

POLITECNICO(PIO)

BASE DE DATOS 2

**PROYECTO FINAL CREANDO COLECCIONES Y
DOCUMENTOS EN MONGO DB**

MIGUEL ANGEL HURTADO ANGULO

12/10/2024

TABLA DE CONTENIDO

1.	Introducción	3
2.	Objetivos.....	4
3.	Estructura de la base de datos	6
4.	Conclusión.....	10

1. Introducción

MongoDB es una base de datos NoSQL orientada a documentos que ofrece una alternativa flexible y escalable a las bases de datos relacionales tradicionales. Diseñada para gestionar grandes volúmenes de datos no estructurados y semi-estructurados, MongoDB almacena la información en un formato llamado BSON (Binary JSON), lo que permite representaciones jerárquicas y complejas de datos dentro de un solo documento. A diferencia de las bases de datos relacionales, donde los datos se dividen en tablas y se relacionan mediante claves, MongoDB organiza la información en colecciones y documentos.

2. Objetivos

2.1 Objetivo general: Desarrollar una aplicación de comercio electrónico (e-commerce) funcional que permita la gestión de usuarios, productos, categorías, carritos de compra y órdenes mediante el uso de MongoDB como base de datos principal.

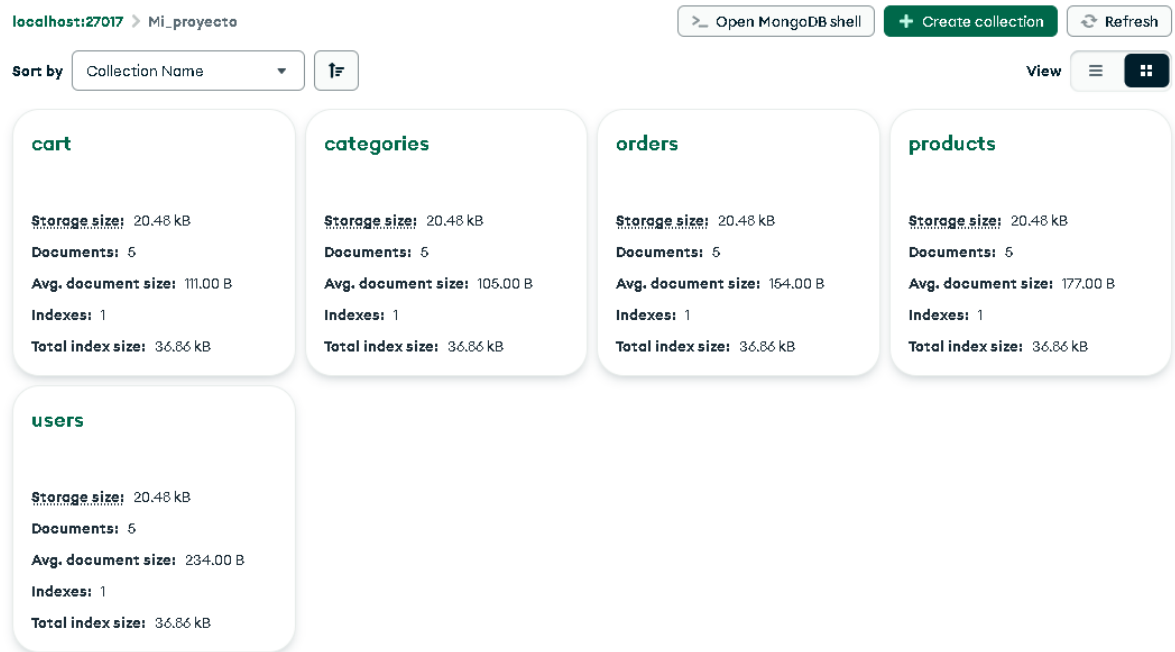
2.2 Objetivos específicos:

- **Diseñar una base de datos en MongoDB** que contenga colecciones para usuarios, productos, categorías, carritos y órdenes, asegurando la correcta relación y organización de los datos.
- **Implementar un backend funcional con Node.js y Express** para manejar las operaciones CRUD de la base de datos y proporcionar una API REST que conecte con el frontend.
- **Desarrollar una interfaz de usuario en React** que permita interactuar con el sistema de manera intuitiva, mostrando productos, gestionando el carrito y confirmando órdenes de compra.
- **Garantizar la integridad y seguridad de los datos** mediante la implementación de validaciones y políticas de control en el backend.
- **Configurar la conexión entre el backend y MongoDB** para realizar operaciones eficientes sobre los datos almacenados.

