



Tema 2: Conceptos Básicos

- **Introducción**
- **Estructura Lógica y Física de un Fichero**
 - **Diseño Lógico**
 - **Diseño Físico**
- **Tipos de Ficheros**
- **Objetivos y Factores en el Diseño de Ficheros**
- **Interacción con Ficheros**
- **Funciones de un Sistema Gestor de Ficheros**



Tema 2.1: Introducción. Enfoques.

Fichero:

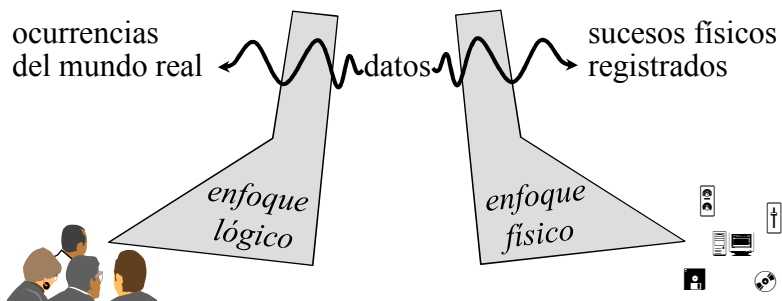
... conjunto organizado de informaciones almacenadas en un soporte común ...

¿qué son “informaciones”?

ocurrencias
del mundo real

datos

sucesos físicos
registrados





Tema 2.2: Estructura Lógica vs. Estructura Física

Estructura Lógica: organización de los datos según los usuarios

Estructura Física: organización de los datos según los soportes

Así se separan dos tipos de problemáticas:

- la organización tiene que ser realista, cubriendo las necesidades de los usuarios

eficacia

- la organización tiene que ser práctica, consumiendo el mínimo de recursos

eficiencia



Tema 2.2: Lógica vs. Física

Unidades mínimas de almacenamiento (sub-atómicas):

- desde el enfoque lógico: *información mínima interpretable*

elemento de datos (campo): unidad mínima de inf. interpretable

*Tipo de un campo: abstracción que describe sus características
(criterio para poder interpretarlo)*

Ejemplos: caracteres C(n), número entero E(n), booleano B, ...

- desde el enfoque físico: *información mínima registrable (accesible)*

en memoria intermedia: el carácter

en soporte: el registro físico o bloque



Tema 2.2: Lógica vs. Física

Contenido de un fichero: colección de informaciones (registros)

- desde el enfoque lógico, colección de informaciones

Registro Lógico: datos referidos a un elemento del fichero.
Es la unidad mínima de proceso de una aplicación
Tamaño de un registro: caracteres que ocupa

- desde el enfoque físico, caracteres almacenados en un soporte

Registro Físico: caracteres que almacenan y acceden juntos.
Es la unidad mínima de acceso del soporte.
Tamaño de un registro: caracteres que caben
Ese espacio sobre el soporte se denomina **bloque**



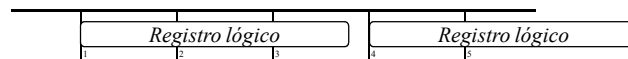
Tema 2.2: Lógica vs. Física

Correspondencia físico-lógica a nivel de registro:

consideración de tamaño: R_l y R_f no suelen coincidir en tamaño

• **Registro Expandido**

Cuando el registro lógico es mayor que el físico (ocupa varios)



• **Bloque (factor de bloqueo)**

Cuando el registro físico es mayor que el lógico (cabén varios)



Factor de Bloqueo: número de registros lógicos que caben en uno físico



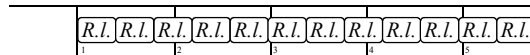
Tema 2.2: Lógica vs. Física

Correspondencia físico-lógica entre registros:

A este nivel, la organización de registros puede ser de dos tipos:

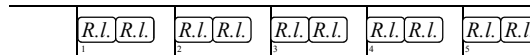
- Organización Consecutiva

Todo registro lógico comienza siempre a continuación del anterior



- Organización No Consecutiva

Todo registro físico comienza con el inicio de un registro lógico



* Sólo los ficheros seriales y secuenciales pueden tener regs. consecutivos.



