

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

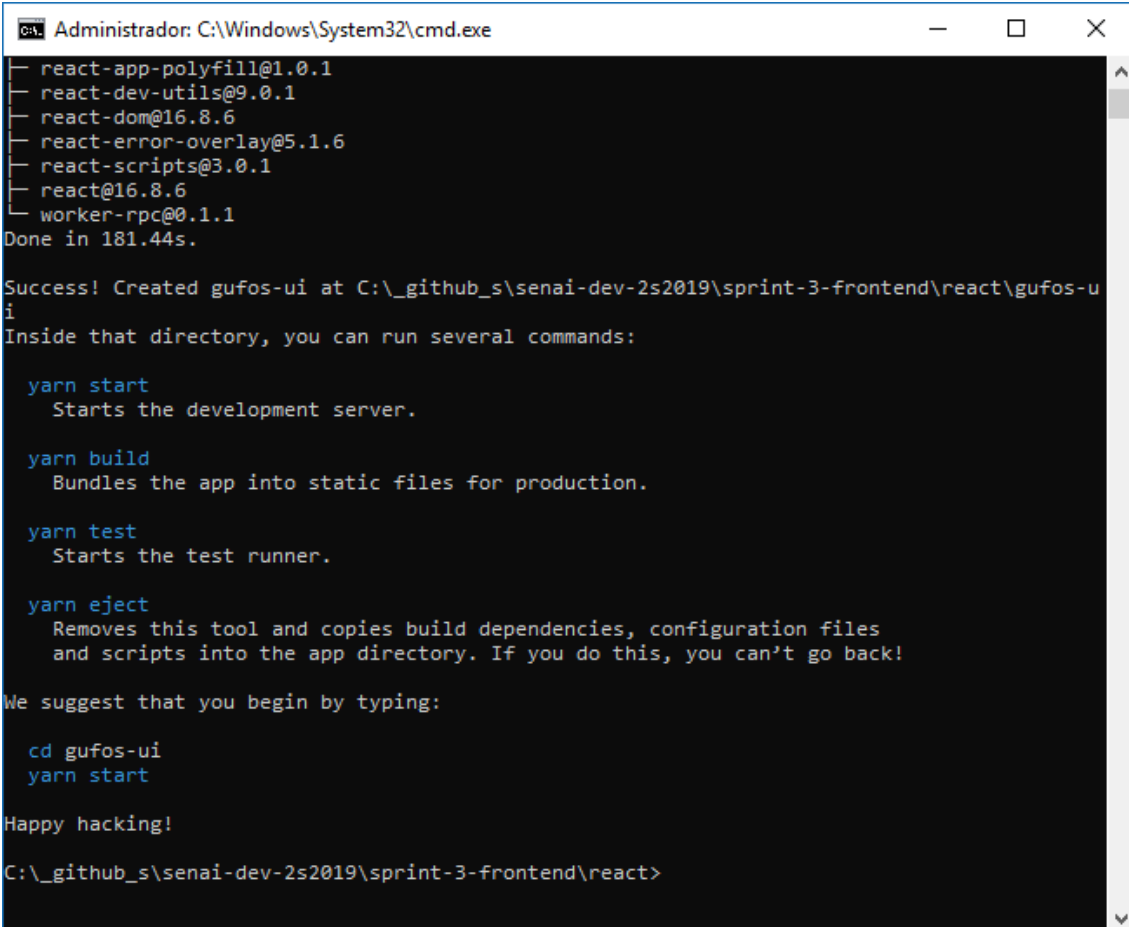
create-react-app gufos-ui

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

Executar Projeto

cd gufos-ui

npm start



```
Administrator: C:\Windows\System32\cmd.exe
- react-app-polyfill@1.0.1
- react-dev-utils@9.0.1
- react-dom@16.8.6
- react-error-overlay@5.1.6
- react-scripts@3.0.1
- react@16.8.6
- worker-rpc@0.1.1
Done in 181.44s.

Success! Created gufos-ui at C:\_github_s\senai-dev-2s2019\sprint-3-frontend\react\gufos-u
i
Inside that directory, you can run several commands:

  yarn start
    Starts the development server.

  yarn build
    Bundles the app into static files for production.

  yarn test
    Starts the test runner.

  yarn eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd gufos-ui
  yarn start

Happy hacking!

C:\_github_s\senai-dev-2s2019\sprint-3-frontend\react>
```

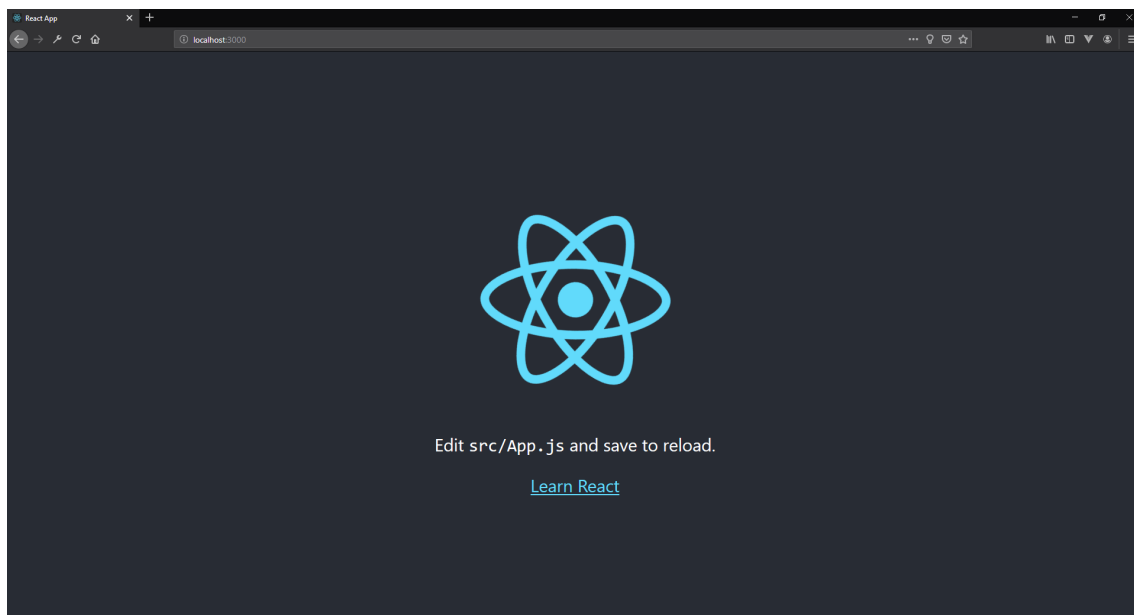
localhost:3000

```
npm
Compiled successfully!

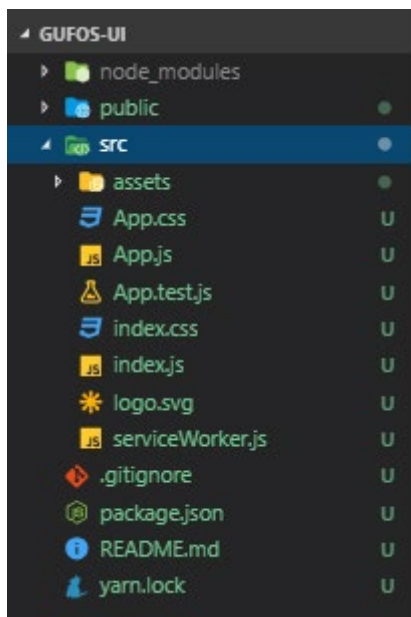
You can now view gufos-ui in the browser.

  Local:            http://localhost:3000/
  On Your Network:  http://192.168.56.1:3000/

Note that the development build is not optimized.
To create a production build, use yarn build.
```



