

## Que tal mais um componente?

### Transcrição

O componente `Painel` foi o primeiro componente reutilizável que criamos. No entanto, por mais ínfima que nossa tag `img` possa ser, teremos que lembrar sempre de adicionar `width: 100%` para que ela se adapte a diversos tamanhos. Para evitar que esqueçamos desse detalhe, que tal criarmos o componente `ImagemResponsiva` que encapsulará a tag `img` com seu estilo?

Vamos criar o arquivo `alurapic/src/components/shared/imagem-responsiva/ImagemResponsiva.vue` :

```
<!-- `alurapic/src/components/shared/imagem-responsiva/ImagemResponsiva.vue` -->

<template>
  
</template>

<script>

export default {
  props: ['url', 'titulo']
}

</script>

<style scoped>
  .imagem-responsiva {
    width: 100%;
  }
</style>
```

Agora, vamos importar nosso componente em `App.vue` e utilizá-lo no lugar da tag `img`. Aliás, não esqueça de remover o estilo de `App` que movemos para dentro do nosso componente:

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

<template>
  <div class="corpo">

    <h1 class="centralizado">{{ titulo }}</h1>

    <input type="search" class="filtro" @input="filtro = $event.target.value" placeholder="filtro">

    <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-painel>
      </li>
    </ul>

  </div>
</template>
```

```
</div>
</template>

<script>

import Paine1 from './components/shared/paine1/Paine1.vue';
import ImagemResponsiva from './components/shared/imagem-responsiva/ImagemResponsiva.vue'

export default {

  components: {

    'meu-paine1': Paine1,
    'imagem-responsiva': ImagemResponsiva
  },

  data () {
    return {
      titulo: 'Alurapic',

      fotos: [],

      filtro: ''
    }
  },

  computed: {
    fotosComFiltro() {
      if (this.filtro) {
        let exp = new RegExp(this.filtro.trim(), 'i');
        return this.fotos.filter(foto => exp.test(foto.titulo));
      } else {
        return this.fotos;
      }
    }
  },

  created() {

    this.$http
      .get('http://localhost:3000/v1/fotos')
      .then(res => res.json())
      .then(fotos => this.fotos = fotos);
  }
}
</script>
<style>

.centralizado {
  text-align: center;
}

.corpo {
  font-family: Helvetica, sans-serif;
  margin: 0 auto;
  width: 96%;
}
```

```
.lista-fotos {  
  list-style: none;  
}  
  
.lista-fotos .lista-fotos-item {  
  display: inline-block;  
}  
  
.filtro {  
  display: block;  
  width: 100%;  
}  
</style>
```

É claro que uma imagem responsiva pode ter outras propriedades, mas a boa notícia é que em todo lugar que nosso componente for utilizado, quando mudarmos sua definição, a alteração será aplicada em todos os lugares.