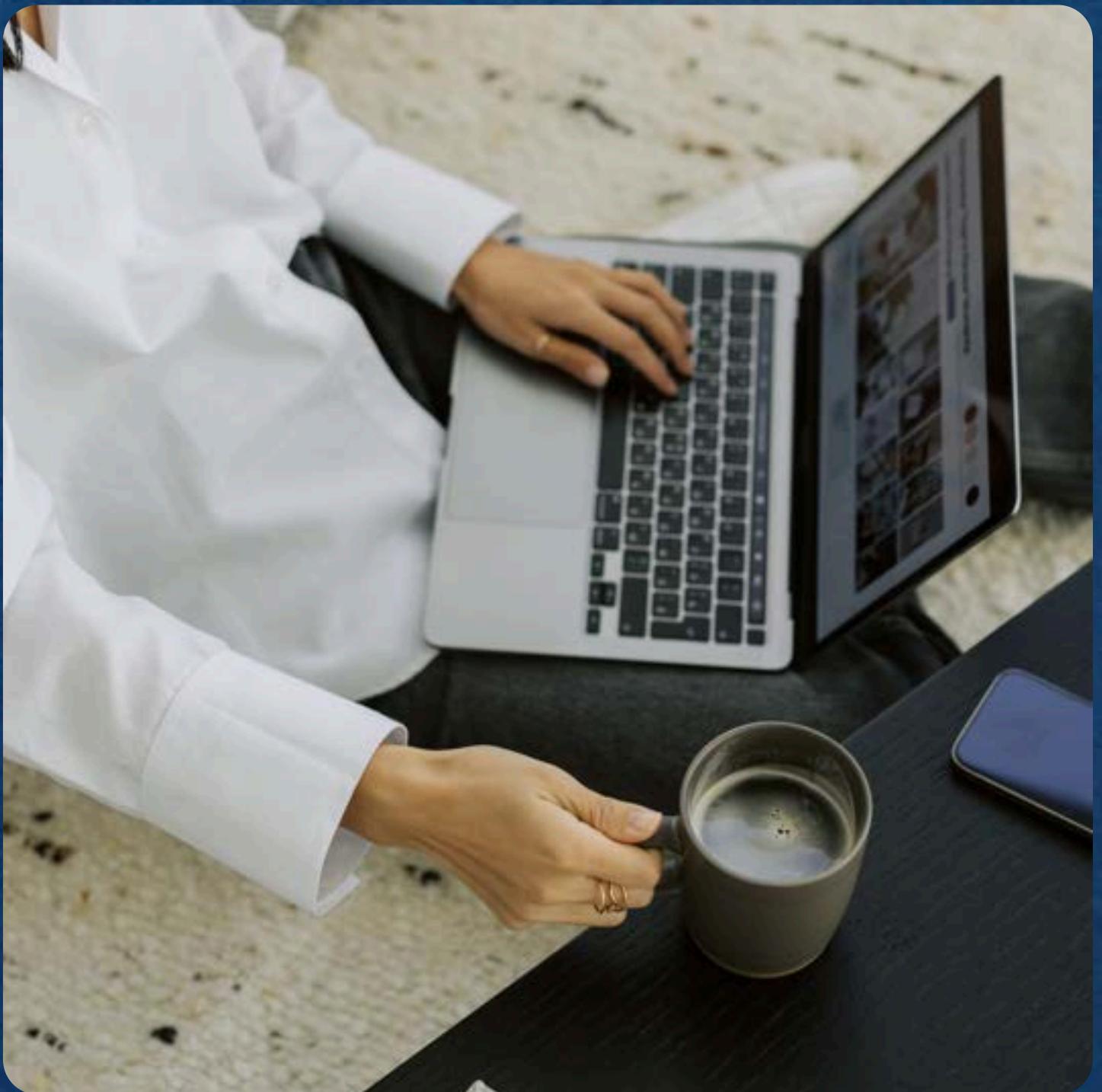


# CLIENTE-SERVIDOR VS NUBE EN UN E-COMMERCE



# VENTAJAS Y DESVENTAJAS

## Cliente-Servidor

### Ventajas

- Control total de hardware y software
- Seguridad bajo administración propia
- Personalización completa
- Rendimiento dedicado

### Desventajas

- Alta inversión inicial
- Escalabilidad limitada
- Mantenimiento complejo
- Riesgo ante fallos físicos

## Nube

### Ventajas

- Escalabilidad inmediata
- Menor inversión inicial (pago por uso)
- Alta disponibilidad y redundancia
- Acceso global desde cualquier lugar
- Actualizaciones automáticas

### Desventajas

- Dependencia del proveedor
- Costos variables a largo plazo
- Latencia en algunos casos
- Riesgos de privacidad y normativas

