

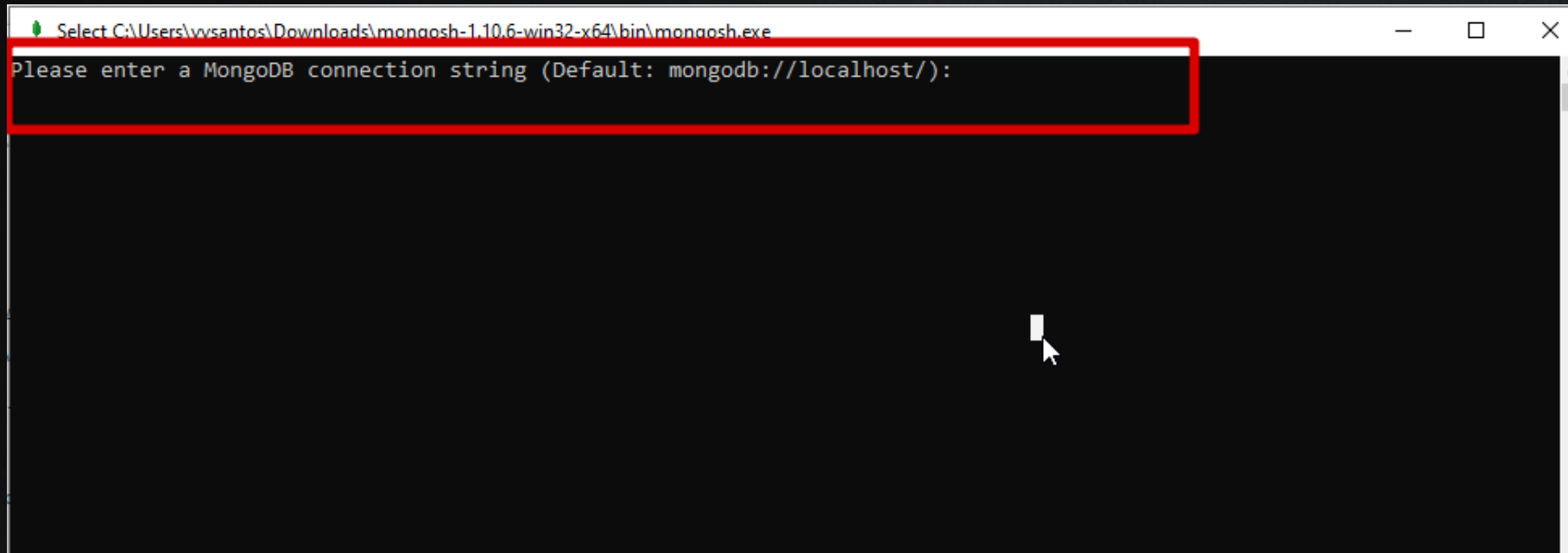
FIAP

AGENDA:

Atividades com documentos em MongoDB

Criar um no database pelo MongoDBSH

1 Conecte no banco de dados utilizando o MongoSH



Select mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

Please enter a MongoDB connection string (Default: mongodb://localhost/):

Current Mongosh Log ID: 6509d18ba11e9e86c65de81f

Connecting to: **mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.10.6**

Using MongoDB: 7.0.1

Using Mongosh: 1.10.6

mongosh 2.0.1 is available for download: <https://www.mongodb.com/try/download/shell>

