

Desenvolvimento Front-end com Frameworks

Fundamentos de React

Agenda

Etapas 4: Framework React JS.

- Jogo de Palavras.
- useEffect Hook.
- TheMealDB.





Jogo de Palavras

EXPLORADOR

...

▼ DR2_E3_A1

> node_modules

> public

▼ src

▼ infra

JS enderecos.js

JS pais.js

▼ pages

Tela1.jsx

Tela2.jsx

Tela3.jsx

App.css

App.jsx

JS App.test.js

index.css

JS index.js

logo.svg

JS reportWebVitals.js

JS setupTests.js

.gitignore

{ } package-lock.json

{ } package.json

i README.md

Tela3.jsx U X

src > pages > Tela3.jsx > Tela3

1 import { useState } from "react";

2

3 export default function Tela3() {

4

5 const palavras = ["elefante", "refrigerante", "maracanã", "saturno", "liderança",
6 "maracujá", "saxofone", "samambaia", "celular", "programação"];

7

8 const [palavra, setPalavra] = useState("");

9

const [embaralhada, setEmbaralhada] = useState("");

10

const [resultado, setResultado] = useState("");

1

2

Tela3.jsx U x

src > pages > Tela3.jsx > Tela3

```
3   export default function Tela3() {  
12     function handleJogar() {  
13       const indice = Math.floor(Math.random() * palavras.length);  
14       let sorteada = palavras[indice];  
15       if (sorteada && sorteada.length > 0) {  
16         let letras = sorteada.split("");  
17         letras.sort(() => Math.random() - 0.5);  
18         setResultado("");  
19         setPalavra(sorteada);  
20         setEmbaralhada(letras.join(""));  
21       }  
22     }  
23  
24     function handleValidar() {  
25       const palpito = document.getElementById("palpite").value;  
26       if (palpito === palavra) {  
27         setResultado("CORRETO");  
28       } else {  
29         setResultado("ERRADO");  
30       }  
31       setEmbaralhada("");  
32     }  
}
```

Tela3.jsx U x

src > pages > Tela3.jsx > ...

```
3   export default function Tela3() {
34     return (
35       <div className="container">
36         <h3>Jogo de Palavras</h3>
37         <p>Neste jogo você terá que adivinhar qual é a palavra que está sendo apresentada com as letras embaralhadas.
38         <button onClick={handleJogar}>Jogar</button>
39         {embaralhada} &&
40         <div>
41           <h4>{embaralhada}</h4>
42           <input id="palpite" />
43           <button onClick={handleValidar}>Validar</button>
44         </div>
45       </div>
46       {resultado} &&
47       <p>A palavra é {palavra} e seu palpite está {resultado}</p>
48     </div>
49   );
50 }
51
```


React App

localhost:3000

Reiniciar para atualizar

AdministrativoConteúdoFerramentasLivros

Jogo de Palavras

Neste jogo você terá que adivinhar qual é a palavra que está sendo apresentada com as letras embaralhadas

Jogar

A palavra é saturno e seu palpite está CORRETO

ElementosConsole

topFiltro

Todos os níveisNenhum problema

2 - maracanãbundle.js:176

5 - maracujábundle.js:176

3 - saturnobundle.js:176



Tela4.jsx U

JS ibge.js U X

src > infra > JS ibge.js > ...

```
1  export default async function listarEstados() {
2    let retorno = [];
3    const url = `https://servicodados.ibge.gov.br/api/v1/localidades/estados?orderBy=nome`;
4    await fetch(url)
5      .then((resposta) => resposta.json())
6      .then((estados) => {
7        retorno = estados;
8        console.log(estados);
9      })
10     .catch((erro) => retorno.erro = erro);
11    return retorno;
12  }
```


