

IW-II

INTERFACE WEB II

Prof. Anderson Vanin

AULA 16 – EVENTOS REACT (onClick, onChange e onSubmit)

Eventos

- Os eventos de React são os mesmos eventos do DOM;
- Ou seja, temos eventos para responder a um click;
- O evento é atrelado a uma tag que irá executá-lo;
- Geralmente um método é atribuído ao evento;
- Este método deve ser criado no componente;

Configurações Iniciais

- Clonar o repositório do projeto
- Posicionar terminal na raiz do projeto
- Instalar as dependências do projeto: **npm i**
- Rodar o comando: **npm start**
- Verificar a execução da ultima aula.

Ajuste do App.js

```
JS App.js M X
primeiro-teste > src > JS App.js > [🔗] default
1  import './App.css';
2
3  function App() {
4      return (
5          <div className="App">
6              <h1>Testando Eventos</h1>
7          </div>
8      );
9  }
10
11  export default App;
```

Componente: Evento.js

The image shows a Visual Studio Code interface with the Explorer sidebar on the left and the Code Editor on the right. The Explorer sidebar shows a project structure under 'PROJETOS_REACT' with a folder 'primeiro-teste' containing 'src' and 'components'. The 'components' folder is expanded, showing files like 'DigaMeuNome.js', 'Evento.js' (highlighted with a red arrow), 'Frase.js', 'Frase.module.css', and 'HelloWorld.js'. The Code Editor shows the content of 'Evento.js' with the following code:

```
1 function Evento(){
2     return(
3         <div>
4
5         </div>
6     )
7 }
8
9 export default Evento
```

A red arrow points from the right side of the image to the JSX element in the code editor, specifically to the opening <div> tag on line 3.

Componente: Evento.js



The image shows a screenshot of a code editor (VS Code) with two panels. The left panel is the Explorer view, showing a project structure. The right panel is the Code Editor view, showing the code for the `Evento.js` component.

Explorer View (Left Panel):

- PROJECTOS_REACT
 - primeiro-teste
 - node_modules
 - public
 - src
 - components
 - JS DigaMeuNome.js**
 - JS Evento.js U** (highlighted with a red arrow)
 - JS Frase.js
 - # Frase.module.css
 - JS HelloWorld.js
 - # App.css
 - JS App.js M
 - # index.css
 - JS index.js

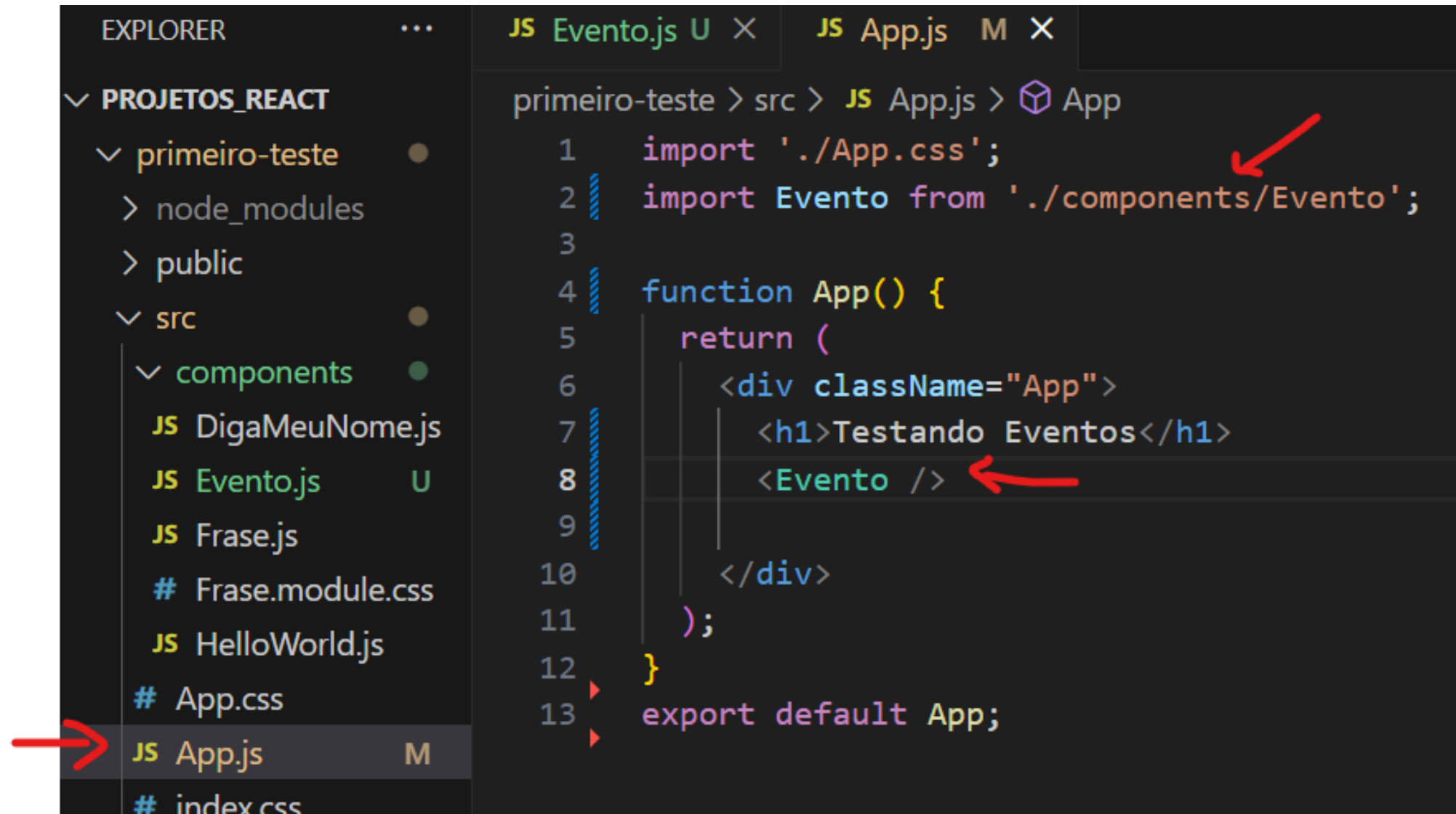
Code Editor View (Right Panel):

