

**APARTADO A.****A1)**

\$gte sería lo equivalente a  $\geq$ , para el caso de que el id no nos salga, bastará con ponerlo a 0, si solo queremos que salgan algunos datos específicos habrá que marcar los que queramos que salgan con 1. En este caso seleccionamos todos los datos excepto el id.

```
db.aficiones.find({Puntuacion:{$gte:9}},{_id:0})
```

Key	Value	Type
(1)	{ 8 fields }	Object
Tema	Fútbol	String
Apodo	Mike	String
Nombre	Cristiano Ronaldo	String
Puntuacion	9.0	Double
Precio	97000000.0	Double
Caracteristicas	{ 2 fields }	Object
ValorCalidad	9e-06	Double
Descuento	0.97	Double
(2)	{ 10 fields }	Object
Tema	Rugby	String
Apodo	Infir	String
Nombre	Torrejon R.C.	String
Puntuacion	10.0	Double
Precio	269.5	Double
direccion	[ 1 element ]	Array
Fundacion	2005.0	Double
Equipos	[ 1 element ]	Array
ValorCalidad	3.636363636364	Double
Descuento	0.98	Double

**A2).**

```
db.aficiones.aggregate({$match:{"Puntuacion":{$gte:9}}},{ "$group":{_id:"$Tema", TotalPrecio:{$sum:"$Precio"},Componentes:{$sum:1}}})
```

Para agrupar utilizaremos aggregate, primero buscaremos los componentes mejor valorados y posteriormente los agruparemos según el Tema, sumaremos el precio y los componentes de cada uno.

Key	Value	Type
(1) Videojuegos	{ 3 fields }	Object
_id	Videojuegos	String
TotalPrecio	179.8	Double
Componentes	4.0	Double
(2) Música	{ 3 fields }	Object
_id	Música	String
TotalPrecio	50.97	Double
Componentes	4.0	Double
(3) Baloncesto	{ 3 fields }	Object
_id	Baloncesto	String
TotalPrecio	135000000.0	Double
Componentes	3.0	Double
(4) Fútbol	{ 3 fields }	Object
_id	Fútbol	String
TotalPrecio	100003300.0	Double
Componentes	5.0	Double
(5) Series	{ 3 fields }	Object
(6) Rugby	{ 3 fields }	Object
(7) Libros	{ 3 fields }	Object
(8) musica	{ 3 fields }	Object

**A3)**

```
db.aficiones.aggregate({$match:{ $or: [ {"Puntuacion":10},
{"Puntuacion":9}, {"Puntuacion":8}, {"Puntuacion":7}, {"Puntuacion":6}, {"Puntuacion":5} ]
}}, {"$group":{_id:"$Puntuacion",Componentes:$sum:1}}})
```

En este caso tenemos que agregar por Puntuacion, pero solo por las que nos piden, que son 10,9,8,7,6,5 por lo primero realizamos una búsqueda de estas y posteriormente las agrupamos para ver que componentes la contienen.

Key	Value	Type
(1) 9.0	{ 2 fields }	Object
_id	9.0	Double
Componentes	15.0	Double
(2) 8.0	{ 2 fields }	Object
_id	8.0	Double
Componentes	13.0	Double
(3) 10.0	{ 2 fields }	Object
(4) 6.0	{ 2 fields }	Object
(5) 7.0	{ 2 fields }	Object
(6) 5.0	{ 2 fields }	Object

```
db.aficiones.aggregate({"$group":{_id:"$Temas",Apodos:{ $sum:"Apodo" }}})
```

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

**A4)**

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

Agregamos lo Temas y posteriormente usamos addToSet que lo que hace es añadir los apodos de cada tema a su respectivo array llamado Apodos dentro de cada tema.

Key	Value	Type
(1) Fútbol	{ 2 fields }	Object
_id	Fútbol	String
Apodos	[ 2 elements ]	Array
[0]	Mike	String
[1]	soFIFA	String
(2) Series	{ 2 fields }	Object
_id	Series	String
Apodos	[ 1 element ]	Array
[0]	Tyrion Lanister	String
(3) Baloncesto	{ 2 fields }	Object
(4) Música	{ 2 fields }	Object
(5) Libros	{ 2 fields }	Object
(6) Rugby	{ 2 fields }	Object
(7) musica	{ 2 fields }	Object
(8) Videojuegos	{ 2 fields }	Object
(9) MotoGP	{ 2 fields }	Object
(10) Ajedrez	{ 2 fields }	Object

**APARTADO B.**

```

var cursor = db.getCollection('aficiones').find({Nombre:{$exists:1}}, {Tema:1, Nombre:1,
Apodo:1, _id:0})
var docu = cursor.next();
var b = cursor.next();
while(cursor.hasNext()){
    //print("-----");
    if(docu.Tema == b.Tema && docu.Nombre==b.Nombre && docu.Apodo != b.Apodo){
        printjson(docu);
    }
    var docu = b;
    var b = cursor.next();
}

