



Licenciatura
Multimédia

Programação III

Frederico Fonseca

ffonseca@ismt.pt



Imagem retirada do website
<https://unsplash.com/photos/MuJHwDHbXUk>

Sumário

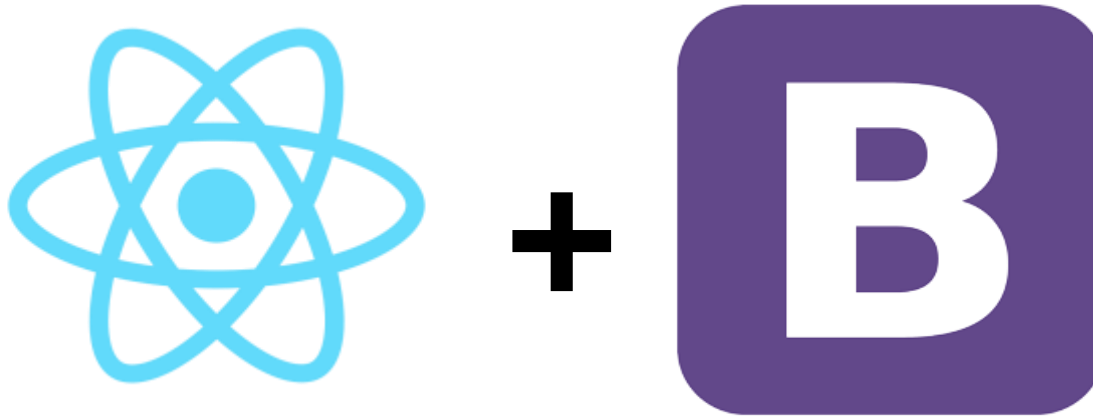
- Bootstrap em React
- React Router
- Axios

Texto de apoio:

Morgan, J. (2021). **How To Code in React.js** [E-book].

Consultado em <https://www.digitalocean.com/community/books/how-to-code-in-react-js-ebook>

React + Bootstrap



React + Bootstrap

- É possível tirar partido da *framework* **Bootstrap** na construção das UIs das nossas aplicações em React;
- Existe **duas formas possíveis** de o fazer:
 - Importar via CDN os ficheiros necessários (solução menos recomendada);
 - A importação dos ficheiros é feita no ficheiro 'index.html' na diretoria 'public';
 - Instalar o módulo '**bootstrap**' do NPM;

```
npm install @popperjs/core --save  
npm install bootstrap --save
```

React + Bootstrap

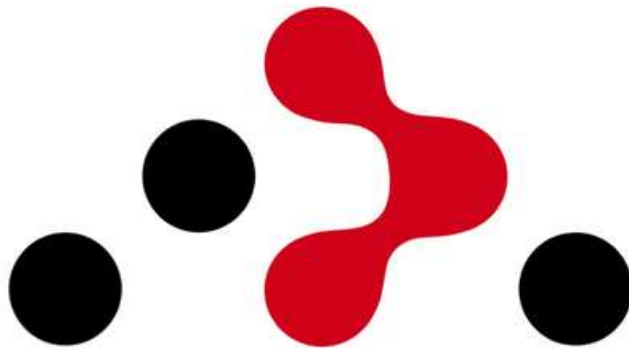
- Por último é necessário importar o(s) ficheiro(s) CSS e JS do *Bootstrap* no início do código do ficheiro principal (App.js):

```
import 'bootstrap/dist/css/bootstrap.min.css';  
import 'bootstrap/dist/js/bootstrap.min.js';
```

- No código do componente apenas necessitamos de definir as classes de acordo com a documentação do *Bootstrap*:

```
<nav className="navbar navbar-expand-lg navbar-light bg-light">  
  (...)  
</nav>
```

React Router



React Router

- Quando começamos a desenvolver a nossa aplicação em React, uma das primeiras coisas que precisamos de definir é a forma como vamos controlar as **rotas**;
 - A navegação da nossa aplicação;
- Por exemplo, as hiperligações dos menus de navegação, os erros (404, etc.), as páginas/componentes a serem invocadas quando um formulário é submetido, etc.;

