

TEMA 2

INDICE

1.- Software de un sistema informático	3
1.1.- Requisitos e instalación: Determinación del equipo necesario.	3
Determinación del equipo necesario.....	3
1.2.- Requisitos e instalación: Ejecución del programa de instalación.....	4
Ejecución del programa de instalación.....	4
1.3.- Requisitos e instalación: Configuración de la aplicación.	5
Configuración de la aplicación.....	5
1.4.- Tipos de aplicaciones informáticas.	5
Aplicaciones de propósito general:	6
Otras aplicaciones de propósito general son:	6
Aplicaciones de propósito específico:	6
1.5.- Licencias software (I).	7
Software propietario	7
1.5.1.- Licencias software (II).	7
Software libre	7
Software de dominio público.....	8
Software con copyleft.....	8
1.5.2.- Ejemplos de software libre.....	8
2.- Sistemas Operativos.....	10
2.1.- Concepto y objetivos de los sistemas operativos	10
Evolución histórica de los sistemas operativos	11
Primera generación (1945-1955)	11
2ª Generación (1955-1965).....	11
3ª Generación (1965-1980).....	12
4ª Generación (1980-hasta hoy).....	12
2.2.- Tipos de sistemas operativos.....	13
Sistemas operativos por su estructura	13
Sistemas operativos por sus servicios	14
Sistemas operativos por su forma.....	14
2.3.- Servicios de los sistemas operativos	15
Núcleo	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.	20
4.2.2.- Asignación de particiones variables.	21
4.2.3.- Memoria virtual.....	22
Técnicas de memoria virtual.....	22
5.- Gestión de la entrada/salida	24
5.1.- Controladores de dispositivo.	24
5.2.- Estructura de datos de la E/S	24
5.3.- Técnicas de la E/S.....	25
5.4.- Planificación de discos.	25
6.- Gestión del sistema de archivos.	27
6.1.- Organización lógica y física.	27
6.2.- Operaciones soportadas por un sistema de archivos	28
6.3.- Rutas de acceso.....	29
7.- Mecanismos de seguridad y protección.	30
8.- Documentación y búsqueda de información técnica.	32

Software de un sistema informático.

Caso práctico

Ya conoces la empresa **BK Programación**, en la que nuestros protagonistas desarrollan su trabajo como profesionales del sector de la informática. Recordemos el perfil de cada uno.

Ada es la fundadora de la empresa y tiene una gran experiencia en programación. Junto a ella han empezado a trabajar en BK Programación dos técnicos superiores informáticos: **María**, dedicada a la implantación de servidores y manteniendo páginas web, y **Juan**, cuyo trabajo principal consiste en desarrollar aplicaciones web e instalar servicios de Internet.

Por otro lado, tenemos a **Ana** y **Carlos**, dos jóvenes que quieren mejorar su formación profesional en la rama de informática. **Ana**, a quien le apasiona el diseño gráfico, parece decantarse por estudiar el ciclo de Desarrollo de Aplicaciones Informáticas Multiplataforma, puesto que le vendría muy bien para mejorar a nivel profesional. **Carlos** está pensando estudiar el ciclo de Desarrollo de Aplicaciones Web, ya que se ajusta mejor a su perfil y tiene curiosidad por aprender a programar en entornos web.

Ana ha colaborado profesionalmente en varias ocasiones con BK Programación. Ha realizado retoque gráfico para algunos de los recursos que aparecen en los sitios web que ha creado y mantiene la empresa. Ana le ha comentado a Carlos que le gustaría realizar las prácticas de empresa en BK Programación, ya que podría aprender bastante y meterse más de lleno en el desarrollo de aplicaciones web. A Carlos le parece una buena idea y decide acompañar a Ana para ir a hablar con la responsable de la empresa, Ada.

1.- Software de un sistema informático.

