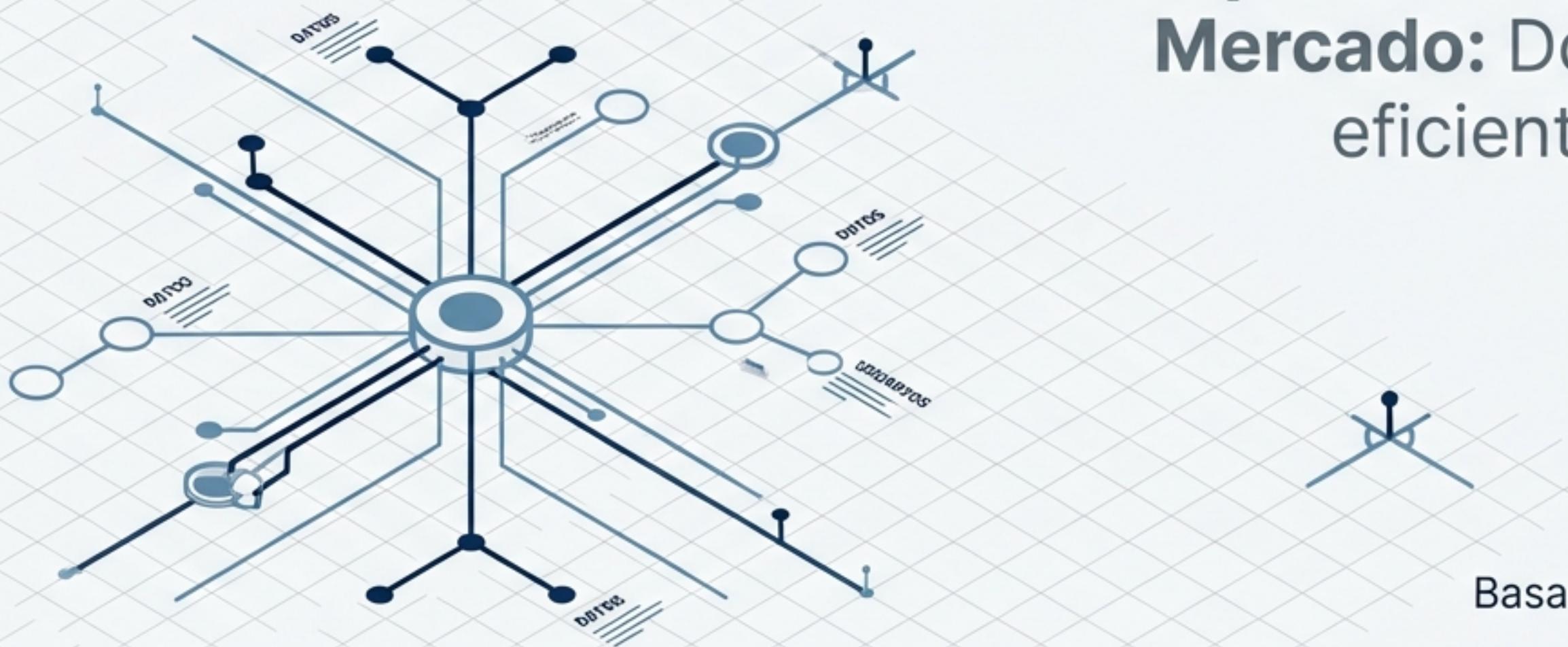


# Sistemas Gestores de Bases de Datos (SGBD)

**Arquitectura, Funcionamiento y Mercado:** Dominando la gestión eficiente de la información.



Basado en la Unidad 3: Base de Datos.

