

Tarea 4- Almacenamiento y Consultas de Datos en Big Data

**Participantes:
Leonardo Hernandez**

Grupo No. 202016911_45

**Tutor:
Frank Rodríguez Achury**

**Universidad Nacional Abierta y a Distancia – UNAD
Escuela Ciencias básicas, tecnología e ingeniería
Programa Ingeniería de Sistemas
Big Data
Colombia
2025**

INTRODUCCIÓN

En esta actividad se trabaja con el manejo de datos usando MongoDB, que es una base de datos NoSQL muy útil para manejar información grande y variada sin necesidad de estructuras rígidas como las bases relacionales. El objetivo principal es aprender como almacenar documentos, crear un catálogo funcional y realizar consultas sencillas y también consultas un poco más avanzadas como las de agregación.

El trabajo se desarrolló por pasos, empezando por el diseño del catálogo, después la creación de la base de datos en MongoDB Compass, la inserción de cien documentos variados y por último la realización de diferentes consultas que permiten analizar y extraer información importante del inventario. Todo esto sirve como base para entender cómo funcionan los sistemas de BigData de forma práctica.

OBJETIVOS

Objetivo general

Implementar una base de datos NoSQL en MongoDB para almacenar un catálogo de productos y realizar consultas básicas y de agregación que permitan analizar el inventario.

Objetivos específicos

- Crear una colección con mínimo 100 documentos sobre un catálogo realista.
- Aplicar operaciones CRUD usando MongoDB Compass.
- Realizar consultas con operadores lógicos y comparativos.
- Ejecutar consultas de agregación y analizar los resultados obtenidos.
- Comprender el uso de estructuras flexibles en bases de datos NoSQL.

FASE UNO – DISEÑO DEL CATÁLOGO

Para esta tarea se diseñó un catálogo de productos variado con categorías como ropa, tecnología, hogar, belleza y accesorios. Cada producto contiene información como nombre, categoría, precio, stock, colores, tallas opcionales y su fecha de registro.

Se escogió este caso porque es cercano a situaciones reales de negocios donde se manejan inventarios cambiantes y cantidades de productos que requieren consultas rápidas y flexibles sin necesidad de modelos rígidos.

El esquema definido es flexible:

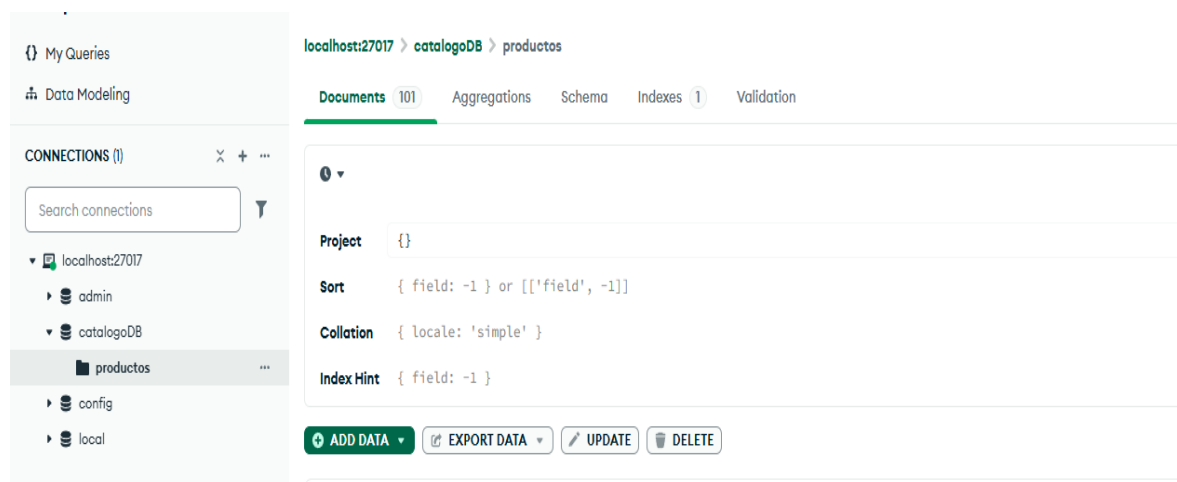
- nombre (string)
- categoria (string)
- precio (number)
- stock (number)
- colores (array)
- tallas (array)
- fechaRegistro (date)

MongoDB es adecuado porque permite almacenar documentos diferentes sin afectar la colección.

