

## AVALIAÇÃO DE DESENVOLVIMENTO DE APLICAÇÃO SPA- VITE + REACT + TYPESCRIPT

- CheckPoint presencial.
- Este projeto deve ser realizado individualmente.
- Crie um projeto SPA vite+react seguindo as solicitações propostas e utilizando o Git.
- Pontuação Total: 100 pontos

### INSTRUÇÕES

Você recebeu um projeto boilerplate de uma aplicação Single Page Application (SPA) desenvolvida com vite+react e TypeScript e mais algumas imagens e ícones.

Seu objetivo criar o projeto realizando os processos básicos de inicialização do boilerplate. Em seguida inicie o desenvolvimento das tarefas descritas abaixo, utilizando os conceitos discutidos em aula. Como essa é uma avaliação sem consulta à internet ou pen-drive, você pode consultar apenas suas anotações manuscritas que são pessoais e intransferíveis.

**Nota: O array de objetos contendo os filmes já está disponível no arquivo movies.ts. Utilize-o para popular a galeria.**

TEMA: Galeria de Filmes

A aplicação deve exibir uma galeria de filmes, onde cada filme terá informações como título, descrição, ano de lançamento, gênero e uma imagem representativa (poster do filme).

### TAREFAS

#### 1. Componentização (30 pontos)

Crie ao menos 3 componentes: Header, MovieCard, e MovieGallery.

O Header deve exibir o título da aplicação.

O MovieGallery deve receber uma lista de filmes como props e renderizar um componente MovieCard para cada filme.

- Dica: Use `type` para definir os tipos das `props` do componente.

#### 2. Props e Tipagem (15 pontos)

Passar os dados de cada filme como props do MovieGallery para o MovieCard.

#### 3. Imagens (10 pontos)

Cada filme deve exibir uma imagem representativa. Utilize imagens fictícias que já estão disponíveis no projeto.

#### 4. Types e Union Types (15 pontos):

Defina um tipo para representar os dados de um filme.

Utilize union types para o gênero do filme ("Ação" | "Comédia" | "Drama" | "Ficção Científica").

#### 5. Destructuring (10 pontos):

Use desestruturação ao receber as props no MovieCard.

#### 6. Estilização com CSS Puro (15 pontos)

Estilize os componentes utilizando apenas CSS puro.

Utilize classes para aplicar os estilos e organize o CSS em arquivos separados.

## 7. Git Local (5 pontos)

Utilize o Git para versionamento local do seu código. Faça ao menos 5 commits que representem marcos importantes no desenvolvimento.

### Entrega:

Salve o projeto na máquina local e mostre os commits realizados ao final da avaliação para a correção.

A entrega não pode conter com a pasta `node_modules`.

O upload do projeto compactado no formato .zip deve ser feita no endereço <https://nac.fiap.com.br>.

Caso a entrega seja realizada com a pasta de dependências 20% do total da nota será descontado.

### Pontuação: 100

*Componentização* = 20

*Props e Tipagem* = 20

*Imagens* = 15

*Deconstructing* = 10

*Intersection Types* = 10

*Estilização com CSS Puro* = 15

*Git Local* = 10