The code editor shows the file path: `primeiro-teste > src > components > JS Evento.js > Evento`. The code is as follows:

```
1 function Evento(){
2
3     function meuEvento(){
4         console.log('Opa, fui ativado!')
5     }
6
7     return(
8         <div>
9             <p>Clique aqui para disparar um evento</p>
10            <button onClick={meuEvento}>Ativar!</button>
11        </div>
12    )
13 }
14
15 export default Evento
```

A red arrow points to the `meuEvento` function definition on line 3.

Importar o componente



The image shows a VS Code editor with two panels. The left panel is the Explorer, showing a project structure for 'PROJETOS_REACT'. The right panel is the Editor, showing the code in 'App.js'.

Explorer Panel:

- PROJETOS_REACT
 - primeiro-teste
 - node_modules
 - public
 - src
 - components
 - DigaMeuNome.js
 - Evento.js U
 - Frase.js
 - Frase.module.css
 - HelloWorld.js
 - App.css
 - App.js M** (highlighted with a red arrow)
 - index.css

Editor Panel (App.js):

```
primeiro-teste > src > JS App.js > App
1  import './App.css';
2  import Evento from './components/Evento';
3
4  function App() {
5    return (
6      <div className="App">
7        <h1>Testando Eventos</h1>
8        <Evento />
9      </div>
10     );
11   }
12   export default App;
```

Red arrows point to the import statement on line 2 and the component usage on line 8 in the code editor.

Testar

Testando Eventos

Clique aqui para disparar um evento

Ativar!

Network Elements Console Sour

top ▼ Filter

Download the React DevTools for a better o
<https://react.dev/link/react-devtools>

Opa, fui ativado!

>

Passando valores

```
primeiro-teste > src > JS App.js > App
1 import './App.css';
2 import Evento from './components/Evento';
3
4 function App() {
5   return (
6     <div className="App">
7       <h1>Testando Eventos</h1>
8       <Evento numero="1"/>
9       <Evento numero="2"/>
10
11     </div>
12   );
13 }
14 export default App;
```

```
primeiro-teste > src > components > JS Evento.js > default
1 function Evento({ numero }) {
2
3   function meuEvento() {
4     console.log(`Opa, fui ativado! ==> ${numero}`);
5   }
6
7   return (
8     <div>
9       <p>Clique aqui para disparar um evento</p>
10      <button onClick={meuEvento}>Ativar!</button>
11    </div>
12  )
13 }
14
15 export default Evento
```