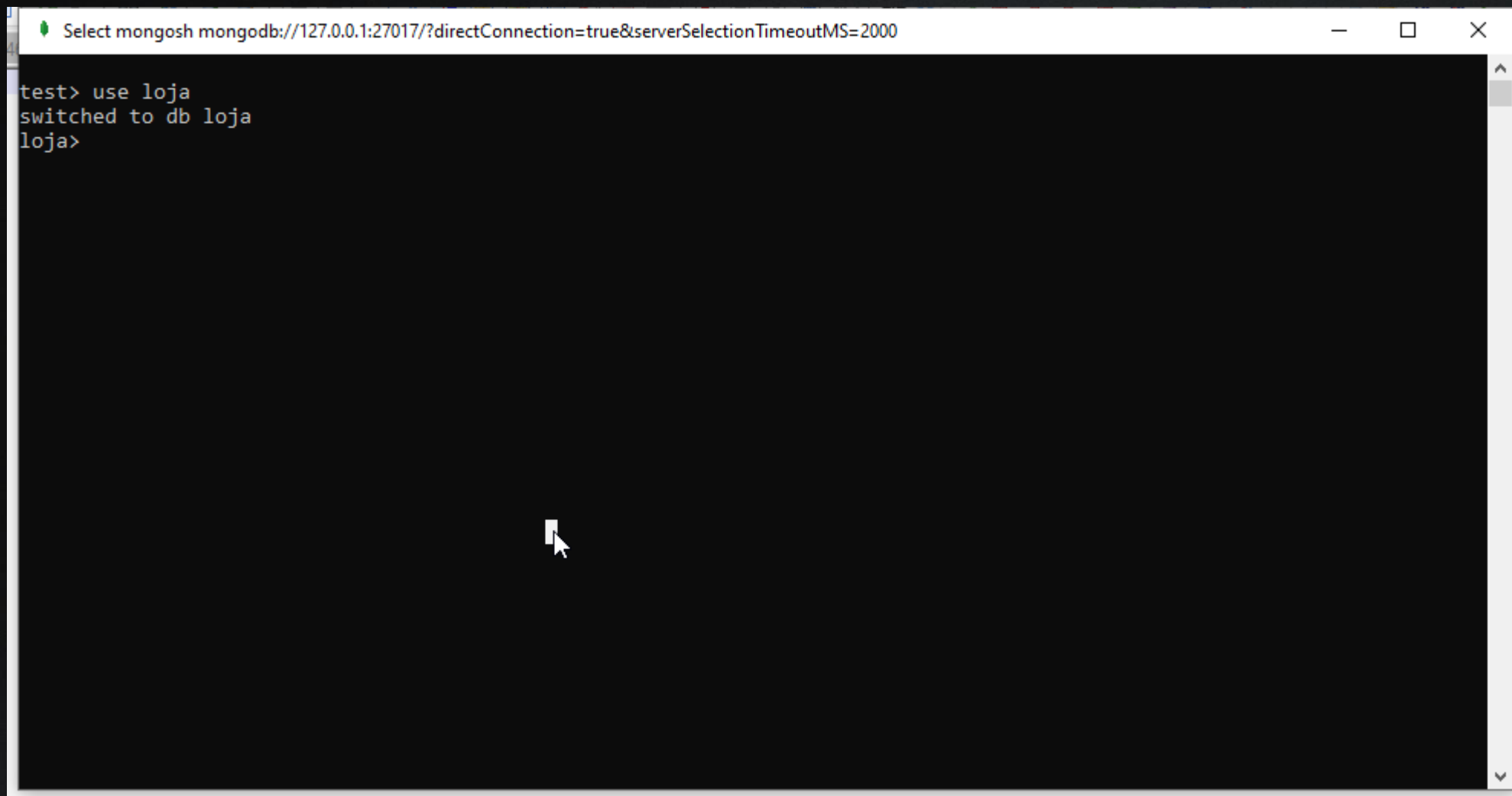
For mongosh info see: <https://docs.mongodb.com/mongosh-shell/>

