

Ejercicio 1: Selección de Arquitectura de Datos (Casos de Uso)

Enunciado: Imagina que trabajas como arquitecto de datos y debes recomendar la tipología de base de datos más adecuada para tres proyectos diferentes de una compañía tecnológica, justificando tu elección con los conceptos de la fuente:

1. **Proyecto A:** Un sistema de monitorización para una red de sensores meteorológicos inteligentes de la ciudad que emite flujos constantes de lecturas analíticas en tiempo real.
2. **Proyecto B:** Un portal de comercio electrónico masivo que necesita gestionar de forma extremadamente rápida la información temporal de los carritos de compra de millones de usuarios simultáneos.
3. **Proyecto C:** Una plataforma para un cuerpo policial que requiere analizar patrones complejos de relaciones, conexiones y asociaciones de llamadas entre sospechosos heterogéneos para prevenir el fraude.

Solución propuesta:

1. **Bases de datos de tiempo:** Son las indicadas para el *Proyecto A* porque están **optimizadas para datos con marcas de tiempo** (*timestamp*) y tienen una **gran utilidad en el ecosistema del IoT** (*Internet of Things*), donde dispositivos y sensores conectados emiten flujos continuos de datos fechados.
2. **Bases de datos clave-valor:** Son la opción ideal para el *Proyecto B* porque guardan los datos de forma muy simple como un grupo de **par-valor**, poseen una alta escalabilidad ante volúmenes masivos de tráfico y resultan **idóneas para administrar carritos de compra** y sesiones de usuario.
3. **Bases de datos de grafos:** Son las adecuadas para el *Proyecto C* porque se fundamentan en la teoría de grafos y su software (GODBMS) está diseñado para trabajar con las **conexiones entre puntos de datos**, lo que las hace excelentes para **analizar relaciones entre puntos heterogéneos** de información y prevenir el fraude.

Ejercicio 2: El Dilema del Esquema (ACID vs. Flexibilidad NoSQL)

Enunciado: Un banco comercial requiere almacenar los registros de transferencias financieras de sus clientes, mientras que una startup multimedia similar a Instagram necesita almacenar el contenido variado (fotos, comentarios, me gusta) de millones de usuarios. Responde según la fuente:

1. ¿Qué modelo de base de datos (Relacional o NoSQL) debería adoptar cada una de estas organizaciones?
2. ¿Qué conjunto de propiedades estándar asegura que las transacciones del banco sean completamente fiables y qué significan sus siglas?
3. ¿Por qué para portales masivos como Facebook o YouTube resultó impracticable continuar utilizando bases de datos relacionales tradicionales para almacenar las publicaciones de sus usuarios?

Solución propuesta:

1. El banco debe adoptar un **modelo relacional**, mientras que la startup de fotos debe implementar un **modelo NoSQL (o no relacional)**.
2. La fiabilidad de las transacciones del banco se garantiza mediante las propiedades **ACID**. Este acrónimo significa **Atomicity** (Atomicidad), **Consistency** (Consistencia), **Isolation** (Aislamiento) y **Durability** (Durabilidad).
3. Porque el modelo NoSQL está pensado para manejar **datos no estructurados o semiestructurados** sin requerir un esquema rígido predefinido. Con el surgimiento de plataformas como Facebook o YouTube, la variedad y el volumen de archivos multimedia (fotos, vídeos) subidos por los usuarios hicieron que **mantenerlos de manera eficiente en una base de datos relacional fuera imposible**.

Ejercicio 3: Reconstrucción de la Línea de Tiempo Histórica

Enunciado: Ordena cronológicamente (del hito más antiguo al más reciente) los siguientes avances históricos del tratamiento de la información y describe brevemente el contexto, autor o tecnologías asociadas a cada uno basándote en la lectura:

- A) Primera referencia documentada en la historia sobre el concepto formal de una "base de datos".
- B) Desarrollo del lenguaje SQL por parte de IBM.
- C) Invención de la máquina automática de tarjetas perforadas.
- D) Definición del modelo relacional por medio de reglas para grandes bancos de datos.
- E) Primera generación de bases de datos comerciales en red y jerárquicas (CODASYL e IMS).

