Vue

Matheus Alberto - 2019005696
Jhonatas Andrade - 2019004705
Wendel Luiz - 2020022457

Tópicos

- Introdução
- Utilização
- Arquitetura
- Ferramentas
- Vantagens e desvantagens
- Comparações
- Vue SFC
- Diretivas
- Tutorial

Introdução

- Vue.js é um framework progressivo para a construção de interfaces de usuário
- Muito usado para criar aplicações single page e interfaces que possuem necessidades de maior interação e experiência para o usuário
- Framework Progressivo

Utilização

Utilizado para a criação de websites de alto desempenho

- Foco na modularização, versatilidade e facilidade de aprendizado
- Requer uma configuração mínima na criação de um projeto

Arquitetura

- Estrutura de componentes
- Escopo isolado
- Renderização com DOM virtual apenas quando necessário

Ferramentas

- Otimização do processo de desenvolvimento - Vue CLI
- Arquitetura de plug-ins
- Vue-devtools
- Documentação oficial e comunidade ativa

Vantagens

- Fácil aprendizagem;
- Versátil;
- Progressivo;
- Desempenho ágil;
- Experiência do usuário;
- Escalabilidade.

Desvantagens

- Complexidade de paradigma;
- Excesso de flexibilidade;
- Recursos um pouco mais limitados em relação a outros frameworks SPA;

Comparação - Vue x React

- Popularidade e mercado de trabalho
- Curva de aprendizagem
- Desempenho (Velocidade)
- Componentes e extensibilidade
- Gerenciamento do estado
- Ecossistema
- Segurança
- Suporte e comunidade

Diretivas

- v-if
- v-for
- v-model
- v-bind
- v-on

Diretivas: v-if

Adiciona e/ou remove um elemento do HTML de acordo com o retorno de uma função ou uma variável

App.vue

```
<div v-if="Math.random() > 0.5">
  Now you see me
</div>
<div v-else>
  Now you don't
</div>
```

App.vue

```
<div v-if="type === 'A'">
  A
</div>
<div v-else-if="type === 'B'">
  B
</div>
<div v-else-if="type === 'C'">
  C
</div>
<div v-else>
  Not A/B/C
</div>
```

Diretivas: v-for

Como qualquer implementação de um laço em um array, como em qualquer linguagem

App.vue

```
<div v-for="item in items">
  {{ item.text }}
</div>
```

App.vue

```
<div v-for="(item, index) in items"></div>
<div v-for="(value, key) in object"></div>
<div v-for="(value, name, index) in object"></div>
```

Diretivas: v-model

Diretiva que recebe uma variável e uma ligação bidirecional entre o HTML e o controller

App.vue

```
<script>
import CustomInput from './CustomInput.vue'

export default {
  components: { CustomInput },
  data() {
    return {
      message: 'hello'
    }
  }
}
</script>

<template>
  <CustomInput v-model="message" /> {{ message }}
</template>
```

Diretivas: v-bind

Usada para ligar um ou mais atributos ou um componente prop a um elemento

App.vue

```
<!-- bind an attribute -->
<img v-bind:src="imageSrc" />

<!-- dynamic attribute name -->
<button v-bind:[key]="value"></button>

<!-- binding an object of attributes -->
<div v-bind="{ id: someProp, 'other-attr': otherProp }"></div>

<!-- pass down parent props in common with a child component -->
<MyComponent v-bind="$props" />
```

Diretivas: v-on

Escuta eventos do DOM e executa JavaScript quando um evento é disparado

App.vue

```
<!-- method handler -->
<button v-on:click="doThis"></button>

<!-- dynamic event -->
<button v-on:[event]="doThis"></button>

<!-- inline statement -->
<button v-on:click="doThat('hello', $event)"></button>

<!-- the click event will be triggered at most once -->
<button v-on:click.once="doThis"></button>

<!-- object syntax -->
<button v-on="{ mousedown: doThis, mouseup: doThat }"></button>
```

