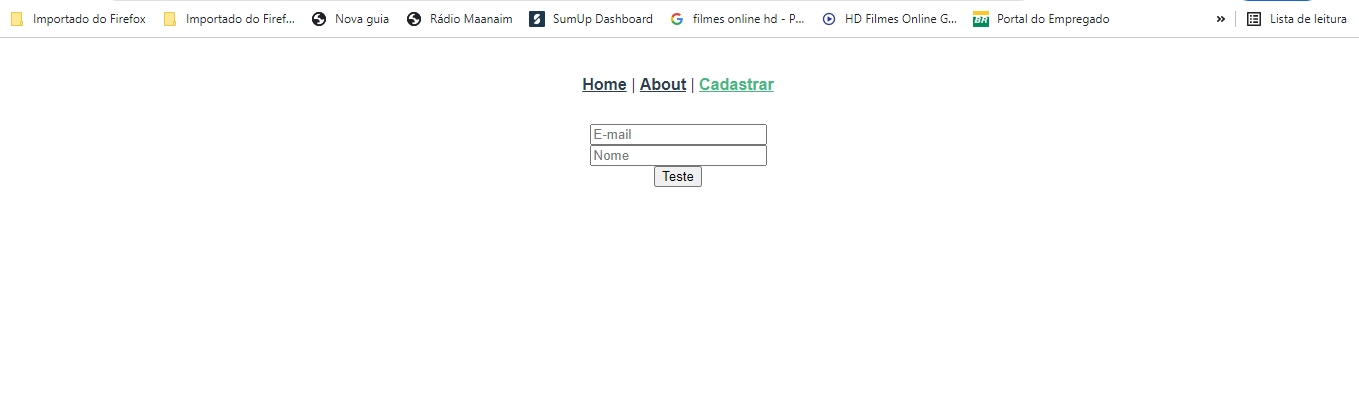
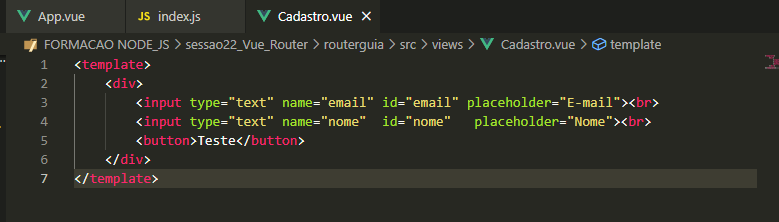
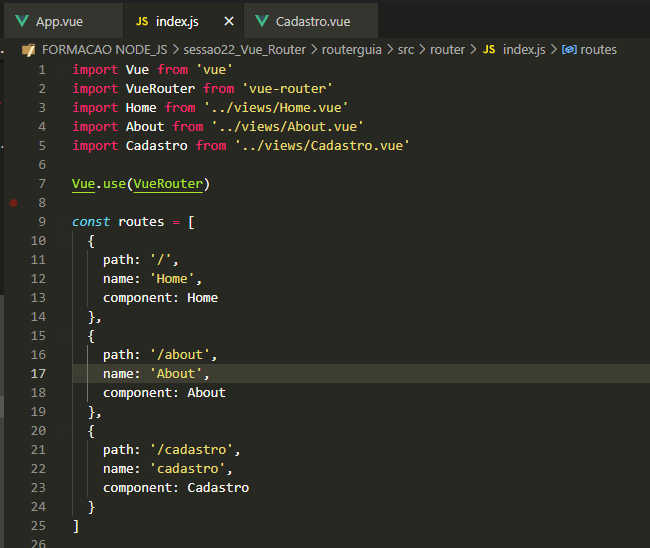
**Criando o Projeto com o Router**

1. Para criar as rotas no vueJs, utilizamos uma biblioteca chamada vouRouter.
2. Para testar essa biblioteca, vamos criar um novo projeto chamado routerguia.
3. Na pasta do projeto digite:
4. ‘vue create routerguia’
5. Selecione a opção ‘Manually select features’ para selecionar o que será adicionado à aplicação vue,
6. Navegue até Router e digite a barra de espaço para seleciona-lo.
7. Tecle enter
8. Tecle enter
9. A próxima tela pergunta se vc quer instalar o history mode. O history mode cria rotas mais amigáveis, porém quando for fazer o deploy será necessário alterar configurações do apache dependendo do servidor.
10. Mas vamos utilizar o history mode assim mesmo.
11. Digite Y
12. Pressione Enter na primeira opção,
13. Pressione Enter na primeira opção
14. Pressione Enter na primeira opção
15. Digite N e tecle Enter
16. Na aplicação podemos ver que foi criado um componente chamado ‘router-view’ responsável por criar a rota de um conteúdo.
17. Temos também outro componete o ‘router-link’ que é o mesmo do <a href:> do html.

