

BASES DE DATOS NO RELACIONAL con mongo

Brayda Solangie Quintana Ruiz

ficha: 2465879

links:

- drive:

https://drive.google.com/drive/folders/1z81APVyi8jY0z4IZln0IsrF28Jh0MEWk?usp=share_link

- git:

https://github.com/bsquintana55/bd_no_relacional_estrenos

INTRODUCCIÓN BASES DE DATOS NOSQL- MONGODB

1. Consulte y realice un cuadro comparativo sobre los tipos de dato que se pueden usar dentro de Mongo DB, indicando el nombre, su descripción o uso y un ejemplo de representación de este.

NOMBRE	DESCRIPCIÓN	EJEMPLO
Objeto (object)	Este tipo de datos almacena documentos incrustados, a los cuales también se les conoce como documentos anidados.	<code>"object": { "name": "Kyle", "age": 24 },</code>
Objeto id	se utiliza la raya al piso (_) para decir que ese campo es el identificador	<code>"_id":</code> <code>..</code>
String	se pone entre comillas ("") son para cadenas de texto	<code>"Deluxe Suite Premium 2",</code>
numéricos	double, int... se utilizan para valores numéricos o de cantidad	<code>: 101,</code>
booleans	(true o false), representar valores de lógica binaria	<code>"boolean": true,</code> <code>" "" ""</code>

<i>undefined</i>	almacena valores indefinidos	<pre>"precio": undefined,</pre>
<i>datos binarios</i>	almacenar datos de 0 y 1, es decir binarios	<pre>"caducidad": "10100100011"</pre>
<i>fecha</i>	es para guardar datos tipo fecha	<pre>> Date() Sun Oct 14 2018 09:33:32 GMT+0530 (India Standard Time)</pre>
<i>arrays (listas)</i>	para almacenar una serie de datos	<pre>"array": ["Hello", 5, false, null, { "key": "value", "number": 6 }],</pre>

2. Consulte el nombre, la definición y uso de cada uno de los operadores propuestos para trabajo con consultas en Mongo DB.

<https://kinsta.com/es/blog/operadores-mongodb/>

OPERADOR	NOMBRE definición	USO
<i>\$eq</i>	<i>igual</i> operador coincide con los valores que son iguales al valor dado	<pre>{ <field>: { \$eq: <value> } }</pre>
<i>\$gt</i>	<i>mayor que</i> operador coincide si los valores son mayores que el valor dado	<pre>{ field: { \$gt: value } }</pre>
<i>\$gte</i>	<i>mayor o igual que</i> operador coincide cuando los valores son mayores o iguales que el valor dado	<pre>{ field: { \$gte: value } }</pre> <i>db.inventory.find({"qty": { \$gte: 25}})</i>
<i>\$lt</i>	<i>menor que</i> operador coincide si los valores son menores que el valor proporcionado	<pre>{ field: { \$lt: value } }</pre> <i>db.inventory.find({"qty": { \$lt: 25}})</i>
<i>\$lte</i>	<i>menor o igual que</i> operador sólo coincide si los valores son menores o iguales que el valor dado	<pre>{ field: { \$lte: value } }</pre> <i>db.inventory.find({"qty": { \$lte: 25}})</i>
<i>\$ne</i>	<i>distinto</i> operador devuelve los documentos en los que el valor especificado no es igual	<pre>{ \$ne: value }</pre> <i>db.inventory.find({ qty: { \$ne: 20 } })</i>
<i>\$in</i>	<i>dentro de</i> operador devuelve los documentos que coinciden con los valores especificados	<pre>{ field: { \$in: [<value1>, <value2>, ... <valueN>] } }</pre> <i>db.collection.find({ "qty": { \$in: [30, 15] } })</i>

<i>\$nin</i>	<i>no dentro de</i> operador devuelve los documentos que no coinciden con los valores dados	<pre>{ field: { \$nin: [<value1>, <value2> ... <valueN>] }</pre> <i>db.collection.find({ "qty": { \$nin: [20, 15] }, { _id: 0 } })</i>
<i>\$or</i>	<i>x O z</i> Selecciona los documentos en los que se cumple al menos una las condiciones de las expresiones	<pre>{ \$or: [{ <expression1> }, { <expression2> },</pre>
<i>\$and</i>	<i>x Y z</i> Selecciona los documentos en los que se cumplen todas las condiciones de las expresiones	<pre>{ \$and: [{ <expression1> }, { <expression2> },</pre>
<i>\$not</i>	<i>NO x <-> z</i> Selecciona los documentos que no coinciden con las condiciones de las expresiones	<pre>{ field: { \$not: { <operator-expression> } } }</pre>
<i>\$nor</i>	<i>a NO b</i> Selecciona los documentos que no cumplen con las condiciones de las expresiones	<pre>{ \$nor: [{ <expression1> }, { <expression2> },</pre>
<i>\$exists</i>	<i>x -> true, false</i> Este operador coincide con los documentos que tienen un campo especificado. Este operador tiene un valor booleano que puede ser true o false	<pre>{ field: { \$exists: <boolean> } }</pre> <i>db.bagofmarbles.find({ red: { \$exists: true } })</i>
<i>\$type</i>	<i>type -> xx</i> Este operador hace coincidir los documentos según el tipo de campo especificado	<pre>{ field: { \$type: <BSON type> } }</pre> <i>db.addressBook.find({ "zipCode": { \$type: "string" } })</i>
<i>\$mod</i>	Puedes utilizar este operador para hacer coincidir los documentos en los que el valor de un campo especificado es igual al resto después de ser dividido por un valor especificado	<pre>db.cars.find ({ qty: { \$mod: [250,0] } })</pre>
<i>\$regex</i>	Ofrece la posibilidad de utilizar expresiones regulares para hacer coincidir cadenas con patrones en las consultas.	<pre>db.cars.find({ price: { \$regex: /\$78900/ } })</pre>
<i>\$elemMatch</i>	Coincide con los documentos que contienen un campo de matriz con al menos un elemento que coincide con todos los criterios de consulta especificados	<pre>{ : { \$elemMatch: { <query1>, <query2>, ... } } }</pre>
<i>\$size</i>	El operador devuelve aquellos documentos en los que el tamaño de la matriz coincide con el número de elementos especificado en el argumento	<pre>{ field: { \$size: value } }</pre>

