



EFA
MORATALAZ

*1º CFGS Desarrollo de
Aplicaciones Multiplataforma*

BASES DE DATOS

JESÚS SANTIAGO RICO

UT1 – INTRODUCCIÓN A LAS BASES DE DATOS





EFA
MORATALAZ

*1º CFGS Desarrollo de Aplicaciones
Multiplataforma*

BASES DE DATOS

UT1 – INTRODUCCIÓN A LAS BASES DE DATOS

1. IMPORTANCIA DEL ALMACENAMIENTO DE DATOS
2. CUALIDADES DE LA INFORMACIÓN
3. LA EVOLUCIÓN HACIA LAS BASES DE DATOS
4. DEFINICIÓN DE BASE DE DATOS
5. OBJETIVOS QUE DEBE CUMPLIR UNA BBDD
6. ARQUITECTURA DE UNA BBDD
7. SISTEMA GESTOR DE BASES DE DATOS (SGBD)
 - 7.1. COMPONENTES DE UN SGBD
8. ARQUITECTURA CLIENTE-SERVIDOR

IMPORTANCIA DEL ALMACENAMIENTO DE DATOS

1

Actualmente el mundo empresarial maneja gran cantidad de datos. Esto puede conducir a:

- **La explosión de la información.** Si no se trata bien el problema.
 - Cifras de los datos generados en un minuto.
 - 695.000 stories en Instagram.
 - 69 millones de mensajes en WhatsApp.
 - 500 horas de contenido en YouTube.
- **Polución informativa.**
 - La información puede perder sus cualidades, no puede cumplir sus objetivos.
 - Puede ser más nociva que beneficiosa.

1. Importancia del almacenamiento de los datos



Fuente: Lori Lewis vía AllAccess

QUALIDADES DE LA INFORMACIÓN



2. Cualidades de la información

- **PRECISIÓN:** Porcentaje de información correcta sobre la información del sistema.
 - Si queremos que la información que hay en el ordenador sea precisa debemos suministrarle datos correctos.
 - Baja precisión implica baja credibilidad.
- **OPORTUNIDAD.** La oportunidad se refiere al tiempo transcurrido desde el momento que se produjo el hecho que originó el dato hasta el momento que la información se pone a disposición del usuario.
 - En general, el valor de la información disminuye con el tiempo.
 - Normalmente la oportunidad resta precisión.
- **COMPLECIÓN.** La información debe ser lo suficientemente completa para cumplir con sus objetivos.
 - Ejemplo: Para poner la nota final de un alumno, se necesitan todas las notas anteriores de ese alumno.
 - La compleción depende de: los datos existentes en el sistema de información y de la capacidad de recuperar todos esos datos cuando se le consulte.

- **SIGNIFICADO:** La información debe ser comprensible e interesante, lo que supone no proporcionar grandes masas de información que por su volumen no puedan ser asimiladas.
- **COHERENCIA:**
 - Toda información debe ser consistente con las reglas del mundo real al que debe representar lo más fielmente posible.
 - Ejemplo: El sistema que almacena la información sobre el expediente del alumnado no debe permitir almacenar un 13 como nota de una asignatura.
- **SEGURIDAD:**
 - La información ha de ser protegida del deterioro.
 - Accesos no autorizados.

LA EVOLUCIÓN HACIA LAS BASES DE DATOS



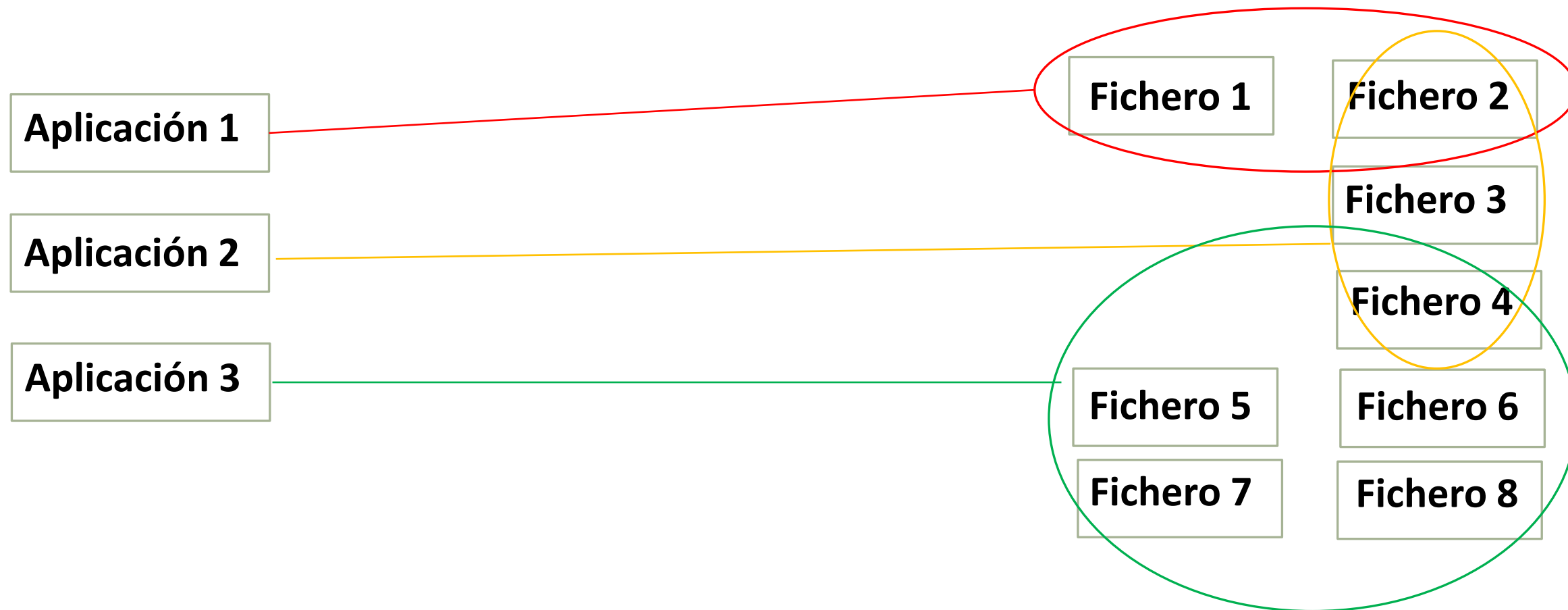
Tradicionalmente los sistemas de información de las empresas se componían de:

