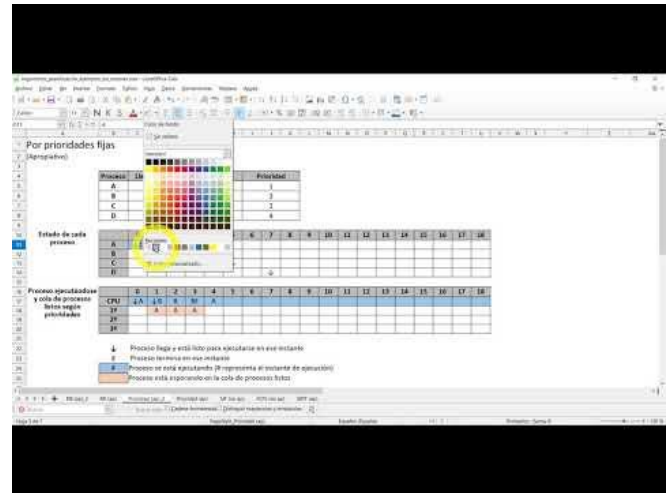
Round Robin
Por prioridad
El trabajo más corto primero
El primero en llegar, primero en ejecutarse
El tiempo restante más corto

TEMA 2: Software de un sistema informático.

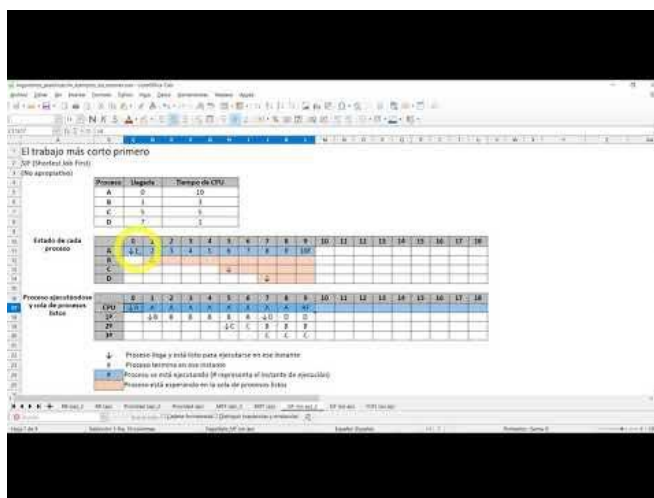
Round Robin



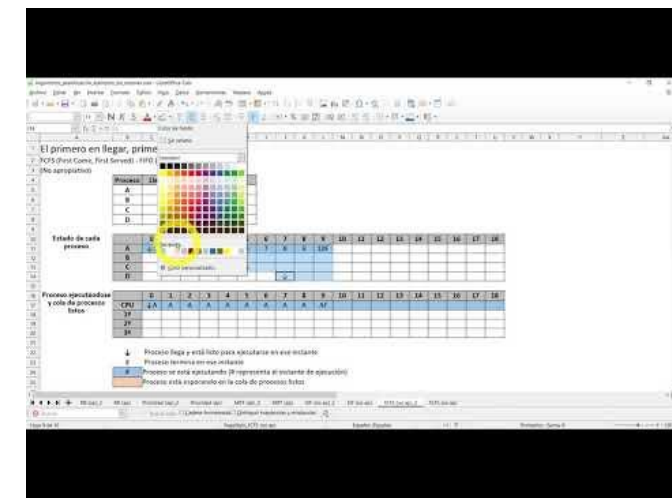
Por prioridades



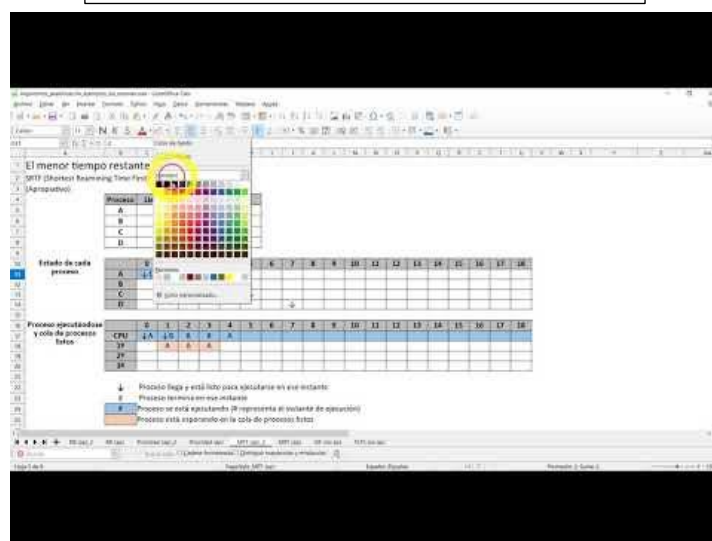
El trabajo más corto primero



El primero en llegar, primero en ejecutarse



El tiempo restante más corto primero.



¿Quieres saber cómo se comunican y sincronizan entre sí los procesos? Visita el siguiente enlace:

[Comunicación y sincronización de procesos](#)

4. GESTIÓN DE MEMORIA.

Hemos visto en la gestión de procesos que el recurso compartido es el procesador. Sin embargo, para que un proceso se pueda ejecutar no sólo requiere tiempo de procesamiento sino también estar cargado en memoria principal. Esto es así, porque **ningún proceso se puede activar antes de que se le asigne el espacio de memoria que requiere**. Así, la memoria se convierte en otro recurso clave que tendrá que gestionar el sistema operativo y la parte encargada de ello se denomina **gestor de memoria**.

La **función principal del gestor de memoria** es la de **asignar memoria principal a los procesos que la soliciten**. Otras funciones serán:

- ✓ **Controlar las zonas de memoria** que están asignadas y cuáles no.
- ✓ **Asignar memoria a los procesos** cuando la necesiten y **retirársela cuando terminen**.
- ✓ **Evitar que un proceso acceda a la zona de memoria asignada a otro proceso**.
- ✓ Gestionar el **intercambio entre memoria principal y memoria secundaria** en los casos en que la memoria principal está completamente ocupada, etc.

De este modo, la gestión de memoria va a tener que cubrir los siguientes **requisitos**:

- ✓ **Reubicación:** En un sistema multitarea la memoria va a estar compartida entre varios procesos, el gestor de memoria debe decidir qué **zonas de memoria asigna a cada proceso** y que zonas descarga.
- ✓ **Protección:** El gestor de memoria debe **evitar que los procesos cargados en memoria interfieran unos con otros** accediendo a zonas de memoria que no les corresponden, Para ello, se comprueba que las referencias a la memoria generadas por un proceso durante su ejecución sólo hacen referencia a la zona de memoria asignada a ese proceso y no acceden a zonas prohibidas, áreas de memoria donde estén otros procesos.
- ✓ **Control de memoria:** El sistema operativo, a través del gestor de memoria, tiene que **controlar las zonas de memoria libres y las asignadas**, además de saber las zonas de memoria que corresponden a cada proceso.
- ✓ **Controlar y evitar en lo posible casos de fragmentación de la memoria:** Existen **dos tipos de fragmentación de la memoria principal**, la **fragmentación interna** y la **externa**. La **fragmentación interna** sucede al **malgastarse el espacio interno de una partición** cuando el proceso o bloque de datos cargado es más pequeño que la partición. Por el contrario, la **fragmentación externa** sucede cuando la memoria externa a todas las particiones se divide cada vez más y van **quedando huecos pequeños y dispersos** en memoria difícilmente reutilizables.
- ✓ **Organización lógica y física:** En ocasiones la memoria principal no es suficiente para proporcionar toda la memoria que necesita un proceso o para almacenar todos los procesos que se pueden ejecutar. Entonces los procesos pueden ser intercambiados a disco y más tarde, si es necesario, vueltos a cargar en memoria. Por lo que el gestor de memoria se encarga de **gestionar la transferencia de información entre la memoria principal y la secundaria (disco)**.