MANEJO DE BASES DE DATOS NOSQL

- con el comando **db** se comprueba que bd se está utilizando
- con el comando **show dbs** está va a mostrar que bd hay en mongo
- se crea la bases de datos y se cambia a esta o se utiliza esta con solo el comando **use** después irá el nombre de la bd que queremos crear.

```
test> db
test
test> show dbs
admin      40.00 KiB
config     60.00 KiB
flores     72.00 KiB
local      40.00 KiB
mascotas   160.00 KiB
test> use pelis_estreno
switched to db pelis_estreno
pelis_estreno>
```

CREAR LAS COLECCIONES

- con el comando **db.createCollection("")** se crea una colección dentro de la bd que se esté utilizando

```
pelis_estreno> db.createCollection("director")
{ ok: 1 }
pelis_estreno> show collections
director
pelis_estreno> db.createCollection("pelicula")
{ ok: 1 }
pelis_estreno> db.createCollection("estreno")
{ ok: 1 }
pelis_estreno> show collections
director
estreno
pelicula
pelis_estreno>
```

INSERTS

- con el comando **db.<collection>.insertOne({})** se crea un documento dentro de la colección elegida, en este caso es solo **1** gracias al metodo **.insertOne({})**
- para explicar rápidamente la estructura de una colección, se componen de **documentos, files y datos.**

se utiliza la estructura de json,

→ *base de datos.colección.método*

db.<collection>.insertOne(

→ *files o propiedades, ejem: **_id** y el **valor** de ese file*

◆ *(para marcar que sea una primary key se utiliza la raya al piso (_) y el nombre del file)*

◆ *abre documento*

{_id: 6,

nombre: 'Kung Fu Panda',

genero: 'infantil, animacion, aventura',

actor_princ: 'panda Po',

duracion: 91

}

◆ *cierra documento*

)

◆ *recordemos que los tipos string van en comillas porque son cadena de texto pero si son números no lo necesitan*

◆ *las flechas de color violeta seras donde se van a hacer modificaciones mas adelante*

INSERTS DE DIRECTOR

db.director.insertOne({_id: 1, nombre: 'Tim Burton', ocupacion: 'director de cine, productor, escritor y dibujante',

→ películas: 'Batman, Beetlejuice, El joven manos de tijeras',

→ nacionalidad: 'Estadounidense'})

```
pelis_estreno> db.director.insertOne({_id: 1, nombre: 'Tim Burton', ocupacion: 'director de cine, productor, escritor y dibujante', películas: 'Batman, Beetlejuice, El joven manos de tijeras', nacionalidad: 'Estadounidense'})
```

db.director.insertOne({_id: 2, nombre: 'Neil Jordan', ocupacion: 'director de cine y novelista', peliculas: 'El juego de las lagrimas, La viuda, Entrevista con el vampiro',

→ nacionalidad: 'irlandés'

}})

```
pelis_estreno> db.director.insertOne({_id: 2, nombre: 'Neil Jordan', ocupacion: 'director de cine y novelista', peliculas: 'El juego de las lagrimas, La viuda, Entrevista con el vampiro', nacionalidad: 'irlandés'})
```

db.director.insertOne({_id: 3, nombre: 'Martin Charles Scorsese', ocupacion: 'director, guionista y productor de cine',

→ peliculas: 'Buenos muchachos,

Taxi Driver, El Lobo de Wall Street', nacionalidad: 'estadounidense'}})

```
pelis_estreno> db.director.insertOne({_id: 3, nombre: 'Martin Charles Scorsese', ocupacion: 'director, guionista y productor de cine', peliculas: 'Buenos muchachos, Taxi Driver, El Lobo de Wall Street', nacionalidad: 'estadounidense'})
```

db.director.insertOne({_id: 4, nombre: 'Steven Spielberg',

→ ocupacion: 'director, guionista y productor de cine',

peliculas: 'Tiburón, Parque Jurasico, E.T., el extraterrestre', nacionalidad: 'estadounidense'}})

```
pelis_estreno> db.director.insertOne({_id: 4, nombre: 'Steven Spielberg', ocupacion: 'director, guionista y productor de cine', peliculas: 'Tiburón, Parque Jurasico, E.T., el extraterrestre', nacionalidad: 'estadounidense'})
```

db.director.insertOne({_id: 5,

→ nombre: 'Quentin Tarantino',

ocupacion: 'director de cine, productor, guionista, editor y actor', peliculas: 'Kill Bill, Bastardos sin gloria, Tiempos violentos', nacionalidad: 'estadounidense'}})

```
pelis_estreno> db.director.insertOne({_id: 5, nombre: 'Quentin Tarantino', ocupacion: 'director de cine, productor, guionista, editor y actor', peliculas: 'Kill Bill, Bastardos sin gloria, Tiempos violentos', nacionalidad: 'estadounidense'})
```

```
db.director.insertOne({_id: 6, nombre: 'Michael Bay', ocupacion: 'cineasta',  
peliculas: 'Transformers, Ambulance, Armagedon', nacionalidad:  
'estadounidense'})
```

```
pelis_estreno> db.director.insertOne({_id: 6, nombre: 'Michael Bay', ocupacion: 'cine  
asta', peliculas: 'Transformers, Ambulance, Armagedon', nacionalidad: 'estadounidense  
'})
```

