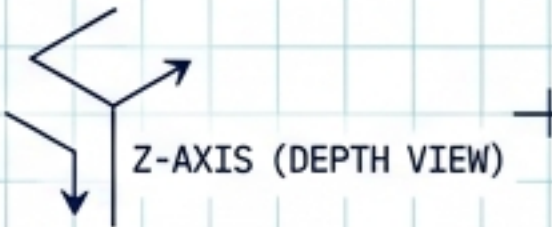


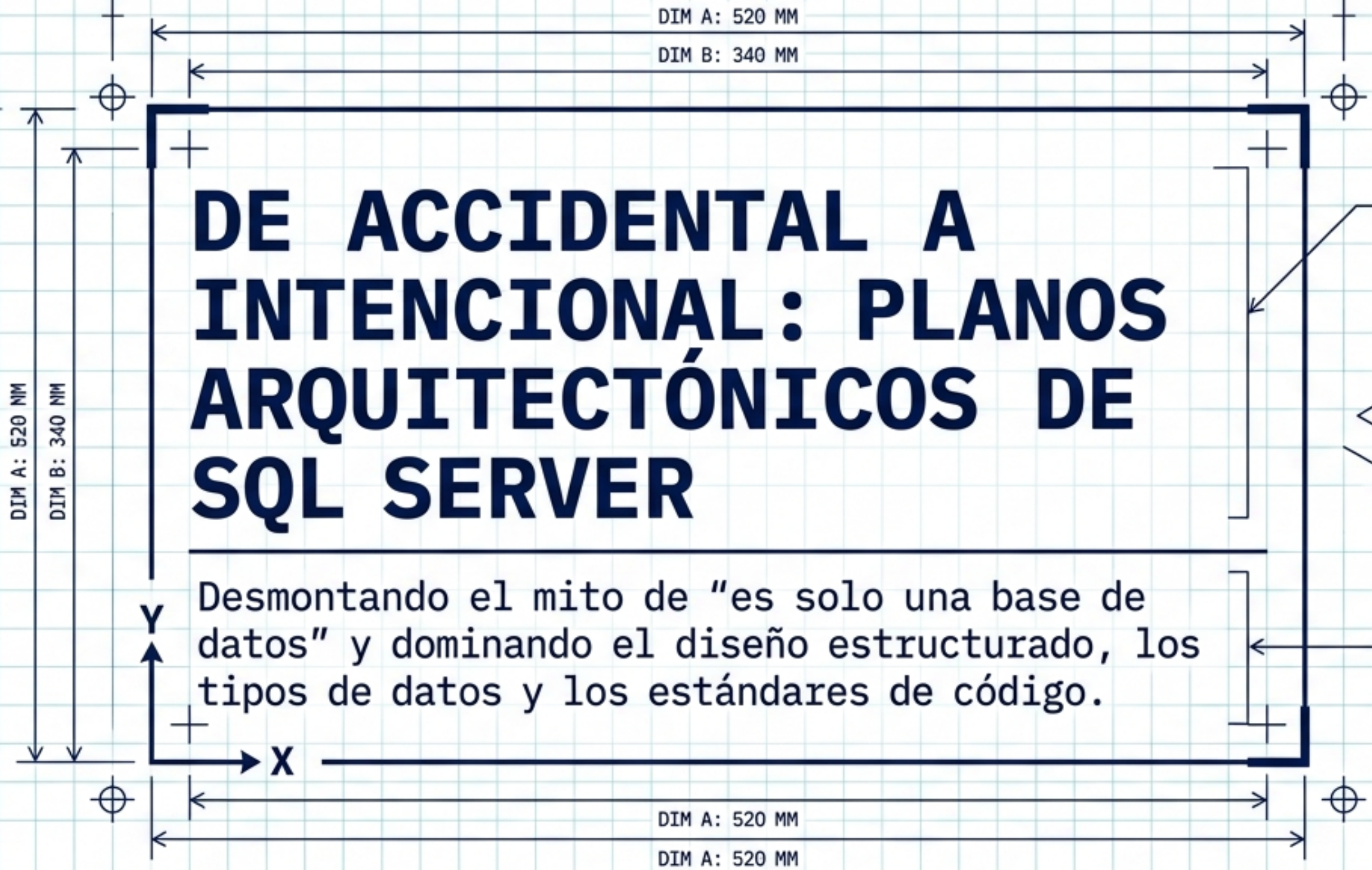
DE ACCIDENTAL A INTENCIONAL: PLANOS ARQUITECTÓNICOS DE SQL SERVER

Desmontando el mito de "es solo una base de datos" y dominando el diseño estructurado, los tipos de datos y los estándares de código.

