

REACT JS



OLHO NO CARRO
CONSULTA VEICULAR



Checktudo
Para a melhor decisão



Hello!

Eduardo Henrique Rotundaro

✉ eduardo.rotundaro@olhonocarro.com.br

in linkedin.com/eduardorotundaro




OLHO NO CARRO
CONSULTA VEICULAR

Checktudo ✓
Para a melhor decisão












Conteúdo do curso

- ☐ Mundo Web
- ☐ Curiosidades
- ☐ JSX
- ☐ State
- ☐ React Router
- ☐ Redux
- ☐ Tecnologias
- ☐ JS declarativo e reativo
- ☐ Componentes
- ☐ Presentational components
- ☐ Axios
- ☐ O React
- ☐ Manipulação do DOM
- ☐ Props
- ☐ Container components
- ☐ Finalizando a aplicação

 To-Do Tarefas About

Tarefas - Cadastro



Descrição	Ações
Treinamento de React.js	 
Estudar Node.js	 
Estudar MongoDB	 
Estudar Vue.js	 

Aplicação - página de to-dos



Sobre - Informações



To-Do App

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Aplicação - página sobre

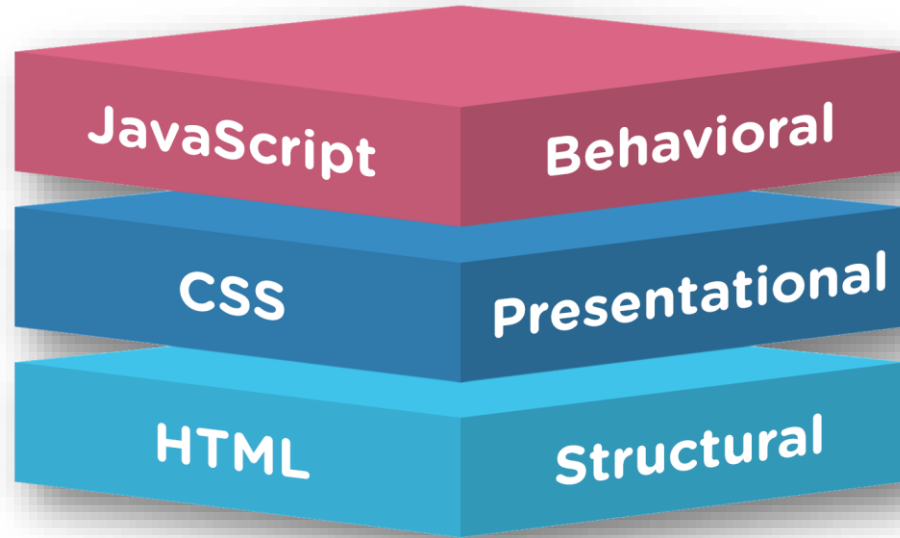


1

A web



.html + .js + .css





Frontend

- É a parte da aplicação que vai interagir diretamente com o usuário
- Qualquer página de um site faz parte do frontend de uma aplicação



Backend

- ◉ De forma geral, é o responsável pela implementação da regra de negócio
- ◉ Faz a comunicação da aplicação cliente com o banco de dados



HTTP requests

- ⦿ Hypertext Transfer Protocol
- ⦿ Métodos utilizados para enviar e receber informações em aplicações web
- ⦿ Alguns métodos: GET, POST , PUT, DELETE, PATCH



DOM

- ◉ Document Object Model
- ◉ É uma representação estruturada de um documento html criada pelo browser, que define um meio pelo qual a estrutura pode ser acessada e manipulada

2

Tecnologias



Bootstrap

- É uma ferramenta gratuita para desenvolvimento componentes de interfaces para sites e aplicações web
- *w3schools.com/bootstrap4/*

