


# Conceptos de Bases de Datos I

1



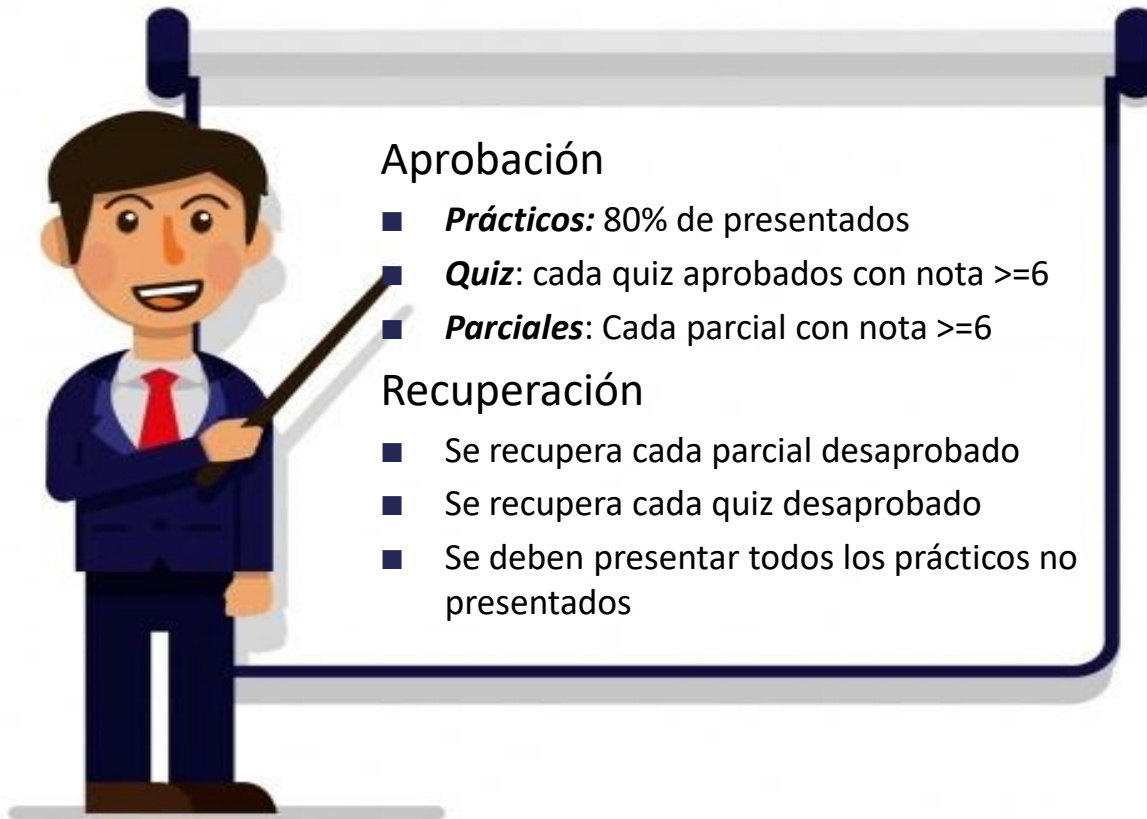
# Conceptos de Bases de Datos I



**Horarios**

- Martes de 15 a 17 hs. Práctica  
(laboratorio A y Fomec)
- Martes de 17 a 19 hs. Teoría  
(Aula 1-0-1)
- Jueves de 17 a 19 hs. Práctica  
(laboratorio A y Fomec)

# Conceptos de Bases de Datos I



# Contenido de la Asignatura:

La primera parte presenta los conceptos universales del mundo de las bases de datos, y su representación, **lo más cercana posible a la realidad**, a través del modelado conceptual y lógico. Revisaremos los fundamentos teóricos del Algebra Relacional que sustenta a las BDs

**En la segunda parte, se iniciará al alumno en el manejo del lenguaje general que permite la administración de Bases de Datos Relacionales. (SQL)**

# Bibliografía:

- ▶ Sistemas de bases de datos. Conceptos fundamentales.  
**Ramez Elmasri/ S.Navathe** – PEARSON EDUCACIÓN (texto principal)
- ▶ Fundamentos y Modelos de Bases de Datos.  
A. De Miguel,M.Piattini – RA-MA
- ▶ Bases de datos, modelos, lenguajes y diseño.  
James L. Johnson – OXFORD
- ▶ Introducción a los sistemas de bases de datos.  
Vol.1. C.J. Date – ADDISON WESLEY

# Definición

Una BD representa **algún aspecto del mundo real**, “minimundo” o “universo de discurso”. Las modificaciones del mini mundo se reflejan en la base de datos. Una BD es un **conjunto de datos lógicamente coherente**, con cierto significado inherente. No es una colección aleatoria de datos. Toda BD se diseña, construye y puebla con datos para un **propósito específico** (Elmasri – Navathe)

Una **BD** es un conjunto de elementos de datos que se **describe a sí mismo** con **relaciones** entre éstos elementos (Johnson)

Una BD es una **colección de datos integrados**, almacenados en **soporte secundario** (no volátil) y con redundancia controlada. Los datos, que ha de ser **compartidos por diferentes usuarios y aplicaciones**, deben mantenerse **independiente de ellos**, y su definición (**estructura**) **única y almacenada junto con los datos** se ha de apoyar en un **modelo de datos**, el cual ha de **permitir captar las relaciones y restricciones del mundo real**. (De Miguel y Piattini)

# Enfoque de BDs

Según el **enfoque de bases de datos** se mantiene un **único almacén de datos** que se define una sola vez y al cual tienen acceso muchos usuarios

**Naturaleza Autodescriptiva:** El sistema no sólo contiene la base de datos misma, sino también una definición o **descripción completa** de la base de datos (meta-datos). Esta descripción se almacena en el **Catálogo del sistema**, que contiene información como la **estructura** de cada archivo, el tipo y el **formato de almacenamiento** de cada elemento de información y diversas **restricciones** que se aplican a los datos

**Abstracción de los Datos:** Hace posible la **independencia** con respecto a los **datos y programas**. La independencia con respecto a los **programas y operaciones** (La **representación conceptual de los datos** que no incluye muchos de los detalles de cómo se almacenan)

# Enfoque de BDs

Datos compartidos y procesamiento de datos multiusuario

El sistema de gestión de bases de datos incluye **software para el control de recurrencia** que asegura que cuando varios usuarios intenten actualizar los mismos datos lo hagan de manera **controlada** para que **el resultado sea correcto y sin interferencias**

**Una base de datos suele tener muchos usuarios, cada uno de los cuales puede requerir una perspectiva o vista diferente de la misma.** Este enfoque proporciona los **mecanismos** para definir diversas vistas. Una **vista**, efectivamente, puede ser un subconjunto de la base de datos o contener **datos virtuales** que se **deriven** de los archivos de bases de datos



