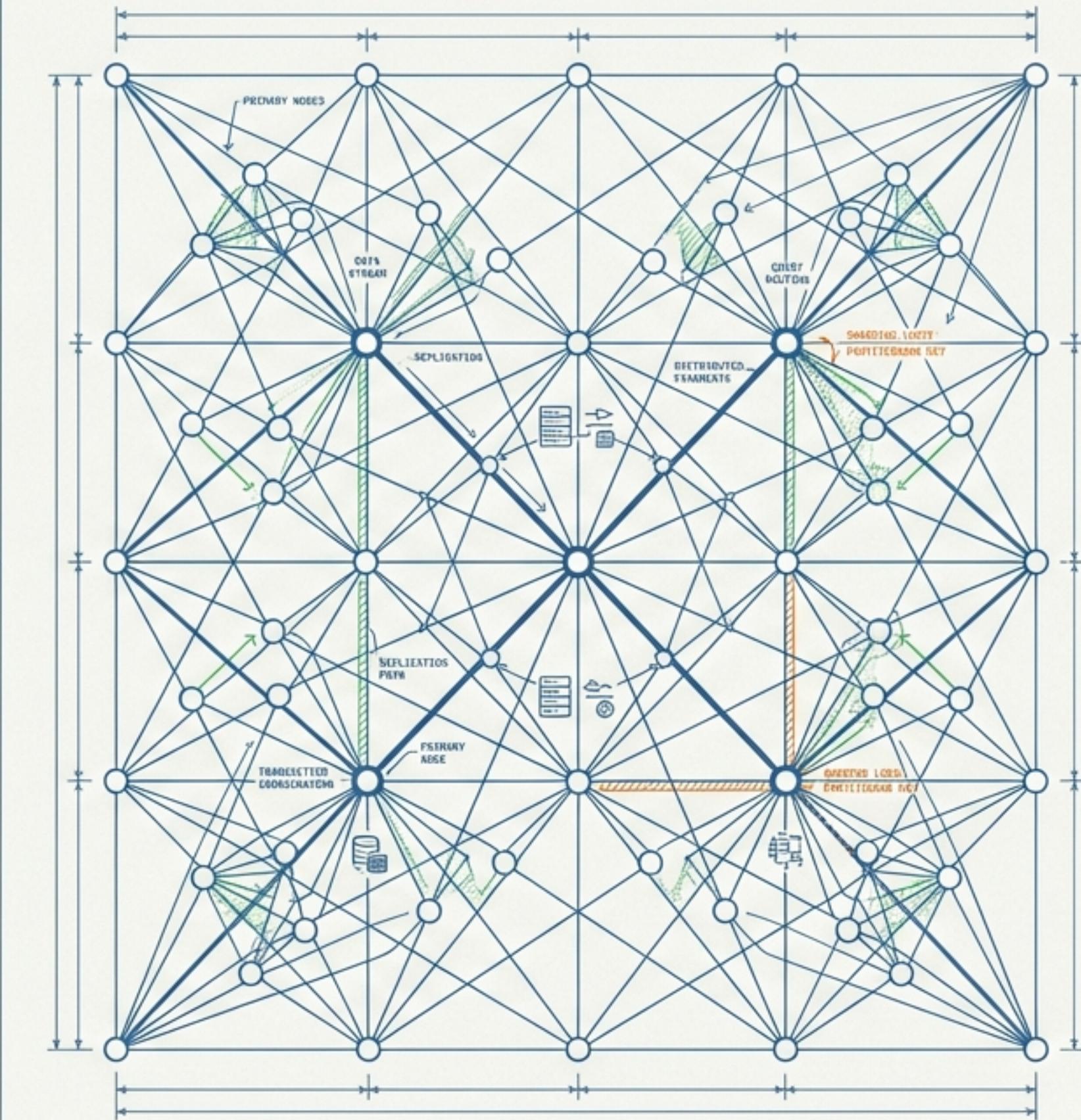


Fundamentos de las Bases de Datos

De la Teoría a la Arquitectura Distribuida

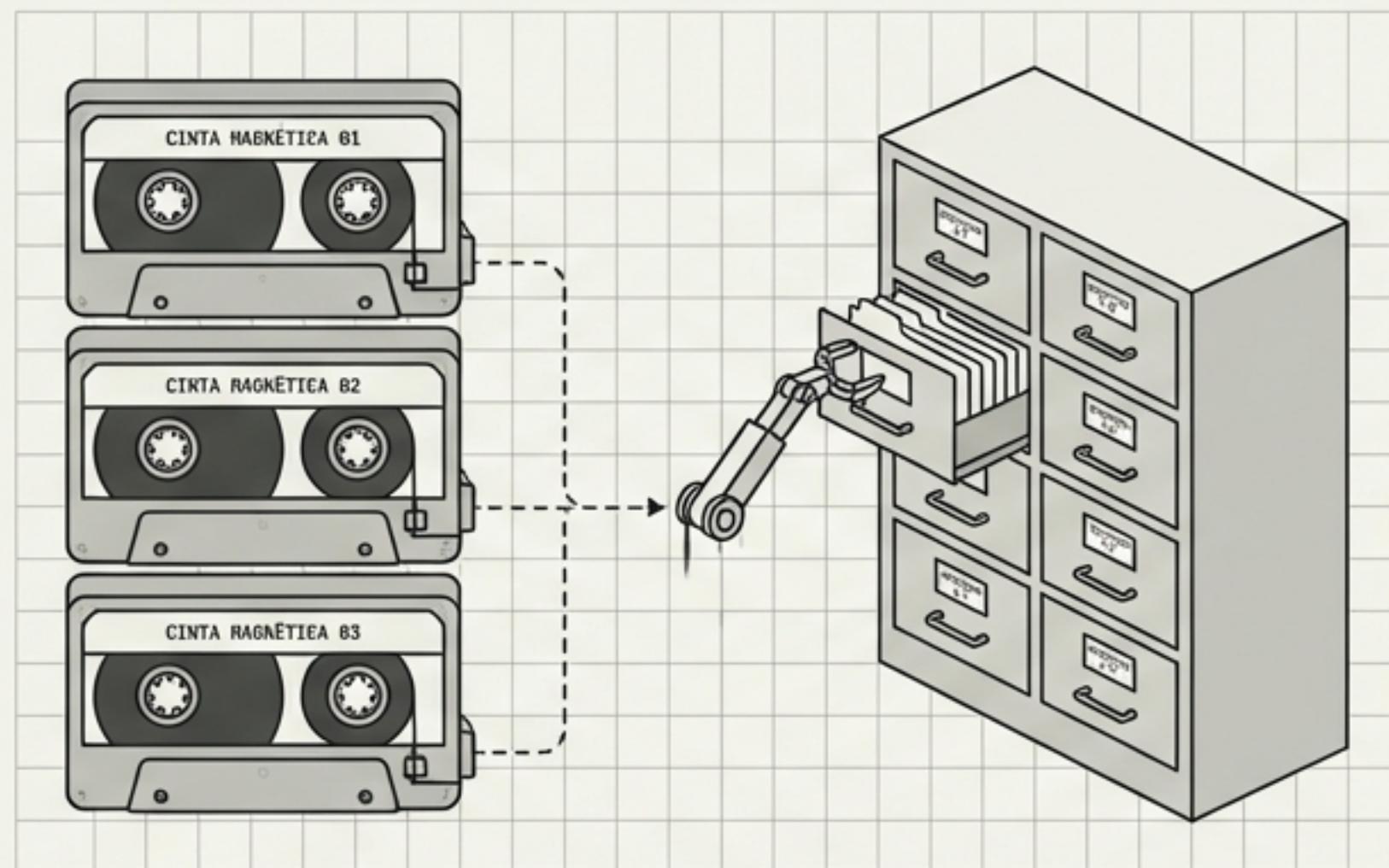
Una guía técnica sobre modelos, fragmentación y diseño de sistemas escalables.