- Grandes ficheros en papel.
- Posteriormente, programas escritos en lenguajes de alto nivel (C, fortran, pascal, etc.) que gestionaban la información que estaba almacenada en ficheros.

El programador era el encargado de almacenar y recuperar la información de los ficheros y dicha información debía cumplir las cualidades vistas anteriormente.

!!!!Enorme complejidad del diseño de dicho programa!!!!

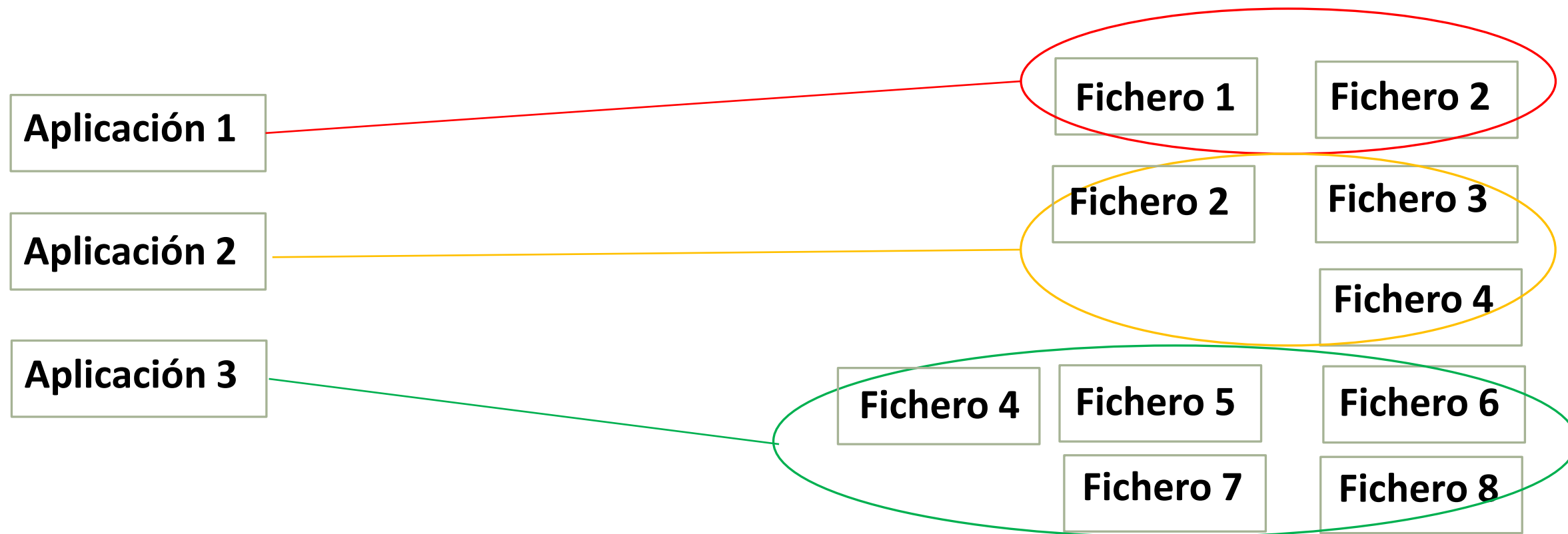
Muchas aplicaciones comparten los mismos ficheros



Muchas aplicaciones comparten los mismos ficheros → DESVENTAJAS

- Si necesitamos variar la estructura del registro añadiendo un campo más, tendríamos que cambiar toda la estructura del fichero y todos los programas que utilizan ese fichero.
- **Lentitud**, cada aplicación trabaja con más información de la que necesita.

