



# **GOVERNO DO ESTADO DE SÃO PAULO**

## **PROGRAMAÇÃO WEB III APLICAÇÃO DE PÁGINA ÚNICA (SPA)**

# PROGRAMAÇÃO WEB III

## APLICAÇÃO DE PÁGINA ÚNICA (SPA)



### ATIVIDADE DA PENÚLTIMA AULA CRIAÇÃO DE TABELA LISTA USANDO CSS:

Teste Lista	
•	ITEM I
	Marcas da Empresa:
•	DELL - 1984
	Marcas da Empresa:
•	Compaq - 1982
	Marcas da Empresa:
•	Faltou a Empresa!! - 0

# PROGRAMAÇÃO WEB III

## APLICAÇÃO DE PÁGINA ÚNICA (SPA)

---



Avançando no **React** vamos para os eventos, (**onClick**, **onSubmit**);

Os eventos do **React** são bem parecidos com o **DOM (Document Object Model)**, simplifica uma interface no próprio **DOM**;

Basicamente ativamos eventos que servem para responder a comandos como um click;

Estes eventos são sempre atrelados a uma **Tag**, que irão executá-los, nesta **Tag** atribuímos um método ou funcionalidades deste **Click**;

Lembrando, que estes métodos sempre devem ser criados no **componente**;

# PROGRAMAÇÃO WEB III

## APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Avançando vamos na pasta **componente** e criamos o Evento.js e sua função:

Na primeira instância, vocês vão perceber que é igual a estrutura dos outros componentes.

The screenshot shows a code editor with a dark theme. On the left is the 'EXPLORADOR' (Explorer) sidebar showing a file tree for 'PROJETO\_REACT'. The tree includes 'meu\_projeto', 'node\_modules', 'public', 'src', and 'components'. Under 'components', several files are listed: 'JS Evento.js' (highlighted), 'JS Form.js', 'JS HelloWorld.js', 'JS Item.js', and 'JS List.js'. The main editor area shows the code for 'JS Evento.js'. The breadcrumb at the top of the editor reads 'meu\_projeto > src > components > JS Evento.js > Evento'. The code is as follows:

```
1  function Evento(){
2  return(
3    <div>
4      <p>Clique para disparar um Evento:</p>
5      <button> Ativar! </button>
6    </div>
7
8  )
9
10 }
11 export default Evento;
12
```

# PROGRAMAÇÃO WEB III

## APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Depois que montamos a estrutura do botão, temos que dar FUNCIONALIDADE:

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

v PROJETO_RE...  [+][-]  [refresh]  [copy]
v meu_projeto
  > node_modules
  > public
  v src
    v components
      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
      # App.css

meu_projeto > src > components > JS Evento.js > [icon] Evento
1  function Evento(){
2
3      function meuEvento(){
4  console.log('Opsss, disparo ativo!!')
5
6      }
7
8  return(
9  <div>
10     <p>Clique para disparar um Evento:</p>
11     <button onClick={meuEvento}> Ativar! </button>
12  </div>
13
14  )
15
16  }
17  export default Evento;
18
```

# PROGRAMAÇÃO WEB III

## APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Vamos no arquivo **App.js** que sempre chama todos as funções do HTML, APAGAMOS TUDO DAS OUTRAS AULA P/ NÃO ATRAPALHAR E INSERIMOS O IMPORT do Evento, **npm start** executa servidor:

The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows the project structure: 'meu\_projeto' (expanded) contains 'node\_modules', 'public', 'src' (expanded), and '.gitignore'. Under 'src', there are files: 'App.css', 'App.js' (selected), 'App.test.js', 'index.css', 'index.js', 'logo.svg', 'reportWebVitals.js', and 'setupTests.js'. The code editor shows the content of 'App.js':

