



GOVERNO DO ESTADO DE SÃO PAULO

PROGRAMAÇÃO WEB III APLICAÇÃO DE PÁGINA ÚNICA (SPA)

PROGRAMAÇÃO WEB III

APLICAÇÃO DE PÁGINA ÚNICA (SPA)



**PENÚLTIMA
TAREFA FOI A CRIAÇÃO DO
LOGIN E SENHA..**

The image shows a development environment with a code editor on the left and a web browser on the right. The code editor displays the `Form.js` file, which contains a `Form` component. The component has a `cadastroUsuario` function that prevents default behavior and logs the user name to the console. It uses `useState` to manage the `name` and `password` states. The component returns a JSX element with a heading `Meu Cadastro`, a form with two input fields (name and password), and a `Cadastrar` button. The web browser shows the rendered application at `localhost:3000/?`. The form is titled `Meu Cadastro` and has a `Ativar!` button. The input fields are labeled `Nome:` and `Senha:`. The `Nome` field contains the text `banana`. The `Senha` field contains three dots. The `Cadastrar` button is visible below the input fields. The browser's developer tools are open, showing the `Elements` panel with the `body` element selected. The `Console` panel shows two messages: `Usuário Form.js:6 teste foi cadastrado com a senha: 123` and `Usuário Form.js:6 banana foi cadastrado com a senha: 666`.

```
JS Form.js JS App.js
meu_projeto > src > components > JS Form.js > Form
2
3 function Form(){
4   function cadastroUsuario(event){
5     event.preventDefault()
6     console.log(`Usuário ${name} foi cadastrado com a senha: ${password}`)
7   }
8   const [name, setName] = useState()
9   const [password, setPassword] = useState()
10
11   return(
12     <div>
13       <h1>Meu Cadastro: </h1>
14       <form onSubmit={cadastroUsuario}>
15         <div>
16           <label htmlFor="name">Nome: </label>
17           <input
18             type="text"
19             id="name"
20             name="name"
21             placeholder="Digite o seu nome."
22             onChange={(e) => setName(e.target.value)}
23           />
24         </div>
25         <div>
26           <label htmlFor="password">Senha: </label>
27           <input
28             type="password"
29             id="password"
30             name="password"
31             placeholder="Digite a sua senha."
32             onChange={(e) => setPassword(e.target.value)}
33           />
34         </div>
35       </form>
36       <button type="button">Cadastrar</button>
37     </div>
38   )
39 }
```

React App
localhost:3000/?
Keep Closet — estu... LeoLabs Intelligent Diagram... Nova digitalização ...

Ativar!

Meu Cadastro:

Nome:
banana

Senha:
...

Cadastrar

Elementos >> 1
html body
Estilos Computado Layout Listener de eventos >>
▼ Grade
Sobrepôr configurações de exibição
Console Novidades Problemas X
top top Filtro
Todos os níveis 1 problema: 1
2 mensagens Usuário Form.js:6 teste foi cadastrado com a senha: 123
2 mensagens ... Nenhum erro
Nenhum aviso
2 informações Usuário Form.js:6 banana foi cadastrado com a senha: 666
Sem mensagem...



Métodos por propriedades:

- Os **métodos** também podem ser passados por props;
- Um componente filho pode ativar o método do seu ancestral;
- Acessando o método por meio de um evento;
- A sintaxe é a mesma de uma **props** de dados: **props.meuEvento**.

PROGRAMAÇÃO WEB III

APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Para recuperar ou passar este método, precisamos primeiro criar uma propriedade do componente de botão:

```
EXPLORADOR  ...  JS Form.js  JS Evento.js 1  JS App.js

PROJETO_REACT
├── meu_projeto
│   ├── node_modules
│   ├── public
│   └── src
│       ├── components
│       │   ├── JS Evento.js 1
│       │   ├── JS Form.js
│       │   ├── JS HelloWorld.js
│       │   ├── JS Item.js
│       │   ├── JS List.js
│       │   ├── JS MyName.js
│       │   ├── JS Pessoa.js
│       │   ├── JS PwebIII.js
│       └── PwebIII.module.css
└── ...

meu_projeto > src > components > JS Evento.js > Evento

1  function Evento({numero}){
2      function meuEvento(){
3          console.log(`Opsss, disparo ativo!!${numero}`)
4      }
5
6      return(
7          <div>
8              <p>Clique para disparar um Evento:</p>
9              <Button text="Primeiro Evento:"/>
10             <button onClick={meuEvento}> Ativar! </button>
11          </div>
12      )
13  }
14
15  export default Evento;
16
```

