

Nombre y apellidos:

Nota: Las preguntas con respuesta múltiple pueden tener más de una respuesta

Indicar qué mostrará la siguiente instrucción:

```
db.coord.find().skip(4).sort({x:1, y:1})
```

- ☐ No mostrará nada, da error porque faltan las llaves dentro de find
- ☐ Se salta los 4 primeros elementos de la colección coord en el orden de inserción y ordena el resto, primero por x, y luego por y para los que tengan la misma x.
- ☒ Ordena los elementos de *coord*, primero por x y luego por y los que tengan la misma x, y muestra los 5 primeros.
- ☐ Ordena según el valor de x+y, y muestra los 5 primeros.

Queremos obtener las personas con edad mayor de 50 años. De cada persona queremos todo menos su mail, ¿qué debemos escribir?:

- ☐ db.personas.find({edad:{\$gt:50}, mail:0})
- ☒ db.personas.find ({edad:{\$gt:50}}, {mail:0})
- ☐ db.personas.find ({edad:{\$gt:50}}, {edad:1, mail:0})
- ☐ db.personas.find ({edad:{\$gt:50}}).skip({mail})

Queremos obtener las personas con edad mayor de 50 años y menor de 60. ¿qué debemos escribir?:

- ☐ db.personas.find({edad:{\$gt:50 && \$lt:60}})
- ☒ db.personas.find({edad:{\$gt:50, \$lt:60}})
- ☐ db.personas.find({edad:{\$gt:50}, edad:{\$lt:60}})
- ☐ db.personas.find({edad:{\$range:[50,60]}})

El array *a* de la colección *datos* tiene números enteros. Queremos devolver todos los documentos en los que *a* contenga un valor 10. ¿Qué es lo más correcto?

- ☐ db.datos.find({a: [10]})
- ☐ db.datos.find({\$elemMatch:{a: {\$eq:10}}})
- ☒ db.datos.find({a:10})
- ☐ db.datos.find({\$in:{a,[10]}})

Si miramos un fichero en MongoDB que contiene una colección. ¿Qué formato veremos?

- ☐ XML
- ☐ OSON
- ☐ JSON
- ☒ BSON
- ☐ OTROSON

Indicar qué se mostrará por pantalla tras esta secuencia de instrucciones:

```
db.aficiones.drop()
db.aficiones.insert({ _id : "Berto", aficiones : [ "damas", "botanica" ] })
db.aficiones.update( { _id : "Berto" }, { $push : { aficiones : "buceo" } } );
db.aficiones.update( { _id : "Berto" }, { $pop : { aficiones : -1 } } );
db.aficiones.update( { _id : "Berto" }, { $addToSet : { aficiones : "montaña" } } );
db.aficiones.update( { _id : "Berto" }, { $pushAll: { aficiones : [ "leer", "damas" ] } } );
db.aficiones.find({_id:"Berto"})
```

Respuesta: { "_id" : "Berto", "aficiones" : ["botanica", "buceo", "montaña", "leer", "damas"] }

¿Qué mostrarán estas instrucciones en la consola de mongo?

```
x = { "a" : 10 };
y = "a";
x[y]++;
print(x.a);
```

Solucion: 11

¿Cuáles de los siguientes documentos podrían mostrarse como respuesta a `db.products.find({ tags : "shiny" })`?

- ☒ { _id : 42, name : "Whizzy Wiz-o-matic", tags:["awesome", "shiny", "green"] }
- ☐ { _id : 704 , name : "Foey Foo-o-tron", tags: ["blue", "mediocre"] }
- ☒ { _id : 1040 , name : "Snappy Snap-o-lux", tags : "shiny" }
- ☐ { _id : 12345 , name : "Quuxinator", tags : [] }

¿Cuáles de los siguientes documentos podrían mostrarse como respuesta a `db.users.find({ friends : { $all : ["Joe", "Bob"] }, favorites : { $in : ["running", "pickles"] } })` ?

- ☐ { name : "William", friends : ["Bob", "Fred"], favorites : ["hamburgers", "running"] }
- ☐ { name : "Stephen", friends : ["Joe", "Pete"], favorites : ["pickles", "swimming"] }
- ☒ { name : "Cliff", friends : ["Pete", "Joe", "Tom", "Bob"], favorites : ["pickles", "cycling"] }
- ☐ { name : "Harry", friends : ["Joe", "Bob"], favorites : ["hot dogs", "swimming"] }

Tenemos una colección foo vacía y escribimos la siguiente orden:

```
db.foo.update( { username : 'bar' }, { '$set' : { 'interests': [ 'cat', 'dog' ] } }, { upsert : true } );
```

¿Cuáles de los siguientes podrían ser documentos en la colección?

- ☐ { "_id" : ObjectId("507b78232e8dfde94c149949"), "interests" : ["cat", "dog"] }
- ☐ { "interests" : ["cat", "dog"], "username" : "bar" }
- ☐ { }
- ☒ { "_id" : ObjectId("507b78232e8dfde94c149949"), "interests" : ["cat", "dog"], "username" : "bar" }