

# UNIDAD 8

## BASES DE DATOS NOSQL: MONGODB

BASES DE DATOS 22/23  
CFGS DAW

### BOLETÍN EJERCICIOS 8.1. MONGODB: DDL Y DQL SOLUCIONES

**Revisado por:**

Abelardo Martínez y Pau Miñana

**Autor:**

Sergio Badal

Licencia Creative Commons



**Reconocimiento - NoComercial - CompartirIgual (by-nc-sa):** No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

## ÍNDICE DE CONTENIDO

1. EJERCICIO LIBROS.....	3
2. EJERCICIO ARTÍCULOS.....	9
3. EJERCICIO MEDICAMENTOS.....	14

## UD7.2. BOLETÍN MONGODB DQL DQL

### 1. EJERCICIO LIBROS

```
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 Hernandez',
    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
  })
```

10. Recuperar todos los libros que en el campo cantidad tienen 50 o más:

```
db.libros.find( {cantidad: { $gte: 50}}).pretty()
```

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

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

```
var valor = { $exists: true }  
var filtro = { precio: valor }  
var orden = { precio: 1 }  
db.libros.find(filtro).sort(orden).limit(1).pretty()
```

```
pruebas> var valor = { $exists: true }  
pruebas> var filtro = { precio: valor }  
pruebas> var orden = { precio: 1 }  
pruebas> db.libros.find(filtro).sort(orden).limit(1).pretty()  
[  
  {  
    _id: 4,  
    titulo: 'Java en 10 minutos',  
    editorial: 'Siglo XXI',  
    precio: 45,  
    cantidad: 1  
  }  
]
```

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

```
var valor = { $exists: false }  
var filtro = { precio: valor }  
var proyeccion = { titulo: 1 }  
db.libros.find(filtro, proyeccion).pretty()
```

```
pruebas> var valor = { $exists: false }
pruebas> var filtro = { precio: valor }
pruebas> var proyeccion = { titulo: 1 }
pruebas> db.libros.find(filtro, proyeccion).pretty()
[ { _id: 1, titulo: 'El aleph' } ]
```

13. Recuperar todos los libros que en el campo cantidad hay un valor distinto a 50:

```
db.libros.find( { cantidad: { $ne : 50 } }).pretty()
```

```
pruebas> db.libros.find( { cantidad: { $ne : 50 } }).pretty()
[
  {
    _id: 2,
    titulo: 'Martin Fierro',
    autor: 'Jose Hernandez',
    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
  }
]
```

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

```
db.libros.find( { precio: { $gte : 20 , $lte : 45 } }).pretty()
db.libros.find( { $and:[{ precio: { $gte : 20 , $lte : 45 } }]}).pretty()
```

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

15. Recuperar todos los libros de la editorial 'Planeta':

```
db.libros.find( { editorial: 'Planeta' }).pretty()  
db.libros.find( { editorial: { $eq: 'Planeta'}}).pretty()
```

```
pruebas> db.libros.find( { editorial: 'Planeta' }).pretty()  
[  
  {  
    _id: 1,  
    titulo: 'El aleph',  
    autor: 'Borges',  
    editorial: 'Planeta',  
    cantidad: 50  
  },  
  {  
    _id: 2,  
    titulo: 'Martin Fierro',  
    autor: 'Jose Hernandez',  
    editorial: 'Planeta',  
    precio: 52,  
    cantidad: 12  
  }  
]
```

16. Recuperar todos los libros que no pertenezcan a la editorial 'Planeta':

```
db.libros.find( { editorial: { $ne : 'Planeta'}}).pretty()
```

```
pruebas> db.libros.find( { editorial: { $ne : 'Planeta' } }).pretty()  
[  
  {  
    _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  
  }  
]
```

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

```
db.libros.find({$or: [{precio:{$gte:50}}, {cantidad:1} ]}).pretty()
```

```
pruebas> db.libros.find({$or: [{precio:{$gte:50}}, {cantidad:1} ]}).pretty()  
[  
  {  
    _id: 2,  
    titulo: 'Martin Fierro',  
    autor: 'Jose Hernandez',  
    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  
  }  
]
```