PROGRAMAÇÃO WEB III

APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Agora dentro dos componentes criamos um botão “Button.js”, porém como são pequenos comandos podemos simplificar o código, e podemos ir já separando os arquivos dos componentes para que não fiquem muito confusos criando pastas com seus nomes:

```
EXPLORADOR    ...    JS Evento.js 1    JS Button.js X

▼ PROJETO_REACT
  ▼ meu_projeto
    / public
  ▼ src
    ▼ components
      ▼ evento
        JS Button.js
        JS Evento.js 1
        JS Form.js

meu_projeto > src > components > evento > JS Button.js > [🔗] def
1  function Button(props){
2      return <button>{props.text}</button>
3
4  }
5  export default Button
```

PROGRAMAÇÃO WEB III

APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Voltando para o evento.js colocamos o import Button from “./evento/Button” e tiramos aquele button velho de onclick:

```
EXPLORADOR    ...    JS Evento.js    JS Button.js

v PROJETO_REACT
v meu_projeto
  public
  src
    components
      evento
        JS Button.js
        JS Evento.js
        JS Form.js
        JS HelloWorld.js
        JS Item.js
        JS List.js
        JS MyName.js
        JS Pessoa.js

meu_projeto > src > components > JS Evento.js > Evento
1  import Button from "./evento/Button";
2
3  function Evento({numero}){
4    function meuEvento(){
5      console.log(`Opsss, disparo ativo!!${numero}`)
6    }
7    return(
8      <div>
9        <p>Clique para disparar um Evento:</p>
10       <Button text="Primeiro Evento:"/>
11       <button onClick={meuEvento}> Ativar! </button>
12     </div>
13   )
}
```

PROGRAMAÇÃO WEB III

APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Modificamos o Button colocando um evento externo que vai ser chamado lá nas propriedades que criamos, através do onClick da própria propriedade do evento:

```
1 import Button from './evento/Button';
2
3 function Evento({numero}){
4   function meuEvento(){
5     console.log(`Opsss, disparo ativo!!${numero}`)
6   }
7   return(
8     <div>
9       <p>Clique para disparar um Evento:</p>
10      <Button event={meuEvento} text="Primeiro Evento:"/>
11    </div>
12  )
13 }
```

```
1 function Button(props){
2   return (
3     <button onClick={props.event}>{props.text}</button>
4   )
5 }
6 export default Button
```