# ¿Cómo se “gestiona” una BD?



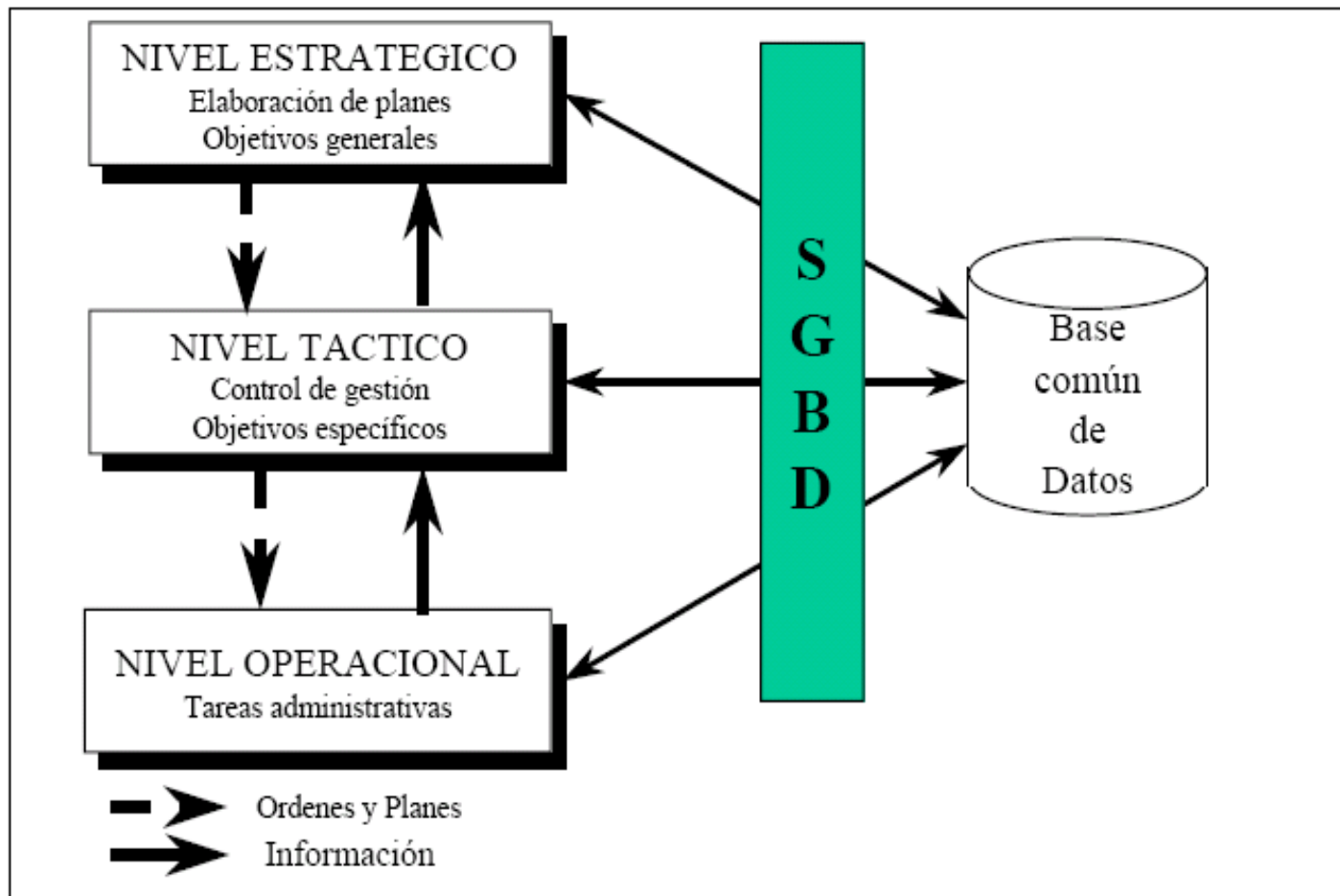
- *¿Qué es un Sistema Gestor de bases de datos?*
- *¿Qué tipos de Usuarios acceden a un sistema de gestión de bases de datos?*
- *¿Cuáles son las características esperables de un DBMS?*

# Definición DBMS

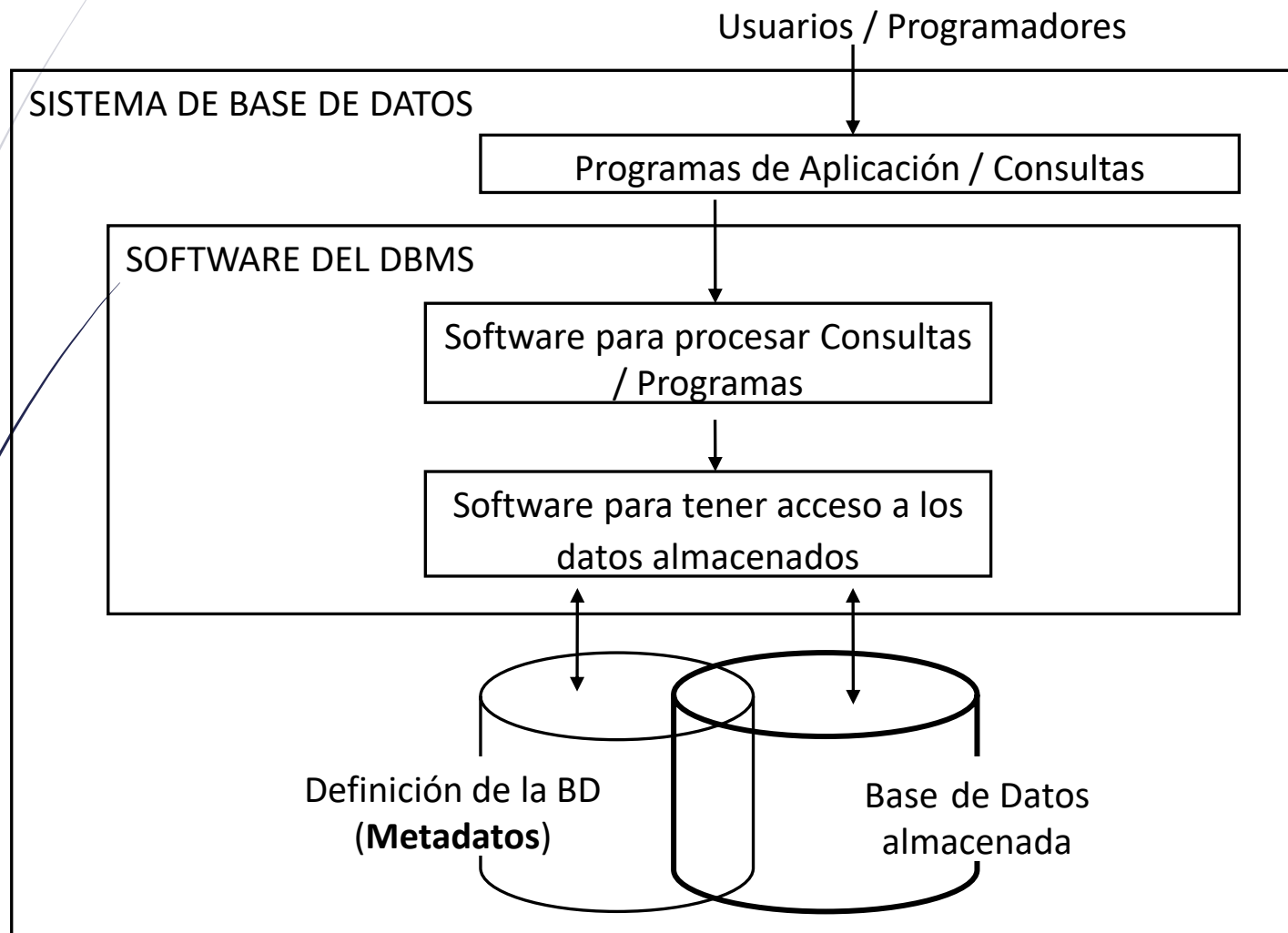
Un **DBMS** (DataBase Management System) es un **conjunto de programas** que permite a los usuarios crear y mantener una BD. Es un sistema de software de propósito general que facilita el **proceso de definir, construir y manipular bases de datos** para diversas aplicaciones (Elmasri – Navathe)