El sistema de gestión de la memoria que se use **dependerá del ordenador y sistema operativo** en particular que se tenga. Las opciones en la gestión de memoria se dividen en función del número de procesos albergados en memoria (monotarea/multitarea) y de si se utiliza memoria real o virtual.

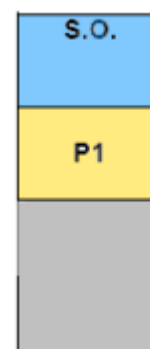
TEMA 2: Software de un sistema informático.

Memoria Real	Memoria Real		Memoria Virtual	
Monotarea	Multitarea		Multitarea	
	Particiones		Memoria virtual paginada	Memoria virtual segmentada
	Fijas	Variables	Combinación	
	Paginación pura	Segmentación pura		
	Relocalización		Protección	

4.1. GESTIÓN DE MEMORIA EN SISTEMAS OPERATIVOS MONOTAREA.

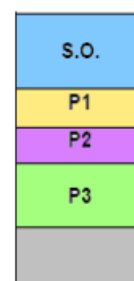
En sus orígenes los sistemas operativos no incluían ningún gestor de memoria, y el programador tenía un control completo sobre el espacio total de memoria. La memoria real se utiliza para almacenar el programa que se esté ejecutando en un momento dado. Conforme los procesos se ejecutan secuencialmente a medida que van terminando los anteriores.

Se trata del esquema más sencillo, en cada momento la memoria alberga un solo proceso y reserva otra zona de la memoria para el sistema operativo. Por ello, se necesita un mecanismo de protección para evitar accesos a la parte del sistema operativo de los procesos de usuario.



4.2. GESTIÓN DE MEMORIA EN SISTEMAS OPERATIVOS MULTITAREA.

Actualmente la mayoría de los sistemas operativos son sistemas multitarea, en los que va a haber varios procesos simultáneamente en ejecución. Para que esto sea posible, todos estos procesos deberán estar también simultáneamente en memoria, pues ésta es una condición necesaria para que un proceso pueda ejecutarse. Por tanto, deberá **haber mecanismos de gestión para distribuir la memoria principal entre todos estos procesos que quieren ejecutarse.**



Intercambio o swapping

Como sabemos la memoria principal es un recurso limitado, por ello puede ocurrir **que haya más procesos esperando a ser cargados en memoria que zonas libres** en la misma. En estos casos, el gestor de memoria sacará de la memoria algunos procesos (bloqueados, suspendidos, que estén esperando a que finalice una operación de entrada/salida, etc.) y los llevará a un área de disco (memoria secundaria), conocida como **área de intercambio o de swap**. A esta operación se la denomina **intercambio o swapping**. Los procesos permanecerán allí hasta que existan huecos libres en memoria y puedan ser recuperados de disco y reubicados en memoria principal.



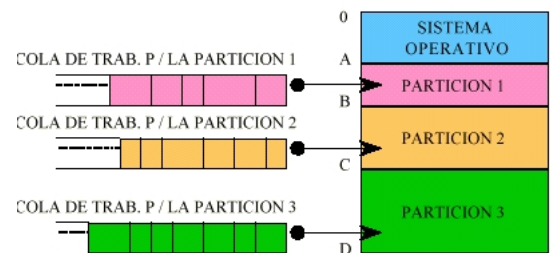
4.2.1. ASIGNACIÓN DE PARTICIONES FIJAS.

Hemos estudiado que el gestor de memoria necesita reservar un espacio de memoria para el sistema operativo y que el resto de la memoria queda para los procesos de usuarios. **Cuando existen varios procesos que requieren ser cargados en memoria el gestor de memoria tiene que organizar el espacio** para ubicarlos.

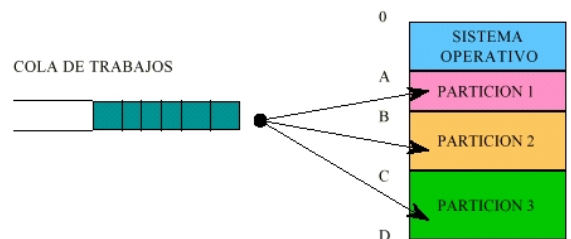
Hay varias alternativas, la primera de ellas es **dividir el espacio de memoria en particiones fijas**. Estas particiones podrán ser todas del mismo tamaño o tener distintos tamaños. Estas particiones se establecen de forma lógica por el sistema operativo y **están predefinidas antes de que lleguen los procesos**. **El número de particiones se mantiene fijo en el tiempo, así como el tamaño de cada una de las particiones**.

La gestión y asignación de particiones a los procesos se puede hacer siguiendo dos tipos de organización:

- ✓ **Una cola por partición:** Se tiene **una cola por cada partición** y se coloca cada trabajo en la cola de la partición más pequeña en que quepa dicho trabajo, a fin de desperdiciar el menor espacio posible. **La planificación de cada cola se hace por separado** y, como cada cola tiene su propia partición, no hay competencia entre las colas por la memoria. La desventaja de este método se hace evidente cuando la cola de una partición grande está vacía y la cola de una partición pequeña está llena.



- ✓ **Una única cola común a todas las particiones:** Se tiene una única cola común para todas las particiones. **El sistema operativo decidirá en que partición se ubica cada proceso**. En función de la disponibilidad de particiones y las necesidades del proceso en cuestión.

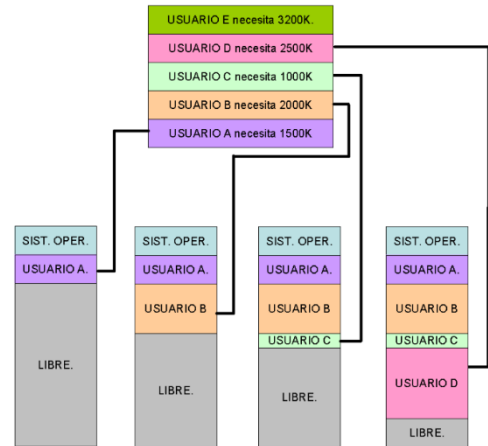


En ambos casos, utilización de una cola por partición o uso de una única cola para los procesos, el gestor de memoria establecerá **mecanismos para impedir que un proceso pueda acceder a una zona de memoria que está fuera de la memoria correspondiente a la partición en la que se encuentra**.