```
meu_projeto > 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;
```

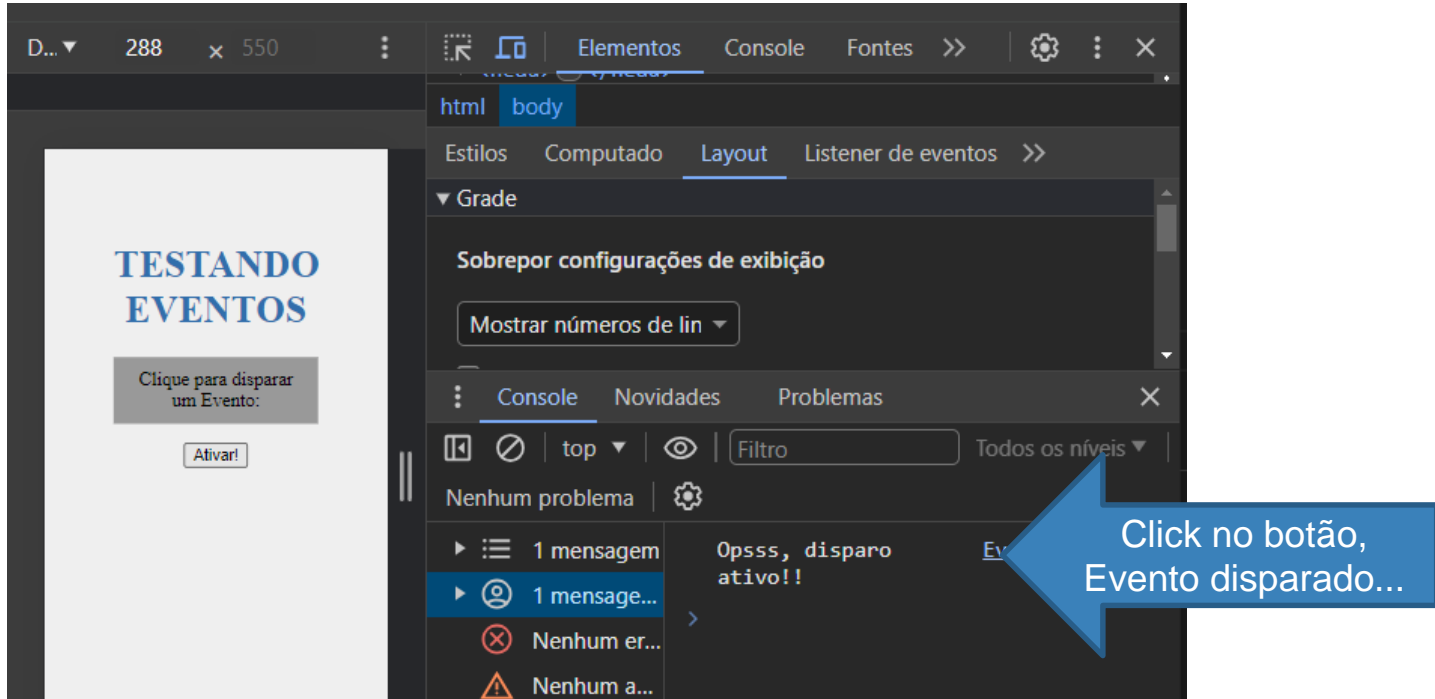
At the bottom of the editor, there is a terminal window with the command prompt 'PS C:\Users\Prof.Augusto\Desktop\projeto\_react\meu\_projeto>' and the command 'npm start' entered.

# PROGRAMAÇÃO WEB III

## APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Entrando no navegador, indo no inspecionar elementos, podemos ver a execução no console, esta função tem a mesma lógica para todas os acontecimentos que eu queira executar na página:



# PROGRAMAÇÃO WEB III

## APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Me baseando da mesma forma que as outras estruturas feitas, posso inserir dados por exemplo e chamar depois este evento clicando no botão:

```
... JS Evento.js JS App.js X
meu_projeto > 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     </div>
11   )
12 }
13 export default App;
```

Posso inserir a  
informação no Evento...



# PROGRAMAÇÃO WEB III

## APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Indo no componente Evento.js colocamos a propriedade da variável do componente:

EXPLORADOR

JS Evento.js

JS App.js

meu\_projeto > src > components > JS Evento.js > Evento > meuEvento

