

## Lista de comandos básicos para MongoDB

### Crear, editar y eliminar bases de datos:

- `use nombre_base_datos` (Crear o seleccionar una base de datos)
- `show dbs` (Listar bases de datos)
- `db.dropDatabase()` (Eliminar una base de datos)

### Crear, editar y eliminar colecciones:

- `db.createCollection("nombre_coleccion")` (Crear una colección)
- `show collections` (Listar colecciones)
- `db.nombre_coleccion.drop()` (Eliminar una colección)

### Operaciones con datos (documentos):

- `db.nombre_coleccion.insertOne({ campo1: "valor1", campo2: "valor2" })` (Insertar un documento)
- `db.nombre_coleccion.insertMany([{ campo1: "valor1", campo2: "valor2" }, { campo1: "valor3", campo2: "valor4" }])` (Insertar varios documentos)
- `db.nombre_coleccion.find()` (Leer todos los documentos)
- `db.nombre_coleccion.find({ campo1: "valor1" })` (Leer documentos con filtro)
- `db.nombre_coleccion.findOne({ campo1: "valor1" })` (Leer un documento específico)
- `db.nombre_coleccion.updateOne({ campo1: "valor1" }, { $set: { campo2: "nuevo_valor" } })` (Actualizar un solo documento)
- `db.nombre_coleccion.updateMany({ campo1: "valor1" }, { $set: { campo2: "nuevo_valor" } })` (Actualizar varios documentos)
- `db.nombre_coleccion.deleteOne({ campo1: "valor1" })` (Eliminar un solo documento)
- `db.nombre_coleccion.deleteMany({ campo1: "valor1" })` (Eliminar varios documentos)

### Búsquedas, filtros y ordenamientos:

- `db.nombre_coleccion.find({ campo1: { $gt: valor } })` (Documentos donde `campo1` es mayor que un valor)
- `db.nombre_coleccion.find({ campo1: { $lt: valor } })` (Documentos donde `campo1` es menor que un valor)
- `db.nombre_coleccion.find({ campo1: { $in: ["valor1", "valor2"] } })` (Documentos donde `campo1` tiene un valor en una lista específica)

- `db.nombre_coleccion.find({ campo1: { $exists: true } })`  
(Documentos donde `campo1` existe)
- `db.nombre_coleccion.find().sort({ campo1: 1 })` (Ordenar documentos por `campo1` en orden ascendente)
- `db.nombre_coleccion.find().sort({ campo1: -1 })` (Ordenar documentos por `campo1` en orden descendente)
- `db.nombre_coleccion.find({ campo1: { $regex: /texto/i } })`  
(Documentos donde `campo1` coincide con una expresión regular)
- `db.nombre_coleccion.find({}, { campo1: 1, campo2: 0 })`  
(Proyección para incluir solo `campo1` y excluir `campo2`)
- `db.nombre_coleccion.find().limit(10)` (Limitar la cantidad de resultados a 10 documentos)
- `db.nombre_coleccion.find().skip(5)` (Saltar los primeros 5 documentos en los resultados)

### Operaciones avanzadas con filtros:

- `$and`: Realiza operación AND  
`db.nombre_coleccion.find({ $and: [{ campo1: "valor1" }, { campo2: "valor2" }] })` (Documentos donde `campo1` es "valor1" y `campo2` es "valor2")
- `$or`: Realiza operación OR  
`db.nombre_coleccion.find({ $or: [{ campo1: "valor1" }, { campo2: "valor2" }] })` (Documentos donde `campo1` es "valor1" o `campo2` es "valor2")
- `$lt`: Coincide con valores menores a un valor especificado  
`db.nombre_coleccion.find({ campo1: { $lt: 100 } })` (Documentos donde `campo1` es menor que 100)
- `$lte`: Coincide con valores menores o iguales a un valor especificado  
`db.nombre_coleccion.find({ campo1: { $lte: 100 } })` (Documentos donde `campo1` es menor o igual a 100)
- `$gt`: Coincide con valores mayores a un valor especificado  
`db.nombre_coleccion.find({ campo1: { $gt: 100 } })` (Documentos donde `campo1` es mayor que 100)
- `$gte`: Coincide con valores mayores o iguales a un valor especificado  
`db.nombre_coleccion.find({ campo1: { $gte: 100 } })` (Documentos donde `campo1` es mayor o igual a 100)
- `$ne`: Coincide con valores que no son iguales a un valor especificado  
`db.nombre_coleccion.find({ campo1: { $ne: "valor1" } })`  
(Documentos donde `campo1` no es "valor1")
- `$eq`: Coincide con valores que son iguales a un valor especificado  
`db.nombre_coleccion.find({ campo1: { $eq: "valor1" } })`  
(Documentos donde `campo1` es "valor1")

### Operaciones adicionales con arrays y subdocumentos:

- `db.nombre_coleccion.distinct("campo1")`  
Devuelve un array con los valores distintos del campo `campo1` en los documentos de la colección  
Ejemplo: `db.nombre_coleccion.distinct("categorias")`
- `$size`: Coincide con el número de elementos especificados en un array  
`db.nombre_coleccion.find({ campo1: { $size: 3 } })` (Documentos donde `campo1` es un array con exactamente 3 elementos)
- `$all`: Coincide con todos los valores definidos dentro de un array  
`db.nombre_coleccion.find({ campo1: { $all: ["valor1", "valor2"] } })` (Documentos donde el array `campo1` contiene "valor1" y "valor2")
- `$elemMatch`: Coincide con algún valor definido dentro de un array que cumpla todas las condiciones  
`db.nombre_coleccion.find({ campo1: { $elemMatch: { subcampo1: "valor1", subcampo2: { $gte: 10 } } } })`  
(Documentos donde el array `campo1` tiene al menos un elemento que cumple las condiciones)

### Operaciones con subdocumentos y expresiones regulares:

- `db.nombre_coleccion.find({ "subdocumento.campo": "valor" })`  
(Documentos donde un subdocumento llamado `subdocumento` tiene un campo `campo` igual a "valor")  
Ejemplo: `db.nombre_coleccion.find({ "direccion.ciudad": "Buenos Aires" })`
- `db.nombre_coleccion.find({ nombre: /^Max$/i })`  
(Documentos donde el campo `nombre` es exactamente "Max", ignorando mayúsculas/minúsculas)