Apresentação – React

Revisão – JSX, Babel

Ordenação:

Criar Projeto

Home

Categorias

React Router

Componente Rodapé

Categorias – Listar API

Categorias – Cadastrar

Login

Eventos – Listar API

Props

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>

<https://facebook.github.io/create-react-app/docs/getting-started>

**Instalar NodeJS**

npm install -g create-react-app

**Criar Projeto**

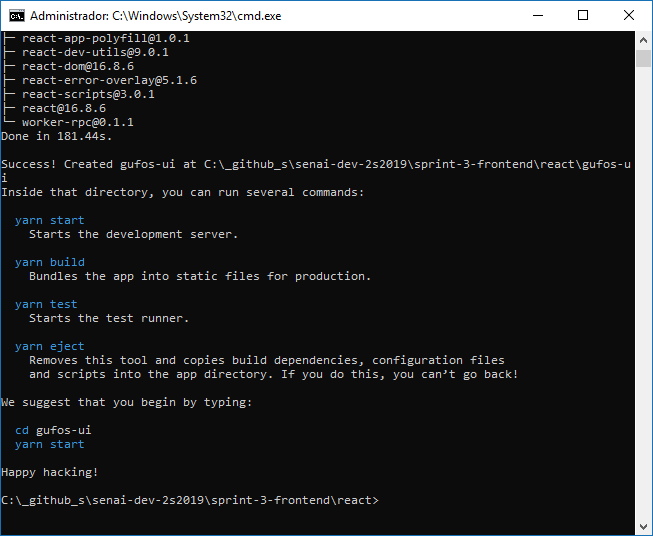
create-react-app gufos-ui

create-react-app is a global command-line utility that you use to create new projects.