The server generated these startup warnings when booting

2023-09-19T13:47:03.370-03:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

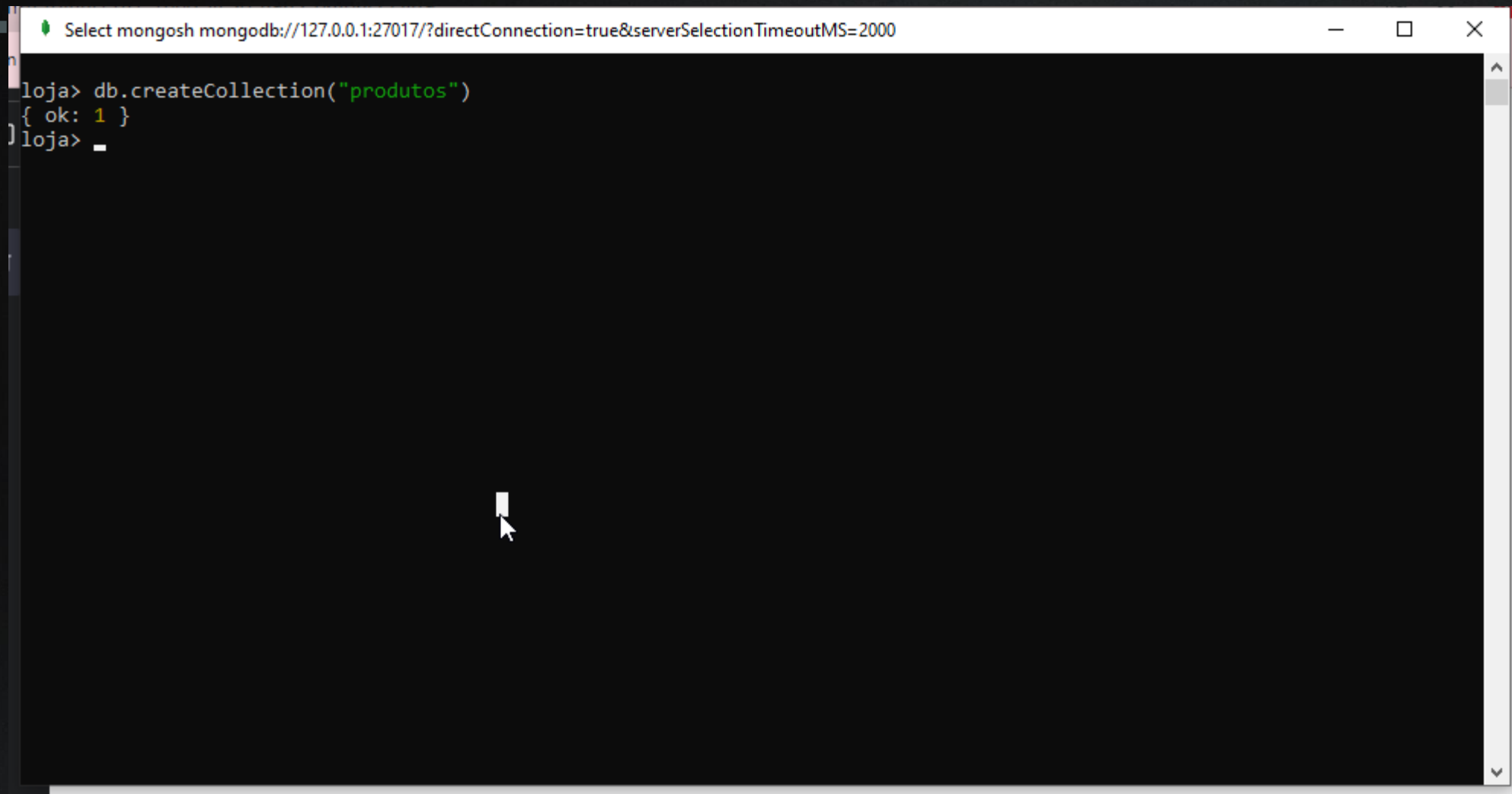
test> _

1. Crie um banco de dados chamado "loja"

A screenshot of a terminal window with a white title bar. The title bar contains a green icon, the text "Select mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000", and standard window controls (minimize, maximize, close). The terminal area has a black background with white text. The text shows a prompt "test>" followed by the command "use loja", a confirmation message "switched to db loja", and a new prompt "loja>". A mouse cursor is visible near the bottom center of the terminal area.

