

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

Unidade Curricular: Desenvolvimento Web – Front-End

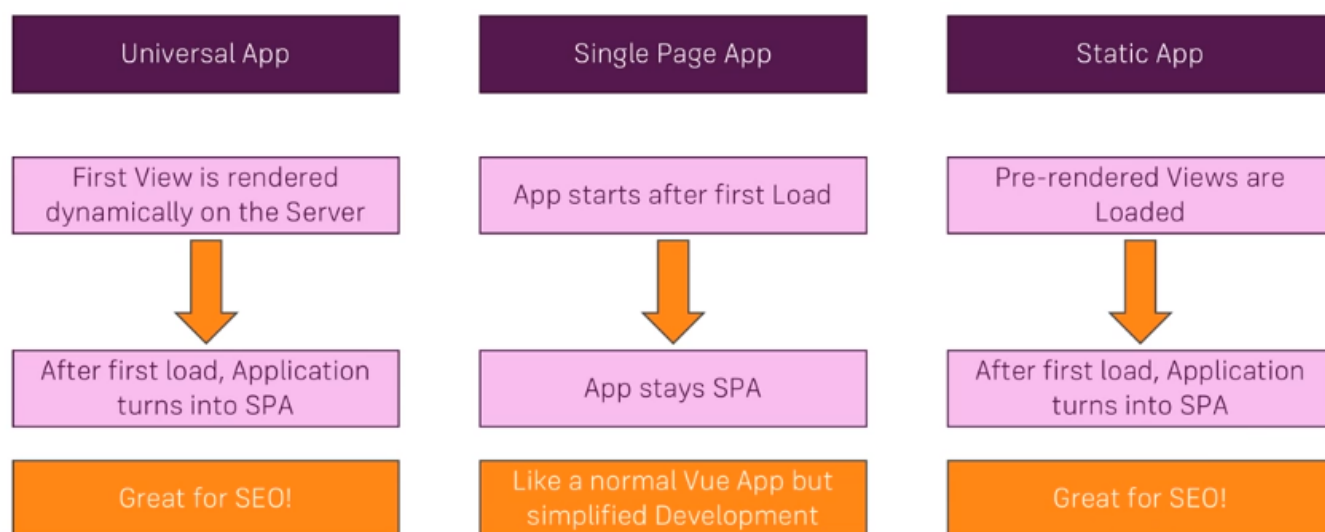
1.º Ano/2.º Semestre

Docente: Marco Miguel Olival Olim

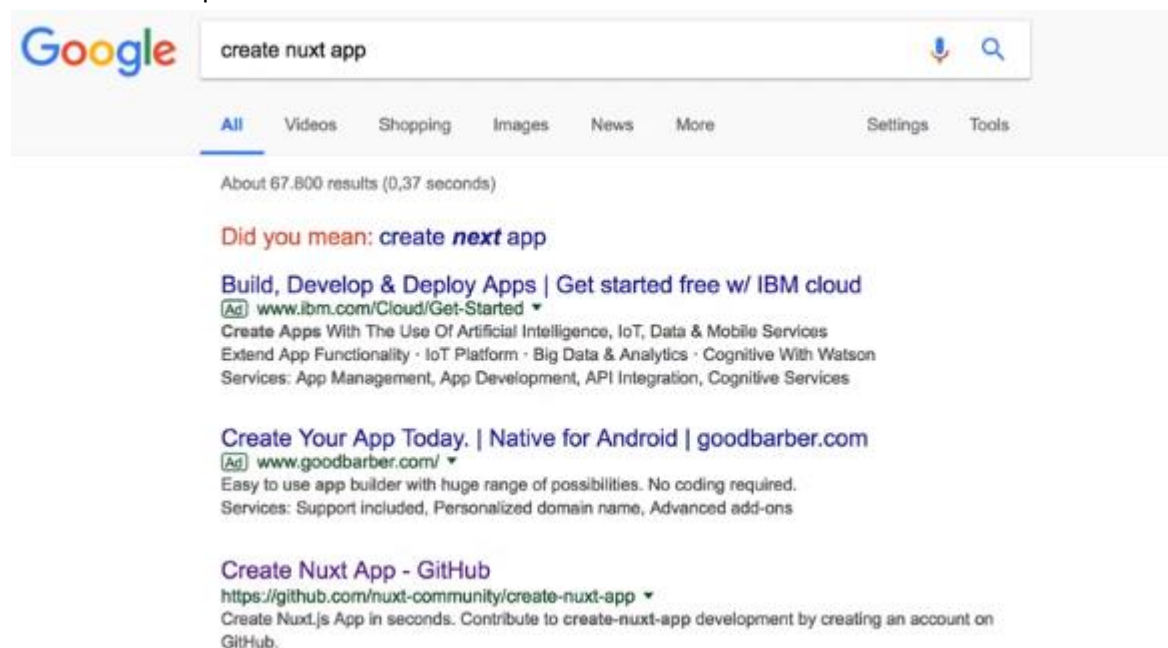
Data 16/03/2018

ESTE EXERCÍCIO PRETENDE ILUSTRAR O FUNCIONAMENTO DO CLI DO VUE JS

À semelhança do CLI utilizado na disciplina anterior, iremos dispor de um ambiente de desenvolvimento mais avançado de VueJS com o NUXT com valências de SSR (server side rendering),



O NUXT está disponível em:



Cofinanciado por:

É instalado globalmente com **npm install -g create-nuxt-app** e inicializa-se com a sintaxe **create-nuxt-app nome**

