



Sistemas de gestión de bases de datos

T8.2006-07



Dpto. Lenguajes y Sistemas Informáticos

Universidad de Alicante

Objetivos de este tema

- presentar la arquitectura básica ANSI/SPARC del SGBD (1972-77)

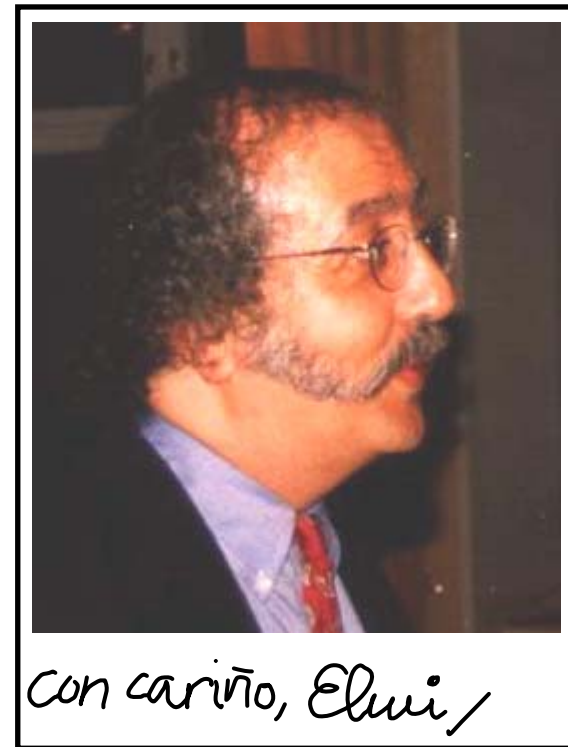


- Standards Planning and Requirements Committee



Bibliografía básica:

- [CELMA97]
- [DATE01]
- [KORT87]
- [ELMASRI02]
- ...



Bibliografía básica:

- [CELMA97]
- [DATE01]
- [KORT87]
- [ELMASRI02]
- Sistemas de bases de datos: un enfoque práctico para diseño, implementación y gestión, Thomas M. Conolly y Carolyn E. Begg, Pearson, Addison Wesley (1269 pags.)



Problemas

- detectados con los sistemas de archivo convencional
 - redundancia de datos
 - dependencia de los programas respecto de los datos
 - insuficientes medidas de seguridad



Problemas, nuevas necesidades

- **redundancia de datos**
 - “Este señor puede que se llame Jonathan del Amor o Vanessa, o vete a saber”
 - degradación de la calidad de los datos
 - disminución del rendimiento del sistema
 - aumento del esfuerzo de mantenimiento
- dependencia de los programas respecto de los datos
- insuficientes medidas de seguridad

Introducción ►

SSGBD

Cliente-
servidor

Problemas, nuevas necesidades

- redundancia de datos
- **dependencia de los programas respecto de los datos**
 - ¿otra vez a modificar y recompilar todos los programas?
 - baja eficiencia en la atención a nuevos requisitos, reorganizaciones, rediseños ...
- insuficientes medidas de seguridad

Introducción ►

SSGBD

Cliente-
servidor

Problemas, nuevas necesidades

- redundancia de datos
- dependencia de los programas respecto de los datos
- **insuficientes medidas de seguridad**
 - control concurrencia
 - evitar colisiones y pérdida de datos en los accesos a datos comunes
 - recuperación de ficheros
 - copias de seguridad, espejos, dietarios ...
 - control de autorizaciones
 - usuarios y permisos



Consecuencias

- Pobre control de los datos
- Capacidades de manipulación de los datos limitadas o inadecuadas
- Excesivo esfuerzo de programación



¿Solución? ¡SGBD!

