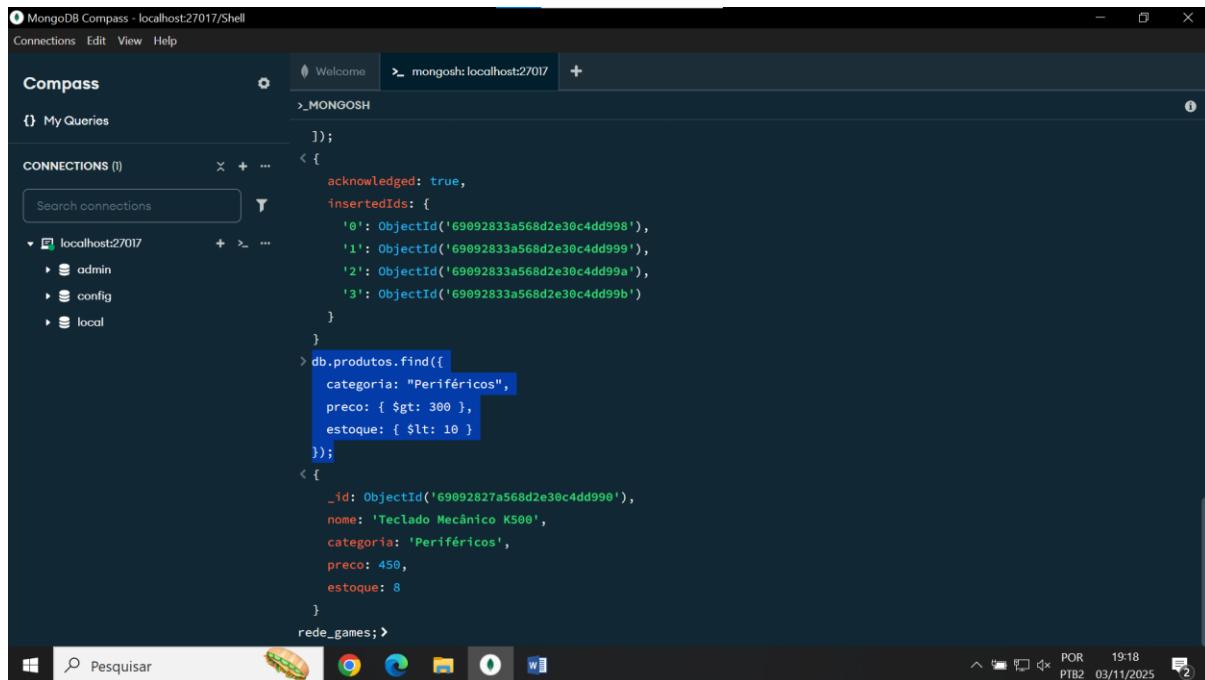


Atividade de Banco de Dados Não Relacional

Nome: Gustavo Henrique Ferreira Hammes

Questão 1 – Consulta com Operadores Lógicos Liste os produtos da categoria "Periféricos" cujo preço seja superior a R\$ 300 e estoque inferior a 10 unidades.

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



The screenshot shows the MongoDB Compass interface with the query results. The results are as follows:

