

08

Estilos e escopo

Transcrição

Criamos nosso primeiro componente que encapsula marcação HTML, dados e seu estilo. Mas que tal vermos como ficará cada item dentro do nosso painel com sombra? É algo que o design não pediu, mas que podemos fazer facilmente usando o seletor universal do CSS.

Alterando Painel.vue :

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

<template>

  <div class="painel">

    <h2 class="painel-titulo">{{ titulo }}</h2>
    <slot class="painel-conteudo">
      </slot>
    </div>

</template>

<script>

export default {
  props: ['titulo']
}
</script>

<style>
.painel {
  padding: 0 auto;
  border: solid 2px grey;
  display: inline-block;
  margin: 5px;
  box-shadow: 5px 5px 10px grey;
  width: 200px;
  height: 100%;
  vertical-align: top;
  text-align: center;
}

.painel .painel-titulo {
  text-align: center;
  border: solid 2px;
  background: lightblue;
  margin: 0 0 15px 0;
  padding: 10px;
  text-transform: uppercase;
}

/* colocando todos os elementos do painel com box shadow */

```

```
* {
  box-shadow: 5px 5px 5px;
}

</style>
```

Quando estilos globais dão problema

Temos um problema. Não só os elementos do painel ficaram com sombra, mas o painel em si, inclusive o título da de App.vue que não tem nada a haver com a história. O problema é que aplicamos um seletor que afetará todos conteúdo da página, não apenas o componente. Dizemos que o estilo do componente Painel vazou do componente para o mundo externo. Podemos resolver isso facilmente especificando nosso seletor:

```
.painel * {
  box-shadow: 5px 5px 5px;
}
```

Estilos com escopo de componente

Mas se tivermos algum outro elemento da nossa página que faça uso da classe painel? Corremos o risco do estilo do nosso componente ser aplicado indevidamente. Se quisermos que os estilos definidos em styles do nosso componente afete apenas o componente, basta adicionarmos a propriedade scoped na tag style. Vamos adicionar esse atributo, inclusive utilizar o seletor universal que usamos anteriormente:

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

<template>

  <div class="painel">

    <h2 class="painel-titulo">{{ titulo }}</h2>
    <slot class="painel-conteudo">
    </slot>
  </div>

</template>

<script>

export default {
  props: ['titulo']
}
</script>

<style scoped>
.painel {
  padding: 0 auto;
  border: solid 2px grey;
  display: inline-block;
  margin: 5px;
  box-shadow: 5px 5px 10px grey;
}
```

```
width: 200px;
height: 100%;
vertical-align: top;
text-align: center;
}

.painel .painel-titulo {
  text-align: center;
  border: solid 2px;
  background: lightblue;
  margin: 0 0 15px 0;
  padding: 10px;
  text-transform: uppercase;
}

* {
  box-shadow: 5px 5px 5px;
}
```

</style>

É claro que não usaremos esse estilo tosco que acabamos de criar, mas veja que as sombras agora afetam apenas o nosso componente. É uma boa prática usar um escopo de componente quando criamos nossos componentes reutilizáveis, evitando assim que os estilos de um componente interfiram no estilos de outro.

Ah, e não esqueça de remover o último seletor, ok?