

# Concepto y Tipologías de Bases de Datos

## Introducción a las Bases de Datos

*Un Sistema Gestor de Bases de Datos (SGBD) o según sus siglas en inglés DBMS (Database Management System) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. La colección de datos, normalmente denominada base de datos, contiene información relevante para una empresa. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente.*

Los sistemas de bases de datos se diseñan para gestionar grandes cantidades de información. La gestión de los datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de la información. Además, los sistemas de bases de datos deben proporcionar la fiabilidad de la información almacenada, a pesar de las caídas del sistema o los intentos de acceso sin autorización. Si los datos van a ser compartidos entre diversos usuarios, el sistema debe evitar posibles resultados anómalos.

Dado que la información es tan importante en la mayoría de las organizaciones, los científicos informáticos han desarrollado un amplio conjunto de conceptos y técnicas para la gestión de los datos.

## Aplicaciones de los Sistemas de Bases de Datos

Los sistemas de bases de datos tienen una amplia gama de aplicaciones en diversos sectores debido a su capacidad para almacenar, gestionar y recuperar grandes volúmenes de datos de manera eficiente. Algunas aplicaciones clave pueden ser:

### Aplicaciones Empresariales

**Sistemas de Gestión de Clientes (CRM):** Utilizados por empresas para almacenar y gestionar información sobre clientes, mejorar el servicio y personalizar las interacciones.

**Sistemas de Gestión de Recursos Humanos (HRMS):** Bases de datos que almacenan información de empleados, desde datos personales hasta registros de salario, vacaciones y evaluaciones de desempeño.

**ERP (Enterprise Resource Planning):** Integran diversos procesos empresariales como la contabilidad, finanzas, compras, y producción en una sola plataforma que depende de una base de datos centralizada.

### E-commerce

**Gestión de Inventario:** Aplicaciones que utilizan bases de datos para rastrear productos, stock, proveedores y ventas.

**Personalización de la Experiencia del Usuario:** Se almacenan datos sobre el comportamiento y preferencias de los usuarios para ofrecer recomendaciones personalizadas en sitios de compras en línea.

### Sector Salud

**Historiales Clínicos Electrónicos (EHR):** Bases de datos que almacenan información médica de pacientes, incluyendo diagnósticos, tratamientos, prescripciones, y resultados de laboratorio.

**Gestión de Hospitales:** Sistemas que gestionan información sobre camas, personal, pacientes, y recursos médicos.

### Sector Bancario y Financiero

**Gestión de Cuentas y Transacciones:** Las bases de datos son fundamentales para el registro de transacciones, saldos, préstamos y operaciones financieras.

**Sistemas de Prevención de Fraude:** Algoritmos que analizan datos en tiempo real para detectar patrones de fraude y realizar alertas automáticas.

### Redes Sociales

**Gestión de Usuarios y Contenido:** Almacenan grandes cantidades de datos de usuarios como perfiles, mensajes, publicaciones, y conexiones entre amigos o seguidores.

**Recomendaciones:** Los algoritmos de recomendación se basan en datos de comportamiento almacenados en bases de datos para sugerir amigos, contenido o publicidad relevante.

### Ciberseguridad

**Detección de Amenazas:** Los sistemas de seguridad recopilan datos de acceso y uso en bases de datos para detectar patrones sospechosos o amenazas de seguridad.

**Análisis Forense:** En incidentes de seguridad, se consultan bases de datos para analizar el origen y el alcance de la brecha.

A lo largo de las últimas cuatro décadas del siglo veinte, el uso de las bases de datos creció en todas las empresas. Las organizaciones han transformado muchas de sus antiguas interfaces telefónicas en interfaces web, permitiendo a los usuarios acceder a diversos servicios en línea. Por ejemplo, al buscar un libro o música en una tienda en línea, se accede a datos almacenados en una base de datos. Los pedidos en línea también se registran en una base de datos. En el caso de los bancos, al consultar el estado de una cuenta o movimientos desde su sitio web, la información proviene de su sistema de bases de datos. Además, la información personal almacenada en bases de datos se utiliza para personalizar la publicidad y registrar accesos web.