```
? Project name my-first-nuxt-app
? Project description My exquisite Nuxt.js project
? Use a custom server framework none
? Use a custom UI framework none
? Choose rendering mode Universal
? Use axios module no
? Use eslint (Use arrow keys)
> no
yes
```

Depois de instalado, poderá executar a aplicação com **npm run dev** como acontecia no Vue CLI

```
npm notice created a lockfile as package-lock.json. You should commit this file.
added 1240 packages in 23.579s
```

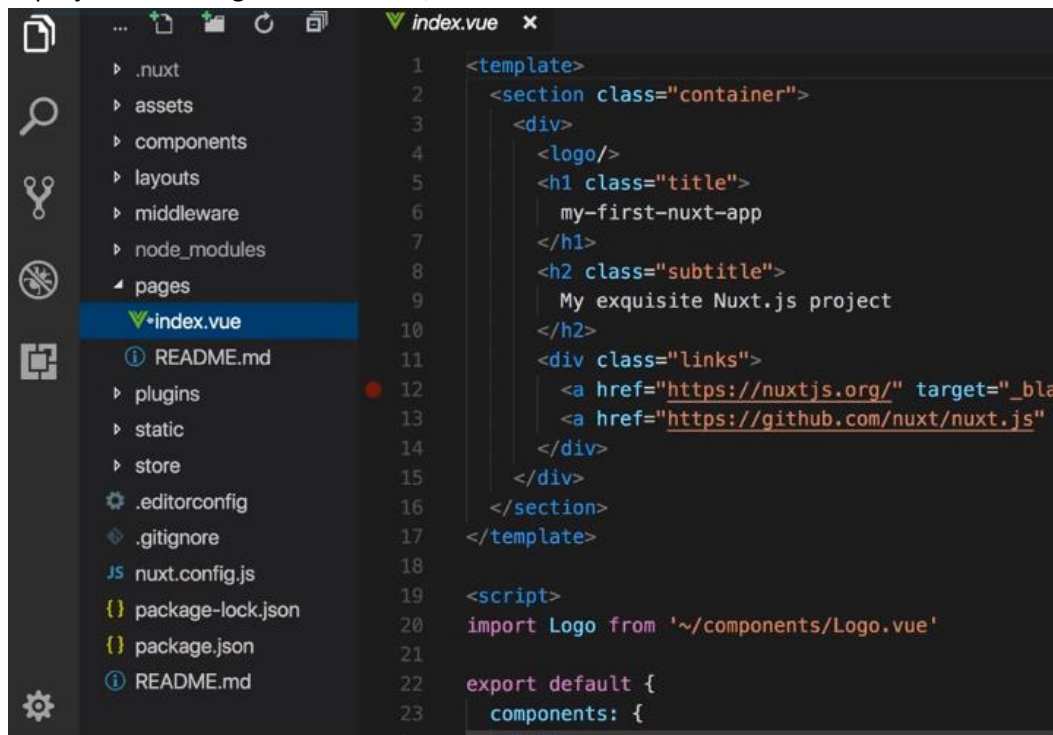
To get started:

```
cd my-first-nuxt-app
npm run dev
```

