Roteiro

React

Antes de iniciar este roteiro, teremos feito um roteiro base com jsx, babel e webpack.

Ordenação:

Criar projeto

Home

TiposEventos

React Router

Component Rodapé

TiposEventos – Listar da API

TiposEventos – Cadastrar chamando a API

Instalar NodeJS

npm install -g create-react-app

create-react-app senai-svigufo-ui

cd senai-svigufo-ui

npm start

Links úteis

<https://medium.com/tableless/o-guia-completo-do-react-e-o-seu-ecossistema-b31a10ecd84f>

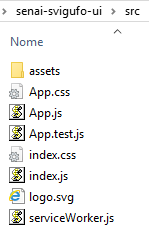
<https://medium.com/tableless/organizando-uma-aplica%C3%A7%C3%A3o-com-react-5b8ea9075596>

<https://medium.com/tableless/webpack-para-react-o-guia-final-cb8a95b369ed>

Criar projeto

Classes

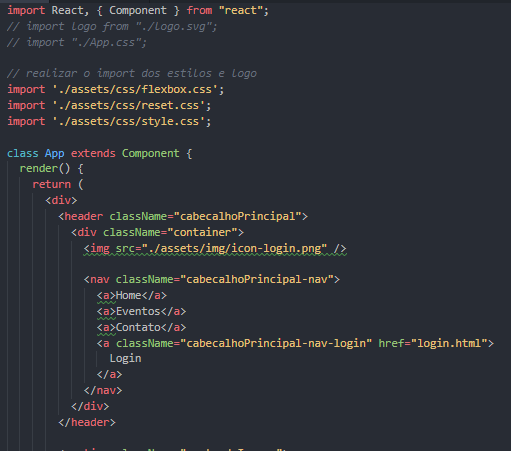
Copiar a pasta assets definida no projeto base e colar dentro de src do projeto do React.



Abrir o App.js e colar o conteúdo dentro de body no render() -> return do conteúdo da home.html do conteúdo base.

Alterar tudo o que for class para className.

Realizar um npm start para verificar se o conteúdo foi gerado.

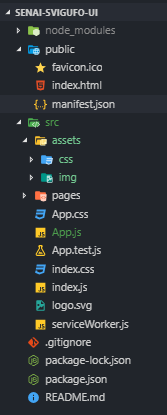


Importando o ícone e o incluindo na tela.

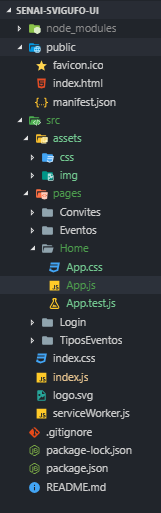


Após termos colocado a home, vamos realizar um cadastro de tipo de evento para verificar que não há diferença entre eles.

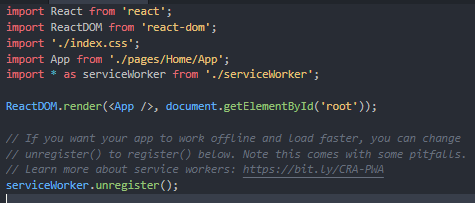
Para melhor organização, vamos criar uma pasta chamada pages.



E aproveitar, criar subpastas para organizar os nossos códigos.



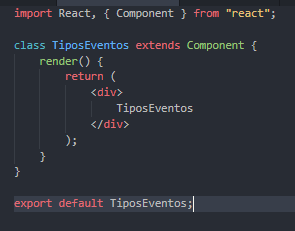
Alterar o index.js para encontrar a página de Home



App.js



Criar TiposEventos.js dentro da pasta de pages/TiposEventos com o conteúdo abaixo.



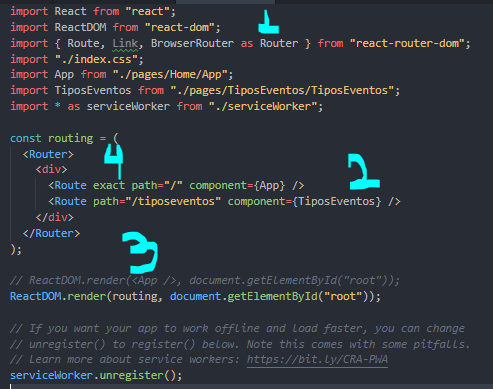
Para que essa página possa ser acessada, precisamos instalar o react-router.

