

Notas - trabalho prático 2

sexta-feira, 3 de junho de 2022

15:51

Objetivos

Exercitar os conceitos vistos em aulas em relação ao banco de dados NoSQL MongoDB.

A partir de dados da Força Aérea Brasileira sobre a aviação civil Brasileira (CENIPA - Ocorrências Aeronáuticas na Aviação Civil Brasileira) vamos importar algumas informações no MongoDB para executar análises.

Enunciado

A base de dados de ocorrências aeronáuticas é gerenciada pelo Centro de Investigação e Prevenção de Acidentes Aeronáuticos (CENIPA). Constam nesta base de dados as ocorrências aeronáuticas notificadas ao CENIPA nos últimos 10 anos e que ocorreram em solo brasileiro. Dentre as informações disponíveis, estão os dados sobre as aeronaves envolvidas, fatalidades, local, data, horário dos eventos e informações taxonômicas típicas das investigações de acidentes (AIG).

Arquivos com os quais trabalharemos:

- Ocorrencia.csv - Informações sobre as ocorrências.
- Ocorrencia_tipo.csv - Informações sobre o tipo de ocorrência.
- Aeronave.csv - Informações sobre as aeronaves envolvidas nas ocorrências

Fonte: Sistema DÉDALO. Disponível em:

<https://dados.gov.br/dataset/ocorrencias-aeronauticas-da-aviacao-civil-brasileira>

- Criando as collections com as regras:

Ocorrencia

```
db.createCollection("ocorrencia", { validator: { $jsonSchema: { bsonType: "object", properties: { id_ocorrencia: { bsonType: "int", description: "is not required" }, classificacao: { bsonType: "string", description: "is not required" }, cidade: { bsonType: "string", description: "is not required" }, uf: { bsonType: "string", description: "is not required" }, pais: { bsonType: "string", description: "is not required" }, data: { bsonType: "date", description: "is not required" }, num_recomendacoes: { bsonType: "int", description: "is not required" } } } } })
```

Ocorrencia tipo

```
db.createCollection("ocorrencia_tipo", { validator: { $jsonSchema: { bsonType: "object", properties: { id_ocorrencia_t: { bsonType: "int", description: "is not required" }, tipo: { bsonType: "string", description: "is not required" } } } } })
```

Aeronave

```
db.createCollection("aeronave", { validator: { $jsonSchema: { bsonType: "object", required: ["assentos", "ano_fabricacao"], properties: { id_ocorrencia_a: { bsonType: "int", description: "is not required" }, matricula: { bsonType: "string", description: "is not required" }, operador_categoria: { bsonType: "string" }, tipo_veiculo: { bsonType: "string", description: "is not required" }, fabricante: { bsonType: "string", description: "is not required" }, modelo: { bsonType: "string", description: "is not required" }, motor_tipo: { bsonType: "string" }, motor_quantidade: { bsonType: "string" }, assentos: { bsonType: "int", minimum: 0, maximum: 1000, description: "must be an integer in [0, 1000 ] and is required" }, ano_fabricacao: { bsonType: "int", minimum: 1900, maximum: 2030, description: "must be an integer in [ 1900, 2030 ] and is required" }, pais_fabricante: { bsonType: "string" }, registro_segmento: { bsonType: "string" }, voo_origem: { bsonType: "string" }, voo_destino: { bsonType: "string" }, fase_operacao: { bsonType: "string" } } } } })
```

Práticas: execute os comandos das práticas abaixo e anote os resultados, pois você vai precisar deles ao responder às questões objetivas.

1) Antes de fazer a importação novamente na collection Aeronave, você precisou limpar os dados que foram importados anteriormente. Você executou o comando `db.aeronave.remove({})`. Quantos documentos foram removidos?

R: 137

2) Verifique o número de documentos carregados na collection "ocorrencia". Você pode usar a função `count()` ou `db.collection.aggregate` com `{ $sum: 1 }`.

```
desafio> db.ocorrencia.find().count()  
5155
```

Recorte de tela efetuado: 03/06/2022 16:34

3) Verifique o número de documentos carregados na collection "ocorrencia_tipo".

```
desafio> db.ocorrencia_tipo.find().count()  
5244
```