Caso práctico

Ana y **Carlos** acaban de llegar a BK Programación y preguntan por **Ada**, ella los recibe en su oficina y se alegra de saludar de nuevo a Ana, colaboradora ocasional de BK Programación. Ana y Carlos le comentan su intención de estudiar ciclos formativos de informática y su interés por realizar, si fuera posible, las prácticas en la empresa. Ada encuentra interesante la posibilidad de incorporar personal con nuevas ideas, ilusión y con conocimientos actualizados. Cree que puede ser una buena experiencia en la que ambas partes, estudiantes y empresa, se benefician.

—Entonces, BK Programación se centra en la parte software de los sistemas informáticos, ¿o también realizáis tareas relacionadas con hardware? —pregunta **Carlos**, tras escuchar atentamente a Ada.

—En BK programación desarrollamos y mantenemos aplicaciones y servicios web. Sin embargo, en ocasiones, a petición de nuestros clientes y contando con trabajadores cualificados para ello, también instalamos y configuramos físicamente servidores, impartimos formación sobre las aplicaciones desarrolladas, actuamos como soporte técnico y asesoramos en la compra de equipos informáticos —comenta Ada.

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. Ejemplos de software son los sistemas operativos, paquetes ofimáticos, compresores, editores de imágenes y un sinfín de programas más o menos específicos según el conocimiento y ámbito profesional del usuario.

1.1.- Requisitos e instalación: Determinación del equipo necesario.

Caso práctico



En BK Programación el software que utilizan es el que los clientes demandan, aunque también hay empresas que piden asesoramiento sobre ello. Por ello, nuestros protagonistas tienen que conocer a fondo todos los **productos software** que el mercado ofrece en cuanto al área de las tecnologías web, sistemas operativos y demás aplicaciones de usuario relacionadas; ya que antes de poner en marcha una aplicación web, con frecuencia, hay que instalar y configurar los servidores que alojarán la aplicación y atenderán las peticiones.

María, encargada en muchas ocasiones de implantar los servicios web de los clientes, siempre se preocupa de conocer a fondo los requisitos y peculiaridades de instalación y configuración de las aplicaciones con las que trabaja, documentándose para ello.

¿Qué ocurre cuando **queremos instalar una aplicación** software en el ordenador? En todo **proceso de instalación** se han de seguir unos pasos que describiremos a continuación. 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.**

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 necesarios** 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 la aplicación 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, 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:** por ejemplo, en disco duro o unidad de almacenamiento externa para aplicaciones portables.
- ✓ **Tarjeta gráfica:** la memoria gráfica necesaria para el buen funcionamiento de la aplicación.
- ✓ **Resolución recomendada del monitor.**

Y de carácter **software**:

- ✓ **Plataforma software:** sistema operativo bajo el que funciona la aplicación, 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:

- ✓ **Equipo básico.**
- ✓ **Equipo opcional.**
- ✓ **Equipo en red.**

1.2.- Requisitos e instalación: Ejecución del programa de instalación.

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**
- ✓ **Instalación personalizada o avanzada**

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 al 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 DVDs o CDs. La aplicación se encuentra normalmente en formato comprimido. El traspaso de 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 ISO (por ejemplo: muchas distribuciones de Linux pueden descargarse en este formato).

1.3.- Requisitos e instalación: Configuración de la aplicación.

Configuración de 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 remoto 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 cookies 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.

1.4.- Tipos de aplicaciones informáticas.

Caso práctico

*Ana y Carlos acaban de terminar su reunión con Ada. De camino a casa, a Carlos le surgen algunas dudas sobre el **tipo de software** con el que trabajan en BK Programación. Ana, que tiene algunos conocimientos más sobre informática que Carlos, le comenta que existen varias clasificaciones de aplicaciones o programas informáticos. Una de las que conoce responde al ámbito en el que se utilizan esas aplicaciones.*

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

- ✓ **Aplicaciones de propósito general.**
- ✓ **Aplicaciones de propósito específico.**

Aplicaciones de propósito general:

Se emplean para el desempeño de funciones no específicas (informes, documentos, presentaciones, gráficos, hojas de cálculo, etc.). Se suelen comercializar en paquetes integrados denominados suites, tales como: Microsoft Office, OpenOffice, StarOffice, Lotus SmartSuite, etc. y se componen de:

- ✓ **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).

