

## Ejercicios MongoDB

### 1. Usando el interfaz de comandos de MongoDB:

- a. Crea una base de datos de nombre AT05\_03.
- b. Añade una colección de nombre equipos.
- c. Añade dos equipos. La estructura de un equipo es:
  - nombre
  - entrenador (persona)
  - jugadores (conjunto de personas)
  - lugar

El tipo persona está compuesto de: dni, nombre, apellidos, año\_nac, telefono, direccion (la dirección estará compuesta de calle y código postal)

- d. Muestra toda la información de los equipos

```
use AT05_03;  
  
db.equipos.find().pretty()
```

- e. Realiza una consulta que muestre los equipos pertenecientes a un lugar concreto.

```
db.equipos.find({'lugar': 'Madrid'})
```

- f. Realiza una consulta que muestre los datos del equipo en el que juegue un jugador con un dni concreto.

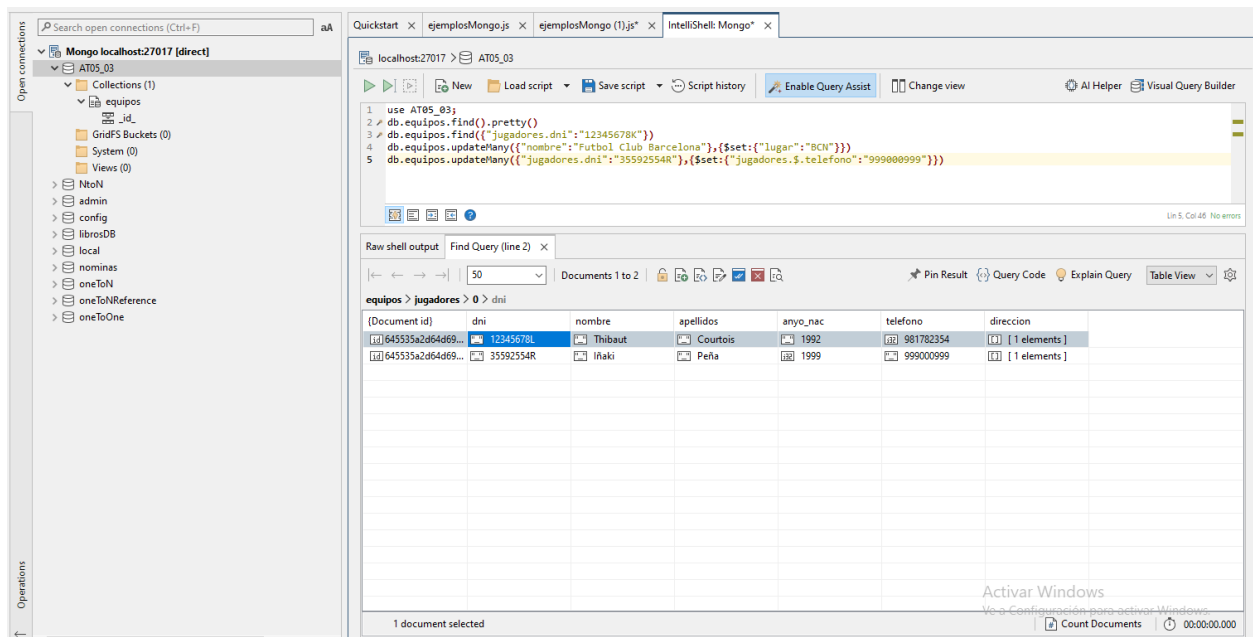
```
db.equipos.find({"jugadores.dni": "12345678K"})
```

- g. Modifica el “lugar” de un equipo con un “nombre” concreto.

```
db.equipos.updateMany({"nombre": "Futbol Club Barcelona"}, {$set: {"lugar": "BCN"}})
```

- h. Modifica el “telefono” de un jugador con un “dni” concreto. En caso de que haya varios documentos con ese jugador deben modificarse todos.

```
db.equipos.updateMany({"jugadores.dni": "35592554R"}, {$set: {"jugadores.$.telefono": "999000999"}})
```



