



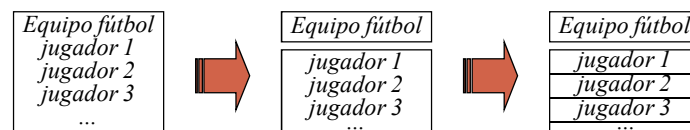
## Tema 3: Asociación de Archivos

- **Introducción**
- **Punteros**
  - Tipos de Puntero
  - Directorios
  - Uso de los Punteros
- **Esquemas de Archivos**
- **El Modelo Jerárquico**
  - Propiedades
  - Notación
  - Modelización
  - Restricciones



### Tema 3.1: Introducción

- La información se separa por cuestiones de economía, consistencia y facilidad de manejo:



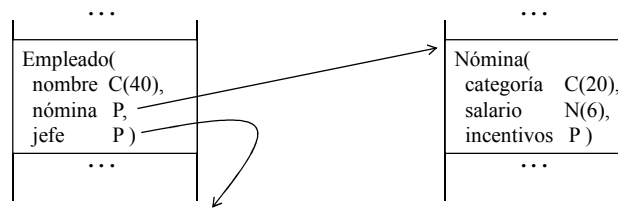
- Las dependencias que existen entre registros son información.
- Esta información se almacena mediante punteros
- Las asociaciones entre ficheros pueden dibujarse mediante un diagrama denominado *Esquema de Archivos*
- El diseño de estos esquemas de archivos recibe el nombre de *Modelización Jerárquica (Modelización de Sistemas Jerárquicos)*



## Tema 3.2: Concepto de Puntero

**“Elemento de datos que permite la localización de un registro”**

- \* Contiene información (de control) para enlazar otra información (registros o fragmentos de un registro, por ejemplo agregados)
- \* Observar que el campo de tipo puntero lleva una semántica asociada, exactamente del mismo modo que cualquier otro tipo de campo.



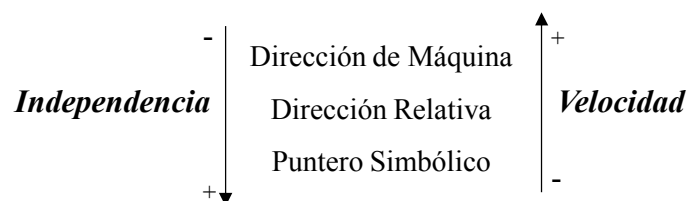
**Restricción Referencial: todo registro apuntado debe existir**



### Tema 3.2.1: Tipos de Puntero

***Tipos de Puntero (según su dominio):***

- **Dirección de Máquina:** la dirección física del registro
- **Dirección Relativa:** del registro en el espacio de dircmto. del fichero
- **Puntero Simbólico:** identificación lógica del registro (identificador)





## Tema 3.2.1: Tipos de Puntero

### ***Tipos de Puntero (según su localización):***

- ***Punteros empotrados***: el puntero se almacena en el registro (campo)  
Para rastrear, requieren procesar todo el registro (menor eficiencia)
- ***Punteros de Directorio***: se almacenan por separado (fichero auxiliar)
  - Son como guías que describen las relaciones entre punteros (y, por tanto, también entre registros), facilitando su localización y manejo. Cada relación registrada (línea) se denomina *entrada del directorio*
  - No siempre encajan con la organización básica de los ficheros.
  - Las entradas de un directorio pueden seguir distintas estructuras



## Tema 3.2.2: Punteros y Directorios

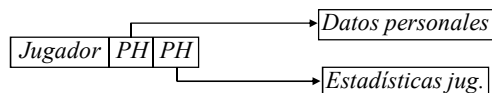
### ***Usos de un Directorio:***

- ***Para relacionar ficheros***: cada entrada guarda punteros a los registros que relaciona. **Ejemplo**: una entrada que contiene tres punteros: uno al fichero *coches*, otro al fichero de *personas*, y otro al de *pólizas*.
- ***Para relacionar dos punteros del mismo registro***: con el fin de poder traducir de uno a otro. Este tipo de directorios se denominan *índices* y se utilizan para mejorar el acceso.
- ***Para asociar una dirección virtual en una posición física***: para poder traducir de una a otra, posibilitando direccionamiento virtual.
  - Las estructuras más habituales son en árbol, y en lista (vector fijo).
  - Si se implementa como vector, se suele hacer coincidir la dirección virtual con la posición en el vector, y el contenido sería el puntero .

