

72.41 - Base de Datos II

Trabajo Práctico Obligatorio:



Grupo G5:

Ben Deyheralde (63559)

Lucas Gonzalez Rouco (63366)

Matias Ignacio Mutz (63590)

Tobias Ves Losada (63342)

Profesores:

Rodriguez Babino, Cecilia

Rodriguez, Guillermo

Índice

Índice	2
Decisiones de Diseño	3

Decisiones de Diseño

1. MongoDB: Una elección ideal para datos estructurados y semi-estructurados

MongoDB es una base de datos NoSQL orientada a documentos que destaca por su flexibilidad y facilidad para manejar datos estructurados y semiestructurados. Su elección para este trabajo se justifica en varios puntos clave:

- Estructura flexible de datos: Los datos de clientes, teléfonos, facturas, detalle de las facturas y productos pueden representarse de manera natural usando documentos JSON, lo que facilita modelar relaciones jerárquicas, como un cliente con múltiples teléfonos o una factura con múltiples productos.
- Consultas rápidas y eficientes: MongoDB permite realizar consultas complejas con su lenguaje de agregación, ideal para responder preguntas como:
 - Obtener los datos de clientes con teléfonos (punto 1).
 - Calcular la cantidad total de facturas por cliente, incluso cuando no tienen ninguna (punto 6).
 - Mostrar el gasto total de cada cliente con IVA incluido (punto 10).
- Fácil escalabilidad y manejo de grandes volúmenes de datos: Si el volumen de datos de clientes o facturas crece considerablemente, MongoDB puede manejar esta carga sin sacrificar rendimiento.
- Actualización y manejo eficiente de datos: Con sus operaciones de escritura (insertar, modificar, eliminar), MongoDB es perfecto para implementar la creación, modificación y eliminación de clientes y productos (puntos 13 y 14).

2. Neo4j: Relación perfecta para datos conectados y análisis de redes

Neo4j es una base de datos de grafos diseñada para modelar relaciones complejas entre entidades. En este caso práctico, Neo4j sobresale por su capacidad de manejar datos altamente relacionales, como los clientes, sus facturas y los productos facturados. Justificamos su uso por los siguientes motivos:

- Modelado natural de relaciones: Los grafos permiten modelar directamente las conexiones entre clientes, facturas y productos. Esto resulta en consultas expresivas y directas para:
 - Identificar los clientes que tienen al menos una factura registrada (punto 4).
 - Listar los productos que han sido facturados al menos una vez (punto 8).
 - Mostrar las facturas que contienen productos de una marca específica, como "Ipsum" (punto 9).
- Lenguaje de consultas intuitivo (Cypher): Neo4j usa Cypher, un lenguaje de consultas declarativo diseñado específicamente para grafos, que facilita tareas como buscar clientes sin facturas (punto 5) o listar productos que aún no han sido facturados (punto 12).
- Rendimiento en consultas relacionales complejas: Consultas que involucran varias relaciones (como unir clientes con facturas y productos) se ejecutan de manera eficiente en Neo4j, mucho más que en bases de datos relacionales tradicionales.
- Visualización y análisis avanzado: Neo4j permite analizar patrones en los datos de manera visual, ideal para explorar conexiones como qué clientes gastan más en productos de marcas específicas o qué productos tienen baja facturación.