# COMPARACION CLIENTE-SERVIDOR FRENTE A LA NUBE

ASPECTO	CLIENTE - SERVIDOR	NUBE
COSTO INICIAL	ALTO	BAJO(POR SU USO)
ESCALABILIDAD	LIMITADA	ALTA Y RAPIDA
MANTENIMIENTO	A CARGO DE LA EMPRESA	A CARGO DEL PROVEEDOR
DISPONIBILIDAD	DEPENDE DE LA INFRESTRUCTURA	ALTA REDUNDANCIA GARANTIZADA
SEGURIDAD	LOCAL(CONTROL DIRECTO)	DELEGADA(RIESGOS TERCEROS)
ACCESIBILIDAD	LIMITADA A RED INTERNA VPN	GLOBAL DESDE INTERNET

# ARQUITECTURA DE BASE DE DATOS PARA UN HOSPITAL REGIONAL

## DETERMINACIÓN DE LA ARQUITECTURA



Un hospital regional es un entorno crítico que maneja información sensible (historias clínicas, resultados de laboratorio, datos de facturación) y requiere alta disponibilidad y seguridad. La arquitectura de base de datos debe reflejar estas prioridades, considerando el contexto organizacional de un hospital (atención 24/7, múltiples departamentos, gran volumen de pacientes y usuarios concurrentes).

# CRITERIOS CLAVE PARA SELECCIONAR LA ARQUITECTURA

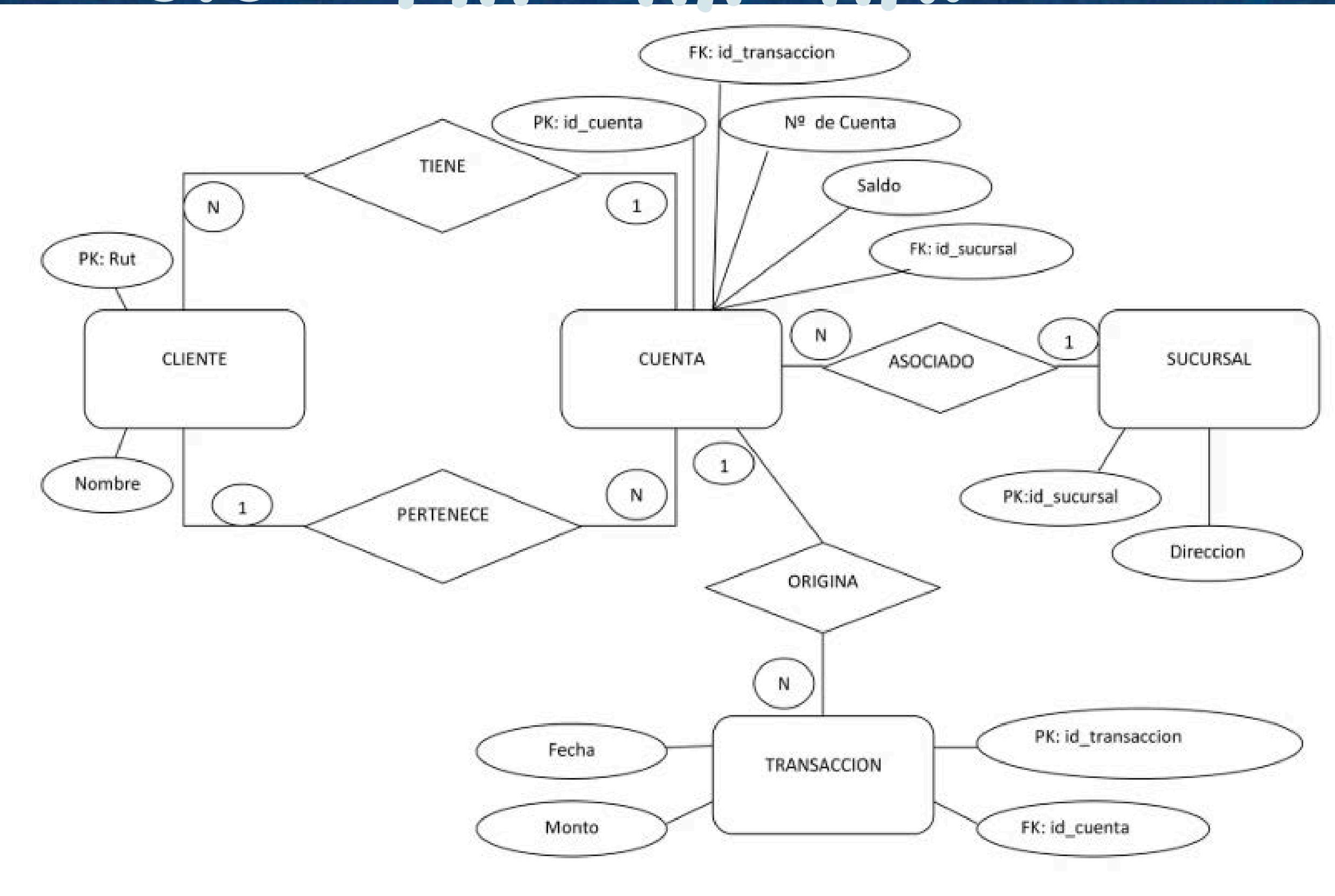
## 1. Disponibilidad y Tolerancia a Fallos