PROGRAMAÇÃO WEB III

APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Vamos dar uma limpada no código, tirar este número de dentro do evento e o console e mudar o texto que vai ser exibido. no App.js apagamos dentro do evento o número também:

```
EXPLORADOR  ...  JS Evento.js  X  JS Button.js

PROJETO_REACT
meu_projeto
src
components
evento
JS Button.js
JS Evento.js
Form.js
HelloWorld.js
Item.js
List.js

meu_projeto > src > components > JS Evento.js > Evento
1  import Button from "../evento/Button";
2
3  function Evento(){
4      function meuEvento(){
5          console.log(`Ativar Primeiro Evento! `)
6      }
7      return (
8          <div>
9              <p>Clique para disparar um Evento:</p>
10             <Button event={meuEvento} text="Primeiro Evento:"/>
11         </div>
12     )
13 }
```

```
EXPLORADOR  ...  JS Evento.js  JS App.js  X  JS Button.js

PROJETO_REACT
meu_projeto
src
components
myName.js
JS Pessoa.js
JS PwebIII.js
# PwebIII.module.css
# App.css
JS App.js
JS App.test.js

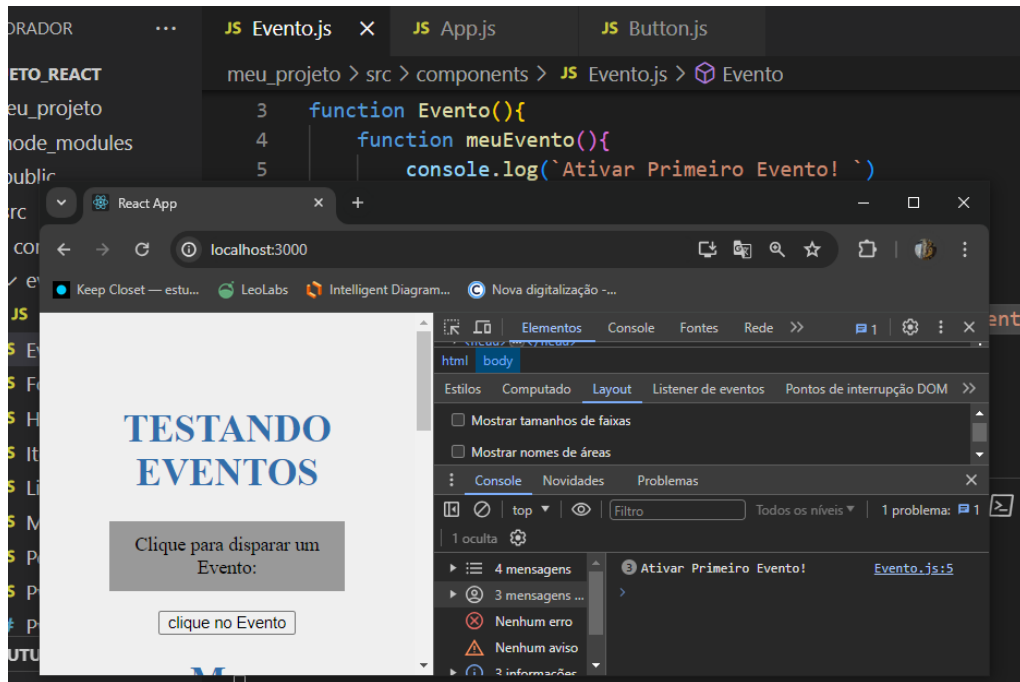
meu_projeto > src > JS App.js > App
1  import './App.css';
2  import Evento from '../components/Evento';
3  import Form from '../components/Form';
4
5  function App() {
6      return (
7          <div className="App">
8              <h1> TESTANDO EVENTOS </h1>
9              <Evento/>
10
11             <Form/>
12         </div>
13     )
14 }
```


PROGRAMAÇÃO WEB III

APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Verificando se deu certo, entrando no console vai aparecer toda vez que clicar “ativar Primeiro Evento, f12 (inspeciona elementos):



PROGRAMAÇÃO WEB III

APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Outro exemplo é reutilizar os mesmos componentes para disparar na mesma propriedade criadas outros eventos com informações distintas:

```
EXPLORADOR  ...  JS Evento.js  JS App.js  JS Button.js

v PROJETO_REACT
v meu_projeto
  > node_modules
  > public
  v src
    v components
      v evento
        JS Button.js
        JS Evento.js
        JS Form.js
        JS HelloWorld.js
        JS Item.js
        JS List.js
        JS MyName.js
        JS Pessoa.js
        JS PwebIII.js
        # PwebIII.module.css

meu_projeto > src > components > JS Evento.js > Evento
1  import Button from "../evento/Button";
2
3  function Evento(){
4    function primeiroEvento(){
5      console.log(`Ativar Primeiro Evento!`)
6    }
7    function segundoEvento(){
8      console.log(`Ativar Segundo Evento!`)
9    }
10   return(
11     <div>
12       <p>Clique para disparar um Evento:</p>
13       <Button event={primeiroEvento} text="Primeiro Evento"/>
14       <Button event={segundoEvento} text="Segundo Evento"/>
15     </div>
16   )
17
18 }
```

PROGRAMAÇÃO WEB III

APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Outro exemplo ativando o mesmo disparos do mesmo componente:

JS Evento.js

JS App.js

JS Button.js

meu_projeto > src > components > JS Evento.js > Evento

```
1 import Button from "../evento/Button";
2
3 function Evento(){
4   function primeiroEvento(){
5     console.log(`Ativar Primeiro Evento!`)
6   }
7   function segundoEvento(){
8     console.log(`Ativar Segundo Evento!`)
9   }
10  return(
11    <div>
12      <p>Clique para disparar um Evento:</p>
13      <Button event={primeiroEvento} text="Pri
14      <Button event={segundoEvento} text="Segu
15    </div>
16  )
17 }
18 }
```

React App

localhost:3000

Keep Closet — estu... LeoLabs Intelligent Diagram... Nova digitalização -...

TESTANDO
EVENTOS

Clique para disparar um
Evento:

Primeiro Evento

Segundo Evento

Elementos Console Fontes Rede

html body

Estilos Computado Layout Listener de eventos Pontos

☐ Mostrar tamanhos de faixas

☐ Mostrar nomes de áreas

Console Novidades Problemas

top Filtro Todos os n

1 oculta

10 mensagens

9 mensagens ...

Nenhum erro

Nenhum aviso

9 informações

Ativar Segundo Evento:

Ativar Primeiro Evento!

Ativar Segundo Evento!

Ativar Primeiro Evento!



Exemplos de **Renderização por Condições:**

Podemos submeter a exibição de alguns elementos a um **if else**, estas ações são conhecidas como renderização condicional, envolvendo as Tags em chaves { }, sabemos que dentro das chaves podemos executar os códigos **JavaScript**, assim criamos nossas condições podemos também utilizar os **states** para criar as condições.

PROGRAMAÇÃO WEB III

APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Vamos dentro da pasta componentes e criamos uma condicional.js em seguida importamos o state, criamos um pequeno formulário de email e montamos a função de retorno no console para testar :

```
EXPLORADOR    ...    JS App.js    JS Condicional.js 1 X

v PROJETO_REACT
v meu_projeto
  > node_modules
  > public
  v src
    v components
      JS Button.js
      JS Condicional.js 1
      JS Evento.js
      JS Form.js
      JS HelloWorld.js
      JS Item.js
      JS List.js
      JS MyName.js
      JS Pessoa.js
      JS PwebIII.js
    > ESTRUTURA DO CÓDIGO

meu_projeto > src > components > JS Condicional.js > Condicional > enviarEmail
1  import {useState} from 'react'
2  function Condicional(){
3
4      function enviarEmail(e){
5          e.preventDefault()
6          console.log('test email')
7      }
8
9      return (
10         <div>
11             <h2>Cadastre o Seu email:</h2>
12             <form>
13                 <input type="email" placeholder="Digite o seu E-mail:" />
14                 <button onClick={enviarEmail}>Enviar-email</button>
15             </form>
16         </div>
17     )
18 }
19 export default Condicional
```

PROGRAMAÇÃO WEB III

APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Feito isto limpamos todo o App.js e adicionamos os novos parâmetros além do import até ai tudo muito parecido com as outras formas de criar a estrutura:

The screenshot displays a code editor with a file explorer on the left and a browser preview on the right. The file explorer shows the project structure: PROJETO_REACT, meu_projeto, src, components, App.css, App.js, App.test.js, index.css, index.js, logo.svg, reportWebVitals.js, setupTests.js, .gitignore, package-lock.json, and package.json. The code editor shows the App.js file with the following code:

```
1 import './App.css';
2 import Conditional from './components/Condiciona
3
4 function App() {
5   return (
6     <div className="App">
7       <h1> Renderização Condicional </h1>
8       <Conditional/>
9     </div>
10  )
11 }
12 export default App;
13
```

The browser preview shows the rendered application with the title "Renderização Condicional" and a form to "Cadastre o Seu email:" with a text input field and an "Enviar-email" button.