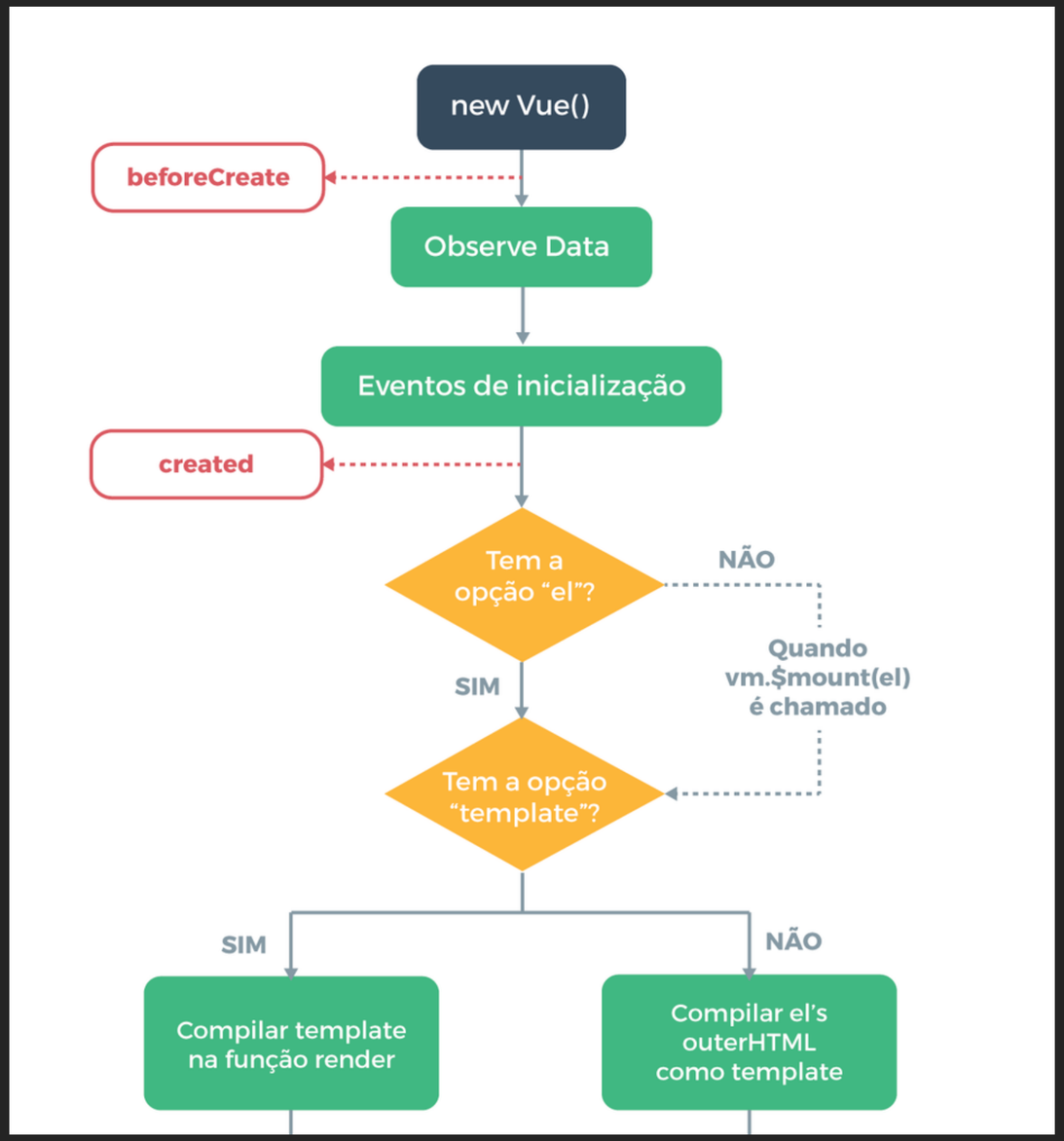
Vue Single File Component (SFC)

