

# OverVue: conhecendo o framework Vue.js





# Karoline Setoue

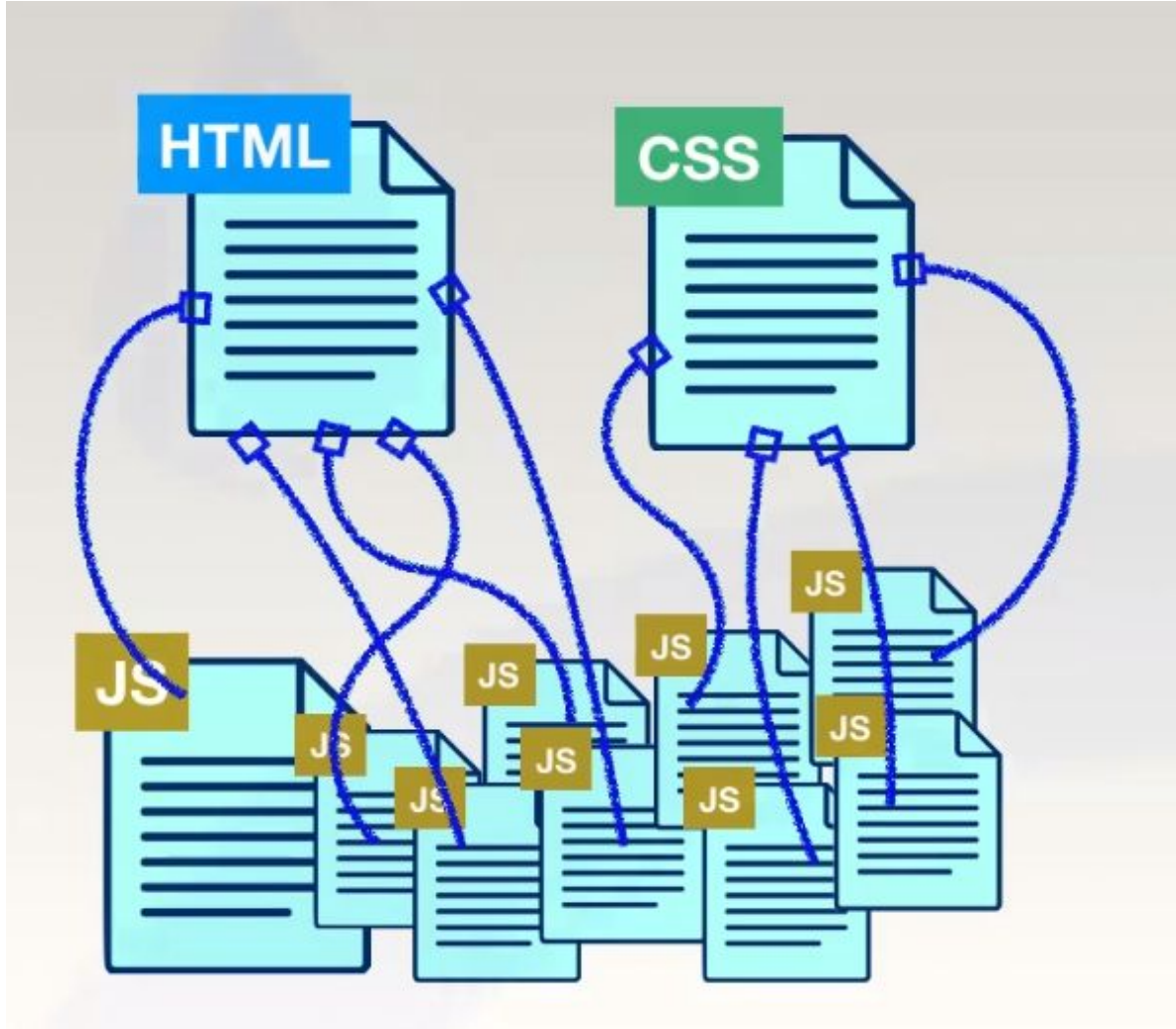
 [linkedin.com/in/karoline-setoue/](https://linkedin.com/in/karoline-setoue/)