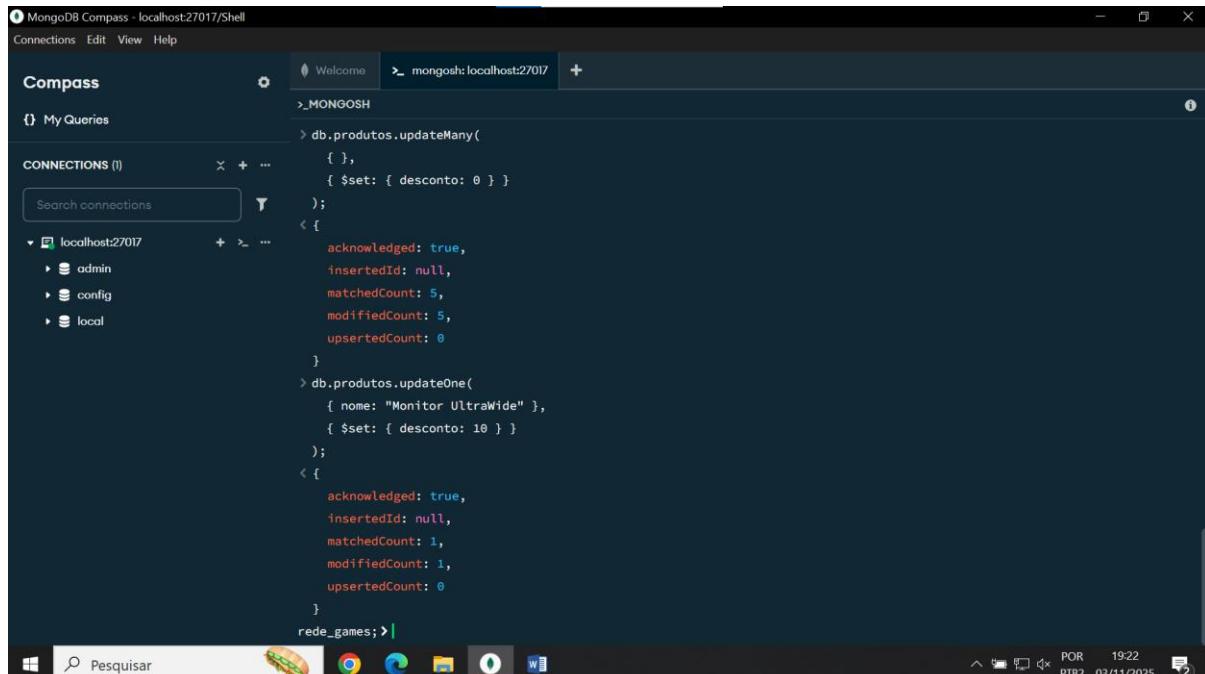
```
acknowledged: true,
insertedIds: {
  '0': ObjectId('69092833a568d2e30c4dd998'),
  '1': ObjectId('69092833a568d2e30c4dd999'),
  '2': ObjectId('69092833a568d2e30c4dd99a'),
  '3': ObjectId('69092833a568d2e30c4dd99b')
}
[{"_id": ObjectId("69092827a568d2e30c4dd998"),
  "nome": "Teclado Mecânico K500",
  "categoria": "Periféricos",
  "preco": 450,
  "estoque": 8}
```

Questão 2 – Inserção com Operadores de Atualização Adicione o campo "desconto": 0 em todos os produtos e, depois, atualize o "Monitor UltraWide" para aplicar desconto de 10%

```
db.produtos.updateMany(
  {},
  { $set: { desconto: 0 } }
);
```

```
db.produtos.updateOne(  
  { nome: "Monitor UltraWide" },  
  { $set: { desconto: 10 } }  
);
```

```
db.produtos.find();
```



The screenshot shows the MongoDB Compass interface. The title bar reads "MongoDB Compass - localhost:27017/Shell". The left sidebar shows "Compass" and "My Queries". Under "CONNECTIONS", there is a list of databases: "localhost:27017" (selected), "admin", "config", and "local". The main area displays the MongoDB shell command history. The commands entered were:

```
> db.produtos.updateMany(  
  { },  
  { $set: { desconto: 0 } }  
);  
< {  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 5,  
  modifiedCount: 5,  
  upsertedCount: 0  
}  
> db.produtos.updateOne(  
  { nome: "Monitor UltraWide" },  
  { $set: { desconto: 10 } }  
);  
< {  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}  
rede_games;>
```

The status bar at the bottom right shows "POR 19:22 PTB2 03/11/2025".

