

# Módulo 2

Código: 484

Bases de datos

Técnico Superior en Desarrollo de  
Aplicaciones Multiplataforma



# UNIDAD 1

## Sistemas de Almacenamiento de la Información



### 3. Contenidos

CONTENIDOS CONCEPTUALES		CONTENIDOS PROCEDIMENTALES
<b>Tipos de datos simples</b>	<ul style="list-style-type: none"><li>• Datos de tipo entero</li><li>• Datos de tipo real</li><li>• Datos de tipo lógico</li><li>• Datos de tipo carácter</li></ul>	<ul style="list-style-type: none"><li>• Clasificar datos simples según su tipo.</li><li>• Obtener resultados mediante tablas de verdad.</li><li>• Asignación entre valores numéricos y caracteres utilizando tabla ASCII.</li></ul>
<b>Estructuras de datos</b>	<ul style="list-style-type: none"><li>• Arrays y Matrices</li><li>• Registros</li><li>• Listas, pilas y colas</li><li>• Árboles</li><li>• Grafos</li></ul>	<ul style="list-style-type: none"><li>• Clasificar estructuras de datos según sus tipos simples y número de componentes.</li><li>• Calcular la posición de un elemento en matrices unidimensionales y bidimensionales.</li><li>• Clasificar listas.</li><li>• Identificar componentes de un árbol.</li><li>• Recorrer árboles.</li><li>• Calcular camino más corto en un grafo.</li></ul>
<b>Ficheros</b>	<ul style="list-style-type: none"><li>• Ficheros secuenciales</li><li>• Ficheros de acceso directo</li><li>• Ficheros secuenciales indexados</li></ul>	<ul style="list-style-type: none"><li>• Clasificar ficheros según su contenido y organización.</li><li>• Buscar un dato concreto según organización del fichero.</li></ul>

### 3. Contenidos

CONTENIDOS CONCEPTUALES			CONTENIDOS PROCEDIMENTALES
Sistemas Gestores de Bases de Datos.	<ul style="list-style-type: none"><li>• Características</li><li>• Utilidades</li><li>• Arquitectura</li></ul>		<ul style="list-style-type: none"><li>• Clasificar SGBDs según sean comerciales o no.</li><li>• Valorar la necesidad de los SGBDs.</li></ul>
Bases de Datos	<ul style="list-style-type: none"><li>• <b>Características</b></li></ul>		<ul style="list-style-type: none"><li>• Identificar fases del proceso de diseño de una Base de datos.</li><li>• Identificar estructura de una BBDD según su modelo lógico.</li><li>• Obtener el campo que relaciona las diferentes tablas de un modelo relacional dado.</li></ul>
	<ul style="list-style-type: none"><li>• <b>Proceso de diseño</b></li></ul>	<ul style="list-style-type: none"><li>• Modelos lógicos para el diseño</li><li>• BBDD Jerárquicas</li><li>• BBDD en Red</li><li>• BBDD Orientadas a Objetos</li><li>• BBDD Relacionales</li></ul>	

# Sesión 1. Contenidos teóricos

---

## DATOS

- Cualquier objeto que pueda ser manejado por el ordenador es lo que se conoce como **dato**. Un ejemplo de dato es un carácter leído desde un teclado o un número que se encuentra almacenado en memoria.
- El tipo de dato identifica la naturaleza del mismo y, como resultado, las restricciones asociadas a él: valores que puede tomar, qué transformaciones que puede ocasionar y operaciones se pueden llevar a cabo con él.

# Sesión 1. Contenidos teóricos

---

## TIPOS DE DATOS

No en todos los lenguajes de programación existen los mismos tipos de datos. A continuación se muestran los más frecuentes:

### Datos de tipo entero

Toma valores del conjunto de números que no tienen parte decimal. Engloba los números naturales, positivos y negativos y el 0. El número de valores diferentes que se pueden generar depende del número de bits que se utiliza en su representación. Se obtiene calculando  $2^n$ , donde  $n$  es dicha cantidad.

Todas las operaciones llevadas a cabo con este tipo de datos son exactas, salvo que se produzcan desbordamientos.

# Sesión 1. Contenidos teóricos

---

## TIPOS DE DATOS

### Datos de tipo real

Representa una aproximación a un número real, es decir, un número con parte decimal. Como ocurre con los números enteros está limitado superior e inferiormente según la cantidad de memoria que haya disponible para almacenarlo.

# Sesión 1. Contenidos teóricos

---

## TIPOS DE DATOS

### Datos de tipo lógico

Toma **valores booleanos**. Es decir, uno de entre: **verdadero o falso** (V, F o 0,1).

Sobre ellos pueden actuar los llamados **operadores lógicos**, siendo Y, O, NO (AND, OR y NOT) los más comunes.

Dependiendo el lenguaje de programación, también podemos encontrar implementados otros operadores como: NO-Y, NO-O, y NO-exclusivo (NAN, NOR, XOR).

## TIPOS DE DATOS

### Datos de tipo carácter

Como su propio nombre indica, representa conjuntos finitos y ordenados de caracteres. Uno de los conjuntos más usuales es el código ASCII. Existen otros como BCD de intercambio normalizado y BCD ampliado, también conocido como EBCDIC.

La única operación interna sobre datos de tipo carácter es la asignación (existen funciones de conversión de tipo).



# Sesión 1. Contenidos teóricos

---

## ESTRUCTURAS DE DATOS o tipo de dato estructurado

Tipos de datos formados por el conjunto de otros datos.

**Un dato de tipo estructurado está compuesto por una serie de datos que guardan alguna relación entre ellos.**

Cuando se trabaja con estructuras de datos, es deseable:

- **Eficiencia en el uso de la memoria:** cuanto menos usemos, mejor.
- **Rapidez de acceso:** acceder de forma rápida a los datos almacenados.

## ESTRUCTURAS DE DATOS. *Clasificación*

- Diferenciamos las estructuras de datos en **homogéneas** o **heterogéneas** según todos los datos elementales que la forman sean del mismo tipo o no.
- Distinguir entre estructuras de datos **dinámicas** o **estáticas** es otra clasificación común. Las primeras se declaran sin especificar el número de componentes que van a tener y van ocupando más memoria según va creciendo durante la ejecución del programa. En caso contrario, será **estática**.

# Sesión 1. Contenidos teóricos

## ESTRUCTURAS DE DATOS. Arrays

- Estructura de datos **homogénea** en que se asocia uno o más **índices** a cada dato contenido. Estos índices determinan la **posición** del dato dentro del array.
- Estas estructuras pueden ser de varias **dimensiones** y tendrán un índice por cada una de ellas.

### Almacenamiento en memoria de un vector

- Los elementos contenidos en un vector se almacenan en posiciones de memoria
- Imaginemos que la posición de memoria del primer elemento es “I” y que cada elemento ocupa “T” bytes. Podemos localizar el elemento “x” con la siguiente operación:  $I + (x-1)*T$

Índices	0	1	2	3	4	5	6	7	8
Datos	7	5	3	0	2	9	1	4	6

Longitud = 9

Array NxM - Matriz [6][4]						
	0	1	2	3	4	5
0	7	5	3	0	2	9
1	6	4	1	6	4	1
2	7	5	3	0	2	9
3	6	4	1	6	4	1

Columnas = 6

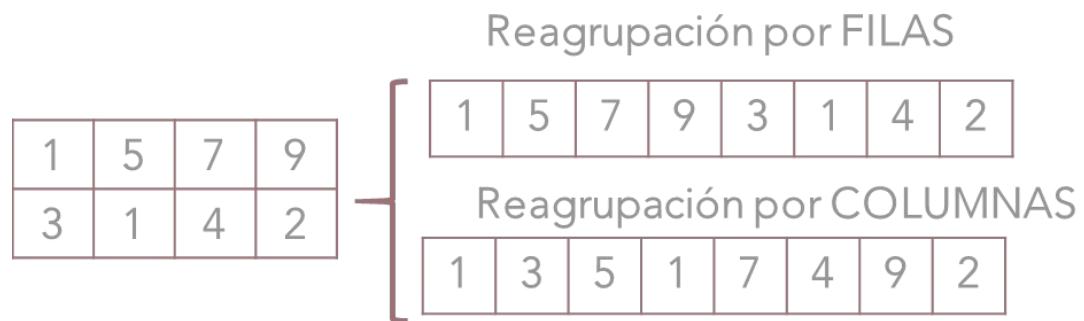
Filas = 4

# Sesión 1. Contenidos teóricos

## ESTRUCTURAS DE DATOS. *Arrays*

### Almacenamiento en memoria de las matrices: Reagrupación por FILAS

- Se basa en posicionar una fila detrás de otra hasta almacenar la matriz completa.
- Consideremos que “I” es la dirección de inicio de almacenamiento de la matriz, “T” el espacio de memoria en bytes que ocupa cada elemento, “LonFil” es el número de celdas que contiene cada fila y “f” y “c” la fila y la columna del elemento al que queremos acceder.



