Felipe Ribeiro de Lima

3 DSM

Banco de Dados não relacional

Atividade 1

Tarefa: Proponha a modelagem em MongoDB que melhor se adequa ao cenário e justifique usando os conceitos das Aulas 1 e 2

A melhor opção é utilizar o padrão **Embedding**, incorporando o *array* de manutenções dentro do documento principal do carro.

- **Justificativa:** O desempenho é a prioridade. O Embedding permite recuperar o carro e **todo** o seu histórico de manutenção em **uma única consulta** (sem a necessidade de *joins*), o que é ideal para **análises rápidas**. Embora o número de manutenções seja grande, o volume de dados de cada manutenção (data, tipo, mecânico) é pequeno o suficiente para caber dentro do limite de 16MB do documento MongoDB.

QUESTÃO 2 — Inserção de Dados (CRUD / Aula 3) Insira 3 registros de telemetria na coleção leituras contendo: carro, sensor, valor e data/hora.

```
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('691bab0f8f9c5efab969a680'),
    '1': ObjectId('691bab0f8f9c5efab969a681'),
    '2': ObjectId('691bab0f8f9c5efab969a682')
  }
}
```

QUESTÃO 3 — Consultas com Operadores Lógicos (Aula 4) Liste todas as leituras onde: • o sensor seja “temperatura_motor” OU “pressao_oleo”, • E o valor seja maior que 90.

```
> db.leituras.find({
    $or: [
      { "sensor": "temperatura_motor" },
      { "sensor": "pressao_oleo" }
    ],
    "valor": { $gt: 90 }
  });
< {
    _id: ObjectId('691bab0f8f9c5efab969a680'),
    carro: 'GT-R',
    sensor: 'temperatura_motor',
    valor: 95.2,
    data_hora: 2025-11-17T18:00:00.000Z
  }
```

QUESTÃO 4 — Atualização Avançada (Aula 4) O carro “GT-R” teve uma falha no sensor de temperatura. Atualize todas as leituras desse carro adicionando o campo: "status_sensor": "verificar" E remova o campo codigo_defeito, caso exista.

```
> db.leituras.updateMany(
    { "carro": "GT-R" }, // Filtro
    {
      $set: { "status_sensor": "verificar" },
      $unset: { "codigo_defeito": "" }
    }
  );
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 2,
    modifiedCount: 2,
    upsertedCount: 0
  }
```

QUESTÃO 5 — Paginação (Aula 4)

Liste as 5 leituras mais recentes do sensor

“velocidade”, ignorando as primeiras 10.

```
> db.leituras.updateMany(
    { "carro": "GT-R" }, // Filtro
    {
      $set: { "status_sensor": "verificar" },
      $unset: { "codigo_defeito": "" }
    }
  );
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 2,
    modifiedCount: 2,
    upsertedCount: 0
  }
> db.leituras.find({
    "sensor": "velocidade"
  })
  .sort({ "data_hora": -1 })
  .skip(10)
  .limit(5);
```

QUESTÃO 6 — Agregação (Aula 6)

Calcule a média de temperatura do motor por carro,

ordenando os maiores valores primeiro.

```
db.leituras.find({
  "sensor": "velocidade"
})
.sort({ "data_hora": -1 })
.skip(10)
.limit(5);

db.leituras.aggregate([
  {

    $match: { "sensor": "temperatura_motor" }
  },
  {

    $group: {
      "_id": "$carro",
      "media_temperatura": { $avg: "$valor" }
    }
  },
  {

    $sort: { "media_temperatura": -1 }
  }
]);
{
  _id: 'GT-R',
  media_temperatura: 95.2
}
```

QUESTÃO 7 — API Node.js (Aula 7)

A equipe precisa registrar uma nova leitura via API

REST.