Tema 2.2: Lógica vs. Física

Correspondencia físico-lógica a nivel de operación:

Debe existir una correspondencia (mapping) a nivel práctico

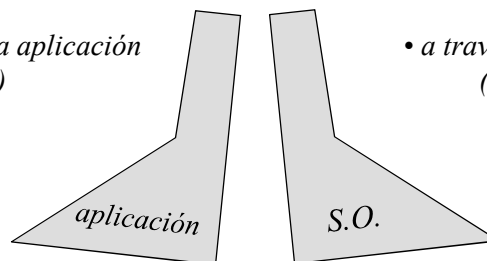
¿cómo se conectan las estructuras lógica y física?

- mediante la aplicación
(alto nivel)

- a través del S.O.
(bajo nivel)



usuarios



soportes



Tema 2.2: Lógica vs. Física

Diseño: *concepción original de un objeto u obra destinado a la producción en serie*

Diseño Físico:

“Determinación de la organización física de un fichero”

Organización física: disposición de los registros en el soporte, relativa a su implementación, orden, direccionamiento, apuntamientos, etc.

Diseño Lógico:

“Descripción de la estructura lógica de los registros de un fichero”

Estructura lógica (de un registro): descripción y disposición de los elementos de un registro, que en conjunto definen un individuo.

Diseño Físico del registro Lógico:

“Implementación de un registro lógico en secuencias de bytes”

Diseño Físico-Lógico (registro): descripción de las cadenas de *bytes* utilizadas para almacenar registros, y de los convenios necesarios para su interpretación.



Tema 2.2.1: Diseño Lógico:

Elementos de un Registro

- **Elemento de datos (campo):** unidad mínima e indivisible de datos que interviene en un proceso
Notación: *campo* tipo(tamaño)
(si es opcional, se encerrará entre corchetes [...])
- **Agregado de datos:** colección de elementos (campos y/o agregados)
 - **Vector:** agregado compuesto por un número fijo de elementos cuya interpretación es complementaria (entre todos los elementos definen un concepto).
Notación: $(elemento_1; elemento_2; \dots; elemento_n)$
 - **Grupo Repetitivo:** agregado compuesto por un número fijo o variable de elementos cuya interpretación es común (el mismo concepto para definir cada elemento)
Notación: $(elemento)^*$ para 0..N elementos
o bien $(elemento)^+$ para 1..N elementos



Tema 2.2.1: Diseño Lógico

Elementos de un Registro. Ejemplos:

DNI	N(8)
Nombre	C(20)
Fecha	(Día N(2); Mes N(2); Año N(4))
Hijos	(DNI N(8); Nombre C(20))*
Coche (opt.)	[Matrícula N(4); Letra C(3)]

Tipo de Elemento: abstracción de un campo o agregado que describe sus características. Consta de tipo de datos (Carácter/Número/Byte) y longitud.

Ocurrencia: valor concreto que toma un tipo (referido a un individuo)

Ejemplo: 03555555
Juan Valdés
12-05-1960
(13333333, José Valdés); (13333334, María Valdés)



Tema 2.2.1: Diseño Lógico

Objetivos del Diseño Lógico

- Fiel representación de la realidad
- Organización física transparente al usuario
- Facilitar a los usuarios la interacción con los datos
- Evitar redundancias lógicas:

“representar dos veces la misma información”

EFICACIA



Tema 2.2.1: Diseño Lógico

Ejemplo: fichero que registre datos de mis películas en DVD

- Necesitaré registrar su título (el más largo tiene 42 letras)
- El año de su realización y el nombre de la productora
- El género: *drama, comedia, suspense, acción, histórica, c.ficción*
- Quiero también registrar el nombre del director
- También de los actores y/o actrices (al menos uno, máximo cinco)
- Quiero saber quién ha hecho la banda sonora, y si la tengo necesitaré registrar en que formato(s): CD, CAS, LP

* 'nombre de persona' estará compuesta de nombre y apellido, de 20 car. cada uno



Tema 2.2.1: Diseño Lógico

Ejemplo: solución diseño lógico

películasDVD

Título	C(42)		
Año	N(4)		
Productora	C(20)		
Género	C(9)		
Director	(nombre C(20),	apellido	C(20))
Actores	(nombre C(20),	apellido	C(20)) ⁺⁵
Autor B.S.	(nombre C(20),	apellido	C(20))
Formato	C(3) ^{*3}		