Entonces, la dirección de memoria que ocuparía el elemento viene dada por la siguiente fórmula

$$I + T * [C-1 + (F-1) * LonFil]$$

# Sesión 1. Contenidos teóricos

## ESTRUCTURAS DE DATOS. *Arrays*

### Almacenamiento en memoria de las matrices: Reagrupación por COLUMNAS.

- Se basa en posicionar una columna detrás de otra hasta almacenar la matriz completa.
- Supongamos que “I” es la dirección de inicio de almacenamiento de la matriz, “T” el espacio de memoria en bytes que ocupa cada elemento, “LonCol” es el número de celdas que contiene cada columna y f y c la fila y la columna del elemento al que queremos acceder.



Entonces, la dirección de memoria que ocuparía el elemento viene dada por la siguiente fórmula

$$I + T * [F-1 + (C-1) * LonCol]$$

# Sesión 1. Contenidos teóricos

---

## ESTRUCTURAS DE DATOS. *String*

### Cadena de caracteres

- Estructura homogénea compuesta por caracteres. En este tipo de variables se puede almacenar una palabra, un párrafo, etc.
- El valor que se asigna a este tipo de datos se escribe, normalmente, entre comillas.
- Sobre este tipo de datos se pueden realizar operaciones como ***Concatenación, Extracción de subcadena, Comparación de cadenas, Obtención de longitud***, entre otras.

# Sesión 1. Contenidos teóricos

## ESTRUCTURAS DE DATOS. *Registros*

Estructura heterogénea formada por un conjunto de elementos relacionados entre sí por referenciar a una misma entidad. Cada elemento particular, aquí contenido, se le denomina **campo**.

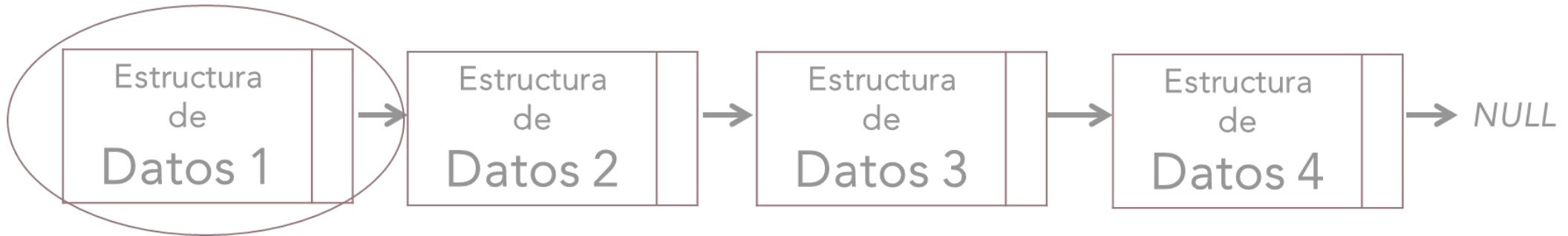
ALUMNO		
	<ul style="list-style-type: none"><li>• Nombre</li><li>• Apellidos</li><li>• Nota</li></ul>	

Cuando se define un registro, se especifica el nombre y el tipo de cada uno de los campos. A su vez, cada campo puede ser de un tipo estructurado.

# Sesión 1. Contenidos teóricos

## ESTRUCTURAS DE DATOS. *Listas*

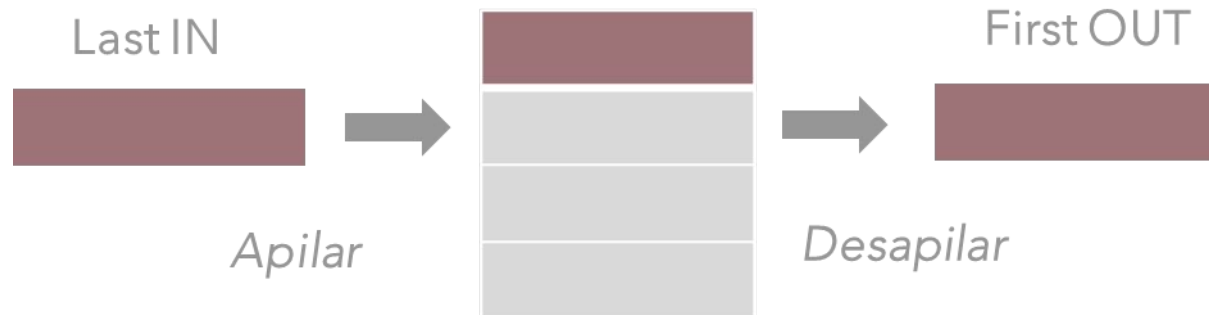
Estructura de datos **dinámica** formada por registros. Estos registros contienen como mínimo dos campos, donde uno de ellos referencia al siguiente.



# Sesión 1. Contenidos teóricos

## ESTRUCTURAS DE DATOS. *Pilas*

- Las **pilas** o listas **LIFO (Last In, First Out)** son un tipo de lista en que se añaden y eliminan los elementos sólo en uno de sus extremos.
- Este tipo de estructuras se utilizan en informática para almacenar instrucciones desde las que se hacen las llamadas a las subrutinas.

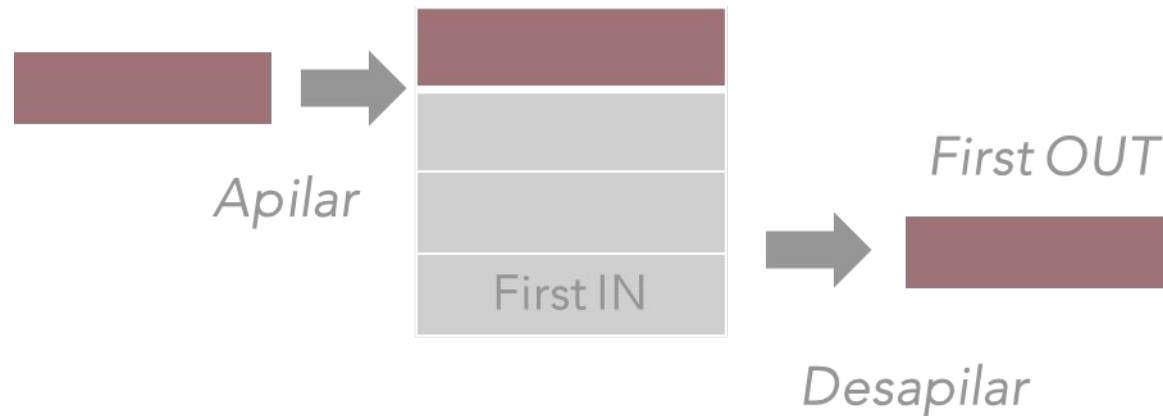




# Sesión 1. Contenidos teóricos

## ESTRUCTURAS DE DATOS. Colas

Otro tipo de lista es la denominada **cola** o **lista FIFO (First In, First Out)**, aquí, las inserciones se realizan en el extremo opuesto de las extracciones.



