



Universidad de Granada

decsai.ugr.es

Fundamentos de Bases de Datos

Grado en Ingeniería Informática

Tema 2: Arquitectura de un SGBD



DECSAI

**Departamento de Ciencias de la
Computación e Inteligencia Artificial**

- 1. Una arquitectura con tres niveles**
- 2. Correspondencias entre niveles**
- 3. Lenguajes de una BD**
- 4. Enfoques para la arquitectura de un SGBD**
- 5. El administrador de la BD**



- 1. Una arquitectura con tres niveles**
- 2. Correspondencias entre niveles**
- 3. Lenguajes de una BD**
- 4. Enfoques para la arquitectura de un SGBD**
- 5. El administrador de la BD**



¿Por qué organizar en niveles?

- Los **usuarios** pueden acceder a los **mismos datos**, pero desde **distintas perspectivas**.
 - Si un usuario cambia la forma de ver los datos no influye en el resto.
- La **organización global** de los datos puede **cambiarse sin afectar a los usuarios**.
- Los **usuarios no** tienen por qué **gestionar** aspectos relativos a la **representación física de los datos**.
 - El administrador de la BD puede cambiar la forma de representar los datos sin influir en los usuarios.

La percepción de los datos en un SGBD puede hacerse siguiendo tres niveles de abstracción:

- Nivel Interno
- Nivel Conceptual
- Nivel Externo
 - ANSI/SPARC
 - Precedente de dos niveles: DBTG - CODASYL

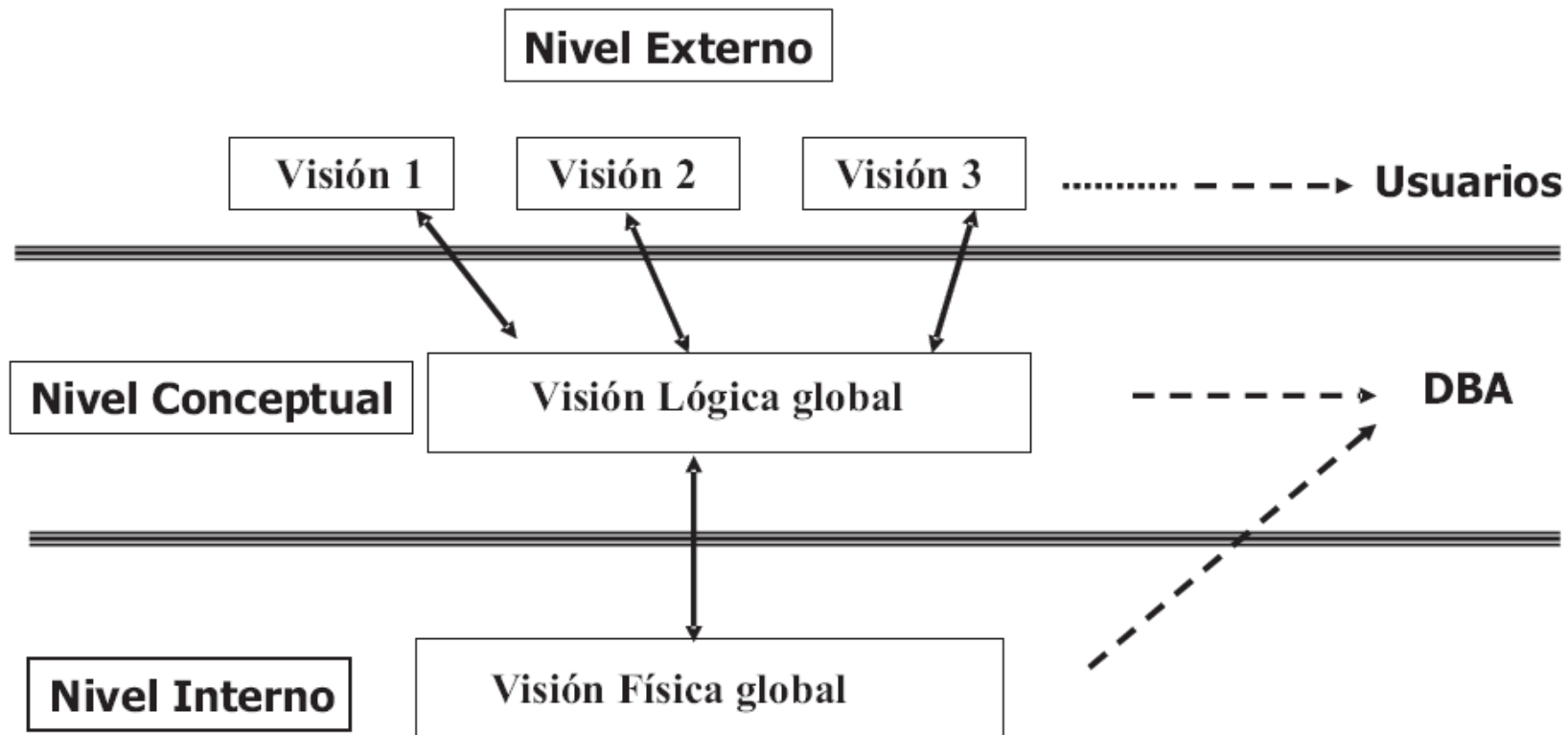
La percepción de los datos en un SGBD puede hacerse siguiendo tres niveles de abstracción:

Definición 2.1 (Nivel Interno). *Constituye la representación de la BD más cercana a la estructura de almacenamiento físico. Por tanto, es la capa donde se establece la forma en que se implantan las estructuras de datos que organizan los niveles superiores.*

Definición 2.2 (Nivel Conceptual). *Supone una abstracción global de la BD que integra y aglutina todas las percepciones que los usuarios tienen de ella.*

Definición 2.3 (Nivel Externo). *A este nivel se definen todas las percepciones particulares de la BD por parte de los usuarios. Cada usuario puede tener su propia visión de la BD.*

La percepción de los datos en un SGBD puede hacerse siguiendo tres niveles de abstracción:



La percepción de los datos en un SGBD puede hacerse siguiendo tres niveles de abstracción:

Nivel Externo

- Parte de la BD que es relevante para cada usuario.
 - Sólo aquellas entidades, relaciones y atributos que le son de interés.
 - Representadas de la forma que le interesa al usuario:
 - Ejemplos:
 - » Nombre completo o nombre y apellidos
 - » Fecha o día, mes y año
 - » ...
 - Datos calculados a partir de los que hay:
 - Edad
 - Ventas totales
 - ...

La percepción de los datos en un SGBD puede hacerse siguiendo tres niveles de abstracción:

Nivel Conceptual

- Visión global de los datos.
- Estructura lógica de los datos:
Qué datos están almacenados y qué relaciones hay entre ellos.
- Este nivel **representa**:
 - Todas las **entidades, atributos y relaciones**.
 - Las **restricciones** que afectan a los datos.
 - Información **semántica sobre los datos**.
 - Información de **seguridad** y de **integridad**.
- Da **soporte** a cada vista externa.
- No debe contener **ningún detalle de almacenamiento**.

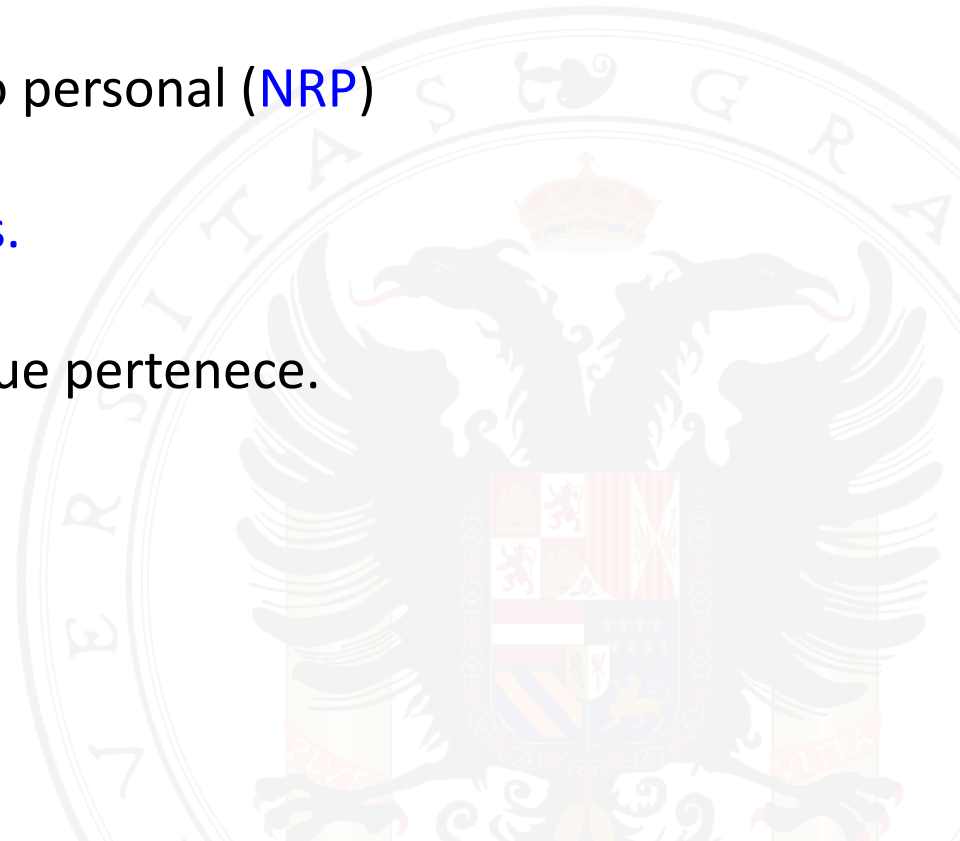
La percepción de los datos en un SGBD puede hacerse siguiendo tres niveles de abstracción:

Nivel Interno

- Representación física de la BD en el ordenador.
- Cómo están almacenados los datos.
- Busca el rendimiento óptimo del sistema.
- Representa:
 - Estructuras de datos.
 - Organizaciones en ficheros.
 - Comunicación con el SO para gestionar el uso de unidades de almacenamiento.
 - Compresión de datos, cifrado ...
- Parte de las responsabilidades de este nivel las realiza el SO:
 - Nivel físico.
 - No existe una división clara, depende de cada SGBD y de cada SO.

