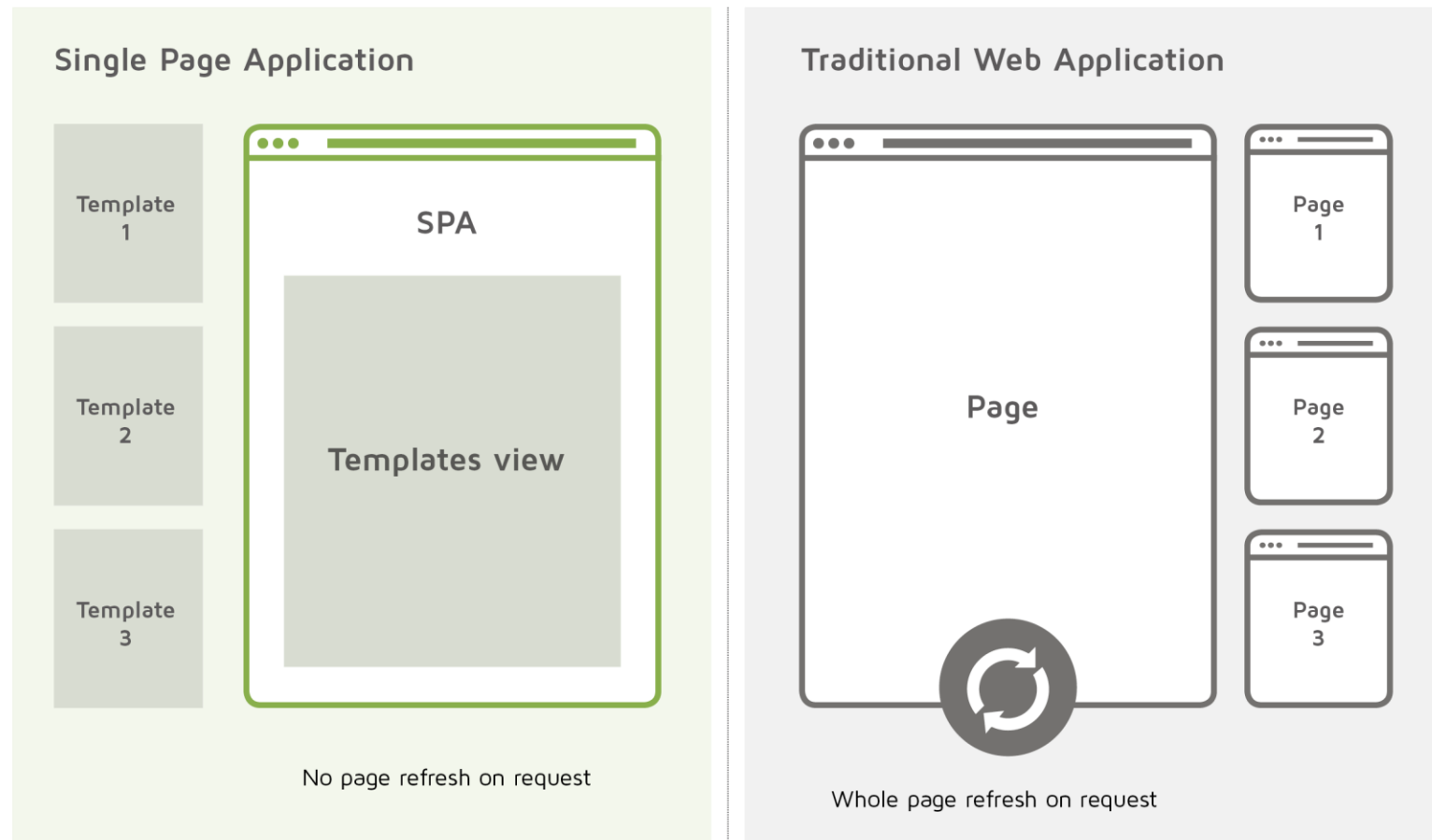




PUC Minas
Virtual

Arquitetura Single Page Application (SPA)

Samuel Martins

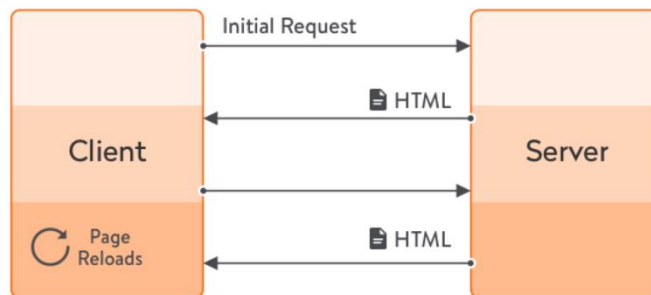


Fonte: <https://www.digitalclaritygroup.com/single-page-application-make-sense/>

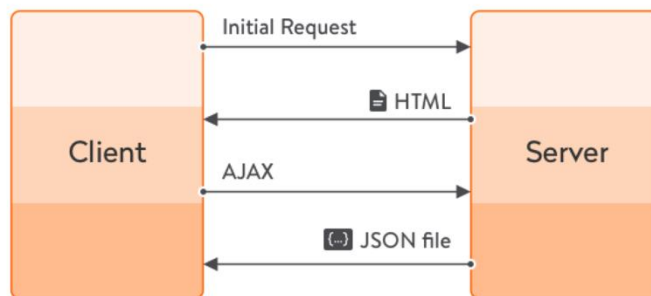
Arquitetura SPA - características

- Em tradução livre: aplicações de uma única página;
- Ao invés de recarregar toda a página, apenas parte do conteúdo é reescrito na navegação entre as telas;
- Definição de uma parte comum da aplicação. Demais conteúdos (templates) são escritos como componentes e renderizados ao entrar na página;
- Fluidez na navegação, uma vez que apenas parte do conteúdo é trocado nas mudanças de páginas.

