

Portfólio — Bancos de Dados Não Relacionais

Este relatório tem como objetivo apresentar as atividades realizadas como parte dos requisitos para a avaliação parcial do semestre, aplicando conceitos de **Bancos de Dados Não Relacionais**. O foco principal é a criação e manipulação de um banco de dados não relacional no **MongoDB Community Server**, utilizando o **MongoDB Compass** para inserir e atualizar documentos em uma collection.

Informações Adicionais

Nome: Raphael Henrique Vieira Coelho
Matrícula: 3481350205
Curso: Inteligência de Mercado e Análise de Dados
Instituição: Faculdade Anhanguera
Período: 5º Semestre / 2025
E-mail: raphael.phael@gmail.com

Atividades Propostas

- Criar um banco de dados no MongoDB Compass, inserir e atualizar documentos em uma collection.
- Realizar pesquisas e consultas em um banco de dados não relacional (MongoDB).
- Elaborar um relatório com as atividades realizadas, incluindo prints das telas e explicações sobre os processos executados, incluindo: Introdução, Metodologia, Resultados e Conclusão.

Checklist

- Acessar o **MongoDB Compass**
- Criar um banco de dados no **MongoDB**
- Criar uma collection em um banco de dados
- Inserir documentos na collection criada
- Atualizar a collection **vendas** no banco de dados **lojadb** de acordo com as informações das tabelas.
- Navegar até a collection **vendas** do banco de dados **lojadb**
- Realizar as 5 consultas especificadas.

1. Introdução

Este relatório detalha a prática realizada com o **MongoDB**, um banco de dados **NoSQL** orientado a documentos. O objetivo foi compreender os fundamentos de bancos não relacionais, utilizando inserções, atualizações e consultas sobre dados de clientes e suas compras.

Diferentemente dos bancos de dados relacionais, os bancos NoSQL, como o MongoDB, oferecem flexibilidade de estrutura, escalabilidade horizontal e melhor desempenho para dados semiestruturados. São amplamente utilizados em aplicações modernas, como e-commerce, redes sociais e sistemas de recomendação.

2. Métodos

2.1 Configuração do Ambiente e Ferramentas

As seguintes ferramentas foram utilizadas:

- **MongoDB Community Server:** O banco de dados NoSQL orientado a documentos.
- **MongoDB Compass:** Interface gráfica para interagir com o banco de dados **MongoDB**.
- **VS Code:** Editor de texto para documentação.

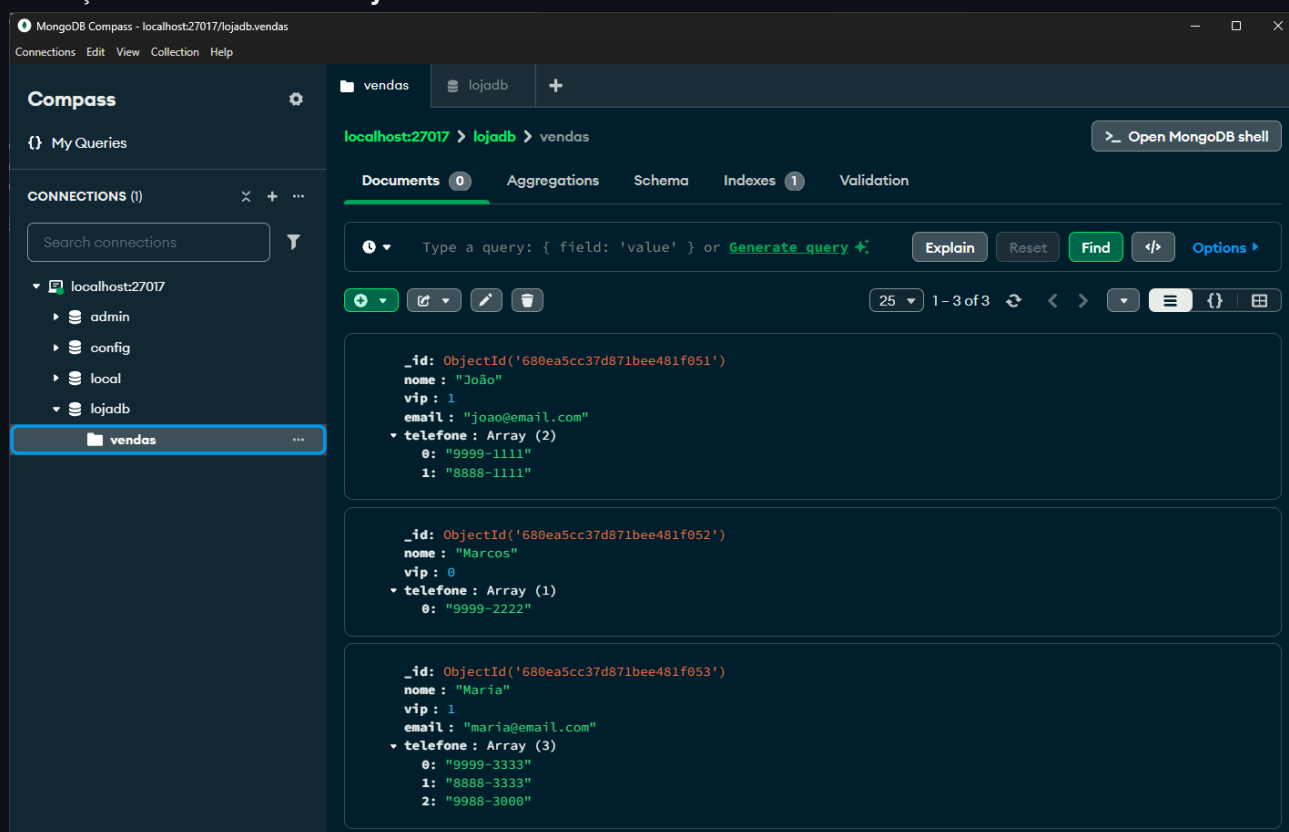
2.2 Aquisição e Preparação dos Dados (Procedimento/Atividade Nº 1)

