

# Curso de React

## CFB Cursos (Bruno Pinho Campos)

Vídeos do curso:

[https://www.youtube.com/watch?v=1LhX2u6\\_BJE&list=PLx4x\\_zx8csUh752BVDGZkxYpY9IS40fyC](https://www.youtube.com/watch?v=1LhX2u6_BJE&list=PLx4x_zx8csUh752BVDGZkxYpY9IS40fyC)

Resumo feito por Roberto Pinheiro

<https://github.com/betopinheiro1005>

### Aula 01 - Introdução ao React

O React é uma biblioteca JavaScript de código aberto com foco em criar interfaces de usuário em páginas web. É mantido pelo Facebook, Instagram, outras empresas e uma comunidade de desenvolvedores individuais.

- Criado pelo Facebook
- Biblioteca para criação de interfaces de usuário
- Roda Javascript no servidor e retorna a página pronta
- Baseado em Javascript ES6
- React só atualiza os componentes que sofrerão alterações

### Fluxo de dados unidirecional

IMPORTANTE: O componente filho não consegue alterar uma informação do componente pai, somente consegue ler esta informação. Caso o componente filho precise alterar a informação do componente pai, ele precisa chamar um método do componente pai para fazer essa alteração. Quando o componente pai altera a informação isso se reflete no componente filho automaticamente.

### Tudo é Javascript

1. Todo conteúdo inserido é Javascript
2. Usamos uma extensão de sintaxe chamada JSX que nos possibilita escrever HTML junto de Javascript

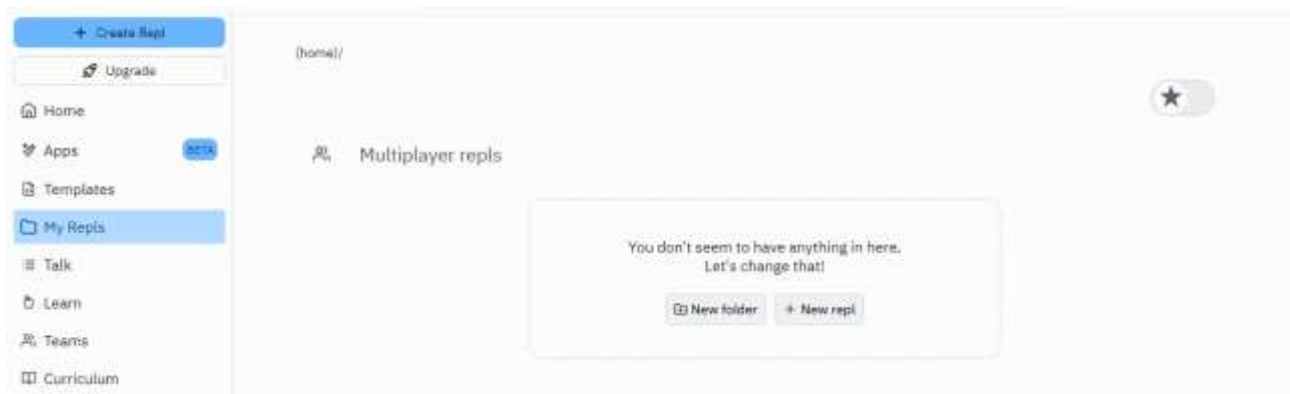
## Replit

O replit permite criar e testar aplicativos de diversas linguagens, sem a necessidade de instalá-los no seu computador.

<https://replit.com/>



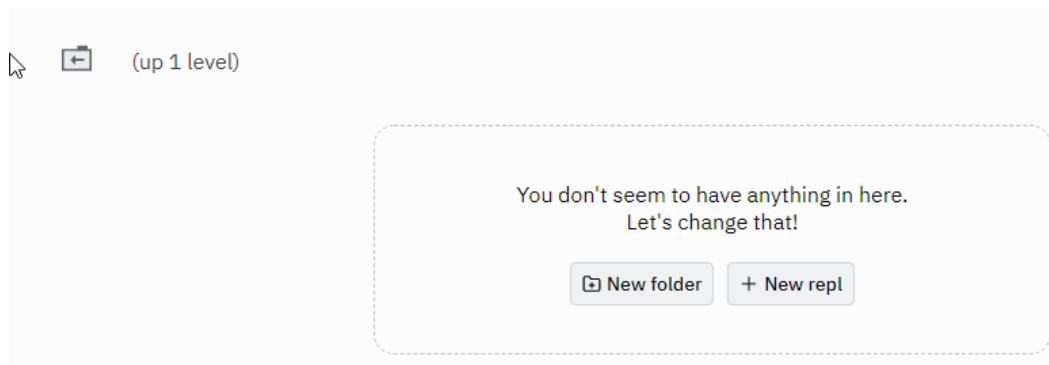
- Acesse e cadastre-se.



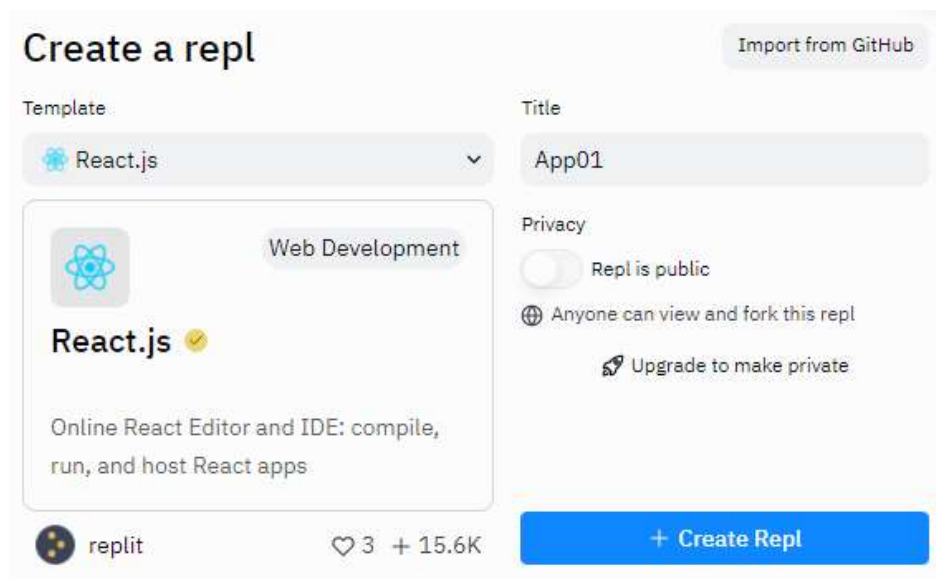
- Clique em **MyRepls**

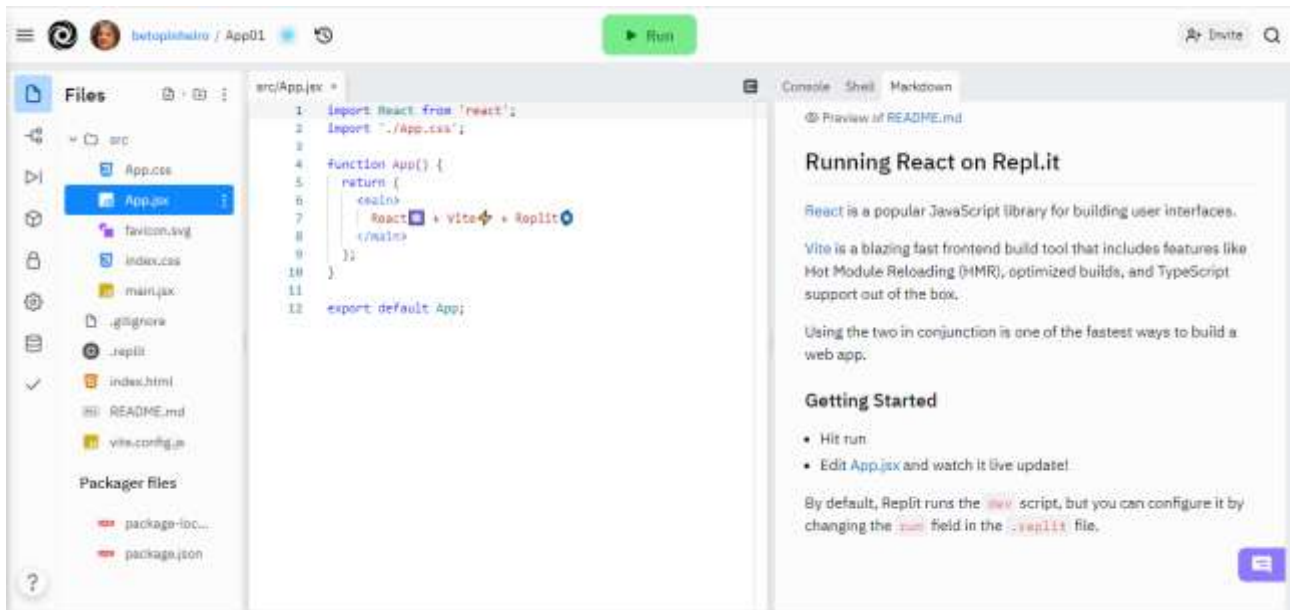
- Crie uma nova pasta chamada "**Curso de React**" e clique nela.



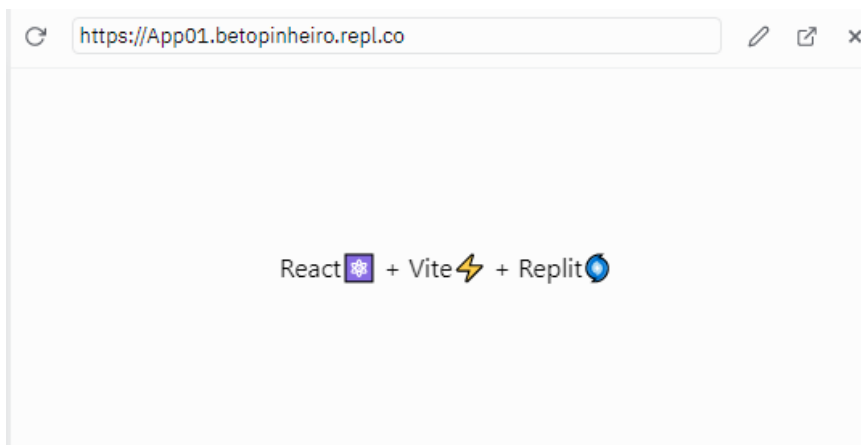


- Crie um novo repl e seleccione como template "**React JS**" e nomeie como **App01**:





- Clique no botão **Run**



## Aula 02 - Instalação do Node e primeiro App criado diretamente pelo PC

Vamos preparar o ambiente Windows instalando o Node e já vamos criar nosso primeiro app React diretamente pelo computador.

- Baixe o **Node** e instale.
- Baixe o **Visual Studio Code** e instale.
- Crie a pasta do aplicativo: **App01**

### Criando o aplicativo

- No terminal do Windows, entre na pasta do aplicativo, e execute o comando:

```
npx create-react-app app01
```

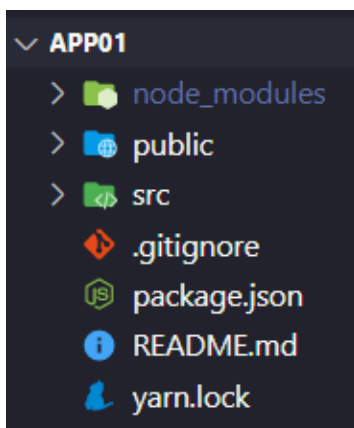
- Aguarde, o processo de criação é longo e demorado.

### Iniciando o aplicativo

```
cd app01
```

- Para abrir o aplicativo no Visual Studio Code, execute o comando:

```
code .
```



- Abra um terminal (PowerShell) e execute o comando:

```
npm start
```

<http://localhost:3000/>



#### src\App.js

```
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Edite <code>src/App.js</code> e salve para recarregar.
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          Aprenda React
        </a>
      </header>
    </div>
  );
}

export default App;
```

## app01\src\App.css

```
.App {
  text-align: center;
}

.App-logo {
  height: 40vmin;
  pointer-events: none;
}

@media (prefers-reduced-motion: no-preference) {
  .App-logo {
    animation: App-logo-spin infinite 20s linear;
  }
}

.App-header {
  background-color: #282c34;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-size: calc(10px + 2vmin);
  color: white;
}

.App-link {
  color: #61dafb;
}

@keyframes App-logo-spin {
  from {
    transform: rotate(0deg);
  }
  to {
    transform: rotate(360deg);
  }
}
```



## Aula 03 - Entendendo a estrutura básica de um App React

Vamos aprender sobre a estrutura básica de um App React e o código básico (esqueleto) de um componente em React.

### public\index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <!--
      manifest.json provides metadata used when your web app is installed on a
      user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-
manifest/
    -->
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <!--
      Notice the use of %PUBLIC_URL% in the tags above.
      It will be replaced with the URL of the `public` folder during the build.
      Only files inside the `public` folder can be referenced from the HTML.

      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
      work correctly both with client-side routing and a non-root public URL.
      Learn how to configure a non-root public URL by running `npm run build`.
    -->
    <title>React App</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
    <!--
      This HTML file is a template.
      If you open it directly in the browser, you will see an empty page.

      You can add webfonts, meta tags, or analytics to this file.
      The build step will place the bundled scripts into the <body> tag.

      To begin the development, run `npm start` or `yarn start`.
      To create a production bundle, use `npm run build` or `yarn build`.
    -->
  </body>
</html>
```



### src\index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

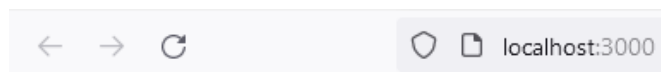
ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

### src\App.js

```
import React from 'react';

export default function App() {
  return (
    <section>
      <p>CFB Cursos</p>
      <br />
      <p>Curso de React</p>
    </section>
  );
}
```



CFB Cursos

Curso de React

## Aula 04 - Expressões em JSX Inserção de imagens no App Constantes e Variáveis

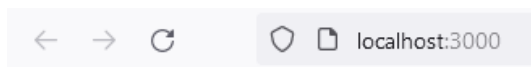
Vamos aprender sobre: Expressões em JSX, como inserir o Javascript no JSX. Inserção de imagens no App React, como inserir imagens que estão na pasta public e na pasta src. Constantes e Variáveis em React, como trabalhar com constantes e variáveis simples nos Apps React.

### Expressões em JSX

#### src\App.js

```
import React from 'react';

export default function App() {
  const canal = "CFB Cursos";
  const curso = "Curso de React";
  return (
    <section>
      <p>Canal: {canal}</p>
      <p>{curso}</p>
    </section>
  );
}
```



Canal: CFB Cursos

Curso de React

#### src\App.js

```
import React from 'react';

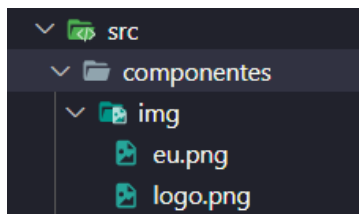
export default function App() {
  const canal = () => {
    return "CFB Cursos"
  };
  function curso(){
    return "Curso de React";
  }
  return (
    <section>
      <p>Canal: {canal()}</p>
      <p>{curso()}</p>
    </section>
  );
}
```



Canal: CFB Cursos

Curso de React

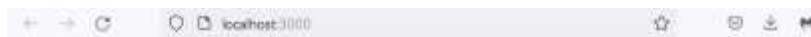
## Inserindo imagens



src\App.js

```
import React from 'react';  
import Eu from './componentes/img/eu.png';  
import Logo from './componentes/img/logo.png';
```