<https://medium.com/collabcode/roteamento-no-react-com-os-poderes-do-react-router-v4-fbc191b9937d>

Parar o projeto e executar o seguinte comando. Para dar suporte à navegação do projeto.

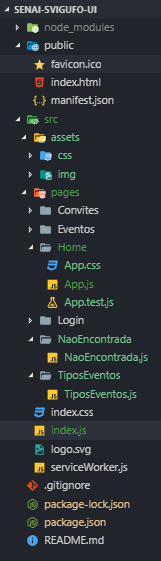
npm install --save react-router-dom

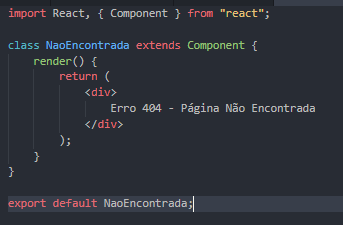
index.js



Sugestão de Página Não Encontrada

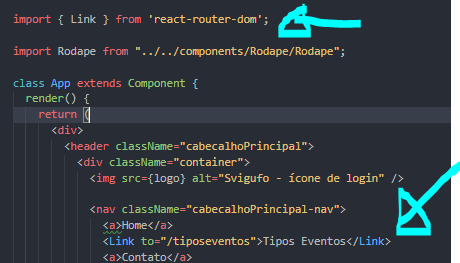
Criar uma pasta chamada NaoEncontrada bem como uma página NaoEncontrada.js com o seguinte conteúdo.





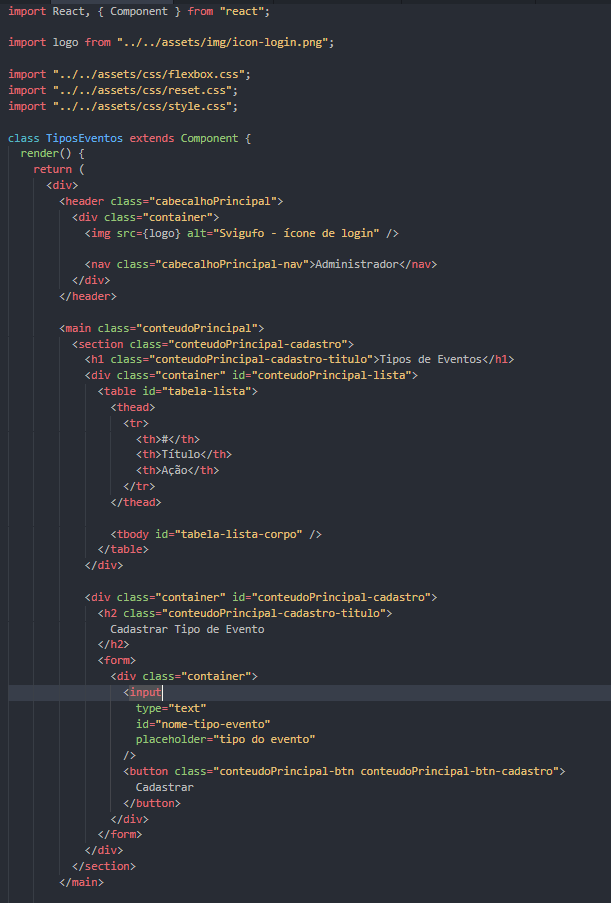


Criando uma rota de exemplo



TiposEventos.js

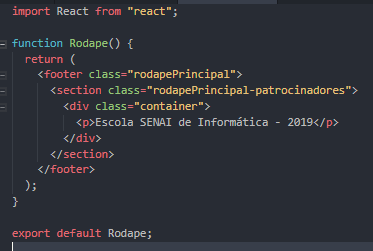
Copiar todo o conteúdo para a tela de TiposEventos.



Lembrar de trocar o que for class para className.

