### Entrar en MongoDB:

* sudo systemctl start mongod / brew services start mongodb-community (MAC OS)
* mongosh

### Ver:

* *Mostrar contenido de una coleccion*:
  + Mostrar todo
    - db.nombreColeccion.find()
  + Mostrar solo la primera
    - db.nombreColeccion.findOne()
  + Mostrar una columna
    - db.coleccion.find({columna:"valor}) -> si la columna es compuesta entre "" ("columna.columna": valor)
  + db.persoa({"nome":{**$eq**:"luis"}}) -> los que se llaman luis
  + db.persoa({"nome":{**$ne**:"luis"}}) -> los que no se llaman luis
  + db.persoa({"edade":{**$gt**:30}}) -> mayor que 30
  + db.persoa({"edade":{**$gte**:31}}) -> mayor o igual que 30
  + db.persoa({"edade":{**$lt:**30}}) -> menor que 30
  + db.persoa({"edade":{**$lte:**31}}) -> menor o igual que 31
  + db.persoa({"edade":{$ne:30}}) -> los que no tengan el campo edade o diferenteDe30
  + db.persoas.find({"nome":{**$in:**["luis","ana"]}}) -> que sean luis o ana
  + db.persoas.find({"nome":{**$nin:**["luis","ana"]}}) -> no sean ni luis ni ana
    - $nin -> incluye los que no tienen ese cam, **IMPORTANTE!!**, Si el objeto no tiene el campo nombre te los da igual, por lo que debemos hacer lo siguiente .
    - db.persoas.find( **{**edade: **{exists: true, $nin:[**30,40**]} }** ) -> para que solo salga los que tengan el campo edad y no tenga ni 30 ni 40
  + db.persoas.find( **{ $or: [ {**nome:"felix"**} , {**edade:31**} ] }**) -> una o otra
  + db.persoas.find( **{ $and: [ {**nome:"ana"**}** , **{**edade: **{**$gte:31**} } ] }** ) -> que se ana y tenga 31 o mas años
  + db.persoas.find**( {**edade: **{$not:{$gt:**30**}} }** ) -> los que no tengan edad o tengan-de31
    - $not: -> siempreincluye cuando ese campo NO existe
  + db.persoas.find( **{}, {edade: 1}** )
    - SOLO muestra el campo edade (todas las edades)
  + db.persoas.find( **{}, {edade: 0}** )
    - Que muestre todo excepto edade
  + db.persoas.find( **{}, {edade: 1, nome:1}** )
    - mostrar edad y nombre
  + db.persoas.find( {}, {edade: 1, nome:1, **\_id:0**} )
    - Para no mostrar el id (si no, sale siempre)
    - Solo se puede incluir y excluir si la exclusion es el **\_id**
  + db.persoas.find( {edade:{$gt:30}}, **{edade: 1, nome:1, \_id:0}** )
    - Edad mayor de 30 y que solo muestre edad y nombre
  + Que contengan una parte de texto
    - db.persoas.find({nome:**{$regex:"l"}**})
      * Saca los que tengan l
    - db.persoas.find({nome:{$regex:"**^l**"}})

Los que empiezen por l

* + - db.persoas.find({nome:{$regex:"**$s**"}})
      * Que terminen por s
    - db.persoas.find({**nome:/i/**})
      * Que tengan i
    - db.persoas.find({**nome:{$in:[/u/,/x/]**}}
      * Que tenga la letra U o la X
* *Mostrar colecciones:*
  + show collections
  + show tables
* *Bases de datos*:
  + show dbs

### Borrado:

* *Contenido Collections*
  + db.persoas.deleteOne({...})
    - tienes que especificar cual borrar para que no borre el primero que encuentre (... es la condicion)
  + db.persoas.deleteMany({})
    - borra todo el contenido
    - si pones una condicion borra los de la condicion
* *Collections:*
  + db.nombreTabla.drop()
* *Bases de datos:*
  + db.dropDatabase()

### Crear:

* Crear un dato/collecion:
  + (las colecciones se crean insertando una fila):
  + db.nombreTabla.insertOne({“nome”:”luis”, numeros:[2,3,56,89]})
    - Si creas 2 campos que se llamen igual, no se crean dos campos, coge el ultimo
* Crear una fila con id personalizado
  + db.persoas.insertOne({\_id:1, nome:"pedro"})

### Actualización

* Modificar todo el documento/fila por el nuevo:
  + db.people.replaceOne({\_id:1},{nome:"luis"})
    - Si tiene campo hello lo cambia por nome, si tiene 4 lo cambia por solo nombre
  + db.people.replaceOne({},{telefono:4444})
    - Cambia el primero que encuentra
* Modificar un documento/fila o añadirle un campo:
  + db.people.**updateOne**({name:"Smith"},{**$set:**{children:2.5}})
    - Busca el primer documento/fila del 1 campo(name en este caso) y cambia el segundo parametro, si no existe lo CREA
    - NO PUEDE CAMBIAR EL ID
  + db.people.updateOne({name:"Jones"}, {$set:{age:50}},**{upsert:true}**)
    - Si no encuentra el campo de condicion lo crea
* Modificar varios/todo los documentos/filas o añadirles un campo:
  + db.people.**updateMany({}**,{$set:{city:"NYC"}})
* Borrar un dato de un documento/fila:
  + db.people.updateOne({name:"Smith"},{**$unset**:{childre:443}})
    - Después del campo que quieres borrar no tienes por que poner el dato exacto de ese campo, puedes poner lo que quieras
* Modificar varios documentos/filas:
  + db.people.**updateMany**({name:"Smith"},{$set:{children:2.5}})
* Incrementar un campo:
  + db.books.updateOne({\_id:1}, {**$inc**:{stock:**1**}})
    - El numero que pones es en cuanto lo incrementas
* Disminuir un campo:
  + db.books.updateOne({\_id:1}, {**$inc**:{**stock**:**-**1}})
    - El numero que pones es en cuanto lo disminuyes
* Modificar varios campos:
  + db.books.updateOne({\_id:1}, **{$inc:**{stock:18},**$set:**{item:"ABCD"}**}**)
  + db.books.updateOne({\_id:1}, {$inc:{stock:5},$set:{item:"ABC123", "info.publisher":2222, tags:["software"], "ratings.1":{by:"xyz", rating: 3}}}

### Conectarse a una BD o crearla:

(Si al crearla si no metes ninguna coleccion no se ve en el show dbs)

* use nombreBaseDatos