

# ESCOLA E FACULDADE DE TECNOLOGIA SENAI ROBERTO MANGE

## DESENVOLVIMENTO DE SISTEMAS

JULHO DE 2022 — SENAI-SP



ESCOLA SENAI "ROBERTO MANGE"  
UM. RECONHECIDA EM. INDUSTRIAL

**SENAI**  
SÃO PAULO

## ESTRUTURA DO VUE

✧ Vamos aproveitar e entender a estrutura básica de qualquer página do vue:

```
index.vue x
pages > index.vue > Vetur > template > div > p
1  <script setup>
2    console.log("Hello Home Page!!!")
3  </script>
4
5  <template>
6    <div>
7      <h1>My Home Page!</h1>
8      <p>Welcome!!!</p>
9    </div>
10 </template>
11
12 <style scoped>
13   h1{
14     color: grey;
15   }
16 </style>
```

✧ **<template>** → esta é uma tag obrigatória em qualquer arquivo vue, e é nela onde ficará o html de nossa página.

É importante observar que debaixo da tag template só pode existir uma tag, ou seja, no exemplo ao lado, se queremos colocar várias tags (h1 e p, por exemplo) temos que colocá-las dentro de uma única tag, que no caso estamos usando a div mas poderia ser uma section, article, main, etc....

✧ **<script setup>** → dentro desta tag não obrigatória podemos colocar linguagem javascript. O atributo "setup" indica que este script é lido apenas antes de iniciar a página acessada.

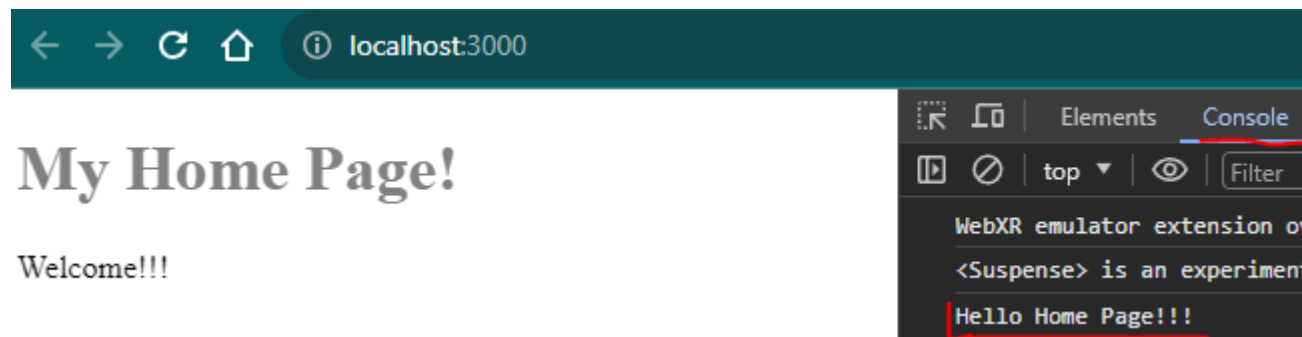
✧ **<style>** → dentro da tag style usamos o CSS da página, ou seja, as configurações que queremos alterar no aspecto visual da mesma. Esta tag não é obrigatória. O atributo "scoped" indica que este css deve ser válido apenas neste escopo, ou seja, apenas nesta página, sem afetar outras.



## ESTRUTURA DO VUE

✨ Vamos aproveitar e entender a estrutura básica de qualquer página do vue:

```
index.vue x
pages > index.vue > Vetur > template > div > p
1  <script setup>
2    console.log("Hello Home Page!!!")
3  </script>
4
5  <template>
6    <div>
7      <h1>My Home Page!</h1>
8      <p>Welcome!!!</p>
9    </div>
10 </template>
11
12 <style scoped>
13   h1{
14     color: grey;
15   }
16 </style>
```

