



UTN-FRBA

GESTION DE DATOS

ESTRUCTURA DE UN DBMS

DIRECTOR CATEDRA: ING. ENRIQUE REINOSA

CONCEPTOS

- ▶ **BD:** Conjunto de datos interrelacionados que se ajustan a una serie de modelos preestablecidos que recogen información de interés de objetos del mundo real.
- ▶ **DBMS:** Software encargado de gestionar los datos de la BD. Su misión es proporcionar mecanismos de acceso a los datos para almacenar, definir y recuperar información de forma eficiente. Existen además una serie de aplicaciones en torno al DBMS que aportan interfaces sencillos para manejar los datos.

PROPIEDADES

- ▶ Para que un producto se considere un Motor de Base de Datos (DBMS) debe cumplir con determinadas propiedades. Dichas propiedades se agrupan dentro de la sigla **ACID**.
- ▶ **ACID**: es un acrónimo de **A**tomicity, **C**onsistency, **I**solation and **D**urability: Atomicidad, Consistencia, Aislamiento y Durabilidad en español.

ATOMICIDAD

- ▶ **Atomicidad** es la propiedad que asegura que una operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias. Se dice que una operación es atómica cuando es imposible para otra parte de un sistema encontrar pasos intermedios. Si esta operación consiste en una serie de pasos, todos ellos ocurren o ninguno. Por ejemplo, en el caso de una transacción bancaria o se ejecuta tanto el depósito y la deducción o ninguna acción es realizada. Es una característica de los sistemas transaccionales.

CONSISTENCIA

- **Consistencia:** (*Integridad*). Es la propiedad que asegura que sólo se empieza aquello que se puede terminar. Por lo tanto se ejecutan aquellas operaciones que no van a romper las reglas y directrices de *Integridad* de la base de datos. La propiedad de consistencia sostiene que cualquier transacción llevará a la base de datos desde un estado válido a otro también válido. "La Integridad de la Base de Datos nos permite asegurar que los datos son exactos y consistentes, es decir que estén siempre intactos, sean siempre los esperados y que de ninguna manera cambian ni se deformen. De esta manera podemos garantizar que la información que se presenta al usuario será siempre la misma."

AISLAMIENTO

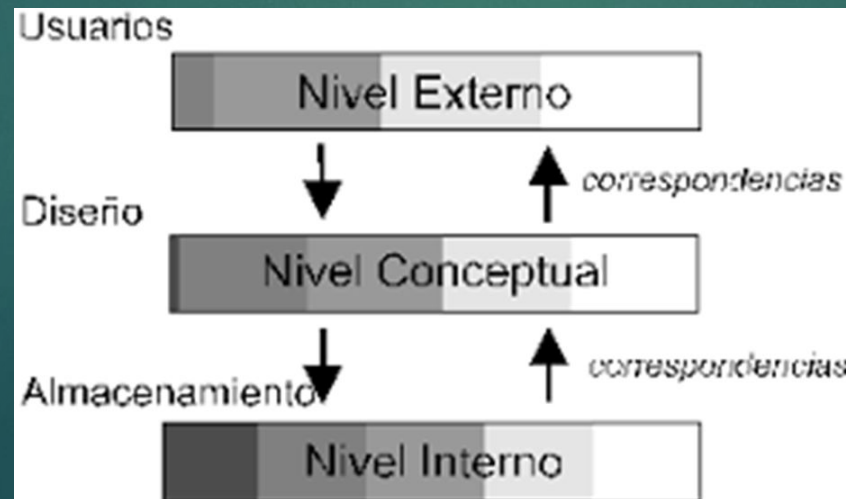
- ▶ **Aislamiento:** Esta propiedad asegura que una operación no puede afectar a otras. Esto asegura que la realización de dos transacciones sobre la misma información sean independientes y no generen ningún tipo de error. Esta propiedad define cómo y cuándo los cambios producidos por una operación se hacen visibles para las demás operaciones concurrentes. El aislamiento puede alcanzarse en distintos niveles, siendo el parámetro esencial a la hora de seleccionar un DBMS.