## Sistemas de Archivos Convencionales vs Sistemas Bases de Datos

El sistema de procesamiento de archivos que utilizan algunas empresas, como una caja de ahorros, permite almacenar y gestionar información de clientes y cuentas mediante programas de aplicación que manejan archivos, por ejemplo, para hacer abonos, calcular saldos o generar operaciones. Sin embargo, este enfoque tiene varios inconvenientes:

1. **Redundancia e inconsistencia de datos:** Diferentes programadores pueden crear archivos con formatos distintos, generando duplicación de información y posibles inconsistencias, como cambios no reflejados en todos los archivos.
2. **Dificultad en el acceso a los datos:** Si surge una nueva necesidad, como listar clientes en un distrito específico, es necesario crear un nuevo programa o hacerlo manualmente, lo cual es ineficiente.
3. **Aislamiento de datos:** Los datos dispersos en varios archivos dificultan la creación de nuevos programas de aplicación para recuperar información específica.
4. **Problemas de integridad:** Las restricciones de consistencia, como un saldo mínimo en cuentas, son difíciles de mantener cuando se deben modificar múltiples programas.
5. **Problemas de atomicidad:** En caso de fallos, como en una transferencia de fondos, se corre el riesgo de que los datos queden en un estado inconsistente.
6. **Anomalías en el acceso concurrente:** Si varios usuarios actualizan los datos simultáneamente, pueden producirse inconsistencias, como un saldo incorrecto en una cuenta.
7. **Problemas de seguridad:** Es complicado controlar qué usuarios tienen acceso a qué información, lo que puede comprometer la seguridad.

Estos problemas motivaron el desarrollo de los Sistemas de Gestión de Bases de Datos (SGBD), que resuelven estas dificultades proporcionando una mejor gestión de datos en términos de acceso, integridad, atomicidad, seguridad, y concurrencia.

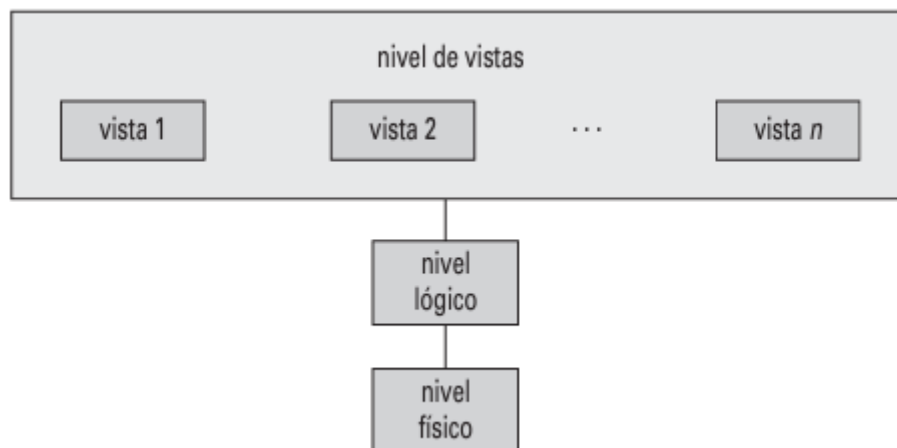
## Abstracción de datos

Un sistema de bases de datos es una colección de archivos interrelacionados y un conjunto de programas que permiten a los usuarios acceder y modificar estos archivos. Uno de los propósitos principales de un sistema de bases de datos es proporcionar a los usuarios una visión abstracta de los datos. Es decir, el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos.

Para que el sistema sea útil debe **recuperar los datos eficientemente**. Esta preocupación ha conducido al diseño de estructuras de datos complejas para la representación de los datos en la base de datos. Como muchos usuarios de sistemas de bases de datos no están familiarizados con computadores, los desarrolladores esconden la complejidad a los usuarios a través de varios niveles de abstracción para simplificar la interacción de los usuarios con el sistema:

- **Nivel físico:** El nivel más bajo de abstracción describe cómo se almacenan realmente los datos. En el nivel físico se describen en detalle las estructuras de datos complejas de bajo nivel.
- **Nivel lógico:** El siguiente nivel más alto de abstracción describe qué datos se almacenan en la base de datos y qué relaciones existen entre esos datos. La base de datos completa se describe así en términos de un número pequeño de estructuras relativamente simples. Aunque la implementación de estructuras simples en el nivel lógico puede involucrar estructuras complejas del nivel físico, los usuarios del nivel lógico no necesitan preocuparse de esta complejidad. Los administradores de bases de datos, que deben decidir la información que se mantiene en la base de datos, usan el nivel lógico de abstracción.