Recorte de tela efetuado: 03/06/2022 16:34

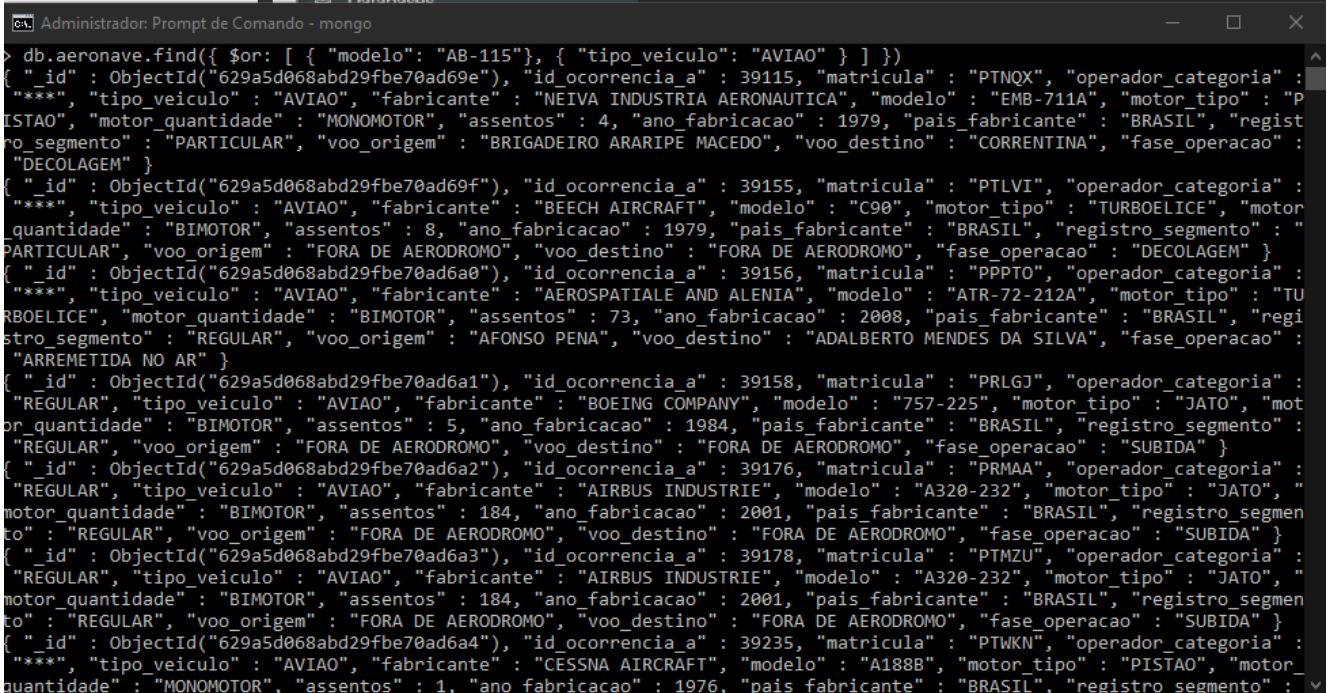
4) Verifique o número de documentos carregados na collection "aeronave"

```
desafio> db.aeronave.find().count()  
5213
```

Recorte de tela efetuado: 03/06/2022 16:35

5) Execute um comando find() na collection aeronave com modelo= 'AB-115' OU tipo_veiculo = 'AVIAO'.

```
db.aeronave.find({ $or: [ { "modelo": "AB-115"}, { "tipo_veiculo": "AVIAO" } ] })
```



The screenshot shows a MongoDB command prompt window titled "Administrador: Prompt de Comando - mongo". The command executed is `db.aeronave.find({ $or: [{ "modelo": "AB-115"}, { "tipo_veiculo": "AVIAO" }] })`. The output displays several JSON documents. The first document is for a "PTNQX" aircraft. The second document is for a "PTLVI" aircraft. The third document is for a "PRLGJ" aircraft. The fourth document is for a "PRMAA" aircraft. The fifth document is for a "PTMZU" aircraft. The sixth document is for an "A320-232" aircraft. The seventh document is for an "A188B" aircraft. The eighth document is for a "CESSNA AIRCRAFT" aircraft.

