

DAW/DAM. UD 8. BASES DE DATOS NOSQL: MONGODB. ACTIVIDADES NO EVALUABLES PARTE 1 (SOLUCIONADO)

DAW/DAM. Bases de datos (BD)

UD 8. BASES DE DATOS NOSQL: MONGODB

Parte 1. DDL y DQL. Prácticas no evaluables (solucionado)

Abelardo Martínez y Pau Miñana

Basado y modificado de Sergio Badal (www.sergiobadal.com) y Raquel Torres.

Curso 2023-2024

Aspectos a tener en cuenta

Importante

Estas actividades son opcionales y no evaluables pero es recomendable hacerlas para un mejor aprendizaje de la asignatura.

Si buscas las soluciones por Internet o preguntas al oráculo de ChatGPT, te estarás engañando a ti mismo. Ten en cuenta que **ChatGPT no es infalible ni todopoderoso.**

Es una gran herramienta para agilizar el trabajo una vez se domina una materia, pero usarlo como atajo en el momento de adquirir habilidades y conocimientos básicos perjudica gravemente tu aprendizaje. Si lo utilizas para obtener soluciones o asesoramiento respecto a las tuyas, revisa cuidadosamente las soluciones propuestas igualmente. Intenta resolver las actividades utilizando los recursos que hemos visto y la documentación extendida que encontrarás en el “Aula Virtual”.

Recomendaciones

Importante

- **No uses NUNCA tildes, ni eñes, ni espacios, ni caracteres no alfanuméricos** (salvo el guión bajo) **en los metadatos** (nombres de elementos de una base de datos).
- Sé coherente con el uso de mayúsculas/minúsculas.

1. Colección Libros

Actividad no evaluable

Utiliza la base de datos **pruebas** para crear la colección **"libros"** y los datos siguientes:

```
// -----  
// Colección: libros  
// -----  
  
//poblar la base de datos  
use pruebas  
db.libros.drop()  
  
db.libros.insertOne(  
  {  
    _id: 1,  
    titulo: 'El aleph',  
    autor: 'Borges',  
    editorial: 'Planeta',  
    cantidad: 50  
  }  
)  
db.libros.insertOne(  
  {  
    _id: 2,  
    titulo: 'Martin Fierro',  
    autor: 'Jose Hernández',  
    editorial: 'Planeta',  
    precio: 52,  
    cantidad: 12  
  }  
)  
db.libros.insertOne(  
  {  
    _id: 3,  
    titulo: 'Aprenda PHP',  
    autor: 'Mario Molina',  
    editorial: 'Siglo XXI',  
    precio: 50,
```

```
        cantidad: 20
    }
)
db.libros.insertOne(
{
    _id: 4,
    titulo: 'Java en 10 minutos',
    editorial: 'Siglo XXI',
    precio: 45,
    cantidad: 1
}
)
```

1.1. Ejercicio

- a) Recuperar todos los libros que en el campo cantidad tienen 50 o más.
- b) Recuperar todos los libros que en el campo cantidad hay un valor distinto a 50.

Solución

a) Consulta 1

```
var j_valor      = {$gte:50}
var j_condicion  = {cantidad:j_valor}
db.libros.find(j_condicion)
```

```
pruebas> var j_valor      = {$gte:50}

pruebas> var j_condicion  = {cantidad:j_valor}

pruebas> db.libros.find(j_condicion)
[
  {
    _id: 1,
    titulo: 'El aleph',
    autor: 'Borges',
    editorial: 'Planeta',
    cantidad: 50
  }
]
```

b) Consulta 2

```
var j_valor      = {$ne:50}  
var j_condicion = {cantidad:j_valor}  
db.libros.find(j_condicion)
```

```
pruebas> var j_valor      = {$ne:50}  
  
pruebas> var j_condicion = {cantidad:j_valor}  
  
pruebas> db.libros.find(j_condicion)  
[  
  {  
    _id: 2,  
    titulo: 'Martin Fierro',  
    autor: 'Jose Hernández',  
    editorial: 'Planeta',  
    precio: 52,  
    cantidad: 12  
  },  
  {  
    _id: 3,  
    titulo: 'Aprenda PHP',  
    autor: 'Mario Molina',  
    editorial: 'Siglo XXI',  
    precio: 50,  
    cantidad: 20  
  },  
  {  
    _id: 4,  
    titulo: 'Java en 10 minutos',  
    editorial: 'Siglo XXI',  
    precio: 45,  
    cantidad: 1  
  }  
]
```

1.2. Ejercicio

Recuperar toda la información sobre el libro más barato, incluyendo el id.