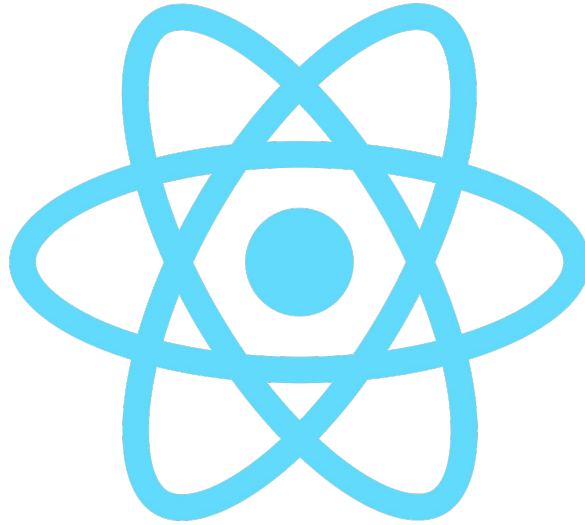
 [facebook.com/karoline.setoue](https://facebook.com/karoline.setoue)

 [github.com/ksetoue](https://github.com/ksetoue)









# DOM

- Representação dos componentes na página
- Árvore de componentes

# Virtual DOM

- Representação virtual do DOM
- Aqui entram os frameworks



# Vue.js



# O que é Vue.js?

- Framework progressivo de js para construção de interfaces



# O que é Vue.js?

- Framework progressivo de js para construção de interfaces
- Indicado para construção de SPAs (Single Page Application)





# O que é Vue.js?

- Framework progressivo de js para construção de interfaces
- Indicado para construção de SPAs (Single Page Application)
- Renderização via Virtual Dom



# O que é Vue.js?

- Framework progressivo de js para construção de interfaces
- Indicado para construção de SPAs (Single Page Application)
- Renderização via Virtual Dom
- Componentes!



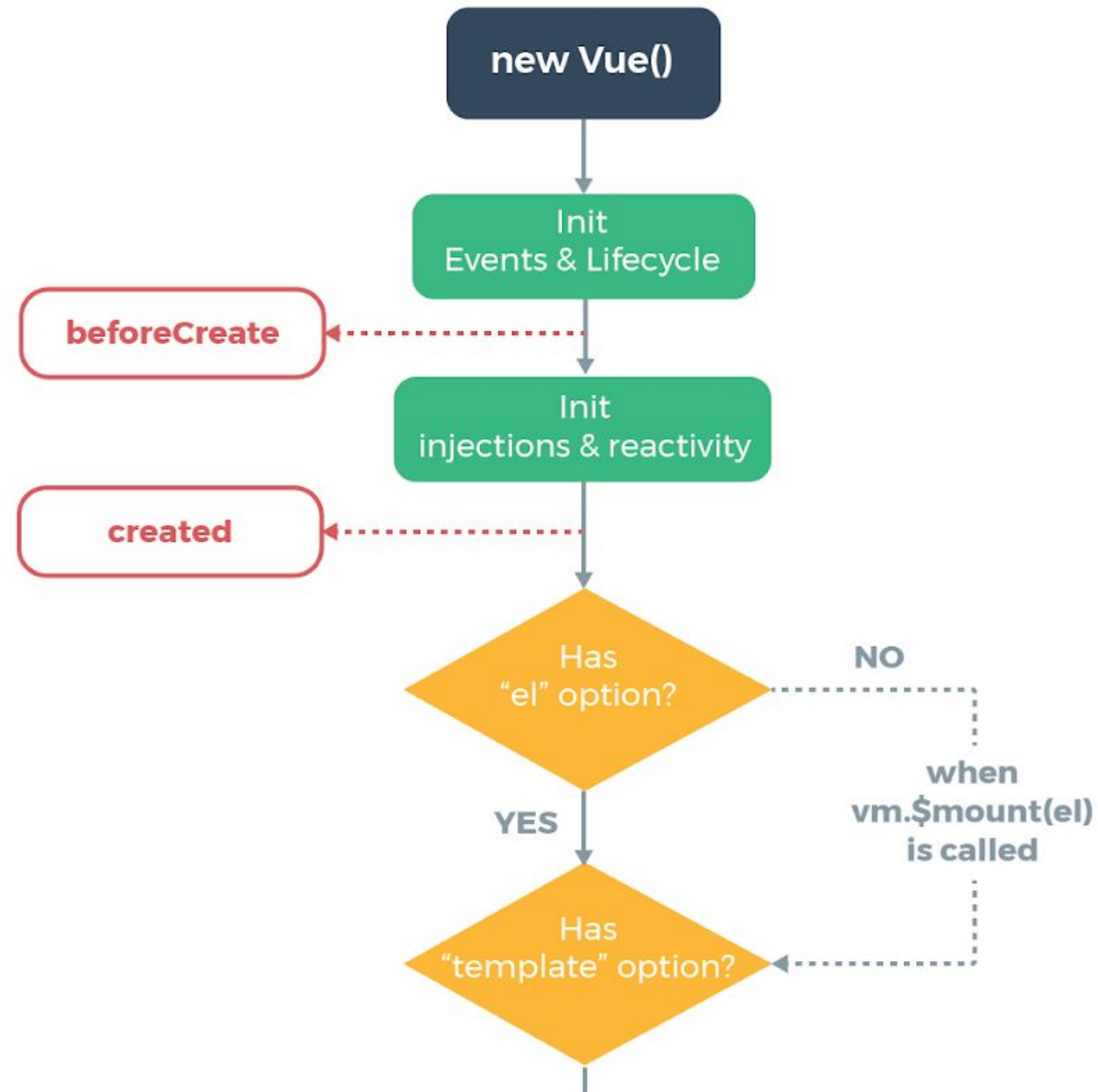
# O que é Vue.js?