PROGRAMAÇÃO WEB III

APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Agora vamos colocar umas constantes dentro da condicional, e setar a trajetória através do onChange já utilizando useState :

```
JS App.js  JS Condicional.js 1
1  import {useState} from 'react'
2  function Condicional(){
3
4      const [email, setEmail] = useState()
5
6
7      function enviarEmail(e){
8          e.preventDefault()
9          console.log('test email')
10 }
11
12 return (
13     <div>
14         <h2>Cadastre o Seu email:</h2>
15         <form>
16             <input type="email" placeholder="Digite o seu E-mail:"
17                 onChange={(e) => setEmail(e.target.value)} />
18
19             <button onClick={enviarEmail}>Enviar-email</button>
20         </form>
```

PROGRAMAÇÃO WEB III

APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Na sequência, colocamos mais uma `const [userEmail, setUserEmail]`, que vai servir como simulação na transição entre as variáveis de teste, é muito comum e usado para testes de validações por exemplo:

The image shows a development environment with a code editor on the left and a web browser on the right.

Code Editor (Left): Displays the file `JS Condicional.js` with the following code:

```
1 import {useState} from 'react'
2 function Condicional(){
3
4     const [email, setEmail] = useState()
5     const [userEmail, setUserEmail] = useState()
6
7     function enviarEmail(e){
8         e.preventDefault()
9         setUserEmail(email)
10        console.log(userEmail)
11    }
12
13    return (
14        <div>
15            <h2>Cadastre o Seu email:</h2>
16            <form>
17                <input type="email" placeholder="Digite o seu E-mail:"
18                    onChange={(e) => setEmail(e.target.value)} />
19
20                <button onClick={enviarEmail}>Enviar-email</button>
21            </form>
22        </div>
23    )
24 }
```

Web Browser (Right): Shows the rendered application at `localhost:3000`. The page title is "Renderização Condicional". The main heading is "Cadastre o Seu email:". Below the heading is a form with an input field containing the text "teste" and a button labeled "Enviar-email".

The browser's developer tools are open, showing the "Layout" tab. The "Grade" panel is visible, showing the page structure. The "Console" panel shows 1 message, and the "Problemas" panel shows 1 problem, both related to the file `Condicional.js:10`.

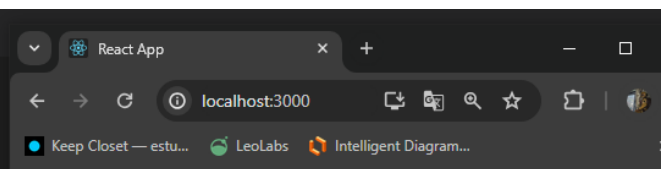
PROGRAMAÇÃO WEB III

APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Ainda para testar o valor, podemos mostrar direto no layout da tela as variáveis, para isto só inserir entre `{userEmail}` dentro das divs e remover o `console.log`:

```
JS App.js JS Condicional.js X
meu_projeto > src > components > JS Condicional.js > Condicional > enviarEmail
1 import {useState} from 'react'
2 function Condicional(){
3
4     const [email, setEmail] = useState()
5     const [userEmail, setUserEmail] = useState()
6
7     function enviarEmail(e){
8         e.preventDefault()
9         setUserEmail(email)
10    }
11    return (
12        <div>
13            <h2>Cadastre o Seu email:</h2>
14            <form>
15
16                <input type="email" placeholder="Digite o seu E-mail:"
17                    onChange={(e) => setEmail(e.target.value)} />
18
19                <button onClick={enviarEmail}>Enviar-email</button>
20                {userEmail}
21            </form>
22        </div>
23    )
```



Renderização Condicional

Cadastre o Seu email:

teste@teste Enviar-email teste@teste

PROGRAMAÇÃO WEB III

APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Brincando um pouco, com as condicionais, dentro do userEmail podemos colocar o `&&` e que vai comparar o userEmail e se tiver algo armazenado ele será exibido dentro da html:

```
function enviarEmail(e){  
}  
return (  
  <div>  
    <h2>Cadastre o Seu email:</h2>  
    <form>  
      <input type="email" placeholder="Digite o seu E-m"   
        onChange={(e) => setEmail(e.target.value)} />  
      <button onClick={enviarEmail}>Enviar-email</button>  
      {userEmail && (<div>  
        <p>O e-mail do usuário é: {userEmail}</p>  
      </div>)}  
    </form>  
  </div>  
)
```