A atividade consistiu em criar um banco de dados **lojadb** para simular o contexto de uma loja, registrando informações de clientes e suas vendas. Estou utilizando o **Shell** para executar os comandos.

Etapa 1: Criação do Banco e Collection e Inserção Inicial

1. Criação do banco de dados **lojadb** com a collection **vendas**.
2. Inserção dos dados básicos dos clientes na collection **vendas**, onde o campo **telefone** é um array.

a. Criação do banco de dados **lojadb** e da collection **vendas**.



a. Inserção dos dados iniciais dos clientes.

```
// db-vendas-insertMany
db.vendas.insertMany([
  {
    "nome": "João",
    "vip": 1,
    "email": "joao@email.com",
    "telefone": ["9999-1111", "8888-1111"]
  },
  {
    "nome": "Marcos",
    "vip": 0,
    "telefone": ["9999-2222"]
  },
  {
    "nome": "Maria",
    "vip": 1,
    "email": "maria@email.com",
    "telefone": ["9999-3333", "8888-3333", "9988-3000"]
  }
])
```

Etapa 2: Atualização dos Documentos

1. Atualização com o campo **endereço** (objeto aninhado: `rua`, `numero`, `complemento`, `cidade`, `estado`).

a. Atualização do endereço de João.

```
_id: ObjectId('680eb6f675543da9c8fb1503')
nome: "João"
vip: 1
email: "joao@email.com"
▼ telefone: Array (2)
  0: "9999-1111"
  1: "8888-1111"
▼ endereco: Object
  rua: "Rua Um"
  numero: 1000
  complemento: "Apto 1 Bloco 1"
  cidade: "São Paulo"
  estado: "SP"
```

```
// db-vendas-update-joao-endereco
db.vendas.updateOne(
  { "nome": "João" },
  {
    $set: {
      "endereco": {
        "rua": "Rua Um",
        "numero": 1000,
        "complemento": "Apto 1 Bloco 1",
        "cidade": "São Paulo",
        "estado": "SP"
      }
    }
  }
)
```

```
}  
)
```

a. Atualização do endereço de Marcos.

```
_id: ObjectId('680eb6f675543da9c8fb1504')  
nome : "Marcos"  
vip : 0  
▼ telefone : Array (1)  
  0: "9999-2222"  
▼ endereco : Object  
  rua : "Rua Dois"  
  numero : 4000  
  cidade : "Campinas"  
  estado : "SP"
```

```
// db-vendas-update-marcos-endereco  
db.vendas.updateOne(  
  { "nome": "Marcos" },  
  {  
    $set: {  
      "endereco": {  
        "rua": "Rua Dois",  
        "numero": 4000,  
        "cidade": "Campinas",  
        "estado": "SP"  
      }  
    }  
  }  
)
```

