

Primeros Pasos

Instalar

```
sudo apt install mongodb
```

Comprobar estado

```
sudo service mongodb status
```

Abrir y salir de MongoDB CLI

```
mongo
```

```
exit
```

- Se monta en `localhost:27017`

MongoDB Compass

Descargamos la aplicación de la web de mongo. SO Ubuntu (.deb)

- Ruta: `mongodb://localhost:27017`

Shell MongoDB

- Ver BBDD: `show databases` | `dbs`
- Seleccionar una `use nombreDB`. Si no existe, la crea, y la guarda en el momento que creamos una primera colección.
- Borrar base de datos seleccionada: `db.dropDatabase()`

Manejo de colecciones y documentos

- Podemos crear una colección con `db.createCollection("name")` o creando un primer documento:

```
db.productos.insert({"nombre":"champú", "marca":"Loreal"})
```

- Ver las colecciones: `show collections`

- Obtener todos los documentos:

```
db.productos.find()
```

- Salida en formato JSON:

```
db.productos.find().pretty()
```

- Para recuperar un solo documento:

```
db.productos.findOne({"nombre": "champú"})
```

- Borrar docs:

```
db.users.remove({"nombre": "John"})
```

- Para actualizar un dato de uno o más documentos:

```
db.productos.update({"nombre": "galletas"}, {"nombre": "pastas"})
```

```
db.productos.update({"nombre": "galletas"}, {"nombre": "pastas"}, {"upsert": false, multi: true})
```

IMPORTANTE: Sin el \$set, “machaca” todos los atributos del documento.

- Añadir un nuevo dato a un documento:

```
db.productos.update({"nombre": "Champú"}, {$set: {"Precio": 27.85}})
```

- Incrementar el valor de un numérico en un documento:

```
db.productos.update({"nombre": "Champú"}, {$inc: {"precio": 3.47}})
```

- Incrementar el valor de un numérico en todos los documentos:

```
db.alumns.update({}, {$inc: {"precio": 2.00}}, false, true)
```

Filtrado y ordenación

- Buscar documentos: `db.usuarios.find({"nombre":"Paco"})`
- `db.myCollection.find({ "subjects.music" : { $exists : true } })`
- Buscar y ordenar por edad: `db.usuarios.find().sort({"edad": 1})`.
Si ponemos -1 en edad ordenará de mayor a menor.
- Limitar la salida: `db.alumnos.find().limit(3)`
- Para que vaya saltando resultados: `db.alumnos.skip(1)`
- Recuperar todos los documentos que contengan un registro en concreto: `db.alumnos.find({"matricula": {$exists: true}})`
- Buscar los documentos que tengan un registro mayor o igual al indicado: `db.alumnos.count({"edad": {$gte: 22, $lt: 30}})`
 - Mayor que: `$gt`
 - Mayor o igual: `$gte`
 - menor que: `$lt`
 - menor o igual: `$lte`
- Contar los documentos que tengan un registro concreto: `db.alumnos.count({"curso":"3ºB"})`
- Para match en un array

```
db.restaurants.find({grades : { $elemMatch:{"score":{$gt : 90}}}});
```

- Not equal

```
db.students.find({name: {$ne: "María"}})
```

- Match with several options

```
db.collection.find({property: {$in/$nin: [value, value]}})
```

Todos estos filtros se pueden ir combinando con `find()` para obtener el resultado exacto que se pueda requerir.

Notación de punto

Con la notación de punto asignada en las claves de los registros se podrá acceder a objetos anidados:

```
[{  
  "name": "Pedro",  
  "lastname": "Martinez Bravo",
```

```

"subjects": {
  "music": "1º",
  "art": "2º",
  "history": "1º"
}, {
  "name": "Aurora",
  "lastname": "Losantos Ramos",
  "subjects": {
    "music": "3º",
    "art": "1º",
    "math": "1º"
  }
}, {
  "name": "Enrique",
  "lastname": "Vargas Marinos",
  "subjects": {
    "music": "1º",
    "math": "2º",
    "history": "1º"
  }
}]

```

- Ahora podemos filtrar por una propiedad anidada:

```
db.alumnos.find({"subjects.music": "1º"}).pretty()
```

Referencias en MongoDB

Existe un modo de crear relaciones en MongoDB parecido a MySQL y se conocen como referencias:

- Tenemos un profesor de Literatura: `db.professors.insert({"name": "Miguel", "subject": "Literatura"})`
- Ahora se debe recuperar el ObjectId del profesor Miguel, para ello, se accede con `findOne`: `const literatureProfessor = db.professors.findOne({"name": "Miguel"})`
- Vamos a relacionar al profesor Miguel con los alumnos de Literatura de 1º:

```
db.alumnos.update({"subjects.literature": "1º"}, {$set: {"professor": literatureProfessor._id}}
```

Ahora si se vuelve a realizar un find con los alumnos se podrá comprobar que aparece el identificador del profesor referenciados con los alumnos.

- Si ejecutamos `db.alumnos.find({"professor": literatureProfessor._id})` encontrará a los alumnos referenciados con este profesor.

Copia de seguridad de datos MongoDB

- Para hacer copia desde terminal: `mongodump -h 127.0.0.1:27017 -d tienda -o tienda`
- Para restaurar desde terminal: `mongorestore -h 127.0.0.1:27017 <directorio>`
- Para importar documentos: `mongoimport --db dbName --collection collectionName --file fileName.json`