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

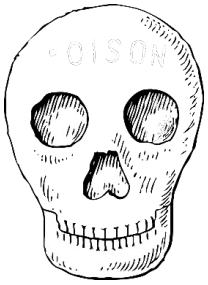
¿Por qué organizar en niveles?

- 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 → independencia lógica.
- 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.

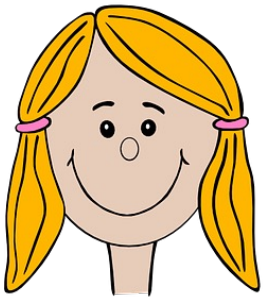
La percepción de los datos en un SGBD puede hacerse siguiendo tres niveles de abstracción

- Propuesta por el grupo de estudio en sistemas de administración de bases de datos de ANSI/SPARC.
- Precedente de dos niveles propuesto por el subgrupo de CODASYL (Conference on data system) denominado DBTG (Data base task group).

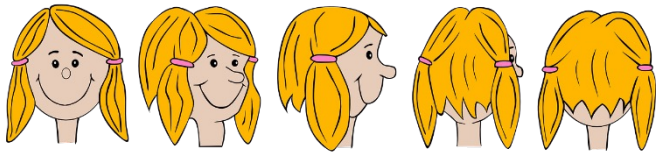
- Tres niveles:
  - Nivel interno.
  - Nivel conceptual.
  - Nivel externo.



- Nivel interno:
  - Sería como observar a una persona como un conjunto de huesos, músculos y órganos.
  - Solo personal cualificado es capaz de entender ese nivel.



- Nivel conceptual:
  - Sería como observar a una persona tal y como la percibimos habitualmente y conocer todas sus facetas.

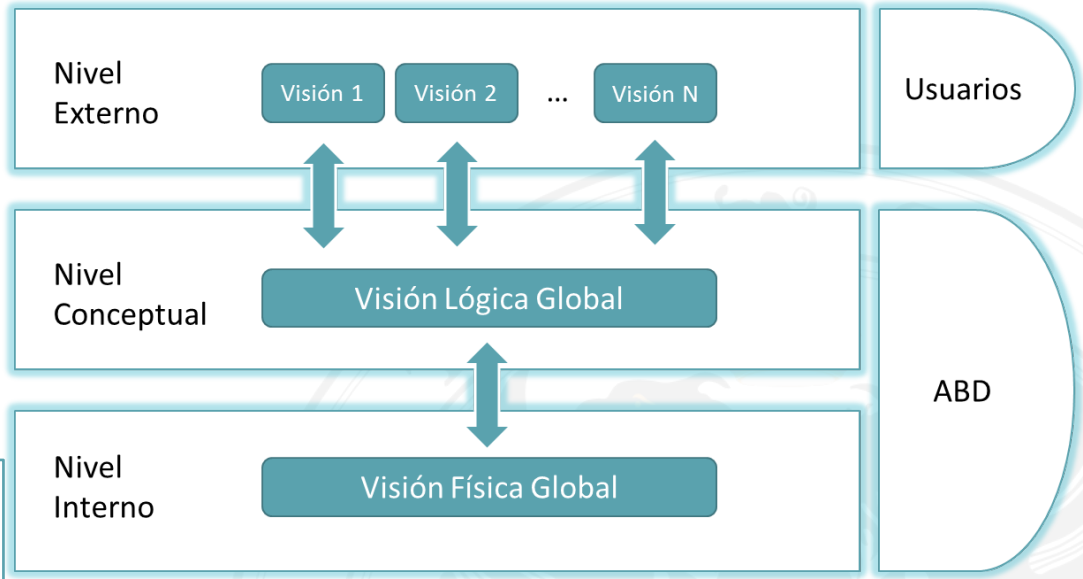


- Nivel externo:
  - La persona la podemos percibir de distintas maneras.

Aquí se definen las distintas percepciones particulares de la base de datos por parte de los usuarios.

Abstracción global de la base de datos. Integra todas las percepciones que los distintos usuarios tienen de la base de datos.

Representación de la base de datos más cercana a la estructura de almacenamiento físico. Estructuras de datos sobre las que se sustentan los niveles superiores.



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, etc.
- Datos calculados a partir de los que hay:
  - Edad.
  - Ventas totales, etc.

