

PROYECTO PRIMER TRIMESTRE BASES DE DATOS NO RELACIONALES

En este proyecto vamos a utilizar MongoDB, que es una base de datos distribuida, basada en documentos y de uso general, es una base de datos documental, lo que significa que almacena datos en forma de documentos tipo JSON los cuales usaremos para guardar nuestras inserciones de datos y nuestras consultas.

Yo he creado mi propia base de datos ya que creo que es más didáctico que copiarla de internet, mi base de datos está inspirada en la estructura de la clase, con 15 ordenadores en uso más el del profesor, en total 16, cada ordenador tiene dedicados en la base de datos una serie de campos, de los distintos tipos que hemos visto en clase, como pueden ser, tipo texto, tipo entero, tipo documento...

La base de datos tiene de nombre clase, y la colección en la que trabajaremos se llamará ordenadores. Cada ordenadores dispone de los campos:

- Nombre, de tipo cadena de caracteres
- RAM, de tipo entero
- Procesador, de tipo documento, con sus correspondientes campos:
 - Marca, de tipo cadena de caracteres
 - Modelo, de tipo cadena de caracteres
 - Núcleos, de tipo entero
- Teclado, de tipo Booleano
- Ratón, de tipo Booleano

El proyecto tiene la siguiente estructura de directorios:



Un directorio principal al que he llamado proyecto01 y dentro de este, podemos distinguir dos directorios, src para contener los archivos js y docs, para contener este documento pdf y un archivo readme.md con una breve introducción al proyecto

Tras explicar que es mongo y como tenemos estructurado el proyecto vamos a pasar a explicar las inserciones y consultas que he realizado.

Las inserciones las he hecho en dos veces, primero inserte un array de documentos con el que cree la base de datos y la colección, primero use el comando de mongo **use clase** para cambiar la base de datos en uso a una nueva que estuviera vacía, y una vez hecho esto cargue el documento que contenía las inserciones con load("inserciones.js"), me di cuenta de que faltaba un equipo así que lo inserté posteriormente con un insertOne()

```
db.ordenadores.insertMany([
  { nombre: "pc-01", ram: 8, procesador: { marca: "intel", modelo: "i3", nucleos: 2 }, teclado : true , raton: false},
  { nombre: "pc-02", ram: 12, procesador: { marca: "intel", modelo: "i5", nucleos: 4 }, teclado : true , raton: true},
  { nombre: "pc-03", ram: 6, procesador: { marca: "amd", modelo: "ryzen3", nucleos: 4 }, teclado : false , raton: false},
  { nombre: "pc-04", ram: 16, procesador: { marca: "intel", modelo: "pentium", nucleos: 2 }, teclado : true , raton: true},
  { nombre: "pc-05", ram: 8, procesador: { marca: "amd", modelo: "ryzen5", nucleos: 6 }, teclado : true , raton: false},
  { nombre: "pc-06", ram: 64, procesador: { marca: "intel", modelo: "xeon", nucleos: 12 }, teclado : false , raton: false},
  { nombre: "pc-07", ram: 6, procesador: { marca: "intel", modelo: "i3", nucleos: 4 }, teclado : false , raton: true},
  { nombre: "pc-08", ram: 4, procesador: { marca: "amd", modelo: "ryzen3", nucleos: 4 }, teclado : false , raton: true},
  { nombre: "pc-09", ram: 10, procesador: { marca: "intel", modelo: "pentium", nucleos: 2 }, teclado : true , raton: true},
  { nombre: "pc-10", ram: 16, procesador: { marca: "amd", modelo: "ryzen3", nucleos: 4 }, teclado : true , raton: true},
  { nombre: "pc-11", ram: 8, procesador: { marca: "amd", modelo: "ryzen5", nucleos: 6 }, teclado : true , raton: true},
  { nombre: "pc-12", ram: 8, procesador: { marca: "intel", modelo: "i3", nucleos: 4 }, teclado : false , raton: true},
  { nombre: "pc-13", ram: 4, procesador: { marca: "intel", modelo: "i5", nucleos: 6 }, teclado : false , raton: true},
  { nombre: "pc-14", ram: 8, procesador: { marca: "amd", modelo: "fx", nucleos: 4 }, teclado : true , raton: true},
  { nombre: "pc-15", ram: 6, procesador: { marca: "intel", modelo: "i3", nucleos: 2 }, teclado : true , raton: true}
]);
```

```
db.ordenadores.insertOne({ nombre: "pc-profesor", ram: 8, procesador: { marca: "amd", modelo: "threadripper", nucleos: 16}, teclado:true, raton:false})
```