```

### APARTADO C.

Tenemos que cambiar la consulta principal para que solo busque puntuaciones intermedias para esto usaremos un and que mire las puntuaciones > 4 y >9: {\$and : [{Puntuacion:{\$gt:4}}, {Puntuacion:{\$lt:9}}]}.

```

var cursor = db.getCollection('aficiones').find({$and : [{Puntuacion:{$gt:4}},
{Puntuacion:{$lt:9}}, {Puntuacion: {$exists:true}}, {Nombre: {$exists:true}}]}, {"Tema":1,
"Nombre":1, "Apodo":1, _id:0})
var docu = cursor.next();
var b = cursor.next();
while(cursor.hasNext()){
    //print("-----");
    if(docu.Tema == b.Tema && docu.Nombre==b.Nombre && docu.Apodo != b.Apodo){
        printjson(docu);
    }
    var docu = b;
    var b = cursor.next();
}

```

### APARTADO D.

#### D1) Coger el máximo Precio para cada Tema

```

db.aficiones.aggregate([
    { "$group":
        { "_id": "$Tema",
          "PrecioMax": { "$max": "$Precio" }
        }
    }
])

```

```

/* 1 */
{

```

```

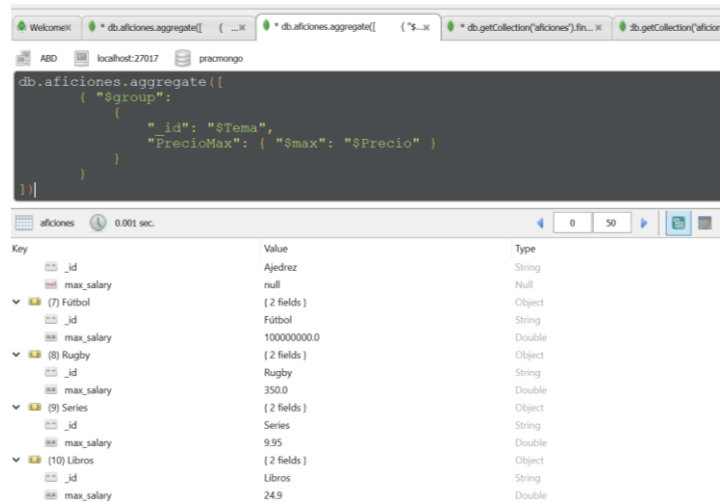
    "_id": "Videojuegos",
    "max_salary": 59.95
  }
/* 7 */
{
  "_id": "Fútbol",
  "max_salary": 100000000.0
}

/* 8 */
{
  "_id": "Rugby",
  "max_salary": 350.0
}

/* 9 */
{
  "_id": "Series",
  "max_salary": 9.95
}

/* 10 */
{
  "_id": "Libros",
  "max_salary": 24.9
}

```

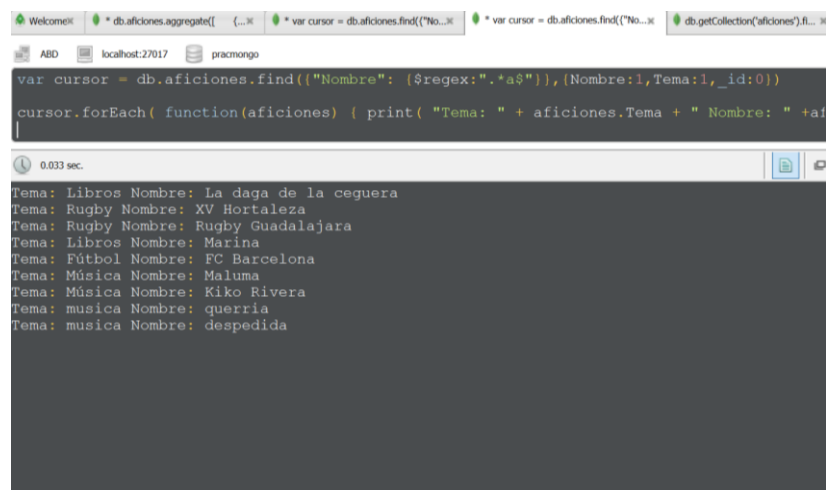


Key	Value	Type
_id	Ajedrez	String
max_salary	null	Null
(7) Fútbol	{ 2 fields }	Object
_id	Fútbol	String
max_salary	100000000.0	Double
(8) Rugby	{ 2 fields }	Object
_id	Rugby	String
max_salary	350.0	Double
(9) Series	{ 2 fields }	Object
_id	Series	String
max_salary	9.95	Double
(10) Libros	{ 2 fields }	Object
_id	Libros	String
max_salary	24.9	Double

## D2. Conseguir el Tema y el nombre de aquellos nombres que terminen en a

```
var cursor = db.aficiones.find({"Nombre": {$regex:".a$"}},{Nombre:1,Tema:1,_id:0})
```

```
cursor.forEach( function(aficiones) { print( "Tema: " + aficiones.Tema + " Nombre: " + aficiones.Nombre ); });
```



```
var cursor = db.aficiones.find({"Nombre": {$regex:".a$"}},{Nombre:1,Tema:1,_id:0})
cursor.forEach( function(aficiones) { print( "Tema: " + aficiones.Tema + " Nombre: " + aficiones.Nombre ); });
```

Tema	Nombre
Libros	La daga de la ceguera
Rugby	XV Hortaleza
Rugby	Rugby Guadalajara
Libros	Marina
Fútbol	FC Barcelona
Música	Maluma
Música	Kiko Rivera
musica	querria
musica	despedida

