

ANGULAR VS **REACT VS VUE**

QUAL O MELHOR?

**FRAMEWORK
E
LIBRARY**



FRAMEWORK

COMPONENTS

FORMS

ROUTER

HTTP

•

•

•

FILOSOFIA

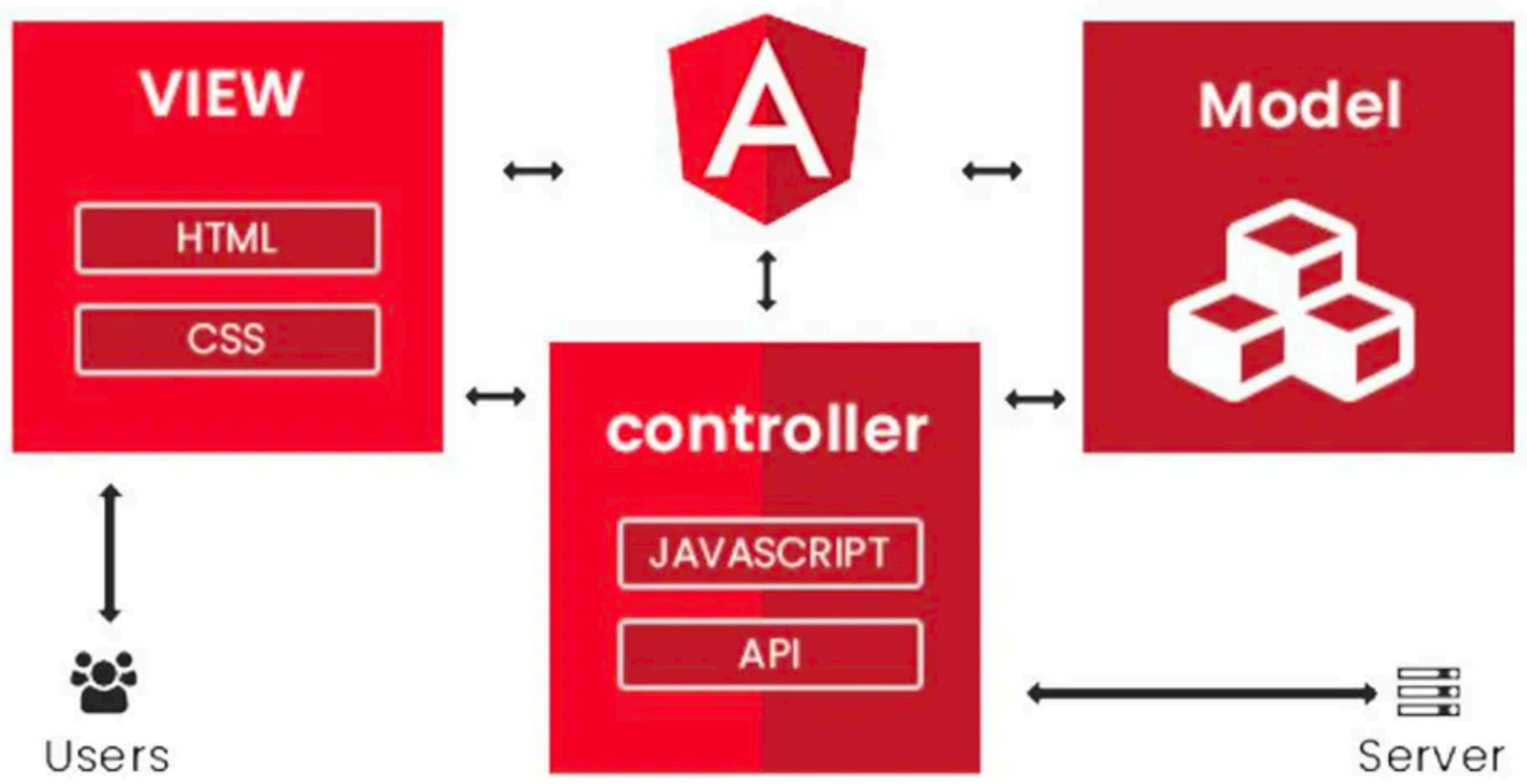
MVC

FORTEMENTE ORIENTADO À OBJETOS

INJEÇÃO DE DEPENDÊNCIA

TYPESCRIPT (FORTEMENTE TIPADO)