Cada aplicación tiene sus propios ficheros, aunque se compartan



Cada aplicación tiene sus propios ficheros, aunque se compartan → DESVENTAJAS

- Los datos no se comparten sino que se duplican, existe **redundancia** → **polución informativa**.
- **Lentitud**, debido al control de la redundancia.

Otras DESVENTAJAS por el uso de ficheros

- Control de accesos concurrentes.
- Recuperación de ficheros por fallos de disco o errores del software.
- La vigilancia de accesos (confidencialidad).

Todas estas desventajas nos lleva a la necesidad de un nuevo enfoque

- **La independencia de los datos con respecto a las aplicaciones que los utilizan de modo que ningún cambio en la estructura de datos afectará a los programas que los usen → BASES DE DATOS**

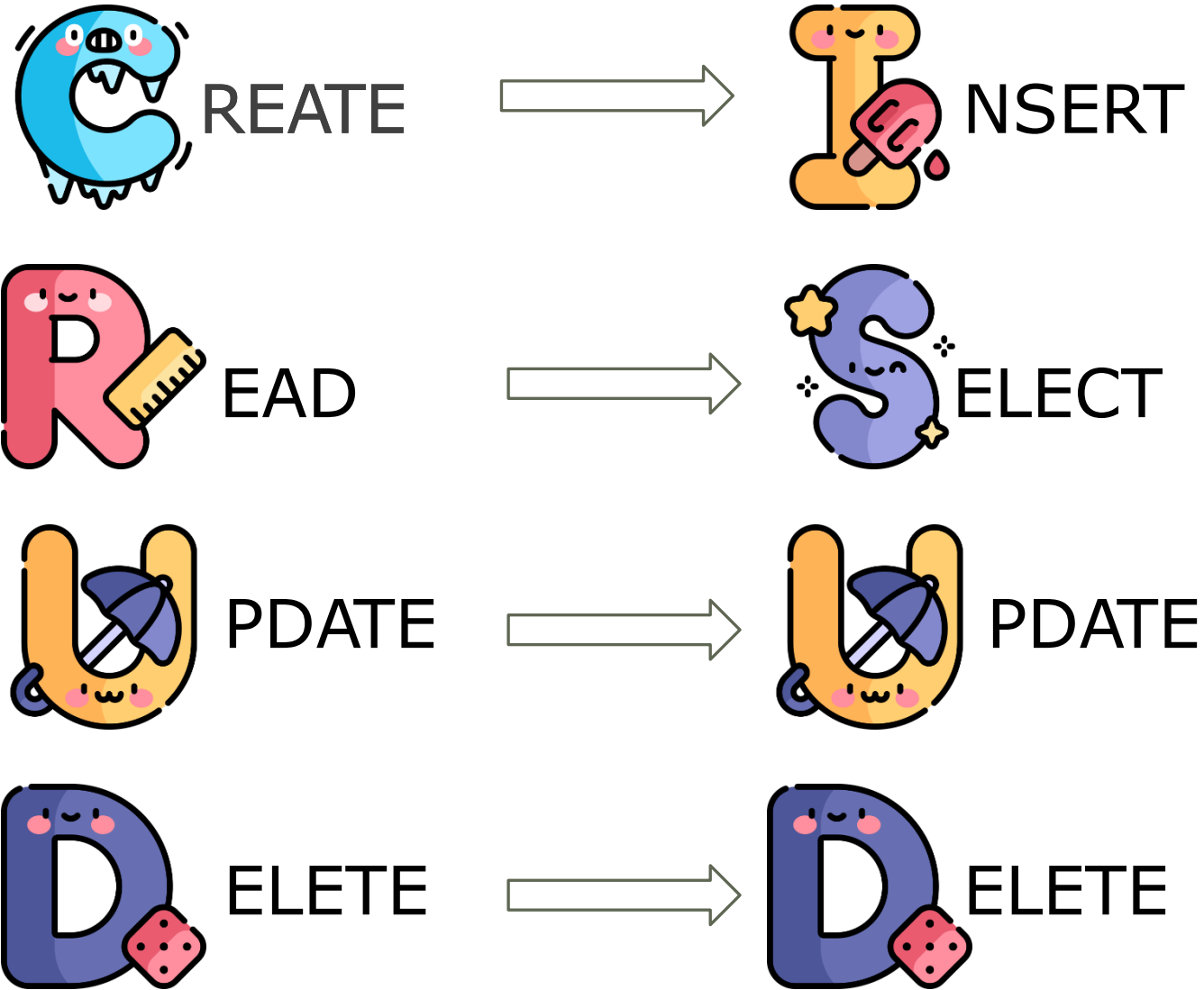
DEFINICIÓN DE BASE DE DATOS



Una **base de datos** es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático. Se caracterizan por:

- Servir a las aplicaciones de la mejor manera posible → **información sin polución.**
- Existir independencia entre datos y programas que lo manejan.