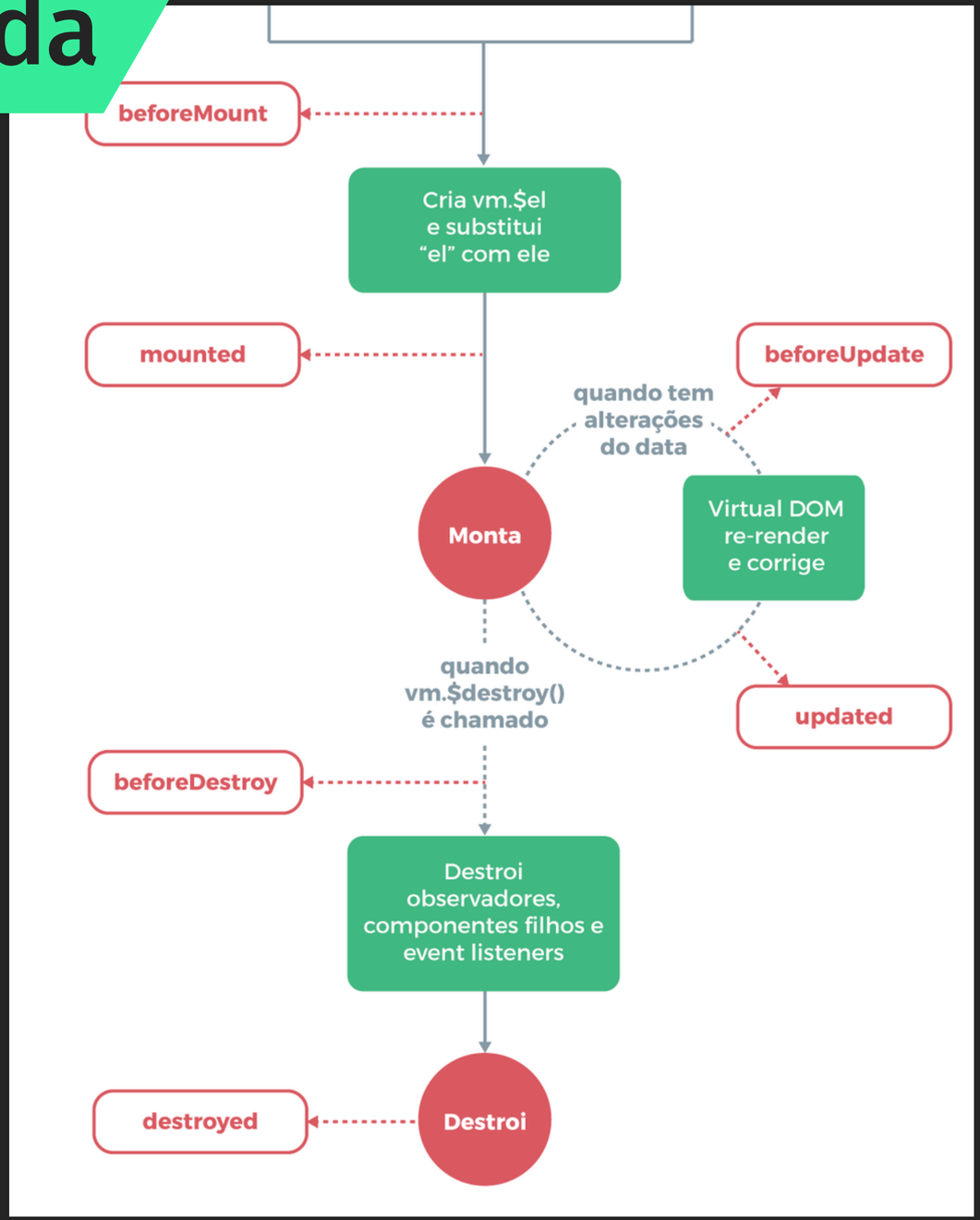
- Um SFC é um bloco reutilizável de código. Essa abordagem encapsula HTML, CSS e JavaScript que contém relação uns com os outros. O formato do arquivo de um SFC é “.vue”.
- Cada arquivo .vue consiste em três tipos principais de blocos: <template>, <script> e <style>.

Ciclo de vida da instância

- Cada instância Vue passa por uma série de etapas em sua inicialização, por exemplo: configurar a observação de dados, compilar o template, montar a instância no DOM, atualizar o DOM quando os dados forem alterados.
- Gatilhos do ciclo de vida: beforeCreate, created, beforeMount, mounted, beforeUpdate, updated, beforeDestroy, destroyed.

Diagrama do Ciclo de Vida





Componentes

- Componentes são instâncias reutilizáveis;

```vue
<script>
import ButtonCounter from './ButtonCounter.vue'

export default {
  components: {
    ButtonCounter
  }
}
</script>

<template>
  <h1>Here is a child component!</h1>
  <ButtonCounter />
</template>
```

Props

- Propriedades são atributos personalizados que você pode registrar em um componente;

```
<!-- BlogPost.vue -->
<script>
export default {
  props: ['title']
}
</script>

<template>
  <h4>{{ title }}</h4>
</template>
```

Tutorial


```
> npm init vue@latest
```

- Inicializará o processo de instalação.

```
> cd <nome-do-projeto>
> npm install
> npm run dev
```

- Instalará as dependências e Inicializará o aplicativo no modo "watch".

Tutorial On-line

https://vuejs.org/tutorial