Solución

```
var j_valor = {$exists: true}
var j_filtro = {precio: j_valor}
var j_orden = {precio:1}
db.libros.find(j_filtro).sort(j_orden).limit(1)
```

```
pruebas> var j_valor = {$exists: true}

pruebas> var j_filtro = {precio: j_valor}

pruebas> var j_orden = {precio:1}

pruebas> db.libros.find(j_filtro).sort(j_orden).limit(1)
[
  {
    _id: 4,
    titulo: 'Java en 10 minutos',
    editorial: 'Siglo XXI',
    precio: 45,
    cantidad: 1
  }
]
```


1.3. Ejercicio

Recuperar el título e id de los libros que NO tienen un precio indicado.

Solución

```
var j_valor      = { $exists: false }
var j_filtro     = { precio: j_valor }
var j_proyeccion = { titulo: 1 }
db.libros.find(j_filtro, j_proyeccion)
```

```
pruebas> var j_valor      = { $exists: false }
pruebas> var j_filtro     = { precio: j_valor }
pruebas> var j_proyeccion = { titulo: 1 }
pruebas> db.libros.find(j_filtro, j_proyeccion)
[ { _id: 1, titulo: 'El aleph' } ]
```

1.4. Ejercicio

Recuperar todos los libros cuyo precio esté comprendido entre 20 y 45.

Solución

```
//opción 1
var j_valor1      = {$gte:20}
var j_condicion1  = {precio:j_valor1}
var j_valor2      = {$lte:45}
var j_condicion2  = {precio:j_valor2}
var j_condicionFinal = {$and:[j_condicion1, j_condicion2]}
db.libros.find(j_condicionFinal)
//opción 2
var j_valor      = {$gte:20, $lte:45}
var j_condicion  = {precio:j_valor}
db.libros.find(j_condicion)
```

```
pruebas> var j_valor1      = {$gte:20}

pruebas> var j_condicion1  = {precio:j_valor1}

pruebas> var j_valor2      = {$lte:45}

pruebas> var j_condicion2  = {precio:j_valor2}

pruebas> var j_condicionFinal = {$and:[j_condicion1, j_condicion2]}

pruebas> db.libros.find(j_condicionFinal)
[
  {
    _id: 4,
```

```
    titulo: 'Java en 10 minutos',  
    editorial: 'Siglo XXI',  
    precio: 45,  
    cantidad: 1  
  }  
]
```

```
pruebas> var j_valor      = {$gte:20, $lte:45}  
  
pruebas> var j_condicion = {precio:j_valor}  
  
pruebas> db.libros.find(j_condicion)  
[  
  {  
    _id: 4,  
    titulo: 'Java en 10 minutos',  
    editorial: 'Siglo XXI',  
    precio: 45,  
    cantidad: 1  
  }  
]
```

1.5. Ejercicio

- a) Recuperar todos los libros de la editorial 'Planeta'.
- b) Recuperar todos los libros que no pertenezcan a la editorial 'Planeta'.

Solución

a) Consulta 1

```
//opción 1
var j_condicion = {editorial: 'Planeta'}
db.libros.find(j_condicion)
//opción 2
var j_valor      = 'Planeta'
var j_condicion = {editorial: j_valor}
db.libros.find(j_condicion)
```

```
pruebas> var j_condicion = {editorial: 'Planeta'}

pruebas> db.libros.find(j_condicion)
[
  {
    _id: 1,
    titulo: 'El aleph',
    autor: 'Borges',
    editorial: 'Planeta',
    cantidad: 50
  },
  {
    _id: 2,
    titulo: 'Martin Fierro',
    autor: 'Jose Hernández',
```

```

    editorial: 'Planeta',
    precio: 52,
    cantidad: 12
  }
]

```

b) Consulta 2

```

var j_valor      = {$ne:'Planeta'}
var j_condicion = {editorial:j_valor}
db.libros.find(j_condicion)

```

```

pruebas> var j_valor      = {$ne:'Planeta'}

pruebas> var j_condicion = {editorial:j_valor}

pruebas> db.libros.find(j_condicion)
[
  {
    _id: 3,
    titulo: 'Aprenda PHP',
    autor: 'Mario Molina',
    editorial: 'Siglo XXI',
    precio: 50,
    cantidad: 20
  },
  {
    _id: 4,
    titulo: 'Java en 10 minutos',
    editorial: 'Siglo XXI',
    precio: 45,
    cantidad: 1
  }
]

```

1.6. Ejercicio

Recuperar los libros que tienen un precio mayor o igual a 50 o la cantidad es 1.