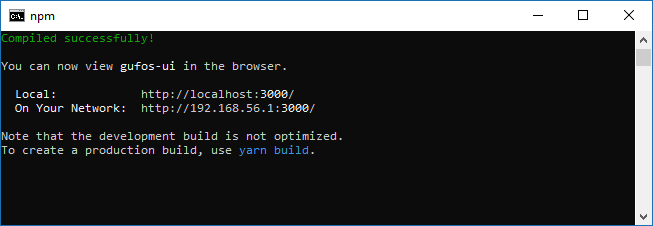
**Executar Projeto**

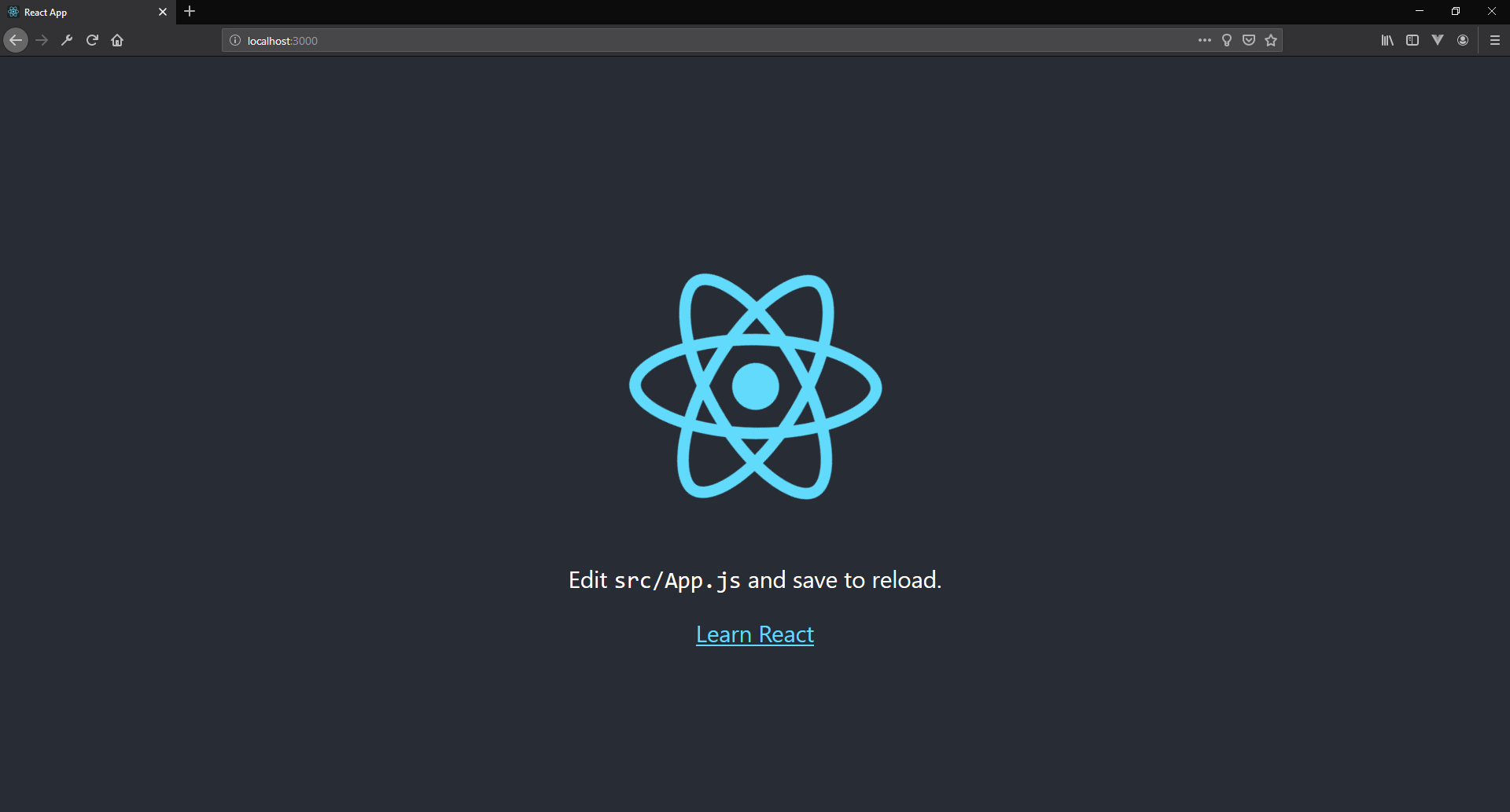
cd gufos-ui

npm start

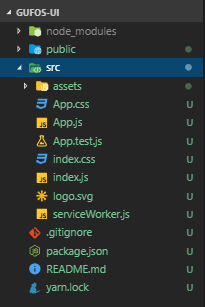


localhost:3000





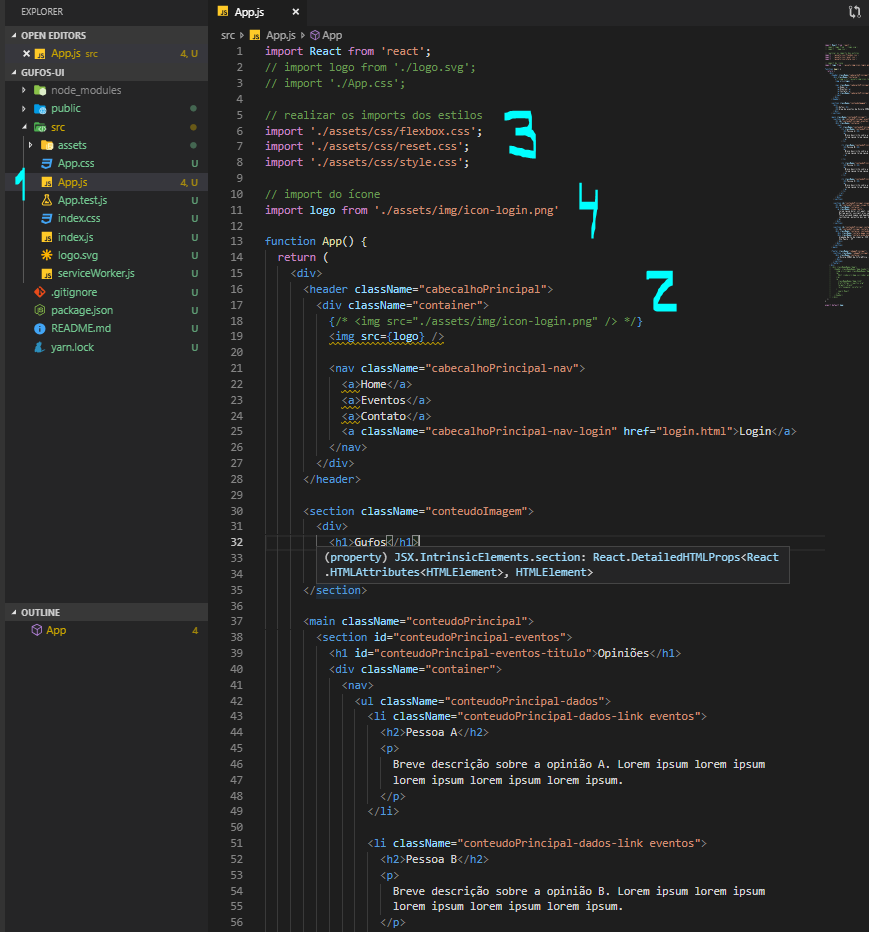
Dentro do projeto do React – gufos-ui, criar uma pasta chamada assets e copier o conteúdo da mesma pasta em gufos.base para este local.



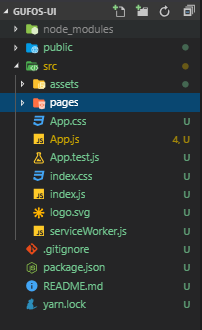
Copiar todo o conteúdo de home.html e colar em App.js.

Alterar os imports do css e das imagens.

Alterar tudo o que for class para className.



