



## Nosso botão de 1001 utilidades!

### Transcrição

*Começando deste ponto? Você pode fazer o [DOWNLOAD](https://s3.amazonaws.com/caelum-online-public/vue/stages/07-alurapic.zip) (<https://s3.amazonaws.com/caelum-online-public/vue/stages/07-alurapic.zip>) completo do projeto do capítulo anterior e continuar seus estudos a partir deste capítulo. Será necessário abrir seu terminal, entrar dentro da pasta `alurapic` e executar o comando `npm install` para baixar novamente todas as dependências da aplicação.*

Como já configuramos as rotas da aplicação, nada nos impede de partirmos para nossa página de cadastro. No entanto, podemos expandir ainda mais nosso conhecimento sobre criação de componentes.

Em breve, precisaremos implementar a funcionalidade de exclusão de fotos. Vamos deixar preparado o terreno adicionando um botão com o título `remove`, mas espere um pouco. Quais são as características desse botão?

O botão deve ter a cor vermelha e letra branca, porque remoção é algo sério e o vermelho chama atenção. Ele deve ter um rótulo, ou seja, um nome. Além disso, precisamos indicar seu tipo, se vai ser `button` ou `submit` por exemplo. Por fim, queremos que esse botão exiba uma janela de confirmação da operação. Agora eu te pergunto, e se quisermos o mesmo botão na cor azul e sem confirmação? Teremos que criar outro botão.

Você já deve ter percebido que o botão é um forte candidato para tornarmos um componente. Ele será bastante configurável e será usado não apenas em `Home`, mas em `Cadastro` também em breve.

Vamos criar o componente `alurapic/src/shared/botao/Botao.vue`:

```
<!-- alurapic/src/components/shared/botao/Botao.vue -->
<template>
  <button class="botao botao-perigo" :type="tipo">{{rotulo}}
</button>
</template>
<script>
export default {
  props: ['tipo', 'rotulo'],
}
</script>

<style scoped>
  .botao {
    display: inline-block;
    padding: 10px;
    border-radius: 3px;
    margin: 10px;
    font-size: 1.2em;
  }

  .botao-perigo {
    background: firebrick;
    color: white;
  }

  .botao-padrao {
    background: darkcyan;
```

```
    color: white;
  }

</style>
```

[COPIAR CÓDIGO](#)

Vamos por partes. Primeiro, vamos fazer com que o botão receba apenas como parâmetro do componente pai o seu tipo e rótulo. Por enquanto, vamos deixar fixo a classe `botao-perigo`.

Agora que já temos o componente criado, vamos importá-lo em `Home` e utilizá-lo dentro do painel, logo abaixo do nosso componente de imagem responsiva:

```
<!-- alurapic/src/components/home/Home.vue -->

<template>
  <div>
    <h1 class="titulo">Alurapic</h1>

    <input type="search" class="filtro" @input="filtro =
    $event.target.value" placeholder="filtre pelo título da foto">

    <ul class="lista-fotos">

      <li class="lista-fotos-item" v-for="foto of
      fotosComFiltro">

        <meu-painel :titulo="foto.titulo">

          <imagem-responsiva :url="foto.url"
          :titulo="foto.titulo"/>

          <meu-botao rotulo="remover" tipo="button"/>
      </li>
    </ul>
  </div>
</template>
```

```
        </meu-painel>

    </li>

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

<script>

import Paine1 from '../shared/painel/Paine1.vue';
import ImagemResponsiva from '../shared/imagem-
responsiva/ImagemResponsiva.vue'

// Fazendo o import do botão. Não esqueça de adicioná-lo em
components

import Botao from '../shared/botao/Botao.vue';

export default {

  components: {

    'meu-painel': Paine1,
    'imagem-responsiva': ImagemResponsiva,
    'meu-botao': Botao
  },

  // código omitido
}
</script>
<style>

/* código omitido */
</style>
```

[COPIAR CÓDIGO](#)

Isso já é suficiente para que nosso botão seja exibido.