

04

## O poder das diretivas

### Transcrição

Estamos quase chegando ao final do capítulo, mas vem aquela pergunta: e se quisermos exibir mais de uma imagem, qual alteração será necessária em nosso código? Veja que a pergunta é totalmente válida pois `App` será a responsável pela exibição de todas as fotos da nossa aplicação.

Primeiro, vamos renomear a propriedade `foto` para `foto1` em nossa função `data`. Em seguida, vamos adicionar a nova propriedade `foto2` que aponta para outra imagem com outro nome. Com isso, vamos adicionar mais uma tag `img` no template e apontar para a nova propriedade usando o atalho para `v-bind`, o famoso dois pontos. Vamos também adotar a boa prática de exibir uma lista não ordenada através da tag `ul`, onde cada imagem será o conteúdo de uma `li`:

```
<!-- alurapic/src/App.vue -->

<template>
  <div>
    <h1>{{ titulo }}</h1>

    <ul>
      <li>
        
      </li>

      <li>
        
      </li>
    </ul>
  </div>
</template>

<script>
export default {
  data() {
    return {
      titulo: 'Alurapic',
      foto1: {
        url: 'https://encrypted-tbn2.gstatic.com/images?q=tbn:ANd9GcTwV4kVzT5McBdGSgqlVeRzubrNH_',
        titulo: 'Cachorro'
      },
      foto2: {
        url: 'https://encrypted-tbn2.gstatic.com/images?q=tbn:ANd9GcTwV4kVzT5McBdGSgqlVeRzubrNH_',
        titulo: 'Cachorro'
      },
    }
  }
}</script>
```

```
        }
    }
}

</script>

<style>
</style>
```

Excelente, temos duas imagens sendo exibidas. Mas se tivéssemos trinta? Teríamos as propriedades `foto3`, `foto4` e assim por diante, sem falar que teríamos que adicionar em `App` uma tag `img` para cada foto. Isso não precisa ser assim. Para revolucionarmos nosso código, primeiro vamos seguir a boa prática de disponibilizar os dados da foto em um `array`. Com isso, deixaremos de ter várias propriedades, uma para cada foto no objeto retornado pela função `data`:

```
<!-- alurapic/src/App.vue -->

<template>

<div>
  <h1>{{ titulo }}</h1>

  <ul>

    <li>
      
    </li>

    <li>
      
    </li>

  </ul>
</div>

</template>

<script>

// agora temos apenas a propriedade `fotos` que é um array que possui dois objetos que possuem :
export default {

  data() {
    return {
      titulo: 'Alurapic',
      fotos: [
        {
          url: 'https://encrypted-tbn2.gstatic.com/images?q=tbn:ANd9GcTwV4kVzT5McBdGSgqlVeRzubrl',
          titulo: 'Cachorro'
        },
        {
          url: 'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTOhmlmzV4-Sifx5BIC2SXeA-1',
          titulo: 'Gato'
        }
      ]
    }
  }
}
```

```

        }
    }
}

</script>

<style>
</style>

```

Excelente, e agora?

## A diretiva v-for

Mudamos apenas a fonte de dados, mas e o nosso template, como ficará? A ideia é a seguinte, para cada item do nosso array `fotos` queremos construir uma tag `li` com a imagem da foto que está sendo iterada no momento. Queremos algo como um `forEach` em JavaScript. Para isso, o próprio Vue nos disponibiliza a diretiva `v-for` que permite realizar laço dentro do nosso template, vejamos seu uso:

```

<!-- alurapic/src/App.vue -->

<template>

<div>
    <h1>{{ titulo }}</h1>

    <ul>

        <li v-for="foto of fotos">
            
        </li>

    </ul>

</div>
</template>

<script>
export default {

    // código omitido
}

</script>

<style>
</style>

```

Incrível! Para cada item do nosso array `fotos` uma tag `li` é criada. Na diretiva `v-for` indicamos qual lista estamos varrendo, em nosso caso, `fotos`, que faz parte do objeto retornado pela função `data`.

Mas como ter acesso a cada foto da lista? Precisamos ter acesso ao elemento para podermos passar os dados para a tag `img`. Para isso, damos um apelido para cada item da lista, em nosso caso, escolhemos o apelido `foto`. Como temos duas

fotos em nossa lista, a diretiva `v-for` será repetida duas vezes e através de interpolação com `{{ }}` podemos ter acesso aos dados da foto, por exemplo, com `{{ foto.titulo }}` acessamos seu título.

Agora, imagine se no lugar de termos dados fixos esses dados viessem de um servidor, de um API na web? Exibiríamos instantaneamente todas as fotos de uma só vez com pouquíssimo esforço. Aliás, esse será o assunto do próximo capítulo.