*** É FACIL DE APRENDER PARA QUEM VEM DO JAVA**



PONTOS POSITIVOS

VÁRIAS IMPLEMENTAÇÕES PRONTAS QUE FACILITAM O PROCESSO DE DESENVOLVIMENTO;

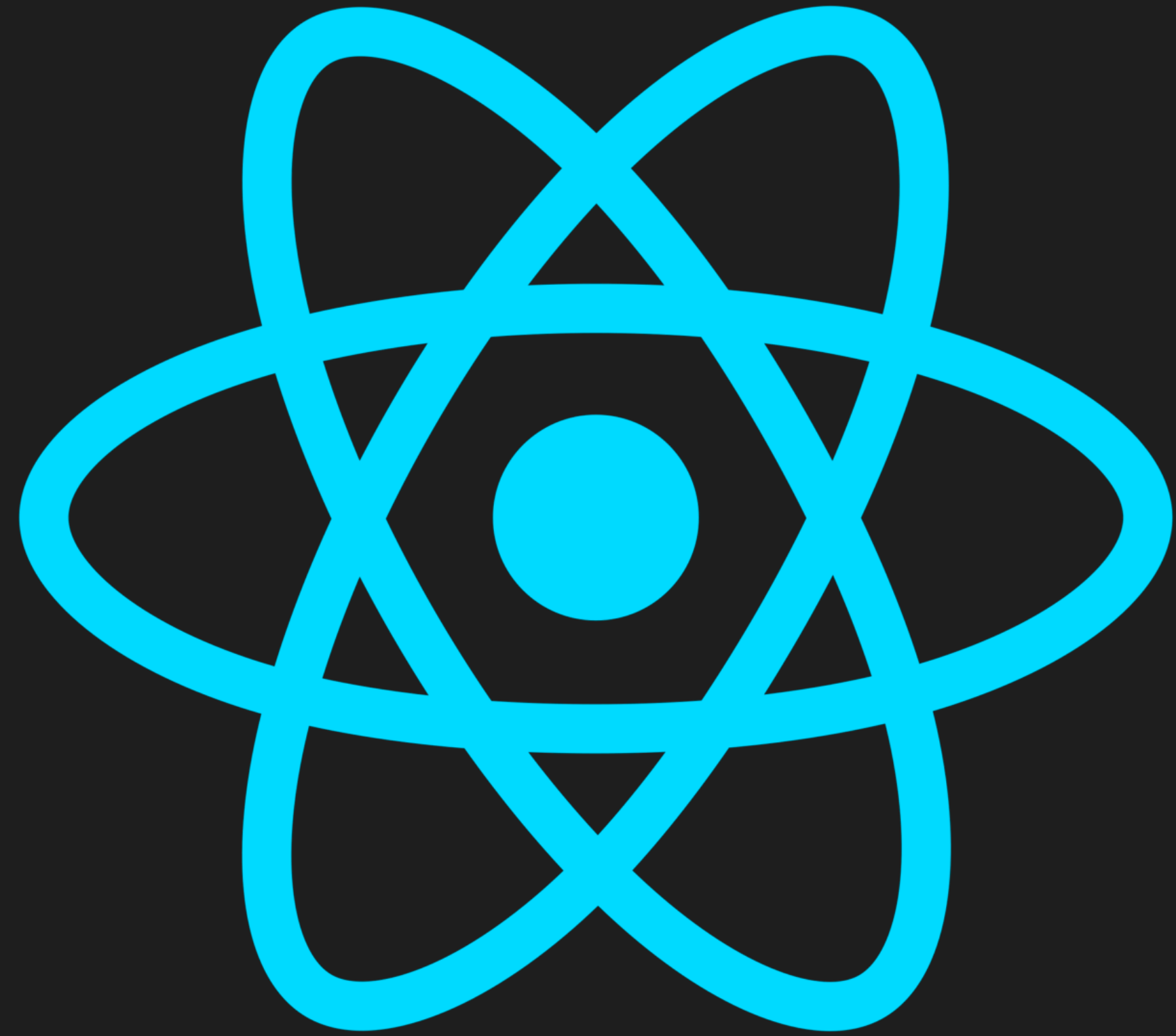
USADO POR GRANDES EMPRESAS. CRIADO E MANTIDO PELA GOOGLE QUE GARANTE BOM SUPORTE E COMUNIDADE ATIVA.

