

# **Preparador Informática**

[www.preparadorinformatica.com](http://www.preparadorinformatica.com)

## **TEMA 37. INFORMÁTICA**

**MODELO DE DATOS JERÁRQUICO Y EN  
RED. ESTRUCTURAS. OPERACIONES**

## **TEMA 37 INF: MODELO DE DATOS JERÁRQUICO Y EN RED. ESTRUCTURAS. OPERACIONES.**

1. INTRODUCCIÓN
2. MODELO DE DATOS
3. MODELO DE DATOS JERÁRQUICO
  - 3.1. ESTRUCTURA
  - 3.2. VENTAJAS
  - 3.3. INCONVENIENTES
  - 3.4. OPERACIONES
4. MODELO DE DATOS EN RED
  - 4.1. ESTRUCTURA
  - 4.2. VENTAJAS
  - 4.3. INCONVENIENTES
  - 4.4. OPERACIONES
5. CONCLUSIÓN
6. BIBLIOGRAFÍA



Preparador Informática



## 1. INTRODUCCIÓN

Cuando IBM creó su Sistema Administrador de Información o IMS, se establecieron las bases para que la gran mayoría de sistemas de gestión de información de los años setenta utilizaran el **modelo jerárquico**. Este modelo también recibe el nombre de modelo en árbol, ya que utiliza una estructura en árbol invertido para la organización de los datos. La información se organiza con una jerarquía en la que la relación entre las entidades de este modelo siempre es del tipo padre/hijo. De tal manera que existen nodos que contienen atributos o campos y que se relacionarán con sus nodos hijos, pudiendo tener cada nodo más de un hijo, pero un nodo siempre tendrá un sólo padre. Los datos de este modelo se almacenan en estructuras lógicas llamadas segmentos. Los segmentos se relacionan entre sí utilizando arcos. La forma visual de este modelo es de árbol invertido, en la parte superior están los padres y en la inferior los hijos.

El **modelo de datos en red** aparece a mediados de los sesenta como respuesta a limitaciones del modelo jerárquico en cuanto a representación de relaciones más complejas. Podemos considerar a IDS (Integrated Data Store) de Bachman como el primer sistema de base de datos en red. Tras él se intentó crear un estándar de modelo de red por parte de CODASYL, siendo un modelo que tuvo gran aceptación a principios de los setenta. El modelo en red organiza la información en registros (también llamados nodos) y enlaces. En los registros se almacenan los datos, mientras que los enlaces permiten relacionar estos datos. En este modelo se pueden representar perfectamente cualquier tipo de relación entre los datos, pero hace muy complicado su manejo. El sistema de gestión de información basado en el modelo en red más popular es el sistema IDMS.

Hoy en día, debido a sus limitaciones, tanto el modelo jerárquico como el modelo en red están en desuso. Se estudian en este tema porque es importante conocer las características de los modelos anteriores al modelo relacional, ya que tanto el modelo jerárquico como el modelo en red hace varias décadas fueron modelos de datos muy relevantes y utilizados.

## 2. MODELO DE DATOS

En informática, un **modelo de datos** es un lenguaje utilizado para la descripción de una base de datos. Con este lenguaje vamos a poder describir las estructuras de los datos (tipos de datos y relaciones entre ellos), las restricciones de integridad (condiciones que deben cumplir los datos, según las necesidades de nuestro modelo basado en la realidad) y las operaciones de manipulación de los datos (insertado, borrado, modificación de datos). Es importante distinguir entre modelo de datos y esquema. Para clasificar los modelos debemos pensar en el nivel de abstracción, es decir, en lo alejado que esté del mundo real:

- Los modelos de datos **conceptuales** son aquellos que describen las estructuras de datos y restricciones de integridad. Se utilizan durante la etapa de análisis de un problema dado, y están orientados a representar los elementos que intervienen y sus relaciones. Ej: Modelo Entidad-Relación.
- Los modelos de datos **lógicos** se centran en las operaciones y se implementan en algún SGBD. Ej: Modelo Relacional.
- Los modelos de datos **físicos**, son estructuras de datos a bajo nivel, implementadas dentro del propio sistema gestor de base de datos.

