

Questão 1

Resposta: A melhor modelagem para o caso é Referência (Referencing). Onde teríamos a coleção carro, na qual contém informações principais dos veículos, com campos: carro, sensor, valor, data_hora, identificação (carro_id). E também a coleção de manutencoes, onde armazena registros de manutencoes realizadas, onde possui o id do carro (carro_id) que será utilizado para relacionar.

Justificativa: Como o número de manutenções por carro pode aumentar muito, o Referencing é mais adequado para coleções com volumes grandes. Além de facilitar as buscas com foco nas manutenções, sendo mais eficiente do que caso fosse Embedding, onde é mais interessante para visualizar ambos (carro e manutenções).

Questão 2

```
db.leituras.insertMany([
  {
    "carro": "GT-R",
    "sensor": "temperatura_motor",
    "valor": 95.5,
    "data_hora": new Date()
  },
  {
    "carro": "Mustang-T",
    "sensor": "pressao_oleo",
    "valor": 45.2,
    "data_hora": new Date()
  },
  {
    "carro": "GT-R",
    "sensor": "velocidade",
    "valor": 280,
    "data_hora": new Date()
  }
])
```

```

> db.leituras.insertMany([
  {
    "carro": "GT-R",
    "sensor": "temperatura_motor",
    "valor": 95.5,
    "data_hora": new Date()
  },
  {
    "carro": "Mustang-T",
    "sensor": "pressao_oleo",
    "valor": 45.2,
    "data_hora": new Date()
  },
  {
    "carro": "GT-R",
    "sensor": "velocidade",
    "valor": 280,
    "data_hora": new Date()
  }
])
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('692dbc11c4a66aeff4c3a466'),
    '1': ObjectId('692dbc11c4a66aeff4c3a467'),
    '2': ObjectId('692dbc11c4a66aeff4c3a468')
  }
}

```

Questão 3

```

db.leituras.find({
  $and: [
    {
      $or: [
        { "sensor": "temperatura_motor" },
        { "sensor": "pressao_oleo" }
      ]
    },
    { "valor": { $gt: 90 } }
  ]
})

```

}}

```
> db.leituras.find({
  $and: [
    {
      $or: [
        { "sensor": "temperatura_motor" },
        { "sensor": "pressao_oleo" }
      ]
    },
    { "valor": { $gt: 90 } }
  ]
})
< {
  _id: ObjectId('692dbc11c4a66a6ff4c3a466'),
  carro: 'GT-R',
  sensor: 'temperatura_motor',
  valor: 95.5,
  data_hora: 2025-12-01T16:02:25.178Z
}
```

Questão 4

```
db.leituras.updateMany(
  { "carro": "GT-R" },
  {
    $set: { "status_sensor": "verificar" },
    $unset: { "codigo_defeito": "" }
  }
)
```

```

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

```

Questão 5

Adicionando mais registros para demonstrar:

```

db.leituras.insertMany([
  // Dados para Paginação (velocidade - 10 registros)
  {
    "carro": "Mustang-T",
    "sensor": "velocidade",
    "valor": 290,
    "data_hora": new Date("2025-11-20T10:00:01Z")
  },
  {
    "carro": "GT-R",
    "sensor": "velocidade",
    "valor": 295,
    "data_hora": new Date("2025-11-20T10:00:02Z")
  },
  {
    "carro": "Mustang-T",
    "sensor": "velocidade",
    "valor": 285,
    "data_hora": new Date("2025-11-20T10:00:03Z")
  },
  {
    "carro": "GT-R",
    "sensor": "velocidade",
    "valor": 305,
    "data_hora": new Date("2025-11-20T10:00:04Z")
  }
])

```

```
},
{
  "carro": "Mustang-T",
  "sensor": "velocidade",
  "valor": 270,
  "data_hora": new Date("2025-11-20T10:00:05Z")
},
{
  "carro": "GT-R",
  "sensor": "velocidade",
  "valor": 310,
  "data_hora": new Date("2025-11-20T10:00:06Z")
},
{
  "carro": "Mustang-T",
  "sensor": "velocidade",
  "valor": 288,
  "data_hora": new Date("2025-11-20T10:00:07Z")
},
{
  "carro": "GT-R",
  "sensor": "velocidade",
  "valor": 292,
  "data_hora": new Date("2025-11-20T10:00:08Z")
},
{
  "carro": "Mustang-T",
  "sensor": "velocidade",
  "valor": 275,
  "data_hora": new Date("2025-11-20T10:00:09Z")
},
{
  "carro": "GT-R",
  "sensor": "velocidade",
  "valor": 300,
  "data_hora": new Date("2025-11-20T10:00:10Z")
},
},
```

```
// Dados Adicionais para Agregação (temperatura_motor)
// 3 registros de temperatura para GT-R e 2 para Mustang-T
```

```
{
  "carro": "GT-R",
  "sensor": "temperatura_motor",
  "valor": 98.2,
  "data_hora": new Date("2025-11-20T10:00:11Z")
},
{
  "carro": "Mustang-T",
```

```

        "sensor": "temperatura_motor",
        "valor": 93.1,
        "data_hora": new Date("2025-11-20T10:00:12Z")
    },
    {
        "carro": "GT-R",
        "sensor": "temperatura_motor",
        "valor": 99.0,
        "data_hora": new Date("2025-11-20T10:00:13Z")
    },
    {
        "carro": "Mustang-T",
        "sensor": "temperatura_motor",
        "valor": 92.5,
        "data_hora": new Date("2025-11-20T10:00:14Z")
    },
    {
        "carro": "GT-R",
        "sensor": "temperatura_motor",
        "valor": 97.5,
        "data_hora": new Date("2025-11-20T10:00:15Z")
    },
    {
        "carro": "GT-R",
        "sensor": "velocidade",
        "valor": 260,
        "data_hora": new Date("2025-11-20T09:00:00Z")
    },
    {
        "carro": "Mustang-T",
        "sensor": "velocidade",
        "valor": 255,
        "data_hora": new Date("2025-11-20T09:00:01Z")
    },
    {
        "carro": "GT-R",
        "sensor": "velocidade",
        "valor": 277,
        "data_hora": new Date("2025-11-20T09:00:02Z")    },
    {
        "carro": "Mustang-T",
        "sensor": "velocidade",
        "valor": 265,
        "data_hora": new Date("2025-11-20T09:00:03Z")
    }
]

```

```

>_MONGOSH

    },
    {
      "carro": "GT-R",
      "sensor": "temperatura_motor",
      "valor": 97.5,
      "data_hora": new Date("2025-11-20T10:00:15Z")
    }
  ])
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('692dbd0cc4a66aeff4c3a469'),
    '1': ObjectId('692dbd0cc4a66aeff4c3a46a'),
    '2': ObjectId('692dbd0cc4a66aeff4c3a46b'),
    '3': ObjectId('692dbd0cc4a66aeff4c3a46c'),
    '4': ObjectId('692dbd0cc4a66aeff4c3a46d'),
    '5': ObjectId('692dbd0cc4a66aeff4c3a46e'),
    '6': ObjectId('692dbd0cc4a66aeff4c3a46f'),
    '7': ObjectId('692dbd0cc4a66aeff4c3a470'),
    '8': ObjectId('692dbd0cc4a66aeff4c3a471'),
    '9': ObjectId('692dbd0cc4a66aeff4c3a472'),
    '10': ObjectId('692dbd0cc4a66aeff4c3a473'),
    '11': ObjectId('692dbd0cc4a66aeff4c3a474'),
    '12': ObjectId('692dbd0cc4a66aeff4c3a475'),
    '13': ObjectId('692dbd0cc4a66aeff4c3a476'),
    '14': ObjectId('692dbd0cc4a66aeff4c3a477')
  }
}

```

```

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

```

```

> db.leituras.find({ "sensor": "velocidade" })
  .sort({ "data_hora": -1 })
  .skip(10)
  .limit(5)
< {
  _id: ObjectId('692dbd0cc4a66aeff4c3a469'),
  carro: 'Mustang-T',
  sensor: 'velocidade',
  valor: 290,
  data_hora: 2025-11-20T10:00:01.000Z
}
{
  _id: ObjectId('692dbe03c4a66aeff4c3a47b'),
  carro: 'Mustang-T',
  sensor: 'velocidade',
  valor: 265,
  data_hora: 2025-11-20T09:00:03.000Z
}
{
  _id: ObjectId('692dbe03c4a66aeff4c3a47a'),
  carro: 'GT-R',
  sensor: 'velocidade',
  valor: 277,
  data_hora: 2025-11-20T09:00:02.000Z
}
{
  _id: ObjectId('692dbe03c4a66aeff4c3a479'),
  carro: 'Mustang-T',
  sensor: 'velocidade',

```

Questão 6

```

db.leituras.aggregate([
  {
    $match: { "sensor": "temperatura_motor" } // Filtra apenas leituras de temperatura do motor
  },
  {
    $group: {
      "_id": "$carro", // Agrupa pelo campo 'carro'
      "mediaTemperatura": { $avg: "$valor" } // Calcula a média do campo 'valor'
    }
  }
])

```



```

    },
    {
      $sort: { "mediaTemperatura": -1 } // Ordena a média em ordem decrescente (maior
primeiro)
    }
  ]
})