INSERTS DE PELÍCULA

→ actor_princ

```
db.pelicula.insertOne(  
{_id: 1,  
nombre: 'El joven manos de tijera',  
→ genero: 'fantasia romatica',  
→ actor_princ: 'Johnny Deep',  
duracion: 126  
})
```

```
pelis_estreno> db.pelicula.insertOne(  
... {_id: 1,  
... nombre: 'El joven manos de tijera',  
... genero: 'fantasia romatica',  
... actor_princ: 'Johnny Deep',  
... duracion: 126  
... })
```

```
db.pelicula.insertOne({_id: 2,  
→ nombre: 'Entrevista con el vampiro',  
genero: 'suspense, drama, romance',  
actor_princ: 'Tom Cruice y Brad Pitt',  
duracion: 122  
})
```

```
pelis_estreno> db.pelicula.insertOne({_id: 2,  
... nombre: 'Entrevista con el vampiro',  
... genero: 'suspense, drama, romance',  
... actor_princ: 'Tom Cruise y Brad Pitt',  
... duracion: 122  
... })
```

db.pelicula.insertOne({_id: 3,

nombre: 'Beetlejuice',

→ genero: 'comedia, fantasia',

actor_princ: 'Winona Ryder y Michael Keaton',

duracion: 92

})

```
pelis_estreno> db.pelicula.insertOne({_id: 3,  
... nombre: 'Beetlejuice',  
... genero: 'comedia, fantasia',  
... actor_princ: 'Winona Ryder y Michael Keaton',  
... duracion: 92  
... })
```

db.pelicula.insertOne(

{_id: 4,

→ nombre: 'Transformers',

genero: 'accion, ficcion',

actor_princ: 'Shia LaBeouf',

→ duracion: 144

})

```
pelis_estreno> db.pelicula.insertOne(  
... {_id: 4,  
... nombre: 'Transformers',  
... genero: 'accion, ficcion',  
... actor_princ: 'Shia LaBeouf',  
... duracion: 144  
... })
```



```
db.pelicula.insertOne({_id: 5,  
nombre: 'Tom y Jerry',  
→ genero: 'aventura, comedia, familiar',  
actor_princ: 'Tom y Jerry',  
duracion: 101  
}))
```

```
pelis_estreno> db.pelicula.insertOne({_id: 5,  
... nombre: 'Tom y Jerry',  
... genero: 'aventura, comedia, familiar',  
... actor_princ: 'Tom y Jerry',  
... duracion: 101  
... })
```

```
db.pelicula.insertOne({_id: 6,  
nombre: 'Kung Fu Panda',  
genero: 'infantil, animacion, aventura',  
→ actor_princ: 'panda Po',  
duracion: 91  
}))
```

```
pelis_estreno> db.pelicula.insertOne({_id: 6,  
... nombre: 'Kung Fu Panda',  
... genero: 'infantil, animacion, aventura',  
... actor_princ: 'panda Po',  
... duracion: 91  
... })
```

INSERTS DE ESTRENO

```
db.estreno.insertOne({_id: 1,  
→ pelicula: 'El joven manos de tijera',
```

año_estreno: 1990, recaudacion:'86 millones USD', lugar_estreno:'Estados Unidos', plataforma:'Disney+'})

```
pelis_estreno> db.estreno.insertOne({_id: 1, pelicula: 'El joven manos de tijera', año_estreno: 1990, recaudacion:'86 millones USD', lugar_estreno:'Estados Unidos', plataforma:'Disney+'})
```

db.estreno.insertOne({_id: 2, pelicula: 'Entrevista con el vampiro',

→ año_estreno: 1994,

recaudacion:'223,7 millones USD', lugar_estreno:'Estados Unidos',
plataforma:'HBOmax'})

```
pelis_estreno> db.estreno.insertOne({_id: 2, pelicula: 'Entrevista con el vampiro', año_estreno: 1994, recaudacion:'223,7 millones USD', lugar_estreno:'Estados Unidos', plataforma:'HBOmax'})
```

db.estreno.insertOne({_id: 3, pelicula: 'Beetlejuice', año_estreno: 1988,

→ recaudacion:'74,2 millones USD',

lugar_estreno:'Estados Unidos', plataforma:'HBOmax'})

```
pelis_estreno> db.estreno.insertOne({_id: 3, pelicula: 'Beetlejuice', año_estreno: 1988, recaudacion:'74,2 millones USD', lugar_estreno:'Estados Unidos', plataforma:'HBOmax'})
```

db.estreno.insertOne({_id: 4, pelicula: 'Transformers', año_estreno: 2007,
recaudacion:'86 millones USD', lugar_estreno:'Estados Unidos',

→ plataforma:'HBOmax'

})

```
pelis_estreno> db.estreno.insertOne({_id: 4, pelicula: 'Transformers', año_estreno: 2007, recaudacion:'86 millones USD', lugar_estreno:'Estados Unidos', plataforma:'HBOmax'})
```

db.estreno.insertOne({_id: 5, pelicula: 'Tom y Jerry', año_estreno: 2021,
recaudacion:'136,5 millones USD', lugar_estreno:'Estados Unidos',

→ plataforma:'HBOmax'

