



Facultad de ingeniería

# Bases de datos documentales

Materia: Laboratorio 2

Alumnos: Gabriel Pérez, Daira Orlandini, Enoc García, Favio Ontón

Profesor: Juan Lopreiato

Ciudad Autónoma de Buenos Aires, Argentina – 2022

## Tabla de contenido

NoSQL .....	2
Bases De Datos Documentales .....	4
Ventajas .....	5
Desventajas: .....	5
Casos de uso .....	6
Administración de contenido .....	6
Catálogos .....	6
Perfiles de usuario.....	7
Big data en tiempo real .....	7
Diferencias con otras DDBB .....	7
Diferencias con bases de datos relacionales .....	7
Diferencias con otras bases NoSQL .....	8
Motores más utilizados .....	8
Escalabilidad.....	9
Bibliografía .....	10

## NoSQL

El término "NoSQL" se refiere a tipos de bases de datos no relacionales que almacenan datos en un formato distinto a las tablas relacionales. Sin embargo, las bases de datos

NoSQL se pueden consultar utilizando API de lenguaje natural, lenguajes de consulta estructurados declarativos y lenguajes de consulta mediante ejemplo, por lo que también se les llama bases de datos "no solo SQL".

Las bases de datos NoSQL se utilizan de forma generalizada en aplicaciones web en tiempo real y big data, ya que sus principales ventajas son los elevados niveles de escalabilidad y disponibilidad.

Las bases de datos NoSQL también son la opción preferida entre los desarrolladores, ya que propician por su propia naturaleza un desarrollo ágil dada su rápida adaptación a los requisitos en constante cambio. Las bases de datos NoSQL permiten almacenar los datos de forma más intuitiva y fácil de entender, o más cercanas a la forma en que las aplicaciones utilizan los datos: no necesitan realizar tantas transformaciones cuando almacenan o recuperan datos utilizando interfaces API de NoSQL. Además, las bases de datos NoSQL pueden aprovechar al máximo la nube para evitar por completo el tiempo de inactividad.

Si se comparan con las bases de datos relacionales, las bases de datos NoSQL son más escalables y ofrecen un mayor rendimiento; además, su modelo de datos aborda varias cuestiones que el modelo relacional pasa por alto tales como:

- Grandes volúmenes de datos estructurados, semiestructurados y no estructurados en constante cambio
- Sprints de desarrollo ágiles, iteración rápida de los esquemas y generación frecuente de código
- Programación orientada a objetos flexible y fácil de usar
- Arquitectura de escalado horizontal distribuida geográficamente, en lugar de una arquitectura monolítica

# Bases De Datos Documentales

Las bases de datos documentales, en inglés, documents stores, se utilizan para la administración de datos semiestructurados. Se trata de datos que no siguen una estructura fija, sino que llevan la estructura casi en sí misma. Sin embargo, con ayuda de marcadores dentro de estos datos, la información puede ordenarse. Debido a la falta de una estructura clara, estos datos no son adecuados para bases de datos relacionales porque su información no puede clasificarse en tablas.

Este modelo es un tipo de base de datos no relacional que ha sido diseñada para almacenar y consultar datos como documentos de tipo JSON. Las bases de datos de documentos facilitan a los desarrolladores el almacenamiento y la consulta de datos en una base de datos mediante el mismo formato de modelo de documentos que emplean en el código de aplicación. La naturaleza flexible, semi-estructurada y jerárquica de los documentos y las bases de datos de documentos permite que evolucionen según las necesidades de las aplicaciones.

Estas crean un par simple: un documento específico se asigna a una clave. En este documento, que puede tener formato XML, JSON o YAML, por ejemplo, se puede encontrar la información propiamente dicha. Como la base de datos no requiere un esquema específico, en un almacén de documentos pueden integrarse diferentes clases de documentos. Los cambios en los documentos no afectan a la base de datos.