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

```
var valor      = { $gt: 20 }
var filtro     = { precio: valor }
var proyeccion = { titulo:1, _id:0 }
var orden      = { precio:-1 }
db.libros.find(filtro, proyeccion).sort(orden).limit(3).pretty()
```

```
pruebas> var valor      = { $gt: 20 }
pruebas> var filtro     = { precio: valor }
pruebas> var proyeccion = { titulo:1, _id:0 }
pruebas> var orden      = { precio:-1 }
pruebas> db.libros.find(filtro, proyeccion).sort(orden).limit(3).pretty()
[
  { titulo: 'Martin Fierro' },
  { titulo: 'Aprenda PHP' },
  { titulo: 'Java en 10 minutos' }
]
```

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

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

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



## 2. EJERCICIO ARTÍCULOS

```
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
}
)
```

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

```
db.articulos.find({ rubro : {$ne: 'impresora'}}).pretty()
```

```
pruebas> db.articulos.find({ rubro : {$ne: 'impresora'}}).pretty()
[
  {
    _id: 3,
    nombre: 'LED 19 PHILIPS',
    rubro: 'monitor',
    precio: 4500,
    stock: 2
  },
  {
    _id: 4,
    nombre: 'LED 22 PHILIPS',
    rubro: 'monitor',
    precio: 5700,
    stock: 4
  },
  {
    _id: 5,
    nombre: 'LED 27 PHILIPS',
    rubro: 'monitor',
    precio: 12000,
    stock: 1
  },
  {
    _id: 6,
    nombre: 'LOGITECH M90',
    rubro: 'mouse',
    precio: 300,
    stock: 4
  }
]
```

21. Imprimir todos los artículos que pertenecen al rubro de 'mouse'

```
db.articulos.find({ rubro : {$eq: 'mouse'}}).pretty()
```

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

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

```
db.articulos.find({ precio : {$gte: 5000}}).pretty()
```

```
pruebas> db.articulos.find({ precio : {$gte: 5000}}).pretty()  
[  
  {  
    _id: 4,  
    nombre: 'LED 22 PHILIPS',  
    rubro: 'monitor',  
    precio: 5700,  
    stock: 4  
  },  
  {  
    _id: 5,  
    nombre: 'LED 27 PHILIPS',  
    rubro: 'monitor',  
    precio: 12000,  
    stock: 1  
  }  
]
```

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

```
db.articulos.find({ rubro : {$eq: 'impresora'}, precio : { $gte: 3500 } }).pretty()
```

```
pruebas> db.articulos.find({ rubro : {$eq: 'impresora'}, precio : { $gte: 3500 } }).pretty()  
[  
  {  
    _id: 2,  
    nombre: 'MULTIFUNCION EPSON EXPRESSION XP241',  
    rubro: 'impresora',  
    precio: 3700,  
    stock: 5  
  }  
]
```

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

```
db.articulos.find({ stock : {$gte: 0, $lte: 4} }).pretty()
```

```
pruebas> db.articulos.find({ stock : {$gte: 0, $lte: 4} }).pretty()
[
  {
    _id: 3,
    nombre: 'LED 19 PHILIPS',
    rubro: 'monitor',
    precio: 4500,
    stock: 2
  },
  {
    _id: 4,
    nombre: 'LED 22 PHILIPS',
    rubro: 'monitor',
    precio: 5700,
    stock: 4
  },
  {
    _id: 5,
    nombre: 'LED 27 PHILIPS',
    rubro: 'monitor',
    precio: 12000,
    stock: 1
  },
  {
    _id: 6,
    nombre: 'LOGITECH M90',
    rubro: 'mouse',
    precio: 300,
    stock: 4
  }
]
```