# Sesión 1. Contenidos teóricos

---

## ESTRUCTURAS DE DATOS. Listas

### Operaciones básicas sobre una lista.

- **Crear:** Operación que creará una lista con la que trabajar.
- **PUSH:** Mediante esta operación se inserta un nuevo elemento en la lista.
- **POP:** Operación contraria a la anterior, extrae un elemento de la lista.
- **Vacía:** Operación que nos permite conocer si una lista tiene o no elementos agregados.

# Sesión 1. Contenidos teóricos

---

## ESTRUCTURAS DE DATOS. Árboles

- Estructura de datos **dinámica**. Sus elementos se denominan nodos y se relacionan de forma **jerárquica**, de forma que todos ellos tienen un único **nodo padre**, a excepción del **nodo raíz** que no tiene y de él descienden todos los demás.
- Cada nodo puede tener varios **nodos hijos**, que son aquellos que le suceden en la jerarquía. Si un nodo no tiene ningún hijo se le llama nodo **terminal u hoja**. Se denomina **grado** al número de hijos que tiene un nodo.

## ESTRUCTURAS DE DATOS. Árboles

### Árboles Binarios

- Tipo especial de árboles con la restricción de que cada nodo tiene, como máximo, 2 hijos; es decir, el grado del árbol es menor o igual a 2.
- Los árboles binarios son los más utilizados en la práctica. Por ejemplo, los árboles de búsqueda, que son árboles ordenados.

# Sesión 1. Contenidos teóricos

---

## ESTRUCTURAS DE DATOS. Árboles

### Recorriendo Árboles

La necesidad más frecuente, cuando se trabaja con árboles binarios, es recorrerlos. Esta operación trata de visitar todos los nodos una única vez. Los diferentes métodos son:

**En profundidad:** se visitan los nodos hasta llegar a las hojas.

**Preorden:** Raíz, subárbol izquierdo, Subárbol derecho.

**Inorden:** Subárbol izquierdo, Raíz, Subárbol derecho.

**Postorden:** Subárbol izquierdo, Subárbol derecho, Raíz.

**En anchura:** se visitan todos los nodos de un mismo grado antes de pasar al siguiente.

# Sesión 1. Contenidos teóricos

---

## ESTRUCTURAS DE DATOS. Grafos

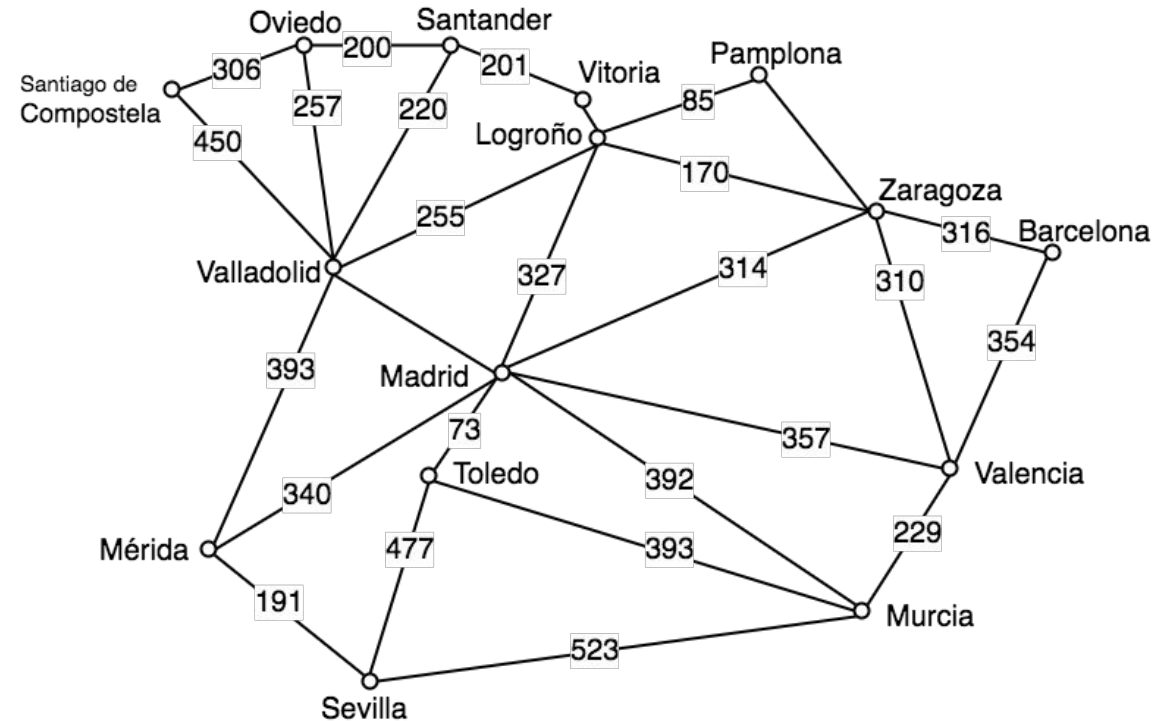
- Estructura de datos dinámica que difiere de los árboles en que cada nodo puede tener varios nodos padre.
- Los elementos que unen los **nodos** se denominan **arcos**. De forma que cada uno agrupa a dos nodos que pueden ser el mismo. Cada arco puede tener un **peso, coste o distancia** y pueden estar o **orientados** o no. En el primer caso se representan mediante una flecha, en el segundo, por un segmento.
- Algunas de las **operaciones básicas que podemos realizar con un grafo son**: crear, añadir vértice, eliminar vértice, añadir arco, eliminar arco, vacío.

# Sesión 1. Contenidos teóricos

## ESTRUCTURAS DE DATOS. Grafos

Algunos **algoritmos** importantes basados en grafos son el algoritmo de *Dijkstra*, el de *Floyd* o el *Warshall*.

Los grafos se recorren igual que los árboles: en **profundidad** o en **anchura**.



# Sesión 1. Contenidos teóricos

---

## SISTEMA DE INFORMACIÓN. Introducción

- Tradicionalmente, el almacenamiento de la información consistía en fichas que se colocaban en armarios. Todas las operaciones que se hacían, como por ejemplo, consultar dicha información. Había que hacerlo de forma manual.
- Con la informática se ha automatizado este proceso, dándole rapidez, calidad y fiabilidad.
- La implantación de los ordenadores ha dado lugar al uso de los denominados archivos informáticos y bases de datos. Además, la llegada de internet ha desembocado en enormes sistemas de información que deben ser accedidos desde diferentes puntos en poco tiempo. Esto sería inviable de no tenerla organizada.

**Los Sistemas de Información (SI) surgen de la necesidad de ordenar la información para que sea accesible.**

# Sesión 1. Contenidos teóricos

---

## SISTEMA DE INFORMACIÓN. Definición

Un **Sistema de Información** es un conjunto de elementos destinados a la organización de información. Sus funciones básicas son:

- Recogida de datos
- Almacenamiento de datos
- Procesamiento de datos
- Elaboración y presentación de datos



# Sesión 1. Contenidos teóricos

---

## SISTEMA DE INFORMACIÓN. Componentes

Un **Sistema de Información** electrónico se compone de los siguientes elementos:

**Datos:** Información que almacena y gestiona el SI

**Hardware:** equipamiento físico utilizado.

**Software:** aplicaciones informáticas específicas que se ocupan de gestionar los datos.

**Recursos humanos:** personal que maneja el sistema de información.

# Sesión 1. Contenidos teóricos

---

## FICHEROS. Introducción

Los **ficheros** o **archivos** siguen siendo, a día de hoy, una herramienta esencial de almacenamiento de información. Algunas de las operaciones básicas que se pueden llevar a cabo en un fichero son:

- Abrir (open)
- Cerrar (close)
- Leer (read)
- Escribir (write)
- Posicionarse (seek)
- Fin de fichero (eof)

# Sesión 1. Contenidos teóricos

---

## FICHEROS. Características

- **Residen en** sistemas de almacenamiento secundario, memoria auxiliar, periférica o externa.
- **Independencia de aplicación con que manejarlo.** El mismo fichero se puede utilizar por diferentes programas.
- **Gran capacidad de almacenamiento:** en teoría, esta capacidad es ilimitada; sin embargo, hay que tener en cuenta que dependemos de la capacidad del soporte donde se almacene la información.

