

IOS – Instituto de
Oportunidade Social

React 02 - Instalação do React



- Compreender a Instalação do React;
- Criar a primeira aplicação em React;

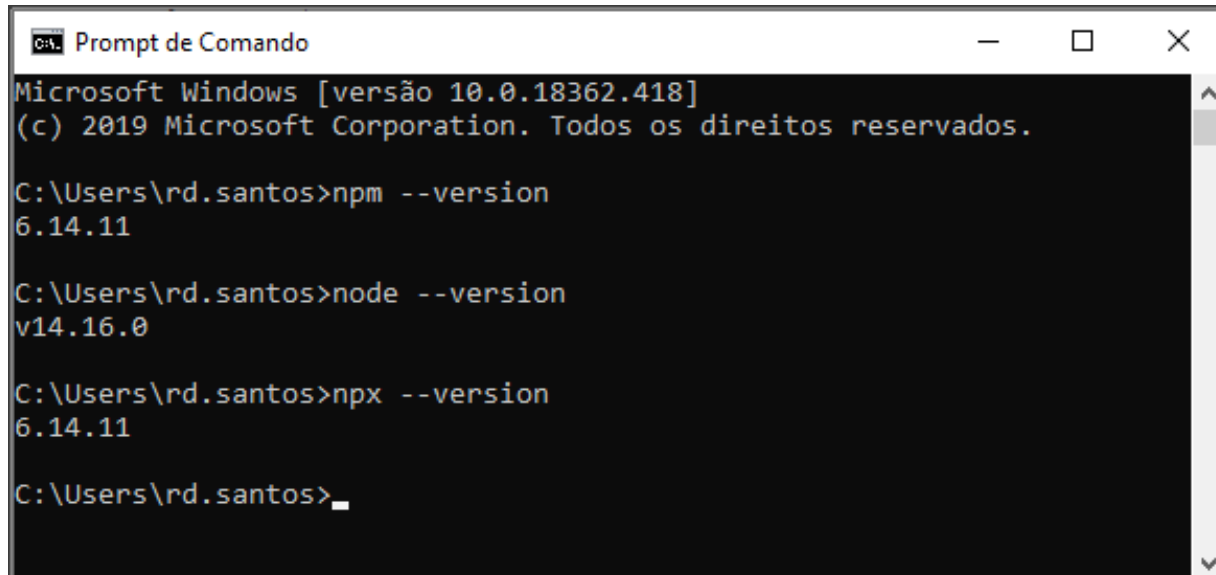
IOS – Instituto de
Oportunidade Social

Criando aplicação em React



Criando aplicação em React

Antes de criar qualquer aplicação em React, você precisa ter o software **Node.js** instalado no seu computador. O Node.js é um software de código aberto que permite **executar** códigos **JavaScript** **fora de um navegador web**. Ele pode ser instalado nos sistemas operacionais Windows, Linux e MacOS.



```
C:\> Prompt de Comando

Microsoft Windows [versão 10.0.18362.418]
(c) 2019 Microsoft Corporation. Todos os direitos reservados.

C:\Users\rd.santos>npm --version
6.14.11

C:\Users\rd.santos>node --version
v14.16.0

C:\Users\rd.santos>npx --version
6.14.11

C:\Users\rd.santos>_
```

Criando aplicação em React

Onde baixar o Node.js

nodejs.org/pt-br/download/

Downloads

Versão LTS Mais Recente: **16.15.0** (includes npm 8.5.5)

Baixe o código fonte do Node.js ou um instalador pré-compilado para o seu sistema, e comece a desenvolver hoje.