Y ahora empezaré a explicar las consultas

La primera consulta es muy pobre y muy simple, es un find, que nos devolverá como resultado los ordenadores que tengan procesadores marca intel.

```
db.ordenadores.find({"procesador.marca" : "intel"})
```

En la segunda hemos utilizado una consulta empleando un campo de tipo Booleano, esta consulta nos devolverá como resultado los ordenadores que tengan teclado.

```
db.ordenadores.find({teclado:true})
```

Esta tercera se complica un poco, ya que utilizaremos en la consulta el operador \$regex, que nos permite utilizar expresiones regulares en nuestras consultas, este find nos mostrará los equipos que empiecen por la cadena de caracteres pc-1.

```
db.ordenadores.find({nombre: {$regex : /^pc-1/ }})
```

Para la siguiente hemos utilizado los operadores \$gt y \$and, al realizar esta consulta nos mostrará los equipos de la clase que tengan más de 8GB de RAM y más de 2 núcleos.

```
db.ordenadores.find({ram:{$gt:8},"procesador.nucleos":{$gt:2}})
```

En la quinta consulta vamos a emplear un operador \$and para mostrar los ordenadores que tengan un i3 y sea necesario instalarle un teclado.

```
db.ordenadores.find({"procesador.modelo":"i3", teclado:false})
```

Para la sexta vamos a realizar una consulta con el operador \$lt que nos mostrará los ordenadores que necesiten una ampliación de ram, ya que 8GB a estas alturas es una cantidad pobre.

```
db.ordenadores.find({ram:{$lt:8}})
```

En la séptima vamos a realizar una consulta utilizando el parámetro \$nin, para que se nos muestre como resultado de la consulta los equipos que no tengan ni un i3 ni un ryzen3

```
db.ordenadores.find({"procesador.modelo":{$nin:["i3", "ryzen3" ]}})
```

Ahora vamos a ver los equipos de intel que no sean ni un i3, ni un i5, ni un i7 en el caso de que hubiese uno, para no realizar una búsqueda empleando las cadenas i3, i5 e i7 vamos a hacer uso de las expresiones regulares, y nos quedaría tal que así:

```
db.ordenadores.find({"procesador.marca": "intel", "procesador.modelo": { $not: /^i/ }})
```

En la siguiente consulta vamos a utilizar los parámetros \$or, \$lt y \$eq, para mostrar como resultado los equipos con menos de 8 o 8 de ram y los que tengan 16, para saber a los que se les puede quitar ram y a los que se los podemos instalar.

```
db.ordenadores.find({ $or:[{ram:{$lte:8}}, {ram:{$eq:12}}]})
```

Como no he implementado aún el operador \$in lo haré ahora para mostrar los equipos que tengan instalados un i3 o un i5

```
db.ordenadores.find( { "procesador.modelo": { $in: [ "i5", "i3" ] } } )
```

Por último quería comprobar si había un equipo que no siguiera el patrón que siguen todos los equipos en su nombre, siendo este, pc-numeros, y esto se puede hacer gracias a las expresiones regulares y al operador \$not

```
db.ordenadores.find({ nombre: {$not : /^pc-[0-9]/ }})
```