```
export default function App() {  
  const canal = () => {  
    return "CFB Cursos"  
  };  
  function curso() {  
    return "Curso de React";  
  }  
  return (  
    <section>  
      <p>Canal: {canal()}</p>  
      <p>{curso()}</p>  
      <img src={Logo} alt="Logo CFB" />  
      <img src={Eu} alt="Bruno P. Campos" />  
    </section>  
  );  
}
```



Canal: CFB Cursos

Curso de React



## Aula 05 - Como usar componentes

### Criando componentes

#### src\componentes\Header.js

```
import React from 'react';
import Logo from './img/logo.png'

export default function Header(){
  return (
    <header>
      <img src={Logo} alt="Logo CFB Cursos" />
      <h1>CFB Cursos</h1>
    </header>
  )
}
```

#### src\componentes\Corpo.js

```
import React from 'react'

export default function Corpo(){
  return(
    <section>
      <h2>Curso de React</h2>
      <p>Se inscreva em nosso Canal</p>
      <p>Ative o sininho e clique no joinha</p>
    </section>
  )
}
```

### Usando os componentes criados

#### src\App.js

```
import React from 'react'
import Header from './componentes/Header'
import Corpo from './componentes/Corpo'

export default function App() {
  return (
    <section>
      <Header/>
      <Corpo/>
    </section>
  );
}
```



## CFB Cursos

### Curso de React

Se inscreva em nosso Canal

Ative o sininho e clique no joinha

## Aula 06 - Enviar propriedades para os componentes com PROPS

Vamos aprender como enviar propriedades e funções para os componentes com PROPS.

### src\App.js

```
import React from 'react'
import Header from './componentes/Header'
import Corpo from './componentes/Corpo'
import Dados from './componentes/Dados'

export default function App() {
  const canal = 'CFB Cursos'
  const youtube = 'youtube.com/cfbcursos'
  const curso = 'React'

  return (
    <section>
      <Header/>
      <Corpo/>
      <Dados canal={canal} youtube={youtube} curso={curso} />
    </section>
  );
}
```

### src\componentes\Dados.js

```
import React from 'react'

export default function Dados(props){
  return (
    <section>
      <p>Canal: {props.canal}</p>
      <p>Youtube: {props.youtube}</p>
      <p>Curso: {props.curso}</p>
    </section>
  )
}
```



### CFB Cursos

#### Curso de React

Se inscreva em nosso Canal

Ative o sininho e clique no joinha

Canal: CFB Cursos

Youtube: youtube.com/cfbcursos

Curso: React

### src\App.js

```
import React from 'react'
import Header from './componentes/Header'
import Corpo from './componentes/Corpo'

export default function App() {

  return (
    <section>
      <Header/>
      <Corpo/>
    </section>
  );
}
```

### src\componentes\Corpo.js

```
import React from 'react'
import Dados from './Dados'

export default function Corpo(){
  const canal = 'CFB Cursos'
  const youtube = 'youtube.com/cfbcursos'
  const curso = 'React'

  return(
    <section>
      <h2>Curso de React</h2>
      <p>Se inscreva em nosso Canal</p>
      <p>Ative o sininho e clique no joinha</p>
      <Dados canal={canal} youtube={youtube} curso={curso} />
    </section>
  )
}
```

### src\componentes\Dados.js

```
import React from 'react'

export default function Dados(props){
  return (
    <section>
      <p>Canal: {props.canal}</p>
      <p>Youtube: {props.youtube}</p>
      <p>Curso: {props.curso}</p>
    </section>
  )
}
```



## CFB Cursos

### Curso de React

Se inscreva em nosso Canal

Ative o sininho e clique no joinha

Canal: CFB Cursos

Youtube: [youtube.com/cfbcursos](https://youtube.com/cfbcursos)

Curso: React



## Aula 07 - Enviar funções para os componentes

Vamos aprender como enviar funções para os componentes.

### src\componentes\Corpo.js

```
import React from 'react'
import Dados from './Dados'

export default function Corpo(){
  const canal = () => {
    return 'CFB Cursos'
  }
  const youtube = 'youtube.com/cfbcursos'
  const curso = 'React'
  const somar = (v1, v2) =>{
    return v1+v2
  }

  return(
    <section>
      <h2>Curso de React</h2>
      <p>Se inscreva em nosso Canal</p>
      <p>Ative o sininho e clique no joinha</p>
      <Dados
        canal={canal}
        youtube={youtube}
        curso={curso}
        somar={somar}
      />
    </section>
  )
}
```

### src\componentes\Dados.js

```
import React from 'react'

export default function Dados(props){
  let n1=10
  let n2=5
  return (
    <section>
      <p>Canal: {props.canal()}</p>
      <p>Youtube: {props.youtube}</p>
      <p>Curso: {props.curso}</p>
      <p>{n1} + {n2} = {props.somar(n1,n2)}</p>
    </section>
  )
}
```



## CFB Cursos

### Curso de React

Se inscreva em nosso Canal

Ative o sininho e clique no joinha

Canal: CFB Cursos

Youtube: [youtube.com/cfbcursos](https://youtube.com/cfbcursos)

Curso: React

10 + 5 = 15

## Aula 08 - Formas de usar CSS em React

Vamos aprender as várias formas de como usar CSS em páginas React.

### src\App.js

```
import React from 'react'
import './App.css'

export default function App() {

  const textoDestaque = {
    color: '#00f',
    fontSize: '3em'
  }

  return (
    <section className="caixa">
      <h1 style={{color: '#00f', fontSize: '5em'}}>CFB Cursos</h1>
      <h2 style={textoDestaque}>Curso de React</h2>
      <p className="texto">Se inscreva em nosso canal e nos siga no Instagram</p>
      <a href="https://www.youtube.com/cfbcursos" target="_blank">CFB Cursos</a>
    </section>
  );
}
```

### public\index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
    <!--
      manifest.json provides metadata used when your web app is installed on a
      user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/
    -->
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <!--
      Notice the use of %PUBLIC_URL% in the tags above.
      It will be replaced with the URL of the `public` folder during the build.
      Only files inside the `public` folder can be referenced from the HTML.

      Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
      work correctly both with client-side routing and a non-root public URL.
      Learn how to configure a non-root public URL by running `npm run build`.
    -->
```

```
<title>React App</title>
</head>
<body style="background-color: #ccc;">
  <noscript>You need to enable JavaScript to run this app.</noscript>
  <div id="root"></div>
  <!--
    This HTML file is a template.
    If you open it directly in the browser, you will see an empty page.

    You can add webfonts, meta tags, or analytics to this file.
    The build step will place the bundled scripts into the <body> tag.

    To begin the development, run `npm start` or `yarn start`.
    To create a production bundle, use `npm run build` or `yarn build`.
  -->
</body>
</html>
```

### src\App.css

```
body{
  margin-left: 20px;
}
.texto{
  color: #0f0;
  font-size: 1em;
}
.caixa{
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
}
.caixa a{
  text-decoration: none;
  color: #f00;
}
.caixa a:hover{
  background-color: #f00;
  color: #fff;
}
```



## Aula 09 - Como funciona a renderização de componentes em React

Vamos aprender mais sobre como funciona a renderização de componentes.

### src\componentes\Relogio.js

```
import React from 'react'

export default function Relogio() {
  return(
    <p>{new Date().toLocaleTimeString()}</p>
  )
}
```

### src\App.js

```
import React from 'react'
import './App.css'
import Relogio from './componentes/Relogio'

export default function App() {

  const textoDestaque = {
    color: '#00f',
    fontSize: '3em'
  }

  return (
    <section className="caixa">
      <Relogio/>
      <h1 style={{color: '#f00', fontSize: '5em'}}>CFB Cursos</h1>
      <h2 style={textoDestaque}>Curso de React</h2>
      <p className="texto">Se inscreva em nosso canal e nos siga no Instagram</p>
      <a href="https://www.youtube.com/cfbcursos" target="_blank">CFB Cursos</a>
    </section>
  );
}
```

**src\index.js**

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
```

```
function tick(){
  ReactDOM.render(
    <React.StrictMode>
      <App />
    </React.StrictMode>,
    document.getElementById('root')
  );
}
```

```
setInterval(tick,1000);
```

```
// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```



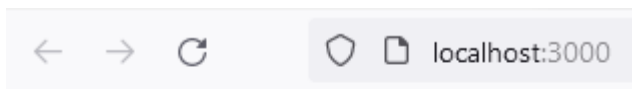
## Aula 10 - O que é State e como usar State em React

src\App.js

```
import React,{useState} from 'react'
import './App.css'

export default function App() {
  const [num, setNum] = useState(10)

  return (
    <section>
      <p>Valor do state num: {num}</p>
      <button onClick={()=>setNum(num+10)}>Soma 10</button>
    </section>
  );
}
```



Valor do state num: 10

Soma 10



Valor do state num: 20

Soma 10



### src\App.js

```
import React,{useState} from 'react'
import './App.css'
import Numero from './componentes/Numero'

export default function App() {
  const [num, setNum] = useState(10)
  const [nome, setNome] = useState('Bruno')

  return (
    <section>
      <p>Valor do state num em App: {num}</p>
      <Numero num={num} setNum={setNum}/>
      <p>{nome}</p>
    </section>
  );
}
```

### src\componentes\Numero.js

```
import React from 'react'

export default function Numero(props){
  return (
    <p>
      <p>Valor do state num em Numero: {props.num}</p>
      <button onClick={()=>props.setNum(props.num+10)}>Soma 10</button>
    </p>
  )
}
```

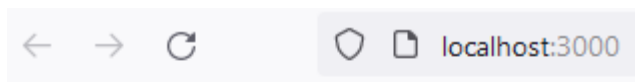


Valor do state num em App: 10

Valor do state num em Numero: 10

Soma 10

Bruno



Valor do state num em App: 20

Valor do state num em Numero: 20

Soma 10

Bruno

## Aula 11 - Eventos em React

Vamos aprender como trabalhar com eventos, além de passar funções para eventos de componentes.

### app01\src\App.js

```
import React, { useState } from 'react';
import './App.css';
import Led from './componentes/Led';

export default function App() {

  const [ligado, setLigado] = useState(false)

  const cancelar = (obj) => {
    return obj.preventDefault();
  };

  return (
    <>
      <Led ligado={ligado} setLigado={setLigado} />
      <a
        style={{display: "block"}}
        href="http://youtube.com/cfbcursos"
        target="_blank"
        onClick={(e) => cancelar(e)}
      >
        CFB Cursos
      </a>
    </>
  );
}
```

### app01\src\componentes\Led.js

```
import React from 'react';
import LedVerde from './img/led_verde.png';
import LedVermelho from './img/led_vermelho.png';

export default function Led(props) {

  return (
    <>
      <img style={{ width: '50px' }} src={props.ligado ? LedVerde : LedVermelho} alt="led vermelho" />
      <button onClick={() => props.setLigado(!props.ligado)}>
        {props.ligado ? "Desligar" : "Ligar"}
      </button>
    </>
  );
}
```



Ligar

[CFB Cursos](#)



Desligar

[CFB Cursos](#)

## Aula 12 - Como usar o recurso de Renderização condicional em React

Vamos aprender como usar o recurso de Renderização condicional. Com este recurso podemos indicar se um componentes será mostrado/renderizado ou não, além de outras possibilidades.

### app01\src\App.js

```
import React, { useState } from 'react';
import './App.css';
```

```
export default function App() {
```

```
  const [log, setLog] = useState(false)
  const [cor, setCor] = useState(1)
```

```
  const vermelho = { color: "#f00" }
  const verde = { color: "#0f0" }
  const azul = { color: "#00f" }
```

```
  const mudaCor = () => {
    setCor(cor+1)
    if(cor > 2){
      setCor(1)
    }
  }
}
```

```
  const retCor = (c) => {
    if(c === 1){
      return vermelho
    } else if (c === 2) {
      return verde
    } else {
      return azul
    }
  }
}
```

```
  const msglogin = () => {
    return "Usuário logado!"
  }
```

```
  const msglogoff = () => {
    return "Favor logar!"
  }
```

```
  const cumprimento = () => {
    const hora = new Date().getHours()

    if(hora >= 0 && hora < 13){
      return <p>Bom dia!</p>
    } else if (hora >= 13 && hora < 18) {
      return <p>Boa tarde!</p>
    } else {
      return <p>Boa noite!</p>
    }
  }
}
```

```
return (  
  <>  
    <h1 style={retCor(cor)}>CFB Cursos</h1>  
    {cumprimento()}  
    <p>{log ? msglogin() : msglogout()}</p>  
    <button onClick={() => setLog(!log)}>{log ? "Logout" : "Login"}</button>  
    <button onClick={() => mudaCor()}>Muda cor</button>  
  </>  
)  
};  
}
```

## CFB Cursos

Bom dia!

Favor logar!

Login Muda cor

## CFB Cursos

Bom dia!

Usuário logado!

Logout Muda cor

## CFB Cursos

Bom dia!

Usuário logado!

Logout Muda cor

# CFB Cursos

Bom dia!

Usuário logado!

Logoff

Muda cor

## Aula 13 - Trabalhando com LISTA em React usando a função MAP

- A função map() permite percorrer os elementos de um array e executar uma ação para cada elemento.

app01\src\App.js

```
import React from 'react';
import './App.css';

export default function App() {

  const carros1 = ["HRV", "Golf", "Focus", "Cruze", "Argo"]

  const carros2 = [
    { categoria: "Esporte", preco: 110000.00, modelo: "Golf GTI" },
    { categoria: "Esporte", preco: 120000.00, modelo: "Camaro" },
    { categoria: "SUV", preco: 85000.00, modelo: "HRV" },
    { categoria: "SUV", preco: 83000.00, modelo: "T-Cross" },
    { categoria: "Utilitário", preco: 120000.00, modelo: "Hillux" },
    { categoria: "Utilitário", preco: 90000.00, modelo: "Ranger" },
  ]

  const listaCarros1 = carros1.map((elemento, index) => <li key={index}>{elemento}</li>)
  const listaCarros2 = carros2.map((elemento, index) =>
    <li key={index}>
      Modelo: {elemento.modelo} - Categoria: {elemento.categoria} - Preço: R${elemento.preco}
    </li>)

  return (
    <>
      <h1>Lista de carros - 1</h1>
      <ol>
        {listaCarros1}
      </ol>

      <h1>Lista de carros - 2</h1>
      <ol>
        {listaCarros2}
      </ol>
    </>
  );
}
```



## Lista de carros - 1

1. HRV
2. Golf
3. Focus
4. Cruze
5. Argo

## Lista de carros - 2

1. Modelo: Golf GTI - Categoria: Esporte - Preço: R\$110000
2. Modelo: Camaro - Categoria: Esporte - Preço: R\$120000
3. Modelo: HRV - Categoria: SUV - Preço: R\$85000
4. Modelo: T-Cross - Categoria: SUV - Preço: R\$83000
5. Modelo: Hillux - Categoria: Utilitário - Preço: R\$120000
6. Modelo: Ranger - Categoria: Utilitário - Preço: R\$90000

## Aula 14 - Manipulando elementos de formulário com React

### Input

app01\src\App.js

```
import React, { useState } from 'react';
import './App.css';

export default function App() {

  const [nome, setNome] = useState("")

  return (
    <>
      <form action="">

        <label htmlFor="fname">Digite seu nome</label>
        <input
          type="text"
          name="fname"
          value={nome}
          onChange={(e) => setNome(e.target.value)}
        />
        <p>Nome digitado: {nome}</p>

      </form>
    </>
  );
}
```

