

Soluciones a los ejercicios propuestos

Soluciones a los ejercicios del capítulo 3

1) Insertar los documentos dados en una base de datos llamada «media» en una única operación.

Figura 1

```
> var doc1={ "tipo": "libro",
...   "titulo": "Java para todos",
...   "ISBN": "987-1-2344-5334-8",
...   "editorial": "Anaya",
...   "autor": ["Pepe Caballero", "Isabel Sanz", "Tinoteo Marino"],
...   "capítulos": [
...     <"capitulo":1,
...       "titulo":"Primeros pasos en Java",
...       "longitud": 20
...     >,
...     <"capitulo":2,
...       "titulo":"Primeros pasos en Java",
...       "longitud": 25
...     >
...   ]
... }
> var doc2={ "tipo": "CD",
...   "artista":"Los piratas",
...   "Titulo": "Recuerdos",
...   "canciones": [
...     <"cancion":1,
...       "titulo":"Adios mi barco",
...       "longitud": "3:20"
...     >,
...     <"cancion":2,
...       "titulo":"Pajaritos",
...       "longitud": "4:15"
...     >
...   ]
... }
>
```

Figura 2

```

> doc3={ "tipo": "DVD",
...   "Titulo": "Matrix",
...   "estreno": 1999,
...   "actores": [
...     "Keanu Reeves",
...     "Carry-Anne Moss",
...     "Laurence Fishburne",
...     "Hugo Weaving",
...     "Gloria Foster",
...     "Joe Pantoliano"
...   ]
... }
{
  "tipo": "DVD",
  "Titulo": "Matrix",
  "estreno": 1999,
  "actores": [
    "Keanu Reeves",
    "Carry-Anne Moss",
    "Laurence Fishburne",
    "Hugo Weaving",
    "Gloria Foster",
    "Joe Pantoliano"
  ]
}
> db.nedia.insert([doc1,doc2,doc3])
BulkWriteResult({
  "writeErrors": [ ],
  "writeConcernErrors": [ ],
  "nInserted": 3,
  "nUpserted": 0,
  "nMatched": 0,
  "nModified": 0,
  "nRemoved": 0,
  "upserted": [ ]
})
> db.nedia.find()
{ "_id": ObjectId("54ddc7929d235c1da3b376de"), "tipo": "libro", "titulo": "Java para todos", "ISBN": "987-1-2344-5334-8", "editorial": "Anaya", "Autor": [ "Pepe Caballero", "Isabel Sanz", "Tinoteo Marino" ], "capitulos": [ { "capitulo": 1, "titulo": "Primeros pasos en Java", "longitud": 20 }, { "capitulo": 2, "titulo": "Primeros pasos en Java", "longitud": 25 } ] }
{ "_id": ObjectId("54ddc7929d235c1da3b376df"), "tipo": "CD", "Artista": "Los Piratas", "Titulo": "Recuerdos", "canciones": [ { "cancion": 1, "titulo": "A Dios mi barco", "longitud": "3:20" }, { "cancion": 2, "titulo": "Pajaritos", "longitud": "4:15" } ] }
{ "_id": ObjectId("54ddc7929d235c1da3b376e0"), "tipo": "DVD", "Titulo": "Matrix", "estreno": 1999, "actores": [ "Keanu Reeves", "Carry-Anne Moss", "Laurence Fishburne", "Hugo Weaving", "Gloria Foster", "Joe Pantoliano" ] }
>

```

2) Actualizar el documento que hace referencia a la película «Matrix» de manera que se cambie su estructura a:

```

{ "tipo": "DVD",
  "Titulo": "Matrix",
  "estreno": 1999,
  "genero": "accion"
}

```

Figura 3

```

> db.media.update(<<"_id" : ObjectId("54ddc7929d235c1da3b376e0")>>,< "tipo": "DVD"
, "Titulo": "Matrix", "estreno": 1999, "genero": "accion" > >
WriteResult(<<"nMatched" : 1, "nUpserted" : 0, "nModified" : 1 >>)
> db.media.findOne(<<"_id" : ObjectId("54ddc7929d235c1da3b376e0")>>)
{
  "_id" : ObjectId("54ddc7929d235c1da3b376e0"),
  "tipo" : "DVD",
  "Titulo" : "Matrix",
  "estreno" : 1999,
  "genero" : "accion"
}

```

3) Considerar un nuevo documento para la colección media:

```

{ "tipo": "Libro",
  "Titulo": "Constantinopla",
  "capitulos": 12,
  "leidos": 3
}

```

Añadir el documento a la colección media y a continuación incrementar en cinco unidades el valor de la clave «leídos».

Figura 4

```

> var doc4=< "tipo": "Libro",
... "Titulo": "Constantinopla",
... "capitulos": 12,
... "leidos": 3
... >
> db.media.insert<doc4>
WriteResult(<<"nInserted" : 1 >>)
> db.media.find<>
< "_id" : ObjectId("54ddc7929d235c1da3b376de"), "tipo" : "libro", "titulo" : "Ja
va para todos", "ISBN" : "987-1-2344-5334-8", "editorial" : "Anaya", "autor" : [
"Pepe Caballero", "Isabel Sans", "Timoteo Marino" ], "capitulos" : [ < "capitul
o" : 1, "titulo" : "Primeros pasos en Java", "longitud" : 20 >, < "capitulo" : 2
, "titulo" : "Primeros pasos en Java", "longitud" : 25 > ] >
< "_id" : ObjectId("54ddc7929d235c1da3b376df"), "tipo" : "CD", "Artista" : "Los
piratas", "Titulo" : "Recuerdos", "canciones" : [ < "cancion" : 1, "Titulo" : "ñ
dios mi barco", "longitud" : "3:20" >, < "cancion" : 2, "titulo" : "Fajaritos",
"longitud" : "4:15" > ] >
< "_id" : ObjectId("54ddc7929d235c1da3b376e0"), "tipo" : "DVD", "Titulo" : "Matr
ix", "estreno" : 1999, "genero" : "accion" >
< "_id" : ObjectId("54ddc9b69d235c1da3b376e1"), "tipo" : "Libro", "Titulo" : "Co
nstantinopla", "capitulos" : 12, "leidos" : 3 >
>

```

Figura 5

```

> db.nedia.update(<< "_id" : ObjectId("54ddc9b69d235c1da3b376e1") >>, << "$inc": {"leidos": 5} >>)
WriteResult(<< "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 >>)
> db.nedia.findOne(<< "_id" : ObjectId("54ddc9b69d235c1da3b376e1") >>)
{
  "_id" : ObjectId("54ddc9b69d235c1da3b376e1"),
  "tipo" : "Libro",
  "Titulo" : "Constantinopla",
  "capitulos" : 12,
  "leidos" : 8
}

```

4) Actualizar el documento referido a la película «Matrix» cambiando el valor de la clave «género» a «ciencia ficción».

Figura 6

```

> db.nedia.update(<< "_id" : ObjectId("54ddc7929d235c1da3b376e0") >>, << "$set": {"genero": "ciencia ficcion"} >>)
WriteResult(<< "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 >>)
> db.nedia.findOne(<< "_id" : ObjectId("54ddc7929d235c1da3b376e0") >>)
{
  "_id" : ObjectId("54ddc7929d235c1da3b376e0"),
  "tipo" : "DUD",
  "Titulo" : "Matrix",
  "estreno" : 1999,
  "genero" : "ciencia ficcion"
}

```

5) Actualizar el documento referido al libro «Java para todos», de manera que se elimine la clave «editorial».

Figura 7

```

> db.nedia.update(<< "_id" : ObjectId("54ddc7929d235c1da3b376de") >>, << "$unset": {"editorial": 1} >>)
WriteResult(<< "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 >>)
> db.nedia.findOne(<< "_id" : ObjectId("54ddc7929d235c1da3b376de") >>)
{
  "_id" : ObjectId("54ddc7929d235c1da3b376de"),
  "tipo" : "libro",
  "titulo" : "Java para todos",
  "ISBN" : "987-1-2344-5334-8",
  "autor" : [
    "Pepe Caballero",
    "Isabel Sanz",
    "Timoteo Marino"
  ],
  "capitulos" : [
    <
      "capitulo" : 1,
      "titulo" : "Primeros pasos en Java",
      "longitud" : 20
    >,
    <
      "capitulo" : 2,
      "titulo" : "Primeros pasos en Java",
      "longitud" : 25
    >
  ]
}

```

6) Actualizar el documento referido al libro «Java para todos» añadiendo el autor «María Sancho» al array de autores.

Figura 8

```
> db.media.update<<"_id" : ObjectId<"54ddc7929d235c1da3b376de">>,<"$push":{"auto
res":"María Sancho"}>>
WriteResult<<"nMatched" : 1, "nUpserted" : 0, "nModified" : 1 >>
> db.media.findOne<<"_id" : ObjectId<"54ddc7929d235c1da3b376de">>>
{
  "_id" : ObjectId<"54ddc7929d235c1da3b376de">,
  "tipo" : "libro",
  "titulo" : "Java para todos",
  "ISBN" : "987-1-2344-5334-8",
  "Autor" : [
    "Pepe Caballero",
    "Isabel Sanz",
    "Tinoteo Marino",
    "María Sancho"
  ],
  "capítulos" : [
    {
      "capitulo" : 1,
      "titulo" : "Primeros pasos en Java",
      "longitud" : 20
    },
    {
      "capitulo" : 2,
      "titulo" : "Primeros pasos en Java",
      "longitud" : 25
    }
  ]
}
```

7) Actualizar el documento referido a la película «Matrix» añadiendo al array «actores» los valores de «Antonio Banderas» y «Brad Pitt» en una única operación.

