

Curso Técnico Superior Profissional em Tecnologias e Programação de Sistemas de Informação

Unidade Curricular: Desenvolvimento Web – Front-End

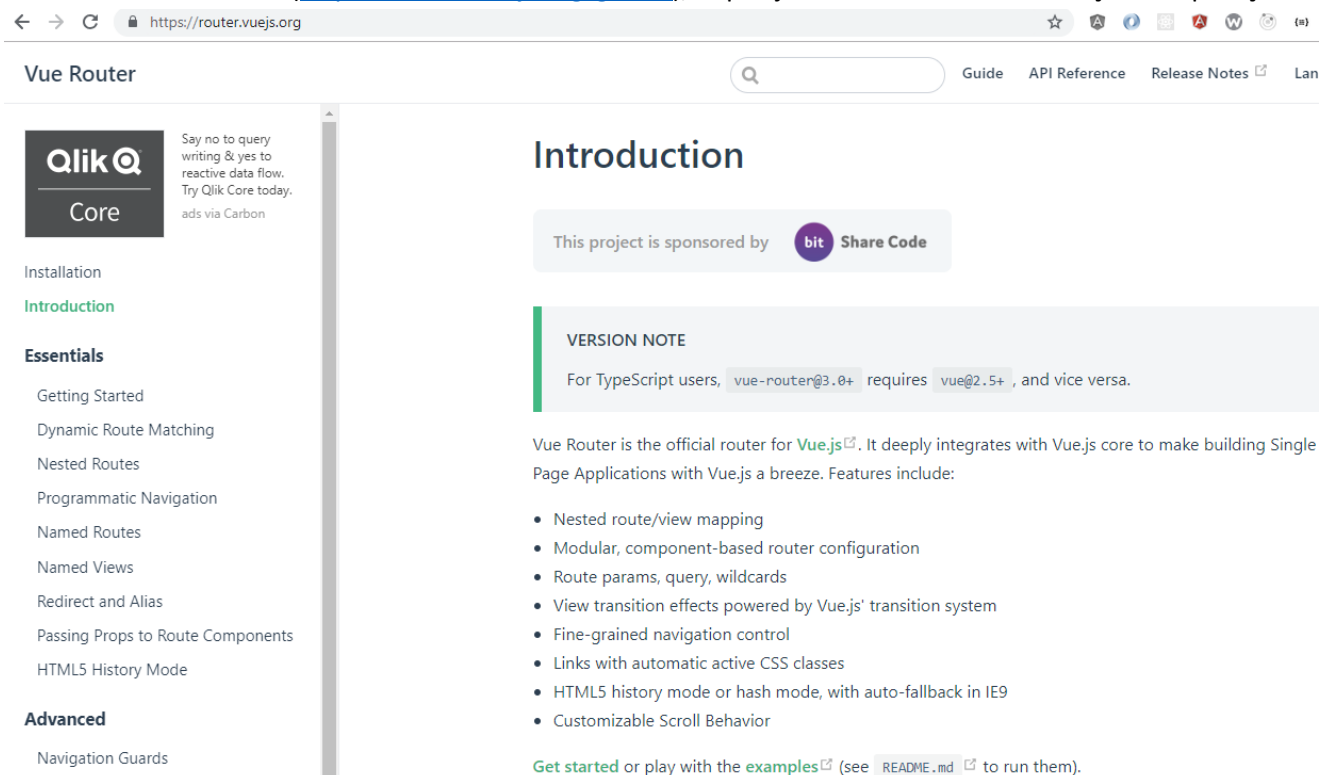
1º Ano/2º Semestre

Docente: Marco Miguel Olival Olim

Data 26/04/2019

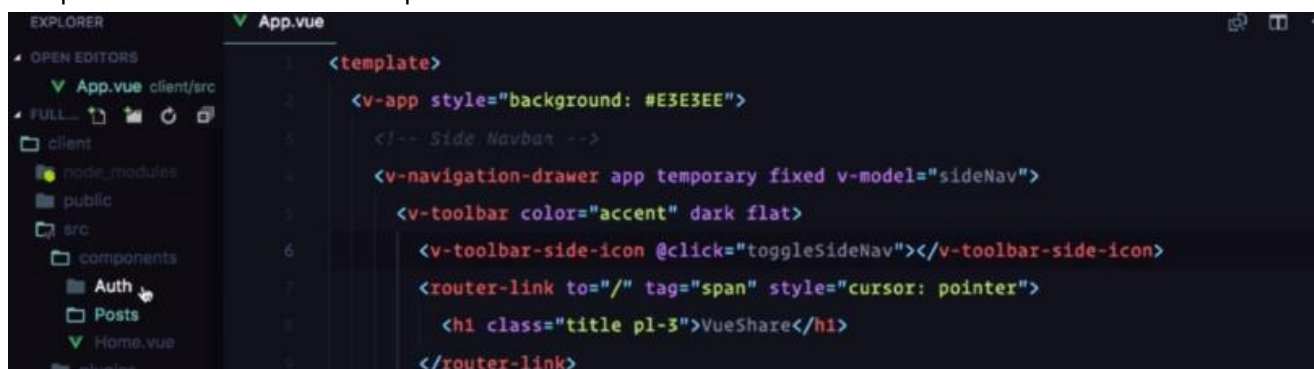
ESTE EXERCÍCIO ABORDA A UTILIZAÇÃO DE ROTAS NO VUE CLI

Ao criarmos a barra de navegação inserimos alguns botões que neste momento ainda não têm funcionalidade. Atendendo a que a nossa aplicação em VueJS é composta por componentes, apenas temos de mapear esses componentes aos endereços desejados no browser. Para operacionalizar essa funcionalidade utilizamos o Vue router (<https://router.vuejs.org/guide/>), o qual já foi sendo instalado na criação da aplicação.



The screenshot shows the official Vue Router website. The left sidebar contains a navigation menu with links to 'Installation', 'Introduction' (highlighted), 'Essentials' (with sub-links like 'Getting Started', 'Dynamic Route Matching', etc.), and 'Advanced' (with 