- **Nivel de vistas:** El nivel más alto de abstracción describe sólo parte de la base de datos completa. A pesar del uso de estructuras más simples en el nivel lógico, queda algo de complejidad, debido a la variedad de información almacenada en una gran base de datos. Muchos usuarios del sistema de base de datos no necesitan toda esta información. En su lugar, tales usuarios necesitan acceder sólo a una parte de la base de datos. Para que su interacción con el sistema se simplifique, se define la abstracción del nivel de vistas. El sistema puede proporcionar muchas vistas para la misma base de datos.



La figura anterior muestra la relación entre los tres niveles de abstracción. Una analogía con el concepto de tipos de datos en lenguajes de programación puede clarificar la distinción entre los niveles de abstracción. La mayoría de lenguajes de programación de alto nivel soportan la estructura de tipo registro.

## Esquemas de Bases de Datos

Las bases de datos van cambiando a lo largo del tiempo conforme la información se inserta y borra. La colección de información almacenada en la base de datos en un momento particular se denomina un ejemplar de la base de datos. El diseño completo de la base de datos se llama el esquema de la base de datos. Los esquemas son raramente modificados, si es que lo son alguna vez. El concepto de esquemas y ejemplares de bases de datos se puede entender por analogía con un programa escrito en un lenguaje de programación. Un esquema de base de datos corresponde a las declaraciones de variables (junto con definiciones de tipos asociadas) en un programa. Cada variable tiene un valor particular en un instante de tiempo. Los valores de las variables en un programa en un instante de tiempo corresponde a un ejemplar de un esquema de bases de datos. Los sistemas de bases de datos tiene varios esquemas divididos de acuerdo a los niveles de abstracción que se han discutido. El esquema físico describe el diseño físico en el nivel físico, mientras que el esquema lógico describe el diseño de la base de datos en el nivel lógico. Una base de datos puede tener también varios esquemas en el nivel de vistas, a menudo denominados subesquemas, que describen diferentes vistas de la base de datos. De éstos, el esquema lógico es con mucho el más importante, en términos de su efecto en los programas de aplicación, ya que los programadores construyen las aplicaciones usando el esquema lógico. El esquema físico está oculto bajo el esquema lógico, y puede ser fácilmente cambiado usualmente sin afectar a los programas de aplicación. Los programas de aplicación se dice que muestran independencia física de datos si no dependen del esquema físico y, por tanto, no deben ser modificados si cambia el esquema físico. Se estudiarán los lenguajes para la descripción de los esquemas, después de introducir la noción de modelos de datos en el siguiente apartado.

## Arquitectura de un SGBD

Un sistema de bases de datos se divide en módulos que se encargan de cada una de las responsabilidades del sistema completo. Los componentes funcionales de un sistema de bases de datos se pueden dividir a grandes rasgos en los componentes gestor de almacenamiento y procesador de consultas. El gestor de consultas es importante porque las bases de datos requieren normalmente una gran cantidad de espacio de almacenamiento. Las bases de datos corporativas tienen un tamaño de entre cientos de gigabytes y, para las mayores bases de datos, terabytes de datos. Un gigabyte son 1.000 megabytes (1.000 millones de bytes), y un terabyte es 1 millón de megabytes (1 billón de bytes). Debido a que la memoria principal de los computadores no puede almacenar esta gran cantidad de información, esta se almacena en discos. Los datos se trasladan entre el disco de almacenamiento y la memoria principal cuando es necesario. Como la transferencia de datos a y desde el disco es lenta comparada con la velocidad de la unidad central de procesamiento, es fundamental que el sistema de base de datos estructure los datos para minimizar la necesidad de movimiento de datos entre el disco y la memoria principal. El procesador de consultas es importante porque ayuda al sistema de bases de datos a simplificar y facilitar el acceso a los datos. Las vistas de alto nivel ayudan a conseguir este objetivo. Con ellas, los usuarios del sistema no deberían ser molestados innecesariamente con los detalles físicos de implementación del sistema. Sin embargo, el rápido procesamiento de las actualizaciones y de las consultas es importante. Es trabajo del sistema de bases de datos traducir las actualizaciones y las consultas escritas en un lenguaje no procedimental, en el nivel lógico, en una secuencia de operaciones en el nivel físico.