DURABILIDAD

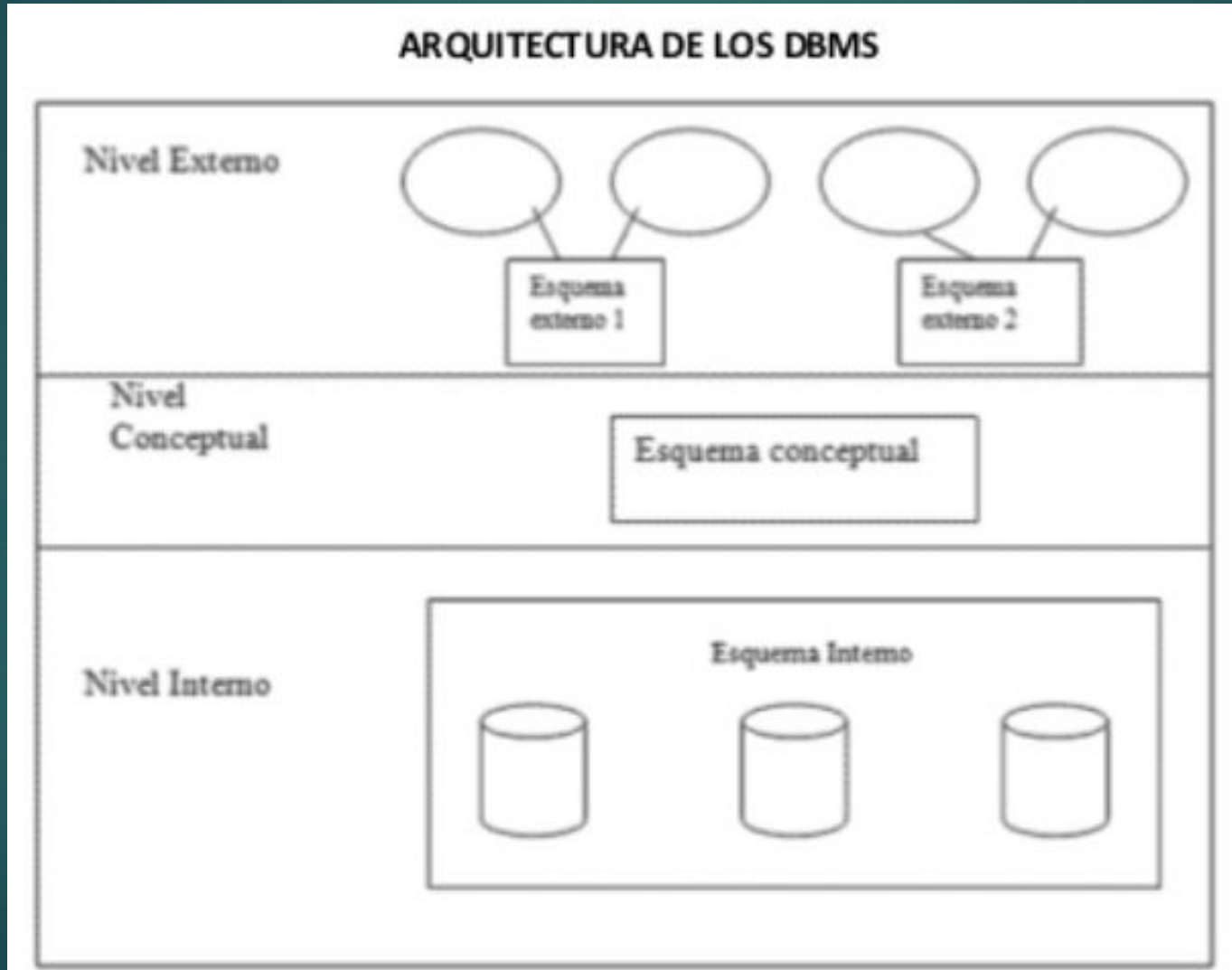
- **Durabilidad:** (*Persistencia*). Esta propiedad asegura que una vez realizada la operación, esta persistirá y no se podrá deshacer aunque falle el sistema y que de esta forma los datos sobrevivan de alguna manera. la **persistencia** es la acción de preservar la información de un objeto de forma permanente (guardado), pero a su vez también se refiere a poder recuperar la información del mismo (leerlo) para que pueda ser nuevamente utilizado.