PRIMARY SCHEMATIC TITLE
BLOCK: DATABASE
ARCHITECTURE



PROJECT SCOPE:
STRUCTURE & STANDARDS



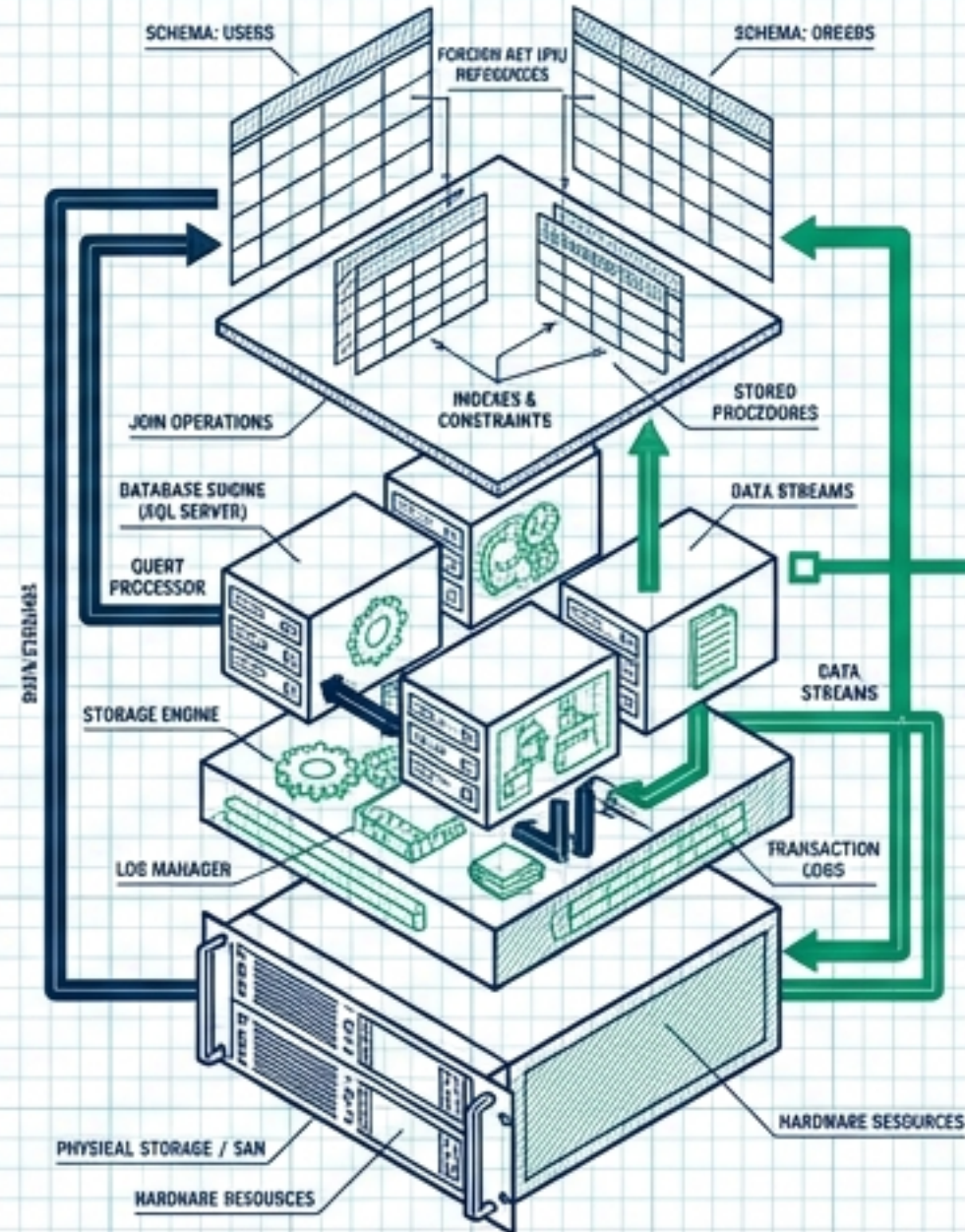
EL ERROR CERO: "ES SOLO UNA BASE DE DATOS"

EL MITO



La suposición de simplicidad lleva a presupuestos bajos, falta de planificación y la creación del DBA Accidental.

LA REALIDAD

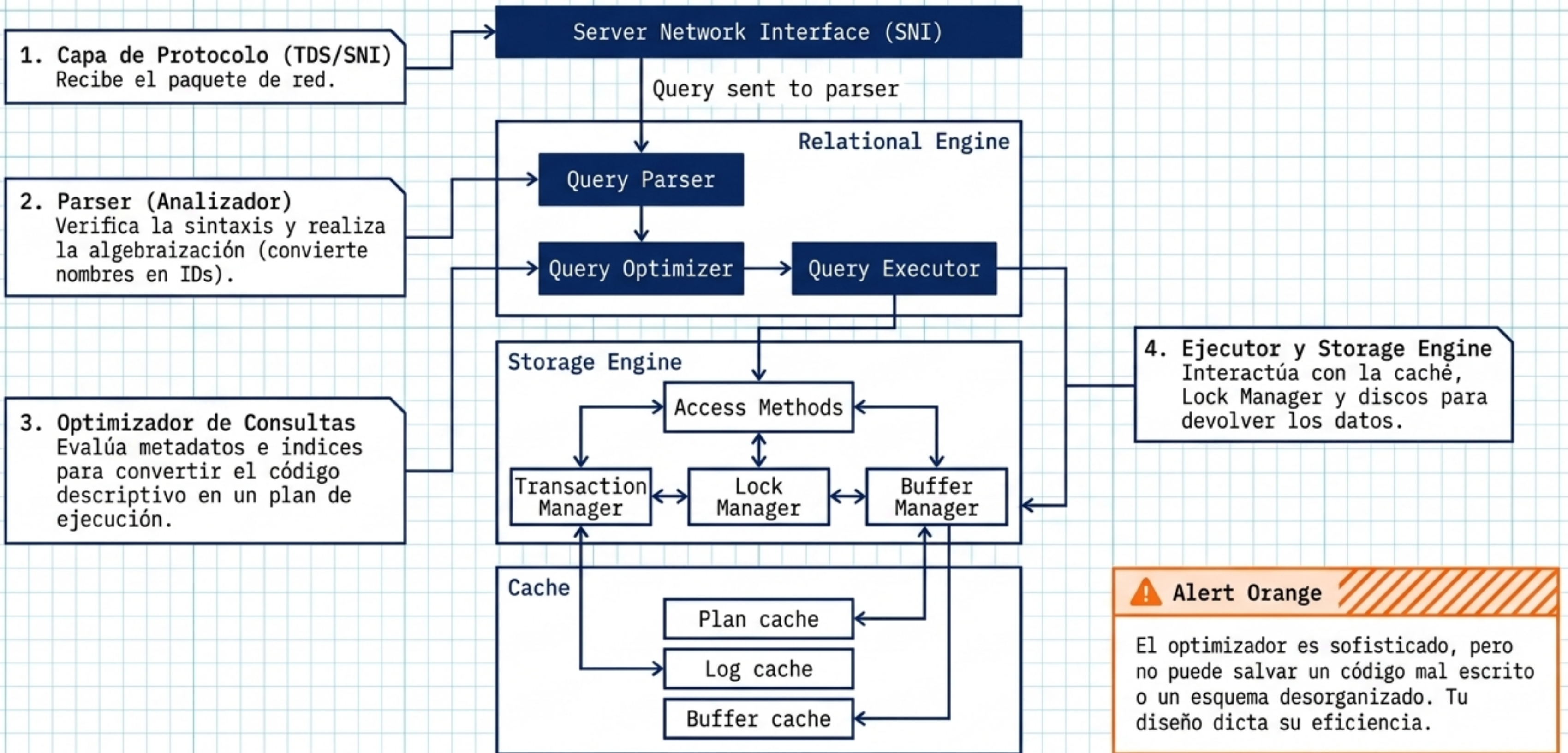


SQL Server es un motor de base de datos relacional de alta ingeniería. Cada decisión intuitiva o "por defecto" genera deuda técnica, pérdida de rendimiento y vulnerabilidades a escala.



UN MINUTO PARA APRENDER,
TODA UNA VIDA PARA DOMINAR.