```
1 function Evento({numero}){  
2   function meuEvento(){  
3     console.log(`Opsss, disparo ativo!!${numero}`)  
4   }  
5  
6   return(  
7     <div>
```

Utilizar acento grave..

# PROGRAMAÇÃO WEB III

## APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Indo no componente **Evento.js** colocamos a propriedade da variável do componente:

The screenshot shows a web browser window with a SPA application. The main content area displays the text "TESTANDO EVENTOS" in blue. Below this, there are two identical gray boxes, each containing the text "Clique para disparar um Evento:" and a button labeled "Ativar!". The browser's developer tools are open, showing the "Console" tab. The console log shows two messages: "Opsss, disparo ativo!!1" and "Opsss, disparo ativo!!2", both linked to "Evento.js:3". A blue arrow points to the console log with the text "Executando os dois cliques..".

html body

Estilos Computado Layout Listener de eventos >>

▼ Grade

Sobrepôr configurações de exibição

Mostrar números de lin ▼

Console Novidades Problemas

Nenhum problema

2 message... Opsss, disparo ativo!!1 [Evento.js:3](#)

2 message... Opsss, disparo ativo!!2 [Evento.js:3](#)

Nenhum er...

Nenhum a...

2 informaç...

Sem mens...

# PROGRAMAÇÃO WEB III

## APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Outro exemplo, criando um Form.js, e buscando os dados de um formulário com click:

```
EXPLORADOR  ...  JS Evento.js  JS Form.js  JS App.js  # App.css

PROJETO_REACT
├── meu_projeto
│   ├── node_modules
│   ├── public
│   └── src
│       ├── components
│       │   ├── 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
│       │   ├── # App.css
│       │   ├── JS App.js
│       │   ├── JS App.test.js
│       │   ├── # index.css
│       │   ├── JS index.js
│       │   └── logo.svg
└── ...

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

Não esquecer da função..

# PROGRAMAÇÃO WEB III

## APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Outro exemplo, dentro do App.js importamos o Form.js:

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

v PROJETO_REACT
v meu_projeto
  > node_modules
  > public
  v src
    > components
    # App.css
    JS App.js
    JS App.test.js
    # index.css
    JS index.js
    logo.svg
    JS reportWebVitals.js
    JS setupTests.js
    .gitignore

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 numero="1"/>
10       <Evento numero="2"/>
11       <Form/>
12     </div>
13   )
14 }
15 export default App;
16
```

Porém surge um problema, os dados ficam no cache mas não são chamados no console..

# PROGRAMAÇÃO WEB III

## APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Então temos que fazer uma pequena modificação na função, pausando ela antes da execução dos clicks e assim direcionar os dados da página para o script:

```
EXPLORADOR  ...  JS Evento.js  JS Form.js  JS App.js  # App.css

PROJETO_REACT
├── meu_projeto
│   ├── node_modules
│   ├── public
│   └── src
│       ├── components
│       │   ├── 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
│       ├── # App.css
│       ├── JS App.js
│       ├── JS App.test.js
│       └── # index.css
└── ...

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

preventDefault: é uma pequena função crucial, que ajuda no controle dos eventos dando uma pausa...

# PROGRAMAÇÃO WEB III

## APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Entrando no console podemos ver que a execução foi aceita:

The screenshot shows a web browser window displaying a single-page application (SPA) running on localhost:3000. The application has a dark theme and a central content area with the title "TESTANDO EVENTOS" and "Meu Cadastro:". Below the title, there are two buttons labeled "Ativar!". The main content area contains a form with a text input field containing the word "banana" and a "Cadastrar" button. The browser's developer console is open, showing a message "Cadastrou Usuário!!!" and a log entry "Form.js". A blue arrow points to the "Form.js" log entry in the console.

```
meu_projeto > src > components >  
1 function Form(){  
2  
3   function cadastra  
4     event.preventDefault  
5     console.log(  
6   )  
7   }  
8   return(  
9     <div>  
10      <h1>Meu C  
11      <form onS  
12      <div>  
13      </div>  
14      <div>  
15      </div>  
16      </div>  
17    )  
18  )  
19  
20 }  
21 export default Form
```

# PROGRAMAÇÃO WEB III

## APLICAÇÃO DE PÁGINA ÚNICA (SPA)

---

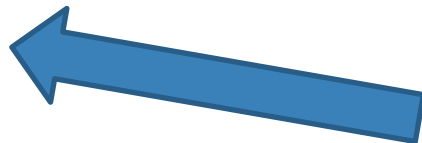


**Próximo useState e o onChange:**

**Existem diversos hook's, eles são funções que permitem você “ligar-se” aos recursos dos componentes do React;**

**Com ele conseguimos manusear o estado de um componente de forma simples;**

**Os hook, são bem projetados para serem executados nos eventos, com isto podemos atrelar um evento a uma mudança de estado.**