Un DBMS proporciona el método de organización necesario para el **almacenamiento y recuperación flexibles de grandes cantidades de datos**. Es un producto de software que presta soporte al almacenamiento confiable de los datos, pone en marcha las estructuras para **mantener relaciones y restricciones** más funciones que se ocupan de otras tareas, como son el acceso simultáneo, **seguridad y respaldo de datos** (Johnson)

# Sistemas de Gestión de bases de datos (DBMS)



# Entorno simplificado de un Sistema de BD



# Tipos de Usuario de un DBMS

- **Administrador:** Tiene el control sobre el sistema, datos y programas. Administra los recursos compartidos, controlando su uso
- **Diseñadores:** Se encargan de identificar los datos que se almacenarán y de elegir las estructuras apropiadas para representar y almacenar dichos datos
- **Usuarios finales:** Simples o paramétricos. Esporádicos. Avanzados
- **Otros Usuarios:** Implementadores de DBMS, Creadores de Interfases, Operadores y mantenimiento

## Administrador de bases de datos - DBA

1

- Coordina el trabajo de los diseñadores respecto a la Definición del Esquema, de la estructura de almacenamiento y los métodos de acceso.

2

- Supervisa la modificación del esquema y la organización física.

3

- Autoriza el acceso a la base de datos, de coordinar y vigilar su empleo.

4

- Adquiere recursos necesarios de software y hardware.

5

- Respecto a la Concesión de la autorización para el acceso a los datos, es responsable de violaciones de seguridad. También de respuestas lentas del sistema.

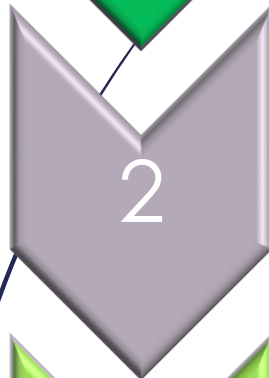
6

- Especifica las restricciones de Integridad.

# Diseñadores



- Se encargan de identificar los datos que se almacenarán y de elegir las estructuras apropiadas para representar y almacenar dichos datos.



- Casi siempre interactúan con cada uno de los grupos de usuarios potenciales



- y desarrollan una vista que satisfaga los requerimientos de datos y de procesamiento para ese grupo

# Usuarios Finales

Son las personas que necesitan tener acceso a la bd para consultarla, actualizarla y generar informes. Existen varias categorías:

## Simples o paramétricos

- constituyen la mayor porción de los usuarios finales y su principal función gira en torno a las consultas y actualizaciones constantes de la base de datos, utilizando tipos standard de estas operaciones que se han programado y probado con mucho cuidado

## Esporádicos

- tienen acceso de vez en cuando y posiblemente requieran información diferente en cada ocasión. Aprenden a emplear sólo unos cuantos recursos. Suelen ser gerentes de nivel medio o alto

## Avanzados

- son los que conocen en forma cabal los recursos del DBMS y lo utilizan para satisfacer requerimientos complejos

NO necesitan saber mucho de los recursos que proporcion a el DBMS



# Otros Usuarios

Personas a las que no les interesa la base de datos misma. Tienen que ver con el **diseño, creación y operación** del software y entorno del sistema del DBMS.

Diseñadores e  
Implementado-  
res del DBMS

- éstos se encargan de diseñar e implementar los módulos e interfaces del DBMS en forma de “paquetes de software”

Creadores de  
Herramientas

- se ocupan de diseñar e implementar paquetes especiales llamados herramientas. Las herramientas son paquetes de software que facilitan el diseño y el empleo de los sistemas de bases de datos, y que ayudan a elevar el rendimiento.

Operadores y  
Personal de  
Mantenimiento

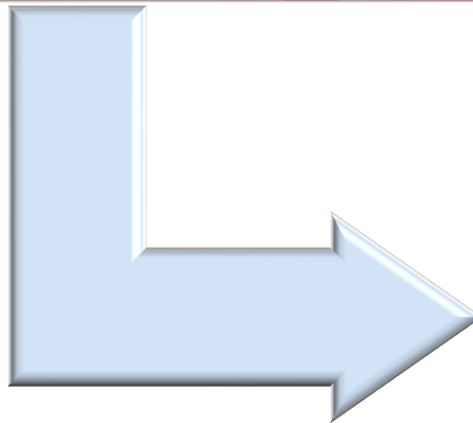
- tiene a su cargo el funcionamiento y mantenimiento reales del entorno de hardware y software del sistema de bd

# ¿Cuáles son las características esperables de un DBMS?

- **Control de la Redundancia**
- **Cumplimiento de las restricciones de Integridad**
- **Sharing de datos y archivos**
- **Restricción de Accesos No autorizados**
- **Suministro de múltiples interfaces con los usuarios**
- **Respaldo y recuperación**

# Control de Redundancia

En los sistemas sin bases de datos, cada aplicación tiene sus **propios archivos privados** para manejar sus aplicaciones de procesamiento de datos, lo que provoca considerable redundancia en los datos almacenados, **con el consecuente desperdicio de espacio de almacenamiento.**

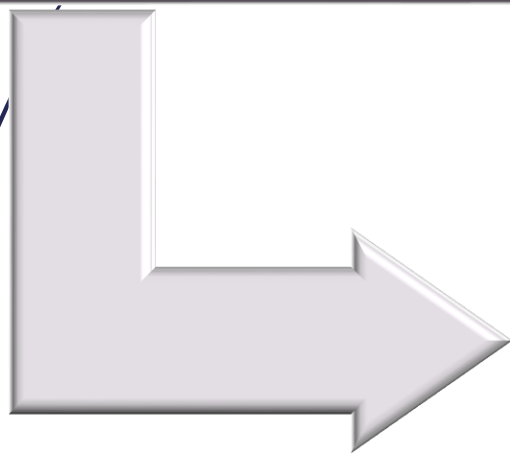


A veces, y no pocas, esta redundancia provoca varios **problemas**, no sólo el del almacenamiento (peor aún si las bd's son grandes), sino también una **duplicación del trabajo** porque es necesario realizar varias **actualizaciones lógicas** (como introducir los datos de un nuevo registro) tantas veces como archivos contengan una estructura igual (igual entidad).