Deverá obter a seguinte página em <http://localhost:3000>

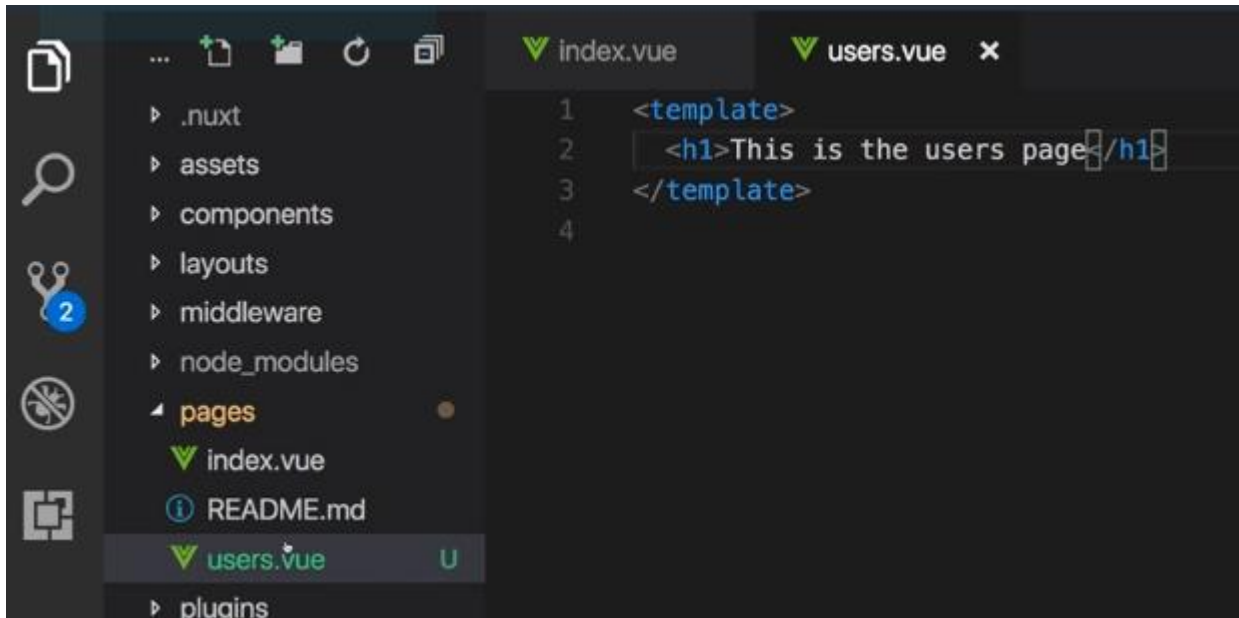


O projeto terá a seguinte estrutura, usando o Visual Studio Code

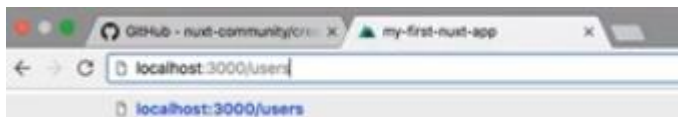


Cofinanciado por:

Para adicionar uma nova página a este site basta criar um ficheiro com a extensão .vue no diretório pages



Esta rota fica logo disponível para navegação no browser



Tradicionalmente estas rotas são definidas em formato JSON, mas o NUXT possibilita a simples estruturação de pastas para resolver estas rotas

```
{
  path: '/',
  component: WelcomeComponent
},
{
  path: '/products',
  component: ProductsComponent
},
{
  path: '/products/:id',
  component: ProductComponent
},
}
```