Recorte de tela efetuado: 03/06/2022 16:49

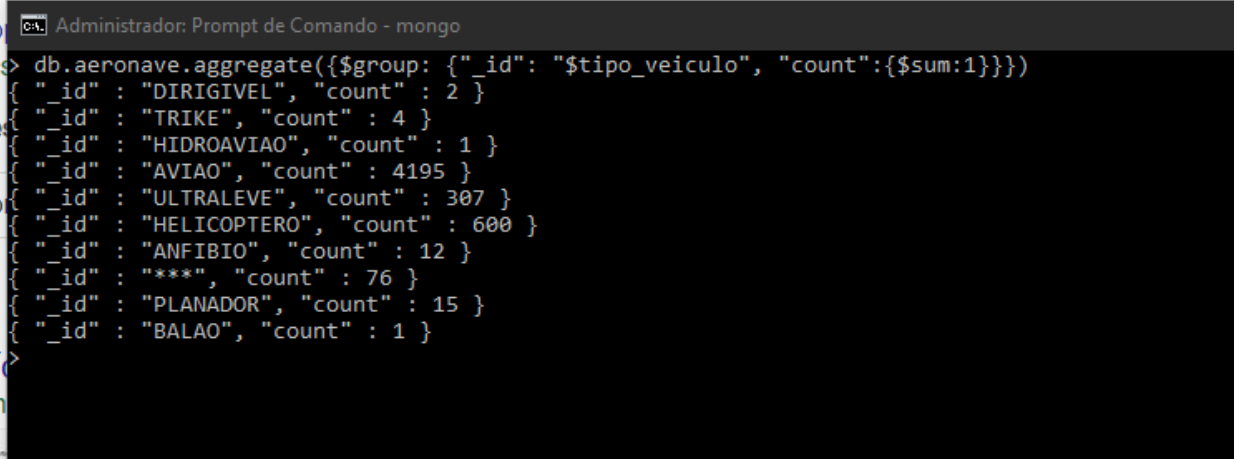
Limite a consulta para trazer apenas os 5 primeiros documentos

```
db.aeronave.find({ $or: [ { "modelo": "AB-115"}, { "tipo_veiculo": "AVIAO" } ] },  
{_id: 0, matricula:1}).limit(5).pretty()
```


Anfibio

7) Execute um comando aggregate() na collection aeronave para agrupar os documentos pelo campo tipo_veiculo fazendo uma contagem (\$sum:1) para cada tipo_veiculo.

```
db.aeronave.aggregate({$group: {"_id": "$tipo_veiculo", "count":{$sum:1}}})
```



```
Administrador: Prompt de Comando - mongo
> db.aeronave.aggregate({$group: {"_id": "$tipo_veiculo", "count":{$sum:1}}})
{"_id" : "DIRIGIVEL", "count" : 2 }
{"_id" : "TRIKE", "count" : 4 }
{"_id" : "HIDROAVIAO", "count" : 1 }
{"_id" : "AVIAO", "count" : 4195 }
{"_id" : "ULTRALEVE", "count" : 307 }
{"_id" : "HELICOPTERO", "count" : 600 }
{"_id" : "ANFIBIO", "count" : 12 }
{"_id" : "****", "count" : 76 }
{"_id" : "PLANADOR", "count" : 15 }
{"_id" : "BALAO", "count" : 1 }
```

Recorte de tela efetuado: 03/06/2022 18:01

(? Refazer)

8) Execute um comando find() na collection ocorrencia para buscar os documentos com o field num_recomendacoes menor ou igual (\$lte) a 5 (<=5). Limite o resultado da sua consulta em 10 documentos. Anote o maior valor para a recomendação encontrada

```
db.ocorrencia.find({num_recomendacoes: {$lte: 5}}).sort({num_recomendacoes:-1}).limit(10).pretty()
```

```
Administrador: Prompt de Comando - mongo
db.ocorrencia.find({num_recomendacoes: {$lte: 5}}).limit(10).sort({num_recomendacoes:-1}).pretty()

  "_id" : ObjectId("629a5d8aabd29fbe70aff87"),
  "id_ocorrencia" : 39318,
  "classificacao" : "ACIDENTE",
  "cidade" : "SENADOR JOSE PORFIRIO",
  "uf" : "PA",
  "pais" : "BRASIL",
  "data" : ISODate("2010-01-25T16:20:00Z"),
  "num_recomendacoes" : 5

  "_id" : ObjectId("629a5d8aabd29fbe70b01bd"),
  "id_ocorrencia" : 43778,
  "classificacao" : "ACIDENTE",
  "cidade" : "CAMPO GRANDE",
  "uf" : "MS",
  "pais" : "BRASIL",
  "data" : ISODate("2011-01-28T20:50:00Z"),
  "num_recomendacoes" : 5

  "_id" : ObjectId("629a5d8aabd29fbe70b0234"),
  "id_ocorrencia" : 44248,
  "classificacao" : "ACIDENTE",
  "cidade" : "SANTO ANDRE",
  "uf" : "SP",
  "pais" : "BRASIL",
  "data" : ISODate("2011-04-08T20:10:00Z"),
  "num_recomendacoes" : 5

  "_id" : ObjectId("629a5d8aabd29fbe70b0016"),
  "id_ocorrencia" : 40732,
  "classificacao" : "INCIDENTE GRAVE",
  "cidade" : "CAMACARI",
  "uf" : "BA",
  "pais" : "BRASIL",
  "data" : ISODate("2010-05-08T15:05:00Z"),
  "num_recomendacoes" : 5

  "_id" : ObjectId("629a5d8aabd29fbe70b006e"),
  "id_ocorrencia" : 41834,
  "classificacao" : "INCIDENTE GRAVE",
  "cidade" : "PONTA GROSSA",
  "uf" : "PR",
  "pais" : "BRASIL",
  "data" : ISODate("2010-07-10T20:35:00Z"),
  "num_recomendacoes" : 5

  "_id" : ObjectId("629a5d8aabd29fbe70b01e6"),
  "id_ocorrencia" : 43908,
  "classificacao" : "ACIDENTE",
  "cidade" : "PEJUCARA",
  "uf" : "RS",
  "pais" : "BRASIL",
  "data" : ISODate("2011-02-15T13:10:00Z"),
  "num_recomendacoes" : 5
```