# Cumplimiento de las restricciones de Integridad

Es posible que los archivos que representen los mismos datos se tornen inconsistentes **porque la actualización se haya aplicado a algunos archivos y a otros no**.

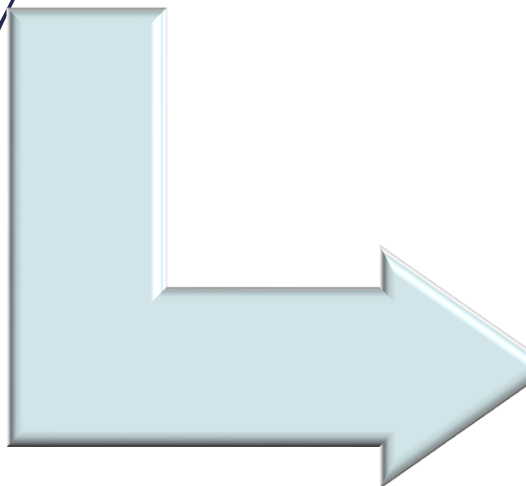
Incluso si la actualización se aplica a *todos* los archivos apropiados, persiste la posibilidad de que los datos relacionados también resulten con problemas de consistencia porque cada grupo de usuarios aplica las actualizaciones de manera independiente.



La inconsistencia entre varias entradas que supuestamente representan el mismo “hecho” es un ejemplo **de falta de integridad**. Con el enfoque de bases de datos, las vistas de los diferentes grupos de usuarios se integran durante el diseño de la misma, siendo almacenados en un mismo lugar, evitando inconsistencias y ahorrando espacio de almacenamiento.

# Sharing de datos y archivos

El **compartir archivos** implica no sólo que las aplicaciones ya existentes pueden acceder a la misma información de la base de datos, sino **también** que se pueden desarrollar aplicaciones nuevas para trabajar con los *mismos datos* almacenados.



Dicho de otro modo, es posible **satisfacer las necesidades** de información de las aplicaciones nuevas **sin tener que almacenar datos adicionales.**

# Restricción de Accesos No autorizados

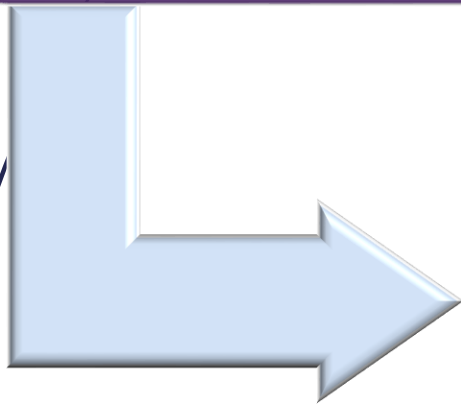
Cuando muchos usuarios comparten una misma base de datos, **es probable que no todos tengan autorización para tener acceso** a toda la información que contiene

Al tener **jurisdicción completa** sobre la base de datos, el DBMS puede:

Asegurar que el acceso a la base de datos sea **sólo a través de canales apropiados y definir las verificaciones de seguridad** por realizar cuando se intente acceder a información delicada.

# Suministro de múltiples interfaces con los usuarios

Puesto que muchos tipos de usuarios con diversos niveles de conocimientos técnicos utilizan las bases de datos, **el DBMS debe ofrecer diferentes interfases.**

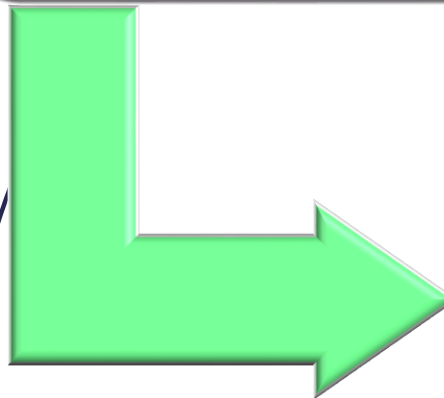


Entre éstas podemos mencionar:

- los lenguajes de consulta para usuarios esporádicos,
- las interfases del lenguaje de programación para programadores de aplicaciones,**
- las formas y los códigos de órdenes para los usuarios paramétricos y
- las interfases controladas por menús y en lenguaje natural para los usuarios autónomos.**

# Respaldo y recuperación

Todo DBMS debe contar con **recursos para recuperarse de fallos de hardware o software**. Para ello está el sistema de respaldo y recuperación.



Por ejemplo, si el *sistema falla mientras se está ejecutando un complejo programa de actualización*, este subsistema se asegurará que la base de datos se **restaure al estado en el que estaba antes de que comenzara la ejecución del programa**



# Funciones de un DBMS

Una forma de caracterizar la función general del DBMS es decir que ***constituye*** la interfaz entre el usuario y el sistema de base de datos.

- Definición
- Manipulación
- Seguridad e integridad de los datos
- Recuperación y concurrencia de los datos
- Diccionario de Datos
- Desempeño

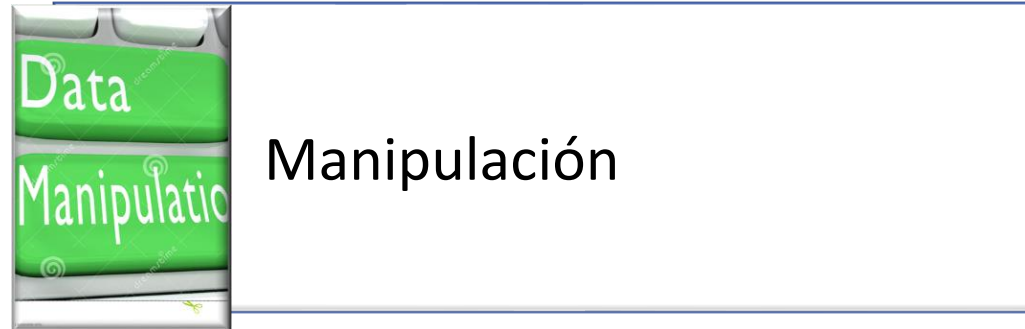
# Funciones de un DBMS



## Definición

El DBMS debe ser capaz de aceptar definiciones de datos (esquemas externos, el esquema conceptual, el esquema interno, y todas las correspondencias asociadas), es decir, debe incluir **componentes procesadores de lenguajes** para cada uno de los diversos lenguajes de definición de datos (DDL)

# Funciones de un DBMS



El DBMS debe ser capaz de *atender las solicitudes del usuario* para extraer, y quizá poner al día, datos que ya existen en la base de datos, o para agregar en ella datos nuevos, es decir, debe incluir un componente procesador de lenguaje de manipulación de datos (DML).

