

Caso Práctico Guiado en MongoDB

CONTENIDO

Descripción del caso	1
Creación de la base de datos	3
Inserción de documentos en colecciones	4
Consulta de documentos	6
Creación de índices	8
Modificación de documentos	9
Eliminar documentos	10
Eliminar colección	10
Eliminar la base de datos	10

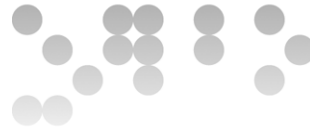
1. DESCRIPCIÓN DEL CASO

Una pequeña empresa de emprendedores ha concebido una aplicación para dispositivos móviles que permita la compra-venta de cualquier tipo de artículo entre particulares.

Para que un usuario pueda hacer uso de la aplicación debe registrarse con un nombre de usuario y una contraseña y aportar una dirección de correo electrónico y la dirección de un punto de venta preferente donde entregar los artículos.

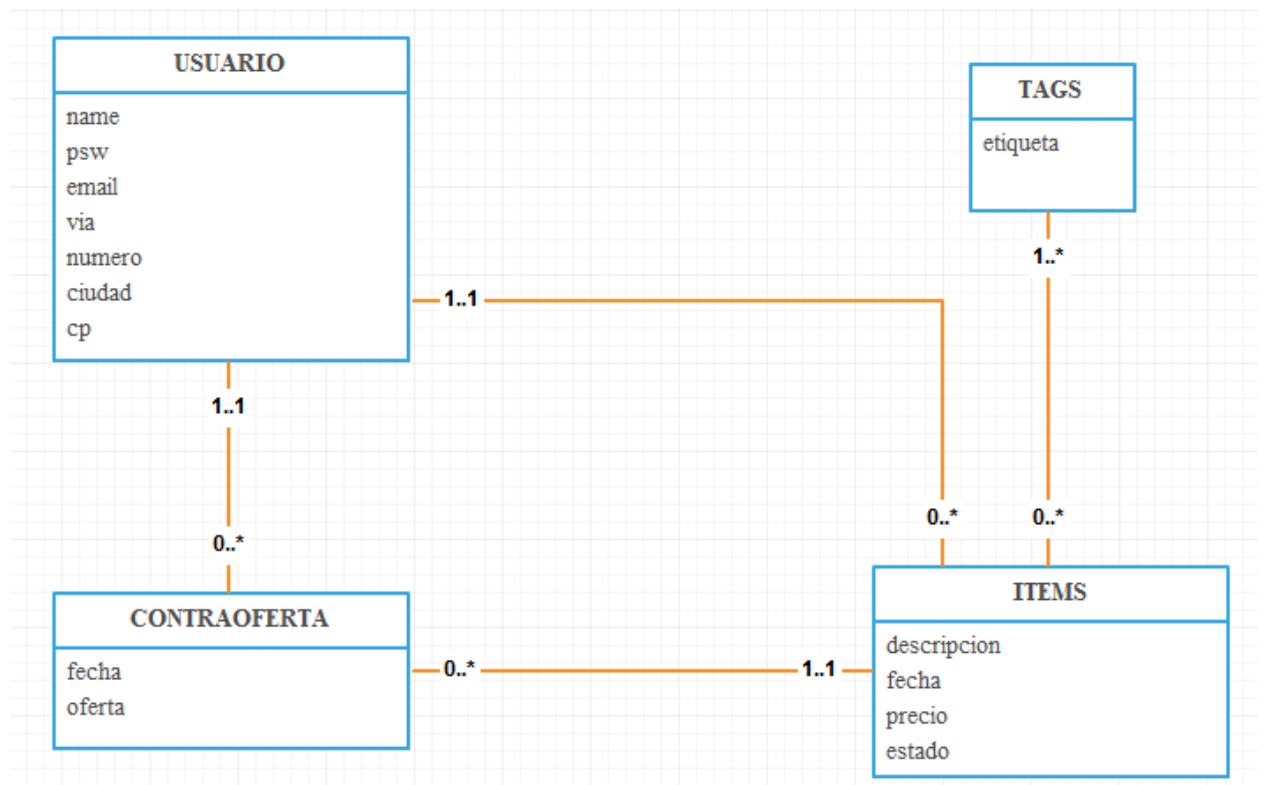
Para poner un ítem a la venta el vendedor debe indicar una breve descripción del artículo, el precio de salida y la lista de etiquetas que lo describen. La aplicación incluirá también la fecha en la que el artículo sale a la venta.