Recorte de tela efetuado: 04/06/2022 10:25

```
Administrador: Prompt de Comando - mongo
{
{
  "_id" : ObjectId("629a5d8a8abd29fbe70b02ac"),
  "id_ocorrencia" : 44652,
  "classificacao" : "INCIDENTE GRAVE",
  "cidade" : "NOVO HAMBURGO",
  "uf" : "RS",
  "pais" : "BRASIL",
  "data" : ISODate("2011-06-19T13:00:00Z"),
  "num_recomendacoes" : 5
}
{
  "_id" : ObjectId("629a5d8a8abd29fbe70b02a9"),
  "id_ocorrencia" : 44647,
  "classificacao" : "ACIDENTE",
  "cidade" : "SALVADOR",
  "uf" : "BA",
  "pais" : "BRASIL",
  "data" : ISODate("2011-06-18T19:07:00Z"),
  "num_recomendacoes" : 5
}
{
  "_id" : ObjectId("629a5d8a8abd29fbe70b016c"),
  "id_ocorrencia" : 43472,
  "classificacao" : "ACIDENTE",
  "cidade" : "PICOS",
  "uf" : "PI",
  "pais" : "BRASIL",
  "data" : ISODate("2010-12-13T12:30:00Z"),
  "num_recomendacoes" : 5
}
{
  "_id" : ObjectId("629a5d8a8abd29fbe70b002c"),
  "id_ocorrencia" : 41155,
  "classificacao" : "ACIDENTE",
  "cidade" : "JUNDIAI",
  "uf" : "SP",
  "pais" : "BRASIL",
  "data" : ISODate("2010-05-20T19:35:00Z"),
  "num_recomendacoes" : 5
}
>
```

Recorte de tela efetuado: 04/06/2022 10:26

. Pela PRÁTICA 8, executamos o comando find() na collection ocorrencia para buscar os documentos com o campo num_recomendacoes menor ou igual (\$lte) a 5 (<=5). Considerando os 10 primeiros documentos retornados, qual foi o valor MÁXIMO que apareceu para os números de recomendações (num_recomendacoes)?

9) Execute um comando aggregate() na collection ocorrencia para buscar os documentos conforme abaixo.

- Field num_recomendacoes menor ou igual (\$lte) a 5 (≤ 5).
- Agrupar pelo field uf (\$uf).
- Fazer a contagem (\$sum:1).
- Ordenar de forma descendente

```
db.ocorrencia.aggregate([{$match: {num_recomendacoes: {$lte:5}}}]])
```

R:

```
db.ocorrencia.aggregate([
  { $match: { num_recomendacoes: {$lte: 5 } } },
  { $group: { _id: "$uf", count: { $sum:1 } } },
  { $sort: { count: -1 } }
])
```



```

> db.ocorrencia.aggregate([
...     { $match: { num_recomendacoes: { $lte: 5 } } },
...     { $group: { _id: "$uf", count: { $sum: 1 } } },
...     { $sort: { count: -1 } }
... ])
{ "_id" : "SP", "count" : 1211 }
{ "_id" : "MG", "count" : 474 }
{ "_id" : "RJ", "count" : 469 }
{ "_id" : "PR", "count" : 423 }
{ "_id" : "RS", "count" : 325 }
{ "_id" : "GO", "count" : 278 }
{ "_id" : "PA", "count" : 252 }
{ "_id" : "MT", "count" : 248 }
{ "_id" : "AM", "count" : 209 }
{ "_id" : "BA", "count" : 196 }
{ "_id" : "SC", "count" : 160 }
{ "_id" : "MS", "count" : 131 }
{ "_id" : "DF", "count" : 124 }
{ "_id" : "PE", "count" : 85 }
{ "_id" : "CE", "count" : 76 }
{ "_id" : "MA", "count" : 67 }
{ "_id" : "ES", "count" : 66 }
{ "_id" : "AC", "count" : 51 }
{ "_id" : "RR", "count" : 49 }
{ "_id" : "TO", "count" : 46 }
Type "it" for more
> it
{ "_id" : "PI", "count" : 34 }
{ "_id" : "RO", "count" : 27 }
{ "_id" : "AL", "count" : 27 }
{ "_id" : "PB", "count" : 24 }
{ "_id" : "SE", "count" : 20 }
{ "_id" : "RN", "count" : 14 }
{ "_id" : "AP", "count" : 9 }
{ "_id" : "****", "count" : 2 }
>