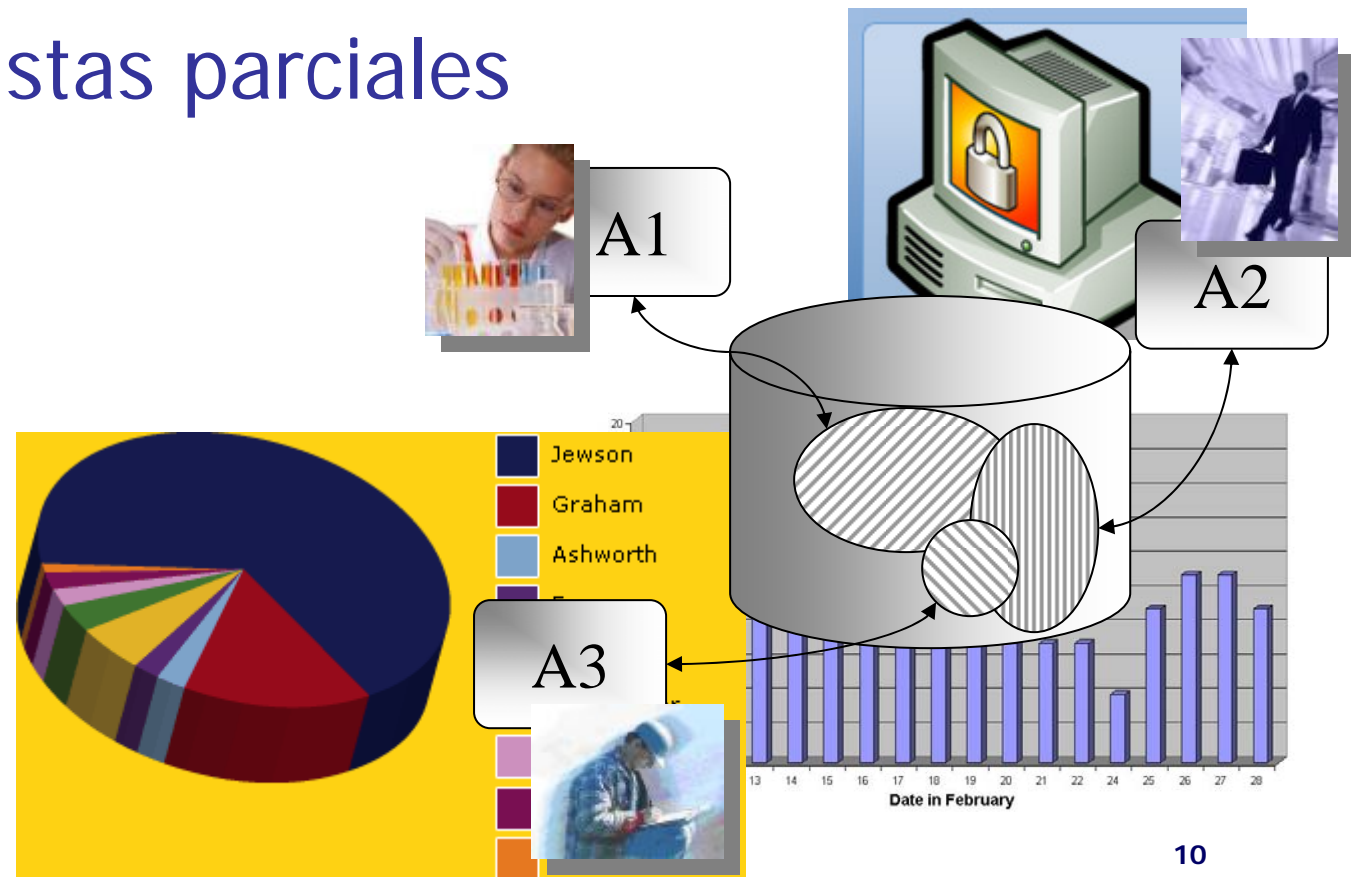
- Descripción centralizada de los datos
- Vistas parciales



Introducción

SSGBD ►

Cliente-servidor



Objetivos

- Independencia

- modificar la definición de un dato no implica reprogramar / recompilar una aplicación que **NO** lo usa
 - si aumento el código de artículo, ¿debo modificar la aplicación de nóminas?

- Integridad

- los datos deben ser correctos
 - el nombre del empleado 100 es Manolo, siempre

- Seguridad

- acceso restringido a quien y como esté autorizado
 - La tabla de vendedores sólo es accesible por alumnos de BD1 y profesores: los profesores pueden leer y escribir, y los alumnos sólo leer.

Introducción

SSGBD ►

Cliente-
servidor

¿Cómo conseguirlo?