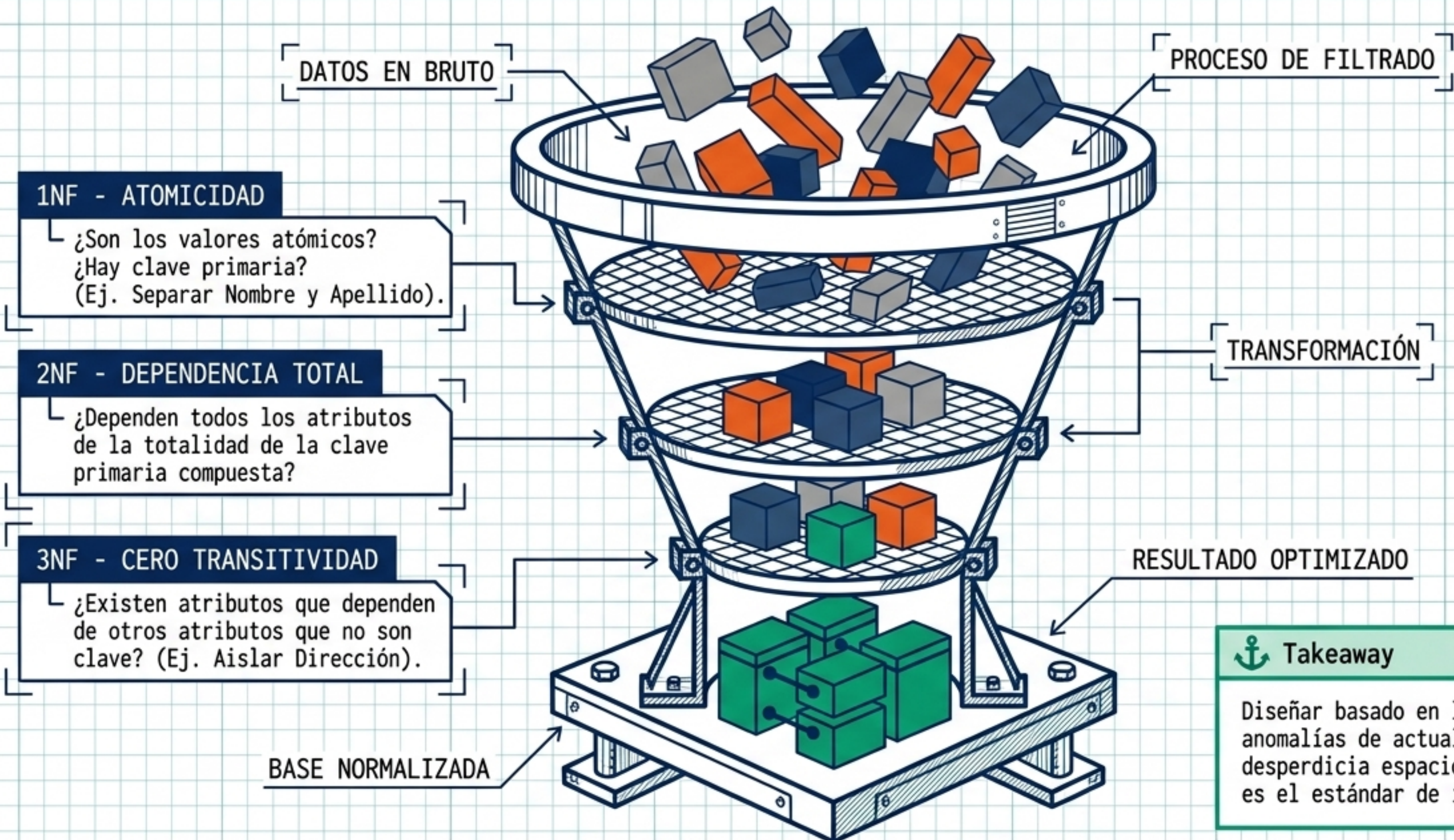
LA MAQUINARIA INTERNA: EL VIAJE DE UNA CONSULTA



⚠ Alert Orange

El optimizador es sofisticado, pero no puede salvar un código mal escrito o un esquema desorganizado. Tu diseño dicta su eficiencia.

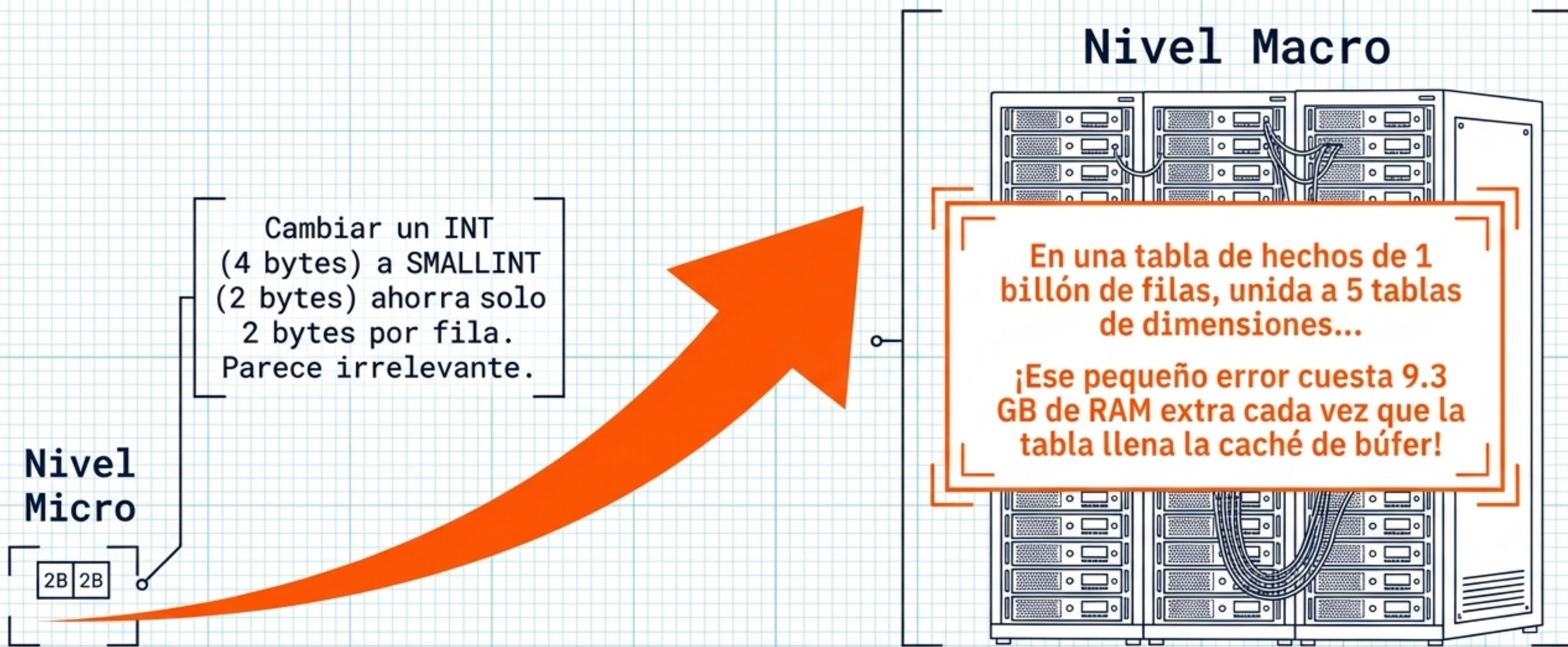
EL FILTRO DE NORMALIZACIÓN (EVITANDO EL ERROR #11)



