



# 1. Software de un Sistema Informático.

Autor	ⓧ Xerach Casanova
Clase	Sistemas Informáticos
Fecha	@Dec 9, 2020 6:40 AM

## 1. Definición de Sistemas informáticos

## 2. Software

### 2.1. Requisitos e Instalación

### 2.2. Tipos de App. Informáticas

#### **Aplicaciones de propósito general**

#### **Aplicaciones de propósito específico**

### 2.3. Licencias de Software

#### **Software propietario o privativo**

Software Libre. Licencia GNU

Protección jurídica del software (Copyright)

Copyleft

## 3. Introducción a sistemas operativos

### 3.1. Definición, objetivos, kernel y subsistemas de los sistemas operativos.

Kernel o núcleo del S.O.

### 3.2. Tipos de sistemas operativos

Clasificación de sistemas operativos según su estructura

Clasificación de los sistemas operativos por los servicios que ofrecen

Clasificación de los sistemas operativos por su forma

## 4. Gestión de los procesos.

### 4.1. Estado de los procesos. Bloque de control de procesos

Definición de proceso

Estado de procesos

Transición entre estado de procesos

Bloque de control de procesos

### 4.2 Planificador de procesos. Algoritmos

Algoritmos de planificador de procesos

## 5. Gestión de memoria

### 5.1. Particiones fijas, variables y paginación

Primer sistema: gestión de memoria con particiones fijas

Segundo sistema: gestión de memoria con particiones variables

Tercer sistema: paginación

5.2. Memoria Virtual (Windows) - Swap Linux

Tendencia actual con memoria swap o memoria virtual.

6. Gestión de E/S

6.1. Estructura y transferencia de datos

Estructura de datos de la E/S

Transferencia de los datos de la E/S

7. Gestión de archivos

7.1. Organización lógica: directorios y ficheros

Organización lógica del sistema de archivos

Jerarquía de directorios. Árboles

Unidades físicas y lógicas

Mapa conceptual

# 1. Definición de Sistemas informáticos

Es el conjunto de una o más computadoras, software asociado, terminales, periféricos, componente humano, componentes físicos, medios de transmisión de la información, etc... que constituyen un todo autónomo capaz de tratar la información.

Se divide en tres apartados:

- **Hardware:** Son los componentes físicos, tangibles, tales como tarjetas gráficas, procesadores, memoria RAM, etc...
- **Software:** Son los procesos lógicos y aritméticos según se vayan programando.
- **Componente humano:** Quien crea los sistemas informáticos, los desarrolla e implementa y quien los utiliza.

## 2. Software

### 2.1. Requisitos e Instalación

El software está formado por los programas, las estructuras de datos y la documentación: Sistemas operativos, compresores, programas de diseño

gráfico, paquetes ofimáticos, etc...

Para la instalación de una aplicación se dan los siguientes pasos.

1. **Determinación del equipo necesario:** Una aplicación no puede instalarse en un sistema operativo o plataforma distinta para el que fue diseñada. Tampoco puede ser instalada si no cumple unos requisitos mínimos (hardware mínimo necesario o software previo). También se suele hablar de hardware recomendado, que es el hardware opcional o recomendables, normalmente superiores a los requisitos mínimos para que la aplicación funcione de manera óptima.
2. **Ejecución del programa de Instalación:** Normalmente el proceso de instalación se divide en dos partes: Instalación básica, la cual se hará en función de los elementos del propio equipo y de las especificaciones básicas del fabricante e instalación avanzada, que permitirá al usuario elegir el tipo de instalación del software según sus necesidades o de lo que vaya a utilizar.
3. **Configuración de la aplicación:** Una vez instalada la aplicación se debe configurar el entorno de trabajo y una serie de parámetros entre los que se encuentra la configuración de pantalla: ventanas, colores, tamaños de textos, etc... y la configuración de directorios de trabajo: archivos, proyectos, plantillas, etc.

## 2.2. Tipos de App. Informáticas

### Aplicaciones de propósito general

Son aplicaciones que solemos utilizar todos los usuarios, se suelen integrar en paquetes tipo Suite: Microsoft Office, etc...

- Editores de texto plano
- Procesadores de texto
- Hojas de cálculo:
- Comunicación: correo, agendas electrónicas, etc...
- Presentación: Power Point, etc...
- Herramientas de acceso y gestión de BBDD tales como Access
- Editores de XML y HTML

- Otras utilidades y herramientas como navegadores web, antivirus, compresores, lectores de PDF, etc...

