

**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 optionales 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.

The screenshot shows the MongoDB Compass interface. On the left, the sidebar displays 'My Queries' and 'Data Modeling'. Under 'CONNECTIONS (1)', it lists 'localhost:27017' with sub-options for 'admin', 'catalogoDB' (which is expanded to show 'productos'), 'config', and 'local'. The main panel shows the 'localhost:27017 > catalogoDB > productos' path. At the top, there are tabs for 'Documents' (101), 'Aggregations', 'Schema', 'Indexes' (1), and 'Validation'. Below the tabs, there are fields for 'Project' (empty), 'Sort' (set to '{ field: -1 } or [['field', -1]]), 'Collation' (set to { locale: 'simple' }), and 'Index Hint' (set to { field: -1 }). At the bottom of the main panel are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'.

### 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 categoría

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 categorías como Ropa tienen mas registros, mientras que otras cuentan con menos variedad. Esto ayuda a entender mejor como esta distribuido el inventario general.

Documents 101 Aggregations Schema

ALL RESULTS

`_id: "Belleza"  
totalProductos : 17`

`_id: "Calzado"  
totalProductos : 8`

`_id: "Ropa"  
totalProductos : 16`

`_id: "Tecnología"  
totalProductos : 21`

`_id: "Accesorios"  
totalProductos : 18`

`_id: "Hogar"  
totalProductos : 20`

### 3.2 Precio promedio por categoría

Mediante `$avg` se calculó el valor promedio de los productos de cada categoría. Esto es útil para conocer si hay categorías más costosas que otras y comparar si los precios están equilibrados. Por ejemplo, tecnología suele tener precios más altos, mientras que accesorios o belleza tienen valores más bajos.

Documents 101 Aggregations Schema Indexes 1 Validation

\$group

#### 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 : 50555.555555555555
```

### 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

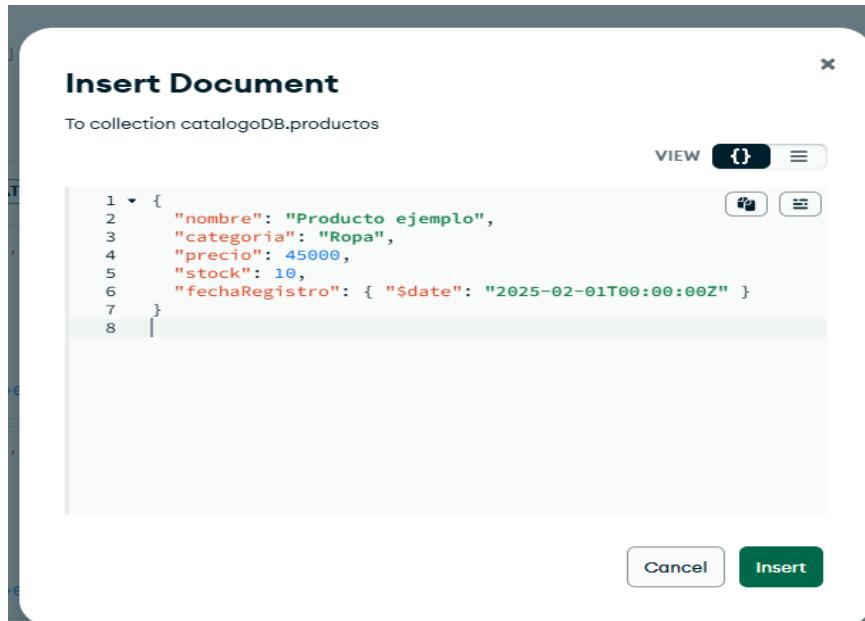
\$group

#### ALL RESULTS

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

## Consultas básicas (CRUD)

### 4.1 Insertar documento



### 4.2 Consultar

	_id	nombre	categoria	precio	stock	fechaRegistro
1	ObjectId('69212a56a596c24f64d6964f')	"Camiseta oversize"	"Ropa"	45000	20	2025-01-01T00:00:00.000+00:00
2	ObjectId('69212a56a596c24f64d6964f')	"Camiseta básica"	"Ropa"	35000	50	2025-01-05T00:00:00.000+00:00
3	ObjectId('69212a56a596c24f64d69650')	"Pantalón de mezclilla"	"Ropa"	85000	18	2025-01-08T00:00:00.000+00:00
4	ObjectId('69212a56a596c24f64d69651')	"Sudadera con capucha"	"Ropa"	120000	12	2025-01-10T00:00:00.000+00:00

## 4.3 Actualizar

**Update 1 document**

catalogoDB.productos

**Filter** ⓘ

```
{ nombre: 'Camiseta oversize' }
```

**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

0 ▾ { "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 segun 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 catálogo de productos variado, donde cada registro puede tener campos diferentes.

También se comprendió cómo funcionan las consultas CRUD y las consultas con operadores, que ayudan a buscar información de forma rápida y precisa. Además, las consultas de agregación dieron información importante para analizar el inventario, como las categorías más usadas, los precios promedios y el stock total.

En general la actividad me permitió entender mejor cómo aplicar conceptos de Big Data con herramientas actuales, y cómo realizar análisis básicos directamente desde la base de datos.