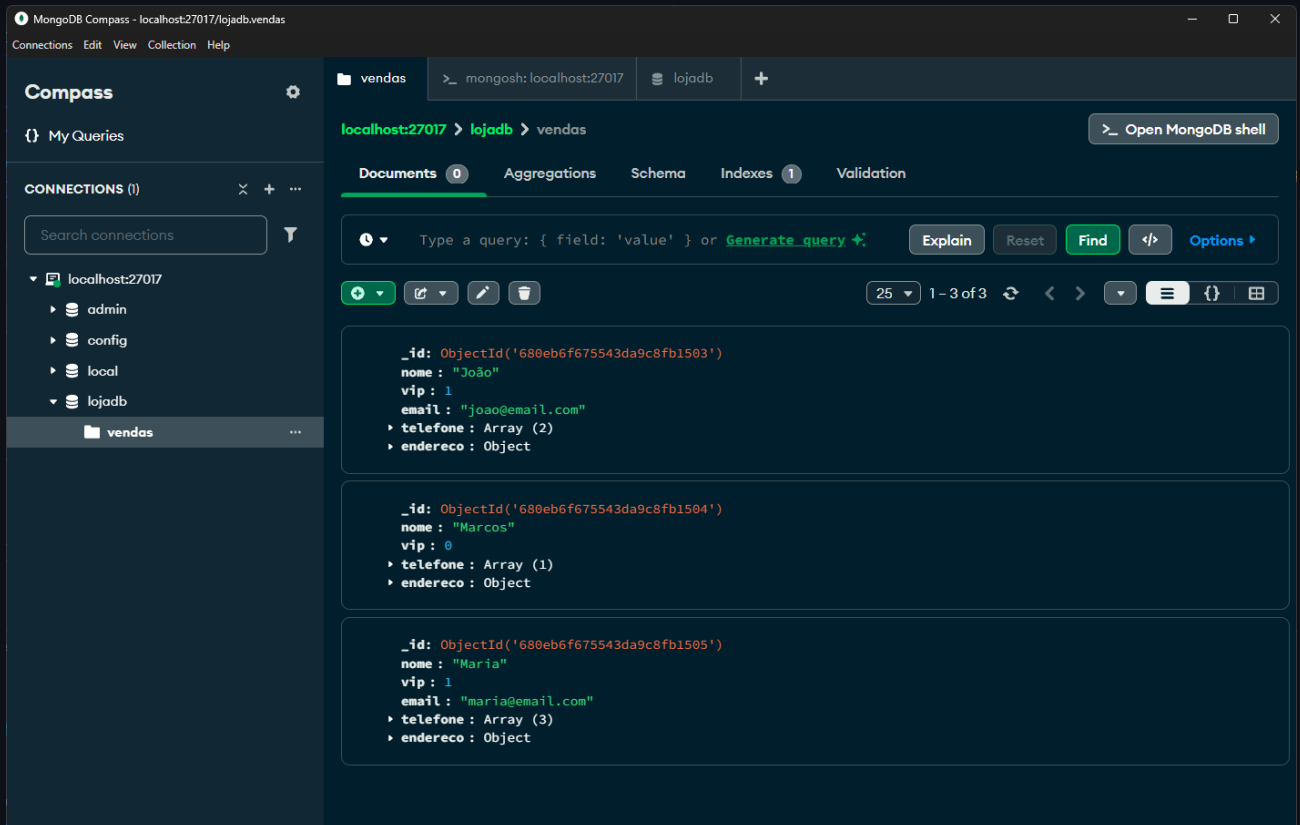
a. Atualização do endereço de Maria.

```
_id: ObjectId('680eb6f675543da9c8fb1505')  
nome : "Maria"  
vip : 1  
email : "maria@email.com"  
▼ telefone : Array (3)  
  0: "9999-3333"  
  1: "8888-3333"  
  2: "9988-3000"  
▼ endereco : Object  
  rua : "Rua Três"  
  numero : 3000  
  cidade : "Londrina"  
  estado : "PR"
```

```
// db-vendas-update-maria-endereco  
db.vendas.updateOne(  
  { "nome": "Maria" },  
  {  
    $set: {  
      "endereco": {  
        "rua": "Rua Três",  
        "numero": 3000,  
        "cidade": "Londrina",  
        "estado": "PR"  
      }  
    }  
  }  
)
```

```
}  
)
```

a. Atualização geral dos endereços.



2. Atualização com o campo **compras** (array de objetos: `nome_produto`, `preco`, `quantidade`).

a. Atualização das compras de João.

```
_id: ObjectId('680eb6f675543da9c8fb1503')  
nome: "João"  
vip: 1  
email: "joao@email.com"  
▼ telefone: Array (2)  
  0: "9999-1111"  
  1: "8888-1111"  
▼ endereco: Object  
  rua: "Rua Um"  
  numero: 1000  
  complemento: "Apto 1 Bloco 1"  
  cidade: "São Paulo"  
  estado: "SP"  
▼ compras: Array (1)  
  0: Object  
    nome_produto: "notebook"  
    preco: 5000  
    quantidade: 1
```

```
// db-vendas-update-joao-compras  
db.vendas.updateOne(  
  { "nome": "João" },  
  {  
    $set: {  
      "compras": [  
        { "nome_produto": "notebook", "preco": 5000.00, "quantidade": 1 }  
      ]  
    }  
  }  
)
```

```
}  
)
```

