

Capítulo 4 – MongoDB

Introdução

O **MongoDB** é um banco de dados NoSQL orientado a documentos. Armazena informações em **coleções de documentos BSON** (JSON binário).

- Estrutura flexível, sem schema fixo.
- Nome vem de *humongous* (enorme).
- Popular em aplicações web modernas.

Estrutura básica

- Banco → coleções.
- Coleção → documentos.
- Documento → pares chave/valor.

Exemplo:

```
{
  "titulo": "Dom Casmurro",
  "autor": "Machado de Assis",
  "ano": 1899,
  "edicoes": [ { "idioma": "Português", "ano": 1900 } ],
  "editora": { "nome": "Garnier", "pais": "Brasil" }
}
```

Dia 1 – CRUD e Nesting

Criando banco

```
use biblioteca
```

Inserindo

```
db.livros.insert({ titulo: "1984", autor: "George Orwell", ano: 1949 })
```

Consultando

```
db.livros.find()
db.livros.findOne({ titulo: "1984" })
```

Atualizando

```
db.livros.update(  
  { titulo: "1984" },  
  { $set: { paginas: 350 } }  
)
```

Deletando

```
db.livros.remove({ titulo: "1984" })
```

Nesting

```
db.livros.insert({  
  titulo: "O Senhor dos Anéis",  
  autor: "Tolkien",  
  edicoes: [ { idioma: "PT", ano: 2000 }, { idioma: "EN", ano: 1954 } ],  
  editora: { nome: "Allen & Unwin", pais: "Reino Unido" }  
})
```

Consulta aninhada:

```
db.livros.find({ "editora.nome": "Allen & Unwin" })
```

Dia 2 – Indexação, Agregação e MapReduce

Índices

```
db.livros.createIndex({ titulo: 1 })
```

Operadores

```
db.livros.find({ ano: { $in: [2010, 2012] } })
```

Aggregation Framework

```
db.livros.aggregate([  
  { $group: { _id: "$ano", total: { $sum: 1 } } },  
  { $sort: { total: -1 } }  
])
```

MapReduce

```
map = function() { this.autores.forEach(a => emit(a, 1)); };
reduce = function(chave, valores) { return Array.sum(valores); };
db.livros.mapReduce(map, reduce, { out: "contagem_autores" })
```

Dia 3 – Replica Sets, Sharding, GeoSpatial e GridFS

Replica Sets

```
rs.initiate({ _id: "meuReplicaSet", members: [
  { _id: 0, host: "localhost:27017" },
  { _id: 1, host: "localhost:27018" },
  { _id: 2, host: "localhost:27019" }
]})
```

Sharding

- Distribui dados em shards.
- mongos roteia queries.
- Chaves ruins → hotspots.
- Chaves boas → distribuição equilibrada.

GeoSpatial

```
db.locais.createIndex({ coordenadas: "2dsphere" })
```

GridFS

```
mongofiles -d meusArquivos put foto.png
```

Wrap-Up

- CRUD simples e natural.
- Indexação e Aggregation poderosas.
- Replica Sets e Sharding garantem escala.
- GeoSpatial e GridFS ampliam aplicações.

👉 MongoDB = simplicidade + escalabilidade + robustez.