'Navigation Guards'). The main content area is titled 'Introduction' and includes a 'VERSION NOTE' for TypeScript users, a list of features (Nested route/view mapping, Modular configuration, etc.), and a 'Get started' link.

Vamos agora criar os componentes para onde irão ser direcionadas as nossas rotas (em router.js). Em components criamos duas novas pastas: **Auth** e **Posts**

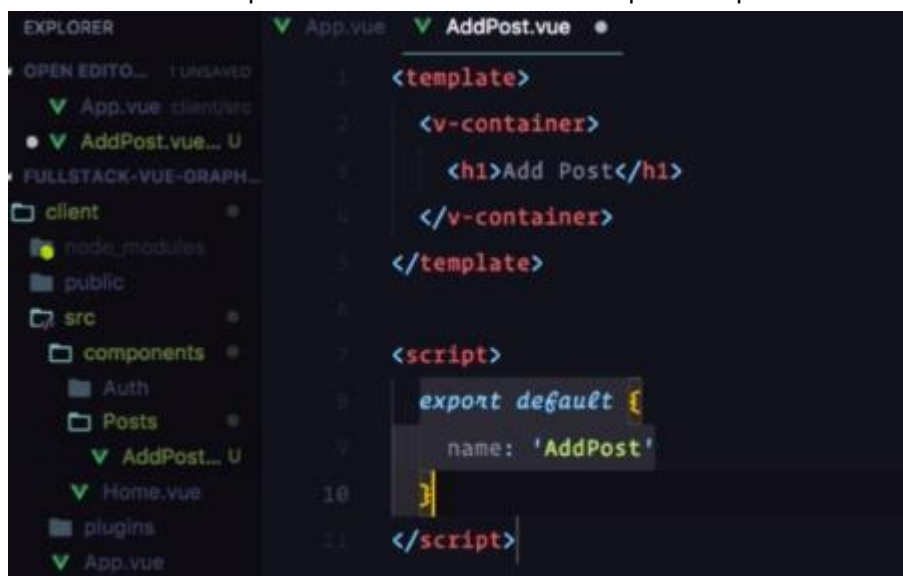


The screenshot shows a code editor with the 'App.vue' file open. The file structure on the left shows 'components' containing 'Auth' and 'Posts' folders. The main code area shows the following Vue.js template:

```
<template>
  <v-app style="background: #E3E3EE">
    <!-- Side Navbar -->
    <v-navigation-drawer app temporary fixed v-model="sideNav">
      <v-toolbar color="accent" dark flat>
        <v-toolbar-side-icon @click="toggleSideNav"></v-toolbar-side-icon>
        <router-link to="/" tag="span" style="cursor: pointer">
          <h1 class="title pl-3">VueShare</h1>
        </router-link>
```

