



Introdução ao ReactJS

Filipe Gonçalves
Pedro Lopes

Aula 12 - 12 de Dezembro de 2022

Índice

Visão Geral

01



Passos Iniciais e Conceitos

02



Pôr em Prática

03



04

Jogos Olímpicos



05

Informações



Visão Geral

ReactJS



FÁCIL DE APRENDER

Não implica conhecimento profundo de outras linguagens




BASEADO EM COMPONENTES

Componentes gerem o seu próprio estado, escrito em JS



DECLARATIVO

Criação fácil de interfaces interativas



“Learn once, write anywhere”
We don’t make assumptions about the rest of
your technology stack, so you can develop
new features in React without rewriting
existing code.

— **ReactJS**

Instalação

<https://www.knowledgehut.com/blog/web-development/install-react-on-mac>



<https://docs.microsoft.com/en-us/windows/dev-environment/javascript/react-on-windows>



<https://www.tecmint.com/install-reactjs-on-ubuntu/>



Criar uma aplicação



```
npx create-react-app my-app
```

```
cd my-app
```

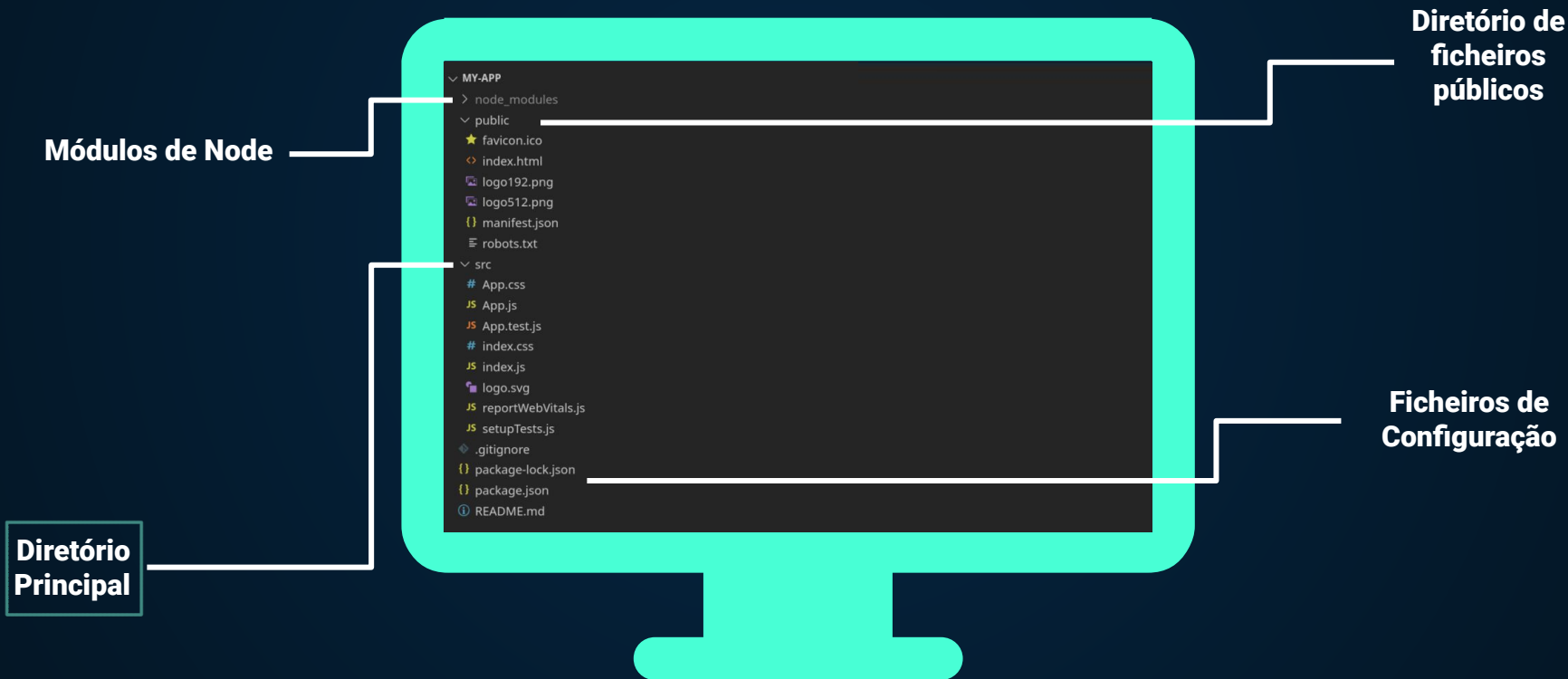
```
npm start
```





Passos Iniciais e Conceitos

Estrutura Inicial



Estrutura Melhorada

Diretório de Componentes

Diretório de Páginas

```
MY-APP
├── node_modules
├── public
│   ├── favicon.ico
│   ├── index.html
│   ├── logo192.png
│   ├── logo512.png
│   ├── manifest.json
│   └── robots.txt
├── src
│   ├── components
│   ├── images
│   ├── pages
│   ├── App.css
│   ├── App.js
│   ├── App.test.js
│   ├── index.css
│   ├── index.js
│   ├── logo.svg
│   ├── reportWebVitals.js
│   └── setupTests.js
├── .gitignore
├── package-lock.json
├── package.json
└── README.md
```

Diretório de Imagens

Instalar Packages



```
npm install react-bootstrap bootstrap  
npm install cdbreact
```

```
import 'react-bootstrap/<component>'  
import <components> from 'cdbreact'
```



App.js

Imports

```
1 import logo from './logo.svg';
2 import './App.css';
3
4 function App() {
5   return (
6     <div className="App">
7       <header className="App-header">
8         <img src={logo} className="App-logo" alt="logo" />
9         <p>
10           Edit <code>src/App.js</code> and save to reload.
11         </p>
12         <a
13           className="App-link"
14           href="https://reactjs.org"
15           target="_blank"
16           rel="noopener noreferrer"
17         >
18           Learn React
19         </a>
20       </header>
21     </div>
22   );
23 }
24
25 export default App;
```

Exportar a
função

Página
HTML



Pôr em Prática

Componentes

```
class HelloMessage extends React.Component {  
  render() {  
    return <div>Hello {this.props.name}</div>;  
  }  
}  
  
root.render(<HelloMessage name="Taylor" />);
```

`render()` recebe dados como **argumento de entrada** e retorna o que deve ser **displayed**

Componentes com React Bootstrap



```
function NavbarITW() {  
  return (  