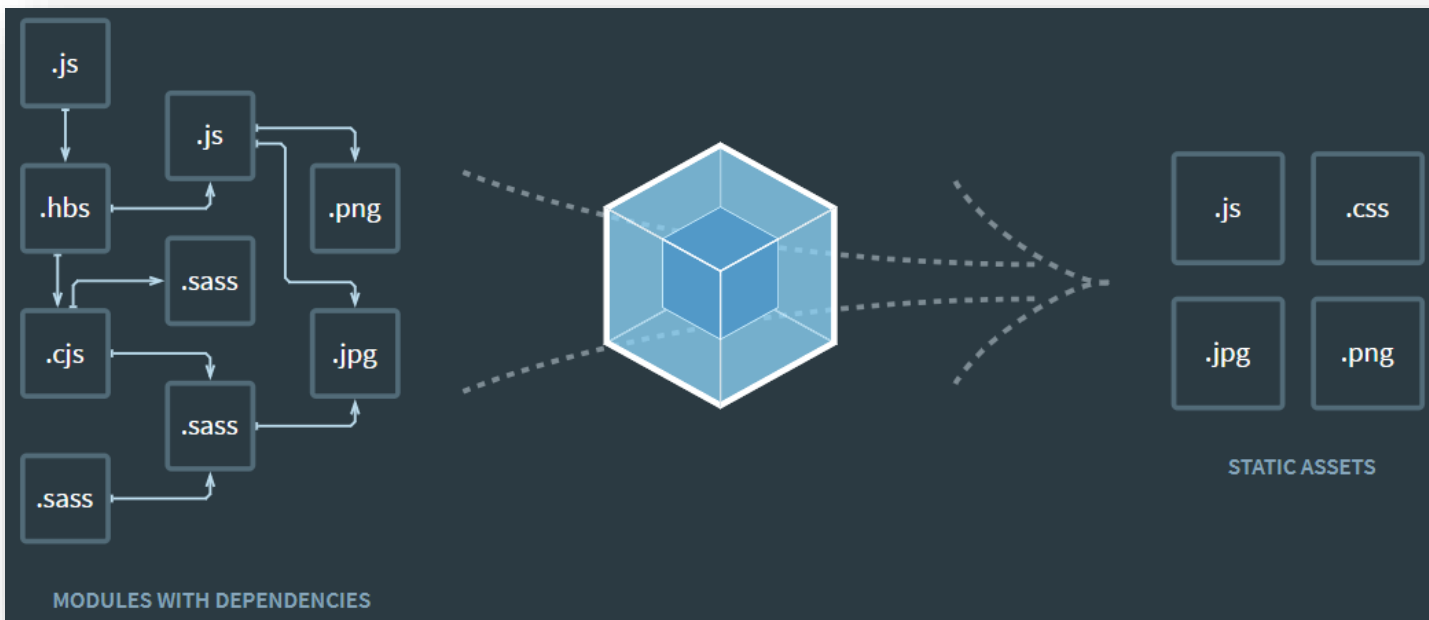


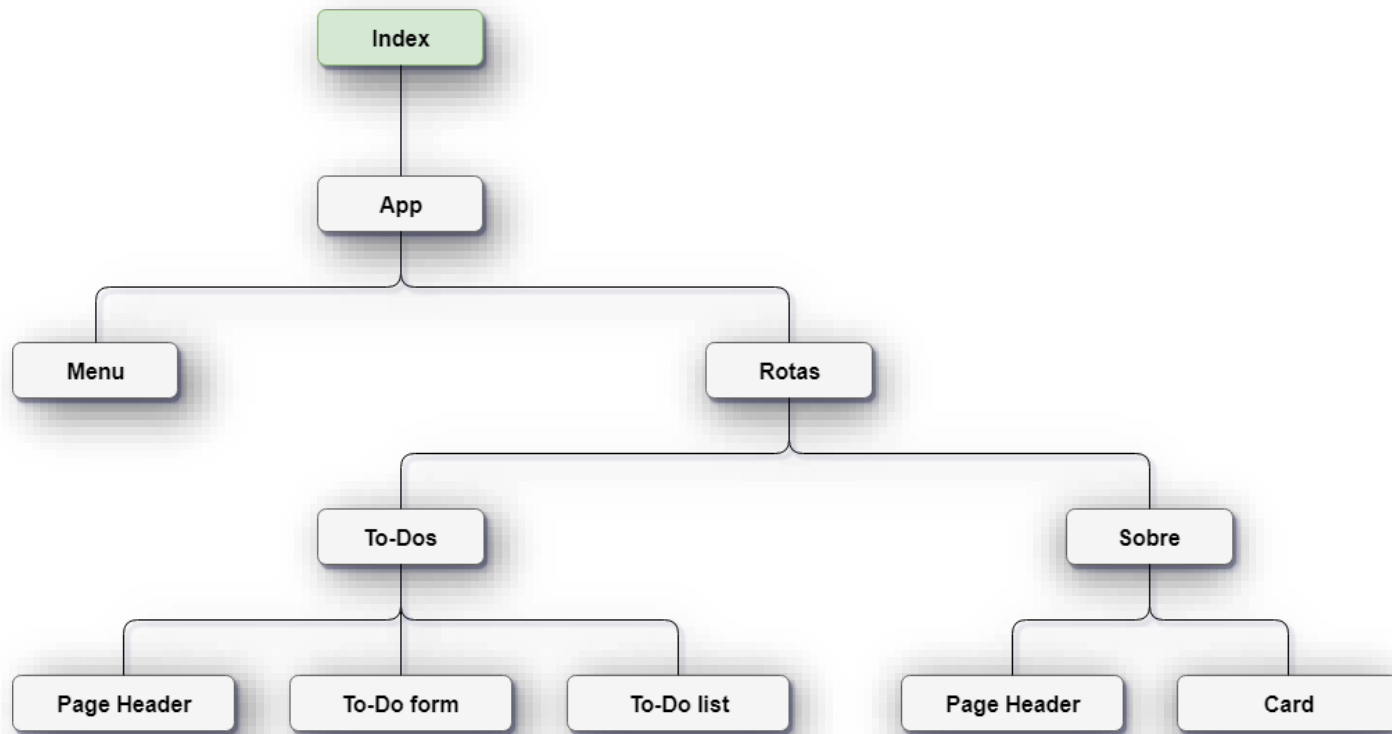
Font Awesome

- Conjunto de ferramentas de fontes e ícones baseados em CSS
- *fontawesome.com/*



Webpack





Estrutura da aplicação



2.1

Configuração

```
> cd backend  
> npm install  
  
> cd ..  
> cd frontend  
> npm install
```