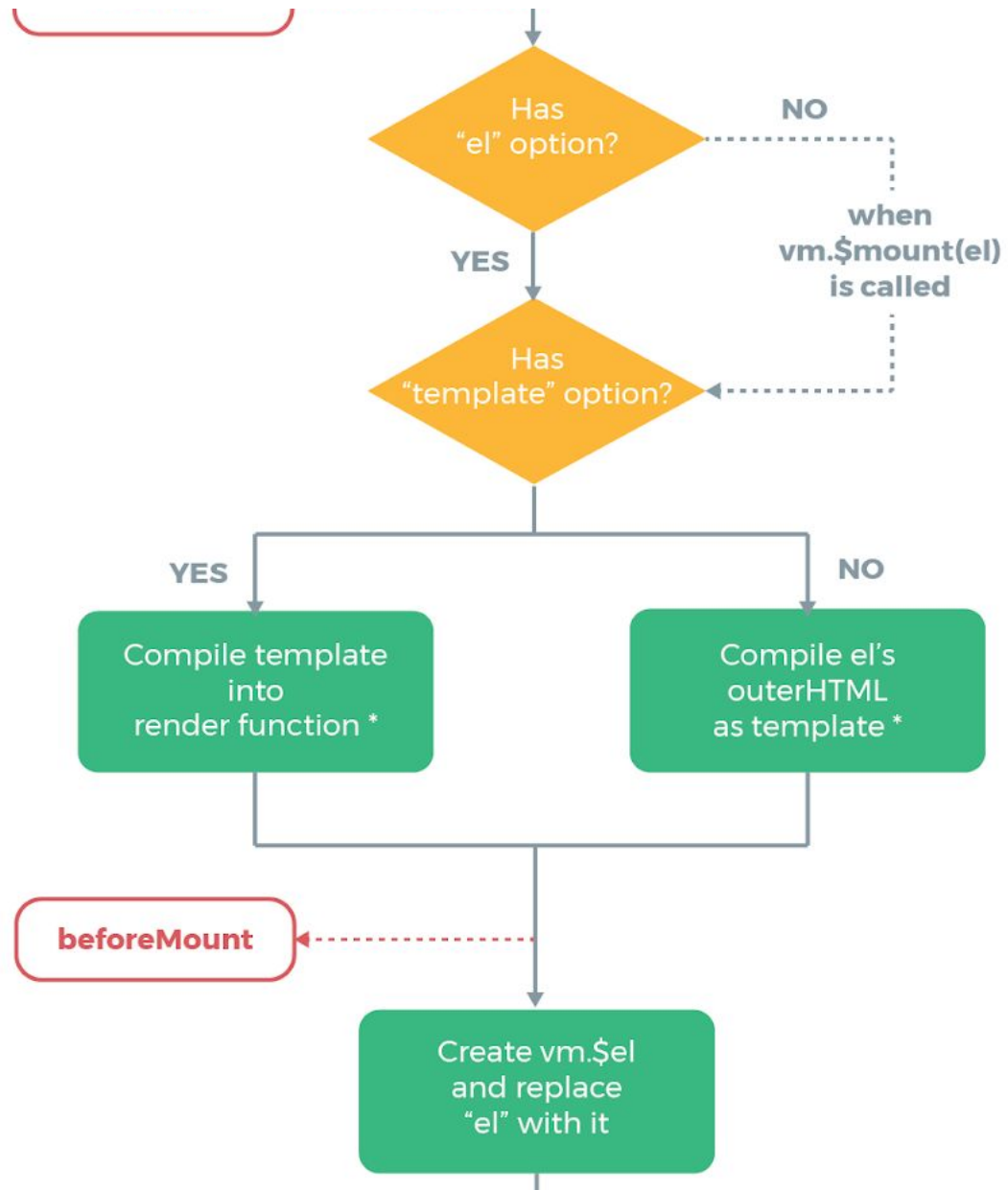
- Framework progressivo de js para construção de interfaces
- Indicado para construção de SPAs (Single Page Application)
- Renderização via Virtual Dom
- Componentes!
- Reatividade!