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



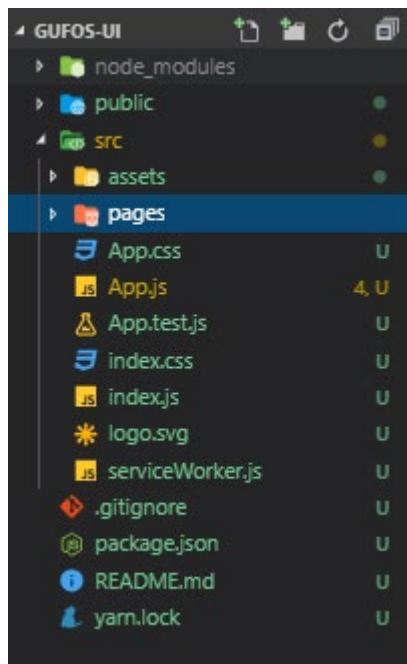
Copiar todo o conteúdo de home.html e colar em App.js.

Alterar os imports do css e das imagens.

Alterar tudo o que for class para className.

```
1 import React from 'react';
2 // import logo from './logo.svg';
3 // import './App.css';
4
5 // realizar os imports dos estilos
6 import './assets/css/flexbox.css';
7 import './assets/css/reset.css';
8 import './assets/css/style.css';
9
10 // import do ícone
11 import logo from './assets/img/icon-login.png'
12
13 function App() {
14   return (
15     <div>
16       <header className="cabecalhoPrincipal">
17         <div className="container">
18           { /* <img src='./assets/img/icon-login.png' /> */ }
19           <img src={logo} />
20
21           <nav className="cabecalhoPrincipal-nav">
22             <a>Home</a>
23             <a>Eventos</a>
24             <a>Contato</a>
25             <a className="cabecalhoPrincipal-nav-login" href="login.html">Login</a>
26           </nav>
27         </div>
28       </header>
29
30       <section className="conteudoImagem">
31         <div>
32           <h1>Gufos</h1>
33           (property) JSX.IntrinsicElements.section: React.DetailedHTMLProps<React
34             .HTMLAttributes<HTMLElement>, HTMLElement>
35         </section>
36
37       <main className="conteudoPrincipal">
38         <section id="conteudoPrincipal-eventos">
39           <h1 id="conteudoPrincipal-eventos-titulo">Opiniões</h1>
40           <div className="container">
41             <nav>
42               <ul className="conteudoPrincipal-dados">
43                 <li className="conteudoPrincipal-dados-link eventos">
44                   <h2>Pessoa A</h2>
45                   <p>
46                     Breve descrição sobre a opinião A. Lorem ipsum lorem ipsum
47                     lorem ipsum lorem ipsum lorem ipsum.
48                   </p>
49                 </li>
50
51                 <li className="conteudoPrincipal-dados-link eventos">
52                   <h2>Pessoa B</h2>
53                   <p>
54                     Breve descrição sobre a opinião B. Lorem ipsum lorem ipsum
55                     lorem ipsum lorem ipsum lorem ipsum.
56                   </p>
57                 </li>
58               </ul>
59             </nav>
60           </div>
61         </section>
62       </main>
63     </div>
64   );
65 }
```

Criar uma pasta chamada pages para melhor organização.



Criar pastas dentro de pages para organizar o nosso conteúdo.

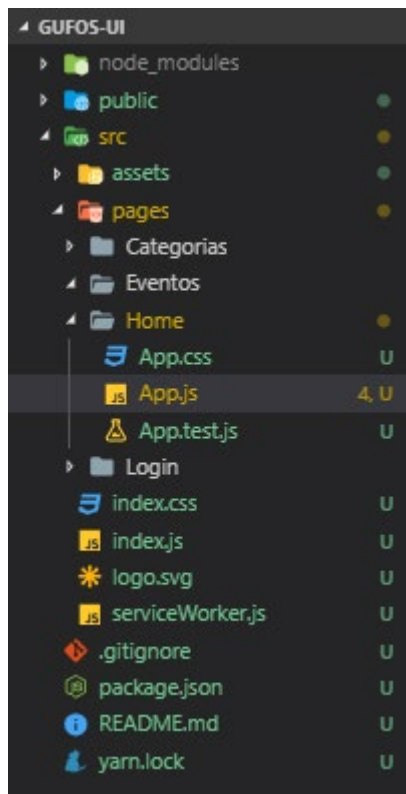
Eventos

Categorias

Home

Login

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



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

```
src > index.js
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import './index.css';
4  import App from './pages/Home/App';
5  import * as serviceWorker from './serviceWorker';
6
7  ReactDOM.render(<App />, document.getElementById('root'));
8
9  // If you want your app to work offline and load faster, you can change
10 // unregister() to register() below. Note this comes with some pitfalls.
11 // Learn more about service workers: https://bit.ly/CRA-PWA
12 serviceWorker.unregister();
13
```

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

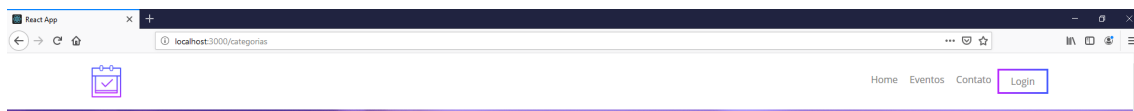
```
App.js x index.js
src > pages > Home > App.js > App
1 import React from 'react';
2 // import logo from './logo.svg';
3 // import './App.css';
4
5 // realizar os imports dos estilos
6 import '../assets/css/flexbox.css';
7 import '../assets/css/reset.css';
8 import '../assets/css/style.css';
9
10 // import do ícone
11 import logo from '../assets/img/icon-login.png'
12
13 function App() {
14   return (
15     <div>
16       <header className="cabecalhoPrincipal">
17         <div className="container">
18           { /*  */ }
19           <img src={logo} />
20         </div>
21       </header>
22     </div>
1
```

Vamos começar com a parte de Categorias dos eventos.