Además de esto, puede surgir el problema de la **fragmentación**, la cual se produce, cuando en la memoria hay áreas ocupadas intercaladas con áreas libres; es decir, cuando no hay una única área ocupada ni una única área libre.

4.2.2. ASIGNACIÓN DE PARTICIONES VARIABLES.

Con la asignación de particiones fijas se tiene la desventaja de que no se aprovecha, con frecuencia, todo el tamaño de cada partición, ya que el proceso se adapta a los tamaños fijos ya preestablecidos en memoria. En este punto se plantea una segunda alternativa, la asignación de memoria a los procesos mediante particiones variables. La idea es **crear las particiones dinámicamente, conforme llegan los procesos y en función de los tamaños de estos**. Este método de gestión de memoria se conoce con el nombre de **asignación de la memoria con particiones variables**. Es una técnica más realista que aprovecha mejor el espacio de la memoria.



- ✓ Este mecanismo se ajusta a la realidad de que el número y tamaño de los procesos varía dinámicamente y, por tanto, lo lógico es que no se está sujeto a un número fijo de particiones que pudieran ser muy grandes o demasiado pequeñas, con lo que **se consigue un mejor uso de la memoria aunque a costa de una mayor complejidad**.
- ✓ En la asignación de particiones variables, el sistema operativo debe llevar el control de qué partes de la memoria están disponibles y cuáles libres.

4.2.3. MEMORIA VIRTUAL.

Hasta este momento los procesos se cargaban enteros en la memoria, pero podría suceder que existan procesos grandes que no quepan en las particiones de la memoria y por tanto, no puedan ser cargados por completo en la memoria.

La **memoria virtual** da una solución a estos casos, ya que **permite dividir los procesos en varias partes y cargar sólo algunas de ellas en memoria**. La memoria virtual se basa en el uso de las técnicas de **paginación o segmentación**.

¿En qué consisten la técnica de paginación y segmentación? Conoce su funcionamiento básico por medio del siguiente recurso:

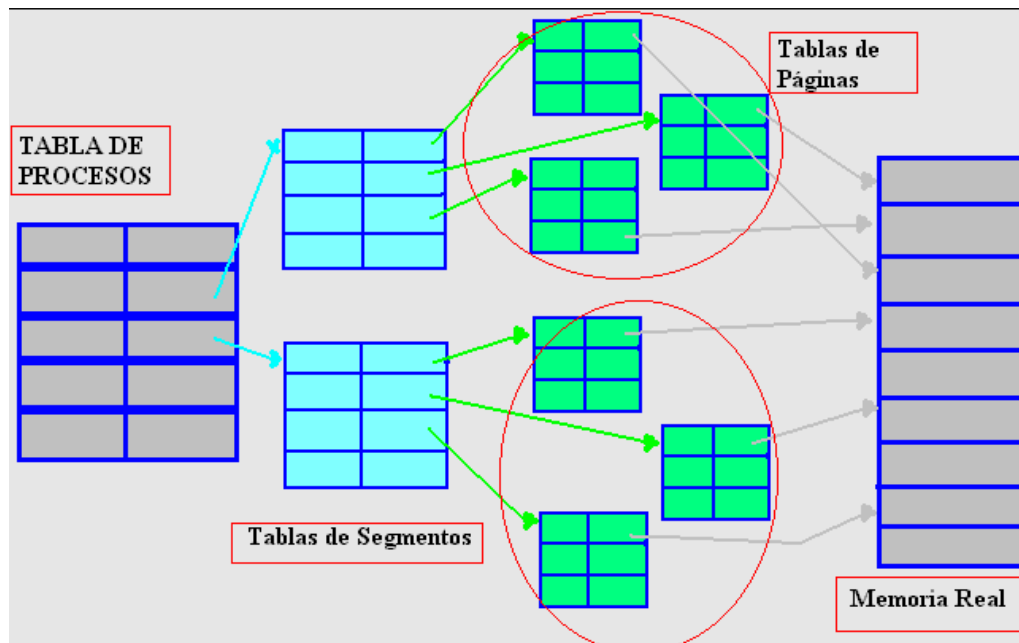
Ver Anexo II.- Gestión de memoria: Técnicas de memoria virtual

Como hemos comentado, no todas las partes de un proceso pueden estar cargadas en memoria en un instante determinado. Por ello, cuando un proceso haga referencia a un parte que no se encuentre asignada en memoria provocará un **fallo de página o segmento**, y el gestor de memoria traerá dicha parte del proceso de disco a memoria.

TEMA 2: Software de un sistema informático.

La utilización de las técnicas de paginación o segmentación por parte de la memoria virtual se conocen como:

- ✓ **Memoria Virtual Paginada:** Sigue el funcionamiento de la paginación simple, pero **no es necesario cargar todas las páginas de un proceso** para que éste pueda ejecutarse. Las páginas que no se encuentren y se necesiten se traerán posteriormente a memoria de manera automática. Reduce la fragmentación
- ✓ **Memoria Virtual Segmentada:** En este caso la operación sería la misma que en la segmentación simple, pero tampoco será necesario cargar todos los segmentos de un proceso. Si se necesitan más segmentos no asignados en memoria se traerán en el momento en que sean referenciados.
- ✓ **Combinación de las técnicas de segmentación y paginación:** En la figura siguiente vemos el funcionamiento de la combinación de ambas técnicas.



¿Sabes cómo gestionan Linux y Windows la memoria? Consulta el siguiente enlace para averiguarlo:

[El uso de la memoria en Windows y Linux](#)

5. GESTIÓN DE ENTRADA/SALIDA.

Anteriormente, vimos que una de las funciones del ordenador era procesar la información, dicha información la obtiene y muestra a través de los periféricos. La parte del **sistema operativo** que se encarga de este proceso es la **gestión de la E/S (entrada/salida)**. En la primera unidad estudiamos los periféricos y recordamos que se clasificaban en periféricos:

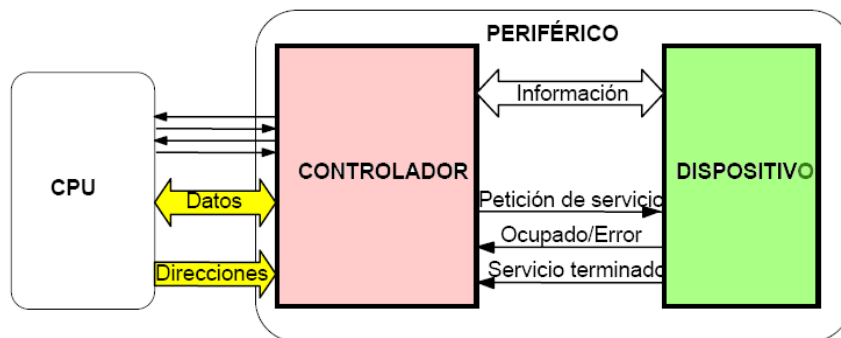
- ✓ **De entrada:** son periféricos que reciben información y la transmiten al ordenador para su procesamiento, por ejemplo: el ratón, el teclado, el escáner, etc.
- ✓ **De salida:** periféricos que presentan la información procesada por el ordenador, por ejemplo: un monitor normal, la impresora, el plóter⁸ (para impresión de planos y cartografía), etc.
- ✓ **De entrada y salida:** Aúnan ambas funciones, por ejemplo: el monitor táctil, el disco duro, unidad de lectura y grabación de DVD, etc.