Questão 3 – Incremento de Valores Aumente o estoque de todos os produtos da categoria “Áudio” em 3 unidades.

```
db.produtos.updateMany(  
  { categoria: "Audio" },  
  { $inc: { estoque: 3 } }  
);  
  
db.produtos.find({ categoria: "Audio" });
```

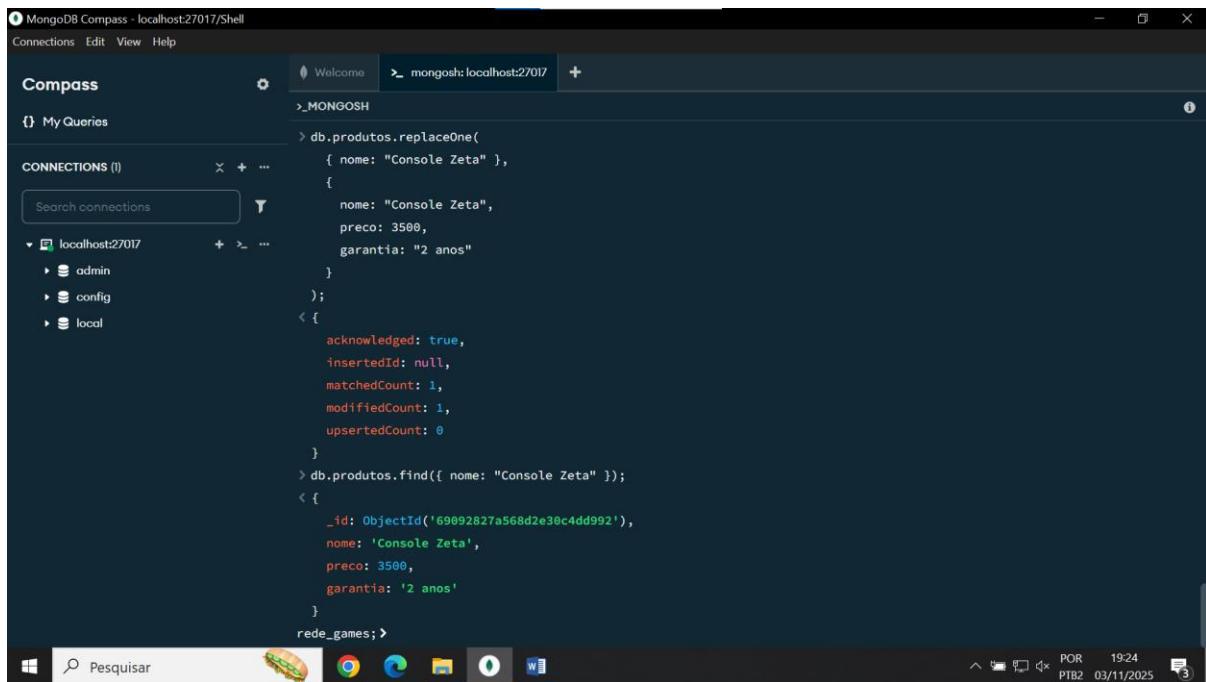
The screenshot shows the MongoDB Compass interface connected to 'mongosh: localhost:27017'. The left sidebar lists connections: 'localhost:27017' (selected), 'admin', 'config', and 'local'. The main area displays a command-line interface with the following code:

```
>_MONGOSH
    { $inc: { estoque: 3 } }
);
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 0,
    modifiedCount: 0,
    upsertedCount: 0
}
> // Este comando filtra pela categoria "Audio" para conferir o novo estoque
db.produtos.find({ categoria: "Audio" });
<
> db.produtos.find({ categoria: "Audio" });
<
> db.produtos.find({ categoria: "Áudio" });
<
{
    _id: ObjectId('69092827a568d2e30c4dd993'),
    nome: 'Headset ProSound',
    categoria: 'Áudio',
    preco: 600,
    estoque: 12,
    desconto: 0
}
rede_games;>
```

