

Relatório — Infraestrutura MongoDB

Aluno: Caio Barroso | **Disciplina:** Infraestrutura MongoDB | **Professor:** Ricardo Barrionuevo Garcia

Rubrica / Evidências (o que o avaliador procura)

Item da rubrica	Onde está no relatório	Evidência (print)
1) Diferenças e semelhanças (SQL vs NoSQL)	Parte 1.1	Texto
1) NoSQL é melhor/pior? Destaques e justificativa	Parte 1.2	Texto
1) Cenário NoSQL e cenário SQL	Parte 1.3	Texto
1) Tipos de NoSQL + características + fontes	Parte 1.4	Texto
2) A4 — Conteúdo das coleções	A4	evidencias/A04_conteudo_colecoes.png
2) A5 — Filtro igualdade em DEPARTAMENTO	A5	evidencias/A05_filtro_igualdade_departamento.png
2) A6 — Salário > 2000 em FUNCIONARIO	A6	evidencias/A06_salario_maior_2000.png
2) A7 — distinct(nome) em DEPENDENTE	A7	evidencias/A07_distinct_dependente_nome.png
2) A8 — update em uma coleção	A8	evidencias/A08_update_funcionario.png
2) A9 — delete/remove em uma coleção	A9	evidencias/A09_delete_dependente.png
3) B2 — Modelagem 1-N (referência)	B2	evidencias/B01_modelagem_1N.png
3) B3 — Modelagem N-N	B3	evidencias/B02_modelagem_NN.png
3) C2/C3/C4 — Índices/TTL/Text/Agregações (itens principais)	C Índices e C Agregações	evidencias/C01_indexar1_indices.png evidencias/C02_ttl_antes.png evidencias/C03_ttl_depois.png evidencias/C04_text_index_busca.png evidencias/C05_agregacoes_por_uf.png
4) D3/D4 — Replica set (criação + leitura no secundário)	D	evidencias/D01_mongod_rs_subindo.png evidencias/D02_rs_status.png evidencias/D13_insert_5_filmes_primary.png evidencias/D14_read_filmes_secondary.png
4) E3 — Sharding (mongos + add shards + shardCollection + distribuição)	E	evidencias/E06_mongos_conectado_27050.png evidencias/E07_add_shards_status.png evidencias/E08_enable_sharding_db.png evidencias/E09_shard_collection.png evidencias/E10_insert_1000_docs.png evidencias/E11_distribuicao_shards.png
4) F1/F2/F3 — Storage engines e Segurança	F1, F2, F3	evidencias/F01_mmapv1_storage_engine.png evidencias/F02_mmapv1_produtos_insert_4.png evidencias/F04_wiredTiger_storage_engine.png evidencias/F05_wiredTiger_lugares_insert_4.png evidencias/F09_secure_login_db_connectionStatus.png evidencias/F11_createUser_CB_desenv.png evidencias/F12_login_CB_desenv.png evidencias/F13_insert_funcionarios_4.png
4) G9–G13 — Backup/kill/delete/restore/validação	G	evidencias/G11_backup_mongodump.png evidencias/G10_taskkill.png evidencias/G11_apagar_dados.png evidencias/G12_restore_ok.png evidencias/G13_contagem_colecoes_exerc4b.png

Parte 1 — Conceitos

1.1 — 3 diferenças e 3 semelhanças entre SQL e NoSQL

Diferenças (3)

- 1) SQL trabalha tipicamente com **esquema rígido** (tabelas/columnas e constraints); NoSQL tende a aceitar **esquema flexível**, facilitando evoluções incrementais do modelo.
- 2) SQL costuma priorizar **consistência forte** e transações com múltiplas entidades; NoSQL frequentemente prioriza **escala/alta disponibilidade**, aceitando consistência eventual em alguns cenários.
- 3) SQL é muito bom para dados relacionais e joins; NoSQL geralmente evita joins complexos (ou usa alternativas), apostando em **modelagem orientada ao acesso e distribuição horizontal**.

Semelhanças (3)

- 1) Ambos existem para **persistência e consulta** de dados de forma confiável.
- 2) Ambos suportam **índices**, controle de acesso, logs e rotinas de backup/restore.
- 3) Em ambos, performance depende de modelagem + índices + padrão de consultas e de operação (monitoramento/capacidade).

1.2 — NoSQL é melhor ou pior que SQL?

Resposta

Não existe “melhor/pior” universal. **NoSQL** se destaca em **escala horizontal**, alta taxa de leitura/escrita, dados semi-estruturados e evolução rápida de schema. **SQL** se destaca quando a prioridade é **integridade forte**, transações complexas e relacionamento intenso entre entidades (joins) com regras rígidas. A decisão correta depende de volume, padrão de acesso, SLA de consistência e custo operacional.

1.3 — Um cenário NoSQL e um cenário SQL

Cenário NoSQL

Telemetria/eventos de aplicação e perfis de usuário que evoluem (campos variáveis), com necessidade de escalar por volume e consultas por atributos/ids. Ex.: logs, eventos de navegação, catálogo com campos opcionais.

Cenário SQL

Sistema financeiro/ERP com regras fortes, auditoria e transações entre várias entidades (pedido/pagamento/faturamento), exigindo consistência e integridade referencial.

1.4 — Tipos de bancos NoSQL e características

Resposta

- **Key-Value:** chave → valor. Muito rápido; bom para cache, sessão, feature flags, leitura direta por chave.
- **Document:** documentos (JSON/BSON). Flexível; bom para dados semi-estruturados e evolução rápida (ex.: MongoDB).
- **Wide-column/Column-family:** famílias de colunas; bom para escala distribuída e consultas por partição.
- **Graph:** nós/arestas; excelente para relacionamentos, recomendação, fraude e análise de redes.

Fontes:

- AWS (conceitos e categorias NoSQL): <https://docs.aws.amazon.com/whitepapers/latest/choosing-an-aws-nosql-database/conclusion.html>
- MongoDB (conceito de document database): <https://www.mongodb.com/resources/basics/databases/document-databases>

Parte 2 — Prática (A: Instalações e criações)

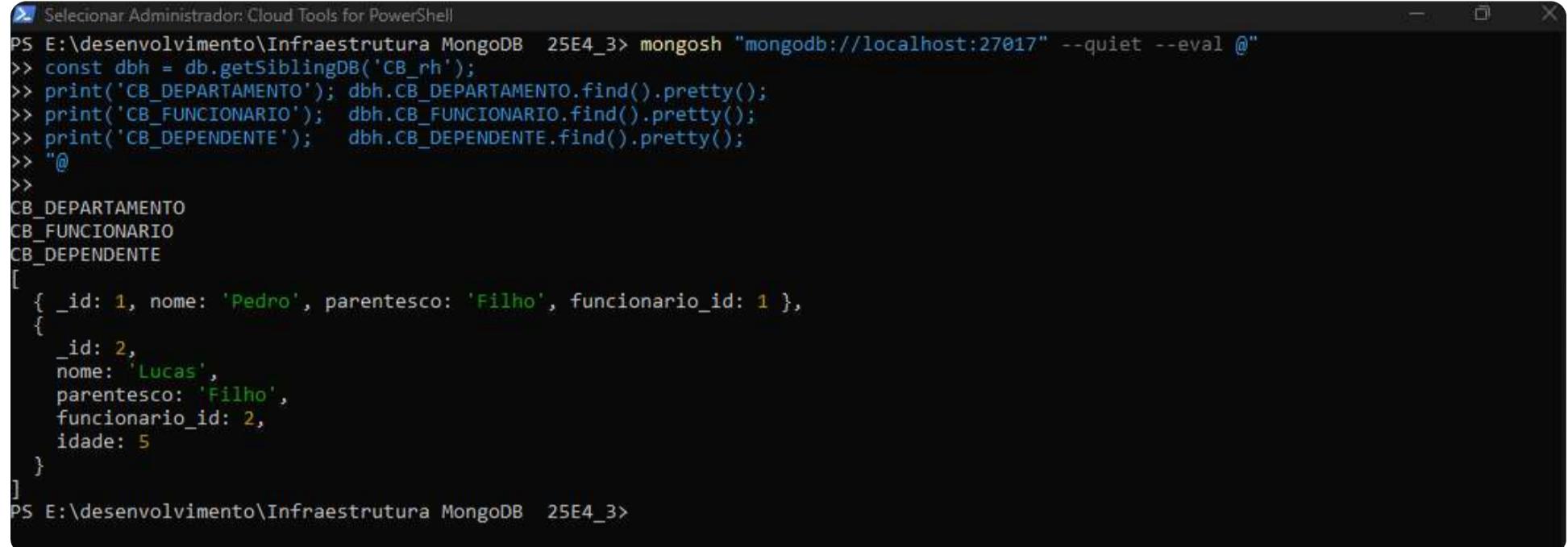
A4 — Mostrar o conteúdo de cada coleção

O que foi pedido

Mostrar o conteúdo das coleções do database **CB_rh**.