Figura 9

```
> db.media.update<<"_id" : ObjectId<"54ddc7929d235c1da3b376e0">>,<"$pushAll":{"a
ctores":["Antonio Banderas","Brad Pitt"]}>>
WriteResult<<"nMatched" : 1, "nUpserted" : 0, "nModified" : 1 >>
> db.media.findOne<<"_id" : ObjectId<"54ddc7929d235c1da3b376e0">>>
{
  "_id" : ObjectId<"54ddc7929d235c1da3b376e0">,
  "tipo" : "DUD",
  "Titulo" : "Matrix",
  "estreno" : 1999,
  "actores" : [
    "Keanu Reeves",
    "Carry-Anne Moss",
    "Laurence Fishburne",
    "Hugo Weaving",
    "Gloria Foster",
    "Joe Pantoliano",
    "Antonio Banderas",
    "Brad Pitt"
  ],
  "genero" : "ciencia ficcion"
}
```

8) Actualizar el documento referido a la película «Matrix» añadiendo al array actores los valores «Joe Pantoliano», «Brad Pitt» y «Natalie Portman», en caso de que no se encuentren, y si se encuentran, no se hace nada.

Figura 10

```

> db.media.update({"_id" : ObjectId("54ddc7929d235c1da3b376e0")}, {"$addToSet": {"actores": {"$each": ["Joe Pantoliano", "Brad Pitt", "Natalie Portman"]}}})
WriteResult({"nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.media.findOne({"_id" : ObjectId("54ddc7929d235c1da3b376e0")})
{
  "_id" : ObjectId("54ddc7929d235c1da3b376e0"),
  "tipo" : "DUD",
  "Titulo" : "Matrix",
  "estreno" : 1999,
  "actores" : [
    "Keanu Reeves",
    "Carry-Anne Moss",
    "Laurence Fishburne",
    "Hugo Weaving",
    "Gloria Foster",
    "Joe Pantoliano",
    "Antonio Banderas",
    "Brad Pitt",
    "Natalie Portman"
  ],
  "genero" : "ciencia ficcion"
}

```

9) Actualizar el documento referido a la película «Matrix» eliminando del array el primer y último actor.

Figura 11

```

> db.media.findOne({"_id" : ObjectId("54ddc7929d235c1da3b376e0")})
{
  "_id" : ObjectId("54ddc7929d235c1da3b376e0"),
  "tipo" : "DUD",
  "Titulo" : "Matrix",
  "estreno" : 1999,
  "actores" : [
    "Keanu Reeves",
    "Carry-Anne Moss",
    "Laurence Fishburne",
    "Hugo Weaving",
    "Gloria Foster",
    "Joe Pantoliano",
    "Antonio Banderas",
    "Brad Pitt"
  ],
  "genero" : "ciencia ficcion"
}
> db.media.update({"_id" : ObjectId("54ddc7929d235c1da3b376e0")}, {"$pop": {"actores": -1}})
WriteResult({"nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.media.findOne({"_id" : ObjectId("54ddc7929d235c1da3b376e0")})
{
  "_id" : ObjectId("54ddc7929d235c1da3b376e0"),
  "tipo" : "DUD",
  "Titulo" : "Matrix",
  "estreno" : 1999,
  "actores" : [
    "Carry-Anne Moss",
    "Laurence Fishburne",
    "Hugo Weaving",
    "Gloria Foster",
    "Joe Pantoliano",
    "Antonio Banderas",
    "Brad Pitt"
  ],
  "genero" : "ciencia ficcion"
}

```

10) Actualizar el documento referido a la película «Matrix» añadiendo al array «actores» los valores «Joe Pantoliano» y «Antonio Banderas».

Figura 12

```
> db.media.update<<"_id" : ObjectId<"54ddc7929d235c1da3b376e0">>,<{"$pushAll":{"actores":["Antonio Banderas","Joe Pantolino"]}}>
WriteResult<<"nMatched" : 1, "nUpserted" : 0, "nModified" : 1 >>
> db.media.findOne<<"_id" : ObjectId<"54ddc7929d235c1da3b376e0">>
{
  "_id" : ObjectId<"54ddc7929d235c1da3b376e0">,
  "tipo" : "DUD",
  "Titulo" : "Matrix",
  "estreno" : 1999,
  "actores" : [
    "Carry-Anne Moss",
    "Laurence Fishburne",
    "Hugo Weaving",
    "Gloria Foster",
    "Joe Pantoliano",
    "Antonio Banderas",
    "Brad Pitt",
    "Antonio Banderas",
    "Joe Pantolino"
  ],
  "genero" : "ciencia ficcion"
}
```

11) Actualizar el documento referido a la película «Matrix» eliminando todas las apariciones en el array «actores» de los valores «Joe Pantoliano» y «Antonio Banderas».

Figura 13

```
> doc4=<"tipo": "DUD", "Titulo": "Blade Runner", "estreno": 1982 >
< "tipo" : "DUD", "Titulo" : "Blade Runner", "estreno" : 1982 >
> doc5=<"tipo": "DUD", "Titulo": "Toy Story 3", "estreno": 2010 >
< "tipo" : "DUD", "Titulo" : "Toy Story 3", "estreno" : 2010 >
> db.media.insert<[doc4,doc5]>
BulkWriteResult<<
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 2,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
>>
```

12) Actualizar el documento referido al disco «Recuerdos» y añadir una nueva canción al array «canciones»:

```
{ "cancion": 5,
  "titulo": "El atardecer",
  "longitud": "6:50"
}
```

Figura 14

```
> db.media.find(<<"estreno":{"$gt":2000},{"actores":0}>>
{ "_id" : ObjectId("54dfb8534fa9e7f2f218dde4"), "tipo" : "DVD", "Titulo" : "Toy
Story 3", "estreno" : 2010 }
> db.media.find(<<"estreno":{"$gte":1999},{"actores":0}>>
{ "_id" : ObjectId("54dfa30e4fa9e7f2f218dde8"), "tipo" : "DVD", "Titulo" : "Matr
ix", "estreno" : 1999 }
{ "_id" : ObjectId("54dfb8534fa9e7f2f218dde4"), "tipo" : "DVD", "Titulo" : "Toy
Story 3", "estreno" : 2010 }
> db.media.find(<<"estreno":{"$lt":1999},{"actores":0}>>
{ "_id" : ObjectId("54dfb8534fa9e7f2f218dde9"), "tipo" : "DVD", "Titulo" : "Blad
e Runner", "estreno" : 1982 }
> db.media.find(<<"estreno":{"$lte":1999},{"actores":0}>>
{ "_id" : ObjectId("54dfa30e4fa9e7f2f218dde8"), "tipo" : "DVD", "Titulo" : "Matr
ix", "estreno" : 1999 }
{ "_id" : ObjectId("54dfb8534fa9e7f2f218dde9"), "tipo" : "DVD", "Titulo" : "Blad
e Runner", "estreno" : 1982 }
> db.media.find(<<"estreno":{"$gte":1999,"$lt":2010},{"actores":0}>>
{ "_id" : ObjectId("54dfa30e4fa9e7f2f218dde8"), "tipo" : "DVD", "Titulo" : "Matr
ix", "estreno" : 1999 }
```

13) Actualizar el documento referido al disco «Recuerdos» de manera que la canción «El atardecer» tenga asignado el número 3 en vez de 5.

Figura 15

```
> db.media.find(<<"tipo":"libro", "autor":{"$ne":"Camilo José Cela"}>>
{ "_id" : ObjectId("54dfa30e4fa9e7f2f218dde6"), "tipo" : "libro", "titulo" : "Ja
va para todos", "ISBN" : "987-1-2344-5334-8", "editorial" : "Anaya", "autor" : [
"Pepe Caballero", "Isabel Sanz", "Tinoteo Marino" ], "capitulos" : [ { "capitul
o" : 1, "titulo" : "Primeros pasos en Java", "longitud" : 20 }, { "capitulo" : 2
, "titulo" : "Primeros pasos en Java", "longitud" : 25 } ] }
```

14) Actualizar el documento referido al disco «Recuerdos», de manera que en una sola operación se cambie el nombre del artista a «Los piratillas» y se muestre el documento resultante.

Figura 16

```
> db.media.find(<<"estreno":{"$in":[1999,2005,2006]},{"actores":0}>>
{ "_id" : ObjectId("54dfa30e4fa9e7f2f218dde8"), "tipo" : "DVD", "Titulo" : "Matr
ix", "estreno" : 1999 }
```


15) Renombrar el nombre de la colección «media» a «multi-media».

Figura 17

```
> db.media.find(<<"estreno":<{"$nin": [1999, 2005, 2006]}, "tipo": "DVD">>, <<"actores": 0>>
>
< "_id" : ObjectId("54dfb8534fa9e7f2f218dde9"), "tipo" : "DVD", "Titulo" : "Blade Runner", "estreno" : 1982 >
< "_id" : ObjectId("54dfb8534fa9e7f2f218ddea"), "tipo" : "DVD", "Titulo" : "Toy Story 3", "estreno" : 2010 >
```

Soluciones a los ejercicios del capítulo 4

1) Insertar los documentos dados en una base de datos llamada «media» en una única operación.