## Aplicaciones de propósito específico

Aplicaciones para funciones específicas, técnicas o de gestión:

- Facturación, contabilidad, etc...: Factucont, Contaplus, etc...
- Desarrollo: Visual Studio, Java, etc...
- Herramientas de administración de BBDD tales como PhpMyAdmin, Oracle, etc...
- Herramientas Ad-Hoc especializadas: gestión empresarial, monitores bursátiles, etc...
- Herramientas de diseño y maquetación: Photoshop, Illustrator, Indesign, Corel Draw, etc...
- herramientas científicas y de ingeniería utilizadas para investigaciones en universidades, etc...

## 2.3. Licencias de Software

Se encargan de establecer un **contrato** entre el autor de una aplicación, sometido a propiedad intelectual y derechos de autor y el usuario final. Definen con precisión los derechos y deberes de ambas partes (**actos de explotación legales**).

El software puede ser privativo o libre

### Software propietario o privativo

La empresa que crea el software lo vende con límites sobre su uso, modificación y redistribución. Los tipos de licencia más utilizadas son:

- **CLUF** (contrato de licencia para usuario final), se limita su uso para un número limitado de ordenadores o nº de instalaciones.
- **OEM**, licencia para el equipo en el que viene instalado de fábrica, incluso si se estropea dicho equipo no podrá utilizarse en otro.
- **Licencia por volumen o corporativas**: para empresas que adquieren licencias para una cantidad grande de equipos.

También existen casos particulares de software propietario tales como:

- **Freeware:** Distribución libre del software para su uso, pero no se permite su modificación ni suele estar disponible el código fuente. Por ejemplo antivirus gratuitos para particulares pero no para empresas.
- **Shareware:** Límite de uso o funcionalidades. Distribución libre del software para su uso durante un periodo de prueba, pasado ese periodo debe adquirirse una copia legal o desinstalar el software. Por ejemplo WinRar.

## Software Libre. Licencia GNU

El software puede ser gratis y libre, pero libre no implica gratis. Esto quiere decir que se puede cobrar por el coste que conlleva la distribución del software libre, pero incluso se puede distribuir comercialmente, pues en el concepto de libre no habla nada de gratuidad.

El software libre lleva implícitas estas cuatro particularidades y si no se cumple alguna, se trata de software privativo:

- Uso libre para cualquier propósito
- Modificación del mismo para adaptarlo a tus propias necesidades
- Distribuir copias
- Añadir mejoras y hacerlas públicas.

Un tipo de licencia libre muy utilizado es el GNU-GPL, en la que cualquier modificación implica que se debe citar al autor original y redistribuir con licencia GNU-GPL, para evitar que nadie se apropie de software libre.

## Protección jurídica del software (Copyright)

El software no se puede patentar como el hardware, ya que se trata de algo intangible. El software se rige por la protección intelectual modificada. Una vez realizada la inscripción en el registro (el cual es público) se podrá anteponer el símbolo © en la obra añadiendo lugar y año de divulgación.

## Copyleft

El mundo libre tiene sus propios derechos. El copyleft se creó para asegurar que se cumplen las licencias GNU-GPL. El símbolo es una © invertida.

### 3. Sistemas Operativos

## 3. Introducción a sistemas operativos

### 3.1. Definición, objetivos, kernel y subsistemas de los sistemas operativos.

Un sistema operativo es un conjunto de programas que se encargan de gestionar los recursos de hardware y software del equipo, por lo que actúa como interfaz entre los programas de aplicación de usuario y el hardware puro.

#### Objetivos:

- Abstraer al usuario del hardware.
- Eficiencia: permite que los recursos del ordenador se utilicen de la manera más eficiente posible.
- Accede a dispositivos de E/S de manera que facilite al usuario utilizar los mismos de una manera sencilla.
- Estructura y ofrece un conjunto de operaciones para el sistema de archivos.
- Detecta y da respuesta a errores.
- Gestiona las redes y los usuarios para compartir recursos y datos.



#### Kernel o núcleo del S.O.

Se encarga de gestionar todas las funciones anteriores, representa una pequeña parte del S.O. pero es de las más que se utiliza. Reside en la memoria RAM continuamente mientras que otras partes del S.O. se cargan en la RAM solo cuando se necesitan.

