



# Introdução ao React JS

11 de Janeiro 2022

# NA AULA DE HOJE:

01

Visão Geral

02

Passos  
iniciais e  
conceitos

03

Pôr em  
prática

04

Adorar  
React 



# 01

## Visão Geral

## Declarativo

Criação fácil de interfaces  
interativas

## Baseado em componentes

Escritos em JS

## Fácil de aprender

Não implica o conhecimento  
profundo de outras  
tecnologias



“Learn once, write anywhere”

–React JS

# INSTALAÇÃO

- Windows: <https://docs.microsoft.com/en-us/windows/dev-environment/javascript/react-on-windows>
- MacOS: <https://www.knowledgehut.com/blog/web-development/install-react-on-mac>
- Linux: <https://www.tecmint.com/install-reactjs-on-ubuntu/>

# CRIAR UMA APLICAÇÃO E INICIÁ-LA

- `npx create-react-app my-app` (ou `"npx create-react-app@latest my-app"` em caso de erro com versões)
- `cd my-app`
- `npm start`

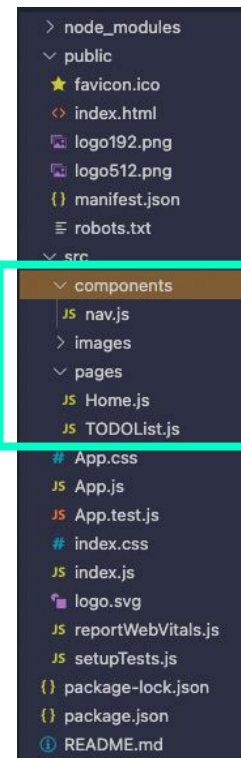
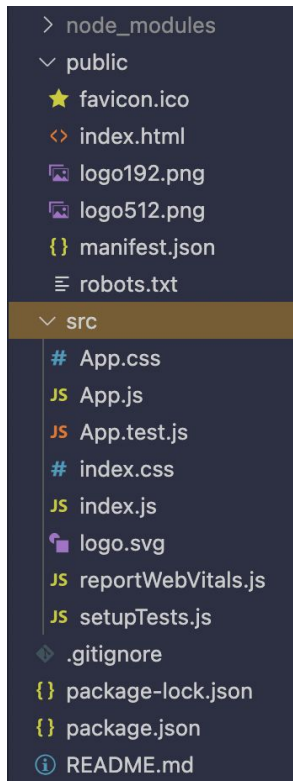


02

Passos iniciais e conceitos



# ESTRUTURA INICIAL



# INSTALAÇÃO DO BOOTSTRAP

- Parar a aplicação e correr, dentro do diretório da mesma, `npm install --save bootstrap`
- Dentro do ficheiro `index.js`, fazer o respetivo import - `import 'bootstrap/dist/css/bootstrap.css';`

# APP.JS

```
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;
```

# HOOKS

```
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>
  );
}
```

\*Retirado de <https://reactjs.org/docs/hooks-intro.html>

## Function and Class Components

The simplest way to define a component is to write a JavaScript function:

```
function Welcome(props) {  
  return <h1>Hello, {props.name}</h1>;  
}
```

This function is a valid React component because it accepts a single “props” (which stands for properties) object argument with data and returns a React element. We call such components “function components” because they are literally JavaScript functions.

You can also use an [ES6 class](#) to define a component:

```
class Welcome extends React.Component {  
  render() {  
    return <h1>Hello, {this.props.name}</h1>;  
  }  
}
```

The above two components are equivalent from React’s point of view.

Function and Class components both have some additional features that we will discuss in the [next sections](#).

# ATENÇÃO!

O uso de classes, em vez de funções, podem trazer problemas futuros, como por exemplo em chamadas dinâmicas a uma API.

\*Retirado de <https://reactjs.org/docs/components-and-props.html>



# 03

## Pôr em prática

Código disponível em: [https://github.com/sofiamoniz/aula\\_react](https://github.com/sofiamoniz/aula_react)



# Routing em App.js

# App.js

```
import React from 'react';
import './App.css';
import Nav from './components/nav';
import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";
import Home from './pages/Home';
import TODOList from './pages/TODOList';
import Drivers from './pages/Drivers';
import DriverDetails from './pages/DriverDetails';

function App() {
  return (
    <div className="App">
      <Router>
        <Nav />
        <Routes>
          <Route path="/" exact element={<Home />} />
          <Route path="/TODOList" element={<TODOList />} />
          <Route path="/Drivers" element={<Drivers />} />
          <Route path="/DriverDetails/:id" element={<DriverDetails />} />
        </Routes>
      </Router>
    </div>
  );
}

export default App;
```





# Components

# Navbar (nav.js)

Começar por instalar o React Router: *"npm install react-router-dom@6"* !

```
import React from 'react';
import '../App.css';
import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";

function Nav(){

  const navStyle={
    textDecoration: 'none'
  }

  return(
    <nav className="navbar navbar-expand-lg navbar-light bg-light">
      <div className="collapse navbar-collapse" id="navbarSupportedContent">
        <ul className="navbar-nav mr-auto">
          <li className="nav-item">
            <Link to="/" style={navStyle} id="nav_item">
              <a className="nav-link">Home</a>
            </Link>
          </li>
          <li className="nav-item">
            <Link to="/TODOList" style={navStyle} id="nav_item">
              <a className="nav-link">TODO list</a>
            </Link>
          </li>
          <li className="nav-item">
            <Link to="/Drivers" style={navStyle} id="nav_item">
              <a className="nav-link">Drivers</a>
            </Link>
          </li>
        </ul>
      </div>
    </nav>
  )
}

export default Nav;
```



# Pages

# Welcome to your ReactJs app!

# Home.js

```
import React from 'react';

function Home(){
  return(
    <div>
      <h1>Welcome to your ReactJs app!</h1>
    </div>
  )
}

export default Home;
```

# Your TODO list

Nothing to do!

# Your TODO list

Add

Fazer o trabalho de ITW

Delete

# TODOList.js

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

function TODOList(){
  const [tasks, setTasks] = useState([]);
  const [newTask, setNewTask] = React.useState(null);

  const addNewTask = () => {
    setTasks([...tasks, newTask]);
    setNewTask("");
  };

  const removeTask = (entry) => {
    setTasks(tasks.filter(item => item !== entry));
  };
}
```



# TODOList.js

```
const checkTasks = () => {
  if(tasks.length){
    return(
      <div>
        <div className="container">
          <div className="row" style={{marginTop:"5%}}>
            <div className="container">
              {tasks.map(entry =>
                <div className="row">
                  <div className="col-sm-6">
                    <p>{entry}</p>
                  </div>
                  <div className="col-sm-6">
                    <input type="button" value="Delete" name={entry} onClick={() => removeTask(entry)}>
                  </div>
                </div>
              )}
            </div>
          </div>
        </div>
      )
    }
    else{
      return(
        <p style={{marginTop:"5%}}>Nothing to do!</p>
      )
    }
  }
}
```

# TODOList.js

```
return (  
  <div>  
    <h1>Your TODO list</h1>  
    <div className="container">  
      <div className="row" style={{marginTop:"5%}}>  
        <div className="container">  
          <div className="row">  
            <div className="col-sm-6">  
              <input type="text" className="form-control input-normal" placeholder="Insert new task"  
                value={newTask} onChange={(event) => setNewTask(event.target.value)} />  
            </div>  
            <div className="col-sm-6">  
              <input type="button" onClick={addNewTask} value="Add" />  
            </div>  
          </div>  
        </div>  
      </div>  
      {checkTasks()}  
    </div>  
  </div>  
);  
}  
  
export default TODOList;
```

## Get to know the drivers

Name	Nationality
<a href="#">Adolf Brudes</a>	German
<a href="#">Adolfo Cruz</a>	Argentine
<a href="#">Adrián Campos</a>	Spanish
<a href="#">Adrian Sutil</a>	German
<a href="#">Aguri Suzuki</a>	Japanese
<a href="#">Al Herman</a>	American
<a href="#">Al Keller</a>	American
<a href="#">Al Pease</a>	Canadian
<a href="#">Alain de Changy</a>	Belgian
<a href="#">Alain Prost</a>	French
<a href="#">Alan Brown</a>	British
<a href="#">Alan Jones</a>	Australian
<a href="#">Alan Rees</a>	British

<a href="#">Al Herman</a>	American
<a href="#">Al Keller</a>	American
<a href="#">Al Pease</a>	Canadian
<a href="#">Alain de Changy</a>	Belgian
<a href="#">Alain Prost</a>	French
<a href="#">Alan Brown</a>	British
<a href="#">Alan Jones</a>	Australian
<a href="#">Alan Rees</a>	British
<a href="#">Alan Rollinson</a>	British
<a href="#">Alan Stacey</a>	British
<a href="#">Albert Scherrer</a>	Swiss
<a href="#">Alberto Ascari</a>	Italian
<a href="#">Alberto Colombo</a>	Italian
<a href="#">Alberto Crespo</a>	Argentine
<a href="#">Alberto Rodriguez Larreta</a>	Argentine

# Drivers.js

```
import React, {useEffect} from 'react';
import { Link } from "react-router-dom";

function Drivers(){

  const [driversList, setDriversList] = React.useState([]);
  const [page, setPage] = React.useState(1);
  const pageSize = 20;

  const fetchState = async () => {
    const fetchItem = await fetch('http://192.168.160.58/Formula1/api/Drivers?page=' + page + '&pagesize=' + pageSize);
    const item = await fetchItem.json();
    setDriversList(item.List);
  };

  const Td = ({ children, to }) => {
    // Conditionally wrapping content into a link
    const ContentTag = to ? Link : 'div';
    return (
      <td>
        <ContentTag to={to}>{children}</ContentTag>
      </td>
    );
  }
}
```

# Drivers.js

```
const retrieveDrivers = () => {
  if (driversList.length){
    return (
      <div>
        <table className="table">
          <thead>
            <tr>
              <th scope="col">Name</th>
              <th scope="col">Nationality</th>
            </tr>
          </thead>
          <tbody>
            {driversList.map(driver =>
              <tr key={driver.DriverId}>
                <td to={`/DriverDetails/${driver.DriverId}`}>{driver.Name}</td>
                <td>{driver.Nationality}</td>
              </tr>
            )}
          </tbody>
        </table>
        <nav aria-label="Page navigation">
          <ul className="pagination">
            <li className="page-item"><a className="page-link" onClick={changePageBackward}>Previous</a></li>
            <li className="page-item"><a className="page-link" onClick={changePageForward}>Next</a></li>
          </ul>
        </nav>
      </div>
    )
  }
  else{
    return(
      <div>
        <p>Waiting for API to fetch!</p>
      </div>
    )
  }
}
```

# Drivers.js

```
const changePageForward = () => {
  setPage(page+1);
  fetchState();
}

const changePageBackward = () => {
  if (page > 1){
    setPage(page-1);
    fetchState();
  }
}

useEffect(() => {
  fetchState();
}, [])

return(
  <div>
    <h1>Get to know the drivers</h1>
    {retrieveDrivers()}
  </div>
)
}

export default Drivers;
```

# Lewis Hamilton



Nationality: British

[Click here to know more](#)

## Races:

Circuit Name	Year	Race Name
Albert Park Grand Prix Circuit	2007	Australian Grand Prix



Races:

Circuit Name	Year	Race Name
Albert Park Grand Prix Circuit	2007	Australian Grand Prix
Bahrain International Circuit	2007	Bahrain Grand Prix
Circuit de Spa-Francorchamps	2007	Belgian Grand Prix
Autódromo José Carlos Pace	2007	Brazilian Grand Prix
Silverstone Circuit	2007	British Grand Prix
Circuit Gilles Villeneuve	2007	Canadian Grand Prix
Shanghai International Circuit	2007	Chinese Grand Prix
Nürburgring	2007	European Grand Prix
Circuit de Nevers Magny-Cours	2007	French Grand Prix
Hungaroring	2007	Hungarian Grand Prix
Autodromo Nazionale di Monza	2007	Italian Grand Prix
Fuji Speedway	2007	Japanese Grand Prix
Singapore International Circuit	2007	Malaysian Grand Prix

# DriverDetails.js

```
import React, {useEffect} from 'react';
import { useParams } from "react-router-dom";

function DriverDetails() {

  const { id } = useParams()
  const [driverDetails, setDriverDetails] = React.useState({});
  const [driverRaces, setDriverRaces] = React.useState({});

  const fetchState = async () => {
    const fetchItem = await fetch('http://192.168.160.58/Formula1/api/Drivers/Driver?id=' + id);
    const item = await fetchItem.json();
    setDriverDetails(item);
    setDriverRaces(item.Races);
  };

  const retrieveRaces = () => {
    if(driverRaces.length > 0){
      return (
        <div>
          <h3 style={{marginTop: '5%', marginBottom: '5%'}}>Races:</h3>
          <table className="table">
            <thead>
              <tr>
                <th scope="col">Circuit Name</th>
                <th scope="col">Year</th>
                <th scope="col">Race Name</th>
              </tr>
            </thead>
            <tbody>
              {driverRaces.map(race =>
                <tr key={race.RaceId}>
                  <td>{race.CircuitName}</td>
                  <td>{race.Year}</td>
                  <td>{race.Name}</td>
                </tr>
              )}
            </tbody>
          </table>
        </div>
      )
    }
  }
}
```


# DriverDetails.js

```
const retrieveImage = (image) =>{
  if(image != null){
    return(
      <img
        src={image}
        alt="new"
        style={{width: 200, height: 'auto', marginTop:10}}
      />
    )
  }
}

useEffect(() => {
  fetchState();
},[])

return(
  <div>
    <div>
      {driverDetails.Name || driverDetails.Nationality || driverDetails.Url ? (
        <div>
          <h1 style={{marginTop: '5%'}}>{driverDetails.Name}</h1>
          {retrieveImage(driverDetails.ImageUrl)}
          <p>Nationality: {driverDetails.Nationality}</p>
          <a href={driverDetails.Url}>Click here to know more</a>
          {retrieveRaces()}
        </div>
      ) : (
        <p>Waiting for API to fetch!</p>
      )}
    </div>
  </div>
)

export default DriverDetails;
```



# Fácil conversão para mobile, com React Native!

Guia para instalação:  
<https://reactnative.dev/docs/environment-setup>

04

Adorar React





# Obrigada!

Dúvidas?

E-mail: [sofiamoniz@ua.pt](mailto:sofiamoniz@ua.pt)