Una solicitud en DML puede ser planeada, es decir, prevista con mucho tiempo antes de su primera ejecución, por lo que el DBA afinó el diseño físico de la base de datos a fin de garantizar un buen desempeño. Y también puede ser no planeada, es decir, que surge de improviso, por lo que el diseño físico de la base de datos puede ser o no ideal para la solicitud específica de que se trate.

# Funciones de un DBMS



Seguridad e integridad de los datos

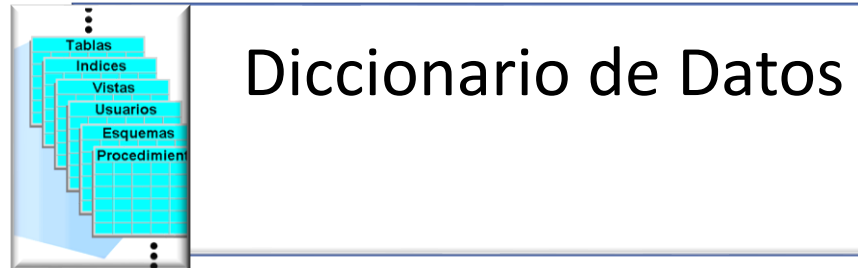
El DBMS debe *supervisar las solicitudes* de los usuarios y *rechazar los intentos* de violar las medidas de seguridad e integridad definidas por el DBA.



Recuperación y concurrencia de los datos

El DBMS debe *cuidar del cumplimiento de ciertos controles de recuperación* y concurrencia, o en su defecto delegarlo a algún componente de software relacionado como el Administrador de transacciones.

# Funciones de un DBMS



El DBMS debe incluir una función de diccionario de datos, entendiéndose éste como **una base de datos misma**, no del usuario, sino **del sistema**. El contenido del diccionario puede considerarse como datos acerca de los datos (Metadatos).



## Desempeño

El DBMS debe **ejecutar todas las funciones** descritas en la forma más **eficiente** posible

# Componentes funcionales - DBMS

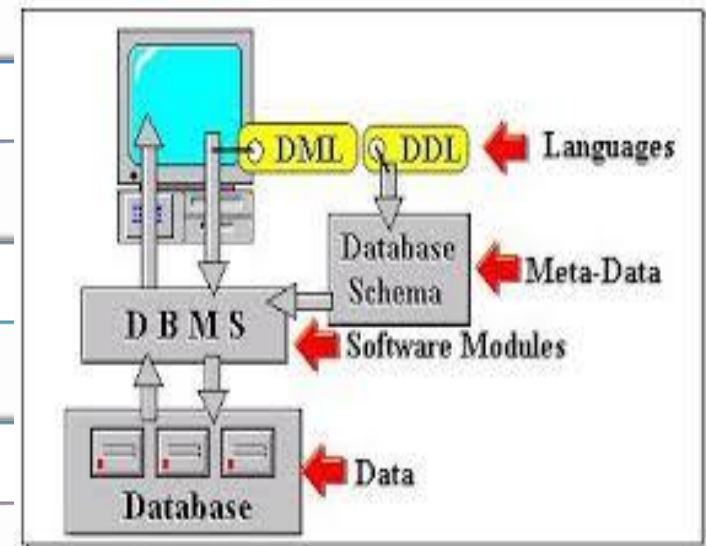
Gestor de datos almacenados del DBMS  
(el Gestor de Archivos),

Compilador de DDL

Catálogo del sistema

Procesador de la base de datos

Compilador de consultas



# Gestor de datos almacenados de un DBMS

El acceso a disco suele controlarlo primordialmente el Sistema operativo (S.O.) que programa la entrada/salida del disco.



El **GDA** controla el acceso a la información almacenada en el disco, ya sea que forme parte de la **bd** o del **catálogo**.

Este último puede emplear servicios básicos del S.O. para transferir los datos de bajo nivel entre el disco y la memoria principal del computador, pero controla otros aspectos de la transferencia de datos, como el manejo de las áreas de almacenamiento intermedio (buffers) en la memoria principal.



Una vez que los datos estén en dicho **almacenamiento intermedio**, otros módulos del DBMS podrán **procesarlos**.

# Compilador de DDL y Catalogo del Sistema



## Compilador de DDL

procesa las definiciones de esquemas, especificadas en el DDL, y almacena las descripciones de los esquemas (metadatos) en el catálogo del DBMS.

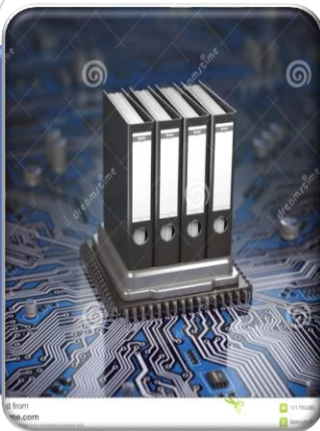


## Catálogo

Contiene información como los nombres de los archivos y de los elementos de información, los detalles de almacenamiento de cada archivo, la inf. de correspondencia entre los esquemas y restricciones.



# Procesador de Base de Datos y Compilador de Consultas



## Procesador de la base de datos

- Se encarga, en tiempo de ejecución, de los accesos a ella durante la ejecución; recibe las operaciones de obtención o de actualización y las ejecuta sobre la base de datos.
- El acceso a disco se tiene mediante el Gestor de datos almacenados.



## Compilador de consultas

- Maneja las consultas de alto nivel que se introducen interactivamente. Analiza la sintaxis y el contenido de las consultas y luego genera llamadas al procesador en tiempo de ejecución para atender la solicitud.

# Lenguajes de Bases de Datos

## **DDL**

*Lenguaje de Definición de Datos*

- Permite especificar los esquemas conceptual e interno y las correspondencias entre ambos, sobretodo en los DBMS en que no se mantiene una separación estricta de niveles. En ellos en DBA y los diseñadores utilizan este mismo lenguaje.

## **SDL**

*Lenguaje de Definición de Almacenamiento*

- Se utiliza para especificar solamente el esquema interno. Las correspondencias entre los dos esquemas se pueden especificar en cualquiera de los dos lenguajes (SDL o DDL).