Tradicionalmente los modelos de datos se clasifican en tres grupos: **jerárquico, en red y relacionales**. Las bases de datos jerárquicas fueron las primeras en aparecer. La información se representa en forma de árbol. El problema con este tipo de estructura es que no todas las bases de datos se adaptaban a la estructura en árbol. En un intento de eliminar la rigidez de las bases de datos jerárquicas, se desarrolló la estructura en red, en la cual se permitían todo tipo de relaciones. Cualquier conjunto de información es representable mediante una base de datos en red. Por último en los años 70 de la mano de E.F.Codd, aparecieron las bases de datos relacionales, que pretendían obtener una mayor flexibilidad y rigor en el tratamiento de datos.

## 3. MODELO DE DATOS JERÁRQUICO

Un modelo de datos jerárquico utiliza jerarquías o árboles para la representación lógica de los datos. Los archivos son organizados en jerarquías, y normalmente cada uno de ellos se corresponde con una de las entidades de la base de datos. Se representan en forma de árboles jerárquicos.



Los árboles, como instrumentos para la representación de estructuras de datos, presentan problemas por su poca flexibilidad, lo que da origen a una falta de adaptación a muchas organizaciones reales.

No se ha llegado a una formalización matemática del modelo y de sus lenguajes, como ha ocurrido en el caso del relacional; ni tampoco se ha intentado su estandarización, a pesar de lo cual los productos jerárquicos (sobre todo, el IMS y el DL/I de IBM como máximos exponentes de estos sistemas) consiguieron altas cuotas de mercado, aunque la actual difusión de la tecnología relacional los han llevado a convertirse en sistemas superados, lo cual no quiere decir que no persistan todavía importantes aplicaciones soportadas en estos productos.

El producto comercial de tipo Jerárquico más extendido y el único que ha llegado hasta nuestros días es el IMS de IBM con su lenguaje de datos DL/I2. Otro sistema Jerárquico, el System 2000 también tuvo una alta aceptación comercial.

### **3.1. ESTRUCTURA**

En el modelo jerárquico sólo se pueden modelar relaciones 1:N (uno a varios) así que sólo podremos representar situaciones de la vida real donde predominen estas relaciones.

Al igual que con el modelo de red, el principal problema de los sistemas de bases de datos jerárquicos es el de la poca independencia de los programas respecto a cómo están almacenados los datos, lo que dificulta además la programación de software de acceso a estos sistemas. La implementación del modelo Jerárquico en los productos se lleva a cabo en base a punteros; estructura física que varía según los productos.

Es un modelo muy rígido en el que las diferentes entidades de las que está compuesta una determinada situación, se organizan en niveles múltiples de acuerdo a una estricta relación padre/hijo, de manera que un padre puede tener más de un hijo, todos ellos localizados en el mismo nivel, y un hijo únicamente puede tener un padre situado en el nivel inmediatamente superior al suyo.

Esta estricta relación padre/hijo implica que no puedan establecerse relaciones entre segmentos dentro de un mismo nivel.



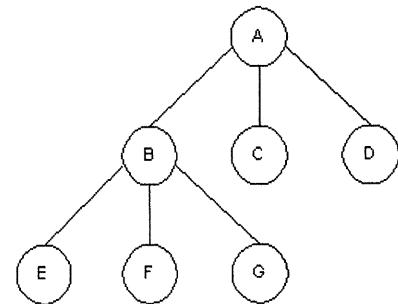
La representación gráfica de un modelo jerárquico se realiza mediante la estructura de árbol invertido, en la que el nivel superior está ocupado por una única entidad, bajo la cual se distribuyen el resto de las entidades en niveles que se van ramificando. Los diferentes niveles quedan unidos por medio de las relaciones. Las entidades se denominan en el caso particular del modelo jerárquico segmentos, mientras que los atributos reciben el nombre de campos.

Los segmentos, se organizan en niveles de manera que en un mismo nivel estén todos aquellos segmentos que dependen de un segmento de nivel inmediatamente superior.

Nivel 1:  
Raíz

Nivel 2

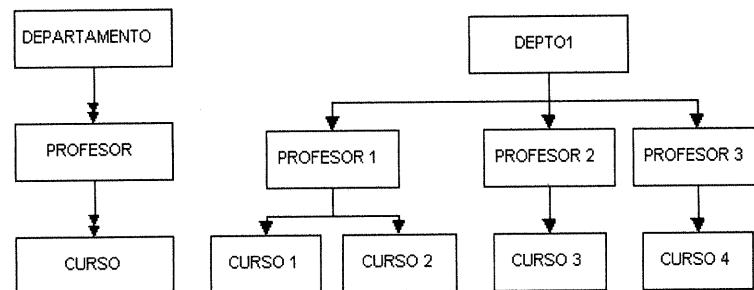
Nivel 3



Por ejemplo, veamos en el siguiente esquema como cada departamento es una entidad que mantiene una relación de uno a muchos con los profesores, que a su vez mantienen una relación de uno a muchos con los cursos que imparten.

