

Universidade Estadual do Piauí - UESPI
Curso de Tecnologia em Sistemas de Computação
Disciplina: Programação Web
Professor: Eyder Rios

2ª Avaliação
Trabalho de Implementação
Data de Entrega: 16/11/2025

Tema

Catálogo Interativo de Filmes e Séries com APIs Públicas

O objetivo é desenvolver uma aplicação Web responsiva que consuma dados de uma API REST pública, exibindo filmes e séries com filtros, detalhes e interação.

Tecnologias obrigatórias:
HTML5 + CSS (puro ou Tailwind CDN) + JavaScript ES6+
Sem frameworks de JavaScript

Resumo da Aplicação

Construir um catálogo interativo filmes e séries que:

1. Lista lançamentos recentes
2. Permite paginação e filtragem
3. Mostra detalhes de filmes/séries
4. Exibe temporadas/episódios quando houver
5. Trata erros e carrega dados com feedback visual

Requisitos Funcionais (o que o sistema deve fazer)

1. Exibir os títulos lançados desde o início do ano corrente, com título, tipo, gênero, ano e imagem (limitado a uma página)
2. Implementar paginação dos resultados
3. Implementar filtro por gênero
4. Página/Seção de detalhes do título, incluindo sinopse e rating
5. Exibir temporadas e episódios quando houver
6. Lidar com campos ausentes (ex.: sem sinopse)
7. Tratar exibição de textos longos de forma adequada
8. Exibir spinner de carregamento durante requisições
9. Tratar erros da API com mensagens amigáveis
10. Página/Seção sobre os membros da equipe
11. Interface clara, responsiva, intuitiva e bem elaborada
12. Rodapé contendo os nomes dos autores do projetos

Requisitos Não Funcionais (elementos que o sistema deve incorporar)

1. HTML5 Semântico
2. Responsividade (mobile-first)
3. JavaScript modular
4. Tratamento de exceções

É recomendado aplicar técnicas aprendidas em outras disciplinas:
POO, estrutura de dados, modelagem, JSON, princípios de UX/UI, análise e tratamento de erros, boas práticas de software e documentação.

Fator de Autenticidade (FA)

Propósito

Promover desenvolvimento contínuo, evitar “entrega de última hora” e incentivar boas práticas de versionamento.

Definição

$$FA = \min(1, (C/5) \times (I/6))$$

Onde:

- C: commits válidos (conteúdo real, mensagem clara, contribuição concreta)
- I: intervalo em horas entre primeiro e último commit (mínimo 6h para FA ideal)

Exemplo

4 commits válidos

8 horas entre o primeiro e o último commit

$$\begin{aligned} FA &= \min(1, (4/5) \times (8/6)) \\ FA &= \min(1, 0.8 \times 1.33) \\ FA &= \min(1, 1.064) \\ FA &= 1 \end{aligned}$$

Commits concentrados em poucas horas ⇒ FA baixo

Evolução distribuída com mensagens descritivas ⇒ FA alto

Nota Final

$$\text{Nota Final} = \text{Nota Técnica} \times FA$$

Entrega

- Repositório GitHub público
- README.md contendo:
 - descrição do projeto
 - integrantes (até 4 membros)
 - APIs utilizadas
 - instruções de execução
- Link enviado via SIGAA
- **Data de entrega: 12/11/2025**
- **Equipes de até 4 (quatro) alunos**

Critérios de Avaliação

Nº	Critério	Peso
1	Interface + Responsividade	25%
2	Integração com API + Tratamento de erros	25%
3	JS modular + DOM + Assíncrono	20%
4	Documentação e estrutura do projeto	15%
5	Estilo, estética e criatividade	15%

Orientações finais

- Priorize qualidade e usabilidade
- Mensagens de commits explicativas e frequentes
- Seções organizadas e navegação fluida
- Lembre-se: o usuário final não é técnico
- Criatividade será valorizada