**Pontifícia Universidade Católica de Minas Gerais**

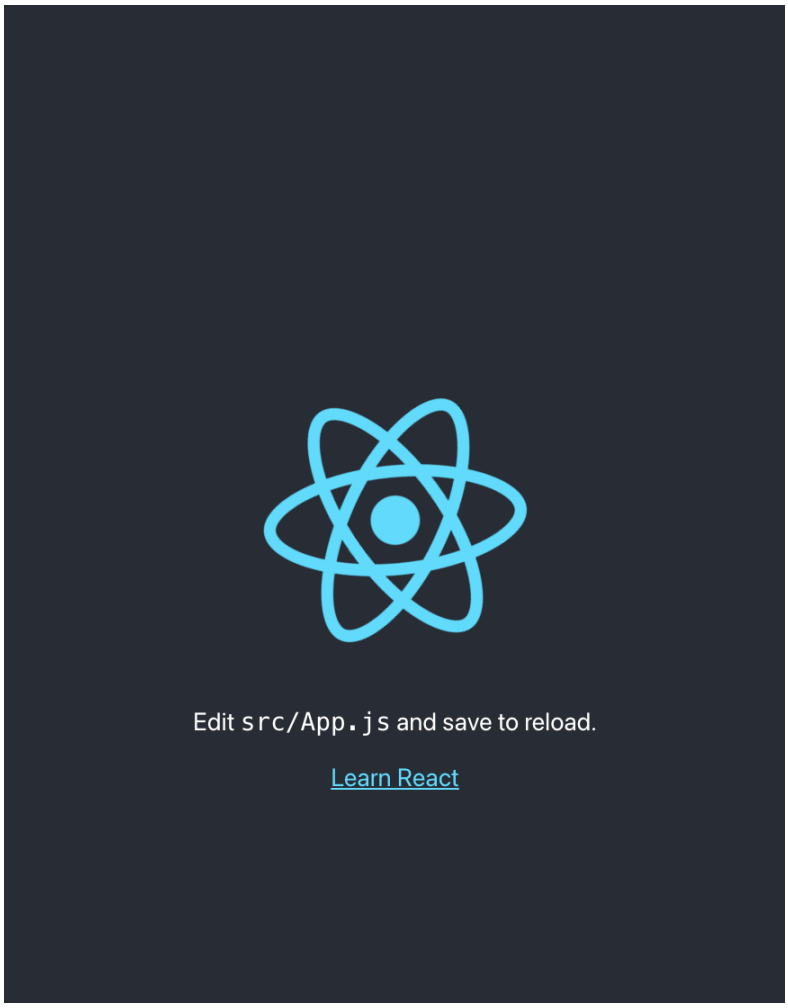
Curso: Desenvolvimento web Full Stack
Disciplina: Frameworks front end: React
Professor: Samuel Martins
Valor: 15pts

# Exercício 2

Para este exercício, iremos criar um cliente para o The Movies Database, uma API gratuita para consulta de filmes e séries! Vamos listar os filmes mais populares de acordo com a plataforma e adicionar uma tela de detalhes para que possamos ver mais informações à respeito do filme. Para isso, siga os passos abaixo:

## Passo 1

Inicialize um novo projeto com o seguinte comando: npx create-react-app react-movies. Para confirmar que a aplicação está funcionando corretamente, entre na pasta via terminal e rode o comando npm start. Você deverá ver uma tela similar à esta:



## Passo 2

Instale algumas bibliotecas auxiliares que serão utilizadas no projeto. Para isso, rode o seguinte comando: **npm install axios react-router-dom**.

## Passo 3

Vamos criar uma classe para implementar a integração da nossa aplicação com a API externa. Para isso, crie uma pasta “api” e um arquivo dentro da pasta criada com o nome de **MoviesService.js**. Neste arquivo, iremos colocar o seguinte conteúdo:

```
import axios from 'axios';

const API_KEY = 'd416af5d4faee64e25ab001d87aab5c3';
const BASE_URL = 'https://api.themoviedb.org/3/';
const withBaseUrl = (path) => `${BASE_URL}${path}?api_key=${API_KEY}`

export class MoviesService {
    static getMovies() {
        return axios(withBaseUrl('movie/popular'))
    }

    static getMovieDetail(id) {
        return axios(withBaseUrl(`movie/${id}`))
    }
}
```

Na segunda linha, fique à vontade para substituir a API key com a sua prórpia. Você pode encontrá-la no painel de controle da sua conta do TheMovieDB: https://www.themoviedb.org/settings/api.

