**Pontifícia Universidade Católica de Minas Gerais**

|  |  |
| --- | --- |
| Curso: | Arquitetura de Software Distribuído |
| Disciplina: | Arquitetura de Front Ends |
| Professor: | Samuel Martins |
| Aluno: | Tiago de Oliveira Raposo |

**Exercício 1**

1. **Como vantagens, podemos citar:**
   * **o desempenho de navegação em uma SPA é otimizado, devido ao pré-carregamento de toda a página no início do acesso;**
   * **os estados entre telas são mais fáceis de serem mantidos;**
   * **a transição do front end para outras plataformas (web para mobile, por exemplo) fica mais simples.**

**As desvantagens são:**

* **otimizações para motores de busca são mais complicadas, já que os crawlers não costumam executar códigos em JavaScript;**
* **é preciso haver uma preocupação com o histórico de navegação, que não é automático como em uma aplicação de várias páginas;**
* **o carregamento inicial da página pode se tornar um problema.**

1. **O profissional de arquitetura de front end é o responsável por definir as tecnologias usadas na construção do front end, orientar as equipes de design e desenvolvimento em torno do contexto arquitetural e tecnológico, criar provas de conceito, garantir a conformidade do sistema com o planejamento da arquitetura, e centralizar as decisões arquiteturais do produto.**
2. **O código CSS em bloco traz um desempenho muito maior em relação aos códigos separados em arquivo, e deve ser usado quando a performance da aplicação não estiver satisfatória. No entanto, a reusabilidade e manutenibilidade do código se torna pior, pois fica restrita a um único arquivo HTML.**
3. **Vantagens:** 
   * **Facilita a escolha de nomes, diminuindo conflitos e agilizando a escolha dos mesmos, já que define regras claras a respeito;**
   * **Deixa facilmente visível a separação dos elementos, especialmente dentro da hierarquia (quem é bloco, quem é elemento filho e quem é modificador);**
   * **Torna o entendimento dos estados dos elementos mais rápida, pois há classes específicas de modificadores que tornam visíveis alterações nos estados do elemento.**

**Desvantagens:**

* + **Nomes de classes se tornam muito grandes, poluindo o HTML;**
  + **As classes não são muito reutilizáveis, pois se tornam dependentes das classes pai;**
  + **No caso de um bloco com vários elementos, haverá uma classe de modificador para cada elemento, independente de terem os mesmos atributos.**

1. **Três técnicas que ajudam a evitar o override de estilos no CSS:**
   * **O estilo de código CSS chamado BEM determina regras para se nomear as classes, considerando uma hierarquia bem definida de escopos para cada classe. Ao colocar apenas o nome do bloco como único, se garante que todas as classes filhas também serão únicas, pois herdam o nome da classe do bloco.**
   * **O SASS fornece ferramentas para se colocar classes como dependentes de outras ou de ids de elementos HTML, por exemplo. A organização do CSS se torna mais estruturada e lógica para o programador, que pode criar interdependências entre arquivos CSS e HTML, e o compilador SASS resolve as unicidades de nomes.**
   * **CSS funcional determina a criação de pequenas classes, reutilizáveis em todo o código HTML. Com isso, todas as páginas podem usar um mesmo CSS, com poucas exceções que podem ser tratadas com estilos inline, por exemplo.**
2. **Parcel: o Parcel pega um arquivo de entrada de código e percorre todas as referências e dependências declaradas para gerar uma árvore de recursos (*assets*). A partir disso, ele constrói uma árvore de pacotes (*bundles*), e empacota todos eles em um arquivo final para o navegador. Parcel aceita uma grande quantidade de frameworks diferentes, trabalhando com os compiladores para gerar os arquivos JavaScript.**

**Browserify: o Browserify pega um arquivo de entrada e gera um arquivo JavaScript que agrega todos os outros scripts referenciados por require(). Esse arquivo é um bundle.js, que pode ser referenciado com uma tag <script> no HTML.**

1. **Escolhi o Angular como framework, e a metodologia de escrita de CSS foi o BEM.**

**O desenvolvimento foi bem simples, sem obstáculos.**

**Criei uma aplicação Angular por já trabalhar com a tecnologia (queria aprender React, mas fiquei sem tempo). No entanto, notei que a pasta de projeto tem 350MB, e o tempo de compilação é de 10 segundos em uma boa máquina. Possivelmente, devido ao escopo do projeto, o React fosse mais adequado.**

**A metodologia BEM facilita muito a produção de CSS hierarquizado, mantendo o código muito bem organizado e de fácil leitura. Um problema que percebi, e que pode ser relevante em projetos maiores, é que o hover de ações (a cor do elemento muda para royalblue) não é reutilizável, já que sempre será específica do elemento HTML. Na BEM, o CSS tende a copiar a estrutura do HTML.**

**Link do GitHub do item 7:**

<https://github.com/TiagoRaposoBR/puc-afe-aula01>