Tema 2.2.2: Diseño Físico

Objetivos del Diseño Físico

Se buscará gastar menos recursos para el mismo resultado:

- Espacio (disminuir la ocupación y el tamaño total)
- Tiempo de Respuesta (minimizar los accesos)
- Mantenimiento (coste de reorganización)
- Desarrollo (coste de desarrollo y modificación)

EFICIENCIA



Tema 2.2.2: Diseño Físico

Diseño Físico: dependencia del soporte

- Tamaño de bloque
- Tiempo de Acceso (localización y acceso)
- Tasa de Transferencia (caracteres por unidad de tiempo)

Optimizaciones de bajo nivel:

- buscar soportes adecuados (tamaño bloque, tipo de acceso)
- minimizar el tiempo de localización (distribución adecuada)
- introducir redundancia física
 “grabar lo mismo varias veces”
- uso de memorias intermedias



Tema 2.3: Tipos de Fichero

Hay varias taxonomías, atendiendo a distintos criterios:

Atendiendo al tamaño de sus registros:

Longitud Fija: los registros fijos y de formato definido

Longitud Variable: registros variables y de formato definido

Longitud Indefinida: registros variables y sin formato definido

Tipos de Registro:

- Longitud fija: siempre se utiliza todo su tamaño
- Longitud variable: sus ocurrencias son de diversa longitud (su tamaño es el del mayor de sus ocurrencias)
- Longitud indefinida: se desconoce la longitud de sus ocurrencias



Tema 2.3: Tipos de Fichero

Ejemplo: *diseño físico-lógico de fichero fijo (corresp. al d. lógico en 2-12)*

películasDVD

Título	b(42)		
Año	b(4)		
Productora	b(20)		
Género	b(9)		
Director	(nombre b(20),	apellido b(20))	
Actor/iz 1	(nombre b(20),	apellido b(20))	
Actor/iz 2	(nombre b(20),	apellido b(20))	
Actor/iz 3	(nombre b(20),	apellido b(20))	
Actor/iz 4	(nombre b(20),	apellido b(20))	
Actor/iz 5	(nombre b(20),	apellido b(20))	
Autor B.S.	(nombre b(20),	apellido b(20))	
Formato1	b(3)		
Formato2	b(3)		
Formato3	b(3)		

/ Notar que se contempla cualquier registro, hasta el más grande posible */*



Tema 2.3: Tipos de Fichero

- Según el formato de sus registros
 - **homogéneos**: todos los registros siguen el mismo patrón
 - **heterogéneos**: contienen varios tipos de registro
 - En general, los ficheros heterogéneos son variables o indefinidos
 - La heterogeneidad puede afectar a todo el registro o sólo a una parte
 - Precisan un mecanismo de diferenciación del registro (no es razonable reservar espacio para cada tipo y sólo usar uno)
 - Distinguimos dos casos, ocurrencia única y ocurrencia múltiple
 - única: sólo aparece uno de los tipos (o ninguno); utilizaremos un campo *tipo* para saber cuál es en cada ocurrencia
 - múltiple: pueden aparecer varios; necesitamos un campo *mapa* (*)
- * el campo *mapa* es un binario de tamaño igual al n° de tipos distintos considerados (un bit por cada tipo, que adopta el valor 1 si ese tipo ocurre y 0 si no ocurre).



Tema 2.3: Tipos de Fichero

Ejemplo: fichero heterogéneo, con DVD películas y CD musicales

```
Tipo                b(1)
{
  /* película */
  Título            b(42)
  Año               b(4)
  Productora        b(20)
  Género            b(9)
  Director          (nombre b(20),  apellido b(20))
|
  /* musical */
  Título            b(20)
  Discográfica      b(20)
  Género            b(9)
  Autor            (nombre b(20),  apellido b(20))
}

/**/ ¿Cómo funciona el campo 'Tipo'? */
```



Tema 2.3: Otras taxonomías

- Atendiendo a su unidad básica de información:
 - binarios*: unidad básica, el bit
 - textuales*: unidad básica, el carácter
 - tipados*: unidad básica, el registro
- Atendiendo a la función del fichero
 - permanentes*: orientados al almacenamiento
 - temporales*: orientados al manejo
- Por la función de su contenido:
 - maestros* (de situación): información diversa (variada)
 - constantes* (de referencia): pocos valores y poco volátiles
- Por la oportunidad del contenido
 - borrador*: aún no entrado en uso
 - vigente*: en uso
 - histórico*: de uso pasado