Comandos

```
use CB_rh
db.CB_DEPARTAMENTO.find().pretty()
db.CB_FUNCIONARIO.find().pretty()
db.CB_DEPENDENTE.find().pretty()
```



```
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> mongosh "mongodb://localhost:27017" --quiet --eval @@
>> const dbh = db.getSiblingDB('CB_rh');
>> print('CB_DEPARTAMENTO'); dbh.CB_DEPARTAMENTO.find().pretty();
>> print('CB_FUNCIONARIO'); dbh.CB_FUNCIONARIO.find().pretty();
>> print('CB_DEPENDENTE'); dbh.CB_DEPENDENTE.find().pretty();
>> "@
>>
CB_DEPARTAMENTO
CB_FUNCIONARIO
CB_DEPENDENTE
[
  { _id: 1, nome: 'Pedro', parentesco: 'Filho', funcionario_id: 1 },
  {
    _id: 2,
    nome: 'Lucas',
    parentesco: 'Filho',
    funcionario_id: 2,
    idade: 5
  }
]
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3>
```

evidencias/A04_conteudo_colecoes.png

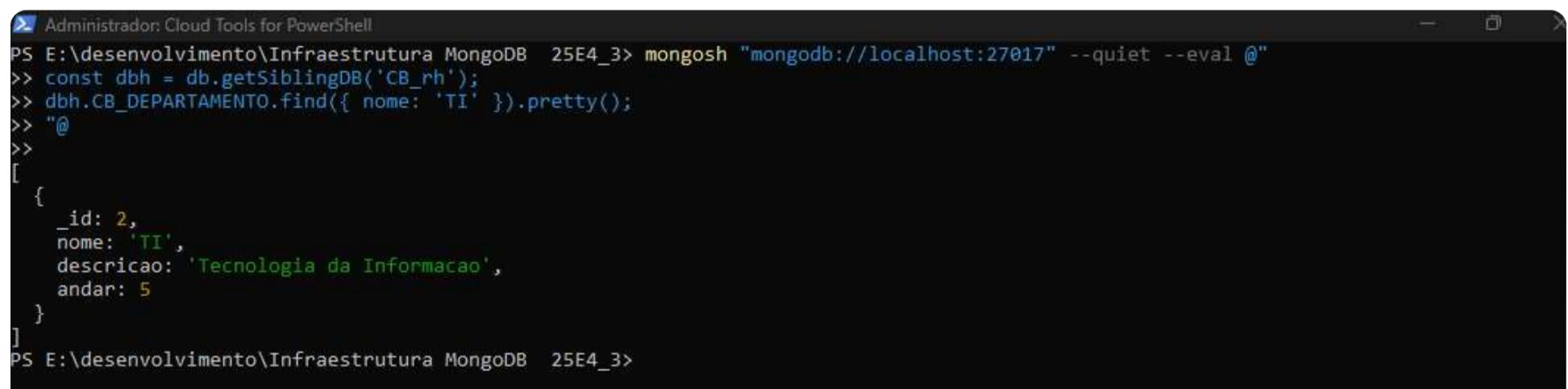
A5 — Filtro por igualdade em DEPARTAMENTO

O que foi pedido

Filtro de igualdade na coleção **CB_DEPARTAMENTO** por descrição.

Comandos

```
use CB_rh
db.CB_DEPARTAMENTO.find({ descricao: "..." }).pretty()
```



```
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> mongosh "mongodb://localhost:27017" --quiet --eval @@
>> const dbh = db.getSiblingDB('CB_rh');
>> dbh.CB_DEPARTAMENTO.find({ nome: 'TI' }).pretty();
>> "@
>>
[
  {
    _id: 2,
    nome: 'TI',
    descricao: 'Tecnologia da Informacao',
    andar: 5
  }
]
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3>
```

evidencias/A05_filtro_igualdade_departamento.png

A6 — Funcionários com salário acima de 2000

O que foi pedido

Mostrar em **CB_FUNCIONARIO** quem tem salário > **2000**.

Comandos

```
use CB_rh
db.CB_FUNCIONARIO.find({ salario: { $gt: 2000 } }).pretty()
```

```
> mongosh mongodb://localhost:27017/?directConnection=true&serverSelectionTimeoutMS=2000
CB_rh> db.CB_FUNCIONARIO.find({ salario: { $gt: 2000 } }).pretty()
...
[ {
  _id: 101,
  nome: 'Ana',
  cargo: 'Analista',
  salario: 2500,
  deptoId: 1
},
{
  _id: 102,
  nome: 'Bruno',
  cargo: 'Dev',
  salario: 5500,
  deptoId: 1,
  skills: [ 'node', 'mongo' ]
}
]
CB_rh>
```

evidencias/A06_salario_maior_2000.png

A7 — Distinct no atributo nome em DEPENDENTE

O que foi pedido

Executar **distinct** no atributo **nome** na coleção **CB_DEPENDENTE**.

Comandos

```
use CB_rh
db.CB_DEPENDENTE.distinct("nome")
```

```
> Administrador: Cloud Tools for PowerShell
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> mongosh "mongodb://localhost:27017" --quiet --eval @@
>> const dbh = db.getSiblingDB('CB_rh');
>> printjson(dbh.CB_DEPENDENTE.distinct('nome'));
>> "@
>>
[ 'Lucas', 'Pedro'
]
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3>
```

evidencias/A07_distinct_dependente_nome.png

A8 — Update em uma coleção

O que foi pedido

Executar um **update** em uma coleção.

Comandos

```
use CB_rh
db.CB_FUNCIONARIO.updateOne(
  { _id: ... },
  { $set: { ... } }
)
db.CB_FUNCIONARIO.find({ _id: ... }).pretty()
```

```
mongosh mongodb://localhost:27017/?directConnection=true&serverSelectionTimeoutMS=2000
CB_rh> db.CB_FUNCIONARIO.updateOne(
...   { _id: 102 },
...   { $set: { salario: 6000, atualizadoEm: new Date() } }
... );
... db.CB_FUNCIONARIO.find({ _id: 102 }).pretty();
...
[
  {
    _id: 102,
    nome: 'Bruno',
    cargo: 'Dev',
    salario: 6000,
    deptoId: 1,
    skills: [ 'node', 'mongo' ],
    atualizadoEm: ISODate('2026-01-07T23:37:51.982Z')
  }
]
CB_rh>
```

evidencias/A08_update_funcionario.png

A9 — Delete (remove) em uma coleção

O que foi pedido

Executar um **delete/remove** em uma coleção.