## D3. Decir por cada tema que si el precio es asequible o no, mayor que 500 no asequible y al revés si.

```

var cursor = db.aficiones.aggregate([
  { "$group":
    { "_id": "$Tema",
      "PrecioTotal": { "$sum": "$Precio" }
    }
  }
])
var a = cursor.next()
while(cursor.hasNext()){
  if(a.PrecioTotal > 500){
    print("Precio no asequible" + a._id.Tema + " con un precioTotal de: " + a.PrecioTotal)
  }
  else{
    print("Precio asequible" + a._id.Tema + " con un precioTotal de: " + a.PrecioTotal)
  }
  var a = cursor.next()
}

```

```

Welcome x * var cursor = db.getCollection("aficiones") x * var cursor = db.aficiones.aggregate([ x * db.getCollection("aficiones").find({$and
ABD localhost:27017 prcmongo
var cursor = db.aficiones.aggregate([
  { "$group":
    { "_id": "$Tema",
      "PrecioTotal": { "$sum": "$Precio" }
    }
  }
])
var a = cursor.next()
while(cursor.hasNext()){
  if(a.PrecioTotal > 500){
    print("Precio no asequible del Tema " + a._id + " con un precioTotal de: " + a.PrecioTotal)
  }
  else{
    print("Precio asequible del Tema " + a._id + " con un precioTotal de: " + a.PrecioTotal)
  }
  var a = cursor.next()
}
robot
0 sec.
Precio asequible del Tema Videojuegos con un precioTotal de: 350.37219500000003
Precio asequible del Tema Música con un precioTotal de: 165.966595
Precio asequible del Tema musica con un precioTotal de: 50
Precio no asequible del Tema Baloncesto con un precioTotal de: 170000000
Precio asequible del Tema MotoGP con un precioTotal de: 0
Precio asequible del Tema Ajedrez con un precioTotal de: 0
Precio no asequible del Tema Fútbol con un precioTotal de: 240006706.561
Precio no asequible del Tema Rugby con un precioTotal de: 2781.6620000000003
Precio asequible del Tema Series con un precioTotal de: 29.849999999999998

```

#### D4. MOSTRAR PARA EL TEMA DE LIBROS CUANTOS LIBROS HAY DE CADA GENERO

```
db.aficiones.aggregate({$match:{"Tema":"Libros"}},{ "$group":{_id:"$genero",Componentes:{$sum:1}}})
```

```

/* 1 */
{
  "_id" : "Fantasia-epica",
  "Componentes" : 3.0
}
/* 2 */
{
  "_id" : "Fantasia",
  "Componentes" : 5.0
}

```

*/\* 3 \*/*

```
{
  "_id" : "Religioso",
  "Componentes" : 1.0
}
```

## APARTADO E.

**E1)**

```
var cursor = db.getCollection('aficiones').find({}, {Tema:1, Nombre:1, _id:0});
```

```
var a = cursor.next();
```

```
while(cursor.hasNext()){
    print("Tema " + a.Tema + " Nombre " + a.Nombre);
    var a = cursor.next();
}
```

The screenshot shows the MongoDB Compass interface. At the top, the database is 'aficiones' and the collection is 'aficiones.aggregate'. The query is: `db.aficiones.aggregate([{$match: {'Puntuacion': {$gte: 9}}}, {$group: {_id: '$Tema', Componentes: {$addToSet: '$Nombre'}}}, NumComponentes: {$sum: 1}}])`. The results are displayed in a table with 8 rows. Each row represents a sport (Tema) and its associated components (Componentes) and the total number of components (NumComponentes).

Key	Value	Type
(1) Rugby	{ 3 fields }	Object
_id	Rugby	String
Componentes	[ 2 elements ]	Array
NumComponentes	2.0	Double
(2) Fútbol	{ 3 fields }	Object
_id	Fútbol	String
Componentes	[ 5 elements ]	Array
NumComponentes	5.0	Double
(3) Videojuegos	{ 3 fields }	Object
_id	Videojuegos	String
Componentes	[ 4 elements ]	Array
NumComponentes	4.0	Double
(4) Series	{ 3 fields }	Object
(5) Libros	{ 3 fields }	Object
(6) Música	{ 3 fields }	Object
(7) musica	{ 3 fields }	Object
(8) Baloncesto	{ 3 fields }	Object

```

Tema Libros Nombre El temor del hombre sabio
Tema Libros Nombre El nombre del viento
Tema Libros Nombre El camino de las sombras
Tema Libros Nombre Mas allá de las sombras
Tema Libros Nombre Al filo de las sombras
Tema Libros Nombre El prisma negro
Tema Libros Nombre La daga de la ceguera
Tema Libros Nombre El ojo fragmentado
Tema Libros Nombre Corazón inquieto: la vida de San Agustín
Tema Fútbol Nombre Cristiano Ronaldo
Tema Fútbol Nombre Isco
Tema Fútbol Nombre Sergio Ramos
Tema MotoGP Nombre undefined
Tema MotoGP Nombre undefined
Tema MotoGP Nombre undefined
Tema MotoGP Nombre undefined
Tema MotoGP Nombre undefined
Tema MotoGP Nombre undefined
Tema MotoGP Nombre undefined
Tema MotoGP Nombre undefined
Tema MotoGP Nombre undefined
Tema MotoGP Nombre undefined
Tema MotoGP Nombre undefined
Tema Rugby Nombre Torrejon R.C.
Tema Rugby Nombre San Isidro R.C
Tema Rugby Nombre CAU Metropolitano
Tema Rugby Nombre Rivas R.C
Tema Rugby Nombre Alcala R.C
Tema Rugby Nombre XV Hortaleza
Tema Rugby Nombre Rugby Guadalajara
Tema Rugby Nombre Quijote Rugby
Tema Rugby Nombre Club de Rugby Ingenieros Industriales Las Rozas