## **VDL**

*Lenguaje de Definición de Vistas*

- Se utiliza para especificar las vistas del usuario y sus correspondencias con el esquema conceptual. Es imprescindible en una verdadera arquitectura de tres esquemas

## **DML**

*Lenguaje de manipulación de Datos*

- Una vez que se han compilado los esquemas de la base de datos y que en ésta se introdujeron datos, los usuarios requerirán algún mecanismo para manipularla (recuperar, insertar, eliminar y modificar datos). Para estos fines el DBMS ofrece este lenguaje

# ¿Cómo empezar?

- *Cómo “representar” una porción de realidad en una base de datos?*
- *Cómo lograr que esta representación sea lo más fiel posible?*
- *Para qué sirve un modelo?*



# ¿Qué es un modelo?

Construcción mental a partir de la realidad en la que se reproducen los principales componentes y relaciones del segmento de la realidad analizada

## ¿Qué es un modelo de datos?

- Conjunto de herramientas conceptuales para describir la representación de la información en términos de datos. Los modelos de datos comprenden aspectos relacionados con Estructuras y tipos de datos, Operaciones y restricciones (Dittrich)
- Conjunto de Conceptos, reglas y convenciones que permiten describir y manipular los datos de la parcela de cierto mundo real que deseamos almacenar en la base de datos (De Miguel)

# ¿Qué es un modelo?

Entonces...

Un **modelo de datos** es un conjunto de conceptos que pueden servir para describir la estructura de una base de datos.

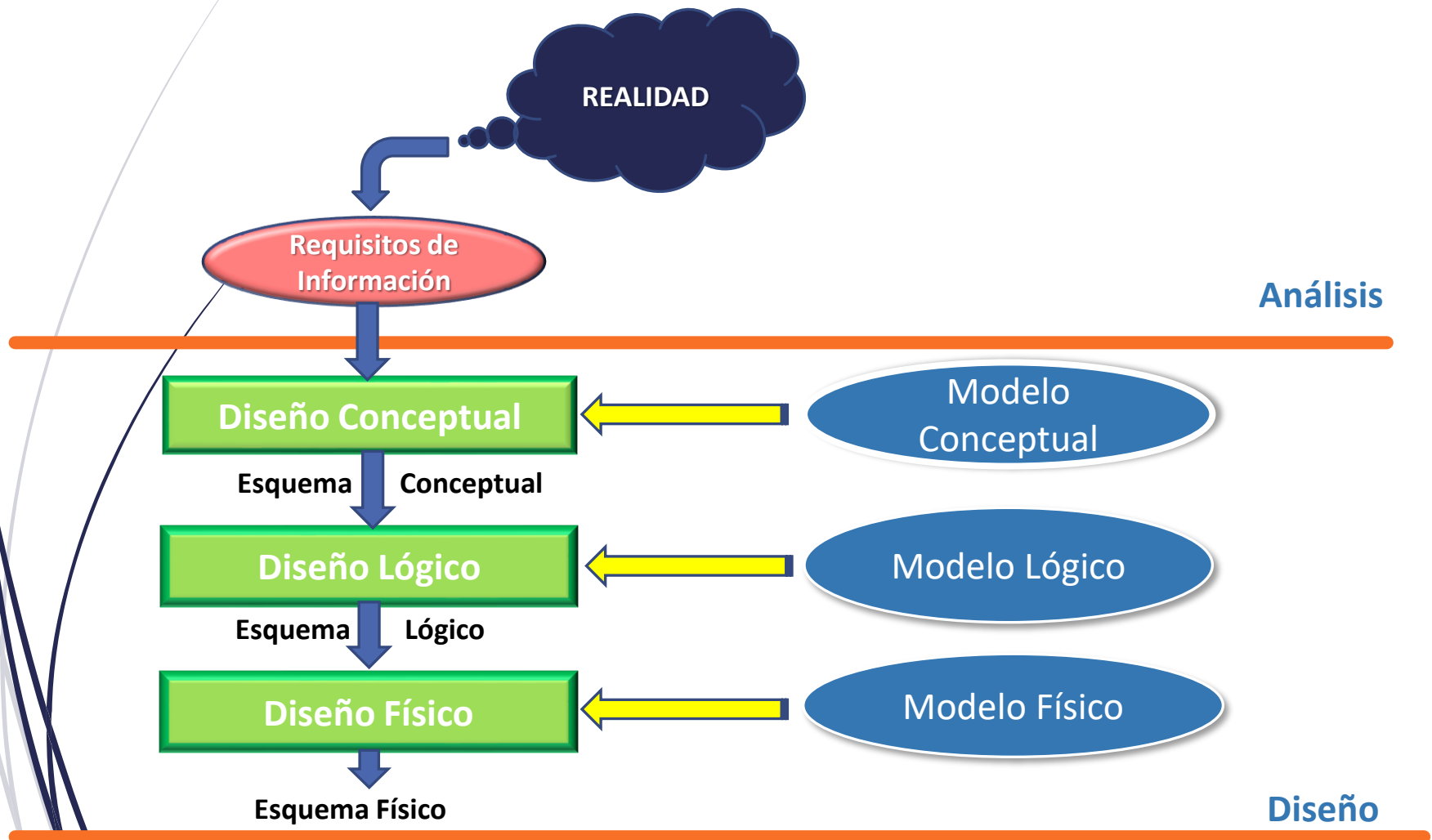
## **Modelo de datos**

***“colección de herramientas conceptuales para describir los datos, relaciones entre ellos, semántica asociada a los datos y restricciones de consistencia”.***

# Modelos de Datos: categorías

- **Modelos Conceptuales** o de alto nivel: disponen de conceptos muy cercanos al modo como la generalidad de los **usuarios perciben los datos**. Uno de los más populares el de **Entidad-Relación**
- **Modelo lógico o Implementable**: Estos ocultan algunos detalles de cómo se almacenan los datos, pero pueden implementarse de manera directa. Son los más utilizados en los DBMS actuales, Representan sus datos valiéndose de estructuras de registros, por lo que a veces se les denomina modelos de datos basados en registros
- **Modelo Físicos** o de bajo nivel: proporcionan conceptos que describen los detalles de cómo se almacenan los datos en el servidor, al representar información como los formatos y ordenamientos de los registros y los caminos de acceso. Un camino de acceso es una estructura que hace eficiente la búsqueda de registros específicos de la base de datos, casi siempre están dirigidos a especialistas en computación, no a los usuarios finales

# Etapas del Diseño de Base de Datos



# Modelos de Datos

## Conceptuales

Disponen de conceptos muy cercanos al modo como la generalidad de los **usuarios perciben los datos**

Uno de los más populares el de **Entidad-Relación**

## Lógicos o Implementables

Entre estos dos extremos hay una clase de modelos de datos cuyos conceptos pueden ser entendidos por los usuarios finales aunque no están demasiado alejados de la forma en que los datos se organizan dentro del computador

Estos ocultan algunos detalles de cómo se almacenan los datos, pero pueden implementarse de manera directa. Son los más utilizados en los DBMS actuales,

Representan sus datos valiéndose de estructuras de registros, por lo que a veces se les denomina modelos de datos basados en registros

## Físicos

Proporcionan conceptos que describen los detalles de cómo se almacenan los datos en el computador, al representar información como los formatos y ordenamientos de los registros y los caminos de acceso

Un camino de acceso es una estructura que hace eficiente la búsqueda de registros específicos de la base de datos

Casi siempre están dirigidos a **especialistas en computación**, no a los usuarios finales.