Multi-page app lifecycle



Single-page app lifecycle



[SPA vs MPA]

Fonte: <https://www.altamira.ai/blog/single-page-app-vs-multi-page-app/>

Arquitetura SPA - características

- **Carga maior em cima do browser**, uma vez que a máquina do usuário é a responsável por renderizar a aplicação;
- **Problemas com SEO**: a aplicação é renderizada dinamicamente, após interpretação dos scripts. Isso faz com que os mecanismos de busca não consigam identificar o conteúdo da página para fazer a indexação de acordo com a busca do usuário;
- **Tempo inicial de renderização um pouco maior**: a página é aberta, uma requisição ao servidor é feita para buscar os scripts, os scripts são interpretados e só após o conteúdo é renderizado na página.

Arquitetura SPA - características

- Mapeamento de Componente + Página. Cada página (rota) é representada por um componente que deve ser implementado, exemplo:
 - `"/" => <Home />`;
 - `"/movies" => <Movies />`;
 - `"/movies/:id" => <MoviesDetail />`.

React Router

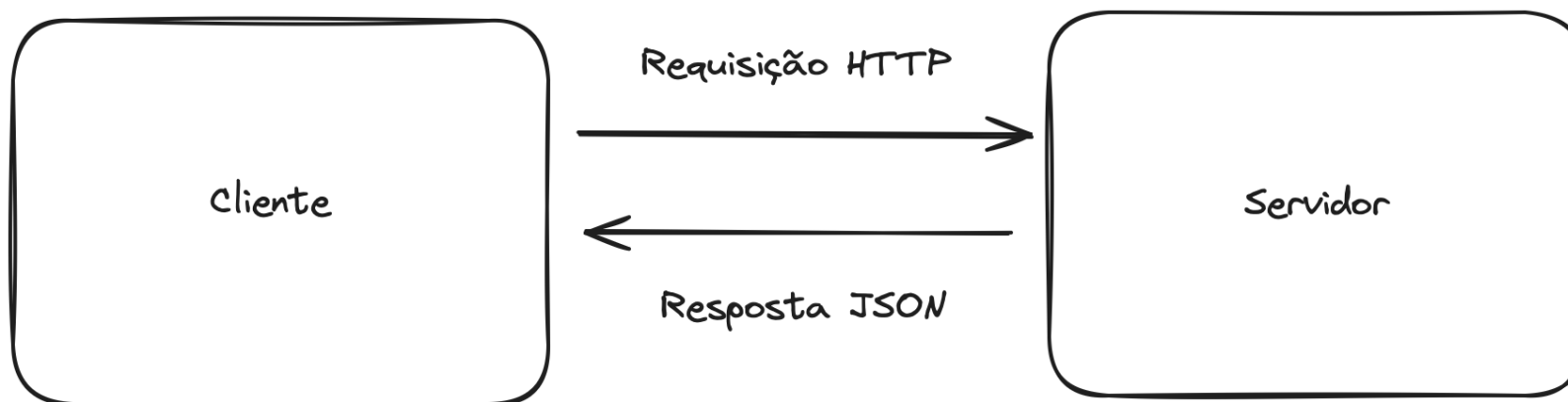
- Biblioteca para React com o objetivo de possibilitar a implementação de múltiplas rotas em uma aplicação;
- Recebe parâmetros via URL;
- Component-driven: todas as rotas nos levam a um componente.

Integrações com API externas

Samuel Martins

Requisições em APIs externas

- Aplicações client-side (front-end) tratam apenas de interações com usuário;
- Grande parte das aplicações corporativas irão ser construídas com base em uma arquitetura cliente -> servidor



Requisições em APIs externas

- **GET:** requisita dados do servidor (listagem de filmes, por exemplo);
- **POST:** envia informações para o servidor (criação de novos recursos, alteração do estado do servidor);
- **DELETE:** remove um recurso do servidor;
- **PUT:** cria um novo recurso ou altera um recurso em sua totalidade;
- **PATCH:** altera parcialmente um recurso;

Axios

- Biblioteca utilizada no front-end para formatar requisições de qualquer natureza (GET, POST, com ou sem passagem de parâmetros).

A X I O S

Caso de uso

- Aplicação para listagem de filmes, com busca na API [TheMovieDB](#);
- Para obter um token:
 - Cadastre-se gratuitamente na plataforma - <https://www.themoviedb.org/signup>;
 - Entre na página Settings -> API e copie o token na área API Key - <https://www.themoviedb.org/settings/api>



PUC Minas
Virtual