Solución

```
var j_valor1      = {$gte:50}
var j_condicion1  = {precio:j_valor1}
var j_condicion2  = {cantidad:1}
var j_condicionFinal = {$or:[j_condicion1, j_condicion2]}
db.libros.find(j_condicionFinal)
```

```
pruebas> var j_valor1      = {$gte:50}

pruebas> var j_condicion1  = {precio:j_valor1}

pruebas> var j_condicion2  = {cantidad:1}

pruebas> var j_condicionFinal = {$or:[j_condicion1, j_condicion2]}

pruebas> db.libros.find(j_condicionFinal)
[
  {
    _id: 2,
    titulo: 'Martin Fierro',
    autor: 'Jose Hernández',
    editorial: 'Planeta',
    precio: 52,
    cantidad: 12
  },
  {
    _id: 3,
```

```
    titulo: 'Aprenda PHP',  
    autor: 'Mario Molina',  
    editorial: 'Siglo XXI',  
    precio: 50,  
    cantidad: 20  
  },  
  {  
    _id: 4,  
    titulo: 'Java en 10 minutos',  
    editorial: 'Siglo XXI',  
    precio: 45,  
    cantidad: 1  
  }  
]
```

1.7. Ejercicio

Recuperar solo el título de los tres primeros libros con precio mayor que 20, ordenados por precio descendente, omitiendo el id.

Solución

```
var j_valor      = { $gt: 20 }
var j_filtro     = { precio: j_valor }
var j_proyeccion = { titulo: 1, _id: 0 }
var j_orden      = { precio: -1 }
db.libros.find(j_filtro, j_proyeccion).sort(j_orden).limit(3)
```

```
pruebas> var j_valor      = { $gt: 20 }

pruebas> var j_filtro     = { precio: j_valor }

pruebas> var j_proyeccion = { titulo: 1, _id: 0 }

pruebas> var j_orden      = { precio: -1 }

pruebas> db.libros.find(j_filtro, j_proyeccion).sort(j_orden).limit(3)
[
  { titulo: 'Martin Fierro' },
  { titulo: 'Aprenda PHP' },
  { titulo: 'Java en 10 minutos' }
]
```


1.8. Ejercicio

Recuperar el título y precio del libro más caro, omitiendo el id.

Solución

```
var j_proyeccion = {titulo:1, precio: 1, _id:0}  
var j_orden      = {precio:-1}  
db.libros.find({}, j_proyeccion).sort(j_orden).limit(1)
```

```
pruebas> var j_proyeccion = {titulo:1, precio: 1, _id:0}  
  
pruebas> var j_orden      = {precio:-1}  
  
pruebas> db.libros.find({}, j_proyeccion).sort(j_orden).limit(1)  
[ { titulo: 'Martin Fierro', precio: 52 } ]
```

2. Colección Artículos

Actividad no evaluable

Utiliza la base de datos **pruebas** para crear la colección "**articulos**" y los datos siguientes:

```
//poblar la base de datos
use pruebas
db.articulos.drop()

db.articulos.insertOne(
  {
    _id: 1,
    nombre: 'MULTIFUNCION HP DESKJET 2675',
    rubro: 'impresora',
    precio: 3000,
    stock: 20
  }
)
db.articulos.insertOne(
  {
    _id: 2,
    nombre: 'MULTIFUNCION EPSON EXPRESSION XP241',
    rubro: 'impresora',
    precio: 3700,
    stock: 5
  }
)
db.articulos.insertOne(
  {
    _id: 3,
    nombre: 'LED 19 PHILIPS',
    rubro: 'monitor',
    precio: 4500,
    stock: 2
  }
)
db.articulos.insertOne(
  {
    _id: 4,
```

```
    nombre: 'LED 22 PHILIPS',
    rubro: 'monitor',
    precio: 5700,
    stock: 4
  }
)
db.articulos.insertOne(
  {
    _id: 5,
    nombre: 'LED 27 PHILIPS',
    rubro: 'monitor',
    precio: 12000,
    stock: 1
  }
)
db.articulos.insertOne(
  {
    _id: 6,
    nombre: 'LOGITECH M90',
    rubro: 'mouse',
    precio: 300,
    stock: 4
  }
)
)
```

2.1. Ejercicio

Imprimir todos los documentos de la colección 'articulos' que no son impresoras.

