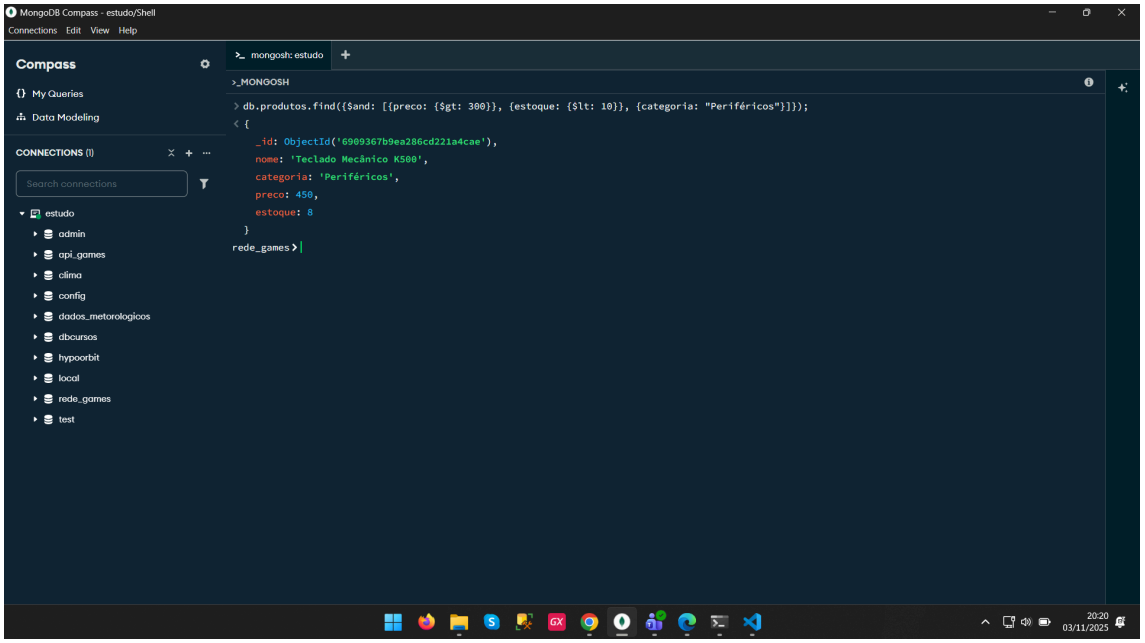
Atividade - Banco de Dados Não Relacional – DSM – Profa. Lucineide
Nome: Carlos Eduardo da Silva Magalhães
Tema: Atividade prática semanas 5 a 8

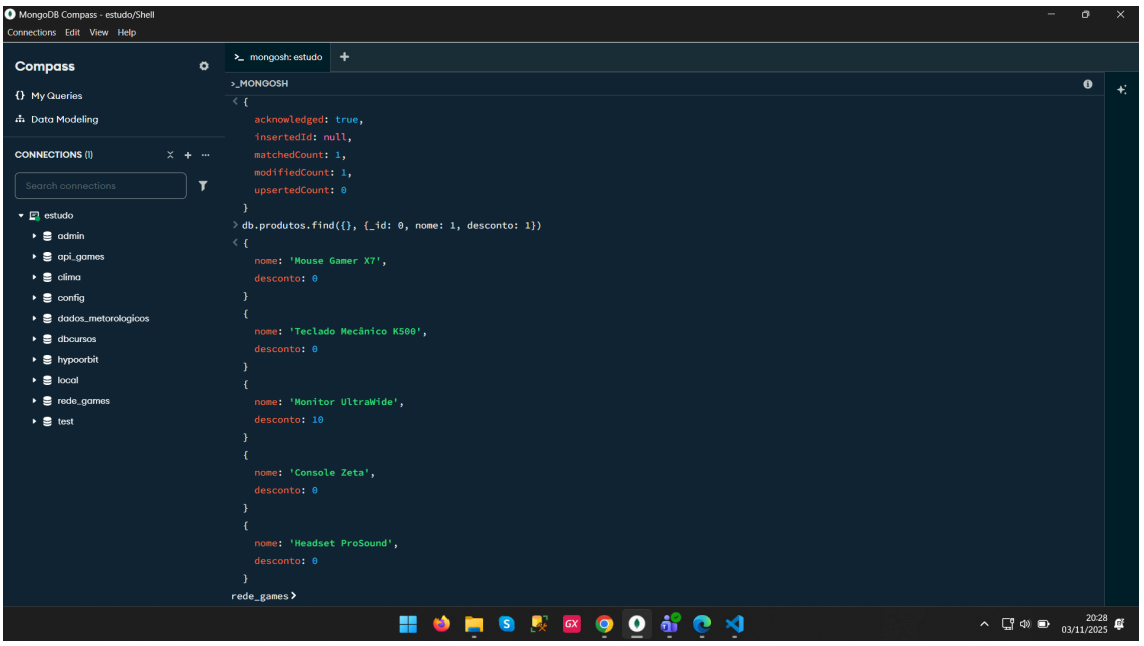
Exercício 01

```
db.produtos.find({$and: [{preco: {$gt: 300}}, {estoque: {$lt: 10}},
{categoria: "Periféricos"}]});
```