Ref: Tema 2: Fundamentos de las bases de datos



La Evolución: De la Gestión de Archivos a la Integridad de Datos

El Pasado: Sistemas de Archivos (Pre-1960s)

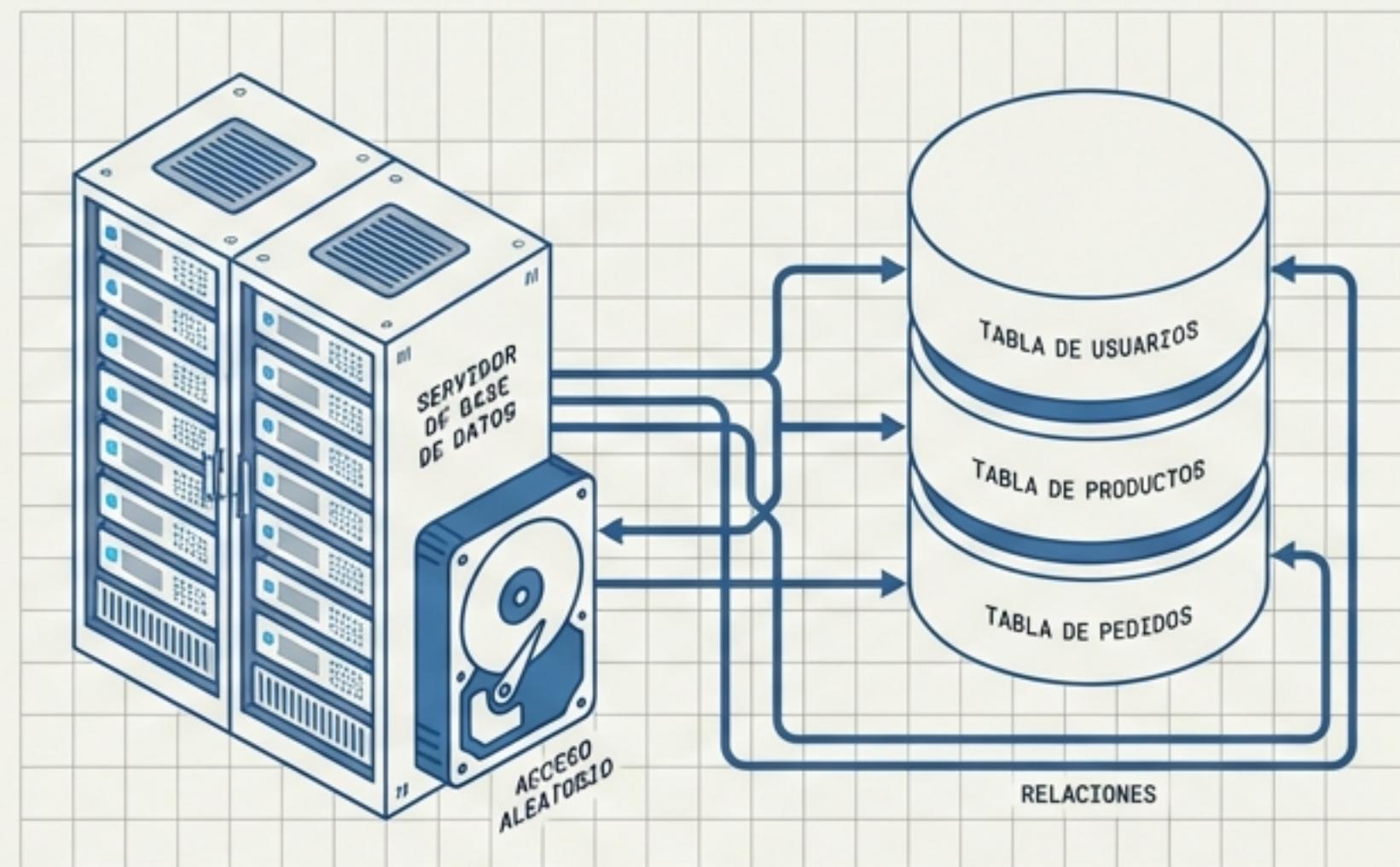


- Procesamiento secuencial
- Duplicidad de datos
- Inconsistencia
- Rigidez estructural



Las bases de datos surgieron para solucionar la falta de coherencia y la redundancia de los antiguos sistemas de gestión de archivos.

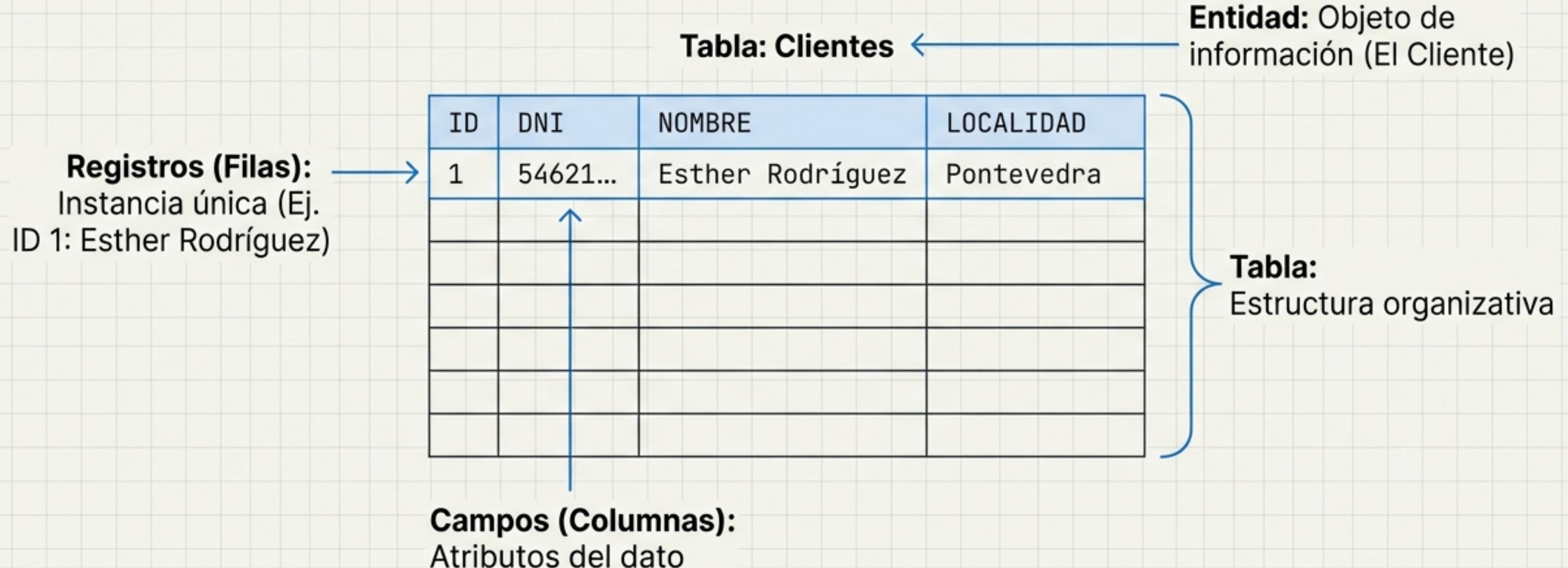
El Presente: Bases de Datos



- Acceso aleatorio (Discos Magnéticos)
- Integridad referencial
- Archivos interrelacionados
- Coherencia y no redundancia

Anatomía de la Base de Datos: Los Bloques de Construcción

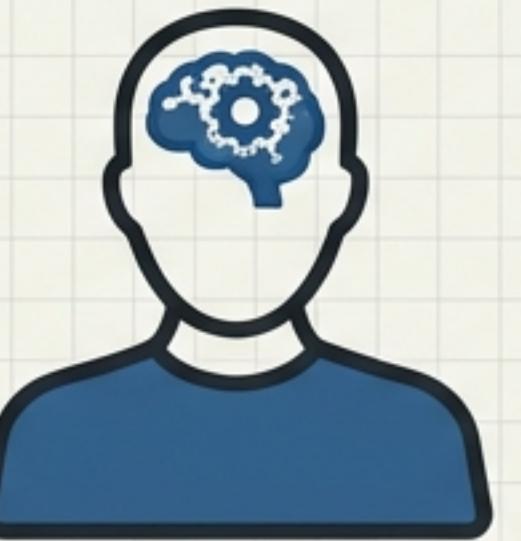
El modelo define la arquitectura: almacenamiento e interrelación.



