

Introdução ao ReactJS

conhecendo a tecnologia ReactJS

| *"react é uma biblioteca para criação de interfaces (UI)"*

React foi criado pelo facebook, por Jordan Walke

O funcionamento do React é baseado em componentes

configurando o ReactJS

configurando o reactJS com `react scripts`

- ☐ abrir no diretório do projeto
- ☐ `npm init`
- ☐ `npm install --save react@17.0.2 react-dom@17.0.2 react-scripts@4.0.3`

configurando o reactJS com `create-react-app`

- ☐ Abrir no diretório do projeto
- ☐ executar os comandos abaixo

```
npx create-react-app my-app  
cd my-app  
npm start
```

JSX

JSX é uma linguagem de marcação junto com JS

Renderizando Elementos

<https://www.youtube.com/watch?v=aJR7f45dBNs>

ecossistema do react

- ☐ React Router
- ☐ Redux
- ☐ Material UI
- ☐ Ant-Design
- ☐ Storybook
- ☐ Gatsby
- ☐ Jest
- ☐ React i18n Next

Configurando o WebPack

<https://www.youtube.com/watch?v=LMCtGvLJT6c>

```
1  const path = require('path')
2  const HtmlWebPackPlugin = require("html-webpack-plugin");
3
4  module.exports = {
5    entry: './src/index.js',
6    output: {
7      path: path.resolve(__dirname, 'dist'),
8      filename: 'bundler.js'
9    },
10   module: {
11     rules: [
12       {
13         test: /\.jsx?$/,
14         exclude: /node_modules/,
15         use: {
16           loader: "babel-loader"
17         }
18       }
19     ]
20   },
21   plugins: [
22     new HtmlWebPackPlugin({
23       template: './src/index.html',
24       filename: './index.html'
25     })
26   ]
27 }
```

Instalação e configuração do ESLint

```
npm install --save-dev eslint babel-eslint eslint-plugin-react eslint-watch
```

Renderização condicional

Renderizar um elemento dependendo do estado da aplicação

if inline

```
{condition && (component)}
```

if-else inline

```
{condition ? (component1) : (component2)}
```

Também é possível usar o if statemente normal

```
if (condition) {  
  {component1}  
} else if (condition){  
  {component2}  
} else {  
  {component3}  
}
```

Listas e chaves

```
const clients = [  
  {  
    id: 1,  
    nome: 'osvaldo'  
  },  
  {  
    id: 2,  
    nome: 'Leonardo'  
  }  
];  
  
const renderClients = ({id, nome}) => {  
  return (  
    <ul type="square">  
      <li key={`client-${id}`}>cliente {id}</li>  
      <li key={`client-${id}`}>nome {nome}</li>  
    </ul>  
  )  
}
```

```
    </ul>
  );
}

clients.map(renderClients);
```

resultado

- cliente 1
- nome osvaldo
- cliente 2
- nome Leonardo

Aviso: Chaves devem ser únicas apenas entre elementos irmãos.

Manipulando Eventos

```
<button onClick={funcaoqualquer}></button>
```

Dicas para pensar do Jeito ReactJS

- ☐ Comece com um mock
- ☐ Separe em uma Hierarquia de componentes
- ☐ Crie uma versão estática em React
- ☐ Identifique a representação mínima(mas completa) do State da UI
- ☐ Identifique onde o state deve ficar
- ☐ Adicione o fluxo de dados inverso

Deixe as coisas mais modulares e abstratos possíveis