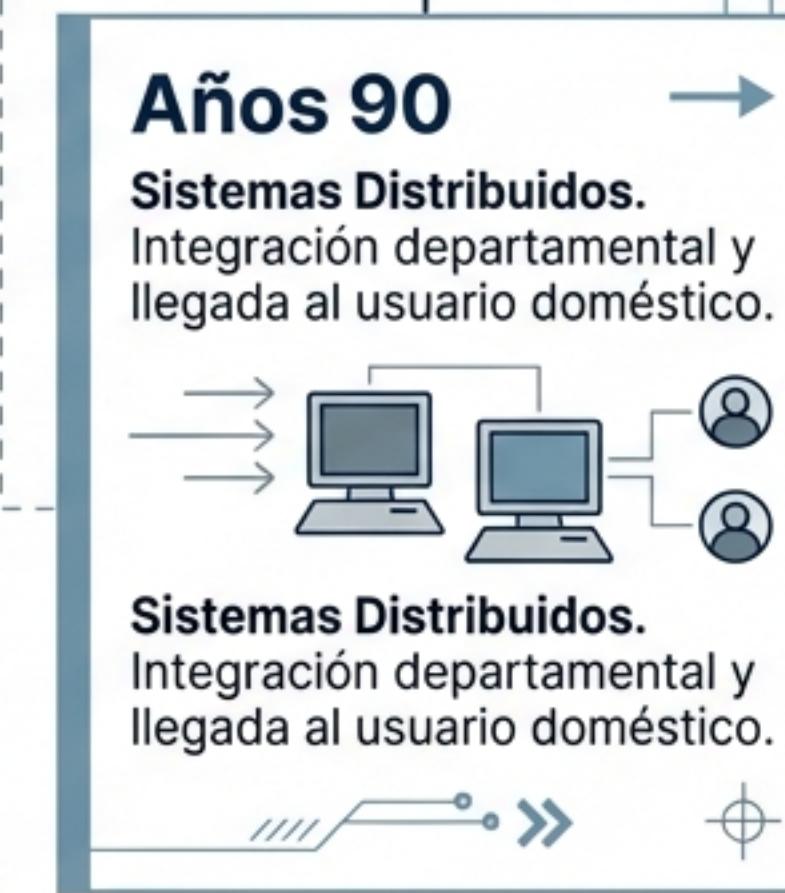
# Evolución Histórica: De la Dependencia del Hardware al Big Data