Clasificación Funcional: Tipología según el Uso

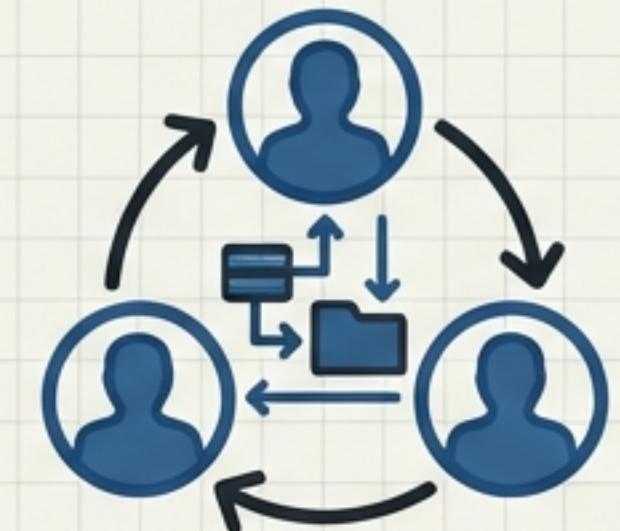
Individual

Un solo usuario asume todos los roles (**administrador**, **programador**). Uso local.



Compartida

Múltiples usuarios con acceso concurrente. Típico en Pymes.



Bancos de Datos

Grandes volúmenes de información específica (Pública o Privada).
Ej: **BASE**.



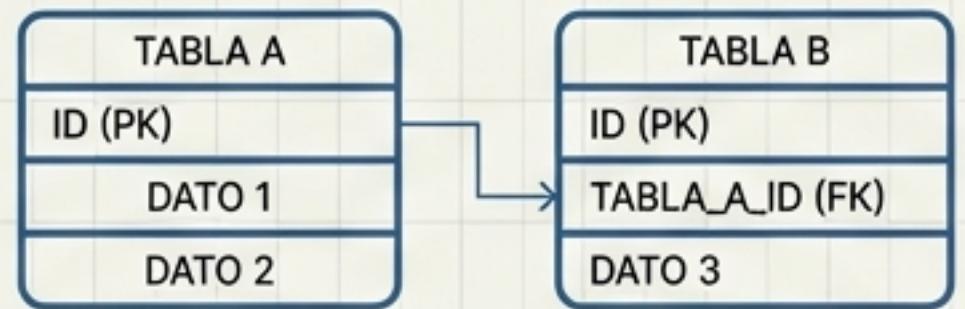
Pública

Acceso abierto a cualquier usuario con medios técnicos.
Ej: **Catastro**.



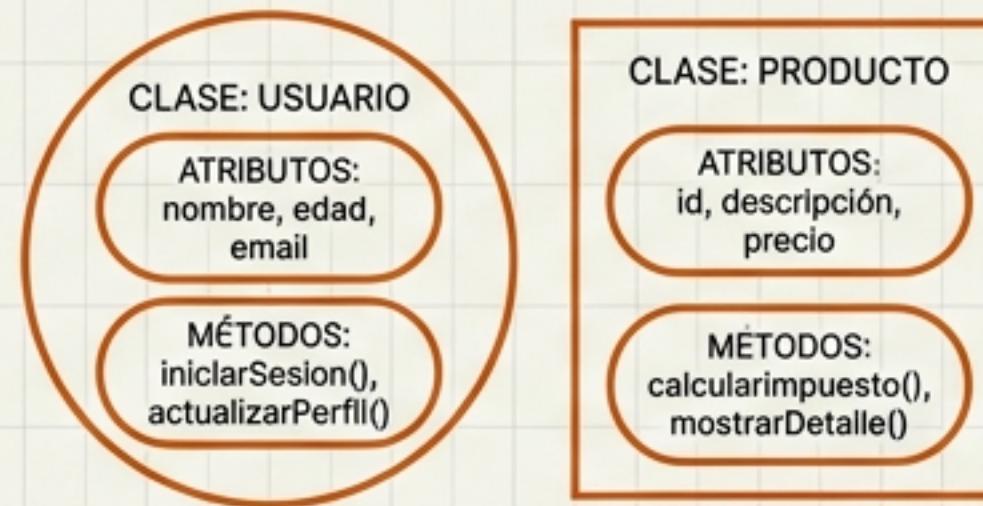
Modelos Estandarizados: Relacional y Orientado a Objetos

BBDD Relacionales (El Estándar)



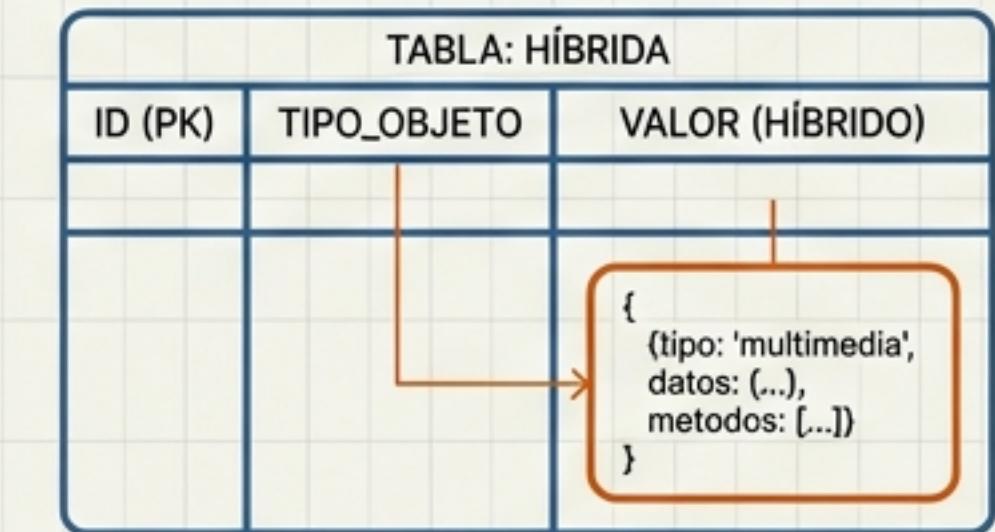
Datos en tablas (filas/columnas).
Integridad referencial garantizada.
Lenguaje SQL.
Ideal para datos estructurados y robustez.

BBDD Orientadas a Objetos



Información como objetos definidos por clases. Soporta multimedia (vídeo, audio).
Flexibilidad para estructuras complejas.

Objeto-Relacionales (Híbrido)



Eficiencia relacional + tipos de datos avanzados.
Adoptado por Oracle, IBM, Microsoft.