Takeaway

Diseñar basado en la intuición crea anomalías de actualización y desperdicia espacio. La normalización es el estándar de ingeniería.

EL EFECTO MULTIPLICADOR: EL PELIGRO DEL INT POR DEFECTO



Cada byte desperdiciado empuja datos útiles fuera de la memoria RAM hacia el disco, destruyendo el rendimiento de las consultas.

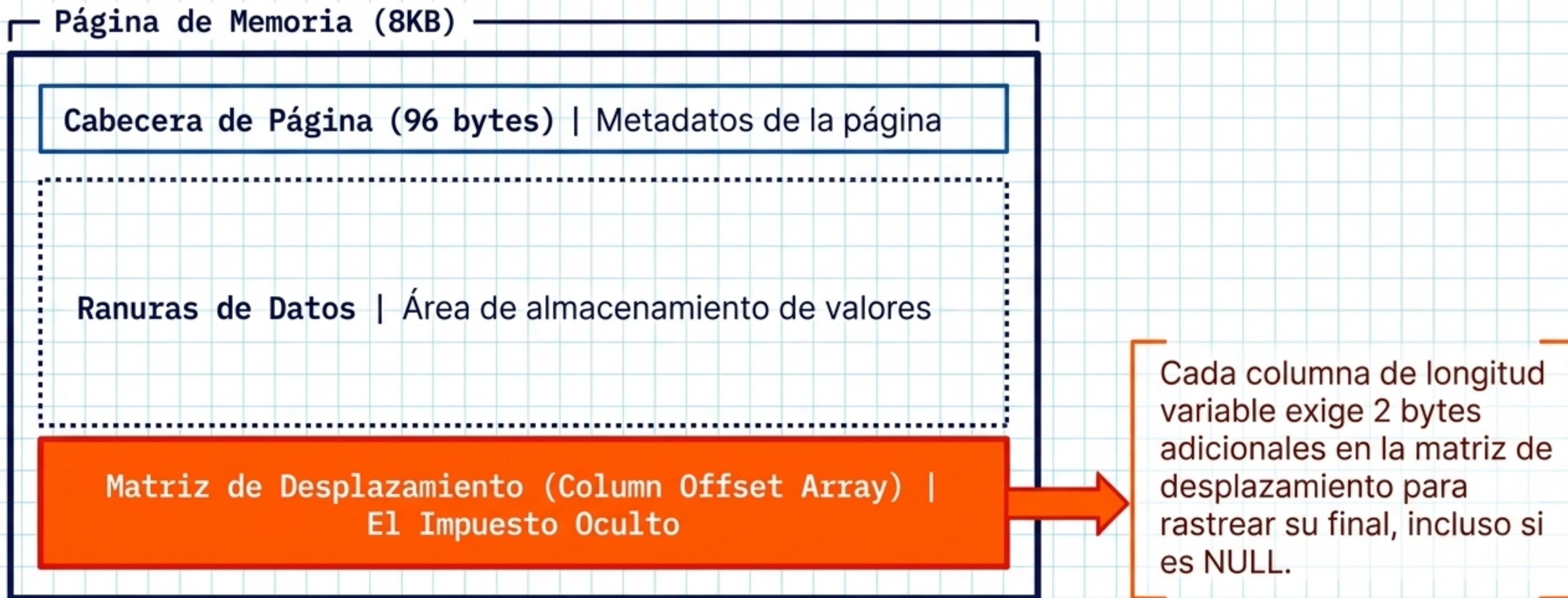
MATRIZ DE DIAGNÓSTICO: LA FAMILIA DE ENTEROS

Data Type	Technical Specs	Architectural Use Case
TINYINT	1 Byte Rango: 0 a 255	Uso Ideal: Estados, tipos definidos pequeños.
SMALLINT	2 Bytes Rango: -32,768 a 32,767	Uso Ideal: Dimensiones de tamaño mediano, años.
INT	4 Bytes Rango: -2.1 Billones a 2.1 Billones	Uso Ideal: Claves primarias estándar. (NO usar por defecto).
BIGINT	8 Bytes Rango: Astronómico	Uso Ideal: Telemetría masiva, data warehousing global.