Figura 18

```
> var doc1 = { "tipo": "libro",
...   "titulo": "Java para todos",
...   "ISBN": "987-1-2344-5334-8",
...   "editorial": "Anaya",
...   "autor": ["Pepe Caballero", "Isabel Sanz", "Timoteo Marino"],
...   "capitulos": [
...     < "capitulo": 1,
...       "titulo": "Primeros pasos en Java",
...       "longitud": 20
...     >,
...     < "capitulo": 2,
...       "titulo": "Primeros pasos en Java",
...       "longitud": 25
...     >
...   ]
... }
> var doc2 = { "tipo": "CD",
...   "artista": "Los piratas",
...   "titulo": "Recuerdos",
...   "canciones": [
...     < "cancion": 1,
...       "titulo": "Adios mi barco",
...       "longitud": "3:20"
...     >,
...     < "cancion": 2,
...       "titulo": "Pajaritos",
...       "longitud": "4:15"
...     >
...   ]
... }
```

Figura 19

```

doc3=< "tipo": "DUD",
... "Titulo": "Matrix",
... "estreno": 1999,
... "actores": [
... "Keanu Reeves",
... "Carry-Anne Moss",
... "Laurence Fishburne",
... "Hugo Weaving",
... "Gloria Foster",
... "Joe Pantoliano"
... ]
... >
{
  "tipo": "DUD",
  "Titulo": "Matrix",
  "estreno": 1999,
  "actores": [
    "Keanu Reeves",
    "Carry-Anne Moss",
    "Laurence Fishburne",
    "Hugo Weaving",
    "Gloria Foster",
    "Joe Pantoliano"
  ]
}
> db.media.insert([doc1,doc2,doc3])
BulkWriteResult<<
  "writeErrors": [ ],
  "writeConcernErrors": [ ],
  "nInserted": 3,
  "nUpserted": 0,
  "nMatched": 0,
  "nModified": 0,
  "nRemoved": 0,
  "upserted": [ ]
>>
> db.media.find<>
{ "_id" : ObjectId("54ddc7929d235c1da3b376de"), "tipo" : "libro", "titulo" : "Java para todos", "ISBN" : "987-1-2344-5334-8", "editorial" : "Anaya", "autor" : [ "Pepe Caballero", "Isabel Sanz", "Timoteo Marino" ], "capitulos" : [ < "capitulo" : 1, "titulo" : "Primeros pasos en Java", "longitud" : 25 > ], < "capitulo" : 2, "titulo" : "Primeros pasos en Java", "longitud" : 25 > ] }
{ "_id" : ObjectId("54ddc7929d235c1da3b376df"), "tipo" : "CD", "artista" : "Los Piratas", "titulo" : "Recuerdos", "canciones" : [ < "cancion" : 1, "titulo" : "A Dios mi barco", "longitud" : "3:20" >, < "cancion" : 2, "titulo" : "Pajaritos", "longitud" : "4:15" > ] }
{ "_id" : ObjectId("54ddc7929d235c1da3b376e0"), "tipo" : "DUD", "Titulo" : "Matrix", "estreno" : 1999, "actores" : [ "Keanu Reeves", "Carry-Anne Moss", "Laurence Fishburne", "Hugo Weaving", "Gloria Foster", "Joe Pantoliano" ] }

```

2) Del documento que hace referencia a la película «Matrix» recuperar el array de actores.

Figura 20

```

> db.media.find(<<"Titulo":"Matrix">>,<<"actores":1>>)
{ "_id" : ObjectId("54dfa39e4fa9e7f2f218dde8"), "actores" : [ "Keanu Reeves", "Carry-Anne Moss", "Laurence Fishburne", "Hugo Weaving", "Gloria Foster", "Joe Pantoliano" ] }

```

3) Del documento que hace referencia a la película «Matrix» recuperar todos los campos de información excepto el array de actores.

Figura 21

```
> db.media.find(<<"Titulo":"Matrix">>,<<"actores":0>>)
< {"_id" : ObjectId("54dfa30e4fa9e7f2f218dde8"), "tipo" : "DVD", "Titulo" : "Matrix", "estreno" : 1999 }
>
```

4) Del documento que hace referencia a la película «Matrix» recuperar un único documento en el que aparezcan solo los campos «tipo» y «título».

Figura 22

```
> db.media.findOne(<<"Titulo":"Matrix">>,<<"tipo":1,"Titulo":1>>)
< {"_id" : ObjectId("54dfa30e4fa9e7f2f218dde8"), "tipo" : "DVD", "Titulo" : "Matrix"}
>
```

5) Recuperar todos los documentos que sean de tipo «libro» y editorial «Anaya» mostrando solo el array «capítulos».

Figura 23

```
> db.media.find(<<"tipo":"libro","editorial":"anaya">>,<<"capitulos":1>>)
< {"_id" : ObjectId("54dfa30e4fa9e7f2f218dde6"), "capitulos" : [ < "capitulo" : 1, "titulo" : "Primeros pasos en Java", "longitud" : 20 >, < "capitulo" : 2, "titulo" : "Primeros pasos en Java", "longitud" : 25 > ] }
>
```

6) Recuperar todos los documentos referidos a canciones que tengan una canción que se denomine «Pajaritos».

Figura 24

```
> db.media.find(<<"canciones.titulo":"Pajaritos">>)
< {"_id" : ObjectId("54dfa30e4fa9e7f2f218dde7"), "tipo" : "CD", "Artista" : "Los Piratas", "Titulo" : "Recuerdos", "canciones" : [ < "cancion" : 1, "titulo" : "A dios mi barco", "longitud" : "3:20" >, < "cancion" : 2, "titulo" : "Pajaritos", "longitud" : "4:15" > ] }
>
```

7) Recuperar todos los documentos en los que Timoteo Marino es autor de un libro.

Figura 25

```
> db.media.find(<<"autor":"Tinoteo Marino">>)
< "_id" : ObjectId("54dfa39e4fa9e7f2f218dde6"), "tipo" : "libro", "titulo" : "Java para todos", "ISBN" : "987-1-2344-5334-8", "editorial" : "Anaya", "autor" : [ "Pepe Caballero", "Isabel Sanz", "Tinoteo Marino" ], "capitulos" : [ < "capitulo" : 1, "titulo" : "Primeros pasos en Java", "longitud" : 20 >, < "capitulo" : 2, "titulo" : "Primeros pasos en Java", "longitud" : 25 > ] >
```

8) Recuperar todos los documentos de la colección «media» ordenados de manera decreciente por el campo «tipo».

Figura 26

```
> db.media.find(<<>>).sort(<<"tipo":-1>>)
< "_id" : ObjectId("54dfa39e4fa9e7f2f218dde6"), "tipo" : "libro", "titulo" : "Java para todos", "ISBN" : "987-1-2344-5334-8", "editorial" : "Anaya", "autor" : [ "Pepe Caballero", "Isabel Sanz", "Tinoteo Marino" ], "capitulos" : [ < "capitulo" : 1, "titulo" : "Primeros pasos en Java", "longitud" : 20 >, < "capitulo" : 2, "titulo" : "Primeros pasos en Java", "longitud" : 25 > ] >
< "_id" : ObjectId("54dfa39e4fa9e7f2f218dde8"), "tipo" : "DVD", "Titulo" : "Matrix", "estreno" : 1999, "actores" : [ "Keanu Reeves", "Carry-Anne Moss", "Laurence Fishburne", "Hugo Weaving", "Gloria Foster", "Joe Pantoliano" ] >
< "_id" : ObjectId("54dfa39e4fa9e7f2f218dde7"), "tipo" : "CD", "Artista" : "Los Piratas", "Titulo" : "Recuerdos", "canciones" : [ < "cancion" : 1, "titulo" : "A Dios ni barco", "longitud" : "3:20" >, < "cancion" : 2, "titulo" : "Pajaritos", "longitud" : "4:15" > ] >
```

9) Recuperar todos los documentos de la colección «media» ordenados de manera decreciente por el campo «tipo». Mostrar solo dos resultados.

Figura 27

```
> db.media.find(<<>>).sort(<<"tipo":1>>).limit(2)
< "_id" : ObjectId("54dfa39e4fa9e7f2f218dde7"), "tipo" : "CD", "Artista" : "Los Piratas", "Titulo" : "Recuerdos", "canciones" : [ < "cancion" : 1, "titulo" : "A Dios ni barco", "longitud" : "3:20" >, < "cancion" : 2, "titulo" : "Pajaritos", "longitud" : "4:15" > ] >
< "_id" : ObjectId("54dfa39e4fa9e7f2f218dde8"), "tipo" : "DVD", "Titulo" : "Matrix", "estreno" : 1999, "actores" : [ "Keanu Reeves", "Carry-Anne Moss", "Laurence Fishburne", "Hugo Weaving", "Gloria Foster", "Joe Pantoliano" ] >
> =
```

10) Recuperar todos los documentos de la colección «media» ordenados de manera decreciente por el campo «tipo». Saltarse el primer resultado.