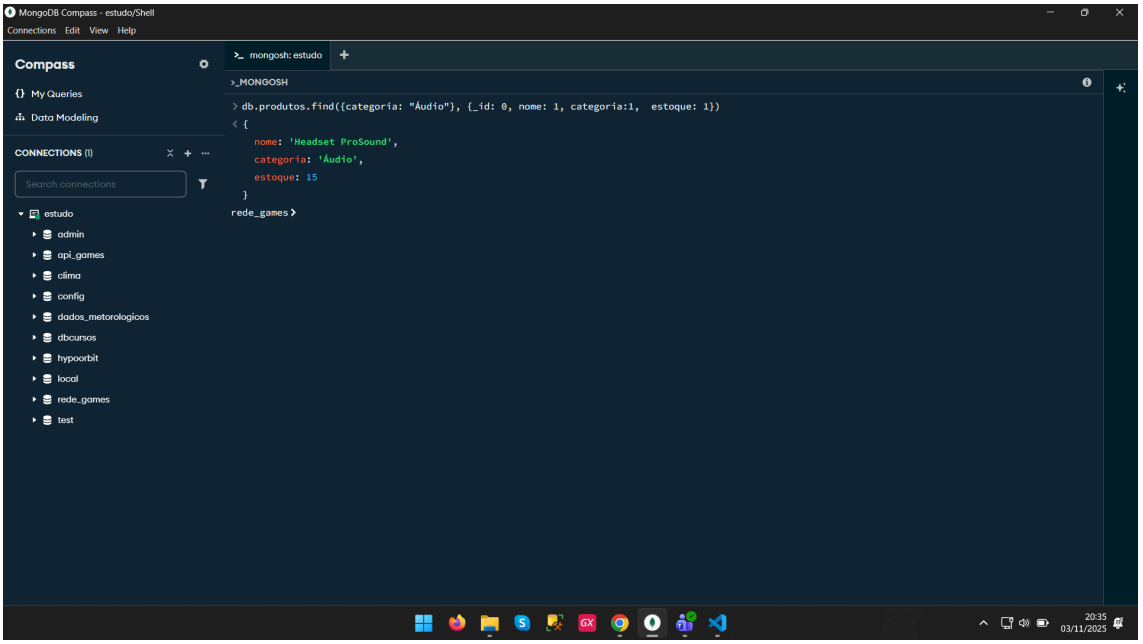
Exercício 02

```
db.produtos.updateMany({}, { $set: { desconto: 0.00 } });
db.produtos.updateOne({nome: "Monitor UltraWide"},{$set: {desconto:
10}});
db.produtos.find({}, {_id: 0, nome: 1, desconto: 1});
```