5

Tela4.jsx U x

JS ibge.js U

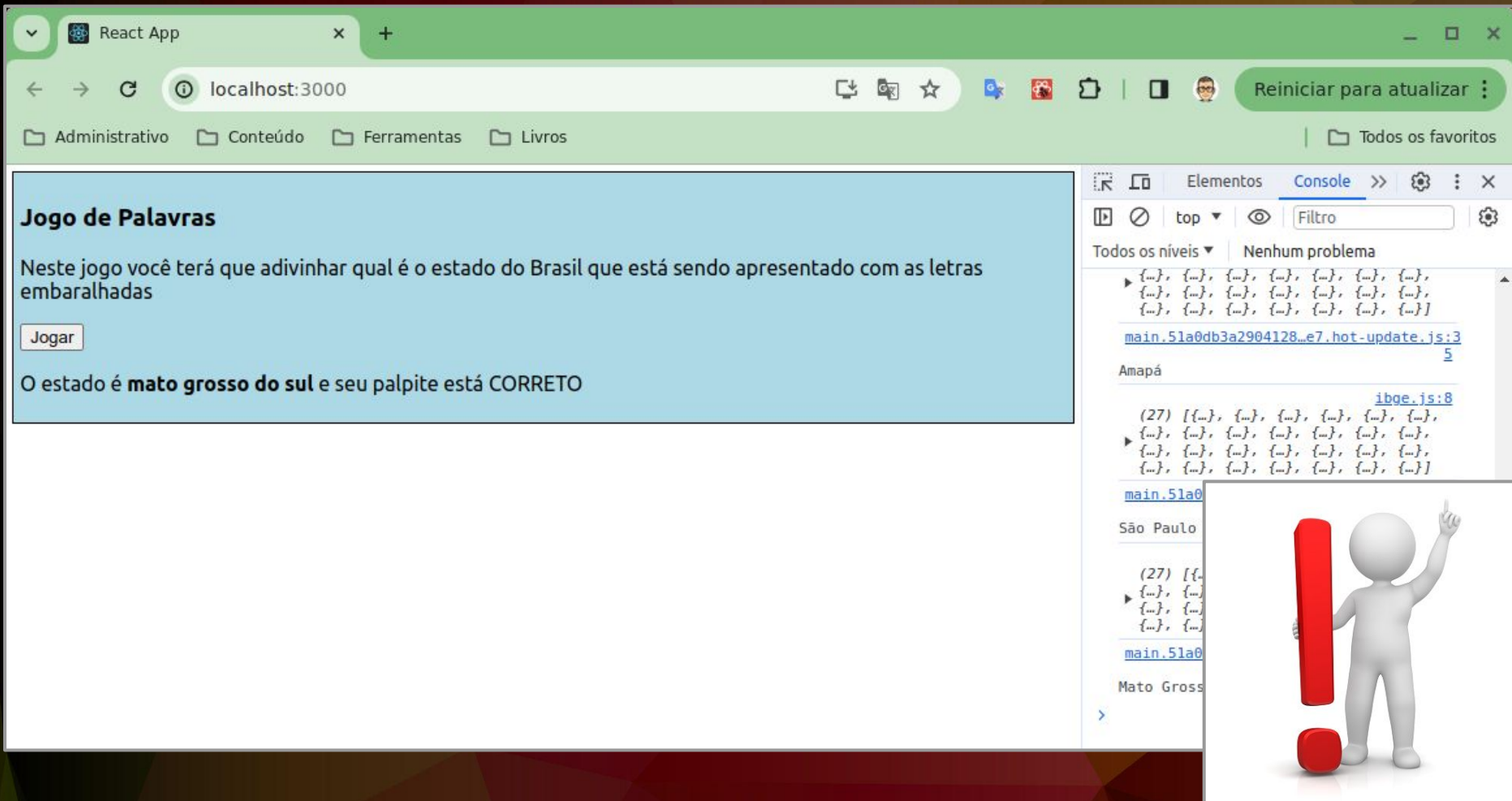
src > pages > Tela4.jsx > ...

```
1  import { useState } from "react";
2  import listarEstados from "../infra/ibge";
3
4  export default function Tela4() {
5
6      const [palavra, setPalavra] = useState("");
7      const [embaralhada, setEmbaralhada] = useState("");
8      const [resultado, setResultado] = useState("");
9
10     async function handleJogar() {
11         const palavras = await listarEstados();
12         const indice = Math.floor(Math.random() * palavras.length);
13         let sorteada = palavras[indice].nome;
14         console.log(sorteada);
15         if (sorteada && sorteada.length > 0) {
16             let letras = sorteada.split("");
17             letras.sort(() => Math.random() - 0.5);
18             setResultado("");
19             setPalavra(sorteada.toLowerCase());
20             setEmbaralhada(letras.join("").toLowerCase().replaceAll(" ", "_"));
21         }
22     }
```