Otras aplicaciones de propósito general son:

- ✓ Herramientas para la comunicación: GroupWare o Trabajo en grupo como gestores de e-mail, servicio de mensajería instantánea, etc. Gestión de FAX.
- ✓ Utilidades y herramientas: como antivirus, navegadores web, gestores de archivos, compresores de archivos, visores de archivos.

Aplicaciones de propósito específico:

Por otro lado, las 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:** por ejemplo ContaPlus.
- ✓ **Entorno 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.
- ✓ **Herramientas "ad-hoc" especializadas:** OCR/OMR, monitores bursátiles, gestión empresarial ERP, 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.

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:

<http://portableapps.com/>

1.5.- Licencias software (I).

Caso práctico

***María**, como asesora en la implantación de aplicaciones software, prefiere la opción de **software libre**, siempre que sea posible y el cliente esté de acuerdo. Sin embargo, **Juan** parece decantarse por la alternativa de **software propietario** que, según su criterio, garantiza un soporte técnico más profesional y fiable. ¡Así que, como os podéis imaginar, el debate está a la orden del día!*

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 como acuerdos concretos se den entre el autor y el usuario. Distinguimos varios **tipos de software** o licencias en función de lo limitadas que estén las acciones del usuario sobre el mismo:

- ✓ **Software propietario.**
- ✓ **Software libre.**
- ✓ **Software semilibre.**
- ✓ **Software de dominio público.**
- ✓ **Software con copyleft.**

Software propietario

Se trata del software cuya redistribución o modificación están prohibidos o necesitan una autorización. Los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo (con o sin modificaciones), o su código fuente no está disponible, o el acceso a éste se encuentra restringido. Así, cuando el usuario adquiere una licencia software propietario lo que se le otorga es el **derecho de uso de la aplicación**.

En el **software propietario** o “no libre” una persona física o jurídica (compañía, corporación, fundación, etc.) posee los derechos de autor sobre un software, negando o no otorgando, al mismo tiempo, los derechos de usar el programa con cualquier propósito; de estudiar cómo funciona el programa y adaptarlo a las propias necesidades (donde el acceso al código fuente es una condición previa); de distribuir copias; o de mejorar el programa y hacer públicas las mejoras (para esto el acceso al código fuente es un requisito previo).

De esta manera, un software sigue siendo no libre aún si el código fuente es hecho público, cuando se mantiene la reserva de derechos sobre el uso, modificación o distribución (por ejemplo, el programa de licencias Shared source, de código abierto para uso académico de Microsoft).

1.5.1.- Licencias software (II).

Software libre

Proporciona al usuario las **cuatro libertades** siguientes, es decir, autoriza para:

- ✓ **Utilizar el programa**, para cualquier propósito.
- ✓ **Estudiar cómo funciona el programa** y adaptarlo a tus necesidades, debe proporcionarse las fuentes, directa o indirectamente, pero siempre de forma fácil y asequible.
- ✓ **Distribuir copias.**
- ✓ **Mejorar el programa y hacer públicas las mejoras 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 FSF. El **software libre suele estar disponible gratuitamente**, o al precio de costo de la distribución a través de otros medios; **sin embargo no es obligatorio que sea así**, por lo tanto **no hay que asociar software libre a "software gratuito"** (denominado usualmente freeware), ya que, conservando su carácter de libre, puede ser distribuido comercialmente.

Software de dominio público

Es aquél que no está protegido con copyright y que no requiere de licencia, pues sus derechos de explotación son para toda la humanidad. 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.