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

```
Categorias.js x
src > pages > Categorias > Categorias.js > ...
1 import React, { Component } from "react";
2
3 class Categorias extends Component {
4   render() {
5     return <div>Categorias</div>;
6   }
7 }
8
9 export default Categorias;
```

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
```

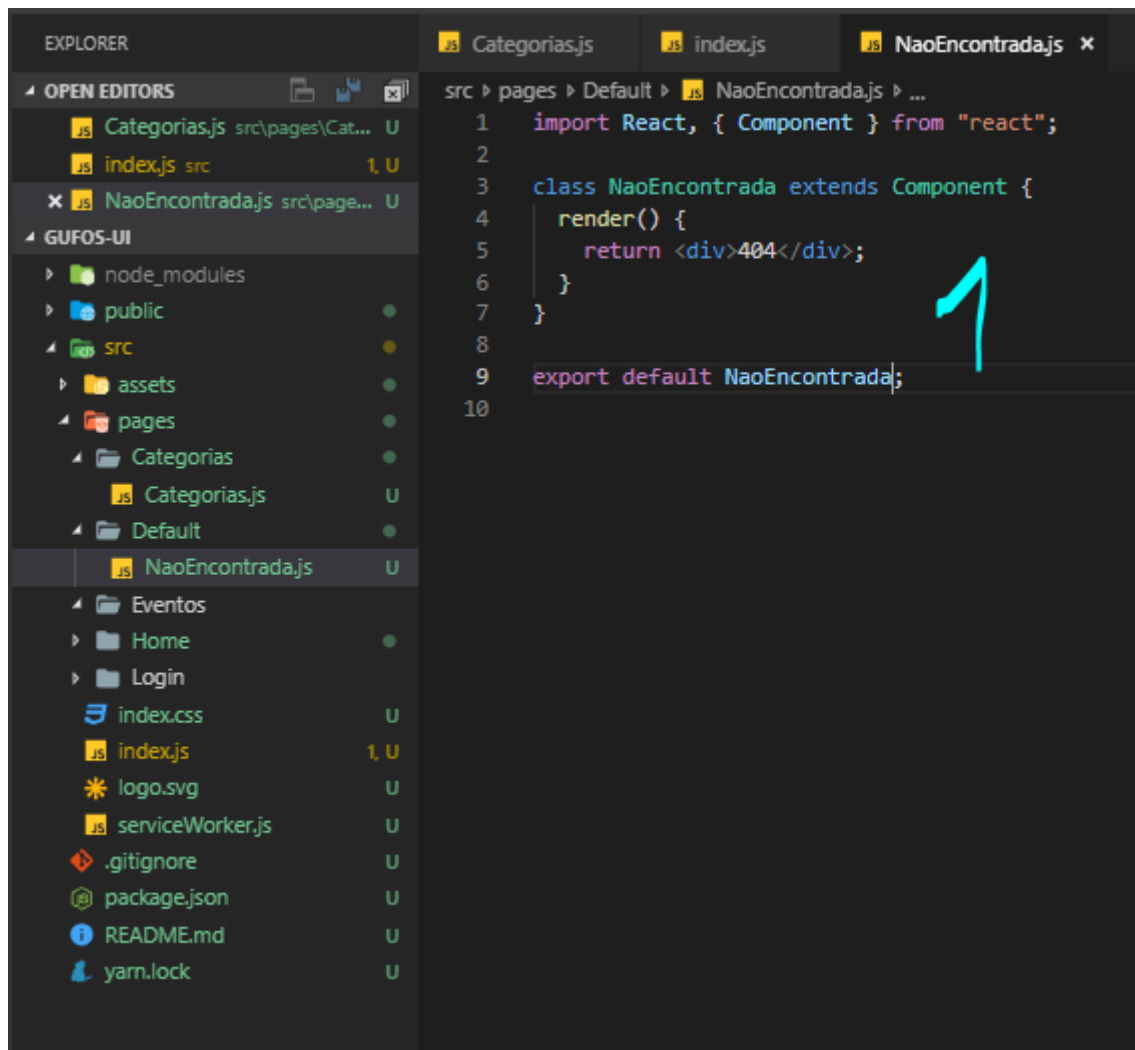
Após ter instalado a dependência acima, altera o index.js para realizar a configuração.

```

src > index.js > ...
1  import React from "react";
2  import ReactDOM from "react-dom";
3  // navegação
4  import { Route, Link, BrowserRouter as Router } from "react-router-dom";
5
6  import "../index.css";
7  import App from "../pages/Home/App";
8  // incluir página
9  import Categorias from "../pages/Categorias/Categorias";
10
11 import * as serviceWorker from "../serviceWorker";
12
13 const routing = (
14   <Router>
15     <div>
16       <Route exact path="/" component={App} />
17       <Route path="/categorias" component={Categorias} />
18     </div>
19   </Router>
20 );
21
22 // ReactDOM.render(<App />, document.getElementById("root"));
23 ReactDOM.render(routing, document.getElementById("root"));
24
25 // If you want your app to work offline and load faster, you can change
26 // unregister() to register() below. Note this comes with some pitfalls.
27 // Learn more about service workers: https://bit.ly/CRA-PWA
28 serviceWorker.unregister();
29

```

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



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

```
Categorias.js  index.js  x  NaoEncontrada.js

src > index.js > ...
1  import React from "react";
2  import ReactDOM from "react-dom";
3  // navegação
4  import { Route, Link, BrowserRouter as Router, Switch } from "react-router-dom";
5
6  import "./index.css";
7  import App from "../pages/Home/App";
8  // incluir página
9  import Categorias from "../pages/Categorias/Categorias";
10 import NaoEncontrada from "../pages/Default/NaoEncontrada";
11
12 import * as serviceWorker from "../serviceWorker";
13
14 const routing = (
15   <Router>
16     <div>
17       <Switch>
18         <Route exact path="/" component={App} />
19         <Route path="/categorias" component={Categorias} />
20         <Route component={NaoEncontrada} />
21       </Switch>
22     </div>
23   </Router>
24 );
25
26 // ReactDOM.render(<App />, document.getElementById("root"));
27 ReactDOM.render(routing, document.getElementById("root"));
28
29 // If you want your app to work offline and load faster, you can change
30 // unregister() to register() below. Note this comes with some pitfalls.
31 // Learn more about service workers: https://bit.ly/CRA-PWA
32 serviceWorker.unregister();
33
```

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

src > pages > Categorias > Categorias.js > Categorias > render

```

1  import React, { Component } from "react";
2
3  // realizar os imports dos estilos
4  import "../../assets/css/flexbox.css";
5  import "../../assets/css/reset.css";
6  import "../../assets/css/style.css";
7
8  // import do ícone
9  import logo from "../../assets/img/icon-login.png";
10
11 class Categorias extends Component {
12   render() {
13     return (
14       <div>
15         <header class="cabecalhoPrincipal">
16           <div class="container">
17             <img src={logo} />
18           </div>
19         </header>
20
21         <main class="conteudoPrincipal">
22           <section class="conteudoPrincipal-cadastro">
23             <h1 class="conteudoPrincipal-cadastro-titulo">Categorias</h1>
24             <div class="container" id="conteudoPrincipal-lista">
25               <table id="tabela-lista">
26                 <thead>
27                   <tr>
28                     <th>#</th>
29                     <th>Título</th>
30                   </tr>
31                 </thead>
32
33                 <tbody id="tabela-lista-corpo" />
34               </table>
35             </div>
36
37             <div class="container" id="conteudoPrincipal-cadastro">
38               <h2 class="conteudoPrincipal-cadastro-titulo">
39                 Cadastrar Categoria
40               </h2>
41               <form>
42                 <div class="container">
43                   <input
44                     type="text"
45                     class="class__categoria"
46                     id="input__categoria"
47                     placeholder="tipo do evento"
48                   />
49                   <button
50                     id="btn__cadastrar"
51                     class="conteudoPrincipal-btn conteudoPrincipal-btn-cadastro"
52                   >
53                     Cadastrar
54                   </button>
55                 </div>
56               </form>
57             </div>
58           </section>
59         </main>
60

```

Alterar tudo de class para className.