    <Navbar bg="dark" variant="dark">  
      <Container>  
        <Navbar.Brand href="#home">Olympic Games <i class="fa-solid fa-trophy"></i></Navbar.Brand>  
        <Navbar.Toggle aria-controls="basic-navbar-nav" />  
        <Navbar.Collapse id="basic-navbar-nav">  
          <Nav className="me-auto">  
            <Nav.Link href="#home">Athletes</Nav.Link>  
          </Nav>  
        </Navbar.Collapse>  
      </Container>  
    </Navbar>  
  );  
}  
  
export default NavbarITW;
```

Páginas

```
<div>
```

```
  <NavbarITW />
```

```
</div>
```

Invocação do componente

Componentes com State

```
class Timer extends React.Component {  
  constructor(props) {  
    super(props);  
    this.state = { seconds: 0 };  
  }  
  
  tick() {  
    this.setState(state => ({  
      seconds: state.seconds + 1  
    }));  
  }  
  
  componentDidMount() {  
    this.interval = setInterval(() => this.tick(), 1000);  
  }  
  
  render() {  
    return ( <div> Seconds: {this.state.seconds} </div> );  
  }  
}  
  
root.render(<Timer />);
```

Props são os
argumentos de entrada

Podemos **guardar** o state numa variável e
atualizar o seu valor quando quisermos

componentDidMount() é uma
função que é chamada sempre que
o **componente** é criado



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

function Example() {

  // Declare a new state variable, which we'll call "count"
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}
```

State Hook

- Permite manter o estado de uma variável entre renders
- Retorna o valor atual e uma função de atualização



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

function Example() {
  const [count, setCount] = useState(0);
  useEffect(() => {
    // Update the document title using the browser API
    document.title = `You clicked ${count} times`;
  }); // Always runs again

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}
```

Effect Hook

- Garante a atualização do **componente**



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

function Example() {
  const [count, setCount] = useState(0);
  useEffect(() => {
    // Update the document title using the browser API
    document.title = `You clicked ${count} times`;
  }, [count]); // runs when count changes

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}
```

Effect Hook

- Neste caso o componente só é atualizado caso a variável **count** mude



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

export default function Counter() {
  const [count, setCount] = useState(0);

  useEffect(() => {
    const intervalId = setInterval(() => {
      setCount(c => c + 1); // ✅ Pass a state updater
    }, 1000);
    return () => clearInterval(intervalId);
  }, []); // ✅ Now count is not a dependency
  // runs in periods of 1 second

  return <h1>{count}</h1>;
}
```