```

Recorte de tela efetuado: 04/06/2022 11:27

10) Execute um comando aggregate() na collection ocorrencia para buscar a média dos números de recomendações (\$num_recomendacoes)

```
db.ocorrencia.aggregate({$group: {'_id': null, 'media':
{$avg:$num_recomendacoes}}})
```

```

C:\> Administrador: Prompt de Comando - mongo
> db.ocorrencia.aggregate({$group: {'_id': null, 'media': {$avg:$num_recomendacoes}}})
{ "_id" : null, "media" : 0.3258971871968962 }
>

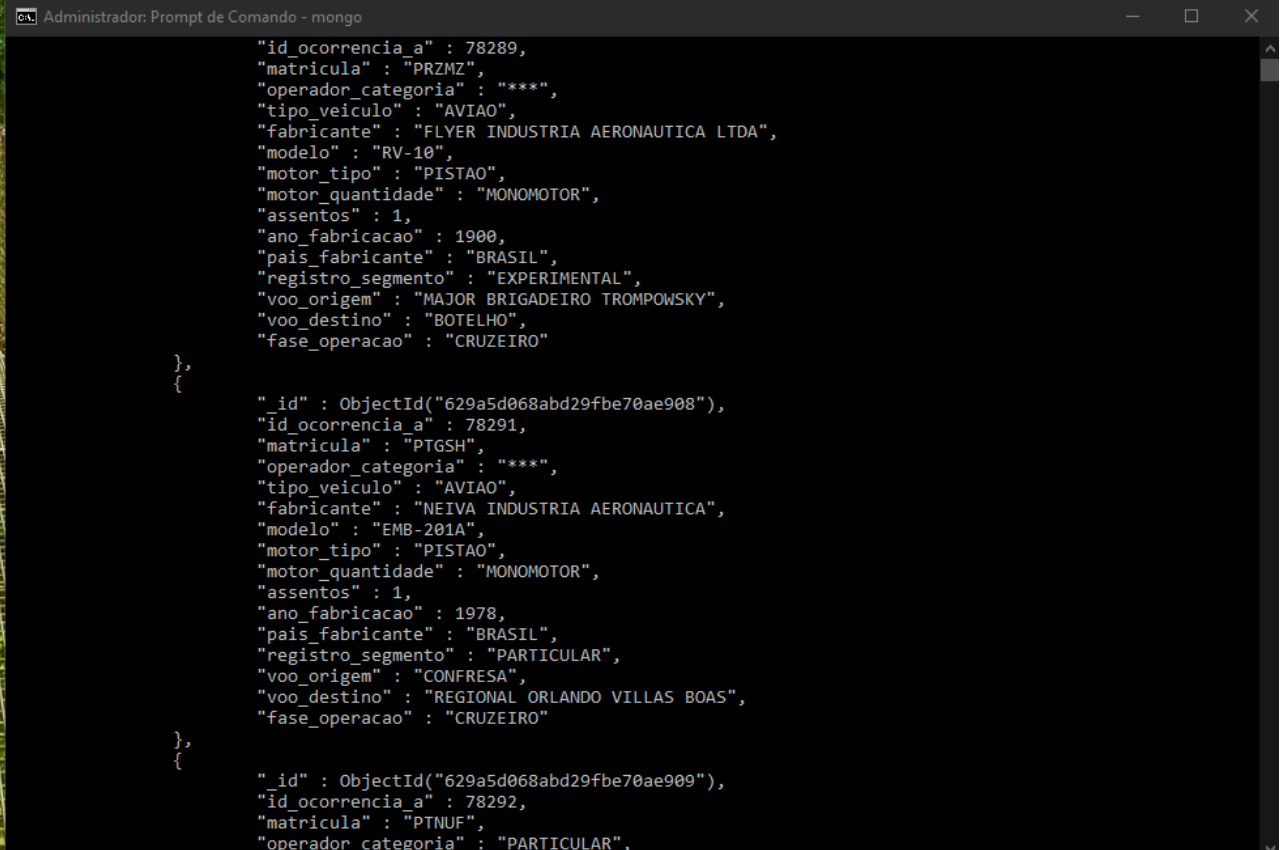
```

11) Execute um comando lookup aggregate() na collection aeronave fazendo uma junção com a collection ocorrencia. Limite o resultado do lookup aggregate() em dois documentos para facilitar a visão do que acontece.