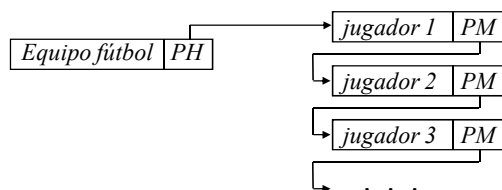


## Tema 3.2.3: Uso de los Punteros

- Según la semántica de la dependencia, se tienen:
  - puntero hijo:** el registro referenciado es parte del primero

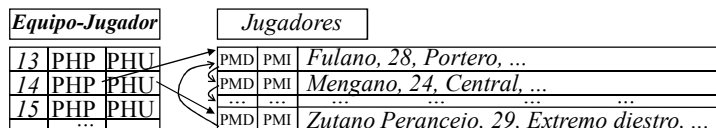


- puntero mellizo:** el referenciado es análogo al primero



## Tema 3.2.3: Uso de los Punteros

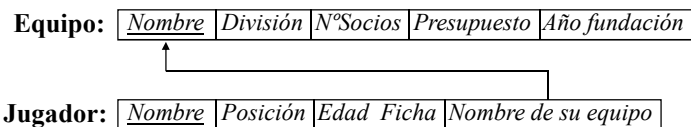
- Los registros almacenan a menudo un 'puntero a siguiente' (**Puntero Mellizo Directo**), y un 'puntero a anterior' (**Puntero Mellizo Inverso**)
- Cuando hay multiplicidad de hijos, se almacenan punteros al primero (**PHP**) y al último (**PHU**). Entre los hijos existirán punteros mellizos
- Los punteros de un registro se suelen agrupar en la cabecera (prefijo) para agilizar así la localización y manejo de los datos.
- Otras veces se recogen en un fichero auxiliar (**directorio**)





## Tema 3.3: Esquemas de Archivos

- Los Esquemas de Archivos recogen la definición de los registros de cada archivo, no las ocurrencias almacenadas en ellos.
- Para dotar de mayor independencia a los elementos del esquema, se pueden plantear las dependencias a nivel lógico.
- En este caso, el identificador lógico del registro referenciado (clave primaria) se incluye en el referenciante (puntero lógico).
- Si un registro padre puede tener un número variable de hijos, la referencia habrá de aparecer en los hijos.



## Tema 3.4: Modelización

### Introducción

- Cuando el volumen de información es muy elevado, es necesario establecer organizaciones de la misma (a nivel lógico).
- La descripción de esa organización recibe el nombre de modelado.

### **Modelo:**

“ Conjunto de normas necesarias para describir datos, los vínculos que presentan entre ellos, y las restricciones que se les aplican.”

### Clasificación

- **Jerárquicos:** los datos siguen una estructura arbórea
- **En Red:** como el anterior, pero con conexiones cruzadas
- **Relacionales:** las generalidades son relaciones entre datos (atributos o columnas) que definen cada ocurrencia (fila)



## Tema 3.4: Sistemas Jerárquicos

### **Introducción**

- Es un estándar *de facto* los primeros macro-sistemas de información seguían este modelo, y sus sucesores han ido manteniendo terminología y enfoque.
- En un modelo de datos, el diseño es a nivel lógico aunque exista siempre una conexión con el nivel físico. En este modelo la conexión es más cercana
- Basa sus estructuras (arbóreas) en el postulado de que en la naturaleza de las clasificaciones de datos está la jerarquía.
- El ejemplo prototípico de sistema gestor basado en este modelo es el IMS (Information Management System) de IBM

<http://www-306.ibm.com/software/data/ims/>  
<ftp://ftp.software.ibm.com/software/data/ims/shelf/presentations/IMSOverview.pdf>



## Tema 3.4.1: Propiedades

### **Propiedades de los esquemas jerárquicos de archivos :**

- Existe un archivo que no es hijo de ningún otro (raíz).
- Todos los archivos (menos el raíz) son hijos de otro.
- Los archivos que no tienen hijos se denominan hojas.
- Un archivo padre puede tener varios hijos (no al revés).