El **sistema operativo** hace que los **dispositivos** se conecten al sistema y **realicen sus funciones** de forma adecuada y eficiente. El **sistema operativo abstrae de la complejidad y peculiaridad hardware de cada periférico para que las aplicaciones de usuario puedan hacer uso de los periféricos de una manera estandarizada y más sencilla**. El sistema operativo actúa pues como **intermediario** entre ellos, gracias a los **controladores de dispositivo**.

5.1. CONTROLADORES DE DISPOSITIVO.

¿Cómo pueden entenderse los programas de aplicación con los dispositivos periféricos? Hay multitud de tipos y fabricantes de periféricos, esto conlleva que tanto el sistema operativo como los fabricantes de periféricos deben **estandarizar el acceso a los dispositivos** utilizando lo que se denominan **controladores de dispositivos (device drivers)**.

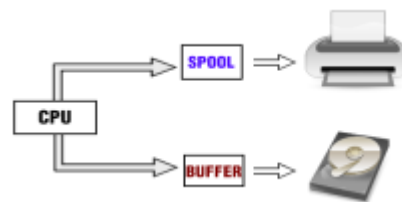
Un periférico siempre tiene dos partes: un **controlador**, se encarga de la comunicación con la CPU y un **dispositivo** mecánico, electromecánico o electromagnético. El controlador es un software, generalmente, suministrado por el fabricante del dispositivo o bien por el desarrollador del sistema operativo. De esta manera, estos controladores actúan como **interfaz entre los programas y el hardware**.



⁸ Es una impresora de grandes dimensiones para la impresión de planos, cartelería, cartografía, etc.

5.2. ESTRUCTURA DE DATOS DE LA E/S

Otro punto importante es la **estructura de datos** que utilizan los dispositivos periféricos para manejar la información y comunicación entre dispositivos o entre estos y la CPU. Las más utilizadas son los **spools** y los **buffers**.



- ✓ **Spools:** Los **datos de salida se almacenan** de forma temporal en una **cola** situada en un **dispositivo de almacenamiento** masivo (spool), hasta que el dispositivo periférico requerido se encuentre libre. De este modo se evita que un programa quede retenido porque el periférico no esté disponible. El sistema operativo dispone de llamadas para añadir y eliminar archivos del **spool**. Se utiliza en **dispositivos que no admiten intercalación**, como ocurre en la impresora, ya que no puede empezar con otro hasta que no ha terminado.
- ✓ **Buffers:** Es para **dispositivos que pueden atender peticiones de distintos orígenes**. En este caso, los datos no tienen que enviarse completos, pueden enviarse porciones que el **buffer** retiene de forma temporal. También se utilizan para acoplar velocidades de distintos dispositivos. Así, si un dispositivo lento va a recibir información más rápido de lo que puede atenderla se emplea un **buffer** para retener temporalmente la información hasta que el dispositivo pueda asimilarla. Esto ocurre entre una grabadora de DVD y el disco duro, ya que la primera funciona a una menor velocidad que el segundo.

5.3. TÉCNICAS DE LA E/S.

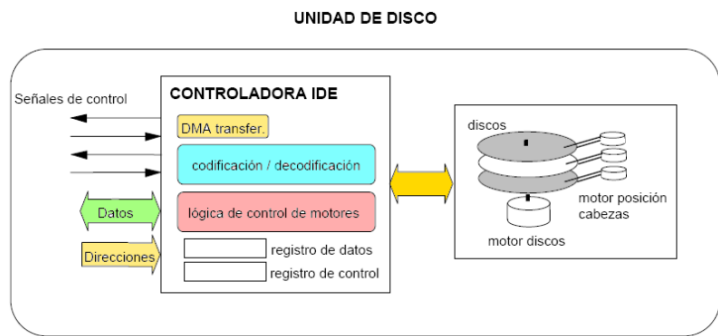
Vamos a conocer las distintas formas de funcionamiento de la E/S en los sistemas operativos según la intervención de la CPU. En estos momentos tenemos:

	Sin interrupciones	Con interrupciones
Transferencia de E/S a memoria a través de la CPU	E/S programada	E/S por interrupciones
Transferencia de E/S directa a memoria		Acceso directo a memoria (DMA)

- ✓ **E/S programada:** la CPU tiene todo el protagonismo ya que inicia y lleva a cabo la transferencia. Esta técnica **repercute en la velocidad de proceso del ordenador** porque la CPU debe dejar todo lo que está haciendo para ocuparse del proceso de entrada/salida.
- ✓ **E/S por interrupciones:** la CPU ejecuta la transferencia, pero el inicio es pedido por el periférico que indica así su disponibilidad. La CPU no pregunta a los dispositivos, sino que son estos los que la avisan cuando es necesario.
- ✓ **Acceso directo a memoria (DMA):** la transferencia es realizada por un controlador especializado. Esta técnica acelera enormemente el proceso de la E/S y libera a la CPU de trabajo. Lo habitual es que los datos que se quieren escribir en el dispositivo o que son leídos del dispositivo provengan o vayan a la memoria del ordenador, pues bien, en este caso la CPU inicia el proceso, pero luego este continúa sin necesitar a la CPU, con lo que **se acelera mucho el proceso de entrada/salida y se libera a la CPU del proceso**.

5.4. PLANIFICACIÓN DE DISCOS.

En la anterior unidad estudiamos la estructura y funcionamiento del disco duro, en este apartado conoceremos la **forma u orden** en que el **sistema operativo atiende las peticiones de lectura/escritura en disco**. Para ello utiliza lo que se conoce como **algoritmos de planificación del disco**. Antes de estudiarlas, fíjate en la figura sobre el **funcionamiento de la E/S en una unidad de disco**:



Los algoritmos de **planificación del disco** más importantes son:

Algoritmos de planificación de disco

FCFS (First Come, First Served)
 SSF (Shortest Seek First)
 Scan (Algoritmo del ascensor)
 C-Scan (Scan circular)

¿Cómo funcionan cada uno de estos algoritmos?