- Arquitectura
 - distintos niveles de representación de los datos, buscando la independencia entre ellas
 - cada esquema tiene una función
 - Esquema conceptual: *qué*
 - Esquema lógico: *cómo*
 - Esquema interno: *almacenamiento, eficiencia*
 - Esquemas externos: *explotación*

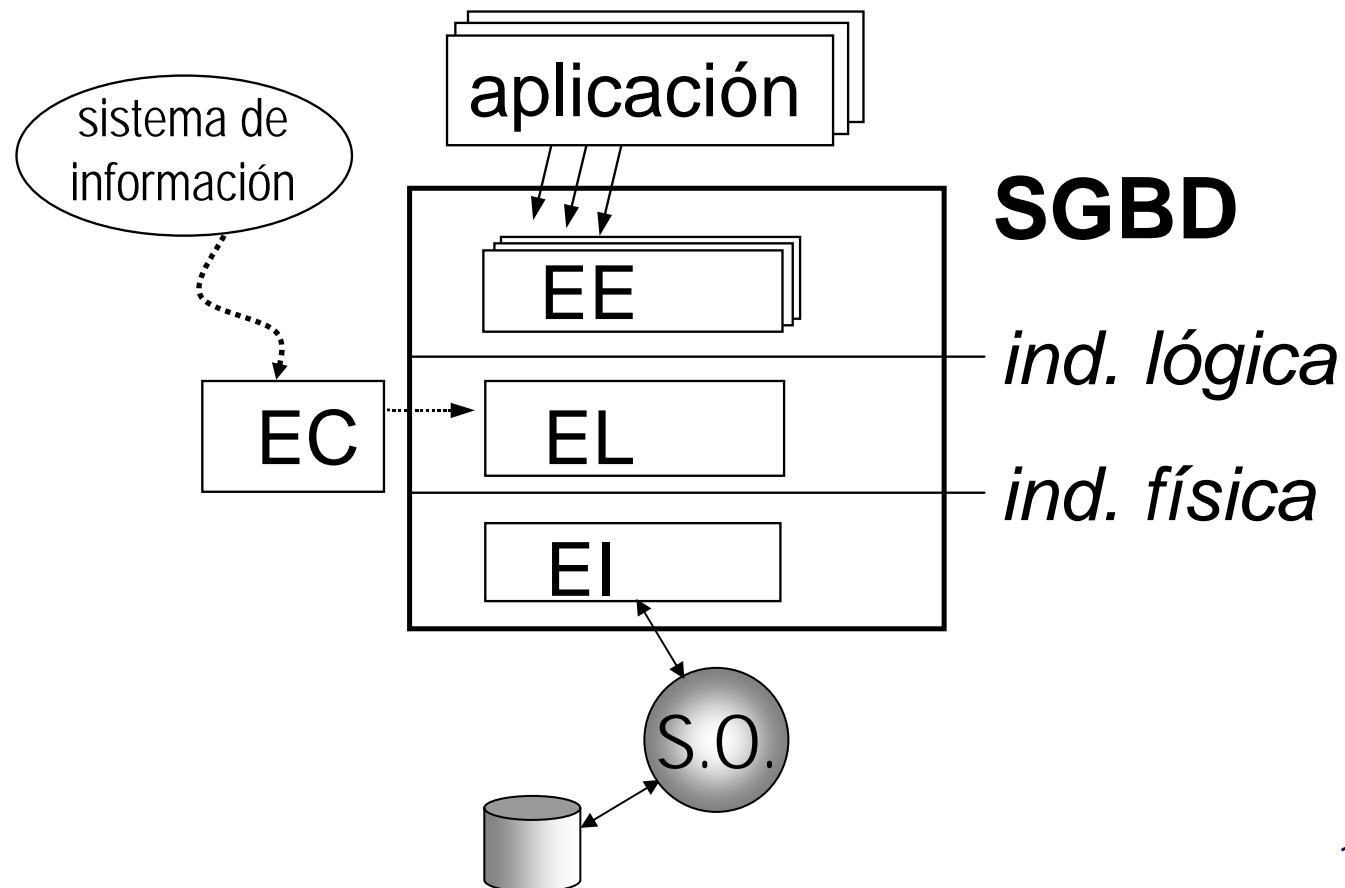
Introducción

SSGBD ►

Cliente-
servidor

¿Cómo conseguirlo?

- Arquitectura

**SGBD***ind. lógica**ind. física*

Arquitectura de esquemas

- “Manolo’s kioskos S.A. es una gran empresa con delegaciones en todo el mundo”

- ya pero, ¿qué es una delegación?
 - un kiosko
- ¿de cuántas “delegaciones” estamos hablando?
 - ... 3
- todas en la ciudad
 - sí, todas en San Vicente

y usted quiere controlar sus pedidos

- sí, tengo un local de 100m² donde voy recibiendo las chuches, el tabaco (¡uy!, no, ya no), los juguetes de a 20 duros, los duvedeses, ...)