Estructura lógica

Ejemplo de base de datos



Los segmentos, en función de su situación en el árbol y de sus características, pueden denominarse como:

- 1) **SEGMENTO PADRE:** Es aquél que tiene descendientes, todos ellos localizados en el mismo nivel.
- 2) **SEGMENTO HIJO:** Es aquél que depende de un segmento de nivel superior. Todos los hijos de un mismo padre están en el mismo nivel del árbol.
- 3) **SEGMENTO RAÍZ:** El segmento raíz de una base de datos jerárquica es el padre que no tiene padre.

La raíz siempre es única y ocupa el nivel superior del árbol.

Una ocurrencia de un segmento de una base de datos jerárquica es el conjunto de valores particulares que toman todos los campos que lo componen en un momento determinado.

Un **REGISTRO** de la base de datos es el conjunto formado por una ocurrencia del segmento raíz y todas las ocurrencias del resto de los segmentos de la base de datos que dependen jerárquicamente de dicha ocurrencia raíz.

Un modelo de datos jerárquico recorre los distintos nodos de un árbol en un *preorden* que requiere tres pasos siendo un camino de acceso a los datos único:

1. Visitar la raíz.
2. Visitar el hijo más a la izquierda, si lo hubiera, que no haya sido visitado.
3. Si todos los descendientes del segmento considerado se han visitado, volver a su padre e ir al punto 1.

Podemos enumerar las siguientes **características** de las estructuras jerárquicas:

- El árbol se organiza en un conjunto de niveles.
- El nodo raíz, el más alto de la jerarquía, se corresponde con el nivel 0.
- Los arcos representan las asociaciones jerárquicas entre dos entidades y no tienen nombre, ya que no es necesario porque entre dos conjuntos de datos sólo puede haber una interrelación.
- Mientras que un nodo de nivel superior (padre) puede tener un número ilimitado de nodos de nivel inferior (hijos), al nodo de nivel inferior sólo le puede corresponder un único nodo de nivel superior. En otras palabras, un padre puede tener varios hijos, pero un hijo sólo tiene un padre.
- Todo nodo, a excepción del nodo raíz, ha de tener obligatoriamente un padre.
- Se llaman *hojas* los nodos que no tienen descendientes.
- Se llama *altura* al número de niveles de la estructura jerárquica.
- Se denomina *momento* al número de nodos.
- El número de hojas del árbol se llama *peso*.
- Sólo están permitidas las interrelaciones 1:1 ó 1:N
- Cada nodo no terminal y sus descendientes forman un subárbol, de forma que un árbol es una estructura recursiva.
- Cada nodo consta de uno o más campos.

- Cuando se elimina un registro padre se deben eliminar todos los registros hijos (integridad de los datos).

Entre las **restricciones** propias de este modelo se pueden resaltar:

- A) Cada árbol debe tener un único segmento raíz.
- B) No puede definirse más de una relación entre dos segmentos en un árbol.
- C) No se permiten las relaciones reflexivas de un segmento consigo mismo.
- D) No se permiten las relaciones N:M.
- E) No se permite que exista un hijo con más de un padre.
- F) Para cualquier acceso a la información almacenada, es obligatorio el acceso por la raíz del árbol, excepto en el caso de utilizar un índice secundario.
- G) El árbol debe recorrer siempre de acuerdo a un orden prefijado: el camino jerárquico.
- H) La estructura del árbol, una vez creada, no se puede modificar.

### 3.2. VENTAJAS

- La mayor ventaja de un modelo jerárquico es la existencia de sistemas de manejo de BD probados que usan este modelo como estructura básica.
- Sencillez de comprensión, relativa simplicidad y facilidad de uso del modelo jerárquico y la familiaridad de los usuarios de procesamiento de datos con la jerarquía.
- Existe una reducción de la dependencia de los datos.
- La predicción del funcionamiento se simplifica a través de relaciones predefinidas.