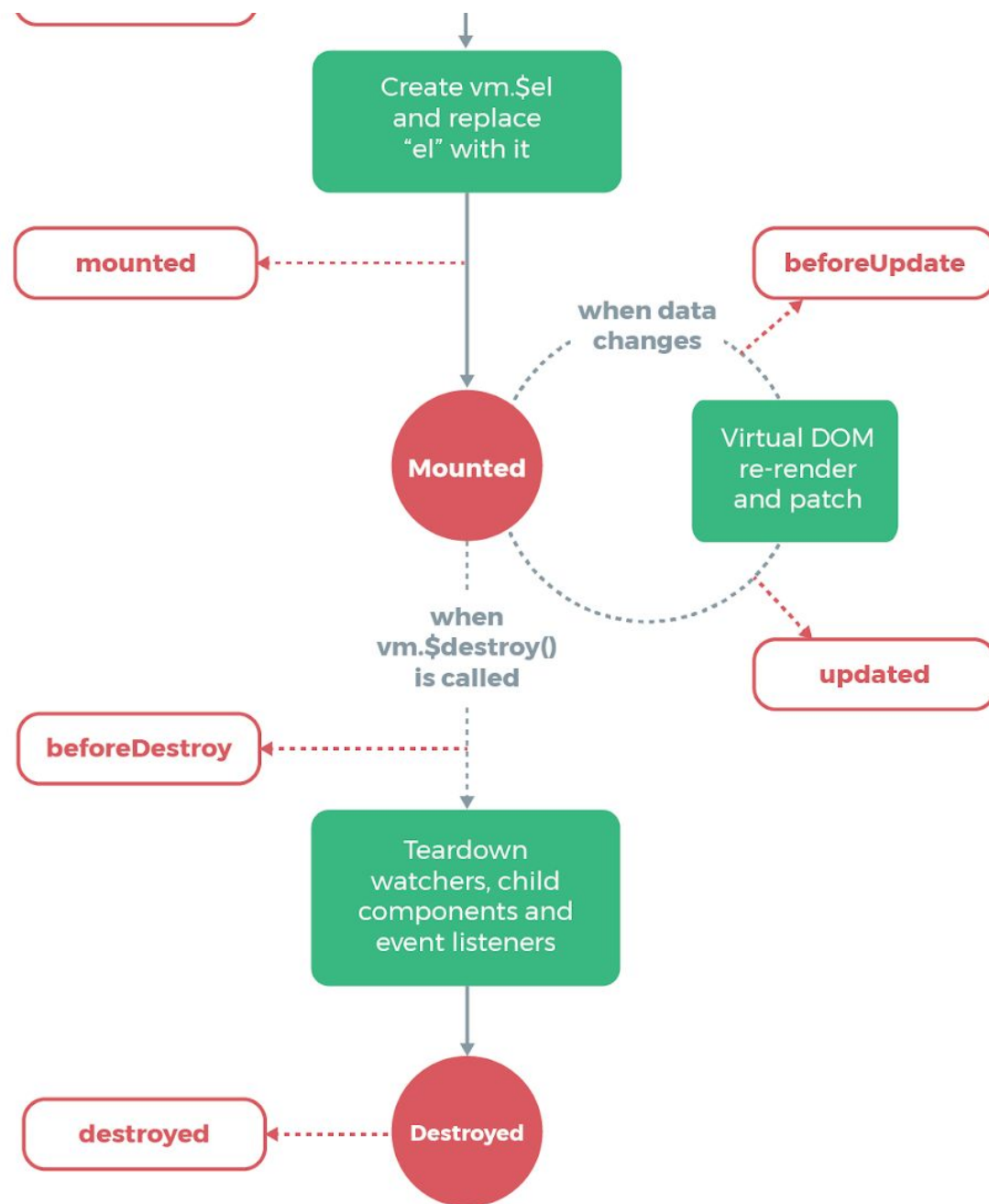
# Por onde começar?











