



Vue.js

DESENVOLVIMENTO EVOLUTIVO E MODULARIZADO

Quem sou

- Frederico Lima (lima.frederico.87@gmail.com / @fredorlima)
- Graduação
 - UEFS - Engenharia da Computação
 - UFBA - Ciência da Computação
 - FTC - Sistemas de Informação
- Profissional
 - Desenvolvedor Fullstack - 2006
 - Webfeira / UFBA / Momentum / Huggy.io (atual)
 - Desenvolvedor Fullstack e Gerente de Projetos
 - Spotlab / Linceweb
- Palmeiras tem Mundial SIM!!!

Introdução

- Pronúncia: /vju:/ (como view)
- Vue.js é um framework progressivo
 - Focado em soluções incrementais
 - Modularizável e Escalável
 - Facilmente integrável com pacotes third-parts e componentes
- Criação de interfaces com o usuário
- Reativo
 - Mudanças do lado do Vue desencadeiam alterações no front-end
 - Perrogativa: Manipular DOM é custoso, entediante, disruptivo (de-evolutiva) e suscetível a erros

O que é um Framework?

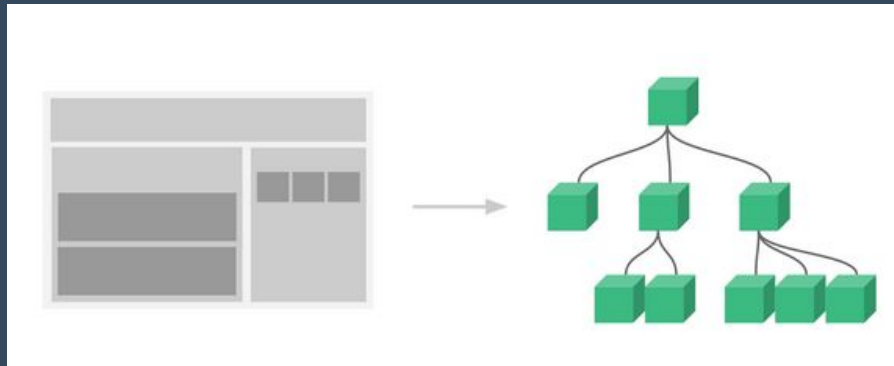
- Abstração de códigos comuns
- Funcionalidades genéricas
- Fluxo de controle ditado pelo framework, ao contrário de bibliotecas
- Benefícios
 - Redução no tempo de desenvolvimento / Aumento de produtividade
 - Reutilização de código (componentização)
 - Conhecimento universal
 - Consistência das aplicações / Padronização
- Front-end, back-end, micro-frameworks
- Vue.JS, Bootstrap, Ionic, Angular, Laravel, Lumen, Slim

Porque usar?

- O framework nos permite renderizar dados no DOM usando sintaxe de template
- Facilita a manipulação de eventos
- Flexibilidade
 - Adaptável para qualquer tipo de projeto
 - Pode trabalhar em conjunto com qualquer outra linguagem (Ruby, PHP, Python, Node, Java, Go ou qualquer outra)
- Curva de aprendizagem muito muito curta
- Performance
- *[Apresentação exemplos A]*

Componentes

- Permitem o desenvolvimento de módulos independente e em larga escala
- Auto-contidos e reutilizáveis
- São como instâncias do Vue com parâmetros definidos
- Aplicação pode ser abstraída como uma árvore de componentes
- *[Apresentação exemplo B]*



Dados Computados

- Operam como "virtual fields"
- Código mais limpo e manutenível
- Utilizados principalmente para manipulação dados
- *[Apresentação exemplo C]*

Observadores

- Recurso para reação a alterações de dados
- Utilização ideal para operações assíncronas
- *[Apresentação exemplo D]*

Outros recursos

- Transições e animações
- Mixins
 - Distribuição de funcionalidades reutilizáveis através de componentes
- Diretivas customizadas
 - v-demo
- Rotas
- Manipulação de estado (vuex)

Fontes

- <https://vuejs.org/>
- <https://vuex.vuejs.org/>
- <https://github.com/fredorlima/vuejs-demos>

Obrigado! Perguntas?!