Se encarga de controlar y administrar los servicios y peticiones de los subsistemas, que son:

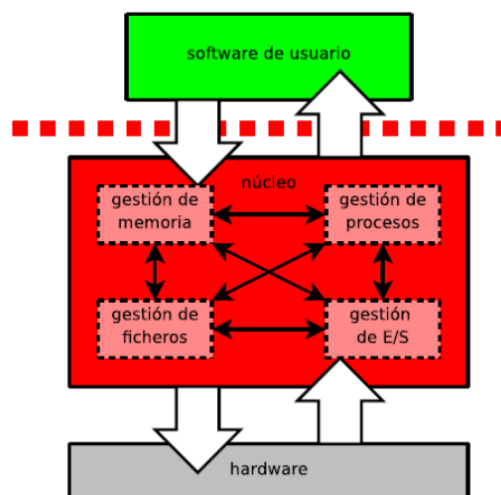
- Gestión de procesos
- Gestión de memoria
- Gestión de E/S
- Gestión del sistema de archivos



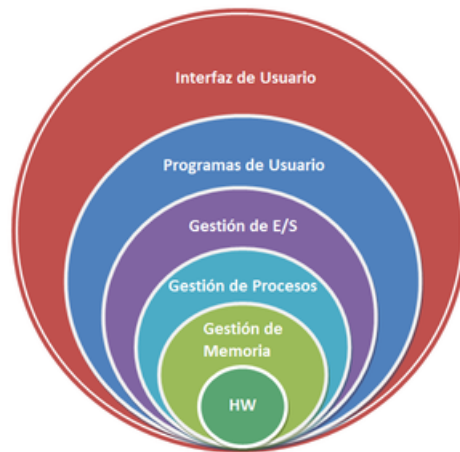
## 3.2. Tipos de sistemas operativos

### Clasificación de sistemas operativos según su estructura

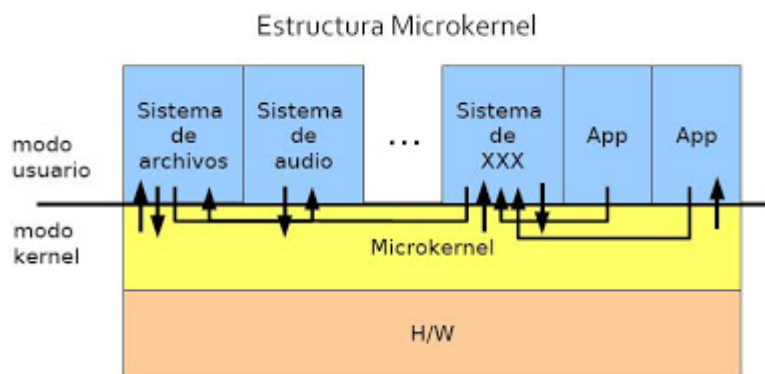
**Monolíticos:** Estructura de los primeros S.O. Se trata de un solo programa desarrollado con rutinas entrelazadas que se llaman entre sí. Ejemplos: MS-DOS, FreeBSD



**Jerárquicos o de capas:** Conforme las necesidades de los usuarios aumentaban, también lo hacía la complejidad y funciones de los S.O. Así que fue necesario realizar una mayor organización de ellos, dividiéndolos en partes más pequeñas y capas, de tal forma que la capa superior es la interfaz de usuario y la inferior es el hardware. Cada capa puede interactuar con su capa inmediatamente inferior. Ejemplos: MULTICS, VENUS, THE.



**Micronúcleo o Microkernel:** La idea consiste en tener un núcleo que brinde los servicios mínimos de manejo de procesos, memoria y que provea la comunicación entre procesos. Todos los restantes servicios se construyen como procesos separados del micronúcleo, que ejecutan en modo usuario. Estos sistemas tienen como ventaja un diseño simple y funcional, que aumenta la portabilidad y la escalabilidad. Para agregar un nuevo servicio no es necesario modificar el núcleo, y es más seguro ya que los servicios corren en modo usuario.



**Cliente / Servidor:** Basado en la estructura microkernel. En los sistemas operativos modernos, los sistemas cliente-servidor nacen con la finalidad de minimizar el núcleo (kernel), trasladando el código de todos sus servicios a las capas superiores; y el núcleo sólo deberá controlar la comunicación, que se realiza mediante mensajes, entre clientes y servidores o servidores y hardware. El objetivo es desarrollar la mayoría de las funciones del sistema operativo como procesos de usuario. Un proceso de usuario, llamado en este caso proceso cliente, envía una solicitud a un proceso servidor, que realiza el trabajo y devuelve la respuesta.



**Máquina virtual:** Integran distintos sistemas operativos en una sola máquina, dando la sensación de máquinas diferentes. En cada una de ellas, se puede ejecutar un sistema operativo distinto.