Software con copyleft

Es el software libre cuyos términos de distribución no permiten a los redistribuidores agregar ninguna restricción adicional cuando lo redistribuyen o modifican, o sea, la versión modificada debe ser también libre.

Existen otros tipos de software, tales como:

- ✓ **Freeware**: Programa totalmente gratuito. Es posible que requiera que nos registremos, pero siempre de forma gratuita.
- ✓ **Shareware (Demo)**: Se trata de una versión reducida del programa, con algunas funciones desactivadas para que podamos probarlo y decidir si lo vamos a comprar o no.
- ✓ **Shareware (Versión limitada por tiempo)**: Se trata de una versión totalmente funcional por un cierto número de días (normalmente 30, pero puede variar según la compañía) tras la cual no lo podremos usar o se verá reducida su funcionalidad. Su objetivo es poder probar la aplicación y luego decidir si la compraremos o no.

1.5.2- Ejemplos de software libre.

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.

Descubre más sobre la filosofía del software libre de una manera amena y visual visitando este mapa conceptual:

http://es.wikipedia.org/wiki/Archivo:Mapa_conceptual_del_software_libre.svg

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

http://noticias.juridicas.com/base_datos/Admin/rdleg1-1996.l1t7.html

2.- Sistemas Operativos.

Caso práctico

En **BK Programación**, los protagonistas de nuestra historia deben conocer a fondo todos los productos que el mercado ofrece en cuanto a software de sistema, y así aconsejar con garantías cuál se adecua mejor a las necesidades y situaciones que su clientela le pueda plantear.

Juan dice que deben conocer a fondo la teoría en la que están basados los desarrollos de los sistemas operativos. Conociendo como funcionan los sistemas “por dentro” podrán llegar a ser unos buenos profesionales con criterio propio, capaces de valorar las características de cada sistema. Y lo que es más importante, conociendo los entresijos de las máquinas y el software que van a utilizar podrán sacarles el máximo partido.

En **BK Programación** existen diferentes posturas como vinos en un apartado anterior. Esa es la razón de que no se cierran a ninguna opción, por ello, la empresa cuenta con personal especializado en sistemas **Linux** y para los sistemas **Windows** de **Microsoft** y **Mac** de **Apple**.

En **BK Programación** conocen la importancia de estar al día de los productos software del mercado y los nuevos que aparecerán. Para comprender y valorar lo particular de estos productos deben conocer antes lo general. Por lo que tanto **Juan** como **María** están al día de estos conocimientos técnicos para saber comparar las posibilidades que un sistema ofrece frente a otro.

En este apartado introduciremos gran parte de la teoría en la que están basados los sistemas operativos actuales. El **sistema operativo 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.- Concepto y objetivos 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:

Evolución histórica de los sistemas operativos.

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, éste 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.

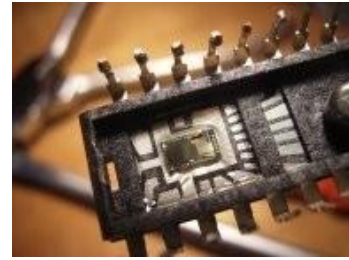


Procesamiento Fuera de línea (Offline)

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**.



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.

¿Quieres conocer cómo se crearon dos de las empresas más importantes de informática, Apple Computer y Microsoft? Visita este interesante enlace:

<http://www.venagraphica.es/blog/wp-content/uploads/file/losPrimerosPcs.pdf>

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.



2.2.- Tipos de sistemas operativos.

Ahora vamos a clasificar los sistemas operativos en base a su estructura, servicios que suministran y por su forma.