← → ↻ ⓘ localhost:3000

Digite seu nome

Nome digitado: Bruno P. Campos

## Usando manipuladores

app01\src\App.js

```
import React, { useState } from 'react';
import './App.css';

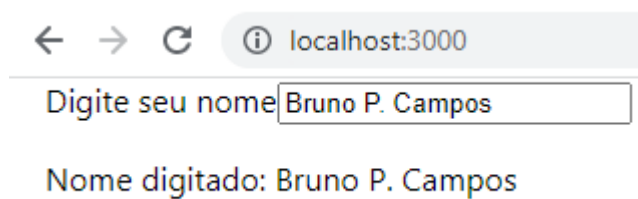
export default function App() {

  const [nome, setNome] = useState("")
  const handleChangeNome = (e) => {
    setNome(e.target.value)
  }

  return (
    <>
      <form action="">

        <label htmlFor="fnome">Digite seu nome</label>
        <input
          type="text"
          name="fnome"
          value={nome}
          onChange={(e)=>handleChangeNome(e)}
        />
        <p>Nome digitado: {nome}</p>

      </form>
    </>
  );
}
```



## Select

### app01\src\App.js

```
import React, { useState } from 'react';
import './App.css';

export default function App() {

  const [nome, setNome] = useState("")
  const [carro, setCarro] = useState('HRV')

  const handleChangeNome = (e) => {
    setNome(e.target.value)
  }

  return (
    <>
      <form action="">

        <label htmlFor="fnome">Digite seu nome</label>
        <input
          type="text"
          name="fnome"
          value={nome}
          onChange={(e)=>handleChangeNome(e)}
        />
        <p>Nome digitado: {nome}</p>

        <label htmlFor={carro}>Selecione um carro</label>
        <select value={carro} onChange={(e)=>setCarro(e.target.value)}>
          <option value="HRV">HRV</option>
          <option value="Golf">Golf</option>
          <option value="Cruze">Cruze</option>
          <option value="Argo">Argo</option>
        </select>
        <p>Carro selecionado: {carro}</p>

      </form>
    </>
  );
}
```

← → ↻ ⓘ localhost:3000

Digite seu nome

Nome digitado:

Selecione um carro

Carro selecionado: HRV

← → ↻ ⓘ localhost:3000

Digite seu nome

Nome digitado:

Selecione um carro

Carro selecionado: Golf

## Aula 15 - Manipulando objetos com um state para vários componentes

app01\src\App.js

```
import React, { useState } from 'react';
import './App.css';

export default function App() {

  const [form, setForm] = useState({ "nome": "", "curso": "", "ano": "" })

  const handleFormChange = (e) => {
    if(e.target.getAttribute('name') === "fname"){
      setForm({ "nome": e.target.value, "curso": form.curso, "ano": form.ano })
    } else if (e.target.getAttribute('name') === "fcurso") {
      setForm({ "nome": form.nome, "curso": e.target.value, "ano": form.ano })
    } else if (e.target.getAttribute('name') === "fano") {
      setForm({ "nome": form.nome, "curso": form.curso, "ano": e.target.value })
    }
  }

  return (
    <>
      <form action="">

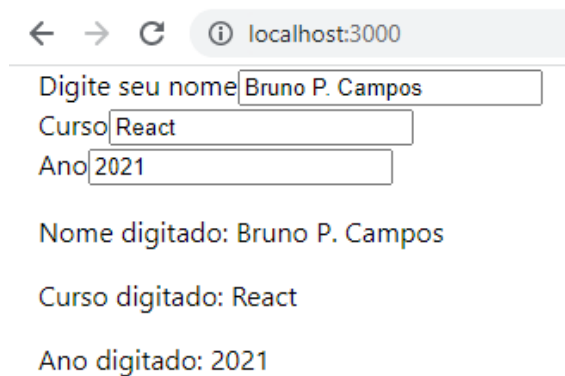
        <label htmlFor="fname">Digite seu nome</label>
        <input type="text" name="fname" value={form.nome} onChange={(e) => handleFormChange(e)} /><br />

        <label htmlFor="fcurso">Curso</label>
        <input type="text" name="fcurso" value={form.curso} onChange={(e) => handleFormChange(e)} /><br />

        <label htmlFor="fano">Ano</label>
        <input type="text" name="fano" value={form.ano} onChange={(e) => handleFormChange(e)} /><br />

        <p>Nome digitado: {form.nome}</p>
        <p>Curso digitado: {form.curso}</p>
        <p>Ano digitado: {form.ano}</p>

      </form>
    </>
  );
}
```



← → ↻ ⓘ localhost:3000

Digite seu nome Bruno P. Campos

Curso React

Ano 2021

Nome digitado: Bruno P. Campos

Curso digitado: React

Ano digitado: 2021

## Aula 16 - Elevação de state

### app01\src\componentes\Nota.js

```
import React from 'react'

function Nota(props){
  return(
    <div>
      <legend>Informe a nota {props.num}</legend>
      <input type="text" value={props.nota} onChange={(e) => props.setNota(e.target.value)} />
    </div>
  )
}
export default Nota
```

### app01\src\componentes\Resultado.js

```
import React from 'react'

function Resultado(props) {
  return (
    <div>
      <p>Soma das notas: {props.somaNotas}</p>
      <p>{props.somaNotas >= 24 ? "Aprovado" : "Reprovado"}</p>
    </div>
  )
}
export default Resultado
```

### app01\src\App.js

```
import React, { useState } from 'react';
import './App.css';
import Nota from './componentes/Nota';
import Resultado from './componentes/Resultado';

export default function App() {
  const [nota1, setNota1] = useState(0)
  const [nota2, setNota2] = useState(0)
  const [nota3, setNota3] = useState(0)
  const [nota4, setNota4] = useState(0)

  return (
    <>
      <Nota num={1} nota={nota1} setNota={setNota1} />
      <Nota num={2} nota={nota2} setNota={setNota2} />
      <Nota num={3} nota={nota3} setNota={setNota3} />
      <Nota num={4} nota={nota4} setNota={setNota4} />
      <Resultado somaNotas={parseFloat(nota1)+parseFloat(nota2)+parseFloat(nota3)+parseFloat(nota4)} />
    </>
  );
}
```

Informe a nota 1

Informe a nota 2

Informe a nota 3

Informe a nota 4

Soma das notas: 0

Reprovado

Informe a nota 1

Informe a nota 2

Informe a nota 3

Informe a nota 4

Soma das notas: 22

Reprovado

Informe a nota 1

Informe a nota 2

Informe a nota 3

Informe a nota 4

Soma das notas: 25

Aprovado



## Aula 17 - Solução do desafio da aula 16 sobre elevação de state

### app01\src\componentes\Nota.js

```
import React from 'react'

function Nota(props){
  return(
    <div>
      <legend>Informe a nota {props.num}</legend>
      <input type="text" name={props.nome} value={props.nota} onChange={(e) => props.setNota(e)} />
    </div>
  )
}

export default Nota
```

### app01\src\App.js

```
import React, { useState } from 'react';
import './App.css';
import Nota from './componentes/Nota';
import Resultado from './componentes/Resultado';

export default function App() {

  const [notas, setNotas] = useState({ "nota1": "0", "nota2": "0", "nota3": "0", "nota4": "0" })

  const handleSetNotas = (e) => {
    if (e.target.getAttribute("name") === "nota1") {
      setNotas({ "nota1": e.target.value, "nota2": notas.nota2, "nota3": notas.nota3, "nota4": notas.nota4 })
    } else if (e.target.getAttribute("name") === "nota2") {
      setNotas({ "nota1": notas.nota1, "nota2": e.target.value, "nota3": notas.nota3, "nota4": notas.nota4 })
    } else if (e.target.getAttribute("name") === "nota3") {
      setNotas({ "nota1": notas.nota1, "nota2": notas.nota2, "nota3": e.target.value, "nota4": notas.nota4 })
    } else if (e.target.getAttribute("name") === "nota4") {
      setNotas({ "nota1": notas.nota1, "nota2": notas.nota2, "nota3": notas.nota3, "nota4": e.target.value })
    }
  }

  return (
    <>
      <Nota nome={'nota1'} num={1} nota={notas.nota1} setNota={handleSetNotas} />
      <Nota nome={'nota2'} num={2} nota={notas.nota2} setNota={handleSetNotas} />
      <Nota nome={'nota3'} num={3} nota={notas.nota3} setNota={handleSetNotas} />
      <Nota nome={'nota4'} num={4} nota={notas.nota4} setNota={handleSetNotas} />
      <Resultado
        somaNotas={parseFloat(notas.nota1)+parseFloat(notas.nota2)+parseFloat(notas.nota3)+parseFloat(notas.nota4)} />
    </>
  );
}
```

← → ↻ ⓘ localhost:3000

Informe a nota 1

Informe a nota 2

Informe a nota 3

Informe a nota 4

Soma das notas: 0

Reprovado

← → ↻ ⓘ localhost:3000

Informe a nota 1

Informe a nota 2

Informe a nota 3

Informe a nota 4

Soma das notas: 22

Reprovado

← → ↻ ⓘ localhost:3000

Informe a nota 1

Informe a nota 2

Informe a nota 3

Informe a nota 4

Soma das notas: 25

Aprovado

## Aula 18 - Aprendendo sobre CONTENÇÃO em React

### app01\src\componentes\Canal.js

```
import React from 'react'
```

```
function Canal(){  
  return(  
    <>  
      <h1>CFB Cursos</h1>  
    </>  
  )  
}
```

```
export default Canal
```

### app01\src\componentes\Caixa.js

```
import React from 'react'
```

```
function Caixa(props){  
  return(  
    <>  
      <p>{props.site}</p>  
      {props.children}  
      {props.children[0]}  
      {props.children[1]}  
    </>  
  )  
}
```

```
export default Caixa
```

### app01\src\App.js

```
import React from 'react';
```

```
import './App.css';
```

```
import Caixa from './componentes/Caixa';
```

```
import Canal from './componentes/Canal';
```

```
export default function App() {
```

```
  return (  
    <>  
      <div>  
        <Caixa site="www.cfbcursos.com.br">  
          <Canal />  
          <p>Curso de React</p>  
        </Caixa>  
      </div>  
    </>  
  );  
}
```

← → ↻ ⓘ localhost:3000

www.cfbcursos.com.br

# CFB Cursos

Curso de React

# CFB Cursos

Curso de React

## Aula 19 - Aprendendo sobre o Hook useEffect

app01\src\App.js

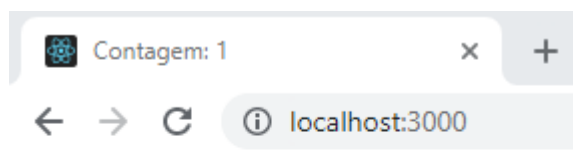
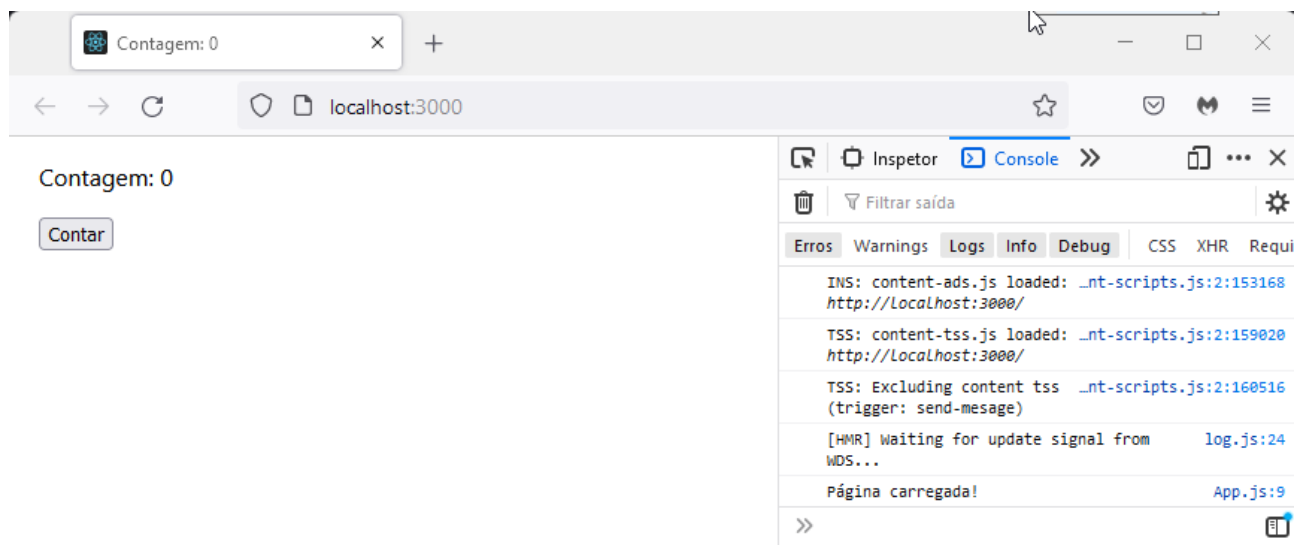
```
import React, { useEffect, useState } from 'react';
import './App.css';

export default function App() {

  const [contagem, setContagem] = useState(0)

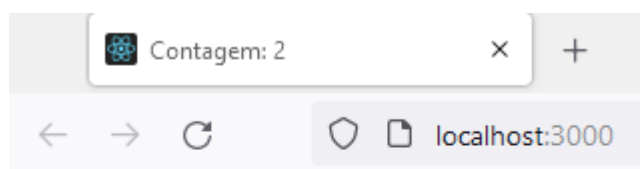
  useEffect(() => {
    console.log("Página carregada!")
    document.title = "Contagem: " + contagem
  }, [contagem])

  return (
    <>
      <p>Contagem: {contagem}</p>
      <button onClick={()=>setContagem(contagem+1)}>Contar</button>
    </>
  );
}
```



Contagem: 1

Contar



Contagem: 2

Contar

## Aula 20 - LocalStorage em React

```
/* Criando a chave nome */
localStorage.setItem("nome", "Bruno")

/* Trocando o valor da chave */
localStorage.setItem("nome", "Theo")

/* Recuperando o valor da chave nome */
localStorage.getItem("nome")

/* Removendo a chave nome */
localStorage.removeItem("nome")
```

### app01\src\App.css

```
body{
  margin-left: 20px;
}
.texto{
  color: #0f0;
  font-size: 1em;
}
.caixa{
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
}
.caixa a{
  text-decoration: none;
  color: #f00;
}
.caixa a:hover{
  background-color: #f00;
  color: #fff;
}
button{
  margin: 5px 10px 5px 0px;
  padding: 5px 15px;
}
input[type="text"], label {
  margin: 5px 10px 5px 0px;
}
```

