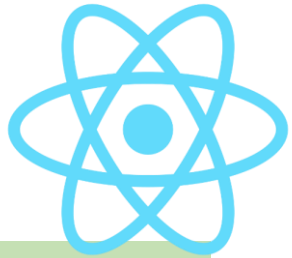
RECODE pro

Usando Create React App para projetos React



Vamos criar um projeto usando a ferramenta `create-react-app`

Se você possui o NPM e o Node.js instalados, é possível criar um aplicativo React instalando primeiro o aplicativo create-react-app

1º passo - Instale o create-react-app executando este comando no seu terminal:

```
npm install -g create-react-app
```

2º passo - Agora você está pronto para criar seu primeiro aplicativo React!
Execute este comando para criar um aplicativo React chamado app01

```
npx create-react-app app01
```

O create-react-app irá configurar tudo que você precisa para executar um aplicativo React.

3º - passo - Entre na pasta com comando : **cd app01**
Execute o comando **npm start** e o aplicativo vai ser exibido no navegador padrão do seu sistema

RECODE pro

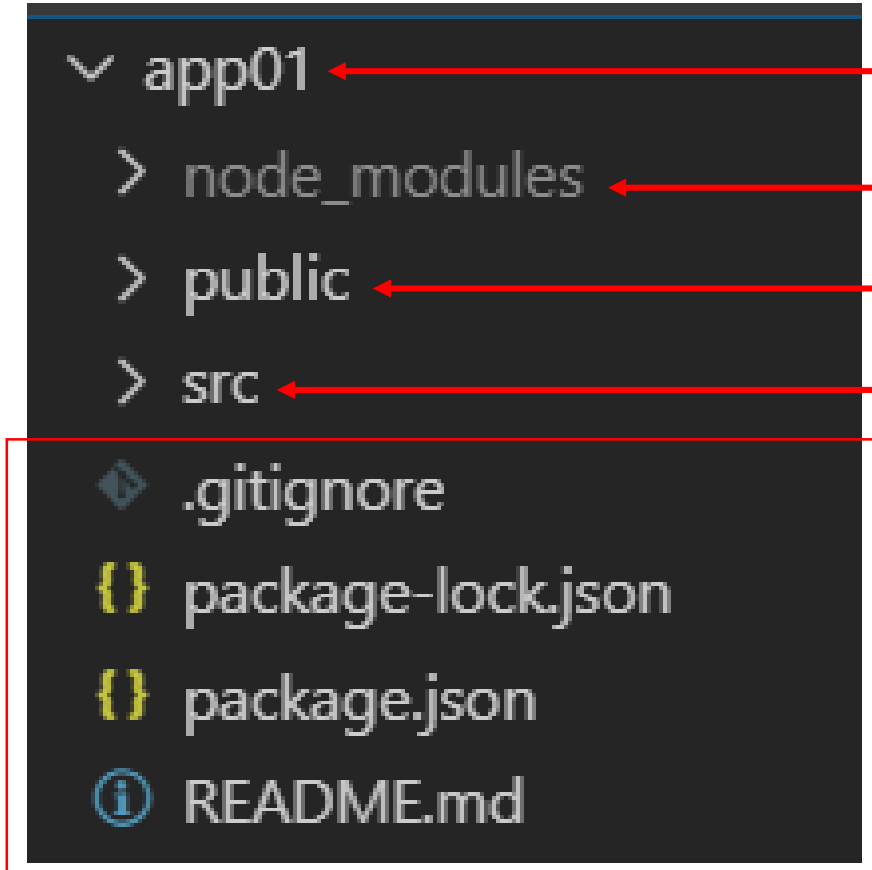
Usando Create React App para projetos React

Projeto padrão sendo executado no navegador

RECODE pro

Usando Create React App para projetos React

Estrutura básica de um projeto React



Pasta raiz da aplicação

Módulos(bibliotecas) do node (dependências para o funcionamento do projeto)