```

**E2)**

```
db.aficiones.aggregate({$match:{"Puntuacion":{$gte:9}}, {"$group":{"_id":"$Tema",
Componentes:{$addToSet:"$Nombre"}, NumComponentes:{$sum:1}}})
```

Ejemplo en texto:

```
/* 1 */
```

```
{
  "_id": "Rugby",
  "Componentes": [
    "San Isidro R.C",
```

```

    "Torrejon R.C."
  ],
  "NumComponentes" : 2.0
}

```

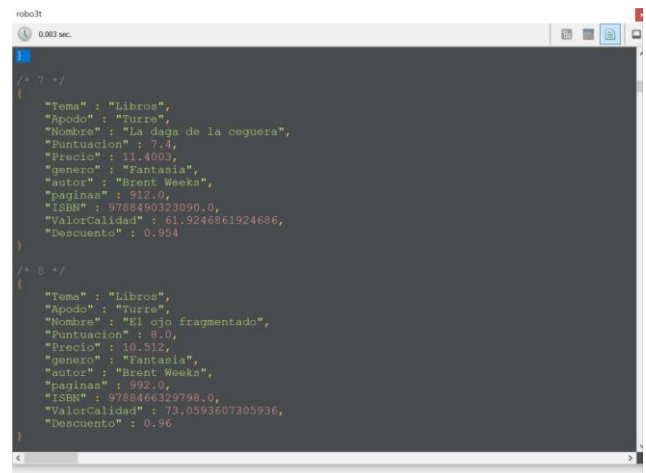
## APARTADO F.

En este apartado tenemos que añadir descuento según su función, la manera de plantearlo ha sido la siguiente: si la puntuación es menor igual que 7 entonces pondremos un descuento de 0.7 ( $\text{aficiones.Puntuacion} - 7 \leq 0$ ) es decir será el precio por 0.3, si tenemos una puntuación de 7 y un precio de 25€  $\rightarrow 25 \times 0.9 = 22.5$  es decir se ha realizado un descuento del 10%, si es mayor ( $\text{aficiones.Puntuacion} - 7 > 0$ ) entonces restaremos la puntuación  $- 7$  y eso lo multiplicaremos  $\times 0.01$  es decir si la puntuación es de 8.2 nos daría 0.012 y eso se lo sumamos a 0.95, cuanto más alto sea la puntuación más alto será el porcentaje y por lo tanto menor el descuento.

```

var cursor = db.aficiones.find();
cursor.forEach(function(aficiones){
  if(aficiones.Puntuacion - 7 > 0){
    var porcentaje =(0.95 + (parseFloat(aficiones.Puntuacion) - 7)*0.01);
  }
  else{
    var porcentaje = 0.9;
  }
  aficiones.Descuento = porcentaje;
  if(aficiones.Precio >= 0){
    aficiones.Precio = aficiones.Precio*porcentaje;
  }
  else {
    aficiones.Precio = 0;
  }
  print(aficiones);
  db.aficiones.save(aficiones);
})
/* 1 */
{
  "Tema" : "Libros",
  "Apodo" : "Turre",
  "Nombre" : "El temor del hombre sabio",
  "Puntuacion" : 8.2,
  "Precio" : 23.9538,
  "genero" : "Fantasia",
  "autor" : "Patrick Rothfuss",
  "paginas" : 1200.0,
  "ISBN" : 9788401339639.0,

```



```

/* 1 */
{
  "Tema" : "Libros",
  "Apodo" : "Turre",
  "Nombre" : "La daga de la ceguera",
  "Puntuacion" : 7.4,
  "Precio" : 11.4003,
  "genero" : "Fantasia",
  "autor" : "Brent Weeks",
  "paginas" : 912.0,
  "ISBN" : 9788440323090.0,
  "ValorCalidad" : 61.9246861924686,
  "Descuento" : 0.954
}

/* 2 */
{
  "Tema" : "Libros",
  "Apodo" : "Turre",
  "Nombre" : "El ojo fragmentado",
  "Puntuacion" : 8.0,
  "Precio" : 10.512,
  "genero" : "Fantasia",
  "autor" : "Brent Weeks",
  "paginas" : 922.0,
  "ISBN" : 9788440329798.0,
  "ValorCalidad" : 73.0593607305936,
  "Descuento" : 0.96
}