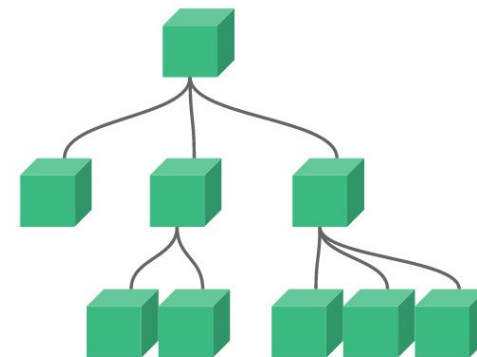
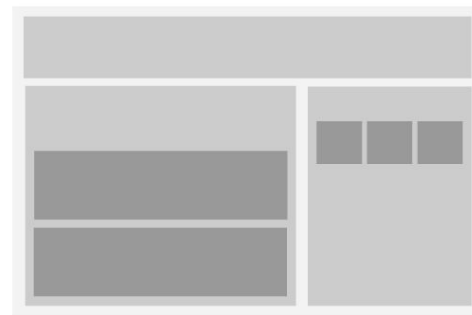
Toda aplicação em Vue,  
começa com a criação  
da instância do Vue

```
1 import Vue from 'vue'
2 import App from './App.vue'
3
4 new Vue({
5   el: '#app',
6   render: h => h(App)
7 })
8
9
```



# Componentes

- Conceito importante do Vue
- Permite escalabilidade
- ideia de componentes pequenos e reutilizáveis



```
1 <template>
2   <div class="hello">
3     <h1> Component de {{ name }} </h1>
4   </div>
5 </template>
6
7 <script>
8 export default {
9   name: 'HelloWorld',
10  data () {
11    return {
12      name: 'Menu'
13    }
14  },
15 }
16 </script>
17
18
19 <style scoped>
20 h1, h2 {
21   font-weight: normal;
22   color: #42b983;
23 }
24 </style>
```

Componentes podem ser reutilizados

Quando o template é o mesmo mas o conteúdo (dado) muda, você pode transformar o código em componente



```
1 <template>
2   ..<div class="hello">
3     ...<h1> Component de {{ name }} </h1>
4   ..</div>
5 </template>
6
7 <script>
8 export default {
9   ..name: 'HelloWorld',
10   ..data () {
11     ...return {
12       ...name: 'Menu'
13     }
14   },
15 }
16 </script>
17
18
19 <style scoped>
20 h1, h2 {
21   ..font-weight: normal;
22   ..color: ■ #42b983;
23 }
24 </style>
```