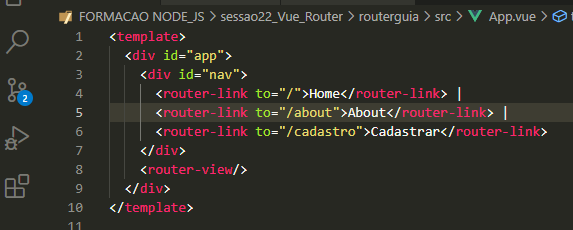
**Criando a primeira rota**

1. Para criar uma rota, devemos navegar até a pasta router e abrir o arquivo index.js
2. No arquivo index vamos até o array routes e para cada nova rota que for criada, basta somente colar uma virgula, abrir um novo objeto, neste objeto vamos inserir o path da rota, o name e o componente que será aberto com a nova rota. Neste caso criamos uma rota ‘/cadastro’ que irá abrir o componente cadastro.
3. 
4. Para testar basta digitar a url no navegador: localhost:8080/cadastro e aparecerá o conteúdo do componente cadastro..
5. 

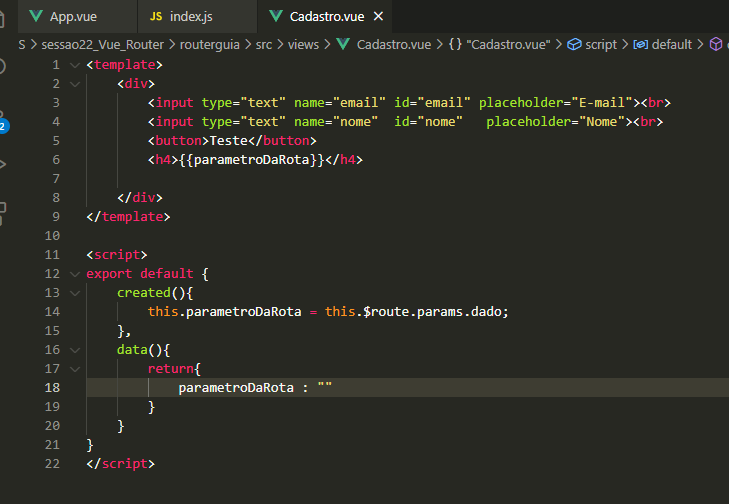
**Criando um componente para a rota**

1. Navegue até a pasta view e crie um arquivo chamado Cadastro.vue
2. No arquivo crie um template com uma div principal
3. Insira input tipo texto para email e nome e botão teste.
4. 
5. Após a criação do componente, vá no arquivo index.js e importe o componte criado, após isso basta ir na rota que criamos e adicioná-lo ao array de rotas.
6. 

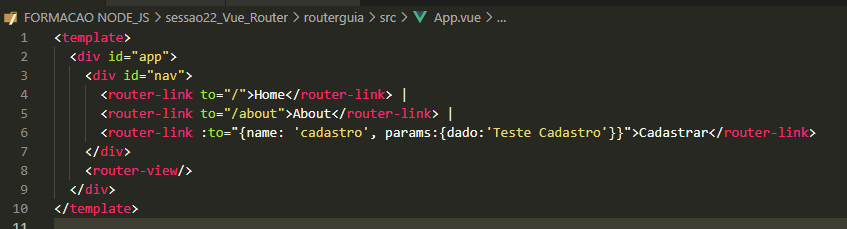
**Criando um link para o Cadastro**

1. No arquivo APP.vue crie um novo router-link para acessar a view do cadastro.
2. 

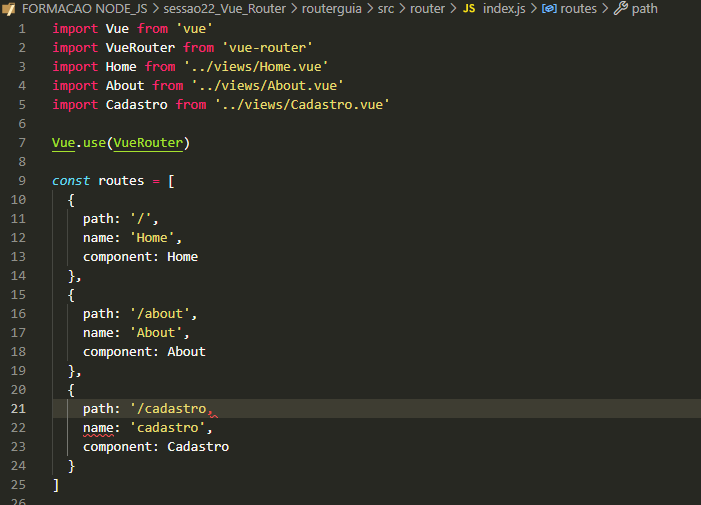
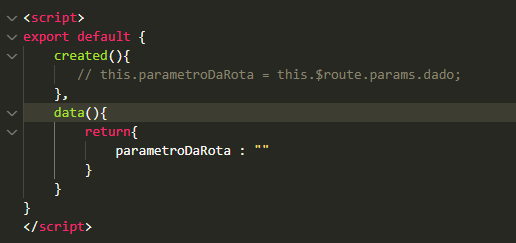
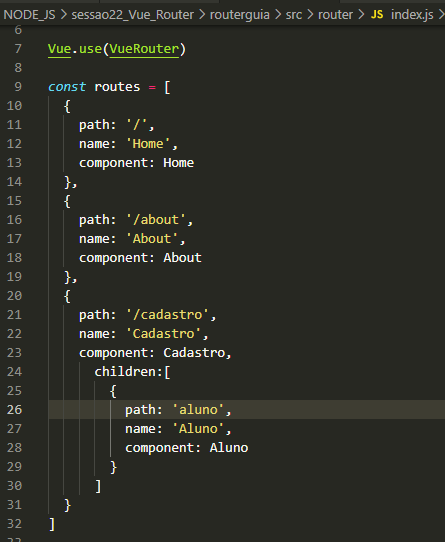
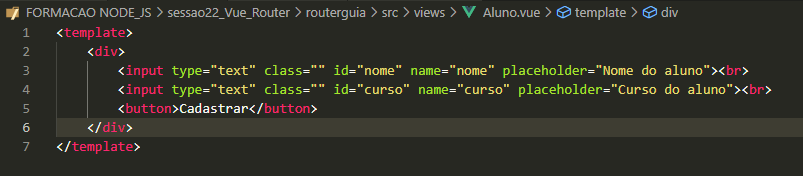
**Criando rotas dinâmicas com parâmetros**

1. Para criar rotas dinâmicas com parâmetros, basta ir no arquivo index.js no array de rotas e incluir o parâmetro em frente ao caminho, neste caso vamos inserir o parâmetro chamado :dado
2. Após isso vamos no componente que conversa com a rota e criamos dentro da área de script um método chamado created, que será chamado sempre que o componente for criado. No método created nós acessamos o parâmetro informado na rota, através de $route.parans.”nome\_do\_parâmetro”. Lembrando que sempre que for referênciar devemos utilizar o this. Se utilizarmos um console.log($route.parans.dado) veremos o parâmetro que passamos na rota.
3. Vamos definir a função data com uma variável chamada parametroDaRota, essa variável será passada para o componente e mostrado dentro de um h4.
4. <h4>{{parametroDaRota}}</h4>
5. Mas para isso temos que referenciar a variável recebendo o dado que foi digitado na rota.
6. Isso dentro da função created().
7. This.parametroDaRota = this.$route.params.dado;
8. 

**Criando links dinâmicos**

1. No arquivo App.vue vamos fazer um bind no link que vamos transformar em dinâmico, apagar a rota e passar um objeto javaScript com o nome da rota e o parâmetro criado no array de rotas.
2. 

**Nested Routes (Rotas aninhadas)**

1. Para colocarmos uma rota dentro de outra, vamos até o arquivo index.js no array de rotas, vamos apagar o parâmetro :dado passado na rota cadastro.
2. 
3. Vamos comentar também a criação da variável parâmetroDaRota no componente Cadastro
4. 
5. No arquivo index.js no array de rotas vamos criar uma rota filha para a rota cadastro, lembrando que a rota children é um array e por isso podemos criar várias rotas filhas.
6. 
7. Vamor criar um componente chamado Aluno para testar.
8. 
9. Importe o componente no arquivo index.js para a rota funcionar e no componente da rota pai, devemos inserir um <router-view> para que o componente filho seja exibido.
10. 