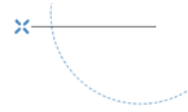


## Ejercicios de Aplicación – MongoDB (E-commerce y Consultas Avanzadas)



### Modelado de colecciones para una app de e-commerce

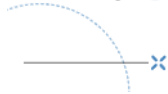
#### Contexto: 🧑🏻‍🤝‍🧑🏻

MongoDB permite almacenar información flexible y semiestructurada, ideal para representar productos, usuarios y órdenes en una tienda online. Esta actividad busca ejercitar el diseño de documentos optimizados.

#### Consigna: 📝

Diseña una estructura de colección para una tienda online que permita guardar información de productos, incluyendo variantes (talle, color), reviews y disponibilidad por sucursal.

**Tiempo** ⌚: 30 minutos



#### 🔗 Paso a paso:

1. En grupo, definan los campos que incluya cada documento de producto.
2. Estructuren el documento con objetos anidados y arrays cuando sea útil.
3. Justifiquen por qué usaron esa estructura (qué datos embebieron, qué dejaron planos, etc.).
4. Escriban un ejemplo en formato JSON.
5. Compartan y discutan su modelo con otro grupo.

### 1. Campos definidos

- producto\_id
- nombre
- categoría
- precio\_base
- variantes (array con objetos de color, talle, stock)
- reviews (array con usuario, comentario y calificación)
- disponibilidad (array por sucursal)

### 2. Estructura del documento (JSON)

```
{
  "producto_id": "P001",
  "nombre": "Zapatilla Running Pro",
  "categoria": "Calzado",
  "precio_base": 54990,
  "variantes": [
    { "color": "Rojo", "talle": 42, "stock": 10 },
    { "color": "Negro", "talle": 43, "stock": 5 }
  ],
  "reviews": [
    { "usuario": "ana123", "comentario": "Muy cómodas", "puntuacion": 5 },
```

```
{ "usuario": "luis456", "comentario": "Buen agarre", "puntuacion": 4 }
],
"disponibilidad": [
  { "sucursal": "Santiago", "stock_total": 8 },
  { "sucursal": "Antofagasta", "stock_total": 7 }
]
}
```

### 3. Justificación del diseño

- Se usaron arrays y objetos embebidos para representar múltiples variantes y reviews.
- Permite consultas eficientes sin necesidad de relaciones externas.
- Optimiza lectura y coherencia para datos jerárquicos.

## Consultas avanzadas con operadores MongoDB

### Contexto: 🧑🏫

MongoDB permite realizar consultas muy específicas con operadores como \$gt, \$in, \$regex, \$and y \$or. Esta actividad entrena el uso práctico de estas herramientas.

### Consigna: 📝

A partir de una colección de usuarios con datos de edad, ciudad, intereses y fecha de registro, redacta consultas que respondan preguntas del negocio usando operadores avanzados.

**Tiempo** ⌚: 30 minutos

### 🔗 Paso a paso:

1. El docente entregará un esquema de colección ejemplo.
2. En parejas, escriban consultas para responder:
  - ¿Qué usuarios son mayores de 30 y viven en Rosario?
  - ¿Qué usuarios se registraron después de 2023-01-01 y tienen "data" como interés?
  - ¿Cuántos usuarios tienen nombre que comienza con "A" o viven en Córdoba?
3. Utilicen los operadores correctos y proyección para limitar los campos.
4. Validar consultas entre grupos y revisar resultados esperados.

## Ejercicio 2: Consultas avanzadas con operadores MongoDB

### Esquema base del documento usuarios

```
{
  "nombre": "Ana López",
  "edad": 34,
  "ciudad": "Rosario",
  "intereses": ["data", "ciencia", "tecnología"],
  "fecha_registro": ISODate("2023-02-15T00:00:00Z")
}
```

### Consultas requeridas

1. Usuarios mayores de 30 y que viven en Rosario:

```
db.usuarios.find({
  $and: [
    { edad: { $gt: 30 } },
    { ciudad: "Rosario" }
  ]
})
```

2. Usuarios registrados después del 01/01/2023 y que tienen 'data' como interés:

```
db.usuarios.find({
  $and: [
    { fecha_registro: { $gt: ISODate("2023-01-01T00:00:00Z") } },
    { intereses: "data" }
  ]
})
```

3. Usuarios cuyo nombre comienza con 'A' o que viven en Córdoba:

```
db.usuarios.find({
  $or: [
    { nombre: { $regex: /^A/, $options: "i" } },
    { ciudad: "Córdoba" }
  ]
})
```