Ejemplo: Gestión docente universitaria

- Item básico **PROFESOR**
 - **Identificado** por:
 - Número de registro personal (**NRP**)
 - **Caracterizado** por:
 - **Nombre y apellidos.**
 - **Sueldo**
 - **Departamento** al que pertenece.



Ejemplo: Gestión docente universitaria

Visión conceptual

Profesor = registro de

NRP	campo alfanumérico de 10 caracteres,
Apellidos	campo alfanumérico de 30 caracteres,
Nombre	campo alfanumérico de 20 caracteres,
Sueldo	campo decimal de 8+2 dígitos,
Departamento	campo alfanumérico de 30 caracteres

fin Profesor.

Ejemplo: Gestión docente universitaria

Visión externa 1

- Gestión de **personal**
- Lenguaje A

```
TYPE Profesor IS RECORD (
    NRP VARCHAR2(10),
    Apellidos VARCHAR2(30),
    Nombre VARCHAR2(20),
    Sueldo NUMBER(8,2)
);
```

Visión externa 2

- Ordenación **académica**
- Lenguaje B

```
TYPE Profesor = RECORD
    NRP : STRING[10];
    Apellidos : STRING[30];
    Nombre : STRING[20];
    Departamento : STRING[30];
END;
```

Ejemplo: Gestión docente universitaria

Visión interna

`Profesor_interno BYTES=74`

`NRP TYPE=BYTES(10),OFFSET=0`

`Apellidos TYPE=BYTES(30),OFFSET=10`

`Nombre TYPE=BYTES(20),OFFSET=40`

`Sueldo TYPE=WORD(2),OFFSET=60`

`Departamento TYPE=BYTES(10),OFFSET=64.`

1. Una arquitectura con tres niveles
2. **Correspondencias entre niveles**
3. Lenguajes de una BD
4. Enfoques para la arquitectura de un SGBD
5. El administrador de la BD



Transformación o correspondencia entre niveles:

- Conjunto de *normas* que establece *cómo* se *definen* los datos de un nivel en términos de otro.
- Mecanismo *fundamental* para el establecimiento de la *independencia*:
 - *Lógica*
 - *Física*

Transformación **conceptual / interna**:

- **Cómo** se organizan las **entidades lógicas** del nivel conceptual **en términos de registros y campos** almacenados en el **nivel interno**.
- **Independencia física**:
 1. **Cambio** en el nivel **interno**.
 2. Se **cambia** la **correspondencia**.
 3. **No varía** el nivel **conceptual**.

Transformación **externa / conceptual**:

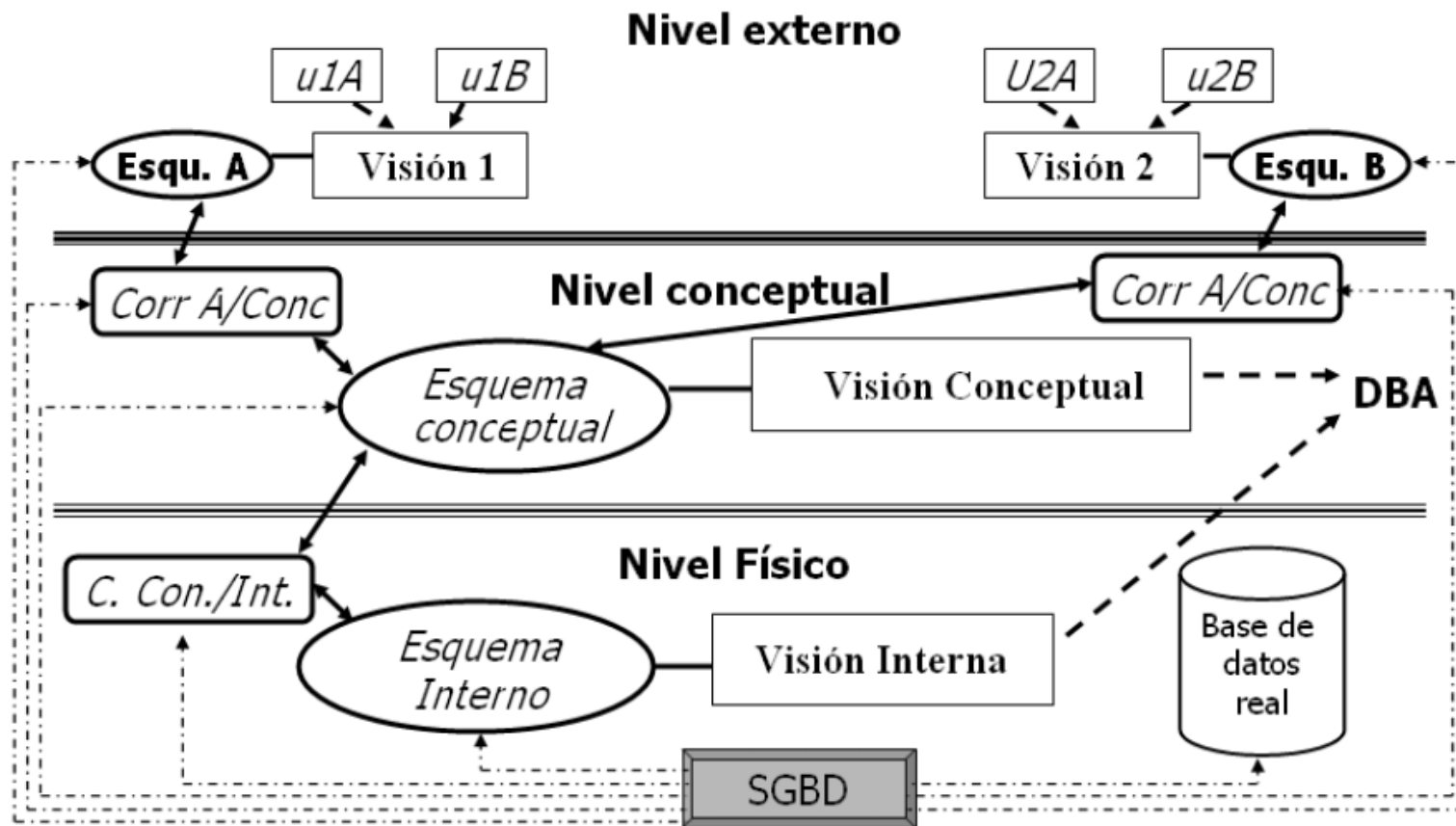
- Describe un **esquema externo** en **términos** del **esquema conceptual** subyacente.
- **Independencia lógica**:
 1. **Cambios** en el nivel **conceptual**.
 2. Se **cambia** la **correspondencia**.
 3. **No varía** el nivel **externo**.

No siempre es posible

Transformación **externa / externa**:

- Algunos SGBDs permiten describir **esquemas externos** en términos de **otros esquemas externos**.
- Independencia lógica:
 1. Cambios en el **esquema** externo **subyacente**.
 2. Se **cambia** la **correspondencia**.
 3. **No varía** el **esquema** externo **dependiente**.

Transformaciones y correspondencias:



1. Una arquitectura con tres niveles
2. Correspondencias entre niveles
3. **Lenguajes de una BD**
4. Enfoques para la arquitectura de un SGBD
5. El administrador de la BD



Recomendación **ANSI/SPARC**:

- Disponer de un **lenguaje** específico **orientado a los datos**:
 - **Definición** de datos.
 - **Control** de datos.
 - **Manipulación** de datos.
- **Sublenguaje** de **datos**: **DSL**
 - **Implementado** en el propio **SGBD**.
 - Tiene distintas partes:
 - **DDL** (*Data Definition Language*)
 - **DCL** (*Data Control Language*)
 - **DML** (*Data Manipulation Language*)

Recomendación **ANSI/SPARC**:

Definición 2.4 (DDL). *(Del inglés, Data Definition Language) O sublenguaje de definición de datos. Subconjunto del DSL destinado a la definición de estructuras de datos y esquemas en la BD.*

Definición 2.5 (DML). *(Del inglés, Data Manipulation Language) O sublenguaje de manipulación de datos. Subconjunto del DSL mediante el que podemos introducir datos en los esquemas, modificarlos, eliminarlos y consultarlos. También debe permitir consultar la estructura de los esquemas definidos en la BD.*

Definición 2.6 (DCL). *(Del inglés, Data Control Language) O sublenguaje de control de datos, que permite gestionar los requisitos de acceso a los datos y otras tareas administrativas sobre la BD.*

- ANSI/SPARC recomienda disponer de un DDL, un DML y un DCL para cada nivel de la arquitectura.
- En la práctica todos estos sublenguajes se presentan bajo una implementación única.
 - Cada sentencia trabaja sobre uno o varios niveles.
 - Un sistema de privilegios discrimina quién puede ejecutar qué y en qué nivel.
- La industria ha seguido caminos diferentes → lenguajes de datos diferentes.
 - Aparecen intentos de estandarizar los lenguajes de datos.
- Ejemplo destacado:
 - SQL y sus estandarizaciones:
 - SQL89
 - SQL92
 - SQL3

Desarrollo de aplicaciones: Lenguaje anfitrión o de aplicación.