## app01\src\App.js

```
import React, { useState } from 'react';
import './App.css';

export default function App() {

  const [nome, setNome] = useState()

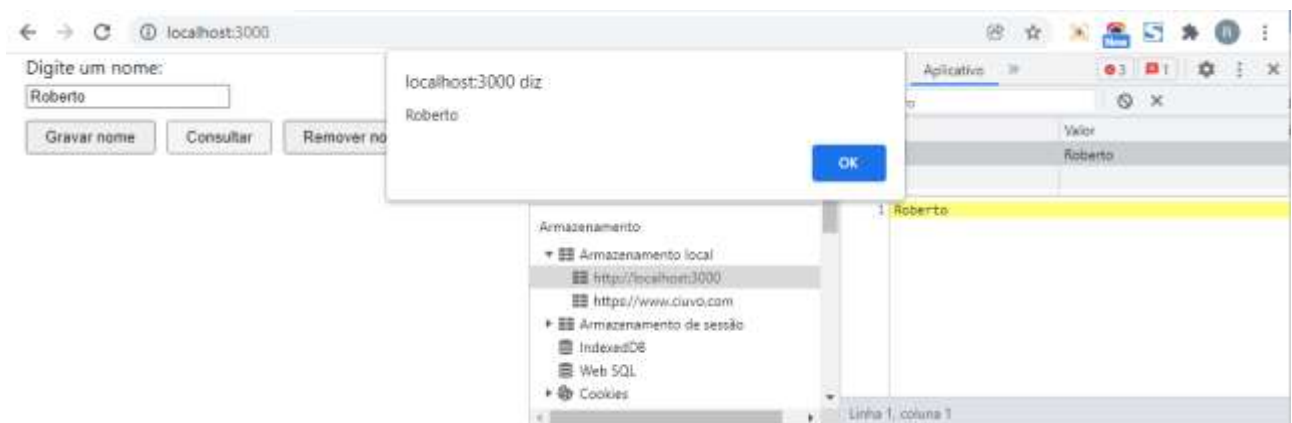
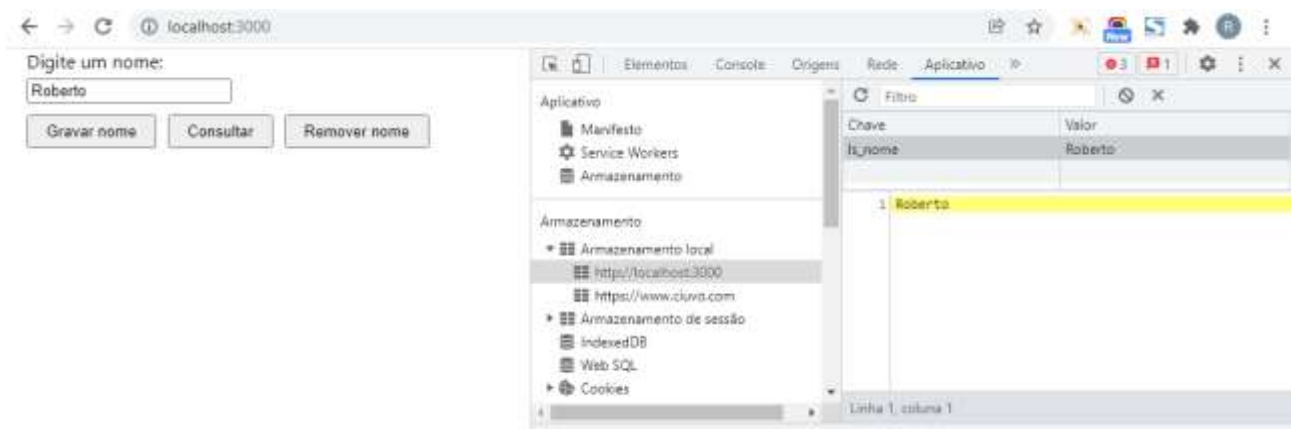
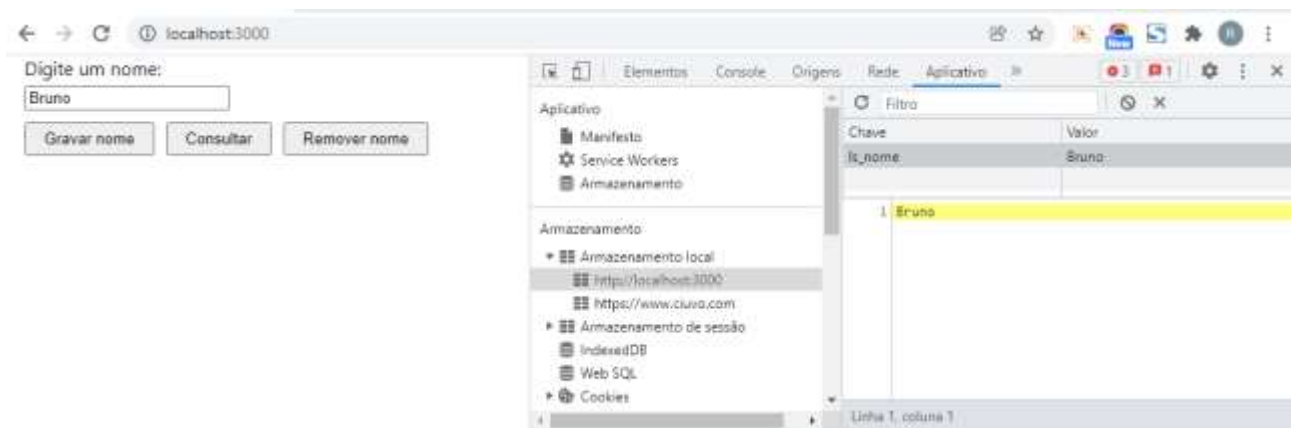
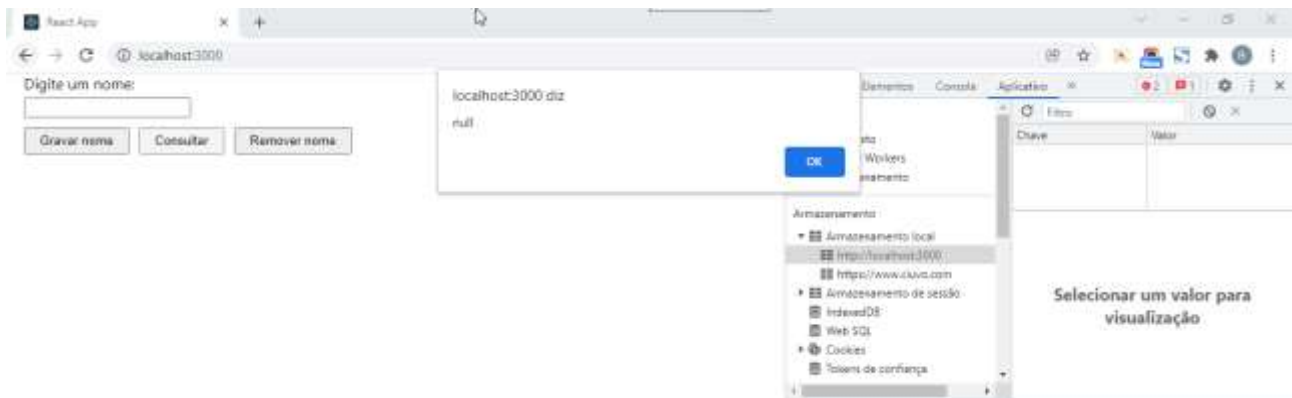
  const armazenar = (chave, valor) => {
    localStorage.setItem(chave, valor)
    alert("Nome armazenado com sucesso!")
  }

  const consultar = (chave) => {
    alert(localStorage.getItem(chave))
  }

  const apagar = (chave) => {
    localStorage.removeItem(chave)
    alert("Nome removido com sucesso!")
  }

  return (
    <>
      <label htmlFor="nome">Digite um nome: </label><br />
      <input type="text" name="nome" value={nome} onChange={(e) => setNome(e.target.value)} /><br />
      <button onClick={() => armazenar("ls_nome", nome)}>Gravar nome</button>
      <button onClick={() => consultar("ls_nome")}>Consultar</button>
      <button onClick={() => apagar("ls_nome")}>Remover nome</button>
    </>
  );
}
```





← → ↻ localhost:3000

Digite um nome:

Roberto

Gravar nome Consultar Remover nome

Aplicativo

Manifesto  
Service Workers  
Armazenamento

Armazenamento

Armazenamento local

- http://localhost:3000
- https://www.ciuvo.com

Armazenamento de sessão

- IndexedDB
- Web SQL

Cookies

Filtro

Chave	Valor
-------	-------

Selecionar um valor para visualização

## Aula 21 - Criando Componentes de Classe em React

**app01\src\componentes\Classe.js**

```
import React from 'react'

class Classe extends React.Component {

  constructor(props){
    super(props)
  }

  render(){
    return(
      <>
        <h1>Primeiro Componente de Classe</h1>
        <p>Canal: {this.props.canal } </p>
        <p>Curso: {this.props.curso}</p>
      </>
    )
  }
}

export default Classe
```

**app01\src\App.js**

```
import React from 'react';
import './App.css';
import Classe from './componentes/Classe';

export default function App() {

  return (
    <>
      <h1>Componentes de classe</h1>
      <Classe canal="CFB Cursos" curso="React" />
    </>
  );
}
```

# Componentes de classe

## Primeiro Componente de Classe

Canal: CFB Cursos

Curso: React

## Aula 22 - STATE em Componentes de Classe ReactJS

### app01\src\componentes\Carro.js

```
import React from 'react'

class Carro extends React.Component {

  constructor(props) {
    super(props)
    this.modelo = 'Golf'
    this.state = {
      ligado: false,
      velAtual: 0
    }
  }

  ligarDesligar(){
    this.setState({ligado: !this.state.ligado})
  }

  render() {
    return (
      <>
        <h1>Meu carro</h1>
        <p>Modelo: {this.modelo}</p>
        <p>Ligado: {this.state.ligado ? "Sim" : "Não"}</p>
        <p>Velocidade atual: {this.state.velAtual}</p>
        <button onClick={() => this.ligarDesligar()}>{this.state.ligado ? "Desligar carro" : "Ligar carro"}</button>
      </>
    )
  }
}

export default Carro
```

### app01\src\App.js

```
import React from 'react';
import './App.css';
import Carro from './componentes/Carro';

export default function App() {

  return (
    <>
      <h1>Componentes de classe</h1>
      <Carro />
    </>
  );
}
```

## Componentes de classe

### Meu carro

Modelo: Golf

Ligado: Não

Velocidade atual: 0

Ligar carro

## Componentes de classe

### Meu carro

Modelo: Golf

Ligado: Sim

Velocidade atual: 0

Desligar carro

## Aula 23 - Manipulando STATE em Componentes de Classe ReactJS

### app01\src\App.js

```
import React from 'react';
import './App.css';
import Carro from './componentes/Carro';

export default function App() {

  return (
    <>
      <h1>Componentes de classe</h1>
      <Carro fator={10} />
    </>
  );
}
```

### app01\src\componentes\Carro.js

```
import React from 'react'

class Carro extends React.Component {

  constructor(props) {
    super(props)
    this.modelo = 'Golf'
    this.state = {
      ligado: false,
      velAtual: 0
    }
  }

  ligarDesligar(){
    // this.setState({ligado: !this.state.ligado})

    // this.setState(
    //   function (state) {
    //     return {
    //       ligado: !state.ligado
    //     }
    //   }
    // )

    this.setState(
      (state) => (
        {
          ligado: !state.ligado
        }
      )
    )
  }
}
```

```

    }

    // acelerar() {
    //   this.setState(
    //     (
    //       (state,props) => (
    //         {velAtual: this.state.velAtual + this.props.fator}
    //       )
    //     )
    //   )
    // }

    acelerar() {
      this.setState(
        (state, props) => (
          { velAtual: state.velAtual + props.fator }
        )
      )
    }

    render() {
      return (
        <>
          <h1>Meu carro</h1>
          <p>Modelo: {this.modelo}</p>
          <p>Ligado: {this.state.ligado ? "Sim" : "Não"}</p>
          <p>Velocidade atual: {this.state.velAtual}</p>
          <button onClick={() => this.ligarDesligar()}>{this.state.ligado ? "Desligar carro" : "Ligar carro"}</button>
          <button onClick={() => this.acelerar()}>Acelerar</button>
        </>
      )
    }
  }
}

```

export default Carro

← → ↻ ⓘ localhost:3000

## Componentes de classe

### Meu carro

Modelo: Golf

Ligado: Não

Velocidade atual: 0

Ligar carro

Acelerar





localhost:3000

## Componentes de classe

### Meu carro

Modelo: Golf

Ligado: Não

Velocidade atual: 10

Ligar carro

Acelerar



localhost:3000

## Componentes de classe

### Meu carro

Modelo: Golf

Ligado: Não

Velocidade atual: 20

Ligar carro

Acelerar

## Aula 24 - Como usar o método BIND em React

app01\src\componentes\Carro.js

```
import React from 'react'
```

```
class Carro extends React.Component {
```

```
  constructor(props) {  
    super(props)  
    this.modelo = 'Golf'  
    this.state = {  
      ligado: false,  
      velAtual: 0  
    }  
    this.ligarDesligar.bind(this)  
  }
```

```
  ligarDesligar(){  
    this.setState({ligado: !this.state.ligado})  
  }
```

```
  acelerar() {  
    this.setState(  
      (state, props) => (  
        { velAtual: state.velAtual + props.fator }  
      )  
    )  
  }
```

```
  render() {  
    return (  
      <>  
        <h1>Meu carro</h1>  
        <p>Modelo: {this.modelo}</p>  
        <p>Ligado: {this.state.ligado ? "Sim" : "Não"}</p>  
        <p>Velocidade atual: {this.state.velAtual}</p>  
        <button onClick={this.ligarDesligar}>{this.state.ligado ? "Desligar carro" : "Ligar carro"}</button>  
        <button onClick={() => this.acelerar()}>Acelerar</button>  
      </>  
    )  
  }  
}
```

```
export default Carro
```

## Componentes de classe

### Meu carro

Modelo: Golf

Ligado: Não

Velocidade atual: 0

Ligar carro

## Componentes de classe

### Meu carro

Modelo: Golf

Ligado: Sim

Velocidade atual: 0

Desligar carro

## Aula 25 - Ciclo de Vida dos componentes em REACT

### Montagem de componente

Na montagem, quando uma instância do componente está sendo criado no DOM.

1. constructor()
2. static getDerivedStateFromProps()
3. render()
4. componentDidMount()

### Atualização de componente

Na atualização do componente, por alterações causadas em props ou state, quando o componente está sendo re-renderizado.

1. static getDerivedStateFromProps()
2. shouldComponentUpdate()
3. render()
4. getSnapshotBeforeUpdate()
5. componentDidUpdate()

### Desmontagem de componente

1. componentWillUnmount()

### Tratamento de erros

Em tratamento de erros, quando existir algum erro em algum momento no componente.

1. static getDerivedStateFromError()
2. componentDidCatch()

### app01\src\componentes\Carro.js

```
import React from 'react'
```

```
class Carro extends React.Component {
```

```
  constructor(props) {
    super(props)
    this.modelo = 'Golf'
    this.state = {
      ligado: false,
      velAtual: 0
    }
    this.ligarDesligar=this.ligarDesligar.bind(this)
  }
```

```
ligarDesligar(){
  this.setState({ligado: !this.state.ligado})
}

acelerar() {
  this.setState(
    (state, props) => (
      { velAtual: state.velAtual + props.fator }
    )
  )
}

componentDidMount() {
  console.log("O carro foi criado!")
}

// componentDidUpdate() {
//   console.log("O carro foi atualizado!")
// }

componentWillUnmount() {
  console.log("O carro foi removido!")
}

render() {
  return (
    <>
      <h1>Meu carro</h1>
      <p>Modelo: {this.modelo}</p>
      <p>Ligado: {this.state.ligado ? "Sim" : "Não"}</p>
      <p>Velocidade atual: {this.state.velAtual}</p>
      <button onClick={this.ligarDesligar}>{this.state.ligado ? "Desligar carro" : "Ligar carro"}</button>
      <button onClick={() => this.acelerar()}>Acelerar</button>
    </>
  )
}

export default Carro
```

## app01\src\App.js

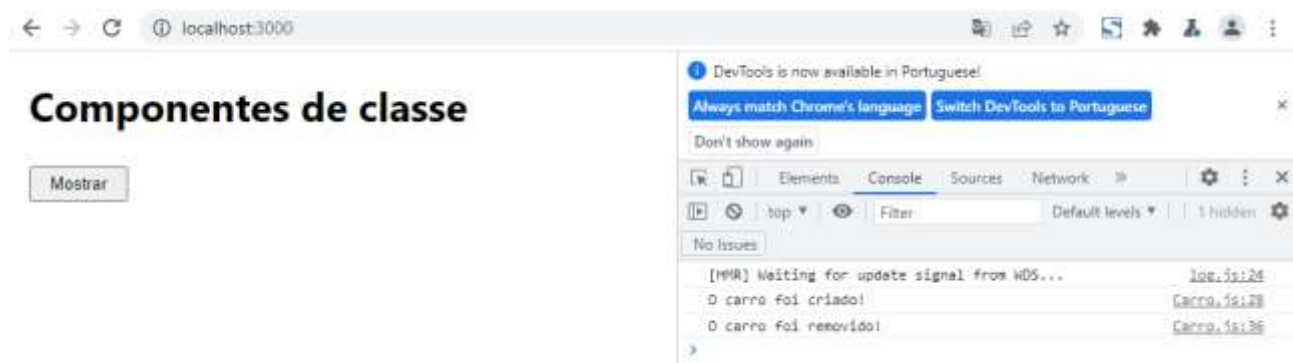
```
import React, { useState } from 'react';
import './App.css';
import Carro from './componentes/Carro';

export default function App() {

  const [carro, setCarro] = useState(true)

  const mostrarOcultarCarro = () => {
    setCarro(!carro)
  }

  return (
    <>
      <h1>Componentes de classe</h1>
      {carro ? <Carro fator={10} /> : ""}
      <button onClick={() => mostrarOcultarCarro()}>{carro ? "Ocultar" : "Mostrar"}</button>
    </>
  );
}
```