# PROGRAMAÇÃO WEB III

## APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Vamos dar uma turbinada com o useState no Form.js:

The screenshot shows a code editor with the following code in `Form.js`:

```
1 function Form(){
2   function cadastrarUsuario(event){
3     console.log('Cadastrou Usuário!!!')
4   }
5   return(
6     <div>
7       <h1>Meu Cadastro: </h1>
8       <form onSubmit={cadastrarUsuario}>
9         <div>
10           <label htmlFor="name">Nome: </label>
11           <input type="text" id="name" name="name" placeholder="Digite o seu nome."/>
12         </div>
13         <div>
14           <label htmlFor="password">Senha: </label>
15           <input type="password" id="password" name="password" placeholder="Digite a sua senha."/>
16         </div>
17         <div>
18           <input type="submit" value="Cadastrar"/>
19         </div>
20       </form>
21     </div>
22   )
23 }
24 export default Form
```

A blue arrow points to the `htmlFor` attribute in the labels, with the text: "Detalhe p/ o atributo FOR que seria reservada no JavaScript então usamos (htmlFor) no label...".

The browser preview shows the rendered form:

**Meu Cadastro:**

Nome:

Senha:



# PROGRAMAÇÃO WEB III

## APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Agora que montamos o escopo, dentro ainda do formulário (Form.js), importamos o `useState`, e adicionamos no `console.log` o `name`, colocamos uma constante para armazenar os dados, e colocamos o `onChange` para manipular a variável buscando no HTML e jogando no `name`:

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

```
1 import { useState } from 'react'
2
3 function Form(){
4   function cadastrarUsuario(event){
5     event.preventDefault()
6     console.log(name)
7     console.log('Cadastrou Usuário!!!')
8   }
9
10  const [name, setName] = useState()
11
12  return(
13    <div>
14      <h1>Meu Cadastro: </h1>
15      <form onSubmit={cadastrarUsuario}>
16        <div>
17          <label htmlFor="name">Nome: </label>
18          <input
19            type="text"
20            id="name"
21            name="name"
22            placeholder="Digite o seu nome."
23            onChange={(e) => setName(e.target.value)} />
24        </div>
25      </form>
26    </div>
27  )
28 }
```

The web browser on the right shows the rendered application. The title is "Meu Cadastro:". Below the title, there is a form with two input fields: "Nome: Digite o seu nome." and "Senha: Digite a sua senha.". A "Cadastrar" button is located below the "Senha" field.

# PROGRAMAÇÃO WEB III

## APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Agora podemos inspecionar os elementos e verificar que o nome foi anexado ao script do js basicamente este é o hook:

The image displays a web application running on a browser at `localhost:3000/?`. The application is a single-page application (SPA) for user registration. The browser window shows the URL and several tabs, including 'React App'. The page content includes a button labeled 'Ativar!', a message 'Clique para disparar um Evento:', another 'Ativar!' button, and a registration form titled 'Meu Cadastro:'. The form has fields for 'Nome:' (with 'teste' entered), 'Senha:', and a 'Cadastrar' button. Below the form, there is a message 'Digite a sua senha.' and a 'Cadastrar' button.

The background shows the code editor with the following JavaScript code in `Form.js`:

```
1 import { useState } from 'react'
2
3 function Form(){
4   function cadastrarUsuario(event){
5     event.preventDefault()
6     console.log(name)
7     console.log('Cadastrou Usuário!!!')
8   }
9
10  const [name, setName] = useState()
11
12  return(
13    <div>
14      <h1>Meu Cadastro: </h1>
15      <form onSubmit=cadastrarUsuario>
16        <div>
17          <label htmlFor="name">Nome:
18          <input
19            type="text"
20            id="name"
21            name="name"
22            placeholder="Digite o seu nome"
23            onChange={(e) => setName(e.target.value)}
24          </div>
25          <div>
```

The right side of the image shows the browser's developer tools. The 'Elements' panel is open, showing the HTML structure of the page. The 'Console' panel is also open, showing the following log messages:

- 5 mensagens
- 3 mensagens do usuário...
- main.18aec... 1
- main.f8ff17b9... 2
- Nenhum erro
- Nenhum aviso
- 3 informações
- 2 mensagens detalhadas...

The 'Form.js:16' and 'Form.js:7' links in the console indicate the source of the log messages.