Tema 2.4: Factores y Objetivos

Objetivos en el Diseño de Ficheros

- Tiempo de acceso
- Volumen y Ocupación

Factores que Influyen en la organización del fichero

- Actividad y tipo de los procesos
- Volatilidad
- Crecimiento



Tema 2.4.1: Tiempo de Acceso

Cálculo del tiempo de acceso a registro físico (bloque):

$$t_{\text{bloque}} = t_{\text{localización}} + t_{\text{transferencia}}$$

- El tiempo de localización será el número de localizaciones por el tiempo empleado en cada una
- El tiempo de transferencia será el número de bloques transferidos por el tiempo empleado leer un bloque

Cálculo del tiempo de acceso a registro aleatorio:

$$t_{\text{aleatorio}} = n^{\circ} \text{ accesos} \cdot t_{\text{bloque}}$$

- * ¿A cuál de todos los registros del fichero se refiere el acceso aleatorio?
En general, al peor caso o al caso medio (tiempos máximo y medio)



Tema 2.4.1: Coste Global (en Tiempo) de una Organización

Frecuencia Relativa de un Proceso:

Todo sistema de archivos estará sometido a un cjto. de procesos $P \equiv \{P_1..P_n\}$

Cada proceso tendrá una frecuencia f_i asociada, referida a...

- a una unidad de tiempo (por ejemplo, segundos u horas)
- frecuencia relativa: al conjunto P de todos los procesos, tal que $\sum_{i=1..n} f_i = 1$

Coste Global (referido a una organización del Sistema de Archivos):

- Una organización física del Sistema de Archivos (O_k) define todas las organizaciones base de los archivos que incluye, vínculos entre los mismos, y las organizaciones auxiliares con las que cuenta.
- Cada proceso P_i tendrá en O_k un coste C_i asociado (y expresado en nº accesos o en tiempo), que dependerá directamente de la organización.
- El coste global de la organización se define como (y puede venir referida a una unidad de tiempo)

$$C(O_k, P) = \sum_{i=1..n} C_i \cdot f_i$$



Tema 2.4.2: Volumen y Ocupación

Volumen y Ocupación de un fichero

- **volumen**: número de caracteres que contiene
- **ocupación útil**: caracteres útiles del fichero
(nº registros · tamaño medio de un registro)

volumen \geq ocupación útil

$$\text{densidad (\%)} = \frac{\text{ocupación útil}}{\text{volumen real}} \cdot 100 = \frac{\text{tamaño medio de un registro}}{\text{tamaño real de un registro}} \cdot 100$$

$$\text{mejora (\%)} = \frac{\text{densidad}_{\text{final}}}{\text{densidad}_{\text{inicial}}} \cdot 100$$



Tema 2.4.2: Factores y Dispositivos

Factores referidos a un dispositivo

Volumen y ocupación referidos a un fichero sobre un dispositivo cambian sensiblemente, porque hay que tener en cuenta el bloque.

volumen fichero = bloques ocupados · tamaño bloque

siendo $n^{\circ} \text{registros totales} \leq \text{bloques ocupados} \cdot \text{factor de bloque}$

$$\text{densidad (\%)} = \frac{\text{ocupación útil}}{\text{volumen fichero}} \cdot 100$$

soporte ideal: el tamaño del bloque coincide con el del registro, y el volumen real del fichero será $n^{\circ} \cdot \text{registros} \cdot \text{tamaño registro}$

* *esta consideración de bloque también debe aplicarse al resto de factores*



Tema 2.4.2: Factores y Dispositivos

Ejemplo: densidad del fichero de películas en DVD

- Título (el más largo tiene 42 letras, y de media 30)
 - El año de su realización siempre son 4 dígitos
 - La productora son 20 car. como máximo, y de media 9 car.
 - El género, 9 caracteres como máximo y de media se usan 8.1
 - Los nombres de personas usan, en media, 25 caracteres (máximo 40)
 - Se han llegado a registrar hasta 5 actores/actrices, y de media 2.8
 - 'Formato' siempre gasta 3 caracteres, pero sólo aparece la mitad de las veces
- ❖ ***Calcular la densidad ideal y la densidad de un fichero de 100.000 regs. en un soporte de bloque es 1Kb (comparar esta densidad en los casos de disponer registros consecutivos y de tenerlos no consecutivos).***