Comandos

```
use CB_rh
db.CB_DEPENDENTE.deleteOne({ _id: ... })
db.CB_DEPENDENTE.find().pretty()
```

```
mongosh mongodb://localhost:27017/?directConnection=true&serverSelectionTimeoutMS=2000
CB_rh> db.CB_DEPENDENTE.deleteOne({ _id: 1003 });
... db.CB_DEPENDENTE.find().pretty();
...
[
  { _id: 1001, nome: 'Davi', parentesco: 'Filho', funcionarioId: 101 },
  { _id: 1002, nome: 'Eva', parentesco: 'Filha', funcionarioId: 102, idade: 7
}
]
CB_rh>
```

evidencias/A09_delete_dependente.png

Parte 2 — Prática (B: Modelagem)

B2 — Modelagem 1-N com referência

Comandos (resumo)

```
use CB_modelo
db.CB_col1a.insertMany([ {...}, {...} ])

db.CB_col2a.insertMany([
  { ..., col1a_id: ObjectId("...") },
  { ..., col1a_id: ObjectId("...") },
  { ..., col1a_id: ObjectId("...") },
  { ..., col1a_id: ObjectId("...") }
])
```

Evidência

evidencias/B01_modelagem_1N.png

```
Administrador: Cloud Tools for PowerShell
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> $path = ".\scripts\B_modelagem\B01_1N.js"
>>
>> @'
>> /**
>> * Parte 2B - Modelagem 1:N (referência)
>> * DB: CB_modelo
>> * Coleções: CB_col1a (lado 1), CB_col2a (lado N)
>> */
>>
>> const dbm = db.getSiblingDB("CB_modelo");
>>
>> // drop seguro
>> const cols = dbm.getCollectionNames();
>> if (cols.includes("CB_col1a")) dbm.CB_col1a.drop();
>> if (cols.includes("CB_col2a")) dbm.CB_col2a.drop();
>>
>> dbm.createCollection("CB_col1a");
>> dbm.createCollection("CB_col2a");
>>
>> dbm.CB_col1a.insertMany([
>>   { _id: 1, nome: "Mae_A", tipo: "MAE" },
>>   { _id: 2, nome: "Mae_B", tipo: "MAE" }
>> ]);
>>
>> dbm.CB_col2a.insertMany([
>>   { _id: 101, nome: "Filha_A1", maeId: 1 },
>>   { _id: 102, nome: "Filha_A2", maeId: 1 },
>>   { _id: 201, nome: "Filha_B1", maeId: 2 },
>>   { _id: 202, nome: "Filha_B2", maeId: 2 }
>> ]);
>>
>> print("OK: 1:N criado. Prova: filhas da Mae_A (maeId=1)");
>>
>> // força materialização + impressão
>> const filhas = dbm.CB_col2a.find({ maeId: 1 }).toArray();
>> printjson(filhas);
>> '@ | Set-Content -Encoding UTF8 $path
>>
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> mongosh --file .\scripts\B_modelagem\B01_1N.js
>>
OK: 1:N criado. Prova: filhas da Mae_A (maeId=1)
[
  {
    _id: 101,
    nome: 'Filha_A1',
    maeId: 1
  },
  {
    _id: 102,
    nome: 'Filha_A2',
    maeId: 1
  }
]
```

evidencias/B01_modelagem_1N.png

B3 — Modelagem N-N

Comandos (resumo)

```
use CB_modelo
db.CB_col1b.insertMany([ {...}, {...} ])
db.CB_col2b.insertMany([ {...}, {...} ])
// se houver coleção de ligação, ela foi criada e populada no mesmo f
```

Evidência

evidencias/B02_modelagem_NN.png

```
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> $path = ".\scripts\B_modelagem\B02_NN.js"
>>
>> @'
>> /**
>> * Parte 2B - Modelagem N:N (referência via coleção de relacionamento)
>> * DB: CB_modelo
>> * Coleções: CB_col1b, CB_col2b, CB_rel1b2b
>> */
>>
>> const dbm = db.getSiblingDB("CB_modelo");
>>
>> const cols = dbm.getCollectionNames();
>> if (cols.includes("CB_col1b")) dbm.CB_col1b.drop();
>> if (cols.includes("CB_col2b")) dbm.CB_col2b.drop();
>> if (cols.includes("CB_rel1b2b")) dbm.CB_rel1b2b.drop();
>>
>> dbm.createCollection("CB_col1b");
>> dbm.createCollection("CB_col2b");
>> dbm.createCollection("CB_rel1b2b");
>>
>> // Entidade A: 3 docs
>> dbm.CB_col1b.insertMany([
>>   { _id: 1, nome: "A1" },
>>   { _id: 2, nome: "A2" },
>>   { _id: 3, nome: "A3" }
>> ]);
>>
>> // Entidade B: 3 docs
>> dbm.CB_col2b.insertMany([
>>   { _id: 10, nome: "B10" },
>>   { _id: 20, nome: "B20" },
>>   { _id: 30, nome: "B30" }
>> ]);
>>
>> // Relações (6 docs) para provar N:N
>> dbm.CB_rel1b2b.insertMany([
>>   { _id: 1001, aId: 1, bId: 10 },
>>   { _id: 1002, aId: 1, bId: 20 },
>>   { _id: 1003, aId: 2, bId: 20 },
>>   { _id: 1004, aId: 2, bId: 30 },
>>   { _id: 1005, aId: 3, bId: 10 },
>>   { _id: 1006, aId: 3, bId: 30 }
>> ]);
>>
>> print("OK: N:N criado.");
>>
>> print("Prova 1: relações de A1 (aId=1)");
>> printjson(dbm.CB_rel1b2b.find({ aId: 1 }).toArray());
>>
>> print("Prova 2: A's relacionados ao B20 (bId=20)");
>> printjson(dbm.CB_rel1b2b.find({ bId: 20 }).toArray());
>> '@ | Set-Content -Encoding UTF8 $path
>>
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> mongosh --file .\scripts\B_modelagem\B02_NN.js
>>
OK: N:N criado.
Prova 1: relações de A1 (aId=1)
[
  {
    _id: 1001,
    aId: 1,
    bId: 10
  },
  {
    _id: 1002,
    aId: 1,
    bId: 20
  }
]
Prova 2: A's relacionados ao B20 (bId=20)
[
  {
    _id: 1002,
    aId: 1,
    bId: 20
  },
  {
    _id: 1003,
    aId: 2,
    bId: 20
  }
]
```

evidencias/B02_modelagem_NN.png

Parte 2 — Prática (C: Índices e Agregações)

C — Índices (indexar1/indexar2) e TTL

Comandos (resumo)

```
use CB_indeagreg

db.CB_indexar1.createIndex({ campoAsc: 1 })
db.CB_indexar1.createIndex({ campoUnico: -1 }, { unique: true })
db.CB_indexar1.createIndex({ campoA: 1, campoB: 1 })
db.CB_indexar1.createIndex({ tags: 1 })
db.CB_indexar1.createIndex({ opcional: 1 }, { sparse: true })
db.CB_indexar1.createIndex({ expiraEm: 1 }, { expireAfterSeconds: 20 })

db.CB_indexar2.createIndex({ a:"text", b:"text", c:"text", d:"text" })
db.CB_indexar2.find({ $text: { $search: "..." } })
```

Evidências

evidencias/C01_indexar1_indices.png
evidencias/C02_ttl_anter.png
evidencias/C03_ttl_depois.png
evidencias/C04_text_index_busca.png