Años 60

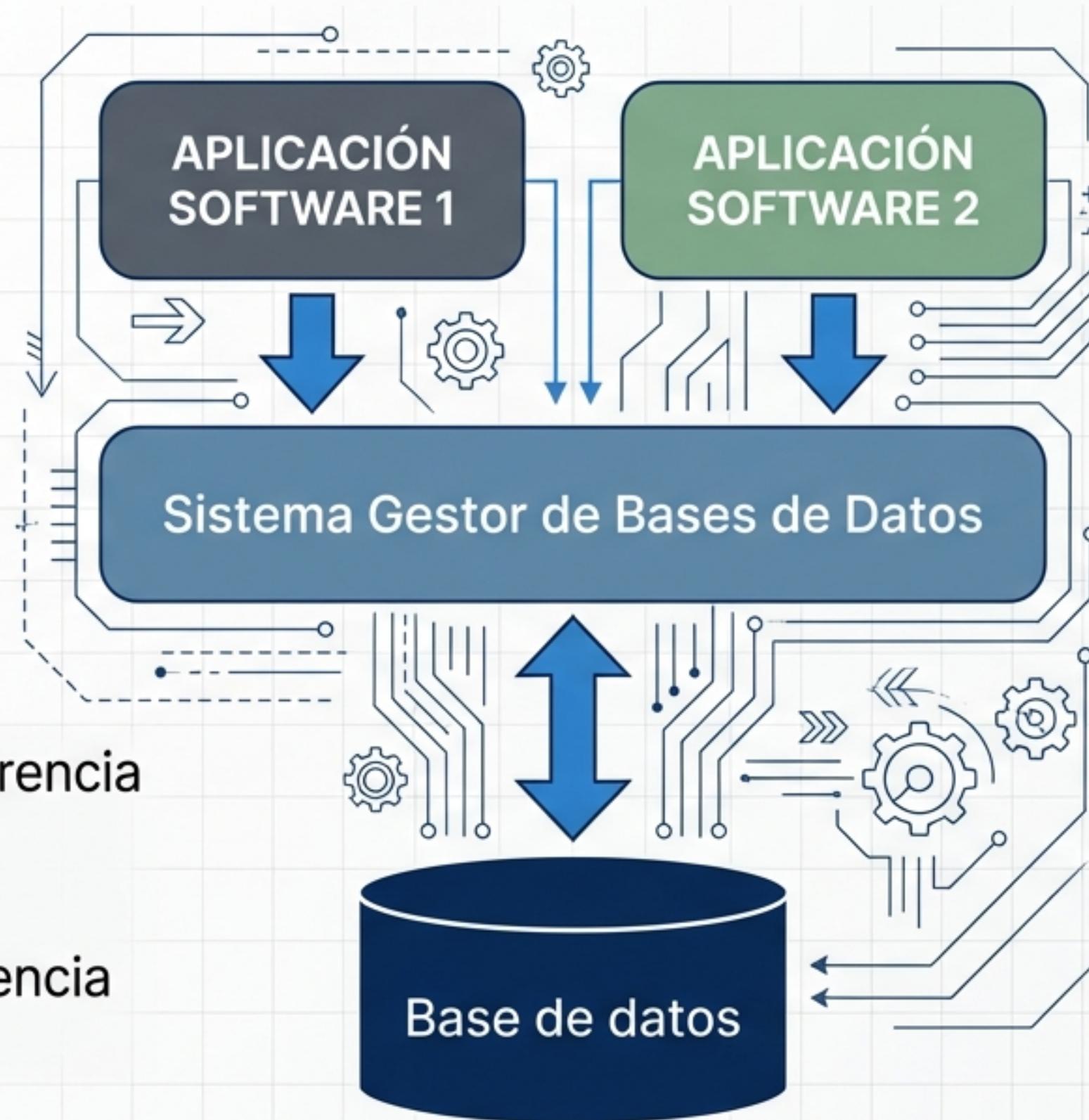
Años 80

Años 90

Actualidad



# El SGBD como Intermediario Crítico de la Información



- Eficiencia y Transparencia
- Control y Seguridad
- Integridad de datos
- Gestión de Conurrencia
- Abstracción

## Definición:

Software que facilita la gestión y manipulación de datos, garantizando disponibilidad y seguridad. Interfaz crítica entre usuario y dato.

# El Salto Cualitativo: SGBD vs. Ficheros Clásicos

## Gestión por Ficheros (El Problema)



- Redundancia incontrolada.
- Dificultad de acceso y búsqueda.
- Inseguridad y falta de integridad.