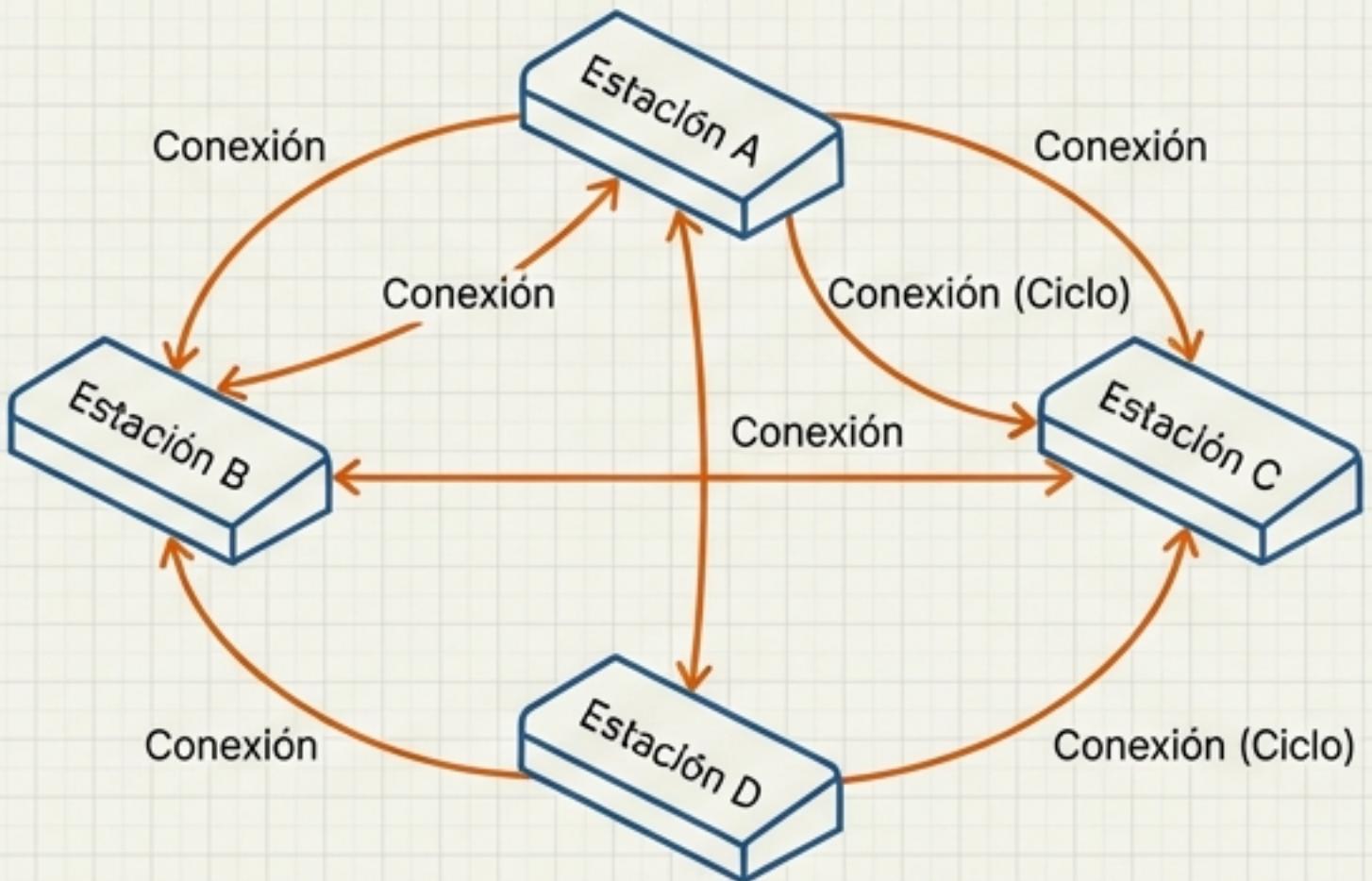
Modelos Especializados y Estructuras Jerárquicas

Modelo Jerárquico



Relación Padre-Hijo. Un hijo solo tiene un padre.
Estructura rígida.

Modelo en Red

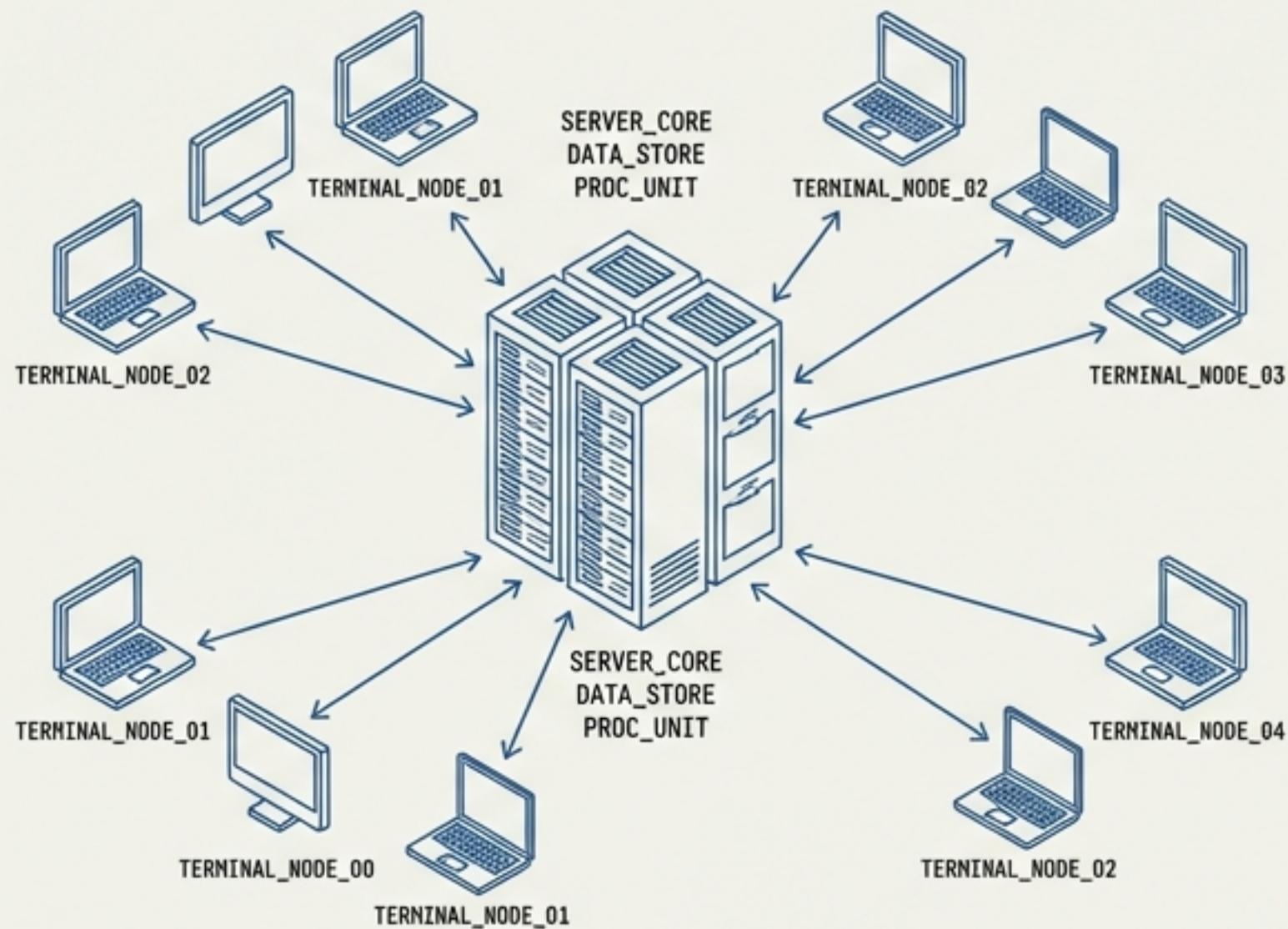


Evolución Flexible. Relaciones circulares permitidas. Un nodo puede tener varios padres.