A fin de que los usuarios de la aplicación puedan buscar artículos que se vendan cerca de su zona, la aplicación también guarda la longitud y latitud del punto de venta introducido por el vendedor del objeto. La dirección definitiva de entrega puede variar, ya que comprador y vendedor se pueden poner de acuerdo mediante correo electrónico para acordar el lugar que mejor les venga a ambos.

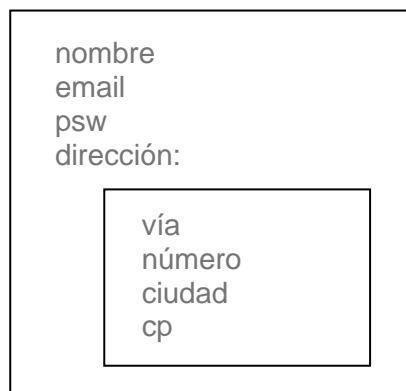


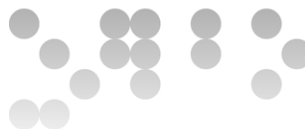
Los compradores interesados en un determinado artículo pueden hacer contraofertas, indicando su precio. La aplicación debe también guardar de cada contraoferta cuándo se realizó y quién la realizó.

A efectos de visualización de resultados de las búsquedas, es importante reflejar el estado de un artículo: disponible, vendido o descatalogado. Pasan a este último estado aquellos artículos que hayan sido puestos a la venta hace más de 12 meses y no hayan sido vendidos).

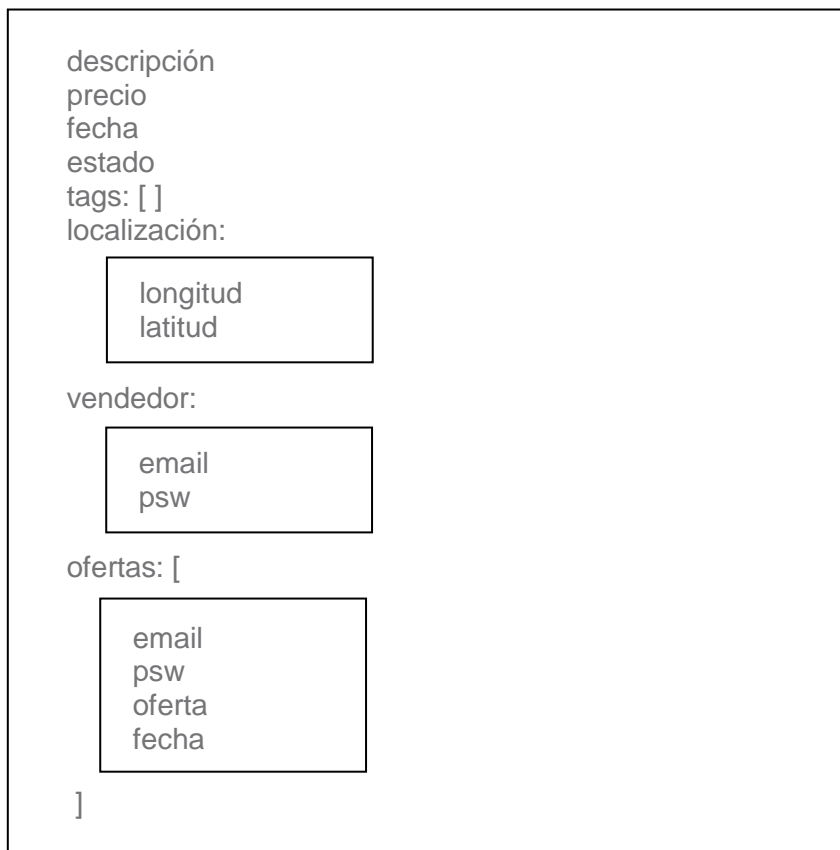


La estructura del agregado *usuario* quedaría:





El agregado *ítem* tendrá la siguiente estructura:



2. CREACIÓN DE LA BASE DE DATOS

Arrancar el servicio de MongoDB

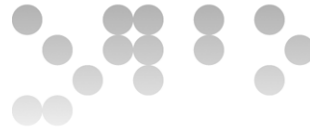
```
sudo service mongod start
```

Comprobar la correcta instalación

Este paso no es necesario si utilizáis la máquina virtual que podéis descargaros del aula.

```
cat /var/log/mongodb/mongod.log
```

```
[initandlisten] waiting for connections on port 27017
```



Parar el servicio MongoDB

```
sudo service mongod stop
```

Conexión a MongoDB a través del shell

```
mongo
```

Visualizar las bases de datos existentes

```
show dbs
```

Consultar la base de datos actual

```
db
```

Crear una base de datos

Las bases de datos se crean en el momento que se dota de contenido a alguna de sus colecciones.

Para indicar la base de datos sobre la que vamos a trabajar:

```
use ventasParticulares
```

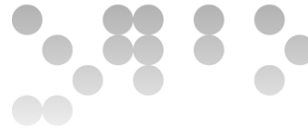
3. INSERCIÓN DE DOCUMENTOS EN COLECCIONES

Existen dos formas de añadir documentos a una colección:

1. Crear una variable con el documento y posteriormente añadirla a la colección con la función *insert*.
2. Insertar directamente el documento en la función *insert*.

Veamos cada una de ellas.

```
item1={
  descripcion: "Mando xBox negro",
  fecha: new Date ("2015, 3, 21"),
  precio: 10,
  tags: ["consolas", "xbox", "entretenimiento"],
  vendedor: {email: "pperez@gmail.com", psw: "pperez"},
  localizacion: {longitude: 37.743671, latitude: -2.552276},
  estado: "disponible",
  contraofertas: [
    {email: "llopez@gmail.com", psw: "llopez", oferta: 8,
      fecha: new Date("2015, 4, 2")},
  ]
}
```



```

        {email: "ggomez@gmail.com", psw: "ggomez", oferta: 7, fecha: new
Date("2015, 4, 13")}
    ]
}

db.items.insert(item1)

db.items.insert({
    descripcion: "Mando Wii Mario",
    fecha: new Date ("2013, 10, 2"),
    precio: 8,
    tags: ["consolas", "wii", "entretenimiento"],
    vendedor: {email: "ffernandez@gmail.com", psw:
"ffernandez"},
    localizacion: {longitude: 38.743671, latitude: -10.552276},
    estado: "vendido",
    comprador: {email: "llopez@gmail.com", psw: "llopez"},
    contraofertas: [
        {email: "llopez@gmail.com", psw: "llopez", oferta: 7, fecha: new
Date("2013, 10, 20")},
        {email: "aalonso@gmail.com", psw: "ggomez", oferta: 5, fecha: new
Date("2013, 10, 19")}
    ]
})

db.items.insert({
    descripcion: "Thermomix",
    fecha: new Date ("2015, 5, 2"),
    precio: 80,
    tags: ["robot cocina", "menaje"],
    vendedor: {email: "ffernandez@gmail.com", psw: "ffernandez"},
    localizacion: {longitude: 38.743671, latitude: -10.552276},
    estado: "disponible"
})

```

Se pueden también insertar documentos de forma masiva insertando un array de documentos.