### Gestor de almacenamiento

Un gestor de almacenamiento es un módulo de programa que proporciona la interfaz entre los datos de bajo nivel en la base de datos y los programas de aplicación y consultas emitidas al sistema. El gestor de almacenamiento es responsable de la interacción con el gestor de archivos. Los datos en bruto se almacenan en disco usando un sistema de archivos, que está disponible habitualmente en un sistema operativo convencional. El gestor de almacenamiento traduce las diferentes instrucciones a órdenes de un sistema de archivos de bajo nivel. Así, el gestor de almacenamiento es responsable del almacenamiento, recuperación y actualización de los datos en la base de datos. Los componentes del gestor de almacenamiento incluyen:

- Gestor de autorización e integridad, que comprueba que se satisfagan las restricciones de integridad y la autorización de los usuarios para acceder a los datos.
- Gestor de transacciones, que asegura que la base de datos quede en un estado consistente (correcto) a pesar de los fallos del sistema, y que las ejecuciones de transacciones concurrentes ocurran sin conflictos.
- Gestor de archivos, que gestiona la reserva de espacio de almacenamiento de disco y las estructuras de datos usadas para representar la información almacenada en disco.
- Gestor de memoria intermedia, que es responsable de traer los datos del disco de almacenamiento a memoria principal y decidir qué datos tratar en memoria caché. El gestor de memoria intermedia es una parte crítica del sistema de bases de datos, ya que permite que la base de datos maneje tamaños de datos que son mucho mayores que el tamaño de la memoria principal. El gestor de almacenamiento implementa varias estructuras de datos como parte de la implementación física del sistema:

- Archivos de datos, que almacenan la base de datos en sí.
- Diccionario de datos, que almacena metadatos acerca de la estructura de la base de datos, en particular, el esquema de la base de datos.
- Índices, que proporcionan acceso rápido a elementos de datos que tienen valores particulares.

## Gestor de transacciones

Varias operaciones sobre la base de datos forman a menudo una única unidad lógica de trabajo. Un ejemplo es la transferencia de fondos, en el que una cuenta (A) se carga y otra cuenta (B) se abona. Claramente es esencial que, o bien tanto el cargo como el abono tengan lugar, o bien no ocurra ninguno. Es decir, la transferencia de fondos debe ocurrir por completo o no ocurrir en absoluto. Este requisito de todo o nada se denomina **atomicidad**. Además, es esencial que la ejecución de la transferencia de fondos preserve la consistencia de la base de datos. Es decir, el valor de la suma  $A + B$  se debe preservar. Este requisito de corrección se llama **consistencia**. Finalmente, tras la ejecución correcta de la transferencia de fondos, los nuevos valores de las cuentas A y B deben persistir, a pesar de la posibilidad de fallo del sistema. Este requisito de persistencia se llama **durabilidad**. Una transacción es una colección de operaciones que se lleva a cabo como una única función lógica en una aplicación de bases de datos. Cada transacción es una unidad de **atomicidad** y consistencia. Así, se requiere que las transacciones no violen ninguna restricción de consistencia de la base de datos. Es decir, si la base de datos era consistente cuando la transacción comenzó, la base de datos debe ser consistente cuando la transacción termine con éxito. Sin embargo, durante la ejecución de una transacción, puede ser necesario permitir inconsistencias temporalmente, ya que o el cargo de A o el abono de B se debe realizar uno antes que otro. Esta inconsistencia temporal, aunque necesaria, puede conducir a dificultades si ocurre un fallo. Es responsabilidad del programador definir adecuadamente las diferentes transacciones, de tal manera que cada una preserve la consistencia de la base de datos. Por ejemplo, la transacción para transferir fondos de la cuenta A a la cuenta B se podría definir como compuesta de dos operaciones separadas: una que carga la cuenta A y otra que abona la cuenta B. La ejecución de estas dos operaciones uno después del otro preservará realmente la consistencia. Sin embargo, cada operación en sí misma no transforma la base de datos de un estado consistente en otro nuevo estado consistente. Así, estas operaciones no son transacciones. Asegurar las propiedades de **atomicidad** y **durabilidad** es responsabilidad del propio sistema de bases de datos, específicamente del componente de gestión de transacciones. En ausencia de fallos, toda transacción completada con éxito y atómica se archiva fácilmente. Sin embargo, debido a diversos tipos de fallos, una transacción puede no siempre completar su ejecución con éxito. Si se asegura la propiedad de atomicidad, una transacción que falle no debe tener efecto en el estado de la base de datos. Así, la base de datos se restaura al estado en que estaba antes de que la transacción en cuestión comenzara su ejecución. El sistema de bases de datos debe realizar la recuperación de fallos, es decir, detectar los fallos del sistema y restaurar la base de datos al estado que existía antes de que ocurriera el fallo. Finalmente, cuando varias transacciones actualizan la base de datos concurrentemente, la consistencia de los datos puede no ser preservada, incluso aunque cada transacción individualmente sea correcta. Es responsabilidad del **gestor de control de concurrencia** controlar la interacción entre las transacciones concurrentes para asegurar la consistencia de la base de datos. Los sistemas de bases de datos diseñados para uso sobre pequeños computadores personales pueden no tener todas las características vistas. Por ejemplo, muchos sistemas pequeños imponen la restricción de permitir el acceso a un único usuario a la base de datos en un instante de tiempo. Otros dejan las tareas de copias de seguridad y recuperación a los usuarios. Estas restricciones permiten un gestor de datos más pequeño, con menos requisitos de recursos físicos, especialmente de memoria principal. Aunque tales enfoques de bajo coste y prestaciones son