a. Atualização das compras de Marcos.

```
_id: ObjectId('680eb6f675543da9c8fb1504')  
nome: "Marcos"  
vip: 0  
▼ telefone: Array (1)  
  0: "9999-2222"  
▼ endereco: Object  
  rua: "Rua Dois"  
  numero: 4000  
  cidade: "Campinas"  
  estado: "SP"  
▼ compras: Array (2)  
  ▼ 0: Object  
    nome_produto: "Caderno"  
    preco: 20  
    quantidade: 1  
  ▼ 1: Object  
    nome_produto: "Caneta"  
    preco: 3  
    quantidade: 5
```

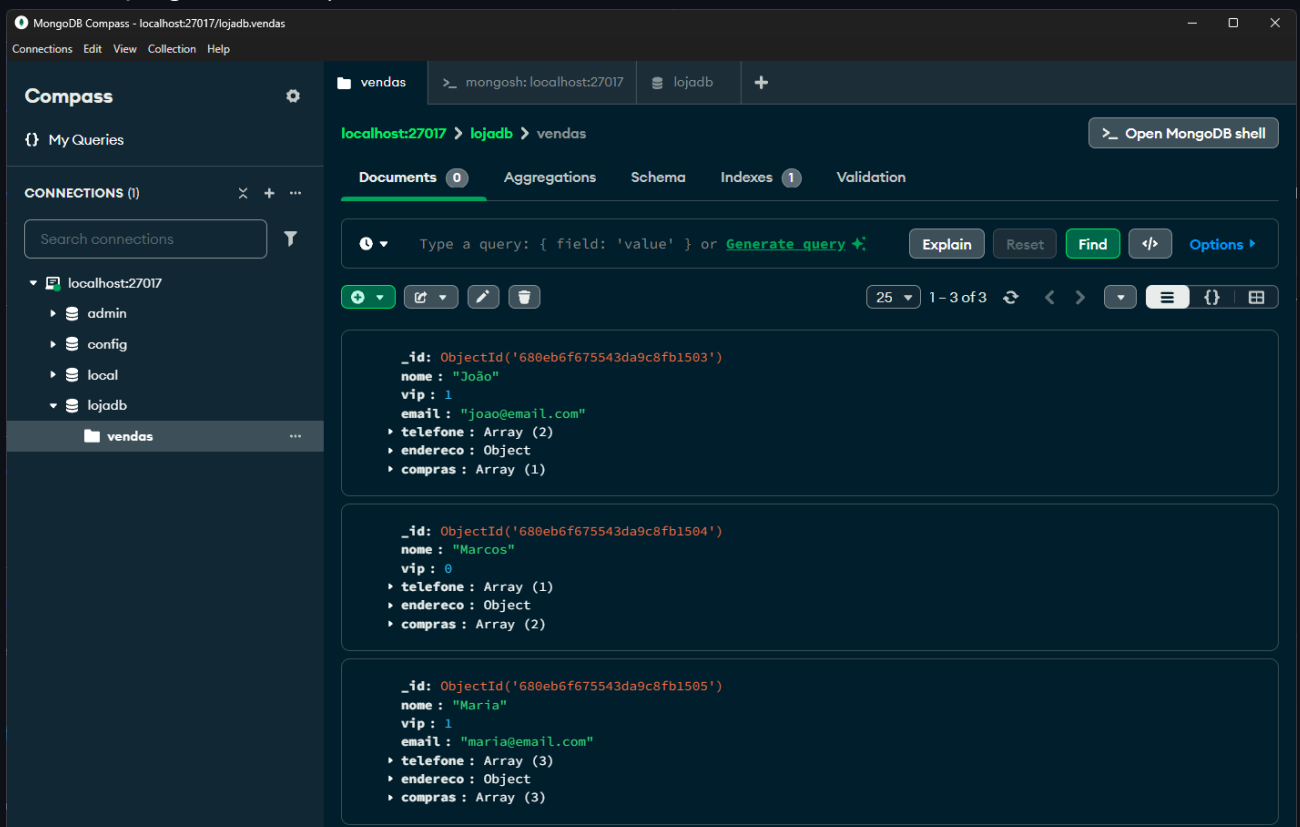
```
// db-vendas-update-marcos-compras  
db.vendas.updateOne(  
  { "nome": "Marcos" },  
  {  
    $set: {  
      "compras": [  
        { "nome_produto": "Caderno", "preco": 20.00, "quantidade": 1 },  
        { "nome_produto": "Caneta", "preco": 3.00, "quantidade": 5 }  
      ]  
    }  
  }  
)
```