### **Consideraciones:**

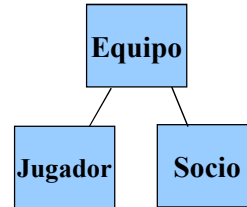
- En general, modelar un problema puede requerir varias jerarquías. Un Esquema de BB.DD. Jerárquica es la colección de subesquemas que describen y recogen el universo de discurso tratado.
- Aún así, el modelo jerárquico presenta muchas limitaciones



## Tema 3.4.2: Notaciones

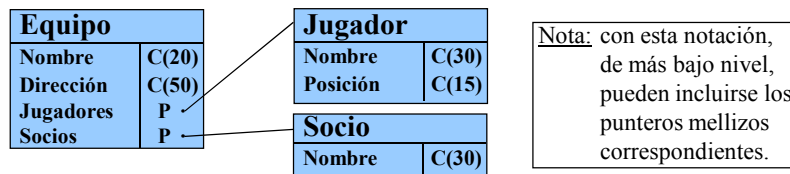
### Notación 1:

- los archivos se representan en una caja
- los vínculos con un arco (no dirigido)
- los hijos se sitúan debajo



### Notación 2:

- los archivos contienen la descripción de sus campos
- el esquema se representa entonces de izquierda a derecha



## Tema 3.4.3: Modelización (I)

- La información debe dividirse en varias unidades (archivos) atendiendo a criterios de:
  - uniformidad (registros homogéneos)
  - no redundancia (no repetir información)
  - acceso (no recuperar campos no requeridos)
  - proceso (no procesar registros innecesariamente)
  - almacenamiento y distribución
- Cada archivo tiene un significado conceptualmente, que describe su función en el universo de discurso.
- Esos archivos se vinculan mediante punteros padre-hijo y mellizos (¡la asociación entre dos archivos siempre es padre-hijo!)



### Tema 3.4.3: Modelización (II)

- La cardinalidad de una asociación entre archivos describe la cantidad de registros de cada archivo que participa en esa asociación
- Los tipos de cardinalidad que vamos a considerar: **1-1**, **1-N**, y **M-N**
- En asociaciones de tipo 1-1, se privilegia como padre a uno de ellos
- En asociaciones de tipo 1-N, se distingue entre padre (1) e hijo (N)  
Los punteros padre-hijo son típicos para implementar la asociación del padre con el primer hijo, y los punteros mellizos para relacionar todos los registros ‘hermanos’ entre sí (haciendo una cadena).
- Sin embargo, un archivo sólo puede tener un padre (no puede tomar parte en varias asociaciones en las que participe como N).
- Las asociaciones M-N no se pueden representar:  
un registro hijo no puede tener varios padres (en el mismo archivo).



### Tema 3.4.3: Modelización (III)

Se pueden crear varias jerarquías, distribuyendo los archivos.  
No es admisible repetir archivos (¡sería una redundancia!)

*Vínculos Virtuales: pueden asociar archivos de distintas jerarquías*

*Vínculos Reales: siempre asocian archivos de la misma jerarquía*

- Los vínculos virtuales tienen distinta naturaleza, y por ello:
  - un archivo puede tener varios padres virtuales
  - dos archivos pueden ser padres virtuales recíprocamente (M-N)
- Los vínculos virtuales pueden representarse:
  - *con un puntero (empotrado) al padre virtual*
  - *con un directorio que apunta a cada hijo virtual*  
(*el padre virtual apunta a la primera entrada, que contiene el puntero virtual al primer hijo y un puntero mellizo a la siguiente entrada; y así sucesivamente hasta la última entrada enlazada*)