suficientes para bases de datos personales pequeñas, son inadecuadas para satisfacer las necesidades de una empresa de media a gran escala.

## Usuarios del SGBD y administradores

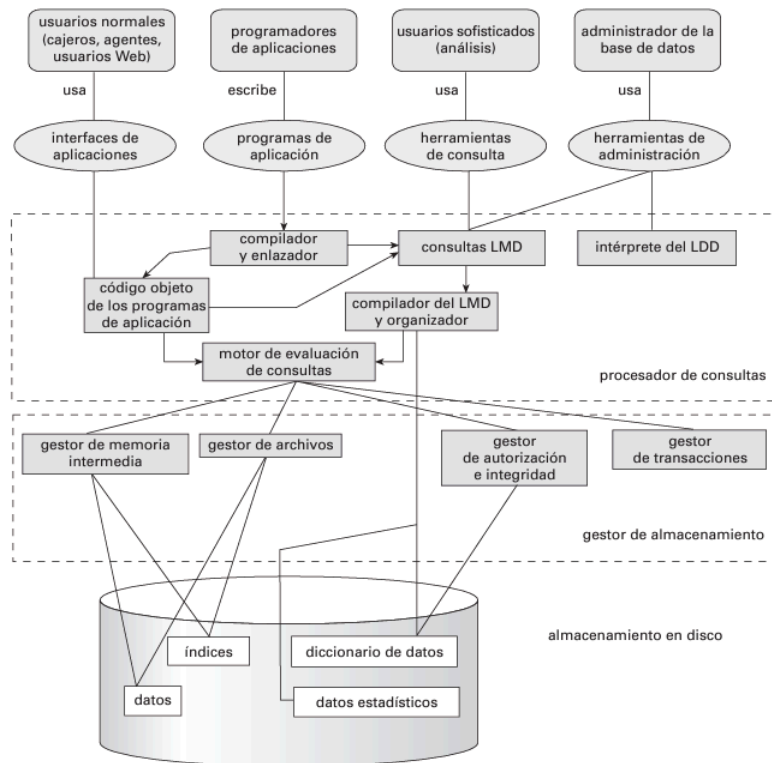
Hay cuatro tipos diferentes de usuarios de un sistema de base de datos, diferenciados por la forma en que ellos esperan interactuar con el sistema. Se han diseñado diferentes tipo de interfaces de usuario para diferentes tipos de usuarios