FASE DOS – IMPLEMENTACIÓN EN MONGODB

1. Creación de la base de datos y colección

Se creó la BD llamada catalogoDB y dentro la colección productos usando MongoDB Compass.



2. Inserción de 100 documentos

Se insertaron cien documentos con información realista y variada.

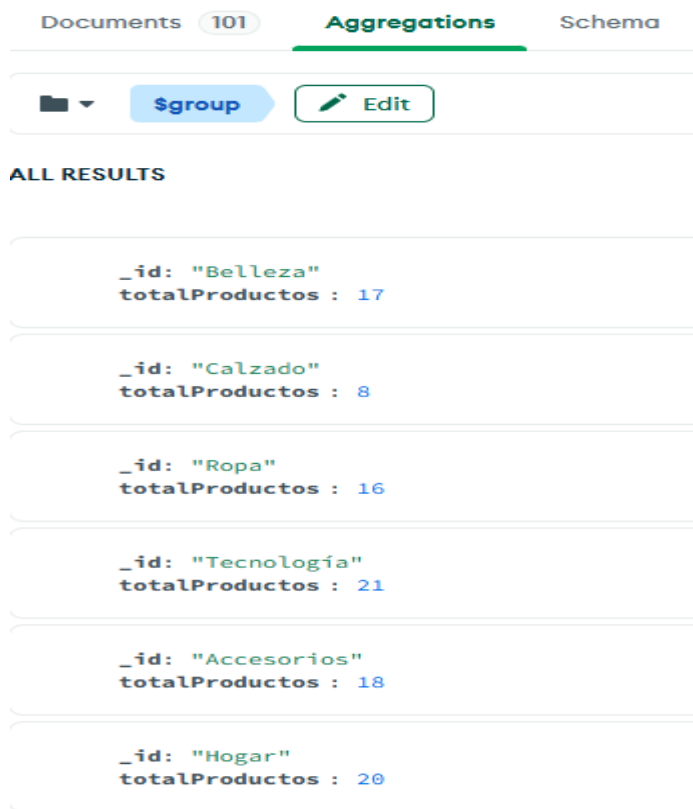
Los datos se cargaron por medio de insertMany en Compass.

Consultas de agregación

3.1 Conteo de productos por categoria

Se uso un \$group para agrupar por el campo categoria y contar cuantos productos hay en cada una.

Con esto se pudo ver que algunas categorias como Ropa tienen mas registros, mientras que otras cuentan con menos variedad. Esto ayuda a entender mejor como esta distribuido el inventario general.



3.2 Precio promedio por categoria

Mediante \$avg se calculó el valor promedio de los productos de cada categoria. Esto es util para conocer si hay categorias mas costosas que otras y comparar si los precios estan equilibrados. Por ejemplo, tecnologia suele tener precios mas altos, mientras que accesorios o belleza tienen valores mas bajos.

Documents 101 Aggregations Schema Indexes 1 Validation



\$group



Edit

ALL RESULTS

```
_id: "Hogar"
promedioPrecio : 104000
```

```
_id: "Calzado"
promedioPrecio : 132500
```

```
_id: "Ropa"
promedioPrecio : 71750
```

```
_id: "Tecnología"
promedioPrecio : 185000
```

```
_id: "Belleza"
promedioPrecio : 55941.17647058824
```

```
_id: "Accesorios"
promedioPrecio : 50555.55555555555
```

3.3 Stock total

Con \$sum se sumaron todas las cantidades del stock de los productos.

Este dato permite saber cuántas unidades totales hay en inventario, lo cual ayuda a revisar si el nivel de abastecimiento es suficiente o si se necesitan reposiciones.