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.
- 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, etc.
- 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

- Se puede representar con la siguiente estructura:

```
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

- La sintaxis del lenguaje interno podría ser:

```
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.
```

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.

- Garantizar la independencia física es posible:
  1. Cambios en el nivel interno.
  2. Se cambia la correspondencia.
  3. De forma que no varía el nivel conceptual.

Transformación externa / conceptual

- Describe un esquema externo en términos del esquema conceptual subyacente.
- Definir la correspondencia entre cada esquema externo y el conceptual sobre el que se construye.

- Garantizar la independencia lógica:
  1. Cambios en el nivel conceptual.
  2. Se cambia la correspondencia.
  3. No varía el nivel externo.

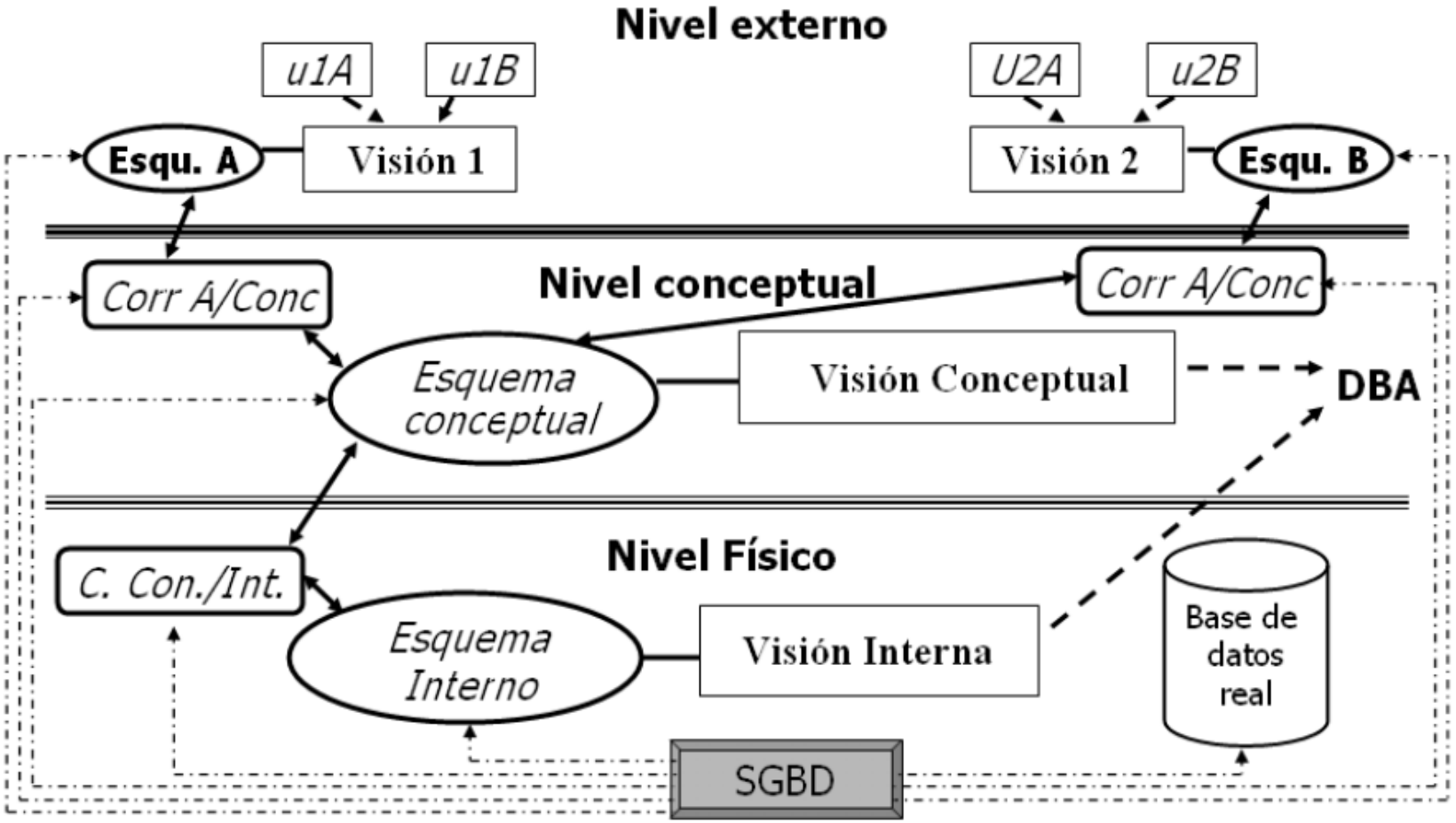