As definições de dados  
estão dentro dos  
componentes



```

1  <template>
2  ..<div class="hello">
3  .....<h1>Outro Componente</h1>
4  .....<button @click="addCounter()">Me clique</button>
5  .....<p>O valor do contador é: ui + {{ counter }}</p>
6  ..</div>
7  </template>
8
9  <script>
10 export default {
11   ..name: 'OutroComponente',
12
13   ..data () {
14     ....return {
15       ....msg: 'Olar! esse é o seu primeiro App em Vue.js',
16       ....counter: 0
17     }
18   },
19
20   ..methods: {
21     ....addCounter () {
22       ....this.counter++
23     }
24   }
25
26 }
27 </script>
28
29 <style scoped>
30 </style>
31

```

Cada componente tem  
seus próprios métodos





# Exemplo



# Outros elementos

- v-if
- v-show
- v-bind
- this.\$emit
- props
- mixins
- v-model

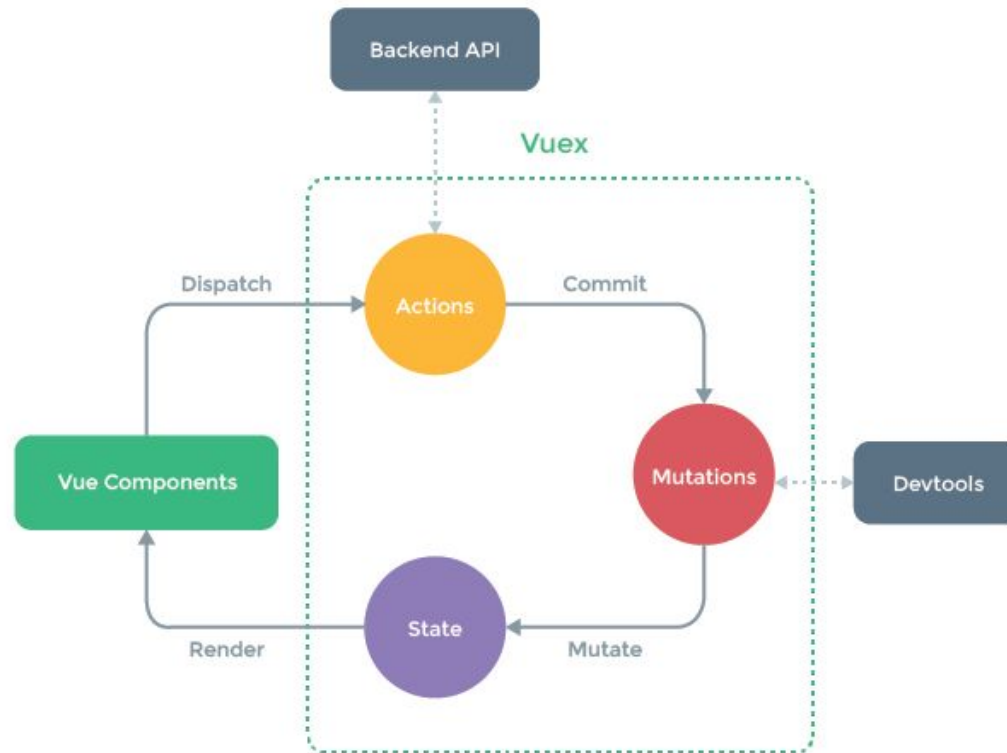


# Vuex

- Lib/padrão de gerenciamento de estados
- armazenamento centralizado de dados
- alterações nos dados feitas de modo previsível usando mutations



# principais componentes



vuex todomvc example

Bill

localhost:8080/todomvc/

todos

What needs to be done?

ElementsVueConsoleSourcesNetworkTimelineProfilesApplicationSecurityAuditsRedux

Ready. Detected Vue 2.1.8.

ComponentsVuexEventsRefresh

Filter mutationsCommit AllRevert AllRecordingExportImport

Base Stateinspectedactive15:36:18

state: Object

todos: Array[0]



“Flux libraries are like glasses: you’ll know when you need them.”