```

```

> db.leituras.aggregate([
  {
    $match: { "sensor": "temperatura_motor" }
  },
  {
    $group: {
      "_id": "$carro",
      "mediaTemperatura": { $avg: "$valor" }
    }
  },
  {
    $sort: { "mediaTemperatura": -1 }
  }
])
< {
  _id: 'GT-R',
  mediaTemperatura: 97.55
}
{
  _id: 'Mustang-T',
  mediaTemperatura: 92.8
}

```

Questão 7

```

const express = require('express');
const router = express.Router();
const Leitura = require('../model/leitura');

router.post('/leituras', async (req, res) => {
  try {
    const { carro, sensor, valor, data_hora } = req.body;

    if (!carro || !sensor || valor === undefined) {

```

```

        return res.status(400).json({ error: "Campos 'carro',
'sensor' e 'valor' são obrigatórios." });
    }

    const novaLeitura = new Leitura({
        carro,
        sensor,
        valor,
        data_hora: data_hora || new Date()
    });

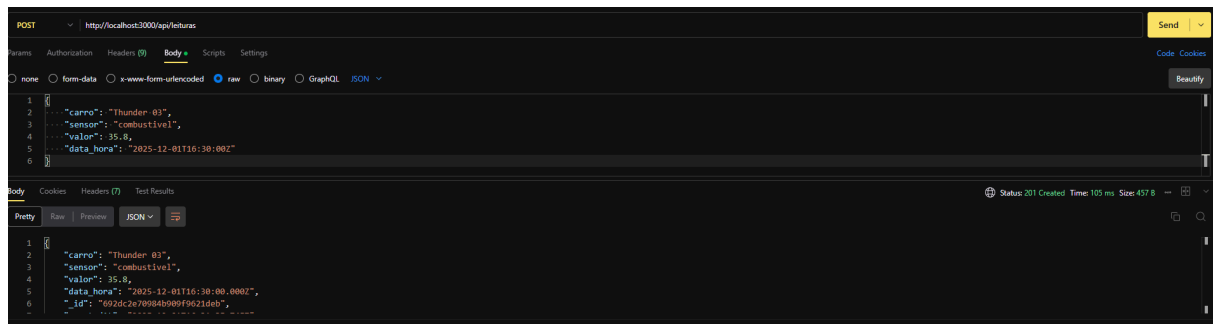
    await novaLeitura.save();

    res.status(201).json(novaLeitura);

} catch (error) {
    console.error(error);
    res.status(500).json({ error: "Erro ao registrar leitura: " +
error.message });
}
});

module.exports = router;

```



Questão 8

```

Buscando dados de clima para Jacarei,br...
Registro de clima salvo com sucesso: {
  local: 'Jacarei,br',
  temperatura: 31.4,
  unidade: 'Celsius',
  data_registro: 2025-12-01T16:43:57.181Z,
  descricao: 'few clouds',
  _id: new ObjectId('692dc5cd3498e86c734ba63d'),
  __v: 0
}

```

localhost:27017 > revisaobd3 > climas Open MongoDB shell

Documents 1 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#) Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE 25 1 - 1 of 1 ⌂ {} ⌂

```

_id: ObjectId('692dc5cd3498e86c734ba63d')
local: "Jacarei,br"
temperatura: 31.4
unidade: "Celsius"
data_registro: 2025-12-01T16:43:57.181+00:00
descricao: "few clouds"
__v: 0

```

Questão 9

use revisaobd3

db.createUser({ user: "engenheiroCorrida", pwd: "123", roles: [{ role: "read", db: "revisaobd3"}] })

```

> db.createUser({ user: "engenheiroCorrida", pwd: "123", roles: [ { role: "read", db: "revisaobd3"} ] })
< { ok: 1 }

```

Questão 10

```

JS index.js X
src > JS index.js > ...
5  const backupPath = path.join(__dirname, 'backups', `telemetria_backup_${Date.now()}`);
6  const dbName = 'revisaobd3';
7
8  const backupCommand = `mkdir -p "${backupPath}" && mongodump --db ${dbName} --out "${backupPath}"`;
9
10
11  cron.schedule('00 22 * * *', () => {
12    console.log(`Iniciando backup diário da telemetria (${dbName}) às 22:00...`);
13
14    exec(backupCommand, (error, stdout, stderr) => {
15      if (error) {
16        console.error(`Erro no backup: ${error.message}`);
17        return;
18      }
19      if (stderr) {
20        console.error(`Erro (stderr) no backup: ${stderr}`);
21        return;
22      }
23      console.log(`Backup concluído com sucesso em: ${backupPath}`);
24      console.log(stdout);
25    });
26  }, {
27    scheduled: true,
28    timezone: "America/Sao_Paulo"
29  });
30
31  console.log('Agendador de backup iniciado. Próximo backup: 22:00.');
```