Tema 2.4.2: Optimización de Espacio

Campos de control: mejoran el manejo (ahorrando espacio)

Elemento de Datos

- Existencia:* en campos opcionales, indica si aparece o no
- Longitud:* en campos variables, indica el número de caracteres
- Reiteración:* en grupos repetitivos, indica el número de ocurrencias
- Fin de Campo:* en campos indefinidos, indican que acaba

Registro

- Fin (inicio) de Registro:* separa registros consecutivos en el soporte
- Tipo:* indica el tipo de registro a continuación (f. heterogéneos)
- Mapa:* indica los registros que se aplican (f. heterogéneos)



Tema 2.4.2: Optimización de Espacio

Ejemplo: (ver 2-26)

películasDVD

```
longitud_titulo b(1)
Título          b(longitud_titulo)
Año            b(4)
long_productora b(1)
Productora     b(long_productora)
Género         b(9)
Director       (long_nombre b(1), nombre b(long_nombre),
               long_apellido b(1), apellido b(long_apellido))
num_actores    b(1)
Actores        (long_nombre b(1), nombre b(long_nombre),
               long_apellido b(1), apellido b(long_apellido))num_actores
Autor B.S.     (long_nombre b(1), nombre b(long_nombre),
               long_apellido b(1), apellido b(long_apellido))
Existe_formato b(1)
Formato        b(3)existe_formato
```

* convenio aplicado: todas las cadenas de bytes codificadas en ASCII

Calcular la densidad ideal del diseño, y compararla con la obtenida en 2-26



Tema 2.4.2: Optimización de Espacio

Codificación de campos: codificar el contenido del campo

Algunos Tipos:

1. **Codificación numérica:** no almacenar dígitos, representar números

Ejemplo: DNI car(9) → byte(4)

2. **Enumerados:** equivalen a una codificación numérica natural (se puede establecer una codificación de ese tipo)

Ejemplo: color ∈ {blanco, amarillo, naranja, rojo, verde, azul, negro}
color car(8) → byte(1)

- **Pseudoenumerados:** incluyen un valor 'otros', cuya ocurrencia implica la aparición explícita del valor a continuación en el registro.

Se puede representar de siguiente modo:

```
color          byte(1)
[color_raro    byte(8)]color='otros'
```

3. **Agrupación de varios campos:** por ejemplo, booleanos en un mapa



Tema 2.4.2: Optimización de Espacio

Observaciones sobre la Codificación:

- Es un cambio de convenio en el almacenamiento
Ejemplo: en lugar de almacenar dígitos decimales en bytes ASCII, se puede decidir almacenar dígitos binarios en signo/magnitud.
- Si se hace, debe ser completamente transparente al usuario.
 - Los que necesitan conocer y compartir el convenio son la aplicación que escribe los registros, y la aplicación que los lee.
 - La que los muestra, lo hará del mejor modo para el usuario.

Efectos de la codificación: ejemplo, DNI car(9)

- Antes de ser codificado: ocupa **9** car. útiles / su tamaño es de **9** car. reales;
- Después de codificarlo: ocupa **4** car. útiles / su tamaño es de **4** car. reales

***“ La codificación disminuye el tamaño,
pero no disminuye significativamente la densidad ”***



Tema 2.4.2: Optimización de Espacio

- **Codificación numérica:**
- *para un rango de números naturales [0..m]*
¿cuántos caracteres son necesarios para codificar el número?
 - se halla la longitud (nº de dígitos) del número codificado en binario
$$l = \log_2(m),,$$
 donde m es el máximo del dominio
 - se halla el número de caracteres $c = l / 8$ (1 caracter = 8 bits)
- *para un rango de números enteros [n..m] (m positivo)*
representación en exceso a n: para almacenar x, tomamos $x-n$
el nuevo rango es $[0..(m-n)]$ y procedemos como en el 1º caso
- *para un rango de números enteros negativo [n..m] (n y m negativos)*
dado que el signo es un factor k constante, no se almacena (tomamos x/k)
el nuevo rango es positivo, y se procede como en los casos anteriores



