Laboratorio de mongoDB

Ao final do documento encóntranse as instrucións para levantar un servidor de MongoDB.

Import e export de datos

Manual: https://www.mongodb.com/docs/manual/

Carga de documentos JSON

Importa un obxecto JSON

```
mongoimport --db <db-name> --collection <coll-name> --type json --file nombrearchivo.json
```

```
mongoimport -d <db-name> -c <coll-name> --type json --file nombrearchivo.json
```

O ficheiro JSON pode conter unha serie de obxectos JSON un a continuación do outro, sen separación de comas, nin corchetes nin nada.

contacto.json

```
{
   "nome":"fulano",
   "tlf":"666555444"
}
```

Importa un obxecto JSON

```
mongoimport --db libreta --collection contactos --type json --file contacto.json
```

contactos.json

```
{
   "nome":"fulanito",
   "tlf":"666555111"
}
{
   "nome":"menganita",
   "tlf":"666555222"
}
```

Importa un obxecto JSON

```
mongoimport --db libreta --collection contactos --type json --file contactos.json
```

Se pola contra se trata dunha lista/array hai que indicalo

Importa un array/lista de obxectos JSON

```
mongoimport --db <db-name> --collection <coll-name> --type json --jsonArray  --file nombrearchivo.json
```

lista_contactos.json

```
[{
   "nome":"mengana",
   "tlf":"666555333"
},
{
   "nome":"moriarty",
   "tlf":"666555222"
}
]
```

```
mongoimport --db libreta --collection contactos --type json --jsonArray --file lista_contactos.json
```

Tamén é posible importar documentos CSV/TSV

```
mongoimport -d <db-name> -c <coll-name> --type tsv --headerline --drop --file nombrearchivo.json
```

Export de documentos

En lugar do comando mongoimport pódese utilizar mongoexport para salvar datos a ficheiro desde a base de datos.

```
mongoexport --db libreta --collection contactos --type json --out export_contactos.json
```

```
mongoexport -d libreta -c contactos --type json --out export_contactos.json
```

É posible exportar unicamente algúns campos de interese, ou tamén o resultado dunha consulta.

Por exemplo:

```
mongoexport --db libreta --collection contactos --out datos.json --query='{"nome":"fulano"}'
```

Import/Export de ficheiros binarios BSON

Tamén é posible utilizar mongodump ou mongorestore para traballar con ficheiros binarios BSON.

```
mongodump -d nombreBaseDatos nombreFichero.bson
mongorestore -d nombreBaseDatos nombreFichero.bson
```

Búsquedas – (Query)

Manual: https://www.mongodb.com/docs/manual/
https://www.mongodb.com/docs/manual/crud/

Búsqueda simple

```
db.coleccion.find()
```

Búsqueda con condición simple

```
db.coleccion.find({query})
db.coleccion.find({"chave": valor})
```

Búsqueda con varias condicións

```
db.coleccion.find({"chave": valor, "chave2": valor2})
```

Búsqueda con condicionais
Pódense usar comparadores $lt, $lte, $gt, $gte, $ne. $in, ...

```
db.coleccions.find({"chave_numerica" : { "$gte" : 10 }})
```

Outros operadores

$in, $nin
$and, $or
$not, $nor
$exists
$type
...

Proxección
A función “find()” devolve o documento ao completo, mais podemos seleccionar os campos de interese, é dicir, facer unha proxección.

Podemos proxectar incluíndo:

```
db.coleccion.find({"chave": valor},{campo1si:1, campo2si:1, ..., campoNsi:1})
```

Ou excluíndo:

```
db.coleccion.find({"chave": valor},{campo1non:0, campo2si:0, ..., campoNsi:0})
```

Cursores e métodos de cursor

O método find() devolve un cursor. Existen diferentes métodos de cursor que podemos utilizar:
db.collectrion.find().sort()
.count()
.limit()
.skip()
.toArray()

Exemplo

Carga de datos

Exporto datos da SampleAPI – Beers
https://sampleapis.com/api-list/beers
Esta API devolve unha lista de obxectos JSON con datos sobre diferentes cervexas
A páxina web ofrece diferentes endpoints: ale, stouts e red-ale. Neste exemplo utilizaremos **ale**

```
curl -o ale.json https://api.sampleapis.com/beers/ale
```

```
mongoimport --db <db-name> --collection <coll-name> --type json --jsonArray --file nombrearchivo.json
```

```
mongoimport --db drinks --collection ale --type json --jsonArray --file ale.json
```

```
mongosh
show dbs
show collections
db.ale.findOne()
```

