

# BASES DE DATOS

## TEMA 2

Arquitectura de un Sistema de Gestión de Bases de Datos

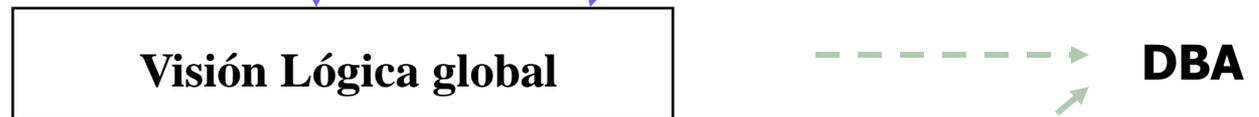
# 2.1 y 2.2 Arquitectura en 3 niveles

*Independencia -> ANSI/SPARC (1975)*

**Nivel externo (Todas las percepciones de la BD)**



**Nivel conceptual (Abstracción global de la BD que integra todas las vistas)**



**Nivel Físico (Cercana a la estructura de almacenamiento físico de la BD)**



## 2.3. Lenguajes de una base de datos

### Definiciones.

DSL: (Data System Language) Implantado en el SGBD:

- ❑ DDL: Lenguaje orientado a la **definición** de datos.
- ❑ DML: Lenguaje orientado a la **manipulación** de datos.
- ❑ DCL: Lenguaje de **control** de datos.
- ❑ Lenguajes anfitrión (Host Lenguaje): lenguaje de aplicación de alto nivel que incluye al DSL como una ampliación.

### Recomendaciones ANSI/SPARC y realidad actual.

- ❑ Deberán existir DCL, DDL y DML para cada nivel
- ❑ En la práctica las sentencias son distinguibles a nivel de categoría pero la separación de los lenguajes conforme a los niveles de la arquitectura se establece de acuerdo con el tipo de sentencia que se puede ejecutar cada usuario según privilegios.
- ❑ Deberá haber lenguajes anfitriones para ser usados a nivel conceptual y externo

## 2.3. Lenguajes de una base de datos

Acoplamiento entre el lenguaje de aplicación y el de datos.

**Lenguaje Anfitrión:** empleado para desarrollar sobre el SO anfitrión.

- Propósito General ( Java ó C++ ) ó específicas de BD (Developer, PowerBuilder ó Delphi).
- Débilmente Acoplados. Se distingue la separación entre ambos. Se dota al leng. de prop. general de mecanismos para interactuar con el DSL de la BD. Implementación : APIs, ODBC, JDBC y DSL Inmerso.
- Fuertemente Acoplados. Se parte del DSL y se extiende con estructuras de control, entrada/salida o de interfaz. Implementación : PL/SQL.

## 2.3. Lenguajes de una base de datos

### Situación actual:

- DML independiente del DBMS, aparición de estándares (SQL)
- Aparición de herramientas para el desarrollo de aplicaciones: diseñadores de informes, diseñadores de pantallas, etc...



### Lenguajes de 4ª Generación

- Usuarios no-terminales y DBAs comparten DDL y lenguajes anfitriones. No hay diferencias entre lenguaje conceptual, externo e interno.
- Desarrollo de sistemas en conexión con BD (JDBC, Developer2000 etc..)
- Desarrollo de nuevas formas de almacenamiento y acceso orientadas a la toma de decisiones (Data Warehousing)

## 2.4, 2.5 y 2.6 Niveles Externo, conceptual e interno.

**Esquema:** descripción de una base de datos interpretable por el SGBD. Los esquemas deben estar **almacenados en el propio SGBD**.

**Esquema externo:** describe la estructura lógica de la vista de una base de datos que necesitan una o varias aplicaciones concretas. El nivel Externo nos proporciona las distintas perspectivas de una misma BD mediante los esquemas externos. Estos se crean el DDL del nivel externo. Los esquemas externos se crearan basándonos en el esquema conceptual o en otros esquemas externos.