Este es un ejemplo en formato JSON

```
{
  '_id' : 1,
  'artistName' : { 'Iron Maiden' },
  'albums' : [
    {
      'albumname' : 'The Book of Souls',
      'datereleased' : 2015,
      'genre' : 'Hard Rock'
    }, {
      'albumname' : 'Killers',
      'datereleased' : 1981,
      'genre' : 'Hard Rock'
    }, {
      'albumname' : 'Powerslave',
      'datereleased' : 1984,
      'genre' : 'Hard Rock'
    }, {
      'albumname' : 'Somewhere in Time',
      'datereleased' : 1986,
      'genre' : 'Hard Rock'
    }
  ]
}
```

# Ventajas

- **Creación más rápida de documentos y mantenimiento:** es muy simple crear un documento y aparte de esto, no requiere de mucho mantenimiento.
- **Sin esquema:** estos son muy buenos para retener datos existentes en volúmenes masivos porque no hay absolutamente ninguna restricción en el formato y la estructura del almacenamiento de los datos.
- **Formatos abiertos:** tiene un proceso de compilación muy simple que usa XML, JSON y sus otras formas.
- **Control de versiones incorporados:** tiene un control de versiones incorporado, lo que significa que a medida que los documentos aumentan de tamaño, existe la posibilidad que aumentan en complejidad. El control de versiones reduce los conflictos.
- **Facilidad de integrar datos nuevos:** mientras que, en una base de datos relacional, el nuevo punto de información se debe insertar en todos los registros de datos, en una base de datos orientada a documentos basta con integrar la información nueva en solo unos pocos registros. El contenido adicional se puede añadir a otros documentos, pero esto no es necesario.
- Tiene un modelo que facilita la tarea de actualización de datos.
- De igual forma, mantienen un gran formato para contener muchos datos e información.
- Da la posibilidad de consultar y mover datos semiestructurados, aunque no tenga estructura definida.
- Es de las mejores cuando se trata de guardar grandes cantidades de información.
- Cuentan con buenos motores de búsqueda y buenas propiedades de indexación.

# Desventajas:

- **Redundancia:** elementos relacionales no encajan en el modelo documental, lo que implica que los datos se insertan de manera redundante en los documentos al no poder referenciar otro documento para extraer la información. Hacer referencias entre documentos puede hacer crecer de manera exponencial el volumen de datos, por lo que es preferible en esos casos usar modelos relacionales.
- **Atomicidad débil:** carece de soporte para transacciones ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad) de múltiples documentos. Un cambio en el modelo de datos del documento que involucre dos colecciones requerirá que ejecutemos dos consultas separadas, es decir, una para cada colección. Aquí es donde rompe los requisitos de atomicidad.

- **Limitaciones de la verificación de consistencia:** se pueden buscar las colecciones y los documentos que no están conectados a una colección de autor, pero hacerlo podría crear un problema en el rendimiento de la base de datos.
- **Seguridad:** hoy en día, muchas aplicaciones web carecen de seguridad, lo que a su vez da como resultado la fuga de datos confidenciales. Entonces se convierte en un punto de preocupación, uno debe prestar atención a las vulnerabilidades de las aplicaciones web.

## Casos de uso

### Administración de contenido

Para administrar eficazmente el contenido, debe poder recopilar y agrupar contenido de una variedad de orígenes y enviárselo al cliente. Debido a su esquema flexible, las bases de datos documentales son perfectas para recopilar y almacenar cualquier tipo de datos. Puede utilizarlas para crear e incorporar nuevos tipos de contenido, incluido el contenido generado por el usuario, como imágenes, comentarios, y vídeos

Una base de datos de documentos es una excelente opción para aplicaciones de administración de contenido, como blogs y plataformas de vídeo. Con una base de datos documental, cada entidad que rastrea la aplicación se puede almacenar como un único documento. La base de datos de documentos es más intuitiva para que un desarrollador actualiza una aplicación a medida que evolucionan los requisitos. Además, si el modelo de datos necesita cambiar, solo se deben actualizar los documentos afectados. No se requiere actualización del esquema y no es necesario tiempo de inactividad de la base de datos para realizar los cambios.

### Catálogos

Las bases de datos de documentos son eficientes y efectivas para almacenar información de catálogo. Por ejemplo, en una aplicación de e-commerce, los diferentes productos generalmente tienen diferentes números de atributos. La administración de miles de atributos en bases de datos relacionales no es eficiente y afecta al rendimiento de lectura. Al utilizar una base de datos de documentos, los atributos de cada producto se pueden describir en un solo documento para que la administración sea fácil y la velocidad de lectura sea más rápida. Cambiar los atributos de un producto no afectará a otros.

## Perfiles de usuario

Como las bases de datos documentales tienen un esquema flexible, pueden almacenar documentos que tengan atributos y valores de datos diferentes. Las bases de datos documentales son una solución práctica para los perfiles online en los que diferentes usuarios proporcionan diferentes tipos de información. Mediante una base de datos documental, puede almacenar cada perfil de usuario de forma eficaz almacenando solo los atributos que son específicos de cada usuario.

