

Laço de repetição e diretivas personalizadas



DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

AGENDA

- Conceito
- Renderização de texto e HTML
- Bind e actions
- Interligações em formulários
- Diretivas condicionais
- **Loops**
- **Diretivas personalizadas**

Diretiva: v-for

Podemos utilizar a diretiva v-for para renderizar uma lista de elementos com base nos dados de um Array. A diretiva requer uma sintaxe especial na forma de item in items, onde items é a fonte de dados e item é um **apelido** para o elemento que estiver sendo iterado.

```
<div class="mt-2">
  <ul>
    <li v-for="item in lista">
      {{ item }}
    </li>
  </ul>
</div>
```

- Item1
- Item2
- Item3
- Item4

Diretiva: v-for key

É recomendado oferecer um atributo key para v-for sempre que possível, a menos que esteja iterando conteúdo DOM simples, ou esteja intencionalmente se apoiando no comportamento padrão para ganho de desempenho, pois o Vue irá poder rastrear a identidade de cada nó e assim reutilizar e reordenar elementos existentes, defina um atributo key único para cada item

```
<ul>
  <li v-for="item in lista" :key="item">
    {{ item }}
  </li>
</ul>
```

```
<ul>
  <li v-for="(item, index) in lista" :key="index">
    {{ item }}
  </li>
</ul>
```

Diretiva personalizada

Vue permite registrar suas próprias diretivas personalizadas. A forma primária de abstração e reuso de código são componentes - no entanto, pode haver casos em que você só precisa de um acesso de baixo nível ao DOM em elementos simples, e aí diretivas personalizadas seriam úteis



The image shows a Vue.js directive example. It features a large text label "Gilmar" in a bold, dark font. Below it, the label "Nome" is positioned above a text input field. The input field contains the text "Gilmar" and is highlighted with a blue border, indicating it is the active element. This visualizes a custom directive being applied to a standard HTML input element.

Registrando uma diretiva

No Vue, ao criar a aplicação no main, podemos criar uma diretiva pela propriedade "directive" do app. Após criar a aplicação basta inserir as diretivas passando como parâmetro primeiramente o nome seguido de uma função que chamará as ações da diretiva

```
import { createApp } from 'vue'
import App from './App.vue'

let app = createApp(App);

//Registra a diretiva v-color-red
app.directive('color-red', {
  updated(el) {
    el.style = 'color:red'
  }
})

app.mount('#app')
```

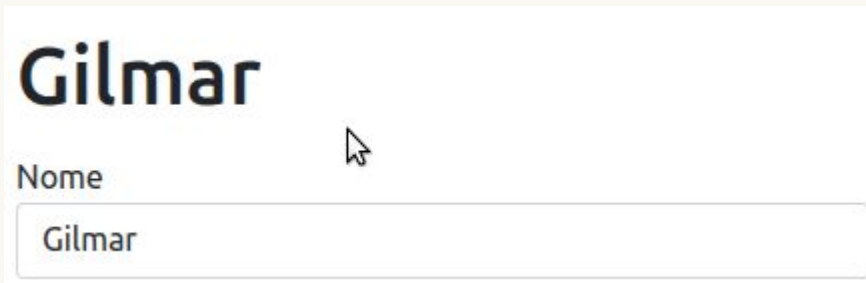
```
<h1 v-text="reserva.nome" v-color-red></h1>
<div class="col-md-4">
  <label for="">Nome</label>
  <input type="text" class="form-control" v-model="reserva.nome" required>
</div>
```

Funções de gatilhos (Hook Functions)

As funções de hook são as funções que são executadas conforme o estado da diretiva e há uma variedade disponível para uso (todas opcionais), como por exemplo: “created”, “beforeMounted”, “mounted”, “beforeUpdate”, “updated”, “beforeUnmounted” e “unmounted”.

```
//Registra a diretiva v-color-red
app.directive('color-red', {
  beforeMount(el) {
    el.addEventListener('click', () => {
      el.style = 'color:blue'
    })
  },

  updated(el) {
    el.addEventListener('click', () => {
      el.style = 'color:red'
    })
  }
})
```



Podemos passar parâmetros para nossa diretiva através do “binding”, atributo opcional da função de nossa diretiva. O binding recupera o valor passado em diretiva pela variável “value”

```
//Registra a diretiva v-color
app.directive('color', {
  beforeMount(el, binding) {
    el.addEventListener('click', () => {
      el.style = `color:${binding.value}`
    })
  }
})
```

```
<h1 v-text="reserva.nome" v-color="'red'"></h1>
<div class="col-md-4">
  <label for="">Nome</label>
  <input type="text" class="form-control" v-model="reserva.nome" required>
</div>
```

Gilmar

Nome

Gilmar

Modifiers

Podemos criar modifiers, acessando pelo "binding.modifiers":

```
//Registra a diretiva v-color
app.directive('color', {
  beforeMount(el, binding) {
    el.addEventListener('click', () => {
      if (binding.modifiers.font) {
        el.style = `color:${binding.value}`
      } else if (binding.modifiers.background) {
        el.style = `background-color:${binding.value}`
      }
    })
  }
})
```

```
<h1 v-text="reserva.nome" v-color.font="'red'"></h1>
<h1 v-text="reserva.nome" v-color.background="'red'"></h1>
```

Gilmar

Gilmar

Nome

Gilmar

Por fim, podemos receber as informações do elemento através do “vnode”, recebida no terceiro parâmetro (opcional) da função:

```
//Registra a diretiva v-color
app.directive('color', {
  beforeMount(el, binding, vnode) {
    el.addEventListener('click', () => {
      if (vnode.type === 'h1') {
        el.style = `color:${binding.value}`
      } else {
        el.style = `color:${binding.value};font-size: 9px`
      }
    })
  }
})
```

```
<h1 v-text="reserva.nome" v-color="'red'"></h1>
<h2 v-text="reserva.nome" v-color="'red'"></h2>
```



Gilmar

Gilmar



DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!



<LAB365>