Ocorrencia x aeronave

```
db.aeronave.aggregate([
  { $lookup:
    {
      from: "aeronave",
      localField: ("id_ocorrencia_a "),
      foreignField: ("id_ocorrencia "),
      as: "ocorrencia_aeronave"
    }
  },
  {$limit: 2}
]).pretty()
```

Um dentro do outro



```
Administrator: Prompt de Comando - mongo
{
  "_id" : ObjectId("629a5d068abd29f70ae908"),
  "id_ocorrencia_a" : 78289,
  "matricula" : "PRZMZ",
  "operador_categoria" : "****",
  "tipo_veiculo" : "AVIAO",
  "fabricante" : "FLYER INDUSTRIA AERONAUTICA LTDA",
  "modelo" : "RV-10",
  "motor_tipo" : "PISTAO",
  "motor_quantidade" : "MONOMOTOR",
  "assentos" : 1,
  "ano_fabricacao" : 1900,
  "pais_fabricante" : "BRASIL",
  "registro_segmento" : "EXPERIMENTAL",
  "voo_origem" : "MAJOR BRIGADEIRO TROMPOWSKY",
  "voo_destino" : "BOTELHO",
  "fase_operacao" : "CRUZEIRO"
},
{
  "_id" : ObjectId("629a5d068abd29f70ae909"),
  "id_ocorrencia_a" : 78291,
  "matricula" : "PTGSH",
  "operador_categoria" : "****",
  "tipo_veiculo" : "AVIAO",
  "fabricante" : "NEIVA INDUSTRIA AERONAUTICA",
  "modelo" : "EMB-201A",
  "motor_tipo" : "PISTAO",
  "motor_quantidade" : "MONOMOTOR",
  "assentos" : 1,
  "ano_fabricacao" : 1978,
  "pais_fabricante" : "BRASIL",
  "registro_segmento" : "PARTICULAR",
  "voo_origem" : "CONFRESA",
  "voo_destino" : "REGIONAL ORLANDO VILLAS BOAS",
  "fase_operacao" : "CRUZEIRO"
},
{
  "_id" : ObjectId("629a5d068abd29f70ae909"),
  "id_ocorrencia_a" : 78292,
  "matricula" : "PTNUF",
  "operador_categoria" : "PARTICULAR",
}
```

12) Execute um comando lookup aggregate() na collection ocorrencia fazendo uma junção com a collection ocorrencia_tipo. Limite o resultado do lookup aggregate() em dois documentos para facilitar a visão do que acontece.

```
db.aeronave.aggregate([
  {
    $lookup:
    {
      from: "ocorrencia",
      localField: ("id_ocorrencia_t"),
      foreignField: ("id_ocorrencia"),
      as: "ocorrencia_octipo"
    }
  },
  {
    $limit: 2
  }
]).pretty()
```

```
CA. Administrador: Prompt de Comando - mongo
>
> db.aeronave.aggregate([
... {
...   $lookup:
...   {
...     from: "ocorrencia",
...     localField: ("id_ocorrencia_t"),
...     foreignField: ("id_ocorrencia"),
...     as: "ocorrencia_octipo"
...   }
... },
... {
...   $limit: 2
... }
... ]).pretty()
{
  "_id" : ObjectId("629a5d068abd29fbe70ad69e"),
  "id_ocorrencia_a" : 39115,
  "matricula" : "PTNQX",
  "operador_categoria" : "****",
  "tipo_veiculo" : "AVIAO",
  "fabricante" : "NEIVA INDUSTRIA AERONAUTICA",
  "modelo" : "EMB-711A",
  "motor_tipo" : "PISTAO",
  "motor_quantidade" : "MONOMOTOR",
  "assentos" : 4,
  "ano_fabricacao" : 1979,
  "pais_fabricante" : "BRASIL",
  "registro_segmento" : "PARTICULAR",
  "voo_origem" : "BRIGADEIRO ARARIPE MACEDO",
  "voo_destino" : "CORRENTINA",
  "fase_operacao" : "DECOLAGEM",
  "ocorrencia_octipo" : [ ]
}
{
  "_id" : ObjectId("629a5d068abd29fbe70ad69f"),
  "id_ocorrencia_a" : 39155,
  "matricula" : "PTLVI",
  "operador_categoria" : "****",
  "tipo_veiculo" : "AVIAO",
  "fabricante" : "BEECH AIRCRAFT",
  "modelo" : "C90",
  "motor_tipo" : "TURBOELICE",
  "motor_quantidade" : "BIMOTOR",
  "assentos" : 8,
  "ano_fabricacao" : 1979,
  "pais_fabricante" : "BRASIL",
  "registro_segmento" : "PARTICULAR",
  "voo_origem" : "FORA DE AERODROMO",
  "voo_destino" : "FORA DE AERODROMO",
  "fase_operacao" : "DECOLAGEM",
  "ocorrencia_octipo" : [ ]
}
```