Tipos de sistemas operativos		
Por estructura	Por sus servicios	Por su forma
Monolíticos	Monousuario	Sistema operativo en red
Jerárquicos	Multiusuario	
Máquina Virtual	Monotarea	Sistema operativo distribuido
Microkernel o Cliente-Servidor	Multitarea	
Monolíticos	Monoprocesador	
	Multiprocesador	

Sistemas operativos por su estructura

Monolíticos: Es la estructura de los primeros sistemas operativos, consistía en un solo programa desarrollado con rutinas entrelazadas que podían llamarse entre sí. Por lo general, eran sistemas operativos hechos a medida, pero difíciles de mantener

Jerárquicos: 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, diferenciadas por funciones y con una interfaz clara para interoperar con los demás elementos. Un ejemplo de este tipo de sistemas operativos fue MULTICS.

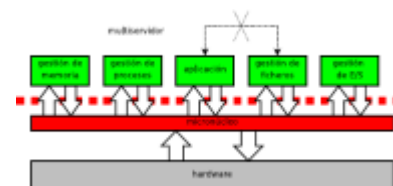
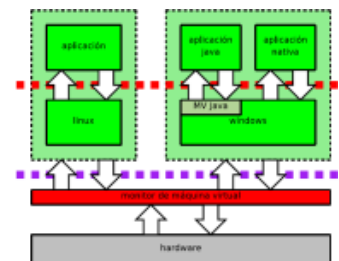
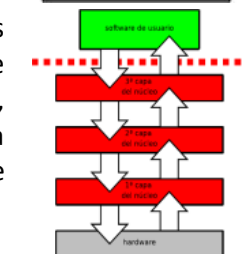
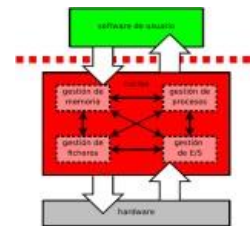
Máquina Virtual: El objetivo de los sistemas operativos es el de integrar distintos sistemas operativos dando la sensación de ser varias máquinas diferentes. Presentan una interfaz a cada proceso, mostrando una máquina que parece idéntica a la máquina real subyacente. Estas máquinas no son máquinas extendidas, 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.

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 MAC OS X o AIX.

Sistemas operativos por sus servicios

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 varios 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.



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.

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: Un sistema operativo multiprocesador se refiere al número de procesadores del sistema, éste es más de uno y el sistema operativo es capaz de utilizarlos todos para distribuir su carga de trabajo. 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).

Sistemas operativos por su forma

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.

Echa un vistazo a esta interesante comparativa entre Windows Vista y Ubuntu:

<http://www.cibm.es/ficheros/59b9b1ce1d6ea7c46f1d4f3c9f546a04.pdf>

2.3.- Servicios de los sistemas operativos.

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.

Núcleo

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



3.- Gestión de procesos.

Caso práctico

Ana y Carlos tienen dudas sobre lo que afecta a la eficiencia de un ordenador. María y Juan les explican que el rendimiento efectivo de un ordenador no sólo se basa en sus prestaciones hardware, sino también en el software que se ejecute y en el tipo de carga de trabajo que procese. La ejecución de los distintos programas que se lanzan en un equipo se traduce en multitud de procesos que requieren recursos. Uno de los principales recursos del ordenador es el **procesador**.

María y Juan le comentan a Ana y Carlos con más de detalle en qué consiste la **gestión de procesos** y que opciones existen.

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 en 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.**
- ✓ **En ejecución.**
- ✓ **Bloqueado.**



Los **procesos en estado listo** son los que pueden pasar a estado de ejecución si el planificador del sistema operativo los selecciona, esto es, cuando llegue su turno (según el orden de llegada o prioridad).

Los **procesos en estado de ejecución** son los que se están ejecutando en el procesador en un momento dado.

Los procesos que se encuentran en **estado bloqueado** están 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 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** con 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**.
- ✓ **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**:

- ✓ Todos los procesos en algún momento obtienen su **turno de ejecución o intervalos de tiempo** de ejecución hasta su terminación con éxito.
- ✓ El sistema debe **finalizar el mayor número de procesos** por unidad tiempo.
- ✓ **El usuario no percibirá tiempos de espera** demasiado largos.
- ✓ **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”.