Questão 4 – Atualização e Substituição Substitua completamente o documento do produto “Console Zeta” por uma nova versão, mantendo o mesmo nome, mas com preço atualizado para 3500 e campo novo “garantia”: “2 anos”.

```
db.produtos.replaceOne(
    { nome: "Console Zeta" },
    {
        nome: "Console Zeta",
        preco: 3500,
        garantia: "2 anos"
    }
);

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



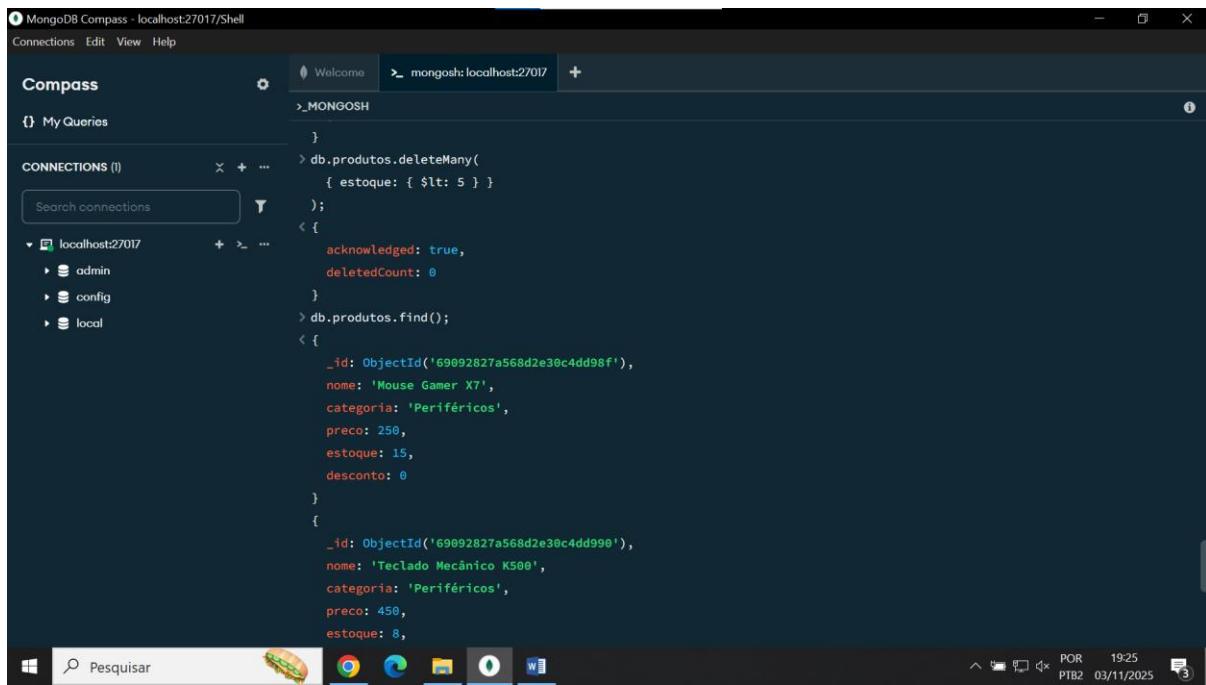
The screenshot shows the MongoDB Compass application interface. On the left, there's a sidebar titled 'Compass' with sections for 'My Queries' and 'CONNECTIONS'. Under 'CONNECTIONS', it lists 'localhost:27017' which is currently selected, showing connections to 'admin', 'config', and 'local' databases. The main area is a code editor with the title 'mongosh: localhost:27017'. It contains the following MongoDB shell script:

```
> db.produtos.replaceOne(
  { nome: "Console Zeta" },
  {
    nome: "Console Zeta",
    preco: 3500,
    garantia: "2 anos"
  );
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
> db.produtos.find({ nome: "Console Zeta" });
< [
  {
    _id: ObjectId('69092827a568d2e30c4dd992'),
    nome: 'Console Zeta',
    preco: 3500,
    garantia: '2 anos'
  }
]
rede_games;
```

Questão 5 – Remoção Condisional Remova todos os produtos cujo estoque seja menor que 5

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

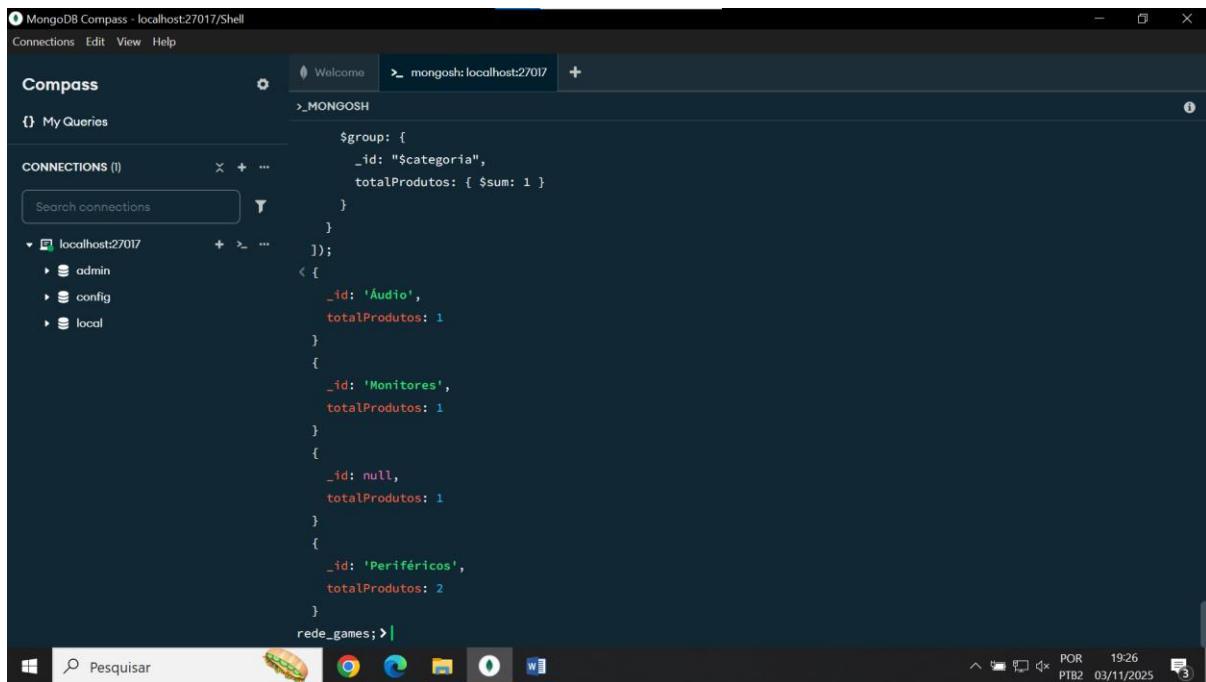
```
db.produtos.find();
```