```
class Categorias extends Component {
  render() {
    return (
      <div>
        <header className="cabecalhoPrincipal">
          <div className="container">
            <img src={logo} />
          </div>
        </header>

        <main className="conteudoPrincipal">
          <section className="conteudoPrincipal-cadastro">
            <h1 className="conteudoPrincipal-cadastro-titulo">Categorias</h1>
            <div className="container" id="conteudoPrincipal-lista">
              <table id="tabela-lista">
                <thead>
                  <tr>
                    <th>#</th>
                    <th>Título</th>
                  </tr>
                </thead>

                <tbody id="tabela-lista-corpo" />
              </table>
            </div>

            <div className="container" id="conteudoPrincipal-cadastro">
              <h2 className="conteudoPrincipal-cadastro-titulo">
                Cadastrar Categoria
              </h2>
              <form>
                <div className="container">
                  <input
                    type="text"
                    className="className_categoria"
                    id="input_categoria"
                    placeholder="tipo do evento"
                  />
                </div>
              </form>
            </div>
          </section>
        </main>
      </div>
    );
  }
}
```

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.

The screenshot shows the VS Code interface. On the left, the Explorer sidebar displays the project structure. A red arrow points to the 'Rodape.js' file within the 'src > componentes' directory. The main editor area shows the code for 'Rodape.js', which is a React component. A red '2' is placed next to the closing tag of the component function.

```
1 import React from "react";
2
3 function Rodape() {
4   return (
5     <footer className="rodapePrincipal">
6       <section className="rodapePrincipal-patrocinadores">
7         <div className="container">
8           <p>Escola SENAI de Informática - 2019</p>
9         </div>
10      </section>
11    </footer>
12  );
13 }
14
15 export default Rodape;
```

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

The screenshot shows the VS Code interface with the 'App.js' file open. The code shows various imports for React, CSS files, and the 'Rodape' component. A red arrow points to the import statement for 'Rodape'.

```
1 import React from "react";
2 // import logo from './logo.svg';
3 // import './App.css';
4
5 // realizar os imports dos estilos
6 import "../../assets/css/flexbox.css";
7 import "../../assets/css/reset.css";
8 import "../../assets/css/style.css";
9
10 // import do ícone
11 import logo from "../../assets/img/icon-login.png";
12
13 import Rodape from "../../componentes/Rodape";
14
15 function App() {
16   return (
```

Incluí-lo no corpo.

A close-up of the JSX in 'App.js' showing the closing tags for the main container. A red arrow points to the location where the 'Rodape' component is being added.

```
</div>
</section>
</main>
<Rodape />
</div>
```

Realizar a mesma alteração no arquivo Categorias.js.

```
Categories.js x Rodape.js App.js NaoEncontrada.js
src > pages > Categorias > Categories.js > render
1  import React, { Component } from "react";
2
3  // realizar os imports dos estilos
4  import "../../assets/css/flexbox.css";
5  import "../../assets/css/reset.css";
6  import "../../assets/css/style.css";
7
8  // import do ícone
9  import logo from "../../assets/img/icon-login.png";
10
11 // importar Rodape
12 import Rodape from "../../componentes/Rodape";
13
```

```
<Rodape />

{/* <footer className="rodapePrincipal">
  <section className="rodapePrincipal-patrocinadores">
    <div className="container">
      <p>Escola SENAI de Informática - 2019</p>
    </div>
  </section>
</footer> */}
```

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.

```

class Categorias extends Component {
  constructor() {
    super();
    this.state = {
      lista: [
        { idCategoria: 1, nome: "Design" },
        { idCategoria: 2, nome: "Jogos" },
        { idCategoria: 3, nome: "Meetup" }
      ]
    };
  }

  render() {
    return (
      <div>
        <header className="cabecalhoPrincipal">
          <div className="container">
            <img src={logo} />
          </div>
        </header>

        <main className="conteudoPrincipal">
          <section className="conteudoPrincipal-cadastro">
            <h1 className="conteudoPrincipal-cadastro-titulo">Categorias</h1>
            <div className="container" id="conteudoPrincipal-lista">
              <table id="tabela-lista">
                <thead>
                  <tr>
                    <th>#</th>
                    <th>Título</th>
                  </tr>
                </thead>

                <tbody id="tabela-lista-corpo">
                  {this.state.lista.map(function(categoria) {
                    return (
                      <tr key={categoria.idCategoria}>
                        <td>{categoria.idCategoria}</td>
                        <td>{categoria.nome}</td>
                      </tr>
                    );
                  })}
                </tbody>
              </table>
            </div>
          </section>
        </main>
      </div>
    );
  }
}

```

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.

```

class Categorias extends Component {
  constructor() {
    super();
    this.state = {
      lista: []
      // [
      //   { idCategoria: 1, nome: "Design" },
      //   { idCategoria: 2, nome: "Jogos" },
      //   { idCategoria: 3, nome: "Meetup" }
      // ]
    };
  }

  componentDidMount() {
    fetch("http://localhost:5000/api/categorias")
      .then(response => response.json())
      .then(data => this.setState({ lista: data }));
  }
}

```

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

Alterar a construção da lista.