LTS
Recomendado Para a Maioria dos Usuários

Atual
Recursos Mais Recentes


Instalador Windows
node-v16.15.0-x64.msi


Instalador macOS
node-v16.15.0.pkg


Código fonte
node-v16.15.0.tar.gz

Instalador Windows (.msi)	32-bit	64-bit
Binário para Windows (.zip)	32-bit	64-bit
Instalador macOS (.pkg)	64-bit / ARM64	
Binário para macOS (.tar.gz)	64-bit	ARM64
Binários para Linux (x64)	64-bit	
Binários para Linux (ARM)	ARMv7	ARMv8
Código fonte	node-v16.15.0.tar.gz	

Criando aplicação em React

Precisamos do Node.js para utilizarmos os gerenciadores pacotes (bibliotecas) **NPM (Node Package Manager)** e **NPX (Node Package Execute)** e também para executarmos os códigos JavaScript necessários para rodar uma aplicação React. O software Node.js está disponível para download em: <https://nodejs.org/pt-br/>.

O **NPX** será utilizado para **criar a aplicação React**, enquanto o **NPM** será utilizado para **instalar bibliotecas** de terceiros, que podem ser necessárias e não são instaladas por padrão ao criar a aplicação React.



Importante! O fato de você desenvolver a sua aplicação React e executá-la em um ambiente Node.js, faz-se necessário o uso de muito recurso computacional. Isto é, a execução e atualização de uma aplicação React no navegador pode ficar lenta em computadores com pouco poder computacional.

IOS – Instituto de
Oportunidade Social

Instalando aplicação React



Instalando Aplicação React



Para instalar/criar uma aplicação React você deve escolher um **diretório de workspace** no VS Code. Por exemplo, no nosso caso todas as aplicações serão criadas no **diretório React** e esse diretório.

Escolhido o diretório local, você deve usar o seguinte comando no terminal da sua IDE para criar a sua aplicação React:

`npx create-react-app my-app`

Onde: **my-app** deve ser substituído pelo **nome da sua aplicação** que será criada. O nome da aplicação deve conter apenas **letras minúsculas** e **não pode** conter **espaços em branco** ou **caracteres especiais**.

Instalando Aplicação React



Assim podemos executar no terminal podemos por exemplo executar o comando:

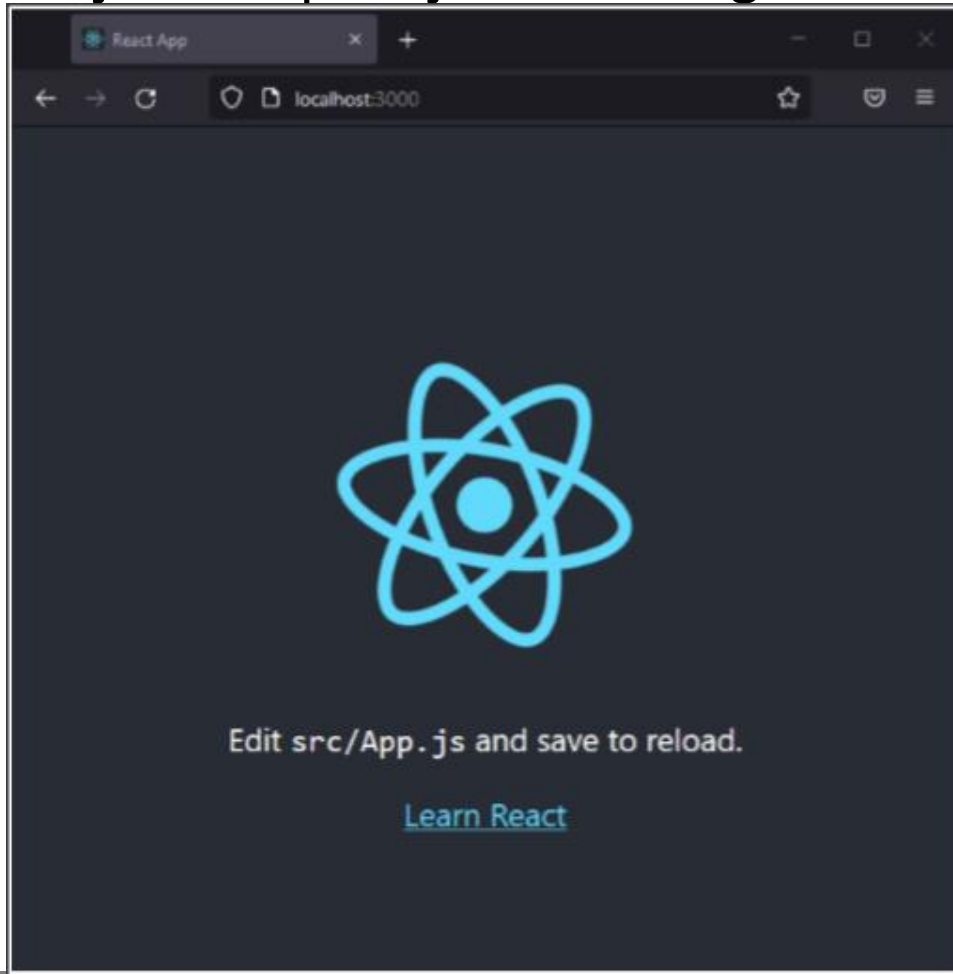
```
npx create-react-app tema_02_exemplo_01
```