- Requisito: Es el criterio más crítico, ya que un fallo podría poner en riesgo la vida de los pacientes (cirugías, monitoreo de UCI, urgencias).
- Recomendación: Se necesita una arquitectura que garantice redundancia y replicación de datos (como se menciona en el texto).
  - Opción: Un sistema de bases de datos de alta disponibilidad (HA) con replicación sincrónica o asíncrona entre servidores locales (en on-premise o local) o en una nube privada/híbrida. Esto permite la commutación por error (failover) inmediata si el servidor principal falla.

## 4. Costo y Recursos Disponibles

- Requisito: Depende del presupuesto. Las soluciones on-premise son costosas en inversión inicial de hardware y personal, mientras que la nube tiene costos operativos variables.
- Recomendación: Un hospital regional típicamente tiene un presupuesto fijo.
  - Opción: Una inversión en un sistema local robusto y escalable podría ser preferible si ya se cuenta con la infraestructura y el personal especializado. Si el know-how de TI es limitado, una solución cloud (nube) podría ser más eficiente a largo plazo, externalizando la gestión del hardware y la escalabilidad.

# ENTIDAD RELACION DE UN SISTEMA BANCARIO



ADECIDIR QUÉ TABLAS  
CONVIENE DESNORMALIZAR  
EN UN SISTEMA DE REPORTES

# ANÁLISIS DE DESNORMALIZACIÓN PARA UN SISTEMA DE REPORTES

La desnormalización es una técnica de optimización del diseño que consiste en reintroducir intencionalmente cierta redundancia de datos en las tablas de la base de datos. Se hace para mejorar el rendimiento de las consultas, especialmente en sistemas donde la lectura de datos (consultas y reportes) es mucho más frecuente que la escritura/actualización.

Un sistema de reportes (o Data Warehouse) es el candidato ideal para la desnormalización, ya que su objetivo principal es consolidar y analizar grandes volúmenes de información de manera rápida.

# TABLAS A DESNORMALIZAR

tabla original (normalizada)	informacion de reporte requerido	propuesta de desnormalatizacion	justificacion
VENTAS	Reporte de "Ventas por Nombre de Cliente" y "Ventas por Nombre de Producto"	Agregar el <b>Nombre_Cliente</b> y el <b>Nombre_Producto</b> directamente a la tabla <b>VENTAS</b> .	La consulta original requeriría dos JOINs (VENTAS → CLIENTES y VENTAS → PRODUCTOS). Al agregar los nombres, se elimina la necesidad de los JOINs, <b>acelerando drásticamente los reportes</b> que agrupan o filtran por estos campos.
PRODUCTOS	Reporte de "Existencias por Categoría de Producto".	Agregar la <b>Descripción_Categoría</b> a la tabla <b>PRODUCTOS</b> . (Si la Categoría está en una tabla separada).	Se evita el JOIN entre la tabla <b>PRODUCTOS</b> y la tabla <b>CATEGORIAS</b> . Esto simplifica las consultas de inventario y evita que el reporte falle si la tabla de categorías no está disponible.
DETALLE_PEDIDO	Reporte de "Ganancia Bruta por Pedido" (que requiere el costo de los productos en el momento de la venta).	Agregar el <b>Costo_Unitario_al_Vender</b> y el <b>Precio_Final_Unitario</b> a la tabla <b>DETALLE_PEDIDO</b> .	Motivo histórico y de integridad: Estos valores cambian con el tiempo. Desnormalizar y almacenar los valores del momento de la transacción asegura que los reportes históricos sean precisos, independientemente de los precios o costos actuales.  Exportar a Hojas de cálculo

# PRINCIPIO CLAVE DE DECISIÓN

La decisión de desnormalizar siempre implica un balance entre rendimiento y mantenimiento:

1. Tablas de Hechos (Facts): Son las tablas centrales que registran los eventos (ej. VENTAS, TRANSACCIONES). Estas son las mejores candidatas para desnormalizar, incrustando los atributos descriptivos de las Tablas de Dimensión (ej. CLIENTE, PRODUCTO).
2. Riesgo de Mantenimiento: La desventaja de desnormalizar es que, si se actualiza el nombre de un cliente en la tabla CLIENTES, también se debe actualizar en todos los registros de la tabla desnormalizada VENTAS (lo que se conoce como anomalía de actualización).

En un sistema de reportes, este riesgo se gestiona asumiendo que:

- Las actualizaciones de datos en las tablas de hechos son raras o controladas.
- El beneficio de acelerar cientos o miles de reportes diarios supera el costo ocasional de una actualización más compleja.