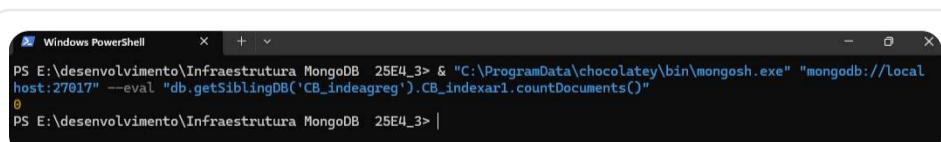
```
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> & "C:\ProgramData\chocolatey\bin\mongosh.exe" --file .\scripts\CB_indexar1_setup_indexes_aggs.js
OK: CB_indexar1 criado + indices configurados.
Índices CB_indexar1:
[
  {
    v: 2,
    key: {
      _id: 1
    },
    name: '_id_'
  },
  {
    v: 2,
    key: {
      codigo: 1
    },
    name: 'codigo_1'
  },
  {
    v: 2,
    key: {
      email: -1
    },
    name: 'email_-1',
    unique: true
  },
  {
    v: 2,
    key: {
      categoria: 1,
      preco: 1
    },
    name: 'categoria_1_preco_1'
  },
  {
    v: 2,
    key: {
      tags: 1
    },
    name: 'tags_1'
  },
  {
    v: 2,
    key: {
      email: 1
    },
    name: 'email_1',
    sparse: true
  },
  {
    v: 2,
    key: {
      expiresAt: 1
    },
    name: 'expiresAt_1',
    expireAfterSeconds: 0
  }
]
OK: CB_indexar2 criado + indice textual.
Índices CB_indexar2:
[
  {
    v: 2,
    key: {
      _id: 1
    },
    name: '_id_'
  },
  {
    v: 2,
    key: {
      _fts: 'text',
      _ftsx: 1
    },
    name: 'titulo_text_resumo_text_autor_text',
    weights: {
      autor: 1,
      resumo: 1,
      titulo: 1
    },
    default_language: 'english',
    language_override: 'language',
    textIndexVersion: 3
  }
]
```

evidencias/C01_indexar1_indices.png



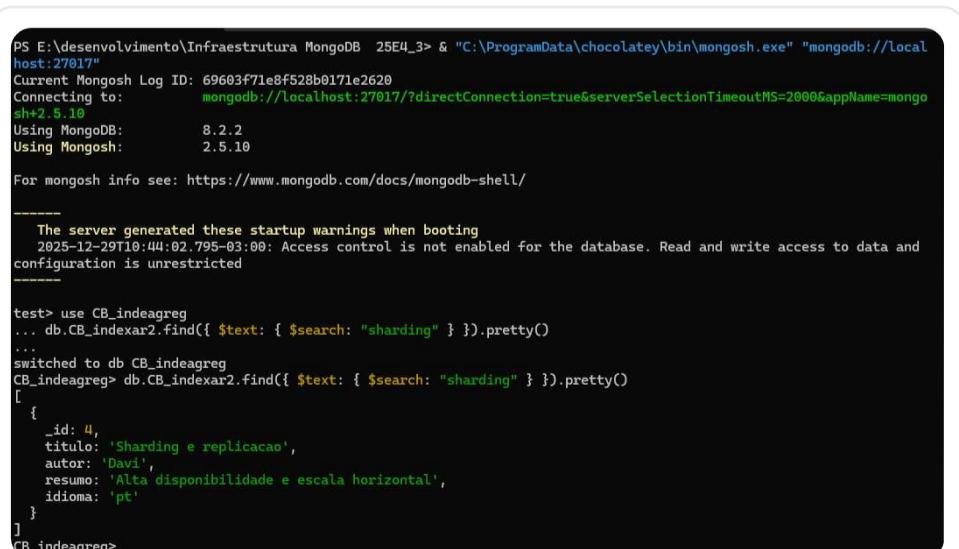
```
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> & "C:\ProgramData\chocolatey\bin\mongosh.exe" "mongodb://localhost:27017" --eval "db.getSiblingDB('CB_indeagreg').CB_indexar1.countDocuments()"
4
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3>
```

evidencias/C02_ttl_anter.png



```
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> & "C:\ProgramData\chocolatey\bin\mongosh.exe" "mongodb://localhost:27017" --eval "db.getSiblingDB('CB_indeagreg').CB_indexar1.countDocuments()"
0
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> |
```

evidencias/C03_ttl_depois.png



```
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> & "C:\ProgramData\chocolatey\bin\mongosh.exe" "mongodb://localhost:27017" --eval "db.getSiblingDB('CB_indeagreg').CB_indexar2.countDocuments()"
0
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> |
```

```
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> & "C:\ProgramData\chocolatey\bin\mongosh.exe" "mongodb://localhost:27017" --eval "db.getSiblingDB('CB_indeagreg').CB_indexar2.find({ $text: { $search: 'sharding' } }).pretty()"
Current Mongosh Log ID: 69603f71e8f528b0171e2620
Connecting to: mongodb://localhost:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongo
sh+2.5.10
Using MongoDB: 8.2.2
Using Mongosh: 2.5.10
For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/
-----
The server generated these startup warnings when booting
2025-12-29T10:44:02.795-03:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----
test> use CB_indeagreg
... db.CB_indexar2.find({ $text: { $search: "sharding" } }).pretty()
...
switched to db CB_indeagreg
CB_indeagreg> db.CB_indexar2.find({ $text: { $search: "sharding" } }).pretty()
[
  {
    _id: 4,
    titulo: 'Sharding e replicacao',
    autor: 'Davi',
    resumo: 'Alta disponibilidade e escala horizontal',
    idioma: 'pt'
  }
]
CB_indeagreg>
```

evidencias/C04_text_index_busca.png

C — Agregações por UF em CB_Venda

Comando (resumo)

```
use CB_indeagreg
db.CB_Venda.aggregate([
  { $group: {
    _id: "$UF_Venda",
    qtd: { $sum: 1 },
    total: { $sum: "$Valor_Venda" },
    media: { $avg: "$Valor_Venda" },
    maior: { $max: "$Valor_Venda" },
    menor: { $min: "$Valor_Venda" }
  }},
  { $sort: { _id: 1 } }
])
])
```

Evidências

evidencias/C05_agregacoes_por_uf.png
evidencias/C07_venda_count_16.png

```
Windows PowerShell
OK: CB_Venda criado com 16 docs.
AGG: count por UF
[
  {
    _id: 'MG',
    count: 4
  },
  {
    _id: 'PR',
    count: 4
  },
  {
    _id: 'RJ',
    count: 4
  },
  {
    _id: 'SP',
    count: 4
  }
]
AGG: total (sum) por UF
[
  {
    _id: 'MG',
    total: 1000
  },
  {
    _id: 'PR',
    total: 1120
  },
  {
    _id: 'RJ',
    total: 1200
  },
  {
    _id: 'SP',
    total: 1240
  }
]
AGG: média (avg) por UF
[
  {
    _id: 'MG',
    media: 250
  },
  {
    _id: 'PR',
    media: 280
  },
  {
    _id: 'RJ',
    media: 300
  },
  {
    _id: 'SP',
    media: 310
  }
]
AGG: max por UF
[
  {
    _id: 'MG',
    max: 400
  },
  {
    _id: 'PR',
    max: 430
  },
  {
    _id: 'RJ',
    max: 450
  },
  {
    _id: 'SP',
    max: 460
  }
]
AGG: min por UF
[
  {
    _id: 'MG',
    min: 100
  },
  {
    _id: 'PR',
    min: 130
  },
  {
    _id: 'RJ',
    min: 150
  },
  {
    _id: 'SP',
    min: 160
  }
]
```

evidencias/C05_agregacoes_por_uf.png

```
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> & "C:\ProgramData\chocolatey\bin\mongosh.exe" "mongodb://localhost:27017" --quiet
>> & "C:\ProgramData\chocolatey\bin\mongosh.exe" "mongodb://localhost:27017" --quiet --eval "db.getSiblingDB('CB_indeagreg').CB_Venda.countDocuments()"
>> & "C:\ProgramData\chocolatey\bin\mongosh.exe" "mongodb://localhost:27017" --quiet --eval "db.getSiblingDB('CB_indeagreg').CB_Venda.findOne()"
>> 16
{
  _id: ObjectId('69614c653e1ebbea9fie2621'),
  Cod_Venda: 1,
  UF_Venda: 'SP',
  Desc_Prod_Vendido: 'Produto_SP_1',
  Valor_Venda: 160
}
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3>
```

evidencias/C07_venda_count_16.png

Parte 2 — Prática (D: Replicação)

