



01

A rota de alteração

Transcrição

Começando deste ponto? Você pode fazer o [DOWNLOAD](https://s3.amazonaws.com/caelum-online-public/vue/stages/11-alurapic.zip) (<https://s3.amazonaws.com/caelum-online-public/vue/stages/11-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.

Precisamos implementar a funcionalidade de alteração de fotos. Para isso, em `Home` vamos adicionar um novo botão ao lado do botão `Remover` :

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

<template>
  <div>
    <h1 class="centralizado">Alurapic</h1>
    <p v-show="mensagem" class="centralizado">{{
mensagem }}</p>
    <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" v-meu-transform:scale.animate="1.2"/>

```

```
    <!-- novo botão! -->

    <router-link :to="{ name: 'cadastro' }">
      <meu-botao
        rotulo="Alterar"
        tipo="button"/>
    </router-link>

    <meu-botao
      rotulo="remover"
      tipo="button"
      estilo="perigo"
      :confirmacao="true"
      @botaoAtivado="remove(foto)"/>
  </meu-painel>
</li>
</ul>
</div>
</template>
<!-- código posterior omitido -->
```

[COPIAR CÓDIGO](#)

Veja que ele usa a rota nomeada `cadastro`. Quando clicamos no botão, somos direcionados para `Cadastro`. No entanto, precisamos enviar o ID da foto que estamos alterado para `Cadastro` para que seja possível buscamos a foto da nossa API exibindo-a no formulário.

Para que isso seja possível, vamos alterar nosso arquivo `routes.js` e adicionar um curinga na rota:

```
import Home from './components/home/Home.vue';
import Cadastro from './components/cadastro/Cadastro.vue';

export const routes = [
```

```

    { path: '', name: 'home', component: Home, titulo:
'Home', menu: true },
    { path: '/cadastro/:id', name: 'cadastro', component:
Cadastro, titulo: 'Cadastro', menu: true },
    { path: '*', component: Home, menu: false }
];

```

[COPIAR CÓDIGO](#)

Agora, toda vez que endereços como `cadastro/1` ou `cadastro/22` forem acessados, o Vue Router saberá que é essa a rota que deve ser seguida.

Agora, no template de `Home`, podemos realizar uma navegação para `cadastro` passando como parâmetro o ID da foto. Isso acarreta uma mudança em nosso `router-link`:

```

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

<template>
  <div>
    <h1 class="centralizado">Alurapic</h1>
    <p v-show="mensagem" class="centralizado">{{
mensagem }}</p>
    <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" v-meu-transform:scale.animate="1.2"/>

          <router-link :to="{ name: 'cadastro',
params: { id : foto._id }}">
            <meu-botao

```

```
        rotulo="Alterar"
        tipo="button"/>
    </router-link>

    <meu-botao
        rotulo="remover"
        tipo="button"
        estilo="perigo"
        :confirmacao="true"
        @botaoAtivado="remove(foto)"/>
    </meu-painel>
</li>
</ul>
</div>
</template>
<!-- código posterior omitido -->
```

[COPIAR CÓDIGO](#)

Reparem que além do name, o objeto passado para a propriedade `to` agora possui a propriedade `params`. Essa propriedade é outro objeto JavaScript cuja a propriedade `tem` que ser exatamente o nome do curinga que usamos na rota. O valor da propriedade é o ID da foto.

Quando clicamos no botão, somos direcionados para `/cadastro/IdDaFotoAqui`. Excelente, mas temos um problema. Se clicamos no menu `Home` e depois no menu `Cadastro` continuamos em `Home`. Isso acontece porque `/cadastro` deixou de ser uma rota válida, rotas válidas agora são aquelas que possuem `/cadastro/` seguido de qualquer valor. E agora?

Para resolvermos, vamos voltar a rota para o jeito anterior e adicionar mais uma rota, essa sim parametrizada e que não faz parte do menu:

```
// alurapic/src/routes.js
```

```
import Home from './components/home/Home.vue';
```

```
import Cadastro from './components/cadastro/Cadastro.vue';

export const routes = [

  { path: '', name: 'home', component: Home, titulo:
'Home', menu: true },
  { path: '/cadastro', name: 'cadastro', component:
Cadastro, titulo: 'Cadastro', menu: true },
  { path: '/cadastro/:id', name: 'altera', component:
Cadastro, titulo: 'Cadastro', menu: false },
  { path: '*', component: Home, menu: false }

];
```

[COPIAR CÓDIGO](#)

A nova rota tem o nome `altera`, não podemos nos esquecer de alterar esse nome no `router-link` de `Home`:

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

<template>
  <div>
    <h1 class="centralizado">Alurapic</h1>
    <p v-show="mensagem" class="centralizado">{{
mensagem }}</p>
    <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" v-meu-transform:scale.animate="1.2"/>

          <router-link :to="{ name: 'altera', params:
{ id : foto._id }}">
```

```
        <meu-botao
          rotulo="Alterar"
          tipo="button"/>
      </router-link>

      <meu-botao
        rotulo="remover"
        tipo="button"
        estilo="perigo"
        :confirmacao="true"
        @botaoAtivado="remove(foto)"/>
    </meu-painel>
  </li>
</ul>
</div>
</template>
<!-- código posterior omitido -->
```

[COPIAR CÓDIGO](#)

Agora, conseguimos acessar `Cadastro` diretamente ou passando parâmetro para ela. Precisamos agora é descobrir como conseguimos extrair o ID da foto passado para `Cadastro` para que possamos então buscar a foto em nossa API.