Instalação de dependências



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="app.css">
    <title>To-Do App</title>
  </head>
  <body>
    <div id="app" class="container"></div>
    <script src="app.js"></script>
  </body>
</html>
```

index.html



3

O React

*Uma biblioteca declarativa, eficiente
e flexível para criação de interfaces
de usuário.*

“



Origem

- Tecnologia mantida pelo Facebook, Instagram e uma comunidade de desenvolvedores
- Foi criado como 'solução' para problemas que outros frameworks apresentavam, relacionados à performance



O React

- É uma biblioteca, ou seja, uma coleção de funcionalidades relacionadas, que são utilizadas para resolver problemas específicos
- No caso do React, a criação de interfaces de usuário reaproveitáveis



Libs

- O React sozinho não é capaz de construir uma aplicação complexa
- Então, as aplicações React serão compostas por várias bibliotecas especializadas

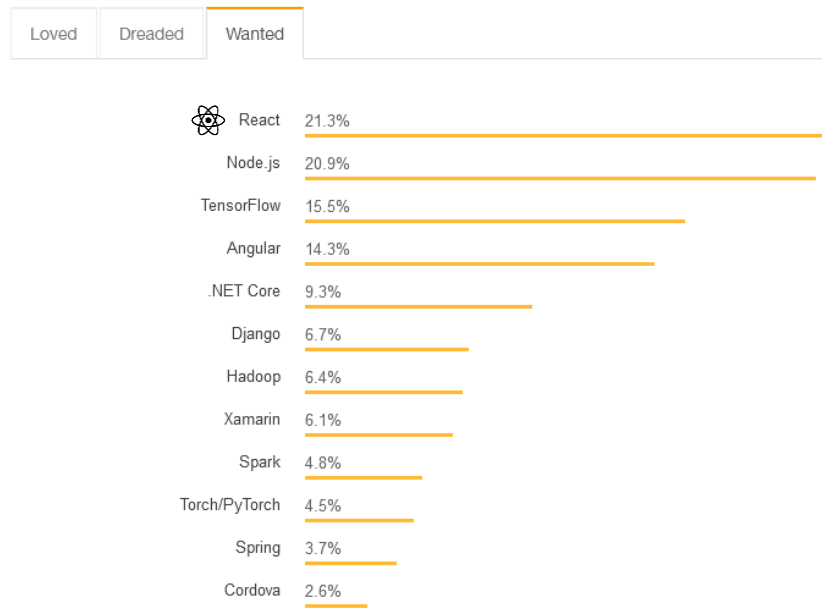
4

Curiosidades

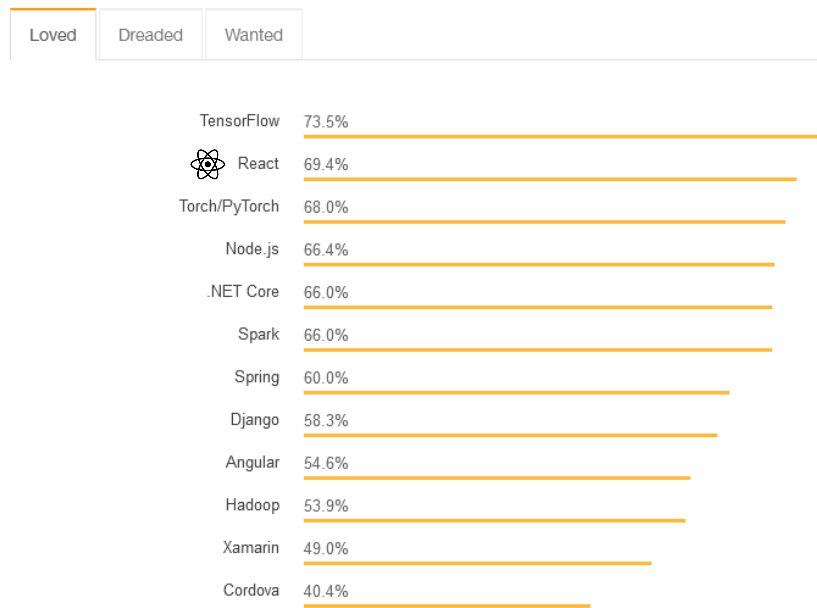


StackOverflow Developer Survey 2018

Most Loved, Dreaded, and Wanted Frameworks, Libraries, and Tools

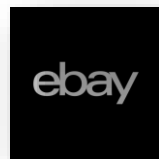


Most Loved, Dreaded, and Wanted Frameworks, Libraries, and Tools





Utilização



5

JS declarativo e reativo



Javascript declarativo

- O desenvolvedor precisa apenas focar no resultado, e não na forma como é atingido
- O que na prática, resulta em um aplicação funcional com pouco código Javascript



Javascript reativo

- O ponto forte do React é a criação de componentes reutilizáveis, que possuem dois atributos principais: state e props
- Sempre que o estado de um componente é alterado, ele é renderizado

6

React DOM



Manipulação do DOM

- A manipulação do DOM é um dos principais causadores de gargalos de performance em site e web apps
- O React propoe a criação do seu próprio DOM, mais eficiente e onde os componentes vivem



Virtual DOM

- O React possui uma representação virtual do html na memória - o virtual DOM
- Toda vez que um estado muda e tem-se uma nova árvore html para ser escrita no DOM, o React vai apenas escrever as diferenças



Performance

- O React compara o virtual DOM com uma imagem do DOM feita antes da atualização e descobre o que mudou, atualizando somente os componentes que mudaram de estado
- Este processo, conhecido como reconciliação de árvores, resulta em um enorme ganho de performance

7

JSX



A sintaxe

- É uma especificação de sintaxe para escrever Javascript como se estivesse escrevendo html
- O browser não consegue interpretar o JSX, então é necessário usar um transpilador, como por exemplo o Babel

```
import React from 'react'  
import ReactDOM from 'react-dom'  
import App from './main/app'  
  
ReactDOM.render( <App></App>, document.getElementById('app') )
```

index.jsx



8

Componentes



Tudo é componente

- O modo como o React trabalha para criar as Uis é por meio da quebra de toda a estrutura da aplicação em componentes
- O objetivo é separar todo o código em pequenas partes, ou seja, em arquivos diferentes



Máxima reutilização

- Isso torna possível aumentar ao máximo a reutilização desses componentes na aplicação
- Os componentes React:
 - Possuem um estado e propriedades
 - Podem conter outros componentes

Tarefas - Cadastro

Adicione uma tarefa



Descrição

Ações

Treinamento de React.js



Estudar Node.js



Estudar MongoDB



Estudar Vue.js



Menu