- Em React utilizamos o módulo '**react-router-dom**' do NPM para definir e controlar as rotas da nossa aplicação;
 - Esta biblioteca utiliza a interface [History](#) dos browsers;
 - As rotas são definidas em **formato JSX**;
- Para instalar o módulo '**react-router-dom**' é necessário executar o seguinte comando:

```
npm install react-router-dom --save
```


- A sintaxe para definir uma rota inclui o:
 - **path** - caminho a ter em consideração;
 - **element** – elemento (componente, página, etc.) a ser renderizado;

```
<Routes>
  <Route path="/" element={ <App /> } />
  <Route path="*" element={ <NotFound /> } />
</Routes>
```

- O `path` aceita alguns formatos para definição da rota:

- Rota estática

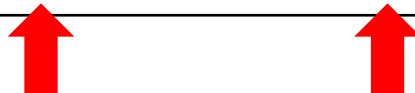
```
path='<caminho>'
```

- Rota dinâmica com parâmetros

```
path='<caminho>/:<parâmetro>'
```

- Rota dinâmica com parâmetros opcionais

```
path='<caminho>(/:<parâmetro>)'
```



- É necessário importar o módulo (*package*) no ficheiro principal:

```
import {BrowserRouter as Router, Routes, Route, Link} from 'react-router-dom';
```

- É necessário colocar o código de toda a nossa app dentro da tag `<Router>`, conforme o exemplo que se segue:

```
<Router>  
  //código já existente da nossa App  
</Router>
```

- Os parâmetros passados pelas rotas são recebidos pelos componentes através do método `useParams()`;

```
import React from 'react';  
import { useParams } from 'react-router-dom';  
  
const Pessoa = () => {  
  const {id} = useParams();  
  return (  
    <h1>ID: {id}</h1>  
  );  
};  
  
export default Pessoa;
```

- Vejamos um exemplo:

```
import React from 'react';
import ListaFilmes from './pages/filmes/list';
import CriarFilme from './pages/filmes/create';
import EditarFilme from './pages/filmes/edit';
import {BrowserRouter as Router, Routes, Route, Link} from 'react-router-dom';


function App() {
  return (
    <Router>
      <div className="App">
        <h1>Exemplo de Rotas em React</h1>
      </div>

      <Routes>
        <Route path="/" element={ <ListaFilmes /> } />
        <Route path="/create" element={ <CriarFilme /> } />
        <Route path="/edit/:id" element={ <EditarFilme /> } />
      </Routes>
    </Router>
  );
}

export default App;
```

- No caso do CRUD de coleções (*collections*) é costume agrupar-se as rotas, conforme é exemplificado no código seguinte;
- Propositadamente o caminho (*path*) da rota de editar e apagar filme está diferente do exemplo anterior;
 - Neste caso, a rota de editar ficaria '**filme/1/delete**';

```
<Route path="/filmes">
  <Route index element={ <ListaFilmes /> } />
  <Route path="create" element={ <CriarFilme /> } />
  <Route path=":id" element={ <VerFilme /> } />
  <Route path=":id/edit" element={ <EditarFilme /> } />
  <Route path=":id/delete" element={ <ApagarFilme /> } />
</Route>
```



Hiperligações

- Também é possível criar **hiperligações** com recurso ao `<Link>`;
 - Permite fazer a **navegação entre páginas**;
- A sintaxe deste componente é:
 - **to** - o destino da hiperligação, que pode ser passado via texto (*string*) ou objeto (*object*);
 - **activeStyle** (*opcional*) - nome do(s) estilo(s) a serem utilizados pelo elemento;
 - **activeClassName** (*opcional*) - nome do estilo CSS a ser utilizado pelo elemento quando a hiperligação tiver o estado 'active';

- Vejamos um exemplo utilizando o formato string:

```
<Link to='/utilizador/123'>Utilizador 123</Link>  
<Link to='/utilizador?id=123'>Utilizador 123</Link>
```

- Vejamos um exemplo utilizando o formato object:

```
<Link to={{pathname: "/utilizador",  
  search: "?id=123",  
  hash: "#user-123",  
  state: { fromDashboard: true }  
}}>Utilizador 123</Link>
```