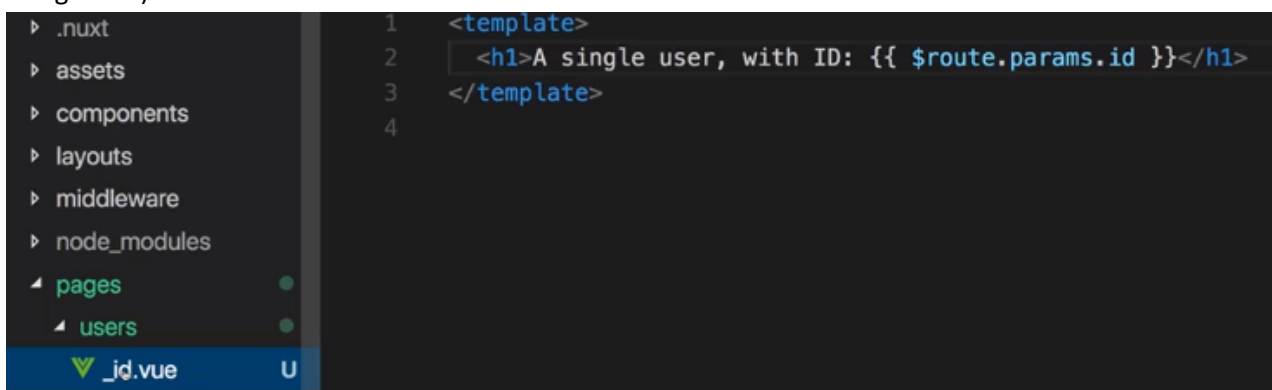


/
index.vue

/products
index.vue

/products/_id
index.vue

Em relação a rotas dinâmicas, estas tanto pode ser criadas pelo diretório e um ficheiro index.vue (nome obrigatório) ou utilizando um ficheiro com o nome da rota dinâmica



Cofinanciado por:

Para navegar para estas novas páginas devemos utilizar a sintaxe própria do router do NUXT, com nuxt-link, para evitar o page reload ao utilizar

```
1 <template>
2   <section class="container">
3     <nuxt-link to="/users">Users</nuxt-link>
4   </section>
5 </template>
6
7 <script>
8   import Logo from '~/components/Logo.vue'
9
10  export default {
11    components: {
12      Logo
13    }
14  }
15 </script>
```

Para navegar através do código é utilizada a sintaxe do vue router que serve de base ao NUXT

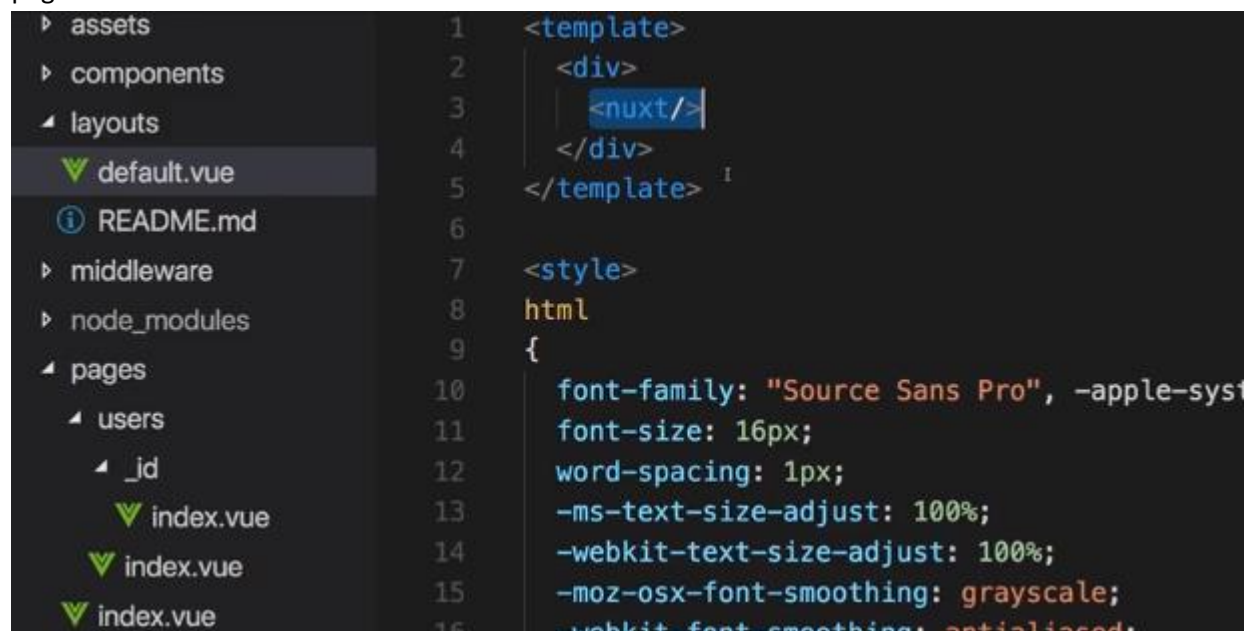
```
3   <input type="text" v-model="userId">
4   <button @click="onLoadUser">Load User</button>
5 </div>
6 </template>
7
8 <script>
9   export default {
10     data() {
11       return {
12         userId: ''
13       }
14     },
15     methods: {
16       onLoadUser() {
17         this.$router.push('/users/' + this.userId)
```