Criando apenas um rodapé.

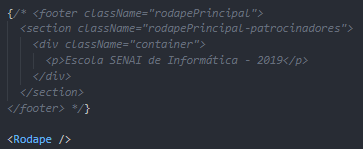
Criar uma pasta chamada componentes com uma pasta chamada Rodape e um arquivo chamado Rodape.js.



Lembrar de trocar o que for class para className.

App.js





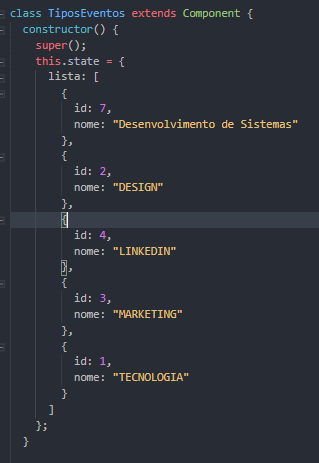
Realizar o mesmo no TiposEventos.js



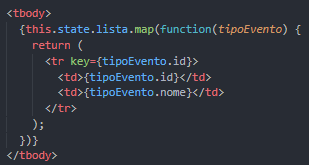


TiposEventos.js

Mostrar os tipos de eventos da API



Lembrar de deixar sem o key. Depois incluir.



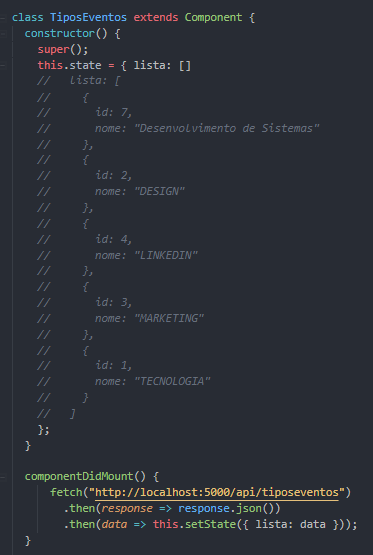
Ciclo de Vida

<https://medium.com/@edmo_/m%C3%A9todos-do-ciclo-de-vida-de-componentes-reactjs-um-mergulho-profundo-332ed7b3b782>

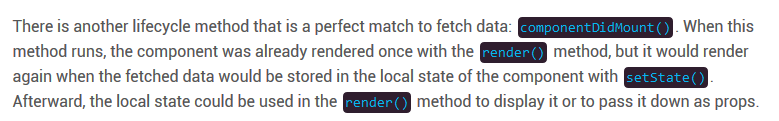
**Uso:** componentWillMount é usado para inicializar os estados ou adereços, há um grande debate para fundir isso com o construtor.

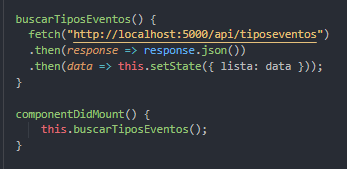
**Onde você deve fazer as chamadas API?**