D — Replica Set CB_rsposmit (primário, 2 secundários, árbitro)

Comandos (resumo)

```
rs.initiate(...)

rs.add("127.0.0.1:<sec1>")
rs.add("127.0.0.1:<sec2>")
rs.addArb("127.0.0.1:<arb>")

use CB_rsposmit
db.CB_col_filmes.insertMany([ {...}, {...}, {...}, {...}, {...} ])

// nó secundário
rs.slaveOk()
db.CB_col_filmes.find().pretty()
```

Evidências

evidencias/D01_mongod_rs_subindo.png
evidencias/D02_rs_status.png
evidencias/D13_insert_5_filmes_primary.png
evidencias/D14_read_filmes_secondary.png

```
mongosh mongodb://localhost:27020/?directConnection=true&serverSelectionTimeoutMS=2000
PS E:\desenvolvimento\Infraestrutura MongoDB_25E4_3> $ports = 27017,27018,27019,27020
>> foreach($p in $ports){
>> & "C:\ProgramData\chocolatey\bin\mongosh.exe" "mongodb://localhost:$p" --quiet --eval "db.runCommand({ping:1})" | Out-Host
>> Write-Host "OK porta $p"
>>
>> }
ok: 1
OK porta 27017
ok: 1
OK porta 27018
ok: 1
OK porta 27019
ok: 1
OK porta 27020
PS E:\desenvolvimento\Infraestrutura MongoDB_25E4_3>
```

evidencias/D01_mongod_rs_subindo.png

```
s0 [direct: secondary] test> rs.status().members.map(m => { name: m.name, stateStr: m.stateStr, health: m.health })
...
[
  { name: 'localhost:27017', stateStr: 'PRIMARY', health: 1 },
  { name: 'localhost:27018', stateStr: 'SECONDARY', health: 1 },
  { name: 'localhost:27019', stateStr: 'SECONDARY', health: 1 },
  { name: 'localhost:27020', stateStr: 'ARBITER', health: 1 }
]
s0 [direct: primary] test>
```

evidencias/D02_rs_status.png

```
mongosh mongodb://localhost:27017/?directConnection=true&serverSelectionTimeoutMS=2000
CB_rsposmit [direct: primary] CB_rsposmit_db> db.CB_col_filmes.drop()
...
db.CB_col_filmes.insertMany([
  { _id: 1, titulo: 'Matrix', ano: 1999, genero: 'Sci-Fi' },
  { _id: 2, titulo: 'Interestelar', ano: 2014, genero: 'Sci-Fi' },
  { _id: 3, titulo: 'Cidade de Deus', ano: 2002, genero: 'Drama' },
  { _id: 4, titulo: 'O Poderoso Chefão', ano: 1972, genero: 'Crime' },
  { _id: 5, titulo: 'Parasita', ano: 2019, genero: 'Thriller' }
])
...
db.CB_col_filmes.find().sort({_id:1}).pretty()
{
  "_id": 1, "titulo": "Matrix", "ano": 1999, "genero": "Sci-Fi" },
  "_id": 2, "titulo": "Interestelar", "ano": 2014, "genero": "Sci-Fi" },
  "_id": 3, "titulo": "Cidade de Deus", "ano": 2002, "genero": "Drama" },
  "_id": 4, "titulo": "O Poderoso Chefão", "ano": 1972, "genero": "Crime" },
  "_id": 5, "titulo": "Parasita", "ano": 2019, "genero": "Thriller" }
CB_rsposmit [direct: primary] CB_rsposmit_db>
```

evidencias/D13_insert_5_filmes_primary.png

```
mongosh mongodb://localhost:27018/?directConnection=true&serverSelectionTimeoutMS=2000
PS C:\WINDOWS\system32> & "C:\ProgramData\chocolatey\bin\mongosh.exe" "mongodb://localhost:27018"
>> Current Mongosh Log ID: 6961553e26f45776151e2620
Connecting to: mongodb://localhost:27018/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.10
Using MongoDB: 8.2.2
CB_rsposmit [direct: secondary] test> rs.secondaryOk()
...
DeprecationWarning: .setSecondaryOk() is deprecated. Use .setReadPref("primaryPreferred") instead
Setting read preference from "primary" to "primaryPreferred"
CB_rsposmit [direct: secondary] test> use CB_rsposmit_db
...
switched to db CB_rsposmit_db
CB_rsposmit [direct: secondary] CB_rsposmit_db> db.CB_col_filmes.find().sort({_id:1}).pretty()
...
[
  { _id: 1, titulo: 'Matrix', ano: 1999, genero: 'Sci-Fi' },
  { _id: 2, titulo: 'Interestelar', ano: 2014, genero: 'Sci-Fi' },
  { _id: 3, titulo: 'Cidade de Deus', ano: 2002, genero: 'Drama' },
  { _id: 4, titulo: 'O Poderoso Chefão', ano: 1972, genero: 'Crime' },
  { _id: 5, titulo: 'Parasita', ano: 2019, genero: 'Thriller' }
]
CB_rsposmit [direct: secondary] CB_rsposmit_db>
```

evidencias/D14_read_filmes_secondary.png

Parte 2 — Prática (E: Particionamento / Sharding)

E — Sharding (configRS + 4 shards + mongos)

Comandos (resumo)

```
sh.addShard("127.0.0.1:27200")
sh.addShard("127.0.0.1:27201")
sh.addShard("127.0.0.1:27202")
sh.addShard("127.0.0.1:27203")

sh.enableSharding("CB_algumdb")
sh.shardCollection("CB_algumdb.CB_algumacol", { chave: 1 })

for (var i=1; i<=1000; i++){
  db.CB_algumacol.insert({ chave:i, payload:"x" })
}

db.CB_algumacol.getShardDistribution()
```

Evidências

evidencias/E06_mongos_conectado_27050.png
evidencias/E07_add_shards_status.png
evidencias/E08_enable_sharding_db.png
evidencias/E09_shard_collection.png
evidencias/E10_insert_1000_docs.png
evidencias/E11_distribuicao_shards.png

```

mongosh mongodb://localhost:27050/?directConnection=true&serverSelectionTimeoutMS=2000
>>
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> netstat -ano | findstr ":27050"
TCP 127.0.0.1:27050 0.0.0.0: LISTENING 67120
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> & "C:\ProgramData\chocolatey\bin\mongosh.exe" "mongodb://localhost:27050"
Current Mongosh Log ID: 69619875ceb258804e1a2628
Connecting to: mongodb://localhost:27050/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.10
Using Mongos: 8.2.2
Using Mongosh: 2.5.10
For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/
-----
The server generated these startup warnings when booting
2026-01-09T19:23:08.675-03:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2026-01-09T19:23:08.676-03:00: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning
-----
[direct: mongos] test>