```

```
"ValorCalidad" : 32.9317269076305,
"Descuento" : 0.962
}
/* 2 */
{
  "Tema" : "Libros",
  "Apodo" : "Turre",
  "Nombre" : "El nombre del viento",
  "Puntuacion" : 7.9,
  "Precio" : 19.8513,
  "genero" : "Fantasia",
  "autor" : "Patrick Rothfuss",
  "paginas" : 880.0,
  "ISBN" : 9788401337208.0,
  "ValorCalidad" : 38.1642512077295,
  "Descuento" : 0.959
}
/* 3 */
{
  "Tema" : "Libros",
  "Apodo" : "Turre",
  "Nombre" : "El camino de las sombras",
  "Puntuacion" : 8.3,
  "Precio" : 9.58185,
  "genero" : "Fantasia-epica",
  "autor" : "Brent Weeks",
  "paginas" : 592.0,
  "ISBN" : 9788499893679.0,
  "ValorCalidad" : 83.4170854271357,
  "Descuento" : 0.963
}
/* 4 */
{
  "Tema" : "Libros",
  "Apodo" : "Turre",
  "Nombre" : "Mas allá de las sombras",
  "Puntuacion" : 8.3,
  "Precio" : 10.54485,
  "genero" : "Fantasia-epica",
  "autor" : "Brent Weeks",
  "paginas" : 624.0,
  "ISBN" : 9788499894348.0,
  "ValorCalidad" : 75.7990867579909,
  "Descuento" : 0.963
}
/* 5 */
{
```



```

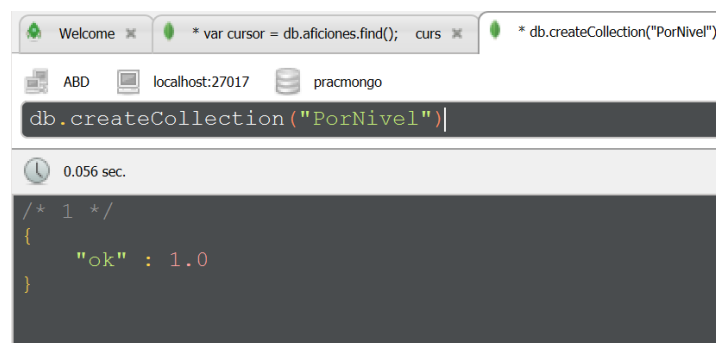
    "Tema" : "Libros",
    "Apodo" : "Turre",
    "Nombre" : "Al filo de las sombras",
    "Puntuacion" : 8.0,
    "Precio" : 19.872,
    "genero" : "Fantasia-epica",
    "autor" : "Brent Weeks",
    "paginas" : 576.0,
    "ISBN" : 9788499893983.0,
    "ValorCalidad" : 38.6473429951691,
    "Descuento" : 0.96
  }

```

## APARTADO G.

### G.1)

```
db.createCollection("PorNivel")
```



### G.2)

```

var cursor = db.aficiones.find({});
var array1= [];
var array2 = [];
var array3= [];
var array4= [];
var a = cursor.next();
while(cursor.hasNext()){
    var punt = a.Puntuacion*10;
    if(punt <=30){
        array1.push(a);
    }
    else if(punt <= 50){
        array2.push(a);
    }
    else if(punt <= 70){
        array3.push(a);
    }
    else{
        array4.push(a);
    }
}

```

```

    }
    a = cursor.next();
}
db.PorNivel.insert({"NomCal": "nivel_1", "Componentes": array1})
db.PorNivel.insert({"NomCal": "nivel_2", "Componentes": array2})
db.PorNivel.insert({"NomCal": "nivel_3", "Componentes": array3})
db.PorNivel.insert({"NomCal": "nivel_4", "Componentes": array4})

```

Key	Value	Type
(3) ObjectId("5ec8150a282a26db62f61fd1")	{ 3 fields }	Object
_id	ObjectId("5ec8150a282a26db62f61fd1")	ObjectId
NomCal	nivel_3	String
Componentes	[ 12 elements ]	Array
[0]	{ 11 fields }	Object
[1]	{ 11 fields }	Object
[2]	{ 11 fields }	Object
[3]	{ 10 fields }	Object
[4]	{ 10 fields }	Object
[5]	{ 10 fields }	Object
[6]	{ 10 fields }	Object
[7]	{ 10 fields }	Object
[8]	{ 10 fields }	Object
[9]	{ 10 fields }	Object
[10]	{ 11 fields }	Object
[11]	{ 10 fields }	Object
(4) ObjectId("5ec8150a282a26db62f61fd2")	{ 3 fields }	Object
_id	ObjectId("5ec8150a282a26db62f61fd2")	ObjectId
NomCal	nivel_4	String
Componentes	[ 76 elements ]	Array

### G.3)

```

var cursor = db.getCollection('PorNivel').find();
var docu = cursor.next();
while(cursor.hasNext()){
    print("-----");
    printjson(docu);
    var docu = cursor.next();
}

```

```

{
  "Puntuacion" : 5,
  "Precio" : 5.8983390000000002,
  "Genero" : "Pop",
  "Discografica" : "Red Hill",
  "ValorCalidad" : 55.61735261401557,
  "Descuento" : 5
},
{
  "_id" : ObjectId("5746cfce6046b47d407366b"),
  "Tema" : "Música",
  "Apodo" : "dios",
  "Nombre" : "Drake",
  "Puntuacion" : 4,
  "Precio" : 9.834939,
  "Genero" : "Hip hop",
  "Discografica" : "Universal Motowns",
  "ValorCalidad" : 26.6844563042028,
  "Descuento" : 6
}
-----
{
  "_id" : ObjectId("5ec8150a282a26db62f61fd1"),
  "NomCal" : "nivel_3",
  "Componentes" : [
    {
      "_id" : ObjectId("5746cf7de6046b47d407362f"),
      "Tema" : "Rugby",
      "Apodo" : "Infor",
      "Nombre" : "Rivas R.C.",
      "Puntuacion" : 7,
      "Precio" : 300,

```

**G.4)**

```
db.PorNivel.aggregate( [  
  { $unwind: "$Componentes" },  
  { $sort: {"Componentes.Precio":1}},  
  { $project:{"NomCal":1,"Componentes.Nombre":1,"Componentes.Precio":1,_id:0}},  
  { $limit:5}]);
```