```
import React from 'react'

export default props => (
  <nav className="navbar navbar-expand-sm bg-dark navbar-dark">
    <a className="navbar-brand" href="#">
      <i className="fa fa-calendar-check-o"></i> To-Do
    </a>
    <ul className="navbar-nav">
      <li className="nav-item">
        <a className="nav-link" href="#/todos">Tarefas</a>
      </li>
      <li className="nav-item">
        <a className="nav-link" href="#/about">About</a>
      </li>
    </ul>
  </nav>
)
```

menu.jsx



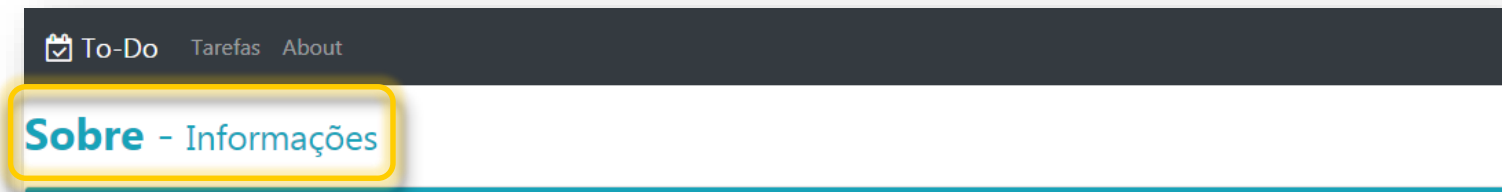
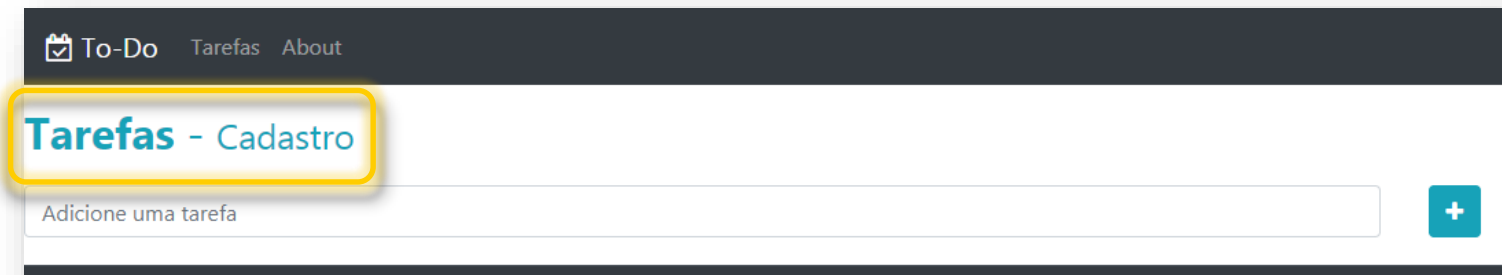
9

Props



Parâmetros

- Props (abreviação de 'properties') são informações que podem ser passadas para um componente
- Podem strings, números ou mesmo uma função



Page Header



```
import React from 'react'