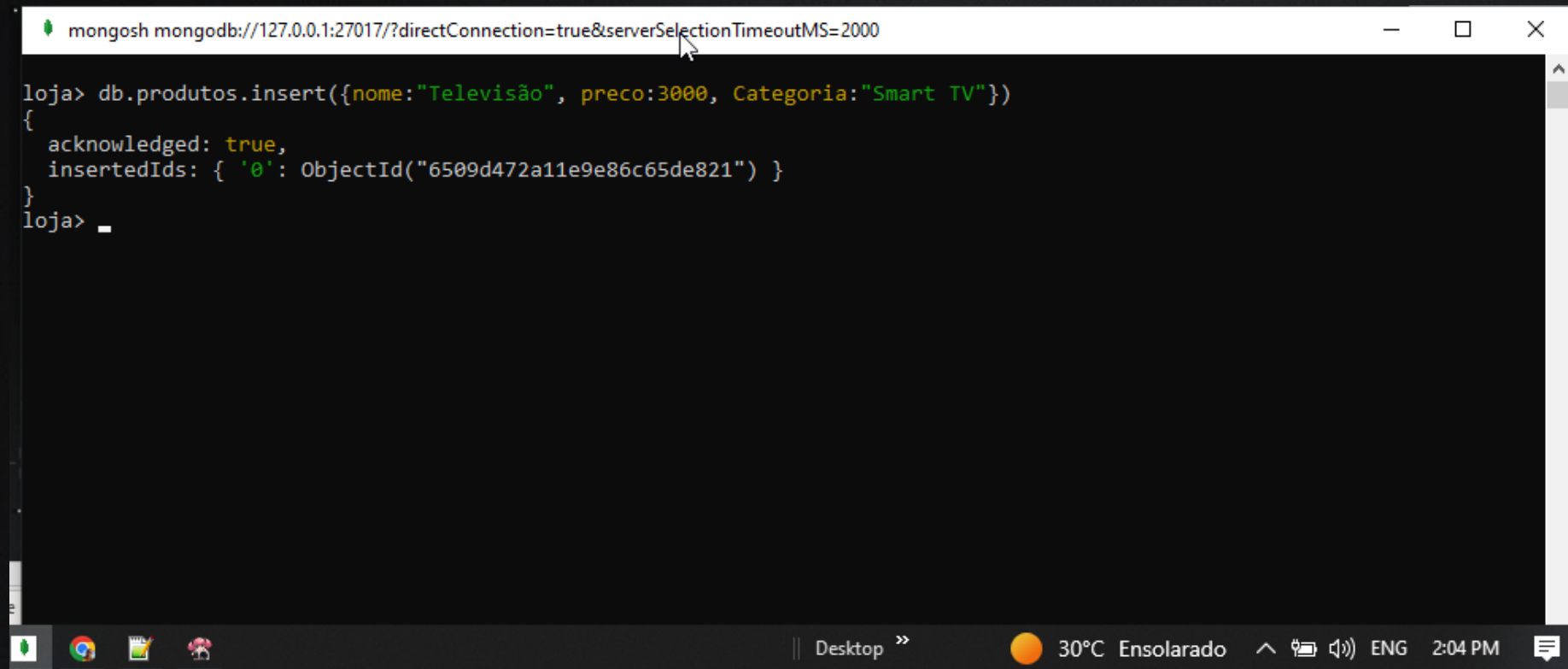
```
Select mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
test> use loja
switched to db loja
loja>
```

2. Crie uma coleção chamada "produtos" no banco de dados "loja"

A screenshot of a terminal window titled "Select mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000". The terminal shows a MongoDB shell session for the "loja" database. The command "db.createCollection('produtos')" is entered, and the output is "{ ok: 1 }". The prompt "loja>" is visible at the end of the line.

```
Select mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
loja> db.createCollection("produtos")
{ ok: 1 }
loja>
```


3. Insira um novo produto na coleção "produtos" com campos como "nome," "preço," e "categoria."



The screenshot shows a terminal window titled "mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000". The prompt is "loja>". The user has entered the command: `db.produtos.insert({nome:"Televisão", preco:3000, Categoria:"Smart TV"})`. The output is a JSON object: `{ acknowledged: true, insertedIds: { '0': ObjectId("6509d472a11e9e86c65de821") } }`. The prompt is now "loja> _". The taskbar at the bottom shows icons for a terminal, Chrome, and other applications, along with system status information: "Desktop", "30°C Ensolarado", "ENG", and "2:04 PM".