Resultado, nos salen los que no tienen precio obviamente porque serán los más baratos.

```
/* 1 */  
{  
  "NomCal" : "nivel_4",  
  "Componentes" : {  
    "Precio" : 0.0  
  }  
}  
/* 2 */  
{  
  "NomCal" : "nivel_3",  
  "Componentes" : {  
    "Nombre" : "World of Warships",  
    "Precio" : 0.0  
  }  
}  
/* 3 */  
{  
  "NomCal" : "nivel_4",  
  "Componentes" : {  
    "Precio" : 0.0  
  }  
}  
/* 4 */  
{  
  "NomCal" : "nivel_4",  
  "Componentes" : {  
    "Precio" : 0.0  
  }  
}  
/* 5 */  
{  
  "NomCal" : "nivel_1",  
  "Componentes" : {  
    "Nombre" : "Cuentame",  
    "Precio" : 0.0  
  }  
}
```

**G.5)**

```

db.PorNivel.aggregate( [
  { $unwind: "$Componentes" },
  { $sort: {"Componentes.Precio":-1}},
  { $group : { _id:"$NomCal", Componentes:{$push:"$Componentes"}}}
]).forEach(function(aficiones){
  aficiones.Componentes.shift();
  aficiones.Componentes.shift();
  print(aficiones);
  db.PorNivel.update( { NomCal : aficiones.NomCal }, aficiones );
});

```

Key	Value	Type
(1) ObjectId("5ec823e1282a26db62f61fd3")	{ 3 fields }	Object
_id	ObjectId("5ec823e1282a26db62f61fd3")	ObjectId
NomCal	nivel_1	String
Componentes	[ 5 elements ]	Array
(2) ObjectId("5ec823e1282a26db62f61fd4")	{ 3 fields }	Object
_id	ObjectId("5ec823e1282a26db62f61fd4")	ObjectId
NomCal	nivel_2	String
Componentes	[ 5 elements ]	Array
(3) ObjectId("5ec823e1282a26db62f61fd5")	{ 3 fields }	Object
_id	ObjectId("5ec823e1282a26db62f61fd5")	ObjectId
NomCal	nivel_3	String
Componentes	[ 12 elements ]	Array
(4) ObjectId("5ec823e1282a26db62f61fd6")	{ 3 fields }	Object
_id	ObjectId("5ec823e1282a26db62f61fd6")	ObjectId
NomCal	nivel_4	String
Componentes	[ 76 elements ]	Array

Key	Value	Type
(1)	{ 1 field }	Object
Componentes	[ 74 elements ]	Array
(2)	{ 1 field }	Object
Componentes	[ 3 elements ]	Array
(3)	{ 1 field }	Object
Componentes	[ 10 elements ]	Array
(4)	{ 1 field }	Object
Componentes	[ 3 elements ]	Array

Vemos como efectivamente a disminuido el número de cada nivel en 2.

## APARTADO 4.- USANDO COLECCIONES LIMITADAS

a)

```
db.createCollection("superGuai")
```

b)

```
db.PorNivel.aggregate( { $unwind: "$Componentes" }, { $project: { _id:0 , Componente:
"$Componentes" } }, { $limit: 5 } ).forEach(function(PorNivel) {
    db.superGuai.insert(PorNivel);
})
```

The screenshot shows the Robo 3T interface with the following content:

```
db.PorNivel.aggregate(
{ $unwind: "$Componentes" },
{ $project: { _id:0 , Componente: "$Componentes" } },
{ $limit: 5 }
).forEach(function(PorNivel) {
    db.superGuai.insert(PorNivel);
})
```

Below the code, the execution time is shown as 0.104 sec. The output log displays the following messages:

```
Inserted 1 record(s) in 3ms
Inserted 1 record(s) in 3ms
Inserted 1 record(s) in 5ms
Inserted 1 record(s) in 1ms
Inserted 1 record(s) in 1ms
Inserted 1 record(s) in 1ms
Inserted 1 record(s) in 1ms
Inserted 1 record(s) in 2ms
Inserted 1 record(s) in 1ms
Inserted 1 record(s) in 0ms
Inserted 1 record(s) in 0ms
Inserted 1 record(s) in 1ms
Inserted 1 record(s) in 3ms
Inserted 1 record(s) in 1ms
```

Comprobamos que se ha insertado correctamente:

The screenshot shows the Robo 3T interface with the following content:

The left sidebar shows the database structure:

- ABD (4)
  - System
  - config
  - pracmongo
    - Collections (3)
      - PorNivel
      - aficiones
      - superGuai
    - Functions
    - Users

The main window shows the execution of the query:

```
db.getCollection('superGuai').find({})
```

The execution time is 0.002 sec. The results are displayed in a table with columns Key, Value, and Type.