## Clasificación de los sistemas operativos por los servicios que ofrecen

### Por número de usuarios.

- **Monousuario:** solo un usuario utiliza el S.O. Ejemplo: MS-Dos
- **Multiusuario:** Trabajan varios usuarios en el mismo S.O. Bien en el mismo ordenador, o desde otro a través de consolas. Ejemplos: Unix, Windows 7, 8, 10, GNU Linux...

### Por número de procesos o tareas.

- **Monoproceso / monotarea:** Solo se puede ejecutar un proceso a la vez. Ejemplo: MS-Dos.
- **Multiproceso / multitarea:** Se ejecutan varios procesos a la vez, aunque la multitarea real no existe, simplemente se reparte el tiempo de manera que lo parezca. Ejemplo: Windows actuales, Linux, MacOS...

### Por número de procesadores.

- **Monoprocesador:** El S.O. solo se puede utilizar en ordenadores con un solo procesador. Ejemplo: MS-Dos
- **Multiprocesador:** El S.O. se utiliza en ordenadores con varios procesadores. Ejemplo: Windows actuales, Linux, MacOS...

## Clasificación de los sistemas operativos por su forma

**Sistemas operativos en red:** capaces de interactuar con sistemas operativos de otras máquinas a través de la red, para intercambiar archivos, etc. Ejemplo: Windows Server, Linux...

**Sistemas operativos distribuidos:** Las funciones se distribuyen entre distintos S.O. logrando integrar recursos (impresoras, unidades de respaldo, memorias, etc...) en una sola máquina virtual a la que accede el usuario. Ejemplo: MOSIX

# 4. Gestión de los procesos.

## 4.1. Estado de los procesos. Bloque de control de procesos

### Definición de proceso

Cuando se abre un programa, se ejecuta un proceso principal. Todos los programas ejecutan instrucciones de manera secuencial. Las instrucciones son subprocesos dentro del proceso principal, el cual dura hasta que se cierra el programa. Estos procesos finalizan de forma correcta o por error.

### Estado de procesos

- **En ejecución o activos:** es aquel que está ejecutándose, solo se puede ejecutar uno por procesador.
- **Preparado o en espera:** listos para ejecutar a la espera de que un procesador se libere.
- **Bloqueado o suspendido:** No se puede ejecutar porque le falta algún recurso. Cuando lo obtiene pasa a estar en espera.
- **Muerto:** cuando el proceso ha terminado o el S.O ha detectado un error fatal.

### Transición entre estado de procesos

El planificador de procesos tiene listas independientes para cada uno de los estados.

Cuando se crea un proceso se comprueba y si se puede ejecutar pasa a la lista de procesos en espera y cuando lo decida el planificador pasará a ejecución.

Una vez en ejecución el proceso puede pasar a cualquier otro estado:

- Puede pasar a preparado si el planificador decide ejecutar otro proceso.
- Puede pasar a bloqueado si el proceso necesita algún dato o está en conflicto con otro proceso. Cuando está listo para ejecutar pasa al listado de procesos preparados.

Todos los procesos pueden pasar a estar muertos.

### Bloque de control de procesos

Es donde el sistema tiene toda la información de los procesos, los cuales contienen:

- Identificador único de proceso (PID)
- Estado del proceso.
- Prioridad.
- Dirección de memoria donde está guardada la información relativa al proceso.
- Información contable para que el planificador realice su trabajo: Hora de inicio, tiempo de espera, tiempo de ejecución que resta....

## 4.2 Planificador de procesos. Algoritmos

### Algoritmos de planificador de procesos

El planificador de procesos es el que decide que proceso pasa a ejecución, cuanto tiempo se está ejecutando, puede decidir ponerlo en espera, para dar paso a otro proceso, etc... Es lo que se llama multiprogramación y aunque solo se está ejecutando un proceso, da la sensación de que hay varios procesos ejecutándose en paralelo.

El planificador de procesos utiliza algoritmos para decidir que proceso pasa a estar activo, su objetivo es acabar la ejecución de todos los procesos cuanto antes.