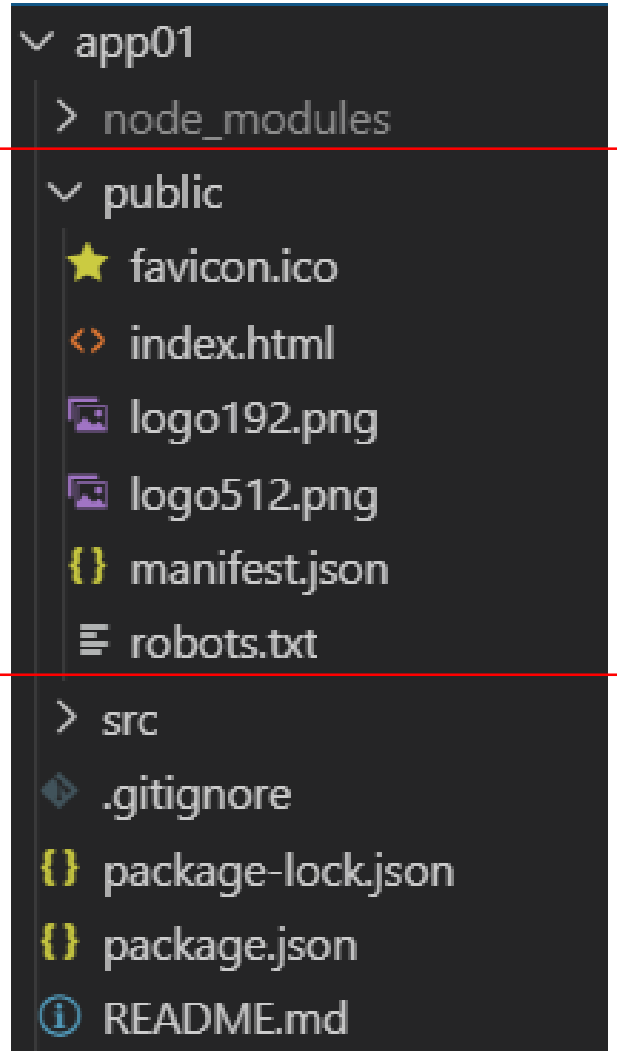
Na pasta public nós temos a index.html um ícone favicon e um manifest.json:

Nessa pasta estão os principais arquivos da nossa aplicação

Esses arquivos são de configurações do nosso projeto
e serão vistos em um outro momento

RECODE pro

Usando Create React App para projetos React

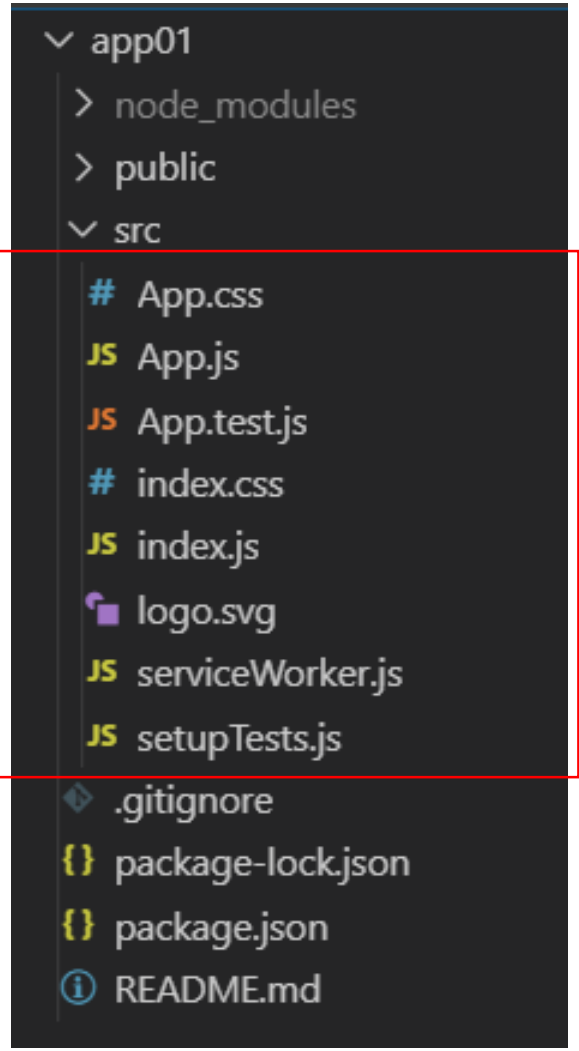


Na pasta public nós temos a index.html um ícone favicon e um manifest.json:
O favicon é o icone que aparece na aba do browser. O index é o html que carrega todo o projeto e o manifest.json é onde você seta os dados como nome do projeto cor de fundo e etc.