Introducción

SSGBD ►

Cliente-servidor



LS#

Arquitectura de esquemas

- “Manolo’s kioskos S.A. es una gran empresa con delegaciones en todo el mundo”
 - El sistema de información se compone de
 - delegaciones (kioskos)
 - empleados (kioskeros)
 - artículos (chuches, colecciones, periodicos, ...)
 - proveedores
 - los artículos son servidos por los proveedores
 - $\text{card}(\text{artículo}, \text{servir}) = (1, 1)$ $\text{card}(\text{proveedor}, \text{servir}) = (0, N)$
 - los empleados se encargan de las delegaciones
 - $\text{card}(\text{empleado}, \text{encarga}) = (0, 1)$
 $\text{card}(\text{kiosko}, \text{encarga}) = (0, 1)$
 - ...

Introducción

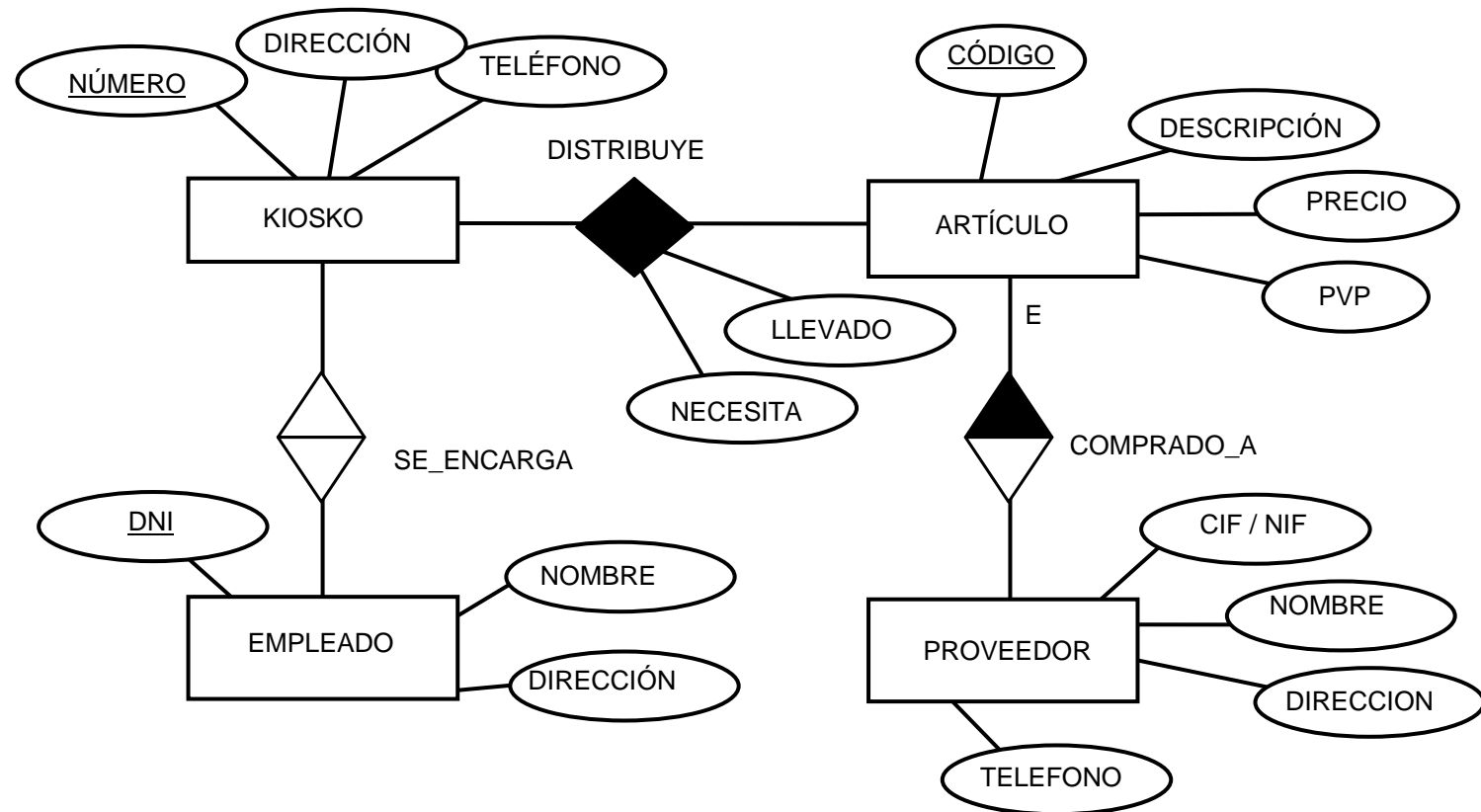
SSGBD ►

Cliente-
servidor



Arquitectura de esquemas

- Esquema **conceptual**



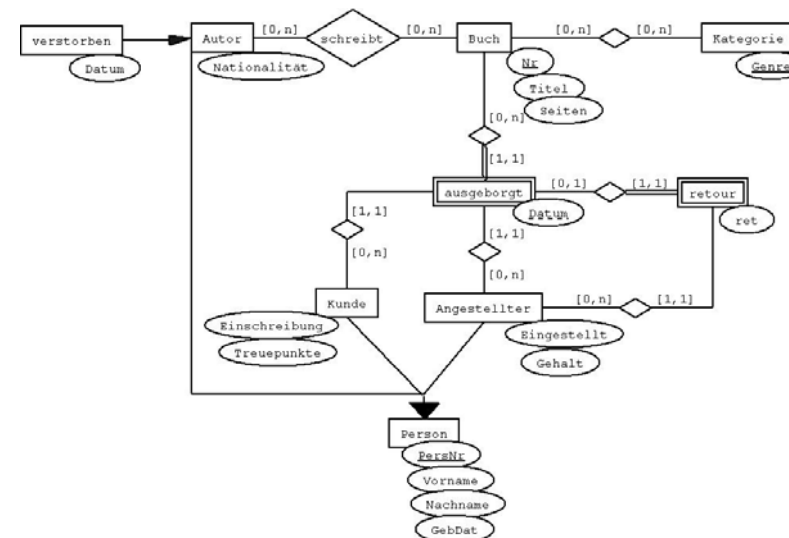
Introducción

SSGBD ►

Cliente-
servidor

Arquitectura de esquemas

- Esquema **conceptual**
 - en un modelo de datos de alto nivel
 - clases de objetos, atributos, correspondencias
 - descripción de los datos (y los procesos) independiente de cualquier software o hardware



Arquitectura de esquemas

- Voy a utilizar un SGBDR
 - “traducir” el esquema anterior a “tablas”
 - este “gran empresario” necesita
 - delegaciones: número entero(0 a 99), dirección cadena(100), teléfono cadena (12)
 - empleados: nif cadena(10), dirección cadena(120), cc cadena (24)