## Aula 26 - Estrutura de um componente de CLASSE em React

app01\src\componentes\BaseClasse.js

```
import React from "react";

export default class BaseClasse extends React.Component {
  constructor(props) {
    //Para permitir o uso de props
    super(props)

    //state
    this.state = {
      canal: 'CFB Cursos',
      curso: 'React',
      ativo: true,
      nome: this.props.nomeAluno
    }
    this.status = this.props.status

    //bindagem
    this.ad = this.ativarDesativar.bind(this)

    //Instruções do construtor
  }

  //Função para manipular state
  ativarDesativar() {
    this.setState(
      state => ({
        ativo: !state.ativo
      })
    )
  }

  componentDidMount() {
    console.log('O componente foi criado')
  }

  // componentDidMount() {
  //   console.log('O componente foi atualizado')
  // }

  componentWillUnmount() {
    console.log('O componente foi removido')
  }

  render() {
    return (
      <>
        <h1>Componente de Classe</h1>
        <button onClick={this.state.ad}>Ativar/Desativar</button>
        <button onClick={() => this.state.ativarDesativar}>Ativar/Desativar</button>
      </>
    )
  }
}
```

## Aula 27 - Variáveis Globais Static em React

**app01\src\componentes\Globais.js**

```
export default class Globais{
  static canal = "CFB Cursos"
  static curso = "Curso de React"
  static ano = 2021
  static resumo = "Curso de React do canal CFB Cursos"
}
```

**app01\src\componentes\Info.js**

```
import React from 'react';
import Globais from './Globais';

export default function Info() {

  return (
    <>
      <p>{'Canal: ' + Globais.canal}</p>
      <p>{'Curso: ' + Globais.curso}</p>
      <p>{'Ano: ' + Globais.ano}</p>
    </>
  );
}
```



## app01\src\App.js

```
import React, { useState } from 'react';
import './App.css';
import Globais from './componentes/Globais';
import Info from './componentes/Info';


export default function App() {

  const [resumo,setResumo]=useState(Globais.resumo)

  const gravarResumo=()=>{
    Globais.resumo=resumo;
  }

  const verResumo=()=>{
    alert(Globais.resumo)
  }

  return (
    <>
      <Info />
      <hr />
      <p>{'Canal: ' + Globais.canal}</p>
      <p>{'Curso: ' + Globais.curso}</p>
      <p>{'Ano: ' + Globais.ano}</p>
      <hr />
      <input size="100" type="text" value={resumo} onChange={(e)=>setResumo(e.target.value)} />
      <br />
      <button onClick={() => gravarResumo()}>Gravar resumo</button>
      <button onClick={() => verResumo()}>Ver resumo</button>
    </>
  );
}
```

← → ↺ ⓘ localhost:3000 

---

Canal: CFB Cursos

Curso: Curso de React

Ano: 2021

---

Canal: CFB Cursos

Curso: Curso de React

Ano: 2021

---

Curso de React do canal CFB Cursos

Gravar resumoVer resumo

- Clique no botão "Ver resumo"

The screenshot shows a web browser window with the address bar at localhost:3000. The page contains a form with three input fields: "Canal: CFB Cursos", "Curso: Curso de React", and "Ano: 2021". A modal dialog is open, displaying the text "localhost:3000 diz" and "Curso de React do canal CFB Cursos", with an "OK" button.

Canal: CFB Cursos

Curso: Curso de React

Ano: 2021

localhost:3000 diz

Curso de React do canal CFB Cursos

OK

Canal: CFB Cursos

Curso: Curso de React

Ano: 2021

Curso de React do canal CFB Cursos

Gravar resumo Ver resumo

- Altere o texto no input:

The screenshot shows the same web browser window, but the input field for the course name now contains the text "Curso de React do canal CFB Cursos, ano 2021".

Canal: CFB Cursos

Curso: Curso de React

Ano: 2021

Canal: CFB Cursos

Curso: Curso de React

Ano: 2021

Curso de React do canal CFB Cursos, ano 2021

Gravar resumo Ver resumo

- Clique no botão "Gravar resumo" e em seguida clique no botão "Ver resumo";

The screenshot shows the web browser window after clicking "Gravar resumo" and "Ver resumo". The input field now contains "Curso de React do canal CFB Cursos, ano 2021". A modal dialog is open, displaying the text "localhost:3000 diz" and "Curso de React do canal CFB Cursos, ano 2021", with an "OK" button.

Canal: CFB Cursos

Curso: Curso de React

Ano: 2021

localhost:3000 diz

Curso de React do canal CFB Cursos, ano 2021

OK

Canal: CFB Cursos

Curso: Curso de React

Ano: 2021

Curso de React do canal CFB Cursos, ano 2021

Gravar resumo Ver resumo

**Observação:** Se você alterar o texto no campo input e em seguida clicar no botão "Ver resumo", o alert não exibirá essa alteração. Para que isso aconteça é necessário primeiro clicar no botão "Gravar resumo"

## Aula 28 - Navegando entre telas de forma nativa em React

**app01\src\componentes\Pagina1.js**

```
import React from 'react';

export default function Pagina1() {

  return (
    <div>
      <h1>Página 1</h1>
      <h3>Curso de React</h3>
    </div>
  );
}
```

**app01\src\componentes\Pagina2.js**

```
import React from 'react';

export default function Pagina2() {

  return (
    <div>
      <h1>Página 2</h1>
      <h3>CFB Cursos</h3>
    </div>
  );
}
```

## app01\src\App.js

```
import React, { useEffect, useState } from 'react';
import './App.css';
import Pagina1 from './componentes/Pagina1';
import Pagina2 from './componentes/Pagina2';

export default function App() {

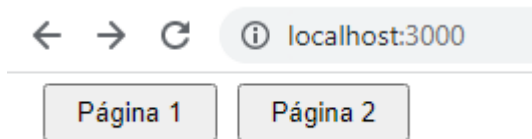
  const [pagina, setPagina] = useState(0)

  useEffect(
    () => {
      const url=window.location.href
      const res=url.split('?')
      setPagina(res[1])
    }, [pagina]
  )

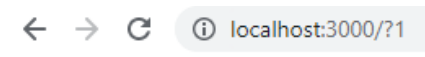
  const LinksPaginas=(p)=>{
    if(p===1){
      window.open('http://localhost:3000?1', '_self')
    } else if(p===2){
      window.open('http://localhost:3000?2', '_self')
    }
  }

  const retornarPagina=()=>{
    if(pagina==1){
      return <Pagina1 />
    } else if (pagina==2) {
      return <Pagina2 />
    } else {
      return <div>
        <button onClick={()=>LinksPaginas(1)}>Página 1</button>
        <button onClick={()=>LinksPaginas(2)}>Página 2</button>
      </div>
    }
  }

  return (
    <>
      {retornarPagina()}
    </>
  );
}
```



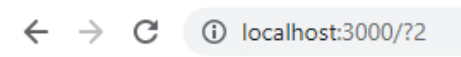
- Clique no botão "Página 1"



# Página 1

**Curso de React**

- Volte para a página anterior e clique no botão "Página 2"



# Página 2

**CFB Cursos**

## Aula 29 - Que tal praticar um pouco de React [Exercício de React]

app01\src\App.js

```
import React, { useState } from 'react';
import './App.css';

const carros = [
  { categoria: "Esporte", preco: "110000", modelo: "Golf GTI" },
  { categoria: "Esporte", preco: "120000", modelo: "Camaro" },
  { categoria: "SUV", preco: "85000", modelo: "HRV" },
  { categoria: "SUV", preco: "83000", modelo: "T-Cross" },
  { categoria: "Utilitário", preco: "125000", modelo: "Hilux" },
  { categoria: "Utilitário", preco: "90000", modelo: "Ranger" }
]

const linhas = (cat) => {
  const li = []
  carros.forEach(
    (carro) => {
      if(carro.categoria.toUpperCase() === cat.toUpperCase() || cat === ""){
        li.push(
          <tr>
            <td>{carro.categoria}</td>
            <td>{carro.preco}</td>
            <td>{carro.modelo}</td>
          </tr>
        )
      }
    }
  )
  return li
}

const TabelaCarros = (cat) => {
  return(
    <table border='1' style={{borderCollapse:'collapse'}}>
      <thead>
        <tr>
          <th>Categoria</th>
          <th>Preço</th>
          <th>Modelo</th>
        </tr>
      </thead>
      <tbody>
        {linhas(cat)}
      </tbody>
    </table>
  )
}
```

```

const pesquisaCategoria = (cat, scat) => {
  return(
    <div>
      <label>Digite uma categoria</label>
      <input type="text" value={cat} onChange={(e)=>scat(e.target.value)} />
    </div>
  )
}

export default function App() {
  const [categoria, setCategoria] = useState("")
  return (
    <>
      {pesquisaCategoria(categoria, setCategoria)}
      <br />
      {TabelaCarros(categoria)}
    </>
  );
}

```

← → ↻ ⓘ localhost:3000

Digite uma categoria

Categoria	Preço	Modelo
Esporte	110000	Golf GTI
Esporte	120000	Camaro
SUV	85000	HRV
SUV	83000	T-Cross
Utilitário	125000	Hilux
Utilitário	90000	Ranger

← → ↻ ⓘ localhost:3000

Digite uma categoria

Categoria	Preço	Modelo
SUV	85000	HRV
SUV	83000	T-Cross



← → ↻ ⓘ localhost:3000

Digite uma categoria

<b>Categoria</b>	<b>Preço</b>	<b>Modelo</b>
Esporte	110000	Golf GTI
Esporte	120000	Camaro

← → ↻ ⓘ localhost:3000

Digite uma categoria

<b>Categoria</b>	<b>Preço</b>	<b>Modelo</b>
Utilitário	125000	Hilux
Utilitário	90000	Ranger

## Aula 30 - Como criar calculadora de IMC em React

app01\src\App.js

```
import React, { useState } from 'react';
import './App.css';
```

```
const tabelaIMC = () => {
  return(
    <table border='1' style={{ borderCollapse: 'collapse' }}>
      <thead>
        <tr>
          <th>Classificação</th>
          <th>IMC</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>Abaixo do peso</td>
          <td>Abaixo de 18,5</td>
        </tr>
        <tr>
          <td>Peso normal</td>
          <td>Entre 18,5 e 24,9</td>
        </tr>
        <tr>
          <td>Sobrepeso</td>
          <td>Entre 25 e 29,9</td>
        </tr>
        <tr>
          <td>Obesidade Grau I</td>
          <td>Entre 30 e 34,9</td>
        </tr>
        <tr>
          <td>Obesidade Grau II</td>
          <td>Entre 35 e 39,9</td>
        </tr>
        <tr>
          <td>Obesidade Grau III ou mórbida</td>
          <td>Maior que 40</td>
        </tr>
      </tbody>
    </table>
  )
}
```

```
const fpeso = (p, sp) => {
  return (
    <div>
      <label>
        Peso
        <input type="text" value={p} onChange={(e) => { sp(e.target.value) }} />
      </label>
    </div>
  )
}
```

```

        </label>
      </div>
    )
  }

const faltura = (a, sa) => {
  return (
    <div>
      <label>
        Altura
        <input type="text" value={a} onChange={(e) => { sa(e.target.value) }} />
      </label>
    </div>
  )
}

const fcalcular = (p, a, sr) => {
  const calc = () => {
    sr(p/(a*a))
  }

  return (
    <div>
      <button onClick={calc}>Calcular</button>
    </div>
  )
}

const fresultado = (r) => {
  return(
    <div>
      <p>Resultado: {r.toFixed(2)}</p>
    </div>
  )
}

export default function App() {

  const [peso, setPeso] = useState(0)
  const [altura, setAltura] = useState(0)
  const [resultado, setResultado] = useState(0)

  return (
    <>
      {fpeso(peso, setPeso)}
      {faltura(altura, setAltura)}
      {fcalcular(peso, altura, setResultado)}
      {fresultado(resultado)}
      {tabelaIMC()}
    </>
  );
}

```

Peso

Altura

Calcular

Resultado: 27.55

Classificação	IMC
Abaixo do peso	Abaixo de 18,5
Peso normal	Entre 18,5 e 24,9
Sobrepeso	Entre 25 e 29,9
Obesidade Grau I	Entre 30 e 34,9
Obesidade Grau II	Entre 35 e 39,9
Obesidade Grau III ou mórbida	Maior que 40

## Aula 31 - Solução do desafio da aula 30

app01\src\componentes\TabelaIMC.js

```
import React from 'react';

export default function tabelaIMC() {
  return (
    <table border='1' style={{ borderCollapse: 'collapse' }}>
      <thead>
        <tr>
          <th>Classificação</th>
          <th>IMC</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>Abaixo do peso</td>
          <td>Abaixo de 18,5</td>
        </tr>
        <tr>
          <td>Peso normal</td>
          <td>Entre 18,5 e 24,9</td>
        </tr>
        <tr>
          <td>Sobrepeso</td>
          <td>Entre 25 e 29,9</td>
        </tr>
        <tr>
          <td>Obesidade Grau I</td>
          <td>Entre 30 e 34,9</td>
        </tr>
        <tr>
          <td>Obesidade Grau II</td>
          <td>Entre 35 e 39,9</td>
        </tr>
        <tr>
          <td>Obesidade Grau III ou mórbida</td>
          <td>Maior que 40</td>
        </tr>
      </tbody>
    </table>
  )
}
```

### app01\src\componentes\Peso.js

```
import React from 'react';

export default function Peso (props) {
  return (
    <div>
      <label>
        Peso
        <input type="text" value={props.p} onChange={(e) => { props.sp(e.target.value) }} />
      </label>
    </div>
  )
}
```

### app01\src\componentes\Altura.js

```
import React from 'react';

export default function Altura(props) {
  return (
    <div>
      <label>
        Altura
        <input type="text" value={props.a} onChange={(e) => { props.sa(e.target.value) }} />
      </label>
    </div>
  )
}
```

### app01\src\componentes\CalcularIMC.js

```
import React from 'react';

export default function CalcularIMC(props) {
  const calc = () => {
    props.sr(props.p / (props.a * props.a))
  }

  return (
    <div>
      <button onClick={calc}>Calcular</button>
    </div>
  )
}
```

### app01\src\componentes\ResultadoIMC.js

```
import React from 'react';

export default function ResultadoIMC (props) {
  return (
    <div>
      <p>Resultado: {props.r.toFixed(2)}</p>
    </div>
  )
}
```

### app01\src\App.js

```
import React, { useState } from 'react'
import './App.css'
import Altura from './componentes/Altura'
import CalcularIMC from './componentes/CalcularIMC'
import Peso from './componentes/Peso'
import ResultadoIMC from './componentes/ResultadoIMC'
import TabelaIMC from './componentes/TabelaIMC'

export default function App() {

  const [peso, setPeso] = useState(0)
  const [altura, setAltura] = useState(0)
  const [resultado, setResultado] = useState(0)

  return (
    <>
      <Peso p={peso} sp={setPeso} />
      <Altura a={altura} sa={setAltura} />
      <CalcularIMC p={peso} a={altura} sr={setResultado} />
      <ResultadoIMC r={resultado} />
      <TabelaIMC />
    </>
  );
}
```