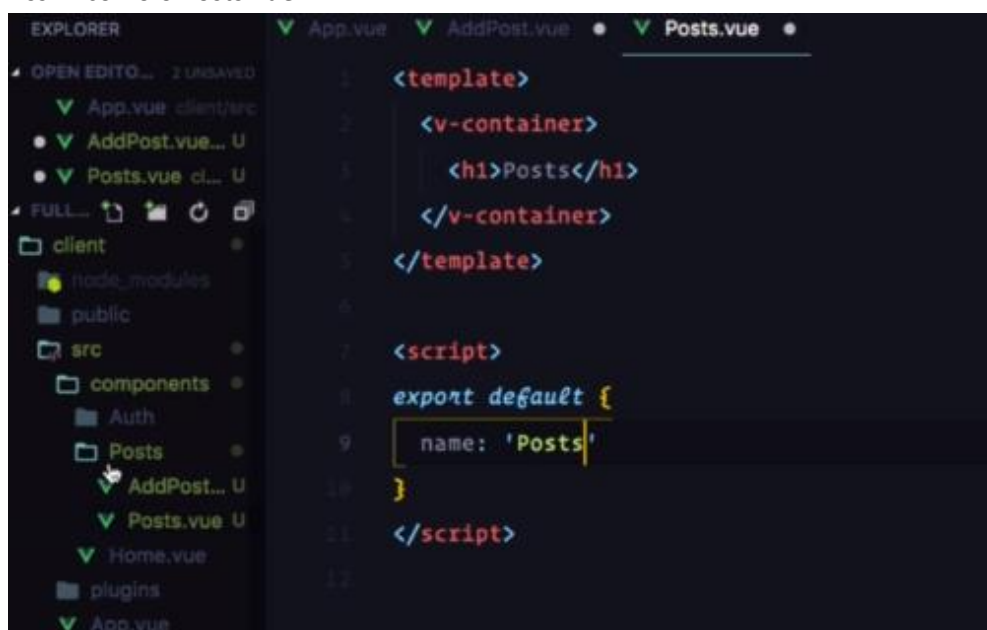
Cofinanciado por:

Inicialmente os componentes terão a mesma markup. O componente AddPost.vue terá o seguinte formato:



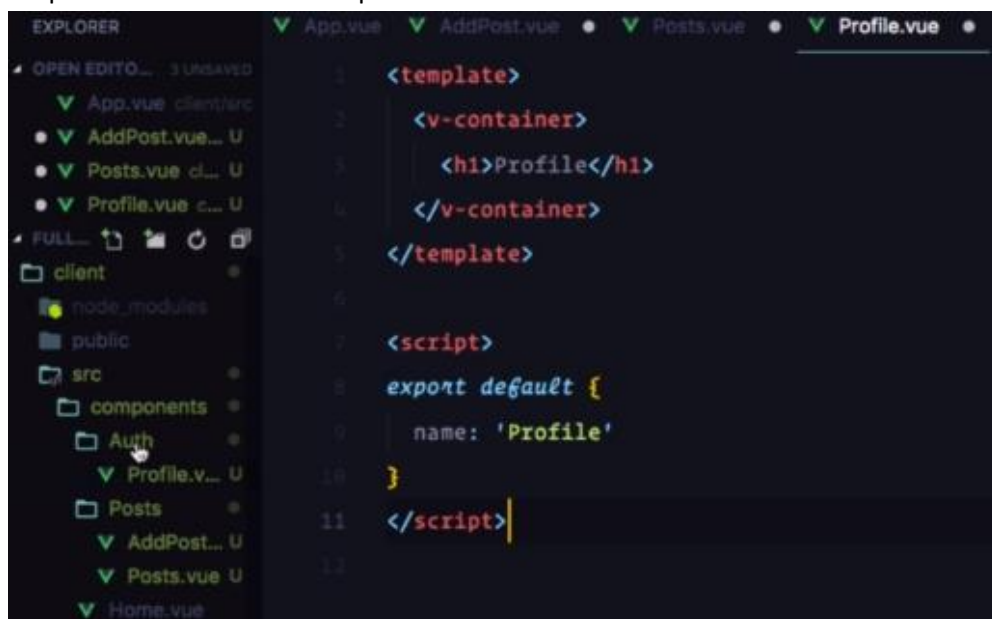
```
1 <template>
2   <v-container>
3     <h1>Add Post</h1>
4   </v-container>
5 </template>
6
7 <script>
8   export default {
9     name: 'AddPost'
10  }
11 </script>
```

Assim como o Posts.vue



```
1 <template>
2   <v-container>
3     <h1>Posts</h1>
4   </v-container>
5 </template>
6
7 <script>
8   export default {
9     name: 'Posts'
10  }
11 </script>
```