4. Definición de base de datos



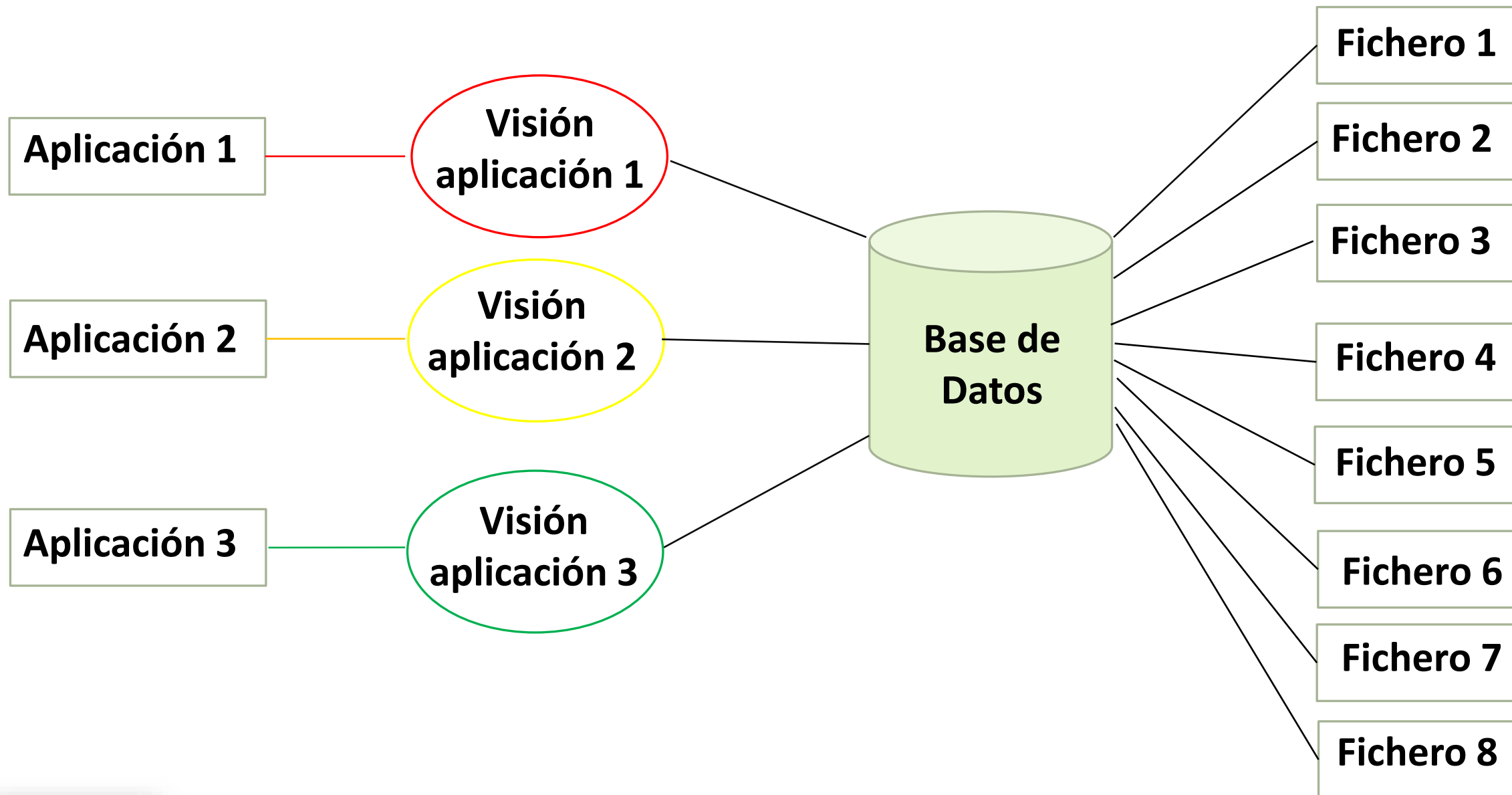
En una base de datos se guardan información de dos tipos:

- **Datos de usuario:** Son los datos de las aplicaciones de usuario.

Ejemplo: Datos del alumno como: apellido, nombre, asignaturas, etc.

- **Datos del sistema:** Son los datos que necesita la base de datos para gestionarse a sí misma.

Ejemplo: Datos de usuarios que acceden a la base de datos como contraseña, privilegios, espacio asignado, etc.



VENTAJAS:

- Las aplicaciones no saben en que ficheros está la información.
- La aplicación trabaja con la visión que le proporciona la base de datos.
- Si una aplicación quiere añadir más información sólo tiene que ampliar su visión.