### 3.3. INCONVENIENTES

- Las relaciones muchos a muchos pueden implantarse sólo de una manera deficiente. Esto puede traer como consecuencia redundancia en los datos almacenados. Sabemos que, en el nivel lógico, la redundancia no es necesariamente mala, por el contrario, promueve la simplicidad. Sin embargo, en el nivel físico la redundancia es indeseable.





- Como resultado del estricto ordenamiento jerárquico, las operaciones conocidas como inserción y de supresión se vuelven muy complejas.
- Toda alta, a no ser que corresponda a un nodo raíz, debe tener un padre.
- La eliminación de padres trae como consecuencia la eliminación de hijos. Como resultado de esto, los usuarios deben tener cuidado cuando tengan la necesidad de poner en funcionamiento la operación de supresión.
- La "raíz" es de tipo nodo dominante. Cualquier nodo hijo es accesible solamente a través de su nodo padre.
- Respecto a las restricciones de usuario, el modelo jerárquico no ofrece ninguna posibilidad de definir las.

### 3.4. OPERACIONES

La base de datos IMS de IBM fue el primer sistema de base de datos jerárquico y el más importante. IMS dispone de dos elementos para definir la estructura de la base de datos:

- DBD (DataBase Description): donde se define la estructura de la BD (los registros que la componen y las relaciones jerárquicas que hay entre ellos).
- PCB (Program Communication Block): subconjunto del DBD generado por medio de la eliminación de determinadas ramas del árbol. Sería el equivalente a una vista del árbol.

Las versiones más modernas disponen de un DDL que permite crear y modificar bases de datos, tablas, claves y relaciones, por lo que es el propio SGBD el encargado de llevar a cabo todas estas operaciones.

Con respecto a las operaciones de manipulación de datos en el sistema jerárquico, éstas constan de dos fases: primero se necesita *localizar* (seleccionar) los datos sobre los que va a trabajar para realizar a continuación la acción de recuperación o actualización sobre dichos datos.

**A) Localización o selección:** La función de selección jerárquica es de tipo navegacional, es decir, trabaja registro a registro. Dada la sencillez del modelo, la función de selección es también muy sencilla, existiendo únicamente las siguientes formas básicas de búsqueda:

- Seleccionar un determinado registro que cumpla una cierta condición. En el lenguaje DL/I se realizará este tipo de selección mediante una sentencia (GET UNIQUE -GU-) que activará el primer registro que cumpla la condición especificada en el predicado que acompaña a la sentencia.
- Seleccionar el siguiente registro, que se encuentra perfectamente definido al existir un único camino jerárquico. También en este caso se puede especificar una condición que habrá de cumplir el registro para ser seleccionado. En DL/I se utiliza una sentencia (GET NEXT -GN-) que selecciona y, al mismo tiempo recupera el siguiente registro en el pre orden.
- Seleccionar el siguiente registro dentro de un padre. Esta sentencia (GET NEXT PARENT -GNP-) es análoga a la anterior, pero la selección termina cuando no haya más descendientes de ese padre.
- Seleccionar el registro padre de otro dado (que ha sido activado previamente) se conoce como normalización jerárquica ascendente, mientras que la selección de descendientes se llama normalización jerárquica descendente.

**B) Acción:** Una vez seleccionado un registro, se tendrá que realizar sobre él una acción, sea de recuperación o de actualización.

La recuperación, que va asociada a la selección en el DL/I, consiste en llevar el registro marcado como activo en la selección realizada previamente al área de entrada/salida. Se utiliza la sentencia GET.

En cuanto a la actualización, es preciso distinguir entre:

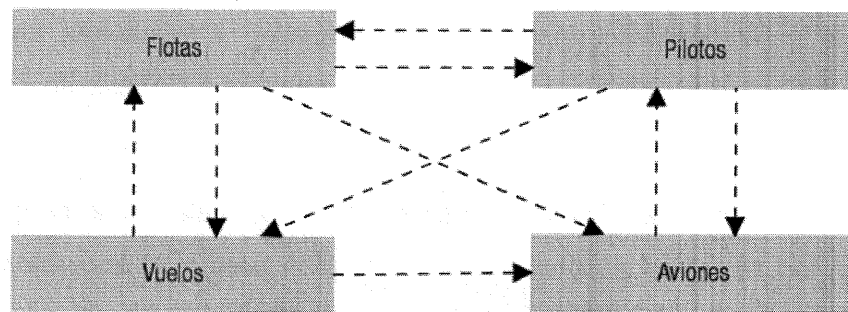
- *Insertar* un conjunto de datos (INSERT -ISRT-)
- *Borrar* un conjunto de datos (DELETE -DLET-)
- *Reemplazar* -modificar- uno o más campos de un registro (REPLACE -REPL-)