Crie a rota POST /leituras usando Express + Mongoose

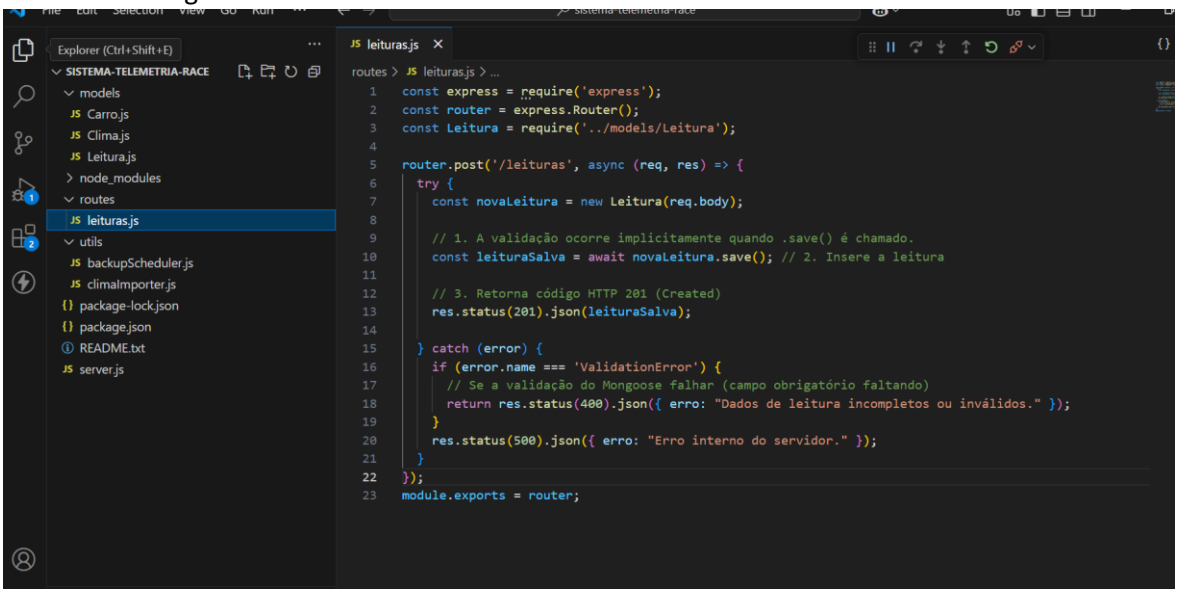
que:

1. Valida se os campos obrigatórios foram

enviados

2. Insere a leitura

3. Retorna código HTTP 201



QUESTÃO 8 — Consumo de API Externa (Aula 7)

A equipe deseja importar automaticamente a

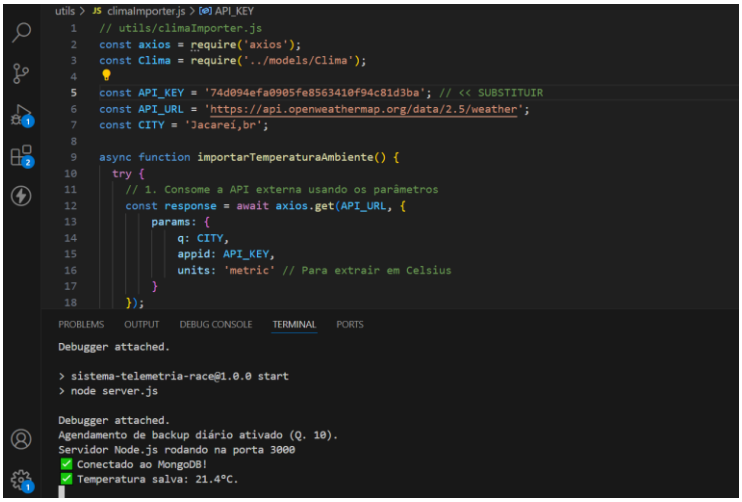
temperatura ambiente antes das corridas.

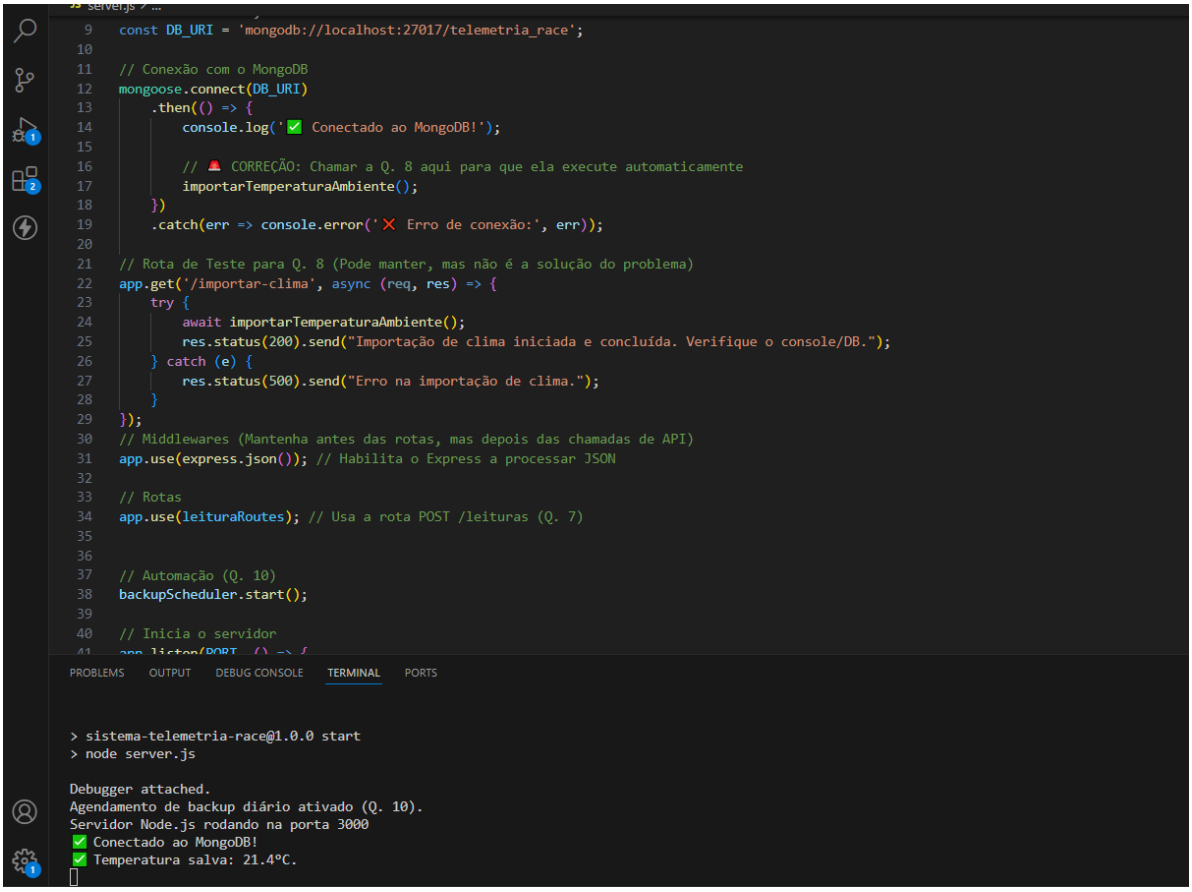
Crie uma função Node.js usando axios que:

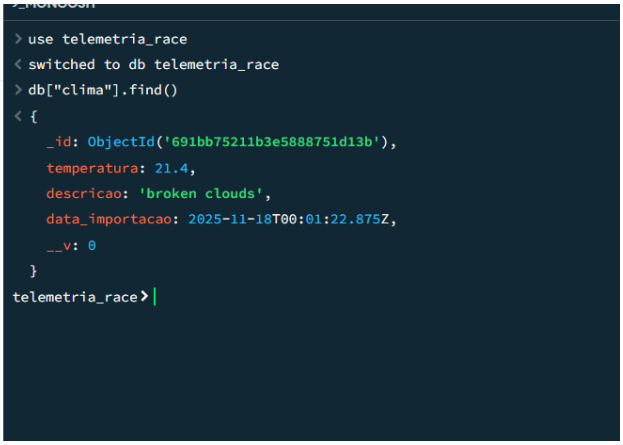
• consome a API externa,

• extrai temp,

• salva na coleção clima.







QUESTÃO 9 — Segurança no MongoDB (Aula 8)

Crie um usuário chamado engenheiroCorrida com acesso apenas de leitura ao banco telemetria_race.

```
> use admin

  db.createUser({
    user: "engenheiroCorrida",
    pwd: "SENHA_SEGURA_DO_ENGENHEIRO",
    roles: [
      { role: "read", db: "telemetria_race" }
    ]
  });
< switched to db admin
admin>
```

QUESTÃO 10 — Backup e Restauração (Aula 8)

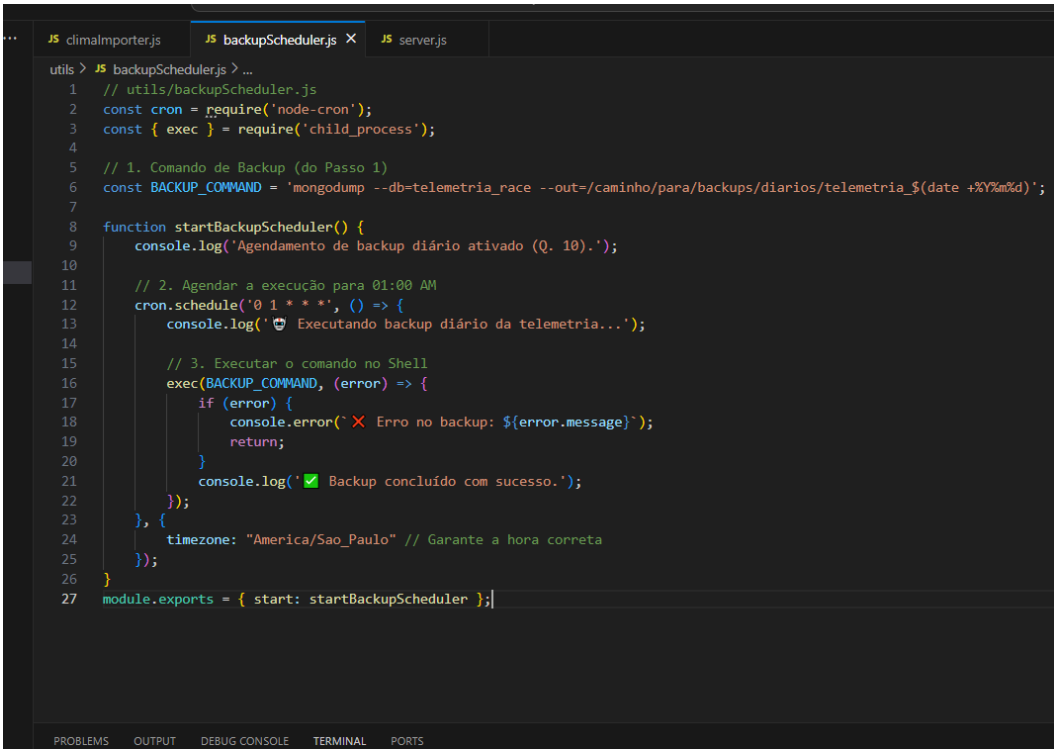
A equipe quer fazer backup diário dos dados de telemetria.