})

```
pelis_estreno> db.estreno.insertOne({_id: 5, pelicula: 'Tom y Jerry', año_estreno: 2021, recaudacion: '136,5 millones USD', lugar_estreno: 'Estados Unidos', plataforma: 'HBO max'})
```

db.estreno.insertOne({_id: 6, pelicula: 'Kung Fu Panda', año_estreno: 2008,

→ recaudacion: '631,7 millones USD',

lugar_estreno: 'Estados Unidos', plataforma: 'Netflix'})

```
pelis_estreno> db.estreno.insertOne({_id: 6, pelicula: 'Kung Fu Panda', año_estreno: 2008, recaudacion: '631,7 millones USD', lugar_estreno: 'Estados Unidos', plataforma: 'Netflix'})
```

CONSULTAS

CONSULTAS DE DIRECTOR

db.director.find().pretty()

- Esta consulta trae de manera organizada los documentos y sus fields

```
pelis_estreno> db.director.find().pretty()
[
  {
    _id: 1,
    nombre: 'Tim Burton',
    ocupacion: 'director de cine, productor, escritor',
    peliculas: 'Batman, Beetlejuice',
    nacionalidad: 'estadounidense'
  },
  {
    _id: 2,
    nombre: 'Neil Jordan',
    ocupacion: 'director de cine y novelista',
    peliculas: 'El juego de las lagrimas, La viuda, En
```

db.director.find().limit(3)

En este caso la consulta traerá solo 3 documentos, esto se hace con el metodo .limit(), donde se da el valor de la cantidad de los documentos que se quiera mostrar.

```

pelis_estreno> db.director.find().limit(3)
[
  {
    _id: 1,
    nombre: 'Tim Burton',
    ocupacion: 'director de cine, productor, escritor y dibujante',
    peliculas: 'Batman, Beetlejuice, El joven manos de tijeras',
    nacionalidad: 'Estadounidense'
  },
  {
    _id: 2,
    nombre: 'Neil Jordan',
    ocupacion: 'director de cine y novelista',
    peliculas: 'El juego de las lagrimas, La viuda, Entrevista con el vampiro',
    nacionalidad: 'irlandés'
  },
  {
    _id: 3,
    nombre: 'Martin Charles Scorsese',
    ocupacion: 'director, guionista y productor de cine',
    peliculas: 'Buenos muchachos, Taxi Driver, El Lobo de Wall Street',
    nacionalidad: 'estadounidense'
  }
]

```

`db.director.findOne()`

Esta consulta traera el primer documento que encuentre, osea el 1

```

pelis_estreno> db.director.findOne()
{
  _id: 1,
  nombre: 'Tim Burton',
  ocupacion: 'director de cine, productor, escritor y dibujante',
  peliculas: 'Batman, Beetlejuice, El joven manos de tijeras',
  nacionalidad: 'Estadounidense'
}

```

`db.director.find({nacionalidad:'estadounidense'})`

En este caso la consulta traerá todos los documentos donde el director tenga una nacionalidad estadounidense, en el segundo caso trae otro debido a que está en mayúscula la primera palabra.

```

pelis_estreno> db.director.find({nacionalidad:'estadounidense'})
[
  {
    _id: 3,
    nombre: 'Martin Charles Scorsese',
    ocupacion: 'director, guionista y productor de cine',
    peliculas: 'Buenos muchachos, Taxi Driver, El Lobo de Wall Street',
    nacionalidad: 'estadounidense'
  },
  {
    _id: 4,
    nombre: 'Steven Spielberg',
    ocupacion: 'director, guionista y productor de cine',
    peliculas: 'Tiburón, Parque Jurasico, E.T., el extraterrestre',
    nacionalidad: 'estadounidense'
  },
  {
    _id: 5,
    nombre: 'Quentin Tarantino',
    ocupacion: 'director de cine, productor, guionista, editor y actor',
    peliculas: 'Kill Bill, Bastardos sin gloria, Tiempos violentos',
    nacionalidad: 'estadounidense'
  },
  {
    _id: 6,
    nombre: 'Michael Bay',
    ocupacion: 'cineasta',
    peliculas: 'Transformers, Ambulance, Armagedon',
    nacionalidad: 'estadounidense'
  }
]
pelis_estreno>

```

```

pelis_estreno> db.director.find({nacionalidad:'Estadounidense'})
[
  {
    _id: 1,
    nombre: 'Tim Burton',
    ocupacion: 'director de cine, productor, escritor y dibujante',
    peliculas: 'Batman, Beetlejuice, El joven manos de tijeras',
    nacionalidad: 'Estadounidense'
  }
]
pelis_estreno>

```

db.director.find({ocupacion: 'director, guionista y productor de cine'})

en esta consulta deberá mostrar cada director que sea, directo, guionista y productor de cine en este caso muestra solo a dos

```

pelis_estreno> db.director.find({ocupacion: 'director, guionista y productor de cine'})
[
  {
    _id: 3,
    nombre: 'Martin Charles Scorsese',
    ocupacion: 'director, guionista y productor de cine',
    peliculas: 'Buenos muchachos, Taxi Driver, El Lobo de Wall Street',
    nacionalidad: 'estadounidense'
  },
  {
    _id: 4,
    nombre: 'Steven Spielberg',
    ocupacion: 'director, guionista y productor de cine',
    peliculas: 'Tiburón, Parque Jurasico, E.T., el extraterrestre',
    nacionalidad: 'estadounidense'
  }
]
pelis_estreno>

