

React

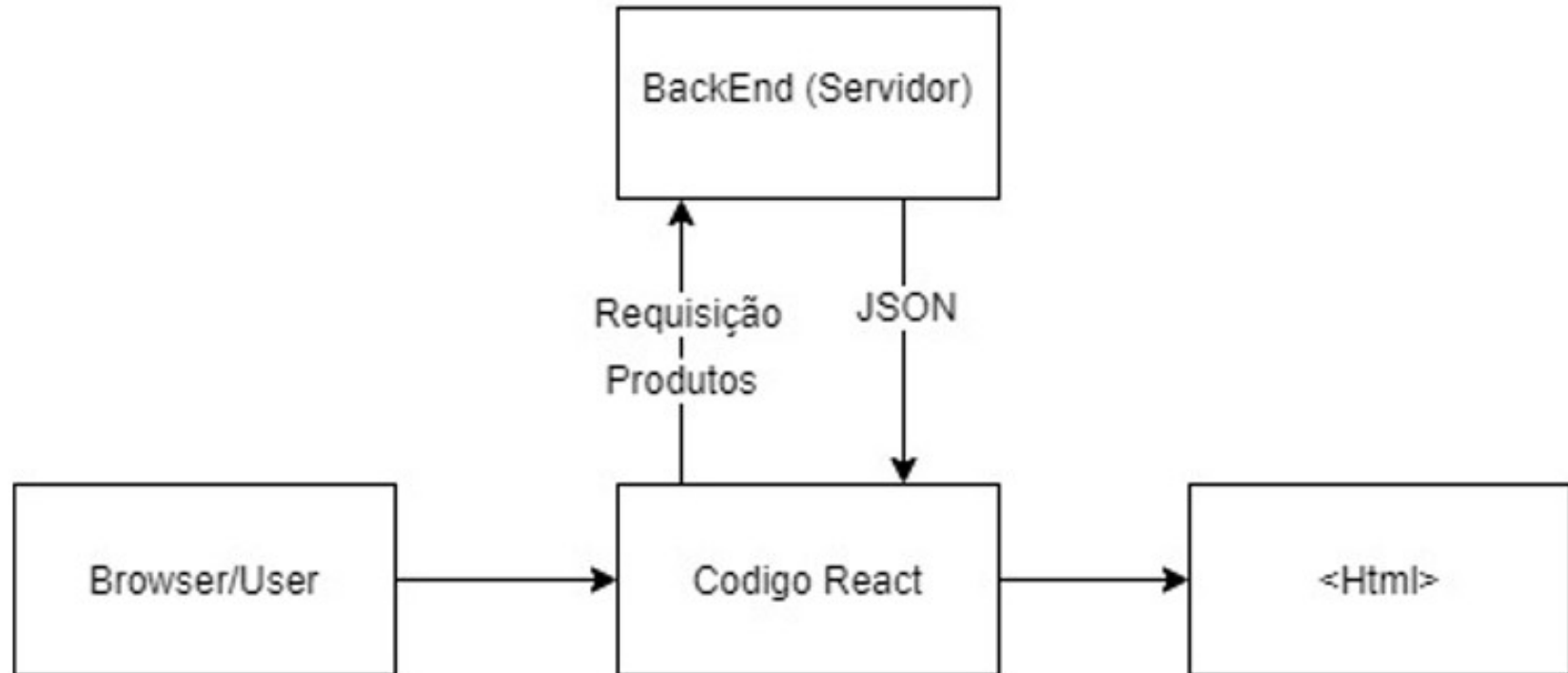
O que é React?

- React é uma biblioteca JavaScript, para desenvolvimento de aplicações front-end.
- Apresentada e mantida pelo Facebook.
- Foi desenvolvido para criar interfaces de usuário de maneira simples e eficiente.
- Baseado em componentes reutilizáveis.
- Criar visualizações simples rápida e eficiente.
- O React gerencia os estados, e conforme os dados mudam ele atualiza e renderiza apenas os componentes necessários.

React

- O React usa SPA (Single Page Application) são aplicações Web que tem como objetivo trazer uma experiência similar a de um APP desktop para o usuário (sem os reloads e etc comuns aos sites)

SPA - Single Page Application

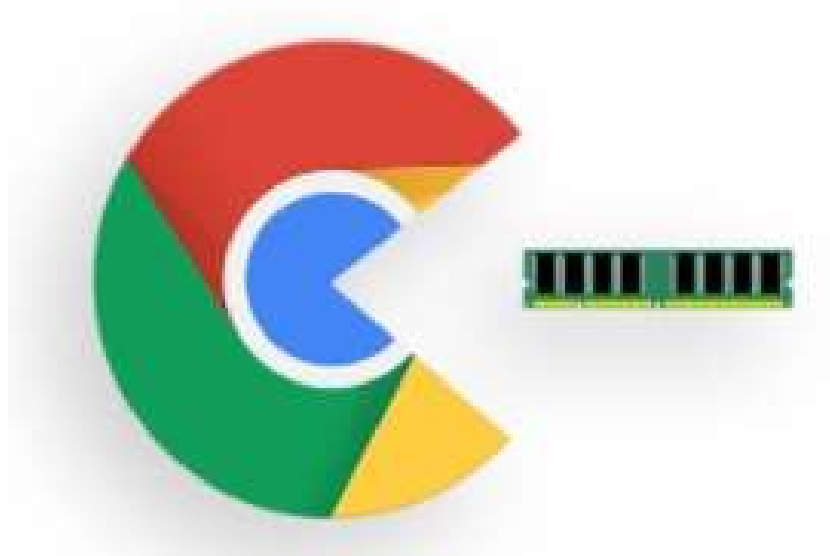


Vantagens SPA

- É possível prover interações na aplicação sem a necessidade de recarregamento, oferecendo uma experiência muito mais fluída para o usuário;
- Pode ocasionar “alívio” ao servidor da aplicação, já que não há a necessidade de se realizar uma requisição ao servidor a cada vez que é necessário carregar um novo conteúdo.