Tela4.jsx U x JS ibge.js U

src > pages > Tela4.jsx > ...

```
4 export default function Tela4() {
34   return (
35     <div className="container">
36       <h3>Jogo de Palavras</h3>
37       <p>Neste jogo você terá que adivinhar qual é o estado do Brasil que está sendo apresentado com as letras emb
38       <button onClick={handleJogar}>Jogar</button>
39       {embaralhada} &&
40       <div>
41         <h4 style={{letterSpacing: 3}}>{embaralhada}</h4>
42         <input id="palpite" />
43         <button onClick={handleValidar}>Validar</button>
44       </div>
45     </div>
46     {resultado} &&
47     <p>O estado é <b>{palavra}</b> e seu palpite está {resultado}</p>
48   )
49 </div>
50 );
51 }
```

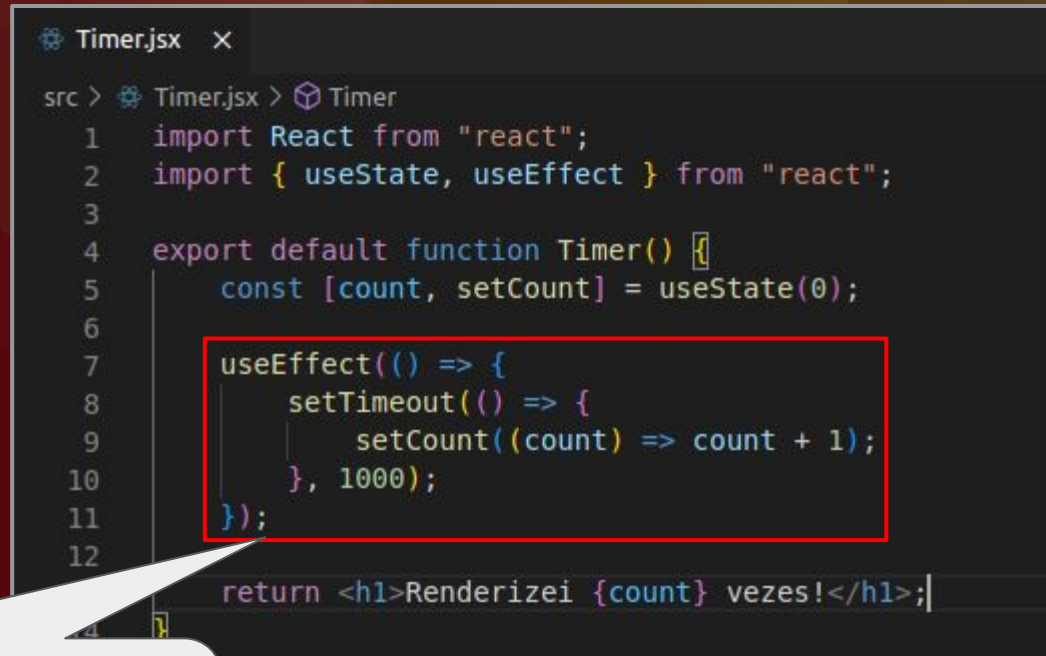




useEffect Hook

O **useEffect** permite que você execute código depois da renderização, como complemento desta - “efeitos colaterais”.

Alguns exemplos de efeitos colaterais são: buscar dados, atualizar diretamente o DOM e temporizadores.

A screenshot of a code editor window titled 'Timer.jsx'. The code defines a function 'Timer' that uses 'useState' to manage a 'count' and 'useEffect' to set a timeout that increments the count every 1000ms. The 'useEffect' call is highlighted with a red rectangle. The component returns an HTML element showing the current count.

```
Timer.jsx x
src > Timer.jsx > Timer
1  import React from "react";
2  import { useState, useEffect } from "react";
3
4  export default function Timer() {
5    const [count, setCount] = useState(0);
6
7    useEffect(() => {
8      setTimeout(() => {
9        setCount((count) => count + 1);
10     }, 1000);
11   });
12
13   return <h1>Renderizei {count} vezes!</h1>;
14 }
```

useEffect é **executado em cada renderização**.

Isso significa que, quando a contagem muda, ocorre uma renderização, que aciona outro efeito.

Podemos incluir o segundo parâmetro que aceita um array e opcionalmente podemos passar dependências neste array.

Quando é um array vazio, useEffect só roda uma vez.