**Esquema conceptual:** describe la estructura lógica global de la base de datos mediante **un modelo abstracto de dato comprensible por el SGBD**. Integra todos los esquemas externos. Debe incluir:

- Descripción de atributos.
- Descripción de entidades y conexiones.
- Restricciones de integridad asociadas a la semántica.

**Esquema interno:** Representación abstracta de la estructura de almacenamiento proporcionada por el SO sobre el que se ejecuta el SGBD. Describe la estructura interna, ficheros de registros almacenados, campos que componen los registros, formas de acceso, etc..

El SGBD proporcionará mecanismos para garantizar la inmutabilidad (independencia) de los esquemas superiores en caso de cambios en estructuras mas internas. Mediante transformaciones entre niveles.

## 2.7. Transformaciones entre niveles.

*La conexión entre esquemas se hace mediante correspondencias (conceptual/interna, externa/conceptual) que deben estar **almacenadas y ser interpretables por el SGBD.***

**Conceptual/interna** preserva la independencia física. Implementada por el SGBD. Aspectos gestionados por el DBA (índices y clusters).

Registros y Campos lógicos  Registros y campos almacenados

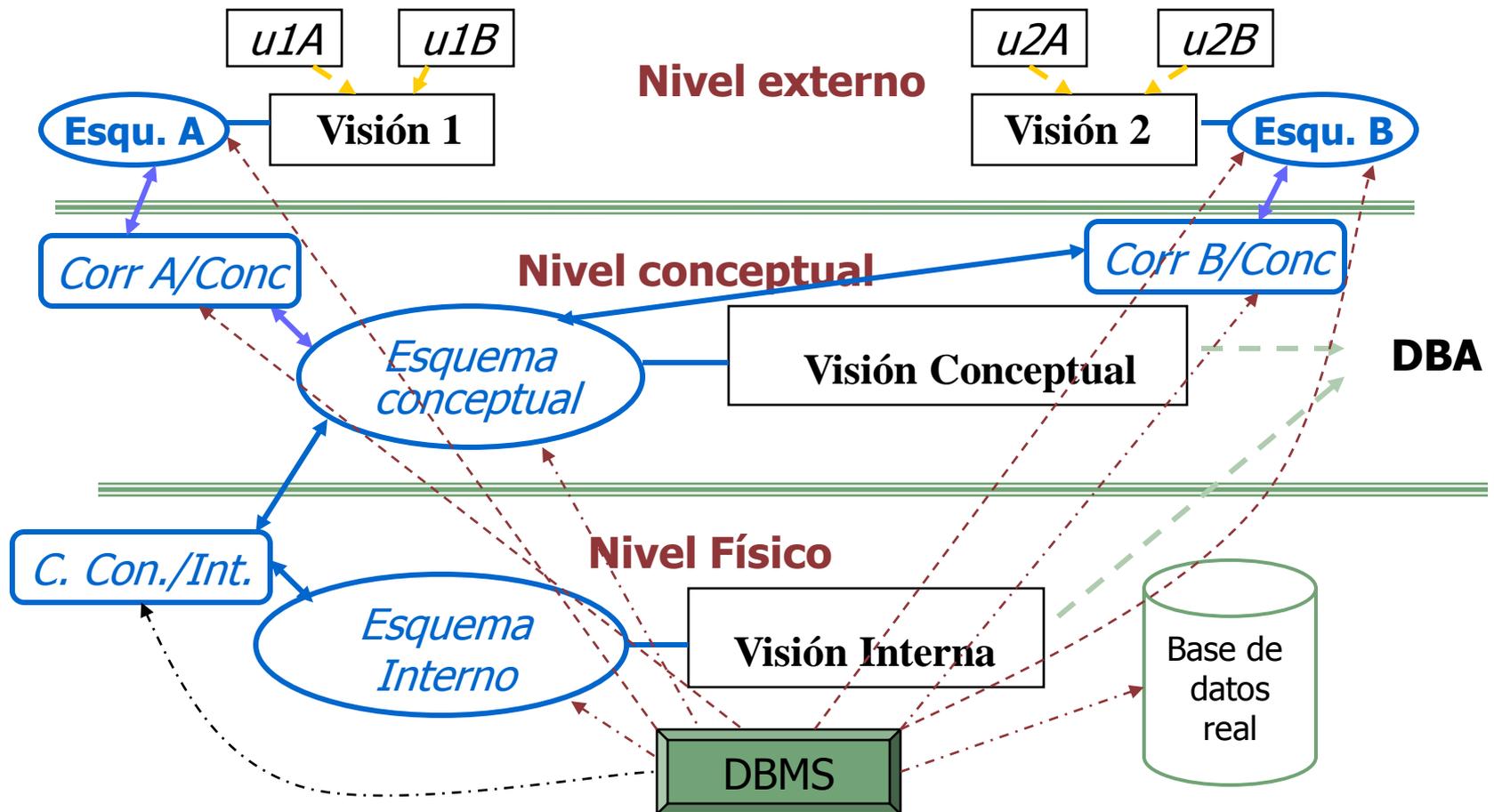
**Externa/conceptual** y **externa/externa** preservan la independencia lógica. Usuario de acuerdo con el DBA mediante el DDL.