export default props => (
  <header className='page-header mrg-u-b mrg-t mrg-b'>
    <h2 className="text-info">
      <strong>{props.name}</strong>&nbsp;-&nbsp;<small>{props.small}</small>
    </h2>
  </header>
)
```

pageHeader.jsx

</>

Sobre - Informações



To-Do App

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Aplicação - página sobre

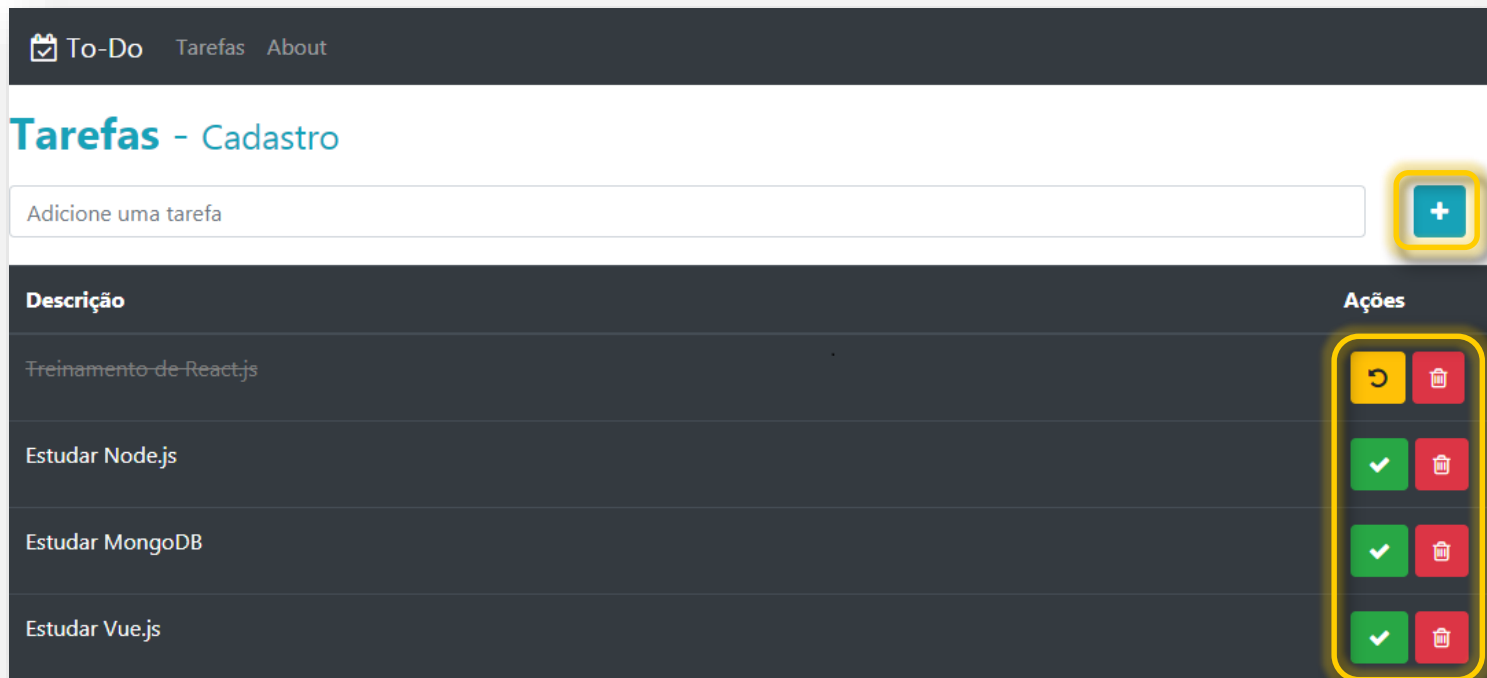


```
import React from 'react'
import PageHeader from '../components/pageHeader'

export default props => (
  <div>
    <PageHeader name="Sobre" small="Informações"></PageHeader>
    <div className="card">
      <div className="card-img-top bg-info">
        <center><i className="fa fa-3x fa-calendar-check-o aboutImg text-white"></i></center>
      </div>
      <div className="card-body bg-dark text-white">
        <h4 className="card-title">To-Do App</h4>
        <p className="card-text">
          Lorem ipsum dolor...
        </p>
      </div>
    </div>
  </div>
)
```

about.jsx

</>



Botões



```
import React from 'react'

export default props => {
  if(props.test) return props.children
  else return false
}
```

if.jsx



```
import React from 'react'
import If from './if'

export default props => (
  <If test={!props.hide}>
    <button className={'mrg-1 btn btn-' + props.style} onClick={props.onClick} >
      <i className={'fa fa-' + props.icon }></i>
    </button>
  </If>
)
```

iconButton.jsx

</>

10

State



Dados

- O estado da aplicação pode ser definido como o lugar de onde os dados vem e se transformam ao longo do tempo
- Os componentes do React podem ser categorizados em: presentational (sem estado) e container (com estado)

11

Presentational components



Sem estado

- Componentes do tipo presentational se importam apenas com a apresentação dos dados
- Podem ser escritos utilizando uma função simples

12

Container components



Com estado

- Componentes do tipo container tem que lidar com alguma lógica ou transformação de dados, e não só a apresentação dos dados
- Por esse motivo necessitam de um estado, e devem ser uma classe

13

React-router



Páginas da aplicação

- Biblioteca completa para o controle de rotas na aplicação
- Permite a configuração através do formato do JSX