Passando valores

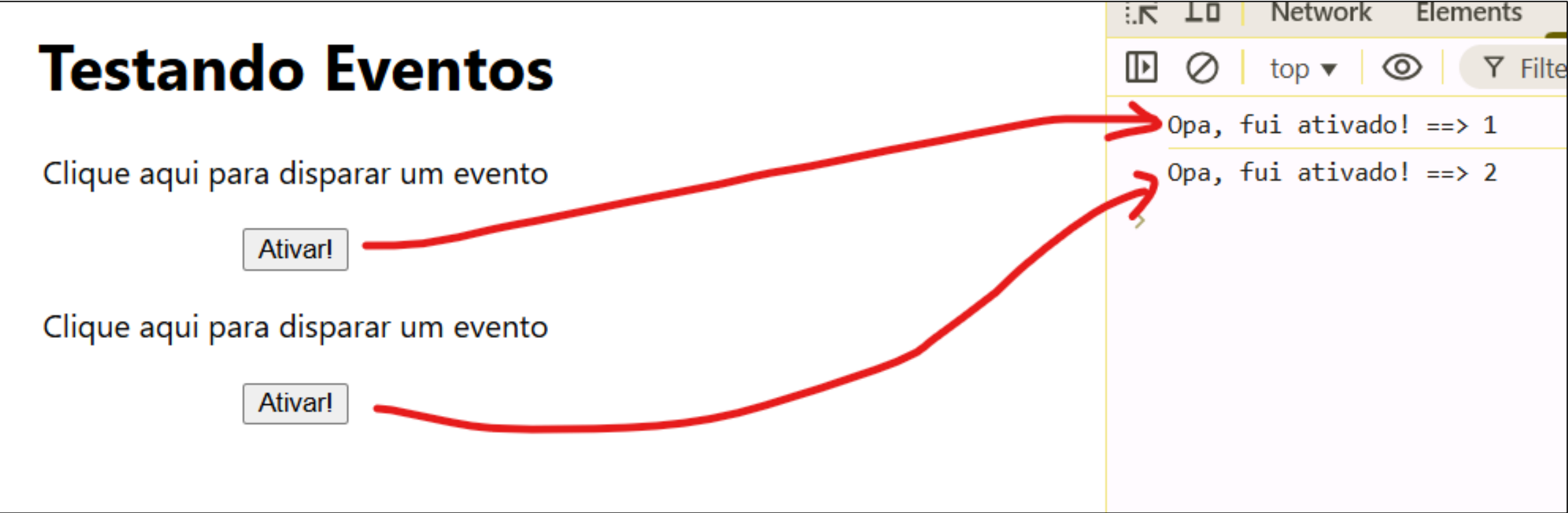
Testando Eventos

Clique aqui para disparar um evento

Ativar!

Clique aqui para disparar um evento

Ativar!



The screenshot shows a web application interface on the left and a browser's developer tools network log on the right. The interface has two identical sections, each with the text 'Clique aqui para disparar um evento' and a button labeled 'Ativar!'. Red arrows originate from each button and point to the network log. The network log, titled 'Network', shows two entries. The first entry is 'Opa, fui ativado! ==> 1' and the second is 'Opa, fui ativado! ==> 2'. The log also includes standard browser developer tool icons and a filter input.