1. **Algoritmo FCFS (First Come, First Served):** 'Primero en llegar, primero en ser servido' es la planificación más sencilla, emplea una cola tipo **FIFO** (*First In, First Out* – **Primero en entrar, primero en salir**). Se da servicio a las solicitudes de acceso a disco de la cola según el orden de llegada de dichas solicitudes. Este algoritmo parece justo, pero en promedio, puede dar lugar a **tiempos bastante grandes**.
2. **Algoritmo SSF (Shortest Seek First)** La planificación **SSF** (*Shortest Seek First* – **Primero la búsqueda más cercana**) atiende primero la solicitud de la cola de solicitudes pendientes que quiere acceder al cilindro más cercano al de la solicitud actual, que se está procesando. Es decir, atiende primero la petición que requiere el menor movimiento de la cabeza de lectura/escritura desde su posición actual. Sin embargo, pueden llegar solicitudes que impliquen cilindros próximos al actual, por lo que estas solicitudes serán atendidas enseguida mientras que otras que llegaron antes, con cilindros más alejados, no se atenderán. Esto puede causar un **bloqueo indefinido**.
3. **Algoritmo Scan o algoritmo del ascensor:** La planificación *Scan*, también conocida como algoritmo del ascensor porque va dando servicio a las solicitudes que van encontrando en el sentido en el que se van desplazando las cabezas de lectura/escritura. Cuando no hay más solicitudes en ese sentido, o se llega al extremo, se invierte el sentido para hacer lo mismo otra vez pero yendo hacia el otro lado. Por tanto, en este algoritmo es necesario tener un bit que indique el sentido del movimiento. Este algoritmo evita el bloqueo indefinido que se puede producir con la planificación **SSF**. Este algoritmo es más **apropiado para sistemas que hacen gran uso del disco**.
4. **Algoritmo C-Scan o algoritmo Scan Circular:** En esta planificación la cabeza se mueve de un extremo del disco al otro atendiendo las solicitudes que va encontrando, pero al llegar al extremo opuesto, regresa de inmediato al otro sin servir ninguna solicitud; ya que en la zona próxima a dicho extremo habrá pocas solicitudes, puesto que las solicitudes para acceder a los cilindros de dicha zona acaban de ser atendidas.

6. GESTIÓN DE SISTEMA DE ARCHIVOS.

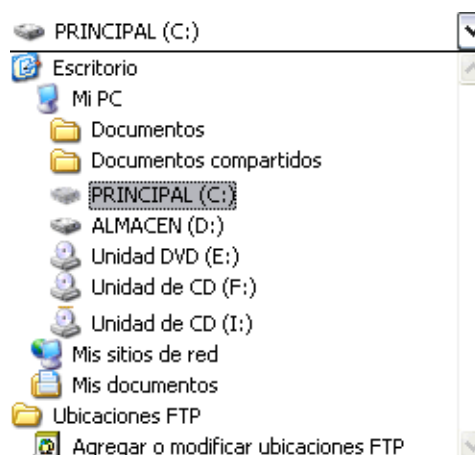
Esta parte del sistema operativo gestiona el servicio de almacenamiento, por lo que permite crear, modificar, borrar archivos y directorios y para ello utiliza generalmente una **estructura jerárquica**.

Cada sistema operativo utilizará su propio sistema de archivos, no obstante las operaciones que se pueden realizar sobre el sistema de archivos son bastante similares. Así, todos los sistemas de archivos actuales utilizan los **directorios** o **carpetas** para organizar a los archivos.

El **sistema de archivos** es el software que provee al sistema **operativo**, a los **programas de aplicación** y a **usuarios** de las funciones para operar con archivos y directorios almacenados en disco proporcionando mecanismos de protección y seguridad.

Los **objetivos más importantes** en la implementación de un **sistema de archivos** son:

- ✓ **Optimizar el rendimiento mediante un acceso rápido para recuperar la información contenida en archivos:** No se debe ralentizar el sistema en general por una deficiente gestión de los medios de almacenamiento, discos duros.
- ✓ **Fácil actualización:** Los cambios (añadir, borrar y modificar) no deben suponer una tarea complicada para el usuario y las aplicaciones.
- ✓ **Economía de almacenamiento:** Intentar que los archivos desperdicien la menor cantidad de espacio en disco posible. Es muy importante evitar la fragmentación de los discos.
- ✓ **Mantenimiento sencillo:** Evitar las operaciones complicadas a usuarios y programas, ocultando los detalles y proporcionando un acceso estandarizado a los archivos.
- ✓ **Fiabilidad para asegurar la confianza en los datos:** Deben proveer sistemas que aseguren que los datos escritos o leídos (entradas/salidas) sean correctos y fiables. También se debe minimizar o eliminar la posibilidad de pérdida o destrucción de datos.
- ✓ **Incorporar mecanismos de seguridad y permisos:** Esto es especialmente importante en sistemas de archivos de sistemas operativos multiusuario. Se debe poder proteger los archivos de un usuario del acceso de los demás usuarios. Por ejemplo, estableciendo permisos de escritura, lectura o ejecución.
- ✓ **Control de concurrencia:** Se debe controlar y asegurar el acceso correcto a los archivos por parte de varios usuarios a un tiempo, posiblemente bloqueando el archivo en uso hasta que termine la operación de modificación en curso.



6.1. ORGANIZACIÓN LÓGICA Y FÍSICA.

Se suele diferenciar entre la **organización de discos** a nivel **físico** (hardware) y **lógico** (software). El nivel físico de almacenamiento de datos en un disco duro consiste en el formateo en pistas, sectores, cilindros y platos. Pero esto es **muy dependiente del hardware** concreto que se esté usando y además **funciona a muy bajo nivel**.

Los **sistemas de archivos** deben proveer una capa de **abstracción** que **oculte los detalles puramente hardware** al usuario y le permita utilizar el medio de almacenamiento (disco) de una forma intuitiva y cómoda, por supuesto más cercana a los hábitos humanos de organización de la información. Éste es el **nivel lógico** del sistema de archivos y naturalmente en el que estamos más interesados.

A esto se le llama **organización del sistema de archivos** y suele coincidir en todos los sistemas de archivos actuales, utilizando el esquema de almacenamiento en **archivos** y la organización en **carpetas o directorios**.

¿En qué consisten los **archivos** y **carpetas**? y ¿cómo los **gestiona el sistema de archivos**?

Archivos: Es el elemento central de la mayoría de programas de aplicación. Los archivos o ficheros son estructuras de datos en disco donde se almacena la información y los programas de un ordenador. Pueden tener diversas estructuras y ello dependerá del sistema de archivos de nuestro sistema operativo y de la extensión del mismo.

Cada **archivo** de un sistema tendrá unas **características, o atributos**, que lo identifican y le sirven al sistema de archivos y al sistema operativo para manejarlo correctamente. Los **atributos** pueden variar de un sistema a otro, pero suelen coincidir al menos en los siguientes:

- ✓ **Nombre:** Identificador principal del archivo para el usuario. Cada sistema operativo establece las reglas para nombrar a los archivos, en cuanto a longitud y caracteres permitidos.
- ✓ **Extensión:** La extensión de un archivo son los caracteres que se colocan al final del nombre del un archivo para **especificar su tipo de contenido**. Por ejemplo, la extensión “.TXT” indica que el archivo es de texto o la extensión “.EXE” indica que el archivo es un programa ejecutable.
- ✓ **Permisos:** El sistema de archivos debe controlar qué usuarios están autorizados a utilizar cada archivo y qué operaciones pueden realizar. Por ejemplo, un archivo puede tener permiso de lectura y escritura para un usuario y en cambio otro usuario solo podrá utilizar el archivo en modo de lectura.
- ✓ **Creador:** Identificador del usuario que creó el archivo.
- ✓ **Propietario:** Identificador del usuario que es el propietario actual del archivo.
- ✓ **Fecha de creación:** Fecha y hora de la creación del archivo.
- ✓ **Fecha del último acceso:** Fecha y hora del último acceso al archivo.
- ✓ **Fecha de la última modificación:** Fecha y hora de la última modificación al archivo.
- ✓ **Tamaño actual:** Número de bytes que ocupa el archivo en el disco duro del ordenador.

