

INTRODUÇÃO AO MONGODB

//Componentes do MongoDB

mongod – O servidor do banco de dados

mongos - Sharding router.

mongo - O cliente do banco de dados / database shell (uses interactive javascript).

- 1) O MongoDB utiliza um diretório base para armazenar os dados.

Ex.: Criar o diretório **C:\data\db**

- 2) Chamar o *daemon* do **mongodb**, com o comando “C:\mongodb-win32-x86_64-2008plus-ssl-4.0.9\bin\mongod”

```
//Especificar outra localizacao do banco de dados  
mongod.exe --dbpath "d:\set up\mongodb\data"
```

- 3) Chamar o shell do mongodb com o comando **mongo**.

//Connexão

//Especificar porta

```
mongo --port 27017
```

//Especificar host e porta

```
mongo --host mongodb0.example.com --port 27017
```

```
mongo --host mongodb0.example.com:27017
```

```
mongo "mongodb://mongodb0.example.com:27017"
```

//Conexão utilizando autenticação

```
mongo --username alice --password --authenticationDatabase admin --  
host mongodb0.examples.com --port 28015
```

//Conecta-se ao banco de dados

```
db = connect("localhost:27017/myDatabase")
```

//Banco de dados

//Mostra todos os bancos de dados

```
show dbs
```

//Mostra o banco de dados atual

```
db
```

//Muda para outro banco de dados

```
use <database>
```

//Dropando um banco de dados

```
db.dropDatabase()
```

//Autenticação

//Cria um usuário adminsterador

```
use admin

db.createUser(
  {
    user: "myUserAdmin",
    pwd: "abc123",
    roles: [ { role: "userAdminAnyDatabase", db: "admin" }, "readWriteAnyDatabase" ]
  }
)
```

//Reinicia o servidor

```
db.adminCommand( { shutdown: 1 } )
```

//Habilita autenticação no servidor

```
mongod --auth --port 27017 --dbpath Z:\data\db
```

// A partir de agora, os clients que se conectarem serão solicitados a fornecer credenciais.

//Conecta-se ao banco de dados

```
mongo --port 27017 -u "myUserAdmin" --authenticationDatabase "admin" -p
```

//Collection == Tabela

//Mostra todas as collections do banco de dados

```
show collections
```

//Mostrando dados da collection alunos

```
db.getCollection("alunos").find()  
ou
```

```
db.alunos.find()
```

//Criar um documento

```
db.colors.save({name:"red",value:"FF0000"});
```

//Verificar se o documento foi criado

```
db.colors.find();  
{ "_id" : ObjectId("5ce1e8a84d9202990067419d"), "name" : "red", "value" : "FF0000" }
```

//Inserindo dados

```
db.usuarios.insert( {  
    nome: "Usuario Teste",  
    cidade: "Curitiba",  
    estado: "Parana"  
}  
)  
WriteResult({ "nInserted" : 1 })
```

```
db.usuarios.find()  
{ "_id" : ObjectId("5ce1e9d64d9202990067419e"), "nome" : "Usuario Teste", "cidade" : "Curitiba", "estado" : "Parana0" }
```

//Usando variaveis/objetos

```
MeusDados = {  
    nome: "Professor",  
    cidade: "Curitiba",
```

```
        estado: "Parana"
    }
    db.meudb.save(MeusDados)
```

//Usando estruturas mais complexas

```
MeusDados = {
    nome: "Professor",
    endereco: {cidade: "Curitiba", estado: "Parana"}
}
```

//Remove usuário cujo estado seja Parana

```
db.usuarios.remove( { estado: "Parana" } )
```

//Atualizando dados

```
db.usuarios.update({ "nome" : "Usuario Teste" },{$set:{ "cidade" : "New York"}},{multi:true})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

```
db.usuarios.find()
{ "_id" : ObjectId("5ce1ec514d920299006741a0"), "nome" : "Usuario Teste", "cidade" : "New York", "estado" : "Parana" }
```

//Cria objeto/variavel

```
MeusDados = {
    nome: "Marta",
    id: 1,
    assuntos: [ "C", "C++", "Java" ]
}
```

//Insere dados

```
db.alunos.insert(MeusDados)
```

//Consultas

//Cria coleção TiposSanguineos

```
db.TiposSanguineos.insertMany([
  { desc: "A+", cod: 1 },
  { desc: "A-", cod: 2 },
  { desc: "B+", cod: 3 },
  { desc: "B-", cod: 4 },
  { desc: "AB+", cod: 5 },
  { desc: "AB-", cod: 6 },
  { desc: "O+", cod: 7 },
  { desc: "O-", cod: 8 }
])
```

// Retorna todos os registros da coleção

```
db.TiposSanguineos.find( {} )
```

//Retorna apenas os itens cuja descrição seja igual a A+.