ARQUITECTURA ANSI DE UN DBMS

- ▶ La Arquitectura de un DBMS se compone de **tres niveles** hoy en la arquitectura de software **conocidas como capas**.
- ▶ Esta arquitectura si bien viene desde la década del 70 es la utilizada actualmente para la mayoría de desarrollo en tres capas.
- ▶ El DBMS está conformado por los siguientes niveles: **interno o físico**, **conceptual o lógico** y **externo o de usuario**.



ARQUITECTURA ANSI DE UN DBMS



ARQUITECTURA ANSI DE UN DBMS

- ▶ **Permite vistas de usuario independientes y personalizadas:** Cada usuario debe ser capaz de acceder a los datos, pero tiene una vista personalizada diferente de los datos. Éstos deben ser independientes: los cambios en una vista no deben afectar a las demás.
- ▶ **Oculta los detalles físicos de almacenamiento a los usuarios:** Los usuarios no deberían tener que lidiar con los detalles de almacenamiento de la base de datos.
- ▶ **El administrador de la base de datos debe ser capaz de cambiar las estructuras:** Modifica la estructura de almacenamiento de la BD sin afectar la vista de los usuarios.
- ▶ **La estructura interna de la base de datos no debería verse afectada por cambios en los aspectos físicos del almacenamiento:** por ejemplo, un cambio a un nuevo disco.

VISTA DESDE EL ALMACENAMIENTO

- ▶ **Nivel externo (Vistas individuales de los usuarios)**

Una vista de usuario describe una parte de la base de datos que es relevante para un usuario en particular. Excluye datos irrelevantes, así como los datos que el usuario no está autorizado a acceder.

- ▶ **Nivel Conceptual (Vista conceptual)**

El nivel conceptual es una forma de describir los datos que se almacenan dentro de la base de datos y cómo los datos están relacionados entre sí. Este nivel no especifica cómo se almacenan físicamente los datos.

- ▶ **Nivel interno (Vista de almacenamiento)**

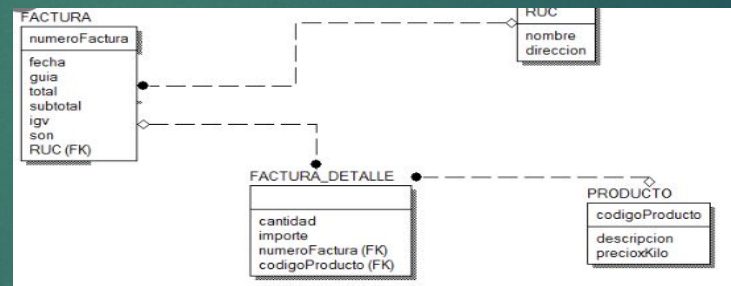
El nivel interno implica la forma en que la base de datos se representa físicamente en el sistema informático. En él se describe cómo los datos se almacenan en la base de datos y en el hardware del equipo.

VISTA DESDE EL ALMACENAMIENTO

► Nivel externo

Alimentación	100	367	369	371	373	210	377
Viáticos	370	377	389	401	413	240	397
Estudios	375	387	409	431	453	270	417
Ocio	380	397	429	461	493	300	437
Hogar	385	407	449	491	533	330	457
Mascotas	390	417	469	521	573	360	477
Servicios	395	427	489	551	613	390	497
Personal	400	437	509	581	653	420	517
Celular	405	447	529	611	693	450	537
Otros	410	457	549	641	733	480	557
TOTAL							

► Nivel Conceptual



► Nivel interno

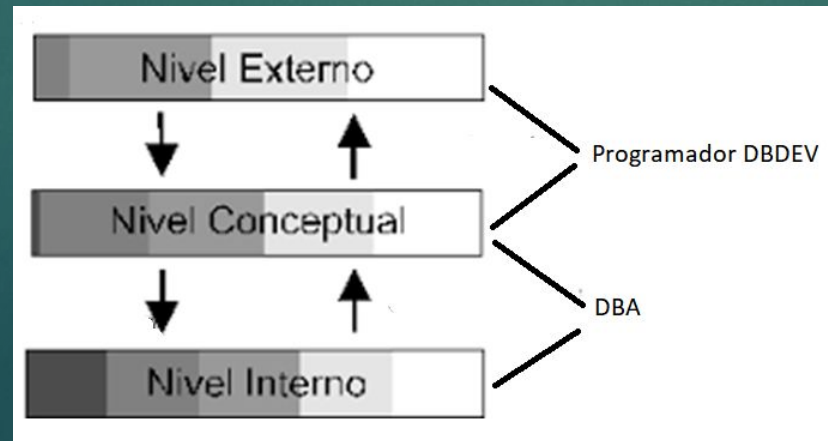
The image shows two side-by-side grid representations of internal storage. The left grid contains several colored blocks (green, cyan, orange) with labels such as "Seguro auto A (enchesión: 1)", "Seguro auto B (enchesión: 1)", "Seguro auto C (enchesión: 1)", and "Seguro auto D (enchesión: 1)". The right grid is mostly empty, representing a different state or a different set of data.