# Sesión 1. Contenidos teóricos

---

## FICHEROS. Clasificación

Una de las diferentes clasificaciones que se pueden hacer de los ficheros es por su organización. Atendiendo a este aspecto, distinguimos entre:

- Ficheros secuenciales
- Ficheros de acceso directo o aleatorio
- Ficheros secuenciales indexados

# Sesión 1. Contenidos teóricos

## FICHEROS SECUENCIALES

- Tipo de archivo que almacena y recupera sus registros de la forma más simple, se guardan uno tras otro (no existe ninguna posición vacía) de modo que para leer los últimos datos hay que leer los anteriores.
- No se pueden hacer operaciones de escritura ni lectura cuando se está leyendo o escribiendo.



# Sesión 1. Contenidos teóricos

## FICHEROS SECUENCIALES

Un ejemplo de uso de ficheros secuenciales puede ser un fichero en el que se almacena una tabla de las notas de una asignatura. Por cada alumno almacenaremos su nombre y su puntuación.

Lucía	8	Andrés	7	Virginia	6	Saúl	9	Héctor	5
-------	---	--------	---	----------	---	------	---	--------	---

Los nombres pueden tener diferentes longitudes de modo que la longitud de los registros puede ser variable. Con ello, conseguimos un mejor aprovechamiento de memoria.

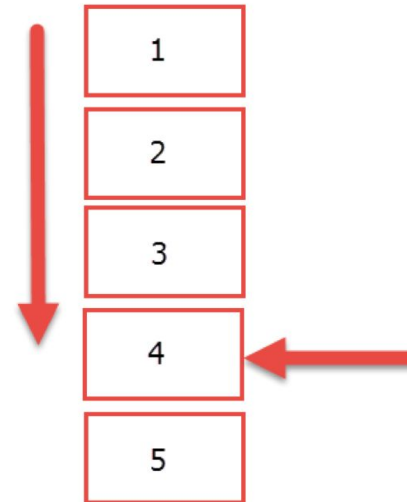
# Sesión 1. Contenidos teóricos

## FICHEROS DE ACCESO DIRECTO

- En este tipo de ficheros podemos dirigirnos directamente al registro que deseamos leer. Es necesario saber la posición donde está.
- Su principal ventaja es la rapidez de acceso a cualquiera de los registros
- Por contra, todos los registros han de tener el mismo tamaño.

### Direct Access *vs.* Sequential Access

With **sequential access**, elements #1,2, 3 must be processed before element #4 can be processed.



With **direct access**, element #4 in the list can be accessed without having to process the elements before it.

NN/g

# Sesión 1. Contenidos teóricos

---

## FICHEROS SECUENCIALES INDEXADOS

Supongamos un fichero secuencial indexado que contiene los datos de los artículos que se venden en una tienda de informática. Los registros que vamos a almacenar contendrán la siguiente información.

Código del Artículo	Precio del Artículo	Nombre del Artículo
---------------------	---------------------	---------------------

Cada registro contendrá tres campos:

1. **Código del artículo:** es la clave principal.
2. **Precio del artículo:** precio de venta del producto.
3. **Nombre del artículo:** breve descripción del mismo



# Sesión 1. Contenidos teóricos

---

## FICHEROS SECUENCIALES INDEXADOS

Los ficheros indexados constan de tres partes:

1. **Área primaria o de datos:** El área primaria se divide en diferentes segmentos. Contiene los datos en sí ordenados de forma ascendente por su campo clave.
2. **Área de índices:** Esta zona es un archivo secuencial en que se almacena, de cada segmento, la dirección de comienzo y la clave más alta. Así, podemos acceder a un registro como si se tratase del índice de un libro.
3. **Área de excedentes o de overflow:** sirve para añadir nuevos registros. No se introducen en el área primaria para evitar una reorganización del mismo.

Este tipo de ficheros requieren de un **campo clave** para ordenar los datos.

A su vez, todos los registros han de tener el mismo tamaño.

El soporte de almacenamiento ha de **permitir el acceso directo**.

# Sesión 1. Contenidos teóricos

## FICHEROS SECUENCIALES INDEXADOS

Supongamos que tenemos almacenados los siguientes datos en cada uno de los ficheros.

Área de Índices

23	D1	51	D2	81	D3	99	D4
----	----	----	----	----	----	----	----

Área de Datos

D1	13	4,75	Cuaderno	15	16,99	Carpeta	23	6,45	Separadores
D2	34	25	Libro	45	3,45	Agenda	51	20,50	Calculadora
D3	65	10,99	Estuche	69	37,58	Mochila	81	3,05	Archivadora
D4	91	2,85	Pack Bolígrafos	93	21	Maletín	99	75	Accesorios de clasificación

Área de Overflow

31	7,95	Papelera	D2	100	0,99	Portalápices
----	------	----------	----	-----	------	--------------

# Sesión 1. Contenidos teóricos

---

## FICHEROS SECUENCIALES INDEXADOS

- Al iniciar una búsqueda, debemos averiguar el segmento en que se encuentra el registro que buscamos. Para ello lo primero es localizar, en el área de índices, el primer registro que supere en valor a la clave que estamos buscando.
- Una vez posicionados en el segmento correcto se busca el registro de forma secuencial. En caso de que no esté, el siguiente paso será buscar en el área de excedentes hasta hallarlo. Si el registro sigue sin aparecer, podemos concluir que no se encuentra en el archivo.

# Sesión 1. Contenidos teóricos

---

## FICHEROS SECUENCIALES INDEXADOS. Caso 1

### 1. El dato a buscar se encuentra en el área de datos

Supongamos que estamos buscando un **estuche**, que es el artículo que tiene código **65**. Lo primero que hacemos es **recorrer secuencialmente el área de índices** e ir comparando nuestra clave, con la del elemento que busquemos, hasta encontrar alguna que supere a la nuestra. En este ejemplo, la primera que supera la clave es la **D3, con el código 81**. Lo que quiere decir que el dato se encuentra en el segmento 3. A continuación accedemos al segmento tercero, y comparamos nuestra clave con la del primer registro de este segmento. Como vemos, ambas coinciden, luego hemos encontrado el registro buscado.

# Sesión 1. Contenidos teóricos

---

## FICHEROS SECUENCIALES INDEXADOS. Caso 2

### 2. El dato está en la zona de overflow

Ahora vamos a buscar la **Papelera**, que tiene como código el número **31**. Lo primero que hacemos es **acceder a la zona de índices**, hasta encontrar un segmento que tenga el código superior al de nuestro artículo. En nuestro caso sería el **segmento 2 con el código 51**. Accedemos al segundo segmento y vamos buscando secuencialmente hasta encontrar nuestro registro o alguno que lo supere. El primer artículo es un **monitor con el código 34**, que supera al 31 de la papelera. **Abrimos la zona de excedentes y buscamos secuencialmente** nuestro registro. Leemos el primer registro y comprobamos que tienen el mismo código, hemos encontrado el registro.