```
db.TiposSanguineos.find( { desc: "A+" } )
```

//O operador **\$in**:

```
db.TiposSanguineos.find( { desc: { $in: [ "A+", "O+" ] } } )
```

//O operador **\$or** e o operador **\$gt (maior que)**:

```
db.TiposSanguineos.find( { $or: [ { desc: "A+" }, { cod: { $gt: 4 } } ] } )
```

Outros operadores:

\$lt (menor que)

\$lte (menor ou igual)

\$gte (maior ou igual)

//Índices possibilitam consultas por texto

```
db.TiposSanguineos.createIndex( { desc: "text" } )
```

//Executa a consulta

```
db.TiposSanguineos.find( { $text: { $search: "O" } } )
```

```
{ "_id" : ObjectId("5ce439e52d89562e0ab21c73"), "desc" : "O-", "cod" : 8 }
```

```
{ "_id" : ObjectId("5ce439e52d89562e0ab21c72"), "desc" : "O+", "cod" : 7 }
```


//Insert Many

//Inserindo múltiplos registros

```
try {
  db.products.insertMany( [
    { item: "card", qty: 15 },
    { item: "envelope", qty: 20 },
    { item: "stamps" , qty: 30 }
  ] );
} catch (e) {
  print (e);
}
```

//Inserindo múltiplos registros com id pre-definido.

```
try {
  db.products.insertMany( [
    { _id: 10, item: "large box", qty: 20 },
    { _id: 11, item: "small box", qty: 55 },
    { _id: 12, item: "medium box", qty: 30 }
  ] );
} catch (e) {
  print (e);
}
```

//Ao tentar reinserir os registros com mesmo id, obtem-se a seguinte mensagem de erro.

```
> try {
...   db.products.insertMany( [
...     { _id: 10, item: "large box", qty: 20 },
...     { _id: 11, item: "small box", qty: 55 },
...     { _id: 12, item: "medium box", qty: 30 }
...   ] );
... } catch (e) {
...   print (e);
... }
```

```
... }
BulkWriteError({
  "writeErrors" : [
    {
      "index" : 0,
      "code" : 11000,
      "errmsg" : "E11000 duplicate key error collection: teste.products index: _id_ dup key: { : 10.0 }",
      "op" : {
        "_id" : 10,
        "item" : "large box",
        "qty" : 20
      }
    }
  ],
  "writeConcernErrors" : [ ],
  "nInserted" : 0,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})
```

//Agregação de coleções (dados)

//Cria coleção TiposSanguineos

```
db.TiposSanguineos.insertMany(  
  [  
    { desc: "A+", cod: 1 },  
    { desc: "A-", cod: 2 },  
    { desc: "B+", cod: 3 },  
    { desc: "B-", cod: 4 },  
    { desc: "AB+", cod: 5 },  
    { desc: "AB-", cod: 6 },  
    { desc: "O+", cod: 7 },  
    { desc: "O-", cod: 8 }  
  ]  
)
```

//Cria coleção Alunos

```
db.alunos.insert( {  
  nome: "Maria",  
  matricula: 1,  
  idade: 9,  
  tiposanguineo: 7  
}  
)
```

//Agregação Alunos<-->TiposSanguineos

```
db.alunos.aggregate( [  
  {  
    $lookup:  
    {  
      from: "TiposSanguineos",  
      localField: "tiposanguineo",  
      foreignField: "cod",  
      as: "TipoSanguineoAluno"  
    }  
  }  
])
```

//Resultado da agregação

```
{
  "_id" : ObjectId("5ce3517541fd3d27f63046db"),
  "nome" : "Maria",
  "matricula" : 1,
  "idade" : 9,
  "tiposanguineo" : 7,
  "TipoSanguineoAluno" : [
    {
      "_id" : ObjectId("5ce3513641fd3d27f63046d9"),
      "desc" : " O+",
      "cod" : 7
    }
  ]
}
```

//Inserindo dados binários (ex. arquivos mp4, jpg, etc)

//Verificar collections antes de ler os arquivos

```
> show collections
TiposSanguineos
colors
meudb
products
stores
usuarios
>
```

//Utilizar o programa mongofiles

```
C:\mongodb-win32-x86_64-2008plus-ssl-4.0.9\bin>mongofiles -d teste put C:\Users\sergio\Desktop\Karate\Bunkai.mp4
2019-05-28T16:09:16.601-0300  connected to: localhost
2019-05-28T16:09:17.229-0300  added file: C:\Users\sergio\Desktop\Karate\Bunkai.mp4
```

//Estado das collections após inserir o arquivo.