Suponga que un usuario decide añadir o eliminar la información de su perfil. En este caso, su documento podría reemplazarse fácilmente por una versión actualizada que contuviera los atributos y datos recién añadidos u omitir todos los atributos y datos recién omitidos. Las bases de datos documentales administran fácilmente este nivel de detalle y fluidez.

## Big data en tiempo real

Históricamente, la capacidad de extraer información de datos operativos se ha visto obstaculizada por el hecho de que las bases de datos operativas y las bases de datos de análisis se mantenían en diferentes entornos: operativos y de negocio, respectivamente. Ser capaces de extraer información operativa en tiempo real es fundamental en un entorno empresarial altamente competitivo. Mediante el uso de bases de datos documentales, una empresa puede almacenar y administrar datos operativos de cualquier origen e incluir los datos de forma simultánea en el motor de BI elegido para su análisis. No es necesario tener dos entornos.

## Diferencias con otras DDBB

### Diferencias con bases de datos relacionales

Las bases de datos relacionales están diseñadas para aplicaciones de procesamiento de transacciones online (OLTP) altamente coherentes y transaccionales, y son buenas para el procesamiento analítico online (OLAP). Por el contrario, las bases de datos NoSQL están diseñadas para varios patrones de acceso a datos que incluyen aplicaciones de baja latencia. Las bases de datos de búsqueda NoSQL están diseñadas para hacer análisis sobre datos semiestructurados.

El modelo relacional normaliza los datos en tablas conformadas por filas y columnas. Un esquema define estrictamente las tablas, las filas, las columnas, los índices, las relaciones entre las tablas y otros elementos de las bases de datos. La base de datos impone la integridad referencial en las relaciones entre tablas. Mientras que las bases de datos

NoSQL proporcionan una variedad de modelos de datos, como clave-valor, documentos y gráficos, que están optimizados para el rendimiento y la escala.

## Diferencias con otras bases NoSQL

Las bases de datos documentales se caracterizan por admitir estructuras de datos complejas y flexibles contra otras bases de datos NoSQL como la de clave-valor ya que los documentos admiten muchos pares de clave-valor además de que pares clave-matriz o documentos anidados. Por otra parte, las bases de datos documentales al tener estructuras más complejas son menos performantes en algunos contextos en comparación con bases de datos NoSQL de clave-valor o de columnas. Además, las bases de datos documentales no están pensadas para crear relaciones entre los elementos de la base (en este caso, documentos), cosa que es posible en bases de datos de grafos.

## Motores más utilizados

Las bases de datos documentales son de gran importancia, en particular para el desarrollo de aplicaciones web. Debido a la demanda resultante del desarrollo web, hay ahora numerosos sistemas de gestión de bases de datos (SGBD) en el mercado. La siguiente selección enumera los más conocidos:

- BaseX: este proyecto open source utiliza Java y XML. BaseX viene con una interfaz de usuario.
- CouchDB: la Apache Software Foundation lanzó el software de código abierto CouchDB. El sistema de gestión de bases de datos está escrito en Erlang, utiliza JavaScript y se utiliza en aplicaciones de Ubuntu y Facebook, entre otras.
- Elasticsearch: este motor de búsqueda funciona con una base de datos documental. Para esto, emplea documentos JSON.
- eXist: el SGBD eXist, de código abierto, se ejecuta en una máquina virtual Java y, por lo tanto, puede utilizarse con independencia del sistema operativo. Utiliza principalmente documentos XML.
- MongoDB: MongoDB es la base de datos NoSQL más extendida del mundo. El software está escrito en C++ y emplea documentos similares a JSON.
- SimpleDB: con SimpleDB (escrito en Erlang), Amazon ha desarrollado su propio SGBD para los servicios en la nube de la compañía. El proveedor cobra una tarifa por su uso.
- DynamoDB. Azure Cosmos. RavenDB.



# Escalabilidad

Las bases de datos NoSQL normalmente se pueden particionar porque los patrones de acceso son escalables mediante el uso de arquitectura distribuida para aumentar el rendimiento que proporciona un rendimiento constante a una escala casi ilimitada.

## Bibliografía

- <https://www.oracle.com/ar/database/nosql/what-is-nosql/>
- <https://www.mongodb.com/es/nosql-explained>
- <https://aws.amazon.com/es/nosql/>
- <https://www.geeksforgeeks.org/document-databases-in-nosql/>
- <https://www.diegocalvo.es/caracteristicas-y-comparativa-de-las-bases-de-datos-nosql/>