```

listarCategorias() {
  fetch("http://localhost:5000/api/categorias")
    .then(response => response.json())
    .then(data => this.setState({ lista: data }));
}

componentDidMount() {
  this.listarCategorias();
}

```

Cadastrar uma categoria


```

class Categorias extends Component {
  constructor() {
    super();
    this.state = {
      lista: [],
      nome: ""
      // [
      //   { idCategoria: 1, nome: "Design" },
      //   { idCategoria: 2, nome: "Jogos" },
      //   { idCategoria: 3, nome: "Meetup" }
      // ]
    };

    this.cadastrarCategoria = this.cadastrarCategoria.bind(this);

    cadastrarCategoria(event) {
      event.preventDefault();
      console.log("hello");
      console.log(this.state.nome);
    }

    atualizaNome(event) {
      this.setState({ nome: event.target.value })
    }

    listarCategorias() {
      fetch("http://localhost:5000/api/categorias")
        .then(response => response.json())
        .then(data => this.setState({ lista: data }));
    }

    componentDidMount() { ...
    }

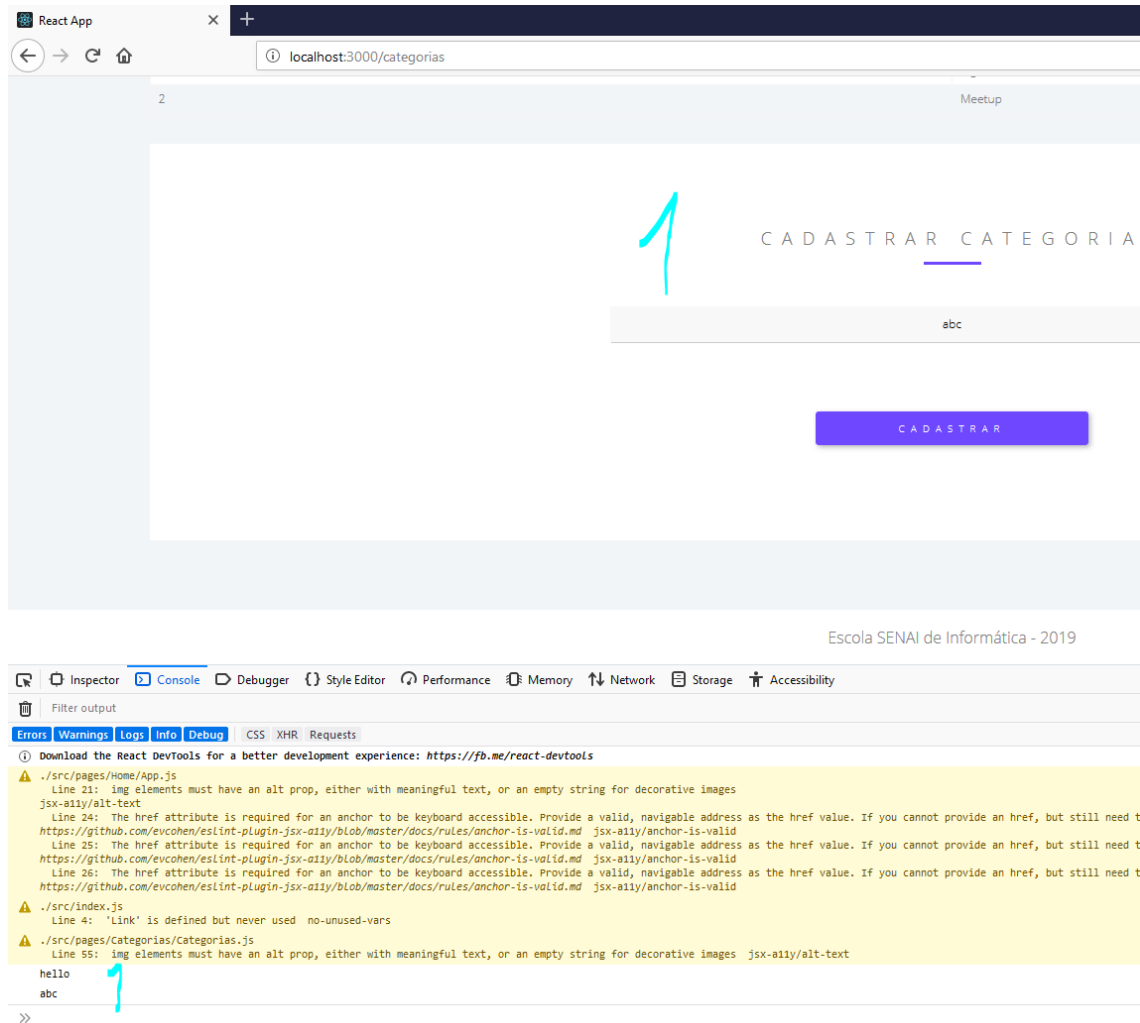
    render() {
      return (
        <div>
          <header className="cabecalhoPrincipal">
            <div className="container">
              <img src={logo} />
            </div>
          </header>

          <main className="conteudoPrincipal">
            <section className="conteudoPrincipal-cadastro">
              <h1 className="conteudoPrincipal-cadastro-titulo">Categorias</h1>
              <div className="container" id="conteudoPrincipal-lista">...

              <div className="container" id="conteudoPrincipal-cadastro">
                <h2 className="conteudoPrincipal-cadastro-titulo">
                  Cadastrar Categoria
                </h2>
                <form onSubmit={this.cadastrarCategoria}>
                  <div className="container">
                    <input
                      type="text"
                      value={this.state.nome}
                      onChange={this.atualizaNome.bind(this)}

```

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.

```
cadastrarCategoria(event) {  
  event.preventDefault();  
  fetch("http://localhost:5000/api/categorias", {  
    method: "POST",  
    body: JSON.stringify({ nome: this.state.nome }),  
    headers: {  
      "Content-Type": "application/json"  
    }  
  })  
  .then(response => response.json())  
  .then(this.listarCategorias())  
  .catch(error => console.log(error));  
}
```

Login

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

Criar o arquivo Login.js e adicionar o conteúdo.

```
import React from "react";

import logo from "../../assets/img/icon-login.png";

import "../../assets/css/login.css";

class Login extends React.Component {
  render() {
    return (
      <section class="container flex">
        <div class="img_login">
          <div class="img_overlay" />
        </div>

        <div class="item_login">
          <div class="row">
            <div class="item">
              <img src={logo} />
            </div>
            <div class="item" id="item_title">
              <p class="text_login" id="item_description">
                Bem-vindo! Faça login para acessar sua conta.
              </p>
            </div>
          </div>
          <form>
            <div class="item">
              <input
                class="input_login"
                placeholder="username"
                type="text"
                name="username"
                id="login_email"
              />
            </div>
            <div class="item">
              <input
                class="input_login"
                placeholder="password"
                type="password"
                name="password"
                id="login_password"
              />
            </div>
            <div class="item">
              <button class="btn btn_login" id="btn_login">
                Login
              </button>
            </div>
          </form>
        </div>
      </section>
    );
  }
}

export default Login;
```

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

```
index.js > ...
import React from "react";
import ReactDOM from "react-dom";
// navegação
import { Route, Link, BrowserRouter as Router, Switch } from "react-router-dom";

import "../index.css";
import App from "../pages/Home/App";
// incluir página
import Categorias from "../pages/Categorias/Categorias";

import Login from "../pages/Login/Login"; 1

import NaoEncontrada from "../pages/Default/NaoEncontrada";

import * as serviceWorker from "../serviceWorker";

const routing = (
  <Router>
    <div>
      <Switch>
        <Route exact path="/" component={App} />
        <Route path="/categorias" component={Categorias} />
        <Route path="/login" component={Login} />
        <Route component={NaoEncontrada} />
      </Switch>
    </div>
  </Router>
);

// ReactDOM.render(<App />, document.getElementById("root"));
ReactDOM.render(routing, document.getElementById("root"));

// If you want your app to work offline and load faster, you can change
// unregister() to register() below. Note this comes with some pitfalls.
// Learn more about service workers: https://bit.ly/CRA-PWA
serviceWorker.unregister();
```

Falar sobre os escopos do CSS.