Renderização Condicional

Cadastre o Seu email:

O e-mail do usuário é: teste@teste

PROGRAMAÇÃO WEB III

APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Para limpar a variável que foi visualizada criamos outra function e um botão onClick para limpar, deixamos o setUserEmail vazio que quando clicamos limpa o estado:

```
setUserEmail(email)
}
function limparEmail(){
  setUserEmail('')
}
return (
  <div>
    <h2>Cadastre o Seu email:</h2>
    <form>
      <input type="email" placeholder="Digite o seu E-mail:"
        onChange={(e) => setEmail(e.target.value)} />
      <button onClick={enviarEmail}>Enviar-email</button>
      {userEmail && (<div>
        <p>O e-mail do usuário é: {userEmail}</p>
        <button onClick={limparEmail}>Limpar e-mail</button>
      </div>)}
    </form>
  </div>
)
```

Renderização Condicional

Cadastre o Seu email:

O e-mail do usuário é: teste@teste



Renderizando listas:

Para renderizar uma lista vamos necessariamente precisar de um **array**, depois utilizamos a função **map**, serve para percorrer os itens assim renderizar o que é mostrado na tela, e podemos também unir operadores condicionais.

PROGRAMAÇÃO WEB III

APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Vamos dar uma limpada na tela e criar já a lista nas const meultens, depois entramos nos componentes e criamos Lista.js colocamos a função MAP e usamos a props:

```
meu_projeto > src > JS App.js > App
1  import './App.css';
2  import Lista from './components/Lista'
3
4  function App() {
5
6      const meusItens = ['React', 'Cordova', 'Monaca']
7
8      return (
9          <div className="App">
10             <h1> Renderização de Listas </h1>
11             <Lista itens = {meusItens}/>
12             </div>
13         )
14     }
15     export default App;
```

```
meu_projeto > src > components > JS Lista.js > Lista > itens.map
1  function Lista({itens}){
2      return(
3          <>
4              <h3>Lista do Ds:</h3>
5              {
6                  itens.map((item) => (
7                      <p> {item}</p>
8                  )))
9              </>
10          )
11      }
12      export default Lista
```

PROGRAMAÇÃO WEB III

APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Como podemos ver a lista foi exibida:

The image shows a code editor on the left and a web browser on the right. The code editor displays the `App.js` file with the following code:

```
meu_projeto > src > JS App.js > App
1 import './App.css';
2 import Lista from './components/Lista';
3
4 function App() {
5
6   const meusItens = ['React', 'Cordova', 'Monaca'];
7
8   return (
9     <div className="App">
10       <h1> Renderização de Listas
11       <Lista itens={meusItens} />
12     </div>
13   );
14 }
15 export default App;
```

The web browser on the right shows the rendered application. The title is "Renderização de Listas". Below the title, there is a heading "Lista do Ds:" followed by a list of three items: "React", "Cordova", and "Monaca". Each item is displayed in a gray rectangular box.

PROGRAMAÇÃO WEB III

APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Como podemos ver a lista foi exibida, porem ocorreu alguns erros ai que entra a bamiarra do react para resolver a lista prendendo ela nas chaves Keys do array:

The image shows a development environment with a code editor on the left and a web browser on the right.

Code Editor (Left): The file explorer shows the path `meu_projeto > src > components > JS Lista.js > Lista > itens.map()`. The code in `Lista.js` is as follows:

```
1 function Lista({itens}){
2   return(
3     <>
4       <h3>Lista do DS:</h3>
5       {
6         itens.map((item, index) => (
7           <p key={index}>{item}</p>
8         ))
9       }
10    </>
11  )
12 }
13 export default Lista
```