Exercício 03

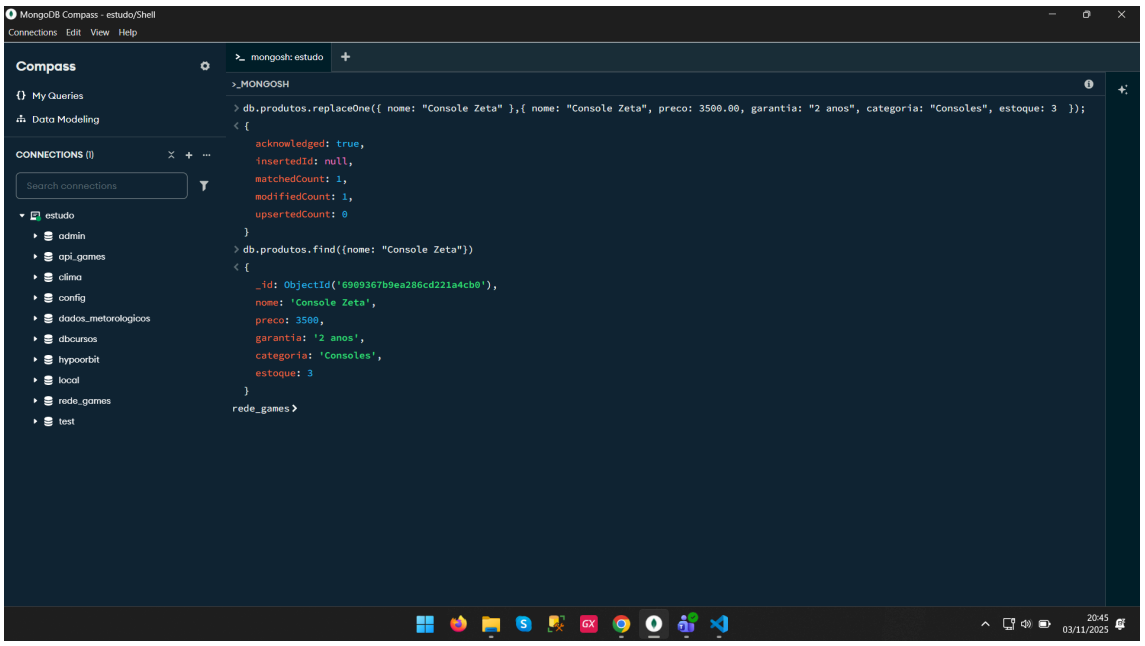
```
db.produtos.updateMany({categoria: "Áudio"}, {$inc: {estoque: 3}});
db.produtos.find({}, {_id: 0, nome: 1, categoria:1,  estoque: 1});
```



Exercício 04

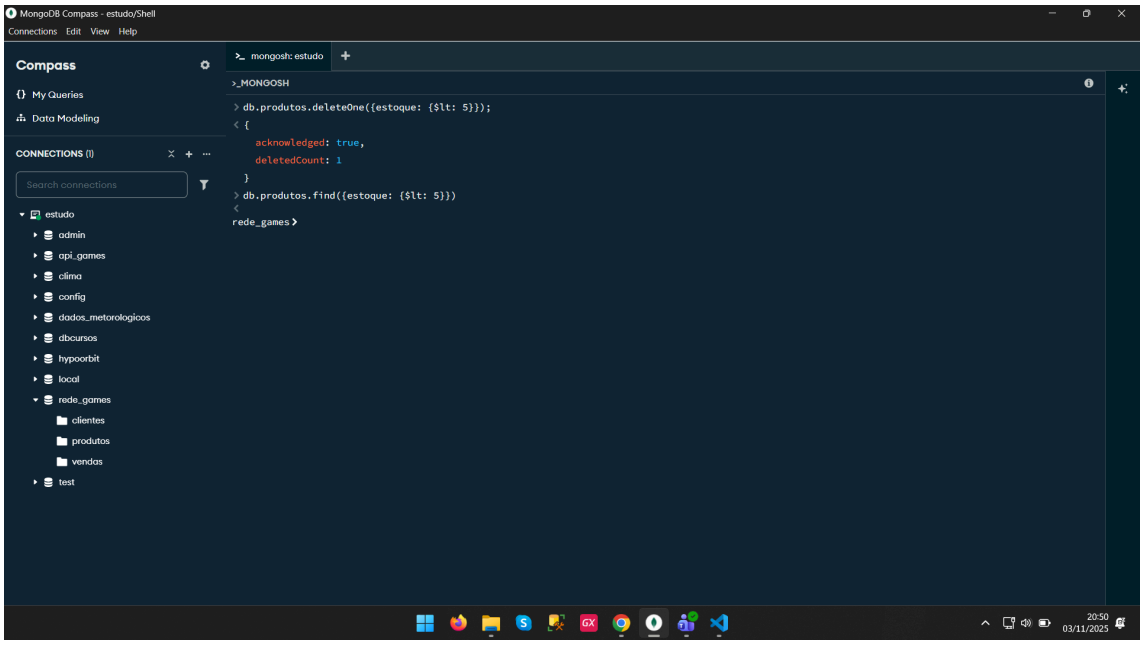
```
db.produtos.replaceOne({ nome: "Console Zeta" },{ nome: "Console Zeta",
preco: 3500.00, garantia: "2 anos", categoria: "Consoles", estoque: 3
});
```

```
db.produtos.find({nome: "Console Zeta"});
```