## Aula 32 - Componentes funcionais para componentes de classe [Desafio de React]

app01\src\componentes\TabelaIMC.js

```
import React from 'react';
```

```
export default class TabelaIMC extends React.Component {  
  render(){  
    return (  
      <table border='1' style={{ borderCollapse: 'collapse' }}>  
        <thead>  
          <tr>  
            <th>Classificação</th>  
            <th>IMC</th>  
          </tr>  
        </thead>  
        <tbody>  
          <tr>  
            <td>Abaixo do peso</td>  
            <td>Abaixo de 18,5</td>  
          </tr>  
          <tr>  
            <td>Peso normal</td>  
            <td>Entre 18,5 e 24,9</td>  
          </tr>  
          <tr>  
            <td>Sobrepeso</td>  
            <td>Entre 25 e 29,9</td>  
          </tr>  
          <tr>  
            <td>Obesidade Grau I</td>  
            <td>Entre 30 e 34,9</td>  
          </tr>  
          <tr>  
            <td>Obesidade Grau II</td>  
            <td>Entre 35 e 39,9</td>  
          </tr>  
          <tr>  
            <td>Obesidade Grau III ou mórbida</td>  
            <td>Maior que 40</td>  
          </tr>  
        </tbody>  
      </table>  
    )  
  }  
}
```



### app01\src\componentes\Peso.js

```
import React from 'react';

export default class Peso extends React.Component {
  constructor(){
    super()
  }
  render(){
    return (
      <div>
        <label>
          Peso
          <input type="text" value={this.props.p} onChange={(e) => {this.props.sp(e.target.value) }} />
        </label>
      </div>
    )
  }
}
```

### app01\src\componentes\Altura.js

```
import React from 'react';

export default class Altura extends React.Component {
  constructor() {
    super()
  }
  render() {
    return (
      <div>
        <label>
          Altura
          <input type="text" value={this.props.a} onChange={(e) => { this.props.sa(e.target.value) }} />
        </label>
      </div>
    )
  }
}
```

### app01\src\componentes\CalcularIMC.js

```
import React from 'react';

export default class CalcularIMC extends React.Component {
  constructor(){
    super()
    this.calc=this.fcalc.bind(this)
  }
  fcalc = () => {
    const r = this.props.p/(this.props.a * this.props.a)
    this.props.sr(r)
    // console.log(r)
  }
  render(){
    return (
      <div>
        <button onClick={this.calc}>Calcular</button>
      </div>
    )
  }
}
```

### app01\src\componentes\ResultadoIMC.js

```
import React from 'react';

export default class ResultadoIMC extends React.Component {
  constructor(){
    super()
  }
  render(){
    return (
      <div>
        <p>Resultado: {this.props.r.toFixed(2)}</p>
      </div>
    )
  }
}
```

## app01\src\App.js

```
import React, { useState } from 'react'
import './App.css'
import Altura from './componentes/Altura'
import CalcularIMC from './componentes/CalcularIMC'
import Peso from './componentes/Peso'
import ResultadoIMC from './componentes/ResultadoIMC'
import TabelaIMC from './componentes/TabelaIMC'

export default function App() {

  const [peso, setPeso] = useState(0)
  const [altura, setAltura] = useState(0)
  const [resultado, setResultado] = useState(0)

  return (
    <>
      <Peso p={peso} sp={setPeso} />
      <Altura a={altura} sa={setAltura} />
      <CalcularIMC p={peso} a={altura} sr={setResultado} />
      <ResultadoIMC r={resultado} />
      <TabelaIMC />
    </>
  );
}
```

← → ↻ 📄 localhost:3000

Peso

Altura

Calcular

Resultado: 27.55

Classificação	IMC
Abaixo do peso	Abaixo de 18,5
Peso normal	Entre 18,5 e 24,9
Sobrepeso	Entre 25 e 29,9
Obesidade Grau I	Entre 30 e 34,9
Obesidade Grau II	Entre 35 e 39,9
Obesidade Grau III ou mórbida	Maior que 40

## Aula 33 - Calculadora matemática simples em React - Parte 1

app01\src\App.js

```
import React, { useState } from 'react'

export default function App() {

  const [valorTela, setValorTela] = useState('')
  const [resultado, setResultado] = useState(0)
  const [acumulador, setAcumulador] = useState(0)
  const [operado, setOperado] = useState(false)

  // COMPONENTES

  const Tela = (valor, res) => {
    return(
      <div style={cssTela}>
        <span style={cssTelaOper }>{valor}</span>
        <span style={cssTelaRes }>{res}</span>
      </div>
    )
  }

  const Btn = (label, onClick) => {
    return(
      <button style={cssBtn} onClick={onClick}>{label}</button>
    )
  }

  // FUNÇÕES

  const addDigitoTela = (d) => {
    if ((d == '+' || d == '-' || d == '*' || d == '/') && operado) {
      console.log('+-*/')
      setOperado(false)
      setValorTela(resultado + d)
      return
    }
    if (operado) {
      setValorTela(d)
      setOperado(false)
      return
    }
    const valorDigitadoTela = valorTela + d
    setValorTela(valorDigitadoTela)
  }

  const limparMemoria = () => {
    setOperado(false)
    setValorTela('')
    setResultado(0)
  }
}
```

```

    setAcumulador(0)
    return
  }

  // ESTILOS

  const cssTela = {
    display: 'flex',
    paddingLeft: 20,
    paddingRight: 20,
    justifyContent: 'center',
    alignItems: 'flex-start',
    flexDirection: 'column',
    backgroundColor: '#444',
    width: 260
  }

  const cssTelaOper = {
    fontSize: 25,
    color: 'fff',
    height: 20
  }

  const cssTelaRes = {
    fontSize: 50,
    color: 'fff',
    height: 65
  }

  const cssBtn = {
    fontSize: 30,
    height: 75,
    width: 75,
    padding: 20,
    backgroundColor: '#000',
    color: 'fff',
    borderColor: '#000',
    textAlign: 'center',
    outline: 'none'
  }

  return (
    <>

    </>
  );
}

```

## Aula 34 - Calculadora matemática simples em React - Parte 2

app01\src\App.js

```
import React, { useState } from 'react'
// import './App.css'

export default function App() {

  const [valorTela, setValorTela] = useState("")
  const [resultado, setResultado] = useState(0)
  const [acumulador, setAcumulador] = useState(0)
  const [operado, setOperado] = useState(false)

  // COMPONENTES

  const Tela = (valor, res) => {
    return(
      <div style={cssTela}>
        <span style={cssTelaOper}>{valor}</span>
        <span style={cssTelaRes}>{res}</span>
      </div>
    )
  }

  const Btn = (label, onClick) => {
    return(
      <button style={cssBtn} onClick={onClick}>{label}</button>
    )
  }

  // FUNÇÕES

  const addDigitoTela = (d) => {
    if ((d == '+' || d == '-' || d == '*' || d == '/') && operado) {
      console.log('+-* /')
      setOperado(false)
      setValorTela(resultado + d)
      return
    }
    if (operado) {
      setValorTela(d)
      setOperado(false)
      return
    }
    const valorDigitadoTela = valorTela + d
    setValorTela(valorDigitadoTela)
  }

  const limparMemoria = () => {
    setOperado(false)
    setValorTela("")
    setResultado(0)
    setAcumulador(0)
    return
  }
```

```

}

const Operacao = (oper) => {
  if (oper === 'bs') {
    let vtela = valorTela
    vtela = vtela.substring(0, (vtela.length - 1))
    setValorTela(vtela)
    setOperado(false)
    return
  }

  try {
    const r = eval(valorTela) // calculo
    setAcumulador(r)
    setResultado(r)
    setOperado(true)
  } catch {
    setResultado('ERRO')
  }
}

// ESTILOS

const cssContainer = {
  display: 'flex',
  justifyContent: 'flex-start',
  alignItems: 'center',
  flexDirection: 'column',
  width: 300,
  margin: '0 auto',
  marginTop: 30,
  padding: 10,
  paddingTop: 0,
  border: '1px solid #000'
}

const cssBotoes = {
  flexDirection: 'row',
  flexWrap: 'wrap'
}

const cssTela = {
  display: 'flex',
  paddingLeft: 20,
  paddingRight: 20,
  justifyContent: 'center',
  alignItems: 'flex-start',
  flexDirection: 'column',
  backgroundColor: '#444',
  width: 260
}

const cssTelaOper = {
  fontSize: 25,
  color: '#fff',

```

```

    height: 20
  }

  const cssTelaRes = {
    fontSize: 50,
    color: 'fff',
    height: 65
  }

  const cssBtn = {
    fontSize: 30,
    height: 75,
    width: 75,
    padding: 20,
    backgroundColor: '#000',
    color: 'fff',
    borderColor: '#000',
    textAlign: 'center',
    outline: 'none'
  }

  return (
    <>
    <div style={cssContainer}>
      <h3>Calculadora Matemática Simples</h3>
      {Tela(valorTela,resultado)}

      <div style={cssBotoes}>
        {Btn('AC',limparMemoria)}
        {Btn('(', () => addDigitoTela('('))}
        {Btn(')', () => addDigitoTela(')'))}
        {Btn('/', () => addDigitoTela('/')')}
        {Btn('7', () => addDigitoTela('7'))}
        {Btn('8', () => addDigitoTela('8'))}
        {Btn('9', () => addDigitoTela('9'))}
        {Btn('*', () => addDigitoTela('*'))}
        {Btn('4', () => addDigitoTela('4'))}
        {Btn('5', () => addDigitoTela('5'))}
        {Btn('6', () => addDigitoTela('6'))}
        {Btn('-', () => addDigitoTela('-'))}
        {Btn('1', () => addDigitoTela('1'))}
        {Btn('2', () => addDigitoTela('2'))}
        {Btn('3', () => addDigitoTela('3'))}
        {Btn('+', () => addDigitoTela('+'))}
        {Btn('0', () => addDigitoTela('0'))}
        {Btn('.', () => addDigitoTela('.')')}
        {Btn('<', () => Operacao('bs'))}
        {Btn('=', () => Operacao('='))}
      </div>

    </div>
    </>
  );
}

```



### Calculadora Matemática Simples

0

AC ( ) /

7 8 9 \*

4 5 6 -

1 2 3 +

0 . ← =

### Calculadora Matemática Simples

(6\*5)-7

23

AC ( ) /

7 8 9 \*

4 5 6 -

1 2 3 +

0 . ← =

## Aula 35 - Criando um jogo da velha em React JS - Parte 1 [Jogo em React]

app01\src\App.js

```
import React, { useState } from 'react';

export default function App() {

  // ESTILOS

  const tabu = {
    display: 'flex',
    flexDirection: 'column'
  }
  const tabuLinha = {
    display: 'flex',
    flexDirection: 'row'
  }
  const casa = {
    width: 100,
    height: 100,
    display: 'flex',
    justifyContent: 'center',
    alignItems: 'center',
    flexDirection: 'row',
    cursor: 'pointer',
    fontSize: 60,
    border: '1px solid #000'
  }

  const jogolnicial = [['', '', ''], ['', '', ''], ['', '', '']]
  const [jogo, setJogo] = useState(['', '', ''], ['', '', ''], ['', '', ''])
  const [simboloAtual, setSimboloAtual] = useState('x')
  const [jogando, setJogando] = useState(true)

  const tabuleiro = (j) => {
    return(
      <div style={tabu}>
        <div style={tabuLinha}>
          <div style={casa} data-pos='00' onClick="">{j[0][0]}</div>
          <div style={casa} data-pos='01' onClick="">{j[0][1]}</div>
          <div style={casa} data-pos='02' onClick="">{j[0][2]}</div>
        </div>
        <div style={tabuLinha}>
          <div style={casa} data-pos='10' onClick="">{j[1][0]}</div>
          <div style={casa} data-pos='11' onClick="">{j[1][1]}</div>
          <div style={casa} data-pos='12' onClick="">{j[1][2]}</div>
        </div>
        <div style={tabuLinha}>
          <div style={casa} data-pos='20' onClick="">{j[2][0]}</div>
          <div style={casa} data-pos='21' onClick="">{j[2][1]}</div>
          <div style={casa} data-pos='22' onClick="">{j[2][2]}</div>
        </div>
      </div>
    )
  }
}
```

```
return (  
    <>  
  
    </>  
);  
}
```

## Aula 36 - Criando um jogo da velha em React JS - Parte 2 [Jogo em React]

app01\src\App.js

```
import React, { useState } from 'react';

export default function App() {

  // ESTILOS

  const tabu = {
    display: 'flex',
    flexDirection: 'column'
  }
  const tabuLinha = {
    display: 'flex',
    flexDirection: 'row'
  }
  const casa = {
    width: 100,
    height: 100,
    display: 'flex',
    justifyContent: 'center',
    alignItems: 'center',
    flexDirection: 'row',
    cursor: 'pointer',
    fontSize: 60,
    border: '1px solid #000'
  }

  const jogolInicial = [['', '', ''], ['', '', ''], ['', '', '']]

  const [jogo, setJogo] = useState(['', '', ''], ['', '', ''], ['', '', '']))
  const [simboloAtual, setSimboloAtual] = useState('X')
  const [jogando, setJogando] = useState(true)

  const tabuleiro = (j) => {
    return(
      <div style={tabu}>
        <div style={tabuLinha}>
          <div style={casa} data-pos='00' onClick="">{j[0][0]}</div>
          <div style={casa} data-pos='01' onClick="">{j[0][1]}</div>
          <div style={casa} data-pos='02' onClick="">{j[0][2]}</div>
        </div>
        <div style={tabuLinha}>
          <div style={casa} data-pos='10' onClick="">{j[1][0]}</div>
          <div style={casa} data-pos='11' onClick="">{j[1][1]}</div>
          <div style={casa} data-pos='12' onClick="">{j[1][2]}</div>
        </div>
        <div style={tabuLinha}>
          <div style={casa} data-pos='20' onClick="">{j[2][0]}</div>
          <div style={casa} data-pos='21' onClick="">{j[2][1]}</div>
          <div style={casa} data-pos='22' onClick="">{j[2][2]}</div>
        </div>
      </div>
    )
  }
}
```

```

const verificaVitoria = () => {
  // linhas
  let pontos = 0
  let vitoria = false
  for(let l=0; l<3; l++){
    pontos = 0;
    for (let c=0; c<3; c++) {
      if(jogo[l][c] == simboloAtual){
        pontos++
      }
    }
    if(pontos >= 3){
      vitoria = true
      break
    }
  }

  // colunas
  for (let c=0; c<3; c++) {
    pontos = 0
    for (let l=0; l<3; l++) {
      if (jogo[l][c] == simboloAtual) {
        pontos++
      }
    }
    if (pontos >= 3) {
      vitoria = true
      break
    }
  }

  // diagonais
  pontos = 0
  for (let d=0; d<3; d++) {
    if (jogo[d][d]==simboloAtual) {
      pontos++
    }
  }
  if (pontos >= 3) {
    vitoria = true
  }

  pontos=0
  let l=0
  for (let c=2; c>=0; c--) {
    if (jogo[l][c] == simboloAtual) {
      pontos++
    }
    l++
  }
  if (pontos >= 3) {
    vitoria = true
  }
  return vitoria
}

const trocaJogador = () => {
  simboloAtual == 'X' ? setSimboloAtual('O') : setSimboloAtual('X')
}

```

```
}

const retPos = (e) => {
  const p = e.target.getAttribute('data-pos')
  const pos = [parseInt(p.substring(0, 1)), parseInt(p.substring(1, 2))]
  return pos
}

const verificaEspacoVazio = (e) => {
  if (jogo[retPos(e)[0]][retPos(e)[1]] == ""){
    return true
  } else {
    return false
  }
}

return (
  <>

  </>
);
}
```