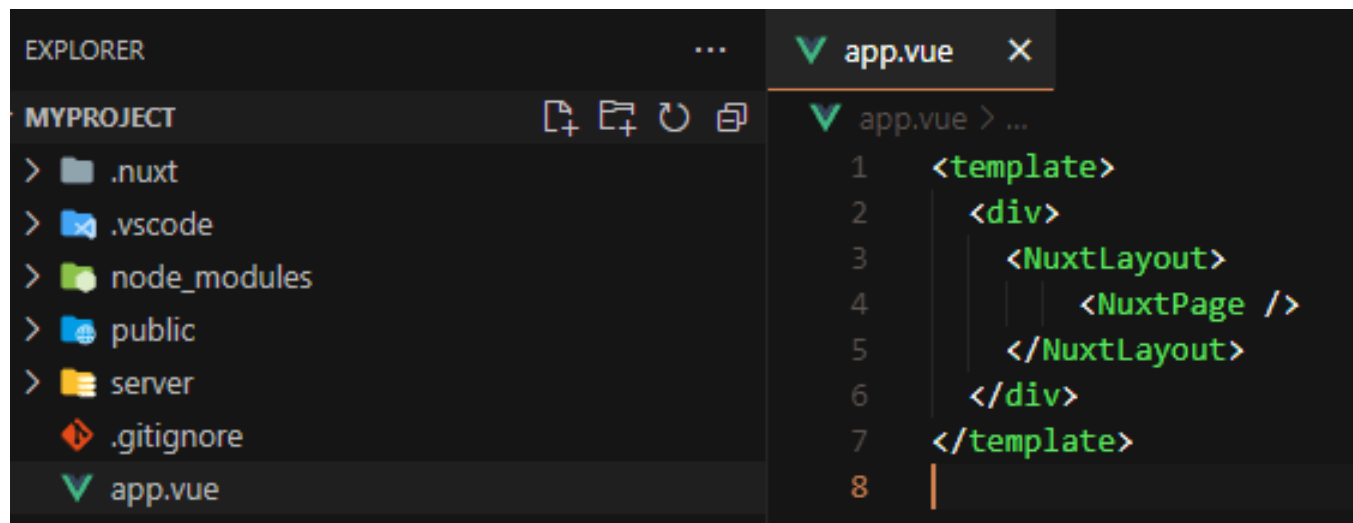


Observe que a página foi exibida com o html que está no **template**, com a cor definida no **style** (grey = cinza), e ao iniciar a página é impresso no console do navegador a mensagem que está dentro do **script** (clique com o botão direito do mouse e acesse o Inspecionar).

## CRIANDO PÁGINAS E ROTAS NO NUXT 3

Muito frequentemente as aplicações possuem várias páginas para organização do conteúdo a ser exibido, portanto aprenderemos com os passos a seguir como criar páginas e fazer com que elas sejam acessíveis através das rotas do servidor frontend.

- 1) Procure o arquivo “app.vue” em sua aplicação, este arquivo é o principal de todo o seu projeto, ou seja, todas as páginas e rotas que você criar serão inseridos automaticamente pelo Nuxt dentro do “app.vue” no momento de renderização. Inclua neste app.vue as tags **NuxtLayout** e **NuxtPage** para indicar que deseja usar Layouts e Páginas em seu projeto:

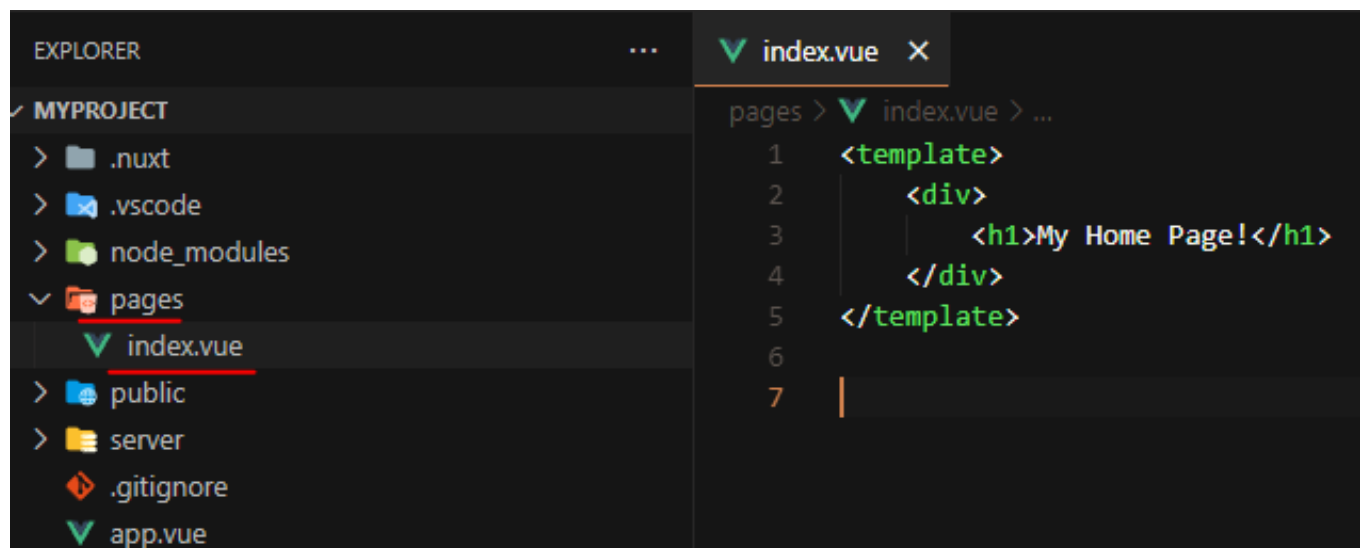


The screenshot shows the VS Code interface. On the left, the 'EXPLORER' sidebar displays the file structure of 'MYPROJECT'. The files listed are: .nuxt, .vscode, node\_modules, public, server, .gitignore, and app.vue. The 'app.vue' file is selected. On the right, the editor shows the content of 'app.vue'. The code is as follows:

```
1 <template>
2   <div>
3     <NuxtLayout>
4       <NuxtPage />
5     </NuxtLayout>
6   </div>
7 </template>
8
```

## CRIANDO PÁGINAS E ROTAS NO NUXT 3

- 2) Crie a pasta “pages” dentro da raiz de seu projeto;
- 3) Crie o arquivo “index.vue” dentro da pasta pages que você criou.



The screenshot shows the VS Code interface. On the left, the Explorer sidebar displays the project structure for 'MYPROJECT'. It includes folders like '.nuxt', '.vscode', 'node\_modules', 'pages', 'public', and 'server', along with files like '.gitignore' and 'app.vue'. The 'pages' folder is expanded, and 'index.vue' is selected. On the right, the editor shows the content of 'index.vue' with the following code:

```
1 <template>
2   <div>
3     <h1>My Home Page!</h1>
4   </div>
5 </template>
6
7 |
```

✎ Dessa maneira, se você seguiu os mesmos nomes indicados, o Nuxt vai entender que você deseja ter várias páginas em seu projeto e a página principal (home page) será a que está imediatamente abaixo da pasta pages com nome “index.vue”.

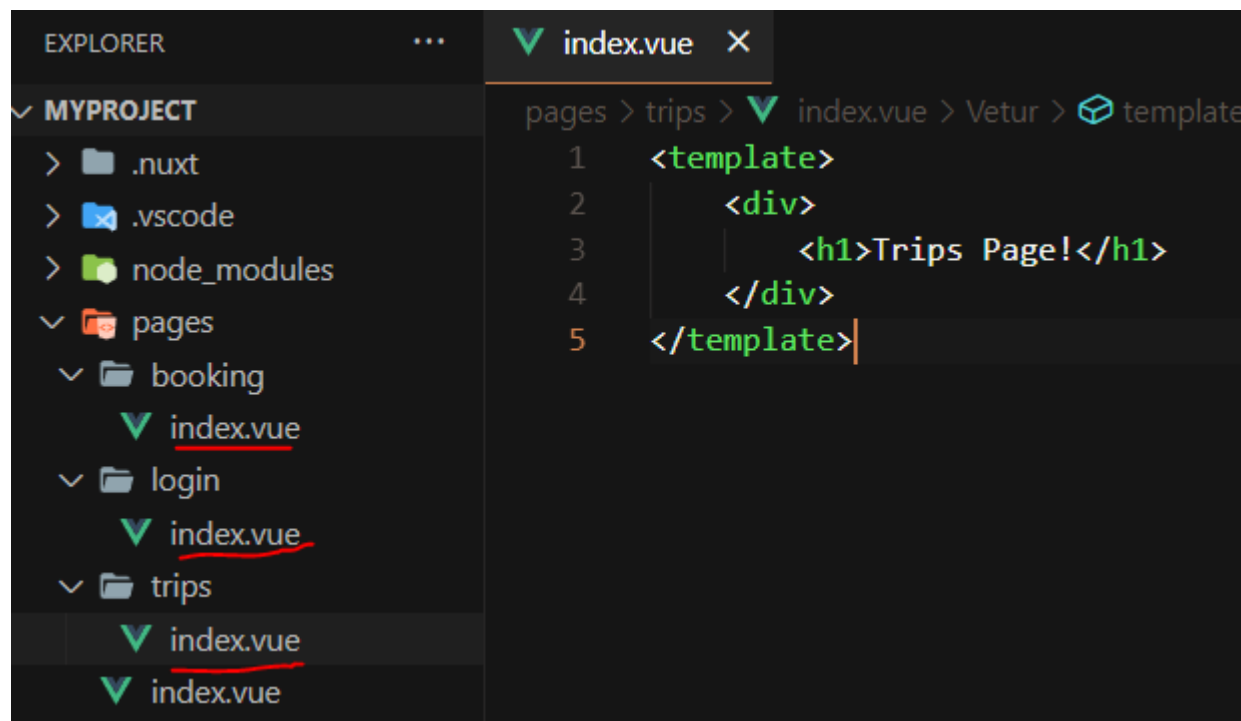
Esta página pode ser acessada quando se usa o endereço principal do servidor frontend:  
<http://localhost:3000>