Quando tem mais parâmetros, condiciona a renderização às mudanças destes - podem ser propriedades ou estados.

1

```
useEffect(() => {  
  //Runs on every render  
});
```

2

```
useEffect(() => {  
  //Runs only on the first render  
}, []);
```

3

```
useEffect(() => {  
  //Runs on the first render  
  //And any time any dependency value changes  
}, [prop, state]);
```


Counter.jsx X

src > Counter.jsx > ...

```
1 import React from "react";
2 import { useState, useEffect } from "react";
3
4 export default function Counter() {
5   const [count, setCount] = useState(0);
6   const [calculation, setCalculation] = useState(0);
7
8   3 useEffect(() => {
9     setCalculation(() => count * 2);
10   }, [count]); // <- add the count variable here
11
12   return (
13     <div>
14       <p>Count: {count}</p>
15       1 <button onClick={() => setCount((c) => c + 1)}>+</button>
16       <p>Calculation: {calculation}</p>
17     </div>
18   );
19 }
```

1

setCount

2

React updates DOM
and refs

3

useEffect



TheMealDB

1

Tela5.jsx U

JS refeicoes.js U X

src > infra > JS refeicoes.js > ...

```
1  export async function listarCategorias() {
2    let retorno = [];
3    const url = `https://www.themealdb.com/api/json/v1/1/categories.php`;
4    await fetch(url)
5      .then((resposta) => resposta.json())
6      .then((categorias) => {
7        retorno = categorias.categories;
8        console.log(retorno);
9      })
10     .catch((erro) => retorno.erro = erro);
11    return retorno;
12  }
```

2

```
Tela5.jsx U x JS refeicoes.js U
src > pages > Tela5.jsx > Tela5
1  import { useEffect, useState } from "react";
2  import { listarCategorias } from "../infra/refeicoes";
3
4  export default function Tela5() {
5
6      const [categorias, setCategorias] = useState([]);
7      const [categoria, setCategoria] = useState({});
8
9      async function carregarCategorias() {
10         const dados = await listarCategorias();
11         setCategorias(dados);
12     }
13
14     useEffect(() => {
15         carregarCategorias();
16     }, []);
```

```
Tela5.jsx U x JS refeicoes.js U
src > pages > Tela5.jsx > Tela5
4 export default function Tela5() {
18   function handleCategoria(event) {
19     const index = event.target.value;
20     console.log(index);
21     if (index >= 0) {
22       setCategoria(categorias[index]);
23     }
24   }
25
26   return (
27     <div className="container">
28       <h3>Categorias de Receitas</h3>
29       <select onChange={handleCategoria}>
30         <option key={0} value={-1}>Selecione...</option>
31         {categorias.map((cat, index) =>
32           <option key={index} value={index}>
33             {cat.strCategory}
34           </option>)
35         }
36       </select>
37       {categoria &&
38         <div>
39           <p className="ladoalado">{categoria.strCategoryDescription}</p>
40           <img className="ladoalado" src={categoria.strCategoryThumb} />
41         </div>
42       }
43     </div>
44   );
45 }
```

3

4

```
Tela5.jsx U # App.css M x JS
src > # App.css > .ladoalado
9 .ladoalado {
10   width: 50%;
11   vertical-align: top;
12   display: inline-table;
13 }
```

React App

localhost:3000


AdministrativoConteúdoFerramentasLivros

Todos os favoritos

Categorias de Receitas

Dessert

Dessert is a course that concludes a meal. The course usually consists of sweet foods, such as confections dishes or fruit, and possibly a beverage such as dessert wine or liqueur, however in the United States it may include coffee, cheeses, nuts, or other savory items regarded as a separate course elsewhere. In some parts of the world, such as much of central and western Africa, and most parts of China, there is no tradition of a dessert course to conclude a meal. The term dessert can apply to many confections, such as biscuits, cakes, cookies, custards, gelatins, ice creams, pastries, pies, puddings, and sweet soups, and tarts. Fruit is also commonly found in dessert courses because of its naturally occurring sweetness. Some cultures sweeten foods that are more commonly savory to create desserts.



Console

1

top

Filtro

Todos os níveis

1 problema: 1

refeicoes.js:8

(14) [{"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}]

refeicoes.js:8

(14) [{"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}, {"-"}]

2

Tela5.jsx:20

>