Criar uma pasta chamada pages para melhor organização.



Criar pastas dentro de pages para organizar o nosso conteúdo.

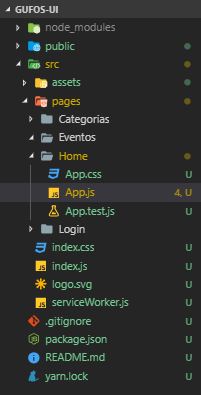
Eventos

Categorias

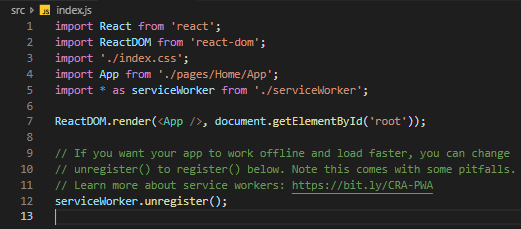
Home

Login

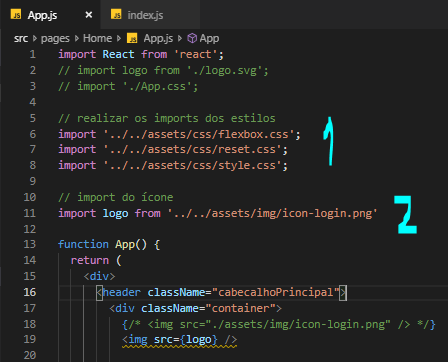
Copiar o conteúdo de App.js, App.test.js e App.css.



Alterar no index.js o destino do App.js.

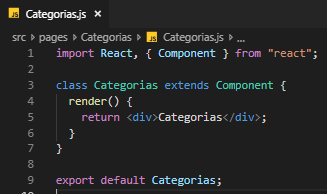


Alterar dentro de App.js, o destino de assets.

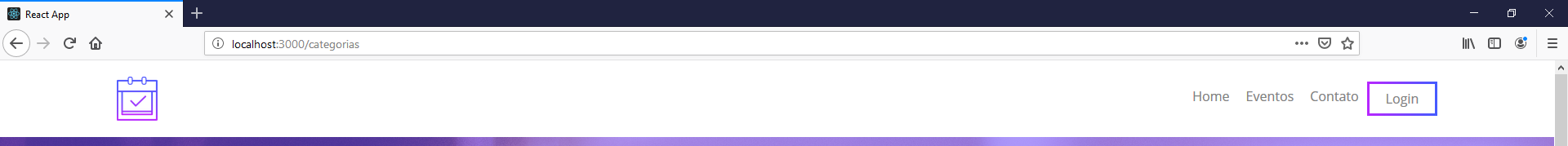


Vamos começar com a parte de Categorias dos eventos.

Dentro da pasta pages -> Categorias, criar um arquivo Categorias.js.



Acessar localhost:3000/categorias



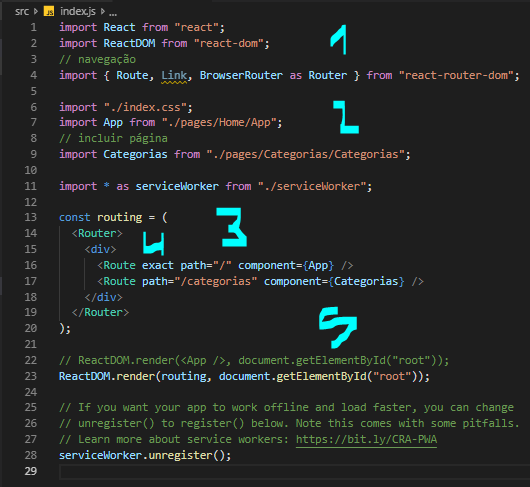
Para que essa página possa ser acessada, precisamos instalar o react-router.

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

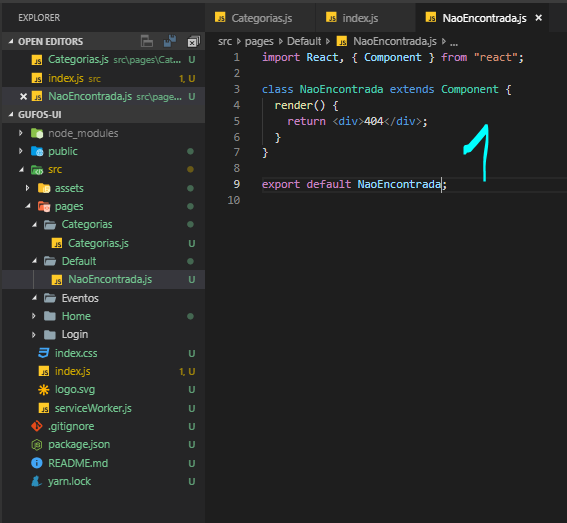
Parar o projeto e executar o seguinte comando. Para dar suporte à navegação do projeto.