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 los mostrados en la imagen:



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

<http://www.redes-linux.com/apuntes/so1/teoria/tema4.pdf>

4.- Gestión de memoria.

Caso práctico

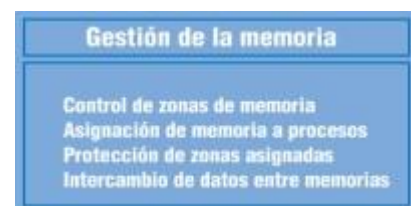
Ana y Carlos tras entender la importancia de una adecuada gestión de procesos, preguntan: —¿Qué ocurre con la memoria principal? ¿Cómo la utiliza el sistema operativo para poder ejecutar los procesos de manera equilibrada? —**Juan** les explica que existen diferentes posibilidades en la gestión de la memoria que hace el sistema operativo. Pero lo primero es tener claro ciertos conceptos sobre el uso de la memoria. Juan comienza su explicación...

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 lasoliciten**.

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 amalgamarse 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, vuelven 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.

Gestión de la memoria con memoria real y virtual				
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.

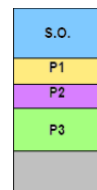
En sus orígenes los sistemas operativos no incluían ningún gestor de memoria, y el programador tenía un control completo sobre el espacio total de memoria. La memoria real se utiliza para almacenar el programa que se esté ejecutando en un momento dado. Conforme los procesos se ejecutan secuencialmente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



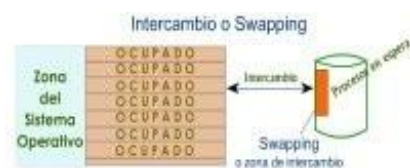
4.2.- Gestión de memoria en sistemas operativos multitarea.

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**

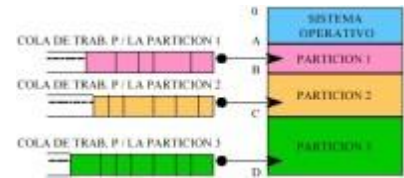
procesos. El número de particiones se mantiene fijo en el tiempo, así como el tamaño de cada una de las particiones.

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

Una cola por partición.

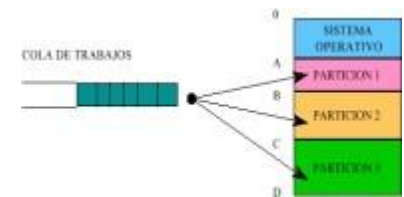
Se tiene **una cola por cada partición** y se coloca cada trabajo en la cola de la partición más pequeña en que quepa dicho trabajo, a fin de desperdiciar el menor espacio posible.

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.**

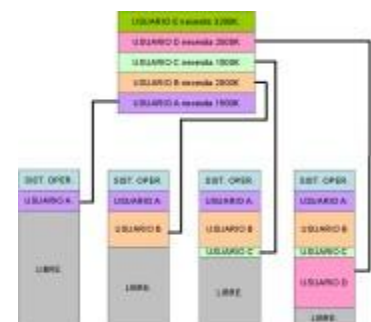
Además de esto, puede surgir el problema de la **fragmentación**, la cual se produce, cuando en la memoria hay áreas ocupadas intercaladas con áreas libres; es decir, cuando no hay una única área ocupada ni una única área libre.

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 cuales 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.

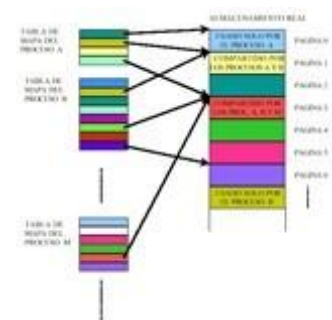
La **memoria virtual** da una solución a estos casos, ya que **permite dividir los procesos en varias partes y cargar sólo algunas de ellas en memoria**. La memoria virtual se basa en el uso de las técnicas de **paginación o segmentación**.