Documents 101 Aggregations Schema Indexes



\$group



Edit

ALL RESULTS



```
_id: null
stockTotal : 2045
```

Consultas básicas (CRUD)



4.1 Insertar documento

Insert Document






To collection catalogoDB.productos

VIEW  

```
1 {
2   "nombre": "Producto ejemplo",
3   "categoria": "Ropa",
4   "precio": 45000,
5   "stock": 10,
6   "fechaRegistro": { "$date": "2025-02-01T00:00:00Z" }
7 }
8
```

4.2 Consultar

Documents 101	Aggregations	Schema	Indexes 1	Validation
<div><div> {}</div><div>Generate query  <input type="button" value="Explain"/> <input type="button" value="Reset"/> <input type="button" value="Find"/></div></div>				
<div><div><input type="button" value="ADD DATA"/> <input type="button" value="EXPORT DATA"/> <input type="button" value="UPDATE"/> <input type="button" value="DELETE"/></div><div>100 1 - 100 of 100   </div></div>				
<pre>_id: ObjectId('6921289da596c24f64d69648') nombre: "Camiseta oversized" categoria: "Ropa" precio: 45000 stock: 20 colores: Array (2) tallas: Array (2) fechaRegistro: 2025-01-01T00:00:00.000+00:00</pre>				
<pre>_id: ObjectId('69212a56a596c24f64d6964f') nombre: "Camiseta básica" categoria: "Ropa" precio: 35000 stock: 50 tallas: Array (3) colores: Array (2) fechaRegistro: 2025-01-05T00:00:00.000+00:00</pre>				
<pre>_id: ObjectId('69212a56a596c24f64d69650') nombre: "Pantalón de mezclilla" categoria: "Ropa" precio: 85000 stock: 18 tallas: Array (3) colores: Array (1) fechaRegistro: 2025-01-08T00:00:00.000+00:00</pre>				
<pre>_id: ObjectId('69212a56a596c24f64d69651') nombre: "Sudadera con capucha" categoria: "Ropa" precio: 120000 stock: 12 tallas: Array (2) colores: Array (2) fechaRegistro: 2025-01-10T00:00:00.000+00:00</pre>				

4.3 Actualizar

Update 1 document

catalogoDB.productos

Filter ⓘ

{ nombre: 'Camiseta oversized' }

Update

[Learn more about Update syntax](#)

```
1 {
2   $set: {
3     "stock": 50
4   },
5 }
```

★ Save Cancel Update 1 document

4.4 Eliminar

Delete 1 document

catalogoDB.productos

Filter ⓘ

{ nombre: 'Producto ejemplo' }

[Export](#)

Preview (sample of 1 document)

```
_id: ObjectId('69213501a596c24f64d696ba')
nombre: "Producto ejemplo"
categoria: "Ropa"
precio: 45000
stock: 10
fechaRegistro: 2025-02-01T00:00:00.000+00:00
```

Cancel Delete 1 document

{ "nombre": "Producto ejemplo", }

Generate query ↗ Expla

ADD DATA

EXPORT DATA

UPDATE

DELETE



100 0 - 0



No results

Try modifying your query to get results.

Consultas con operadores

Precio mayor a 100000

```
{ "precio": { "$gt": 100000 } }
```

Solo productos de Tecnología

```
{ "categoria": "Tecnologia" }
```

Stock entre 10 y 20

```
{ "stock": { "$gte": 10, "$lte": 20 } }
```

Buscar nombres que contengan texto

```
{ "nombre": { "$regex": "cam", "$options": "i" } }
```

Cada consulta permite analizar el inventario de una manera diferente, aplicando filtros específicos según las necesidades de consulta de un negocio real.

CONCLUSIONES

El uso de MongoDB facilita bastante el manejo de grandes cantidades de información, especialmente cuando los datos no siguen un formato rígido. Durante esta actividad se pudo observar como una base de datos NoSQL se adapta muy bien a un catalogo de productos variado, donde cada registro puede tener campos diferentes.

Tambien se comprendio como funcionan las consultas CRUD y las consultas con operadores, que ayudan a buscar informacion de forma rapida y precisa. Ademas, las consultas de agregación dieron informacion importante para analizar el inventario, como las categorias más usadas, los precios promedios y el stock total.

En general la actividad me permitio entender mejor cómo aplicar conceptos de Big Data con herramientas actuales, y como realizar analisis basicos directamente desde la base de datos.