VISTA DESDE LA FUNCIONALIDAD

- ▶ **Nivel Externo:** Es el nivel en el que el usuario interactúa con el DBMS, entendiendo que el usuario puede ser un usuario final o alguna aplicación o lenguaje de programación.
- ▶ **Nivel Conceptual:** Es el nivel donde radica la lógica del DBMS, o sea, donde se definen las reglas de lo que se puede y no se puede hacer. En esta capa se encuentran el analizador sintáctico (parser) y semántico (scanner).
- ▶ **Nivel interno:** Es el nivel que maneja la persistencia de la información en el DBMS, o sea, como se almacena la información.

VISTA DESDE LA FUNCIONALIDAD

- ▶ **Nivel Externo:** El Developer desarrolla aplicaciones desde PL_SQL.
- ▶ **Nivel Conceptual:** El Developer genera modelos lógicos para desarrollar aplicaciones. El DBA define el mejor nivel de diseño desde el punto de vista del DBMS.
- ▶ **Nivel interno:** El DBA define la forma en que se van a almacenar los datos y que se puede hacer o no a nivel programación en función de la configuración del DBMS.



COMPONENTES DE UN DBMS

- ▶ **IPL:** “initial program loader” es un programa de carga inicial que **permite levantar el servicio del DBMS** y disponer la estructura de memoria, cache y disco para el procesamiento de las operaciones.
- ▶ **User Manager:** Es el módulo encargado de **manejar los perfiles,** usuarios y roles de acceso al DBMS.
- ▶ **File Manager:** Es el módulo encargado de la administración ***lógica*** de los archivos que componen al DBMS.
- ▶ **Disk Manager:** Es el módulo encargado de la administración ***física*** de la información persistida en el DBMS.

USER MANAGER

- ▶ El **User Manager** es el encargado de **manejar todos los componentes relacionados con la seguridad del DBMS**. Esto incluye:
 - ▶ Administración de Usuarios
 - ▶ Permisos de acceso
 - ▶ Seguridad Vertical
 - ▶ Seguridad Horizontal

FILE MANAGER

- ▶ El **File Manager** es el encargado de administrar la lógica de almacenamiento de los distintos archivos que componen al DBMS. Al igual que un Sistema Operativo maneja su propia FAT (File Allocation Table). Dentro de sus funciones se encuentran:
 - ▶ Creación de archivos
 - ▶ Eliminación de archivos
 - ▶ Acceso a los archivos
 - ▶ Interconexión con el User Manager para el acceso

DISK MANAGER

- ▶ El **Disk Manager** es el encargado de administrar el espacio físico donde se almacenaran los archivos lógicos administrador por el File Manager. Dentro de sus funciones se encuentran:
 - ▶ Asignación de espacio de almacenamiento
 - ▶ Eliminación de espacio liberado
 - ▶ Acceso a la información física
 - ▶ Comunicación con el SO para el acceso al Disco