¿En qué consisten las técnicas de paginación y segmentación? Conoce su funcionamiento básico por medio del siguiente apartado:

Técnicas de memoria virtual.

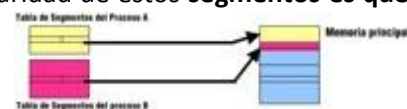
Paginación pura

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 porque estar contiguas en memoria. Como ventaja **reduce la fragmentación externa** de la memoria principal. Sin embargo, puede **aparecer cierta fragmentación interna**.



Segmentación pura

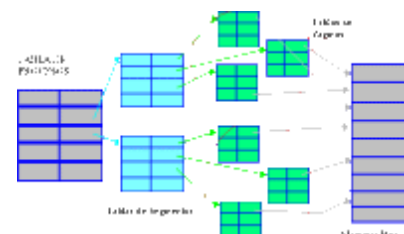
Cada **proceso se divide en una serie de segmentos**. La peculiaridad de estos **segmentos es que su tamaño no tiene que ser el mismo y puede variar hasta un límite máximo**. Un proceso se carga situando todos sus segmentos en particiones dinámicas que no tienen que estar contiguas en memoria. Este sistema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



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.

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:

<http://www.neo-tech.es/wp/doc/presentacion-erc.pdf>

5.- Gestión de la entrada/salida.

Caso práctico

Nuestros protagonistas utilizan a diario una gran variedad de periféricos (impresora, escáner, teclado, ratón, módem, monitor, dispositivos de almacenamiento externo, etc.). Hoy en día, estos dispositivos están preparados en su mayoría para enchufar y empezar a funcionar, algunos requieren la instalación de software específico para que el ordenador pueda trabajar con ellos adecuadamente.

Carlos, que es un gran aficionado a la edición de música y video digitales, no se había planteado hasta este momento cómo interactúan su tarjeta capturadora de vídeo o los altavoces, por ejemplo. Al hilo de este tema mantiene una interesante conversación con **Juan**, que le recuerda que existen diferentes tipos de periféricos, le explica para qué sirven los controladores, qué estructuras de datos utilizan los dispositivos para la transferencia y recepción de información, etc. En definitiva, Juan le descubre a Carlos otra tarea fundamental del sistema operativo, la **gestión de la E/S (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:

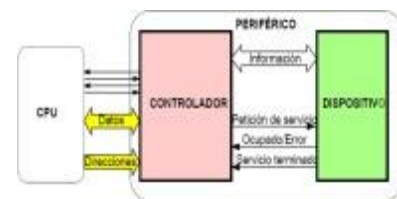
- ✓ **De entrada:** son periféricos que reciben información y la transmiten al ordenador para su procesamiento, por ejemplo: el ratón, el teclado, el escáner, etc.
- ✓ **De salida:** periféricos que presentan la información procesada por el ordenador, por ejemplo: la impresora, el plóter (para impresión de planos y cartografía), etc.
- ✓ **De entrada y salida:** Aúnan ambas funciones, por ejemplo: el monitor, 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.

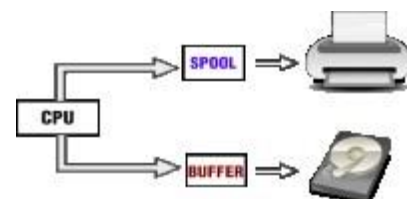
¿Cómo pueden entenderse los programas de aplicación con los dispositivos periféricos? Hay multitud de tipos y fabricantes de periféricos, esto conlleva que tanto el sistema operativo como los fabricantes de periféricos deben **estandarizar el acceso a los dispositivos** utilizando lo que se denominan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



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:

Técnicas para realizar la entrada/salida		
	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:

6.- Gestión del sistema de archivos.