```

evidencias/E06_mongos_conectado_27050.png

```

mongosh mongodb://localhost:27050/?directConnection=true&serverSelectionTimeoutMS=2000
[direct: mongos] test> sh.addShard("sh1RS/localhost:27200")
{
  shardAdded: 'sh1RS',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1767997639, i: 4 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAA='),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1767997639, i: 4 })
}
[direct: mongos] test> sh.addShard("sh2RS/localhost:27201")
{
  shardAdded: 'sh2RS',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1767997648, i: 4 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAA='),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1767997648, i: 4 })
}
[direct: mongos] test> sh.addShard("sh3RS/localhost:27202")
{
  shardAdded: 'sh3RS',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1767997657, i: 4 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAA='),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1767997657, i: 4 })
}
[direct: mongos] test> sh.addShard("sh4RS/localhost:27203")
{
  shardAdded: 'sh4RS',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1767997664, i: 5 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAA='),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1767997664, i: 5 })
}
[direct: mongos] test> sh.status()
...
shardingVersion
{
  _id: ObjectId('69617f2da2f371c2d15874a7'),
  clusterId: ObjectId('69617f2da2f371c2d15874a6')
}
...
shards
[
  {
    _id: 'sh1RS',
    host: 'sh1RS/localhost:27200',
    state: 1,
    topologyTime: Timestamp({ t: 1767997611, i: 32 }),
    replSetConfigVersion: Long('-1')
  },
  {
    _id: 'sh2RS',
    host: 'sh2RS/localhost:27201',
    state: 1,
    topologyTime: Timestamp({ t: 1767997611, i: 89 }),
    replSetConfigVersion: Long('-1')
  },
  {
    _id: 'sh3RS',
    host: 'sh3RS/localhost:27202',
    state: 1,
    topologyTime: Timestamp({ t: 1767997612, i: 25 }),
    replSetConfigVersion: Long('-1')
  },
  {
    _id: 'sh4RS',
    host: 'sh4RS/localhost:27203',
    state: 1,
    topologyTime: Timestamp({ t: 1767997612, i: 76 }),
    replSetConfigVersion: Long('-1')
  }
]
...
active mongoses
[ { '8.2.2': 1 } ]
...
autosplit
{ 'Currently enabled': 'yes' }
...
balancer
{
  'Currently enabled': 'yes',
  'Currently running': 'no',
  'Failed balancer rounds in last 5 attempts': 0,
  'Migration Results for the last 24 hours': 'No recent migrations'
}
...
shardedDataDistribution
[]
...
databases
[
  {
    database: { _id: 'config', primary: 'config', partitioned: true },
    collections: {}
  }
]
[direct: mongos] test>

```

evidencias/E07_add_shards_status.png

```
mongosh mongodb://localhost:27050/?directConnection=true&serverSelectionTimeoutMS=2000
[direct: mongos] test> sh.enableSharding("CB_sharded")
...
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1767997716, i: 21 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAA='),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1767997716, i: 21 })
}
[direct: mongos] test>
```

evidencias/E08_enable_sharding_db.png

```
mongosh mongodb://localhost:27050/?directConnection=true&serverSelectionTimeoutMS=2000
[direct: mongos] test> use CB_sharded
switched to db CB_sharded
[direct: mongos] CB_sharded> db.CB_items.createIndex({ cod: 1 })
cod_1
[direct: mongos] CB_sharded> sh.shardCollection("CB_sharded.CB_items", { cod: 1 })
{
  collectionsharded: 'CB_sharded.CB_items',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1767997810, i: 41 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAA='),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1767997810, i: 40 })
}
[direct: mongos] CB_sharded>
```

evidencias/E09_shard_collection.png

```
[direct: mongos] CB_sharded> for (let i = 1; i <= 1000; i++) {
...   db.CB_items.insertOne({ cod: i, nome: "Item_" + i })
... }
{
  acknowledged: true,
  insertedId: ObjectId('696181a5ceb258804e1e2a08')
}
```

evidencias/E10_insert_1000_docs.png

```
[direct: mongos] CB_sharded> db.CB_items.countDocuments()
... db.CB_items.getShardDistribution()
...
Shard sh4RS at sh4RS/localhost:27203
{
  data: '48KiB',
  docs: 1000,
  chunks: 1,
  'estimated data per chunk': '48KiB',
  'estimated docs per chunk': 1000
}
---
Totals
{
  data: '48KiB',
  docs: 1000,
  chunks: 1,
  'Shard sh4RS': [
    '100 % data',
    '100 % docs in cluster',
    '498 avg obj size on shard'
  ]
}
[direct: mongos] CB_sharded>
```

evidencias/E11_distribuicao_shards.png

Parte 2 — Prática (F: Storage Engines e Segurança)

F1 — Instância mmapv1 + CB_produtos (4 docs)

Comandos (resumo)

```
mongod --port 27060 --dbpath <...> --logpath <...> --logappend --storageEngine mmapv1
mongo mongoDB://127.0.0.1:27060/admin --eval "db.serverStatus().storageEngine"
```

```
use CB_rh
db.CB_produtos.insertMany([ {...}, {...}, {...}, {...} ])
```

```
Administrator: Cloud Tools for PowerShell
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> & "E:\desenvolvimento\Infraestrutura MongoDB 25E4_3\tools\mongodb-4.0.28\bin\mongo.exe"
>> "mongodb://127.0.0.1:27060/admin"
>> --eval "printjson(db.serverStatus().storageEngine)"
>>
MongoDB shell version v4.0.28
connecting to: mongodb://127.0.0.1:27060/admin?gssapiServiceName=mongodb
Implicit session: session { "_id" : UUID("e1db9b93-5d5c-4d69-a338-4f5fb456eee") }
MongoDB server version: 4.0.28
{
  "name" : "mmapv1",
  "supportsCommittedReads" : false,
  "supportssnapshotReadConcern" : false,
  "readOnly" : false,
  "persistent" : true
}
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3>
```

evidencias/F01_mmapv1_storage_engine.png

Evidências

evidencias/F01_mmapv1_storage_engine.png
evidencias/F02_mmapv1_produtos_insert_4.png

```
Administrator: Cloud Tools for PowerShell
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> cd "E:\desenvolvimento\Infraestrutura MongoDB 25E4_3"
>> $path = ".\scripts\F_storage\F01_mmapv1_produtos.js"
>> @'
>> var d = db.getSiblingDB("CB_rh");
>> d.CB_produtos.drop();
>> d.CB_produtos.insertMany([
>>   {_id: 1, nome: "Teclado", preco: 120.0, estoque: 10 },
>>   {_id: 2, nome: "Mouse", preco: 60.0, estoque: 25 },
>>   {_id: 3, nome: "Monitor", preco: 900.0, estoque: 5, obs: "promo" },
>>   {_id: 4, nome: "Cabo USB", preco: 25.0 }
>> ]);
>> print("OK: inseridos = " + d.CB_produtos.countDocuments());
>> print(d.CB_produtos.find().sort({_id:1}).toArray());
>> '@ | Set-Content -Encoding UTF8 $path
>>
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> & "E:\desenvolvimento\Infraestrutura MongoDB 25E4_3\tools\mongodb-4.0.28\bin\mongo.exe"
>> "mongodb://127.0.0.1:27060/CB_rh"
>> ".\scripts\F01_mmapv1_produtos.js"
MongoDB shell version v4.0.28
connecting to: mongodb://127.0.0.1:27060/CB_rh?gssapiServiceName=mongodb
Implicit session: session { "_id" : UUID("bf0f37cb-35c7-4275-91b5-9315d32a5780") }
MongoDB server version: 4.0.28
OK: inseridos = 4
[
  {
    "_id" : 1,
    "nome" : "Teclado",
    "preco" : 120,
    "estoque" : 10
  },
  {
    "_id" : 2,
    "nome" : "Mouse",
    "preco" : 60,
    "estoque" : 25
  },
  {
    "_id" : 3,
    "nome" : "Monitor",
    "preco" : 900,
    "estoque" : 5,
    "obs" : "promo"
  },
  {
    "_id" : 4,
    "nome" : "Cabo USB",
    "preco" : 25
  }
]
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3>
```

evidencias/F02_mmapv1_produtos_insert_4.png

F2 — Instância wiredTiger + CB_lugares (4 docs)