```

CONSULTAS DE PELÍCULAS

- trae el primer documento que su duracion sea de 92 min

```

pelis_estreno> db.pelicula.findOne({duracion:92})
{
  _id: 3,
  nombre: 'Beetlejuice',
  genero: 'comedia, fantasia, historia de fantasmas',
  duracion: 92,
  personaje_princ: 'Winona Ryder y Michael Keaton'
}

```

- trae solo dos documentos

```

pelis_estreno> db.pelicula.find().limit(2)
[
  {
    _id: 1,
    nombre: 'El joven manos de tijera',
    genero: 'fantasia romantica',
    actor_princ: 'Johnny Deep',
    duracion: 126
  },
  {
    _id: 2,
    nombre: 'Entrevista con el vampiro',
    genero: 'suspense, drama, romance',
    actor_princ: 'Tom Cruise y Brad Pitt',
    duracion: 122
  }
]

```

- trae el primer documento que encuentre

```

pelis_estreno> db.pelicula.findOne()
{
  _id: 1,
  nombre: 'El joven manos de tijera',
  genero: 'fantasia romantica',
  actor_princ: 'Johnny Deep',
  duracion: 126
}

```

- trae el documento que tiene el id 6

```

pelis_estreno> db.pelicula.find({_id:6})
[
  {
    _id: 6,
    nombre: 'Kung Fu Panda',
    genero: 'infantil, animacion, aventura',
    actor_princ: 'panda Po',
    duracion: 91
  }
]
pelis_estreno>

```

- trae el documento que tiene el nombre de transformes

```

pelis_estreno> db.pelicula.find({nombre:'Transformers'})
[
  {
    _id: 4,
    nombre: 'Transformers',
    genero: 'accion, ficcion',
    actor_princ: 'Shia LaBeouf',
    duracion: 144
  }
]
pelis_estreno>

```

CONSULTAS DE ESTRENO

- trae todos los documentos

```

pelis_estreno> db.estreno.find()
[
  {
    _id: 1,
    pelicula: 'El joven manos de tijera',
    'año_estreno': 1990,
    recaudacion: '86 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'Disney+'
  },
  {
    _id: 2,
    pelicula: 'Entrevista con el vampiro',
    'año_estreno': 1994,
    recaudacion: '223,7 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'HBOmax'
  },
  {
    _id: 3,
    pelicula: 'Beetlejuice',
    'año_estreno': 1988,
    recaudacion: '74,2 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'HBOmax'
  },
  {
    _id: 4,
    pelicula: 'Transformers',
    'año_estreno': 2007,
    recaudacion: '86 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'HBOmax'
  },
  {
    _id: 5,
    pelicula: 'Tom y Jerry',
    'año_estreno': 2021,
    recaudacion: '136,5 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'HBOmax'
  },
  {
    _id: 6,
    pelicula: 'Kung Fu Panda',
    'año_estreno': 2008,
    recaudacion: '631,7 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'Netflix'
  }
]

```

- trae los documentos que tiene el año de estreno el año 2007

```

pelis_estreno> db.estreno.find({'año_estreno': 2007})
[
  {
    _id: 4,
    pelicula: 'Transformers',
    'año_estreno': 2007,
    recaudacion: '86 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'HBOmax'
  }
]

```

- trae los documentos que tiene el año de estreno mayor del 2000


```

pelis_estreno> db.estreno.find({año_estreno: {$gt: 2007}})
[
  {
    _id: 4,
    película: 'Transformers',
    'año_estreno': 2007,
    recaudacion: '86 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'HBOmax'
  },
  {
    _id: 5,
    película: 'Tom y Jerry',
    'año_estreno': 2021,
    recaudacion: '136,5 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'HBOmax'
  },
  {
    _id: 6,
    película: 'Kung Fu Panda',
    'año_estreno': 2008,
    recaudacion: '631,7 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'Netflix'
  }
]

```

- trae los documentos que tiene el año de estreno mayor o igual que 2007

```

pelis_estreno> db.estreno.find({año_estreno: {$gte: 2007}})
[
  {
    _id: 4,
    película: 'Transformers',
    'año_estreno': 2007,
    recaudacion: '86 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'HBOmax'
  },
  {
    _id: 5,
    película: 'Tom y Jerry',
    'año_estreno': 2021,
    recaudacion: '136,5 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'HBOmax'
  },
  {
    _id: 6,
    película: 'Kung Fu Panda',
    'año_estreno': 2008,
    recaudacion: '631,7 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'Netflix'
  }
]

```

```
db.estreno.find({año_estreno: {$gte: 2007}})
```

- va a traer los documentos que tienen los años de estreno mayor o igual a 2007 y que su plataforma sea HBOmax

```

pelis_estreno> db.estreno.find({año_estreno: {$gte: 2007}, plataforma: 'HBOmax'})
[
  {
    _id: 4,
    película: 'Transformers',
    'año_estreno': 2007,
    recaudacion: '86 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'HBOmax'
  },
  {
    _id: 5,
    película: 'Tom y Jerry',
    'año_estreno': 2021,
    recaudacion: '136,5 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'HBOmax'
  }
]

```

db.estreno.find({año_estreno: {\$gte: 2007}, plataforma: 'HBOmax'})

- encuentra y muestra donde la plataforma corresponda a Disney+ y que ordene los documentos de forma ascendente

```

pelis_estreno> db.estreno.find({'plataforma':'Disney+'}).sort({'_id':1}).pretty()
[
  {
    _id: 1,
    película: 'Edward Scissorhands',
    'año_estreno': 1990,
    recaudacion: '86 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'Disney+'
  },
  {
    _id: 4,
    película: 'Transformers',
    'año_estreno': 2007,
    recaudacion: '86 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'Disney+'
  }
]