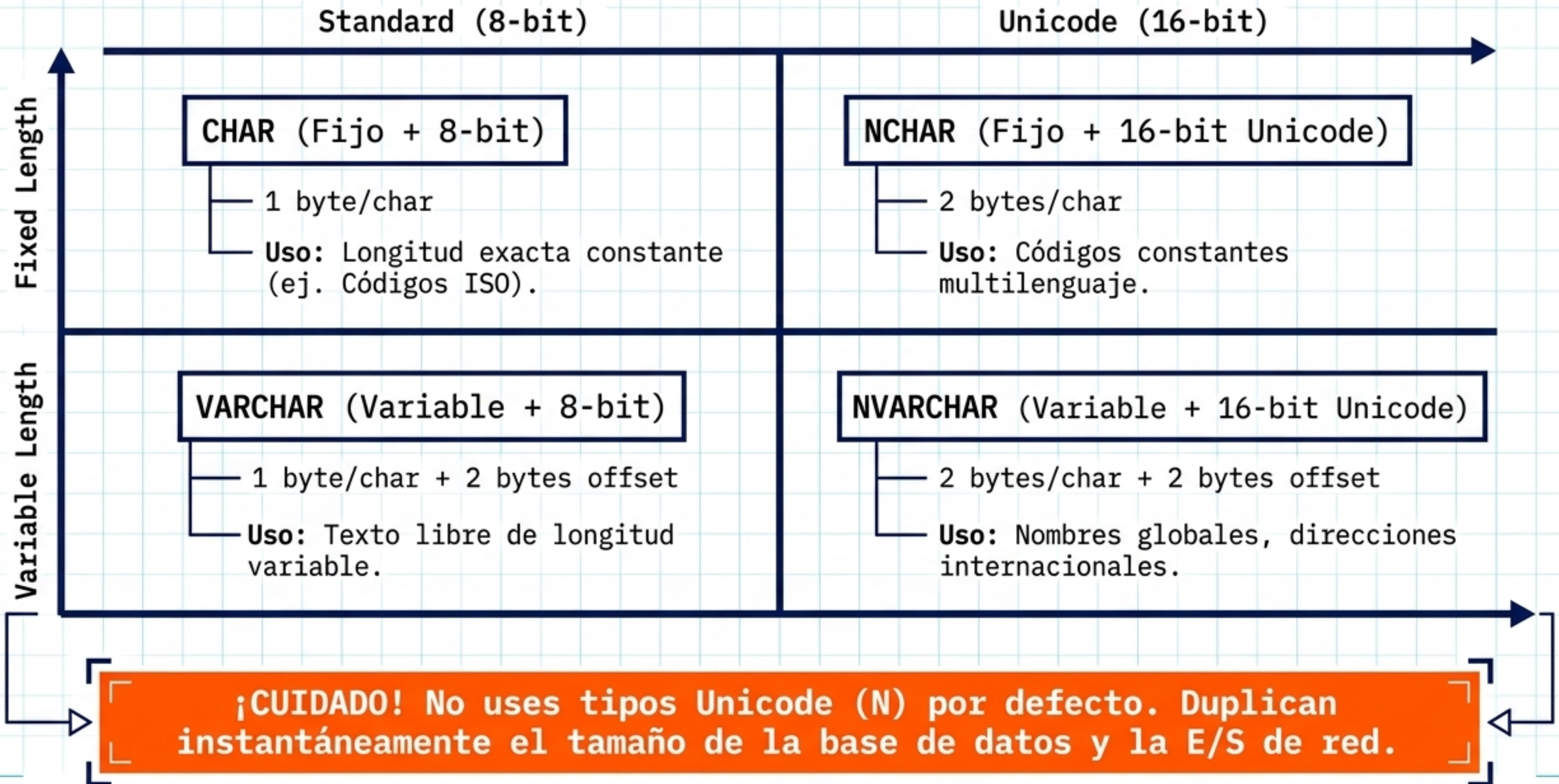
TIP ARQUITECTÓNICO: Usa BIT para banderas booleanas (0, 1). SQL Server optimiza hasta 8 columnas BIT en un solo byte de almacenamiento.

EL IMPUESTO OCULTO DE VARCHAR



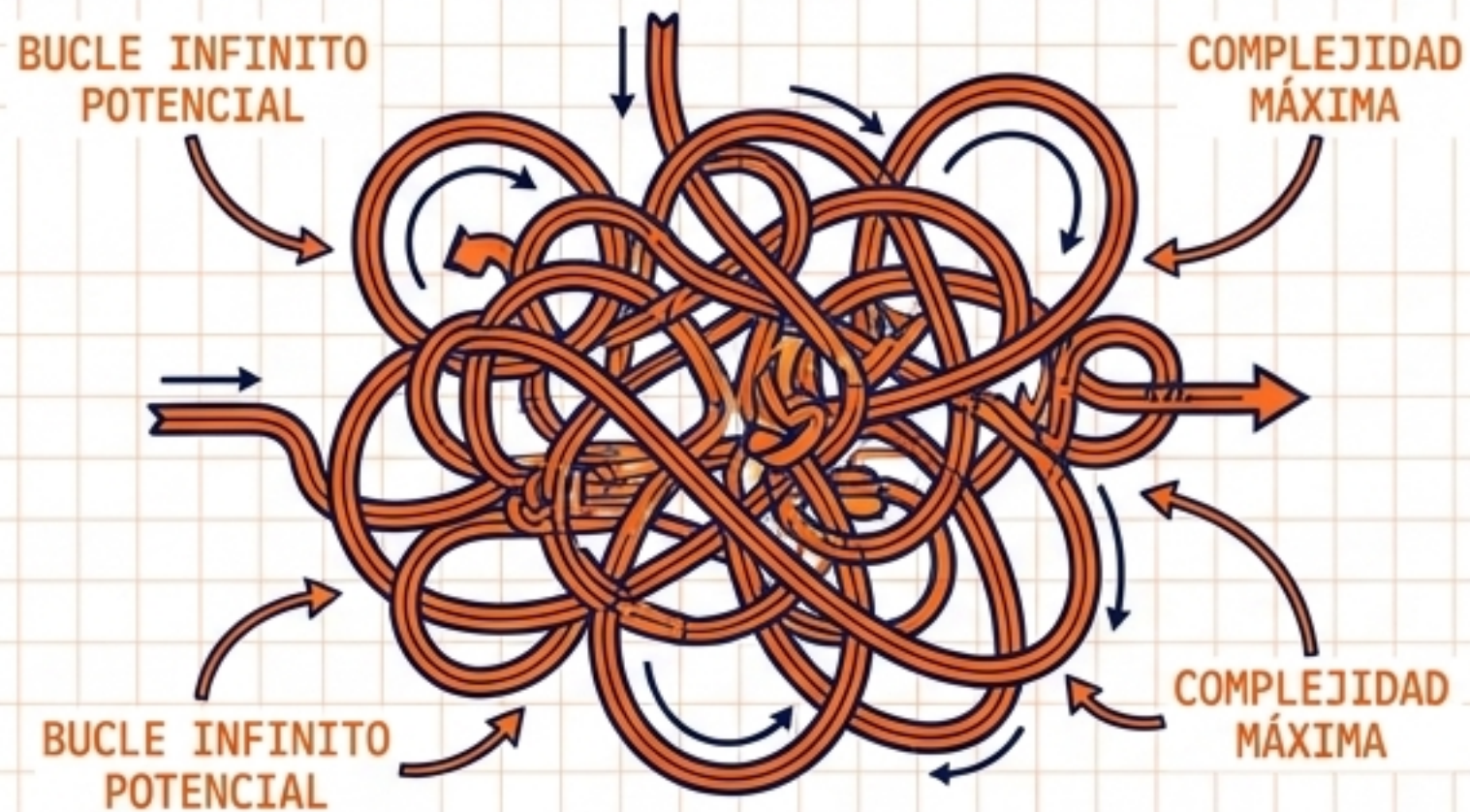
CONCLUSIÓN: Si el dato siempre tiene la misma longitud (Ej. Código Postal de 5 caracteres), VARCHAR(5) desperdicia 2 bytes por fila frente a CHAR(5).