*As chamadas da API devem ser feitas sempre no método componentDidMount.*

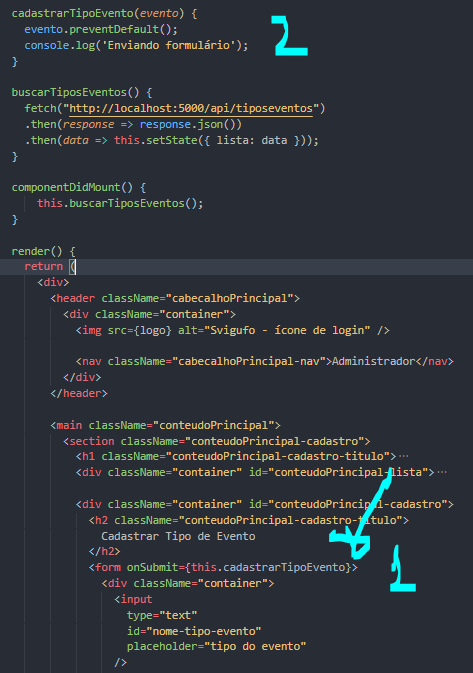


<https://www.robinwieruch.de/react-fetching-data/>





Cadastrar um tipo de evento



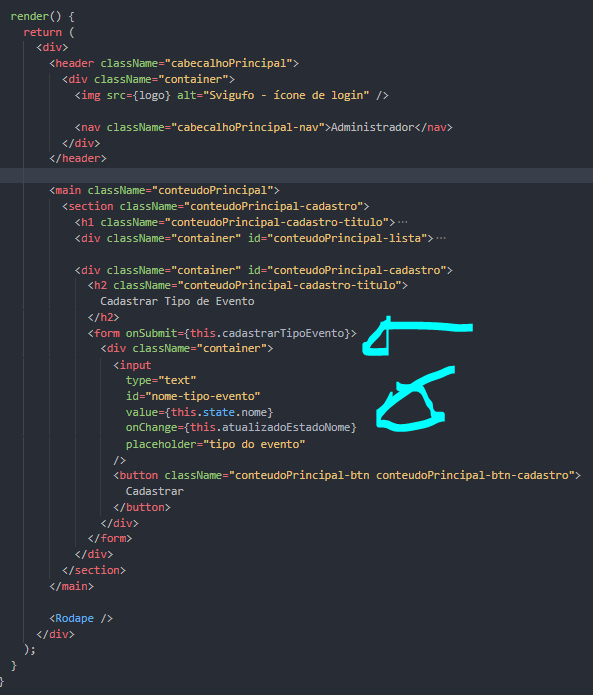
<https://medium.com/@rossbulat/an-introduction-to-using-form-elements-in-react-3778042ff334>

<https://medium.com/capital-one-tech/how-to-work-with-forms-inputs-and-events-in-react-c337171b923b>



LEMBRETE: colocar somente response no formulário, para que ele não forneça nenhum erro e atualize a lista da tela.



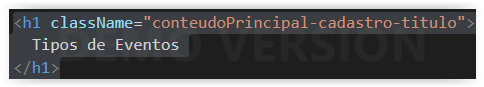


Criar Componente com props.

Va na pasta components e crie um componente chamado Titulo.js



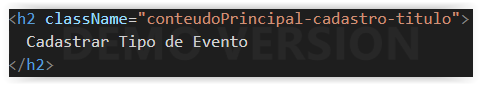
Importe o componente Titulo para TiposEventos.js e altere no render



Para



E



Para



Mostrar props pelo state, incluir titulocadastro e alterar no componente

<Titulo titulo={this.state.titulocadastro} />

Login

Para o login iremos utiliza o axios

<https://github.com/axios/axios>

Instalar

npm install --save axios

Crie uma pasta Login dentro da Pasta Pages

Em render implemente o html de svigufo-js-base e altere onSubmit do form, value e onChange do input.



Implemente o código abaixo em Login.js, defina o state e crie as funções para atualizar o input



Crie a função para efetuarlogin

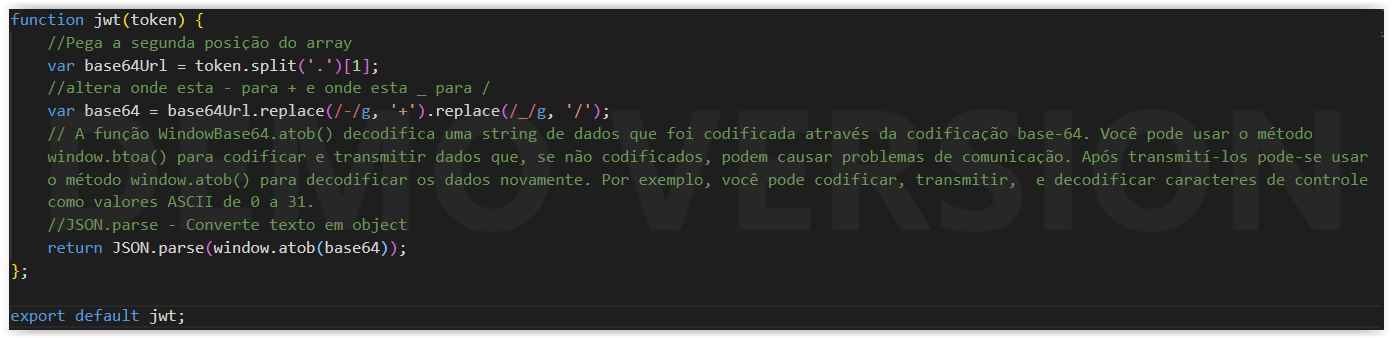
Falar sobre axios, data.status, localstorage(chave, valor) e history.push podendo no valor armazenar um json, fazer primeiro sem