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

```
var valor1      = { rubro : {$eq: 'impresora'}}
var valor2      = { rubro : {$eq: 'monitor'}}
var filtro      = {$or: [valor1, valor2]}
var proyeccion  = {nombre:1, _id:0}
var orden       = {precio:-1}
db.articulos.find(filtro, proyeccion).sort(orden).pretty()
```

```
pruebas> var valor1      = { rubro : {$eq: 'impresora'}}
pruebas> var valor2      = { rubro : {$eq: 'monitor'}}
pruebas> var filtro      = {$or: [valor1, valor2]}
pruebas> var proyeccion  = {nombre:1, _id:0}
pruebas> var orden       = {precio:-1}
pruebas> db.articulos.find(filtro, proyeccion).sort(orden).pretty()
[
  { nombre: 'LED 27 PHILIPS' },
  { nombre: 'LED 22 PHILIPS' },
  { nombre: 'LED 19 PHILIPS' },
  { nombre: 'MULTIFUNCION EPSON EXPRESSION XP241' },
  { nombre: 'MULTIFUNCION HP DESKJET 2675' }
]
```

### 3. EJERCICIO MEDICAMENTOS

```
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
  }
)
```

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

```
db.medicamentos.find({laboratorio:'Roche',precio:{$lt:5}}).pretty()
```

```
pruebas> db.medicamentos.find({laboratorio:'Roche',precio:{$lt:5}}).pretty( )
[
  {
    _id: 2,
    nombre: 'Buscapina',
    laboratorio: 'Roche',
    precio: 4.1,
    cantidad: 200
  }
]
```

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

```
db.medicamentos.find({laboratorio: {$ne:'Bayer'}}).pretty()
```

```
pruebas> db.medicamentos.find({laboratorio: {$ne:'Bayer'}}).pretty()  
[  
  {  
    _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  
  }  
]
```

32. Muestra todos los medicamentos cuyo laboratorio sea "Bayer" y cuya cantidad NO sea=100

```
db.medicamentos.find({laboratorio:'Bayer',cantidad:{$not:{$eq:100}}}).pretty()
```

```
pruebas> db.medicamentos.find({laboratorio:'Bayer',cantidad:{$not:{$eq:100}}}).pretty()  
[  
  {  
    _id: 3,  
    nombre: 'Amoxidal 500',  
    laboratorio: 'Bayer',  
    precio: 15.6,  
    cantidad: 140  
  },  
  {  
    _id: 6,  
    nombre: 'Amoxidal jarabe',  
    laboratorio: 'Bayer',  
    precio: 5.1,  
    cantidad: 50  
  }  
]
```



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

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

```
pruebas> var proyeccion = {nombre:1, laboratorio:1, _id:0}
pruebas> var orden      = {cantidad:-1}
pruebas> db.medicamentos.find({}, proyeccion).sort(orden).pretty()
[
  { 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' }
]
```

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

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

```
pruebas> var proyeccion = {nombre:1, laboratorio:1, _id:0}
pruebas> var orden      = {cantidad:1}
pruebas> db.medicamentos.find({}, proyeccion).sort(orden).limit(1).pretty()
[ { nombre: 'Amoxidal jarabe', laboratorio: 'Bayer' } ]
```

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

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

```
pruebas> var proyeccion = {nombre:1}
pruebas> var orden      = {cantidad:-1}
pruebas> db.medicamentos.find({}, proyeccion).sort(orden).limit(2).pretty()
[
  { _id: 4, nombre: 'Paracetamol 500' },
  { _id: 2, nombre: 'Buscapina' }
]
```

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

```
var proyeccion = {_id:0}
var orden      = {precio:1}
db.medicamentos.find({}, proyeccion).sort(orden).limit(1).pretty()
```

```
pruebas> var proyeccion = {_id:0}
pruebas> var orden      = {precio:1}
pruebas> db.medicamentos.find({}, proyeccion).sort(orden).limit(1).pretty()
[
  {
    nombre: 'Paracetamol 500',
    laboratorio: 'Bago',
    precio: 1.9,
    cantidad: 300
  }
]
```