Directorios: También denominados **carpetas**, son archivos especiales que cumplen la función de almacenar y organizar en su interior a archivos y otros subdirectorios. Son estos los que permiten mantener una cierta organización en el sistema de archivos. La organización en directorios mantiene forma de árbol invertido que comienza por un **directorio principal llamado raíz** y se va **ramificando** en otros directorios que pueden contener archivos y otros directorios. Respecto a los **atributos de un directorio**, como archivos que son coinciden con los atributos de estos.

6.2. OPERACIONES SOPORTADAS POR UN SISTEMA DE ARCHIVOS.

Las **operaciones básicas** sobre **archivos** que la mayoría de los sistemas de archivos soportan son:

- ✓ **Crear:** Los archivos se crean sin datos y después el usuario o alguna aplicación los van llenando.
- ✓ **Borrar:** Si un archivo ya no es necesario debe eliminarse para liberar espacio en disco.
- ✓ **Abrir:** Antes de utilizar el archivo se debe abrir para que el sistema conozca sus atributos, tales como el propietario, fecha de modificación, etc.
- ✓ **Cerrar:** Tras realizar las operaciones deseadas sobre el archivo, éste puede cerrarse para asegurar su integridad y liberar recursos de memoria que tuviera asignados.
- ✓ **Leer:** Los **datos se leen del archivo**; quien hace la llamada (programa) debe especificar la cantidad de datos necesarios y proporcionar un **buffer** para colocarlos.
- ✓ **Escribir:** Los **datos se escriben en el archivo**. El tamaño del archivo puede aumentar si se agregan datos nuevos o no si lo que se hace es actualizar los existentes.
- ✓ **Renombrar:** Permite **modificar el atributo nombre** de un archivo ya existente.

Los sistemas de archivos también suministran un conjunto de **operaciones para los directorios**, las más comunes son: **crear, borrar, abrir, cerrar, renombrar y leer**. Además, existen **otras dos operaciones sobre archivos y directorios** como son la de **crear un enlace y eliminarlo**. La operación de crear un enlace se utiliza para poder acceder a un archivo o directorio desde distintos puntos de la organización de directorios del sistema sin tener que duplicar o copiar el archivo o directorio en cuestión.

6.3. RUTAS DE ACCESO

Los **sistemas de archivos** necesitan una forma de determinar la **localización exacta de un archivo o directorio** en la estructura del árbol de directorios. La ruta de acceso a un archivo o directorio se indica nombrando todos los directorios y subdirectorios que tienen que atravesarse hasta llegar al elemento concreto. Dependiendo del sistema operativo con el que se trabaje cambiará la forma de establecer la ruta de acceso. Por ejemplo, en Windows se utiliza la barra invertida “\” para separar los directorios y en Linux se utiliza la barra “/”. Existen **dos tipos de rutas de acceso**:

- **Ruta de Acceso Absoluta:** Se comienza desde el **directorio raíz** y se va descendiendo en la estructura de directorios **hasta llegar al archivo o directorio buscado**. En las rutas de acceso absolutas se conoce la ubicación exacta. En sistemas Windows las rutas absolutas comienzan con una letra de una unidad seguida del carácter ':' (ejemplo: "D:\Documentos"). En sistemas Linux las rutas absolutas comienzan con la barra '/', que marca el inicio de la estructura de directorios (ejemplo: "/home/luisa/Escritorio/Documento.odt").
- **Ruta de Acceso Relativa:** Se utiliza junto con el concepto de **directorio de trabajo o directorio activo**, que es aquel donde estamos situados en un momento dado. Consiste en escribir la **ruta a partir del directorio activo**, esto se indica con '.' que hace referencia a la localización actual donde nos encontramos y '..' indica el directorio padre. También se puede escribir en primer lugar directamente el nombre de un directorio al que queremos subir sin usar el punto '.'. En las rutas de acceso relativas no se conoce la ubicación exacta del recurso al que se apunta ya que depende de la localización del directorio activo. En los siguientes ejemplos primero se baja uno o dos niveles usando ".." para luego subir a otros directorios.

7. MECANISMOS DE SEGURIDAD Y PROTECCIÓN.

El sistema operativo **debe protegerse activamente a sí mismo y a los usuarios** de acciones accidentales o malintencionadas. Cada vez es más necesaria la seguridad en los sistemas, ya que actualmente la mayoría de los ordenadores se encuentran conectados en red y el número de usuarios y recursos compartidos ha aumentado considerablemente.

Vamos a diferenciar entre **seguridad y protección**. Por **seguridad** nos referimos a una **política** donde se deciden **qué accesos están permitidos, qué usuarios pueden acceder**, en qué **forma** y a qué **recursos**. Por otro lado, la **protección** hace referencia al mecanismo que se utiliza para llevar a cabo la política de seguridad.

Los **requisitos** que debe cumplir un sistema operativo son:

- ✓ **Confidencialidad:** Los elementos del sistema sólo serán visibles por aquellos usuarios o grupos autorizados.
- ✓ **Integridad:** Los elementos del sistema sólo serán modificados por los usuarios o grupos autorizados.
- ✓ **Disponibilidad:** Los elementos del sistema sólo estarán disponibles para usuarios y grupos autorizados.

Los **elementos amenazados** son:

- ✓ Hardware.
- ✓ Software.
- ✓ Datos.
- ✓ Líneas de comunicación.

Pero, **¿cuáles son las posibles acciones accidentales o malintencionadas sobre los elementos amenazados?**

Elemento amenazado	Confidencialidad	Integridad	Disponibilidad
Hardware			Robo o sobrecarga de equipos, eliminando el servicio.
Software	Realización de copias no autorizadas del software.	Alteración de un programa en funcionamiento haciéndolo fallar durante la ejecución o haciéndolo que realice alguna tarea para la que no está programado.	Eliminación de programas, denegando el acceso a los usuarios.
Datos	Lecturas de datos no autorizadas. Revelación de datos ocultos de manera indirecta por análisis de datos estadísticos.	Modificación de archivos existentes o invención de nuevos.	Eliminación de archivos, denegando el acceso a los usuarios.
Líneas de comunicación	Lectura de mensajes. Observación de la muestra de tráfico de mensajes.	Mensajes modificados, retardados, reordenados o duplicados. Invención de mensajes falsos.	Destrucción o eliminación de mensajes. Las líneas de comunicación o redes no se encuentran disponibles.