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
loja> db.produtos.insert({nome:"Televisão", preco:3000, Categoria:"Smart TV"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6509d472a11e9e86c65de821") }
}
loja> _
```

4. Insira mais 4 documentos com os mesmos atributos

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

loja> db.produtos.find()
[
  {
    _id: ObjectId("6509d472a11e9e86c65de821"),
    nome: 'Televisão',
    preco: 3000,
    Categoria: 'Smart TV'
  }
]
loja> db.produtos.insert({nome:"Computador", preco:6000, Categoria:"Notebook"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6509d5dba11e9e86c65de822") }
}
loja> db.produtos.insert({nome:"Ar condicionado", preco:1400, Categoria:"7Mil Btus"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6509d600a11e9e86c65de823") }
}
loja> db.produtos.insert({nome:"SmartPhone", preco:4000, Marca:"Apple"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6509d63ca11e9e86c65de824") }
}
loja> _
```


5 . Consulte os documentos inseridos

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
loja> db.produtos.find()
[
  {
    _id: ObjectId("6509d472a11e9e86c65de821"),
    nome: 'Televisão',
    preco: 3000,
    Categoria: 'Smart TV'
  },
  {
    _id: ObjectId("6509d5dba11e9e86c65de822"),
    nome: 'Computador',
    preco: 6000,
    Categoria: 'Notebook'
  },
  {
    _id: ObjectId("6509d600a11e9e86c65de823"),
    nome: 'Ar condicionado',
    preco: 1400,
    Categoria: '7Mil Btus'
  },
  {
    _id: ObjectId("6509d63ca11e9e86c65de824"),
    nome: 'SmartPhone',
    preco: 4000,
    Marca: 'Apple'
  }
]
loja>
```

6 . Atualizando valores no documento

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

loja> db.produtos.update(
... {nome:"Computador"},
... {$set:{preco: 8.000}}
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
loja> _
```

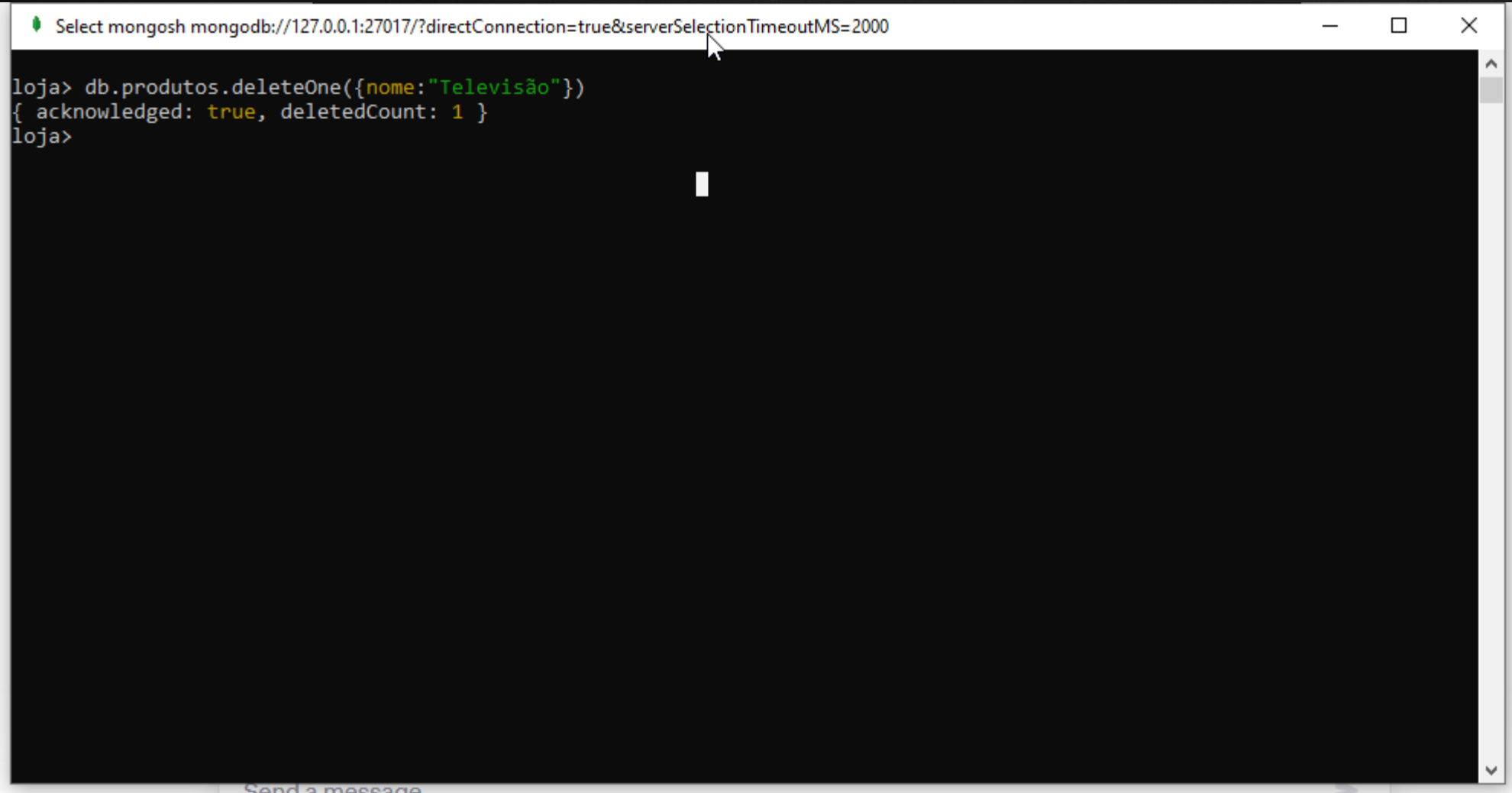
Vergílio Valério dos Santos (pf1788@fiap.com)