Figura 28

```

> db.media.find(<>).sort(<<"tipo":1>>).skip(1)
< "_id" : ObjectId<"54dfa30e4fa9e7f2f218dde8">, "tipo" : "DVD", "Titulo" : "Matrix", "estreno" : 1999, "actores" : [ "Keanu Reeves", "Carry-Anne Moss", "Laurence Fishburne", "Hugo Weaving", "Gloria Foster", "Joe Pantoliano" ] >
< "_id" : ObjectId<"54dfa30e4fa9e7f2f218dde6">, "tipo" : "libro", "titulo" : "Java para todos", "ISBN" : "987-1-2344-5334-8", "editorial" : "Anaya", "Autor" : [ "Pepe Caballero", "Isabel Sanz", "Timoteo Marino" ], "capitulos" : [ < "capitulo" : 1, "titulo" : "Primeros pasos en Java", "longitud" : 20 >, < "capitulo" : 2, "titulo" : "Primeros pasos en Java", "longitud" : 25 > ] >

```

11) Recuperar todos los documentos de la colección «media» ordenados de manera decreciente por el campo «tipo». Recuperar solo dos resultados y saltarse los dos primeros resultados.

Figura 29

```

> db.media.find(<>).sort(<<"tipo":1>>).limit(2).skip(2)
< "_id" : ObjectId<"54dfa30e4fa9e7f2f218dde6">, "tipo" : "libro", "titulo" : "Java para todos", "ISBN" : "987-1-2344-5334-8", "editorial" : "Anaya", "Autor" : [ "Pepe Caballero", "Isabel Sanz", "Timoteo Marino" ], "capitulos" : [ < "capitulo" : 1, "titulo" : "Primeros pasos en Java", "longitud" : 20 >, < "capitulo" : 2, "titulo" : "Primeros pasos en Java", "longitud" : 25 > ] >

```

12) Añadir los siguientes documentos a la colección media:

```

{ "tipo": "DVD", "Titulo": "Blade Runner", "estreno": 1982 }
{ "tipo": "DVD", "Titulo": "Toy Story 3", "estreno": 2010 }

```

Figura 30

```

> doc4=<"tipo": "DVD", "Titulo": "Blade Runner", "estreno": 1982 >
< "tipo" : "DVD", "Titulo" : "Blade Runner", "estreno" : 1982 >
> doc5=<"tipo": "DVD", "Titulo": "Toy Story 3", "estreno": 2010 >
< "tipo" : "DVD", "Titulo" : "Toy Story 3", "estreno" : 2010 >
> db.media.insert([doc4,doc5])
BulkWriteResult<<
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 2,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
>>

```

Realizar las siguientes consultas:

- Recuperar los documentos sobre películas cuya fecha de estreno sea mayor que 2000. En los resultados no mostrar el array de actores.
- Recuperar los documentos sobre películas cuya fecha de estreno sea mayor o igual que 1999. En los resultados no mostrar el array de actores.
- Recuperar los documentos sobre películas cuya fecha de estreno sea menor que 1999. En los resultados no mostrar el array de actores.
- Recuperar los documentos sobre películas cuya fecha de estreno sea menor o igual que 1999. En los resultados no mostrar el array de actores.
- Recuperar los documentos sobre películas cuya fecha de estreno sea mayor o igual que 1999 y menor que 2010. En los resultados no mostrar el array de actores

Figura 31

```

> db.media.find(<<"estreno":{"$gt":2000}>>,{"actores":0})
< "_id" : ObjectId("54dfb8534fa9e7f2f218dde8"), "tipo" : "DVD", "Titulo" : "Toy
Story 3", "estreno" : 2010 >
> db.media.find(<<"estreno":{"$gte":1999}>>,{"actores":0})
< "_id" : ObjectId("54dfa30e4fa9e7f2f218dde8"), "tipo" : "DVD", "Titulo" : "Matr
ix", "estreno" : 1999 >
< "_id" : ObjectId("54dfb8534fa9e7f2f218dde8"), "tipo" : "DVD", "Titulo" : "Toy
Story 3", "estreno" : 2010 >
> db.media.find(<<"estreno":{"$lt":1999}>>,{"actores":0})
< "_id" : ObjectId("54dfb8534fa9e7f2f218dde9"), "tipo" : "DVD", "Titulo" : "Blad
e Runner", "estreno" : 1982 >
> db.media.find(<<"estreno":{"$lte":1999}>>,{"actores":0})
< "_id" : ObjectId("54dfa30e4fa9e7f2f218dde8"), "tipo" : "DVD", "Titulo" : "Matr
ix", "estreno" : 1999 >
< "_id" : ObjectId("54dfb8534fa9e7f2f218dde9"), "tipo" : "DVD", "Titulo" : "Blad
e Runner", "estreno" : 1982 >
> db.media.find(<<"estreno":{"$gte":1999,"$lt":2010}>>,{"actores":0})
< "_id" : ObjectId("54dfa30e4fa9e7f2f218dde8"), "tipo" : "DVD", "Titulo" : "Matr
ix", "estreno" : 1999 >

```

13) Recuperar todos los documentos sobre libros de manera que el autor no sea «Camilo José Cela».

Figura 32

```
> db.media.find(<<"tipo":"libro", "autor":{"$ne":"Camilo José Cela"}>>)
< "_id" : ObjectId("54dfa30e4fa9e7f2f218dde6"), "tipo" : "libro", "titulo" : "Java para todos", "ISBN" : "987-1-2344-5334-8", "editorial" : "Anaya", "autor" : [ "Pepe Caballero", "Isabel Sanz", "Tinoteo Marino" ], "capitulos" : [ < "capitulo" : 1, "titulo" : "Primeros pasos en Java", "longitud" : 20 >, < "capitulo" : 2, "titulo" : "Primeros pasos en Java", "longitud" : 25 > ] >
```

14) Recuperar todos los documentos sobre películas que se hayan estrenado en alguno de los años 1999, 2005 y 2006. En los resultados no mostrar el array de actores.

Figura 33

```
> db.media.find(<<"estreno":{"$in":[1999,2005,2006]}>>,<<"actores":0>>)
< "_id" : ObjectId("54dfa30e4fa9e7f2f218dde8"), "tipo" : "DUD", "Titulo" : "Matrix", "estreno" : 1999 >
```

15) Recuperar todos los documentos sobre películas que no se hayan estrenado en los años 1999, 2005 y 2006. En los resultados no mostrar el array de actores.

Figura 34

```
> db.media.find(<<"estreno":{"$nin":[1999,2005,2006]}>>,"tipo":"DUD"),<<"actores":0>>)
< "_id" : ObjectId("54dfb8534fa9e7f2f218dde9"), "tipo" : "DUD", "Titulo" : "Blade Runner", "estreno" : 1982 >
< "_id" : ObjectId("54dfb8534fa9e7f2f218dde8"), "tipo" : "DUD", "Titulo" : "Toy Story 3", "estreno" : 2010 >
```

16) Recuperar todos los documentos sobre películas que se hayan estrenado en los años 1999 y 2005 exactamente. En los resultados no mostrar el array de actores.

Figura 35

```
> db.media.find(<<"estreno":{"$all":[1999,2005]}>>,"tipo":"DUD"),<<"actores":0>>)
```

17) Recuperar todos los documentos sobre libros que hayan sido escritos por Pepe Caballero e Isabel Sanz y que además

entre los títulos de sus capítulos haya alguno que se denomine «Bucles».

Figura 36

```
> db.media.find(<<"autor":{"$all":["Pepe Caballero","Isabel Sanz"]}, "capitulos.titulo":{"$in":["Bucles"]}>>)
< "_id" : ObjectId("54dfc6824fa9e7f2f218dde8"), "tipo" : "libro", "titulo" : "Java para todos", "ISBN" : "987-1-2344-5334-8", "editorial" : "Anaya", "autor" : ["Pepe Caballero", "Isabel Sanz", "Tinoteo Marino" ], "capitulos" : [ < "capitulo" : 1, "titulo" : "Primeros pasos en Java", "longitud" : 28 >, < "capitulo" : 2, "titulo" : "Bucles", "longitud" : 25 > ] ] >
```

18) Recuperar todos los documentos que tomen en la clave «Título» el valor «Recuerdos» o que tome en la clave «estreno», el valor «1999», y que tome en la clave tipo «DVD».

Figura 37

```
> db.media.find(<<"tipo":"DVD", "$or":{"Titulo":"Recuerdos"},<"estreno":1999}>>)
< "_id" : ObjectId("54dfa39e4fa9e7f2f218dde8"), "tipo" : "DVD", "Titulo" : "Matrix", "estreno" : 1999, "actores" : [ "Keanu Reeves", "Carry-Anne Moss", "Laurence Fishburne", "Hugo Weaving", "Gloria Foster", "Joe Pantoliano" ] >
```

También podría hacerse usando «\$and»:

Figura 38

```
> db.media.find(<<"$and":{"tipo":"DVD"},<"$or":{"Titulo":"Recuerdos"},<"estreno":1999}>>)}>)
< "_id" : ObjectId("54dfa39e4fa9e7f2f218dde8"), "tipo" : "DVD", "Titulo" : "Matrix", "estreno" : 1999, "actores" : [ "Keanu Reeves", "Carry-Anne Moss", "Laurence Fishburne", "Hugo Weaving", "Gloria Foster", "Joe Pantoliano" ] >
```

19) Considerar el documento acerca de la película «Matrix», y recuperar del array de actores:

- Los 3 primeros actores.
- Los últimos 3 actores.
- 3 actores saltándose los 2 dos primeros actores.
- 4 actores saltándose los 5 últimos actores.