- **Usuarios normales:** Son usuarios no sofisticados que interactúan con el sistema mediante la invocación de alguno de los programas de aplicación permanentes que se ha escrito previamente. Por ejemplo, un cajero bancario que necesita transferir 1.000.000 \$ de la cuenta A a la cuenta B invoca un programa llamado transferir. Este programa pide al cajero el importe de dinero a transferir, la cuenta de la que el dinero va a ser transferido y la cuenta a la que el dinero va a ser transferido. Como otro ejemplo, considérese un usuario que desee encontrar su saldo de cuenta en la web. Tal usuario podría acceder a un formulario en el que introduce su número de cuenta. Un programa de aplicación en el servidor web recupera entonces el saldo de la cuenta, usando el número de cuenta proporcionado, y pasa la información al usuario. La interfaz de usuario normal para los usuarios normales es una interfaz de formularios, donde el usuario puede rellenar los campos apropiados del formulario. Los usuarios normales pueden también simplemente leer informes generados de la base de datos.
- **Programadores de aplicaciones.** Son profesionales informáticos que escriben programas de aplicación. Los programadores de aplicaciones pueden elegir entre muchas herramientas para desarrollar interfaces de usuario. Las herramientas de desarrollo rápido de aplicaciones son herramientas que permiten al programador de aplicaciones construir formularios e informes sin escribir un programa. Hay también tipos especiales de lenguajes de programación que combinan estructuras de control imperativo (por ejemplo, para bucles for, bucles while e instrucciones if-then-else) con instrucciones del lenguaje de manipulación de datos. Estos lenguajes, llamados a veces lenguajes de cuarta generación, a menudo incluyen características especiales para facilitar la generación de formularios y la presentación de datos en pantalla. La mayoría de los sistemas de bases de datos comerciales incluyen un lenguaje de cuarta generación.
- **Los usuarios sofisticados** interactúan con el sistema sin programas escritos. En su lugar, ellos forman sus consultas en un lenguaje de consulta de bases de datos. Cada una de estas consultas se envía al procesador de consultas, cuya función es transformar instrucciones de lenguaje de manipulación de datos - LM D(Data Manipulation Language DDL) a instrucciones que el gestor de almacenamiento entienda. Los analistas que envían las consultas para explorar los datos en la base de datos entran en esta categoría. Las herramientas de procesamiento analítico en línea (OLAP, Online Analytical Processing) simplifican la labor de los analistas permitiéndoles ver resúmenes de datos de formas diferentes. Por ejemplo, un analista puede ver las ventas totales por región (por ejemplo, norte, sur, este y oeste), o por producto, o por una combinación de la región y del producto (es decir, las ventas totales de cada producto en cada región). Las herramientas también permiten al analista seleccionar regiones específicas, examinar los datos con más detalle (por ejemplo, ventas por ciudad dentro de una región) o examinar los datos con menos detalle (por ejemplo, agrupando productos por categoría). Otra clase de herramientas para los analistas son las herramientas de recopilación de datos, que les ayudan a encontrar ciertas clases de patrones de datos.

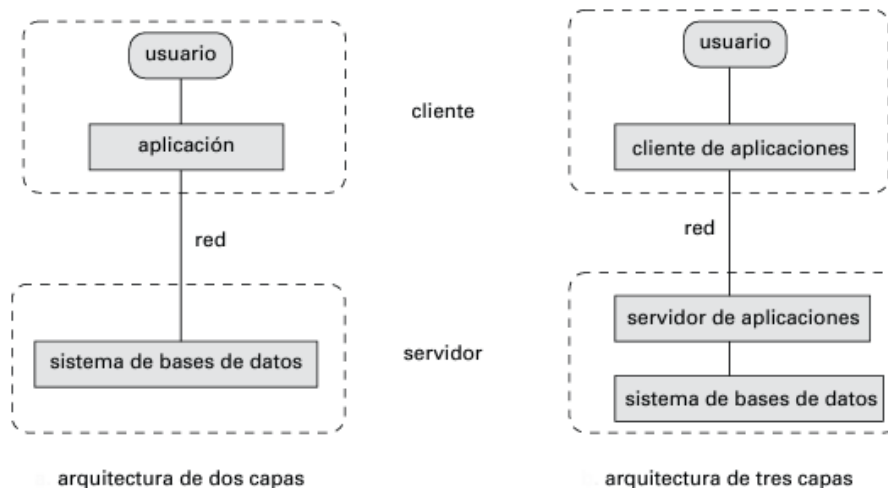
- **Usuarios especializados.** Son usuarios sofisticados que escriben aplicaciones de bases de datos especializadas que no son adecuadas en el marco de procesamiento de datos tradicional. Entre estas aplicaciones están los sistemas de diseño asistido por computador, sistemas de bases de conocimientos y sistemas expertos, sistemas que almacenan los datos con tipos de datos complejos (por ejemplo, datos gráficos y datos de audio) y sistemas de modelado del entorno.
- **Administrador de la base de datos** Una de las principales razones de usar SGBD es tener un control centralizado tanto de los datos como de los programas que acceden a esos datos. La persona que tiene este control central sobre el sistema se llama administrador de la base de datos (Database Administrator DBA). Las funciones del DBA incluyen las siguientes:
  - Definición del esquema. El DBA crea el esquema original de la base de datos escribiendo un conjunto de instrucciones de definición de datos en el lenguaje de definición de datos- LDD (Data Definition Language DDL)
  - Definición de la estructura y del método de acceso.
  - Modificación del esquema y de la organización física. Los ABD realizan cambios en el esquema y en la organización física para reflejar las necesidades cambiantes de la organización, o para alterar la organización física para mejorar el rendimiento.
  - Concesión de autorización para el acceso a los datos. La concesión de diferentes tipos de autorización permite al administrador de la base de datos determinar a qué partes de la base de datos puede acceder cada usuario. La información de autorización se mantiene en una estructura del sistema especial que el sistema de base de datos consulta cuando se intenta el acceso a los datos en el sistema.
  - Mantenimiento rutinario. Algunos ejemplos de actividades rutinarias de mantenimiento del administrado de la base de datos son: copia de seguridad periódica de la base de datos, asegurarse de que haya suficiente espacio libre en disco para las operaciones normales y aumentar el espacio en disco según sea necesario, supervisión de los trabajos que se ejecuten en la base de datos y asegurarse de que el rendimiento no se degrada por tareas muy costosas iniciadas por algunos usuarios.



## Bases de Datos - TUDAI



La mayoría de los usuarios de un sistema de bases de datos no están situados actualmente junto al sistema de bases de datos, sino que se conectan a él a través de una red. Se puede diferenciar entonces entre las máquinas cliente, en donde trabajan los usuarios remotos de la base de datos, y las máquinas servidor, en las que se ejecuta el sistema de bases de datos. Las aplicaciones de bases de datos se dividen usualmente en dos o tres partes.



En una **arquitectura de dos capas**, la aplicación se divide en un componente que reside en la máquina cliente, que llama a la funcionalidad del sistema de bases de datos en la máquina servidor mediante instrucciones del lenguaje de consultas. Los estándares de interfaces de programas de aplicación como ODBC y JDBC se usan para la interacción entre el cliente y el servidor. En cambio, en una **arquitectura de tres capas**, la máquina cliente actúa simplemente como frontal y no contiene ninguna llamada directa a la base de datos. En su lugar, el cliente se

comunica con un servidor de aplicaciones, usualmente mediante una interfaz de formularios. El servidor de aplicaciones, a su vez, se comunica con el sistema de bases de datos para acceder a los datos. La lógica de negocio de la aplicación, que establece las acciones a realizar bajo determinadas condiciones, se incorpora en el servidor de aplicaciones, en lugar de ser distribuida a múltiples clientes. Las aplicaciones de tres capas son más apropiadas para grandes aplicaciones, y para las aplicaciones que se ejecutan en la web.

## **Terminología básica**

Cuando utilizamos el lenguaje natural y decimos, por ejemplo, que una persona ha nacido en 1965, el **dato** (1965) va acompañado de su interpretación (año de nacimiento de una cierta persona) para convertir en **información** a dicho dato. De la misma manera cuando se habla de bases de datos se debe diferenciar entre el **esquema de la base de datos**, o diseño lógico de la misma, y el **ejemplar de la base de datos**, que es una instantánea de los datos de la misma en un momento dado.