

**1) Compare a arquitetura Single Page Application com a Multi Page Application e explique 3 vantagens e 3 desvantagens de cada uma das abordagens.**

***Single Page Application (SPA):***

**Vantagens:**

1. Experiência do usuário interativa: As SPAs carregam uma página inicial única e, em seguida, atualizam o conteúdo dinamicamente conforme o usuário interage com a aplicação. Isso resulta em uma experiência de usuário mais fluida e responsiva, sem interrupções entre as transições de página.
2. Melhor desempenho: Uma vez que o conteúdo é carregado apenas uma vez, as SPAs podem fornecer tempos de carregamento mais rápidos para os usuários subsequentes, já que não é necessário carregar a página inteira novamente.
3. Facilidade de desenvolvimento multiplataforma: As SPAs podem ser desenvolvidas usando frameworks populares, como Angular, React ou Vue.js. Esses frameworks fornecem ferramentas e recursos que simplificam o desenvolvimento multiplataforma, permitindo que os desenvolvedores criem aplicações web e móveis com base no mesmo código.

**Desvantagens:**

1. Requisitos de capacidade do navegador: SPAs dependem fortemente do JavaScript para gerenciar a navegação e atualizações de conteúdo. Isso pode resultar em requisitos mais altos de capacidade do navegador e recursos do dispositivo, tornando as SPAs menos adequadas para dispositivos mais antigos ou navegadores com suporte limitado ao JavaScript.
2. Dificuldades de SEO: Como as SPAs geralmente carregam conteúdo dinamicamente, pode ser desafiador para os motores de busca indexarem todo o conteúdo. É necessário implementar técnicas de otimização de mecanismo de busca específicas para garantir que o conteúdo seja corretamente rastreado e indexado.
3. Complexidade inicial: O desenvolvimento de uma SPA pode exigir um pouco mais de esforço e complexidade inicial em comparação com as MPAs, devido à necessidade de configurar o roteamento, gerenciamento de estado e tratamento adequado de eventos.

***Multi Page Application (MPA):***

**Vantagens:**

1. Facilidade de SEO: As MPAs seguem a abordagem tradicional de carregar páginas individuais ao navegar em um site, o que facilita a indexação de conteúdo pelos motores de busca. Cada página pode ser otimizada individualmente para classificação nos mecanismos de busca.

2. **Compatibilidade com navegadores antigos:** As MPAs são mais compatíveis com navegadores mais antigos, pois não dependem fortemente de JavaScript para o funcionamento básico da navegação e carregamento de páginas.
3. **Desenvolvimento mais simples:** O desenvolvimento de uma MPA pode ser mais simples, pois cada página é tratada separadamente. Isso pode ser benéfico para projetos menores ou quando não há necessidade de recursos avançados de interatividade.

#### **Desvantagens:**

1. **Experiência de usuário fragmentada:** As MPAs exigem recarregamento de página ao navegar, o que pode resultar em uma experiência de usuário menos fluida e mais fragmentada, pois pode haver interrupções perceptíveis entre as transições de página.
2. **Tempos de carregamento mais lentos:** Como cada página precisa ser carregada novamente, as MPAs geralmente têm tempos de carregamento mais lentos em comparação com as SPAs. Isso pode impactar a experiência do usuário, especialmente em conexões de internet mais lentas.
3. **Duplicação de recursos:** Em MPAs, os recursos comuns, como cabeçalhos, rodapés ou scripts, podem precisar ser duplicados em várias páginas. Isso pode resultar em um maior tamanho total de transferência de dados e exigir mais tempo de desenvolvimento para manter a consistência.

## **2) Quais são as principais responsabilidades e atribuições de um(a) profissional de Arquitetura de Front End?**

Um(a) profissional de Arquitetura de Front End é responsável por projetar, planejar e coordenar a arquitetura técnica de uma aplicação ou sistema web no lado do cliente (front end).

Essa função envolve diversas responsabilidades e atribuições, que podem variar de acordo com a empresa e o projeto específico como por exemplo: Planejamento, estratégia, padronização, boas práticas, resolução de problemas, otimização, avaliação e adoção de tecnologias.