OBJETIVOS QUE DEBE CUMPLIR UNA BBDD

5

5. Objetivos que debe cumplir una BBDD

- **VERSATILIDAD:** Los datos podrán utilizarse de múltiples maneras.
- **RENDIMIENTO.** Los datos deben ser suministrados con la mayor rapidez posible.
- **SEGURIDAD Y PRIVACIDAD:** Los datos deberán protegerse frente a pérdidas y frente a usos indebidos o accesos no autorizados.
- **INTEGRIDAD:** La información almacenada en la base de datos debe cumplir ciertas restricciones definidas específicamente por el usuario. El sistema de bases de datos debe determinar si las actualizaciones suponen o no la violación de dichas restricciones.
- **INDEPENDENCIA FÍSICA DE LOS DATOS:** Cuando se produzcan modificaciones en la organización física de los datos, no debe afectar a los programas de aplicación que ya hay en uso.
- **INDEPENDENCIA LÓGICA DE LOS DATOS:** Cuando se cambia la estructura lógica de los datos, por ejemplo, se agregan nuevos campos, esto no debe afectar a los programas de aplicación que utilicen esos datos.
- **AFINACIÓN:** Ajustar la organización física de los datos con el objetivo de mejorar la rapidez de acceso a los mismos. Los cambios en la organización física de los datos no deben afectar a los programas de aplicación que hay en uso (**independencia física de los datos**).

ARQUITECTURA DE UNA BBDD



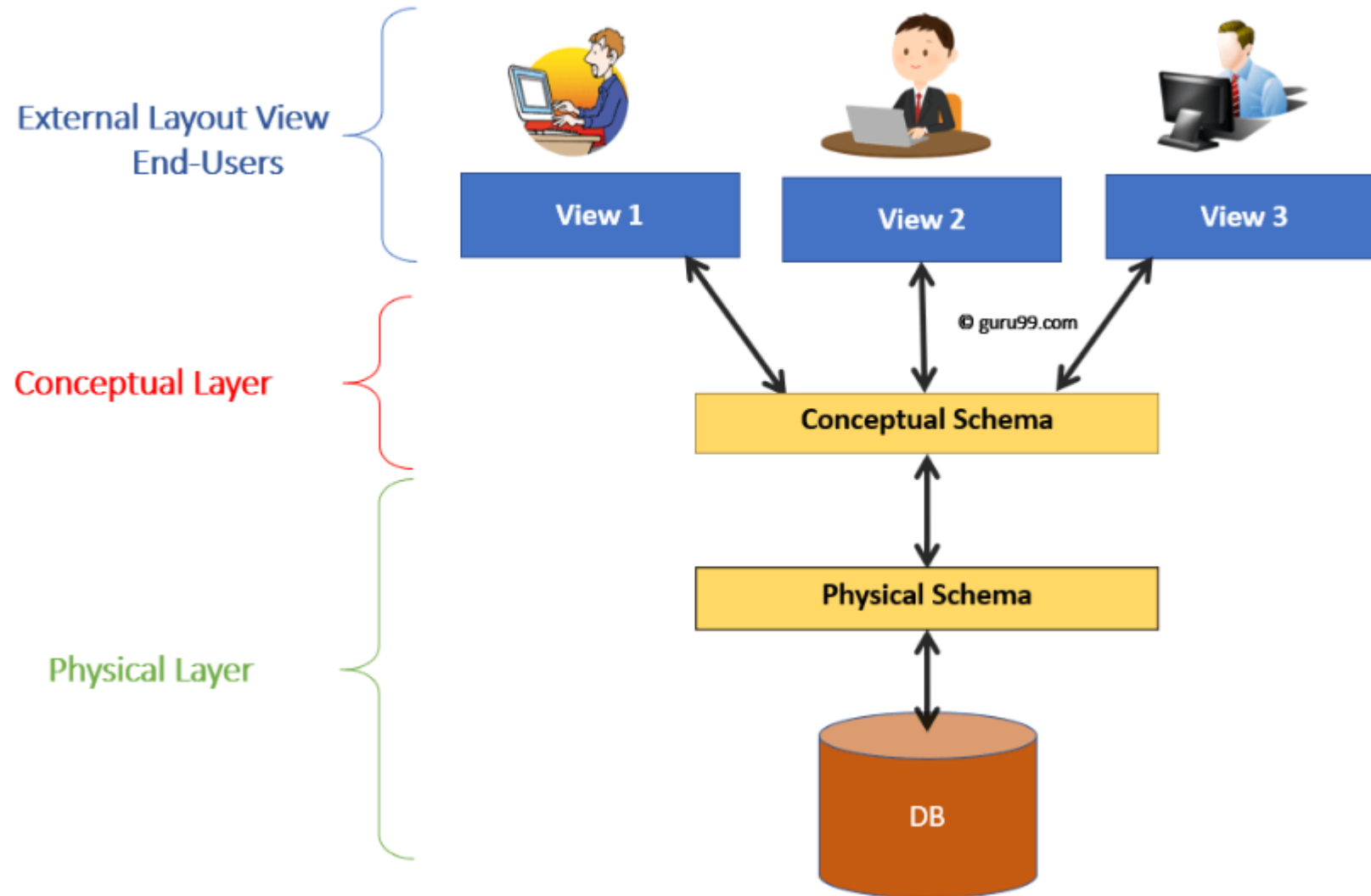
La arquitectura de los sistemas de bases de datos es por **niveles** igual que las de los sistemas en red:

- Los usuarios deben tener una visión lo más abstracta posible.
- No tienen porque saber como están organizados los datos.
- Los datos se deben presentar al usuario final de forma que este sea capaz de interpretarlos y manejarlos.
- No necesita tener la misma visión el informático que administra la base de datos que el administrativo que consulta las nóminas de los empleados.