# Retomando los Modelos implementables también llamados lógicos...

Estos modelos se llaman así porque la bd está estructurada en registros de formato fijo de varios tipos. Cada tipo de registro define un nro fijo de campos de longitud fija.


## Basados en Registros

- Jerárquico
- De Red
- Relacional

## Basados en Objetos

- Modelo orientado a Objetos

# Modelo Relacional de Bases de Datos



Emplea **tablas** para organizar los elementos de datos. Cada *tabla corresponde a una entidad de aplicación*, y *cada fila representa una instancia de ésa entidad*

A pesar de las complicaciones que resultan de relaciones en donde aparecen muchas filas de muchas tablas, este sencillo mecanismo ***soporta relaciones sin recurrir a estructuras auxiliares***, como son listas o índices enlazados

El **lenguaje de consulta estructurado (SQL)** sirve como interfaz uniforme para los usuarios, proporcionando un *conjunto de expresiones standard para almacenar y recuperar datos*

# Modelos Orientados a Objetos



Es el primer modelo pos\_relacional y ofrece una alternativa para las inserciones en tablas para mantener relaciones. Al igual que el anterior, se basa en una colección de objetos.

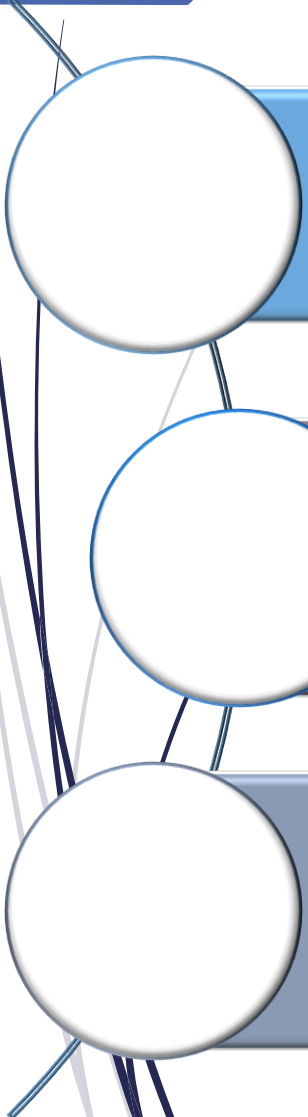


Representa una entidad de aplicación como una clase; una clase captura los atributos y el comportamiento de la entidad. Las instancias individuales de cada clase son llamadas Objetos



Dentro de un objeto, los atributos de clase toman valores específicos, sin embargo las pautas de comportamiento son compartidas por todos los objetos. Estas pautas forman parte del objeto, denominándose métodos (parte de códigos que operan sobre el objeto).

# Concepto de Estado



El **Estado** de la base de datos es el conjunto de datos, llamados **ocurrencias o ejemplares** que la misma posee en un *determinado momento*

Cuando se define una nueva base de datos, el estado correspondiente es **vacío**, sin datos. Cuando cargamos éstos por primera vez, pasa al **estado inicial**. De ahí en adelante, siempre que se aplique una instrucción de actualización a la base de datos **tendremos otro estado**.

El DBMS se encarga en parte de asegurar que todos los estados de la base de datos sean estados válidos; esto es, **que satisfagan la estructura y las restricciones especificadas en el esquema**. Por eso es tan importante especificar un esquema correcto

# Arquitectura de tres niveles

Para poder contar con las características deseables de un DBMS

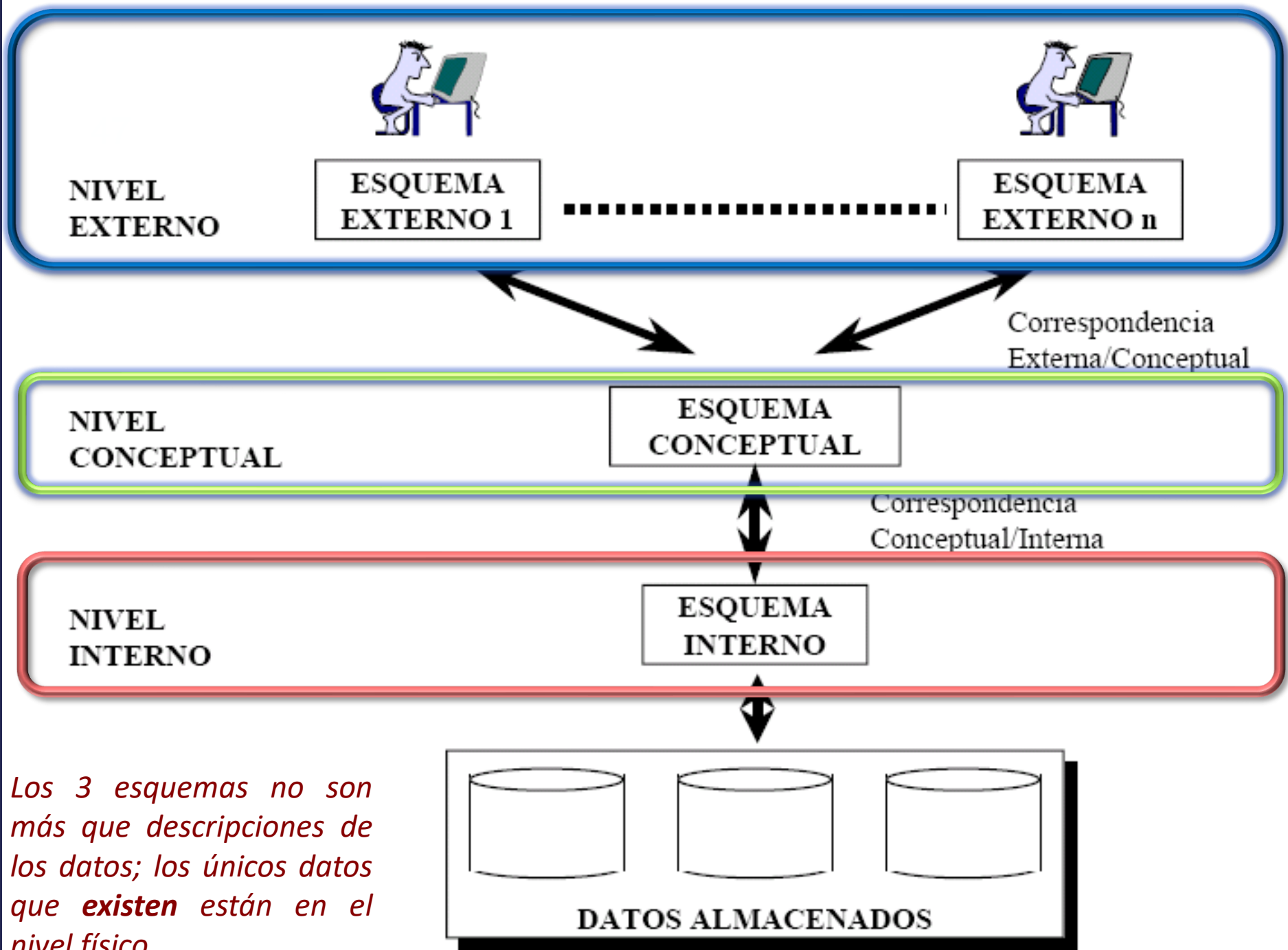
## Arquitectura de tres esquemas

Arquitectura para los sistemas de bases de datos conocida como ANSI / SPARC

independencia  
con respecto a  
programas y  
datos

el manejo de  
múltiples  
vistas de  
usuario

el empleo de  
un catálogo  
para  
almacenar el  
esquema de la  
base de datos



*Los 3 esquemas no son más que descripciones de los datos; los únicos datos que **existen** están en el nivel físico*

