

**Prova III – Banco de Dados não Relacional – DSM – Prof.ª Lucineide – XX/XX/2025**

Nome: \_\_\_\_\_

Valor: 2,0 pts. cada questão

**PROVA PRÁTICA – P3**

**Formato:** estilo ENADE, com estudo de caso, situações-problema e prática real.

**Cobertura:** Aulas 1 a 8 (MongoDB completo, Node.js, APIs, segurança, agregação, modelagem, CRUD).

**TEMA OFICIAL DA PROVA:**

**Sistema Inteligente de Monitoramento Veicular para Competições Automotivas**

*(unindo monitoramento esportivo + mecânica inteligente, excelente para o perfil dos alunos)*

**ESTUDO DE CASO (ENADE – Contexto Inicial)**

A equipe **ThunderRacing**, especializada em competições automobilísticas de alta performance, está desenvolvendo um **Sistema Inteligente de Monitoramento Veicular** baseado em **sensores de telemetria, dados mecânicos, clima, e registro de eventos críticos** (falhas, superaquecimento, desgaste, consumo de combustível).

O sistema utiliza:

- **MongoDB** como base de dados NoSQL
- **Node.js + Express + Mongoose** para a API
- Importação opcional de clima por API externa
- Operadores avançados, agregações e verificações de segurança
- Usuários com diferentes permissões
- Backups automáticos

Os engenheiros pediram sua ajuda para **modelar, consultar, integrar, atualizar e proteger** os dados do sistema.

Com base nesse cenário, responda às questões a seguir.

**QUESTÃO 1 — Modelagem (Embedding x Referencing)**

A equipe deseja armazenar informações dos carros e também **histórico de manutenções**, contendo: data, tipo, mecânico responsável e peças trocadas.

**Situação-problema:**

A manutenção é consultada frequentemente para análises rápidas durante a corrida. Entretanto, o número de manutenções por carro pode chegar a centenas ao longo da temporada.

**Tarefa:**

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

**QUESTÃO 2 — Inserção de Dados (CRUD / Aula 3)**

Insira 3 registros de telemetria na coleção leituras contendo: carro, sensor, valor e data/hora.

**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.

**Prova III – Banco de Dados não Relacional – DSM – Prof.<sup>a</sup> Lucineide – XX/XX/2025****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.

**QUESTÃO 5 — Paginação (Aula 4)**

Liste as **5 leituras mais recentes** do sensor “velocidade”, ignorando as primeiras 10.

**QUESTÃO 6 — Agregação (Aula 6)**

Calcule a **média de temperatura do motor por carro**, ordenando os maiores valores primeiro.

**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`.

**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.

**Instruções de Entrega:**

- Entregar arquivo (único) com os **scripts e prints (tela inteira)** de cada questão.
- A entrega deve ser feita em **formato PDF**.
- O arquivo deve ser nomeado como *Prova03\_NomeCompleto.pdf*
- A tarefa estará disponível no **GitHub Pessoal**