a. Atualização das compras de Maria.

```
_id: ObjectId('680eb6f675543da9c8fb1505')  
nome: "Maria"  
vip: 1  
email: "maria@email.com"  
▼ telefone: Array (3)  
  0: "9999-3333"  
  1: "8888-3333"  
  2: "9988-3000"  
▼ endereco: Object  
  rua: "Rua Três"  
  numero: 3000  
  cidade: "Londrina"  
  estado: "PR"  
▼ compras: Array (3)  
  ▼ 0: Object  
    nome_produto: "Borracha"  
    preco: 2  
    quantidade: 2  
  ▼ 1: Object  
    nome_produto: "Tablet"  
    preco: 2500  
    quantidade: 1  
  ▼ 2: Object  
    nome_produto: "Capa para tablet"  
    preco: 50  
    quantidade: 1
```

```
// db-vendas-update-maria-compras
db.vendas.updateOne(
  { "nome": "Maria" },
  {
    $set: {
      "compras": [
        { "nome produto": "Borracha", "preco": 2.00, "quantidade": 2 },
        { "nome produto": "Tablet", "preco": 2500.00, "quantidade": 1 },
        { "nome produto": "Capa para tablet", "preco": 50.00, "quantidade": 1 }
      ]
    }
  }
)
```

a. Atualização geral das compras.



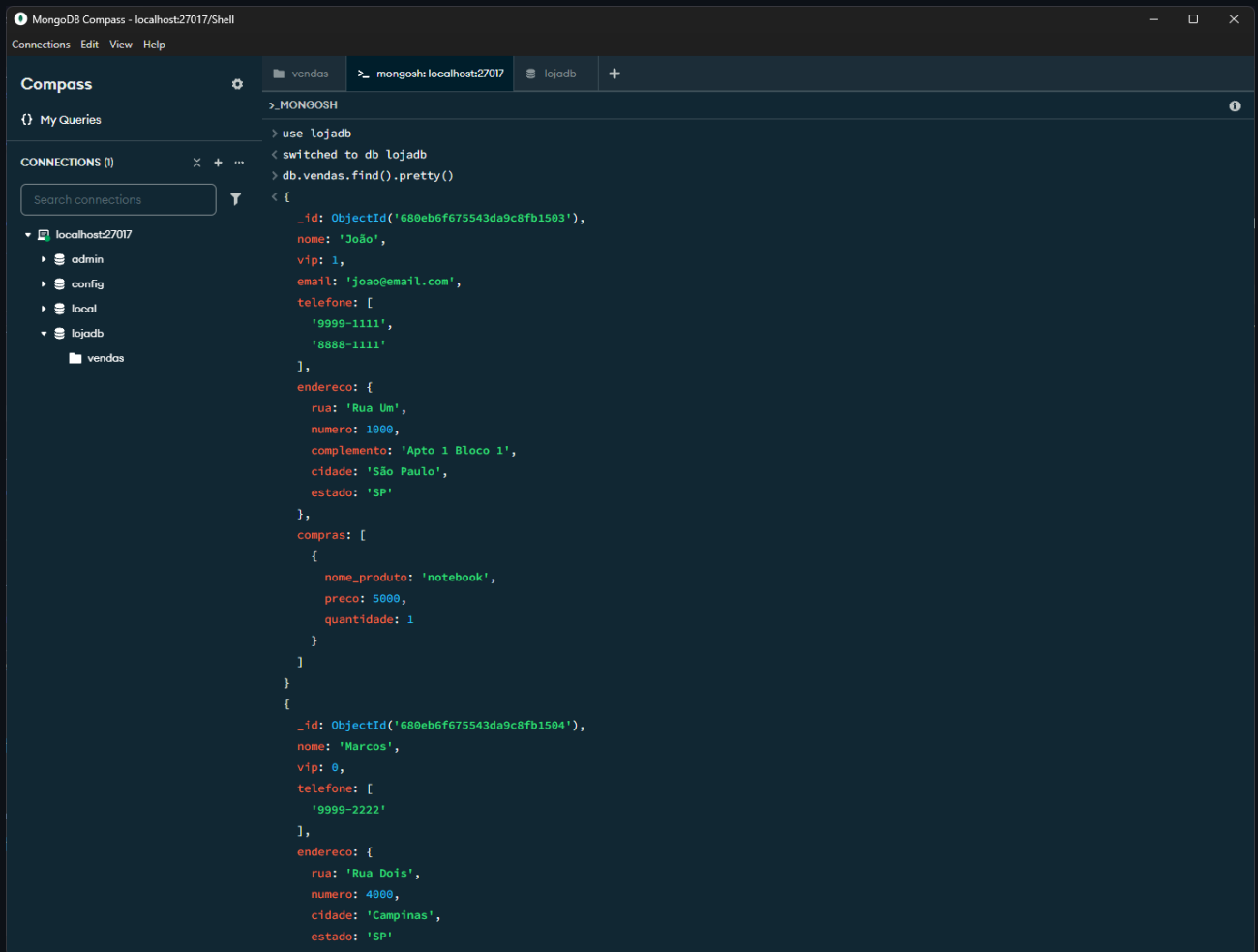
2.3 Processamento de Dados (Procedimento/Atividade Nº 2)

Esta etapa focou na realização de queries na collection **vendas** do banco **lojadb**.