Otros paradigmas: **NoSQL** (Datos no estructurados) y **Deductivas** (Basadas en lógica).

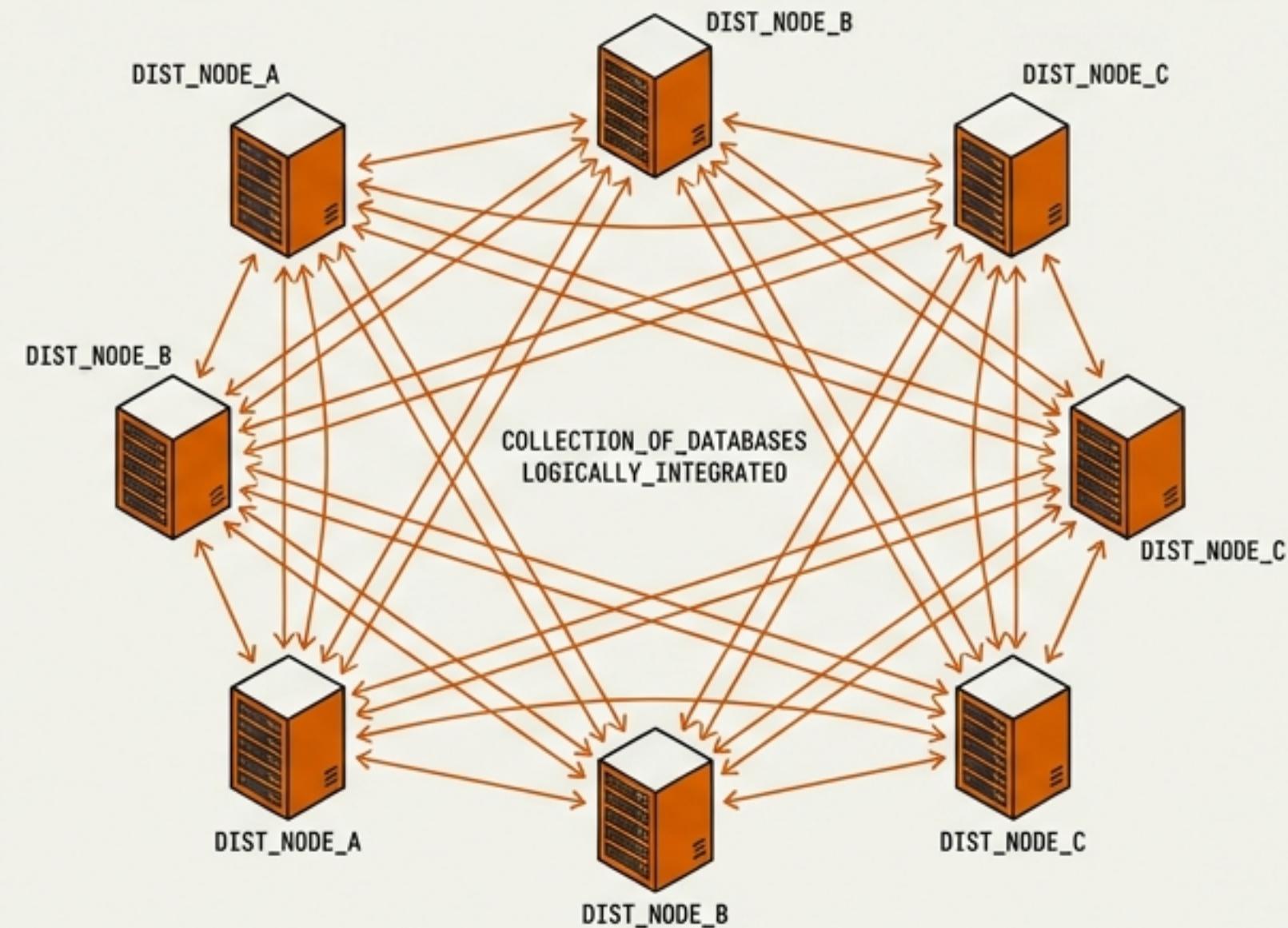
El Salto de Escala: Arquitectura Centralizada vs. Distribuida

Arquitectura Centralizada



Ubicación única. Gestión sencilla.
Riesgo: Punto único de fallo.

Arquitectura Distribuida



Múltiples nodos interconectados. Colección de BBDD
lógicamente integradas, físicamente distribuidas.
Drivers: Amazon, Google, IA.

Arquitectura de Referencia: Esquemas y Capas

Esquema Global

Visión unificada. Define los datos como si no estuvieran distribuidos.



Esquema de Fragmentación

Lógica de división. Cómo se cortan los datos (Horizontal/Vertical).



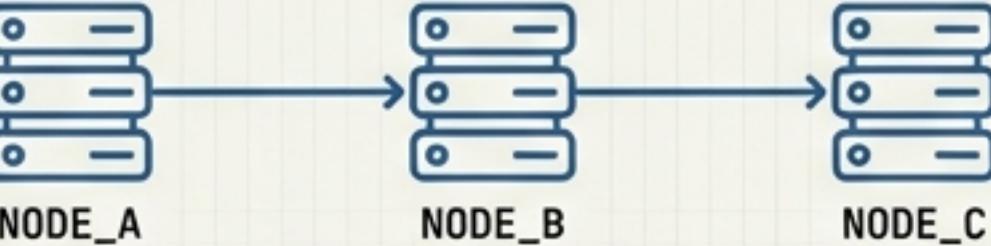
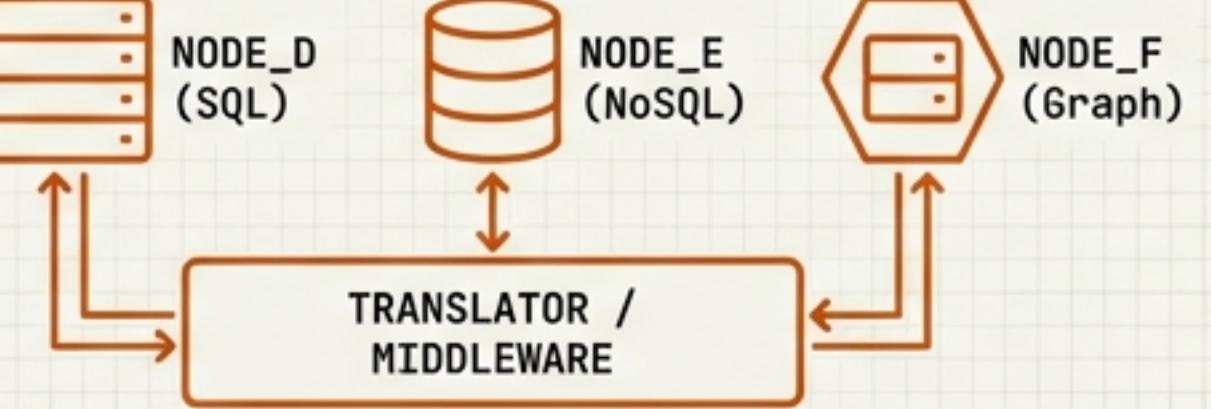
Esquema de Asignación

Mapa físico. Determina en qué nodo reside cada fragmento.

Enfoques de Definición

1. Basado en componentes.
2. Basado en funciones (tipos de usuarios).
3. Basado en datos.

Tipologías de Implementación: Homogénea vs. Heterogénea

A	Definición	B	Visual	C	Características
Homogéneos	Mismo DBMS en todos los nodos.	Visual			Cooperación fluida. Sin usuarios globales obligatorios.
Heterogéneos	Diferentes DBMS entre nodos.				Requiere software intermedio. Interoperabilidad compleja.

Implementaciones Federadas

Autonomía Local

Independencia total del nodo.

