

Frameworks

1 - Atualmente temos 3 frameworks javascript que se destacam no mercado para a construção de SPAs e front-end modernos. Cite quais, qual sua opinião sobre eles ou sobre aquele que possui mais afinidade, e quais as vantagens e por que esses frameworks são adotados em massa pelo mercado?

Atualmente os frameworks mais utilizados para construir Single Page Applications são o Angular, o Vue e o React. Atualmente estou concentrando meus estudos no Angular, não que ele seja melhor que os outros dois e sim por achar ele mais completo e por ser mantido por uma gigante como o Google. Todavia existe espaço para todos eles. O Angular por exemplo possui o conhecido padrão de arquitetura MVC o que facilita muito a construção de aplicações multiplataforma já que o modelo de negócio e a interface controladora fica obrigatoriamente separada da camada de visualização, a única que precisa ser alterada. Podemos utilizar até outros frameworks na view caso achemos necessário. O Vue e o React são mais leves mas são focados mais na camada de visualização. O mercado está adotando atualmente frameworks de desenvolvimento Javascript pois: Javascript é uma das linguagens mais utilizadas no mundo; É multiplataforma podendo inclusive ser utilizado em aplicações para desktop; Possui ampla documentação; Conseguimos construir uma aplicação web utilizando boa parte do código em Javascript, inclusive no servidor se utilizarmos no NodeJS, economizando mão de obra especializada em várias linguagens de programação.

2 - O que é uma SPA? Vê vantagens e/ou desvantagens na utilização desta abordagem? Quais?

SPA ou Single Page Application é à grosso modo uma aplicação completa web que roda em uma página única de forma totalmente dinâmica. O vantajoso desta abordagem é a redução drástica das requisições no servidor visto que a aplicação é praticamente carregada de uma vez no browser. A navegação fica mais fluída principalmente durante troca de páginas ou troca de conteúdos, html, css e lógica javascript são carregados apenas uma vez, apenas o conteúdo dinâmico é requisitado no servidor. Desvantagens, bom, não diria desvantagens e sim que o SPA não é adequado para alguns propósitos como por exemplo uma página promocional ou hotsite que não precisa fazer parte da aplicação SPA, ou ainda a utilização em projetos muito pequenos com poucas linhas de código.

3 - O que é o Virtual DOM? Quais os prós e contras desta abordagem? Quais frameworks atuais utilizam esta abordagem?

O Virtual DOM é um recurso do React que manipula os elementos DOM diretamente pelo framework como uma forma de modelo ou template a vantagem é que reduz a complexidade de acessarmos diretamente o DOM principalmente em Html muito extensos visto que o próprio VDOM acessa e modifica o DOM real.

4 - O que é o "two way data bind"? Quais as diretivas ou como utilizar com Vue, Angular e/ou React?

Utilizamos o "two way data bind" quando queremos que o template html mostre propriedades alteradas em um componente e este componente escute eventos disparados pelo template e execute

alguma lógica simultaneamente. Um bom exemplo é um campo de texto que quando alterado altera também um trecho de texto no html. No Vue utilizamos o v-model como diretiva já no Angular utilizamos o ngModel

6 - Quais as principais maneiras de se trabalhar com estilos no ReactJS?

É implementando o estilo diretamente no código Javascript.

7 - Programação reativa e programação declarativa são dois termos chaves no desenvolvimento em ReactJS, o que significa estes termos e quais as vantagens?

Programação reativa é um paradigma onde a aplicação é construída para “reagir” à eventos disparados e lidar com eles de forma assíncrona, a principal vantagem deste paradigma é conseguir lidar com uma alta carga de eventos simultâneos sem que a aplicação fique lenta ou parada aguardando resposta de um processo síncrono. No paradigma declarativo a aplicação é construída na intenção de “obter” um resultado e não de “chegar” ao resultado fica fácil de entender comparando com a programação imperativa onde criamos um bloco de código que processa um algoritmo e chega à um resultado. Em programação declarativa informamos para a linguagem qual resultado esperamos e a linguagem o retorna. Vantagens, mais fácil de aprender e entender o código, é menos lógico e mais semântico.

8 - Quais as funções das ferramentas ESLint e Prettier? Quais as vantagens de se utilizá-las individualmente e em equipe? Você já utiliza alguma?

São formatadores de código. A utilização deles é importante pois mantém o código limpo e legível, facilita a manutenção e o teste. Geralmente utilizo formatadores de código existentes nos próprios SDK's, o Angular CLI faz um Lint através do ng lint

9 - Se tratando de forms dois conceitos muito presentes no React são componentes não controlados e componentes controlados, o que são estes conceitos? Em quais situações é preferível cada um? Qual você mais utiliza?

Componentes controlados são aqueles onde os dados de um formulário são controlados pelo próprio React e os não controlados são acessados pela própria DOM. É preferível utilizar o não controlado caso você queira utilizar código não React em seu formulário, caso contrário é recomendável utilizar os componentes controlados. Não tenho preferência entre um ou outro.

10 - Há indícios de que uma parte da aplicação ou um componente específico esteja executando renderizações excessivas ou usando mais processamento do que deveria... Quais os procedimentos que você adotaria para diagnosticar e averiguar a situação?

Eu começaria pelos logs e eventos disparados no console do browser, caso não fosse possível descobrir eu faria testes unitários nos últimos serviços atualizados ou criados.

11 – no Redux, "useSelector" e "useDispatch": O que fazem? Em quais momentos utilizar?

O useSelector permite lermos o estado renderização de um elemento e o useDispatch nos permite alterar este estado. Podemos utilizá-lo para fazer um efeito de toogle em um elemento html, alterar visibilidade de conteúdo, alternar a característica de um elemento em múltiplos estados.

12 - Para que serve o "|" (pipe) nos templates do angular?

O pipe altera a formatação de dados do JS para uma forma amigável, legível. Exemplo mudando o formato como data e hora devem ser exibidos no template.

13 - Quais as principais diferenças entre o AngularJS e o Angular 2+?

O AngularJS é obsoleto, ele não dá suporte nativo à plataforma Mobile, não utiliza Typescript, não possui um CLI, a sintaxe não é a mesma.

14 - Quais as principais diretivas estruturais do Angular e qual a função deste tipo de diretiva?

Diretivas estruturais são aquelas capazes de modificar completamente a estrutura do template adicionando e removendo elementos do DOM. Exemplos: NgIf, NgFor, NgSwitch, NgSwitchWhen, NgSwitchDefault

15 - De maneira resumida, quais as finalidades de cada uma destas ferramentas: Gulp, Babel, Npm, Grunt, Yarn, Webpack, Bower? Quais você já utilizou? Por que acredita que este tipo de ferramentas vem se popularizando nos últimos anos?

Gulp = Automatizador de tarefas JS (minify, teste etc)

Babel = Compilador JS ECMA6 para JS Vanilla

Npm = Gerenciador de pacotes JS do NodeJS

Grunt = Automatizador de tarefas JS (minify, teste etc)

Yarn = Gerenciador de pacotes JS do NodeJS

Webpack = Empacotador de código JS

Bower = Gerenciador de dependência para Frontend

Já utilizei outro aplicativo semelhante ao Gulp, utilizei o Babel, Npm e o Yarn. Este tipo de ferramentas se popularizam pois elas garantem produtividade e controle de falhas. Tudo que podemos automatizar é válido.