1. Consulta que retorna todos os documentos da collection.

a. Consulta para retornar todos os documentos.

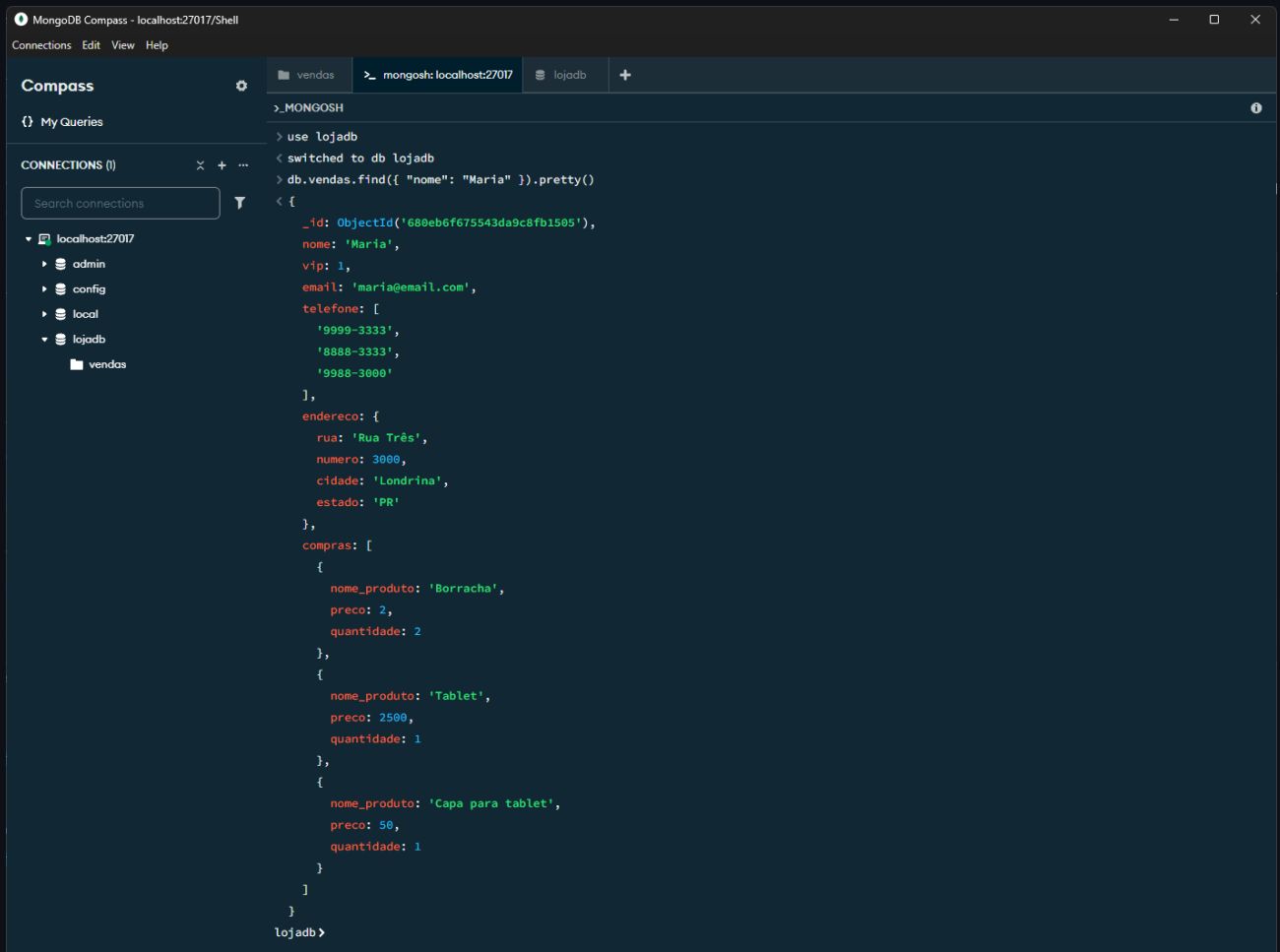
```
// db-vendas-find-all
db.vendas.find().pretty()
```



2. Consulta que localiza as informações da cliente “Maria”.

a. Consulta para localizar informações da cliente Maria.