PONTOS NEGATIVOS

NECESSÁRIO CONHECIMENTOS EM JAVASCRIPT AVANÇADO

VERBOSO;

POUCO FLEXÍVEL, COMO POSSUI UMA ESTRUTURA SÓLIDA, É DIFÍCIL DESVIAR DISSO. (O QUE PODE SER BOM)



LIBRARY

COMPONENTS

FILOSOFIA

ORIENTADO À COMPONENTES (COMPONENTES REATIVOS)

USA FORTEMENTE O PARADIGMA FUNCIONAL (APESAR DE NO PASSADO TER COMEÇADO COM OO)

*** JSX**

PONTOS POSITIVOS

CONCEITOS DE PROGRAMAÇÃO FUNCIONAL TENDEM A SER MENOS VERBOSO QUE OO, O QUE PODE FACILITAR O DESENVOLVIMENTO;

FLEXIBILIDADE PARA CRIAR APLICAÇÕES PEQUENAS OU GRANDES;

LIBERDADE DE TRABALHAR COMO QUISE, INCLUSIVE PODENDO USAR JAVASCRIPT OU TYPESCRIPT;

COMUNIDADE MUITO ATIVA E PREOCUPAÇÃO COM INICIANTE. CRIADA E MANTIDA PELO FACEBOOK, USADA POR PLATAFORMAS COMO TWITTER E NETFLIX.

PONTOS NEGATIVOS

NECESSÁRIO CONHECIMENTOS EM JAVASCRIPT AVANÇADO (ECMAScript);

A FLEXIBILIDADE PODE SER PREJUDICIAL POIS NÃO AJUDA QUEM DESENVOLVE A MANTER UMA BOA ORGANIZAÇÃO;

COMO EVITA ASSUMIR MUITAS RESPONSABILIDADES, O USO DE OUTRAS BIBLIOTECAS É CONSTANTE E, ALÉM DE APRENDER A DESENVOLVER COM REACT, TAMBÉM É IMPORTANTE APRENDER OUTRAS BIBLIOTECAS ESPECÍFICAS (COMO O AXIOS);



LIBRARY

COMPONENTS

FILOSOFIA

ORIENTADO À COMPONENTES (COMPONENTES REATIVOS)

BASEADOS NAS SEPARAÇÕES DE RESPONSABILIDADE DO FRONTEND, TEMPLATE (HTML), SCRIPT (JS) E STYLE (CSS)

VERSÁTIL

PONTOS POSITIVOS

CURVA DE APRENDIZADO BAIXA;

VERSÁTIL;

COMUNIDADE ATIVA E INTERESSADA, E TAMBÉM ABERTA A MELHORIAS;