```

class Login extends React.Component {
  constructor() {
    super();
    this.state = {
      email: "",
      senha: ""
    };
  }

  atualizaEstadoEmail(event) {
    this.setState({ email: event.target.value });
  }

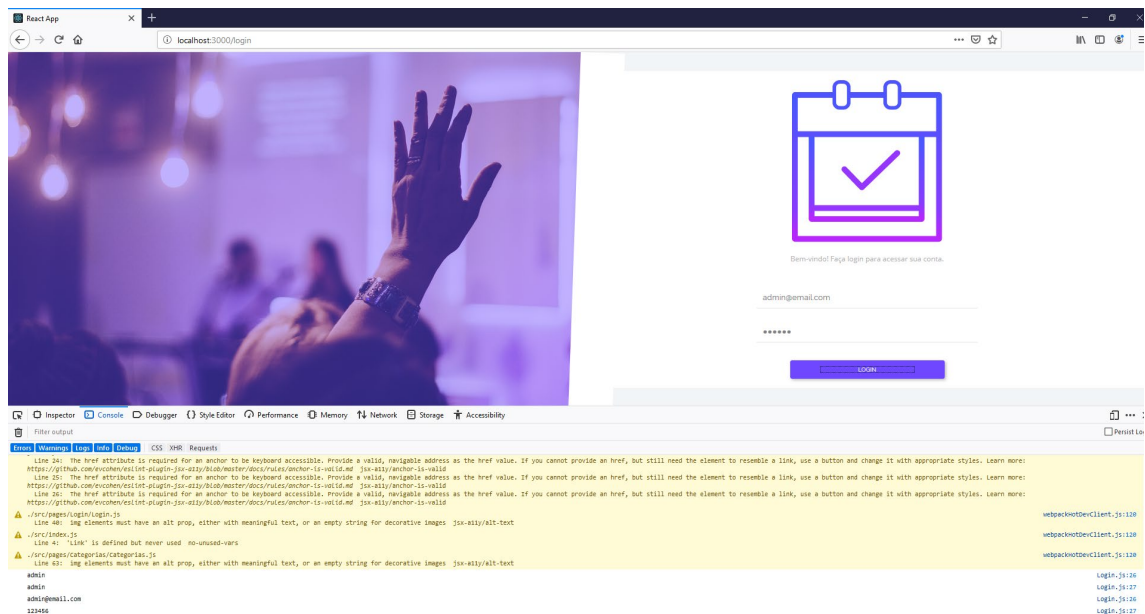
  atualizaEstadoSenha(event) {
    this.setState({ senha: event.target.value });
  }

  efetuaLogin(event) {
    event.preventDefault();
    console.log(this.state.email);
    console.log(this.state.senha);
  }

  render() {
    return (
      <section class="container flex">
        <div class="img_login">
          <div class="img_overlay" />
        </div>

        <div class="item_login">
          <div class="row">
            <div class="item">
              <img src={logo} />
            </div>
            <div class="item" id="item_title">
              <p class="text_login" id="item_description">
                Bem-vindo! Faça login para acessar sua conta.
              </p>
            </div>
            <form onSubmit={this.efetuaLogin.bind(this)}>
              <div class="item">
                <input
                  class="input_login"
                  placeholder="username"
                  type="text"
                  value={this.state.email}
                  name="username"
                  onChange={this.atualizaEstadoEmail.bind(this)}
                  id="login_email"
                />
              </div>
              <div class="item">
                <input
                  value={this.state.senha}
                  onChange={this.atualizaEstadoSenha.bind(this)}
                  class="input_login"
                  placeholder="password"
                  type="password"
                  name="password"
                />
              </div>
            </form>
          </div>
        </div>
      </section>
    );
  }
}

```



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

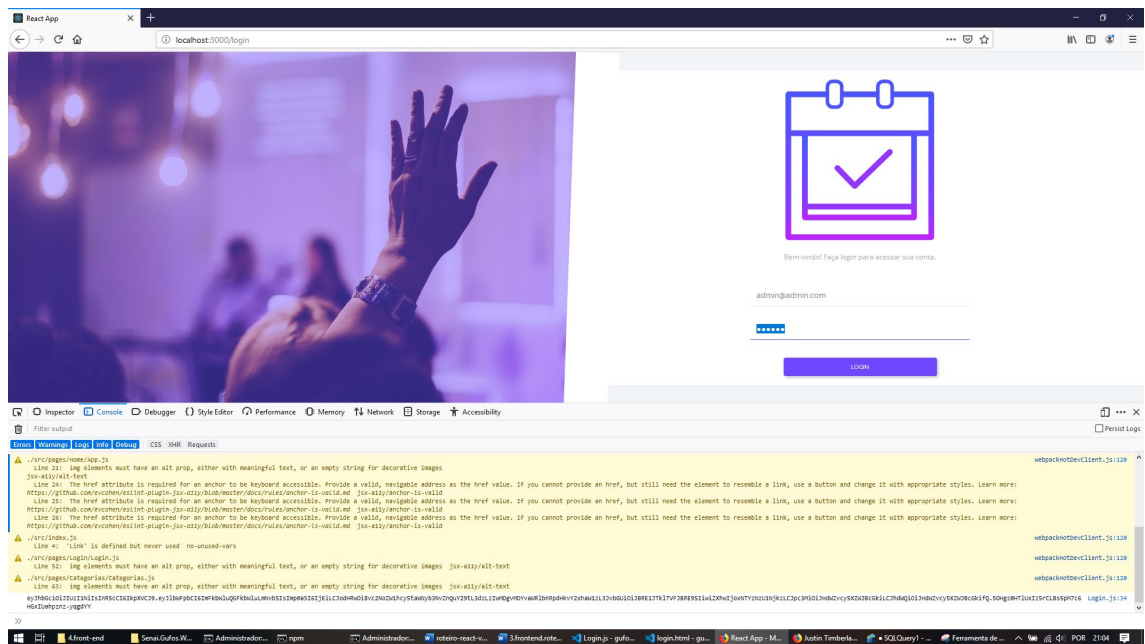
npm install --save axios

```

efetuaLogin(event) {
  event.preventDefault();