Tema 2.4.2: Optimización de Espacio

Ejemplo: películasDVD

		<i>bytes necesarios</i>
longitud_titulo	b(1)	→ b(1)
Título	b(longitud_titulo)	→ media
Año	b(2)	→ b(2)
long_productora	b(1)	→ b(1)
Productora	b(longitud_productora)	→ media
Género	b(1)	→ b(1) (enum)
Director	(l_nom b(1), nombre b(1_nom), l_apel b(1), apellido b(1_apel))	→ media + 2
número_actores	b(1)	→ b(1)
Actores	(l_nom b(1), nombre b(1_nom), l_apel b(1), apellido b(1_apel))	→ n · (m + 2)
Autor B.S.	(l_nom b(1), nombre b(1_nom), l_apel b(1), apellido b(1_apel))	→ media + 2
Formato	b(1)	→ b(1) (enum)



Tema 2.4.3: Actividad y Volatilidad

Tasa de Actividad: porcentaje de reg. Tratados en un proceso

$$T_A = \frac{\text{registros procesados}}{\text{registros totales}} \cdot 100$$

respecto de un proceso

Volatilidad: porcentaje de cambios por unidad de tiempo

(según cada operación, consideraremos tasas de inserción, borrado, y modificación)

$$T_I = \frac{\text{registros insertados}}{\text{registros totales}} \cdot 100$$

$$T_B = \frac{\text{registros borrados}}{\text{registros totales}} \cdot 100$$

$$T_M = \frac{\text{registros modificados}}{\text{registros totales}} \cdot 100$$

referido a unidad de tiempo



Tema 2.4.3: Crecimiento

Tasa de crecimiento: indica en qué proporción aumentan los registros del fichero por unidad de tiempo

$$T_C = T_I - T_B \quad \leftarrow \text{magnitud l\u00f3gica}$$

- la *tasa de crecimiento esperada* permite reservar espacio vac\u00edo (para permitir la llegada de nueva informaci\u00f3n sin reorganizar)
- el espacio f\u00edsico necesario para ese crecimiento depender\u00e1 de la organizaci\u00f3n del fichero: en un fichero serial fijo con reutilizaci\u00f3n de huecos coincidir\u00eda con la tasa de crecimiento, pero en el mismo fichero sin reutilizar huecos coincidir\u00eda con la tasa de inserci\u00f3n.



Tema 2.5: Interacci\u00f3n con Ficheros

Interacci\u00f3n con ficheros:

Operaciones sobre la totalidad:

- Creaci\u00f3n / Destrucci\u00f3n
- Apertura / Cierre
- Consulta de todo el fichero

Operaciones sobre algunos registros

- *consulta selectiva*
- actualizaci\u00f3n: inserci\u00f3n / *modificaci\u00f3n* / borrado

Algunas operaciones (*selectivas*) requieren localizaci\u00f3n



Tema 2.5: Interacción con Ficheros

Clave: campo (o conjunto de campos) que facilitan la interacción de los usuarios con el fichero.
(posibilitan la identificación y/o localización de registros)

Tipos de clave:

- **Clave de Identificación:**
identifica unívocamente un registro en el fichero
(toma valores diferentes para cada registro)
- **Clave de Direccionamiento:** *(no necesariamente identificativa)*
localiza la posición del registro
- **Clave de Ordenación:** *(no necesariamente identificativa)*
criterio de ordenación; puede ser físico (fichero) o lógico (proceso)
- **Clave de Búsqueda:**
información utilizada por los usuarios para localizar registros



Tema 2.5: Interacción con Ficheros

Ejemplo: claves de un fichero

- Clave de identificación: Título
(el título es único, me sirve para identificar un DVD)
- Clave de direccionamiento: Productora
(tengo una estantería por productora, y allí se localizan esos DVD)
- Clave de Ordenación: Año, Título
(los tengo ordenados por año, y luego alfabéticamente según el título)
- Clave de búsqueda: Año, Director



Tema 2.5: Interacción con Ficheros

Cualquier clave debe ser mínima:

“No existe ningún subconjunto de la clave que cumpla las mismas funciones que la totalidad (de dicha clave)”

Ejemplos: claves incorrectas o inadecuadas

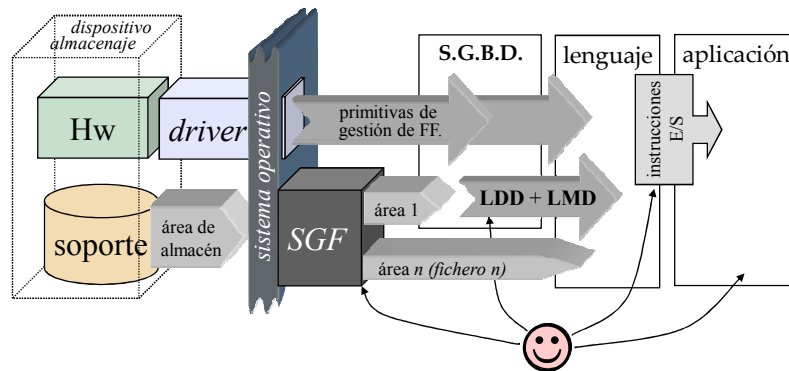
- Clave de identificación: Año, Productora
(probablemente tenga dos películas que coincidan en productora y año)
- Clave de direccionamiento: Formato
(pocos valores posibles, poca capacidad de direccionamiento)
- Clave de Ordenación: Título, Año
(si está ordenado por título, que es único, no tiene sentido el año)
- Clave de Búsqueda: Título, Director
(no es mínima, porque el título por sí solo identifica un solo registro)



Tema 2.6: Gestión de Ficheros

¿Quién gestiona los ficheros sobre el soporte?

- ☞ **El Sistema Gestor de Ficheros (SGF) del Sistema Operativo**
El SGF gestiona áreas individuales de almacenamiento
(identificación + localización + atributos)





Tema 2.6: Gestión de Ficheros

¿Cuáles son las funciones de un SGF? (del S.O. o genérico)

Sus funciones dependen del manejador (del área de almacenamiento).
Es decir, del tipo de soporte, del Sistema Operativo...

De un modo generalizado, se agrupan como sigue:

- **de todo el fichero / área (iniciales)** { CREAR
ASIGNAR
ABRIR / CERRAR . . .
- **de acceso (de un elemento)** { LEER
ESCRIBIR
- **de desplazamiento / localización**
- **otras funciones relacionadas** { • manejo de memorias intermedias
• seguridad (protección y confidencialidad)
• mantenimiento y administración (afinamiento)
• otras propias del soporte



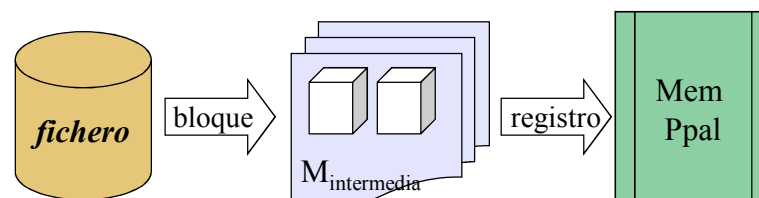
Tema 2.6.1: Memorias Intermedias

lectura: anticipar próximas lecturas (para que estén en Mppal)

- recogiendo todo el bloque leído
- realizando más lecturas en el tiempo de proceso

escritura: demorar escrituras (proceso perezoso)

- varios registros escritos en el mismo bloque (ins/mod)
- evitando operaciones que se corrigen o deshacen





Tema 2.6.1: Mem_{intermedia} Paginada

páginas: agrupación de información en memoria intermedia.

Son como *ventanas* sobre el fichero, gestionadas por el S.O.

- Tamaño de página: depende de soporte y S.O. (suele coincidir con bloque)
- Mecanismo de acceso paginado:
 - al acceder, busca la página y si no está la trae
 - si no hay sitio libre, se hace sitio según su política
 - cuando está ocioso, trae y libera bloques (según el S.O.)
- Políticas de liberación de páginas:
 - LRU/MRU: la que lleva más/menos tiempo sin usarse
 - la de menor tiempo de localización (más cercana)
- Proximidad de registros:
registros más probables en siguiente acceso deben estar en la misma página