npm install --save react-router-dom

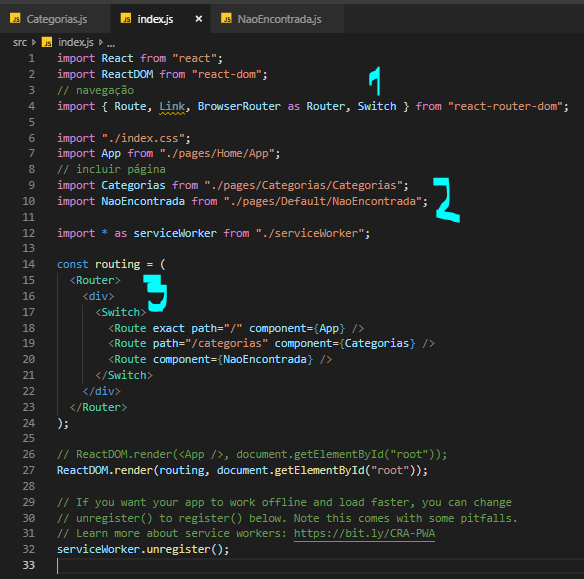
Após ter instalado a dependência acima, altera o index.js para realizer a configuração.



Criar uma página Não Encontrada, caso passe alguma rota que não será encontrada.



Alterar o arquivo index.js para atender a chamada solicitada.



Copiar todo o conteúdo de categorias.html do projeto gufos.base e colar no Categorias.js dentro do return.

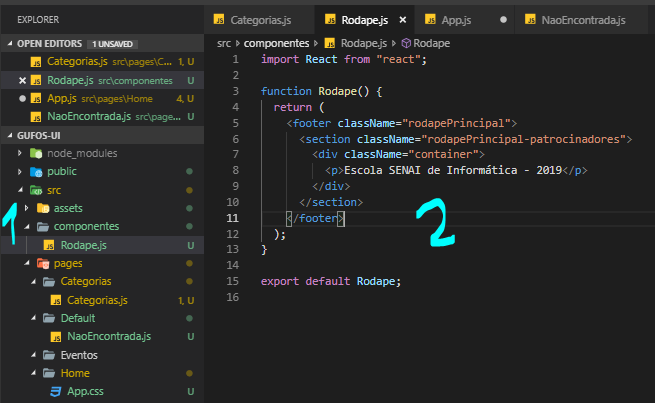


Alterar tudo de class para className.

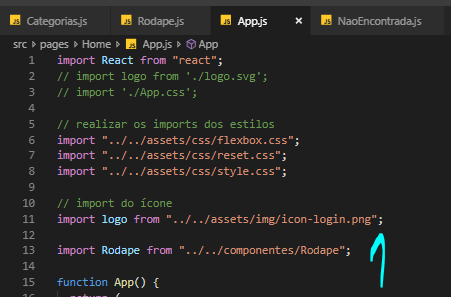


Verificar o componente do rodapé. Tanto na Home quanto em Categorias.

Criar uma nova pasta chamada components e dentro dela um arquivo Rodape.js com o seguinte conteúdo.



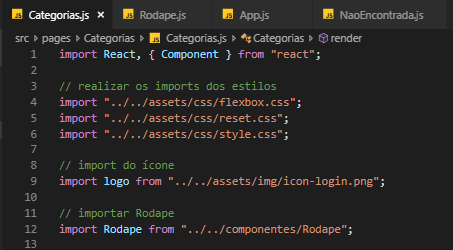
No arquivo App.js, importar o arquivo Rodapé criado.

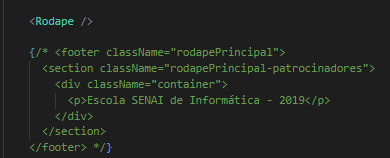


Incluí-lo no corpo.



Realizar a mesma alteração no arquivo Categorias.js.





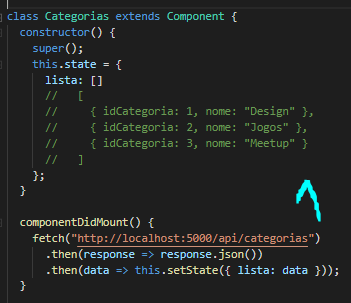
Acessar localhost:3000/ e localhost:3000/categorias e mostrar o resultado do componente criado.

Alterar a lista de categorias para ser apresentada através de uma lista fixa local.



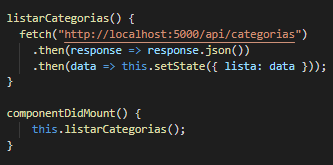
Lembrar de alterar o backend para deixar as categorias sem o [Authorize]. E também incluir o cors.

Alterar para que as chamadas sejam feitas para a API.