Key	Value	Type
(1) ObjectId("Sec9561939a3868ae60549b0")	{ 2 fields }	Object
_id	ObjectId("Sec9561939a3868ae60549b0")	ObjectId
Componente	{ 11 fields }	Object
_id	ObjectId("5746c77de6046b47d407362e")	ObjectId
Tema	Rugby	String
Apodo	Infor	String
Nombre	CAU Metropolitano	String
Puntuacion	3.0	Double
Precio	206.6715	Double
direccion	[ 1 element ]	Array
Fundacion	1962.0	Double
Equipos	[ 1 element ]	Array
ValorCalidad	0.857142857142857	Double
Descuento	0.9	Double
(2) ObjectId("Sec9561939a3868ae60549b1")	{ 2 fields }	Object
(3) ObjectId("Sec9561939a3868ae60549b2")	{ 2 fields }	Object
(4) ObjectId("Sec9561939a3868ae60549b3")	{ 2 fields }	Object
(5) ObjectId("Sec9561939a3868ae60549b4")	{ 2 fields }	Object
(6) ObjectId("Sec9561939a3868ae60549b5")	{ 2 fields }	Object
(7) ObjectId("Sec9561939a3868ae60549b6")	{ 2 fields }	Object

3)

```
db.superGuai.insert({Nombre:"nombre1", NumComponentes:1})
```

```
db.superGuai.insert({Nombre:"nombre2", NumComponentes:2})
```

```
db.superGuai.insert({Nombre:"nombre3", NumComponentes:3})
```

ABD localhost:27017 pracmongo

```
db.superGuai.insert({Nombre:"nombre1", NumComponentes:1})
db.superGuai.insert({Nombre:"nombre2", NumComponentes:2})
db.superGuai.insert({Nombre:"nombre3", NumComponentes:3})
```

db.superGuai.. db.superGuai.. db.superGuai..

0.001 sec.

Inserted 1 record(s) in 2ms

ABD localhost:27017 pracmongo

```
db.getCollection('superGuai').find({"Nombre":"nombre1"})
```

superGuai 0.001 sec.

Key	Value	Type
(1) ObjectId("5ec9575139a3868ae6054a7a")	{ 3 fields }	Object
_id	ObjectId("5ec9575139a3868ae6054a7a")	ObjectId
Nombre	nombre1	String
NumComponentes	1.0	Double

ABD localhost:27017 pracmongo

```
db.getCollection('superGuai').find({"Nombre":"nombre2"})
```

superGuai 0.002 sec.

Key	Value	Type
(1) ObjectId("5ec9575139a3868ae6054a7b")	{ 3 fields }	Object
_id	ObjectId("5ec9575139a3868ae6054a7b")	ObjectId
Nombre	nombre2	String
NumComponentes	2.0	Double

ABD localhost:27017 pracmongo

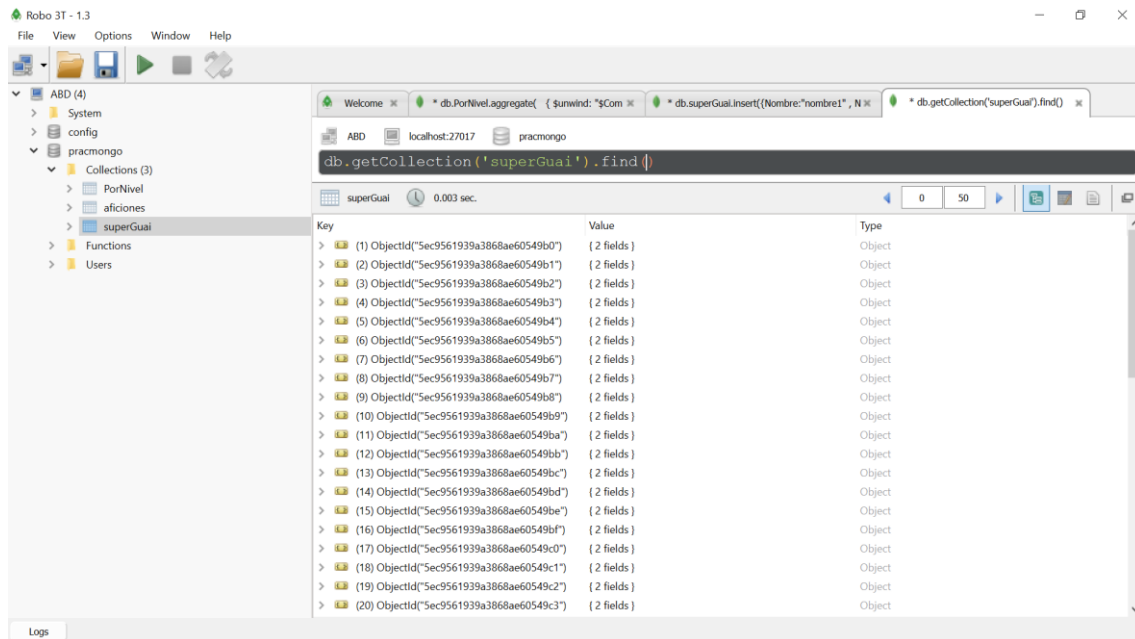
```
db.getCollection('superGuai').find({"Nombre":"nombre3"})
```

superGuai 0.001 sec.