Debido a la naturaleza jerárquica de las conexiones entre registros, las inserciones y borrados de registros requieren consideraciones especiales:

- Cuando un nuevo registro se inserta en una base de datos jerárquica, excepto para la raíz, tiene que ser conectado a un nodo padre previamente seleccionado mediante alguna sentencia de selección. El nuevo registro se inserta como hijo del registro padre seleccionado.
- Cuando un registro se borra en una base de datos jerárquica, excepto si se trata de una hoja, se han de borrar todos los registros descendientes de él.

#### 4. MODELO DE DATOS EN RED

El modelo de datos en red es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico).



Permitiendo relaciones de muchos a muchos entre registros vinculados, lo que implica registros principales múltiples. Basado en la teoría matemática de conjuntos, el modelo se construye con conjuntos de registros relacionados. Cada conjunto consiste de un registro propietario o principal y uno o más registros miembros o secundarios. Un registro puede ser miembro o secundario en múltiples conjuntos, permitiendo que este modelo represente relaciones complejas.

#### Preparador Informática

Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa administrar los datos en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales. Fue muy popular en la década de 1970 después de que fue definido formalmente por la Conference on Data Systems Languages (CODASYL).

## 4.1. ESTRUCTURA


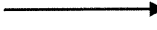

El modelo de datos en red se compone de una colección de registros que se conectan entre sí por medio de ligas.

Un registro se define por medio de uno o varios campos (características propias del registro) y cada uno de esos campos puede tomar un único valor. Entre los registros se establecen relaciones, denominadas ligas. Existen relaciones (ligas) en las que participan sólo dos entidades (binarias) y relaciones en las que participan más de dos entidades (generales) ya sea con o sin atributo descriptivo en la relación.

En el modelo relacional, los tipos de registros equivaldrían a conjuntos de entidades, los registros a entidades, los campos serían el equivalente a los atributos y las ligas serían las restricciones.

El modelo de datos en red general representa las colecciones de registros en forma de nodos de un grafo, y las ligas que se establecen entre éstas mediante arcos que unen dichos nodos.

Un diagrama de estructura de datos es un esquema que representa el diseño de una base de datos en red. Estos diagramas constan de tres componentes básicos:

- Cajas o cuadros: representan un registro.
- Celdas: representan los campos del registro. El campo clave se mostrará subrayado.
- Líneas: representan las ligas entre los registros. En las líneas se pueden indicar la cardinalidad de las relaciones:
  - Si la relación es 1:1 se representa con dos flechas. 
  - Si es 1:N finalizará con punta de flecha en el lado 1. 
  - Si es N:N se representará con una línea sencilla. 

## 4.2. VENTAJAS

- **Flexibilidad:** Las múltiples relaciones padre/hijo permitían a una base de datos en red, representar datos que no tuvieran una estructura jerárquica sencilla.
- **Normalización:** El estándar CODASYL reforzó la popularidad de este modelo.
- **Rendimiento:** A pesar de su superior complejidad las bases de datos en red, reforzaron el rendimiento aproximándolo al de las bases de datos jerárquicas.

## 4.3. INCONVENIENTES

- Al igual que las BD jerárquicas las BD en red resultaban muy rígidas.
- Las relaciones de conjunto y las estructuras de registros tienen que ser especificadas de antemano.
- Modificar la estructura de la base de datos requería típicamente la reconstrucción de la base de datos completa.
- El modelo en red es un modelo tremendamente sencillo de utilizar, pero no deja de tener un carácter general y provoca que en la práctica su instrumentación no resulte nada fácil. Es por esto que los SGBD que se basan en este modelo deben añadir una serie de restricciones a fin de poder implementar la base de datos físicamente y obtener un mayor rendimiento del sistema.

## 4.4. OPERACIONES

Para poder implementar este modelo en una base de datos Codayl y posteriormente realizar sus operaciones sería necesario emplear los siguientes componentes:

- **Elemento de datos:** Unidad de datos más pequeña que se puede referenciar. Puede ser de distintos tipos y puede definirse como dependiente de otros elementos.
- **Agregado de datos:** Sería el equivalente a los campos de un fichero o a los atributos de un modelo entidad relación.