```

MODIFICACIONES

MODIFICACIONES DE DIRECTOR

- quiere decir que cambiará lo que hay dentro del file películas donde el id sea 1

```
pelis_estreno> db.director.update({'_id':1},{ $set:{'peliculas':'Batman, Beetlejuice'}})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
pelis_estreno> db.director.find({'_id': 1})
[
  {
    _id: 1,
    nombre: 'Tim Burton',
    ocupacion: 'director de cine, productor, escritor y dibujante',
    peliculas: 'Batman, Beetlejuice',
    nacionalidad: 'Estadounidense'
  }
]
```

- modificar la nacionalidad del id 2, solamente se quitó la tilde

```
pelis_estreno> db.director.update(
... {'_id':2},
... { $set:
... {'nacionalidad':'irlandes'}
... }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
pelis_estreno> db.director.find({'_id': 2})
[
  {
    _id: 2,
    nombre: 'Neil Jordan',
    ocupacion: 'director de cine y novelista',
    peliculas: 'El juego de las lagrimas, L
    nacionalidad: 'irlandes'
  }
]
```

- se modifica donde la películas sean solamente una, buenos muchachos; en el id 3

```
pelis_estreno> db.director.update(
... {'_id':3},
... {'$set':
... {'peliculas':'Buenos muchachos'}}
... )
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
pelis_estreno> db.director.find({'_id': 3})
[
  {
    _id: 3,
    nombre: 'Martin Charles Scorsese',
    ocupacion: 'director, guionista y p',
    peliculas: 'Buenos muchachos',
  }
]
```

- se modifica donde el id sea 4 y la ocupación del director sea simplemente guionista y productor, quitando la de director

```
pelis_estreno> db.director.update(
... {'_id':4},
... {'$set':
... {'ocupacion':'guionista y productor de cine'}}
... )
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
pelis_estreno> db.director.find({'_id': 4})
[
  {
    _id: 4,
    nombre: 'Steven Spielberg',
    ocupacion: 'guionista y productor de cine',
  }
]
```

- se modifica donde el id sea 5 y que incluya parte del nombre del director

```

pelis_estreno> db.director.update(
... {'_id':5},
... {'$set':
... {'nombre':'Quentin Jerome Tarantino'}}
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
pelis_estreno>

```

```

pelis_estreno> db.director.find({'_id': 5})
[
  {
    _id: 5,
    nombre: 'Quentin Jerome Tarantino',
  }
]

```

- se modifica donde el id sea 1 y que la nacionalidad cambie de Estadounidense a estadounidense

```

pelis_estreno> db.director.update(
... {'_id':1},
... {'$set':
... {'nacionalidad':'estadounidense'}}
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
pelis_estreno>

```

```

pelis_estreno> db.director.find({'_id': 1})
[
  {
    _id: 1,
    nombre: 'Tim Burton',
    ocupacion: 'director de cine, productor, escritor y',
    peliculas: 'Batman, Beetlejuice',
    nacionalidad: 'estadounidense'
  }
]

```

MODIFICACIONES DE PELÍCULAS

- modificara en la colección de pelicula el file actor_princ por personaje_princ

```
db.pelicula.updateMany({}, {$rename: {}})
```

```
{
  "_id": 1,
  "nombre": "El joven manos de tijera",
  "genero": "fantasia romantica",
  "duracion": 126,
  "personaje_princ": "Johnny Deep"
},
{
  "_id": 2,
  "nombre": "Entrevista con el vampiro",
  "genero": "suspenso, drama, romance",
  "duracion": 122,
  "personaje_princ": "Tom Cruice y Brad Pitt"
},
{
  "_id": 3,
  "nombre": "Beetlejuice",
  "genero": "comedia, fantasia",
  "duracion": 92,
  "personaje_princ": "Winona Ryder y Michael Keaton"
},
{
  "_id": 4,
  "nombre": "Transformers",
  "genero": "accion, ficcion",
  "duracion": 144,
  "personaje_princ": "Shia LaBeouf"
},
{
  "_id": 5,
  "nombre": "Tom y Jerry",
  "genero": "aventura, comedia, familiar",
  "duracion": 101,
  "personaje_princ": "Tom y Jerry"
},
{
  "_id": 6,
  "nombre": "Kung Fu Panda",
  "genero": "infantil, animacion, aventura",
  "duracion": 91,
  "personaje_princ": "panda Po"
}
]
pelis_estreno> db.pelicula.find()
```

- cambia en el id 1 el genero “fantasia romantico” a “fantasia, romance”

```

pelis_estreno> db.pelicula.update(
... {'_id':1},
... {'$set':
... {'genero':'fantasia, romance'}
... }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}

```

```

pelis_estreno> db.pelicula.find({'_id':1})
[
  {
    _id: 1,
    nombre: 'El joven manos de tijera',
    genero: 'fantasia, romance',
    duracion: 126,
    personaje_princ: 'Johnny Deep'
  }
]

```

- cambiará los personajes principales en el id 1

```

pelis_estreno> db.pelicula.update(
... {'_id':1},
... {'$set':
... {'personaje_princ':'Johnny Deep y Winona Ryder'}
... }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}

```

```

pelis_estreno> db.pelicula.find({'_id':1})
[
  {
    _id: 1,
    nombre: 'El joven manos de tijera',
    genero: 'fantasia, romance',
    duracion: 126,
    personaje_princ: 'Johnny Deep y Winona Ryder'
  }
]