La arquitectura de los sistemas de bases de datos se compone de tres capas:

- **Capa interna o física:** Se describe la estructura física de la base de datos mediante un esquema interno. En este nivel se describen los ficheros que contienen la información, su organización, ubicación, la forma de acceso a sus registros, su longitud, sus índices.
- **Capa global o conceptual:** Se describe la estructura de toda la base de datos a partir de los requerimientos de los usuarios. En este nivel se ocultan los detalles de almacenamiento y se centra en describir entidades, atributos, relaciones y restricciones.
- **Capa externa:** Es el nivel más cercano al usuario. Representa la visión individual de un usuario o grupo de usuarios de la base de datos.

6. Arquitectura de una BBDD



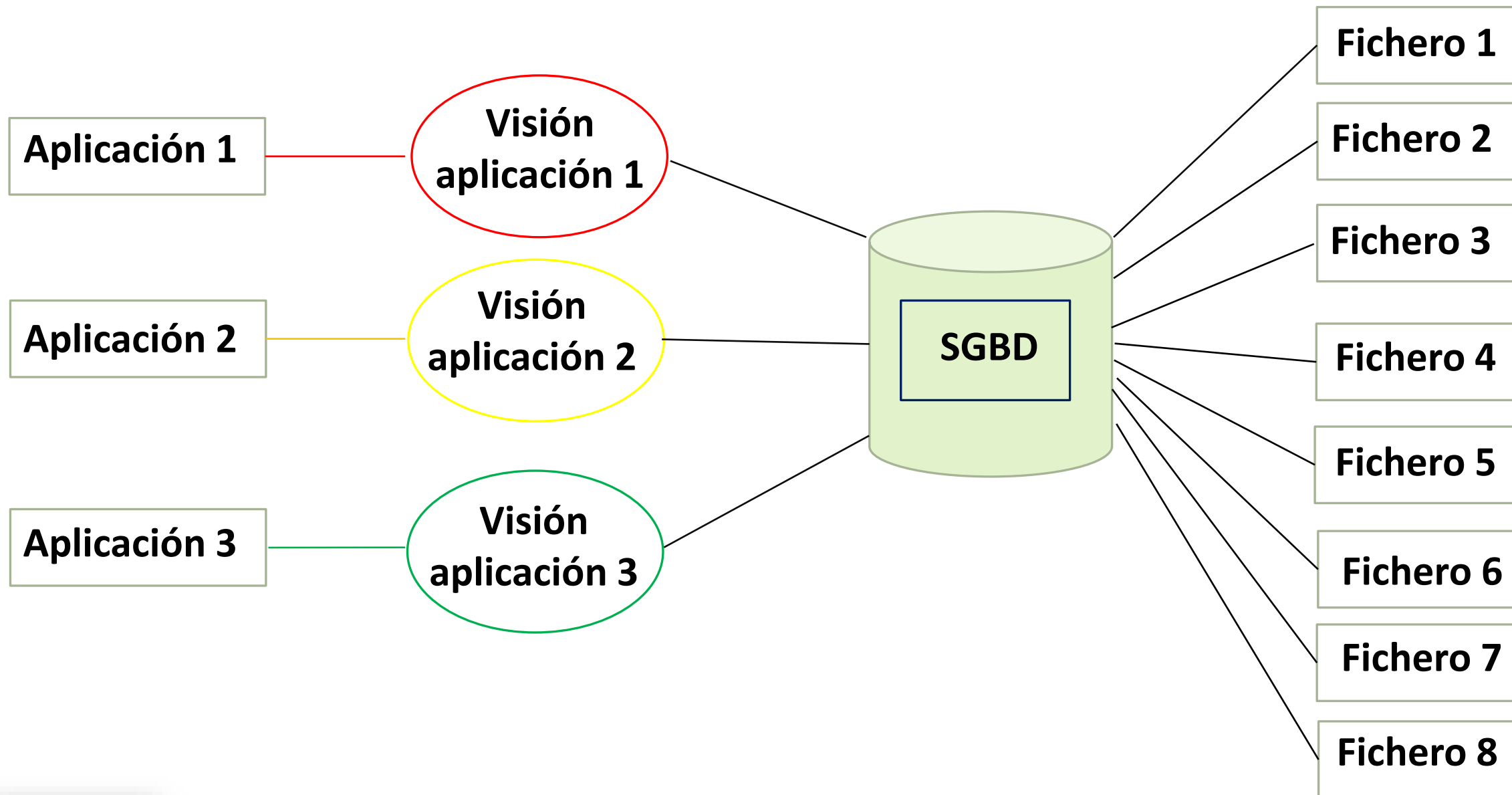
SISTEMA GESTOR DE BASES DE DATOS (SGBD)



El **Sistema Gestor de Base de Datos** es un conjunto de programas que permiten administrar y gestionar la información de una base de datos.