48

## Nivel Interno

describe la estructura física de almacenamiento de los datos

- El **esquema interno** emplea un modelo físico de los datos, y describe todos los detalles para su almacenamiento, así como los caminos de acceso para la base de datos

En este nivel podemos usar un modelo de datos de implementación

## Nivel Conceptual

describe la estructura de toda la base de datos para una comunidad de usuarios

- El **esquema conceptual** oculta los detalles de las estructuras físicas de almacenamiento y se concentra en describir entidades, tipos de datos, vínculos, operaciones de los usuarios y restricciones

## Nivel Externo (o de vistas)

incluye **varios** esquemas externos o vistas de usuario

- **Cada esquema externo** describe la parte de la base de datos que interesa a un grupo de usuarios determinado, y oculta a ése grupo el resto de la base de datos.



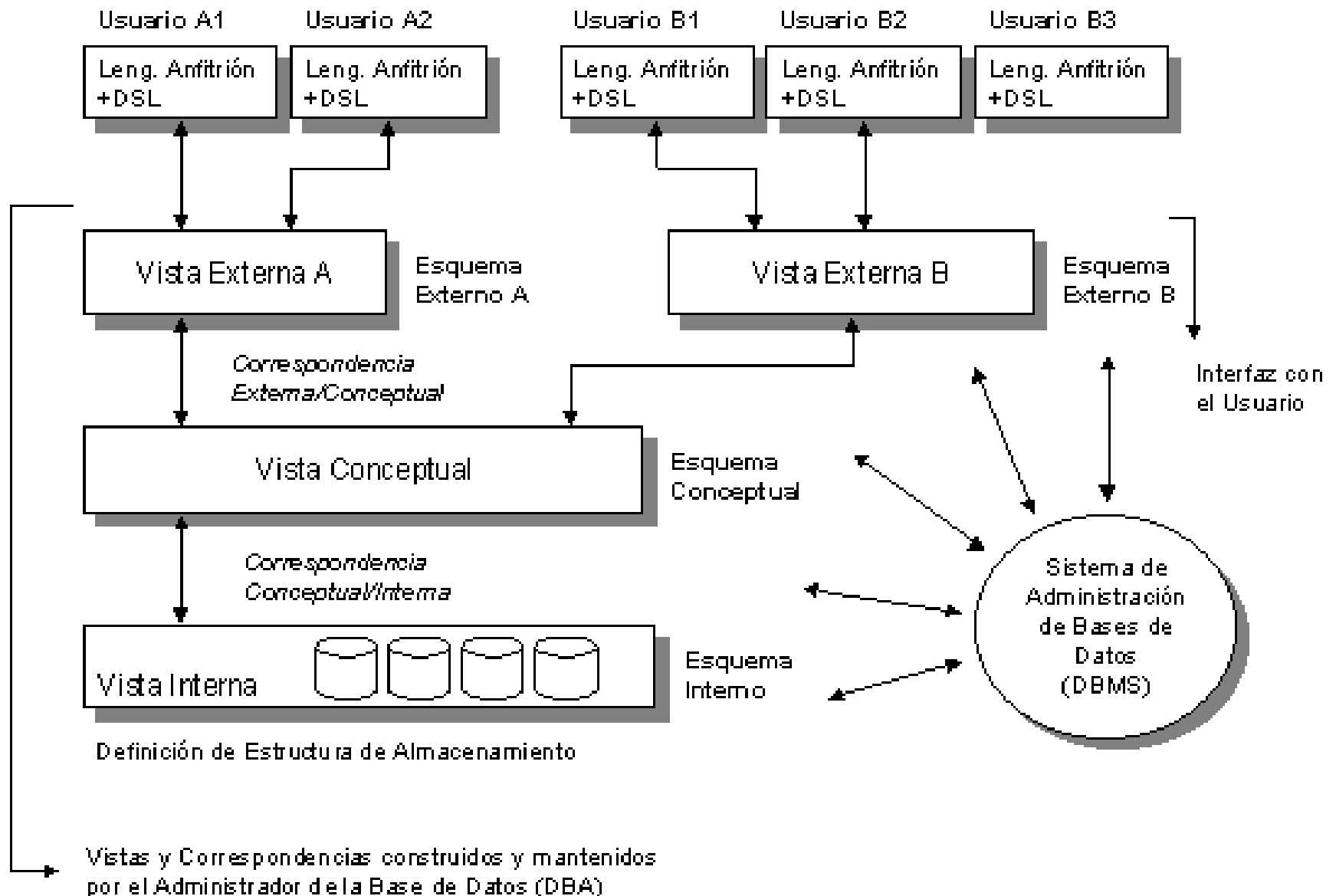


Figura 1.3.- Arquitectura de una Base de Datos

# Independencia con respecto de los datos



Capacidad para modificar el esquema de un nivel de la base de datos **SIN** tener que modificar el esquema del nivel inmediato superior.

Podemos hablar de **Independencia Lógica** e **Independencia Física**

- Capacidad de *modificar el esquema conceptual sin tener que alterar los esquemas externos ni los programas de aplicación.*
- Por ejemplo, podemos ampliar la base de datos (añadiendo un nuevo registro o elemento de información), o para reducir la misma (eliminando un tipo de registro o un elemento de información), en ambos casos modificamos el esquema conceptual, lo que no deberá afectar los programas de aplicación que deberán funcionar igual que antes, luego de esta reorganización lógica. Sólo será preciso modificar la definición de la vista y las correspondencias

## Independencia Física respecto de los datos

- Capacidad de ***modificar el esquema interno sin tener que alterar el esquema conceptual o el de Vistas.***
- A veces es preciso modificar el esquema interno por la necesidad de reorganizar ciertos archivos físicos, por ejemplo, al crear estructuras de acceso adicionales, a fin de mejorar el rendimiento de las operaciones de obtención o actualización. Si la base de datos aún contiene los mismos datos, no deberá ser necesario modificar el esquema conceptual