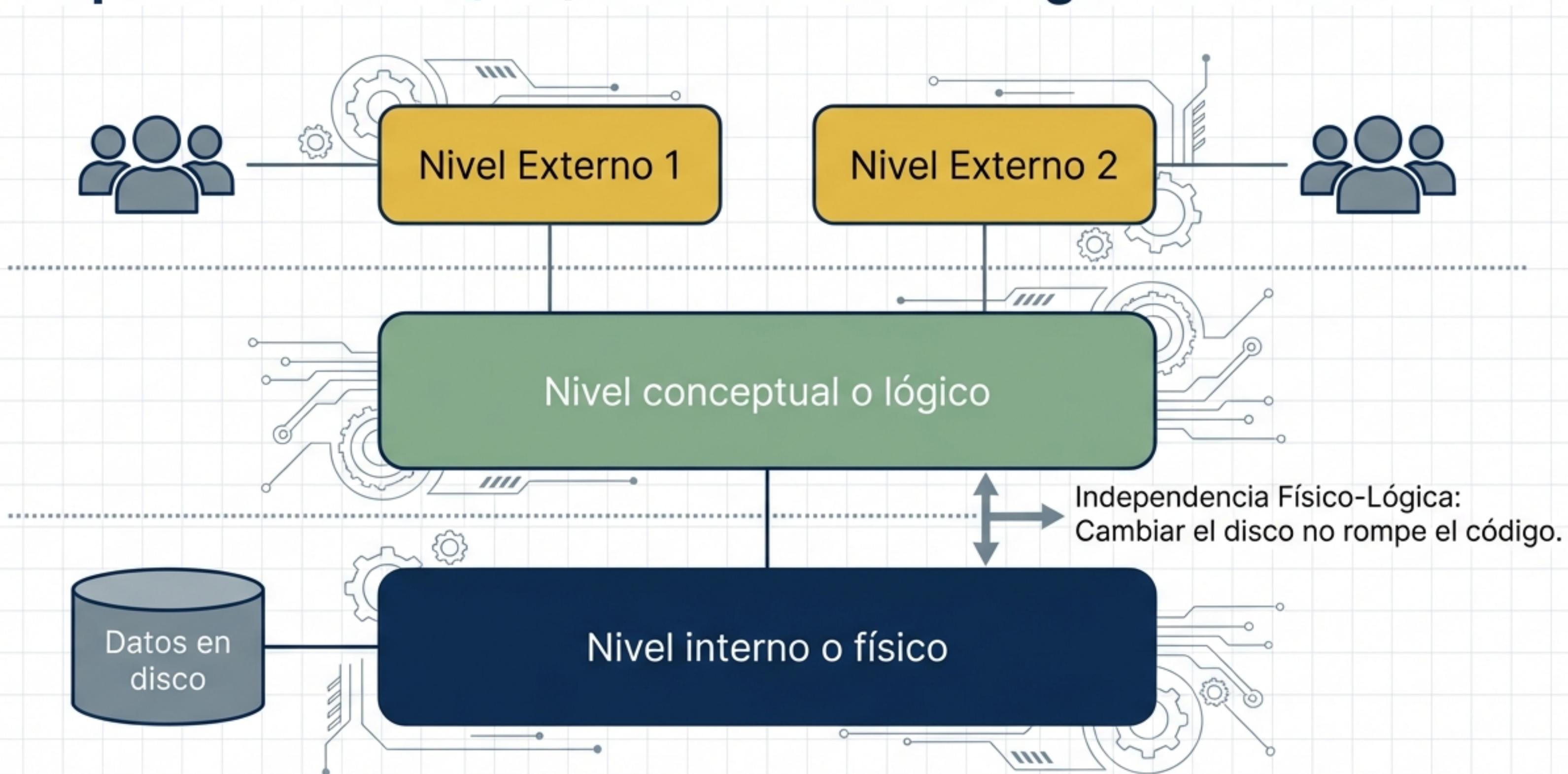


## Gestión por SGBD (La Solución)



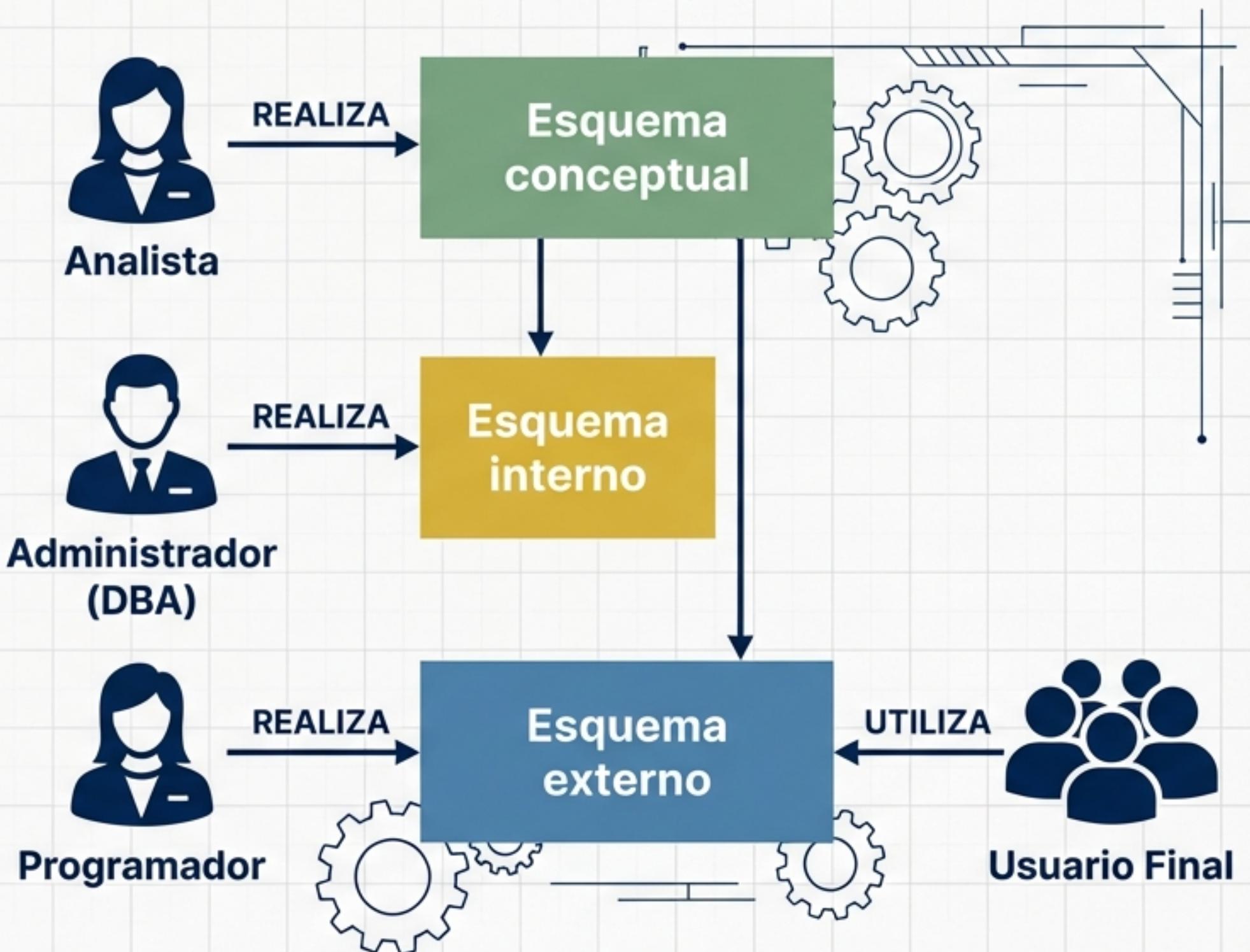
- • Optimización: Mínima redundancia.
- • Acceso Concurrente: Sin conflictos.
- • Seguridad: Control de accesos.
- • Estándares: Consultas vía SQL.