- **Probar y documentar el sistema desarrollado** para asegurar su correcto funcionamiento y facilitar su uso, implementación y mantenimiento en el futuro.

3. Estructura de la base de datos

Lo primero sería iniciar crear una base de datos, en nuestro caso se llamará: **Mi_proyecto**. Que contendrá las siguientes colecciones:



Como se puede observar, en cada colección hay 5 documentos (Que podrían ser más) que nos servirán para trabajar en la base de datos.

Colección de usuarios (users):

- Campos: nombre, correo, dirección, teléfono, rol (cliente o administrador), fecha de registro.
- Crear al menos 5 documentos representando diferentes tipos de usuarios, algunos con el rol de administrador y otros como clientes.

localhost:27017 > Mi_proyecto > users

Documents 5 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

EXPLAIN RESET FIND OPTIONS

ADD DATA EXPORT DATA UPDATE DELETE 25 1 - 5 of 5

	_id ObjectId	nombre String	correo String	direccion String	telefono Int64	
1	ObjectId('675bad9f8a5eca9...')	"Juan Perez"	"juan.perez@gmail.com"	"calle 13 "	3124567890	
2	ObjectId('675bafdd9a5eca9...')	"Maria Lopez"	"Maria.lopez@gmail.com"	"avenida siempre viva 72 "	3136565890	
3	ObjectId('675bb46c9a5eca9...')	"Miguel Hurtado"	"Miguel1813@gmail.com"	"carrera 81 "	3127776360	
4	ObjectId('675bb56a9a5eca9...')	"Salma Yaritza Hurtado"	"salma2004@gmail.com"	"carrera 81 Matias Mulumb...	3057237300	
5	ObjectId('675bb6b98a5eca9...')	"Ana culna"	"Anitax@gmail.com"	"Vereda la chilindrina "	3209087642	

Colección de productos (products):

- Campos: nombre, descripción, precio, imagen, categoría, stock.
- Crear al menos 5 documentos de productos variados, con diferentes categorías y cantidades en stock.

localhost:27017 > Mi_proyecto > products

Documents 5 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

EXPLAIN RESET FIND OPTIONS

ADD DATA EXPORT DATA UPDATE DELETE 25 1 - 5 of 5

	_id ObjectId	nombre String	descripcion String	precio Int64	imagen String	
1	ObjectId('675bb9188a5eca9...')	"Nintendo switch"	"Pantalla OLED 64GB almac...	1700000	"https://www..."	
2	ObjectId('675bec4d8a5eca9...')	"Samsung galaxy A15"	"Version 56, 8GB RAM con ...	1400000	"https://tienc..."	
3	ObjectId('675beca78a5eca9...')	"Armario practinac"	"Productos para el hogar ...	3400000	"https://http..."	
4	ObjectId('675bed078a5eca9...')	"laptop lenovo"	"Ryzen 5500u + 16GB RAM"	1600000	"https://www..."	
5	ObjectId('675bed6d8a5eca9...')	"TV panasonic"	"36 pulgadas, 4k, pantall...	1800000	"https://www.r..."	