```
import React from 'react'
import { Router, Route, Redirect, hashHistory } from 'react-router'

import Todo from '../pages/todo/todo'
import About from '../pages/about/about'

export default props => (
  <Router history={hashHistory} >
    <Route path='/todos' component={Todo} ></Route>
    <Route path='/about' component={About} ></Route>
    <Redirect from='*' to='/todos' ></Redirect>
  </Router>
)
```

router.jsx



14

Axios



Consumindo a API

- Biblioteca simples e completa, que vai permitir consumir APIs
- É através desta biblioteca que a aplicação vai se comunicar com o backend



Endpoints

- ◉ **GET** /server/to-do/ (retorna todos os itens to-do)
- ◉ **POST** /server/to-do/new (cria um item to-do)
- ◉ **DELETE** /server/to-do/remove/:id (remove um item to-do)
- ◉ **PUT** /server/to-do/set-as-done/:id (marca como concluído)
- ◉ **PUT** /server/to-do/set-as-pending/:id (marca como pendente)

```
import axios from 'axios'
```

```
const URL = 'http://localhost:4040/server/to-do/'
```

todo.jsx




```
refresh(){
  axios.get(URL)
  .then( (resp)=>{
    this.setState( { ...this.state, description: '', list: resp.data.data } )
  })
}

handleAdd(){
  const description = this.state.description

  axios.post(`${URL}new`, { description } )
  .then((resp)=>{
    this.refresh()
  })
  .catch((err)=>{
    alert(err.message)
  })
}
```

todo.jsx



```
handleDelete(todo){
  axios.delete(`${URL}remove/${todo._id}`)
  .then((resp)=>{
    this.refresh()
  })
  .catch((err)=>{
    alert(err.message)
  })
}

handleSetAsDone(todo){
  axios.put(`${URL}set-as-done/${todo._id}`)
  .then((resp)=>{
    this.refresh()
  })
  .catch((err)=>{
    alert(err.message)
  })
}
```

todo.jsx



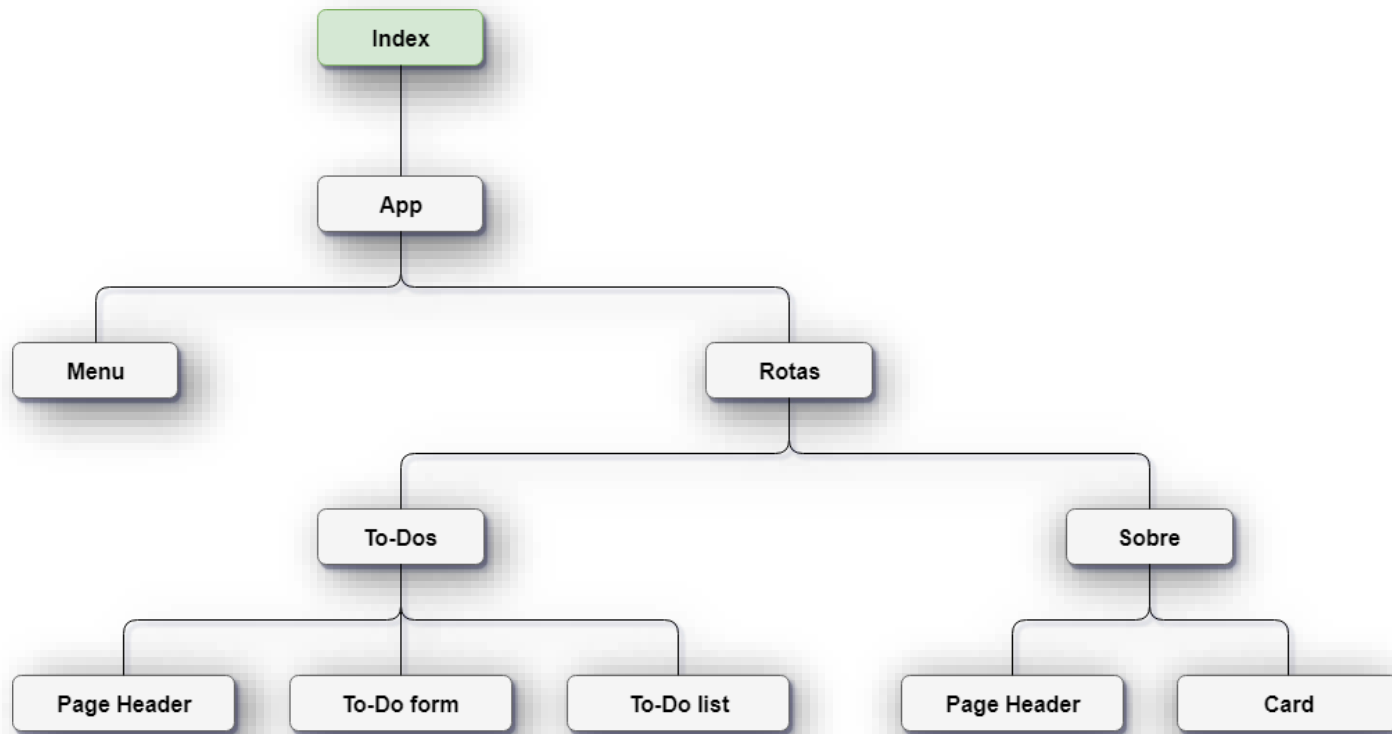
```
handleSetAsPending(todo){  
  axios.put(`${URL}set-as-pending/${todo._id}`)  
    .then((resp)=>{  
      this.refresh()  
    })  
    .catch((err)=>{  
      alert(err.message)  
    })  
}
```

todo.jsx



15

Finalizando a aplicação



Estrutura da aplicação