El SGBD determina la forma en que se implementan y almacenan y el DSL permite definir las.

Algunos SGBD permiten definir el nivel interno en función del SO, usualmente de forma transparente. Permite portabilidad.

## 2.7. Transformaciones entre niveles.

### Arquitectura detallada ANSI/SPARC



## 2.7. Transformaciones entre niveles

### Situación actual

- ❑ Se puede incidir en el nivel físico; pero a través del DDL conceptual (Creando índices y clusters).
- ❑ El DSL permite definir el esquema conceptual, y los esquemas externos y las correspondencias externa/conceptual y externa/externa.
- ❑ El sistema mantiene la correspondencia conceptual/interna.
- ❑ En algunos casos no hay visión externa sino aplicaciones orientadas al usuario.

## 2.8. Misiones del administrador.

### 1.- Definir el Esquema Conceptual

- Análisis semántico del problema, identificando datos operativos.
- Diseño del esquema lógico.
- Implantación del esquema conceptual.

### 2.- Decidir la estructura de almacenamiento

- Diseño físico (Índices y agrupamientos).

### 3.- Conexión con usuarios

- Análisis de requerimientos.
- Diseño del esquema lógico particular de cada rol de usuario.
- Codificación del esquema externo, correspondencias ext/concept.

### 4.- Definir requerimientos de integridad

- Establecer reglas: genéricas (independientes de la semántica del problema) y específicas de la semántica (reglas de negocio).
- Incluir, si es posible, la integridad en el esquema conceptual.

# 2.8. Misiones del administrador:

## 5.- Definir requerimientos de seguridad

- Gestión de usuarios. Determina quienes acceden.
- Gestión de privilegios. Con que nivel de acceso se accede (lectura, modificación, eliminación, inserción).

## 6.- Definir la estrategia de recuperación de fallos

- Servidores en clusters y SGBD redundantes.
- Política de copias de seguridad y discos redundantes (RAID).
- Estrategias de relanzamiento para evitar estados inconsistentes.

## 7.- Optimización de rendimientos

- Gestión del nivel interno. Liberar espacio no utilizado e incrementar los recursos cuando sea necesario.
- Gestión de prioridades de consultas. Optimizar la ejecución de consultas y priorizar la ejecución y uso de los recursos.

## 8.- Monitorización del SGBD

Para cumplir su cometido el DBA necesita herramientas para realizar un seguimiento continuo de la actividad del sistema:

- Accesos de usuarios.
- Niveles de uso
- Eficiencia

## 2.8. Utilidades asociadas al administrador:

- ❑ Herramientas de localización
- ❑ Herramientas de copia y relanzamiento
- ❑ Herramientas de gestión de usuarios
- ❑ Herramientas de análisis:
  - Cálculo estadístico de accesos
  - Resúmenes etc..
- ❑ Herramientas de creación y mantenimiento del **catálogo**

Estructura de información que almacena:

Usuarios

Esquemas

Correspondencias

Integridad

- ❑ El DBA debe ser un profesional altamente cualificado tanto a nivel técnico como en el conocimiento funcional de la organización que gestiona.