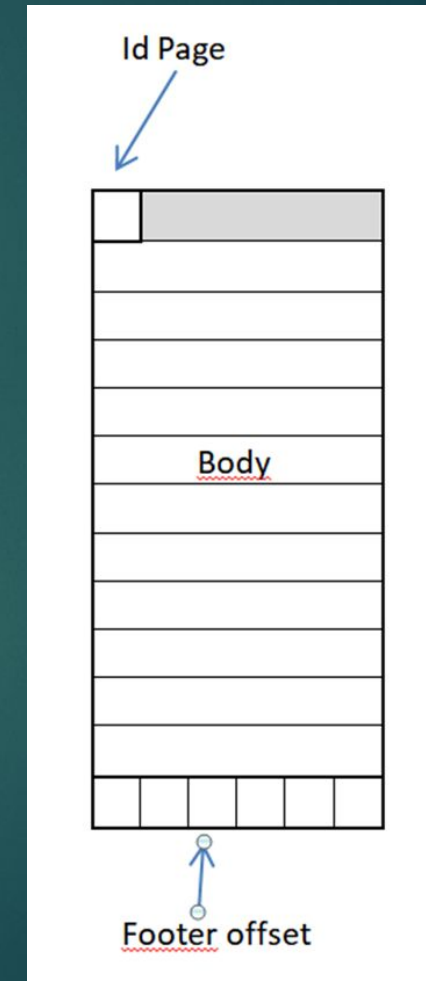
NIVEL INTERNO ALMACENAMIENTO

Existen dos **técnicas de administración de memoria**

- ▶ **Segmentación:** divide la memoria en segmentos, cada uno de los cuales tiene una longitud variable.
- ▶ **Paginación:** divide la memoria en páginas, cada una de las cuales es de longitud fija y de la misma longitud.

PAGINACION

- ▶ **Página:** Una página tiene tres **componentes** el **id page**, el **body page** y el **footer offset**.
 - ▶ **Id:** es la identificación de la página, las cuales se encuentran numeradas y contiguas.
 - ▶ **Body:** es el cuerpo de la página donde se almacena la información. Este cuerpo está **dividido en unidades como registros equivalentes a renglones de una hoja**.
 - ▶ **Footer offset:** es el pie de página **tiene tantas entradas como registros (renglones) que contenga la página**.



PAGINACION

- Los datos se almacenan en **body page** de la página en función como van ingresando, y en el **footer offset** se registra la posición relativa de la fila que se encuentra en esa posición.

9					
123	CARLOS
204	PEDRO
450	JOSE
356	MARIA
0	1	2	3		

Pos	Id	Nombre
0	123	CARLOS
1	204	PEDRO
2	450	JOSE
3	356	MARIA

PAGINACION

En función del tamaño de la fila, puede producirse que sobre espacio en el almacenamiento lo que cual generará una fragmentación.

- ▶ **Fragmentación Externa:** se produce cuando una página es menor al tamaño del cluster mediante el cual asigna espacio el disco y queda un remanente de bytes sin ser utilizados en la grabación
- ▶ **Fragmentación Interna:** es la que se produce cuando la fila a almacenar en el cuerpo de la página es de menor tamaño que la longitud del renglón.

PAGINACION

También puede ocurrir que el tamaño de la fila sea más grande que el tamaño del renglón de la página, lo cual generara que cada fila utilice dos renglones o más para su almacenamiento.

9					
123 CARLOS RO					
DRIGUEZ					
204 PEDRO ALB					
ORNOZ					
450 JOSE MAR					
TINEZ					
0	^	1	^	2	^

+	Pos	Id	Nombre	Apellido
	0	123	CARLOS	RODRIGUEZ
	1	204	PEDRO	ALBORNOZ
	2	450	JOSE	MARTINEZ