Comandos (resumo)

```
mongod --port 27061 --dbpath <...> --logpath <...> --logappend --storageEngine wiredTiger
mongo mongoDB://127.0.0.1:27061/admin --eval "db.serverStatus().storageEngine"
```

```
use CB_rh
db.CB_lugares.insertMany([ {...}, {...}, {...}, {...} ])
```

```
Administrator: Cloud Tools for PowerShell
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> & "E:\desenvolvimento\Infraestrutura MongoDB 25E4_3\tools\mongodb-4.0.28\bin\mongo.exe"
>> "mongodb://127.0.0.1:27061/admin" --quiet --eval "printjson(db.serverStatus().storageEngine)"
>>
{
  "name" : "wiredTiger",
  "supportsCommittedReads" : true,
  "supportssnapshotReadConcern" : true,
  "readOnly" : false,
  "persistent" : true
}
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3>
```

evidencias/F04_wiredTiger_storage_engine.png

Evidências

evidencias/F04_wiredTiger_storage_engine.png
evidencias/F05_wiredTiger_lugares_insert_4.png

```
Administrator: Cloud Tools for PowerShell
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> cd "E:\desenvolvimento\Infraestrutura MongoDB 25E4_3"
>> & "E:\desenvolvimento\Infraestrutura MongoDB 25E4_3\tools\mongodb-4.0.28\bin\mongo.exe"
>> "mongodb://127.0.0.1:27061/CB_rh"
>> ".\scripts\F_storage\F02_wiredTiger_lugares.js"
MongoDB shell version v4.0.28
connecting to: mongodb://127.0.0.1:27061/CB_rh?gssapiServiceName=mongodb
Implicit session: session { "_id" : UUID("id05ba47-6401-4452-bbe6-5c61bc7c2afc") }
MongoDB server version: 4.0.28
OK lugares=4
[
  {
    "_id" : 1,
    "nome" : "Praia",
    "cidade" : "Santos",
    "uf" : "SP"
  },
  {
    "_id" : 2,
    "nome" : "Museu",
    "cidade" : "Rio de Janeiro",
    "uf" : "RJ"
  },
  {
    "_id" : 3,
    "nome" : "Parque",
    "cidade" : "Curitiba",
    "uf" : "PR",
    "obs" : "familia"
  },
  {
    "_id" : 4,
    "nome" : "Cachoeira",
    "cidade" : "Brotas",
    "uf" : "SP"
  }
]
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3>
```

evidencias/F05_wiredTiger_lugares_insert_4.png

F3 — Segurança (auth), usuário dba, usuário CB_desenv, coleção CB_funcionarios

Comandos (resumo)

```
mongod --port 27062 --dbpath <...> --logpath <...> --logappend --auth  
  
mongo mongodb://127.0.0.1:27062/admin --eval "db.createUser({user:'db',  
  
mongo "mongodb://dba:dba123@127.0.0.1:27062/admin" --eval "db.getSibl  
  
mongo "mongodb://CB_desenv:dev123@127.0.0.1:27062/CB_rh" --eval "db.CB_rh
```

Evidências

evidencias/F09_secure_login_dba_connectionStatus.png
evidencias/F11_createUser_CB_desenv.png
evidencias/F12_login_CB_desenv.png
evidencias/F13_insert_funcionarios_4.png

```
Administrator: Cloud Tools for PowerShell
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> & "E:\desenvolvimento\Infraestrutura MongoDB 25E4_3\tools\mongodb-4.0.28\bin\mongo.exe" "mongodb://dba@127.0.0.1:27062/admin?authSource=admin" --quiet --eval "printjson(db.runCommand({connectionStatus:1
}));"
>>>
{
    "authInfo" : {
        "authenticatedUsers" : [
            {
                "user" : "dba",
                "db" : "admin"
            }
        ],
        "authenticatedUserRoles" : [
            {
                "role" : "root",
                "db" : "admin"
            }
        ]
    },
    "ok" : 1
}
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3>
```

evidencias/F09_secure_login_dba_connectionStatus.png

```
Administrador: Cloud Tools for PowerShell
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> & "E:\desenvolvimento\Infraestrutura MongoDB 25E4_3\tools\mongodb-4.0.28\bin\mongo.exe"
>> <"mongodb://127.0.0.1:27062/admin">
>> -u "dba" -p "dba123" --authenticationDatabase "admin" --quiet --eval `
>> "db.getSiblingDB('CB_rh').createUser({user:'CB_desenv',pwd:'devi23',roles:[{role:'readWrite',db:'CB_rh'}]}); print('OK: CB_desenv criado');"
>>
Successfully added user: {
    "user" : "CB_desenv",
    "roles" : [
        {
            "role" : "readWrite",
            "db" : "CB_rh"
        }
    ]
}
OK: CB_desenv criado
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> & "E:\desenvolvimento\Infraestrutura MongoDB 25E4_3\tools\mongodb-4.0.28\bin\mongo.exe"
>> <"mongodb://127.0.0.1:27062/admin">
>> -u "dba" -p "dba123" --authenticationDatabase "admin" --quiet --eval `
>> "var rh=db.getSiblingDB('CB_rh');"
>> "rh.createUser('CB_desenv');"
>> "if(rh) print(' OK: CB_desenv já existe');"
>> "else rh.createUser({user:'CB_desenv',pwd:'devi23',roles:[{role:'readWrite',db:'CB_rh'}]}); print('OK: CB_desenv criado');"
>>
OK: CB_desenv já existe
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3>
```

evidencias/F11_createUser_CB_desenv.png

evidencias/F12_login_CB_desenv.png

```
Administrador: Cloud Tools for PowerShell
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> & "E:\desenvolvimento\Infraestrutura MongoDB 25E4_3\tools\mongodb-4.0.28\bin\mongo.exe"
>> "mongod://127.0.0.1:27062/CB_rh"
>> --u "CB_desenv" -p "dev123" --authenticationDatabase "CB_rh" --quiet --eval `
>> "db.CB_funcionarios.drop();"
>> db.CB_funcionarios.insertMany([
  { _id:1, nome: "Ana", cargo: "Dev", salario: 3500 },
  { _id:2, nome: "Bruno", cargo: "DBA", salario: 5000, obs: "admin" },
  { _id:3, nome: "Carla", cargo: "Analista", salario: 3200 },
  { _id:4, nome: "Diego", cargo: "Estagiario", salario: 1500 }
]);
>> print("OK funcionarios = " + db.CB_funcionarios.count());
>> printjson(db.CB_funcionarios.find().sort({_id:1}).toArray());"
OK funcionarios=4
[ {
    "_id" : 1,
    "nome" : "Ana",
    "cargo" : "Dev",
    "salario" : 3500
},
{
    "_id" : 2,
    "nome" : "Bruno",
    "cargo" : "DBA",
    "salario" : 5000,
    "obs" : "admin"
},
{
    "_id" : 3,
    "nome" : "Carla",
    "cargo" : "Analista",
    "salario" : 3200
},
{
    "_id" : 4,
    "nome" : "Diego",
    "cargo" : "Estagiario",
    "salario" : 1500
}
]
```

evidencias/F13 insert funcionarios 4.pnc

Parte 2 — Prática (G: Depuração, Backup e Restore)

G — Backup/kill/delete/restore/validação (DB exerc4b)

Sequência executada

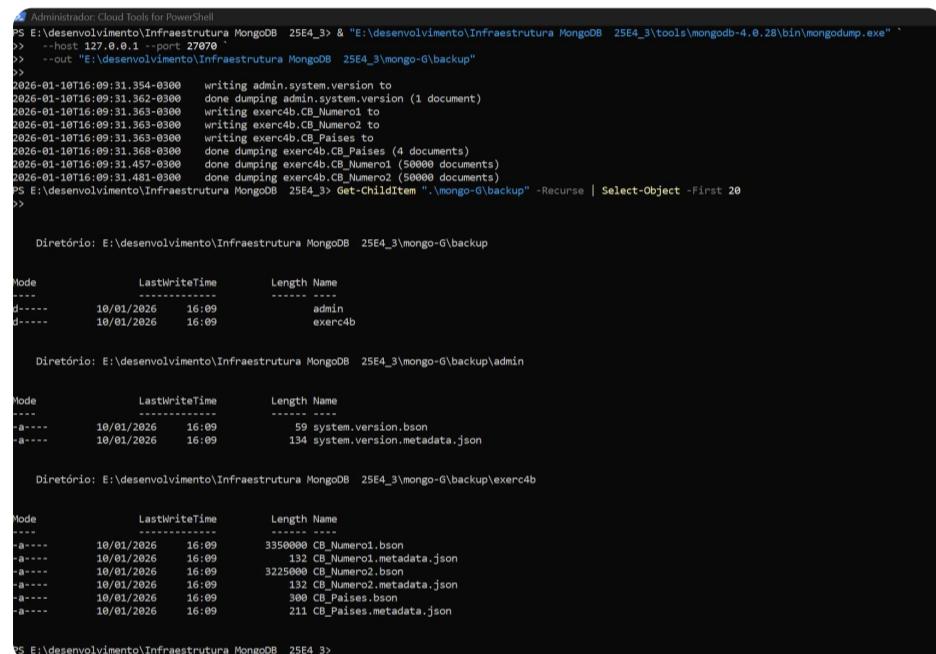
- Backup (mongodump) da instância
- Finalização do processo (taskkill)
- Exclusão do diretório de dados
- Restore (mongorestore)
- Validação das contagens por coleção