Figura 39

```
> db.media.find(<<"Titulo":"Matrix">>,<<"actores":{"$slice":3}>>)
< "_id" : ObjectId("54dfa30e4fa9e7f2f218dde8"), "tipo" : "DUD", "Titulo" : "Matr
ix", "estreno" : 1999, "actores" : [ "Keanu Reeves", "Carry-Anne Moss", "Laurence
e Fishburne" ] >
> db.media.find(<<"Titulo":"Matrix">>,<<"actores":{"$slice":-3}>>)
< "_id" : ObjectId("54dfa30e4fa9e7f2f218dde8"), "tipo" : "DUD", "Titulo" : "Matr
ix", "estreno" : 1999, "actores" : [ "Hugo Weaving", "Gloria Foster", "Joe Panto
liano" ] >
> db.media.find(<<"Titulo":"Matrix">>,<<"actores":{"$slice":[2,3]}>>)
< "_id" : ObjectId("54dfa30e4fa9e7f2f218dde8"), "tipo" : "DUD", "Titulo" : "Matr
ix", "estreno" : 1999, "actores" : [ "Laurence Fishburne", "Hugo Weaving", "Glor
ia Foster" ] >
> db.media.find(<<"Titulo":"Matrix">>,<<"actores":{"$slice":[-5,4]}>>)
< "_id" : ObjectId("54dfa30e4fa9e7f2f218dde8"), "tipo" : "DUD", "Titulo" : "Matr
ix", "estreno" : 1999, "actores" : [ "Carry-Anne Moss", "Laurence Fishburne", "H
ugo Weaving", "Gloria Foster" ] >
>
```

20) Recuperar los documentos referidos a películas que en su campo «estreno» tengan un valor par. No mostrar el array «actores».

Figura 40

```
> db.media.find(<<"estreno":{"$mod":[2,0]}>>,<<"actores":0>>)
< "_id" : ObjectId("54dfb8534fa9e7f2f218dde9"), "tipo" : "DUD", "Titulo" : "Blad
e Runner", "estreno" : 1982 >
< "_id" : ObjectId("54dfb8534fa9e7f2f218dde9"), "tipo" : "DUD", "Titulo" : "Toy
Story 3", "estreno" : 2010 >
```

21) Recuperar los documentos referidos a películas que en su campo «estreno» tengan un valor impar. No mostrar el array «actores».

Figura 41

```
> db.media.find(<<"estreno":{"$mod":[2,1]}>>,<<"actores":0>>)
< "_id" : ObjectId("54dfa30e4fa9e7f2f218dde8"), "tipo" : "DUD", "Titulo" : "Matr
ix", "estreno" : 1999 >
```

22) Recuperar todos los documentos referidos a canciones tales que el número de canciones sea exactamente 2.

Figura 42

```
> db.media.find(<<"canciones":{"$size":2}>>)
< "_id" : ObjectId("54dfa30e4fa9e7f2f218dde7"), "tipo" : "CD", "artista" : "Los
piratas", "Titulo" : "Recuerdos", "canciones" : [ < "cancion" : 1, "titulo" : "A
dios mi barco", "longitud" : "3:20" >, < "cancion" : 2, "titulo" : "Pajaritos",
"longitud" : "4:15" > ] >
```

23) Recuperar todos los documentos que tengan un array de actores.

Figura 43

```
> db.media.find(<<"actores":{"$exists": true}>>)
< {"_id": ObjectId("54dfa30e4fa9e7f2f218dde8"), "tipo": "DVD", "Titulo": "Matrix", "estreno": 1999, "actores": [ "Keanu Reeves", "Carry-Anne Moss", "Laurence Fishburne", "Hugo Weaving", "Gloria Foster", "Joe Pantoliano" ] } >
```

24) Recuperar todos los documentos que no tengan un array de actores.

Figura 44

```
> db.media.find(<<"actores":{"$exists": false}>>)
< {"_id": ObjectId("54dfa30e4fa9e7f2f218dde7"), "tipo": "CD", "Artista": "Los Piratas", "Titulo": "Recuerdos", "canciones": [ < "cancion": 1, "titulo": "Dios mi barco", "longitud": "3:20" >, < "cancion": 2, "titulo": "Pajaritos", "longitud": "4:15" > ] } >
< {"_id": ObjectId("54dfb8534fa9e7f2f218dde9"), "tipo": "DVD", "Titulo": "Blade Runner", "estreno": 1982 } >
< {"_id": ObjectId("54dfb8534fa9e7f2f218ddea"), "tipo": "DVD", "Titulo": "Toy Story 3", "estreno": 2010 } >
< {"_id": ObjectId("54dfc6824fa9e7f2f218ddeb"), "tipo": "libro", "titulo": "Java para todos", "ISBN": "987-1-2344-5334-8", "editorial": "Anaya", "Autor": [ "Pepe Caballero", "Isabel Sanz", "Tinoteo Marino" ], "capitulos": [ < "capitulo": 1, "titulo": "Primeros pasos en Java" "longitud": 20 >, < "capitulo": 2, "titulo": "Bucles", "longitud": 25 > ] } >
```

25) Considerar la siguiente tabla, que asigna a cada tipo de datos BSON con un valor numérico:

Figura 45

#	Data Type	#	Data Type
-1	MiniKey	11	Regular Expression
1	Double	13	JavaScript Code
2	Character string (UTF8)	14	Symbol
3	Embedded object	15	JavaScript Code with scope
4	Embedded array	16	32-bit integer
5	Binary Data	17	Timestamp
7	Object ID	18	64-bit integer
8	Boolean type	127	MaxKey
9	Date type	255	Min Key
10	Null type		

Recuperar todos los documentos que tienen un campo denominado «canciones» cuyo valor sea del tipo un documento embebido.

Figura 46

```
> db.media.find(<<"canciones":<{"$type":3}>>
< "_id" : ObjectId("54dfa30e4fa9e7f2f218dde7"), "tipo" : "CD", "Artista" : "Los
piratas", "Titulo" : "Recuerdos", "canciones" : [ < "cancion" : 1, "titulo" : "A
diós mi barco", "longitud" : "3:20" >, < "cancion" : 2, "titulo" : "Pajaritos",
"longitud" : "4:15" > ] >
>
```

26) Insertar el siguiente documento:

```
{ "tipo": "CD",
  "Artista": "Los piratas",
  "Titulo": "Recuerdos",
  "canciones": [
    { "cancion": 1,
      "titulo": "Adiós mi barco",
      "longitud": "3:20" },
    { "cancion": 3,
      "titulo": "Pajaritos",
      "longitud": "4:15" } ] }
```

Recuperar todos los documentos sobre discos en los que se dan exactamente las siguientes condiciones: existe una canción denominada «Pajaritos» y el número de canción es el 2.

Figura 47

```
> db.media.find(<<"canciones":<{"$elemMatch": <{"cancion":2, "titulo":"Pajaritos"}>}>
< "_id" : ObjectId("54dfa30e4fa9e7f2f218dde7"), "tipo" : "CD", "Artista" : "Los
piratas", "Titulo" : "Recuerdos", "canciones" : [ < "cancion" : 1, "titulo" : "A
diós mi barco", "longitud" : "3:20" >, < "cancion" : 2, "titulo" : "Pajaritos",
"longitud" : "4:15" > ] >
>
```

27) Recuperar todos los documentos sobre discos en los que no se dan exactamente las siguientes condiciones: existe una canción denominada «Pajaritos» y el número de canción es el 2.

Figura 48

```

> db.media.find(<<"canciones":<{"$not":{"$elemMatch":<{"cancion":2,"titulo":"Pajaritos"}}>>>>)
< "_id" : ObjectId("54dfa30e4fa9e7f2f218dde8"), "tipo" : "DVD", "Titulo" : "Matrix", "estreno" : 1999, "actores" : [ "Keanu Reeves", "Carry-Anne Moss", "Laurence Fishburne", "Hugo Weaving", "Gloria Foster", "Joe Pantoliano" ] >
< "_id" : ObjectId("54dfb8534fa9e7f2f218dde9"), "tipo" : "DVD", "Titulo" : "Blade Runner", "estreno" : 1982 >
< "_id" : ObjectId("54dfb8534fa9e7f2f218dde9"), "tipo" : "DVD", "Titulo" : "Toy Story 3", "estreno" : 2010 >
< "_id" : ObjectId("54dfc6824fa9e7f2f218dde8"), "tipo" : "libro", "titulo" : "Java para todos", "ISBN" : "987-1-2344-5334-8", "editorial" : "Anaya", "autor" : [ "Pepe Caballero", "Isabel Sanz", "Tinoteo Marino" ], "capitulos" : [ < "capitulo" : 1, "titulo" : "Primeros pasos en Java", "longitud" : 20 >, < "capitulo" : 2, "titulo" : "Bucles", "longitud" : 25 > ] >

```

28) Encontrar los DVD que sean más antiguos que 1995.

Figura 49

```

> db.media.find(<<"tipo":"DVD","estreno":<{"$lt":1995}>>>)
< "_id" : ObjectId("54dfb8534fa9e7f2f218dde9"), "tipo" : "DVD", "Titulo" : "Blade Runner", "estreno" : 1982 >
> db.media.find(<<"tipo":"DVD","$where":"this.estreno <1995">>>)
< "_id" : ObjectId("54dfb8534fa9e7f2f218dde9"), "tipo" : "DVD", "Titulo" : "Blade Runner", "estreno" : 1982 >
> db.media.find(<<"this.estreno <1995">>>)
< "_id" : ObjectId("54dfb8534fa9e7f2f218dde9"), "tipo" : "DVD", "Titulo" : "Blade Runner", "estreno" : 1982 >
f=function()<return this.estreno<1995;>
function()<return this.estreno<1995;>
> db.media.find(f)
< "_id" : ObjectId("54dfb8534fa9e7f2f218dde9"), "tipo" : "DVD", "Titulo" : "Blade Runner", "estreno" : 1982 >

```

29) Encontrar todos los documentos en los que en el campo «Artista» aparezca la palabra «piratas».

Figura 50

```

> db.media.find(<<"artista":<{$regex:"piratas"}>>>)
< "_id" : ObjectId("54dfa30e4fa9e7f2f218dde7"), "tipo" : "CD", "artista" : "Los piratas", "Titulo" : "Recuerdos", "canciones" : [ < "cancion" : 1, "titulo" : "A Dios ni barco", "longitud" : "3:20" >, < "cancion" : 2, "titulo" : "Pajaritos", "longitud" : "4:15" > ] >

```

30) Encontrar todos los documentos en los que que en el campo «Artista» aparezca la palabra «piratas» y además tengan un campo «Titulo».

Figura 51

```
> db.media.find(<<"artista": <<regex:"piratas">>,"titulo":<<"%exists":true>>>
< "_id" : ObjectId<"54dfa30e4fa9e7f2f218dde7">, "tipo" : "CD", "artista" : "Los
piratas", "titulo" : "Recuerdos", "canciones" : [ <<"cancion" : 1, "titulo" : "A
dios ni barco", "longitud" : "3:20">, <<"cancion" : 2, "titulo" : "Pajaritos",
"longitud" : "4:15"> ] >
>
```

Soluciones a los ejercicios del capítulo 5

1) Seleccionar los documentos de tipo «CD» de manera que solo se muestren en dichos documentos los campos «Artista», «TituloCanción», y un nuevo campo «TitulosCanciones» que contenga un array con las canciones del disco.

Figura 52

```
> db.media.aggregate(<<"$match":<<"tipo":<"CD">>>,< "$project": <<"tipo":1, "artista":1, "TitulosCanciones":<"$canciones.titulo">>>
< "_id" : ObjectId<"5524f03c0ed06f6d9f4555c0">, "tipo" : "CD", "artista" : "Los
piratas", "TitulosCanciones" : [ "Adios ni barco", "Pajaritos" ] >
```

2) Añadir las siguientes películas:

```
{ "tipo": "DVD",      { "tipo": "DVD",      { "tipo": "DVD",
  "Titulo": "Blade Runner",  "Titulo": "Batman",  "Titulo":
"Superman",
  "estreno":1982          "estreno": 1999          "estreno": 1999
}                        }                        }
```

Seleccionar todos los documentos de tipo «DVD» y calcular cuántas películas hay de cada año de estreno, mostrando el año de estreno y el número de películas de cada año.

Figura 53

```
> db.media.aggregate( < {$match: < {tipo: "DVD"} >}, < {$group: < { "_id": "$estreno", count:
t: {$sum: 1} > } >, < {$project: < { "_id": 0, estreno: "$_id", total: "$count" } > } >
< "estreno": 1982, "total": 1 >
< "estreno": 1999, "total": 3 >
```

3) Seleccionar el documento sobre la película «Matrix» y crear un documento por cada uno de los actores que intervienen. En los documentos resultantes solo se mostrará el título y el actor.

Figura 54

```
> db.media.aggregate( < {$match: < {Titulo: "Matrix"} >}, < {$project: < {Titulo: 1, actores
: 1} > } >
< "_id": ObjectId("5524f04c0ed06f6d9f4555c1"), "Titulo": "Matrix", "actores":
[ "Keanu Reeves", "Carry-Anne Moss", "Laurence Fishburne", "Hugo Weaving", "Glo
ria Foster", "Joe Pantoliano" ] >
> db.media.aggregate( < {$match: < {Titulo: "Matrix"} >}, < {$project: < {Titulo: 1, actores
: 1} >}, < {$unwind: "$actores" } > )
< "_id": ObjectId("5524f04c0ed06f6d9f4555c1"), "Titulo": "Matrix", "actores":
"Keanu Reeves" >
< "_id": ObjectId("5524f04c0ed06f6d9f4555c1"), "Titulo": "Matrix", "actores":
"Carry-Anne Moss" >
< "_id": ObjectId("5524f04c0ed06f6d9f4555c1"), "Titulo": "Matrix", "actores":
"Laurence Fishburne" >
< "_id": ObjectId("5524f04c0ed06f6d9f4555c1"), "Titulo": "Matrix", "actores":
"Hugo Weaving" >
< "_id": ObjectId("5524f04c0ed06f6d9f4555c1"), "Titulo": "Matrix", "actores":
"Gloria Foster" >
< "_id": ObjectId("5524f04c0ed06f6d9f4555c1"), "Titulo": "Matrix", "actores":
"Joe Pantoliano" >
```

4) Igual que la consulta anterior, pero se mostrarán solo los tres últimos resultados ordenados por el nombre del actor.

Figura 55

```
> db.media.aggregate( < {$match: < {Titulo: "Matrix"} >}, < {$project: < {Titulo: 1, actores
: 1} >}, < {$unwind: "$actores" } >, < {$sort: < {actores: 1} >}, < {$skip: 3} > )
< "_id": ObjectId("5524f04c0ed06f6d9f4555c1"), "Titulo": "Matrix", "actores":
"Joe Pantoliano" >
< "_id": ObjectId("5524f04c0ed06f6d9f4555c1"), "Titulo": "Matrix", "actores":
"Keanu Reeves" >
< "_id": ObjectId("5524f04c0ed06f6d9f4555c1"), "Titulo": "Matrix", "actores":
"Laurence Fishburne" >
```

5) Obtener las películas distintas que hay con respecto al título, los diferentes años de estrenos, y los diferentes tipos de documentos. Se realizará en tres consultas diferentes.

Figura 56

```
> db.media.distinct<"Titulo">
[ "Recuerdos", "Matrix", "Blade Runner", "Batman", "Superman" ]
> db.media.distinct<"estreno">
[ 1999, 1982 ]
> db.media.distinct<"tipo">
[ "libro", "CD", "DVD" ]
```

6) Agrupar los documentos por «Título», mostrando el título y el total que hay de cada grupo.

Figura 57

```
> db.media.group(<< key: <Titulo:true>, initial:<Total:0>, reduce: function <item,prev> <prev.Total +1>>>)
[
  <
    "Titulo" : "Matrix",
    "Total" : 1
  >,
  <
    "Titulo" : "Blade Runner",
    "Total" : 2
  >,
  <
    "Titulo" : "Batman",
    "Total" : 2
  >,
  <
    "Titulo" : "Superman",
    "Total" : 2
  >,
  <
    "Titulo" : null,
    "Total" : 1
  >
]
```

7) Añadir las siguientes películas:

```
{ "tipo": "DVD",      { "tipo": "DVD",      { "tipo": "DVD",
  "Titulo": "Blade Runner",  "Titulo": "Batman",  "Titulo":
"Superman",
  "estreno":1967           "estreno": 1989      "estreno": 1996
}                          }                          }
```

Repetir el caso anterior pero solo con los documentos que pertenecen a películas.

Figura 58

```

> db.media.group(<key:<Titulo:true>, cond:<"tipo": "DVD">, initial:<Total:0>,
reduce: function <items,prev> <prev.Total +=1>>)
{
  <
    "Titulo" : "Matrix",
    "Total" : 1
  >,
  <
    "Titulo" : "Blade Runner",
    "Total" : 2
  >,
  <
    "Titulo" : "Batman",
    "Total" : 2
  >,
  <
    "Titulo" : "Supernan",
    "Total" : 2
  >
}

```

8) Obtener, usando MapReduce, la suma de los años de los estrenos de cada película. Es decir, deben obtenerse documentos de la forma: {“_id”: “Batman”, “value”:{“TotalPelículas”:3988 } }.

Figura 59

```

> var map= function() < var key=this.Titulo; var value=<"Total": this.estreno>;
emit(key,value); >;
> var reduce=function (keys,value) < var reduced= <TotalPelículas:0>; for <var
i=0; i< values.length; i++> < reduced.TotalPelículas += values[i].Total;> return
reduced; >
> db.media.mapReduce(map.reduce,<out:"resultado", query:<"tipo":"DVD">>)
{
  "result" : "resultado",
  "timeMillis" : 18,
  "counts" : <
    "input" : 7,
    "emit" : 7,
    "reduce" : 3,
    "output" : 4
  >,
  "ok" : 1
}
>
> db.resultado.find()
{ "_id" : "Batman", "value" : < "TotalPelículas" : 3988 > }
{ "_id" : "Blade Runner", "value" : < "TotalPelículas" : 3949 > }
{ "_id" : "Matrix", "value" : < "Total" : 1999 > }
{ "_id" : "Supernan", "value" : < "TotalPelículas" : 3995 > }

```


Soluciones a los ejercicios del capítulo 9

Apartado a)

1) Encontrar el número de productos que cumplen que «\$num-a» es mayor que «\$num-b».

Figura 60

```
> db.prueba.aggregate(<<{"$project":{"producto":1,"num-a":1,"num-b":1,"compa":{"$gt":{"$num-a","$num-b"}}}},<{"$match":{"compa":true}},<{"$group":{"_id":null,"count":{"$sum:1}}}>>
<{"_id":null,"count":99}>
```

Con respecto a la consulta anterior, obtener el plan de ejecución que ha usado el optimizador de consultas.