O método **getMovies** recupera todos os filmes mais populares lançados recentemente. O método **getMovieDetail** recebe um id e recupera os detalhes de um filme específico passado como parâmetro.

## Passo 4

Vamos criar alguns componentes que irão nos auxiliar a listar os filmes recuperados da nossa API. Para isso, crie uma pasta chamada “components” e adicione um arquivo **Movie.js**. Crie também uma pasta “views” e dentro dela, adicione 2 novos arquivos:

- **Movies.js**;
- **MovieDetail.js**;

O arquivo Movies.js será utilizado como nossa página principal. Quando formos implementar o sistema de rotas, ele será o componente responsável pela home page e também por conter a lógica necessária para buscar os filmes e listá-los. O arquivo components/Movie.js (no singular :]) será utilizado para exibir um card de um filme. Vamos preenchê-lo com o seguinte código:

```
import { Link } from "react-router-dom"

export const Movie = ({ movie }) => (
    <div className="movie-item">
        <div>
            <img src={`https://image.tmdb.org/t/p/w200${movie.poster_path}`}
alt="" />
        </div>
        <div className="movie-excerpt">
            <h3>{movie.title}</h3>
            <Link to={`/movie/${movie.id}`} className="btn btn-primary">Ver
detalhes</Link>
        </div>
    </div>
);
```

Ele recebe um objeto **movie** como propriedade e lista as informações de título e imagem. Abra agora o arquivo views/Movies.js (no plural :]) e coloque o seguinte código:

```
import { useEffect, useState } from "react";
import { MoviesService } from "../api/MoviesService";
import { Movie } from "../components/Movie";

export const Movies = () => {
    const [movies, setMovies] = useState([]);

    const getMovies = async () => {
        const {
            data: { results },
        } = await MoviesService.getMovies();

        setMovies(results);
    };
    useEffect(() => {
        getMovies();
    }, []);

    return (
        <div className="container">
            <div className="row gy-5">
                {movies.map((movie) => (
                    <div key={movie.id} className="col-3">
                        <Movie movie={movie} />
                    </div>
                ))}
```

```
            </div>
        </div>
    );
};
```

Este código está basicamente buscando os dados da API no início do componente, e listando os resultados utilizando o componente Movie anteriormente criado.

## Passo 5

Vamos configurar o sistema de rotas da nossa aplicação. Para isso, crie um arquivo dentro da pasta src com o nome ApplicationRoutes.js e coloque o seguinte código:

```
import { Movies } from "./views/Movies";
import {
    createBrowserRouter,
    RouterProvider
  } from "react-router-dom";

const router = createBrowserRouter([
    {
      path: "/",
      element: <Movies />,
    },
  ]);

export const ApplicationRoutes = () => <RouterProvider router={router} />;
```

Abra agora o arquivo index.js e substitua pelo seguinte código:

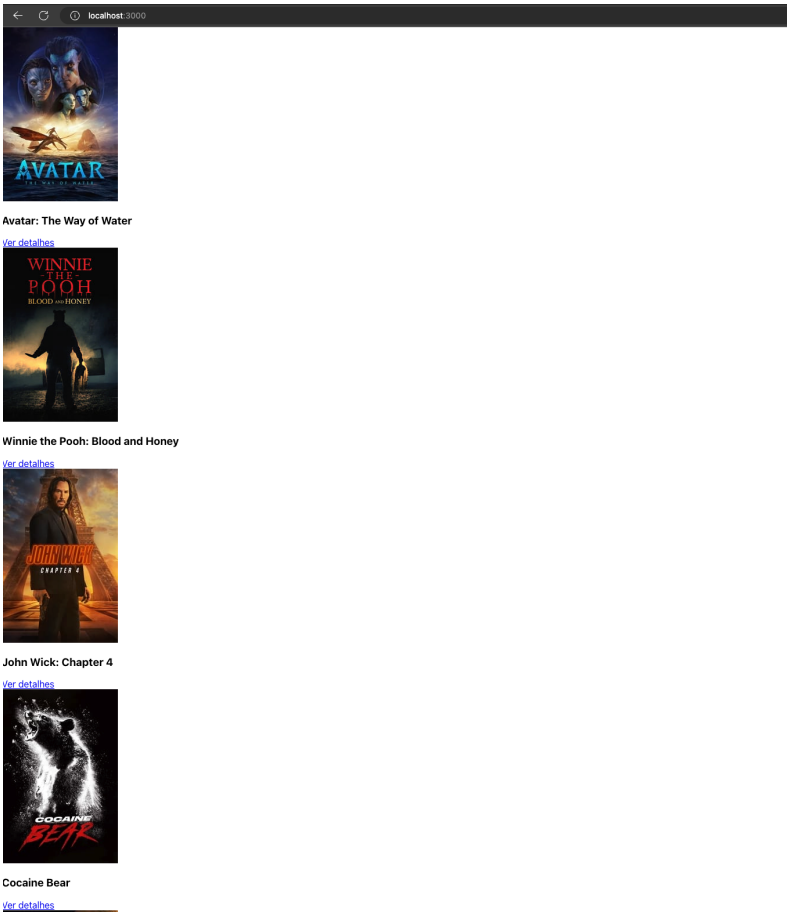
```
import React from "react";
import ReactDOM from "react-dom/client";
import "./index.css";
import reportWebVitals from "./reportWebVitals";
import { ApplicationRoutes } from "./ApplicationRoutes";

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(
    <React.StrictMode>
        <ApplicationRoutes />
    </React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
```

```
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

Este código está mapeando as URLs para os componentes que iremos utilizar na nossa aplicação. A esse ponto, você deverá ser capaz de visualizar algo como o da imagem abaixo:




## Passo 6

Vamos criar agora a páginas de detalhes do filme. Perceba que já temos um botão “Ver detalhes” que foi implementado para nos levar para uma outra página. Mas como não temos a implementação dessa rota ainda, nada irá acontecer quando clicarmos no link. Vamos adicionar um novo arquivo na pasta view com o nome MovieDetail.js:

```jsx
import {useState } from "react";
import { MoviesService } from "../api/MoviesService";
import { useParams } from "react-router-dom";

export function MovieDetail() {
    const [movie, setMovie] = useState({});
    const { movieId } = useParams();

    const getMovie = async () => {
        const { data } = await MoviesService.getMovieDetail(movieId);
        setMovie(data);
    };

    getMovie();

    return (
        <section className="movie-detail">
            <div className="container">
                <div className="row gx-5">
                    <div className="col-6">
                    {movie.poster_path && <img
src={`https://image.tmdb.org/t/p/w400${movie.poster_path}`} alt="" />}
                    </div>
                    <div className="col-6">
                        <h1>{movie.title}</h1>
                        <ul>
                            <li>Budget: {movie.budget}</li>
                            <li>Original language:
{movie.original_language}</li>
                            <li>Popularity: {movie.popularity}</li>
                        </ul>
                    </div>
                </div>
                <div className="row">
                    <div className="col-12">
                        {movie.overview}
                    </div>
                </div>
            </div>
        </section>
    );
};
```

Vamos adicionar mais uma rota no arquivo ApplicationRoutes, como demonstrado a seguir:

```
import { MovieDetail } from "./views/MovieDetail";
import { Movies } from "./views/Movies";
import { createBrowserRouter, RouterProvider } from "react-router-dom";

const router = createBrowserRouter([
    {
        path: "/",
        element: <Movies />,
    },
    {
        path: "/movie/:movieId",
        element: <MovieDetail />,
    },
]);

export const ApplicationRoutes = () => <RouterProvider router={router} />;
```

Agora, ao clicar em “ver detalhes” você deverá ver uma página como esta:


## John Wick: Chapter 4

- Budget: 90000000
- Original language: en
- Popularity: 2803.482

With the price on his head ever increasing, John Wick uncovers a path to defeating The High Table. But before he can earn his freedom, Wick must face off against a new enemy with powerful alliances across the globe and forces that turn old friends into foes.