- No siempre es posible.

Transformación externa / externa

- Algunos SGBDs permiten describir esquemas externos en términos de otros esquemas externos.
- Existencia de correspondencias externa/externa.

- Garantizar la independencia lógica:
  1. Cambios en el esquema externo subyacente.
  2. Se cambia la correspondencia.
  3. No varía el esquema externo dependiente.

Transformaciones y correspondencias



Recomendación ANSI/SPARC

- Disponer de un lenguaje específico orientado a los datos:
  - Definición de datos.
  - Control de datos.
  - Manipulación de datos.

- Se denomina sub lenguaje de datos: DSL
  - Implementado en el propio SGBD.
  - Tiene distintas partes:
    - DDL (*Data Definition Language*)
    - DML (*Data Manipulation Language*)
    - DCL (*Data Control Language*)

Recomendación ANSI/SPARC

- DDL
  - Subconjunto del DSL para definición de estructuras de datos y esquemas de la BD.

- DML
  - Subconjunto del DSL para introducción, modificación, borrado y consulta de datos.
  - También debe permitir consultar los esquemas definidos en la BD.

- DCL
  - Subconjunto del DSL para gestionar los requisitos de acceso a los datos y otras tareas de administración de 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 sub lenguajes 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.
  - Sin separación estricta entre niveles.
- Ejemplo destacado:
  - SQL y sus estandarizaciones:
    - SQL89
    - SQL92
    - SQL3
    - Los SGBD han ido proporcionando soporte.

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 para BD:
    - 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.
- Se llama acoplamiento entre leguaje anfitrión y lenguaje de datos:
  - 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

- Usar APIs de acceso a BD:
  - Acceder a la BD desde el código fuente del lenguaje anfitrión.
  - 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 transforma el código fuente híbrido a un código fuente anfitrión con llamadas a la API de 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:

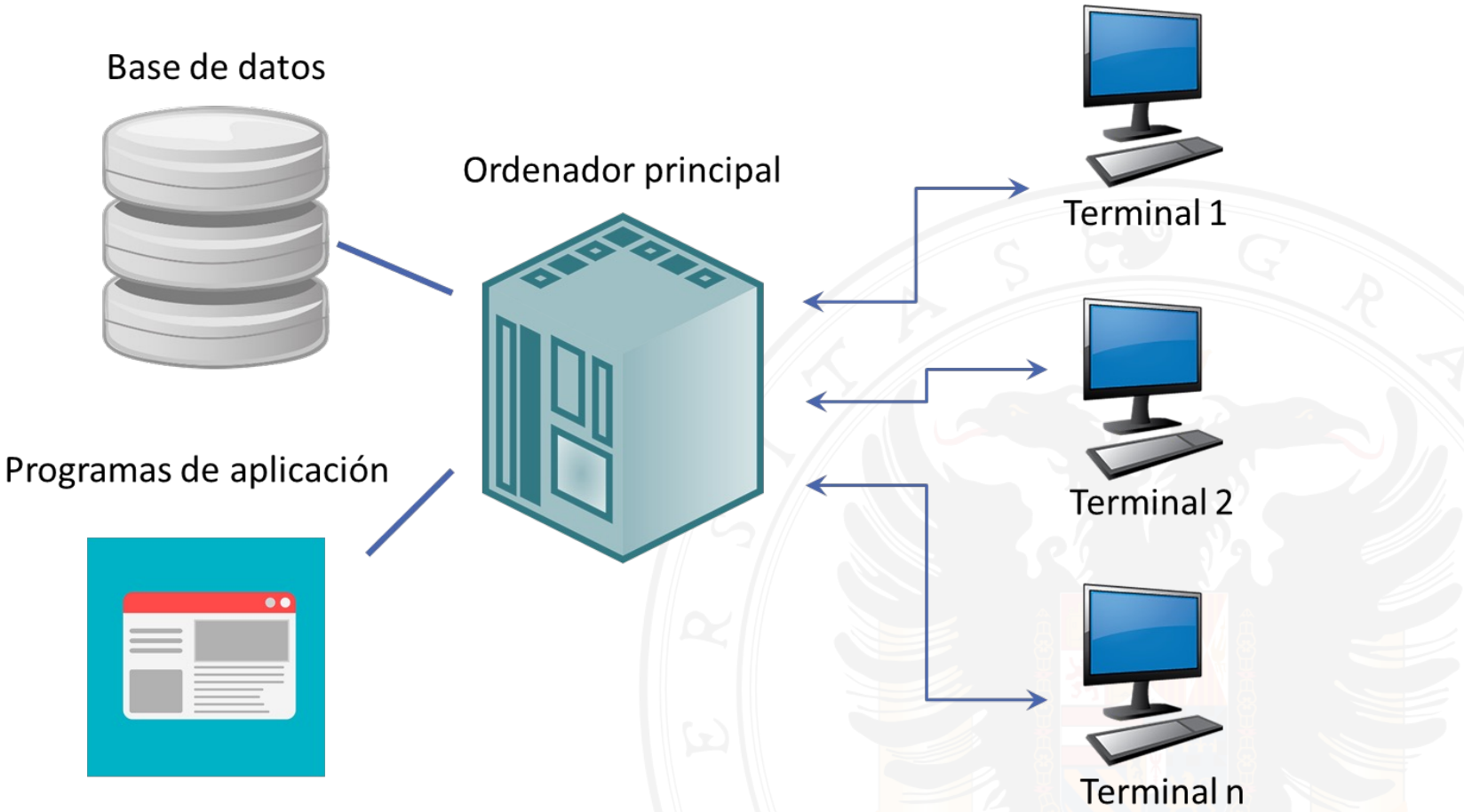
  - Diseñadores de informes.
  - Diseñadores de formularios.
  - Etc.

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

Arquitectura centralizada

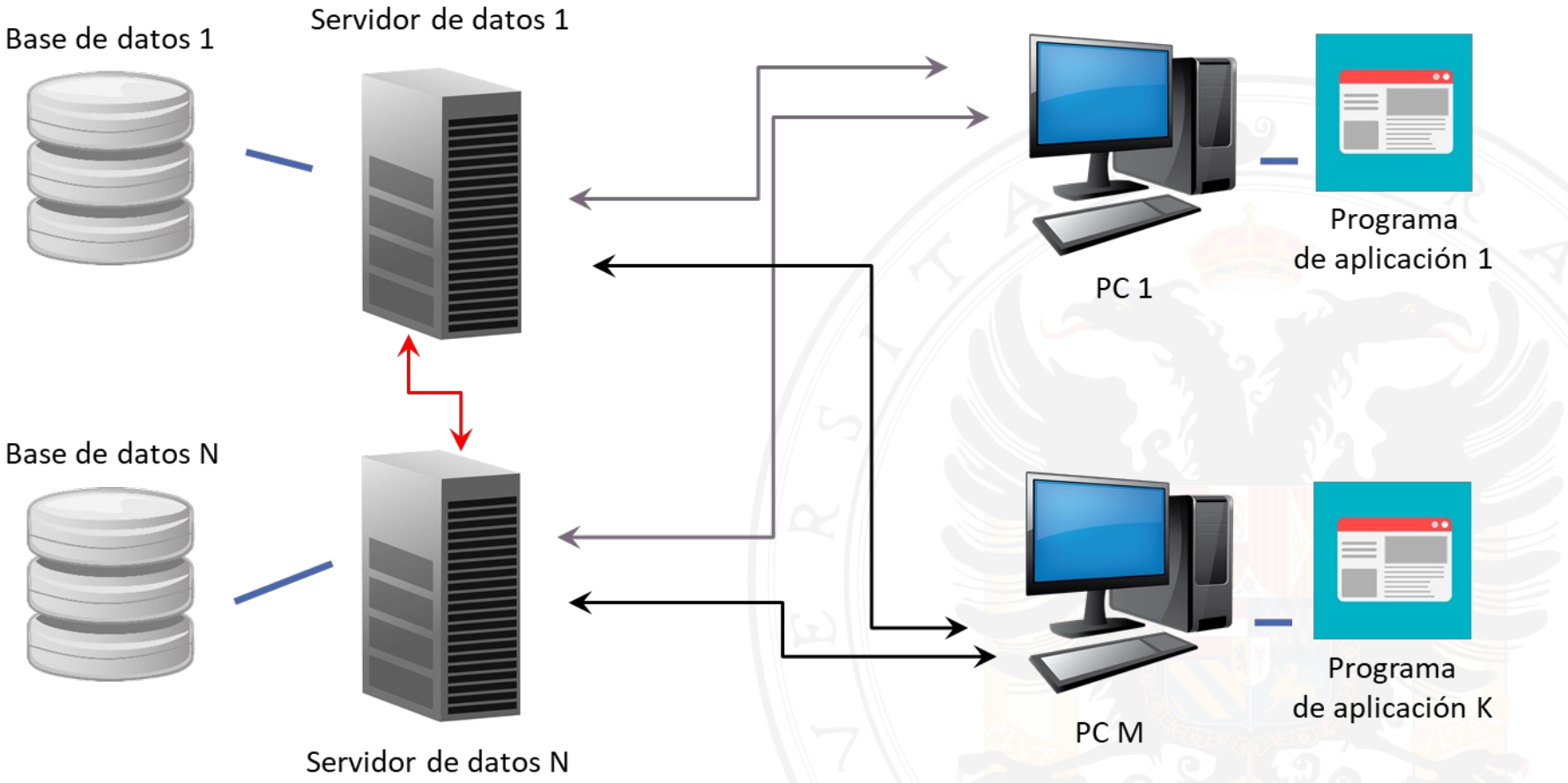


- Problema:
  - Elevado coste de los ordenadores principales.
  - Aparece el PC con precios contenidos.

- Solución:
  - Desplazar la ejecución de los programas de usuario y la interacción hasta los PCs.
    - Reducción de costes en hardware.
    - Disposición de terminales con mayor capacidad.

  - 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.
- 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):

1. Nivel de servidor de datos:
   - Posiblemente distribuido geográficamente.
   - 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):

2. 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.

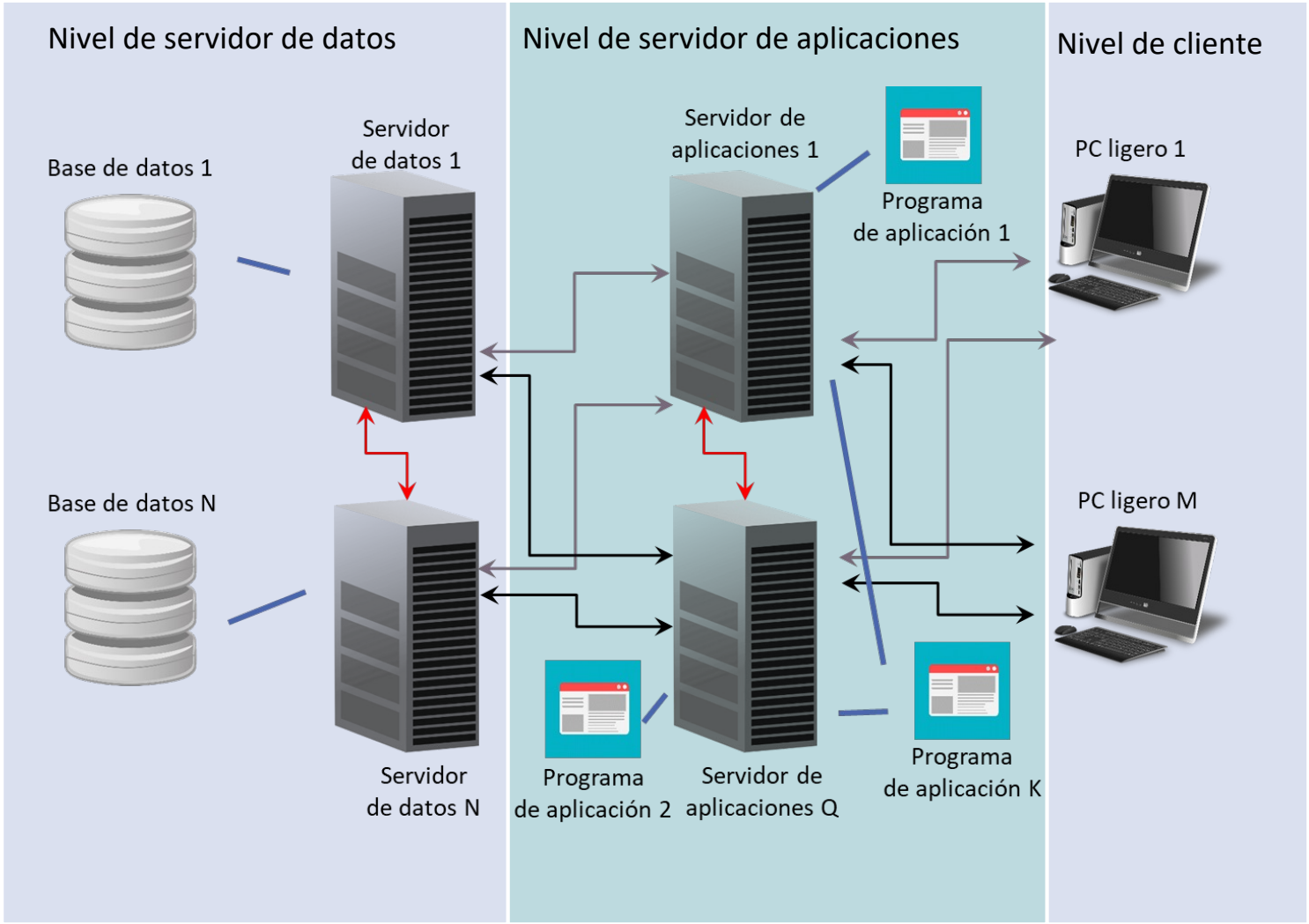
Actualmente:

Arquitectura articulada en tres niveles de procesamiento (III):