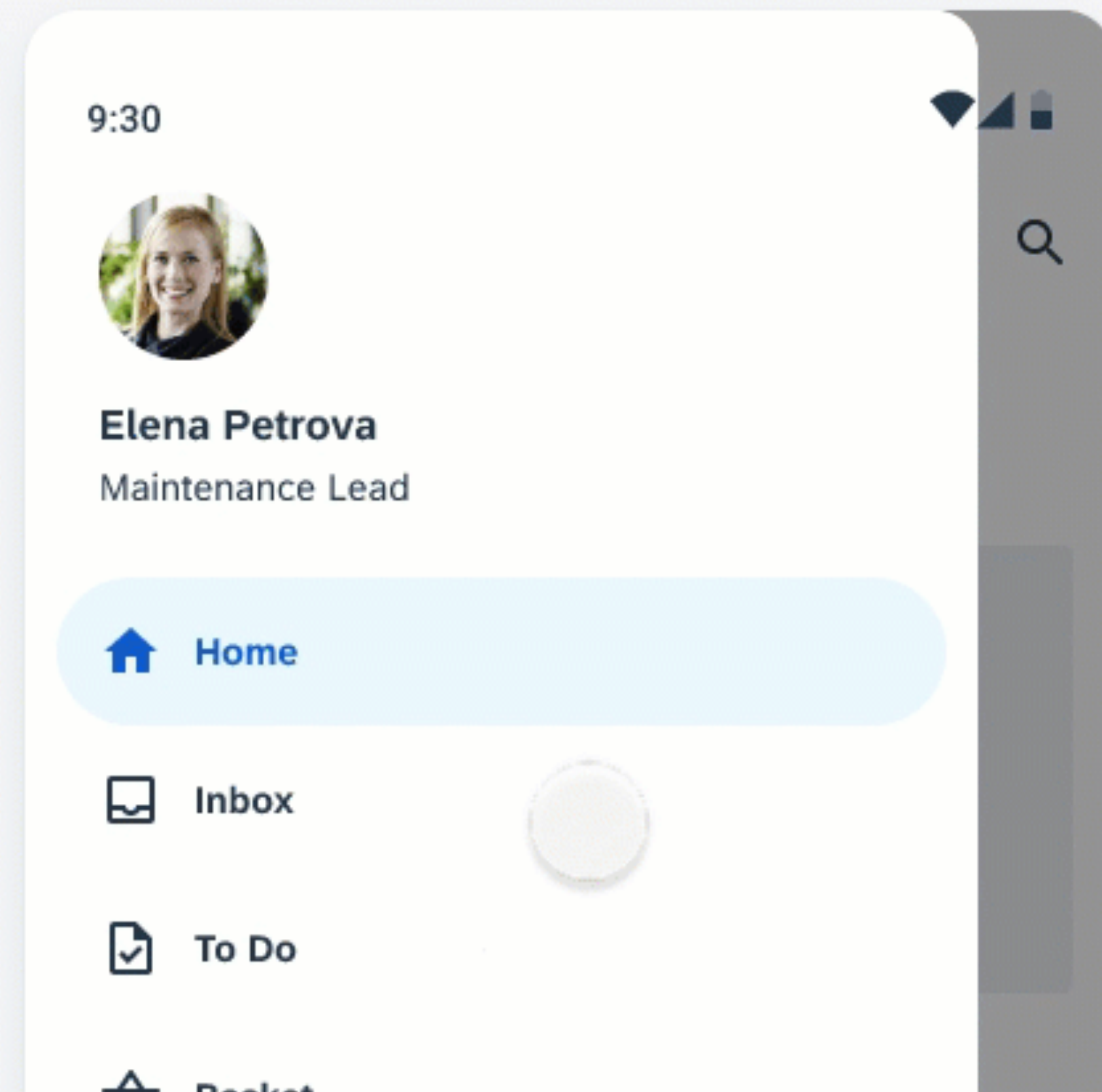
TEM MUITAS CARACTERÍSTICAS DO ANGULAR E DO REACT;

PONTOS NEGATIVOS

SUA POPULARIDADE É MAIS RECENTE, E POR ISSO É MENOS ADOTADO NO MERCADO.

CASOS

REACT VS VUE



REACT

```
const ModalOverlay = styled(Box)`  
  bottom: 0;  
  left: 0;  
  position: fixed;  
  right: 0;  
  top: 0;  
  background-color: ${props => props.theme.components.modal.overlayBG};  
  transform: translateY(200%);  
  z-index: 1000;  
  
  ${props => props.isOpen && 'transform: translateY(0%);'}  
`;  
`;
```


VUE

```
1 <template>
2   <nav
3     class="menu-top"
4     v-bind:class="{ 'open': isOpen }"
5     role="navigation"
6     aria-label="Main"
7   >
8     <ul class="menu-top__list">
9       <li>
10        <button-toggle-sidemenu @click="isOpen = !isOpen"/>
11      </li>
```



Cosméticos

Jequiti

Brasil no corpo e na alma

VUE

```
86 <style scoped lang="stylus">
87   .menu-top
88     position fixed
89     left 0
90     top 0
91     background white
92     width 100%
93     max-width 60px
94     height 100%
95     padding 20px 0
96     box-sizing border-box
97     box-shadow 0 0 18px -10px #ee4f65
98     text-align center
99     z-index 10
100    transition .5s all
101
```

VUE

```
122 .router-link-exact-active.router-link-exact-active
123     font-weight bold
124     border-right 3px solid #e4455d
125     background #fef5f8
126     color #e4455d
127
128 .open
129     max-width 200px
130
131     li:nth-child(n+2) a
132         display flex
133         margin 10px 0
134         padding 25px 0 25px 20px
```