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

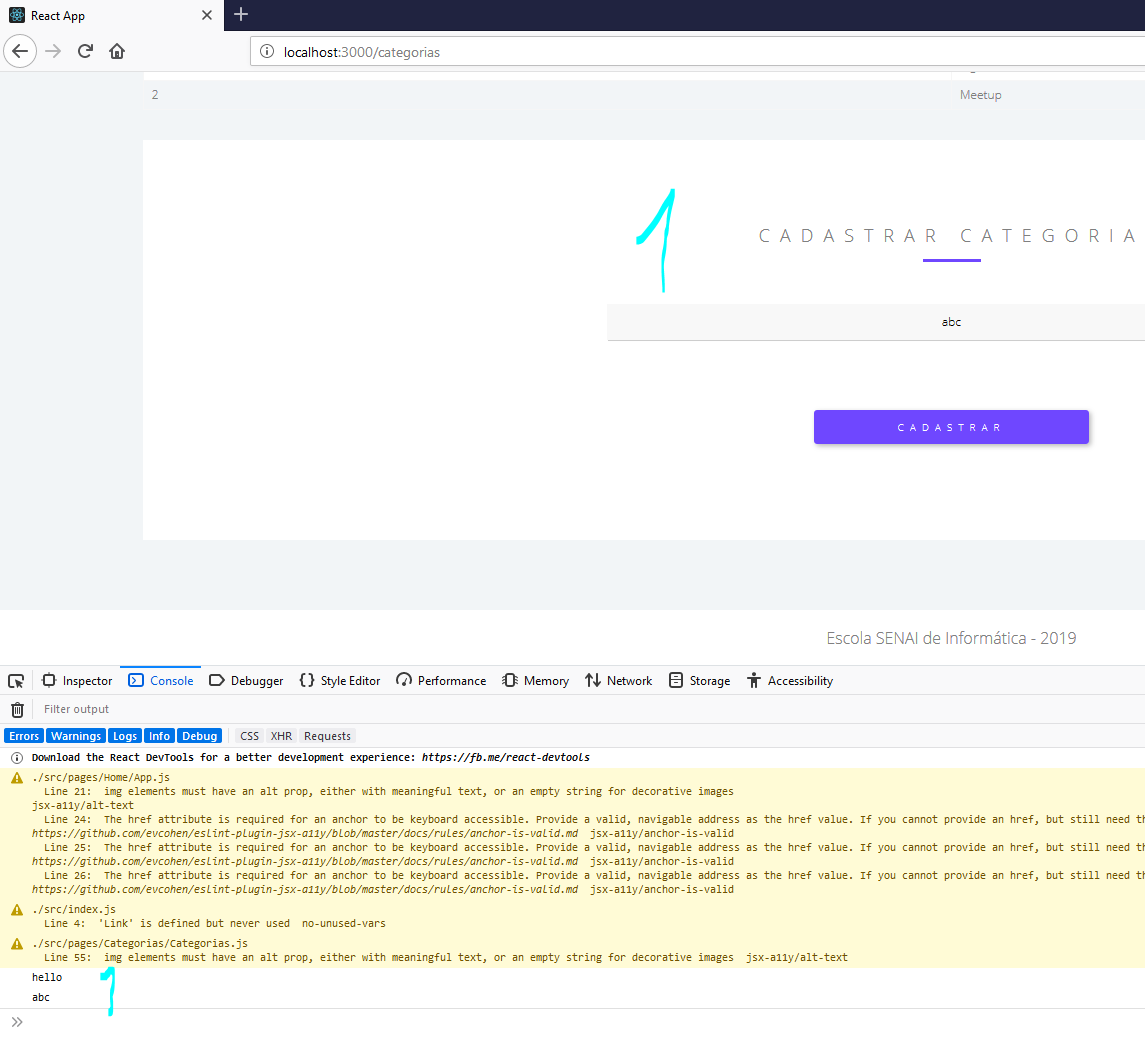
Alterar a construção da lista.



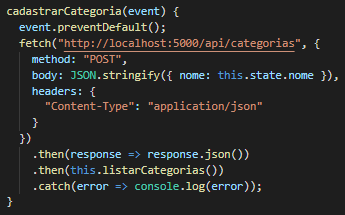
Cadastrar uma categoria



Alterar o formulário para ser chamado. Assim como o nome da Categoria e apenas imprimir o valor que o usuário incluiu ao clicar no botão.



Alterar a categoria para realizar a chamada a API.



Login

<https://blog.rocketseat.com.br/reactjs-autenticacao/>

Criar o arquivo Login.js e adicionar o conteúdo.