## 2.9. Componentes de un SGBD

### Atención de peticiones:

- Usuario (terminal o aplicación) solicita operación sobre la definición o sobre los datos mediante DSL.
- El SGBD analiza la corrección sintáctica y semántica en función de los esquemas y transformaciones entre niveles.
- Crea el plan de ejecución: La forma óptima de acometer la operación desde el nivel más externo al más interno.
- Ejecuta el plan y devuelve un resultado.

Hay que tener en cuenta aspectos como el acceso concurrente, la independencia, el rendimiento y la disponibilidad para el procesamiento definido.

## 2.9. Componentes de un SGBD

### Componentes del SGBD:

- **Definición de Datos.** Interprete y/o compilador para los distintos DDL.
- **Manipulación de Datos.** Ejecución de los distintos DML.
- **Versiones interpretadas y compiladas de los procesadores de DDL y DML.** Dependiendo de la naturaleza de la operación será mejor un sistema.
- **Optimización de la ejecución de sentencias DML.** Elegir la más eficiente de las distintas alternativas de planes de ejecución.
- **Seguridad.** Mantener el esquema de seguridad definido por permisos de usuario y privilegios de acceso.
- **Integridad.** Permitir definir y mantener las reglas de integridad. Disparadores.
- **Gestión de concurrencia y consistencia.** Estrategias de bloqueo. Transacciones.
- **Diccionario de datos.** Catálogo del SGBD.
- **Herramientas de apoyo a la administración.** Desde consola o con interfaces gráficas de usuario.

## 2.10. Enfoques para la arquitectura de un SGBD

### Niveles de procesamiento:

- ▣ Nivel de Servidor de Datos.
- ▣ Nivel de Servidor de Aplicaciones.
- ▣ Nivel de Cliente.

### Arquitecturas:

- ▣ Arquitectura Centralizada.
- ▣ BD Distribuidad y programas de aplicación en aruitectura Cliente/Servidor.
- ▣ BD Distribuida y programas de aplicación en arquitectura de 3 capas.

# Definición de base de datos distribuida

## Idea inicial

Base de datos donde la información está físicamente almacenada en diferentes puntos y en sistemas diferentes.

## Hipótesis de partida

- ▣ Homogeneidad.
- ▣ Autonomía:
  - Sistemas independientes.
  - Distintos tipos de aplicaciones.
- ▣ Red de comunicaciones “lenta”.

## Base de datos distribuida:

*Conjunto de base de datos gestionadas conjuntamente y accesible a usuarios comunes.*

# Objetivos de un sistema distribuido

## Idea inicial

Un sistema distribuido debe aparecer al usuario como un sistema centralizado

- 1.- Transparencia a la localización.
- 2.- Transparencia a la fragmentación.
- 3.- Transparencia a la replicación.

## Ventajas de un sistema distribuido

- ❑ Autonomía local
  - Cada grupo es responsable de sus datos
  - El tratamiento local es más rápido
- ❑ Capacidad y crecimiento incremental
  - Se pueden abordar proyectos de más envergadura
  - Se puede ampliar la información sin cambiar todo el sistema
- ❑ Fiabilidad y disponibilidad
  - Los fallos suelen ser locales
  - Se admite un cierto nivel de redundancia
- ❑ Eficiencia y flexibilidad

# Problemas en un sistema distribuido

---

- ❑ Optimización en el procesamiento de preguntas.
- ❑ Propagación de actualizaciones.
- ❑ Mayor problema con los accesos concurrentes.
- ❑ Problemas con el manejo del catálogo.