Búsqueda

Algunhas procuras:

Búsqueda de documentos que cumpren unha condición

```
db.ale.find({'name':'Guinness Extra Stout'})
```

```
db.ale.find({"id": {$lt:5}});
```

Búsqueda dun valor que conteña o texto

```
db.ale.find({"name" : {$regex : "Guinness"}});
```

Búsqueda con condicións en campos anidados

```
db.ale.find({ "rating.reviews":57 });
```

```
db.ale.find({ "rating.reviews": { $gt: 200} });
```

```
db.ale.find({ "rating.average": { $gt: 4} });
```

Búsquedas con operadores lóxicos

```
db.ale.find({$or: [{'name':'Guinness Extra Stout'},{'name':'Founders All Day IPA'}] })
```

```
db.ale.find({$and: [{'name':'Guinness Extra Stout'},{'rating.average':{ $gt: 3.75}}] })
```

Proxección

```
db.coleccion.find({"chave": valor},{campo1si:1, campo2si:1, ..., campoNsi:1})
```

```
db.ale.find({},{name:1})
# se non queremos o _id hai que excluílo explícitamente
db.ale.find({},{name:1,"_id":0})
```

Algúns métodos de cursor:

#Contar número de resultados

```
db.ale.find({}).count()
```

#limitar resultados

```
db.ale.find({}).limit(3)
```

#ordenar resultados

```
db.ale.find({},{name:1,"_id":0}).sort({"name":"asc"})
```

Proposta de exercicio

Carga nunha mesma database (drinks) pero en distinta colección (stout) os datos de cervexas stout que podes descargar da API en https://sampleapis.com

Descarga os datos a formato json

```
???
```

Carga os datos nunha nova colección

```
???
```

Compara o número de documentos na colección “ale” e na colección “stouts”

```
???
```

Mostra os documentos das cervexas stout

```
???
```

Mostra unicamente o nome e prezo de cada cerveza stouts

```
???
```

Mostra cervexas stout “Chocolate”

```
???
```

Busca cervexas stout con máis de 450 reviews

```
???
```

Cantas encontraches?

```
???
```

Mostra o seu nome, prezo e número de reviews

```
???
```

Conta o número de cervezas cunha puntuación por enriba de 4 e máis de 200 reviews en cada unha das coleccións

```
???
???
```

Mostra o top 5 (nome, prezo e puntuación) de cada unha das coleccións en relación á súa valoración media

```
???
???
```

Engade un novo campo “type” a cada documento, de xeito que indiques se se trata dunha cervexa “ale” ou “stout”.

```
???
???
```

Podes unir o contido de dúas coleccións (nova coleccion: birras)

```
db.stouts.aggregate([ { $unionWith: { coll: "ale" } }, { $out: "birras" } ])
```

Mostra o tipo, nome e prezo do top 10 das cervexas máis caras da nova colección:

```
???
```

Notarás que está a ordenar o campo “price” de xeito alfabético, xa que non é un valor numérico. Podes convertir estes datos usando a seguinte liña:

```
db.collection.updateMany({},  [{$set: { price: {$toDouble: { $substr: ["$price", 1, -1]  }}}}])
```

REPETIMOS: Mostra o tipo, nome e prezo do top 10 das cervexas máis caras da nova colección:

```
???
```

Mostra o nome e prezo da cervexa máis barata

```
???
```

Mostra o nome, prezo e tipo da cervexa máis barata con puntuación maior de 4 e máis de 200 reviews

```
???
```

Exporta a formato json o contido da nova colección “birra”

```
???
```

Extra:

Realiza as mesmas consultas desde un notebook python vía pymongo.

Preparación do escenario

MongoDB con docker

Lanzar un novo contedor de mongo:

```
docker run --name meu-mongo -d -p 27017:27017 mongo
```

Abrir unha consola (interactiva -it) dentro do contedor (consola de bash):

```
docker exec -it meu-mongo bash
```

Executar o cliente de mongo contra localhost:

```
root@maquina_mongo:/# mongosh
```

Opción todo-en-1:

```
docker exec -it meu-mongo mongosh
```

Instalación de ferramentas extra dentro do contedor

Polo xeral os contedores de aplicacións non dispoñen de ferramentas comúns como editores de texto (nano) ou ferramentas que nos permiten descargar datos da rede (cur, wget, ...) pero podemos instalalos facilmente se son necesarios.

```
docker exec -it meu-mongo bash

# actualizar listas de software
apt update

# instalar curl
apt install curl

# instalar wget
apt install wget
```