 - proveedores: cif cadena(12), dirección cadena(100), teléfono cadena(12)
 - artículos: código cadena(5), descripción cadena(50), preciocompra flotante(3,2), pvp flotante(3,2)

Introducción

SSGBD ►

Cliente-
servidor



Arquitectura de esquemas

- Esquema **lógico**

delegacion (numero entero, direccion varchar2(100), teléfono varchar2(12), **primary key (numero)**)

empleado (nif varchar2(10), direccion varchar2(120), cc varchar2(24), **primary key (nif)**)

encargado (nif varchar2(10), numero entero **NOT NULL**, **primary key (nif)**, **unique (numero)** foreign key (nif) references empleado, foreign key (numero) references delegacion)

proveedor (cif varchar2(12), direccion varchar2(100), teléfono varchar2(12), **primary key(cif)**)

artículo (codigo varchar2(5), descripcion varchar2(50), preciocompra float(3,2), pvp float(3,2), cif varchar2(12) **NOT NULL**, **primary key(codigo)**, foreign key (cif) references proveedor)

Introducción

SSGBD ►

Cliente-
servidor

Arquitectura de esquemas

- Esquema **lógico**

- es la “traducción” del esquema conceptual a un modelo de datos comprensible por un SGBD implementado
 - la primera definición ANSI/SPARC no diferenciaba entre conceptual y lógico
- tablas, tipos de datos, relaciones, restricciones, ...
 - según la implementación del modelo subyacente en el SGBD
- independencia física
 - independiente del sistema operativo, organización de ficheros, tecnología de almacenamiento