- Desarrollo de aplicaciones en el SO que trabajen sobre la BD.
 - Propósito general:
 - C/C++
 - Java
 - C#
 - Herramientas de desarrollo específicas:
 - Developer de Oracle.
 - Oracle Application Express (Oracle APEX) .
 - Sysbase PowerBuilder.
 - IBM Rational Application Developer.
 - ...
- Proporciona:
 - Procesamiento avanzado de datos.
 - Gestión de la interfaz de usuario.

Hay que establecer un mecanismo para trasladar de la BD al entorno de procesamiento de la aplicación:

- Estructuras de datos.
- Operaciones.

Acoplamiento:

- Débilmente acoplados:
 - Lenguajes de propósito general.
 - El programador puede distinguir:
 - Sentencias propias del lenguaje anfitrión.
 - Sentencias dispuestas para acceder a la BD a través del DSL.
- Fuertemente acoplados:
 - Lenguajes y herramientas de propósito específico. Mencionadas en la transparencia anterior.
 - Se parte del DSL como elemento central y se le incorporan características procedimentales para facilitar el desarrollo de aplicaciones. P.e. Oracle PL/SQL.

Alternativas para implementar el acoplamiento débil:

- APIs de acceso a BD:
 - ODBC – Open Database Connectivity
 - JDBC – Java Database Connectivity
- DSL inmerso en el código fuente del lenguaje anfitrión:
 1. El programador escribe un código híbrido. Mezcla sentencias del lenguaje anfitrión con sentencias DSL.
 2. Hay un preprocesador que lo transforma en código fuente lenguaje anfitrión con llamadas a API acceso a BD.
 3. Se compila y se enlaza con la biblioteca de acceso a la BD.
 - Ejemplos: SQL inmerso en C, SQLJ, etc.

Alternativas para implementar el acoplamiento fuerte:

- Diversas propuestas (la mayoría **propietarias**)
 - **PL/SQL de Oracle**
 - Extensión Procedural para SQL
- Ejecución de **Java** sobre una **máquina virtual implantada** en el propio **SGBD**.

- También han aparecido numerosos entornos de desarrollo específicos para el desarrollo de aplicaciones de gestión:
 - Diseñadores de **informes**
 - Diseñadores de **formularios**
 - ...



1. Una arquitectura con tres niveles
2. Correspondencias entre niveles
3. Lenguajes de una BD
4. **Enfoques para la arquitectura de un SGBD**
5. El administrador de la BD



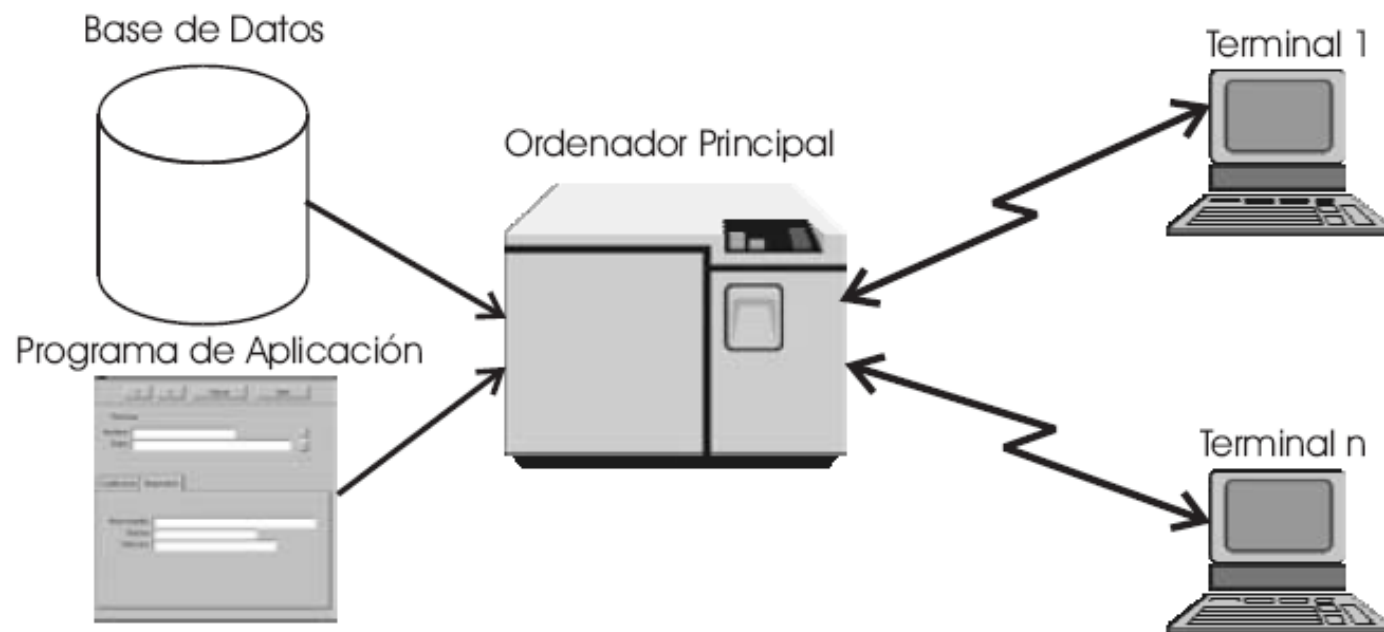
El concepto de SGBDs ha evolucionado bastante.