Effect Hook

- **setInterval** permite a execução de uma função a cada intervalo de tempo
- Permite a remoção da dependência



Jogos Olímpicos

Routes

```

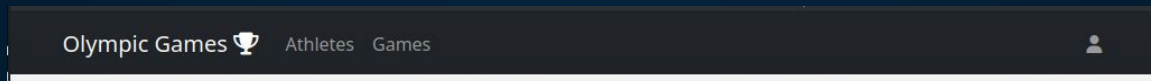
<Router>
  <NavbarITW />
  <br /><br /><br />
  <Routes>
    <Route path="/" exact element={<Home />} />
    <Route path="/404" exact element={<Error />} />
    <Route path="/Home" exact element={<Home />} />
    <Route path="/games" exact element={<Games />} />
    <Route path="/athletes" exact element={<Athletes />} />
  </Routes>
</Router>

```

Referência para o caminho da
página que queremos seguir

Componente para onde queremos ir

Navbar



Alinha os itens à
esquerda

Alinha os itens à
direita

```
function NavbarITW() {
  return (
    <Navbar fixed="top" bg="dark" variant="dark">
      <Container>
        <Navbar.Brand href="/">Olympic Games <i class="fa-solid fa-trophy"></i></Navbar.Brand>
        <Navbar.Toggle aria-controls="basic-navbar-nav" />
        <Navbar.Collapse id="basic-navbar-nav">
          <Nav className="me-auto">
            <Nav.Link href="/athletes">Athletes</Nav.Link>
            <Nav.Link href="/games">Games</Nav.Link>
          </Nav>
          <Nav className="ms-auto">
            <Nav.Link href="" style={{float: "right"}}><i class="fa-solid fa-user"></i></Nav.Link>
          </Nav>
        </Navbar.Collapse>
      </Container>
    </Navbar>
  );
}

export default NavbarITW;
```

Href para os links
predefinidos

Footer

```
<CDBBox>
  <p className="h5 mb-4" style={{ fontWeight: '600' }}>
    Filipe Gonçalves
  </p>
  <CDBBox display="flex">
    <a href="https://www.linkedin.com/in/filipe-gon%C3%A7alves-150789239/" target="_blank" rel="noreferrer">
      <CDBBtn flat color="dark" className="p-2">
        <CDBIcon fab icon="linkedin" />
      </CDBBtn>
    </a>
    <a href="https://github.com/FlipGoncalves" target="_blank" rel="noreferrer">
      <CDBBtn flat color="dark" className="mx-3 p-2">
        <CDBIcon fab icon="github"/>
      </CDBBtn>
    </a>
    <a href="https://www.instagram.com/filipe_asg/" target="_blank" rel="noreferrer">
      <CDBBtn flat color="dark" className="p-2">
        <CDBIcon fab icon="instagram" />
      </CDBBtn>
    </a>
  </CDBBox>
  <p className="h8 mb-4" style={{ marginTop: '15px' }}>
    filipepeg@ua.pt
  </p>
</CDBBox>
```

**Botões
e links**



Filipe Gonçalves



filipepeg@ua.pt

Pedro Lopes



pdfi@ua.pt



Cards

```
const [data, setData] = useState([]);

function getData() {

  let resp = fetch('http://192.168.160.58/0lympics/api/Athletes/'+props["id"], {
    method: 'GET'
  }).then((dados)=>{
    dados.json().then((properties) => {

      console.log(properties);

      var temp = {};

      temp["Name"] = properties["Name"]
      temp["Image"] = properties["Photo"]
      temp["BornPlace"] = properties["BornPlace"]
      temp["BornDate"] = properties["BornDate"]
      temp["Weight"] = properties["Weight"]
      temp["Height"] = properties["Height"]
      temp["Sex"] = properties["Sex"]

      setData(temp);

    })
  }).catch((error) => {
    console.log("error")
  })

  console.log(data);
}

useEffect(() => {
  getData()
  console.log("here" + props["id"])
}, [])
```

O que recebemos da
invocação do componente

```
<Card style={{ width: '18rem' }}>
  <div style={{width: "18rem"}}>
    <Card.Img variant="top" src={data["Image"]} className="card-img"
    style={{height:"20rem", objectFit: "cover"}}/>
  </div>
  <Card.Body>
    <Card.Title align="center">{data["Name"]}</Card.Title>
    <Card.Text>
      <p>Data de Nascimento: {data["BornDate"]}</p>
      <p>Nascido em: {data["BornPlace"]}</p>
      <p>Altura: {data["Height"]}</p>
      <p>Peso: {data["Weight"]}</p>
      <p>Sexo: {data["Sex"]} </p>
    </Card.Text>
  </Card.Body>
</Card>
```

Guardamos a informação recebida da API numa
variável com **setState** e depois **atualizamos** o nosso
Card

Cards



Michael Fred Phelps, II

Data de Nascimento:
1985-06-30T00:00:00

Nascido em: Baltimore, Maryland
(USA)

Altura: 193

Peso: 91

Sexo: M



Paavo Johannes Nurmi

Data de Nascimento:
1897-06-12T00:00:00

Nascido em: Turku (FIN)

Altura: 174

Peso: 65

Sexo: M



Carl Lewis Borack

Data de Nascimento:
1947-07-12T00:00:00

Nascido em: Staten Island, New
York (USA)

Altura: 183

Peso: 76

Sexo: M

Carroussel

```
<Carousel fade variant="dark" md>
```

```
<Carousel.Item>
```

```
<div className="section-2">
```

```
<img
```

```
  style={{objectFit: "fill", width:"100%"}}
```

```
  className="d-block w-100 h-100"
```

```
  src={og1}
```

```
  alt="First slide"
```

```
</div>
```

```
</Carousel.Item>
```

```
<Carousel.Item>
```

```
<div className="section-2">
```

```
<img
```

```
  style={{objectFit: "fill", width:"100%"}}
```

```
  className="d-block w-100 h-100"
```

```
  src={og2}
```

```
  alt="Second slide"
```

```
</div>
```

```
</Carousel.Item>
```

```
</Carousel>
```

Imagem



Tabelas

Modal para abrir as informações adicionais

```
<Container>
  <Table striped bordered hover style={{width: '100%', margin: '5px'}}>
    <thead style={{backgroundColor: 'grey', color: 'white'}}>
      <tr>
        <th>Nome</th>
        <th>Sexo</th>
        <th></th>
      </tr>
    </thead>
    <tbody>
      {data}
    </tbody>
  </Table>
  {modal.show && modal.id && <ModalAthletes closeModal={handleClose} data={modal.id} type={modal.type} />}
</Container>
```

Tipo: “Edit” ou “See”
Id: id do atleta/jogo

Nome	Sexo	
A. Aanantha Sambu Mayavo	M	 
A. Albert	M	 
A. Brun	F	 
A. Charles Six	M	 
A. Christory	M	 
		 

Modal

```
<Modal
  show={true}
  onHide={() => closeModal()}
  centered
  size="lg"
>
  <Modal.Header>
    <Modal.Title id="contained-modal-title-vcenter">
      <h3>{dataFetched[0][ "Name" ]}</h3>
    </Modal.Title>
  </Modal.Header>
  <Modal.Body>
    <div class="d-flex justify-content-between align-items-center mt-4 px-4" align="center">
      <div class="stats">
        <h6 class="mb-0">Nome</h6>
        <span>{dataFetched[0][ "Name" ]}</span>
      </div>
    </div>
  </Modal.Body>
  <Modal.Footer>
    <Button onClick={closeModal}>Close</Button>
  </Modal.Footer>
</Modal>
```

Dados recebidos da API

Aage Carl Christian Lassen

Nome	Sexo	Altura	Peso
Aage Carl Christian Lassen	M	181	62

Close

Forms

```
<Form>
  <Form.Group className="mb-3" controlId="formGames">
    <Form.Label>Nome dos Jogos</Form.Label>
    <Form.Control type="text" placeholder={dataFetched[0][ "Name" ]} />
    <Form.Group className="mb-3">
      <Form.Label as="legend" column sm={6}>
        Estação do Ano
      </Form.Label>
      <Form.Check
        type="radio"
        label="Winter"
        name="formHorizontalRadios"
        id="formHorizontalRadios1"
        checked={dataFetched["Season"] === "Winter"}
      />
    </Form.Group>
    <Form.Label>Ano</Form.Label>
    <Form.Control type="year" placeholder={dataFetched[0][ "Year" ]} />
  </Form.Group>
  <Button variant="primary" type="submit">
    Submit
  </Button>
</Form>
```

Botão de Submit

FormID para depois enviar para a API os dados

Nome dos Jogos

Estação do Ano

☐ Winter

☒ Summer

Ano

Pais

Cidade

Logótipo

Foto



Informações

Links



<https://reactjs.org/docs/getting-started.html>



<https://react-bootstrap.github.io/getting-started/introduction/>



<https://www.devwares.com/docs/contrast/react/index/>



https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch

Filipe Gonçalves

filipeg@ua.pt

Pedro Lopes

pdf1@ua.pt