Los distintos algoritmos son:

- **FIFO (First In - First Out):** Cuando un proceso se ejecuta, lo hace hasta el final, esto significa que pueden bloquearse procesos muy cortos por estar ejecutándose uno muy grande. Una vez en ejecución, se ejecuta hasta el final.
- **SJF (Shortest job first):** Se ejecutan primero los procesos más cortos. Una vez en ejecución, se ejecuta hasta el final. Puede bloquear procesos largos por estar ejecutando procesos más cortos continuamente.
- **SRT (Shortest remaining time):** Se ejecutan primero los procesos más cortos, igual que en SJF, con la diferencia de que los procesos se pueden poner a la espera si llega otro que necesita menos tiempo para ejecutarse. Es algoritmo multiprogramación o expropiativo (se quita el procesador al proceso activo). Se utiliza mucho en ordenadores actuales, en conjunción con otros algoritmos.

- **Round Robin (prioridad circular):** Se establece un tiempo de ejecución para cada proceso, una vez finalizado ese tiempo el proceso en ejecución pasa a la cola de procesos preparados y entra el siguiente proceso en espera. La ventaja es que garantiza un tiempo de respuesta a todos los procesos, pero su desventaja es el tiempo que consume cambiando de procesos. Rendimiento similar a SRT.
- **Prioridad:** Todos los procesos tienen asignada una prioridad por el sistema (el usuario puede asignar y modificar prioridad). Los procesos se ejecutan en orden de la prioridad más alta. Para resolver el problema de que haya procesos que nunca se ejecutan por baja prioridad se cuenta con el algoritmo de prioridad por envejecimiento, que aumenta la prioridad a procesos que llevan más tiempo en espera.

Los algoritmos más utilizados en la actualidad es una combinación de RST y prioridad. FIFO y SJF se utilizan con criterios de desempate.

## 5. Gestión de memoria

### 5.1. Particiones fijas, variables y paginación

Memoria principal o física = Memoria RAM. Memoria secundaria = Disco duro.

Para que un proceso se pueda ejecutar debe estar cargado en memoria, si un proceso no tiene asignado el espacio de memoria que requiere no se puede ejecutar. Además, los S.O. actuales son multitarea y pueden ejecutar varios procesos a la vez, así que debe haber un mecanismo de gestión para distribuir la memoria entre todos los procesos que quieren ejecutarse.

Los primeros algoritmos tenían una gestión más sencilla pero desperdiciaban memoria. Los algoritmos modernos son más complejos pero a cambio son más eficientes.

#### Primer sistema: gestión de memoria con particiones fijas

Se divide la memoria física en varias particiones de un tamaño determinado cada una (cada una de ellas puede ser de distinto tamaño), pero de tamaño fijo. Cada partición se asigna a un solo proceso, pero se produce fragmentación interna y se desperdicia espacio, ya que dentro de cada partición puede sobrar espacio que no se puede utilizar.

## Segundo sistema: gestión de memoria con particiones variables

No existen particiones, a cada proceso se le asigna el tamaño que necesita y cuando termina se libera ese espacio y se junta con el espacio libre de al lado. Los procesos se cargan donde haya espacio y lo que quede libre se utiliza para otro. Se produce fragmentación externa (se desperdicia memoria de manera general), ya que al estar los espacios de memoria libre fragmentados puede haber procesos grandes que no caben en los huecos de memoria que hay.

## Tercer sistema: paginación

La memoria se divide en páginas de igual tamaño y los procesos van utilizando las páginas necesarias que haya libre, aunque no estén contiguas. Es el sistema utilizado en la actualidad y para poder utilizarlo se necesitan máquinas más potentes. El problema de la paginación es que dentro de la memoria poco a poco van quedando desorganizadas. El S.O. resuelve este problema realizando una compactación cada poco tiempo, juntando las páginas libres y las páginas de los procesos.

## 5.2. Memoria Virtual (Windows) - Swap Linux

### Memoria virtual Windows:

Consiste en utilizar memoria del disco duro como si fuera RAM. Suele tener entre 1 y 2 veces el tamaño de la memoria RAM y funciona guardando en la memoria virtual las mismas páginas que en la RAM, de tal forma que si esta última se llena, se pueden borrar para introducir nuevas. Si hacen falta antiguas, se provocará un fallo en la página y el gestor traerá dicha parte del proceso a la RAM. En Windows se utiliza el archivo `pagefile.sys` en la raíz de la partición donde está instalado. Utiliza la paginación dentro del propio archivo.

### Memoria Swap o área de intercambio GNU-Linux

El concepto es el mismo que la memoria virtual, pero en el caso de Linux, se crea una partición expresamente para este fin. La ventaja sobre Windows es que no depende de que la partición de datos de Windows esté muy llena, además que si la swap se encuentra en la primera partición, se gana velocidad, pues esa zona del disco duro es más rápida.