- Paralelamente al desarrollo de la Informática:
 - Forma de gestionar la información.
 - Forma de ejecutar los programas.
 - Forma de interactuar con el usuario.

Inicialmente:

- Esquema centralizado:
 - Toda la carga de gestión y procesamiento de información recaía en servidores centrales.
 - El usuario accedía mediante terminales.
 - En el ordenador principal:
 - SGBD.
 - Programas de aplicación.

Arquitectura centralizada:



a) Arquitectura Centralizada

Problema:

- Elevado coste de los ordenadores principales.
 - Aparece el PC.

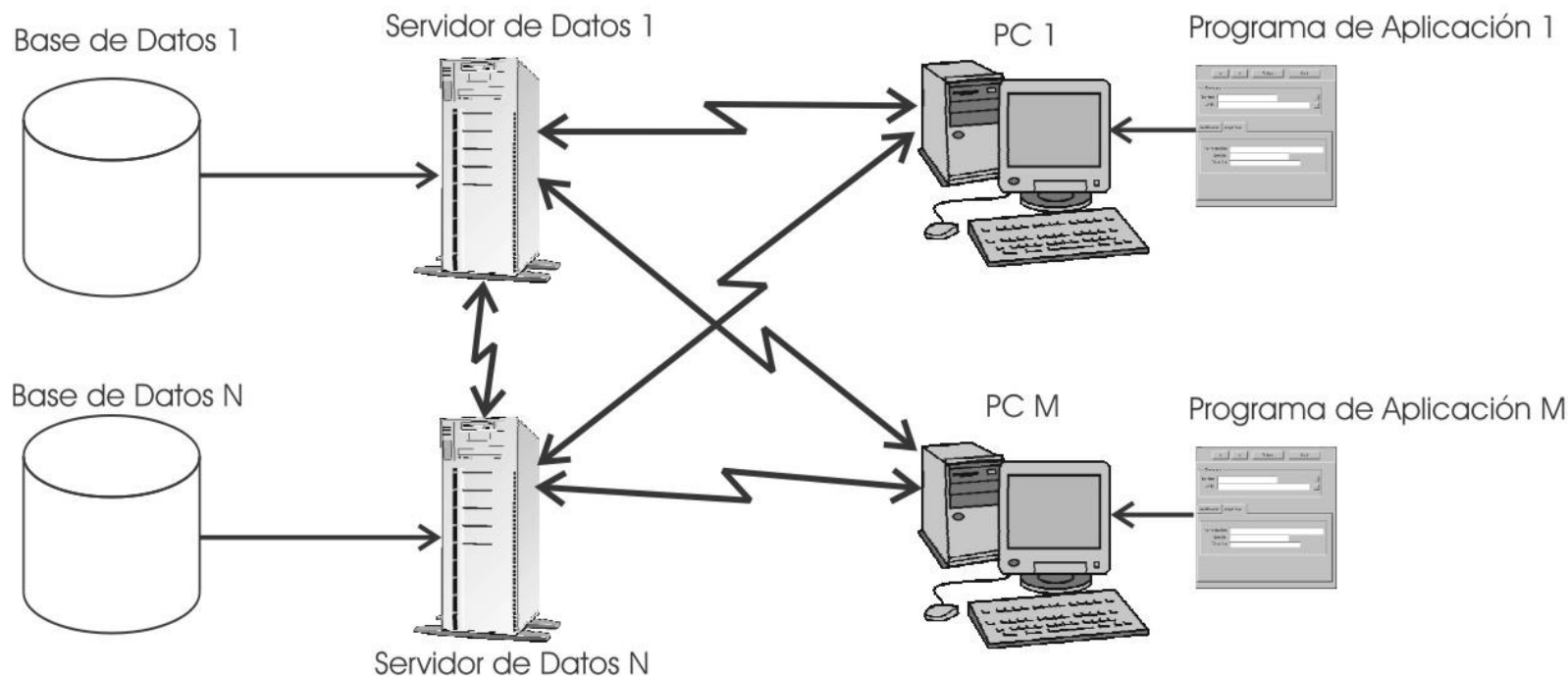
Solución:

- Desplazar la ejecución de los programas de usuario y la interacción hasta los PCs.
 - Reducción de costes en hardware.
- Aproximación Cliente/Servidor:
 - Servidor:
 - Servidor de BD.
 - Servicio de escucha de peticiones.
 - PCs conectados en red con el servidor:
 - Programas de aplicación.
 - Servicio de enlace cliente que interactúa con el servicio de escucha instalado en el servidor.