Key	Value
(1) ObjectId("5ec9575139a3868ae6054a7c")	{ 3 fields }
_id	ObjectId("5ec9575139a3868ae6054a7c")
Nombre	nombre3
NumComponentes	3.0

5)

```
db.getCollection('superGuai').find()
```



## APARTADO 5

```
db.createCollection("Pueblo")
```

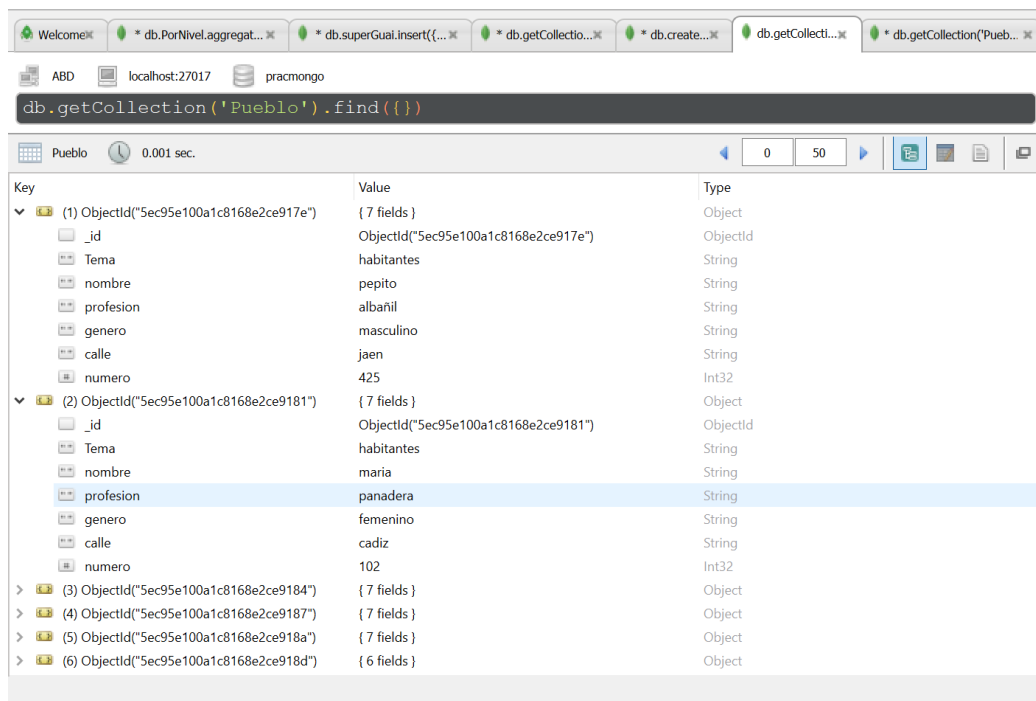
Añadimos dos tipos de temas a nuestra colección la primera será habitantes del pueblo en el cual guardaremos el nombre la profesión el genero la calle donde vive y el numero de la casa, del mismo modo crearemos un segundo tema que se llamará edificios donde almacenaremos el nombre, la calle, el número, el año de creación y el estilo que tiene

```
{
  "Tema" : "habitantes",
  "nombre" : "pepito",
  "profesion" : "albañil",
  "genero" : "masculino",
  "calle" : "jaen",
  "numero" : 425
}
{
  "Tema" : "habitantes",
  "nombre" : "maria",
  "profesion" : "panadera",
  "genero" : "femenino",
  "calle" : "cadiz",
  "numero" : 102
}
{
  "Tema" : "habitantes",
  "nombre" : "pascual",
```

```

"profesion" : "pastor",
"genero" : "masculino",
"calles" : "reina",
"numero" : 234
}
{
"Tema" : "edificios",
"nombre" : "iglesia",
"calles" : "real",
"numero" : 11,
"año": 1815,
"estilo": "romanico"
}
{
"Tema" : "edificios",
"nombre" : "ayuntamiento",
"calles" : "juan maria",
"numero" : 7,
"año": 1953,
"estilo": "modernista"
}
{
"Tema" : "edificios",
"nombre" : "escuela",
"calles" : "las heras",
"numero" : 4,
"año": 1931
}

```



Key	Value	Type
(1) ObjectId("5ec95e100a1c8168e2ce917e")	{ 7 fields }	Object
_id	ObjectId("5ec95e100a1c8168e2ce917e")	ObjectId
Tema	habitantes	String
nombre	pepito	String
profesion	albañil	String
genero	masculino	String
calles	jaen	String
numero	425	Int32
(2) ObjectId("5ec95e100a1c8168e2ce9181")	{ 7 fields }	Object
_id	ObjectId("5ec95e100a1c8168e2ce9181")	ObjectId
Tema	habitantes	String
nombre	maria	String
profesion	panadera	String
genero	femenino	String
calles	cadiz	String
numero	102	Int32
(3) ObjectId("5ec95e100a1c8168e2ce9184")	{ 7 fields }	Object
(4) ObjectId("5ec95e100a1c8168e2ce9187")	{ 7 fields }	Object
(5) ObjectId("5ec95e100a1c8168e2ce918a")	{ 7 fields }	Object
(6) ObjectId("5ec95e100a1c8168e2ce918d")	{ 6 fields }	Object



**APARTADO 5- EXTRA**

Se podría normalizar y desnormalizar para aumentar su rendimiento y así disminuir el tiempo de ejecución de consultas haciendo que los datos sean más accesibles.

```
{
  "Tema" : "fútbol",
  "nombre" : "cristiano ronaldo",
  "genero" : "masculino",
  "domicilio": "Portugal",
  "edad": 36
  "estado" : "casado",
  "hijos": 1,
  "equipos":{
    "Madrid",
    "Juventus"
  }
}
{
  "Tema" : "fútbol",
  "nombre" : "messi",
  "genero" : "masculino",
  "domicilio": "barcelona",
  "edad": 34
  "estado" : "casado",
  "hijos": 1,
  "equipos":{
    "Barça"
  }
}
```