## Tendencia actual con memoria swap o memoria virtual.

Actualmente, al tener espacios de memoria RAM más elevados, se hace innecesario contar con elevada memoria virtual o Swap, se aconseja poner la misma cantidad de memoria virtual o swap que de RAM. En Linux la partición de swap se justifica porque se utiliza cuando hiberna el sistema.

## 6. Gestión de E/S

### 6.1. Estructura y transferencia de datos

Los múltiples tipos de periféricos de los distintos fabricantes estandarizan su acceso desde el S.O. gracias a los controladores de dispositivos (drivers). Estos actúan como interfaz entre los programas y el hardware.

#### Estructura de datos de la E/S

Para manejar la información y comunicación de periféricos con la CPU se utilizan sobretodo dos estructuras:

- **Spool.** Los datos se almacenan en una cola situada en un dispositivo de almacenamiento masivo hasta que el periférico queda libre. Así se evita que el programa se bloquee por no estar el periférico disponible. Se utiliza en dispositivos que no permiten intercalación (como las impresoras).
- **Buffer:** Para dispositivos que pueden atender peticiones de distintos orígenes. No tienen que enviarse datos completos, pueden enviarse porciones que el buffer retiene de manera temporal. También sirve para retener información que llega más rápido de lo que el dispositivo puede procesar (por ejemplo una grabadora de CD/DVD).

#### Transferencia de los datos de la E/S

- **E/S programada.** La CPU se encarga de iniciar y de llevar a cabo toda la transferencia, repercute en la velocidad ya que debe dejar todo lo que está haciendo para encargarse de la transferencia E/S.
- **E/S por interrupciones.** La CPU ejecuta la transferencia pero periférico es el que pide el inicio indicando su disponibilidad.
- **Acceso directo a memoria (DMA).** La transferencia es realizada por un controlador especializado, de esta forma la CPU inicia la transferencia pero es liberada de su ejecución, acelerando mucho el proceso de E/S.

# 7. Gestión de archivos

## 7.1. Organización lógica: directorios y ficheros

Cada S.O. utiliza su propio sistema de archivos, aunque actualmente todos lo hacen de manera muy similar. Todos los S.O. actuales utilizan directorios o carpetas para organizar archivos.

Los objetivos principales de un sistema de archivos son:

- Optimizar el rendimiento mediante un acceso rápido.
- Fácil actualización: Los cambios (añadir, borrar y modificar) no deben suponer una tarea complicada para usuario y aplicaciones.
- Economía de almacenamiento: desperdiciar la menor cantidad de espacio en disco posible. Evitar fragmentación.
- Mantenimiento sencillo: Evitar las operaciones complicadas ocultando los detalles.  
Fiabilidad para asegurar la confianza en los datos escritos o leídos (entradas/salidas).
- Incorporar mecanismos de seguridad y permisos.
- Control de concurrencia: Se debe controlar y asegurar el acceso correcto a los archivos por parte de varios usuarios a un tiempo.

### Organización lógica del sistema de archivos

Permiten una utilización del medio de almacenamiento de una forma intuitiva y sencilla para el usuario sin conocer los detalles físicos del hardware. En casi todos los S.O. se utiliza el esquema de almacenamiento en archivos y organización en carpetas o directorios.

Las características o atributos de los archivos son:

- Nombre: cada S.O. establece sus reglas en cuanto a longitud y caracteres permitidos.
- Extensión: especifica su tipo de contenido.
- Permisos: controla qué usuarios están autorizados a utilizar cada archivo y que operaciones pueden realizar.

- **Propietario:** Identificador del usuario que es el propietario actual del archivo, por defecto quien creó el archivo.
- **Fecha de creación.**
- **Fecha del último acceso.**
- **Fecha de la última modificación.**
- **Tamaño actual:** Número de bytes que ocupa.

Las operaciones básicas sobre archivos o carpetas son: crear, borrar, abrir, cerrar, leer, renombrar y crear enlaces (accesos directos).

## **Jerarquía de directorios. Árboles**

Todos los S.O. distribuyen sus archivos y carpetas en forma de árbol dentro de las unidades de almacenamiento.

## **Unidades físicas y lógicas**

Las unidades físicas son discos duros, CD-ROM, disquetes, etc. Sin embargo los S.O. pueden utilizar unidades lógicas, realizadas por software sobre una unidad de almacenamiento física. Estas particiones se pueden deshacer. En Windows se dividen en letras y cada una de ellas tiene su propio árbol. En Linux no se denominan por letras y aunque están divididas utilizan un solo árbol.

# **Mapa conceptual**