Solución

```
var j_valor      = {$ne:'impresora'}
var j_condicion = {tipo:j_valor}
db.articulos.find(j_condicion)
```

```
pruebas> var j_valor      = {$ne:'impresora'}

pruebas> var j_condicion = {tipo:j_valor}

pruebas> db.articulos.find(j_condicion)
[
  {
    _id: 3,
    nombre: 'LED 19 PHILIPS',
    tipo: 'monitor',
    precio: 4500,
    stock: 2
  },
  {
    _id: 4,
    nombre: 'LED 22 PHILIPS',
    tipo: 'monitor',
    precio: 5700,
    stock: 4
  },
  {
    _id: 5,
```

```
    nombre: 'LED 27 PHILIPS',  
    tipo: 'monitor',  
    precio: 12000,  
    stock: 1  
  },  
  {  
    _id: 6,  
    nombre: 'LOGITECH M90',  
    tipo: 'mouse',  
    precio: 300,  
    stock: 4  
  }  
]
```

2.2. Ejercicio

Imprimir todos los artículos que pertenecen al tipo de 'mouse'.

Solución

```
//opción 1
var j_valor      = {$eq: 'mouse'}
var j_condicion = {tipo: j_valor}
db.articulos.find(j_condicion)
//opción 2
var j_condicion = {tipo: 'mouse'}
db.articulos.find(j_condicion)
```

```
pruebas> var j_condicion = {tipo: 'mouse'}

pruebas> db.articulos.find(j_condicion)
[
  {
    _id: 6,
    nombre: 'LOGITECH M90',
    tipo: 'mouse',
    precio: 300,
    stock: 4
  }
]
```

2.3. Ejercicio

Imprimir todos los artículos con un precio mayor o igual a 5000.

Solución

```
var j_valor      = {$gte: 5000}
var j_condicion  = {precio:j_valor}
db.articulos.find(j_condicion)
```

```
pruebas> var j_valor      = {$gte: 5000}

pruebas> var j_condicion  = {precio:j_valor}

pruebas> db.articulos.find(j_condicion)
[
  {
    _id: 4,
    nombre: 'LED 22 PHILIPS',
    tipo: 'monitor',
    precio: 5700,
    stock: 4
  },
  {
    _id: 5,
    nombre: 'LED 27 PHILIPS',
    tipo: 'monitor',
    precio: 12000,
    stock: 1
  }
]
```


2.4. Ejercicio

Imprimir todas las impresoras que tienen un precio mayor o igual a 3500.

Solución

```
//opción 1
var j_valor1      = {$eq: 'impresora'}
var j_condicion1  = {tipo: j_valor1}
var j_valor2      = {$gte: 3500}
var j_condicion2  = {precio:j_valor2}
var j_condicionFinal = {$and:[j_condicion1, j_condicion2]}
db.articulos.find(j_condicionFinal)

//opción 2
var j_condicion1  = {tipo: 'impresora'}
var j_valor2      = {$gte: 3500}
var j_condicion2  = {precio:j_valor2}
var j_condicionFinal = {$and:[j_condicion1, j_condicion2]}
db.articulos.find(j_condicionFinal)
```

```
pruebas> var j_condicion1      = {tipo: 'impresora'}

pruebas> var j_valor2          = {$gte: 3500}

pruebas> var j_condicion2      = {precio:j_valor2}

pruebas> var j_condicionFinal = {$and:[j_condicion1, j_condicion2]}

pruebas> db.articulos.find(j_condicionFinal)
[
  {
    _id: 2,
```

```
nombre: 'MULTIFUNCION EPSON EXPRESSION XP241',  
tipo: 'impresora',  
precio: 3700,  
stock: 5  
}  
]
```

2.5. Ejercicio

Imprimir todos los artículos cuyo *stock* se encuentra comprendido entre 0 y 4.