Desarrollo de las redes de comunicaciones:

- Enfoque distribuido para los servidores.

Arquitectura distribuida:



b) BD Distribuida y programas de aplicación en arquitectura Cliente/Servidor

Problema:

- Alto coste de mantenimiento de los PCs:
 - Instalación.
 - Configuración.
 - Actualización.

Solución:

- Separar en las aplicaciones:
 - Parte que interactúa con el usuario: interfaz de usuario.
 - Parte de ejecución lógica del programa.

Actualmente:

Arquitectura articulada en tres niveles de procesamiento (I):

– Nivel de Servidor de Datos:

- Posiblemente distribuido.
- El SGBD permite organizar la información de la empresa como una BD global.
- Las peticiones de datos formuladas desde una sede se traducen de forma transparente a peticiones en las sedes donde se encuentran esos datos.

Actualmente:

Arquitectura articulada en tres niveles de procesamiento (II):

- Nivel de Servidor de Aplicaciones:

Son evoluciones de Servidores Web que proporcionan los programas de aplicación a Clientes ligeros, que disponen de entornos de ejecución de aplicaciones:

- Usando estándares.
- Protocolos de red TCP/IP.
- Protocolo HTTP.
- Despliegue de Applets Java a ejecutar en Navegadores con soporte de máquina virtual Java.
- Servlets, JSP, ASP, etc.

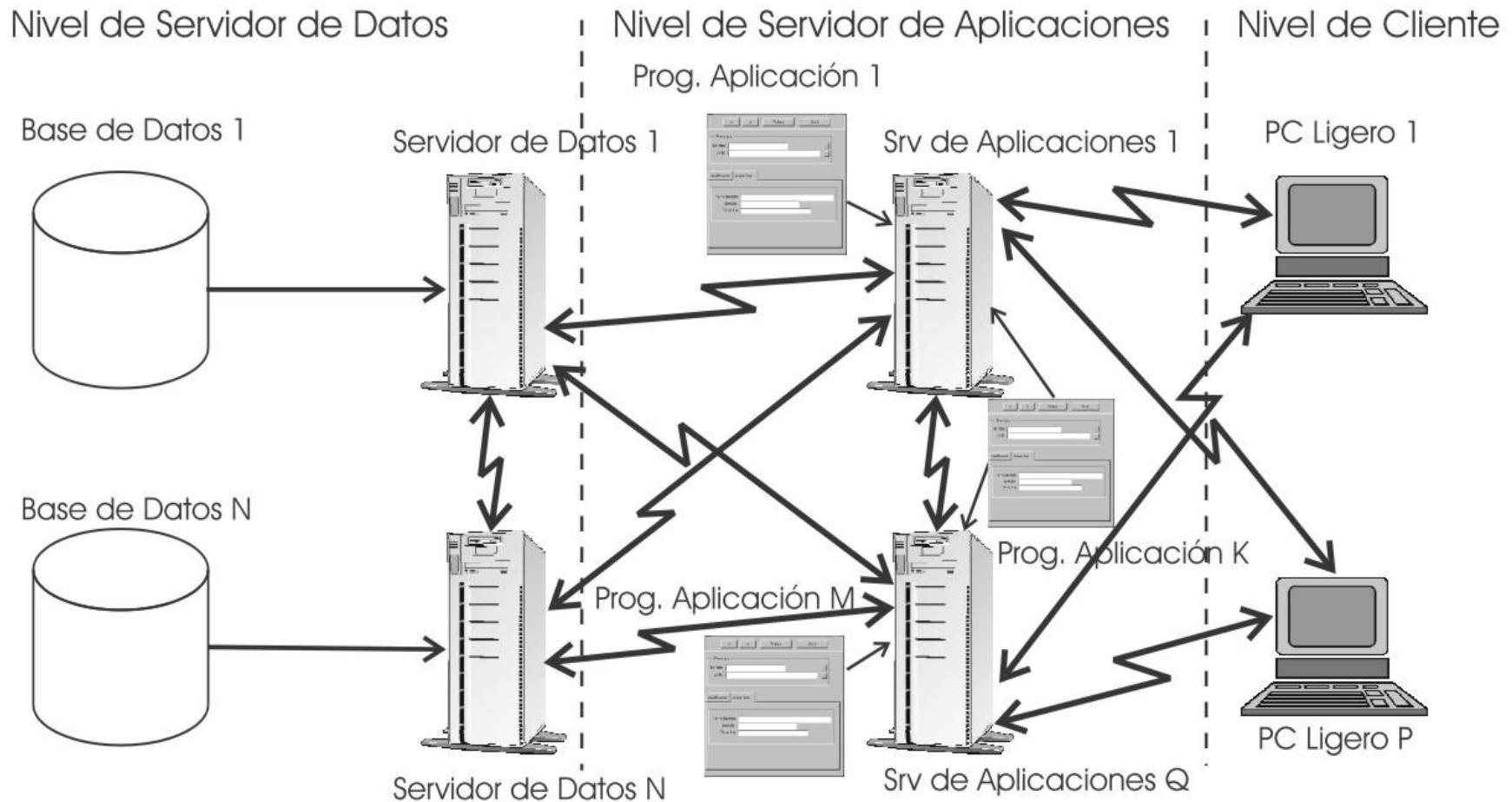
Actualmente:

Arquitectura articulada en tres niveles de procesamiento (III):

- El Nivel de Cliente:

PCs ligeros dotados de configuraciones basadas en estándares abiertos. En muchos casos se pueden ejecutar las aplicaciones desplegadas en un navegador web con soporte de ejecución de código javascript y html avanzado.

- Basados en el carácter portable con que se distribuyen las aplicaciones desde los servidores de aplicaciones.
- Menos dependencia del hardware y del SO a la hora de abordar la ejecución de las aplicaciones.



c) BD Distribuida y programas de aplicación en arquitectura de tres Capas

Ventajas:

- Reducción significativa de costes en cuanto al mantenimiento de los clientes: instalación, configuración y actualización de las aplicaciones realizada en el servidor no en cada cliente.
- Mayor facilidad y flexibilidad para el usuario. Puede acceder desde casi cualquier puesto y a veces desde distintos dispositivos: móviles, tablets, portátil, pc, etc.

Inconvenientes:

- Mayor complejidad en:
 - La configuración y administración de los servidores de aplicaciones.
 - El desarrollo de las aplicaciones conforme a este modelo distribuido.

Ejemplo:

- Usuario del PC invoca desde el navegador la ejecución de una aplicación a través de una URL.
- La parte de interfaz de usuario de la aplicación se puede distribuir como:
 - Un applet Java que se ejecuta en la máquina virtual del navegador.
 - Una serie de paginas HTML generadas desde el servidor de aplicaciones:
 - Servlets.
 - JSP.
 - ASP.
- La interacción del usuario a través de esta interfaz determina la interacción con la parte lógica de la aplicación que se ejecuta en el servidor de aplicaciones:
 - Peticiones de procesamiento.
 - Acceso a datos de la BD.
 - Generación de nuevas páginas o evolución del applet que ofrecen la respuesta al usuario a través de la interfaz de usuario.

1. Una arquitectura con tres niveles
2. Correspondencias entre niveles
3. Lenguajes de una BD
4. Enfoques para la arquitectura de un SGBD
5. **El administrador de la BD**



El DBA es una figura de primordial relevancia en el contexto de los SGBDs:

- **Elaboración del esquema conceptual:**
 - **Análisis** de las necesidades de **información** de la empresa.
 - **Identificación** de los datos operativos.
 - **Elaboración** del **esquema lógico**.
 - **Implantación** del **esquema conceptual**.
- **Decidir la estructura de almacenamiento en el nivel interno**
 - **Esquema interno**.
 - **Correspondencia conceptual/interna** asociada.

El DBA es una figura de primordial relevancia en el contexto de los SGBDs:

- **Conexión con usuarios:**
 - **Análisis** de requerimientos.
 - **Diseño lógico.**
 - **Codificación** del **esquema externo**, **correspondencias ext/concept.**
- **Definir las restricciones de integridad:**
 - Establecer **reglas: genéricas y específicas.**
 - **Incluir**, si es posible, la **integridad en** el **esquema conceptual.**

El DBA es una figura de primordial relevancia en el contexto de los SGBDs:

- Definir e **implantar la política de seguridad**:
 - Gestión de **usuarios**.
 - Gestión de **privilegios**.
- Definir e **implantar la estrategia de recuperación frente a fallos**:
 - Los SOs y los SGBDs suelen incorporar facilidades para afrontar los fallos:
 - **SGBDs redundantes**.
 - **RAID** - Redundant Array of Inexpensive Disks
 - El DBA puede y debe **realizar copias de seguridad de la BD**.
 - Políticas de **gestión de transacciones**.

El DBA es una figura de primordial relevancia en el contexto de los SGBDs:

- Optimización del rendimiento:
 - Liberar espacio no utilizado.
 - Reorganizar las operaciones para que se ejecuten de forma más rápida.
 - Determinar la necesidad de nuevos recursos hardware.
 - Establecer prioridades en el uso de los recursos.
- Monitorizar el SGBD:
 - Seguimiento continuo de la actividad del sistema.
 - Auditar el acceso de los usuarios a los diversos recursos de la BD.
 - Comprobar los niveles de uso de los sistemas de almacenamiento.
 - Evaluar la eficiencia con que se realizan las operaciones.