

Curso:	Ciência de Dados e Inteligência Artificial
Disciplina:	Projeto em Ciência de Dados II
Professora:	Tamara Simões

**OBS.: ESTA É A AP1 APLICADA EM 2025/1.**

**LEIA COM ATENÇÃO AS ORIENTAÇÕES PARA A PROVA:**

- Esta prova é prática e tem a duração de 90 min.
- Todas as orientações para obter acesso aos arquivos base estão no repositório do GitHub.
- Logo após o clone ou o download, **o acesso à internet será totalmente bloqueado.**
- A prova é **individual e sem consulta.**
- Comunique o fim da sua prova para que os arquivos possam ser copiados pela rede. **Os arquivos da sua prova devem estar na área de trabalho.**
- **Todos os computadores estão sendo monitorados.**
- **É expressamente proibido** o uso de qualquer dispositivo eletrônico com armazenamento de memória, inclusive relógios inteligentes.
- **Não é permitido empréstimo** de nenhum material.
- Qualquer tentativa de *cola* levará à **anulação da prova** e o(a) aluno(a) receberá as **sanções cabíveis.**
- A tolerância para entrada do aluno em sala é de **30 minutos**. A saída do aluno da sala é permitida **após 40 minutos**.
- **A interpretação faz parte da prova. Portanto, não serão respondidas dúvidas durante sua realização.** Todas as informações necessárias estão contidas nos enunciados.
- **Todos os arquivos já estão criados. Arquivos fora do escopo serão desconsiderados.**
- Ao fim da prova, comunique à professora para que os arquivos possam ser copiados. **Verifique atentamente se todos os arquivos foram salvos, pois arquivos vazios zeram a pontuação da questão.**
- Resolva as questões utilizando, **apenas**, as ferramentas abordadas em sala de aula. **Resoluções que contenham outros métodos serão desconsiderados!**

## QUESTÃO 1 - Tema: TypeScript e HOFs

---

Você foi contratado(a) para desenvolver um sistema que gerencia os pedidos de uma loja. Para isso, siga as instruções abaixo no arquivo **questao2.ts**:

- a) Crie um tipo *Pedido*, que deve conter as seguintes propriedades: id (número), cliente (string), valor (número), entregue (boolean).
- b) Implemente uma função chamada *filtrarPedidosEntregues*, que recebe um array de *Pedido* e retorna um apenas os pedidos que já foram entregues (entregue: true). Não é permitido utilizar *for*, *for in* ou *for of*.
- c) Crie uma função explicitamente tipada chamada *calcularTotalPedidos*, que recebe um array de *Pedido* e retorna a soma dos valores de todos os pedidos. Não é permitido utilizar *for*, *for in* ou *for of*.
- d) Crie um novo tipo *PedidoSemId*, que deve conter todas as propriedades de *Pedido*, exceto id.

## QUESTÃO 2 - Tema: API Node + Express e HOFs

---

Você foi contratado para desenvolver uma API de catálogo de filmes utilizando Express, que manipule dados do arquivo **questao2\_data.json**. A API deve conter quatro rotas que atendem aos seguintes requisitos, no arquivo **questao2.js** (utilize apenas HOFs em caso de necessidade de laços de repetição):

- a) Implemente a rota **/filmes**, que retorna com o status 200 uma lista de filmes em memória. Permita que os usuários filtrem os filmes pelo título utilizando um como parâmetro o título do filme.
- b) Implemente a rota **/filmes/:id**, que retorna um filme específico pelo seu ID (também armazenado apenas em memória). Se o filme não for encontrado, a rota deve retornar um erro 404 com a mensagem *Filme fora do catálogo*.
- c) Implemente a rota **/filmes**, que recebe um novo filme via body e o salva em **questao2\_data.json**. Retorne o status 201, acompanhado de uma mensagem confirmando o cadastro.
- d) Implemente a rota **/filmes/:id**, que atualiza um filme existente pelo ID, modificando suas informações no arquivo **questao2\_data.json**. Caso o filme não exista, a rota deve retornar um erro 404, também com a mensagem *Filme fora do catálogo*.