

Ejercicio de Aplicación #5 – MongoDB (Ecommerce)

Caso de uso: Comercio Electrónico

Una empresa de ecommerce necesita gestionar información sobre sus clientes, órdenes de compra y productos adquiridos. Se requiere una estructura flexible que permita almacenar y recuperar datos de manera eficiente utilizando MongoDB.

1. Diseño de la estructura

Propuesta 1: Modelo embebido

```
{
  "_id": "cliente1",
  "nombre": "Juan Pérez",
  "email": "juan@mail.com",
  "ordenes": [
    {
      "id_orden": "ORD001",
      "fecha": "2025-07-01",
      "productos": [
        { "nombre": "Laptop", "precio": 1200, "cantidad": 1 },
        { "nombre": "Mouse", "precio": 20, "cantidad": 2 }
      ]
    }
  ]
}
```

Propuesta 2: Modelo con colecciones separadas

Colección: clientes

```
{
  "_id": "cliente1",
  "nombre": "Juan Pérez",
  "email": "juan@mail.com"
}
```

Colección: ordenes

```
{
  "_id": "ORD001",
  "cliente_id": "cliente1",
  "fecha": "2025-07-01",
  "productos": [
    { "nombre": "Laptop", "precio": 1200, "cantidad": 1 },

```

```
    { "nombre": "Mouse", "precio": 20, "cantidad": 2 }  
  ]  
}
```

2. Inserción de datos de ejemplo

```
// clientes  
db.clientes.insertOne({  
  _id: "cliente1",  
  nombre: "Juan Pérez",  
  email: "juan@mail.com"  
})  
  
// ordenes  
db.ordenes.insertOne({  
  _id: "ORD001",  
  cliente_id: "cliente1",  
  fecha: ISODate("2025-07-01T00:00:00Z"),  
  productos: [  
    { nombre: "Laptop", precio: 1200, cantidad: 1 },  
    { nombre: "Mouse", precio: 20, cantidad: 2 }  
  ]  
})
```

3. Consultas

a) Todas las órdenes de un cliente específico:

```
db.ordenes.find({ cliente_id: "cliente1" })
```

b) Todos los productos de una orden específica:

```
db.ordenes.find({ _id: "ORD001" }, { productos: 1, _id: 0 })
```

c) Número total de órdenes por cliente:

```
db.ordenes.aggregate([  
  { $group: { _id: "$cliente_id", total_ordenes: { $sum: 1 } } }  
])
```

4. Comparación MongoDB vs Relacional

| Aspecto | MongoDB | Relacional |
|------------------------|---|-----------------------------|
| Esquema | Flexible | Fijo |
| Rendimiento lectura | Alto con modelo embebido | Bueno con JOINS optimizados |
| Escalabilidad | Horizontal | Limitada (vertical) |
| Agregaciones complejas | Posibles con aggregate | Nativas con SQL |
| Ideal para | Apps modernas, JSON, cambios frecuentes | Transacciones críticas |

Plus: Colección de productos

```
db.productos.insertMany([
  { producto_id: "P001", nombre: "Laptop", precio: 1200, stock: 10 },
  { producto_id: "P002", nombre: "Mouse", precio: 20, stock: 50 },
  { producto_id: "P003", nombre: "Teclado", precio: 40, stock: 30 }
])
```

```
// Consulta
db.productos.find()
```