  Axios.post("http://localhost:5000/api/login", {
    email: this.state.email,
    senha: this.state.senha
  })
    .then(data => {
      if (data.status === 200) {
        console.log(data.data.token);
      } else {
        console.log("erro");
      }
    })
    .catch(erro => console.log(erro));
}

```



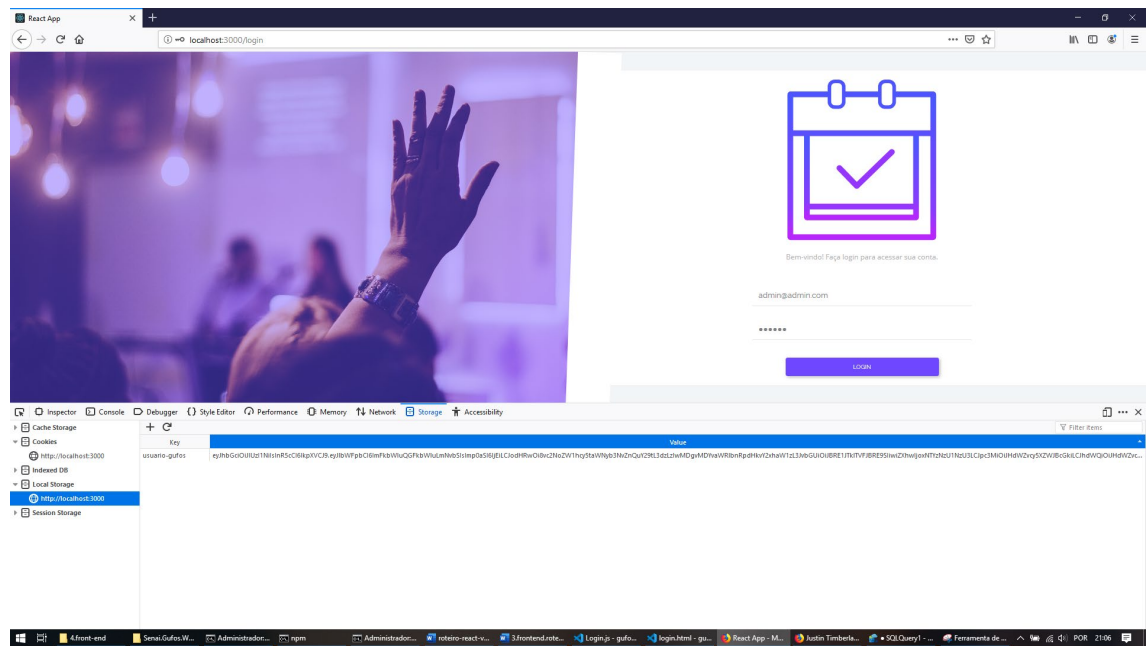
Incluir o localStorage

```

efetuaLogin(event) {
  event.preventDefault();

  Axios.post("http://localhost:5000/api/login", {
    email: this.state.email,
    senha: this.state.senha
  })
    .then(data => {
      if (data.status === 200) {
        console.log(data.data.token);
        localStorage.setItem("usuario-gufos", data.data.token);
      } else {
        console.log("erro");
      }
    })
    .catch(erro => console.log(erro));
}

```



Mostrando uma mensagem de erro caso o usuário seja inválido.


```
class Login extends React.Component {
  constructor() {
    super();
    this.state = {
      email: "",
      senha: "",
      erro: ""
    };
  }

  atualizaEstadoEmail(event) {
    this.setState({ email: event.target.value });
  }

  atualizaEstadoSenha(event) {
    this.setState({ senha: event.target.value });
  }

  efetuaLogin(event) {
    event.preventDefault();

    Axios.post("http://localhost:5000/api/login", {
      email: this.state.email,
      senha: this.state.senha
    })
      .then(data => {
        if (data.status === 200) {
          console.log(data.data.token);
          localStorage.setItem("usuario-gufos", data.data.token);
        } else {
          console.log("erro");
        }
      })
      .catch(erro => {
        this.setState({ erro: "Usuário ou senha inválidos." });
        console.log(erro);
      });
  }
}
```

```

<div class="item_login">
  <div class="row">
    <div class="item">
      <img src={logo} />
    </div>
    <div class="item" id="item_title">
      <p class="text_login" id="item_description">
        Bem-vindo! Faça login para acessar sua conta.
      </p>
    </div>
  </div>
  <form onSubmit={this.efetuaLogin.bind(this)}>
    <div class="item">
      <input
        class="input_login"
        placeholder="username"
        type="text"
        value={this.state.email}
        name="username"
        onChange={this.atualizaEstadoEmail.bind(this)}
        id="login_email"
      />
    </div>
    <div class="item">
      <input
        value={this.state.senha}
        onChange={this.atualizaEstadoSenha.bind(this)}
        class="input_login"
        placeholder="password"
        type="password"
        name="password"
        id="login_password"
      />
    </div>
    <p
      className="text_login"
      style={{ color: "red", textAlign: "center" }}
    >
      {this.state.erro}
    </p>
    <div class="item">
      <button class="btn btn_login" id="btn_login">
        Login
      </button>
    </div>
  </form>
</div>
</div>

```

Mas caso o usuário seja feito com sucesso, redirecionar.

```
efetuaLogin(event) {  
  event.preventDefault();  
  
  Axios.post("http://localhost:5000/api/login", {  
    email: this.state.email,  
    senha: this.state.senha  
  })  
    .then(data => {  
      if (data.status === 200) {  
        console.log(data.data.token);  
        localStorage.setItem("usuario-gufos", data.data.token);  
        this.props.history.push('/categorias');  
      } else {  
        console.log("erro");  
      }  
    })  
    .catch(erro => {  
      this.setState({ erro: "Usuário ou senha inválidos." });  
      console.log(erro);  
    });  
}
```

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

```

import React from "react";
import ReactDOM from "react-dom";
// navegação
import {
  Route,
  Link,
  BrowserRouter as Router,
  Switch,
  Redirect
} from "react-router-dom";

import "./index.css";
import App from "./pages/Home/App";
// incluir página
import Categorias from "./pages/Categorias/Categorias";

import Login from "./pages/Login/Login";

import NaoEncontrada from "./pages/Default/NaoEncontrada";

import * as serviceWorker from "./serviceWorker";

const RotaPrivada = ({ component: Component, ...rest }) => (
  <Route
    {...rest}
    render={props =>
      localStorage.getItem("usuario-gufos") !== null ? (
        <Component {...props} />
      ) : (
        <Redirect
          to={{ pathname: "/login", state: { from: props.location } }}
        />
      )
    }
  />
);

const routing = (
  <Router>
    <div>
      <Switch>
        <Route exact path="/" component={App} />
        { /* <Route path="/categorias" component={Categorias} /> */ }
        <RotaPrivada path="/categorias" component={Categorias} />
        <Route path="/login" component={Login} />
        <Route component={NaoEncontrada} />
      </Switch>
    </div>
  </Router>
);

```

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'.

