

Proyecto final MongoDB 1º Trimestre



Índice:

1. Introducción
2. Explicación inserts
3. Explicación consultas

1.- Introducción

El trabajo que realizaré en este fin de trimestre, trata sobre la venta de ordenadores sobremesa.

Habrà todo tipo de ordenadores que sirvan para ofimática, gaming...

Dentro de los inserts contendrá el nombre del modelo del ordenador y los componentes de ese ordenador y las características.

Dentro de las consultas.js, tengo todas las consultas que hemos dado durante todo el tiempo de estudio. A continuación explicaré cómo hacerlas y en qué problemas me he basado para hacerlas. Aparte de lo aprendido en clase yo he aportado varias consultas que he investigado para comprobar cómo funciona.

2.- Explicación de inserts

Para comenzar los inserts he creado un db.ordenadores.insertOne() para enseñar con una buena presentación como he metido cada dato.

He utilizado:

- Campo documento: Para poner cualquier tipo de palabra
`Modelo:"MSI MAG Codex 5 11TC-668XIB"`
- Campo de números: Lo he puesto para poner números, en este caso lo he utilizado para poner un precio
`precio:1499.9`
- Boleano: Lo he puesto para definir si un campo es si o no
`LEDs: true`
- New Date : Lo he puesto para poner una fecha
`{FechaInicio: new Date("2020-01-01")}`
- Array de fechas: He utilizado este array para determinar cuándo se puede comprar el ordenador y el final de compra porque ya se ha quedado obsoleto comparado a los demás. pero este array lo puse para poner varias fechas

```
Rangodefechadecompra:[{FechaInicio: new Date("2020-01-01")}, {fechaExpedicion: new Date("2025-02-04")}]
```

- Array de números: Lo he utilizado para determinar las versiones de los puertos USB y esto sirve para poner varios números

```
TipoUSB: [2.0, 3.0, 3.1]
```

- Array de documentos:

3.- Explicación de las consultas

Al principio utilizaré un operador por consulta para explicar cómo funciona, después de eso pondré consultas más complicadas.

1º Consulta \$eq: Se utiliza para poner los valores iguales a los que buscas

```
db.ordenadores.find( { "Marca": { $eq: "MSI" } } )
```

2º Consulta \$gt: Se utiliza para poner valores que son mayores que un valor especificado.

```
db.ordenadores.find( { "precio": { $gt: 800 } } ).pretty()
```

3º Consulta \$gte: Se utiliza para poner valores que son mayores o iguales a un valor especificado

```
db.ordenadores.find( { "precio": { $gte: 1000 } } ).pretty()
```

4º Consulta \$in: Se utiliza para buscar cualquiera de los valores especificados en una matriz.

```
db.ordenadores.find( { "Componentes.Almacenamiento.TB": { $in: [ 2, 2.5, 3 ] } } )
```

5º Consulta \$lt: Se utiliza para buscar valores que son menores que un valor especificado.

```
db.ordenadores.find( { "precio": { $lt: 500 } } ).pretty()
```

6º Consulta \$lte: Se utiliza para buscar valores que son menores o iguales a un valor especificado.

```
db.ordenadores.find( { "precio": { $lte: 400 } } ).pretty()
```

7º Consulta \$ne: Se utiliza para buscar los valores que no son iguales a un valor especificado.

```
db.ordenadores.find( { "Componentes.Refrigeracion.Tipo": { $ne: "Líquida" } } ).pretty()
```

8º Consulta \$nin: Se utiliza para buscar valores que no coinciden con ninguno de los valores especificados en una matriz.

```
db.ordenadores.find( { "Componentes.Placabase.Puertos": { $nin: [ "USB 3.0", "Displayport" ] } } )
```

9º Consulta \$and: Se usa para utilizar 2 operadores a la vez: Ejemplo:

```
db.ordenadores.find( { $and: [
  { "Componentes.Placabase.Puertos": { $nin: [ "USB 3.0", "Displayport" ] } },
  { "Componentes.Placabase.Puertos": { $regex: /^HD/i } } ] } ).pretty()
```

10º Consulta \$not: Se utiliza para buscar valores contrarios a los especificados en la matriz

```
db.ordenadores.find( { $nor: [ { "Componentes.Caja.Ventiladores": { $not: { $lt: 4 } } },
  { "Componentes.Refrigeracion.Ventiladores": { $not: { $lt: 2 } } } ] } ).pretty()
```

11º Consulta \$nor: Se utiliza para que te devuelva lo contrario de uno de los valores especificados

```
db.ordenadores.find({ $nor: [ {"Componentes.Caja.Ventiladores":{$not:{$lt: 4}}}, {"Componentes.Refrigeracion.Ventiladores":{$not:{$lt:2}}}] }).pretty()
```

12º Consulta \$or: Se utiliza para buscar uno de los dos valores especificados

```
db.ordenadores.find( { $or: [ { "Precio": { $lt: 700 } }, { "Marca": "MSI" } ] } )
```

13º Consulta \$exists: Se utiliza para buscar coincide con documentos que tienen el campo especificado.

```
db.ordenadores.find({"Componentes.Placabase,LEDs":{$exists:false}}).pretty()
```

14º Consulta \$regex: Se utiliza para buscar documentos donde los valores coinciden con una expresión regular especificada.

```
db.ordenadores.find({ $and: [ {"Componentes.Placabase.Puertos":{$nin:["USB 3.0","Displayport"]}}, {"Componentes.Placabase.Puertos":{$regex:/^HD/i}}] }).pretty()
```

15º Consulta \$size: Se utiliza para buscar documentos si el campo de matriz tiene un tamaño especificado.

```
db.ordenadores.find({ $and: [ {"Componentes.Caja.TipoUSB":{$size:3}}, {"Marca":{$eq:"MSI"}} ] } ).pretty()
```

Ejemplos de consultas más avanzadas y para qué sirven:

Para buscar ordenadores que no tenga 4 ventiladores en la caja o que no tenga 2 ventiladores la refrigeracion.

```
db.ordenadores.find({ $nor: [ {"Componentes.Caja.Ventiladores":{$not:{$lt: 4}}}, {"Componentes.Refrigeracion.Ventiladores":{$not:{$lt:2}}}] }).pretty()
```

Esto sirve para buscar ordenadores de menos de 700 € y que mi ordenador sea marca MSI

```
db.ordenadores.find( { $or: [ { "Precio": { $lt: 700 } }, { "Marca": "MSI" } ] } ).pretty()
```

Este sirve para buscar los USB de tipo 1.0 hasta los USB de tipo 3.1:

```
db.ordenadores.find({"Componentes.Caja.TipoUSB":{"$elemMatch":{"$gte:1.0,$lt:3.1 }}}).pretty()
```

Aportaciones:

sort: Sirve para ordenar en este caso ordena del ordenador más caro al más barato

```
db.ordenadores.find({}).sort ({"precio":1});
```

Create Index: Esto sirve para escanear cada documento en una colección, para seleccionar aquellos documentos que coincidan con la declaración de consulta y te lo resume con un índice.

```
db.ordenadores.createIndex({precio:500})
```

UpdateMany: Este operador sirve para actualizar uno o varios documentos para cambiar sus valores, \$set sirve para indicar el cambio que se va a realizar

```
db.ordenadores.updateMany(  
  { Marca: "MSI" },  
  { $set: { Precio: 500 } }  
)
```

DeleteMany: La función de este operador es borrar los documentos definidos

```
db.ordenadores.deleteMany(  
  { Marca: "MSI"}  
)
```