The screenshot shows the MongoDB Shell interface. On the left, the 'Open connections' panel displays the database 'Mongo localhost:27017 [direct]' and the collection 'equipos'. The main area shows a list of collections: 'equipos', 'GridFS Buckets (0)', 'System (0)', and 'Views (0)'. The 'equipos' collection is selected, and its contents are displayed in a table view. The table has columns: 'Document id', 'dni', 'nombre', 'apellidos', 'anyo\_nac', 'telefono', and 'direccion'. Two documents are shown:

| Document id      | dni       | nombre  | apellidos | anyo_nac | telefono  | direccion    |
|------------------|-----------|---------|-----------|----------|-----------|--------------|
| 645535a2d6d69... | 12345678L | Thibaut | Courtois  | 1992     | 981782354 | [1 elements] |
| 645535a2d6d69... | 35592554R | Iñaki   | Peña      | 1999     | 999000999 | [1 elements] |

The bottom status bar indicates '1 document selected' and 'Count Documents'.

Captura de como he ido ejecutando las sentencias y del resultado de la última sentencia, dónde se puede apreciar el cambio del número de teléfono del jugador con dni 35592554R

**2. Importa la base de datos “AT05.03-Libros.json” usando el comando mongoimport: mongoimport --db AT05\_03 --collection libros --jsonArray --file AT05.03-libros.json y realiza las siguientes operaciones:**

- a. Obtener el nombre de todos los autores (únicamente el nombre)

```
use AT05_03;
db.libros.find({}, {'autor.nombre':1, '_id':0})
```

- b. Obtener los nombres de los autores cuyo apellido sea Proust

- c. Obtener los libros publicados en 1998 o en 2006

- d. Obtener el número de libros de la editorial Addison-Wesley

```
db.libros.find({"editorial": "Addison-Wesley"}).count()
```

- e. Obtener el libro que ocupa la tercera posición

```
db.libros.find({}, {"titulo":1, "_id":0}).skip(2).limit(1)
```

- f. Obtener la lista de autores de cada libro junto con el título

```
db.libros.find({}, {"autor":1, "titulo":1, "_id":0}).pretty()
```

- g. Obtener los títulos y años de publicación de los libros publicados con posterioridad a 2004.

```
db.libros.find( {"publicacion":{"$gt:2004}},
{"titulo":1,"_id":0,"publicacion":"1" } )
```

- h. Obtener los autores de los libros publicados en 1998 o en 2005

```
db.libros.find( {$or:[{"publicacion":2005}, {"publicacion":1998}]},
{"autor":1,"_id":0, "publicacion":"1" } )
```

- i. Obtener el autor, coste y año de publicación de los libros publicados desde 2001 y coste mayor que 20

```
db.libros.find({"publicacion":{"$gte:2001}, "coste":{"$gte:20.00}},
{"autor":1,"_id":0, "publicacion":"1", "coste":1} )
```

- j. Obtener los libros publicados por la editorial Addison-Wesley después de 2005.

```
db.libros.find({"publicacion":{"$gte:2001}, "coste":{"$gte:20.00}},
{"autor":1,"_id":0, "publicacion":"1", "coste":1} )
```

- k. Muestra los nombres de los distintos autores.

```
db.libros.distinct('autor.nombre')
```

- l. Muestra el menor y el mayor precio de los libros y el coste total de todos los libros

```
db.libros.aggregate([{$group: {_id: '$titulo', 'precioMin': {$min: '$coste'}, 'precioMax': {$max: '$coste'}, 'precioTotal': {$sum: '$coste'}}}])
```

- m. Obtén los nombres de los autores y el número de libros que han escrito

```
db.libros.aggregate([{$group: {_id: '$autor.nombre', 'obras': {$sum: 1}}}])
```

- n. Obtener los autores de más de un libro.

```
db.libros.aggregate([{$group: {_id: '$autor.nombre', 'obras': {$sum: 1}}}, {$match: {'obras': {$gt: 1}}}])
```

- o. Guardar una lista de libros con sus autores en una nueva colección llamada “autores”

```
db.libros.aggregate([{$project: {'autor': '$autor', 'titulo': '$titulo'}}, {$out: 'autores'}])
```

- p. Eliminar la colección “autores”

```
db.autores.drop()
```

- q. Asigna todos los libros de la editorial “VOIPA” a la editorial “Galaxia”.

```
db.libros.updateMany({'editorial': 'VOIPA'}, {$set: {'editorial': 'Galaxia'}})
```