Introducción

SSGBD ►

Cliente-servidor

Arquitectura de esquemas

- Esquema **interno**

- muy bonito, ahora “vamos” a hacerlo eficiente
- traducción del esquema lógico a términos entendibles por el sistema operativo
 - ficheros, registros, organización, acceso, índices ...
 - gran parte está determinada por el SO y automatizado en el SGBD
 - decisiones “limitadas” en su diseño por parte del administrador de BD
- Muchos SGBD tienen capacidades limitadas de manipulación del esquema interno
 - muchas veces es transparente al usuario/administrador



Arquitectura de esquemas

- Esquemas externos
 - comunicación entre SGBD y programas de aplicación
 - definición de estructuras de datos propias
 - cobol, C, C++, php, python, ...
 - a rellenar mediante peticiones al SGBD, órdenes *select* desde funciones implementadas en el lenguaje de programación o en librerías
 - el SGBD también puede proporcionar su propio lenguaje de programación
 - generalmente más orientado a manipulación de datos y no tanto a presentación (PL/SQL)

■ independientes del esq. lógico (ind. lógica)

Introducción

SSGBD ►

Cliente-
servidor



```
Private Sub Form_MouseDown(Button As Integer, Shift As Integer, x As
Single, Y As Single)
Dim ReturnValue As Long
If Button = 1 Then
Call ReleaseCapture
ReturnValue = SendMessage(Me.hwnd, WM_NCLBUTTONDOWN,
HTCAPTION, 0&)
End If
End Sub
```



Arquitectura

Introducción

SSGBD ►

Cliente-
servidoresquemas
EXTERNOS

```
( PL / I )
DCL 1 EMPP ,
      2 #EMP CHAR ( 6 ) ,
      2 SAL FIXED BET ( 31 ) ;
```

```
( CÒBOL )
01 EMPC .
      02 EMPNO PIC X ( 6 ) .
      02 DEPTNO PIC X ( 4 ) .
```

esquema
LÓGICO

```
EMPLEAT
  NÚMERO_EMPLEADO      CARÀCTER ( 6 )
  NÚMERO_DEPARTAMENTO   CARÀCTER ( 4 )
  SALARI                NUMÈRIC ( 5 )
```

esquema
INTERNO

```
STORED_EMP    LENGTH=20
  PREFIX      TYPE=BYTE ( 6 ) ,  OFFSET=0
  EMP#        TYPE=BYTE ( 6 ) ,  OFFSET=6 ,  INDEX=EMPX
  DEPT#       TYPE=BYTE ( 4 ) ,  OFFSET=12
  PAY         TYPE=FULLWORD ,  OFFSET=16
```

Administrador

- Responsable del buen funcionamiento del SGBD
 - esquema lógico
 - esquema interno
 - esquemas externos
 - integridad y seguridad
 - recuperación y copias de seguridad
 - rendimiento y cambios en requisitos