MATRIZ DE DECISIÓN PARA ALMACENAMIENTO DE CADENAS



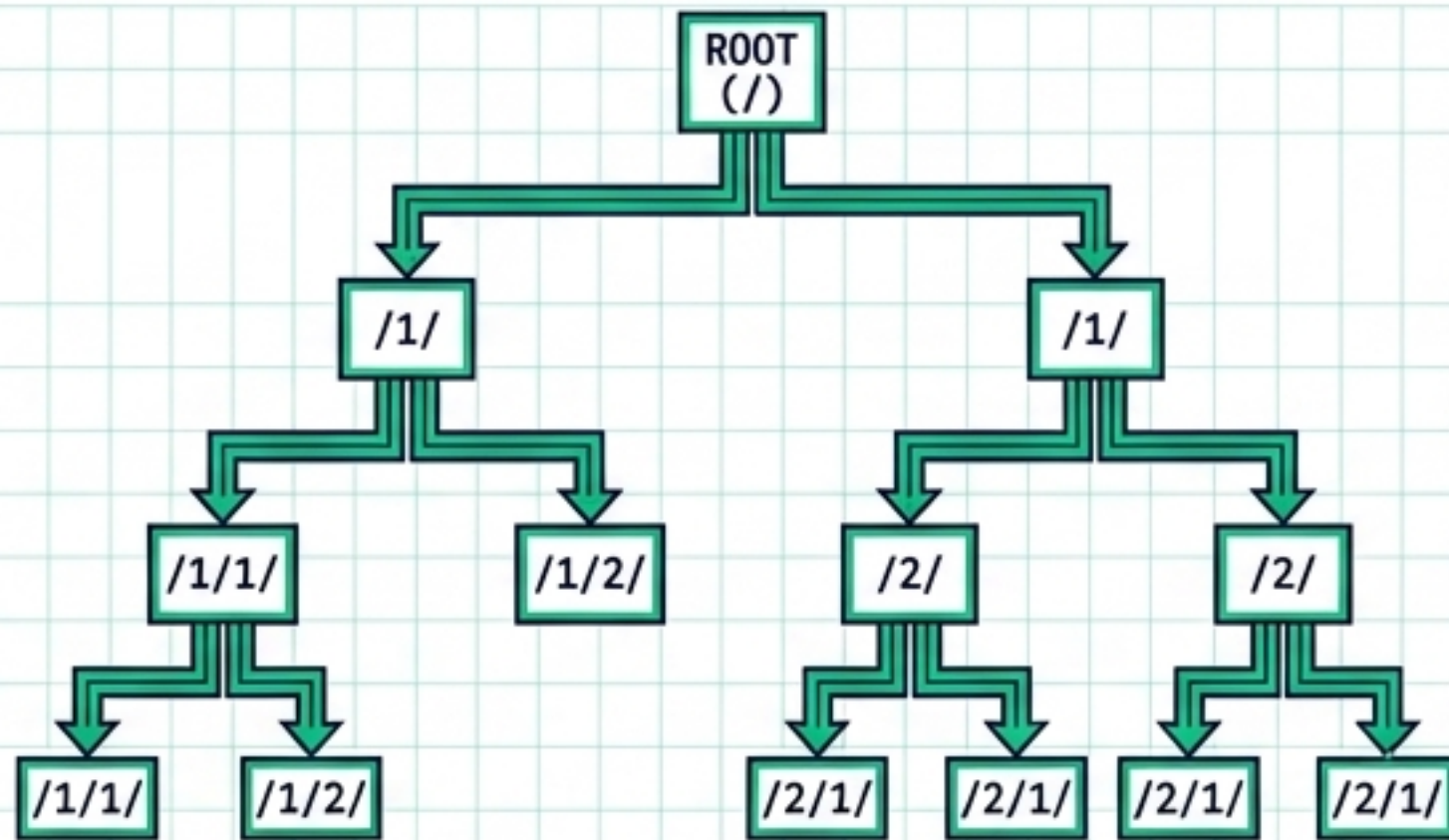
DOMANDO JERARQUÍAS: BUCLES VS. ÁRBOLES

EL BUCLE PELIGROSO (CTE RECURSIVO)



Requiere código complejo (UNION ALL), alto costo de CPU y riesgo de bucles infinitos (requiere protección MAXRECURSION).