Figura 61

```
> db.prueba.aggregate(<<{"$project":{"producto":1,"num-a":1,"num-b":1,"compa":{"$gt":{"$num-a","$num-b"}}}},<{"$match":{"compa":true}},<{"$group":{"_id":null,"count":{"$sum:1}}}>>
<{"_id":null,"count":99}>
```

Figura 62

```
> db.salida.find(<<"compa":true>>).explain()
{"cursor" : "BasicCursor",
  "isMultiKey" : false,
  "n" : 99,
  "nscannedObjects" : 198,
  "nscanned" : 198,
  "nscannedObjectsWithPlans" : 198,
  "nscannedWithPlans" : 198,
  "scanFindOrder" : false,
  "indexOnly" : false,
  "yield" : 1,
  "nChunkSkip" : 0,
  "millis" : 0,
  "server" : "asarasa-U010:27017",
  "filterSet" : false}
```

a) ¿Cómo ha llevado a cabo la operación?

Se ha recorrido completamente la colección (campo «BasicCursor»).

b) ¿Cuántos documentos ha tenido que consultar?

Se han consultado 198 documentos (campo «nscannedObjects»).

- c) ¿Cuánto tarda en realizar la consulta?
Indica 0 milisegundos (campo «millis»).
- d) Obtener las estadísticas de ejecución de la consulta.

Figura 63

```
> db.salida.find(<<"compa":true>>).explain("executionStats")
{
  "cursor" : "BasicCursor",
  "isMultiKey" : false,
  "n" : 99,
  "nscannedObjects" : 198,
  "nscanned" : 198,
  "nscannedObjectsAllPlans" : 198,
  "nscannedAllPlans" : 198,
  "scanAndOrder" : false,
  "indexOnly" : false,
  "nYields" : 1,
  "nChunkSkips" : 0,
  "millis" : 0,
  "allPlans" : [
    {
      "cursor" : "BasicCursor",
      "isMultiKey" : false,
      "n" : 99,
      "nscannedObjects" : 198,
      "nscanned" : 198,
      "scanAndOrder" : false,
      "indexOnly" : false,
      "nChunkSkips" : 0
    }
  ]
}
1.
{
  "server" : "asarasa-UNIO:27017",
  "filterSet" : false,
  "stats" : {
    "type" : "COLLSCAN",
    "works" : 200,
    "yields" : 1,
    "unYields" : 1,
    "invalidates" : 0,
    "advanced" : 99,
    "needTime" : 100,
    "needFetch" : 0,
    "isEOF" : 1,
    "docsTested" : 198,
    "children" : [ ]
  }
}
```

Apartado b)

1) Muestra el plan de ejecución de la consulta que recupera 30 resultados de mesas de color rojo ordenadas por fecha de fabricación. ¿Qué puedes decir de la ejecución en cuanto a tiempo empleado y cómo ha llevado a cabo la operación?

Ha recorrido todos los documentos («BasicCursor») y ha tardado 160 milisegundos.

Figura 64

```

> db.ensayo.find(<<"producto":"nesas","color":"rojo">>).sort(<<"fechafabricacion":1
>>).limit(30).explain()
{
  "clauses" : [
    {
      "cursor" : "BasicCursor",
      "isMultiKey" : false,
      "n" : 30,
      "nscannedObjects" : 90000,
      "nscanned" : 90000,
      "scanAndOrder" : true,
      "indexOnly" : false,
      "nChunkSkips" : 0
    },
    {
      "cursor" : "BasicCursor",
      "isMultiKey" : false,
      "n" : 0,
      "nscannedObjects" : 0,
      "nscanned" : 0,
      "scanAndOrder" : true,
      "indexOnly" : false,
      "nChunkSkips" : 0
    }
  ],
  "cursor" : "QueryOptimizerCursor",
  "n" : 30,
  "nscannedObjects" : 90000,
  "nscanned" : 90000,
  "nscannedObjectsAllPlans" : 90000,
  "nscannedAllPlans" : 90000,
  "scanAndOrder" : false,
  "nFields" : 783,
  "nChunkSkips" : 0,
  "nillis" : 160,
  "server" : "asarasa-0A10:27017",
  "filterSet" : false
}

```

2) Buscar una estrategia para mejorar el tiempo de ejecución y el número de elementos que son necesarios revisar.

Figura 65

```

> db.ensayo.find(<<"producto":"nesas","color":"rojo">>).sort(<<"fechafabricacion":1
>>).limit(30).explain()
{
  "cursor" : "BtreeCursor producto_1_color_1_fechafabricacion_1",
  "isMultiKey" : false,
  "n" : 30,
  "nscannedObjects" : 30,
  "nscanned" : 30,
  "nscannedObjectsAllPlans" : 59,
  "nscannedAllPlans" : 59,
  "scanAndOrder" : false,
  "indexOnly" : false,
  "nFields" : 0,
  "nChunkSkips" : 0,
  "nillis" : 1,
  "indexBounds" : {
    "producto" : [
      {
        "nesas",
        "nesas"
      }
    ],
    "color" : [
      {
        "rojo",
        "rojo"
      }
    ],
    "fechafabricacion" : [
      {
        <
        >
      }
    ]
  },
  "$ninElement" : 1,
  "$maxElement" : 1
},
  "server" : "asarasa-0A10:27017",
  "filterSet" : false
}

```

Soluciones a los ejercicios del capítulo 10

1) Realizar un backup de las bases de datos que se tengan en algún directorio con permiso de escritura. Para ello, utilizar la opción `-out`, que permite redirigir la salida a un directorio diferente al que hay por defecto.

Figura 66

```
G:\mongodb\bin>mongodump --out C:\Ejercicio1
connected to: 127.0.0.1
2015-04-23T22:33:04.403+0200 all dbs
2015-04-23T22:33:04.491+0200 DATABASE: base to C:\Ejercicio1\base
2015-04-23T22:33:04.522+0200 base.system.indexes to C:\Ejercicio1\base\system
.indexes.bson
2015-04-23T22:33:04.536+0200 1 documents
```

Figura 67

Nombre	Fecha de modifica...	Tipo
admin	23/04/2015 22:33	Carpeta de archivos
base	23/04/2015 22:33	Carpeta de archivos
blog	23/04/2015 22:33	Carpeta de archivos
m101	23/04/2015 22:33	Carpeta de archivos
prueba	23/04/2015 22:33	Carpeta de archivos
school	23/04/2015 22:33	Carpeta de archivos
students	23/04/2015 22:33	Carpeta de archivos
test	23/04/2015 22:33	Carpeta de archivos

2) Añadir las bases de datos creadas en los ejercicios de los capítulos anteriores. Una vez cargadas, realizar un backup de la colección media que se encuentra en la base de datos multimedia. Para ello, utilizar las opciones `--db` y `-collection`, que permiten seleccionar una base de datos y colección concreta.

Figura 68

```
C:\mongodb\bin>mongodump --out C:/Nuevo --db multimedia --collection media
connected to: 127.0.0.1
2015-04-24T18:53:49.902+0200 DATABASE: multimedia to C:/Nuevo\multimedia
2015-04-24T18:53:49.916+0200 multimedia.media to C:/Nuevo\multimedia\media.bson
2015-04-24T18:53:49.920+0200 8 documents
2015-04-24T18:53:49.922+0200 Metadata for multimedia.media to C:/Nuevo\multimedia\media.metadata.json
```

Figura 69

Nombre	Fecha de modifica...	Tipo	Tamaño
media.bson	24/04/2015 18:53	Archivo BSON	2 KB
media.metadata.json	24/04/2015 18:53	Archivo JSON	1 KB

3) Abrir una instancia de mongod en el puerto 27019, y desde un terminal de mongo sobre el mismo puerto añadir las bases de datos de los capítulos anteriores y repetir el ejercicio 2. Ahora, para poder hacer un backup, hay que indicarle explícitamente sobre qué host y en qué puerto debe realizarlo. Para ello, se debe encontrar el nombre del host de la máquina, y usar las opciones --host y --port, que permiten indicar sobre qué máquina y en qué puerto debe realizarse el backup.

Figura 70

```
C:\mongodb\bin>mongodump --host Antonio --port 27019 --db multimedia --collection media --out C:/Nuevo
connected to: Antonio:27019
2015-04-24T19:04:23.505+0200 DATABASE: multimedia to C:/Nuevo/multimedia
2015-04-24T19:04:23.519+0200 multimedia.media to C:/Nuevo/multimedia/media.bson
2015-04-24T19:04:23.523+0200 8 documents
2015-04-24T19:04:23.524+0200 Metadata for multimedia.media to C:/Nuevo/multimedia/media.metadata.json
```

4) Restaurar el backup de las bases de datos que se realizó en el ejercicio 1. Para ello, junto al comando mongorestore, se debe indicar el path al directorio en el que se encuentra el backup.

Figura 71

```
C:\mongodb\bin>mongorestore --drop C:/Nuevo
connected to: 127.0.0.1
2015-04-24T19:24:46.012+0200 C:/Nuevo/multimedia/media.bson
2015-04-24T19:24:46.019+0200 going into namespace [multimedia.media]
2015-04-24T19:24:46.020+0200 dropping
3 objects found
2015-04-24T19:24:46.046+0200 Creating index: { key: { _id: 1 }, name: "_id_", ns: "multimedia.media" }
```

5) Considerar los backups que se han realizado en el ejercicio 2. Restaurar los documentos de la base de datos multimedia y de la colección media. Para ello, utilizar las opciones `--db` y `--collection`, que permiten seleccionar una base de datos y colección concreta.