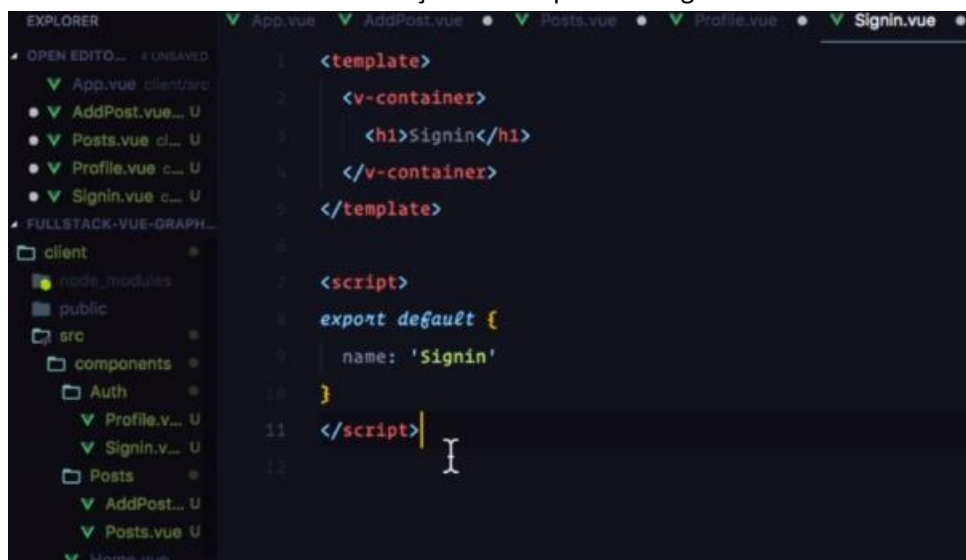
Na pasta Auth criamos o componente Profile.vue



```
1 <template>
2   <v-container>
3     <h1>Profile</h1>
4   </v-container>
5 </template>
6
7 <script>
8   export default {
9     name: 'Profile'
10  }
11 </script>
```

Cofinanciado por:

E continuamos no Auth com a criação do componente Signin.vue



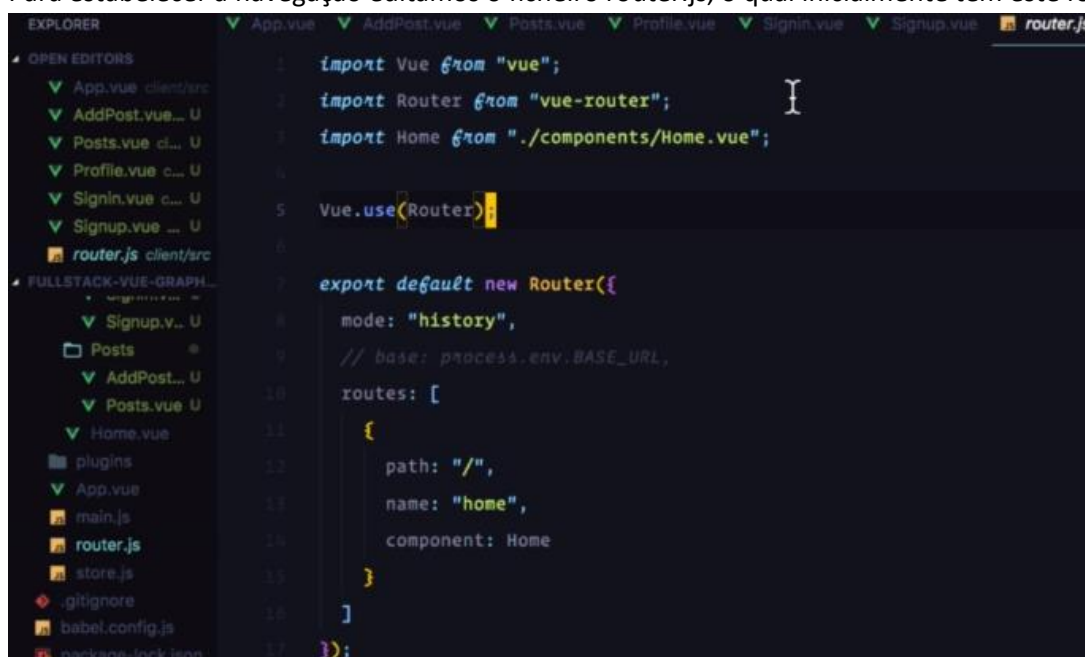
```
1 <template>
2   <v-container>
3     <h1>Signin</h1>
4   </v-container>
5 </template>
6
7 <script>
8   export default {
9     name: 'Signin'
10  }
11 </script>
```

E finalmente concluímos criação de componentes com o Signup.vue



```
1 <template>
2   <v-container>
3     <h1>Signup</h1>
4   </v-container>
5 </template>
6
7 <script>
8   export default {
9     name: 'Signup'
10  }
11 </script>
```