El **SGBD** proporciona al usuario diferentes facilidades para realizar las siguientes tareas:

- Definición de los datos en distintos niveles de abstracción.
- Manipulación de datos (inserción, borrado y consulta).
- Mantenimiento de la integridad de los datos (los datos que se almacenen en la BD debe atender una serie de restricciones definidas en el SGBD).
- Control de privacidad y seguridad de los datos en la base de datos.
- Controlar los accesos concurrentes.



COMPONENTES DE UN SGBD

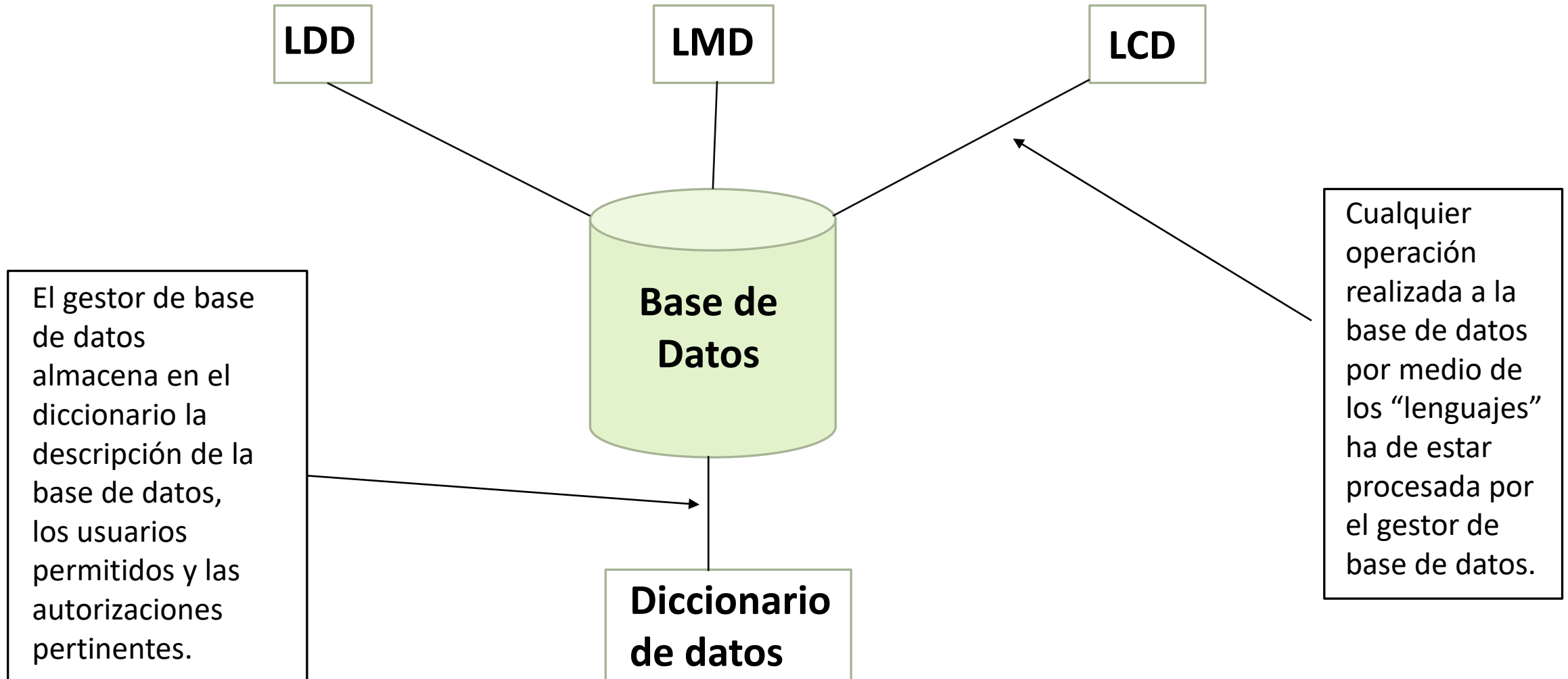
7.1

El **Sistema Gestor de Base de Datos** se compone de:

- **Diccionario de datos.** Se trata de una serie de esquemas que describen el contenido del SGBD, incluyendo distintos objetos con sus propiedades. Algunos de estos objetos son:
 - Restricciones de tablas.
 - Funciones y procedimientos almacenados.
 - Disparadores o Triggers.
- **Lenguajes de datos.** Son lenguajes para la manipulación, acceso, modificación, control, seguridad.... de los datos.
 - Lenguaje de definición de datos (LDD).
 - Lenguaje de manipulación de datos (LMD).
 - Lenguaje de control de datos (LCD).