Colección de categorías (categories):

- Campos: nombre de la categoría, descripción.
- Crear al menos 5 categorías que clasifiquen distintos tipos de productos en el sistema.

localhost:27017 > Mi_proyecto > categories Open MongoDB shell

Documents 5 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#) Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE 25 1 - 5 of 5

categories			
	_id ObjectId	nombre_categoria String	descripcion String
1	ObjectId('675bf4238a5eca8...')	"Videojuegos"	"Consolas y accesorios de..."
2	ObjectId('675bf56f8a5eca8...')	"Hogar"	"Productos para el hogar ..."
3	ObjectId('675bf5b98a5eca8...')	"Celulares"	"Celulares de diferentes ..."
4	ObjectId('675bf5fa8a5eca8...')	"Portatiles"	"Portatiles basico y para..."
5	ObjectId('675bf6188a5eca8...')	"Televisores"	"Televisores de alta gama"

Colección de carrito de compras (cart):

- Campos: ID del usuario, productos (con ID del producto y cantidad).
- Crear al menos 5 documentos de carrito con productos añadidos.

localhost:27017 > Mi_proyecto > cart Open MongoDB shell

Documents 5 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#) Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE 25 1 - 5 of 5

cart					
	_id ObjectId	id_usuario ObjectId	producto String	id_producto ObjectId	id_cantidad Int
1	ObjectId('675e0bd362c8a2e...')	ObjectId('675e390662c8a2e...')	"Nintendo switch"	ObjectId('675e390e62c8a2e...')	1
2	ObjectId('675e0d4562c8a2e...')	ObjectId('675e391562c8a2e...')	"Samsung galaxy"	ObjectId('675e391862c8a2e...')	1
3	ObjectId('675e0eaf62c8a2e...')	ObjectId('675e39be62c8a2e...')	"Armario pratimac"	ObjectId('675e39d762c8a2e...')	1
4	ObjectId('675e0f3562c8a2e...')	ObjectId('675e39f362c8a2e...')	"Laptop Lenovo"	ObjectId('675e39fa62c8a2e...')	1
5	ObjectId('675e0f6662c8a2e...')	ObjectId('675e392362c8a2e...')	"TV panasonic"	ObjectId('675e392862c8a2e...')	1


Colección de orden (orders):

- Campos: ID del usuario, productos (ID del producto, cantidad), estado de la orden (pendiente, pagado, enviado), total.
- Crear al menos 5 documentos de órdenes con diferentes combinaciones de productos y estados de compra.

categorieslocalhost:27017mongodb: localhost:27017orderscartMi_proyectoproductsusers+

localhost:27017Mi_proyectoordersOpen MongoDB shell

Documents5AggregationsSchemaIndexes1Validation





















Type a query: { field: 'value' } or [Generate query](#) 

ExplainResetFindOptions

ADD DATAEXPORT DATAUPDATEDELETE

251 - 5 of 5

orders

	_id ObjectId	id_usuario ObjectId	producto String	id_producto ObjectId	cantidad_prodi	
1	ObjectId('675e106862c8a2e...	ObjectId('675e390662c8a2e...	"Nintendo Switch"	ObjectId('675e390e62c8a2e...	1	   
2	ObjectId('675e122462c8a2e...	ObjectId('675e391562c8a2e...	"Samsung galaxy A15"	ObjectId('675e391862c8a2e...	1	   
3	ObjectId('675e129762c8a2e...	ObjectId('675e38be62c8a2e...	"Armario practimac"	ObjectId('675e38d762c8a2e...	1	   
4	ObjectId('675e130062c8a2e...	ObjectId('675e38f362c8a2e...	"TV panasonic"	ObjectId('675e38fa62c8a2e...	1	   
5	ObjectId('675e150862c8a2e...	ObjectId('675e392362c8a2e...	"Laptop Lenovo"	ObjectId('675e392862c8a2e...	1	   

4. Conclusión

Este proyecto sobre la gestión de relaciones en MongoDB proporciona una comprensión fundamental sobre cómo manejar los datos en una base de datos NoSQL, enfocándose en cómo establecer y verificar relaciones entre documentos. A través del uso de **referencias** y **documentos incrustados**, se logró representar la conexión entre diferentes tipos de datos (como usuarios y pedidos) de manera flexible, lo que refleja la naturaleza flexible de MongoDB.