CASOS

REACT VS VUE

ENQUANTO NO REACT (**UTILIZANDO STYLED COMPONENTE**) PRECISEI DE **UMA ÚNICA LINHA** PARA FAZER O DRAWER, NO VUE PRECISEI FAZER UM **TOGGLE DE CLASSE** (NECESSÁRIO ESCREVER TANTO A CLASSE CSS DO MENU FECHADO, COMO ABERTO)



CASOS REACT VS VUE

ESSE PODER NÃO É DO REACT, MAS SIM DO **JSX** E **STYLED COMPONENTS**

CASOS

REACT VS VUE

JSX



```
<MyButton>  
  Click Me  
</MyButton>
```

REACT API



```
React.createElement(  
  MyButton,  
  {},  
  'Click Me'  
)
```

VUE ACEITA JSX



- [Aprenda](#)
- [Ecossistema](#)
- [Equipe](#)
- [Recursos](#)
- [Patrocine](#)
- [Traduções](#)

Funções de Renderização & JSX

Introdução

Nós, Árvores e DOM Virtual

DOM Virtual

Parâmetros para createElement

O Objeto de Dados em Detalhes

Exemplo Completo

Restrições

Substituindo por Código JavaScript

v-if e v-for

v-model

Eventos e Modificadores

Slots

JSX

Componentes Funcionais

Passando Atributos e Eventos para Elementos/Componentes

Filhos

slots() vs children

Compilação de Templates

Plugins

Filtros

Ferramentas

Componentes Single-File

Testes Unitários

Testing

Funções de Renderização & JSX

Introdução

Vue recomenda que *templates* sejam utilizados para construir seu HTML na grande maioria dos casos. Haverá situações, no entanto, em que você irá realmente precisar de todo o poder de programação do JavaScript. É quando você pode usar a **função render**, uma alternativa aos *templates* mais próxima do compilador.

Vamos mergulhar em um exemplo simples onde a função **render** seria prática. Digamos que você quer gerar cabeçalhos (*headings*) com âncoras:

```
<h1>
  <a name="hello-world" href="#hello-world">
    Olá Mundo!
  </a>
</h1>
```

Para o código HTML acima, você decide que quer esta interface para o componente:

```
<anchored-heading :level="1">Olá Mundo!</anchored-heading>
```

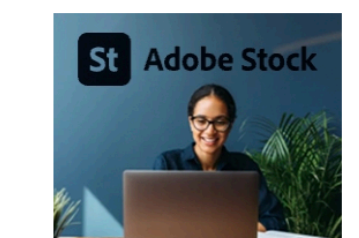
Quando você começar a criar o componente que gera um cabeçalho de acordo com a propriedade **level**, rapidamente chegará a algo assim:

```
<script type="text/x-template" id="anchored-heading-template">
  <h1 v-if="level === 1">
    <slot></slot>
  </h1>
  <h2 v-if="level === 2">
```

Patrocinadores Platina



Seja um Patrocinador



Get 10 free Adobe Stock photos. Start downloading amazing royalty-free stock photos today.

ADS VIA CARBON

Para o código HTML acima, você decide que quer esta interface para o componente:

```
<anchored-heading :level="1">Olá Mundo!</anchored-heading>
```

HTML

Quando você começar a criar o componente que gera um cabeçalho de acordo com a propriedade `level`, rapidamente chegará a algo assim:

```
<script type="text/x-template" id="anchored-heading-template">
  <h1 v-if="level === 1">
    <slot></slot>
  </h1>
  <h2 v-else-if="level === 2">
    <slot></slot>
  </h2>
  <h3 v-else-if="level === 3">
    <slot></slot>
  </h3>
  <h4 v-else-if="level === 4">
    <slot></slot>
  </h4>
  <h5 v-else-if="level === 5">
    <slot></slot>
  </h5>
  <h6 v-else-if="level === 6">
    <slot></slot>
  </h6>
</script>
```

HTML

Enquanto *templates* funcionam muito bem para a maioria dos componentes, está claro que este caso é uma exceção. Então vamos tentar reescrevê-lo usando uma função `render` :

```
Vue.component('anchored-heading', {
  render: function (createElement) {
    return createElement(
      'h' + this.level, // nome do elemento (tag)
      this.$slots.default // vetor de elementos filhos
    )
  },
  props: {
    level: {
      type: Number,
      required: true
    }
  }
})
```

