

NOTA: Son Soluciones Didácticas para mostrar diferentes funcionalidades. Puede haber otras soluciones optimizadas

**** ===== Apartado 1.- =====

- Documentos con Instrucciones y Ejemplos para empezar con MongoDB

*** ===== - Apartado 2.- =====

**** a) De cada Tema:

```
db.aficiones.distinct("Tema") // Lista de temas sin repetir que tengo en la BD
```

```
db.aficiones.distinct("Nombre")    //-- Qué Nombres sin repetir de documentos
```

•

```
****      1. Obtener el nombre, el tema y la puntuación de los componentes mejor
valorados (puntuaciones > = 9) pero no saques el identificador.
```

```
db.aficiones.find({"Puntuacion" :{$gte:
9}}, {"Nombre":1,"Tema":1,"Puntuacion":1, _id:0}).pretty()
```

```
mayor9= micole.find ( { Puntuacion : { $gt: 9 } },{}); // ---- puedo comprobar
si ha salido bien
```

```
if (mayor9) {print("fue bien")} ; // compruebo si ha ido bien
```

•

**** 2. Obtener para cada tema (agrupado): su nombre, el total de cuanto te gastarías si vas a todos los componentes mejor valorados y cuántos componentes has incluido.

```
----- V 1 ----- da el nombre del Tema, la suma y cuántos
hay de ese Tema
db.aficiones.aggregate([
  { $match: { "Puntuacion":{$gte:9} } },
  { $group: { _id: "$Tema",
               total: { $sum: "$Precio" },
               count: { $sum: 1 } } },
  {$sort: { _id:1}}      //clasifico por _id ascendente
                        //      (no pedido())
])
```

```
// hay dos temas que tienen el precio mal: baloncesto, Futbol
// (da mucha cantidad)
```

```
----- NO es lo pedido : para dar solo un total
```

```
db.aficiones.aggregate([
  {$match:{"Puntuacion": {$gt:9}}},
  {$group: {_id: null, PrecioTotal : { $sum : "$Precio"}}}
])
```

.
**** 3. Obtener los componentes valorados para cada uno de los valores
(10,9,8,7,6 y 5) por separado. Obtener los nombres de componentes con ese valor.

----- V1 -----

```
db.aficiones.aggregate(
  {$match : { $or : [ {Puntuacion : {$eq : 10}},
                      {Puntuacion : {$eq : 9}},
                      {Puntuacion : {$eq : 8}},
                      {Puntuacion : {$eq : 7}},
                      {Puntuacion : {$eq : 6}} ] }},
  {$group : { _id : "$Puntuacion", componentes : {$push : "$Nombre"} }
  })
```

----- V2: Si quiero incluir el Tema y Nombre de cada componente

```
db.aficiones.aggregate(
  {$match : { $or : [ {Puntuacion : {$eq : 10}},
                      {Puntuacion : {$eq : 9}},
                      {Puntuacion : {$eq : 8}},
                      {Puntuacion : {$eq : 7}},
                      {Puntuacion : {$eq : 6}}] }},
  {$group : { _id : "$Puntuacion",
              Componentes : {$push: {elTema: "$Tema", suNombre:
"$Nombre"} }
              } }
  })
```

.
**** 4. Lista de Apodos para cada afición (tema)

----- V1 -- NO repite items (necesita los corchetes)

```
db.aficiones.aggregate(
[
  {$group : { _id : "$Tema", Apodos: {$addToSet : "$Apodo"} }
]
)
```

----- V2 (repite apodos para cada docu)

```
db.aficiones.aggregate(
[
  {$group : { _id : "$Tema", Apodos: {$push : "$Apodo"} }
]
)
```

----- V3 (repite mucho) ---- sin agrupar (OK)

```
db.aficiones.aggregate([
  {$project: {"Apodo":1, "Tema":1, _id :0}}
])
```

.
**** b) Lista los Apodos, Componente y Tema en los que coincide al menos un componente (el nombre) del mismo Tema.

----- versión básica :

```
var cursor = null;
// hay docs sin Nombre -> usa exists
var cursor = db.aficiones.find({Nombre: {$exists: true}},
    {"Tema":1, "Apodo":1, "Nombre":1, _id: 0}).sort({"Tema":1,"Nombre":1});
var docu =cursor.next();
var docu2 =cursor.next();

while(cursor.hasNext()){
    if(docu.Tema == docu2.Tema){
        if(docu.Nombre==docu2.Nombre && docu.Apodo != docu2.Apodo){
            print(docu);
        }
        docu = docu2;
        docu2 = cursor.next();
    }
    else {
        docu = docu2;
        docu2 = cursor.next();
    }
}
```

```
/* ver y contar todos */
cursor.pretty(); //--- después del while está al final del cursor
cursor.count(); // --- pero sigue teniendo los mismos documentos

/* para probar necesitas incluir documentos que cumplan la condición
```

.
**** c) Repite la búsqueda anterior para puntuaciones intermedias: más de 4 y menos de 9.

----- V 1 -----

```
var cursor = null;
// hay docs sin Nombre
var cursor = db.aficiones.find({ Puntuacion: { $gt: 4, $lt:9 },
    Nombre: {$exists: true}},
    {"Tema":1, "Apodo":1, "Nombre":1, _id: 0}).sort({"Tema":1,"Nombre":1});
var docu =cursor.next();
var docu2 =cursor.next();

// ----- arreglo
while(cursor.hasNext()){
    if(docu.Tema == docu2.Tema){
        if(docu.Nombre==docu2.Nombre && docu.Apodo != docu2.Apodo)
        {
            print(docu);
            print(docu2);
        }
        docu = docu2;
    }
}
```

```

        docu2 = cursor.next();
    }
    else {
        docu = docu2;
        docu2 = cursor.next();
    }
}

```

.
 **** d) Describe al menos cuatro consultas interesantes para ti, escribe el código y ejecútalas
 **** e) Obtener todos los componentes de tu colección clasificados por tema. Para imprimirlos de una manera ordenada, usa un cursor que llama a una función sin nombre (definida dentro del cursor). Esa función imprime cada ítem que, previamente hemos transformado a json (así le damos un formato más legible)

```

db.aficiones.aggregate( [
  { $group: {
    _id: { tema: "$Tema" },
    count: { $sum: 1 },
    docs: { $addToSet : "$Nombre" }
  }}  ] ).forEach( function(myDoc) { print(myDoc);});

```

.
 **** f) Rebaja un 10% al precio de todos los componentes peor valorados (puntuación < 7).
 Y en la misma actualización añades el atributo Descuento a todas las aficiones que según la puntuación, si es buena de menos % de descuento.

```

db.aficiones.find().forEach(
function (myDoc) {
  var descuento = myDoc.Precio * 0.10;
  var porcentaje = (1 - myDoc.Puntuacion/10) * 10 ;

  if (myDoc.Puntuacion < 7 ){
    myDoc.Precio = myDoc.Precio - descuento;
  }
  myDoc.Descuento = porcentaje;
  print(myDoc);
  db.aficiones.save(myDoc);
});

```

.