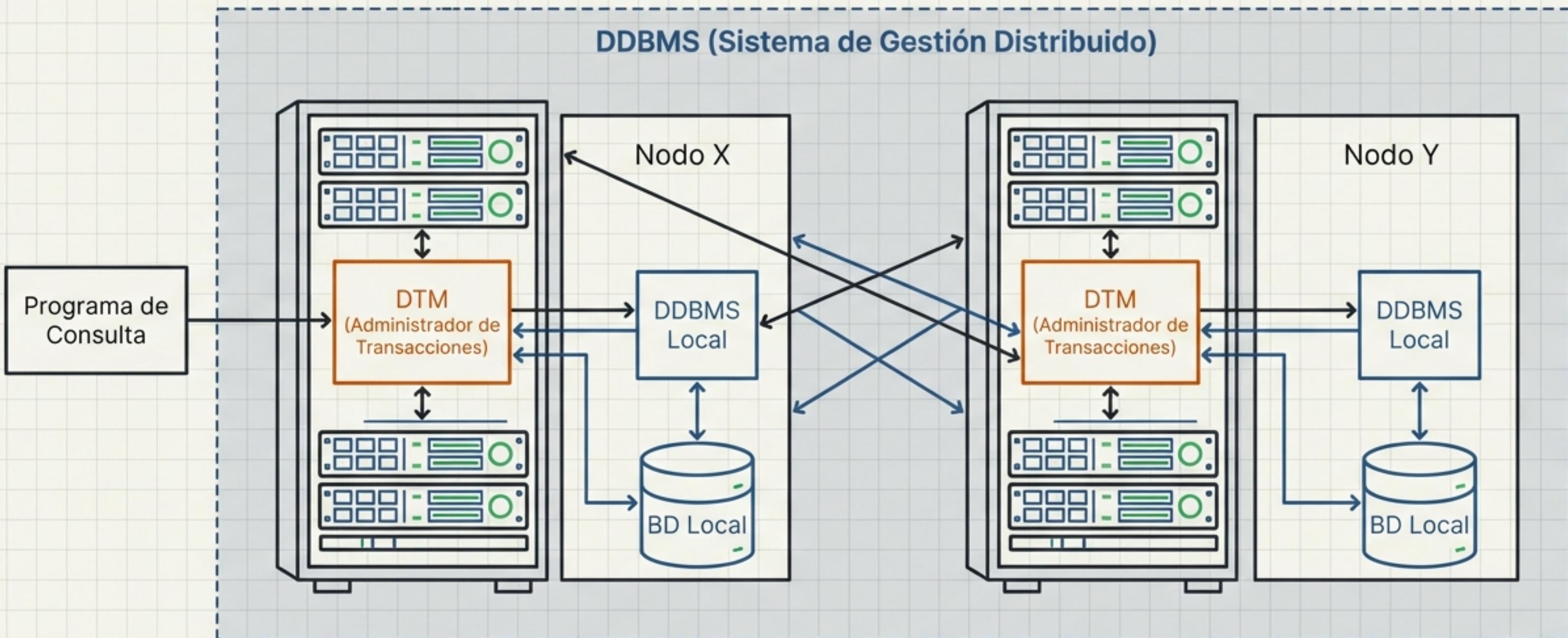
Interoperabilidad

Comunicación entre sistemas diversos.

Consulta Global

Vista unificada de datos dispersos.

Componentes del Sistema: Hardware y Software



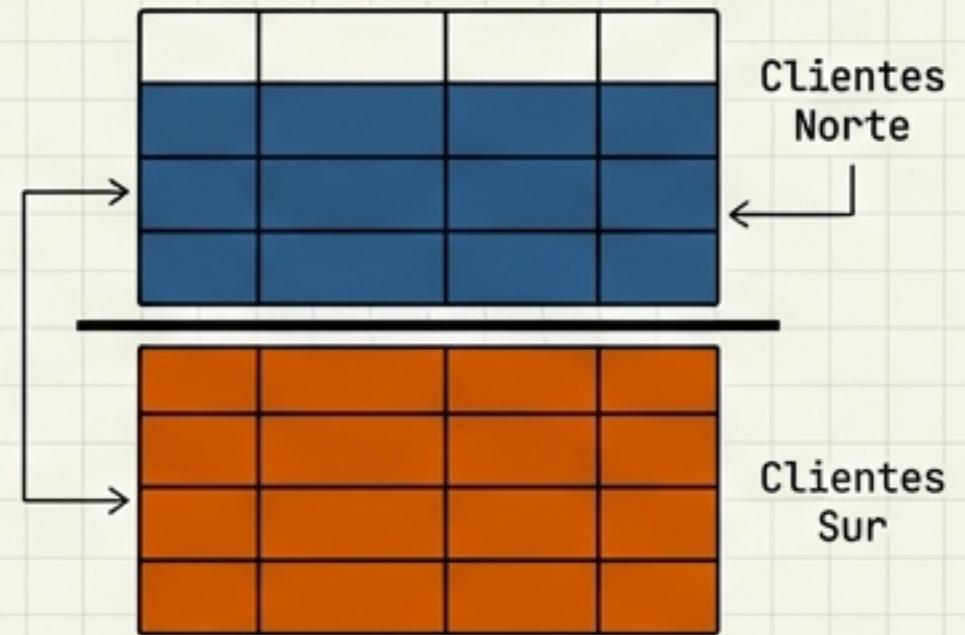
DDBMS: El cerebro gestor.

DTM: Gestiona la integridad y consultas.

Nodo: Servidor físico.

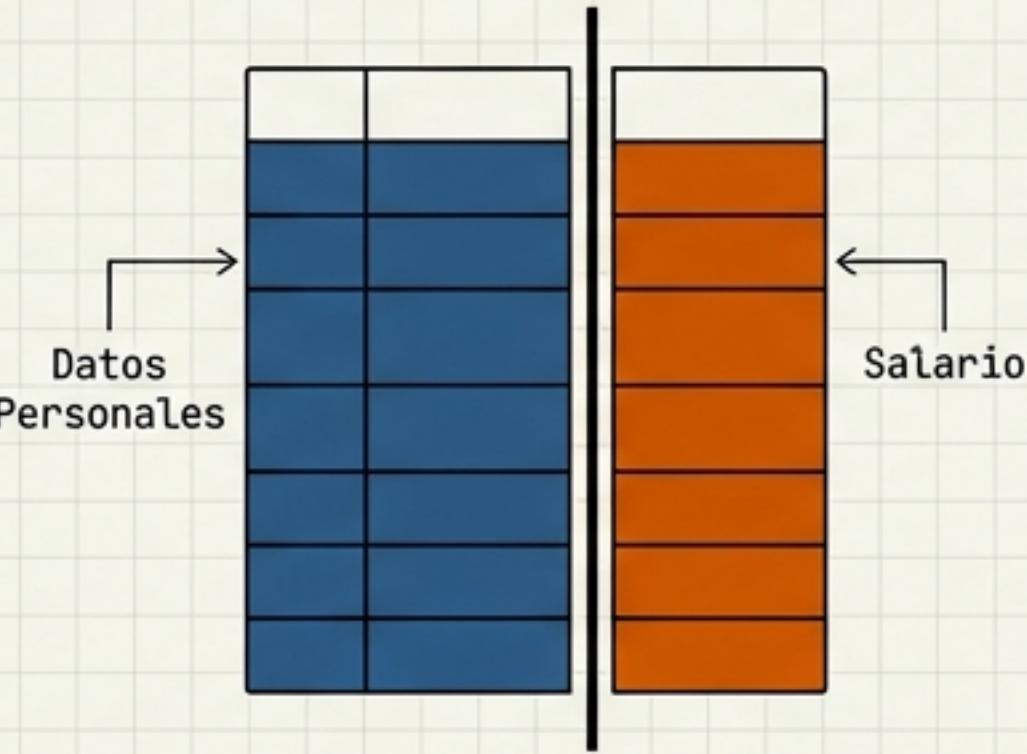
Estrategias de Fragmentación de Información