```
import 'modules/bootstrap/dist/css/bootstrap.min.css'
import 'modules/font-awesome/css/font-awesome.min.css'
import '../styles/custom.css'

import React from 'react'

import Routes from './routes'
import Menu from '../components/menu'

export default props => (
  <div className="container">
    <Menu></Menu>
    <Routes></Routes>
  </div>
)
```

app.jsx



Tarefas - Cadastro

Adicione uma tarefa



Descrição

Ações

Treinamento de React.js



Estudar Node.js



Estudar MongoDB



Estudar Vue.js



Botões



Tarefas - Cadastro

Adicione uma tarefa



Descrição

Ações

Treinamento de React.js



Estudar Node.js



Estudar MongoDB



Estudar Vue.js



Botões



Em frontend/

```
> npm start
```

No browser

<http://127.0.0.1:8080/>

Instalação de dependências



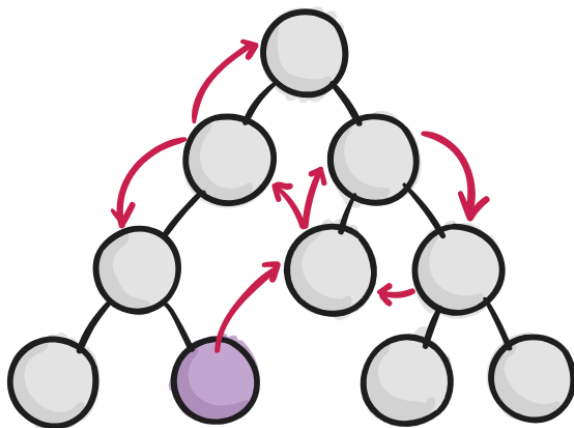
16

Redux

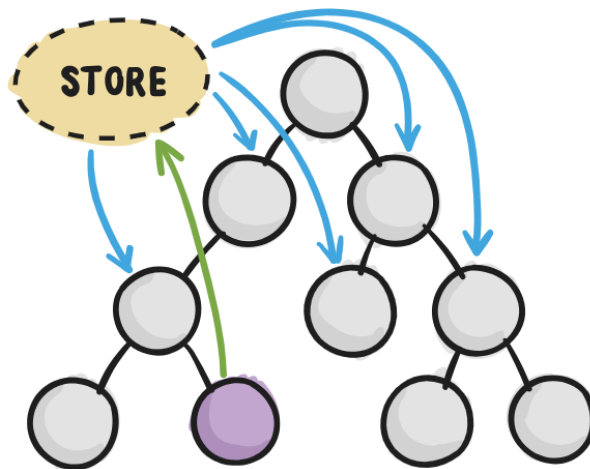


O poder do Redux

WITHOUT REDUX



WITH REDUX



COMPONENT INITIATING CHANGE



Obrigado!

✉ eduardo.rotundaro@olhonocarro.com.br

in linkedin.com/eduardorotundaro

github github.com/EduardoRotundaro/to-do-app-react-js

github github.com/EduardoRotundaro/crud-api-express-mongo



OLHO NO CARRO
CONSULTA VEICULAR

Checktudo ✓
Para a melhor decisão