Network	Elements
Opa, fui ativado! ==> 1	
Opa, fui ativado! ==> 2	

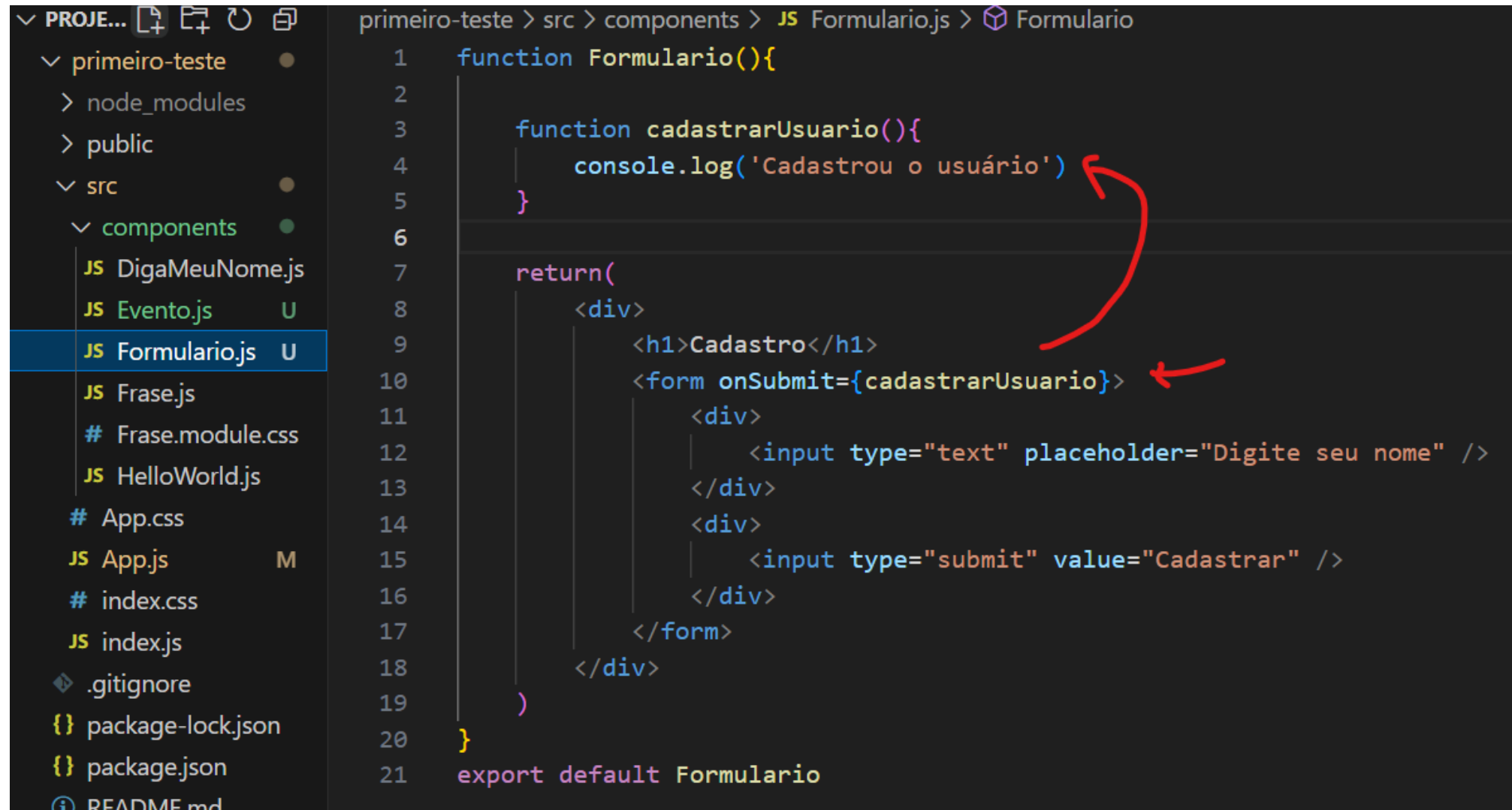
Eventos para formulários

- Criar um componente chamado **Formulario.js**



```
primeiro-teste > src > components > JS Formulario.js > Formulario
1  function Formulario(){
2      return(
3          <div>
4              <h1>Cadastro</h1>
5              <form>
6                  <div>
7                      <input type="text" placeholder="Digite seu nome" />
8                  </div>
9                  <div>
10                     <input type="submit" value="Cadastrar" />
11                 </div>
12             </form>
13         </div>
14     )
15 }
16 export default Formulario
```