Fragmentación Horizontal (Sharding)



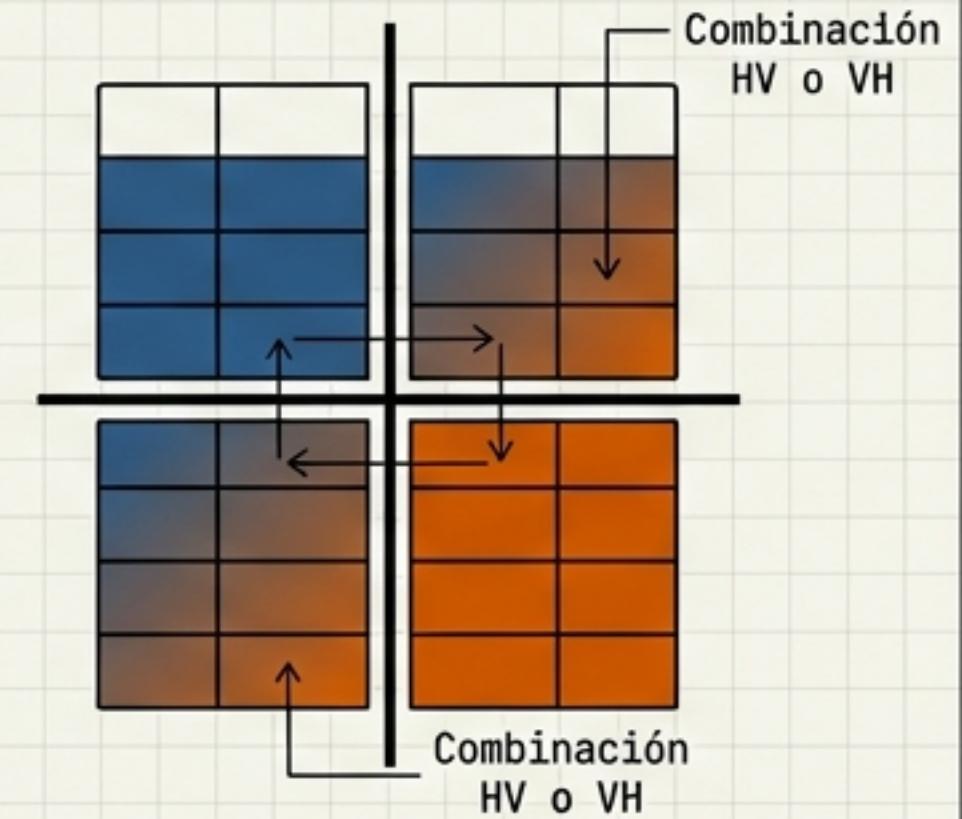
División por registros (Filas).
Ej: Cuentas Norte vs. Sur.

Fragmentación Vertical



División por atributos (Columnas).
Ej: Datos Personales vs. Salario.

Híbrida / Mixta



Combinación HV o VH para optimización precisa.

Regla de Oro: El grado de fragmentación debe ser intermedio. Ni excesivo (ingestionable) ni escaso (ineficiente).

Distribución y Replicación de Datos

Configuración de Nodos



Réplica

Alta disponibilidad
(Copia total).



Particionado

Sin duplicación
(Ahorro).

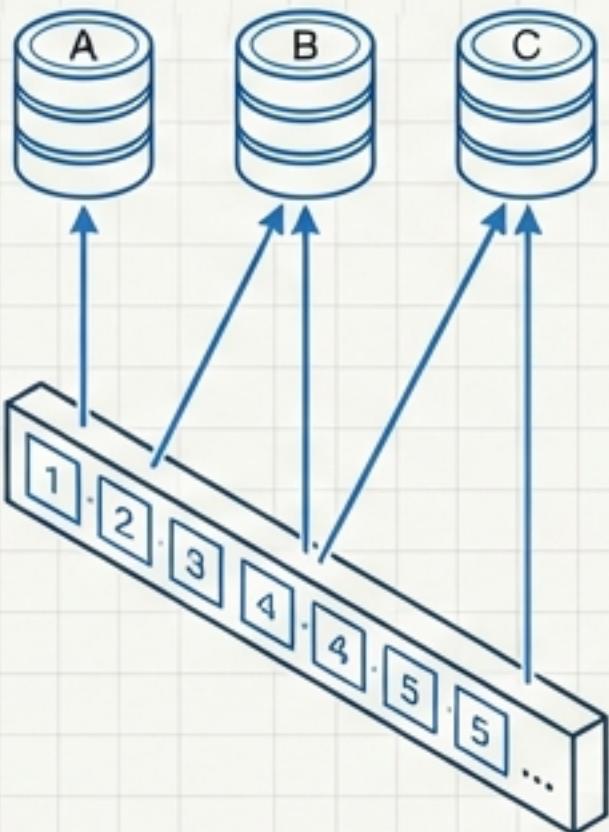


Híbrido

Estándar industrial.