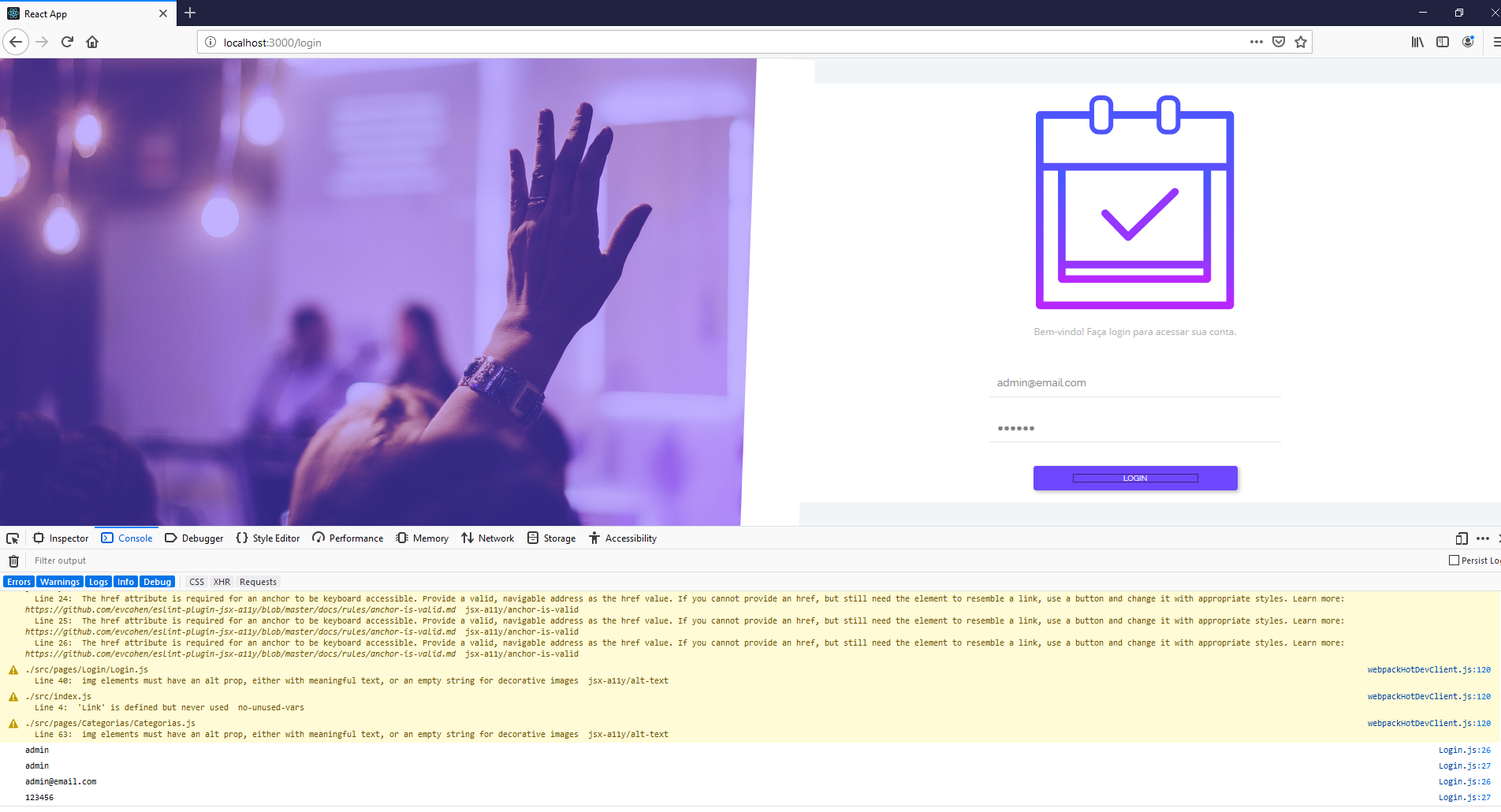


Adicionar nas rotas de navegação (index.js), o Login.



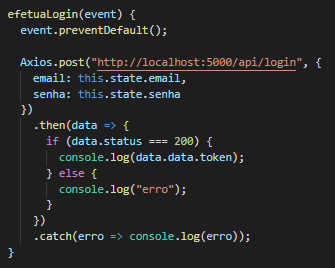
Falar sobre os escopos do CSS.