Desvantagens SPA

- Performance imprevisível: podem ocorrer algumas inconsistências justamente pelo deslocamento do esforço de renderização para o cliente.



O que é Node.js?

- O Node.js é uma runtime de JavaScript.
- Ou seja, uma biblioteca utilizada por um compilador durante a execução do programa.
- Que está construída na V8 engine (escrita em C++) da Google.
- Possibilitando criar softwares em JS no lado servidor.
- Temos então um código JS rodando em C++ para garantir alta performance.

O que é npm?

- O npm é um gerenciador de pacotes do Node.
- Vamos poder utilizar bibliotecas de terceiros, baixando elas pelo npm.
- E também executar determinados scripts no nosso programa.
- Dificilmente um software em Node.js não utiliza o npm.
- Os módulos externos ficam numa pasta chamada node_modules.
- Ela deve ser descartável, ou seja, a cada instalação do projeto baixamos todos os pacotes novamente.

Instalação do Node Windows

- O download do Node.js é feito no site oficial: nodejs.org.
- Vamos baixar um arquivo .msi, que é o instalador.
- É interessante saber que o npm vem junto com o Node.
- Após a instalação podemos testar o Node e o npm em um terminal, para validar a instalação.
- Note que os binários precisam estar no PATH, este passo está na instalação do Node.

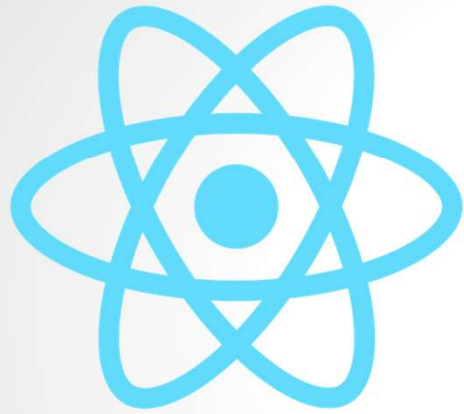
Instalação do Node Linux

- O download do Node.js é feito no site oficial: nodejs.org.
- Execute o comando “sudo apt install nodejs”.
- Após a instalação execute “sudo apt install npm”.
- Após a instalação podemos testar o Node e o npm em um terminal, para validar a instalação.

Fazendo download de um editor

- <https://code.visualstudio.com/>





Primeiro Projeto React

Instalação

- \$ npx create-react-app liga-da-justica

package.json

- É como se fosse uma pagina do caderno de receitas.
- Informações principais do nosso projeto.
- Gerencia as dependências do nosso projeto.

package.json

- Se você escrever `~0.13.0`, você quer somente atualizar patch releases: 0.13.1 está ok, mas 0.14.0 não.
- se você escrever `^0.13.0`, você quer atualizar patches e menor releases: 0.13.1, 0.14.0 e assim por diante.
- se você escrever `0.13.0`, você somente usará esta versão exata do pacote.
- `*0.13.0` qualquer versão do pacote.
- `latest`: a versão mais recente do pacote.
- `>`, `>=`, `<`, `<=1.0`: a versão deve ser superior, superior ou igual, inferior, inferior ou igual à 1.0, respectivamente.

Biblioteca

- \$ npm install react-dom
- Forma da gente trabalhar com react na web.
- Manipulação do nosso JS no HTML.
- Porque React e ReactDOM são separados?
- React serve para criar vários tipos de aplicações e não apenas aplicações para web.
- Exemplo: Aplicativos, Sistemas Desktop, etc.

Pasta

- Onde fica todo código criado por nos da nossa aplicação.
- \$./src

Babel

- Babel é uma ferramenta pra converter nosso código JS para uma maneira que permite todos os browsers entendam.
- What?
- JS é uma linguagem que atualiza muito, então muitas funcionalidades novas que usamos no react os navegadores não entendem.
- **Babel** converte esse código JS para outro JS porém de uma maneira que os navegadores entendam.

Babel

- <https://babeljs.io/>
- Exemplo de código;

```
let pessoa = { nome: 'Chinês'}; console.log(pessoa?.email)
```

JSX

- É o HTML do React.
- Onde vamos declarar as tags de HTML que serão exibidas no navegador.
- Ficam no return do componente.
- Temos algumas diferenças do HTML, por exemplo: class será className.

Webpack

- Na maioria dos projetos é utilizado junto com o babel.
- *No nosso projeto importamos arquivos js e outros como .css, .img, .png, .jpg .less .sass;*
- O webpack estipula uma serie de configurações que agente chama de loader, que ele vai ensinar pra nossa aplicação como que deve tratar cada um desses arquivos, ou converter de maneira entendíveis pelo browser.

Index.jsx

- Adicionar esse código

```
import { createRoot } from 'react-dom/client';  
  
import { App } from './src/App'  
  
const container = document.getElementById('root');  
  
const root = createRoot(container);  
  
root.render(<App />);
```

- createRoot: Cria uma raiz para o container e renderiza React dentro da div root.