JS

JSX

Se você estiver escrevendo muitas funções `render`, pode se tornar cansativo e passível de erros escrever muitas linhas de código como essas:

```
createElement(  
  'anchored-heading', {  
    props: {  
      nivel: 1  
    }  
  }, [  
    createElement('span', 'Alô'),  
    ' Mundo!'  
  ]  
)
```

JS

Especialmente quando a versão usando *template* é tão simples em comparação:

```
<anchored-heading :nivel="1">  
  <span>Alô</span> Mundo!  
</anchored-heading>
```

HTML

Por isso há um **plugin para Babel** destinado à utilização de JSX com o Vue, nos trazendo de volta a uma sintaxe mais semelhante à utilizada em *templates*:

STYLED COMPONENTS

NO VUE

The screenshot shows the GitHub repository page for `styled-components/vue-styled-components`. The repository is public and has 17 watchers, 105 forks, and 1.3k stars. The main content area displays a list of files and folders with their commit history. The right sidebar contains repository metadata and statistics.

Repository Information:

- Repository: `styled-components / vue-styled-components` (Public)
- Watch: 17
- Fork: 105
- Starred: 1.3k
- Code: 13
- Issues: 13
- Pull requests: 24
- Actions: 24
- Projects: 24
- Security: 24
- Insights: 24

Files and Folders:

File/Folder	Commit Message	Time Ago
example	feat: added .attrs examples	3 years ago
src	add 'attrs' support	3 years ago
.babelrc.js	Rename babelrc.js	4 years ago
.editorconfig	Initial commit 🎉	7 years ago
.eslintignore	Updated README, .eslintconfig, utils methods 🚀	6 years ago
.eslintrc.js	Fixed typo in var name, removed useless config	6 years ago
.gitignore	Ignore .idea folder which is used for IDEs such as PhpStorm and Web...	5 years ago
.npmignore	Updated .npmignore and .gitignore 📖	6 years ago
LICENSE	Updated LICENSE and README 📖	7 years ago
README.md	docs(readme): updated readme docs with polymorphic as prop	4 years ago
mocha-bootstrap.js	setup jsdom-global to test vue component properly	5 years ago
package-lock.json	chore: bump 1.6.0 closing #68 #94 #102 #113 #83 #104	3 years ago
package.json	chore: bump 1.6.0 closing #68 #94 #102 #113 #83 #104	3 years ago
rollup.config.js	Merged #76 #77 #78	4 years ago

Repository Metadata:

- About: Visual primitives for the component age. A simple port for Vue of styled-components 🙌
- Tags: `css`, `vue`, `css-in-js`, `vue-styled-components`
- Readme
- MIT license
- 1.3k stars
- 17 watching
- 105 forks
- Report repository

Releases:

- 22 tags

Packages:

- No packages published

Used by: 1.3k

Used by icons:



```
const ModalOverlay = styled(Box)`  
  bottom: 0;  
  left: 0;  
  position: fixed;  
  right: 0;  
  top: 0;  
  background-color: ${props => props.theme.components.modal.overlayBG};  
  transform: translateY(200%);  
  z-index: 1000;  
  
  ${props => props.isOpen && 'transform: translateY(0%);'}  
`;  
;
```


COMPONENTES REUTILIZÁVEIS
É DIFERENTE DE
WEB COMPONENTS

Browser support



CHROME



OPERA



SAFARI



FIREFOX



EDGE



HTML TEMPLATES



CUSTOM ELEMENTS



SHADOW DOM



ES MODULES



Elements



```
<!DOCTYPE html>  
<html lang="en-us" class>  
  ▶ <head> ... </head>
```

```
... ▼ <body> == $0
```

```
  ▶ <catalog-app page="home"> ... </catalog-app>  
    <script async src="https://www.google-analytics.com/analytics.js">  
    </script>  
  ▶ <script> ... </script>  
    <script src="/bower_components/webcomponentsjs/webcomponents-lite.min.  
    js"></script>  
  ▶ <iron-a11y-announcer> ... </iron-a11y-announcer>  
  </body>  
</html>
```