Esse comando **cria um diretório** com o **nome da sua aplicação**, **instala os módulos** (pacote ou bibliotecas) necessárias para executar a aplicação React.

Importante: o comando **npm start** é usado sempre que quisermos **executar e visualizar a aplicação React no navegador**. Esse comando deve ser sempre executado dentro do diretório, onde a aplicação está instalada.

Instalando Aplicação React

Execução da aplicação no navegador:



Instalando Aplicação React



A página mostrada no navegador web é uma **aplicação pré-instalada** quando criamos nosso **projeto React** com o comando **npx create-react-app**. Observe que a aplicação React executa na **porta 3000** de um **servidor local** criado no seu computador com o comando **npm start**.

Caso você queria **terminar a execução** da aplicação React no navegador você deve aperta as **teclas Ctrl+C** no terminal da IDE e fechar o navegador.

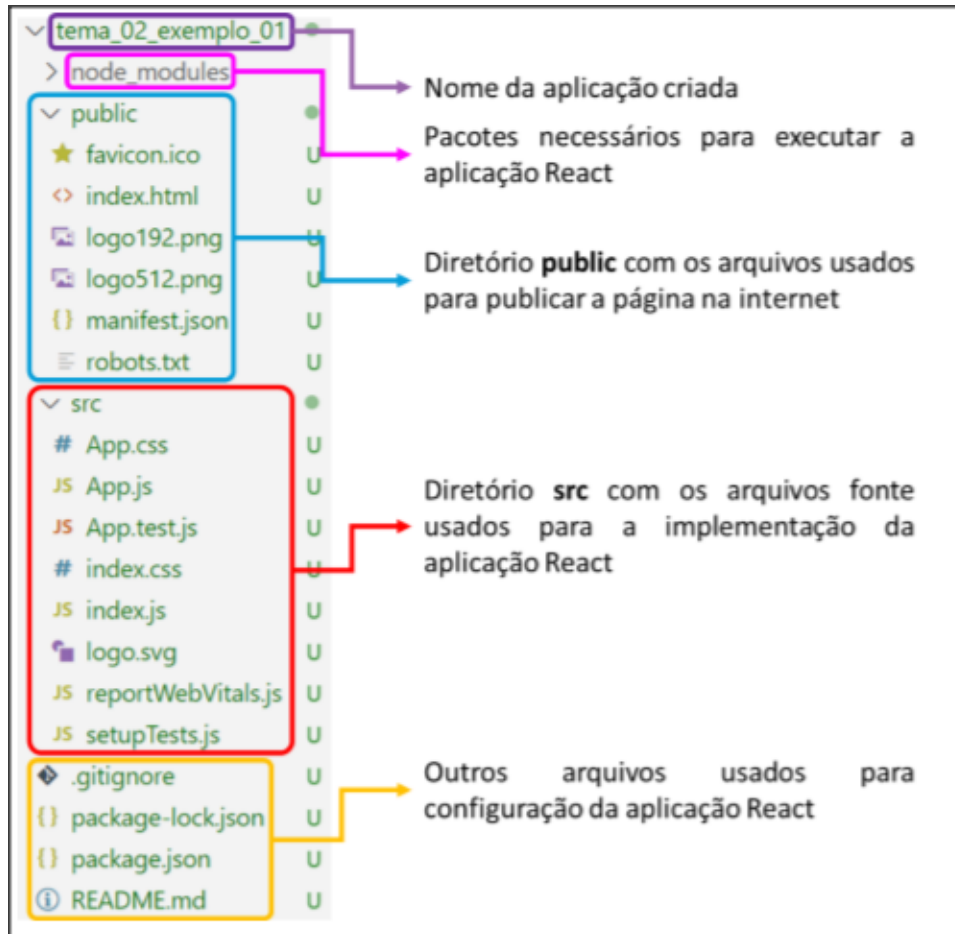
IOS – Instituto de
Oportunidade Social

Estrutura de arquivos do projeto



Estrutura de arquivos do projeto

Organização dos arquivos de uma aplicação em React:



Estrutura de arquivos do projeto

Instalação de Pacotes (**npm install**):



Importante! Se você tiver que instalar algum pacote extra (por exemplo: Bootstrap, react-router, redux, etc.) utilizando o gerenciador de pacotes **NPM**, ele será armazenado no diretório **node_modules**. Além disso, se você abrir o arquivo **.gitignore** verá que o diretório **node_modules** está na lista de ignorados para serem armazenados no repositório online. Isso faz sentido, porque é um diretório com biblioteca que devem estar instaladas no ambiente de desenvolvimento e não no repositório.

Continuando a análise da organização dos arquivos, temos o diretório **public**, que contém os arquivos usados para colocar a **aplicação em um ambiente de produção**. Quando a implementação estiver finalizada você pode executar o comando **npm build**. Ele gerará os códigos (HTML, CSS e JavaScript) e arquivos da aplicação, que devem ser colocados no **servidor de hospedagem do site**.