Solución

```
//opción 1
var j_valor1      = {$gte:0}
var j_condicion1  = {stock:j_valor1}
var j_valor2      = {$lte:4}
var j_condicion2  = {stock:j_valor2}
var j_condicionFinal = {$and:[j_condicion1, j_condicion2]}
db.articulos.find(j_condicionFinal)
//opción 2
var j_valor      = {$gte:0, $lte:4}
var j_condicion  = {stock:j_valor}
db.articulos.find(j_condicion)
```

```
pruebas> var j_valor      = {$gte:0, $lte:4}

pruebas> var j_condicion  = {stock:j_valor}

pruebas> db.articulos.find(j_condicion)
[
  {
    _id: 3,
    nombre: 'LED 19 PHILIPS',
    tipo: 'monitor',
    precio: 4500,
    stock: 2
  },
  {
```

```
_id: 4,  
nombre: 'LED 22 PHILIPS',  
tipo: 'monitor',  
precio: 5700,  
stock: 4  
},  
{  
  _id: 5,  
  nombre: 'LED 27 PHILIPS',  
  tipo: 'monitor',  
  precio: 12000,  
  stock: 1  
},  
{  
  _id: 6,  
  nombre: 'LOGITECH M90',  
  tipo: 'mouse',  
  precio: 300,  
  stock: 4  
}  
]
```

2.6. Ejercicio

Imprimir el nombre de los artículos "monitor o impresora", ordenados por precio descendente.

Solución

```
//opción 1
var j_valor1      = {tipo : {'$eq': 'impresora'}}
var j_valor2      = {tipo : {'$eq': 'monitor'}}
var j_filtro      = {'$or': [j_valor1, j_valor2]}
var j_proyeccion  = {nombre:1, _id:0}
var j_orden       = {precio:-1}
db.articulos.find(j_filtro, j_proyeccion).sort(j_orden)

//opción 2
var j_valor1      = {tipo : 'impresora'}
var j_valor2      = {tipo : 'monitor'}
var j_filtro      = {'$or': [j_valor1, j_valor2]}
var j_proyeccion  = {nombre:1, _id:0}
var j_orden       = {precio:-1}
db.articulos.find(j_filtro, j_proyeccion).sort(j_orden)
```

```
pruebas> var j_valor1      = {tipo : 'impresora'}

pruebas> var j_valor2      = {tipo : 'monitor'}

pruebas> var j_filtro      = {'$or': [j_valor1, j_valor2]}

pruebas> var j_proyeccion  = {nombre:1, _id:0}
```

```
pruebas> var j_orden      = {precio:-1}

pruebas> db.articulos.find(j_filtro, j_proyeccion).sort(j_orden)
[
  { nombre: 'LED 27 PHILIPS' },
  { nombre: 'LED 22 PHILIPS' },
  { nombre: 'LED 19 PHILIPS' },
  { nombre: 'MULTIFUNCION EPSON EXPRESSION XP241' },
  { nombre: 'MULTIFUNCION HP DESKJET 2675' }
]
```

3. Colección Medicamentos

Actividad no evaluable

Utiliza la base de datos **pruebas** para crear la colección "**medicamentos**" y los datos siguientes:

```
// -----  
// Colección: medicamentos  
// -----  
  
//poblar la base de datos  
use pruebas  
db.medicamentos.drop()  
  
db.medicamentos.insertOne(  
  {  
    _id: 1,  
    nombre: 'Sertal',  
    laboratorio: 'Roche',  
    precio: 5.2,  
    cantidad: 101  
  }  
)  
db.medicamentos.insertOne(  
  {  
    _id: 2,  
    nombre: 'Buscapina',  
    laboratorio: 'Roche',  
    precio: 4.10,  
    cantidad: 200  
  }  
)  
db.medicamentos.insertOne(  
  {  
    _id: 3,  
    nombre: 'Amoxidal 500',  
    laboratorio: 'Bayer',  
    precio: 15.60,
```

```
        cantidad: 140
    }
)
db.medicamentos.insertOne(
    {
        _id: 4,
        nombre: 'Paracetamol 500',
        laboratorio: 'Bago',
        precio: 1.90,
        cantidad: 300
    }
)
db.medicamentos.insertOne(
    {
        _id: 5,
        nombre: 'Bayaspirina',
        laboratorio: 'Bayer',
        precio: 2.10,
        cantidad: 100
    }
)
db.medicamentos.insertOne(
    {
        _id: 6,
        nombre: 'Amoxidal jarabe',
        laboratorio: 'Bayer',
        precio: 5.10,
        cantidad: 50
    }
)
)
```


3.1. Ejercicio

Recupera los medicamentos cuyo laboratorio sea 'Roche' y cuyo precio sea menor a 5.

