

## Ajustando a aplicação para SPA

### Transcrição

Agora que temos uma visão geral do que é uma SPA, precisamos realizar um ajuste em nossa aplicação antes de continuar. Hoje, temos o componente `App`, o primeiro a ser carregado pela nossa aplicação e que exibe nossos painéis de fotos que podem ser filtrados. No entanto, em uma SPA, o primeiro componente visual funciona como um placeholder, ou melhor, local no qual outras páginas serão inseridas através do framework. Lembre-se que páginas também são componentes, dessa forma, será em `App` que de acordo com um endereço especial acessado pelo usuário, o Vue saberá qual componente enxertar dentro de `App` para exibi-lo. Entenda-o como uma grande lacuna que hora exibirá um componente, hora exibirá outro. Sendo assim, precisamos extrair todo o código que lista nossas fotos para um novo componente. Vamos chamá-lo de `Home`.

Para acelerar o processo, copie o conteúdo do arquivo `alurapic/src/App.vue` para dentro do arquivo `alurapic/src/components/home/Home.vue`. Com o conteúdo copiado, faremos poucos ajustes. O primeiro, é que a `div` do template não terá a classe `corpo` e o título da página será fixo como `<h1>Alurapic</h1>`. Como o título é fixo, não precisamos da propriedade `titulo` em nossa função `data`, inclusive podemos remover a classe `corpo` da tag `<style>`. Nosso arquivo fica assim:

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

<template>
  <div>

    <h1 class="centralizado">Alurapic</h1>

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

    <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>

<script>

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

export default {

  components: {

    'meu-painel': Painel,
    'imagem-responsiva': ImagemResponsiva
  },

  data () {
```

```

    return {

      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, err => console.log(err));
  }
}
</script>
<style>

  .centralizado {
    text-align: center;
  }

  .lista-fotos {
    list-style: none;
  }

  .lista-fotos .lista-fotos-item {
    display: inline-block;
  }

  .filtro {
    display: block;
    width: 100%;
  }
</style>

```

Veja também que o caminho dos imports precisou mudar para `../shared` para descer um nível, já que o componente agora esta em sua pasta específica. Excelente, agora temos `Home` e `Cadastro` como componentes que correspondem a páginas em nossa aplicação.

Agora, vamos ajustar `App.vue`. Não haverá um código sequer dentro de `<script>` e o único estilo dentro de `<style>` será a classe `corpo`. Seu template terá, por enquanto, apenas a tag `<div class="corpo"></div>`:

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

<template>
  <div class="corpo">
    <!-- é aqui que encontrará os componentes de página que queremos visualizar -->
  </div>
</template>

<script>
</script>

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

Você não deve ter ficado surpreso em ver no browser a página em branco, isso porque como `App` é o primeiro componente a ser carregado, seu `template` está vazio. Chegou a hora de realizarmos as configurações necessárias para que seja possível hora exibir o conteúdo de `Home` e hora o conteúdo de `Cadastro`.