- Dan Abramov, criador do Redux





# Vue-cli

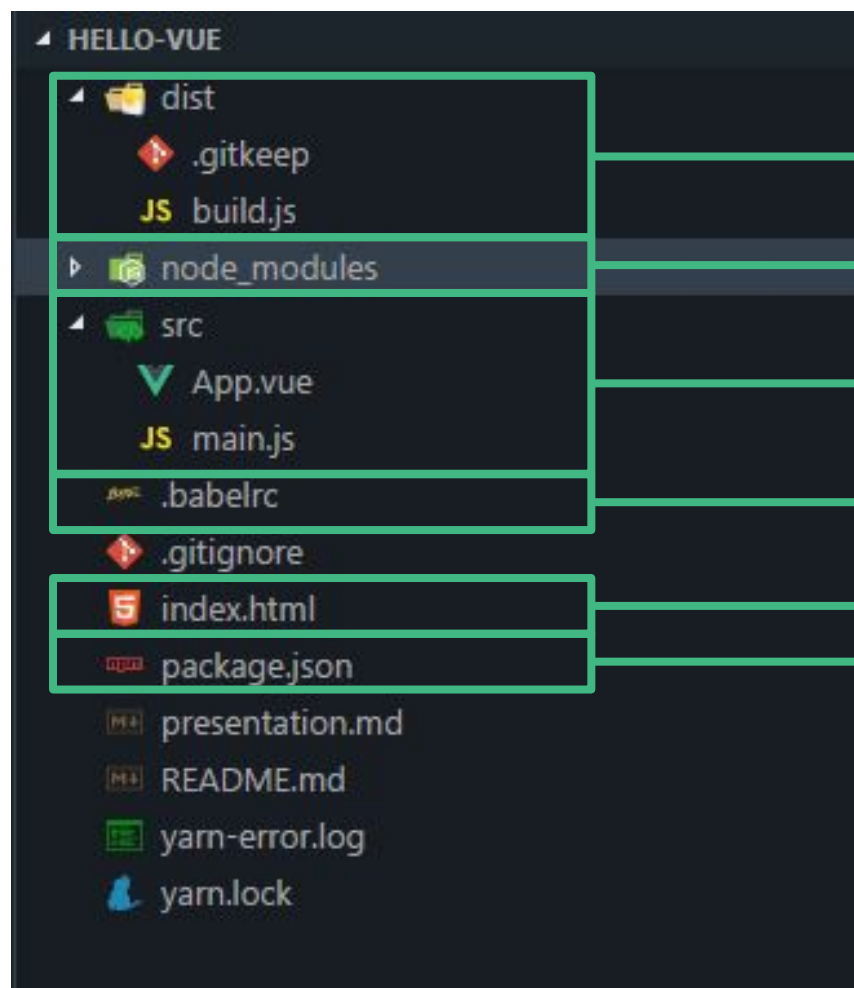


# Templates disponíveis

- webpack
- webpack simple
- browserify
- browserify-simple
- simple



# Estrutura



Onde o arquivo "compilado" vai ficar - todos os arquivos javascript são colocados em um só

pacotes do node que fazem parte do projeto

Pasta com os arquivos da sua aplicação

configurações que ajudam o bundler

Arquivo HTML principal da aplicação

Arquivo de configuração do projeto

Iniciando o projeto:

```
vue init <template> <nome_projeto>
```



# Links úteis

- Guide do Vue: <https://vuejs.org/v2/guide/>
- Docs do Vuex: <https://vuex.vuejs.org/>
- Um artigo bem bacana sobre escolha de tecnologia no projeto:  
<https://medium.com/reverde/why-we-moved-from-angular-2-to-vue-js-and-why-we-didnt-choose-react-ef807d9f4163>





# Obrigada! :)

 [linkedin.com/in/karoline-setoue/](https://linkedin.com/in/karoline-setoue/)

 [facebook.com/karoline.setoue](https://facebook.com/karoline.setoue)

 [github.com/ksetoue](https://github.com/ksetoue)