```

- se cambia el idioma del nombre en el id 2

```

pelis_estreno> db.pelicula.update(
... {'_id':2},
... {'$set':
... {'nombre':'interview with the vampire'}
... }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
pelis_estreno> db.pelicula.find({'_id':2})
[
  {
    _id: 2,
    nombre: 'interview with the vampire',
    genero: 'suspense, drama, romance',
    duracion: 122,
    personaje_princ: 'Tom Cruise y Brad Pitt'
  }
]

```

- se agregara un genero en el id 3

```

pelis_estreno> db.pelicula.update(
... {'_id':3},
... {'$set':
... {'genero':'comedia, fantasia, historia de fantasmas'}
... }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
pelis_estreno> db.pelicula.find({'_id':3})
[
  {
    _id: 3,
    nombre: 'Beetlejuice',
    genero: 'comedia, fantasia, historia de fantasmas',
    duracion: 92,
    personaje_princ: 'Winona Ryder y Michael Keaton'
  }
]

```

- se haran dos modificaciones, el nombre y la duracion en el id 4


```

pelis_estreno> db.pelicula.updateMany(
... {'_id':4},
... {'$set':
... {'nombre':'Transformers: El lado oscuro de la Luna', duracion: 154}
... })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
pelis_estreno> db.pelicula.find({'_id':4})
[
  {
    _id: 4,
    nombre: 'Transformers: El lado oscuro de la Luna',
    genero: 'accion, ficcion',
    duracion: 154,
    personaje_princ: 'Shia LaBeouf'
  }
]

```

- se cambio el genero de “familia” a “animacion” en el id 5

```

pelis_estreno> db.director.update(
... {'_id':5},
... {'$set':
... {'genero':'aventura, comedia, animacion'}
... })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
pelis_estreno> db.pelicula.find({'_id':5})
[
  {
    _id: 5,
    nombre: 'Tom y Jerry',
    genero: 'aventura, comedia, familiar',
    duracion: 101,
    personaje_princ: 'Tom y Jerry'
  }
]

```

- se cambia los personajes principales de la pelicula del id 6

```

pelis_estreno> db.pelicula.update( { '_id': 6 }, { $set: { 'personaje_princ': 'Po, maestro Shifu, Tai Lung' } })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
pelis_estreno> db.director.find({_id:6})
[
  {
    _id: 6,
    nombre: 'Michael Bay',
    ocupacion: 'cineasta',
    peliculas: 'Transformers, Ambulance, Armagedon',
    nacionalidad: 'estadounidense',
    personaje_princ: 'Po, maestro Shifu, Tai Lung'
  }
]
pelis_estreno>

```

MODIFICACIONES DE ESTRENO

- se cambia el idioma de la película

```

pelis_estreno> db.estreno.update(
... { '_id':1},
... {$set:
... {'pelicula':'Edward Scissorhands'}
... })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
pelis_estreno> db.estreno.find({_id:1})
[
  {
    _id: 1,
    pelicula: 'Edward Scissorhands',
    'año_estreno': 1990,
    recaudacion: '86 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'Disney+'
  }
]

```

- se cambia el año de estreno de la película del id 2

```

]
pelis_estreno> db.estreno.update(
... {'_id':2},
... {'$set':
... {'año_estreno':'1995'}}
... })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
pelis_estreno> db.estreno.find({'_id':2})
[
  {
    _id: 2,
    pelicula: 'Entrevista con el vampiro',
    'año_estreno': '1995',
  }
]

```

- se cambia el dinero de la recaudacion dando un valor de 75 millones de dolares en el id 3

```

pelis_estreno> db.estreno.update(
... {'_id':3},
... {'$set':
... {'recaudacion':'75 millones USD'}}
... })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
pelis_estreno> db.estreno.find({'_id':3})
[
  {
    _id: 3,
    pelicula: 'Beetlejuice',
    'año_estreno': 1988,
    recaudacion: '75 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'HBOmax'
  }
]

```

- se modificara la plataforma del id 4 y se pondra en vez de HBOmax a Disney+

```

pelis_estreno> db.estreno.update(
... {'_id':4},
... {'$set':
... {'plataforma':'Disney+'}}
... })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
pelis_estreno> db.estreno.find({'_id':4})
[
  {
    _id: 4,
    pelicula: 'Transformers',
    'año_estreno': 2007,
    recaudacion: '86 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'Disney+'
  }
]

```

- se modifíco la plataforma del id 5

```

mpelis_estreno> db.estreno.update(
... {'_id':5},
... {'$set':
... {'plataforma':'Amazon Prime'}}
... })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}

```

```

pelis_estreno> db.estreno.find({'_id':5})
[
  {
    _id: 5,
    pelicula: 'Tom y Jerry',
    'año_estreno': 2021,
    recaudacion: '136,5 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'Amazon Prime'
  }
]

```

- se modificó donde el dinero de la recaudación sea un valor aproximado

```

pelis_estreno> db.estreno.update(
... {_id:6},
... {$set:
... {'recaudacion':'632 millones USD'}
... })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
pelis_estreno> db.estreno.find({_id:6})
[
  {
    _id: 6,
    pelicula: 'Kung Fu Panda',
    'año_estreno': 2008,
    recaudacion: '632 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'Netflix'
  }
]