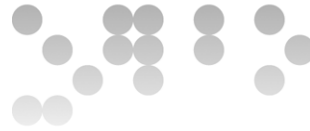
```

usr1={nombre:"Luis López", email: "llopez@gmail.com", psw:"llopez",
direccion: {via:"C/Pez", num:3, ciudad:"Madrid", cp: "28031"}},

usr2={nombre:"Francisco Fernández", email: "ffernandez@gmail.com",
    psw:"ffernandez", direccion: {via:"C/Luna Nueva", num:145,
ciudad: "Barcelona", cp: "08009"}},

usr3={nombre:"Gema Gomez", email: "ggomez@gmail.com", psw:"ggomez",
direccion: {via:"C/Sansa", num:28, ciudad:"Valencia", cp: "46015"}},

```



```
usr4={nombre:"Pepe Perez", email: "pperez@gmail.com", psw:"pperez",
direccion: {via:"C/Sansa", num:79, ciudad:"Valencia", cp: "46015"}},

usr5={nombre:"Ana Alonso", email: "aalonso@gmail.com", psw:"aalonso",
direccion: {via:"C/Luna Llena", num:32, ciudad:"Barcelona", cp:
"08009"}}

db.usuarios.insert([usr1, usr2, usr3, usr4, usr5])
```

4. CONSULTA DE DOCUMENTOS

Mostrar todos los documentos de una colección:

```
db.items.find()
```

En la proyección de los documentos resultantes de la búsqueda se muestra siempre y por defecto el campo que contiene la clave (`_id`). Para no visualizar dicho campo debemos especificarlo explícitamente poniéndolo a 0 en la proyección. Podemos indicar los campos concretos que queremos que aparezcan indicando su nombre en la proyección seguido de :1.

```
db.items.find({descripcion: "Thermomix"},{_id:0, precio:1})
```

Busqueda en arrays

Mostrar la descripción de todos los artículos que estén etiquetados como “consolas”:

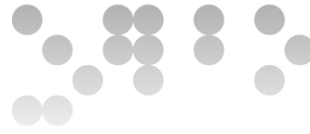
```
db.items.find(
{tags: "consolas"},
{_id:0, descripcion:1})
```

Mostrar la descripción y precio de los artículos que estén etiquetados como “consolas” y “wii”:

```
db.items.find(
{tags: "consolas", tags: "wii"},
{_id:0, descripcion:1, precio:1})
```

Mostrar la descripción y precio de los artículos que estén etiquetados como “consolas” o “wii”:

```
db.items.find(
{tags: {$in:["consolas", "wii"]}},
{_id:0, descripcion:1, precio:1})
```



Busqueda en subagregados

Mostrar la descripción de todos los elementos etiquetados como “consolas” que tengan contraofertas realizadas con posterioridad al 1 de enero de 2014:

```
var start = new Date(2014, 1, 1)

db.items.find(
  {tags: "consolas",
   "contraofertas.fecha": {$gte: start}},
  {descripcion:1})
```

Ordenación de resultados

Mostrar todos los documentos de la colección ítems ordenados de manera que se muestren primero los que tienen contraofertas más recientes:

```
db.items.find().sort( { "contraofertas.fecha": -1 } )
```

Mostrar los usuarios en orden alfabético por su ciudad de residencia:

```
db.usuarios.find().sort( { "direccion.ciudad": 1 } )
```

Agregación

Existe un framework de agregación que permite resumir los resultados de una búsqueda.

Mostrar los emails de los usuarios que tienen algún artículo a la venta (sin mostrar repetidos):

```
db.items.distinct("vendedor.email")
```

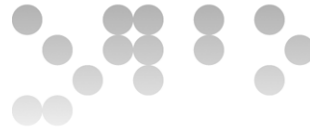
Mostrar cuántos usuarios viven en Valencia:

```
db.usuarios.count({"direccion.ciudad": "Valencia"})
```

(equivalente a `db.usuarios.find({"direccion.ciudad": "Valencia"}).count()`)

Mostrar los artículos que tiene o tuvo a la venta cada usuario:

```
db.items.aggregate([
  {$group: {_id: "$vendedor.email",
            articulos: {$push: "$descripcion"}}}] )
```