Criando o evento de submissão do formulário

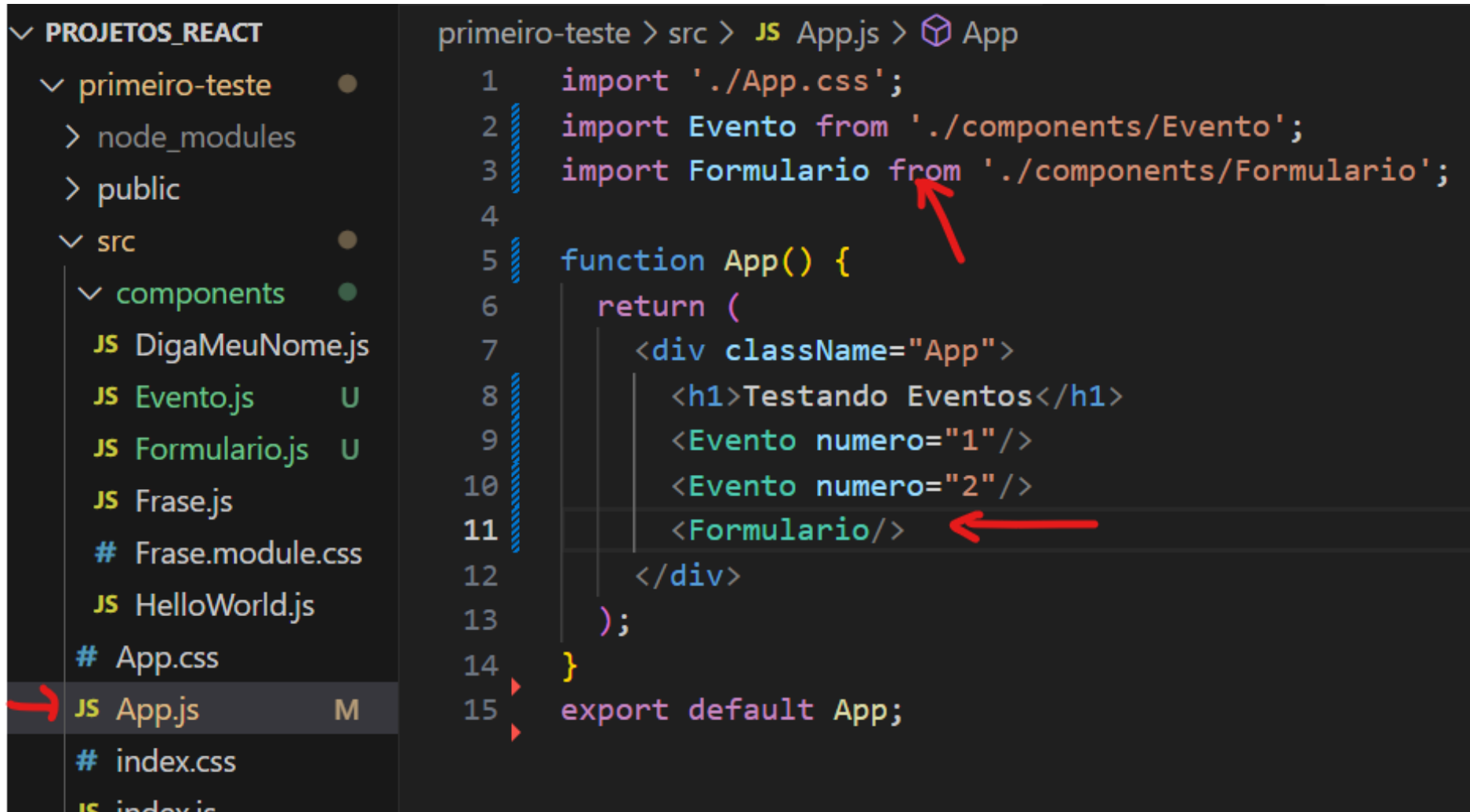


The image shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a file named `Formulario.js` selected. The code editor shows the following code:

```
1 function Formulario(){
2
3   function cadastrarUsuario(){
4     console.log('Cadastrou o usuário')
5   }
6
7   return(
8     <div>
9       <h1>Cadastro</h1>
10      <form onSubmit={cadastrarUsuario}>
11        <div>
12          <input type="text" placeholder="Digite seu nome" />
13        </div>
14        <div>
15          <input type="submit" value="Cadastrar" />
16        </div>
17      </form>
18    </div>
19  )
20 }
21 export default Formulario
```

Two red arrows are drawn on the code. One arrow points from the `onSubmit={cadastrarUsuario}` prop in the `<form>` tag (line 10) to the `cadastrarUsuario` function definition (line 3). The other arrow points from the same `onSubmit` prop to the `<input type="submit" value="Cadastrar" />` tag (line 15).

Importar o componente



```
primeiro-teste > src > JS App.js > App
1  import './App.css';
2  import Evento from './components/Evento';
3  import Formulario from './components/Formulario';
4
5  function App() {
6    return (
7      <div className="App">
8        <h1>Testando Eventos</h1>
9        <Evento numero="1"/>
10       <Evento numero="2"/>
11       <Formulario/>
12     </div>
13   );
14 }
15 export default App;
```

Quando enviamos dados de um formulário, esses dados são enviados para o backend, assim que enviados a pagina é recarregada e os dados são perdidos no frontend. Por isso o evento não foi disparado!

Criando o evento de submissão do formulário

```
primeiro-teste > src > components > JS Formulario.js > Formulario > cadastrarUsuario  
1  function Formulario(){  
2  
3      function cadastrarUsuario(e){  
4          e.preventDefault() ←  
5          console.log('Cadastrou o usuário')  
6      }  
7  
8      return(  
9          <div>  
10             <h1>Cadastro</h1>  
11             <form onSubmit={cadastrarUsuario}>  
12                 <div>  
13                     <input type="text" placeholder="Digite seu nome" />  
14                 </div>  
15                 <div>  
16                     <input type="submit" value="Cadastrar" />  
17                 </div>  
18             </form>  
19         </div>  
20     )  
21 }  
22 export default Formulario
```

Criando o evento de submissão do formulário

Testando Eventos

Clique aqui para disparar um evento

Ativar!

Clique aqui para disparar um evento

Ativar!

Cadastro

Anderson

Cadastrar

Network Elements Console

top Filter

Opa, fui ativado! ==> 1

Opa, fui ativado! ==> 2

Cadastrou o usuário