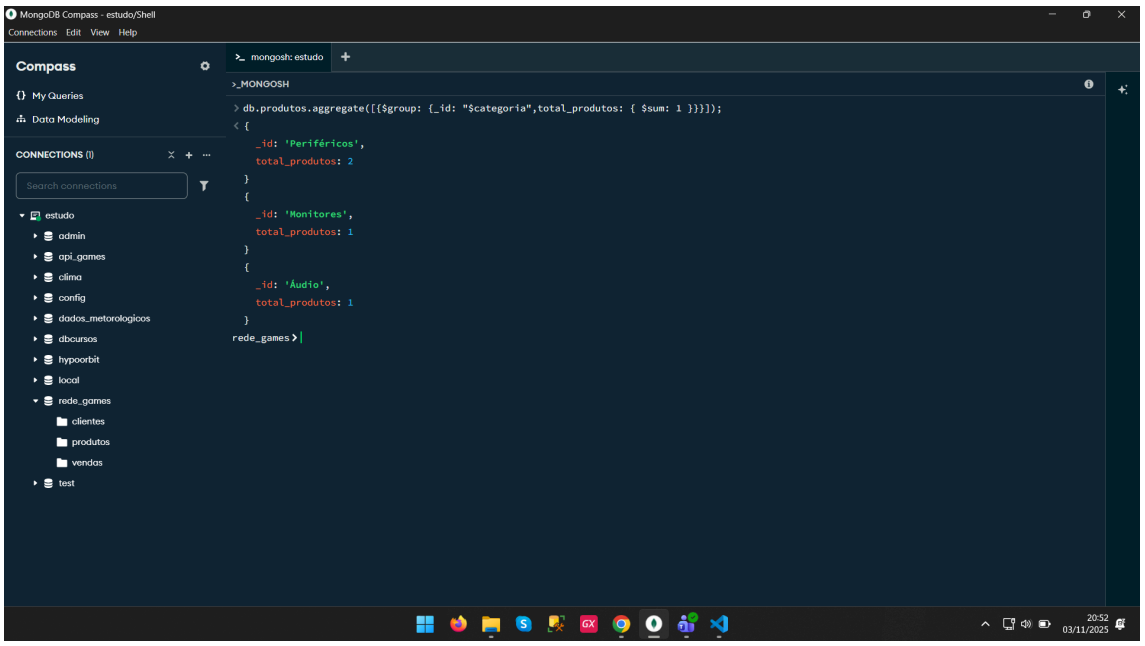
Exercício 05

```
db.produtos.deleteOne({estoque: {$lt: 5}});
db.produtos.find({estoque: {$lt: 5}});
```



