Arquitectura de Datos – Sistema de Gestión Médica

1. Introducción

El sistema implementa una arquitectura híbrida compuesta por tres motores de bases de datos:

- SQL (PostgreSQL) para el núcleo transaccional.
- MongoDB para documentos flexibles y no estructurados.
- Redis para operaciones rápidas en memoria como sesiones, colas y contadores.

Esta arquitectura permite soportar información estructurada, semiestructurada y procesos en tiempo real dentro del mismo sistema, optimizando rendimiento, escalabilidad y flexibilidad.

2. Componentes del Sistema

2.1 Base de datos SQL – PostgreSQL

Se utiliza para la información crítica, estructurada y relacional del sistema:

- paciente
- medico
- cita
- factura
- servicio
- factura_detalle

Justificación del uso de SQL

- Integridad referencial mediante PK y FK.
- Modelo relacional estable.

- Consultas transaccionales complejas (JOIN, GROUP BY, CTE).
- Ideal para información que no cambia de estructura.

2.2 Base de datos documental – MongoDB

Diseñada para información semiestructurada y que crece de forma flexible.

Colecciones implementadas:

1. pacientes_perfil
    - Direcciones, alergias, historiales.
    - Datos variables que no deben rigidizar el modelo SQL.
2. notas_medicas
    - Notas extensas, adjuntos, tipos de consulta.
    - Estructuras anidadas y campos opcionales.
3. inventario_medicamentos
    - Datos con proveedores embebidos.
    - Campos cambiantes según categoría de medicamento.

Justificación del uso de MongoDB

- Permite almacenar documentos con estructuras flexibles.
- Evita múltiples tablas auxiliares.
- Maneja listas y objetos embebidos sin JOINs.
- Ideal para documentos médicos, notas y datos variables.

2.3 Base de datos en memoria – Redis

Redis soporta los procesos que requieren máxima velocidad, como:

- sesiones de usuarios
- contadores automáticos
- colas de atención
- sets de médicos disponibles
- rankings (ZSET)

Estructuras utilizadas:

- STRING
- LIST
- HASH
- SET
- ZSET

Justificación del uso de Redis

- Respuestas en microsegundos.
- Manejo de estados temporales.
- TTL para expiración automática.
- Ideal para colas y métricas en tiempo real.

3. Integración entre SQL, MongoDB y Redis

Flujo integrado típico:

1. El paciente agenda una cita (SQL)
    - Se registra paciente + médico + fecha.
2. El médico añade una nota médica (MongoDB)

    - Se guarda un documento con detalles extensos.
    - Se hace referencia cruzada usando id_cita_sql.
3. Se actualizan contadores y colas en Redis
    - Se incrementa el contador de citas del día (STRING).
    - Se agrega el paciente a la cola de atención (LIST).
    - Se marca el médico como disponible u ocupado (SET).

Beneficios principales

- SQL garantiza integridad.
- MongoDB provee flexibilidad documental.
- Redis aporta velocidad y manejo de estados temporales.
- La combinación aumenta la eficiencia y escalabilidad.

4. Riesgos de la Arquitectura

- El triple motor requiere mayor mantenimiento.
- Consistencia eventual entre SQL y MongoDB.
- Redis requiere políticas de persistencia si se manejan datos sensibles.

5. Conclusión

La arquitectura híbrida integra tres tecnologías complementarias para soportar un sistema médico robusto, escalable y orientado tanto a transacciones como a documentos y operaciones ultrarrápidas.