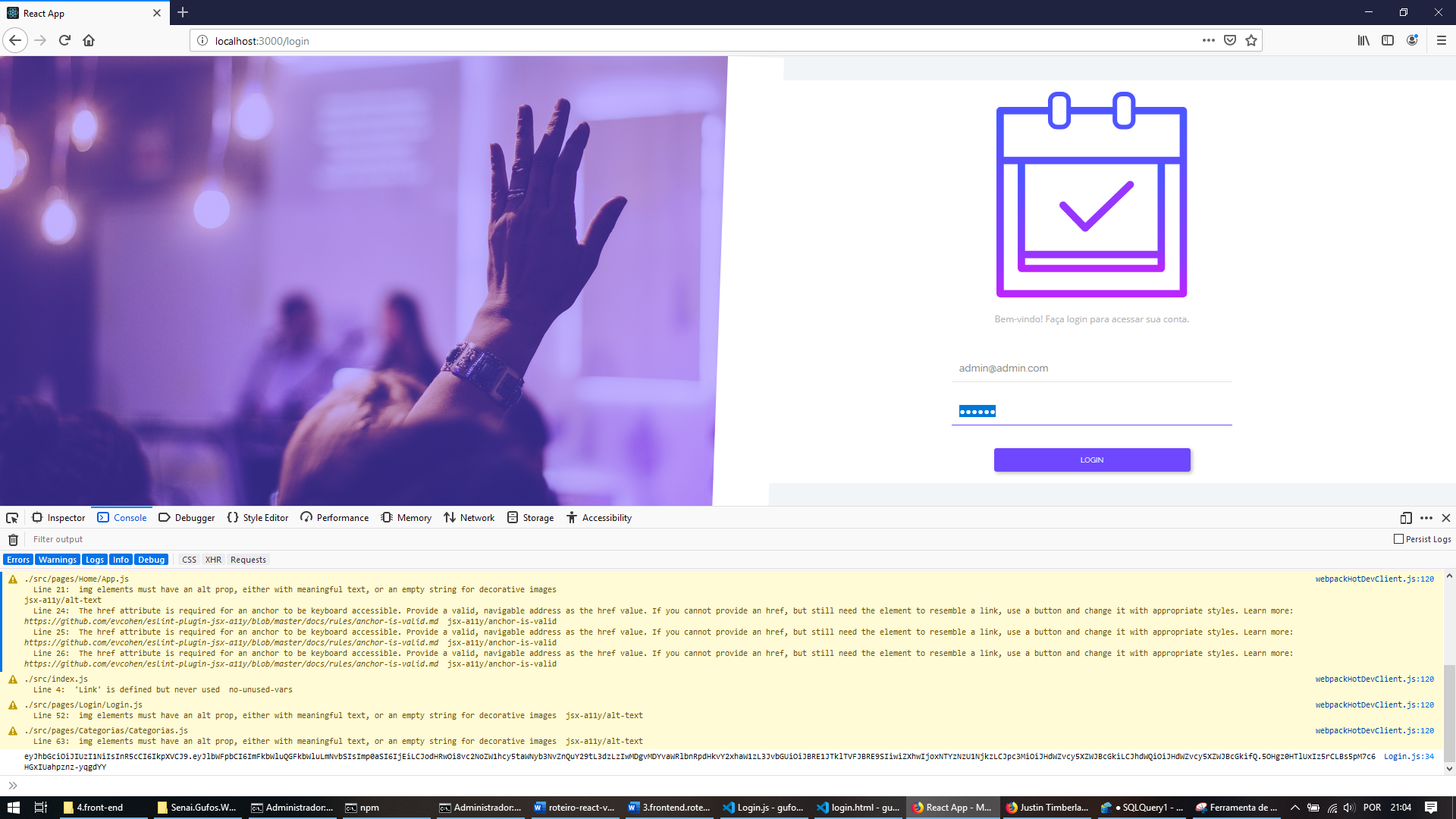




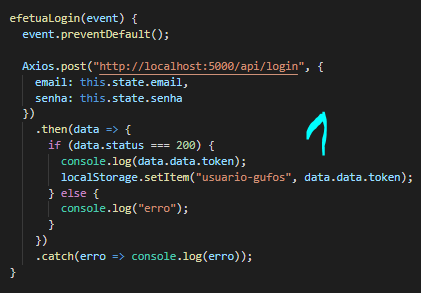
Mostrar outra dependência para realizar as chamadas as API’s.