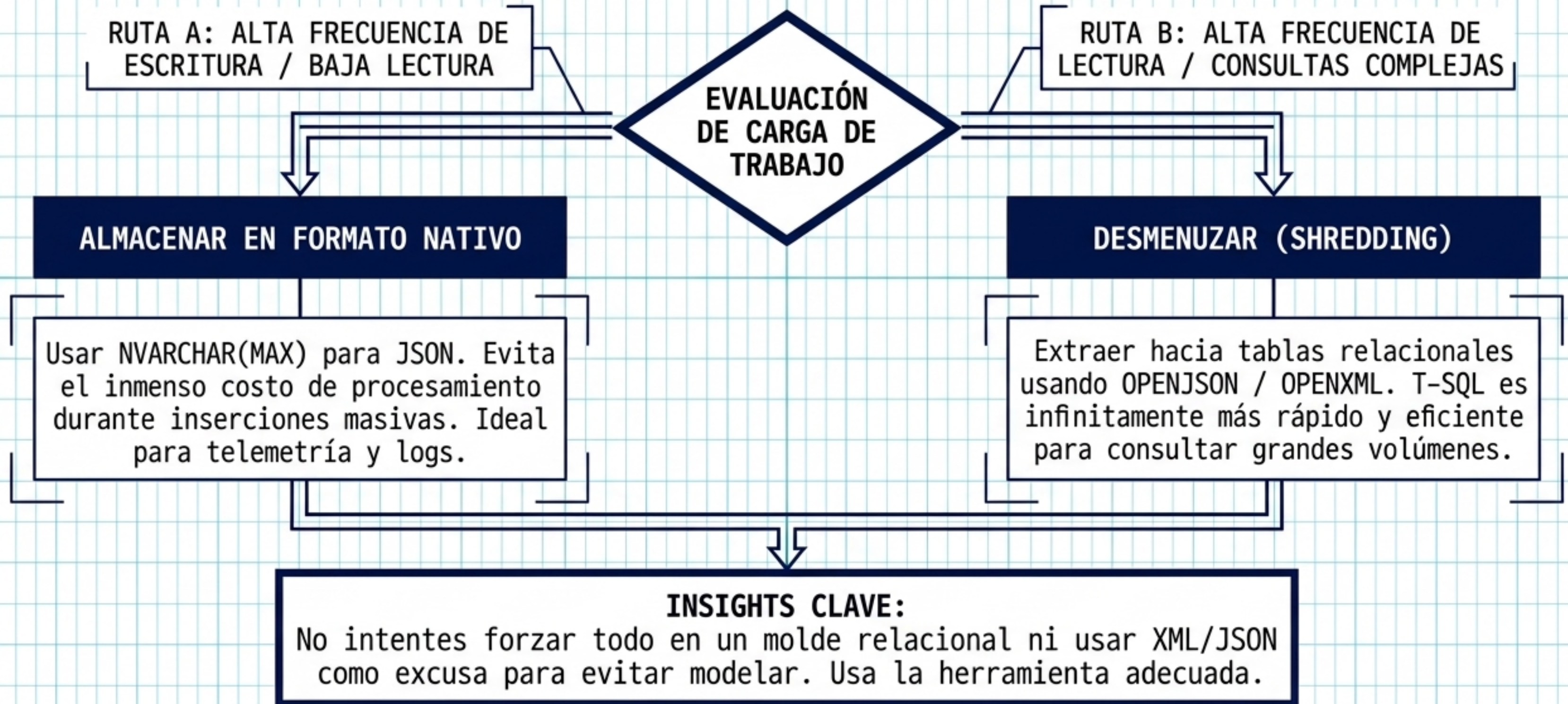
EL ÁRBOL NATIVO (HIERARCHYID)



Almacenamiento nativo basado en rutas (Ej: /1/2/1/). Simple, indexado y altamente optimizado.

Sintaxis limpia: `ManagerHierarchyID.IsDescendantOf(@Manager)`

DATOS SEMIESTRUCTURADOS (XML Y JSON): ¿DESMENUZAR O ALMACENAR?



LA CAPA DE APLICACIÓN: CONVENCIONES Y CARGA COGNITIVA

EL CAOS VISUAL

`tbl_customers`
`tbl_orders`
`sp_update`
`v_customers`

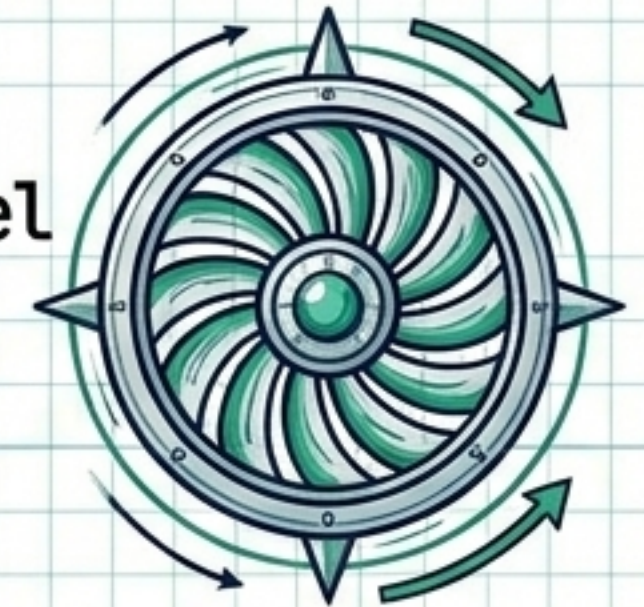


PROCESAMIENTO
CEREBRAL LENTO

Elimina prefijos inútiles (`tbl_`, `v_`). Tu cerebro reconoce patrones; forzarlo a ignorar los primeros 4 caracteres ralentiza drásticamente el escaneo.

EL PLANO LIMPIO

`customers`
`orders`
`updateProductStockLevel`
`customerOrders`



PROCESAMIENTO
CEREBRAL RÁPIDO

Código Autodocumentado: Parámetros como `@Address` son ambiguos. Usa `@BillingAddressID`. El código debe explicarse sin necesidad de comentarios.