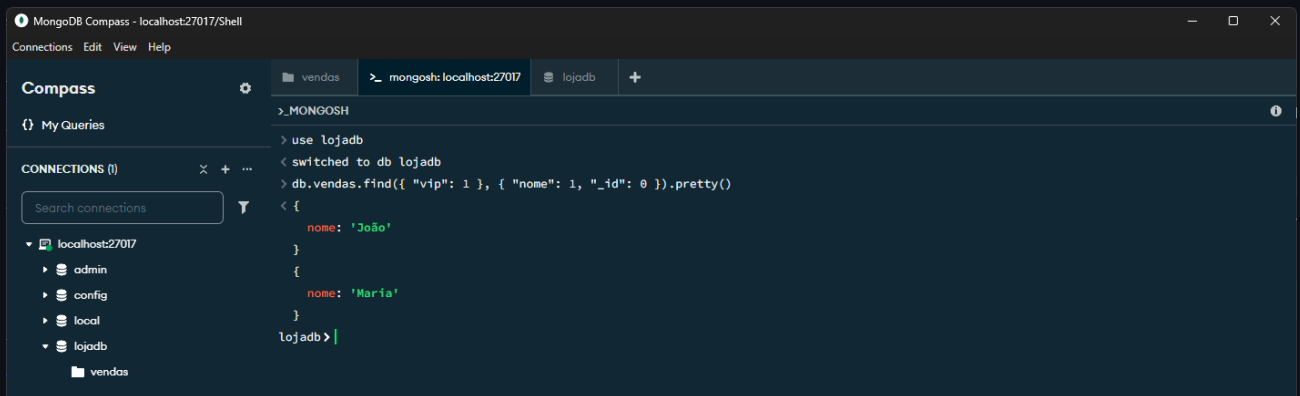
```
// db-vendas-find-maria
db.vendas.find({ "nome": "Maria" }).pretty()
```

3. Consulta que retorna o campo **nome** dos clientes **VIPs**.

a. Consulta para retornar nomes dos clientes VIPs.

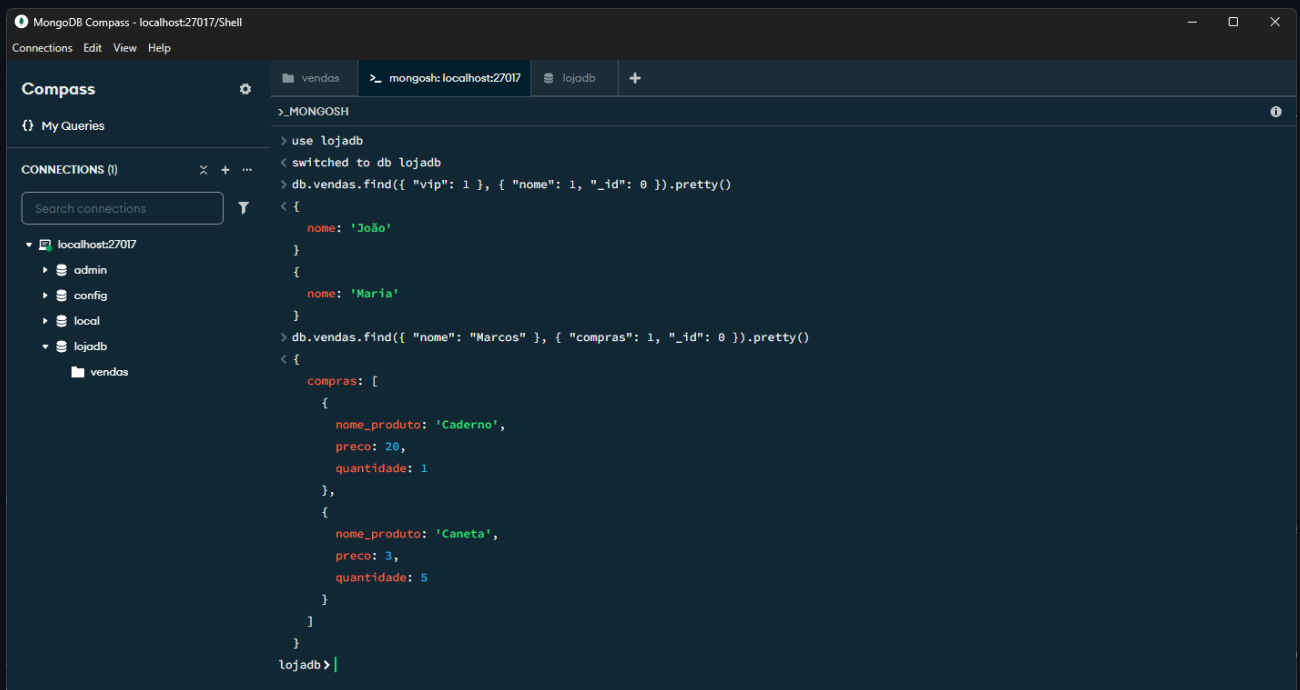
```
// db-vendas-find-vips  
db.vendas.find({ "vip": 1 }, { "nome": 1, "_id": 0 }).pretty()
```