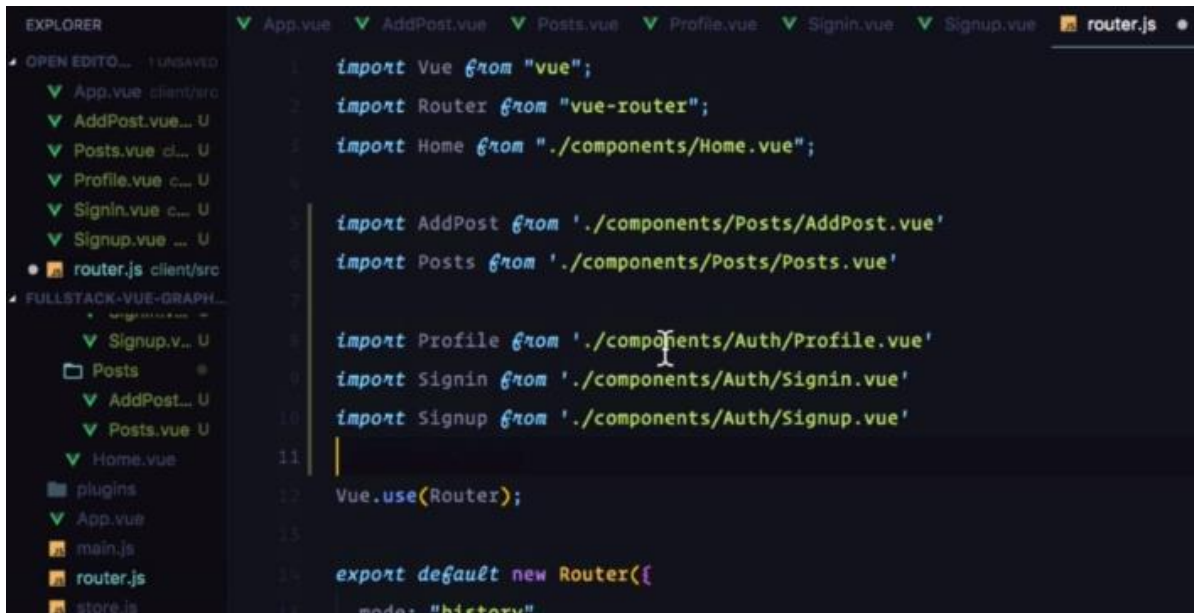
Para estabelecer a navegação editamos o ficheiro router.js, o qual inicialmente tem este formato



```
1 import Vue from "vue";
2 import Router from "vue-router";
3 import Home from "../components/Home.vue";
4
5 Vue.use(Router);
6
7 export default new Router({
8   mode: "history",
9   // base: process.env.BASE_URL,
10  routes: [
11    {
12      path: "/",
13      name: "home",
14      component: Home
15    }
16  ]
17 });
```

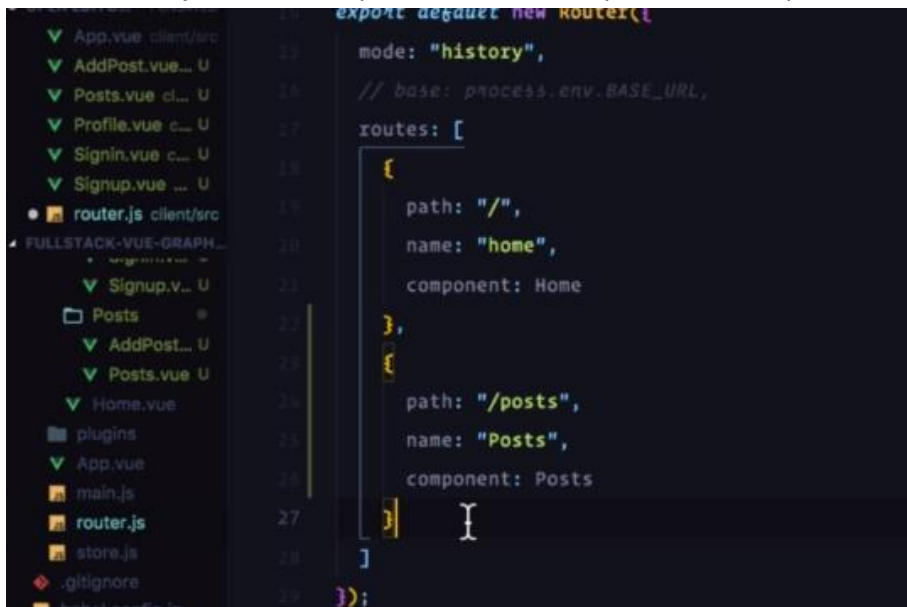
Cofinanciado por:

Caso não queiramos fazer *lazy loading* dos componentes temos de importá-los antes de definir a sua rota



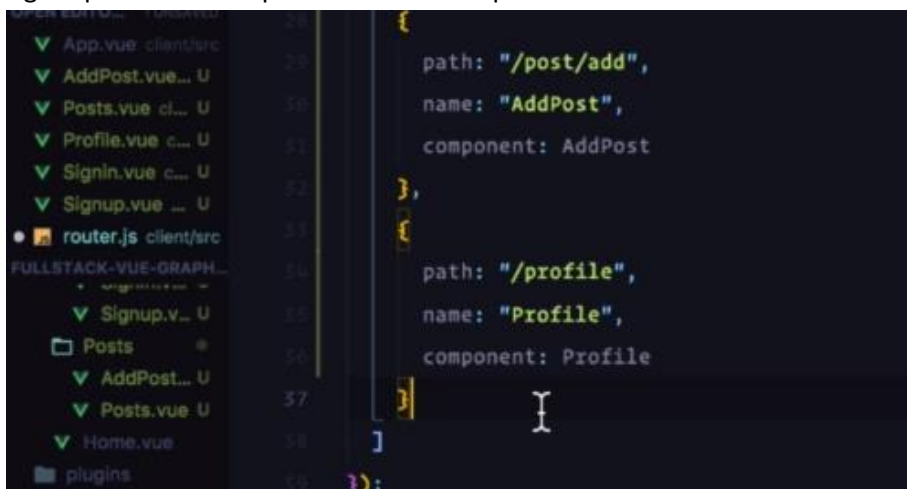
```
1 import Vue from "vue";
2 import Router from "vue-router";
3 import Home from "../components/Home.vue";
4
5 import AddPost from '../components/Posts/AddPost.vue'
6 import Posts from '../components/Posts/Posts.vue'
7
8 import Profile from '../components/Auth/Profile.vue'
9 import Signin from '../components/Auth/Signin.vue'
10 import Signup from '../components/Auth/Signup.vue'
11
12 Vue.use(Router);
13
14 export default new Router({
15   mode: "history"
```

A rota é um objeto JSON em que indicamos o componente correspondente e o caminho pretendido



```
14 export default new Router({
15   mode: "history",
16   // base: process.env.BASE_URL,
17   routes: [
18     {
19       path: "/",
20       name: "home",
21       component: Home
22     },
23     {
24       path: "/posts",
25       name: "Posts",
26       component: Posts
27     }
28   ]
29 });
```

Agora para adicionar posts e consultar o profile:



```
28 {
29   path: "/post/add",
30   name: "AddPost",
31   component: AddPost
32 },
33 {
34   path: "/profile",
35   name: "Profile",
36   component: Profile
37 }
38 ]
39 });
```

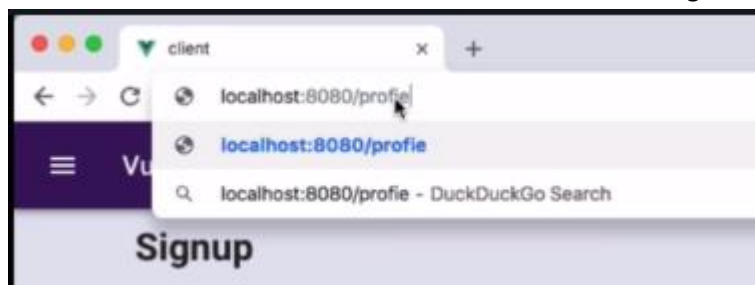
Cofinanciado por:

Por último definimos a rota para o signin e signup:

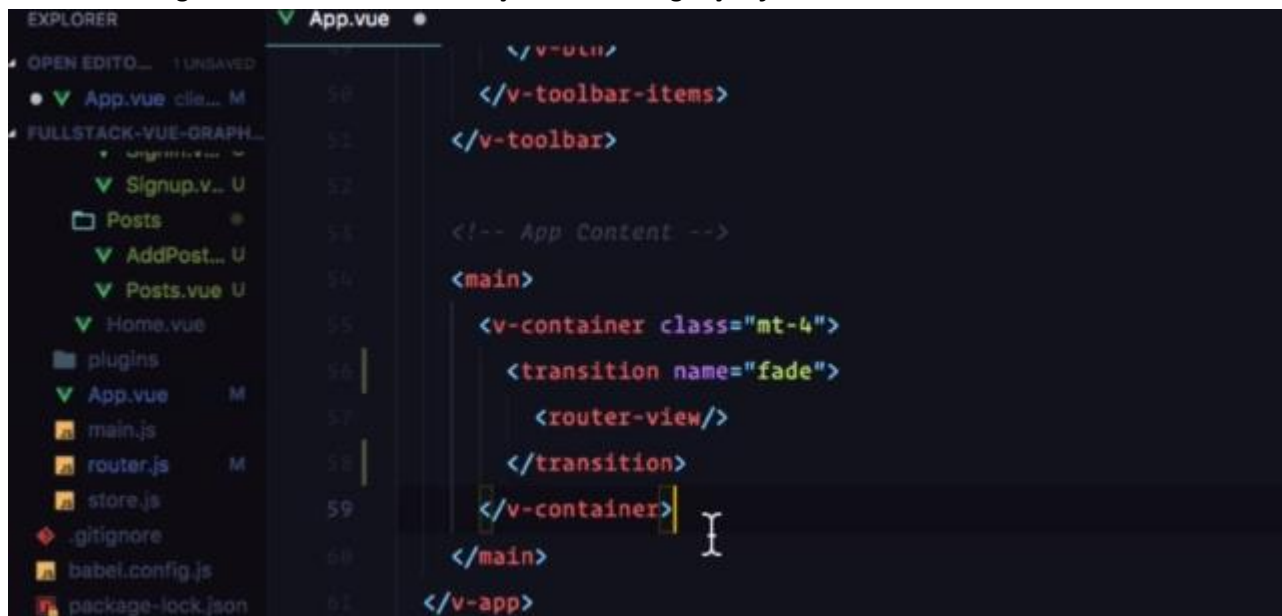


```
18      component: Profile
19    },
20    {
21      path: "/signin",
22      name: "Signin",
23      component: Signin
24    },
25    {
26      path: "/Signup",
27      name: "Signup",
28      component: Signup
29    }
30  ]
31};
```

Podemos testar estas rotas tanto com os botões como digitando o endereço no próprio browser



Tal como no NUXT, podemos definir animações para a transição entre páginas/rotas. Editamos o App.vue e inserimos a tag transition. A nossa animação terá a designação **fade**

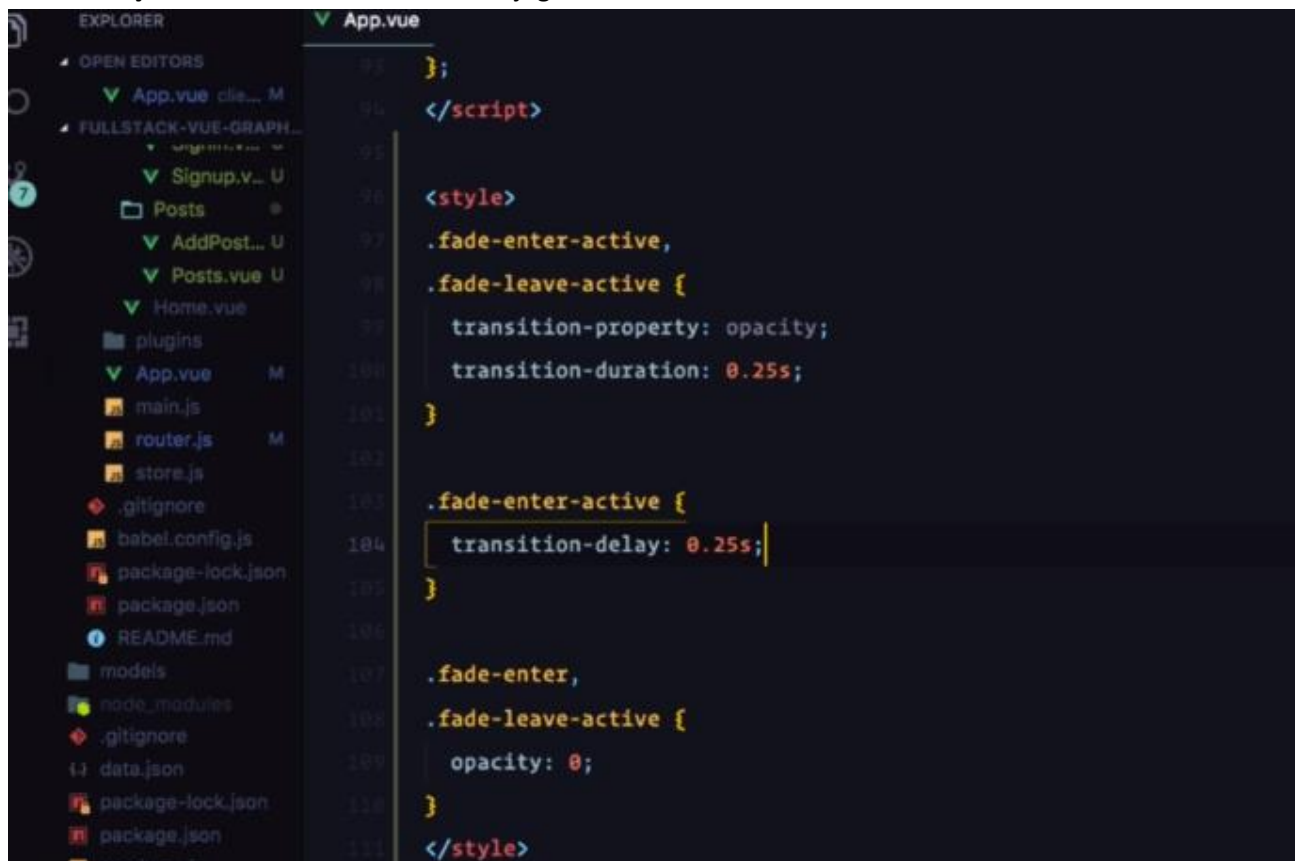


```
50      </v-toolbar-items>
51    </v-toolbar>
52
53    <!-- App Content -->
54    <main>
55      <v-container class="mt-4">
56        <transition name="fade">
57          <router-view/>
58        </transition>
59      </v-container>
60    </main>
61  </v-app>
```