## Aula 37 - Criando um jogo da velha em React JS - Parte 3 [Jogo em React]

app01\src\App.js

```
import React, { useState } from 'react';

export default function App() {

  // ESTILOS

  const tabu = {
    display: 'flex',
    flexDirection: 'column'
  }
  const tabuLinha = {
    display: 'flex',
    flexDirection: 'row'
  }
  const casa = {
    width: 100,
    height: 100,
    display: 'flex',
    justifyContent: 'center',
    alignItems: 'center',
    flexDirection: 'row',
    cursor: 'pointer',
    fontSize: 60,
    border: '1px solid #000'
  }
  const btjogarnovamente = {
    marginTop: 30,
    padding: 20,
    color: '#fff',
    backgroundColor: '#000',
    fontSize: 18,
    cursor: 'pointer'
  }

  const jogolInicial = [['', '', ''], ['', '', ''], ['', '', '']]

  const [jogo, setJogo] = useState(['', '', ''], ['', '', ''], ['', '', '']))
  const [simboloAtual, setSimboloAtual] = useState('X')
  const [jogando, setJogando] = useState(true)

  const tabuleiro = (j) => {
    return(
      <div style={tabu}>
        <div style={tabuLinha}>
          <div style={casa} data-pos='00' onClick={(e) => jogar(e)}>{j[0][0]}</div>
          <div style={casa} data-pos='01' onClick={(e) => jogar(e)}>{j[0][1]}</div>
          <div style={casa} data-pos='02' onClick={(e) => jogar(e)}>{j[0][2]}</div>
        </div>
        <div style={tabuLinha}>
          <div style={casa} data-pos='10' onClick={(e) => jogar(e)}>{j[1][0]}</div>
          <div style={casa} data-pos='11' onClick={(e) => jogar(e)}>{j[1][1]}</div>
          <div style={casa} data-pos='12' onClick={(e) => jogar(e)}>{j[1][2]}</div>
        </div>
      </div>
    )
  }
```

```

<div style={tabuLinha}>
  <div style={casa} data-pos='20' onClick={{e} => jogar(e)}>{j[2][0]}</div>
  <div style={casa} data-pos='21' onClick={{e} => jogar(e)}>{j[2][1]}</div>
  <div style={casa} data-pos='22' onClick={{e} => jogar(e)}>{j[2][2]}</div>
</div>
</div>
)
}

const BtnJogarNovamente = () =>{
  if(!jogando){
    return <button style={btjogarnovamente} onClick={()=>reiniciar()}>Jogar novamente</button>
  }
}

const verificaVitoria = () => {
  // linhas
  let pontos = 0
  let vitoria = false
  for(let l=0; l<3; l++){
    pontos = 0;
    for (let c=0; c<3; c++) {
      if(jogo[l][c] == simboloAtual){
        pontos++
      }
    }
    if(pontos >= 3){
      vitoria = true
      break
    }
  }

  // colunas
  for (let c=0; c<3; c++) {
    pontos = 0
    for (let l=0; l<3; l++) {
      if (jogo[l][c] == simboloAtual) {
        pontos++
      }
    }
    if (pontos >= 3) {
      vitoria = true
      break
    }
  }

  // diagonais
  pontos = 0
  for (let d=0; d<3; d++) {
    if (jogo[d][d]==simboloAtual) {
      pontos++
    }
  }
  if (pontos >= 3) {
    vitoria = true
  }

  pontos=0
  let l=0

```



```

for (let c=2; c>=0; c--) {
  if (jogo[l][c] == simboloAtual) {
    pontos++
  }
  l++
}
if (pontos >= 3) {
  vitoria = true
}
return vitoria
}

```

```

const trocaJogador = () => {
  simboloAtual == 'X' ? setSimboloAtual('O') : setSimboloAtual('X')
}

```

```

const retPos = (e) => {
  const p = e.target.getAttribute('data-pos')
  const pos = [parseInt(p.substring(0, 1)), parseInt(p.substring(1, 2))]
  return pos
}

```

```

const verificaEspacoVazio = (e) => {
  if (jogo[retPos(e)[0]][retPos(e)[1]] == ""){
    return true
  } else {
    return false
  }
}

```

```

const jogar = (e) => {
  if(jogando){
    if (verificaEspacoVazio(e)){
      jogo[retPos(e)[0]][retPos(e)[1]] = simboloAtual
      trocaJogador()
    }
    if (verificaVitoria()){
      trocaJogador()
      alert('Jogador ' + simboloAtual + ' venceu!')
      setJogando(false)
    }
  } else {
    alert('Este espaço não está disponível! Escolha outro.')
  }
}

```

```

const reiniciar = () => {
  setJogando(true)
  setJogo(jogoInicial)
  setSimboloAtual('X')
}

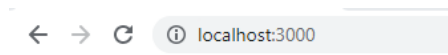
```

```

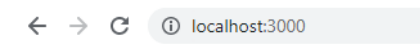
return (
  <>
  <div>
    <p>Quem joga: {simboloAtual}</p>
  </div>
  <div>
    {tabuleiro(jogo)}
  </div>
)

```

```
        </div>
        <div>
            {BtnJogarNovamente()}
        </div>
    </>
);
}
```



Quem joga: X

Quem joga: O

X		

← → ↻

localhost:3000

Quem joga: X

X	X	O
	X	
O	O	

localhost:3000 diz

Jogador X venceu!

OK

← → ↻

localhost:3000

Quem joga: O

X	X	O
	X	
O	O	X

Jogar novamente

## Aula 38 - API simples em Nodejs [API NODE]

### Criando a API

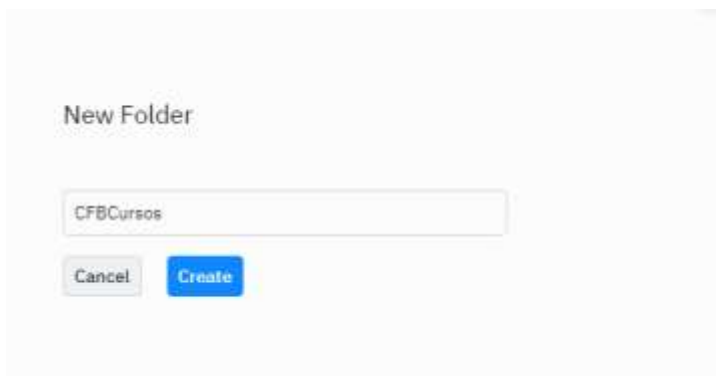
- Acesse sua conta no Replit:

<https://replit.com/>

- Clique em "My Repls"



- Clique no botão "New folder"



- Digite "CFBCursos"

- Clique no botão "Create"


- Acesse a pasta criada e clique no botão "New repl"

- Em **Language**, selecione "Node JS" e no campo **Title** digite "cfbcursos\_api\_react\_exemplo1"

# Create a repl

Template

Node.js

 Languages

**Node.js** ✓


Runtime for writing back-end JavaScript.


Title


cfbcursos\_api\_react\_exemplo1

Privacy

☐ Repl is public

 Anyone can view and fork this repl

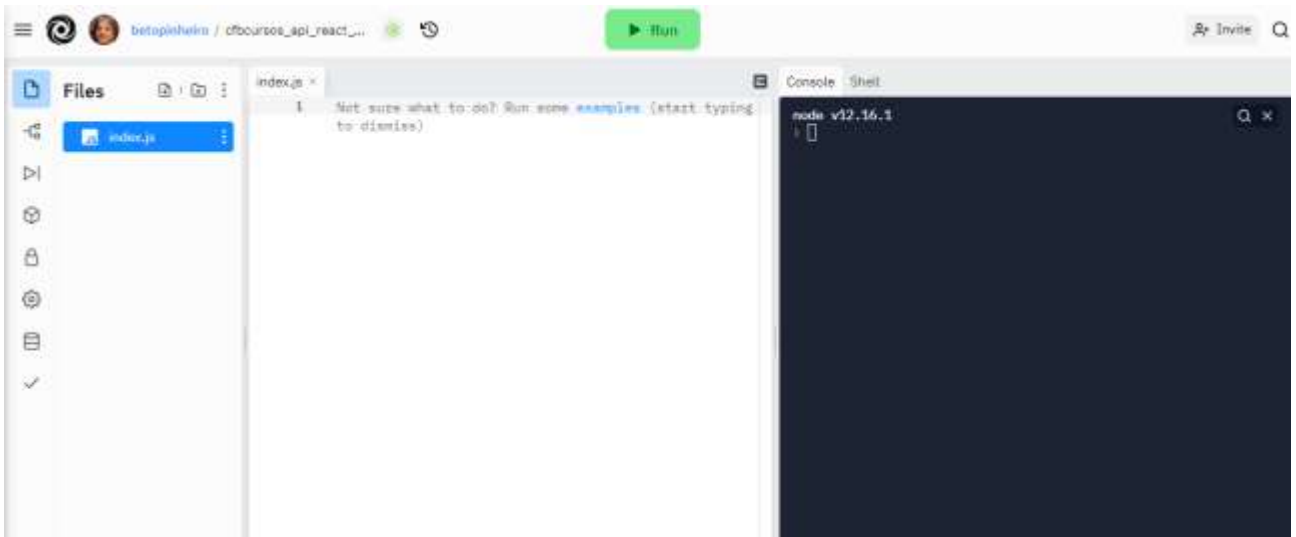
 Upgrade to make private

 replit

11 + 599.6K

+ Create Repl

- Clique no botão "Create Repl"



## index.js

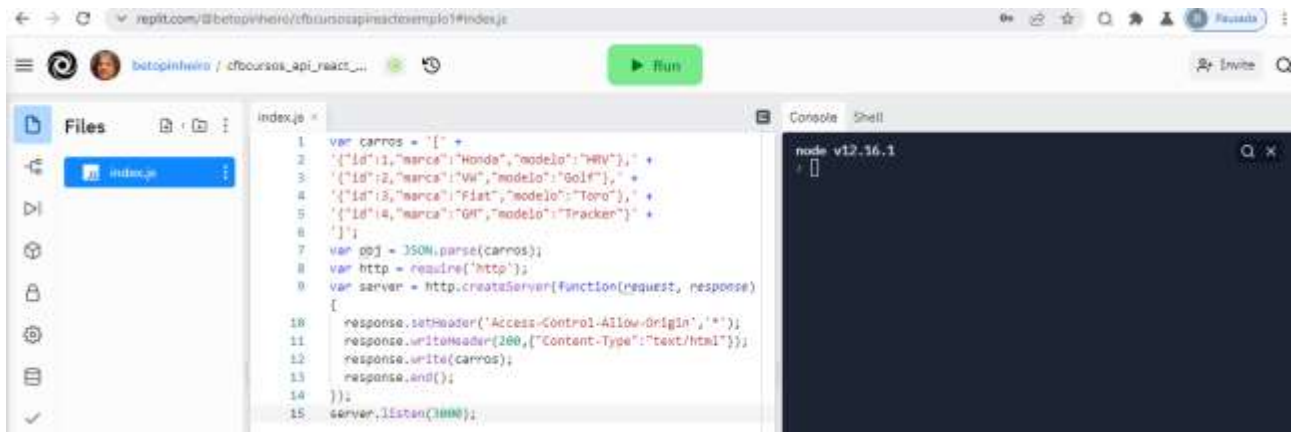
```
var carros = '[' +
'{"id":1,"marca":"Honda","modelo":"HRV"},' +
'{"id":2,"marca":"VW","modelo":"Golf"},' +
'{"id":3,"marca":"Fiat","modelo":"Toro"},' +
'{"id":4,"marca":"GM","modelo":"Tracker"}' +
'];';

var obj = JSON.parse(carros);

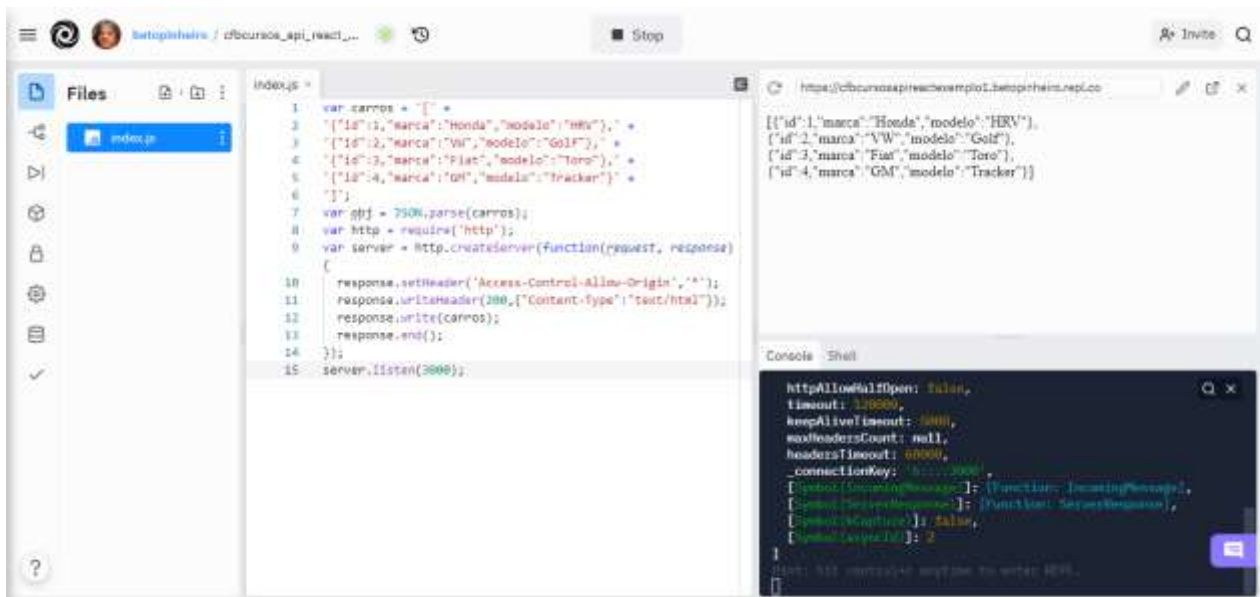
var http = require('http');

var server = http.createServer(function(request, response){
    response.setHeader('Access-Control-Allow-Origin','*');
    response.writeHead(200,{"Content-Type":"text/html"});
    response.write(carros);
    response.end();
});

server.listen(3000);
```



- Clique no botão "Run"



- Para que se possa acessar essa API, o Repl deve estar aberto e rodando.

- O endpoint ou endereço para chamar essa API no React é:

<https://cfbcursosapireactexemplo1.betopinheiro.repl.co>



- Usando a extensão **JSON Viewer** no Google Chrome:



The screenshot shows the JSON Viewer extension interface in Google Chrome. The address bar displays the URL `cfbcursosapireactexemplo1.betopinhoiro.repl.co`. The JSON data is displayed in a syntax-highlighted format with line numbers on the left. The data is an array of four car objects, each with `id`, `marca`, and `modelo` properties. A semi-transparent white box is overlaid on the bottom right of the JSON data.

```
1 // 20220108174655
2 // https://cfbcursosapireactexemplo1.betopinhoiro.repl.co/
3
4 [
5   {
6     "id": 1,
7     "marca": "Honda",
8     "modelo": "HRV"
9   },
10  {
11    "id": 2,
12    "marca": "VW",
13    "modelo": "Golf"
14  },
15  {
16    "id": 3,
17    "marca": "Fiat",
18    "modelo": "Toro"
19  },
20  {
21    "id": 4,
22    "marca": "GM",
23    "modelo": "Tracker"
24  }
25 ]
```

- Agora basta ir no aplicativo React e programar essa chamada.