# Arquitectura ANSI/X3/SPARC: La Estrategia de Abstracción

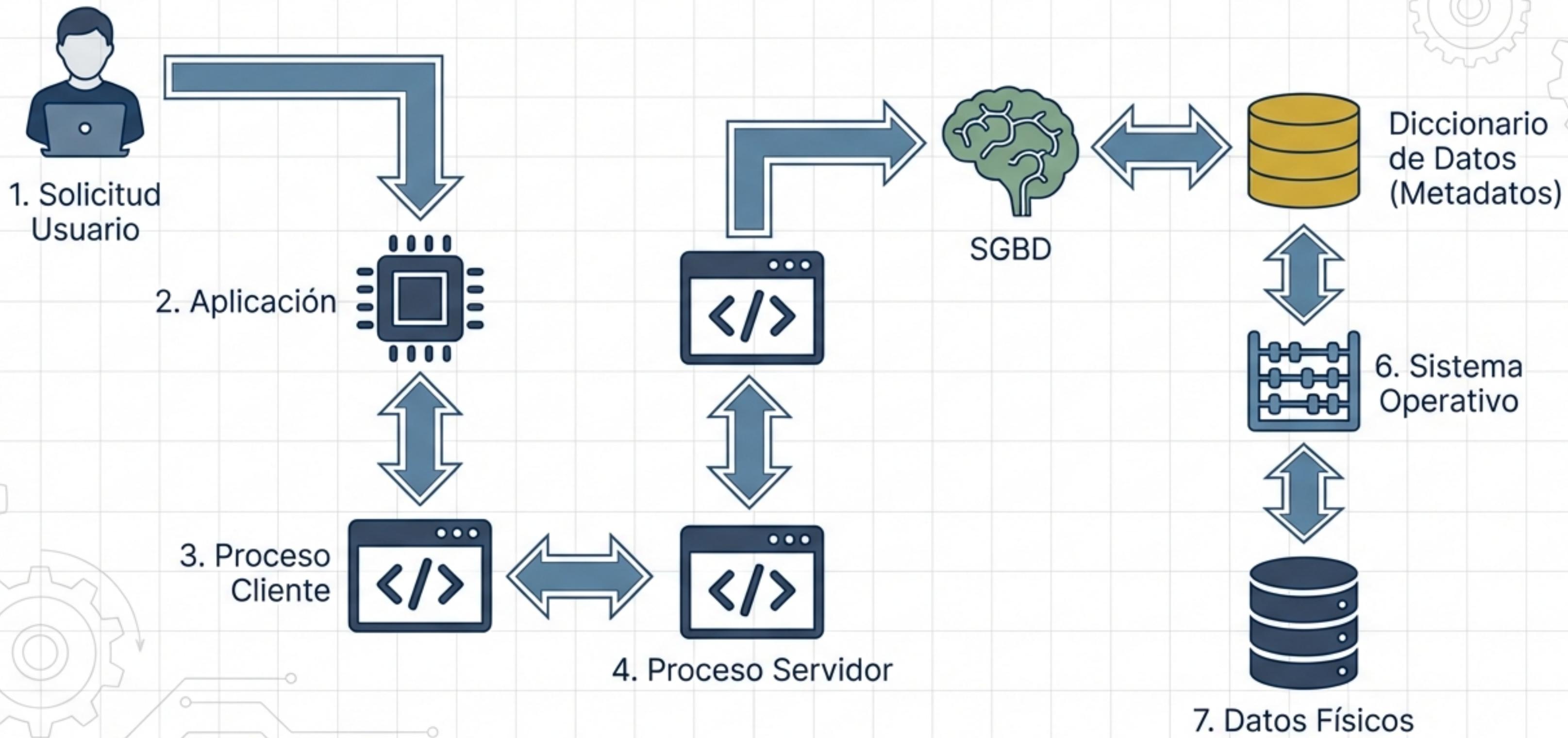


# Componentes del Ecosistema SGBD

- Diccionario de Datos (Metadatos)
- Lenguajes de Programación (SQL)
- Herramientas y Utilidades (Backup/Admin)



# Anatomía de una Consulta: Flujo de Operación Paso a Paso



# Clasificación y Tipología de los SGBD



## Por Modelo Lógico

Relacional (SQL)

NoSQL

Orientado a Objetos

Jerárquico



## Por Tamaño

Ligeros (Access, SQLite)

Alto Rendimiento (Oracle, MongoDB)



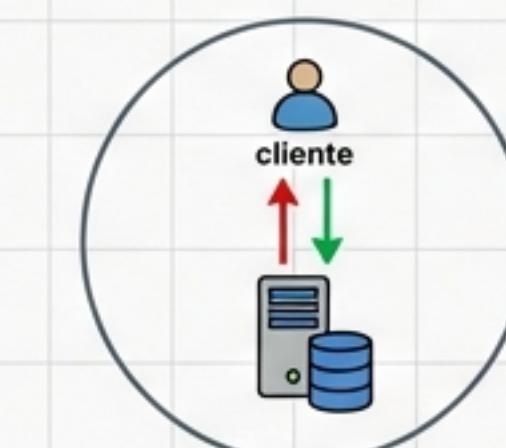
## Por Ubicación

Centralizados

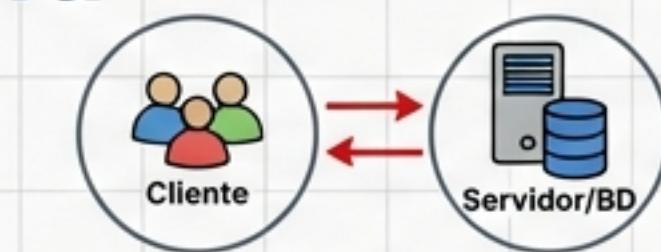
Distribuidos (SGBDD)