13) Vamos analisar a criação de índices na collection ocorrencia_tipo. Execute os comandos abaixo e analise os resultados.

```
db.aeronave.find({voo_destino: "SERRA DO SOL"}).explain();
```

Administrador: Prompt de Comando - mongo

```
> db.aeronave.find({voo_destino: "SERRA DO SOL"}).explain();
{
  "explainVersion" : "1",
  "queryPlanner" : {
    "namespace" : "desafio.aeronave",
    "indexFilterSet" : false,
    "parsedQuery" : {
      "voo_destino" : {
        "$eq" : "SERRA DO SOL"
      }
    },
    "queryHash" : "E1F326EB",
    "planCacheKey" : "7E0A3A4D",
    "maxIndexedOrSolutionsReached" : false,
    "maxIndexedAndSolutionsReached" : false,
    "maxScansToExplodeReached" : false,
    "winningPlan" : {
      "stage" : "COLLSCAN",
      "filter" : {
        "voo_destino" : {
          "$eq" : "SERRA DO SOL"
        }
      }
    },
    "direction" : "forward"
  },
  "rejectedPlans" : [ ]
},
  "command" : {
    "find" : "aeronave",
    "filter" : {
      "voo_destino" : "SERRA DO SOL"
    },
    "$db" : "desafio"
  },
  "serverInfo" : {
    "host" : "DESKTOP-1E02HP0",
    "port" : 27017,
    "version" : "5.0.8",
    "gitVersion" : "c87e1c23421bf79614baf500fda6622bd90f674e"
  },
  "serverParameters" : {
    "internalQueryFacetBufferSizeBytes" : 104857600,
    "internalQueryFacetMaxOutputDocSizeBytes" : 104857600,
    "internalLookupStageIntermediateDocumentMaxSizeBytes" : 104857600,
    "internalDocumentSourceGroupMaxMemoryBytes" : 104857600,
    "internalQueryMaxBlockingSortMemoryUsageBytes" : 104857600,
    "internalQueryProhibitBlockingMergeOnMongoS" : 0,
    "internalQueryMaxAddToSetBytes" : 104857600,
    "internalDocumentSourceSetWindowFieldsMaxMemoryBytes" : 104857600
  },
  "ok" : 1
}
```

Recorte de tela efetuado: 05/06/2022 12:18

```
db.aeronave.find({voo_destino: "SERRA DO  
SOL"}).explain(true).executionStats;
```

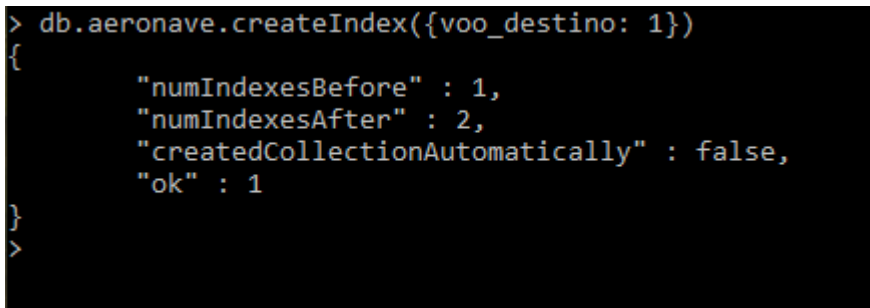


A screenshot of a Windows command prompt window titled "Administrador: Prompt de Comando - mongo". The prompt shows the execution of the command `db.aeronave.find({voo_destino: "SERRA DO SOL"}).explain(true).executionStats;`. The output is a JSON object detailing the execution statistics of the query.

```
> db.aeronave.find({voo_destino: "SERRA DO SOL"}).explain(true).executionStats;
{
  "executionSuccess" : true,
  "nReturned" : 1,
  "executionTimeMillis" : 3,
  "totalKeysExamined" : 0,
  "totalDocsExamined" : 5213,
  "executionStages" : {
    "stage" : "COLLSCAN",
    "filter" : {
      "voo_destino" : {
        "$eq" : "SERRA DO SOL"
      }
    },
    "nReturned" : 1,
    "executionTimeMillisEstimate" : 0,
    "works" : 5215,
    "advanced" : 1,
    "needTime" : 5213,
    "needYield" : 0,
    "saveState" : 5,
    "restoreState" : 5,
    "isEOF" : 1,
    "direction" : "forward",
    "docsExamined" : 5213
  },
  "allPlansExecution" : [ ]
}
```

Recorte de tela efetuado: 05/06/2022 12:19

```
db.aeronave.createIndex({voo_destino: 1})
```