Solución

```
//opción 1
var j_valor1      = {$eq: 'Roche'}
var j_condicion1  = {laboratorio: j_valor1}
var j_valor2      = {$lt: 5}
var j_condicion2  = {precio: j_valor2}
var j_condicionFinal = {$and: [j_condicion1, j_condicion2]}
db.medicamentos.find(j_condicionFinal)

//opción 2
var j_condicion1  = {laboratorio: 'Roche'}
var j_valor2      = {$lt: 5}
var j_condicion2  = {precio: j_valor2}
var j_condicionFinal = {$and: [j_condicion1, j_condicion2]}
db.medicamentos.find(j_condicionFinal)
```

```
pruebas> var j_condicion1      = {laboratorio: 'Roche'}

pruebas> var j_valor2          = {$lt: 5}

pruebas> var j_condicion2      = {precio: j_valor2}

pruebas> var j_condicionFinal = {$and: [j_condicion1, j_condicion2]}

pruebas> db.medicamentos.find(j_condicionFinal)
[
  {
    _id: 2,
```

```
  nombre: 'Buscapina',  
  laboratorio: 'Roche',  
  precio: 4.1,  
  cantidad: 200  
}  
]
```

3.2. Ejercicio

Muestra todos los medicamentos cuyo laboratorio NO sea "Bayer".

Solución

```
var j_valor      = {$ne:'Bayer'}  
var j_condicion = {laboratorio:j_valor}  
db.medicamentos.find(j_condicion)
```

```
pruebas> var j_valor      = {$ne:'Bayer'}  
  
pruebas> var j_condicion = {laboratorio:j_valor}  
  
pruebas> db.medicamentos.find(j_condicion)  
[  
  {  
    _id: 1,  
    nombre: 'Sertal',  
    laboratorio: 'Roche',  
    precio: 5.2,  
    cantidad: 101  
  },  
  {  
    _id: 2,  
    nombre: 'Buscapina',  
    laboratorio: 'Roche',  
    precio: 4.1,  
    cantidad: 200  
  },  
  {  
    _id: 4,
```

```
    nombre: 'Paracetamol 500',  
    laboratorio: 'Bago',  
    precio: 1.9,  
    cantidad: 300  
  }  
]
```

3.3. Ejercicio

Muestra todos los medicamentos cuyo laboratorio sea "Bayer" y cuya cantidad sea distinta de 100.

Solución

```
var j_condicion1    = {laboratorio:'Bayer'}
var j_valor2        = {$ne:100}
var j_condicion2    = {cantidad:j_valor2}
var j_condicionFinal = {$and:[j_condicion1, j_condicion2]}
db.medicamentos.find(j_condicionFinal)
```

```
pruebas> var j_condicion1    = {laboratorio:'Bayer'}

pruebas> var j_valor2        = {$ne:100}

pruebas> var j_condicion2    = {cantidad:j_valor2}

pruebas> var j_condicionFinal = {$and:[j_condicion1, j_condicion2]}

pruebas> db.medicamentos.find(j_condicionFinal)
[
  {
    _id: 3,
    nombre: 'Amoxidal 500',
    laboratorio: 'Bayer',
    precio: 15.6,
    cantidad: 140
  },
  {
    _id: 6,
```

```
    nombre: 'Amoxidal jarabe',  
    laboratorio: 'Bayer',  
    precio: 5.1,  
    cantidad: 50  
  }  
]
```

3.4. Ejercicio

Recupera el nombre y laboratorio de los medicamentos con mayor *stock* (por cantidad descendente).