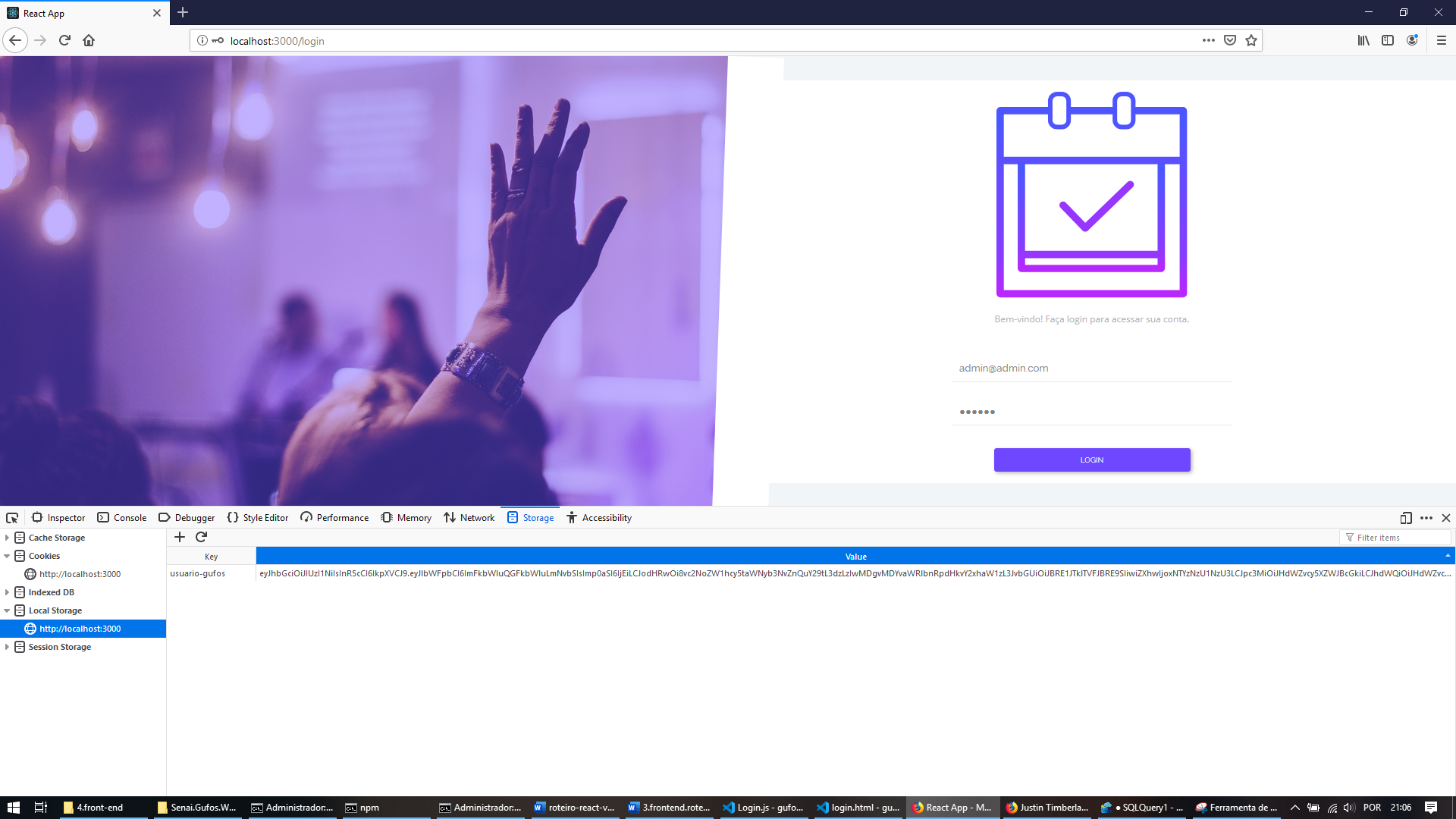
npm install --save axios



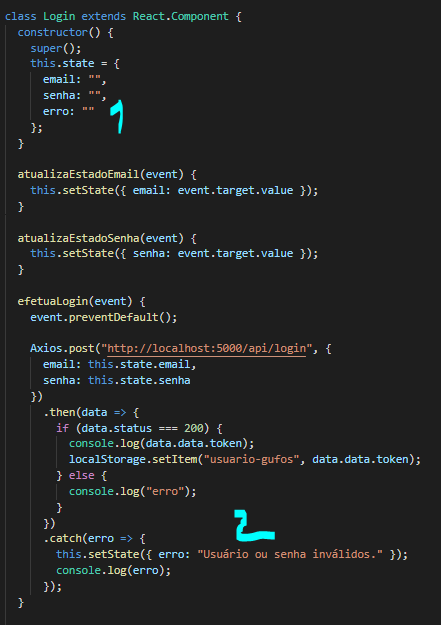


Incluir o localStorage



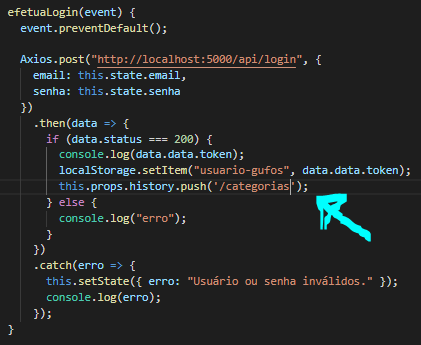


Mostrando uma mensagem de erro caso o usuário seja inválido.





Mas caso o usuário seja feito com sucesso, redirecionar.



No arquivo index.js, restringir para que apenas usuários autenticados, consigam acessar as informações privadas.



Alterar para que a categoria apareça apenas caso tenha o localStorage adicionado.

Eventos – Listar API

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 rota no index.js e importar os eventos.

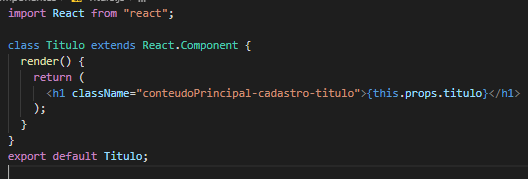




Props

Componente de Título

Criar um arquivo chamado Titulo.js dentro da pasta de Componentes.



E incluir o conteúdo acima.

Importar o título dentro do arquivo de Eventos.js e Categorias.js.

Eventos.js.



Categorias.js