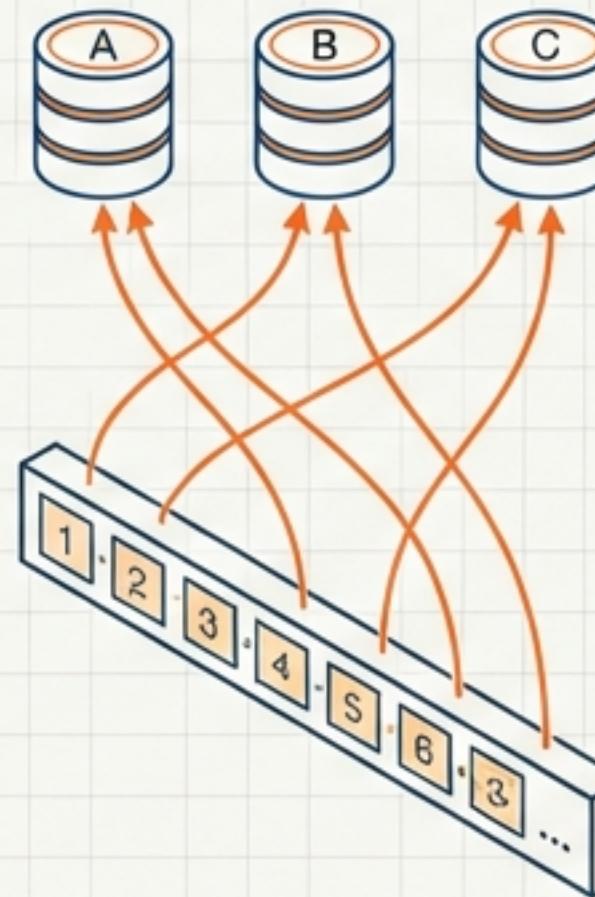
Algoritmos de Distribución

Round-Robin



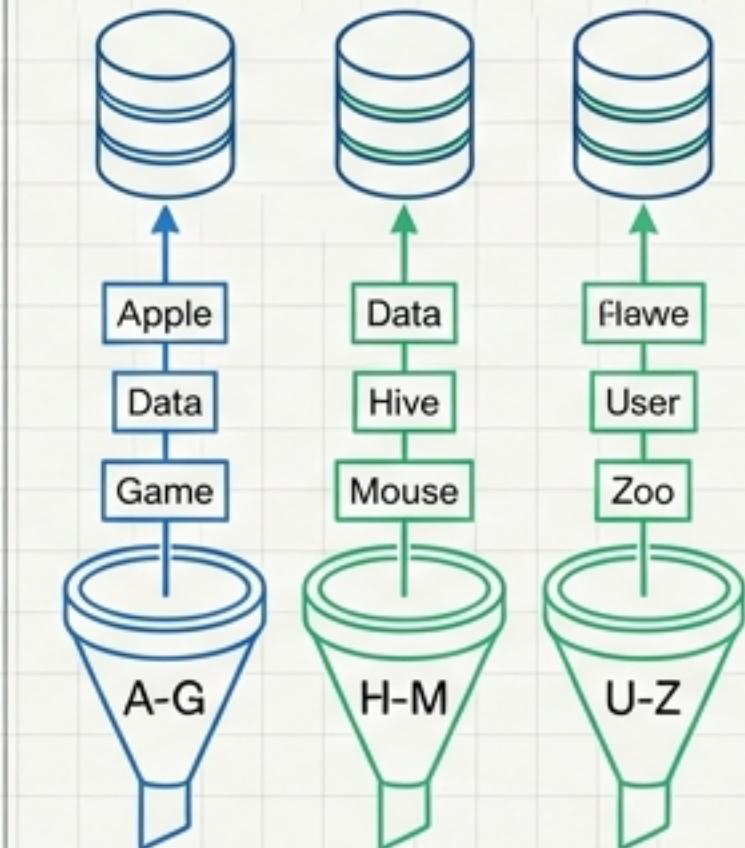
Distribución secuencial
(1->A, 2->B, 3->C).

Hashing



Distribución aleatoria calculada
(Función Hash).

Range



Distribución por rangos de clave.

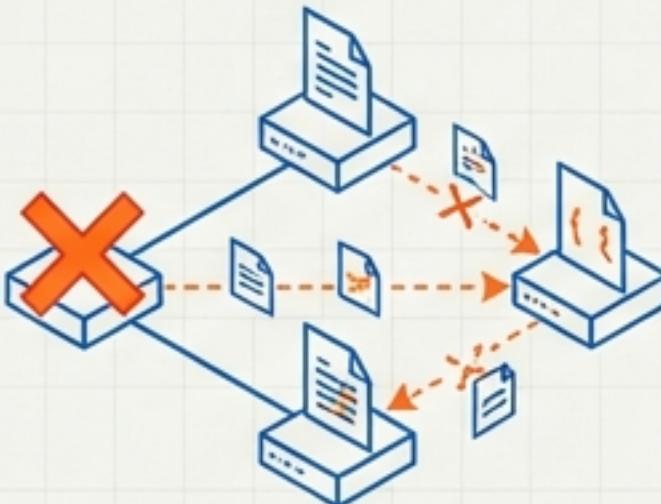
Objetivos: Balanceo de carga Tolerancia a fallos Localidad (baja latencia).

Seguridad, Concurrency y Recuperación

Riesgos

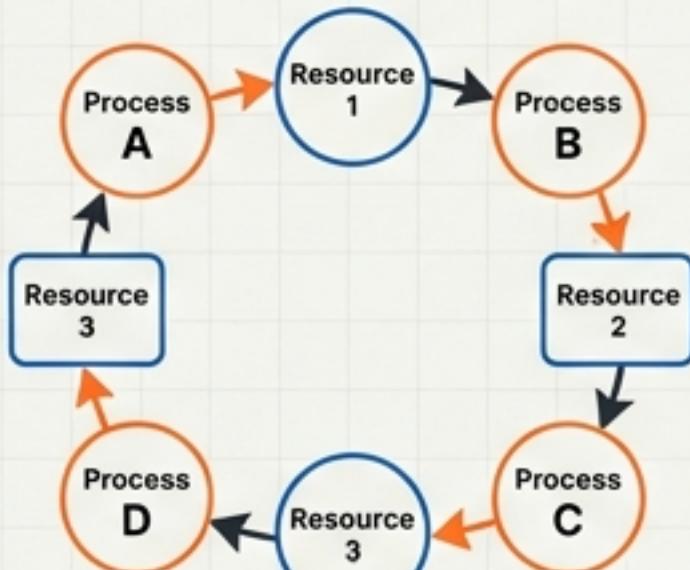
Inconsistencia & Latencia

Fallos en la red o nodos caídos pueden corromper datos.



Interbloqueo (Deadlock)

Procesos esperando recursos mutuamente bloqueados.



Seguridad de Red: Cifrado y VPN para transporte de datos.

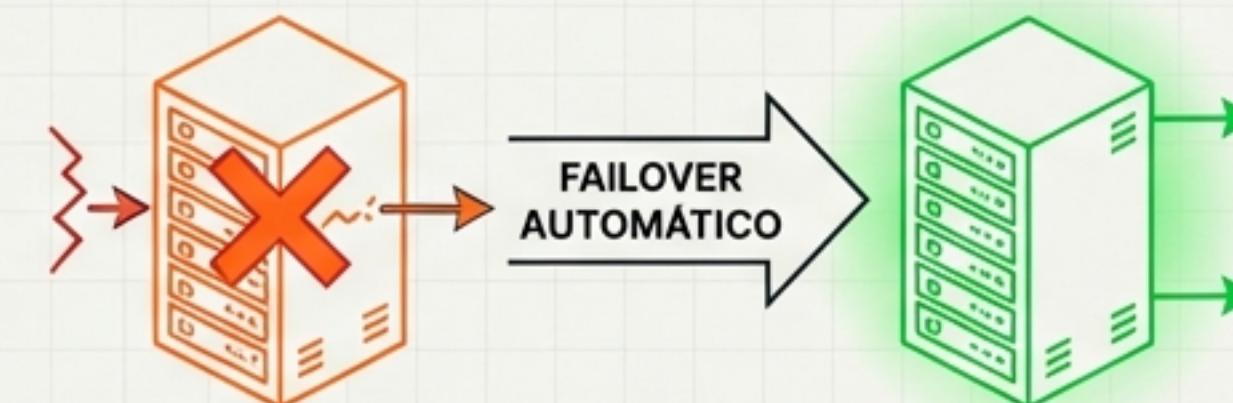
Soluciones

Commit en Dos Fases



Garantiza atomicidad.

Replicación & Failover



Continuidad inmediata del servicio.

Caso Práctico: Diseño para "Tienda y Almacén"



Modelo Elegido: Relacional.
(Razón: Sencillez y rapidez).

Proveedores

Ciientes

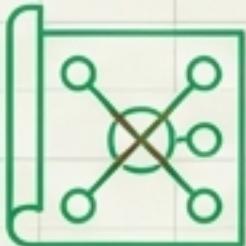
Arquitectura:
Centralizada.

Al disponer de una única tienda física, la complejidad de una distribuida no es necesaria. Un sistema centralizado es suficiente y eficiente.

Resumen Ejecutivo: El Valor de la Distribución



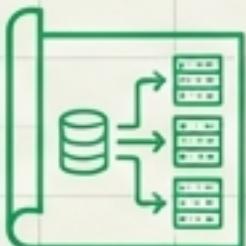
Ventajas



Confiabilidad: Si un nodo falla, el sistema persiste.



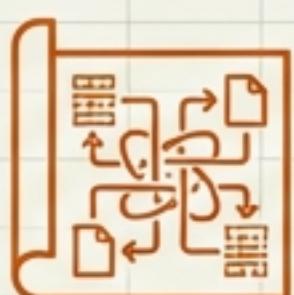
Escalabilidad: Añadir nodos es modular.



Eficiencia: Balanceo de carga inteligente.



Desafíos



Complejidad de Diseño: Requiere esquemas de fragmentación y réplica.



Coste Operativo: Sincronización y control de concurrencia.

La elección del modelo depende del equilibrio entre la necesidad de disponibilidad global y la complejidad de gestión.