Comandos (resumo)

```
mongodump --host 127.0.0.1 --port 27070 --out "E:\...\mongo-G\backup"
taskkill /PID <pid> /F
Remove-Item "E:\...\mongo-G\data\*" -Recurse -Force
mongorestore --host 127.0.0.1 --port 27070 --drop "E:\...\mongo-G\backup"
use exerc4b
db.getCollectionNames().forEach(function(c){
  print(c + " => " + db.getCollection(c).count());
});
```

Evidências exigidas na rubrica

evidencias/G11_backup_mongodump.png
evidencias/G10_taskkill.png
evidencias/G11_apagar_dados.png
evidencias/G12_restore_ok.png
evidencias/G13_contagem_colecoes_exerc4b.png



```
Administrator: Cloud Tools for PowerShell
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> & "E:\desenvolvimento\Infraestrutura MongoDB 25E4_3\tools\mongodb-4.0.28\bin\mongodump.exe" \
>> --host 127.0.0.1 --port 27070
>> --out "E:\desenvolvimento\Infraestrutura MongoDB 25E4_3\mongo-G\backup"
>>
2026-01-10T16:09:31.354-0300  writing admin.system.version to
2026-01-10T16:09:31.362-0300  done dumping admin.system.version (1 document)
2026-01-10T16:09:31.363-0300  writing exerc4b.CB_Numer01 to
2026-01-10T16:09:31.364-0300  writing exerc4b.CB_Numer02 to
2026-01-10T16:09:31.365-0300  writing exerc4b.CB_Paises to
2026-01-10T16:09:31.368-0300  done dumping exerc4b.CB_Paises (4 documents)
2026-01-10T16:09:31.457-0300  done dumping exerc4b.CB_Numer01 (50000 documents)
2026-01-10T16:09:31.481-0300  done dumping exerc4b.CB_Numer02 (50000 documents)
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> Get-ChildItem ".\mongo-G\backup" -Recurse | Select-Object -First 20
>>

Diretório: E:\desenvolvimento\Infraestrutura MongoDB 25E4_3\mongo-G\backup

Mode          LastWriteTime      Length Name
----          -----        ---- 
d----       10/01/2026   16:09                admin
d----       10/01/2026   16:09                exerc4b

Diretório: E:\desenvolvimento\Infraestrutura MongoDB 25E4_3\mongo-G\backup\admin

Mode          LastWriteTime      Length Name
----          -----        ---- 
-a---       10/01/2026   16:09      59 system.version.bson
-a---       10/01/2026   16:09    134 system.version.metadata.json

Diretório: E:\desenvolvimento\Infraestrutura MongoDB 25E4_3\mongo-G\backup\exerc4b

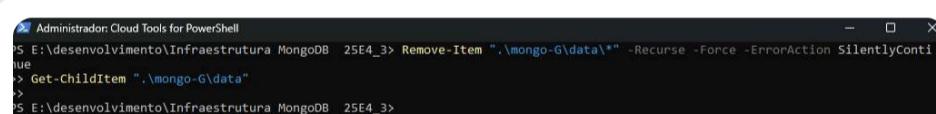
Mode          LastWriteTime      Length Name
----          -----        ---- 
-a---       10/01/2026   16:09  3350000 CB_Numer01.json
-a---       10/01/2026   16:09  3350000 CB_Numer01.metadata.json
-a---       10/01/2026   16:09  3225000 CB_Numer02.json
-a---       10/01/2026   16:09  132 CB_Numer02.metadata.json
-a---       10/01/2026   16:09  300 CB_Paises.json
-a---       10/01/2026   16:09  211 CB_Paises.metadata.json

PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3>
```

evidencias/G11_backup_mongodump.png

G10 taskkill

evidencias/G10_taskkill.png



```
Administrator: Cloud Tools for PowerShell
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> Remove-Item ".\mongo-G\data\*" -Recurse -Force -ErrorAction SilentlyContinue
>>
Get-ChildItem ".\mongo-G\data"
>>
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3>
```

evidencias/G11_apagar_dados.png



```
Administrator: Cloud Tools for PowerShell
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> & "E:\desenvolvimento\Infraestrutura MongoDB 25E4_3\tools\mongodb-4.0.28\bin\mongorestore.exe" \
>> --host 127.0.0.1
>> --port 27070
>> --logpath "E:\desenvolvimento\Infraestrutura MongoDB 25E4_3\mongo-G\logs\mongod_G.log"
>> --logappend
>> --bind_ip 127.0.0.1
>>
2026-01-10T16:33:08.853-0300  preparing collections to restore from
2026-01-10T16:33:08.859-0300  reading metadata for exerc4b.CB_Numer01 from E:\desenvolvimento\Infraestrutura MongoDB 25E4_3\mongo-G\backup\exerc4b.CB_Numer01.metadata.json
2026-01-10T16:33:08.859-0300  reading metadata for exerc4b.CB_Numer02 from E:\desenvolvimento\Infraestrutura MongoDB 25E4_3\mongo-G\backup\exerc4b.CB_Numer02.metadata.json
2026-01-10T16:33:08.865-0300  reading metadata for exerc4b.CB_Paises from E:\desenvolvimento\Infraestrutura MongoDB 25E4_3\mongo-G\backup\exerc4b.CB_Paises.metadata.json
2026-01-10T16:33:08.865-0300  restoring exerc4b.CB_Numer01 from E:\desenvolvimento\Infraestrutura MongoDB 25E4_3\mongo-G\backup\exerc4b.CB_Numer01
2026-01-10T16:33:08.872-0300  restoring exerc4b.CB_Numer02 from E:\desenvolvimento\Infraestrutura MongoDB 25E4_3\mongo-G\backup\exerc4b.CB_Numer02
2026-01-10T16:33:08.877-0300  restoring exerc4b.CB_Paises from E:\desenvolvimento\Infraestrutura MongoDB 25E4_3\mongo-G\backup\exerc4b.CB_Paises
2026-01-10T16:33:08.884-0300  finished restoring exerc4b.CB_Paises (4 documents)
2026-01-10T16:33:09.209-0300  no indexes to restore
2026-01-10T16:33:09.209-0300  finished restoring exerc4b.CB_Numer02 (50000 documents)
2026-01-10T16:33:09.209-0300  finished restoring exerc4b.CB_Numer01 (50000 documents)
2026-01-10T16:33:09.209-0300  done
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3>
```

evidencias/G12_restore_ok.png



```
Administrator: Cloud Tools for PowerShell
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3> & "E:\desenvolvimento\Infraestrutura MongoDB 25E4_3\tools\mongodb-4.0.28\bin\mongo.exe" \
>> "mongod://127.0.0.1:27070/exerc4b" --quiet --eval "db.getCollectionNames().forEach(function(c){ print(c + ' => ' + db.getCollection(c).count()); })"
>>
CB_Numer01 => 50000
CB_Numer02 => 50000
CB_Paises => 4
PS E:\desenvolvimento\Infraestrutura MongoDB 25E4_3>
```

evidencias/G13_contagem_colecoes_exerc4b.png