7. Consultando Valores específicos

```
Select mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

loja> db.produtos.update(
... {nome:"Computador"},
... {$set:{preco: 8.000}}
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
loja> db.produtos.find({nome:"Computador"})
[
  {
    _id: ObjectId("6509d5dba11e9e86c65de822"),
    nome: 'Computador',
    preco: 8,
    Categoria: 'Notebook'
  }
]
loja>
```

8. Excluindo um documento específico



The screenshot shows a terminal window titled "Select mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000". The terminal content shows a MongoDB shell session where the command `db.produtos.deleteOne({nome:"Televisão"})` is executed. The output is `{ acknowledged: true, deletedCount: 1 }`. The prompt `loja>` is visible at the end of the line.

```
loja> db.produtos.deleteOne({nome:"Televisão"})
{ acknowledged: true, deletedCount: 1 }
loja>
```

Send a message

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

loja> db.produtos.deleteOne({nome:"Televisão"})
{ acknowledged: true, deletedCount: 1 }
loja> db.produtos.find()
[
  {
    _id: ObjectId("6509d5dba11e9e86c65de822"),
    nome: 'Computador',
    preco: 8,
    Categoria: 'Notebook'
  },
  {
    _id: ObjectId("6509d600a11e9e86c65de823"),
    nome: 'Ar condicionado',
    preco: 1400,
    Categoria: '7Mil Btus'
  },
  {
    _id: ObjectId("6509d63ca11e9e86c65de824"),
    nome: 'SmartPhone',
    preco: 4000,
    Marca: 'Apple'
  }
]
loja>
```

9 Consulta por Ordenação

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
loja> db.produtos.find().sort({ nome: -1 })
[
  {
    _id: ObjectId("6509d63ca11e9e86c65de824"),
    nome: 'SmartPhone',
    preco: 4000,
    Marca: 'Apple'
  },
  {
    _id: ObjectId("6509d5dba11e9e86c65de822"),
    nome: 'Computador',
    preco: 8,
    Categoria: 'Notebook'
  },
  {
    _id: ObjectId("6509d600a11e9e86c65de823"),
    nome: 'Ar condicionado',
    preco: 1400,
    Categoria: '7Mil Btus'
  }
]
loja> _
```


10 Consulte apenas os nomes e categorias dos produtos.



The screenshot shows a terminal window titled "mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000". The prompt is "loja>". The command entered is "db.produtos.find({}, { nome: 1, preco: 1, _id: 0 })". The output is a JSON array of three objects: [{"nome": "Computador", "preco": 8}, {"nome": "Ar condicionado", "preco": 1400}, {"nome": "SmartPhone", "preco": 4000}]. The prompt "loja>" is followed by a cursor.