LOGICA DE ALMACENAMIENTO

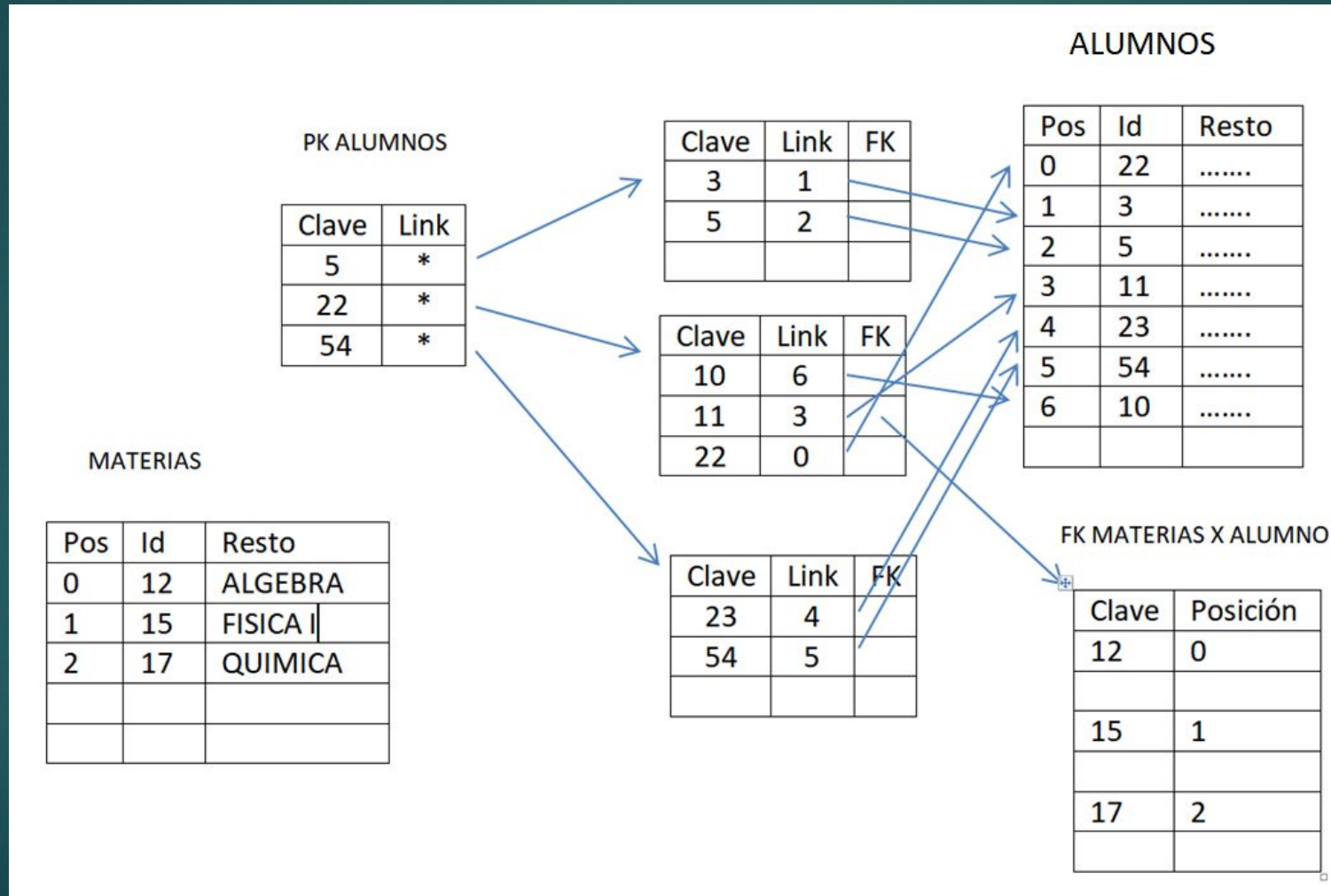
- ▶ **Clustering:** Es la técnica de agrupamiento que permite unificar objetos en función de algún criterio establecido. Existen dos formas de aplicar esta técnica al momento de asignar datos:
 - ▶ **Intra File:** Los objetos se agrupan en función de la pertenencia a un conjunto predeterminado.
 - ▶ **Inter File:** los objetos se agrupan en función a la relación existentes entre los objetos independientemente que pertenezcan a diferentes conjuntos.

LOGICA DE ALMACENAMIENTO

El DBMS utiliza ambas técnicas de almacenamiento para almacenar cosas distintas.

- ▶ **Intra File**: Almacena de esta forma los datos secuenciales, o sea, en una pagina solo coloca filas que se corresponden a una tabla, sin mezclar tablas en la misma página
- ▶ **Inter File**: Almacena de esta forma los índices y PK asociadas a las FK que existan.