# PROGRAMAÇÃO WEB III

## APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Agora podemos montar o mesmo esquema com o login e a senha:

```
EXPLORADOR    ...    JS Form.js    JS App.js
meu_projeto > src > components > JS Form.js > Form

2
3  function Form(){
4      function cadastrarUsuario(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={cadastrarUsuario}>
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                  </div>
24                  <div>
25                      <label htmlFor="password">Senha: </label>
26                      <input
27                          type="password"
28                          id="password"
29                          name="password"
30                          placeholder="Digite a sua senha."
31                          onChange={(e) => setPassword(e.target.value)} />
32                  </div>
33              </div>
34          </div>
35      )
36  }
```

# PROGRAMAÇÃO WEB III

## APLICAÇÃO DE PÁGINA ÚNICA (SPA)



Vai ficar assim, por enquanto, depois quando tiver mais diretivas, vamos trabalhar com todas estas informações no HTML do site, para isto precisam entender os conceitos e comandos do react.

The screenshot displays a development environment with a code editor on the left and a live preview on the right. The code editor shows the `Form.js` file with the following content:

```
function Form(){
  function cadastrarUsuario(event){
    event.preventDefault()
    console.log(`Usuário ${name} foi cadastrado com a senha: ${password}`)
  }

  const [name, setName] = useState()
  const [password, setPassword] = useState()

  return(
    <div>
      <h1>Meu Cadastro: </h1>
      <form onSubmit={cadastrarUsuario}>
        <div>
          <label htmlFor="name">Nome: </label>
          <input
            type="text"
            id="name"
            name="name"
            placeholder="Digite o seu nome."
            onChange={(e) => setName(e.target.value)}
          />
        </div>
        <div>
          <label htmlFor="password">Senha: </label>
          <input
            type="password"
            id="password"
            name="password"
            placeholder="Digite a sua senha."
            onChange={(e) => setPassword(e.target.value)}
          />
        </div>
        <button type="submit">Cadastrar</button>
      </form>
    </div>
  )
}
```

The live preview shows a web browser at `localhost:3000/?` displaying a registration form titled "Meu Cadastro:". The form includes a "Nome:" label, a text input field containing "banana", a "Senha:" label, a password input field, and a "Cadastrar" button. Above the form is a button labeled "Ativar!". The browser's developer tools are open, showing the "Elementos" (Elements) panel with the form structure, the "Console" panel with two messages, and the "Problemas" (Problems) panel.

**Console Messages:**

- Usuário `Form.js:6` teste foi cadastrado com a senha: 123
- Usuário `Form.js:6` banana foi cadastrado com a senha: 666

# PROGRAMAÇÃO WEB III

## APLICAÇÃO DE PÁGINA ÚNICA (SPA)



### TAREFA:

Para o feriado do dia 22/04, criem o cadastro de login e senha no REACT, e entregar no Teams pode ser em dupla, cadastrar o nome dos dois da dupla e senhas no console do js, para print mandem o arquivo tbém...

The screenshot displays a development environment with a code editor on the left and a browser preview on the right.

**Code Editor (JS Form.js):**

```
function Form(){
  function cadastrarUsuario(event){
    event.preventDefault()
    console.log(`Usuário ${name} foi cadastrado com a senha: ${password}`)
  }
  const [name, setName] = useState('')
  const [password, setPassword] = useState('')

  return(
    <div>
      <h1>Meu Cadastro: </h1>
      <form onSubmit={cadastrarUsuario}>
        <div>
          <label htmlFor="name">Nome: </label>
          <input
            type="text"
            id="name"
            name="name"
            placeholder="Digite o seu nome."
            onChange={(e) => setName(e.target.value)}
          />
        </div>
        <div>
          <label htmlFor="password">Senha: </label>
          <input
            type="password"
            id="password"
            name="password"
            placeholder="Digite a sua senha."
            onChange={(e) => setPassword(e.target.value)}
          />
        </div>
        <button type="submit">Cadastrar</button>
      </form>
    </div>
  )
}
```

**Browser Preview (React App):**

The browser shows the rendered form at `localhost:3000/?`. The form has the title "Meu Cadastro:" and contains two input fields: "Nome:" (with the value "banana") and "Senha:" (with a masked value "..."). A "Cadastrar" button is at the bottom.

**Console:**

The console shows two messages:

- Usuário `Form.js:6` teste foi cadastrado com a senha: 123
- Usuário `Form.js:6` banana foi cadastrado com a senha: 666



# **GOVERNO DO ESTADO DE SÃO PAULO**