- Sintaxe do formato *object*:

```
{  
  pathname: 'String representando o caminho para o link',  
  query: 'Um objeto com chave valor dos parâmetros da url',  
  hash: 'Uma hash para colocar na URL, ex: #uma-hash.',  
  state: 'state a ser persistido para o location'  
}
```

- Para mais propriedades consultar:

<https://github.com/ReactTraining/react-router/blob/master/packages/react-router-dom/docs/api/Link.md>

AXIOS

Axios - pedidos API

- O **Axios** é um cliente HTTP que funciona com [XMLHttpRequest](#) (AJAX) ou com a interface [http](#) do Node;
 - Resumidamente, o mesmo bloco de código pode ser utilizado para fazer pedidos em AJAX no *browser* e também no servidor;
 - Todos os pedidos retornam um *promise*, compatível com ES6;
- O Axios permite os métodos HTTP get, post, put, patch, delete e head;
 - `axios.request(config)`
 - `axios.get(url [, config])`

Axios – pedidos API

- `axios.post(url [, data [, config]])`
 - `axios.put(url [, data [, config]])`
 - `axios.patch(url [, data [, config]])`
 - `axios.delete(url [, config])`
 - `axios.head(url [, config])`
-
- Para instalar o módulo '**axios**' é necessário executar o seguinte comando no projeto React:


```
npm install axios --save
```

Axios - Sintaxe

- A sintaxe do pedido HTTP é:



```
axios.get('/edit?id=22')
  .then(function (response) {
    // handle success
    console.log(response);
  })
  .catch(function (error) {
    // handle error
    console.log(error);
  })
  .then(function () {
    // always executed
  });
```



```
axios.post('/edit', {
  id: 22
})
  .then(function (response) {
    // handle success
    console.log(response);
  })
  .catch(function (error) {
    // handle error
    console.log(error);
  });
  .then(function () {
    // always executed
  });
```

Axios - Exemplo

- Exemplo de um pedido GET a uma API:

```
import React, { useEffect, useState } from "react";
import axios from "axios";

export default function ExemploGet() {
  const [filmes, setFilmes] = useState([]);

  useEffect(() => {
    async function getMovies() {
      await axios
        .get(
          "https://api.themoviedb.org/3/movie/top_rated?api_key=<token>&language=pt-PT&page=1"
        )
        .then((res) => {
          setFilmes(res.data.results);
        })
        .catch((error) => {
          console.error("Ups! Ocorreu um erro - " + error);
        });
    }
    getMovies();
  }, []);

  return (
    <div>
      {filmes.map((filme) => {
        return (
          <p key={filme.id}>{filme.title}</p>
        );
      })}
    </div>
  );
}
```

Resultado do exemplo:

Os Condenados de Shawshank
दिलवाले दुल्हनिया ले जायेंगे
O Padrinho
Violeta Evergarden: O Filme
SEUS OLHOS DIZEM
A Lista de Schindler
Doulkyusei (Movie)
Gabriel's Inferno Part II
O Padrinho Parte II
Gabriel's Inferno
Gabriel's Inferno Part III
O Meu Nome
A Viagem de Chihiro
Parasitas
土圓
Doze Homens em Fúria
À Espera de Um Milagre
Filme Dado
O Cavaleiro das Trevas
Evangelho 3.0+1.0: Três Vezes

Axios - CORS

- Para ultrapassar a política de segurança *Cross-Origin Resource Sharing* (CORS) dos browsers é necessário adaptar os cabeçalhos dos pedidos HTTP para:

```
app.use((req, res, next) => {  
  res.header('Access-Control-Allow-Origin', '*');  
  res.header('Access-Control-Allow-Headers', 'Authorization, X-API-KEY, Origin, X-Requested-With, Content-Type, Accept, Access-Control-Allow-Request-Method');  
  res.header('Access-Control-Allow-Methods', 'GET, POST, OPTIONS, PUT, DELETE');  
  res.header('Allow', 'GET, POST, OPTIONS, PUT, DELETE');  
  next();  
});
```