Para hacer frente a estas acciones el **sistema operativo agrupa la seguridad** según tres **aspectos**:

- ✓ **Seguridad en el uso de recursos y servicios y control de acceso:** Utilizar un mecanismo de control de acceso a los recursos que tan sólo permita el acceso si existe el permiso correspondiente. Se establecerán **políticas de permisos** para acceder y operar con **recursos y servicios**.
- ✓ **Seguridad en el acceso al sistema:** Asegurar que sólo entran los usuarios autorizados. Para ello podrán utilizarse un **sistema de contraseñas eficaz** con niveles de acceso diferentes.
- ✓ **Seguridad en el uso de redes:** Evitar que se puedan producir escuchas y alteraciones en los datos que viajan por la red. Se aplicarán **técnicas de cifrado y descifrado de las comunicaciones** a través de la red.

8. DOCUMENTACIÓN Y BÚSQUEDA DE INFORMACIÓN TÉCNICA.

Todo **software** con una cierta complejidad suele venir acompañado de una **documentación**, ésta puede ser en formato digital o papel. Esta documentación toma forma en **manuales, tutoriales y demás guías de referencia** que sirven para mostrar al usuario **cómo se implanta y utiliza una aplicación**. A continuación, veremos los **tipos** de documentación nos podemos encontrar:

- ✓ **Manual de usuario** (con distintos niveles: básico, intermedio, avanzado): Explica en detalle la forma de operar con la aplicación. Las explicaciones de texto suelen venir acompañadas de capturas de pantalla para hacer que el seguimiento sea más fácil y captar la atención del lector.
- ✓ **Manual de Instalación y Configuración del programa:** Dedicado por lo general a la persona encargada de la puesta en funcionamiento del programa. Conlleva la explicación de los pasos de instalación, configuración, carga inicial de datos, si fuera necesaria, y demás pruebas de aceptación antes de que el programa pase a la fase de explotación (cuando comienza a ser utilizada por el usuario final). Este manual puede encontrarse incluido en el manual del administrador que veremos a continuación. En empresas donde se deben poner en marcha aplicaciones en red que requieren ciertos conocimientos técnicos en la configuración de aplicaciones, el perfil de la persona que implanta la aplicación y el del usuario final está claramente diferenciado. Sin embargo, en otras muchas ocasiones la persona que instala, configura y utiliza el programa suele ser la misma, sobretodo en aplicaciones de escritorio.
- ✓ **Manual del Administrador:** Documentación que va dirigida a la persona responsable del correcto funcionamiento, seguridad y rendimiento de la aplicación. Esta persona es, en muchos casos, la misma que instala y configura la aplicación.
- ✓ **Guía de referencia rápida:** Contiene las funciones básicas imprescindibles para instalar, con las opciones por defecto, y comenzar a utilizar una aplicación.

En ocasiones podemos encontrarnos con problemas o dudas técnicas sobre la instalación, configuración o utilización de un programa que no quedan claramente resueltas en la anterior documentación. En esas situaciones existe la posibilidad de buscar información adicional utilizando otros medios, como por ejemplo:

- ✓ Consulta al **soporte técnico** del desarrollador software, vía web, email o teléfono.
- ✓ Consulta en **foros de expertos**.
- ✓ Consulta en **bases de conocimiento**.
- ✓ Consulta en **FAQs** (Frequently Asked Questions – Preguntas Frecuentes).

9. ANEXO I: EVOLUCIÓN HISTÓRICA DE LOS S.O.

El hardware y el software de los sistemas informáticos han evolucionado de forma paralela y conjunta en las últimas décadas. Por lo que la evolución que vamos a ver de los sistemas operativos está estrechamente relacionada con los avances en la arquitectura de los ordenadores que se produjo de cada generación.

Primera generación (1945-1955)

Los primeros ordenadores estaban contruidos con tubos de vacío. En un principio no existían sistemas operativos, se programaba directamente sobre el hardware. Los programas estaban hechos directamente en código máquina y el control de las funciones básicas se realiza mediante paneles enchufables.

Hacia finales de 1950 aparecen las tarjetas perforadas que sustituyen los paneles enchufables. Las tarjetas perforadas supusieron un enorme paso ya que permitían codificar instrucciones de un programa y los datos en una cartulina con puntos que podía interpretar el ordenador. La mayoría de los programas usaban rutinas de E/S y un programa cargador (automatizaba la carga de programas ejecutables en la máquina). Esto constituía una forma rudimentaria de sistema operativo.

2ª Generación (1955-1965)

Esta generación se caracteriza por la aparición de los **transistores** que permitieron la construcción de ordenadores más pequeños y potentes. La programación se realizaba en lenguaje ensamblador y en FORTRAN sobre tarjetas perforadas. Otro aspecto importante de esta generación es el **procesamiento por lotes**, en el cual mientras el sistema operativo está ejecutando un proceso, este último dispone de todos los recursos hasta su finalización. La preparación de los trabajos se realiza a través de un lenguaje de control de trabajos conocido como JCL. El sistema operativo residía en memoria y tenía un programa de control que interpretaba las tarjetas de control, escritas JCL. Dependiendo del contenido de la tarjeta de control el sistema operativo realizaba una acción determinada. Este programa de control es un antecedente de los modernos intérpretes de órdenes.

Procesamiento Fuera de línea (Offline)

Como mejora del procesamiento por lotes surgió el **procesamiento fuera de línea (off-line)**, en el cual las operaciones de carga de datos y salida de resultados de un proceso podían realizarse de forma externa y sin afectar al tiempo que el procesador dedicaba a los procesos. A esto ayudó la aparición de las cintas magnéticas y las impresoras de líneas. Ejemplos de sistemas operativos de la época son **FMS** (Fortran Monitor System) y **IBSYS**.

3ª Generación (1965-1980)

La aparición de los **circuitos integrados (CI)** supuso una mejora consiguiendo un menor tamaño y relación precio/rendimiento respecto de las máquinas de generaciones anteriores. En relación con los sistemas operativos, la característica principal de esta generación fue el desarrollo de la **multiprogramación** y los **sistemas compartidos**. En los sistemas multiprogramados se cargan varios programas en memoria simultáneamente y se alterna su ejecución. Esto maximiza la utilización del procesador. Como evolución de aparecen los sistemas de tiempo compartido donde el tiempo del procesador se comparte entre programas de varios usuarios pudiendo ser programas interactivos. Algunos de los sistemas operativos de esta generación son **OS/360**, **CTSS**, **MULTICS** y **UNIX**.

TEMA 2: Software de un sistema informático.