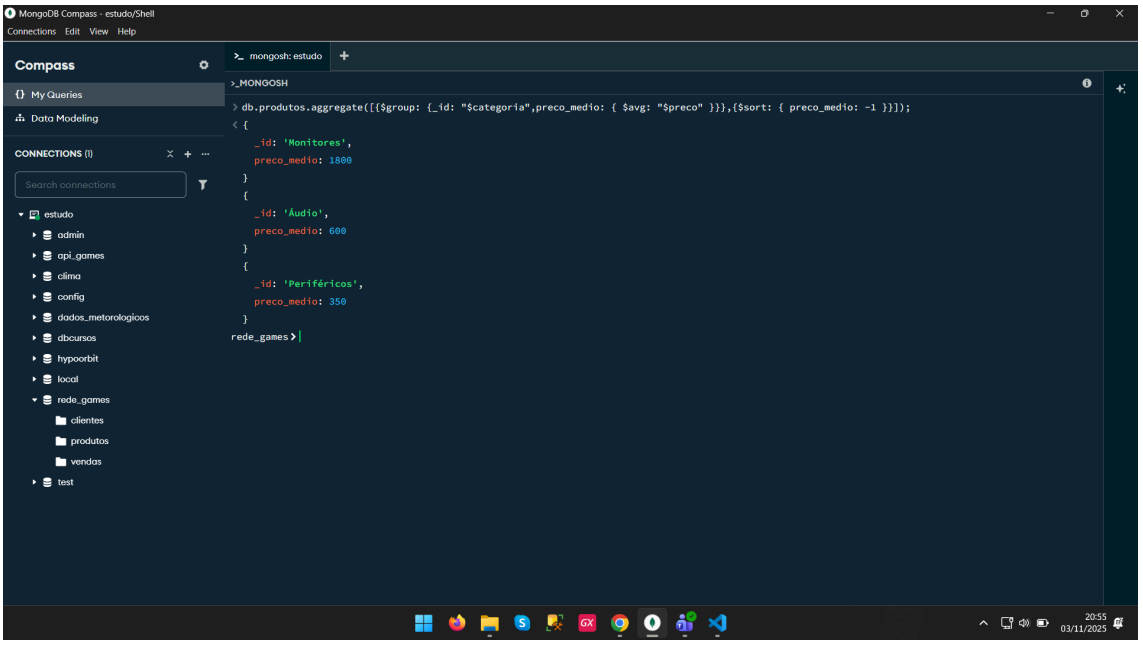
Exercício 06

```
db.produtos.aggregate([{$group: {_id: "$categoria",total_produtos: { $sum: 1 }}}]);
```



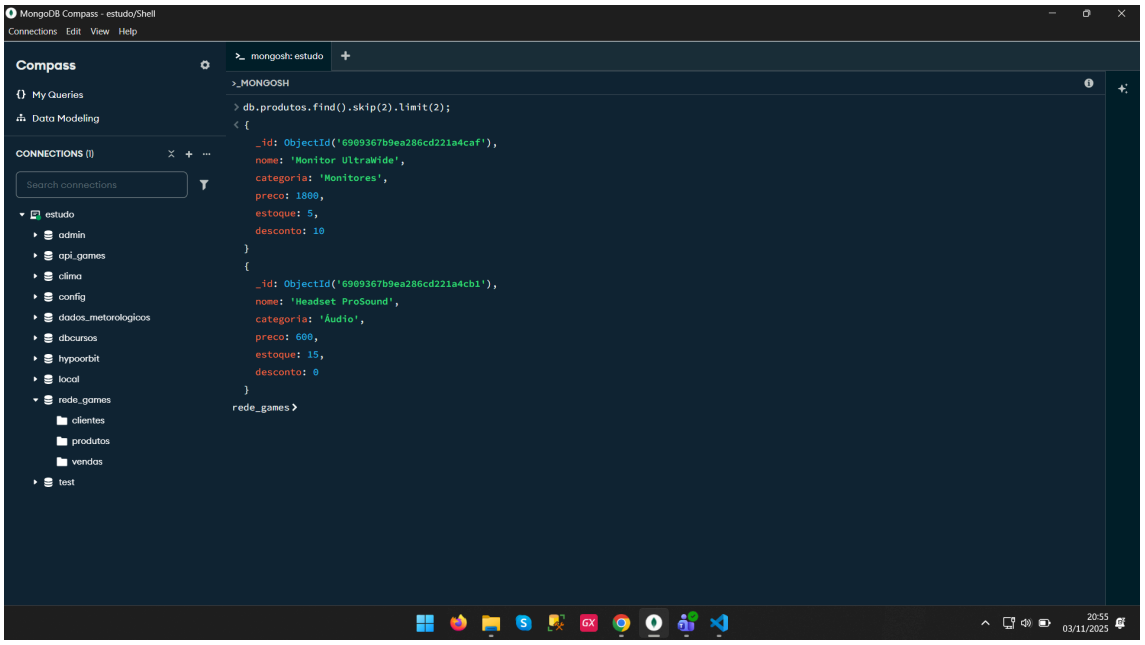
Exercício 07

```
db.produtos.aggregate([{$group: {_id: "$categoria",preco_medio: { $avg: "$preco" }}},{$sort: { preco_medio: -1 }}]);
```