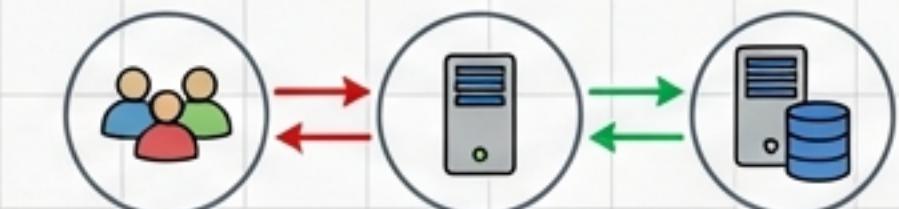
## Por Arquitectura



Monocapa



Bicapa



Cliente

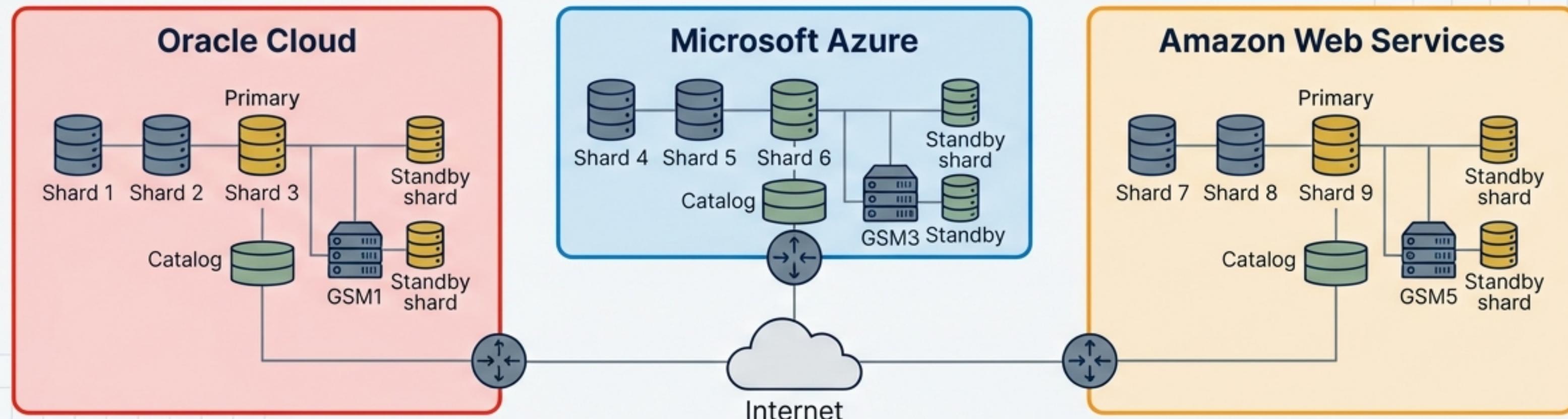
Servidor de Aplicaciones

Servidor de Base de Datos

Multicapa



# El Reto de la Distribución: SGBD Distribuidos (SGBDD)



## Transparencia de Red

El usuario ignora la ubicación física.



## Transparencia de Fragmentación

El usuario ignora si el dato está partido.



## Transparencia de Replicación

El usuario ignora si existen copias.



Ejemplos: Cassandra, PrestoDB, Impala.

# Lenguajes de Bases de Datos: La Familia SQL

## DDL (Definición)

```
CREATE TABLE Clientes (...);  
ALTER TABLE Stock ADD (...);  
DROP INDEX ...
```

## DML (Manipulación)

```
SELECT * FROM Ventas;  
INSERT INTO Pagos VALUES  
(...);  
  
UPDATE ... DELETE ...
```

```
SELECT * FROM Ventas;  
INSERT INTO Pagos VALUES  
UPDATE ... DELETE ...
```

## Control (DCL/TCL)

```
GRANT SELECT ON Tab1 TO  
User;  
  
COMMIT; ROLLBACK;
```

## Note

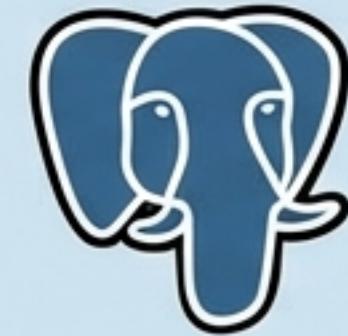
```
// Seleccionar o crear una base de datos  
use miBaseDeDatos  
  
// Crear una colección  
db.createCollection("socios")  
  
// Insertar un solo documento  
db.socios.insertOne({  
  nombre: "Esther",  
  edad: 38,  
  email: "esther.garcia@example.com"  
})
```

# Panorama del Mercado: Soluciones Libres (Open Source)



## MySQL

Líder web.  
Propiedad de  
Oracle. Ideal  
para PHP/Java.



## PostgreSQL

Potencia y  
Objetos. Gran  
comunidad.



## SQLite

Ligero, sin  
servidor.  
Estándar en  
móviles.

## MongoDB

NoSQL.  
Orientado a  
documentos  
JSON.  
Escalabilidad  
horizontal.

# Panorama del Mercado: Soluciones Comerciales

**ORACLE®**

## Oracle Database.

Estándar empresarial.  
Máxima robustez y coste.



## SQL Server.

Integración total Microsoft.  
Potente en Business Intelligence.