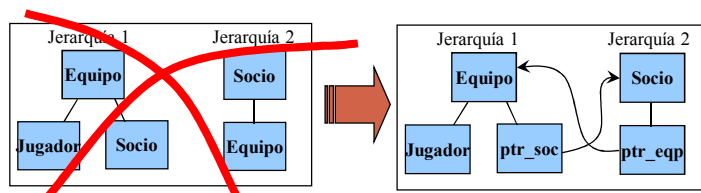




## Tema 3.4.3: Modelización (IV)

### Notación:

- El puntero virtual se representa como un arco dirigido (flecha)
- El directorio es un hijo real cuya etiqueta describe su contenido (ptr\_Socios, dir\_Socios, línea\_Equipo\_Socio, Socio', etc.)



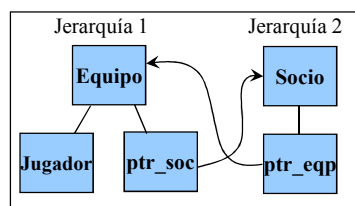
- Un directorio sólo almacena punteros  
(si añadimos atributos pasa a ser un archivo; por ejemplo, si al directorio 'ptr\_eqp' le añadimos 'fecha\_expedición' sería el archivo 'carnet\_socio')



## Tema 3.4.3: Modelización (V)

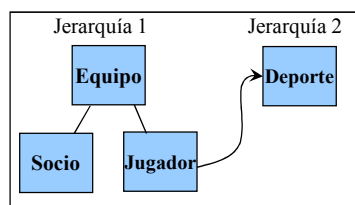
### Ejemplo 1: vínculo M-N

- cualquier equipo cuenta con muchos socios, y cualquier socio puede serlo de varios equipos simultáneamente.



### Ejemplo 2: varios vínculos 1-N

- de un deporte se conocen varios jugadores, y un equipo tiene varios de varios equipos simultáneamente.



#### \* Notas:

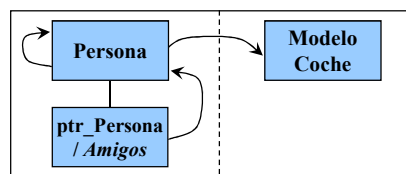
Obsérvese que en este segundo ejemplo podría utilizarse un directorio en lugar de un puntero empotrado. Ambas soluciones son válidas, y debe elegirse una (son redundantes).

Se elegirá la más eficiente según el conjunto de procesos. En ocasiones interesa simultanear ambas por eficiencia (existen procesos que las usan), pero deberá controlarse la redundancia.



### Tema 3.4.3: Modelización (y VI)

- Existen ciertos archivos raíz que sólo tienen hijos virtuales y generalmente son pequeños (un campo) y poco volátiles (casi constantes). Se les denomina ' *ficheros de referencias* ', y se usan para implementar dominios enumerados no cerrados.
- Los vínculos virtuales pueden referenciar archivos de la misma jerarquía. Además, posibilitan las asociaciones reflexivas, como p.e. 'cónyuge' (persona(1)-persona(1)) o 'compañero' (persona(1)-persona(N)).



### Tema 3.4.4: Restricciones

#### Integridad en el Modelo Jerárquico

- *Se resumen en dos normas básicas:*
  - **todo lo apuntado existe**
  - toda ocurrencia no raíz pertenece a un padre (no hay huérfanos)

Desemboca en las siguientes reglas de integridad:

- Cualquier registro hijo puede ser eliminado.
  - Si es hijo real, se deben actualizar los mellizos y el padre real (si es el primero)
- Como los hijos reales forman parte del padre, al borrar un registro padre real se deben eliminar todos sus hijos automáticamente → **Borrado en Cascada**
- Los hijos virtuales pertenecen a otro registro (su padre real) y no pueden eliminarse automáticamente. Para no romper la integridad, se abortarán los borrados de registros con hijos virtuales → **Borrado Restringido**
  - En este caso, también podría contemplarse romper los enlace de los hijos:
    - si se implementa con directorio, eliminando las entradas (B.Cascada)
    - poniendo los punteros empotrados a NIL (borrado 'puesta a nulo')
    - o apuntando a otro registro, por defecto (borrado 'puesta a valor defecto')