```
>_MONGOSH
}
> db.produtos.deleteMany(
  { estoque: { $lt: 5 } }
);
< {
  acknowledged: true,
  deletedCount: 0
}
> db.produtos.find();
< [
  {
    _id: ObjectId('69092827a568d2e30c4dd98f'),
    nome: 'Mouse Gamer X7',
    categoria: 'Periféricos',
    preco: 250,
    estoque: 15,
    desconto: 0
  },
  {
    _id: ObjectId('69092827a568d2e30c4dd990'),
    nome: 'Teclado Mecânico K500',
    categoria: 'Periféricos',
    preco: 450,
    estoque: 8,
    desconto: 0
  }
]
```

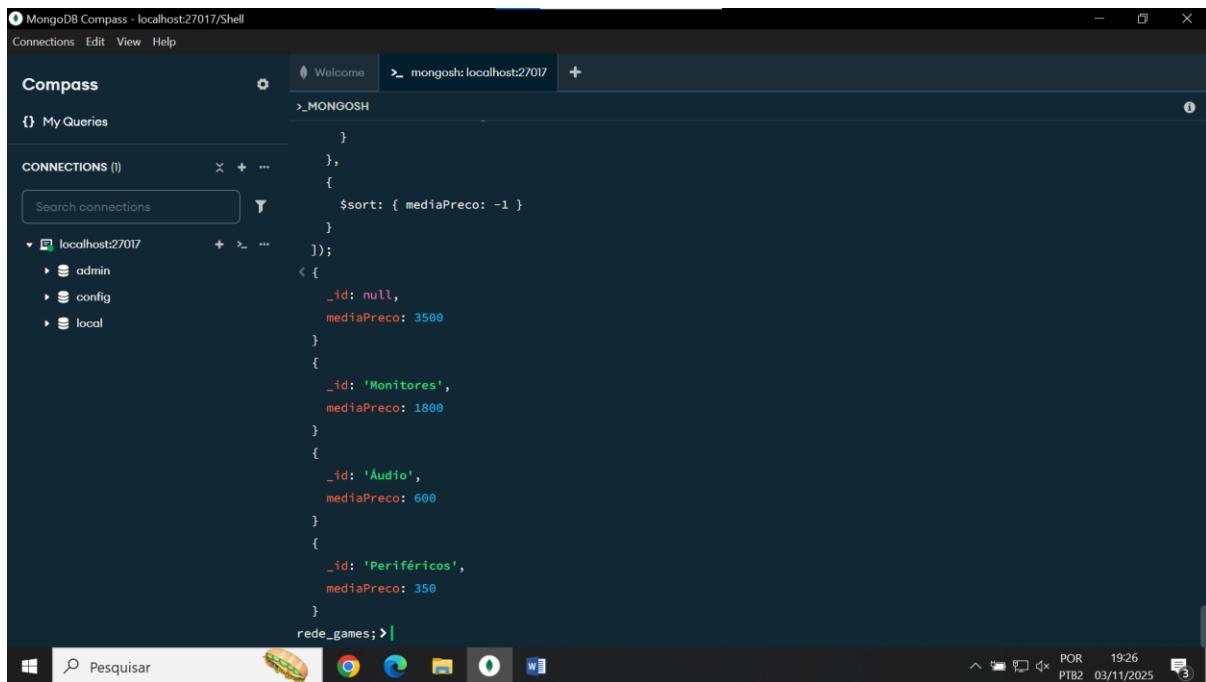
Questão 6 – Agregação e Contagem Conte quantos produtos existem em cada categoria.

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



Questão 7 – Média de Preços por Categoria Calcule o preço médio dos produtos de cada categoria e ordene do maior para o menor.

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

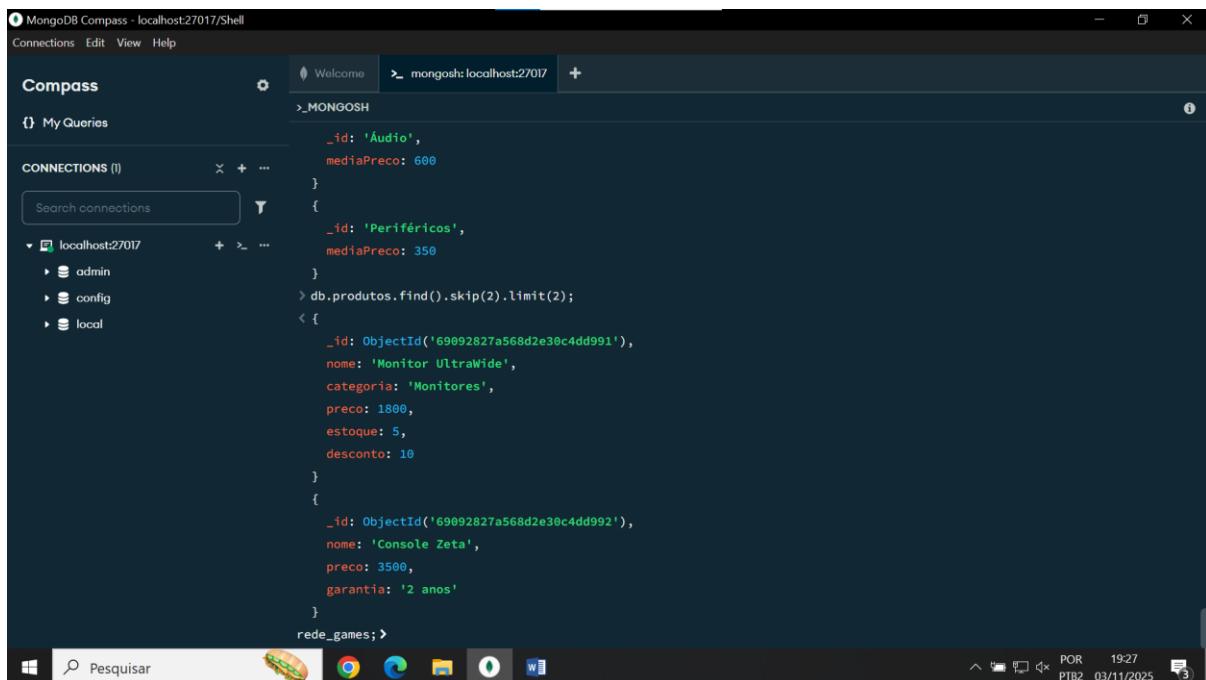


The screenshot shows the MongoDB Compass interface connected to 'localhost:27017'. In the 'mongosh: localhost:27017' tab, a query is being run against the 'rede_games' database. The query is:

```
_id: null,
mediaPreco: 3500
{
  _id: 'Monitores',
  mediaPreco: 1800
}
{
  _id: 'Áudio',
  mediaPreco: 600
}
{
  _id: 'Periféricos',
  mediaPreco: 350
}
rede_games;
```

Questão 8 – Paginação Liste os produtos exibindo apenas 2 resultados por página, mostrando a segunda página.

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



The screenshot shows the MongoDB Compass interface connected to 'localhost:27017'. In the 'mongosh: localhost:27017' tab, a query is being run against the 'rede_games' database. The query is:

```
_id: 'Áudio',
mediaPreco: 600
{
  _id: 'Periféricos',
  mediaPreco: 350
}
> db.produtos.find().skip(2).limit(2);
< [
  {
    _id: ObjectId('69092827a568d2e30c4dd991'),
    nome: 'Monitor UltraWide',
    categoria: 'Monitores',
    preco: 1800,
    estoque: 5,
    desconto: 10
  },
  {
    _id: ObjectId('69092827a568d2e30c4dd992'),
    nome: 'Console Zeta',
    preco: 3500,
    garantia: '2 anos'
  }
]
rede_games;
```

Questão 9 – Indexação e Otimização Crie um índice no campo "categoria" e, em seguida, verifique se está sendo utilizado nas consultas.

```
db.produtos.createIndex({ categoria: 1 });
```

```
db.produtos.find(  
  { categoria: "Periféricos" }  
) .explain("executionStats");
```

The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists 'localhost:27017' with three databases: 'admin', 'config', and 'local'. The main area displays the output of a command executed against the '_MONGOSH' connection:

```
$db: 'rede_games';  
},  
serverInfo: {  
  host: 'DESKTOP-I3FQSDE',  
  port: 27017,  
  version: '8.0.5',  
  gitVersion: 'cb9e2e5e552ee39deale39d7859336456d0c9820'  
},  
serverParameters: {  
  internalQueryFacetBufferSizeBytes: 104857600,  
  internalQueryFacetMaxOutputDocSizeBytes: 104857600,  
  internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,  
  internalDocumentSourceGroupMaxMemoryBytes: 104857600,  
  internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,  
  internalQueryProhibitBlockingMergeOnMongoS: 0,  
  internalQueryMaxAddToSetBytes: 104857600,  
  internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600,  
  internalQueryFrameworkControl: 'trySbeRestricted',  
  internalQueryPlannerIgnoreIndexWithCollationForRegex: 1  
},  
ok: 1  
}  
rede_games,>
```

Questão 10 – Integração com API Crie uma rota em Node.js/Express que exiba todos os produtos com apenas nome e preço, limitando a 5 resultados

```
import express, { Request, Response } from 'express';  
import { MongoClient, Collection } from 'mongodb';  
  
// --- Configuração ---  
const app = express();  
const port = 3000;  
  
const url = 'mongodb://localhost:27017'; // URL de conexão  
const dbName = 'rede_games'; // Nome do banco de dados  
  
// --- Interface para tipar nosso resultado ---  
// Define que o objeto terá APENAS nome e preço  
interface ProdutoProjetado {  
  nome: string;  
  preco: number;  
}
```

```
// --- Função Principal de Conexão ---
async function main() {
    const client = new MongoClient(url);

    try {
        // Conecta ao MongoDB
        await client.connect();
        console.log('Conectado ao servidor MongoDB');

        const db = client.db(dbName);
        const collection: Collection = db.collection('produtos');