## Aula 39 - Como consumir dados de uma API Nodejs [API NODE]

### Instalando o AXIOS

<https://www.npmjs.com/package/axios>

- No terminal, na pasta do aplicativo, entre com o comando:

**npm install axios**

```
C:\xampp\htdocs\App01\app01>npm install axios
npm WARN @babel/plugin-bugfix-v8-spread-parameters-in-optional-chaining@7.16.0 requires a peer of @babel/core@^7.13.0 but none is installed. You must install peer dependencies yourself.
npm WARN tsutils@3.21.0 requires a peer of typescript@>=2.8.0 || >= 3.2.0-dev || >= 3.3.0-dev || >= 3.4.0-dev || >= 3.5.0-dev || >= 3.6.0-dev || >= 3.6.0-beta || >= 3.7.0-dev || >= 3.7.0-beta but none is installed. You must install peer dependencies yourself.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.3.2 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\webpack-chokidar2\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\webpack-dev-server\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

+ axios@0.24.0
added 1 package from 1 contributor and audited 1885 packages in 71.929s

154 packages are looking for funding
  run `npm fund` for details

found 21 vulnerabilities (2 low, 14 moderate, 4 high, 1 critical)
  run `npm audit fix` to fix them, or `npm audit` for details
```

**npm install react-axios**

```
C:\xampp\htdocs\App01\app01>npm install react-axios
npm WARN @babel/plugin-bugfix-v8-spread-parameters-in-optional-chaining@7.16.0 requires a peer of @babel/core@^7.13.0 but none is installed. You must install peer dependencies yourself.
npm WARN tsutils@3.21.0 requires a peer of typescript@>=2.8.0 || >= 3.2.0-dev || >= 3.3.0-dev || >= 3.4.0-dev || >= 3.5.0-dev || >= 3.6.0-dev || >= 3.6.0-beta || >= 3.7.0-dev || >= 3.7.0-beta but none is installed. You must install peer dependencies yourself.
npm WARN react-axios@2.0.5 requires a peer of axios@^0.21.1 but none is installed. You must install peer dependencies yourself.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.3.2 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\webpack-chokidar2\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\webpack-dev-server\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

+ react-axios@2.0.5
added 1 package from 1 contributor and audited 1886 packages in 84.221s

154 packages are looking for funding
  run `npm fund` for details

found 21 vulnerabilities (2 low, 14 moderate, 4 high, 1 critical)
  run `npm audit fix` to fix them, or `npm audit` for details
```



### app01\src\componentes\ListaCarros.js

```
import axios from 'axios'
import React from 'react'

export default class ListaCarros extends React.Component{

  state = {
    carros: []
  }

  componentDidMount(){
    axios.get('https://cfbcursosapireactexemplo1.betopinheiro.repl.co')
    .then(res=>{
      const dadosCarros = res.data
      this.setState({carros:dadosCarros})
    })
  }

  render(){
    return(
      <div>
        {this.state.carros.map(
          carro=> <div key={carro.id}> {carro.id} - {carro.marca} - {carro.modelo}</div>
        )}
      </div>
    )
  }
}
```

### app01\src\App.js

```
import React from 'react';
import './App.css';
import ListaCarros from './componentes/ListaCarros';

export default function App() {

  return (
    <>
      <ListaCarros />
    </>
  );
}
```

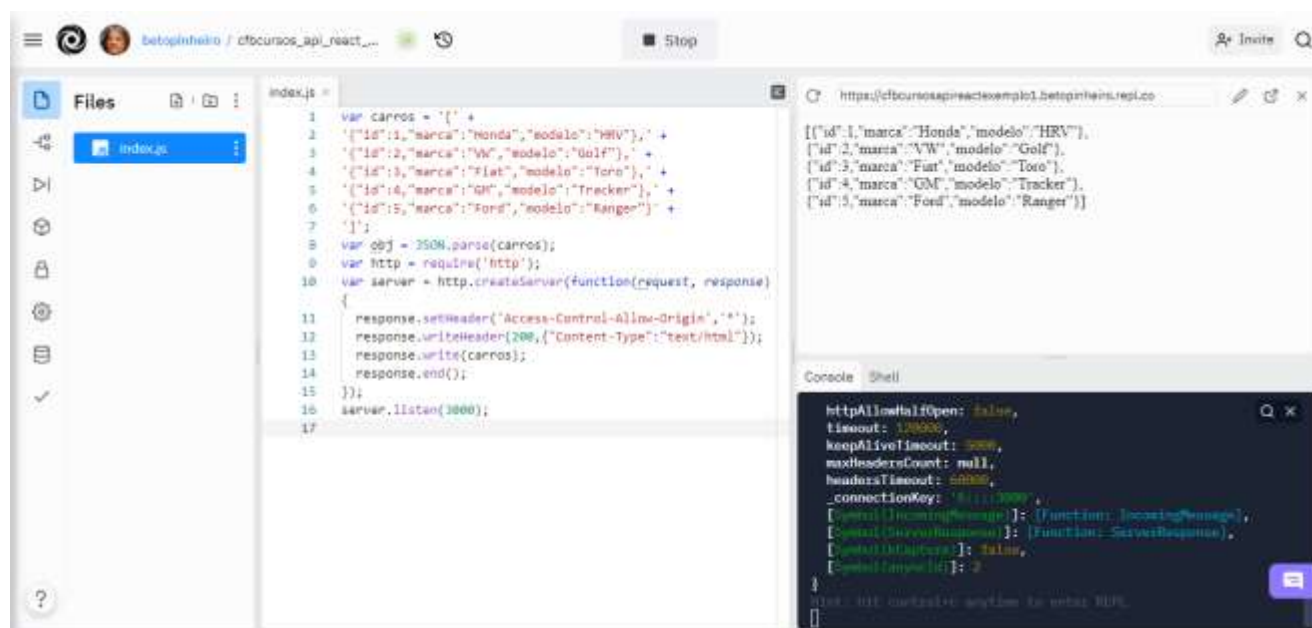
← → ↻ ⓘ localhost:3000

Honda - HRV  
VW - Golf  
Fiat - Toro  
GM - Tracker

- No json (em Replit), insira mais um carro e em seguida clique no botão "Run":

### index.js

```
var carros = '[' +  
'{"id":1,"marca":"Honda","modelo":"HRV"},' +  
'{"id":2,"marca":"VW","modelo":"Golf"},' +  
'{"id":3,"marca":"Fiat","modelo":"Toro"},' +  
'{"id":4,"marca":"GM","modelo":"Tracker"},' +  
'{"id":5,"marca":"Ford","modelo":"Ranger"}' +  
';  
var obj = JSON.parse(carros);  
var http = require('http');  
var server = http.createServer(function(request, response){  
  response.setHeader('Access-Control-Allow-Origin', '*');  
  response.writeHead(200, {"Content-Type": "text/html"});  
  response.write(carros);  
  response.end();  
});  
server.listen(3000);
```



- No browser, atualize a página:



- 1 - Honda - HRV
- 2 - VW - Golf
- 3 - Fiat - Toro
- 4 - GM - Tracker
- 5 - Ford - Ranger

## Aula 40 - Consumindo dados de uma API usando componente funcional [API NODE]

### app01\src\componentes\ListaCarros2.js

```
import axios from 'axios'
import React, { useEffect, useState } from 'react'

export default function ListaCarros2() {

  const [carros, setCarros] = useState([])

  useEffect(() => {
    axios.get('https://cfbcursosapireactexemplo1.betopinheiro.repl.co')
      .then(res => {
        const dadosCarros = res.data
        setCarros(dadosCarros)
      })
  })

  return (
    <div>
      {carros.map(
        carro => <div key={carro.id}>{carro.id} - {carro.marca} - {carro.modelo}</div>
      )}
    </div>
  )
}
```

### app01\src\App.js

```
import React from 'react';
import './App.css';
import ListaCarros2 from './componentes/ListaCarros2';

export default function App() {

  return (
    <>
      <ListaCarros2 />
    </>
  );
}
```

← → ↻ ⓘ localhost:3000

```
1 - Honda - HRV
2 - VW - Golf
3 - Fiat - Toro
4 - GM - Tracker
5 - Ford - Ranger
```

## Aula 41 - Como usar FETCH para consumir API NODE [API NODE]

### app01\src\componentes\ListaCarros3.js

```
import React, { useEffect, useState } from 'react'

export default function ListaCarros2() {

  const [carros, setCarros] = useState([])

  useEffect(() => {
    fetch('https://cfbcursosapireactexemplo1.betopinheiro.repl.co')
      .then(res=>res.json())
      .then(
        (resultado) => {
          setCarros(resultado)
        }
      )
  })

  return (
    <div>
      {carros.map(
        carro => <div key={carro.id}>{carro.id} - {carro.marca} - {carro.modelo}</div>
      )}
    </div>
  )
}
```

### app01\src\App.js

```
import React from 'react';
import './App.css';
import ListaCarros3 from './componentes/ListaCarros3';

export default function App() {

  return (
    <>
      <ListaCarros3 />
    </>
  );
}
```

← → ↻ ⓘ localhost:3000

```
1 - Honda - HRV
2 - VW - Golf
3 - Fiat - Toro
4 - GM - Tracker
5 - Ford - Ranger
```

## Aula 42 - Como criar rotas de navegação no React [React Router]

### Instalando o react-router-dom

<https://www.npmjs.com/package/react-router-dom>

Dentro da pasta da aplicação, execute o comando:

**npm install --save react-router-dom**

```
beto1@DESKTOP-HFTIOCI MINGW64 /c:/xampp/htdocs/App01/app01 (master)
$ npm install --save react-router-dom
npm WARN @babel/plugin-bugfix-v8-spread-parameters-in-optional-chaining@7.16.0 requires a peer of @babel/core@^7.13.0 but none is installed. You must install peer dependencies yourself.
npm WARN react-axios@2.0.5 requires a peer of axios@^0.21.1 but none is installed. You must install peer dependencies yourself.
npm WARN tsutils@3.21.0 requires a peer of typescript@>=2.8.0 || >= 3.2.0-dev || >= 3.3.0-dev || >= 3.4.0-dev || >= 3.5.0-dev || >= 3.6.0-dev || >= 3.6.0-beta || >= 3.7.0-dev || >= 3.7.0-beta but none is installed. You must install peer dependencies yourself.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.3.2 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.3.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\watchpack-chokidar2\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\webpack-dev-server\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

+ react-router-dom@6.2.1
added 3 packages from 1 contributor and audited 1889 packages in 198.012s

154 packages are looking for funding
  run `npm fund` for details

found 21 vulnerabilities (2 low, 14 moderate, 4 high, 1 critical)
  run `npm audit fix` to fix them, or `npm audit` for details
```

### Componentes (Páginas)

**app01\src\componentes\Pagina1.js**

import React from 'react';

export default function Pagina1() {

```
  return (
    <div>
      <h1>Página 1</h1>
      <h3>Curso de React</h3>
    </div>
  );
}
```

### **app01\src\componentes\Pagina2.js**

```
import React from 'react';

export default function Pagina2() {

  return (
    <div>
      <h1>Página 2</h1>
      <h3>CFB Cursos</h3>
    </div>
  );
}
```

### **app01\src\componentes\Pagina3.js**

```
import React from 'react';

export default function Pagina3() {

  return (
    <div>
      <h1>Página 3</h1>
      <h3>React Router</h3>
    </div>
  );
}
```

### **app01\src\componentes\Home.js**

```
import React from 'react';

export default function Home() {

  return (
    <div>
      <h1>Home</h1>
      <h3>Página Principal</h3>
    </div>
  );
}
```

app01\src\App.js

```
import React from 'react';
import { BrowserRouter as Router, Link, Route, Routes } from 'react-router-dom';
import './App.css';
import Home from './componentes/Home';
import Pagina1 from './componentes/Pagina1';
import Pagina2 from './componentes/Pagina2';
import Pagina3 from './componentes/Pagina3';

export default function App() {

  return (
    <>
      <Router>
        <header>
          <Link to='/>Home</Link>
          <Link to='/pag1'>Página 1</Link>
          <Link to='/pag2'>Página 2</Link>
          <Link to='/pag3'>Página 3</Link>
        </header>

        <Routes>
          <Route path='/' element={<Home />} />
          <Route path='/pag1' element={<Pagina1 />} />
          <Route path='/pag2' element={<Pagina2 />} />
          <Route path='/pag3' element={<Pagina3 />} />
        </Routes>
      </Router>
    </>
  );
}
```

← → ↻ ⓘ localhost:3000

[Home](#)[Página 1](#)[Página 2](#)[Página 3](#)

# Home

**Página Principal**

← → ↻ ⓘ localhost:3000/pag1

[Home](#)[Página 1](#)[Página 2](#)[Página 3](#)

# Página 1

**Curso de React**



← → ↻ ⓘ localhost:3000/pag2

[Home](#)[Página 1](#)[Página 2](#)[Página 3](#)

## Página 2

**CFB Cursos**

← → ↻ ⓘ localhost:3000/pag3

[Home](#)[Página 1](#)[Página 2](#)[Página 3](#)

## Página 3

**React Router**

## Aula 43 - Atributo KEY [ReactJS]

Esta propriedade é importante quando estamos trabalhando com qualquer tipo de lista.

app01\src\App.js

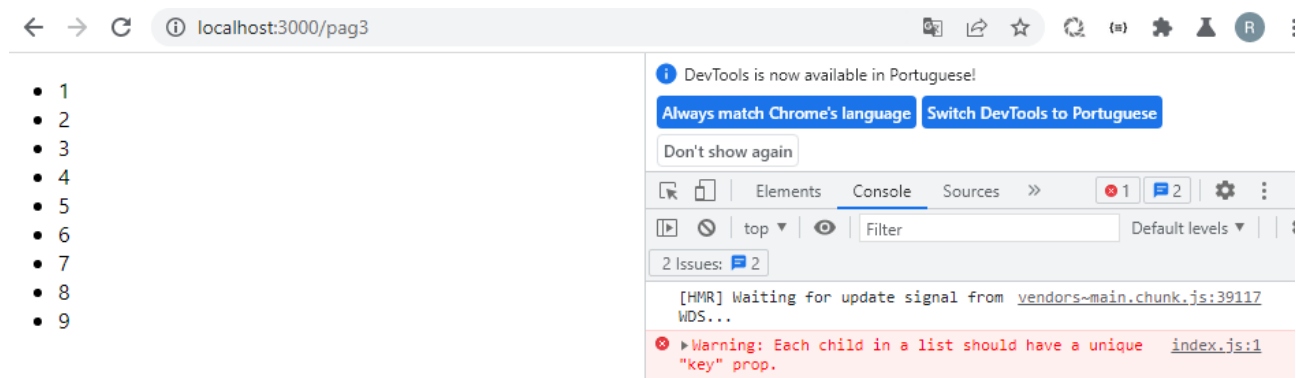
```
import React from 'react';
import './App.css';

function ListaNumeros(props){
  const num = props.numeros
  const lista_numeros = num.map(
    (n) => <li>{n}</li>
  )
  return(<ul>{lista_numeros}</ul>)
}

const array_numeros = [1,2,3,4,5,6,7,8,9]

export default function App() {

  return (
    <>
      <ListaNumeros numeros={array_numeros} />
    </>
  );
}
```



- Perceba a mensagem:

warning: Each child in a list should have a unique "key" prop.

Recomenda-se que essa key seja sempre uma string diferente para cada item da lista.

#### **app01\src\App.js**

```
import React from 'react';
import './App.css';

function ListaNumeros(props){
  const num = props.numeros
  const lista_numeros = num.map(
    (n) => <li key={n.toString()}>{n}</li>
  )
  return(<ul>{lista_numeros}</ul>)
}

const array_numeros = [1,2,3,4,5,6,7,8,9]

export default function App() {

  return (
    <>
      <ListaNumeros numeros={array_numeros} />
    </>
  );
}
```

- Dessa forma, a mensagem de warning não será mais exibida.