Solución

```
var j_proyeccion = {nombre:1, laboratorio:1, _id:0}
var j_orden      = {cantidad:-1}
db.medicamentos.find({}, j_proyeccion).sort(j_orden)
```

```
pruebas> var j_proyeccion = {nombre:1, laboratorio:1, _id:0}

pruebas> var j_orden      = {cantidad:-1}

pruebas> db.medicamentos.find({}, j_proyeccion).sort(j_orden)
[
  { nombre: 'Paracetamol 500', laboratorio: 'Bago' },
  { nombre: 'Buscapina', laboratorio: 'Roche' },
  { nombre: 'Amoxidal 500', laboratorio: 'Bayer' },
  { nombre: 'Sertal', laboratorio: 'Roche' },
  { nombre: 'Bayaspirina', laboratorio: 'Bayer' },
  { nombre: 'Amoxidal jarabe', laboratorio: 'Bayer' }
]
```

3.5. Ejercicio

Recupera el nombre y el laboratorio del medicamento con menor *stock*.

Solución

```
var j_proyeccion = {nombre:1, laboratorio:1, _id:0}
var j_orden      = {cantidad:1}
db.medicamentos.find({}, j_proyeccion).sort(j_orden).limit(1)
```

```
pruebas> var j_proyeccion = {nombre:1, laboratorio:1, _id:0}

pruebas> var j_orden      = {cantidad:1}

pruebas> db.medicamentos.find({}, j_proyeccion).sort(j_orden).limit(1)
[ { nombre: 'Amoxidal jarabe', laboratorio: 'Bayer' } ]
```


3.6. Ejercicio

Recupera el nombre e id de los dos medicamentos con mayor *stock*, ordenados por *stock* descendente.

Solución

```
var j_proyeccion = {nombre:1}
var j_orden      = {cantidad:-1}
db.medicamentos.find({}, j_proyeccion).sort(j_orden).limit(2)
```

```
pruebas> var j_proyeccion = {nombre:1}

pruebas> var j_orden      = {cantidad:-1}

pruebas> db.medicamentos.find({}, j_proyeccion).sort(j_orden).limit(2)
[
  { _id: 4, nombre: 'Paracetamol 500' },
  { _id: 2, nombre: 'Buscapina' }
]
```

3.7. Ejercicio

Recupera todos los datos del medicamento más barato, excepto el id.

Solución

```
var j_proyeccion = {_id:0}
var j_orden      = {precio:1}
db.medicamentos.find({}, j_proyeccion).sort(j_orden).limit(1)
```

```
pruebas> var j_proyeccion = {_id:0}

pruebas> var j_orden      = {precio:1}

pruebas> db.medicamentos.find({}, j_proyeccion).sort(j_orden).limit(1)
[
  {
    nombre: 'Paracetamol 500',
    laboratorio: 'Bago',
    precio: 1.9,
    cantidad: 300
  }
]
```

4. Bibliografía

- ¿Qué es SQL y NoSQL? [Platzi]. <https://www.youtube.com/watch?v=CuAYLX6reXE>
- NO SQL: como se modelan las bbdd no relacionales? [HolaMundo]. <https://www.youtube.com/watch?v=Zdlude8l8w4>
- El concepto NoSQL, o cómo almacenar tus datos en una base de datos no relacional. <https://www.genbeta.com/desarrollo/el-concepto-nosql-o-como-almacenar-tus-datos-en-una-base-de-datos-no-relacional>
- Metodologías ágiles Scrum, Kanban 04 Triángulo de hierro. https://www.youtube.com/watch?v=PdzW4G_hbsw
- Proyectos ágiles. Triángulo de hierro. <https://proyectosagiles.org/triangulo-hierro/>
- Teorema CAP. Píldoras de conocimiento. https://www.youtube.com/watch?v=Ydv-y_oH_CY
- Una introducción a MongoDB. <https://www.genbeta.com/desarrollo/una-introduccion-a-mongodb>
- MongoDB: qué es, cómo funciona y cuándo podemos usarlo (o no). <https://www.genbeta.com/desarrollo/mongodb-que-es-como-functiona-y-cuando-podemos-usarlo-o-no>
- Tutorial gratuito de 30 vídeos. https://www.youtube.com/watch?v=nIOWsnO-d7Q&list=PLXXiznRYETLcJE_4U9qN2pysZOSYyL4Mh



Obra publicada con [Licencia Creative Commons Reconocimiento Compartir igual 4.0](https://creativecommons.org/licenses/by-sa/4.0/)