```
import React, { Component } from "react";

// realizar os imports dos estilos
import "../../assets/css/flexbox.css";
import "../../assets/css/reset.css";
import "../../assets/css/style.css";

// import do ícone
import logo from "../../assets/img/icon-login.png";

// importar Rodape
import Rodape from "../../componentes/Rodape";

class Eventos extends Component {
  render() {
    return (
      <div>
        <header className="cabecalhoPrincipal">
          <div className="container">
            <img src={logo} />
          </div>
        </header>

        <main className="conteudoPrincipal">
          <section className="conteudoPrincipal-cadastro">
            <h1 className="conteudoPrincipal-cadastro-titulo">Eventos</h1>
            <div className="container" id="conteudoPrincipal-lista">
              <table id="tabela-lista">
                <thead>
                  <tr>
                    <th>#</th>
                    <th>Evento</th>
                    <th>Data</th>
                    <th>Acesso Livre</th>
                    <th>Tipo do Evento</th>
                  </tr>
                </thead>

                <tbody id="tabela-lista-corpo" />
              </table>
            </div>
          </section>
        </main>

        <Rodape />
      </div>
    );
  }
}

export default Eventos;
```

Criar rota no index.js e importar os eventos.

```
index.js > ...
import {
  Route,
  Link,
  BrowserRouter as Router,
  Switch,
  Redirect
} from "react-router-dom";

import "../index.css";
import App from "../pages/Home/App";
// incluir página
import Categorias from "../pages/Categorias/Categorias";
import Eventos from "../pages/Eventos/Eventos";

import Login from "../pages/Login/Login";

import NaoEncontrada from "../pages/Default/NaoEncontrada";

import * as serviceWorker from "../serviceWorker";

const RotaPrivada = ({ component: Component, ...rest }) => (
  <Route
    {...rest}
    render={props =>
      localStorage.getItem("usuario-gufos") !== null ? (
        <Component {...props} />
      ) : (
        <Redirect
          to={{ pathname: "/login", state: { from: props.location } }}
        />
      )
    }
  />
);

const routing = (
  <Router>
    <div>
      <Switch>
        <Route exact path="/" component={App} />
        { /* <Route path="/categorias" component={Categorias} /> */ }
        <RotaPrivada path="/categorias" component={Categorias} />
        <RotaPrivada path="/eventos" component={Eventos} />
        <Route path="/login" component={Login} />
        <Route component={NaoEncontrada} />
      </Switch>
    </div>
  </Router>
);

// ReactDOM.render(<App />, document.getElementById("root"));
ReactDOM.render(routing, document.getElementById("root"));

// If you want your app to work offline and load faster, you can change
// unregister() to register() below. Note this comes with some pitfalls.
// Learn more about service workers: https://bit.ly/CRA-PWA
serviceWorker.unregister();
```

```

class Eventos extends Component {
  constructor() {
    super();
    this.state = {
      lista: []
    };
  }

  componentDidMount() {
    Axios.get("http://localhost:5000/api/eventos")
      .then(data => this.setState({ lista: data.data }))
      .catch(erro => {
        console.log(erro);
      });
  }

  render() {
    return (
      <div>
        <header className="cabecalhoPrincipal">
          <div className="container">
            <img src={logo} />
          </div>
        </header>

        <main className="conteudoPrincipal">
          <section className="conteudoPrincipal-cadastro">
            <h1 className="conteudoPrincipal-cadastro-titulo">Eventos</h1>
            <div className="container" id="conteudoPrincipal-lista">
              <table id="tabela-lista">
                <thead>
                  <tr>
                    <th>#</th>
                    <th>Evento</th>
                    <th>Data</th>
                    <th>Ativo</th>
                    <th>Tipo do Evento</th>
                  </tr>
                </thead>

                <tbody id="tabela-lista-corpo">
                  {this.state.lista.map(evento => {
                    return (
                      <tr key={evento.idEvento}>
                        <td>{evento.idEvento}</td>
                        <td>{evento.titulo}</td>
                        <td>{evento.dataEvento}</td>
                        <td>{evento.ativo ? "Sim" : "Não"}</td>
                        <td>{evento.idCategoriaNavigation.nome}</td>
                      </tr>
                    );
                  })}
                </tbody>
              </table>
            </div>
          </section>
        </main>
      </div>
    );
  }
}

```

Props

Componente de Título

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

```
import React from "react";

class Titulo extends React.Component {
  render() {
    return (
      <h1 className="conteudoPrincipal-cadastro-titulo">{this.props.titulo}</h1>
    );
  }
}

export default Titulo;
```

E incluir o conteúdo acima.

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

Eventos.js.

```
// importar Rodape
import Rodape from "../../componentes/Rodape";
import Titulo from "../../componentes/Titulo";

import Axios from "axios";

class Eventos extends Component {
  constructor() {
    super();
    this.state = {
      lista: []
    };
  }

  componentDidMount() {
    Axios.get("http://localhost:5000/api/eventos")
      .then(data => this.setState({ lista: data.data }))
      .catch(erro => {
        console.log(erro);
      });
  }

  render() {
    return (
      <div>
        <header className="cabecalhoPrincipal">
          <div className="container">
            <img src={logo} />
          </div>
        </header>

        <main className="conteudoPrincipal">
          <section className="conteudoPrincipal-cadastro">
            /* <h1 className="conteudoPrincipal-cadastro-titulo">Eventos</h1> */
            <Titulo titulo="Eventos" />
            <div className="container" id="conteudoPrincipal-lista">
              <table id="table">
                <thead>
                  <tr>
                    <th>Nome</th>
                    <th>Data</th>
                    <th>Local</th>
                    <th>Categoria</th>
                    <th>Descrição</th>
                  </tr>
                </thead>
                <tbody>
                  <tr>
                    <td>Eventos</td>
                    <td>20/05/2020</td>
                    <td>Parque</td>
                    <td>Futebol</td>
                    <td>Futebol</td>
                  </tr>
                </tbody>
              </table>
            </div>
          </section>
        </main>
      </div>
    );
  }
}
```


Categorias.js

```
// import do ícone
import logo from "../../assets/img/icon-login.png";

// importar Rodape
import Rodape from "../../componentes/Rodape";

import Titulo from "../../componentes/Titulo";

class Categorias extends Component {
  constructor() {
    super();
    this.state = {
      lista: [],
      nome: ""
      // [
      //   { idCategoria: 1, nome: "Design" },
      //   { idCategoria: 2, nome: "Jogos" },
      //   { idCategoria: 3, nome: "Meetup" }
      // ]
    };

    this.cadastrarCategoria = this.cadastrarCategoria.bind(this);
  }

  cadastrarCategoria(event) { ...
  }

  atualizaNome(event) {
    this.setState({ nome: event.target.value });
  }

  listarCategorias() {
    fetch("http://localhost:5000/api/categorias")
      .then(response => response.json())
      .then(data => this.setState({ lista: data }));
  }

  componentDidMount() {
    this.listarCategorias();
  }

  render() {
    return (
      <div>
        <header className="cabecalhoPrincipal">
          <div className="container">
            <img src={logo} />
          </div>
        </header>

        <main className="conteudoPrincipal">
          <section className="conteudoPrincipal-cadastro">
            { /* <h1 className="conteudoPrincipal-cadastro-titulo">Categorias</h1> */ }
            <Titulo titulo="Teste" />
            <div className="container" id="conteudoPrincipal-lista">
              <table id="tabela-lista">
                <thead>

```