```

ELIMINAR

- en este método lo que hace es eliminar un documento de la colección correspondiente, se hacen las eliminaciones en cada colección
- en el primer “eliminar”, lo que se logra es eliminar el documento donde el id sea 6

```

pelis_estreno> db.director.deleteOne({'_id':6})
{ acknowledged: true, deletedCount: 1 }
pelis_estreno> db.director.find()
[
  {
    _id: 1,
    nombre: 'Tim Burton',
    ocupacion: 'director de cine, productor, escritor y dibujante',
    peliculas: 'Batman, Beetlejuice',
    nacionalidad: 'estadounidense'
  },
  {
    _id: 2,
    nombre: 'Neil Jordan',
    ocupacion: 'director de cine y novelista',
    peliculas: 'El juego de las lagrimas, La viuda, Entrevista con el vampiro',
    nacionalidad: 'irlandes'
  },
  {
    _id: 3,
    nombre: 'Martin Charles Scorsese',
    ocupacion: 'director, guionista y productor de cine',
    peliculas: 'Buenos muchachos',
    nacionalidad: 'estadounidense'
  },
  {
    _id: 4,
    nombre: 'Steven Spielberg',
    ocupacion: 'guionista y productor de cine',
    peliculas: 'Tiburón, Parque Jurasico, E.T., el extraterrestre',
    nacionalidad: 'estadounidense'
  },
  {
    _id: 5,
    nombre: 'Quentin Jerome Tarantino',
    ocupacion: 'director de cine, productor, guionista, editor y actor',
    peliculas: 'Kill Bill, Bastardos sin gloria, Tiempos violentos',
    nacionalidad: 'estadounidense',
    genero: 'aventura, comedia, animacion'
  }
]

```

- en este caso lo unico que cambia del anterior es que se eliminara el documento del id 3

```

pelis_estreno> db.director.deleteOne({'_id':3})
{ acknowledged: true, deletedCount: 1 }
pelis_estreno> db.director.find()
[
  {
    _id: 1,
    nombre: 'Tim Burton',
    ocupacion: 'director de cine, productor, escritor y dibujante',
    peliculas: 'Batman, Beetlejuice',
    nacionalidad: 'estadounidense'
  },
  {
    _id: 2,
    nombre: 'Neil Jordan',
    ocupacion: 'director de cine y novelista',
    peliculas: 'El juego de las lagrimas, La viuda, Entrevista con el vampiro',
    nacionalidad: 'irlandes'
  },
  {
    _id: 4,
    nombre: 'Steven Spielberg',
    ocupacion: 'guionista y productor de cine',
    peliculas: 'Tiburón, Parque Jurasico, E.T., el extraterrestre',
    nacionalidad: 'estadounidense'
  },
  {
    _id: 5,
    nombre: 'Quentin Jerome Tarantino',
    ocupacion: 'director de cine, productor, guionista, editor y actor',
    peliculas: 'Kill Bill, Bastardos sin gloria, Tiempos violentos',
    nacionalidad: 'estadounidense',
    genero: 'aventura, comedia, animacion'
  }
]
pelis_estreno>

```

- se elimina el documento en la colección película con el id 2

```

pelis_estreno> db.pelicula.deleteOne({'_id':2})
{ acknowledged: true, deletedCount: 1 }
pelis_estreno> db.pelicula.find()
[
  {
    _id: 1,
    nombre: 'El joven manos de tijera',
    genero: 'fantasia, romance',
    duracion: 126,
    personaje_princ: 'Johnny Deep y Winona Ryder'
  },
  {
    _id: 3,
    nombre: 'Beetlejuice',
    genero: 'comedia, fantasia, historia de fantasmas',
    duracion: 92,
    personaje_princ: 'Winona Ryder y Michael Keaton'
  },
  {

```

- se elimina el documento de la misma colección de la anterior con el id 5

```

pelis_estreno> db.pelicula.deleteOne({'_id':5})
{ acknowledged: true, deletedCount: 1 }
pelis_estreno> db.pelicula.find()
[
  {
    _id: 1,
    nombre: 'El joven manos de tijera',
    genero: 'fantasia, romance',
    duracion: 126,
    personaje_princ: 'Johnny Deep y Winona Ryder'
  },
  {
    _id: 3,
    nombre: 'Beetlejuice',
    genero: 'comedia, fantasia, historia de fantasmas',
    duracion: 92,
    personaje_princ: 'Winona Ryder y Michael Keaton'
  },
  {
    _id: 4,
    nombre: 'Transformers: El lado oscuro de la Luna',
    genero: 'accion, ficcion',
    duracion: 154,
    personaje_princ: 'Shia LaBeouf'
  },
  {
    _id: 6,
    nombre: 'Kung Fu Panda',
    genero: 'infantil, animacion, aventura',
    duracion: 91,

```

- se elimina de la colección de estrenos el documento con el nombre de la película Tom y Jerry

```
pelis_estreno> db.estreno.deleteOne({'pelicula':'Tom y Jerry'})
{ acknowledged: true, deletedCount: 1 }
pelis_estreno> db.estreno.find()
[
  {
    _id: 1,
    pelicula: 'Edward Scissorhands',
    'año_estreno': 1990,
    recaudacion: '86 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'Disney+'
  },
  {
    _id: 2,
    pelicula: 'Entrevista con el vampiro',
    'año_estreno': '1995',
    recaudacion: '223,7 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'HBOmax'
  },
  {
    _id: 3,
    pelicula: 'Beetlejuice',
    'año_estreno': 1988,
    recaudacion: '75 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'HBOmax'
  },
  {
    _id: 4,
    pelicula: 'Transformers',
    'año_estreno': 2007,
    recaudacion: '86 millones USD',
    lugar_estreno: 'Estados Unidos',
    plataforma: 'Disney+'
  },
  {
    _id: 6,
    pelicula: 'Kung Fu Panda',
    'año_estreno': 2008
  }
]
```

Como podemos ver, ninguno de los documentos eliminados se ven en la consulta general de cada colección, para poder volver a verlos es creandolos de nuevo.