console.log(jwt(localStorage.getItem("usuario-svigufo")));

Mostrar zapimoveis, ifood



Falar sobre reverter o jwt. Criar componente jwt, criar pasta útil e criar componente jwt.js

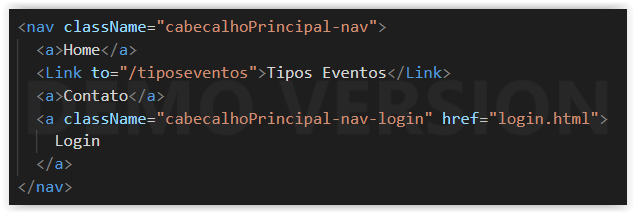


Importar jwt para Login.js, implementar

console.log(jwt(localStorage.getItem("usuario-svigufo")));

Alterar os links de navegação do app.js e importar

import { Link } from "react-router-dom";



Permissão de página

Criar uma pasta services e criar um arquivo auth.js com o código abaixo

export const usuarioAutenticado = () => localStorage.getItem("usuario-svigufo") !== null;

Implementar em Index.js a permissão de Admin e importar auth



Alterar na rota TiposEventos para

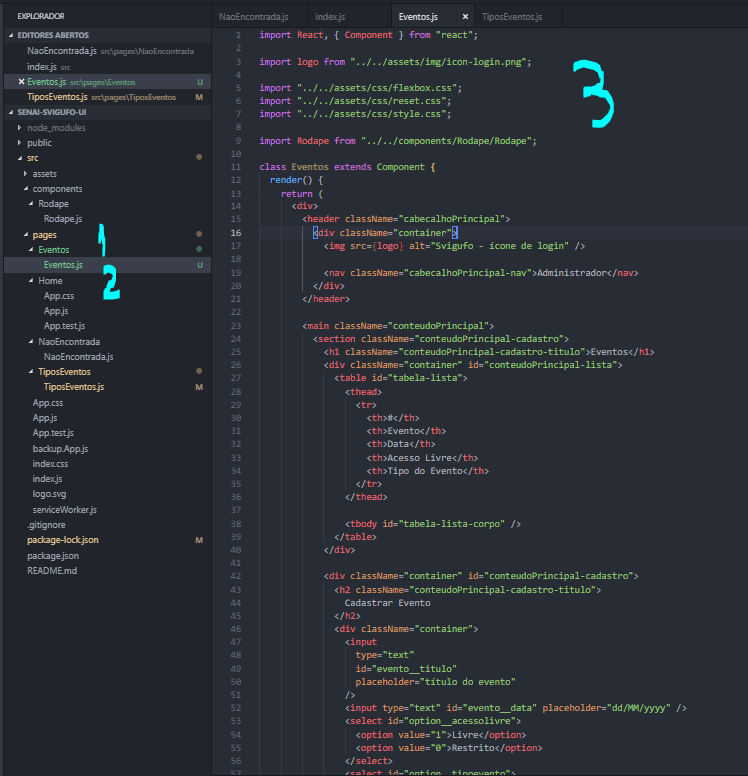
<PermissaoAdmin path="/tiposeventos" component={TiposEventos} />