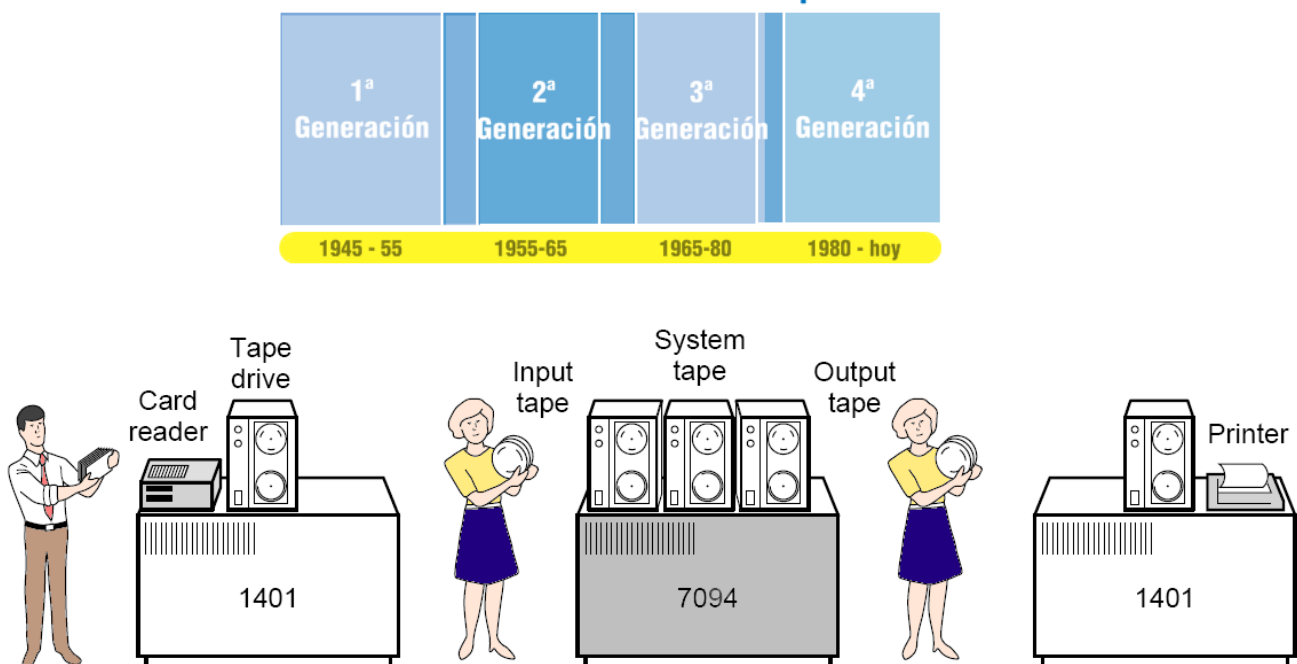
4ª Generación (1980-hasta hoy)

En esta generación se producen grandes avances en la industria hardware como la creación de los **circuitos LSI** (integrados a gran escala). También aparecen los ordenadores personales, entre finales de la anterior generación y principios de la presente. Ejemplos de sistemas operativos de los **primeros ordenadores personales** son **MS-DOS**, desarrollado por **Microsoft, Inc.**, para el IBM PC y **MacOS** de **Apple Computer, Inc.** Steve Jobs, cofundador de Apple, apostó por la **primera interfaz gráfica basada en ventanas, iconos, menús y ratón** a partir de una investigación realizada por Xerox. Siguiendo esta filosofía aparecería **MS Windows**. Durante los 90 apareció **Linux** a partir del núcleo desarrollado por Linus Torvalds. Los sistemas operativos evolucionan hacia sistemas interactivos con una interfaz cada vez más amigable al usuario. Los sistemas Windows han ido evolucionando, con diferentes versiones tanto para escritorio como para servidor (**Windows 3.x, 98, 2000, XP, Vista, 7, Windows Server 2003, 2008**, etc), al igual que lo han hecho Linux (con multitud de distribuciones, Ubuntu, Debian, RedHat, Mandrake, etc) y los sistemas Mac (**Mac OS 8, OS 9, OS X, Mac OS X 10.6 "Snow Leopard"**, entre otros).

Un avance importante fue el desarrollo de **redes de ordenadores** a mediados de los años 80 que ejecutan **sistemas operativos en red** y **sistemas operativos distribuidos**. En un sistema operativo en red los usuarios tienen conocimiento de la existencia de múltiples ordenadores y pueden acceder a máquinas remotas y copiar archivos de un ordenador a otro. En un sistema distribuido los usuarios no saben donde se están ejecutando sus programas o dónde están ubicados sus programas, ya que los recursos de procesamiento, memoria y datos están distribuidos entre los ordenadores de la red, pero todo esto es transparente al usuario.

Actualmente, existen **sistemas operativos integrados**, para una gran diversidad de dispositivos electrónicos, tales como, teléfonos móviles, PDAs (Personal Digital Assistant, Asistente Digital Personal u ordenador de bolsillo), otros dispositivos de comunicaciones e informática y electrodomésticos. Ejemplos de este tipo de sistemas operativos son **PalmOS, WindowsCE, Android OS**, etc. Haremos una referencia especial al último, Android OS, se trata de un sistema operativo basado en Linux. Fue diseñado en un principio para dispositivos móviles, tales como teléfonos inteligentes y tablets, pero actualmente se encuentra en desarrollo para su aplicación también en netbooks y PCs.

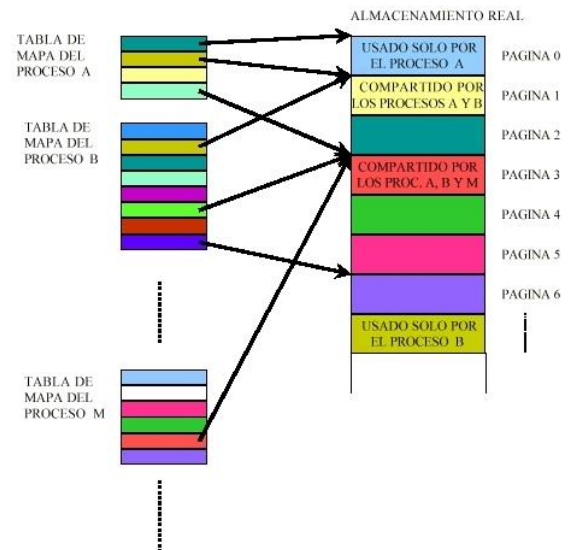
Evolución histórica de los Sistemas Operativos



10. ANEXO II. GESTIÓN DE MEMORIA: TÉCNICAS DE MEMORIA VIRTUAL.

Paginación pura:

La idea es la de **dividir la memoria principal en un conjunto de particiones conocidas como “marcos de página”** de igual tamaño. Cada **proceso se divide a su vez en una serie de partes llamadas “páginas”** del mismo tamaño que los marcos. El proceso se carga en memoria situando todas sus páginas en los marcos de página de la memoria, sin embargo, las páginas no tienen por qué estar contiguas en memoria. Como ventaja **reduce la fragmentación externa** de la memoria principal. Sin embargo, puede **aparecer cierta fragmentación interna**.



Segmentación pura:

Cada **proceso se divide en una serie de segmentos**. La peculiaridad de estos **segmentos es que su tamaño no tiene que ser el mismo y puede variar** hasta un límite máximo. Un proceso se carga situando todos sus segmentos en particiones dinámicas que no tienen que estar contiguas en memoria. Este sistema **reduce la fragmentación interna** de la memoria principal.