4. Consulta que exibe as **compras** efetuadas por "Marcos".

a. Consulta para exibir compras de Marcos.

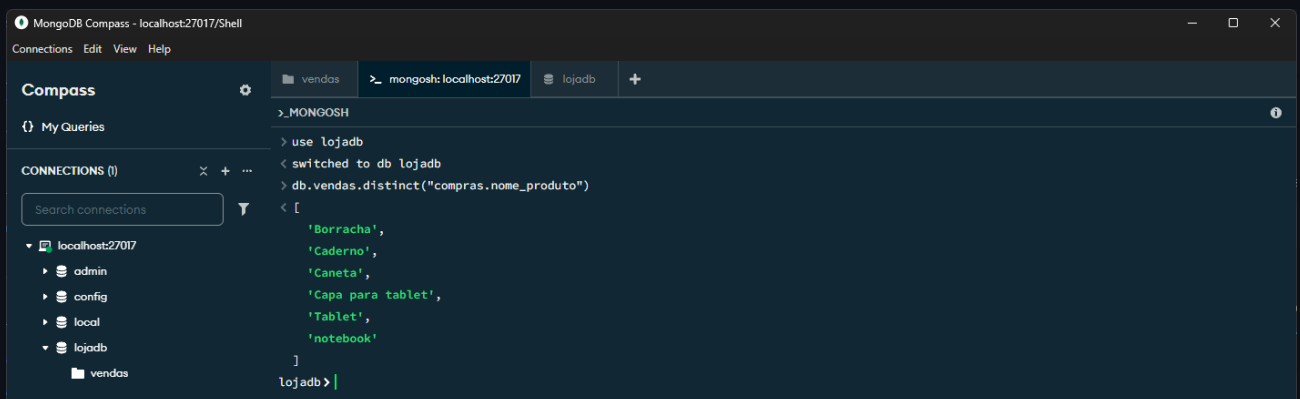
```
// db-vendas-find-marcos-compras  
db.vendas.find({ "nome": "Marcos" }, { "compras": 1, "_id": 0 }).pretty()
```



5. Consulta que retorna os **nomes dos produtos** distintos comprados por todos os clientes.

a. Consulta para retornar nomes dos produtos distintos.

```
// db-vendas-distinct-produtos  
db.vendas.distinct("compras.nome_produto")
```



3. Resultados

As consultas realizadas na base **lojadb** permitiram observar, na prática, como o MongoDB lida com diferentes estruturas de dados. Os principais pontos identificados foram:

- A consulta geral mostrou que os dados foram inseridos e atualizados corretamente.
- A busca por nome facilitou encontrar informações específicas, como o registro da cliente **Maria**.
- A projeção de campos permitiu visualizar apenas os dados necessários, como os nomes dos clientes **VIPs**.
- Consultas em arrays e objetos aninhados, como as compras de cada cliente, mostraram a flexibilidade do modelo de documentos.
- A consulta de produtos distintos ajudou a identificar todos os itens comprados, sem repetições.

Esses resultados mostram que o MongoDB pode ser útil em situações que exigem flexibilidade e consultas variadas, contribuindo para a análise dos dados de forma simples e eficiente.

4. Conclusão

A atividade prática permitiu conhecer, de forma simples, o funcionamento do **MongoDB** para criar, atualizar e consultar dados em um banco não relacional. Foi possível perceber a flexibilidade do modelo de documentos e a facilidade para realizar consultas variadas. A experiência contribuiu para entender conceitos básicos de bancos de dados NoSQL e como aplicá-los em situações do dia a dia. O repositório com os arquivos do projeto pode ser acessado em [GitHub](#).

5. Referências Bibliográficas

MongoDB. (2024). MongoDB Documentation. Disponível em: <https://www.mongodb.com/pt-br/docs/mongodb-shell/crud/>

Markdown Community. (2024). The Markdown Guide. Disponível em: <https://www.markdownguide.org/basic-syntax/>

MongoDB. (2024). O que é um banco de dados de documentos?. Disponível em: <https://www.mongodb.com/pt-br/resources/basics/databases/document-databases>