El Desvío Costoso del Prefijo sp_

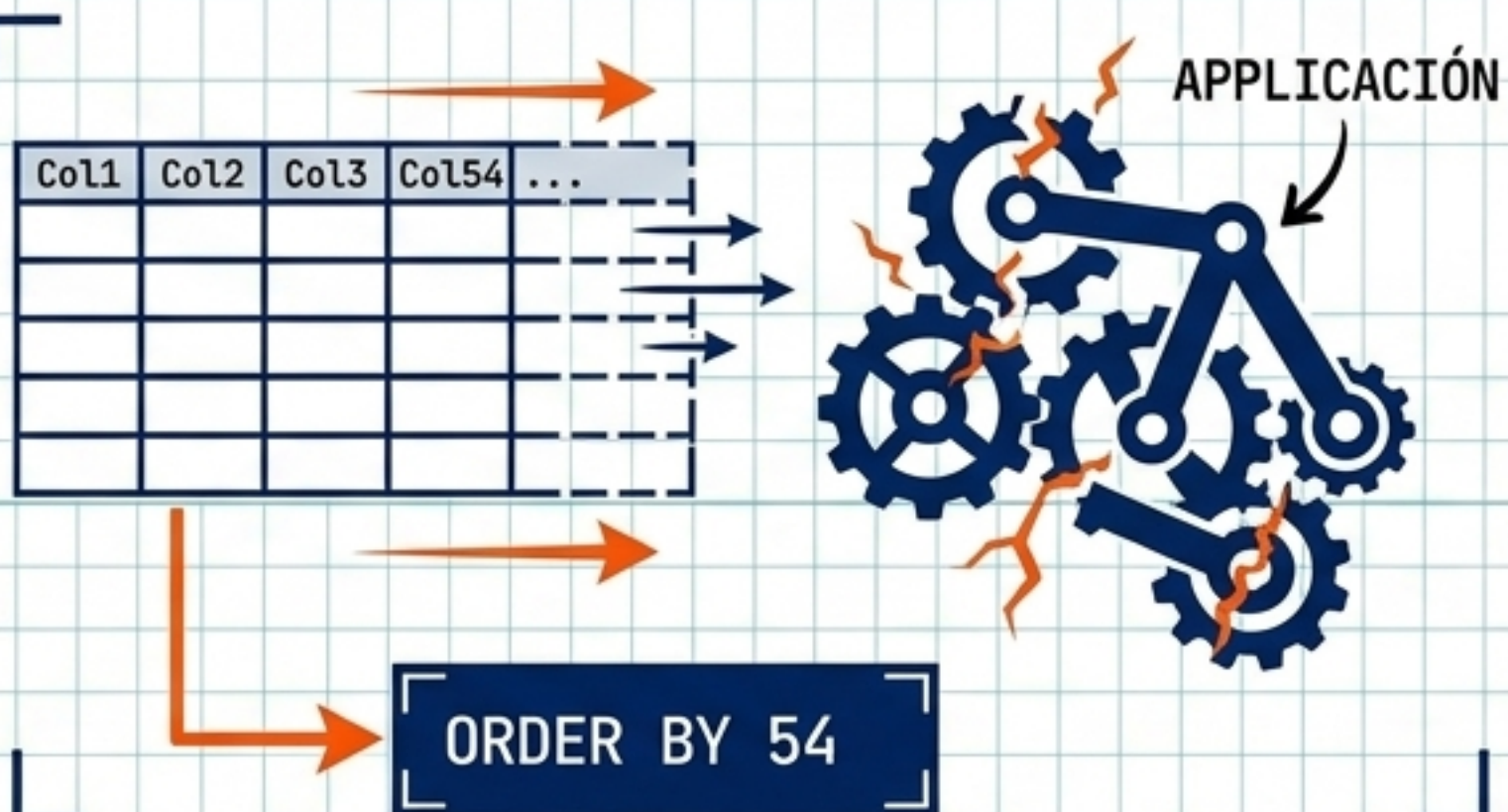


El prefijo sp_ no significa 'stored procedure', significa 'system procedure'. SQL Server siempre buscará primero en la base de datos master.

IMPACTO: Este desvío crea bloqueos innecesarios, picos de CPU y una penalización de rendimiento evitable en cada ejecución.

ESTÁNDARES DE CÓDIGO Y ESTRUCTURAS FRÁGILES

ANTI-PATRÓN 1: POSICIONES ORDINALES



Si cambia el esquema de la tabla, la columna 54 cambia, rompiendo la aplicación silenciosamente. Usa nombres explícitos.

ANTI-PATRÓN 2: CAOS ESTILÍSTICO

```
SeLEct ██████████  
  ↳ from ██████████ ← ?  
  ↳ wHeRe ██████████  
    ↳ SeLEct Tront ██████████  
    ↳ InT ██████████ ← ?  
    ↳ wHeRe ██████████  
  ↳ ObjEct ██████████  
    ↳ Object egt ██████████  
    ↳ wHeRe ██████████
```

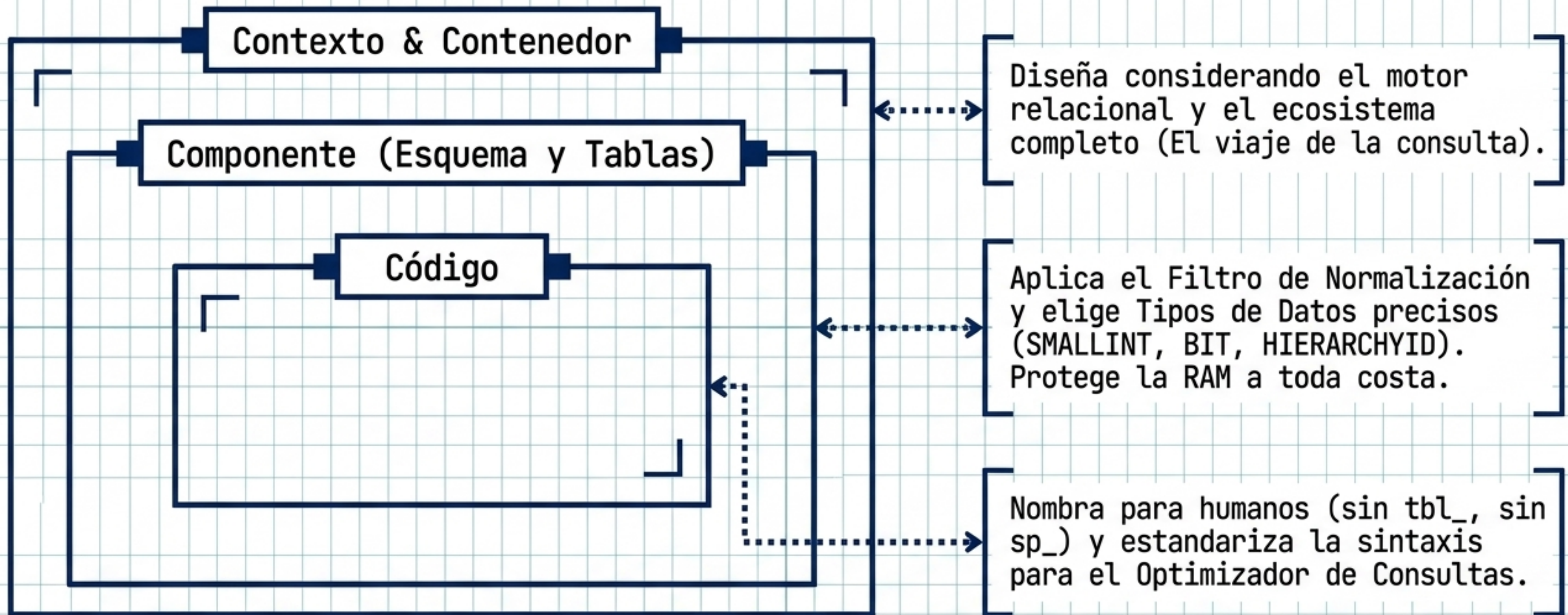
Mezclar espacios, tabulaciones y mayúsculas de forma inconsistente no solo es estéticamente pobre; ralentiza dramáticamente el tiempo de resolución de bugs.



LA FALTA DE ESTÁNDARES NO ES UN SIGNO DE VELOCIDAD,
ES UNA GARANTÍA ABSOLUTA DE DEUDA TÉCNICA FUTURA.



Síntesis: La Arquitectura Intencional (Modelo C4)



Una base de datos no es una simple hoja de cálculo glorificada.
Es un ecosistema interconectado donde los bytes importan.
Actúa como un Arquitecto.