Caso práctico

Carlos y Ana están aprendiendo de la mano de **María y Juan** las principales funciones de los sistemas operativos. Pero ahora se preguntan, ¿qué ocurre con los archivos y directorios? ¿qué información almacena el sistema operativo de estos? ¿qué operaciones pueden realizarse sobre los mismos? ¿existen distintas formas de referenciar un archivo o directorio? Estas y otras cuestiones encuentran respuesta con las explicaciones de **Juan y María**.

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** y **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.

Este es el **nivel lógico** del sistema de archivos y naturalmente en el que estamos más interesados.



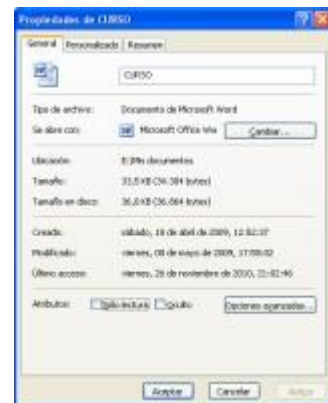
A esto se le llama **organización del sistema de archivos** y suele coincidir en todos los sistemas de archivos actuales, utilizando el esquema de almacenamiento en **archivos** y la organización en **carpetas o directorios**.

¿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:

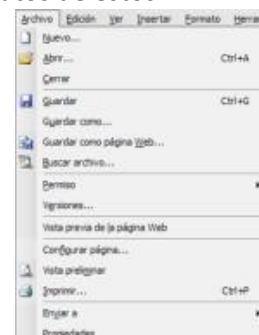
- ✓ **Nombre:** Identificador principal del archivo para el usuario. Cada sistema operativo establece las reglas para nombrar a los archivos, en cuanto a longitud y caracteres permitidos.
- ✓ **Extensión:** La extensión de un archivo son los caracteres que se colocan al final del nombre del un archivo 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 que 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 creo 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 e n 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 “\” 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.

Dirección	C:\Archivos de programa\OpenOffice.org 3\program\scalc.exe
-----------	--

- ✓ **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. En las rutas de acceso relativas no se conoce la ubicación exacta.

Dirección	..\oficina\albaran.jpg
-----------	------------------------

7.- Mecanismos de seguridad y protección.

Caso práctico

*Ada, como responsable de la empresa BK Programación, está muy concienciada de la relevancia de la seguridad en el ámbito informático. Por ello, los trabajadores de la empresa están muy sensibilizados con este tema. Ada decide que es hora de que los nuevos compañeros que acaban de llegar a la empresa reciban un breve seminario sobre **seguridad informática**, en este caso, centrándose en la relacionada con los sistemas operativos y otros elementos relacionados. **María** será la encargada de impartir dicho seminario.*

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 que **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?

Seguridad informática, detección de acciones accidentales o malintencionadas sobre activos informáticos.			
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.

Caso práctico

Cuando en **BK Programación** descubren una nueva aplicación interesante para su trabajo, no dudan en probarla para ver los resultados que aporta y decidir si pueden ofrecerla con garantías a sus clientes. **María** es bastante metódica a la hora de empezar a trabajar con un nuevo producto software, y siempre le gusta documentarse antes de poner en marcha y utilizar una aplicación. Para ello, echa mano de los **manuales** suministrados por el **fabricante de software**, **tutoriales** realizados por usuarios expertos y otros recursos. **Juan** es un experto en la búsqueda de este tipo de **recursos** y se conoce multitud de **fuentes**: foros de expertos, sitios web donde otros usuarios vuelcan problemas técnicos y posibles soluciones a estos. Incluso, ha creado un **repositorio online** de **manuales** con toda la documentación útil encontrada. **Ada** sabe que la gestión y asesoramiento software de BK Programación está en buenas manos.

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, sobretudo 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).
- ✓ etc.