# Sesión 1. Contenidos teóricos

---

## FICHEROS SECUENCIALES INDEXADOS. Caso 3

### 3. El dato no está

Vamos a buscar un artículo cuyo **código es el 66**. Lo primero que hacemos es **buscar el segmento** en el que se encuentra. En este caso saldría el **segmento tercero**. A continuación leemos secuencialmente los registros de dicho segmento hasta que encontremos el registro o alguno con una clave superior. Al leer el segundo registro vemos que su clave supera a 66, luego **debemos abrir la zona de excedentes** e ir leyendo uno a uno todos los registros del mismo, hasta encontrarlo o acceder a una clave mayor que la nuestra. Leemos el primer registro, como su clave es menor que la buscada, **accedemos al segundo**. **Al ser su clave mayor, llegamos a la conclusión de que el dato no se encuentra en el fichero.**

# Sesión 1. Actividades

## ACTIVIDAD 1. Test general de la sesión. 10 preguntas.

1. En todos los lenguajes de programación existen los mismos tipos de datos

a) Verdadera

b) Falsa

2. Si para la representación de un dato de tipo entero se utilizan 4 bits. ¿Cuál es el número de datos distintos que se pueden generar?

a) 8

b) 16

c) 4

d) 32

3. Es un operador lógico

a) AND

b) XOR

c) A y B son ciertas

d) Ninguna de las anteriores

4. Las estructuras de datos son

a) Siempre Homogéneas

b) Estáticas o dinámicas

c) A y B son ciertas

d) Ninguna de las anteriores

5. Las listas son

a) Pilas o Colas

b) Dinámicas

c) A y B son ciertas

d) Ninguna de las anteriores

# Sesión 1. Actividades

## ACTIVIDAD 1. Test general de la sesión. 10 preguntas.

### 6. Los nodos en los árboles binarios

a) Siempre tienen 2 hijos	b) Siempre tienen 2 padres	c) Como máximo tienen 2 hijos.	c) A y B son ciertas
---------------------------	----------------------------	--------------------------------	----------------------

### 7.Si recorremos los nodos de un árbol en profundidad en “Preorden”

a) Visitamos todos los nodos de un mismo grado antes de pasar al siguiente.	b) Visitamos primero raíz, después subárbol izquierdo y por último subárbol derecho.	c) Visitamos primero subárbol izquierdo, después raíz y por último subárbol derecho.	d) Visitamos subárbol izquierdo, después subárbol derecho y por último raíz.
---	--	--	--



# Sesión 1. Actividades

---

## ACTIVIDAD 1. Test general de la sesión. 10 preguntas.

**8. En un grafo los nodos se relacionan mediante arcos. Estos pueden**

- |                          |                          |                      |                              |
|--------------------------|--------------------------|----------------------|------------------------------|
| a) Estar orientados o no | b) Tener un peso o coste | c) A y B son ciertas | d) Ninguna de las anteriores |
|--------------------------|--------------------------|----------------------|------------------------------|

**9. Un sistema de información se compone de**

- |                     |                               |                         |                         |
|---------------------|-------------------------------|-------------------------|-------------------------|
| a) Datos y Software | b) Datos, Software y Hardware | c) A + Recursos Humanos | d) B + Recursos humanos |
|---------------------|-------------------------------|-------------------------|-------------------------|

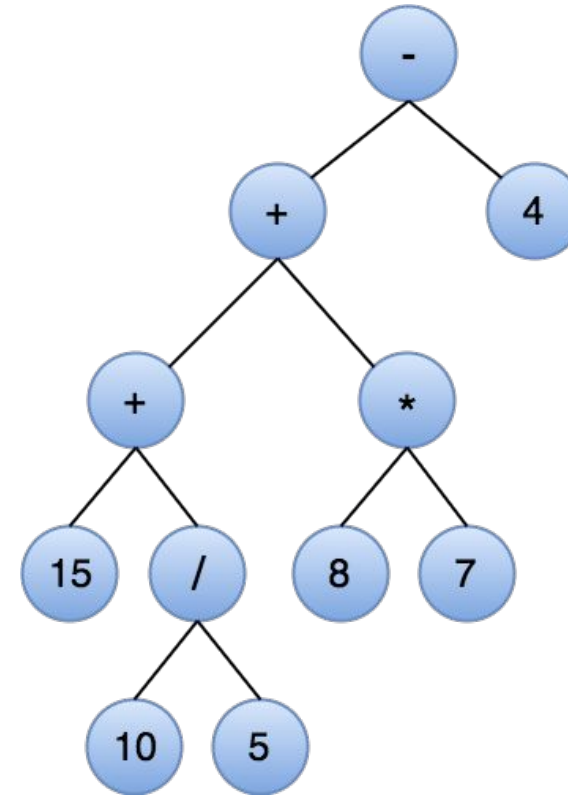
**10. Los ficheros secuenciales indexados contienen**

- |                  |                    |                       |                         |
|------------------|--------------------|-----------------------|-------------------------|
| a) Área de datos | b) Área de índices | c) Área de excedentes | d) Todas las anteriores |
|------------------|--------------------|-----------------------|-------------------------|

# Sesión 1. Actividades

## ACTIVIDAD 4. Métodos de recorrer un árbol dado

- Suponiendo un paréntesis por cada subárbol, ¿Cuál es el resultado de realizar la operación que se plantea al recorrer el siguiente árbol binario en “Profundidad-Inorden”?
- ¿De qué otros modos podríamos recorrer el árbol?
- ¿Cómo sería el recorrido de cada uno de ellos?



# Sesión 1. Actividades

## ACTIVIDAD 5. Búsqueda de un dato en un fichero

Explica los pasos para encontrar la información referente a Rodrigo (Código 60) en el siguiente fichero de acceso secuencial-indexado.

ÁREA DE ÍNDICES		28	D1	39	D2	64	D3	81	D4	
		Código	Nombre	Residencia	Código	Nombre	Residencia	Código	Nombre	Residencia
ÁREA DE DATOS	D1	18	Maria	Madrid	22	David	Málaga	28	Sofia	Córdoba
	D2	35	Juan	Cuenca	36	Raúl	Cáceres	39	Diego	Lugo
	D3	51	Ana	Toledo	57	Estela	Barcelona	64	Eva	Asturias
	D4	73	Elisa	Burgos	78	Hugo	Madrid	81	Carlos	Toledo
ÁREA DE OVERFLOW		60	Rodrigo	Valladolid	75	Irene	León			

# Sesión 2. Contenidos teóricos

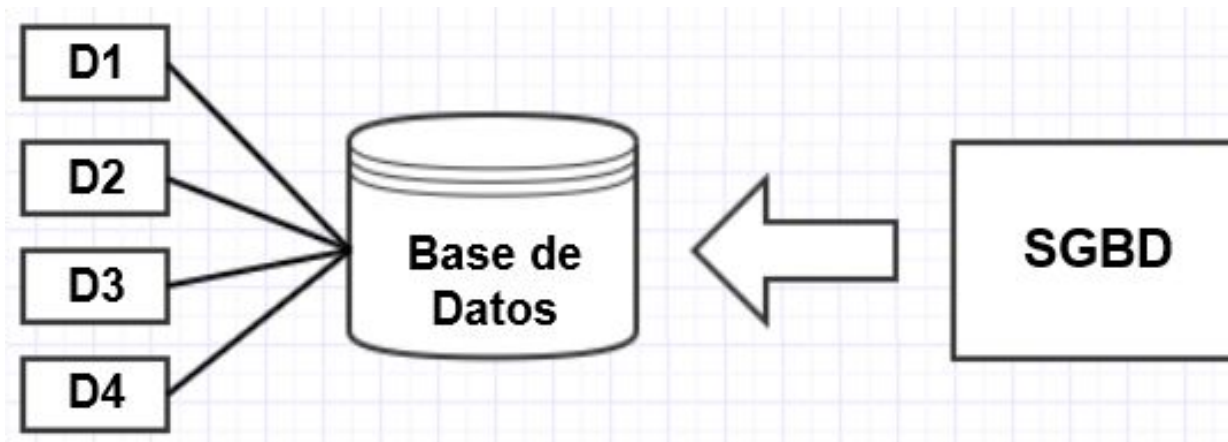
## BASES DE DATOS

Como los Sistemas de ficheros en ocasiones no cubren todas las necesidades, surgieron las Bases de Datos (BD) y los Sistemas Gestores de Bases de Datos (SGBD).