Cofinanciado por:

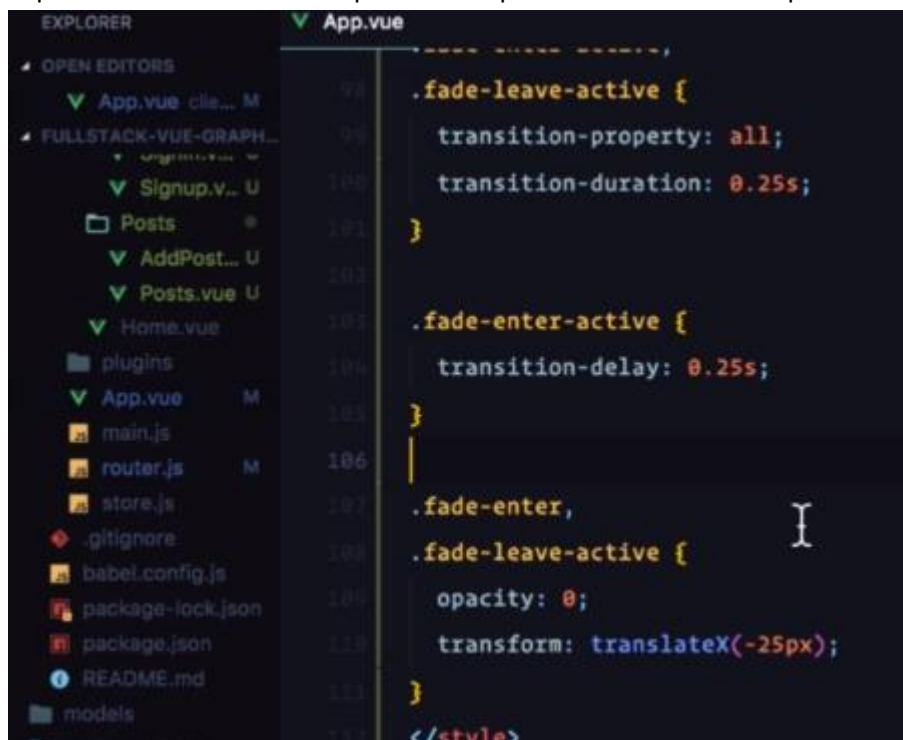


Esta transição é semelhante à utilizada no jogo das fichas anteriores de VueJS



```
App.vue
93 };
94 </script>
95
96 <style>
97   .fade-enter-active,
98   .fade-leave-active {
99     transition-property: opacity;
100     transition-duration: 0.25s;
101   }
102
103   .fade-enter-active {
104     transition-delay: 0.25s;
105   }
106
107   .fade-enter,
108   .fade-leave-active {
109     opacity: 0;
110   }
111 </style>
```

Estas animações de transição aqui propostas não pretendem ser uma receita definitiva. Encoraja-se que experimente com diferentes parâmetros para um resultado mais personalizado.



```
App.vue
98 .fade-leave-active {
99   transition-property: all;
100   transition-duration: 0.25s;
101 }
102
103 .fade-enter-active {
104   transition-delay: 0.25s;
105 }
106
107 .fade-enter,
108 .fade-leave-active {
109   opacity: 0;
110   transform: translateX(-25px);
111 }
112 </style>
```

Cofinanciado por: