

ReactJS

Nota de Compreensão: 3.0 pontos

O que achei do Framework: Apesar da compreensão da documentação ser fácil, não significa que usar o *framework* é. Por grande parte da lógica usada ser a mesma usada em JavaScript, não entendi muito bem por não ser familiarizado com a linguagem. É possível entender a lógica por trás de como o código é escrito mesmo sem saber o que cada comando específico faz. O que achei mais interessante foi o sistema de *render* do React, onde ao invés de ser uma coisa mecânica, como digitar uma expressão e o sistema calculá-la, você basicamente descreve o que quer que o sistema faça e ele traduz em uma informação, uma descrição, que o sistema depois pega e usa para lançar um output equivalente ao que foi descrito previamente.

Aparenta ser um *framework* muito bom para projetos que precisam de uma interface de usuário “forte” e não somente uma grande quantidade de textos na tela. E o fato de poder ser utilizado pelo browser facilita muito pelo rápido acesso ao ambiente de desenvolvimento e pela facilidade de poder editar um projeto em praticamente em qualquer lugar, sendo necessário apenas um conhecimento básico em JS e HTML.

VueJS

Nota de Compreensão: 4.0 pontos

O que achei do Framework: O que achei de melhor nesse *framework* foi a interatividade com o JS de páginas até mesmo já existentes. Com alguns comandos é possível gerar páginas cheias de vida e com interatividade do usuário. Por exemplo, as diretivas (*v-*) como a *v-bind*, que fazem parte do Vue e dão interações ao texto da página (podendo fazer ele se movimentar por exemplo), as condicionais que podem por exemplo fazer um texto desaparecer quando certa condição é alcançada e até mesmo o *v-on* que pode criar botões de input, por exemplo, que fazem algum comando ser executado, como inverter algum texto (no exemplo mais banal). Assim é possível deixar um projeto com funcionalidade e beleza ao mesmo tempo.

Também é de grande ajuda o sistema de “composição em árvore” que lembra uma espécie de fluxograma, onde o programa é operado por diversos componentes menores que se juntam em maiores e assim sucessivamente até

formarem toda a composição do programa. Dessa maneira é mais fácil utilizar o sistema já que os componentes menores podem muitas vezes ser reutilizados para que o trabalho seja mais dinâmico e menos repetitivo.

Foi possível entender o funcionamento do *framework*, mas pelo mesmo motivo anterior foi difícil entender os mínimos detalhes pelo fato de conhecer bem pouco de JS e quase nada de CSS.

AngularJS

Nota de Compreensão: 2.0 pontos

O que achei do Framework: O Angular CLI ajuda a automatizar a criação de projetos, serviços e etc. Para a criação de recursos se usa o comando: `ng g x` (componente, directive, etc), sendo `ng` o comando e `g` a abreviação de generate. É possível usar as linguagens JS, ES6 e principalmente TypeScript (linguagem desenvolvida pela Microsoft). Por ser mais completo tem muitos comandos e é necessário ter na máquina o Node.js para gerenciar os arquivos .json. Foi o framework que mais tive dificuldades de entender como funciona pois são muitos comandos para modificar, salvar, e gerir diferentes partes de cada arquivo.

Testei os 2 primeiros pelo browser e não consegui fazer muito por não conhecer as principais linguagens usadas em cada, tendo alcançado apenas o que estava para ser desenvolvido em ambas as páginas de documentação, um simples Hello World, mas foi mais fácil de entender fazendo do que só lendo.