## **3) Quais são as principais vantagens e desvantagens em se utilizar CSS em bloco de estilos em relação a escrita em múltiplos arquivos externos?**

Ao utilizar CSS em bloco de estilos (inline CSS) você tem vantagens como:

1. **Facilidade de implementação:** O CSS em bloco de estilos é incorporado diretamente no elemento HTML, o que significa que não é necessário fazer referência a arquivos externos. Isso pode simplificar o processo de implementação e reduzir o número de solicitações de recursos adicionais.

2. **Controle de estilo específico:** Ao utilizar CSS em bloco de estilos, é possível aplicar estilos específicos a elementos individuais, sem afetar o restante da página. Isso permite um controle mais granular sobre o estilo de cada elemento.
3. **Prioridade de estilo:** O CSS em bloco de estilos tem uma alta especificidade, o que significa que ele terá uma prioridade maior do que os estilos definidos em arquivos externos. Isso pode ser útil para sobrepor estilos padrão ou aplicar estilos específicos a elementos importantes.

Porém você tem desvantagens como:

1. **Dificuldade de manutenção:** O CSS em bloco de estilos pode se tornar difícil de manter, especialmente em páginas complexas com muitos elementos. O código CSS pode se tornar extenso e desorganizado, tornando difícil rastrear e fazer alterações.
2. **Reutilização limitada:** O CSS em bloco de estilos é específico para cada elemento e não pode ser facilmente reutilizado em outros elementos ou páginas. Isso pode resultar em duplicação de código e dificuldade na manutenção consistente dos estilos.
3. **Baixa modularidade:** O CSS em bloco de estilos não segue o princípio da separação de preocupações (Separation of Concerns), o que pode tornar a arquitetura de estilos menos modular. Isso pode dificultar a reutilização de estilos, a escalabilidade e a colaboração entre desenvolvedores.

Ao utilizar CSS escrita em múltiplos arquivos externos você tem vantagens como:

1. **Modularidade:** Escrever o CSS em múltiplos arquivos externos permite uma abordagem mais modular, onde os estilos podem ser organizados em arquivos separados de acordo com a função, o componente ou a página específica. Isso facilita a reutilização, a manutenção e a colaboração entre desenvolvedores.
2. **Manutenção simplificada:** Com o CSS em arquivos externos, é possível separar e organizar os estilos de forma mais clara e estruturada. Isso torna a manutenção mais fácil, pois é possível fazer alterações em arquivos específicos sem afetar o restante do código.
3. **Cache eficiente:** Ao utilizar arquivos externos, os navegadores podem armazenar em cache os arquivos CSS, resultando em um carregamento mais rápido das páginas subsequentes. Isso pode melhorar o desempenho geral do site.

Porém você tem desvantagens como:

1. **Overhead de solicitação:** Cada arquivo externo CSS requer uma solicitação HTTP separada ao servidor. Se houver muitos arquivos externos, pode haver um aumento no tempo de carregamento da página devido ao overhead das solicitações.
2. **Prioridade de estilo:** Ao utilizar arquivos externos, é importante considerar a ordem em que os arquivos são referenciados, pois a ordem pode afetar a prioridade dos estilos aplicados. Isso requer um planejamento cuidadoso para evitar conflitos e garantir a ordem correta de importação.

3. **Complexidade de configuração:** Ao trabalhar com múltiplos arquivos externos, pode ser necessário configurar ferramentas de construção ou automação para combinar e minimizar os arquivos CSS, a fim de reduzir o número de solicitações. Isso pode adicionar complexidade ao processo de desenvolvimento.

**5) Escolha um dos frameworks comentados em sala de aula: React, Angular ou Vue. Baseando-se na documentação ou nos templates disponíveis no Codesandbox, crie um projeto base e desenvolva uma tela seguindo a estrutura da imagem abaixo.**

Link GitHub: <https://github.com/CarlosHMoura/Arquitetura-Front-End-PUC-Minas>