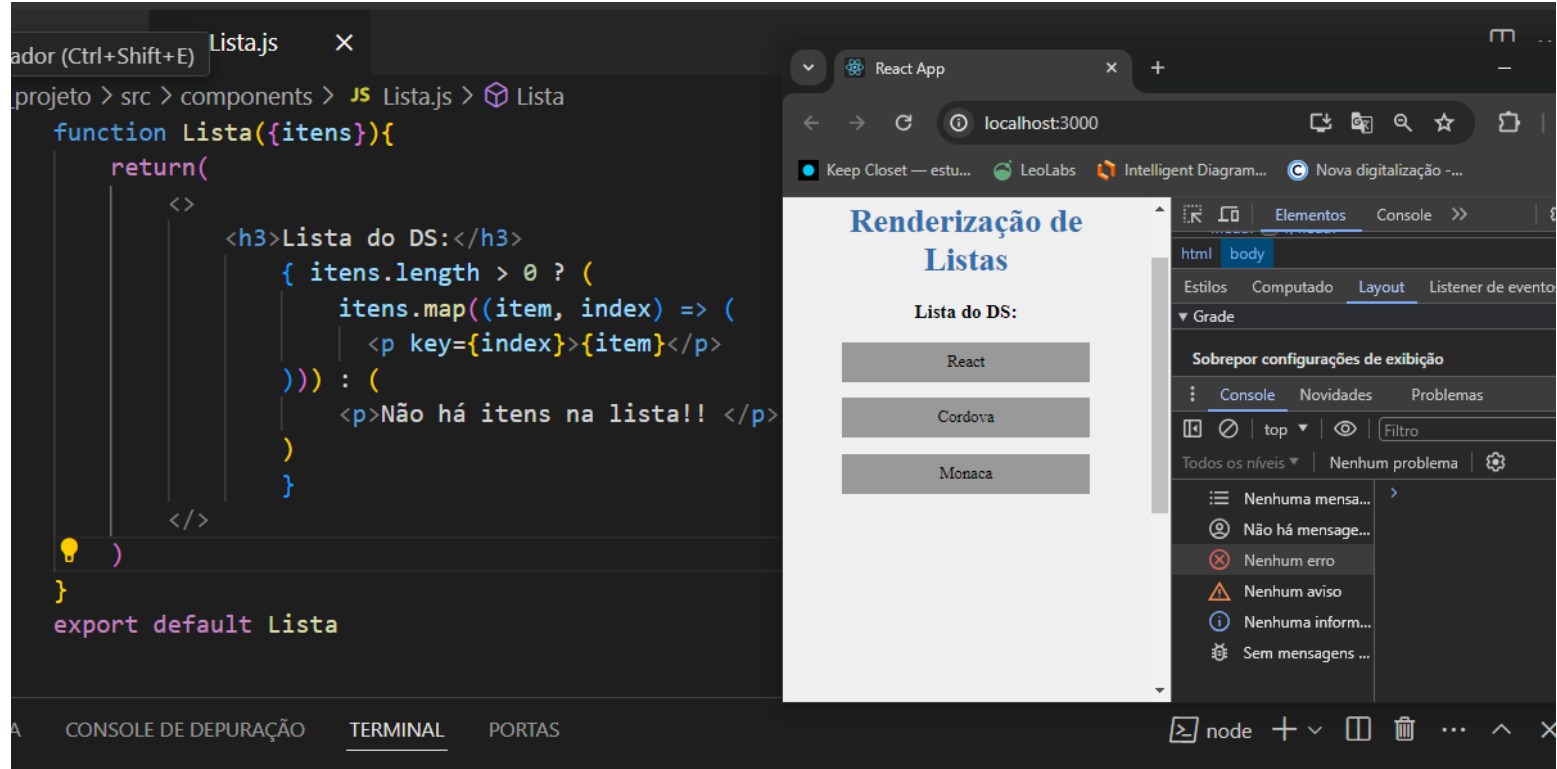
Web Browser (Right): The browser is running `localhost:3000` and displays the rendered application. The title is `Renderização de Listas`. The content shows `Lista do DS:` followed by a list of items: `React`, `Cordova`, and `Monaca`. The browser's developer tools are open, showing the `Elements` panel on the left and the `Console` panel on the right. The `Console` panel shows a message: `Nenhuma mensagem...`.

PROGRAMAÇÃO WEB III

APLICAÇÃO DE PÁGINA ÚNICA (SPA)



COLOCANDO CONDICIONAIS PODEMOS COLOCAR MENSAGENS PARA O USUÁRIO :





GOVERNO DO ESTADO DE SÃO PAULO