```
> show collections
```

```
TiposSanguineos
```

```
colors
```

```
fs.chunks
```

```
fs.files
```

```
meudb
```

```
products
```

```
stores
```

```
usuarios
```

//db.files contém a descrição do arquivo

```
> db.fs.files.find()
```

```
{ "_id" : ObjectId("5ced875c40fbaf42a0ef74dd"), "chunkSize" : 261120, "uploadDate" : ISODate("2019-05-28T19:09:17.250Z"), "length" : 9202371, "md5" : "42269d351b091efbd837fc503b020d5f", "filename" : "C:\\Users\\sergio\\Desktop\\Karate\\Bunkai.mp4" }
```

//db.chunks contém os dados do arquivo

```
db.fs.chunks.find()
```

```
mIEAexXtSZOSXUPJQwiPNwCeQyodlaKkkMlxYGvZV2vfuWOVM8sy3wSSRHma7Q0j8E7CUX8jsh  
WTsd6Hf6/vfGoILzX3HovZWtv+vUDI+Zp8B8p/XqHycwxvI7Y/owV0hj6sUdTSWln4+eWA8JwvPu  
0Wnj4B1ScxkIMH5uCVKKutjFDKq4c1r23XTNpksnrGC2z1I0CyYjAD8cbLtilvqoEbcsQXcP/AIExL0N  
ND7gbDApssw5KW8FcldSBI0TLfYzxhM0oqlcpnRnv2GBsLlk0bBCO2Hnzx8NjdNofN/gq8Z/PLMjxc  
qJ0inMNBKdEbUjD7NkYHGP/sPAGKNMlpj0Wv8FnGXmryg5waSs2Hx3Hn49OyJsPeL0wPFx+CFN  
ASaUpP7CiUCF2Gmqmz8iZKm1jNjWEyODOvif2xvpxuond8NufG7r2zbOX1kFp+1d3FCGf8c1QjPC  
YmWN+c4Frh38RQUBALWeulMBJyPBam1F89Bc7H+Pryjtzs7EmInvkiTCS56bPbnockfcE2D") }
```

//Backup e Restore de dados

//Situação atual da base de dados

```
> show dbs
admin      0.000GB
config     0.000GB
local      0.000GB
myDatabase 0.000GB
test      0.000GB
teste      0.009GB
```

//Backup da base de dados

```
mongodump --db test --out c:\data\bkp
```

//Dropando o banco de dados

```
> use test
switched to db test
> db.dropDatabase()
{ "dropped" : "test", "ok" : 1 }
> show dbs
admin      0.000GB
config     0.000GB
local      0.000GB
myDatabase 0.000GB
teste      0.009GB
```

//Restaurando a base de dados

```
mongorestore --db test c:\data\bkp\test
```

//Verificando a base restaurada

```
> show dbs
admin      0.000GB
config     0.000GB
local      0.000GB
myDatabase 0.000GB
test      0.000GB
teste      0.009GB
```

//Exportação e Importação de dados

Exportação de dados em formato JSON ou CSV.

//Exportar para arquivo em formato json.

```
mongoexport --db test --collection alunos --out alunos.json
```

//Conteúdo do arquivo json exportado

```
{"_id":{"$_oid":"5ce2e4729a8af6c2f5813287"},"nome":"Marta","id":1.0,"assuntos":["C","C++","Java"]}
```

//Conteúdo ALTERADO do arquivo json

```
{"_id":{"$_oid":"5ce2e4729a8af6c2f5813288"},"nome":"Tiago","id":2.0,"assuntos":["Python","CSS","HTML"]}
```

//Importando arquivos

```
mongoimport -d test -c alunos --file alunos.json
```

//Conteúdo collection alunos após importação.

```
> db.alunos.find().pretty()
{
  "_id" : ObjectId("5ce2e4729a8af6c2f5813287"),
  "nome" : "Marta",
  "id" : 1,
  "assuntos" : [
    "C",
    "C++",
    "Java"
  ]
}
{
  "_id" : ObjectId("5ce2e4729a8af6c2f5813288"),
  "nome" : "Tiago",
  "id" : 2,
  "assuntos" : [
    "Python",
    "CSS",
    "HTML"
  ]
}
```

```
]
}
```

// Exportando dados com execução de query. Restringindo os dados exportados.

```
mongoexport -d test -c alunos -q "{nome:'Marta'}" -o alunos2.json
```

```
{"_id":{"_id":"5ce2e4729a8af6c2f5813287"},"nome":"Marta","id":1.0,"assuntos":["C","C++","Java"]}
```

// Exportando dados no formato CSV.

```
mongoexport -d test -c alunos --type csv --fields nome,id,assuntos -o alunos2.csv
```

```
nome,id,assuntos
Marta,1,"[""C""",""C++""",""Java"""]"
Tiago,2,"[""Python""",""CSS""",""HTML"""]"
```