ALMACENAMIENTO DE PK Y FK

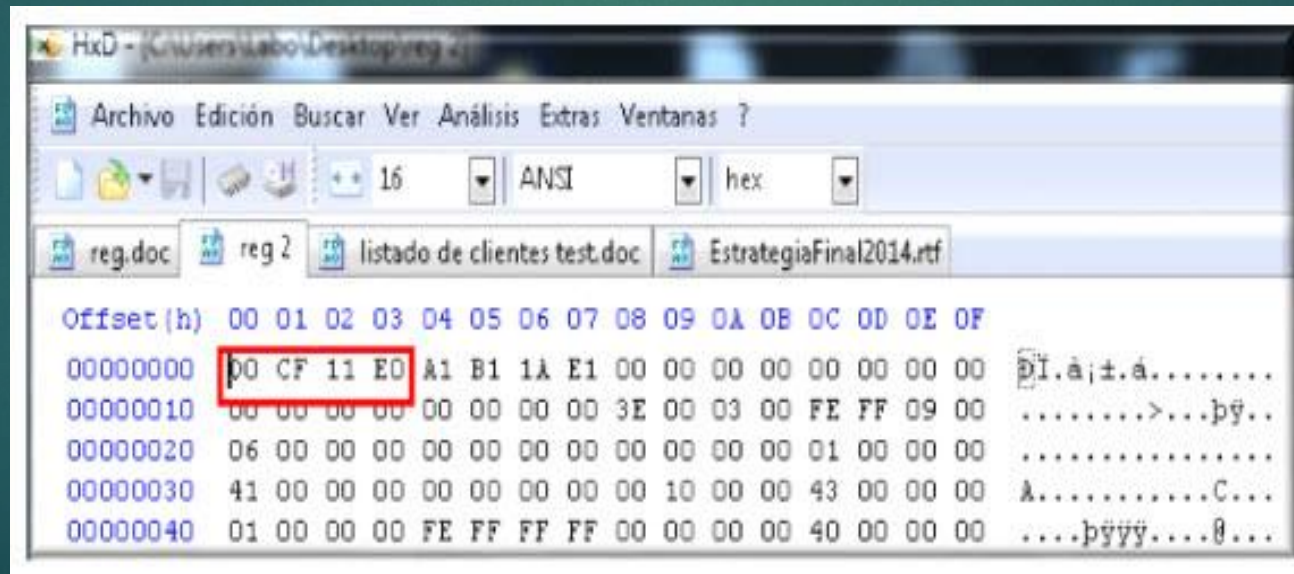


ALMACENAMIENTO DE ARCHIVOS

- ▶ **Archivos:** Dado que el unico formato de archivo existente y manejable por un sistema operativo es el compuesto por un conjunto de caracteres ASCII, es necesario identificar de alguna forma el contenido de los mismos para poder tipificarlos y administrarlos de forma diferencial.
- ▶ **Header:** el cabecero de un archivo es el conjunto de caracteres que se colocan al inicio del mismo y que permiten definir el contenido que continua en el mismo.

TIPOS DE ARCHIVOS

- **Extensión:** La extensión de un archivo esta relacionada con su tipología y de hecho con la aplicación destinada a su apertura y administración. De esta forma **cada tipo de archivo comienza con alguna especificación diferente, lo cual permite identificarlo.**



TIPOS DE ARCHIVOS

Tipo de Archivo	Cabecera	En ASCII
.ZIP	50 4B 03 04	PK
.RAR	52 61 72 21	<u>Rar!</u>
.DOC	D0 CF 11 E0	<u>Đİ.à</u>
.PDF	25 50 44 46	%PDF
.FLV	46 4C 56 01	FLV
.BMP	42 4D F8 A9/ 62 25 / 76 03	BM, <u>BMp%</u> , <u>BMv</u>
.GIF	47 49 46 38 39 61 / 37 61	GIF89a GIF87a
.JPEG	FF D8 FF E0 / FE	JFIF
.PNG	89 50 4E 47	PNG
.SFW	43 57 53 06 / 08	<u>Cws</u>
.MP3	49 44 33 2E /03	ID3
.EXE	4D 5A 50 00 /90 00	MZP / MZ
.DLL	4D 5A 90 00	MZ
Linux <u>bin</u>	7F 45 4C 46	ELF

HEADER DE TABLAS

- El DBMS crea un header para identificar las tablas de la siguiente forma:

```
struct table
{
    long filas;
    long columnas;
    int tamaño_fila;
};

struct columna
{
    char nombre[256];
    char tipo;
    int longitud;
    int decimales;
```

HEADER DE TABLAS EJEMPLO

POS	CODIGO	NOMBRE	SALDO
0	05	CARLOS	12,35
1	03	MARIA	15,23
2	01	JOSE	18.89

struct table

{

long filas;

long columnas;

int tamaño_fila;

};

VALORES

3

3

64

|

struct columna

{

char nombre[256];

char tipo;

int longitud;

int decimales;

boolean notnull;

};

CODIGO

NOMBRE

SALDO

C

C

D

2

50

12

0

0

2

1

0

0