

Atividade de Avaliação - Cadastro de Filmes

📌 Desenvolvimento de uma Aplicação de Catálogo de Filmes (CRUD Fullstack)

O projeto deverá ser desenvolvido de forma individual ou em dupla. No dia da entrega os alunos deverão estar presencialmente na aula para fazer a defesa do seu projeto, onde serão submetidos a questionamentos sobre o código desenvolvido.

Objetivo

Avaliar a capacidade de:

- ✓ Criar um **backend em NodeJS (Express ou NestJS)** com API RESTful
- ✓ Utilizar um **banco de dados relacional (PostgreSQL, MariaDB ou MySQL)**
- ✓ Criar um **frontend em ReactJS** para consumir a API
- ✓ Implementar **CRUD completo (Create, Read, Update, Delete)**

Descrição do Desafio

Você irá construir um **Catálogo de Filmes** onde o usuário poderá:

- **Cadastrar um novo filme** (título, descrição, ano de lançamento, gênero, URL de imagem do pôster)
- **Listar todos os filmes cadastrados**
- **Visualizar detalhes de um filme**
- **Editar informações de um filme**
- **Deletar um filme**

Requisitos do Backend

- Desenvolver usando **NodeJS com Express ou NestJS**
- Implementar **endpoints RESTful**:
 - GET /movies – listar todos os filmes
 - GET /movies/:id – detalhes de um filme
 - POST /movies – criar um novo filme
 - PUT /movies/:id – atualizar filme
 - DELETE /movies/:id – deletar filme

- Utilizar **banco de dados (SQLite, PostgreSQL, MySQL ou MariaDB)** persistindo os dados.
- Retornar **status HTTP adequados** e mensagens claras.
- Validar dados recebidos.

Requisitos do Frontend

- Desenvolver em **ReactJS** utilizando:
 - Hooks (useState, useEffect)
 - Criar rotas para navegação entre páginas:
 - Página de listagem
 - Página de cadastro
 - Página de edição
 - Página de visualização de detalhes
 - Requisições HTTP utilizando **fetch ou axios**
 - Feedback de carregamento e erro
- Layout simples, responsivo e organizado.

Extras (bônus)

- Uso de biblioteca de componentes (Material UI, Ant Design ou Tailwind)
- Paginação e busca no frontend
- Validação de formulário no frontend
- Permitir que o usuário visualize o trailer do filme cadastrado

Entrega

- Código publicado no **Classroom** com documento com instruções de execução.
- Projeto deve funcionar **localmente ao rodar os comandos indicados na documentação do projeto**.

Rubrica de Correção

<i>Critério</i>	<i>Pontuação</i> <i>ão</i> <i>Máxima</i>	<i>Descrição</i>
<i>1. Funcionamento CRUD Completo</i>	30	Backend funcional, rotas corretas, banco funcionando, CRUD completo
<i>2. Frontend funcional e integrado ao backend</i>	25	Tela de cadastro, listagem, edição, detalhes, exclusão integradas
<i>3. Organização do Código e Boas Práticas</i>	15	Separação de responsabilidades, nomeação adequada, versionamento
<i>4. Validação e Tratamento de Erros</i>	10	Validação de dados, mensagens claras de erro, status HTTP corretos
<i>5. Uso de Hooks e Rotas corretamente</i>	10	Uso adequado de useState, useEffect, React Router
<i>6. Layout Responsivo e Usabilidade</i>	10	Layout organizado, responsivo, experiência de uso clara
<i>7. Extras</i>	10	Funcionalidades adicionais e melhorias não obrigatórias

Pontuação Máxima: 110 pontos

Critérios de Desclassificação

- ❌ Código copiado sem créditos ou sem entender
- ❌ Projeto que não executa nenhuma rota do CRUD
- ❌ Projeto entregue fora do prazo sem justificativa prévia

Por que este desafio é legal?

- ✓ Desenvolve habilidades práticas **de mercado**
- ✓ Você constrói algo que poderá usar como **portfólio**
- ✓ Exercita o ciclo completo **frontend + backend + banco**
- ✓ Prepara para entrevistas técnicas e desafios de estágio/emprego