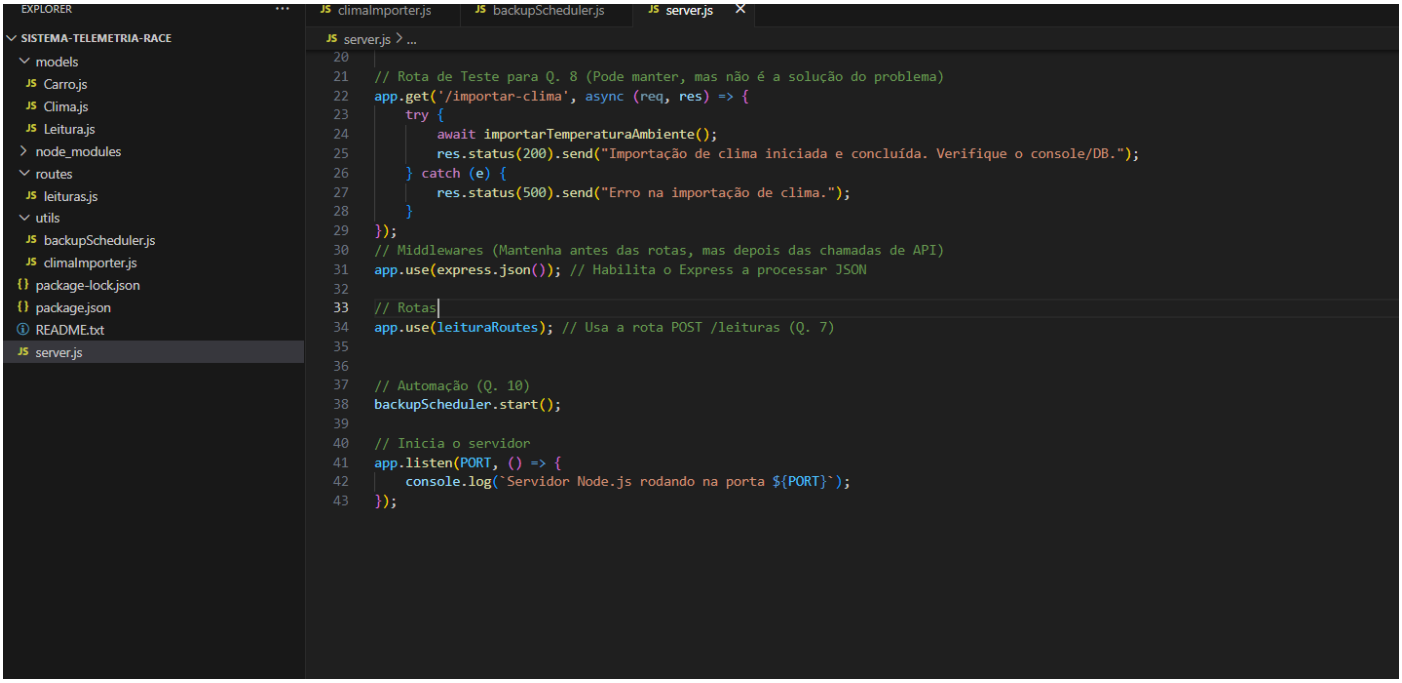
Crie o comando de backup e um exemplo de automação com node-cron.

baixar a **Ferramenta:** Use `mongodump`.

Rodar

mongodump --db="telemetria_race" --out="/caminho/para/backups/diarios/telemetria_$(date +%Y%m%d)"





Ao iniciar o servidor com `npm run dev`, a mensagem **"Agendamento de backup diário ativado (Q. 10)."** aparecerá no console, confirmando que a rotina de automação está pronta para ser executada à 01:00 AM todos os dias.