3. 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.

- 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 de 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 páginas 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.
  - 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.

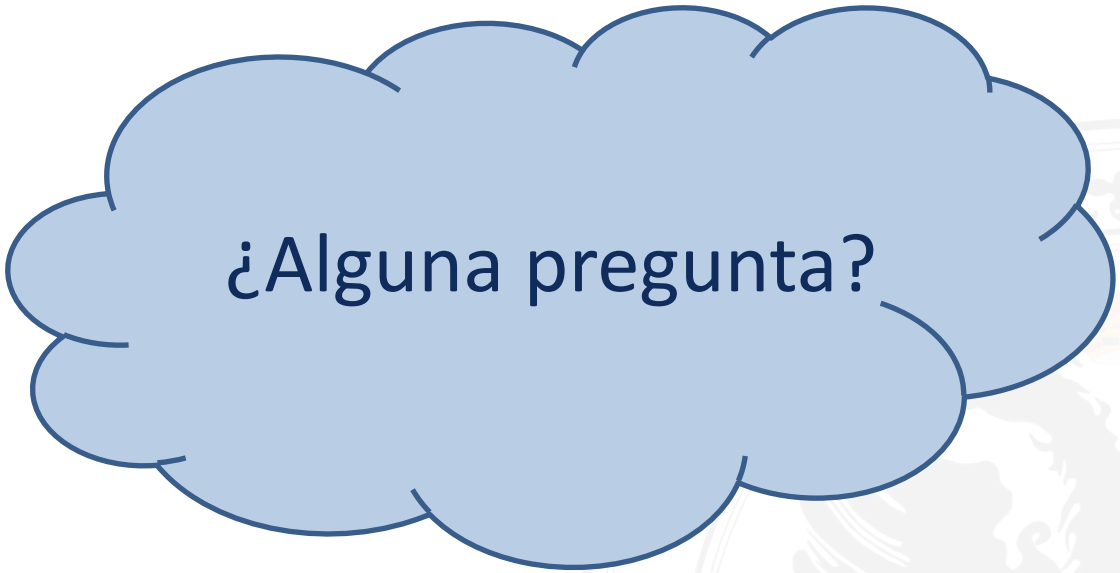
- Conexión con usuarios:
  - Análisis de requerimientos.
  - Diseño lógico.
  - Apoyo al desarrollador de aplicaciones: Codificación del esquema externo, correspondencias externo/conceptual.

- Definir las restricciones de integridad:
  - Establecer reglas: genéricas y específicas.
  - Incluir, si es posible, la integridad en el esquema conceptual.

- Definir e implantar la política de seguridad:
  - Gestión de usuarios.
  - Gestión de privilegios.
- Definir e implantar la estrategia de recuperación frente a fallos:
  - Los SSOO 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.

- 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.


¿Alguna pregunta?