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
loja> db.produtos.find({}, { nome: 1, preco: 1, _id: 0 })
[
  { nome: 'Computador', preco: 8 },
  { nome: 'Ar condicionado', preco: 1400 },
  { nome: 'SmartPhone', preco: 4000 }
]
loja> _
```

11 Limitando a quantidade de linhas

Select mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
loja> db.produtos.find({}, { nome: 1, preco: 1, _id: 0 }).limit(2)
[
  { nome: 'Computador', preco: 8 },
  { nome: 'Ar condicionado', preco: 1400 }
]
loja>
```

```
db.products.insertMany([
  { nome: "Produto 1", preco: 19.99, categoria: "Eletrônicos" },
  { nome: "Produto 2", preco: 29.99, categoria: "Eletrônicos" },
  { nome: "Produto 3", preco: 9.99, categoria: "Eletrônicos" },
  { nome: "Produto 4", preco: 15.99, categoria: "Eletrônicos" },
  { nome: "Produto 5", preco: 39.99, categoria: "Eletrônicos" },
  { nome: "Produto 6", preco: 49.99, categoria: "Eletrônicos" }
])
```

12 Consultas aninhadas

```
Select mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
loja> db.produtos.find({ categoria: "Eletrônicos" })
[
  {
    _id: ObjectId("6509f635a11e9e86c65de82b"),
    nome: 'Produto 1',
    preco: 19.99,
    categoria: 'Eletrônicos'
  },
  {
    _id: ObjectId("6509f635a11e9e86c65de82c"),
    nome: 'Produto 2',
    preco: 29.99,
    categoria: 'Eletrônicos'
  },
  {
    _id: ObjectId("6509f635a11e9e86c65de82d"),
    nome: 'Produto 3',
    preco: 9.99,
    categoria: 'Eletrônicos'
  },
  {
    _id: ObjectId("6509f635a11e9e86c65de82e"),
    nome: 'Produto 4',
    preco: 15.99,
    categoria: 'Eletrônicos'
  },
  {
    _id: ObjectId("6509f635a11e9e86c65de82f"),
    nome: 'Produto 5',
    categoria: 'Eletrônicos'
  }
]
```

13 Atualização aninhadas

```
Select mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
loja> db.produtos.updateMany(
...   { categoria: "Eletrônicos" },
...   { $mul: { price: 1.1 } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 6,
  modifiedCount: 6,
  upsertedCount: 0
}
loja> _
```

14: Exclusão Aninhada:

```
db.produtos.deleteMany({ categoria: "Livros" })
```

15: Criando Index

```
db.produtos.createIndex({ nome: 1 })
```


Exercícios

1. Crie um banco de dados chamado "loja."
2. Crie uma coleção chamada "produtos" no banco de dados "loja."
3. Insira 10 produtos diferentes na coleção "produtos" usando `insertMany`.
4. Consulte todos os produtos na coleção "produtos."
5. Atualize o preço de um produto específico na coleção "produtos."
6. Insira mais cinco produtos diferentes na coleção "produtos" usando `insertMany`.
7. Consulte todos os produtos na coleção "produtos" ordenados por nome em ordem alfabética.
8. Atualize a categoria de todos os produtos de uma marca específica.
9. Exclua um produto específico da coleção "produtos."
10. Consulte apenas os três primeiros produtos na coleção "produtos."
11. Consulte os três produtos mais caros na coleção "produtos," ordenados por preço em ordem decrescente.
12. Exclua todos os produtos com uma quantidade em estoque igual a zero da coleção "produtos."
13. Consulte apenas os nomes e preços dos produtos na coleção "produtos."
14. Aumente o preço de todos os produtos em estoque em 10%.
15. Exclua todos os produtos com preço abaixo de um valor específico da coleção "produtos."