



## Animando transições de página

### Transcrição

Uma das coisas mais bonitas do mundo da programação é podermos aplicar com conhecimento já adquirido e aplicá-lo em outros contextos. No capítulo anterior aprendemos a trabalhar com transições. Se refletirmos por um instante, a saída de uma página para a outra é uma mudança de estado, logo, passível de animação.

Vamos aplicar a mesma transição que fizemos com o componente `Painel` com a única diferença que será executada mais rapidamente. Aprendemos que precisamos envolver o componente desejado como alvo da transição através da tag `transition`. Sendo assim, qual componente você acha que devemos envolver para conseguir realizar transições entre páginas? O componente `router-view` em `App` !.

Alterando `App` :

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

<template>
  <div class="corpo">

    <nav>
      <ul>
        <li v-for="route in routes">
          <router-link :to="route.path" ? route.path : '/'>
```

```
        {{route.titulo}}
      </router-link>
    </li>
  </ul>
</nav>

<transition name="pagina">
  <router-view></router-view>
</transition>

</div>
</template>
<script>

import { routes } from './routes';

export default {

  data() {

    return {

      routes

    }

  }
}
</script>
<style>

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

```
.pagina-enter-active, .pagina-leave-active {  
  transition: opacity .3s  
}  
.pagina-enter, .pagina-leave-active {  
  opacity: 0  
}  
</style>
```

[COPIAR CÓDIGO](#)

Veja que usamos os mesmos estilos, com a diferença que usamos como name do componente `transition` o valor `pagina` e a transição foi reduzida para 200 milissegundos. Experimente clicar nos itens do menu da aplicação e veja a transição entre páginas sendo realizada. Se você domina CSS pode aplicar transições mais elaboradas, o mais importante é saber como integrá-las ao Vue.