**IBM  
DB2**

**IBM DB2**  
Legado Mainframe.  
Automatización y alto rendimiento.

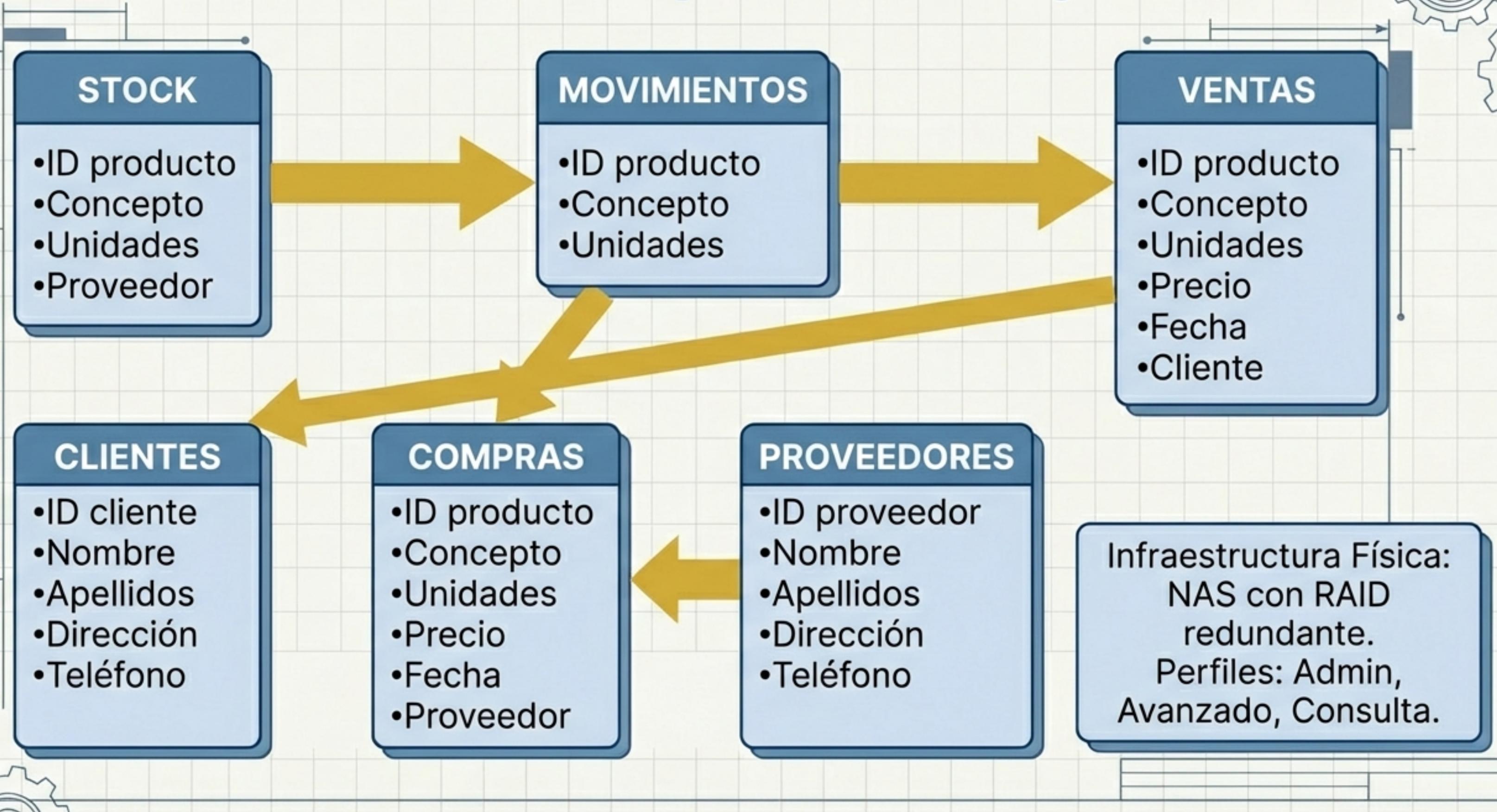
**Sybase-SAP**

Grandes volúmenes y escalabilidad.

# Matriz Comparativa para Toma de Decisiones

SGBD	Ventajas	Inconvenientes
Access	Muy gráfico, simple.	No escala, solo Windows.
SQLite	Sin servidor, portable.	Sin claves foráneas, difícil concurrencia.
MySQL	Rápido, estándar Web.	Soporte limitado en versión libre.
PostgreSQL	Potente, Objetos, Libre.	Lento en algunas config., curva de uso.
Oracle	Líder mundial, seguridad total.	Coste muy elevado, complejo.
MongoDB	Flexible (JSON), Escalable.	Alto consumo disco, redundancia.

# Caso Práctico: Diseño para 'Tienda y Almacén'



# Conclusiones y Claves del Éxito



**Arquitectura de 3 Niveles.** Vital para la independencia físico-lógica.



**Seguridad y Abstracción.** El SGBD es la barrera necesaria entre usuario y dato.



**Elección Estratégica.** Oracle para banca, MySQL para web, SQLite para apps.



**Tendencia Futura.** Distribución transparente y servicios en la nube.