5. CREACIÓN DE ÍNDICES

Creación de un índice sobre el campo descripción:

```
db.items.ensureIndex({"descripcion":1})
```

Creación de un índice sobre el campo email del subagregado vendedor:

```
db.items.ensureIndex({"vendedor.email":1})
```

Creación de un índice sobre el subagregado vendedor:

```
db.items.ensureIndex({"vendedor":1})
```

Creación de un índice compuesto formado por los campos descripción y estado:

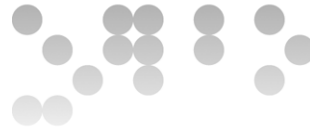
```
db.items.ensureIndex({"descripcion":1,"estado":1})
```

Creación de índices sobre localizaciones geográficas:

```
db.items.ensureIndex({"localizacion":"2dsphere"})
```

Una vez creado este tipo de índice podríamos hacer búsquedas sobre el campo indexado (localización) utilizando criterios de geolocalización. Por ejemplo, mostrar los artículos de la colección items cuyo punto prioritario de venta está en un radio de 1000mts respecto a un punto:

```
db.items.find(
  {"localizacion":
    {$near:
      {$geometry:
        {
          type:"Point",
          coordinates:[ 37.743670, -2.552276]
        },
        $maxDistance:1000
      }
    }
  }
)
```

6. MODIFICACIÓN DE DOCUMENTOS

Modificar el documento con descripción "Mando xBox negro" para que pase a estar vendido al usuario con email llopez@gmail.com y contraseña llopez:

```
db.items.update(
  {descripcion: "Mando xBox negro"},
  {$set:{
    estado: "vendido",
    comprador: {email: "llopez@gmail.com", psw: "llopez"}}
})
```

Actualizar la colección de usuarios para añadir al usuario con los datos: email: "ffernandez@gmail.com", psw:"ffernandez2", via:"C/Escalona", num:8, ciudad: "Barcelona" si no existiera:

```
db.usuarios.update(
  {email: "ffernandez@gmail.com", psw:"ffernandez2"},
  {$set:{email: "ffernandez@gmail.com", psw:"ffernandez2", direccion:
  {via:"C/Escalona", num:8, ciudad: "Barcelona"}}},
  {upsert:true}
)
```

Actualizar la colección de usuarios modificando el nombre de la C/ Escalona por C/Ancha:

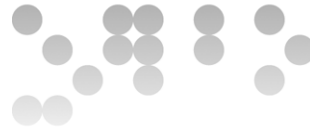
```
db.usuarios.update(
  {"direccion.via":"C/Escalona"},
  {$set: {"direccion.via":"C/Ancha"}},
  {multi:true}
)
```

Añadir una contraoferta al artículo Thermomix a cargo del usuario con email ggomez@gmail.com y contraseña "ggomez" por 73 euros, si no existiera previamente:

```
db.items.update(
  {descripcion:"Thermomix"},
  {$addToSet: {
    contraoferta: {email: "ggomez@gmail.com", psw: "ggomez",
    oferta: 73, fecha: new Date("2015, 5, 9")}}
})
```

Actualizar el documento cuya descripción es Thermomix vendiéndoselo a la oferta anterior:

```
db.items.update(
  {descripcion: "Thermomix"},
  {$inc: {precio: -7},
  $set: {estado: "vendido", comprador: {email: "ggomez@gmail.com",
  psw: "ggomez"}}
}
```



```
)
```

Eliminar el tag “entretenimiento” de todos los documentos de la colección items, siempre que exista:

```
db.items.update(  
  {tags: "entretenimiento" },  
  {$pull: {tags: "entretenimiento"}},  
  {multi: true}  
)
```

7. ELIMINAR DOCUMENTOS

```
db.items.remove({})
```

8. ELIMINAR COLECCIÓN

```
db.items.drop()
```

9. ELIMINAR LA BASE DE DATOS

```
db.dropDatabase()
```