Estrutura de arquivos do projeto



Estrutura da aplicação React:

```
1. <!DOCTYPE html>
2. <html lang="pt-br">
3.   <head>
4.     <meta charset="utf-8" />
5.     <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
6.     <meta name="viewport" content="width=device-width, initial-scale=1" />
7.     <meta name="theme-color" content="#000000" />
8.     <meta
9.       name="description"
10.      content="Web site created using create-react-app"
11.    />
12.    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
13.    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
14.    <title>React App</title>
15.  </head>
16.  <body>
17.    <noscript>You need to enable JavaScript to run this app.</noscript>
18.    <div id="root"></div>
19.  </body>
20. </html>
```

Nessa estrutura você pode alterar o **título do site** na **linha 14**. Observe na linha 18 a instrução **<div id="root"></div>**, local onde toda a **aplicação React** será renderizada.

Estrutura de arquivos do projeto



Em seguida, temos o **diretório src**, que onde criaremos nossos arquivos colocaremos nossos códigos e faremos toda a implementação necessária dentro da aplicação React. Se você abrir o arquivo **index.js**, poderá visualizar o **método render()** usado para **renderizar o componente raiz App** no elemento **<div>** do arquivo **index.html** que contém a ID root.

```
1. import React from 'react';
2. import ReactDOM from 'react-dom';
3. import './index.css';
4. import App from './App';
5. import reportWebVitals from './reportWebVitals';
6.
7. ReactDOM.render(
8.   <React.StrictMode>
9.     <App />
10.   </React.StrictMode>,
11.   document.getElementById('root')
12. );
```


Demais arquivos:

- **index.css:** contém os **estilos CSS** que podem ser **acessados por todos os componentes** da nossa aplicação.
- **App.js:** **componente raiz** da aplicação React que é **renderizado na linha 9** do código do arquivo index.js.
- **App.css:** contém os **estilos CSS** que podem ser **acessados pelo componente App** e pelos seus componentes filhos.