A screenshot of a MongoDB command prompt showing the result of the `db.aeronave.createIndex({voo_destino: 1})` command. The output is a JSON object indicating the successful creation of the index.

```
> db.aeronave.createIndex({voo_destino: 1})
{
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "createdCollectionAutomatically" : false,
  "ok" : 1
}
```

Recorte de tela efetuado: 05/06/2022 12:22

```
db.aeronave.find({voo_destino: "SERRA DO SOL"}).explain();
```

```
Administrador: Prompt de Comando - mongo
> db.aeronave.find({voo_destino: "SERRA DO SOL"}).explain();
{
  "explainVersion" : "1",
  "queryPlanner" : {
    "namespace" : "desafio.aeronave",
    "indexFilterSet" : false,
    "parsedQuery" : {
      "voo_destino" : {
        "$eq" : "SERRA DO SOL"
      }
    },
    "queryHash" : "E1F326EB",
    "planCacheKey" : "ADBC1F70",
    "maxIndexedOrSolutionsReached" : false,
    "maxIndexedAndSolutionsReached" : false,
    "maxScansToExplodeReached" : false,
    "winningPlan" : {
      "stage" : "FETCH",
      "inputStage" : {
        "stage" : "IXSCAN",
        "keyPattern" : {
          "voo_destino" : 1
        },
        "indexName" : "voo_destino_1",
        "isMultiKey" : false,
        "multiKeyPaths" : {
          "voo_destino" : [ ]
        },
        "isUnique" : false,
        "isSparse" : false,
        "isPartial" : false,
        "indexVersion" : 2,
        "direction" : "forward",
        "indexBounds" : {
          "voo_destino" : [
            ["SERRA DO SOL", "SERRA DO SOL"]
          ]
        }
      },
      "rejectedPlans" : [ ]
    },
    "command" : {
      "find" : "aeronave",
      "filter" : {
        "voo_destino" : "SERRA DO SOL"
      },
      "$db" : "desafio"
    },
    "serverInfo" : {
      "host" : "DESKTOP-1E02HP0",
      "port" : 27017,
      "version" : "5.0.8",
      "gitVersion" : "c87e1c23421bf79614baf500fda6622bd90f674e"
    },
    "serverParameters" : {
      "internalQueryFacetBufferSizeBytes" : 104857600,
      "internalQueryFacetMaxOutputDocSizeBytes" : 104857600,
      "internalLookupStageIntermediateDocumentMaxSizeBytes" : 104857600,
      "internalDocumentSourceGroupMaxMemoryBytes" : 104857600,
      "internalQueryMaxBlockingSortMemoryUsageBytes" : 104857600,
      "internalQueryProhibitBlockingMergeOnMongoS" : 0,
      "internalQueryMaxAddToSetBytes" : 104857600,
    }
  }
}
```

Recorte de tela efetuado: 05/06/2022 12:22

```
db.aeronave.find({voo_destino: "SERRA DO SOL"}).explain(true).executionStats;
```

```

CA Administrador: Prompt de Comando - mongo
> db.aeronave.find({voo_destino: "SERRA DO SOL"}).explain(true).executionStats;
{
  "executionSuccess" : true,
  "nReturned" : 1,
  "executionTimeMillis" : 1,
  "totalKeysExamined" : 1,
  "totalDocsExamined" : 1,
  "executionStages" : {
    "stage" : "FETCH",
    "nReturned" : 1,
    "executionTimeMillisEstimate" : 0,
    "works" : 2,
    "advanced" : 1,
    "needTime" : 0,
    "needYield" : 0,
    "saveState" : 0,
    "restoreState" : 0,
    "isEOF" : 1,
    "docsExamined" : 1,
    "alreadyHasObj" : 0,
    "inputStage" : {
      "stage" : "IXSCAN",
      "nReturned" : 1,
      "executionTimeMillisEstimate" : 0,
      "works" : 2,
      "advanced" : 1,
      "needTime" : 0,
      "needYield" : 0,
      "saveState" : 0,
      "restoreState" : 0,
      "isEOF" : 1,
      "keyPattern" : {
        "voo_destino" : 1
      },
      "indexName" : "voo_destino_1",
      "isMultiKey" : false,
      "multiKeyPaths" : {
        "voo_destino" : [ ]
      },
      "isUnique" : false,
      "isSparse" : false,
      "isPartial" : false,
      "indexVersion" : 2,
      "direction" : "forward",
      "indexBounds" : {
        "voo_destino" : [
          ["SERRA DO SOL", "SERRA DO SOL"]
        ]
      },
      "keysExamined" : 1,
      "seeks" : 1,
      "dupsTested" : 0,
      "dupsDropped" : 0
    }
  },
  "allPlansExecution" : [ ]
}
>

```

Se quiser remover o índice é só executar o comando abaixo.

```
db.aeronave.dropIndex({voo_destino: 1})
```