- **Registro:** Conjunto de elementos de datos a los que se le asigna un nombre. Es la unidad básica de acceso y manipulación. Sería el equivalente a los registros en ficheros y a las entidades en el modelo entidad-relación.
- **Conjunto (SET):** Permiten establecer relaciones entre registros. Codasyl permite implementar relaciones 1:N y 1:1. Los conjuntos relacionan jerárquicamente los registros a dos niveles: el nodo raíz es el propietario y los nodos descendientes son los miembros. Un registro puede ser propietario de varios conjuntos y miembro de varios conjuntos, pero no puede ser propietario y miembro de un mismo conjunto. El número de conjuntos que se pueden crear es ilimitado. Físicamente la forma en la que se suelen implementar las relaciones es mediante punteros.
- **Área:** Zonas del soporte físico donde se almacenarán los registros.
- **Clave de base de datos:** Identificador interno único para cada ocurrencia de registro, que proporciona su dirección en la base de datos.

Inicialmente la base de datos Codasyl únicamente proporcionaba un repositorio de datos, por lo que cualquier operación de gestión o acceso a los datos era necesario implementarla directamente por medio de un lenguaje de programación (COBOL), pero con el tiempo se incorporó la posibilidad de gestionar los datos desde el propio sistema gestor. Codasyl propuso la utilización de dos DDLS:

- El lenguaje de definición del esquema (Schema DDL) para describir la estructura de todos los elementos de la BD. Permite definir cuatro tipos de datos: esquemas, áreas, registros y sets.
- El lenguaje de definición de subesquemas (SubSchema DDL) que permite la descripción de subconjuntos del esquema para diferentes usuarios. Por medio de este lenguaje se pueden modificar las contraseñas de los usuarios, los registros que forman parte de un set, el contenido que se mostrará de los registros o el formato de los campos de un registro.

Con respecto al proceso de consulta y modificación de la información, en Codasyl estas tareas dependen en gran medida del programador ya que el SGBD provee de varias operaciones para facilitar el acceso, recuperación y modificación de la información.

Las instrucciones del DML que provee el SGBD son las siguientes:

- **Consulta:** El proceso de consulta consta de dos fases. En primer lugar se ejecuta la instrucción "find" que busca el registro en cuestión y, en segundo lugar, se ejecuta la instrucción "get" que recupera su contenido.
- **Inserción:** Existen dos tipos. A nivel de conjunto (set) y a nivel de registro. La instrucción "insert" permite insertar registros en un set, mientras que la instrucción "store" permite insertar los sets en una base de datos.
- **Modificación:** Utilizamos la instrucción "modify". Para modificar un registro necesitamos consultarlo previamente (buscarlo y recuperarlo), y una vez modificado su contenido utilizamos la instrucción modify para trasladarlo a la base de datos.
- **Eliminación:** Existen también dos instrucciones para eliminar de la BD. "delete" para borrar un registro y "remove" para quitar un registro de un set.

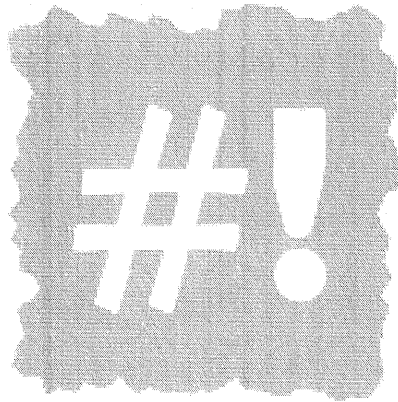
## 5. CONCLUSIÓN

En este tema se ha presentado una visión global de los modelos de datos jerárquico y en red, los cuales tuvieron una gran relevancia hace varias décadas. Para ello se han detallado su estructura y principales ventajas e inconvenientes, así como sus operaciones.

Hoy en día han quedado relegados por otros modelos como el relacional o el objeto-relacional, los cuales solucionan muchos de los problemas que presentaban los modelos jerárquico y en red.

## 6. BIBLIOGRAFÍA

- Date D.J.: **Introducción a los sistemas de bases de datos**. Editorial Addison-Wesley
- De Miguel A,y Piattini M:**Fundamentos y modelos de BBDD**. Edit. Ra-Ma
- Korth H. y Silberschatz: **Fundamentos de bases de datos**. Editorial McGraw-Hill



Preparador Informática