Figura 72

```
C:\mongodb\bin>mongorestore --collection media --db multimedia dump/multimedia/
media.bson
connected to: 127.0.0.1
2015-04-24T19:38:13.278+0200 dump/multimedia/media.bson
2015-04-24T19:38:13.277+0200 going into namespace [multimedia.media]
Restoring to multimedia.media without dropping. Restored data will be inserted w
ithout raising errors; check your server log
8 objects found
2015-04-24T19:38:13.315+0200 Creating index: { key: { _id: 1 }, name: "_id_",
ns: "multimedia.media" }
```

6) Abrir una instancia de mongod en el puerto 27019, y desde un terminal de mongo sobre el mismo puerto restaurar las bases de datos de los capítulos anteriores y repetir el ejercicio 2. Ahora, para poder hacer una restauración, hay que indicarle explícitamente sobre qué host y en qué puerto debe realizarlo. Para ello, se debe encontrar el nombre del host de la máquina, y usar la opción `--host`, que permite indicar sobre qué máquina y en qué puerto debe realizarse la restauración.

Figura 73

```
C:\mongodb\bin>mongorestore --drop --host Antonio --port 27019 --collection medi
a --db multimedia dump/multimedia/media.bson
connected to: Antonio:27019
2015-04-24T19:44:06.432+0200 dump/multimedia/media.bson
2015-04-24T19:44:06.439+0200 going into namespace [multimedia.media]
2015-04-24T19:44:06.439+0200 dropping
8 objects found
2015-04-24T19:44:06.469+0200 Creating index: { key: { _id: 1 }, name: "_id_",
ns: "multimedia.media" }
```

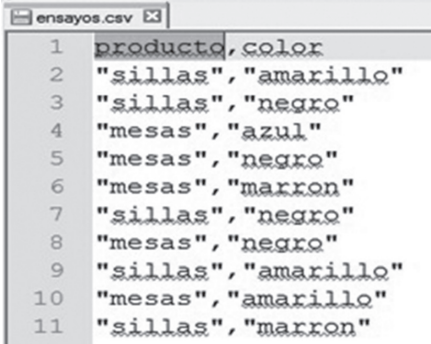
7) Considerar la base de datos que se utilizó en el ejercicio 2 del capítulo sobre optimización. Crear de nuevo esa base de

datos con la colección correspondiente. A continuación, realizar una exportación a un archivo de tipo cvs, exportando únicamente los campos «producto» y «color». Para ello, utilizar las opciones --csv y --fields.

Figura 74

```
C:\mongodb\bin>mongoexport --db ensayos --collection ensayo --csv --fields producto,color --out C:/Nuevo/ensayos.csv
connected to: 127.0.0.1
exported 90000 records
```

Figura 75



Line	producto	color
2	"sillas"	"amarillo"
3	"sillas"	"negro"
4	"mesas"	"azul"
5	"mesas"	"negro"
6	"mesas"	"marron"
7	"sillas"	"negro"
8	"mesas"	"negro"
9	"sillas"	"amarillo"
10	"mesas"	"amarillo"
11	"sillas"	"marron"

8) Considerar la base de datos anterior. A continuación, realizar una exportación a un archivo de tipo cvs, exportando el resultado de la consulta que se hizo en ese ejercicio, es decir, «recupera las mesas de color rojo». Para ello, utilizar las opciones --csv y --query, donde se indicará como un documento en forma de cadena la condición de filtrado.

Figura 76

```
C:\mongodb\bin>mongoexport --db ensayos --collection ensayo --csv --fields producto,color,fechafabricacion --query '{\"producto\":\"mesas\", \"color\":\"rojo\"}' --out C:/Nuevo/ensayo.csv
connected to: 127.0.0.1
exported 6393 records
```

Figura 77

```

Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Mac
ensayos.csv  media.csv  media.json  ensayo.csv
1  producto,color,fechafabricacion
2  "mesas","rojo",2000-04-24T18:07:54.512Z
3  "mesas","rojo",2000-04-24T18:07:54.512Z
4  "mesas","rojo",2000-04-24T18:07:54.512Z
5  "mesas","rojo",2000-04-24T18:07:54.512Z
6  "mesas","rojo",2000-04-24T18:07:54.512Z
7  "mesas","rojo",2000-04-24T18:07:54.512Z
8  "mesas","rojo",2000-04-24T18:07:54.512Z
9  "mesas","rojo",2000-04-24T18:07:54.512Z
10 "mesas","rojo",2000-04-24T18:07:54.512Z
11 "mesas","rojo",2000-04-24T18:07:54.512Z
12 "mesas","rojo",2000-04-24T18:07:54.512Z
13 "mesas","rojo",2000-04-24T18:07:54.512Z
14 "mesas","rojo",2000-04-24T18:07:54.512Z
15 "mesas","rojo",2000-04-24T18:07:54.512Z
16 "mesas","rojo",2000-04-24T18:07:54.512Z

```

9) Abrir una instancia de mongod en el puerto 27019, desde un terminal de mongo sobre el mismo puerto, y repetir el ejercicio 1. Ahora, para poder hacer una exportación, hay que indicarle explícitamente qué host y en qué puerto debe realizarlo. Para ello se debe encontrar el nombre del host de la máquina, y usar las opciones `--host` y `--port`, que permiten indicar sobre qué máquina y en qué puerto debe realizarse la exportación.

Figura 78

```

C:\mongodb\bin>mongoexport --host Antonio --port 27019 --db ensayos --collection
ensayo --csv --fields producto,color --out C:/Nuevo/ensayos.csv
connected to: Antonio:27019
exported 90000 records

```

10) Considerar la base de datos que exportado en los ejercicios anteriores. A continuación, importar el archivo csv exportado a una colección denominada «ensayo2» en la base de datos

«ensayos». Debe indicarse que obtenga los nombres de los campos a partir de la primera línea del archivo csv mediante la opción `--headerline`.

Figura 79

```
C:\mongodb\bin>mongoimport --db ensayos --collection ensayo2 --type=csv --header
line --file C:/Nuevo/ensayos.csv
connected to: 127.0.0.1
2015-04-24T22:03:10.039*0200      Progress: 112347/1600283      7%
2015-04-24T22:03:10.047*0200      6700      2233/second
2015-04-24T22:03:13.001*0200      Progress: 226530/1600283      14%
2015-04-24T22:03:13.009*0200      13500     2250/second
2015-04-24T22:03:16.007*0200      Progress: 318724/1600283      19%
2015-04-24T22:03:16.007*0200      19000     2111/second
2015-04-24T22:03:19.031*0200      Progress: 419370/1600283      26%
2015-04-24T22:03:19.031*0200      25000     2083/second
2015-04-24T22:03:22.087*0200      Progress: 500021/1600283      31%
2015-04-24T22:03:22.089*0200      29800     1986/second
2015-04-24T22:03:25.023*0200      Progress: 542081/1600283      33%
2015-04-24T22:03:25.051*0200      32300     1794/second
2015-04-24T22:03:28.134*0200      Progress: 568949/1600283      35%
2015-04-24T22:03:28.144*0200      33900     1614/second
2015-04-24T22:03:31.061*0200      Progress: 631174/1600283      39%
2015-04-24T22:03:31.061*0200      37600     1566/second
2015-04-24T22:03:34.048*0200      Progress: 691635/1600283      43%
2015-04-24T22:03:34.050*0200      41200     1525/second
2015-04-24T22:03:37.039*0200      Progress: 760301/1600283      47%
2015-04-24T22:03:37.039*0200      45300     1510/second
2015-04-24T22:03:40.024*0200      Progress: 829094/1600283      51%
2015-04-24T22:03:40.030*0200      49400     1496/second
2015-04-24T22:03:43.019*0200      Progress: 896210/1600283      56%
2015-04-24T22:03:43.020*0200      53400     1483/second
2015-04-24T22:03:46.107*0200      Progress: 978275/1600283      61%
2015-04-24T22:03:46.107*0200      58300     1494/second
2015-04-24T22:03:49.026*0200      Progress: 1052234/1600283     65%
2015-04-24T22:03:49.026*0200      62700     1492/second
2015-04-24T22:03:52.005*0200      Progress: 1134458/1600283     70%
2015-04-24T22:03:52.006*0200      67600     1502/second
2015-04-24T22:03:55.003*0200      Progress: 1204946/1600283     75%
2015-04-24T22:03:55.007*0200      71800     1495/second
2015-04-24T22:03:58.005*0200      Progress: 1275296/1600283     79%
2015-04-24T22:03:58.005*0200      76000     1490/second
2015-04-24T22:04:01.001*0200      Progress: 1350920/1600283     84%
2015-04-24T22:04:01.004*0200      80500     1490/second
2015-04-24T22:04:04.073*0200      Progress: 1436516/1600283     89%
2015-04-24T22:04:04.073*0200      85600     1501/second
2015-04-24T22:04:06.971*0200      check 9 90001
2015-04-24T22:04:06.976*0200      imported 90000 objects
```

11) Crear un usuario administrador y un usuario con derechos solo de lectura.

Figura 80

```
> db.createUser({
.. user: "usuario1",
.. pwd: "usuario1",
.. roles: [ "root" ]
.. }
.. )
Successfully added user: < "user" : "usuario1", "roles" : [ "root" ] >
> db.createUser({
.. user: "usuario2",
.. pwd: "usuario2",
.. roles: [ "read" ]
.. }
.. )
Successfully added user: < "user" : "usuario2", "roles" : [ "read" ] >
```