



## Deixando tudo mais alegre com efeitos

### Transcrição

Vue não faz mágicas no que diz respeito à animações. É o desenvolvedor que deve criar os estilos que devem ser aplicados no momento correto para que os efeitos aconteçam. O que o Vue faz é indicar a hora exata na qual nossas classes devem ser aplicadas.

No entanto, o Vue precisa delimitar a área do template no qual fará essas indicações. Como queremos realizar uma transição no conteúdo do nosso `Painel`, vamos envolver nossa `div` com a classe `panel-conteudo` pelo componente `transition`. Obrigatoriamente precisamos escolher um nome para ele. No caso, usaremos `panel-fade`:

```
<!-- alurapic/src/components/shared/panel/Painel.vue -->
```

```
<template>
```

```
<div class="panel">
```

```
<h2 class="panel-titulo" @dblclick="visivel = !visivel">
  {{ titulo }}</h2>
```

```
<transition name="panel-fade">
```

```
<div class="panel-conteudo" v-show="visivel">
```

```
<slot></slot>
```

```
</div>
```

```
</transition>
```

```
</div>

</template>

<script>
// código omitido
</script>

<style>
/* código omitido */
</style>
```

[COPIAR CÓDIGO](#)

Em primeiro lugar, `transition` só pode ser usado com um elemento, se tentarmos colocar mais um elemento como filho do nosso wrapper receberemos um erro.

Se clicarmos em nossa `Painel`, nada ainda acontece. Precisamos criar o estilo para `panel-fade`. O problema é que para realizarmos uma transição, precisamos aplicar estilos diferentes, por exemplo, quando o elemento esta sendo exibido ou quando ele esta sendo escondido.

É por isso que o componente `transition` adiciona dinamicamente nos seus elementos filhos as seguintes classes dinamicamente:

```
panel-fade-enter // antes do elemento ser incluído ou
removido, o estado atual
panel-fade-enter-active // quando o elemento esta sendo
incluído
panel-fade-leave-active // quando o elemento esta sendo
removido
```

[COPIAR CÓDIGO](#)

Sendo assim, podemos fazer o seguinte:

```
<!-- alurapic/src/componets/shared/painel/Painel.vue -->

<template>

  <div class="painel">

    <h2 class="painel-titulo" @dblclick="visivel = !visivel">
      {{ titulo }}</h2>
    <transition name="painel-fade">
      <div class="painel-conteudo" v-show="visivel">
        <slot></slot>
      </div>
    </transition>

  </div>

</template>

<script>
  // código omitido
</script>

<style>
  /* código anterior omitido */

  .painel-fade-enter, .painel-fade-leave-active {
    opacity: 0
  }

</style>
```

[COPIAR CÓDIGO](#)

Por fim, usamos os outros dois estilos para aplicar a transição:

```
<!-- alurapic/src/componets/shared/painel/Painel.vue -->

<template>

  <div class="painel">

    <h2 class="painel-titulo" @dblclick="visivel = !visivel">
      {{ titulo }}</h2>
    <transition name="painel-fade">
      <div class="painel-conteudo" v-show="visivel">
        <slot></slot>
      </div>
    </transition>

  </div>

</template>

<script>
  // código omitido
</script>

<style>
  /* código anterior omitido */

  .painel-fade-enter, .painel-fade-leave-active {
    opacity: 0
  }

  .painel-fade-enter-active, .painel-fade-leave-active {
    transition: opacity .4s
  }
</style>
```

[COPIAR CÓDIGO](#)

Façamos um teste agora. Quando ocultamos e exibimos no elemento um fade é realizado.