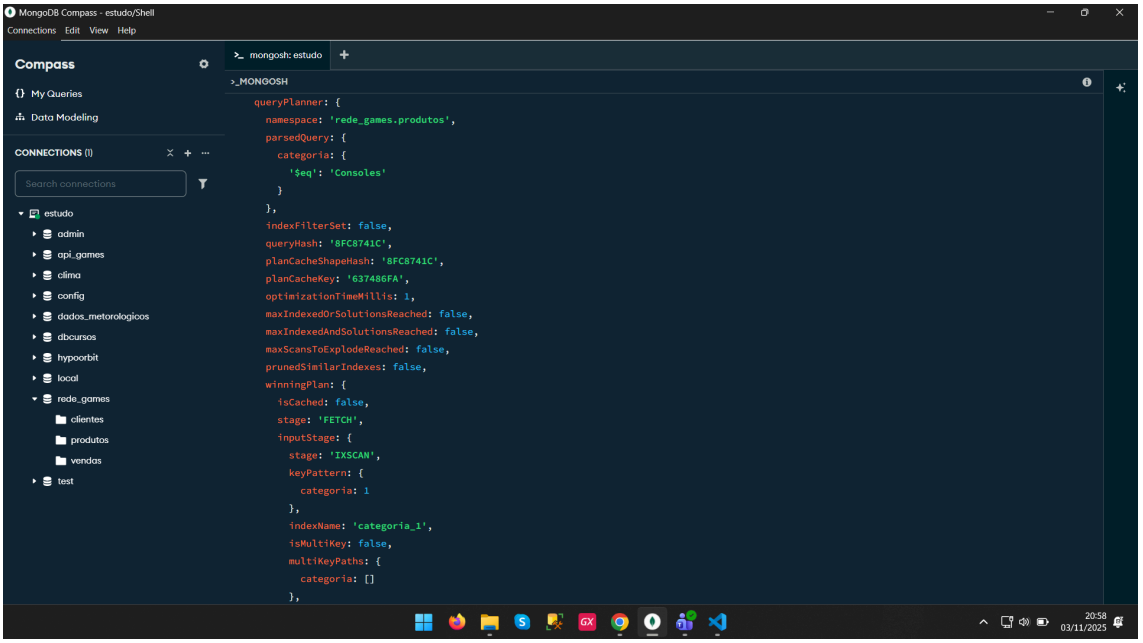
Exercício 08

```
db.produtos.find().skip(2).limit(2);
```



Exercício 09

```
db.produtos.createIndex({ categoria: 1 });
db.produtos.getIndexes();
db.produtos.find({ categoria: "Consoles" }).explain("executionStats");
```



Exercício 10

```
import mongoose, {Document, Schema} from "mongoose";
```

```
export interface IProdutos extends Document {
    nome: string;
    categoria: string;
    preco: number;
    estoque: number;
}

const ProdutoSchema: Schema = new Schema({
    nome: { type: String, required: true },
    categoria: { type: String, required: true },
    preco: { type: Number, required: true },
    estoque: { type: Number, required: true }
});

export default mongoose.model<IProdutos>('Produto', ProdutoSchema);
```

```
import express, {Request, Response} from "express";
import mongoose from "mongoose";
import path from "path";
import Produto from "./models/produtos";

const app = express();
const PORT = 3000;
const MONGO_URI = 'mongodb://localhost:27017/rede_games';

app.use(express.static(path.join(__dirname, "../view")));
app.use(express.json());

mongoose.connect(MONGO_URI)
    .then(() => console.log('MongoDB conectado'))
    .catch(err => console.log('Erro ao conectar ao MongoDB: ', err));


app.get("/api/produtos", async (req: Request, res: Response) => {
    try{
        const produtos = await Produto.find({}, {_id: 0, nome: 1,
preco: 1}).limit(5);
        res.json(produtos);
    }catch(err){
        res.status(500).json({message: 'Erro ao buscar produtos'});
    }
});
```

```
app.listen(PORT, () => {
  console.log(`Servidor rodando em http://localhost:${PORT}`);
  console.log(`API de produtos em
http://localhost:${PORT}/api/produtos`);
});
```