My Home Page!

## CRIANDO PÁGINAS E ROTAS NO NUXT 3

4) Existem inúmeras formas de criarmos outras páginas, porém a mais organizada é criarmos subpastas com o nome das páginas que desejamos dentro da pasta pages, ou seja, se quisermos criar as páginas login, trips e booking, faremos da seguinte maneira:



The screenshot shows the VS Code interface. On the left, the Explorer sidebar displays the project structure: `MYPROJECT` with subfolders `.nuxt`, `.vscode`, `node_modules`, `pages`, `booking`, `login`, and `trips`. Each of these folders contains an `index.vue` file, which are highlighted with red underlines. The main editor area shows the content of `pages > trips > index.vue`. The breadcrumb at the top of the editor indicates the file path: `pages > trips > index.vue > Vetur > template`. The code in the editor is as follows:

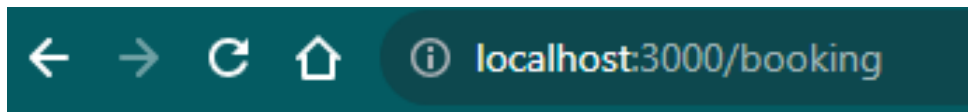
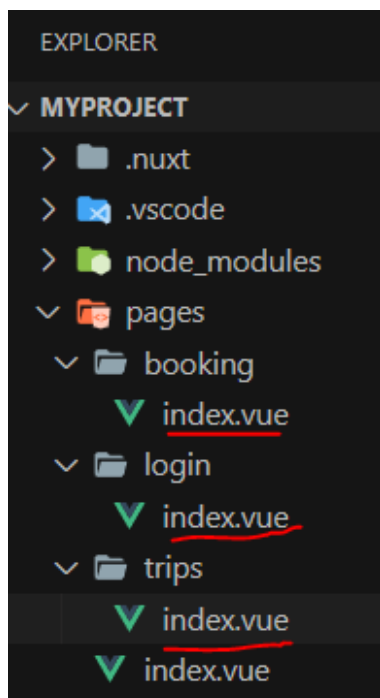
```
1 <template>
2   <div>
3     <h1>Trips Page!</h1>
4   </div>
5 </template>
```

📌 Cada pasta criada dentro da “pages” será considerada uma página com sua rota já configurada automaticamente pelo Nuxt, portanto, jamais crie pastas com espaços ou caracteres especiais com acentuação.

É importante que o arquivo dentro de sua pasta também seja chamado de “index.vue” para que o Nuxt entenda que ele é o arquivo principal que dever ser carregado na rota da página.

Outro ponto importante é que cada página deve ter ao menos a tag template com algum texto qualquer para que você consiga testar.

## CRIANDO PÁGINAS E ROTAS NO NUXT 3



# Booking Page!



# Trips Page!



# Login Page!



## CRIANDO PÁGINAS E ROTAS NO NUXT 3

5) Caso queira criar páginas com endereços dinâmicos, muito utilizadas quando queremos um detalhamento de algum elemento presente em uma lista, basta que criemos um arquivo com nome entre [ ]:

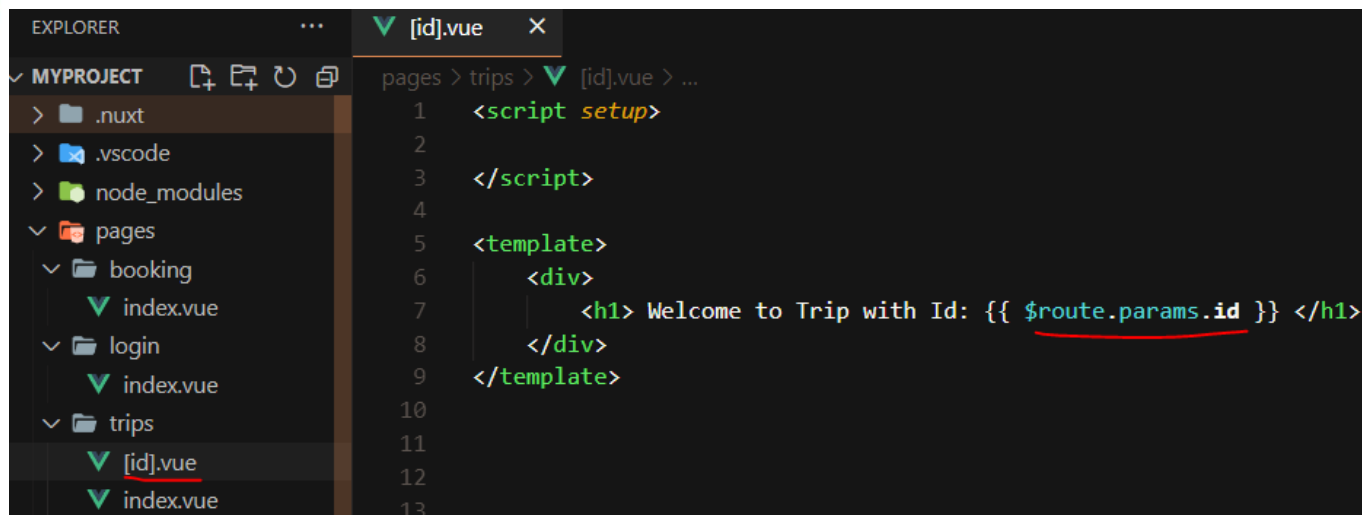
The screenshot shows the VS Code interface. On the left, the Explorer sidebar displays the project structure for 'MYPROJECT'. It includes folders like '.nuxt', '.vscode', 'node\_modules', 'pages', 'booking', 'login', and 'trips'. Under the 'trips' folder, the file '[id].vue' is selected and highlighted. The main editor area shows the content of '[id].vue'. It contains a script setup section and a template section. The template section has a single line of HTML: `<h1> Welcome to Trip with Id: {{ $route.params.id }} </h1>`. The variable `$route.params.id` is underlined in red.

```
1  <script setup>
2
3  </script>
4
5  <template>
6    <div>
7      <h1> Welcome to Trip with Id: {{ $route.params.id }} </h1>
8    </div>
9  </template>
10
11
12
13
```

📌 Note que dentro da pasta trips temos o “index.vue” e também o “[id].vue”, isso significa que quando acessado a rota <http://localhost:3000/trips> estaremos acessando o arquivo “index”, porém quando acessarmos por exemplo <http://localhost:3000/trips/3> estaremos acessando o arquivo “[id].vue” onde o número 3 será passado na variável “id” do objeto usado para as rotas do Nuxt, chamado `$route.params`



## CRIANDO PÁGINAS E ROTAS NO NUXT 3



The screenshot shows the VS Code interface. On the left, the Explorer sidebar displays the project structure: MYPROJECT, .nuxt, .vscode, node\_modules, pages, booking, login, trips, and [id].vue (selected). The main editor shows the content of [id].vue:

```
1 <script setup>
2
3 </script>
4
5 <template>
6   <div>
7     <h1> Welcome to Trip with Id: {{ $route.params.id }} </h1>
8   </div>
9 </template>
```

📌 Observe que o valor exibido na tela é variável, ou seja, podemos usá-lo para obter mais informações por exemplo de uma trip (viagem) em específico, chamando o backend para obter essas informações.

📌 Observe que o objeto route está entre duplas chaves {{ }}, usamos dessa forma quando queremos imprimir no html o valor de uma variável ou objeto.

📌 Outro ponto é que estamos usando o \$ antes do route, de forma simplificada, isso ocorre quando queremos acessar um objeto padrão do Nuxt.

localhost:3000/trips/3

Welcome to Trip with Id: 3

localhost:3000/trips/10

Welcome to Trip with Id: 10