- **Bases de Datos**: *Conjunto de datos* almacenados. Entre ellos existen relaciones lógicas y diseñados para satisfacer los requisitos de una empresa u organización.
- **Sistemas Gestores de Bases de Datos (SGBD)**: Para funcionar, una base de datos no sólo debe almacenar los datos. También debe proveer servicios que permitan consultarlos y administrarlos: control de errores, validación de valores, ejecución de consultas... Todo esto es realizado por un **DBMS** (Database Management System), conocido en castellano como SGBD (Sistema de gestión de bases de datos).

# Sesión 2. Contenidos teóricos

## BASES DE DATOS vs SISTEMA GESTOR DE BASE DE DATOS



# Sesión 2. Contenidos teóricos

## SISTEMAS GESTORES DE BASE DE DATOS

Algunos de los DBMS más conocidos son:

- Comerciales:
  - Oracle Database (Oracle)
  - SQL Server (Microsoft)
  - DB2 (IBM)
- Abiertos:
  - MySQL (Oracle)
  - PostgreSQL o Postgres
  - H2
  - HSQL
  - Derby (Java DB)
  - SQLite



## Sesión 2. Contenidos teóricos

---

### BASES DE DATOS. Características

- **Independencia**. La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- **Redundancia mínima**. Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante.
- **Consistencia**. Vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.
- **Seguridad**. Deben garantizar que esta información se encuentra asegurada frente a usuarios malintencionados.

# Sesión 2. Contenidos teóricos

---

## BASES DE DATOS. Características

- **Integridad**. Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados.
- **Respaldo y recuperación**. Deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos.
- **Control de la concurrencia**. Lo más habitual es que sean muchas las personas que acceden a una base de datos; ésta debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.



# Sesión 2. Contenidos teóricos

---

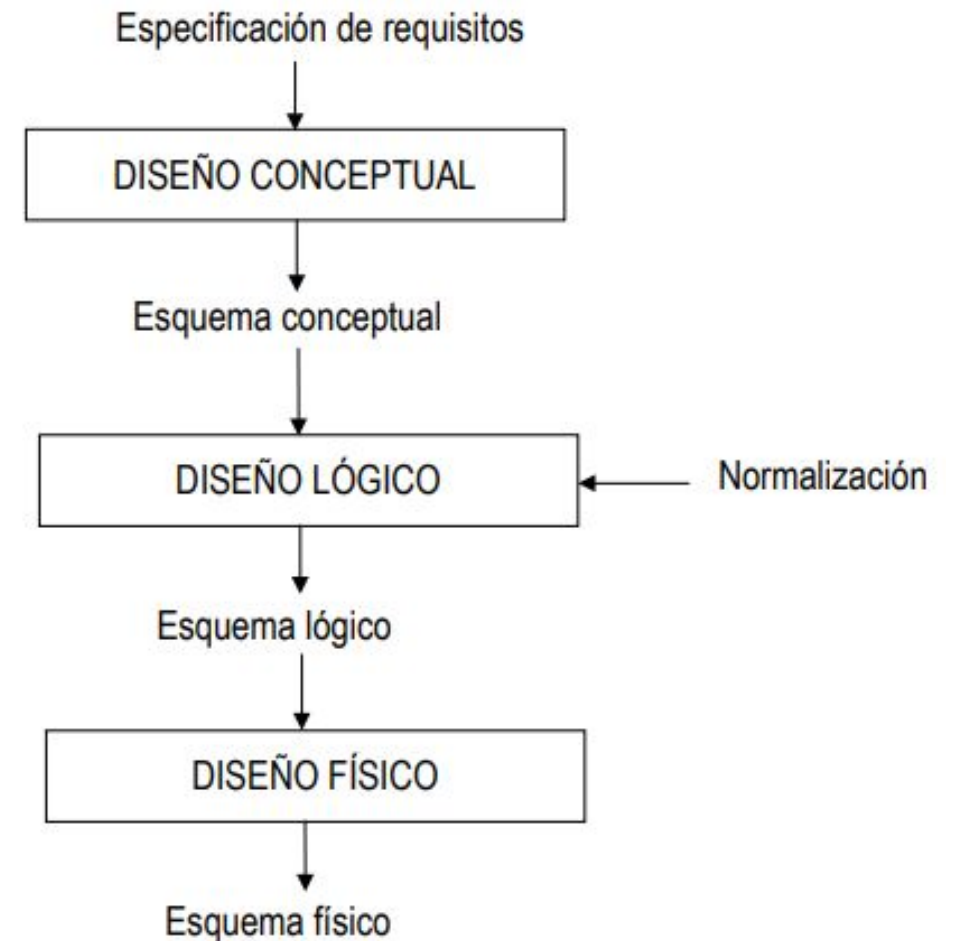
## SISTEMA GESTOR DE BASE DE DATOS. Principales utilidades

- Herramientas para **administrar y crear la estructura física** para almacenamiento de los datos.
- Herramientas para la **manipulación de los datos** (añadir, modificar, borrar o consultar datos).
- Herramientas de **recuperación**.
- Herramientas de **copias de seguridad**.
- Herramientas para la **exportación e importación de datos**.

# Sesión 2. Contenidos teóricos

## PROCESO DE DISEÑO DE UNA BASE DE DATOS

- El proceso de diseño de una base de datos se descompone en **subprocesos**.
- Cada subproceso se realiza con técnicas y métodos específicos.
- Fases:
  - Diseño conceptual
  - Diseño lógico
  - Diseño físico



## Sesión 2. Contenidos teóricos

### PROCESO DE DISEÑO DE UNA BASE DE DATOS: DISEÑO CONCEPTUAL



- **Propósito:** Obtener un esquema completo que lo exprese todo.
- Se parte de la Especificación De Requisitos y se obtiene el Esquema Conceptual
- **Esquema Conceptual:**
  - **Describe** a alto nivel **el contenido** de información de la base de datos.
  - Es **independiente** del modelo de BD y del SGBD (Sistema gestor de bases de datos) que se vaya a utilizar.
- **Modelo conceptual:** Lenguaje que se utiliza para describir esquemas conceptuales.

# Sesión 2. Contenidos teóricos

## PROCESO DE DISEÑO DE UNA BASE DE DATOS: DISEÑO CONCEPTUAL

1º EV



- **Propósito:** Obtener un esquema completo que lo exprese todo.
- Se parte de la Especificación De Requisitos y se obtiene el Esquema Conceptual
- **Esquema Conceptual:**
  - **Describe** a alto nivel **el contenido** de información de la base de datos.
  - Es **independiente** del modelo de BD y del SGBD (Sistema gestor de bases de datos) que se vaya a utilizar.
- **Modelo conceptual:** Lenguaje que se utiliza para describir esquemas conceptuales.

## Sesión 2. Contenidos teóricos

### PROCESO DE DISEÑO DE UNA BASE DE DATOS: DISEÑO LÓGICO



- **Propósito:** Obtener una representación que use de la manera más eficiente los recursos disponibles en el modelo lógico para estructurar datos y modelar restricciones.
- Se parte del Esquema conceptual y se obtiene el Esquema lógico.
- **El Esquema Lógico:**
  - **Describe la estructura** de la base de datos **según el modelos de SGBD que se vaya a utilizar** (jerárquico, red, orientado a objetos, relacional...)
  - Es **dependiente** del modelo de BD que soporte el SGBD que se vaya a utilizar.
- **Modelo lógico:** Lenguaje que se utiliza para describir esquemas lógicos; hay varios modelos lógicos: de red, relacional, orientado a objetos, ...

# Sesión 2. Contenidos teóricos

## PROCESO DE DISEÑO DE UNA BASE DE DATOS: DISEÑO LÓGICO

1º EV



- **Propósito:** Obtener una representación que use de la manera más eficiente los recursos disponibles en el modelo lógico para estructurar datos y modelar restricciones.
- Se parte del Esquema conceptual y se obtiene el Esquema lógico.
- **El Esquema Lógico:**
  - **Describe la estructura** de la base de datos **según el modelos de SGBD que se vaya a utilizar** (jerárquico, red, orientado a objetos, relacional...)
  - Es **dependiente** del modelo de BD que soporte el SGBD que se vaya a utilizar.
- **Modelo lógico:** Lenguaje que se utiliza para describir esquemas lógicos; hay varios modelos lógicos: de red, relacional, orientado a objetos, ...

## Sesión 2. Contenidos teóricos

### PROCESO DE DISEÑO DE UNA BASE DE DATOS: DISEÑO FÍSICO.



- **Propósito:** Obtener el esquema físico final que será implementado en la base de datos por medio del SGBD escogido.
- Se parte del Esquema lógico y se obtiene el Esquema físico.
- **El Esquema Físico:**
  - **Crea las estructuras** de almacenamiento y los datos de la base de datos.
  - Es **dependiente** del SGBD que se vaya a utilizar.
  - Se expresa haciendo uso del lenguaje de definición de datos del SGBD.

`CREATE DATABASE`

`CREATE TABLE`

`CREATE VIEW`

`CREATE INDEX`

`CREATE SCHEMA`

`CREATE SNAPSHOT`

`CREATE CLUSTER`

## Sesión 2. Contenidos teóricos

### PROCESO DE DISEÑO DE UNA BASE DE DATOS: DISEÑO FÍSICO.

2º EV



- **Propósito:** Obtener el esquema físico final que será implementado en la base de datos por medio del SGBD escogido.
- Se parte del Esquema lógico y se obtiene el Esquema físico.
- **El Esquema Físico:**
  - **Crea las estructuras** de almacenamiento y los datos de la base de datos.
  - Es **dependiente** del SGBD que se vaya a utilizar.
  - Se expresa haciendo uso del lenguaje de definición de datos del SGBD.

`CREATE DATABASE`

`CREATE TABLE`

`CREATE VIEW`

`CREATE INDEX`

`CREATE SCHEMA`

`CREATE SNAPSHOT`

`CREATE CLUSTER`



# Sesión 2. Contenidos teóricos

---

## MODELOS LÓGICOS PARA EL DISEÑO DE BASES DE DATOS

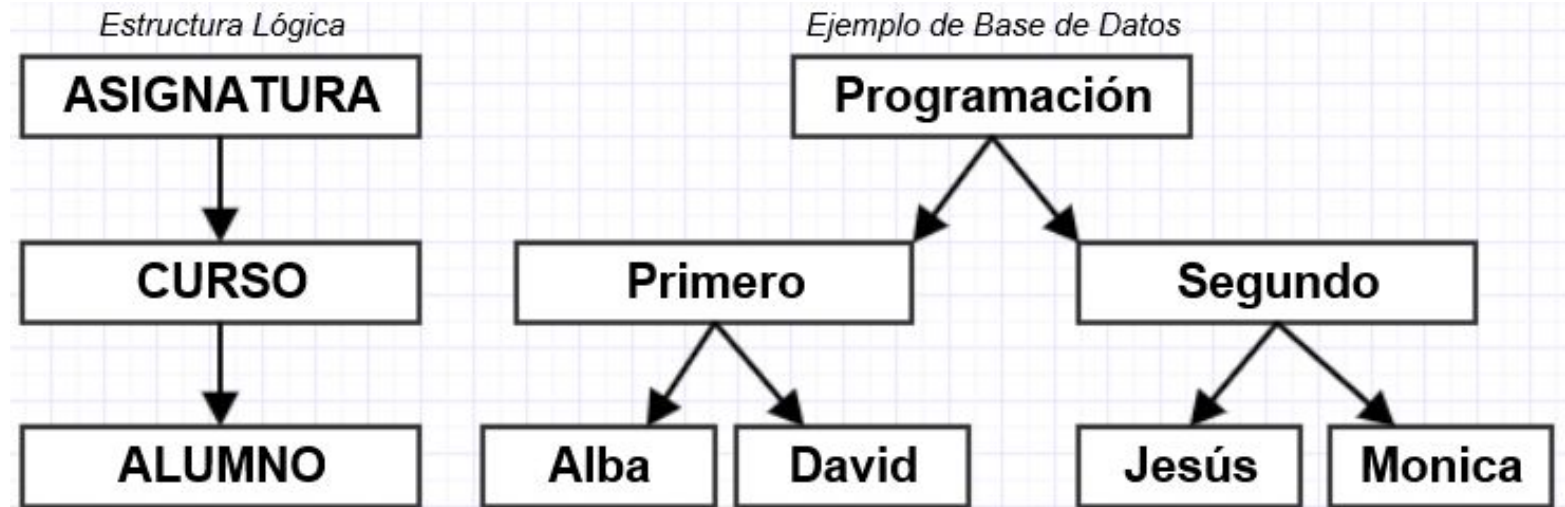
Se pueden distinguir cuatro grandes modelos de bases de datos. Éstos son:

- Base de datos jerárquica
- Base de datos de red
- Base de datos orientada a objetos
- Base de datos relacional

# Sesión 2. Contenidos teóricos

## MODELOS LÓGICOS PARA EL DISEÑO DE BASES DE DATOS. Base de datos Jerárquica

- Una base de datos de tipo jerárquico organiza los datos utilizando **árboles**.
- Cada **nodo** representa un **tipo de registro conceptual**, es decir, una **entidad**. A su vez, cada registro o segmento se compone por **campos**.
- Las **relaciones** entre entidades están representadas por las **ramas**.



## Sesión 2. Contenidos teóricos

---

### MODELOS LÓGICOS PARA EL DISEÑO DE BASES DE DATOS. Base de datos Jerárquica

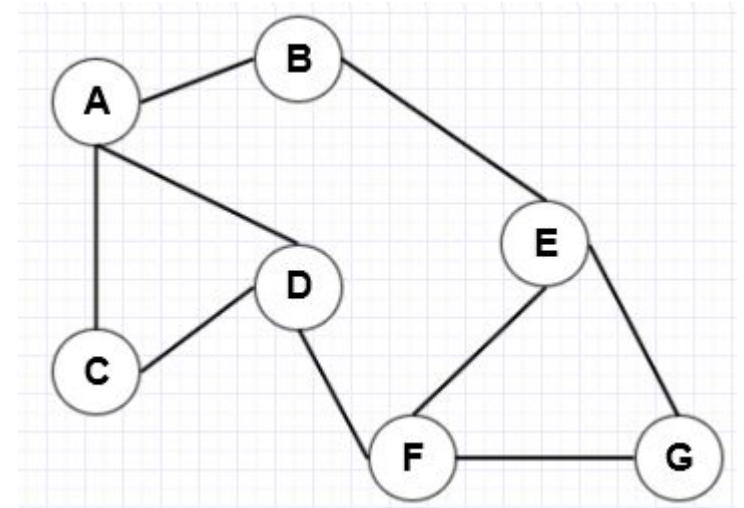
A modo resumen, se enumeran las siguientes características de las bases de datos jerárquicas:

- La organización se dispone en forma de árbol.
- Los segmentos se enlazan utilizando relaciones uno a varios.
- Cada nodo consta de uno o más campos.
- Cada ocurrencia de un registro padre puede tener distinto número de ocurrencias de registros hijos.
- Cuando se elimina un registro padre se deben eliminar todos los registros hijos (integridad de los datos).
- Todo registro hijo debe tener un único registro padre excepto la raíz.

## Sesión 2. Contenidos teóricos

### MODELOS LÓGICOS PARA EL DISEÑO DE BASES DE DATOS. Base de datos en Red

El modelo en red intenta **superar las deficiencias del enfoque jerárquico**, permitiendo el tipo de relaciones de muchos a muchos, mediante la estructura de red o grafo: Al igual que en la estructura jerárquica, **cada nodo puede tener varios hijos, pero a diferencia de ésta, también puede tener varios padres.**



## Sesión 2. Contenidos teóricos

---

### MODELOS LÓGICOS PARA EL DISEÑO DE BASES DE DATOS.

#### Base de datos Orientada a Objetos

Las aplicaciones de las bases de datos en áreas como el diseño asistido por computadora, la ingeniería de software y el procesamiento de documentos no se ajustan al conjunto de suposiciones que se hacen para aplicaciones del estilo de procesamiento de datos. El **modelo de datos orientado por objetos** se ha propuesto para tratar algunos de estos nuevos tipos de aplicaciones.

## Sesión 2. Contenidos teóricos

### MODELOS LÓGICOS PARA EL DISEÑO DE BASES DE DATOS. Base de datos Relacional

Las bases de datos relacionales son **las más conocidas y utilizadas**, tanto por su sencillo manejo, como por estar al alcance de todo tipo de usuarios, al estar muy difundidas.

- Cumple con el modelo relacional postulado por Edgar Frank Codd en 1970

En una base de datos relacional, el objetivo es que el usuario vea las bases de datos como una estructura lógica de relaciones (**tablas**), simple y uniforme.

- Una base de datos se compone de varias tablas o relaciones.
- No pueden existir dos tablas con el mismo nombre ni registro.
- Cada tabla es a su vez un conjunto de campos (columnas) y registros (filas).
- La relación entre tablas se lleva a cabo por medio de las claves primarias y claves foráneas (o ajenas).

# Sesión 2. Actividades

## ACTIVIDAD 7. Test general de la sesión. 10 preguntas

### 1. ¿En qué se diferencia una Base de datos de un DBMS?

a) Son lo mismo	b) DBMS no existe	c) Las bases de datos se componen del conjunto de datos mientras que los DBMS son la herramienta software que permite trabajar con ellas.	d) Los DBMS no son necesarios para trabajar con Bases de Datos.
-----------------	-------------------	---	---

### 2. ¿A qué nos referíamos cuando decimos que una Base de Datos debe ser consistente?

a) Evita que exista información repetida.	b) La información que aparece repetida debe ser coherente.	c) Trata de adoptar las medidas necesarias para garantizar la validez de la información	d) B y C son correctas
---	--	---	------------------------

### 3. ¿En qué fase del proceso de diseño de una Base de Datos se debe tener en cuenta el modelo de BD que soporta el SGBD?

a) Diseño Conceptual	b) Diseño Lógico	c) Diseño Físico	d) B y C son correctas
----------------------	------------------	------------------	------------------------

# Sesión 2. Actividades

## ACTIVIDAD 7. Test general de la sesión. 10 preguntas

**4. Modelo lógico de Bases de Datos que utiliza árboles para su representación.**

- |                    |                |                    |                             |
|--------------------|----------------|--------------------|-----------------------------|
| a) BBDD jerárquica | b) BBDD en red | c) BBDD relacional | d) BBDD Orientada a Objetos |
|--------------------|----------------|--------------------|-----------------------------|

**5. Modelo lógico de Bases de Datos que intenta superar las deficiencias del modelo jerárquico.**

- |                    |                             |                |            |
|--------------------|-----------------------------|----------------|------------|
| a) BBDD relacional | b) BBDD Orientada a Objetos | c) BBDD en red | d) Ninguno |
|--------------------|-----------------------------|----------------|------------|

**6. Modelo lógico de Bases de Datos que surge para adaptarse a nuevos diseños de Ingeniería del Software**

- |                    |                             |                |                    |
|--------------------|-----------------------------|----------------|--------------------|
| a) BBDD relacional | b) BBDD Orientada a Objetos | c) BBDD en red | d) BBDD jerárquica |
|--------------------|-----------------------------|----------------|--------------------|

**7. ¿En qué fase del proceso de diseño de una BBDD se realiza la “normalización”?**

- |                      |                  |                  |                                  |
|----------------------|------------------|------------------|----------------------------------|
| a) Diseño conceptual | b) Diseño lógico | c) Diseño físico | d) Esta fase no se lleva a cabo. |
|----------------------|------------------|------------------|----------------------------------|

**8. Modelo lógico de Bases de Datos cuyo diseño se fundamenta en la utilización de tablas.**

- |                    |                             |                |                    |
|--------------------|-----------------------------|----------------|--------------------|
| a) BBDD relacional | b) BBDD Orientada a Objetos | c) BBDD en red | d) BBDD jerárquica |
|--------------------|-----------------------------|----------------|--------------------|



# Sesión 2. Actividades

---

## ACTIVIDAD 7. Test general de la sesión. 10 preguntas

**9. Modelo lógico de diseño de Bases de Datos basado en grafos en que un nodo puede tener varios padres**

- |                    |                |                    |                             |
|--------------------|----------------|--------------------|-----------------------------|
| a) BBDD jerárquica | b) BBDD en red | c) BBDD relacional | d) BBDD Orientada a Objetos |
|--------------------|----------------|--------------------|-----------------------------|

**10. Modelo lógico de diseño de Bases de Datos más conocido y utilizado.**

- |                    |                |                    |                             |
|--------------------|----------------|--------------------|-----------------------------|
| a) BBDD jerárquica | b) BBDD en red | c) BBDD relacional | d) BBDD Orientada a Objetos |
|--------------------|----------------|--------------------|-----------------------------|

# Sesión 2. Actividades

## ACTIVIDAD 8. Crucigrama. Vocabulario Unidad 1

### VERTICALES

1. Tipo de estructuras en que los datos de que se compone, son de diferentes tipos.
2. Modelo lógico de Base de Datos que intenta solventar las deficiencias del enfoque jerárquico.
3. Algoritmo que permite conocer el camino más corto en un grado.
4. Estructura de datos que se declara sin especificar el número de componentes que tendrá.
5. Método de recorrido de un árbol en que se visitan todos los nodos de un nivel antes de pasar al siguiente.
6. Tipo de dato que únicamente puede tomar los valores de Verdadero o Falso.
7. Toma medidas con objeto de asegurar la validez de los datos almacenados.
8. Fase del proceso de diseño de una Base de Datos en que se parte de los requisitos.
9. Su control evita inconsistencias cuando se accede a la información desde diferentes puntos al mismo tiempo.

### HORIZONTALES

1. Fase del proceso de diseño de una Base de Datos relacional en que se normaliza.
2. Siglas en inglés de Sistema Gestor de Bases de Datos.
3. Tipo de lista cuyo comportamiento es LIFO
4. Última fase del proceso de diseño de una Base de Datos.
5. Tipo de lista cuyo comportamiento es FIFO
6. Hace referencia a la coherencia de la información redundante y a su actualización simultánea.
7. Modelo lógico de Base de Datos más conocido y utilizado.
8. Tipo de dato compuesto de otros que guardan relación entre sí.
9. Tipo de fichero en que los registros se almacenan uno tras otro, de modo que para leer los últimos datos hay que leer también los anteriores.
10. Operación que permite unir varias cadenas de caracteres en una sola.
11. Modelo lógico de Base de Datos que utiliza árboles para su organización.
12. Tipos de árboles cuyos nodos no pueden tener más de dos hijos.

# Sesión 2. Actividades

## ACTIVIDAD 8. Crucigrama. Vocabulario Unidad 1