        // --- Definição da Rota da API ---
        app.get('/produtos', async (req: Request, res: Response) => {
            try {
                // Usamos a tipagem <ProdutoProjetado> para o .project()
                const produtos = await collection
                    .find()
                    .project<ProdutoProjetado>({ nome: 1, preco: 1, _id: 0 }) // 2.
Mostra apenas nome e preço
                    .limit(5) // 3. Limita a 5 resultados
                    .toArray();

                res.json(produtos);

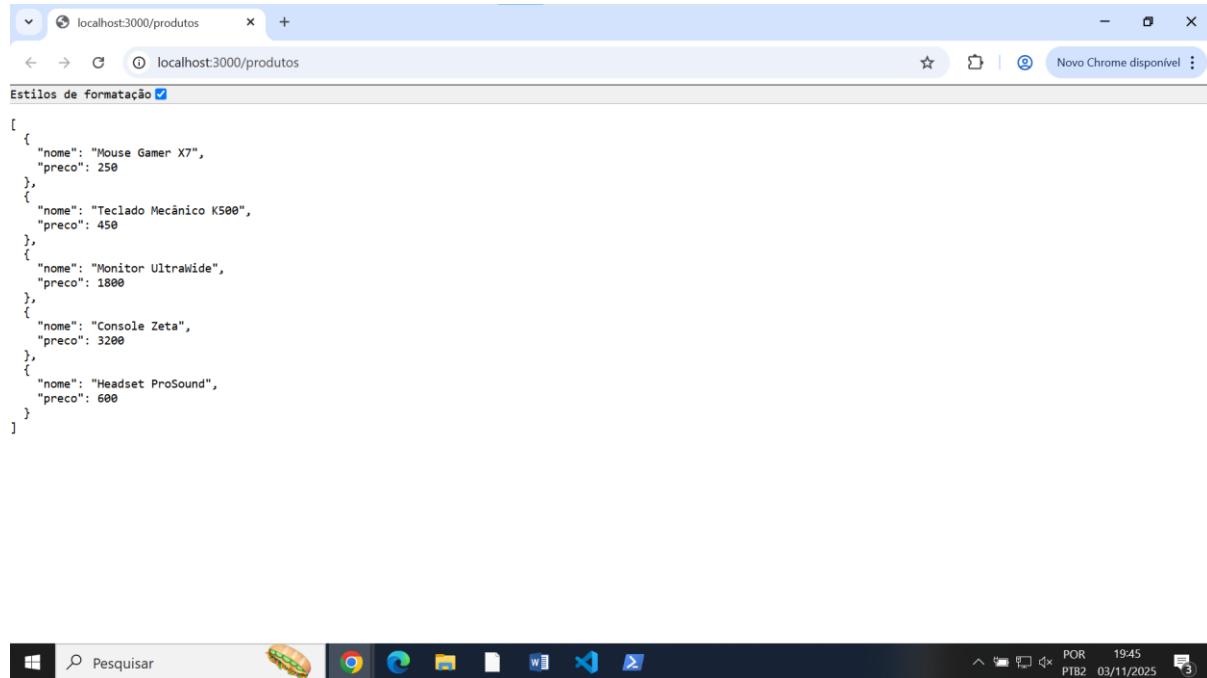
            } catch (err) {
                // Tratamento de erro para a rota
                if (err instanceof Error) {
                    res.status(500).send({ error: 'Falha ao buscar produtos', details: err.message });
                } else {
                    res.status(500).send({ error: 'Ocorreu um erro desconhecido.' });
                }
            }
        });

        // Inicia o servidor Express
        app.listen(port, () => {
            console.log(`API da RedeGames (TS) rodando em
http://localhost:\${port}/produtos`);
        });

    } catch (err) {
        // Tratamento de erro da conexão principal
        console.error('Falha ao conectar ao MongoDB', err);
    }
}
```

```
        process.exit(1); // Encerra a aplicação se não conseguir conectar
    }
}

// Executa a função principal
main();
```



```
[{"nome": "Mouse Gamer X7", "preco": 250}, {"nome": "Teclado Mecânico K500", "preco": 450}, {"nome": "Monitor UltraWide", "preco": 1800}, {"nome": "Console Zeta", "preco": 3200}, {"nome": "Headset ProSound", "preco": 600}]
```