Introdução

Olá! Obrigado pelo interesse em participar do nosso processo seletivo para dev frontend no [Promobit](https://www.promobit.com.br/).

O objetivo do desafio proposto é permitir uma melhor avaliação das suas habilidades como candidato à vaga de frontend. Este desafio deve ser feito apenas por você. Sua implementação e escolha de ferramentas poderá ser questionada em outra etapa.

O desafio

Usando a API de filmes gratuita [themoviedb](https://developers.themoviedb.org/3/getting-started/introduction) em sua versão 3, você será responsável por criar uma listagem dos filmes mais populares do dia, consultando o endpoint [GET /movie/popular](https://developers.themoviedb.org/3/movies/get-popular-movies) para realizar a listagem. Ao clicar em um item dessa listagem, outra página com os detalhes do filme escolhido deve ser exibida. Para acessar mais detalhes sobre o filme, você pode consultar o endpoint [GET /movie/{movie\_id}](https://developers.themoviedb.org/3/movies/get-movie-details).

Temos insights que nos levam a acreditar que os usuários dessa lista costumam ter uma experiência melhor se conseguirem criar um filtro usando seus genêros favoritos. Portanto, você também será responsável por criar filtros de filmes por gênero nessa listagem. Note que um novo endpoint deverá ser consultado para obter uma lista dos possíveis gêneros a serem filtrados, [GET /genre/movie/list](https://developers.themoviedb.org/3/genres/get-movie-list).

Para garantir que o usuário encontre o filme que está procurando, essa lista deverá ser paginada sempre que não existir nenhum filtro de gênero ativo.

Requisitos funcionais

 O usuário deve ter acesso a uma listagem dos filmes mais populares do dia

 O usuário deve conseguir paginar a lista para encontrar novos filmes

 O usuário deve conseguir filtrar os filmes listados por gênero, com a possibilidade de usar mais de um gênero

 Ao realizar filtros, o mesmo deve ser persistido pela paginação

 O usuário deve conseguir remover filtros e a listagem deve ser atualizada de acordo com o filtro removido

 O usário deve ter acesso a uma outra página com detalhes sobre o filme, ao clicar em um item na listagem

 O usuário deve conseguir voltar para a página de listagem de filmes com os filtros ainda ativos

Qualquer tipo de paginação é aceita, mas tenha em mente considerações ao optar por qualquer forma de paginar.

Requisitos não funcionais

 O app deverá ser hospedado em um serviço de cloud, como, por exemplo: [Surge](https://surge.sh/), [Netlify](https://www.netlify.com/), [ZEIT](https://zeit.co/home). Outras plataformas mais robustas como [AWS](https://aws.amazon.com/) também serão aceitas

 O app deverá ser criado usando [React](https://reactjs.org/)

 Na raiz do projeto, será necessário incluir um arquivo README.md com as instruções para construir seu projeto localmente

 O app deverá se comportar da mesma forma na última versão estável dos seguintes browsers: Chrome, Firefox, Edge

 A página com detalhes de um filme deve possuir uma rota própria e estar preparada para ser indexada em mecanismos de pesquisa

 O app deverá ser responsivo

Extras

Usar um framework para server-side rendering (SSR) ou static generation (SSG) como: [Next.js](https://nextjs.org/), [Gatsby](https://www.gatsbyjs.org/), [Nuxt.js](https://nuxtjs.org/)

Aplicar uma solução [CSS-in-JS](https://github.com/MicheleBertoli/css-in-js) ou [CSS Modules](https://github.com/css-modules/css-modules)

Escrever ou configurar testes de qualquer tipo, como: testes estáticos, unidade, integração, etc

Implemente outras funcionalidades que você acredite ser útil para o usuário desse app

Detalhar as razões pelas escolhas de ferramentas e técnicas aplicadas ao desafio. Também pode ser questionado em uma próxima etapa.

Critérios de avaliação

Boas práticas de desenvolvimento como: testes, design patterns, clean code

Domínio das ferramentas e linguagens que compõe um app de frontend moderno

Cuidados com perfomance aplicados ao app, como: estrutura de dados, bundle split de Javascript, eficiência da forma de distribuir e aplicar CSS

Métricas como LCP, FID e TTFB (a última será levada em consideração especialmente caso haja SSR)

Conhecimento em áreas adjacentes ao frontend, como DevSecOps e UX aplicadas ao desafio

Documentação: explicação para construir o app localmente, histórico e workflow de git

Entrega

Para realizar a entrega do desafio, você deverá enviar um email para [devs@promobit.com.br](mailto:devs@promobit.com.br) com o título [Frontend Challenge] Seu nome com o link para seu app e outro para o repósitório com seu código. Exemplo:

[https://link-para-o-app](https://link-para-o-app/)  
<https://github.com/seu-nome/front-end-challenge.git>

Quaisquer outros links ou materiais usados para a construção do app também podem ser enviados aqui, como: protótipos, diagramas, sketches, etc.

Não se esqueça de criar um arquivo README.md contendo as instruções para construir o app localmente.

Feedback

No Promobit, valorizamos muito feedbacks. Acreditamos que o feedback é essencial para melhorar, aprender e facilitar processos. Dessa forma, assim que o seu desafio for submetido, prometemos enviar um feedback técnico em até uma semana usando todos os critérios de avaliação descritos acima.

Dúvidas

Caso haja qualquer dúvida sobre o teste, nos envie um email para [devs@promobit.com.br](mailto:devs@promobit.com.br) com o título: [Frontend Challenge] Briefing da dúvida

Obrigado e bom desafio!