Para validar a rota dinâmica podemos forçar que esta seja um valor numérico

```
3 </template>
4
5 <script>
6   export default {
7     validate (data) {
8       return /\d+$/i.test(data.params.id)
9     }
10  }
11 </script>
12
```

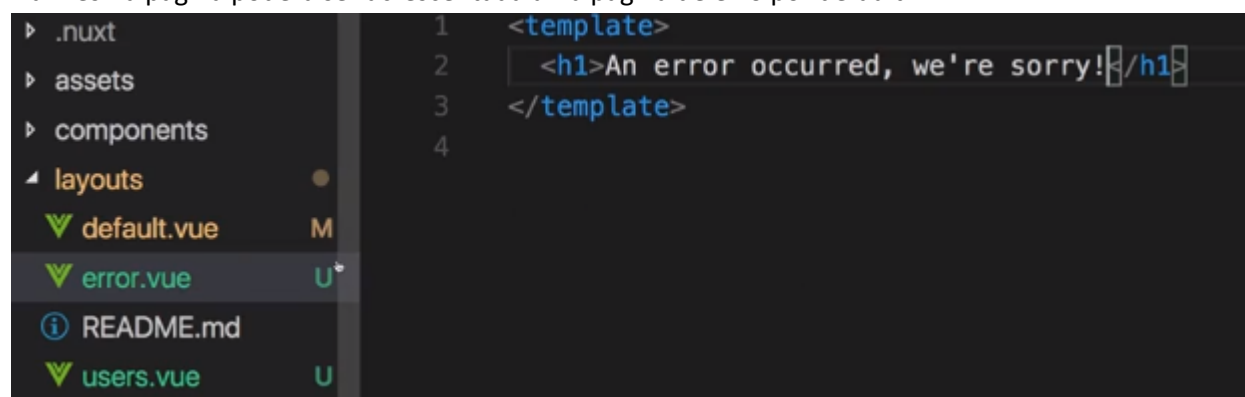
Cofinanciado por:

Se quisermos repetir conteúdo em todas as páginas, como por exemplo um header, footer, barra de navegação, etc... utilizamos o default.vue existente na pasta layout, onde <nuxt/> é onde será renderizada a página atual:



```
1 <template>
2   <div>
3     <nuxt/>
4   </div>
5 </template>
6
7 <style>
8   html
9   {
10    font-family: "Source Sans Pro", -apple-system, sans-serif;
11    font-size: 16px;
12    word-spacing: 1px;
13    -ms-text-size-adjust: 100%;
14    -webkit-text-size-adjust: 100%;
15    -moz-osx-font-smoothing: grayscale;
16    -webkit-font-smoothing: antialiased;
```

Na mesma página poderá ser acrescentada uma página de erro por default



```
1 <template>
2   <h1>An error occurred, we're sorry!</h1>
3 </template>
4
```

Cofinanciado por:

