

UT1 - Introducción al acceso a datos



Luis Javier López López

Índice

1

Concepto de Acceso a Datos

2

Clasificación de los sistemas de almacenamiento de información

3

Modelo de acceso a datos

4

Evolución de las tecnologías de persistencia

5

Arquitectura de una aplicación con acceso a datos

6

Lenguajes y tecnologías para el acceso a datos

7

Seguridad y acceso a datos

8

Tendencias actuales y futuro del acceso a datos



1. Concepto de acceso a datos

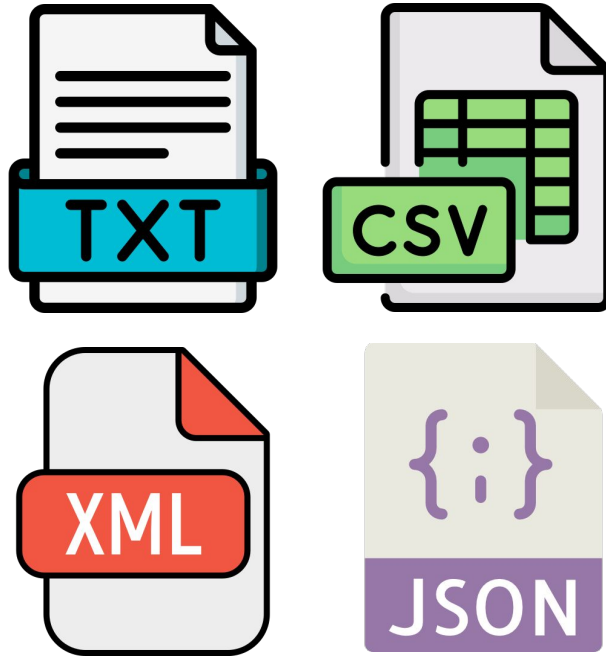


El acceso a datos es el conjunto de técnicas, métodos y herramientas que permiten a una aplicación **almacenar, recuperar, modificar y eliminar información**.

- **Datos:** hechos en bruto, sin contexto (ej: “23, Juan, Madrid”).
- **Información:** datos organizados y con significado (ej: “Juan, de 23 años, vive en Madrid”).
El objetivo del acceso a datos es garantizar que las aplicaciones puedan trabajar con información **de forma eficiente, segura y persistente**.



2. Clasificación de los Sistemas de almacenamiento de información



Ficheros planos

- Almacenan datos en texto o binario.
- Ventajas: sencillos, fáciles de usar.
- Inconvenientes: dificultad en búsquedas, redundancia de datos, falta de integridad.



2. Clasificación de los Sistemas de almacenamiento de información



Bases de datos relacionales (SQL)

- Almacenan datos en texto o binario.
- Ventajas: sencillos, fáciles de usar.
- Inconvenientes: dificultad en búsquedas, redundancia de datos, falta de integridad.





ACTIVIDAD

Recordamos la diferencia entre:

- *Base de datos*
- *Lenguaje de Base de Datos*
- *Sistema Gestor de Base de Datos*



2. Clasificación de los Sistemas de almacenamiento de información

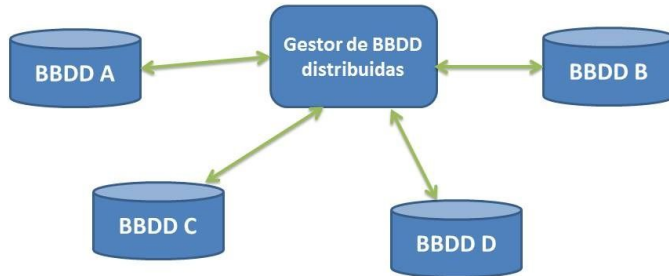
Bases de datos NO relacionales



- Diseñadas para grandes volúmenes y datos no estructurados.
- Tipos: documentales (MongoDB), clave-valor (Redis), columnares (Cassandra), grafos (Neo4j).
- Muy usadas en Big Data y entornos distribuidos.



2. Clasificación de los Sistemas de almacenamiento de información



Otros Sistemas

BBDD en memoria, distribuidas o en la nube.





ACTIVIDAD

Nos han encargado realizar la aplicación para diferentes proyectos de casos reales. Tenemos que averiguar o proponer qué tipo de bases de datos vamos a usar.





ACTIVIDAD

CASO 1 (Colegio San Marcos)

El colegio “San Marcos” lleva el registro de alumnos, asignaturas y calificaciones. Actualmente usan hojas de cálculo, pero quieren una aplicación que garantice que cada alumno solo tiene una matrícula, cada asignatura está bien definida y las notas no se duplican. Además, necesitan generar listados oficiales fácilmente.





ACTIVIDAD

CASO 2 (Tienda de ropa)

Una startup de moda quiere lanzar una web de e-commerce. Los productos tienen descripciones, fotos, tallas y precios. Además, quieren que los clientes puedan dejar comentarios y valoraciones, que pueden variar en número e incluir multimedia. Se espera que haya gran volumen de datos y crecimiento rápido.





ACTIVIDAD

CASO 3 (Caja Registradora)

Un bar pequeño necesita un sistema para guardar los tickets de caja del día y poder revisarlos si hay reclamaciones. El programa solo guarda la fecha, los productos vendidos y el importe total. No es necesario hacer búsquedas complejas, solo revisar algunos tickets concretos de vez en cuando.





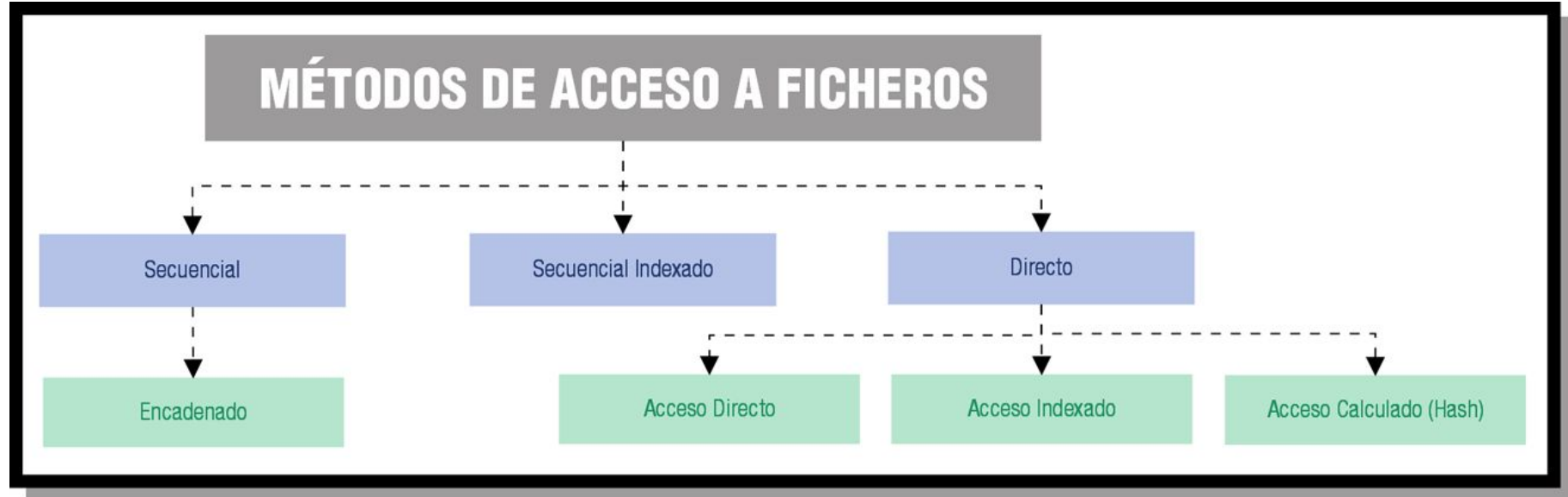
ACTIVIDAD

CASO 4 (Red social uni)

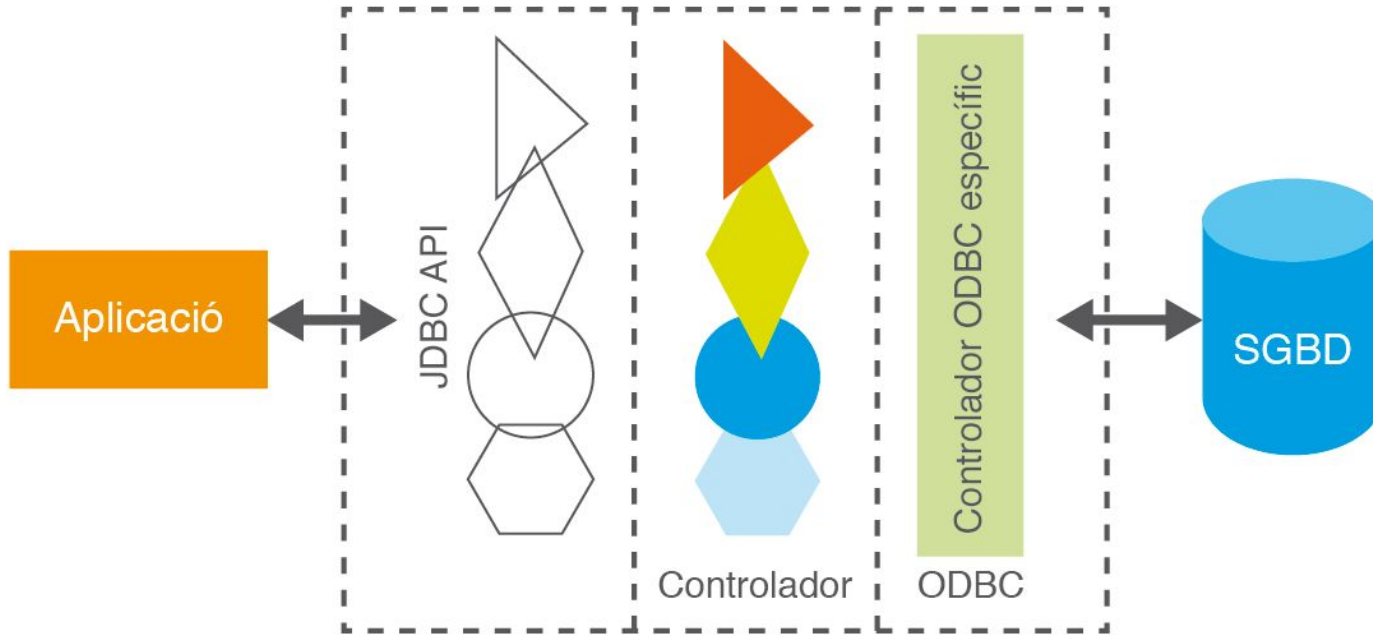
Una universidad quiere crear una red social para sus estudiantes. Los usuarios podrán tener amigos, enviar mensajes privados y pertenecer a grupos. La aplicación necesita gestionar muchas relaciones entre usuarios (amistades, mensajes, interacciones), que pueden crecer de forma exponencial.



3. Modelos de acceso a datos



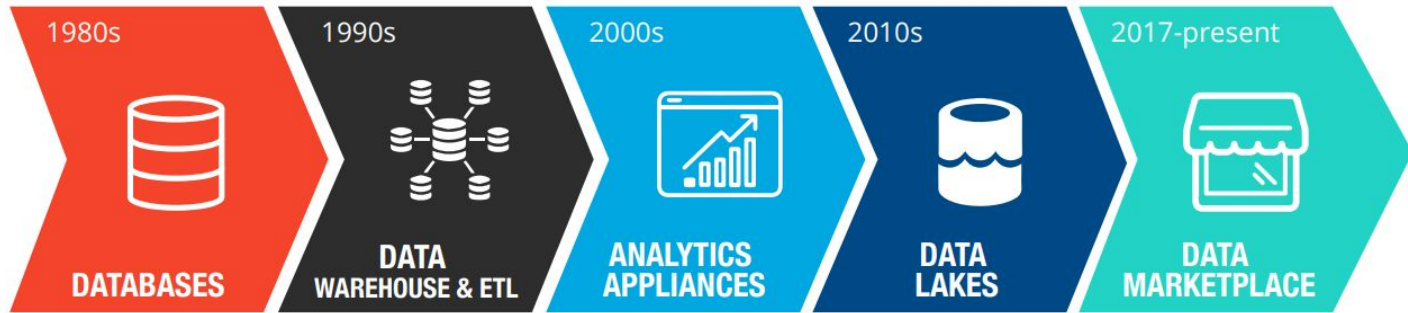
3. Modelos de acceso a datos



3. Modelos de acceso a datos



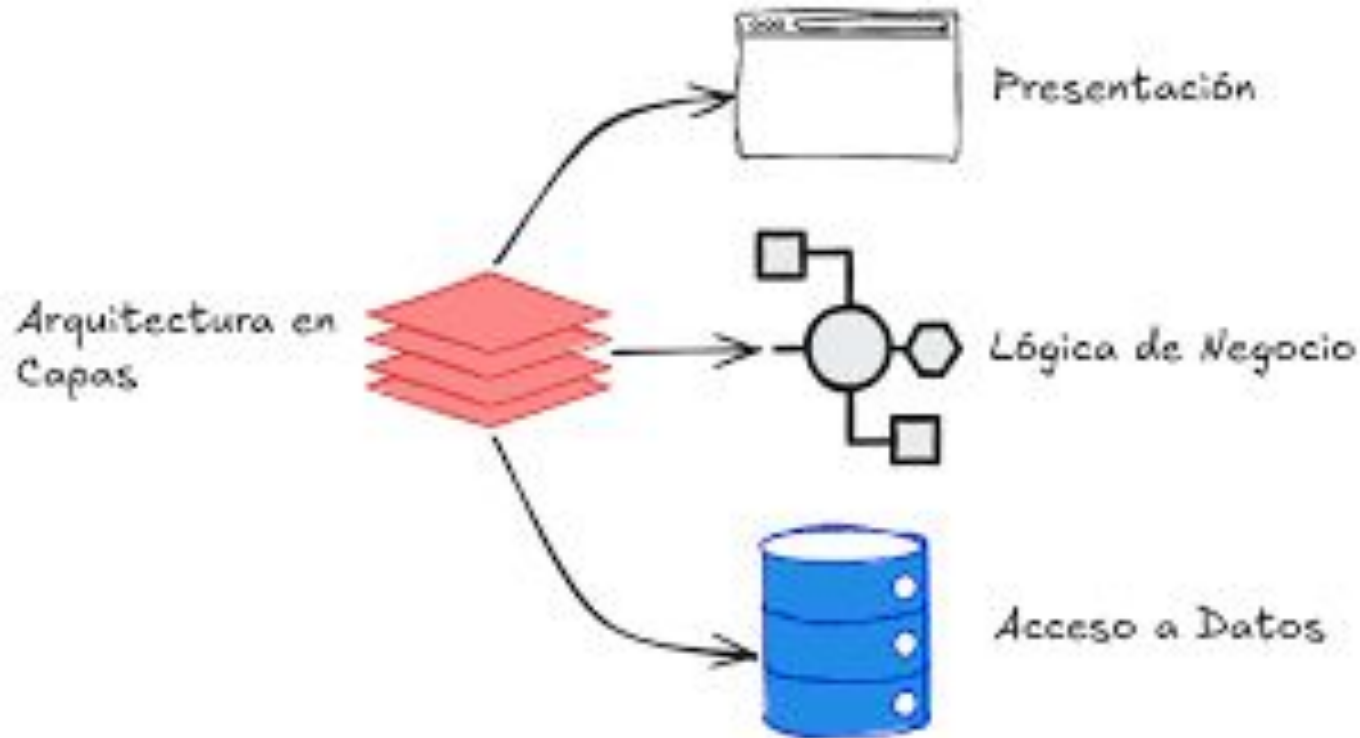
4. Evolución de las tecnologías de persistencia



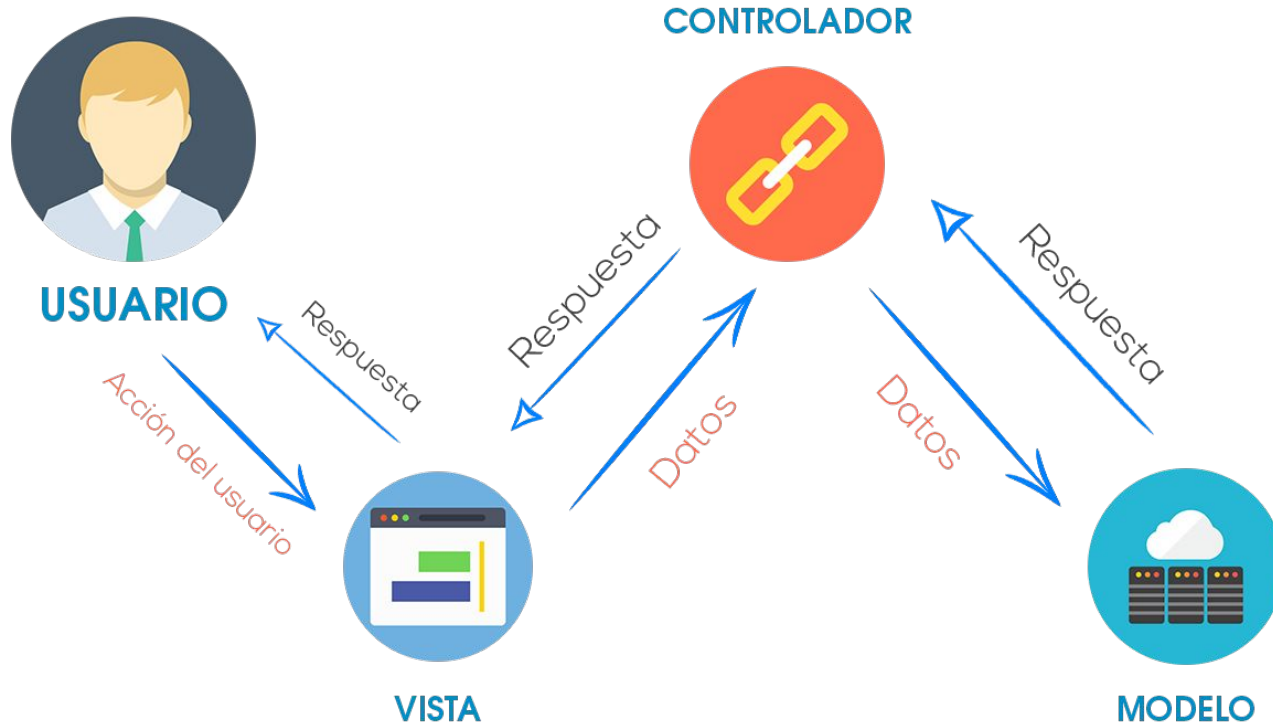
- **Primera etapa:** ficheros de texto o binarios → almacenamiento básico.
- **Bases de datos relacionales:** permiten integridad, consultas complejas y optimización.
- **Arquitectura cliente-servidor:** separación entre aplicación y gestor de datos.
- **Era actual:** datos distribuidos, servicios en la nube, bases de datos híbridas (mezcla de SQL y NoSQL).



5. Arquitectura de una aplicación con Acceso a datos



5. Arquitectura de una aplicación con Acceso a datos



5. Arquitectura de una aplicación con Acceso a datos



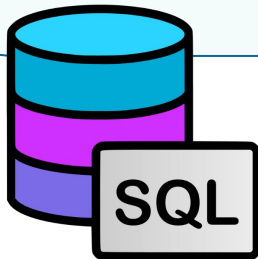
Patrón común: **DAO (Data Access Object)**

- Separa la lógica de negocio del código de acceso a datos.
- Hace más fácil mantener y cambiar la fuente de datos (fichero, SQL, NoSQL).



6. Lenguajes y tecnologías para el acceso a datos

- **SQL (Structured Query Language):** lenguaje estándar para consultar BBDD relacionales.
- **JDBC (Java Database Connectivity):** API en Java para conectarse a BBDD SQL.
- **ORM (Object-Relational Mapping):** herramientas como Hibernate o JPA que permiten trabajar con objetos en vez de consultas SQL.
- **APIs NoSQL:** librerías específicas de cada motor (ejemplo: MongoDB Driver, Redis client).



7. Seguridad y acceso a datos



- **Control de accesos:** usuarios, roles, permisos.
- **Riesgos:**
 - Inyección SQL (ejecutar código malicioso en consultas).
 - Exposición de datos sensibles.
 - Accesos no autorizados.
- **Buenas prácticas:**
 - Usar consultas preparadas.
 - Cifrado de datos sensibles.
 - Políticas de copias de seguridad.



8. Tendencias actuales y futuro del acceso a datos



- **Big Data:** tratamiento de grandes volúmenes de datos en tiempo real.
- **Bases de datos en la nube:** escalables, con mantenimiento gestionado por proveedores (ej: AWS RDS, Azure SQL).
- **Microservicios:** cada servicio puede tener su propia base de datos, optimizada a sus necesidades.
- **Bases de datos multimodelo:** combinan tablas, documentos, grafos en un mismo motor.





ACTIVIDAD FINAL



moodle