Solución propuesta:

1. **C) Máquina automática de tarjetas perforadas (1884):** Inventada por **Herman Hollerith** (nombrado el primer ingeniero estadístico de la historia), lo que representó el primer salto tecnológico desde el registro de datos original en papel.
2. **E) Primera generación en red y jerárquicas (Década de 1960):** Surgieron las bases de datos de red (**CODASYL**) y las jerárquicas (**IMS** de IBM), basadas en listas y árboles. CODASYL era un sistema altamente complejo que requería navegación manual para recorrer los datos conectados.
3. **A) Primera referencia del concepto "base de datos" (1963):** Se escuchó formalmente por primera vez durante un **simposio celebrado en California**.
4. **D) Definición del modelo relacional (Década de 1970):** Formulado por **Edgar Frank Codd** en su artículo sobre grandes bancos de datos compartidos, motivado por la falta de un motor que aprovechara sistemas como IMS sin requerir personal

especializado. Este modelo sustituyó las listas libres por **tablas con registros de longitud fija**.

5. **B) Desarrollo del lenguaje SQL (Década de 1980):** Creado por IBM para plantar cara a **QUEL** (el lenguaje desarrollado bajo el proyecto de investigación INGRES en la UC Berkeley por Michael Stonebraker y Eugene Wong) y estandarizar las consultas relacionales.

Ejercicio 4: Modelado de Datos (Estructuras Jerárquicas vs. Objetos)

Enunciado: Analiza los siguientes requerimientos técnicos y determina qué modelo de almacenamiento de la fuente se adapta a cada uno:

1. **Requerimiento 1:** Una aplicación administrativa que debe almacenar datos organizados estrictamente con un esquema de "árbol genealógico", donde un elemento principal puede tener subramas dependientes. ¿Qué modelo es, qué relación define su esquema y qué ejemplo nativo y cotidiano en los sistemas operativos modernos lo utiliza?
2. **Requerimiento 2:** Una plataforma escrita en lenguajes como Java o C++ que necesita persistir datos directamente como entidades complejas (unificando la información y sus comportamientos) sin dividirlos en tablas y filas. ¿Qué tipo de base de datos es, qué siglas identifican a sus sistemas de administración y qué ejemplos de software cita la fuente?

Solución propuesta:

1. **Bases de datos jerárquicas:** Almacenan los datos utilizando un **modelo padre-hijo** estructurado como ramas. Un ejemplo muy común de este modelo que se ejecuta de forma nativa en entornos de escritorio es el **registro de Windows**.
2. **Bases de datos orientadas a objetos:** Se basan en el paradigma de la programación orientada a objetos (POO), permitiendo que **los datos y sus atributos estén unidos entre sí como un objeto**. Son gestionadas por sistemas **OODBMS** y funcionan de forma idónea con Java o C++. Los ejemplos citados en la fuente son **Wakanda** y **ObjectStore**.

Ejercicio 5: Estructura de Datos (Tablas Relacionales vs. Documentos JSON)

Enunciado: En los ciclos de informática de ASIR, DAW y DAM es fundamental entender cómo se estructuran físicamente los datos según el modelo elegido. Imagina que vas a diseñar un sistema para almacenar la información de los alumnos de un instituto (nombre, apellidos, ciclo formativo y asignaturas matriculadas):

1. **Escenario A:** Si decides utilizar una **base de datos relacional clásica**, ¿cómo se organizaría estructuralmente la información de los alumnos y qué lenguaje estándar emplearías para realizar consultas? Nombra dos motores comerciales aptos para este escenario.

2. **Escenario B:** Si optas por una **base de datos NoSQL orientada a documentos**, ¿en qué formato o tipo de archivo se modelarían los datos de los alumnos en lugar de usar filas y columnas? Nombra dos sistemas de bases de datos de este tipo que se mencionen en el texto.

Solución propuesta:

1. **Escenario A (Relacional):** La información se organizaría repartida en **varias tablas relacionadas entre sí**, donde los datos se distribuyen ordenadamente en **filas y columnas**. El lenguaje estándar empleado para leer, crear, actualizar y eliminar estos datos es el **SQL** (*Structured Query Language*). Como motores relacionales comerciales aptos se pueden utilizar, por ejemplo, **MySQL/MariaDB**, **PostgreSQL**, **Oracle Database** o **Microsoft SQL Server**.
2. **Escenario B (NoSQL Documental):** En lugar de filas y columnas, la información de los alumnos se modelaría en **documentos similares a ficheros JSON**. Como motores de bases de datos orientadas a documentos se pueden utilizar **MongoDB**, **Apache CouchDB** o **Amazon DocumentDB**.