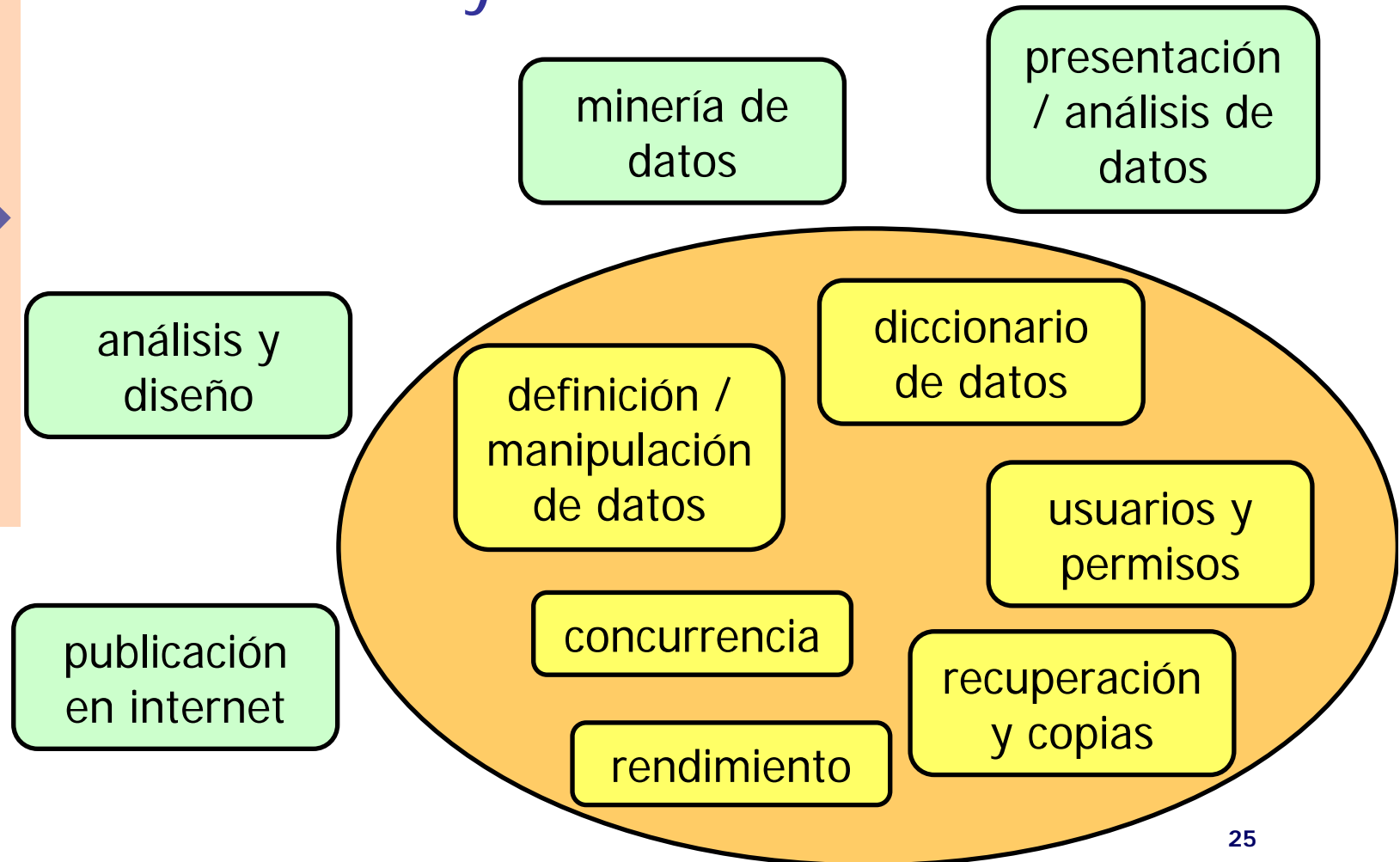
Introducción

SSGBD ►

Cliente-
servidor

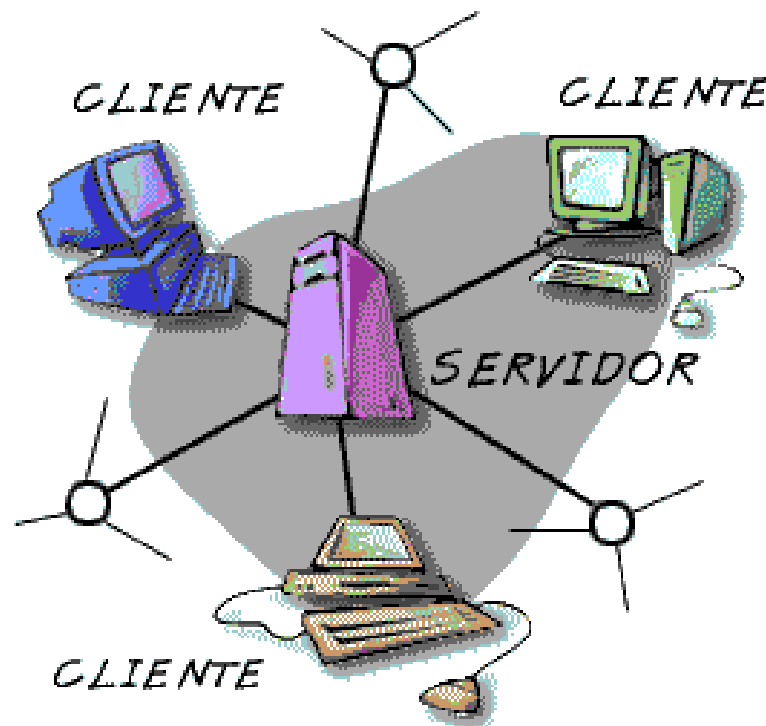
SGBD

- módulos y herramientas auxiliares



Arquitectura cliente-servidor

- Visión de alto nivel de un SGBD como servidor de peticiones



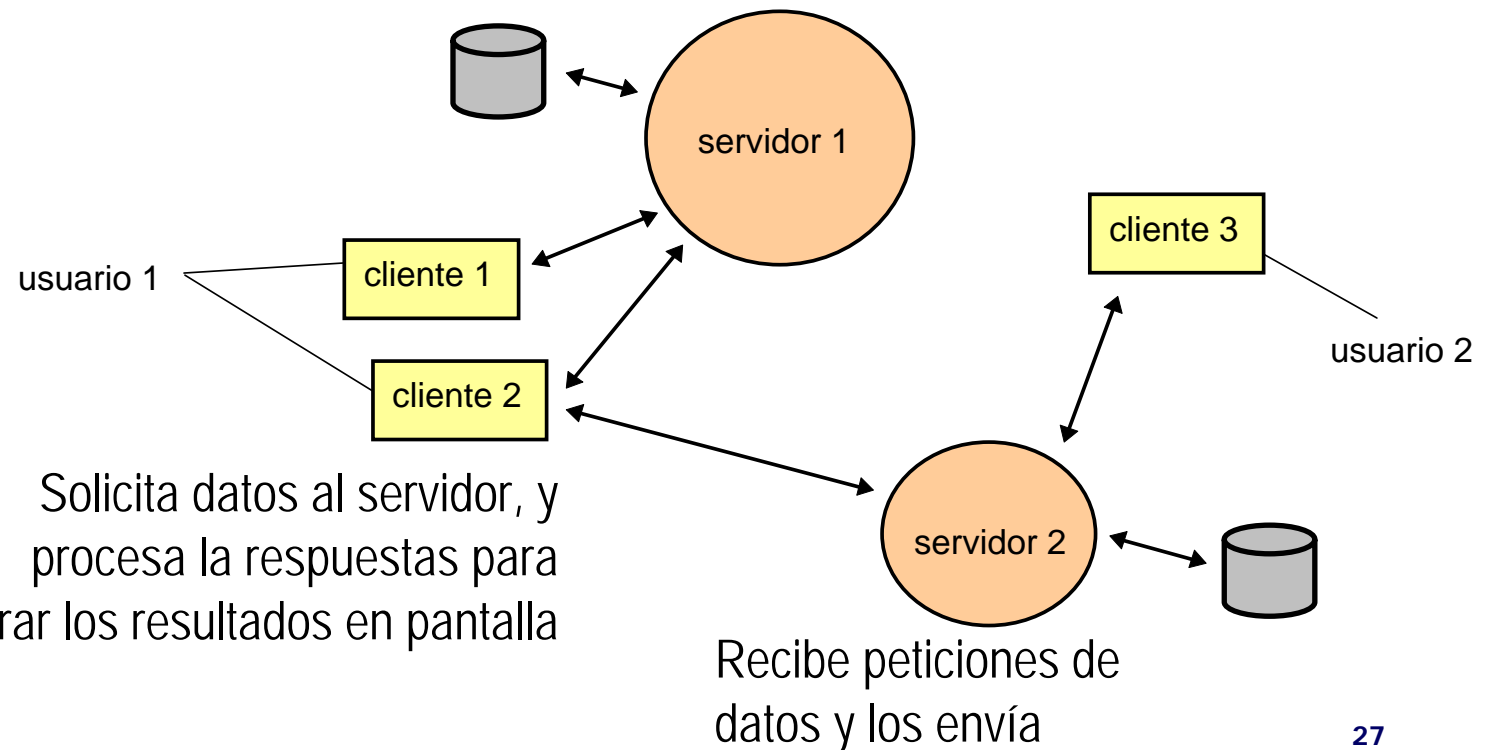
Introducción

SSGBD

Cliente-servidor ▶

Arquitectura cliente-servidor

- descargar al SGBD de labores de presentación
- aprovechar la red, distribución de datos, transparencia, diferentes SGBD



Sistemas de Gestión de Bases de datos

- respuesta a los problemas de dependencia, integridad y seguridad de los sistemas de archivo convencional
- centralizando los datos y proporcionando vistas parciales
- estableciendo varios niveles de definición de datos, de menos a más dependientes del almacenamiento físico
- enriqueciendo la gestión con las herramientas adecuadas, modularidad, rendimiento

Introducción

SSGBD

Cliente-
servidor