Por fim, temos os **arquivos extras**:

- configuração do git (**.gitignore**). Se você não tiver um repositório local configurado no seu workspace ou o git configurado no VS Code esse arquivo não irá aparecer.
- **package.json**: contém a **configuração da aplicação React**, onde é possível visualizar as **dependências**, os **scripts** e outras informações do projeto. A seguir um exemplo desse arquivo:

Estrutura de arquivos do projeto

```
1. {
2.   "name": "tema_02_exemplo_01",
3.   "version": "0.1.0",
4.   "private": true,
5.   "dependencies": {
6.     "@testing-library/jest-dom": "^5.15.0",
7.     "@testing-library/react": "^11.2.7",
8.     "@testing-library/user-event": "^12.8.3",
9.     "react": "^17.0.2",
10.    "react-dom": "^17.0.2",
11.    "react-scripts": "4.0.3",
12.    "web-vitals": "^1.1.2"
13.  },
14.  "scripts": {
15.    "start": "react-scripts start",
16.    "build": "react-scripts build",
17.    "test": "react-scripts test",
18.    "eject": "react-scripts eject"
19.  },
20.  "eslintConfig": {
21.    "extends": [
22.      "react-app",
23.      "react-app/jest"
24.    ]
25.  },
26.  "browserslist": {
```

Estrutura de arquivos do projeto



```
27.   "production": [  
28.     ">0.2%",  
29.     "not dead",  
30.     "not op_mini all"  
31.   ],  
32.   "development": [  
33.     "last 1 chrome version",  
34.     "last 1 firefox version",  
35.     "last 1 safari version"  
36.   ]  
37. }  
38. }
```

Estrutura dos arquivos do projeto Não existe uma forma única de criar a estrutura de diretórios e arquivos no seu projeto React, o que existe é uma recomendação na documentação oficial do React:

<https://ptbr.reactjs.org/docs/faq-structure.html>

Pode-se utilizar um dos dois tipos de agrupamentos de arquivos: por **funcionalidades/rotas** ou por **tipos de arquivos**:

Estrutura de arquivos do projeto

Nos nossos projetos, usaremos o agrupamento por **tipo de arquivos** por ser mais comum e utilizado:

Agrupamento por funcionalidades/rotas	Agrupamento por tipo de arquivos
<code>common/</code> <code>Avatar.js</code> <code>Avatar.css</code> <code>APIUtils.js</code> <code>APIUtils.test.js</code>	<code>api/</code> <code>APIUtils.js</code> <code>APIUtils.test.js</code> <code>ProfileAPI.js</code> <code>UserAPI.js</code>
<code>feed/</code> <code>index.js</code> <code>Feed.js</code> <code>Feed.css</code> <code>FeedStory.js</code> <code>FeedStory.test.js</code> <code>FeedAPI.js</code>	<code>components/</code> <code>Avatar.js</code> <code>Avatar.css</code> <code>Feed.js</code> <code>Feed.css</code> <code>FeedStory.js</code> <code>FeedStory.test.js</code> <code>Profile.js</code> <code>ProfileHeader.js</code> <code>ProfileHeader.css</code>
<code>profile/</code> <code>index.js</code> <code>Profile.js</code> <code>ProfileHeader.js</code> <code>ProfileHeader.css</code> <code>ProfileAPI.j</code>	

IOS – Instituto de
Oportunidade Social

Vamos Praticar



Apostila de React

05.Apostila-React

Páginas 18 a 27

OBS: Acompanhar o passo a passo com o instrutor

IOS – Instituto de
Oportunidade Social

Exercícios



Realizar o processo de Build (**npm build**) do projeto praticado em aula e subir no GitHub, verificar se os diretórios de pacotes **node_modules** estão como exceção pra não subir (**.gitignore**) e se os arquivos buildados funcionam no site: <https://htmlpreview.github.io/> pois os demais exercícios das próximas aulas deverão estar nesse mesmo padrão.