Testar

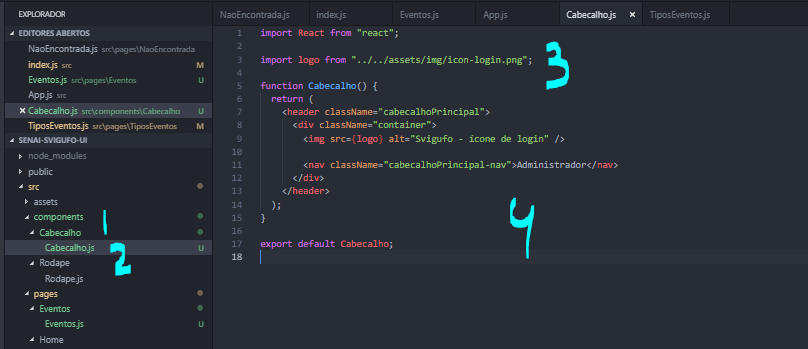
Criar uma pasta chamada Eventos

Criar um arquivo chamado Eventos.js

Realizar os devidos imports e copiar o conteúdo de eventos.html (da pasta base). Lembrar de fechar as tags de inputs e trocar as ‘class’ para ‘className’.

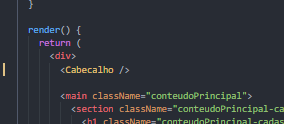


Criar um componente de cabeçalho

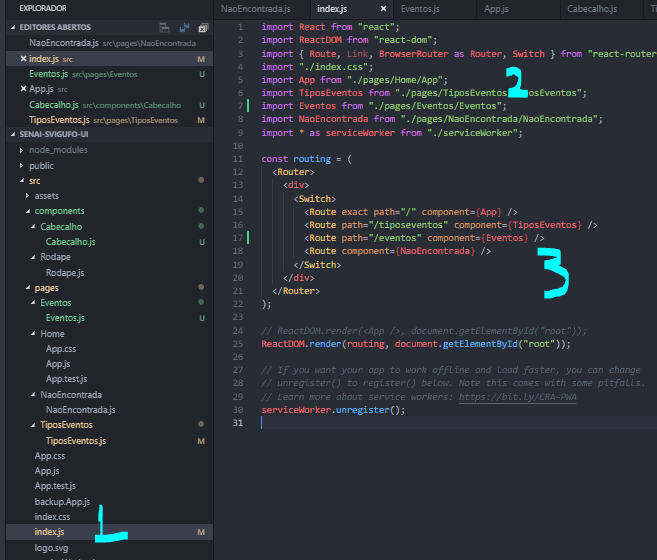


Importar este componente nas duas páginas





Para verificar que a página evento foi criada, devemos incluí-la na rota.



Listar os eventos



Montando o select



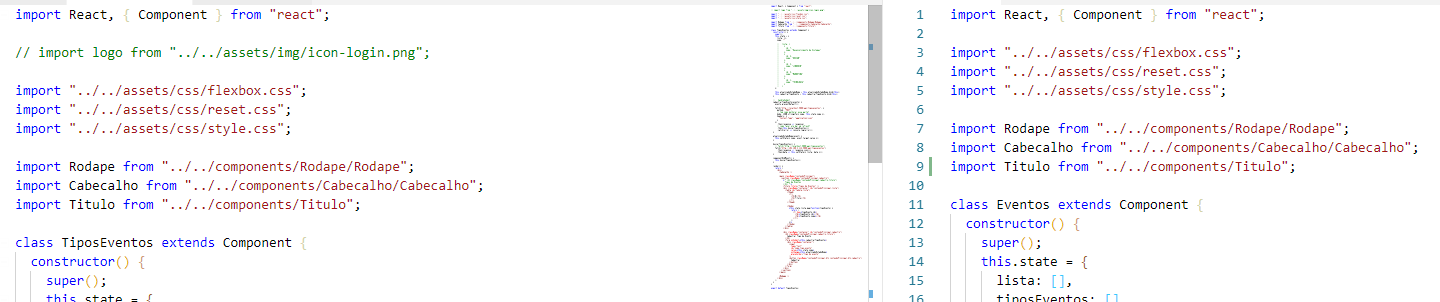
`

Props

Criar um componente de Titulo.js dentro da pasta de componentes.



Importar o título dentro da tela de evento e tipos de eventos.



Atualizar eventos e tipos de eventos com a props correspondente.