- **Lenguaje de definición de datos (LDD).** Se utiliza para definir el esquema global y los distintos subesquemas de la base de datos.
- **Lenguaje de Manipulación de datos (LMD).** Nos permite insertar, modificar, eliminar y consultar datos.
- **Lenguaje de control de datos (LCD).** Control el acceso a la información de la base de datos. Define privilegios, permisos y tipos de acceso, así como control de seguridad.

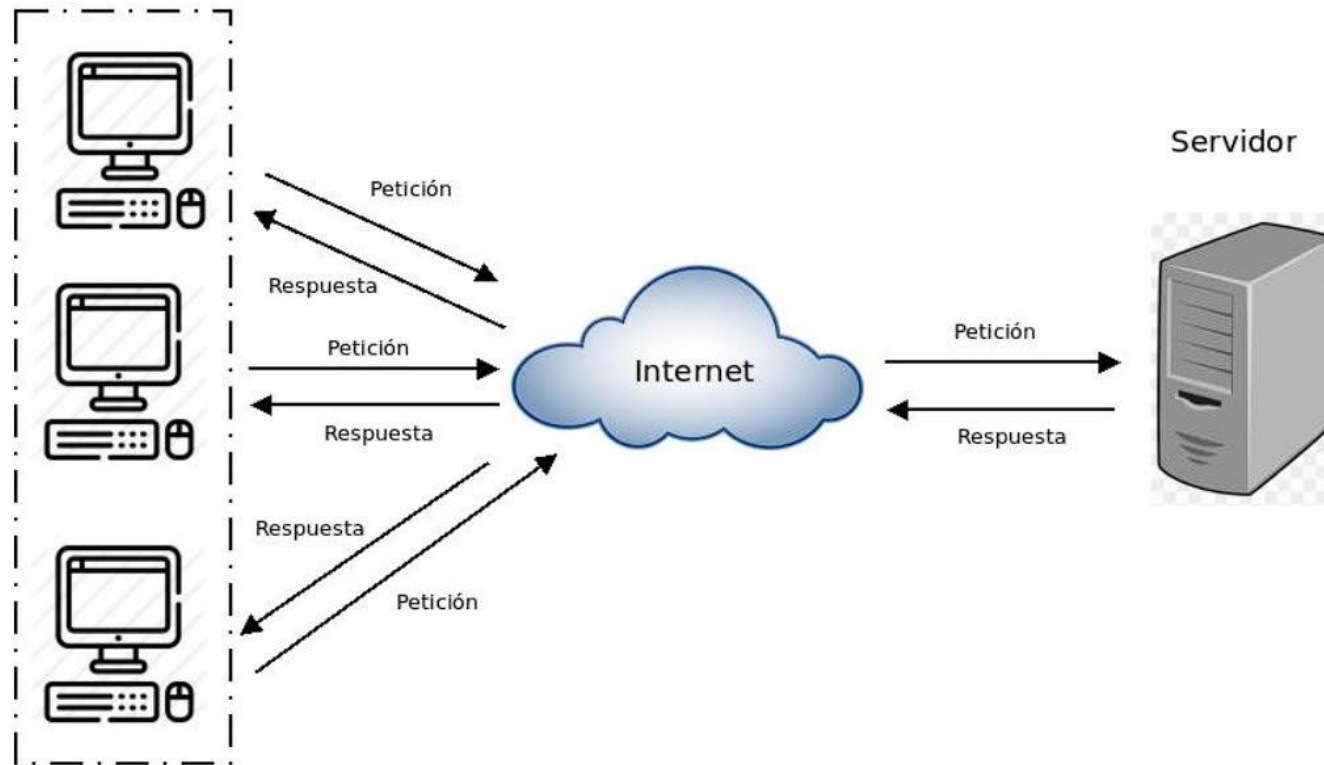
7.1. Componentes de un SGBD



ARQUITECTURA CLIENTE-SERVIDOR



La **arquitectura cliente-servidor** es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados **servidores**, y los demandantes, llamados **clientes**. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta



Uno o varios clientes acceden a un servidor. Las nuevas arquitecturas sustituyen el servidor por un balanceador de carga de manera que N servidores dan respuesta a M clientes.

El software adecuado para una arquitectura cliente servidor posee varios componentes:

- **Software de gestión de datos:** Lleva a cabo la manipulación y gestión de datos requerido por las aplicaciones. Normalmente en el servidor.
- **Software de desarrollo:** Se usa para desarrollar aplicaciones. Normalmente en el cliente.
- **Software de interacción con el usuario:** Implementa funciones que se asocian a la interfaz gráfica de usuario. Normalmente en el cliente.