https://blog.fcamara.com.br/criando-sistemas-de-rotas-com-react/

RECODE pro

Usando Create React App para projetos React

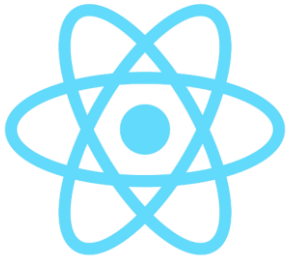


Na pasta src temos o app.css que carrega todo o css dos projetos. App.js que é a página inicial dos projetos. A index.js que é a página principal que irá criar todo o sistema de rotas do seu projeto e o logo que é carregado na página inicial.
A ServiceWorker.js que serve para carregar serviços em background como push notification e etc.

https://blog.fcamara.com.br/criando-sistemas-de-rotas-com-react/

RECODE pro

Usando Create React App para projetos React



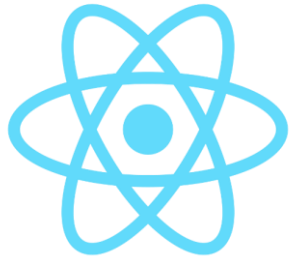
```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import * as serviceWorker from './serviceWorker';

ReactDOM.render(<App />, document.getElementById('root'));

// If you want your app to work offline and load faster, you can change
// unregister() to register() below. Note this comes with some pitfalls.
// Learn more about service workers: https://bit.ly/CRA-PWA
serviceWorker.unregister();
```

Na index.js estamos importando as principais funções para fazer tudo funcionar, porém ele está carregando apenas a página inicial do react

https://blog.fcamara.com.br/criando-sistemas-de-rotas-com-react/

RECODE pro

Usando Create React App para projetos React



```jsx
import React from 'react';
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Edit <code>src/App.js</code> and save to reload.
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          Learn React
        </a>
      </header>
    </div>
  );
}
export default App;
```

Na App.js e renderizado todo o conteúdo da página usando uma function component

https://blog.fcamara.com.br/criando-sistemas-de-rotas-com-react/

RECODE pro

Usando react-router-dom para criar navegação entre paginas

Vamos criar uma página chamada hello.js dentro da pasta src e deixar desta forma:

```
import React, { Component } from 'react';
import './App.css';

class hello extends Component{
    render(){
        return(

            <div className="App">
                <p>HELLO WORLD!!</p>
            </div>
        );
    }
}
export default hello;
```

https://blog.fcamara.com.br/criando-sistemas-de-rotas-com-react/

RECODE pro

Usando react-router-dom para criar navegação entre paginas

Vamos instalar o react-router-dom para poder criar sistema de link e de browserRouter do React, é só ir ao terminal e digitar o comando:

npm install react-router-dom

```
import React from 'react';
import logo from './logo.svg';
import './App.css';
import { Link } from 'react-router-dom';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Edit <code>src/App.js</code> and save to reload.
        </p>
        <Link to="/hello">
          <button>
            Proxima página
          </button>
        </Link>
      </header>
    </div>
  );
}
export default App;
```

Na página app.js importe a função Link, logo após vamos criar um botão com a função Link to =”/hello” ficando conforme a imagem

https://blog.fcamara.com.br/criando-sistemas-de-rotas-com-react/ Flávio Mota

RECODE pro

Usando react-router-dom para criar navegação entre paginas

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import hello from './hello';
import { BrowserRouter, Switch, Route } from 'react-router-dom';
import * as serviceWorker from './serviceWorker';

ReactDOM.render(
    <BrowserRouter>
        <Switch>
            <Route exact path='/' component={App} />
            <Route exact path='/hello' component={hello} />
        </Switch>
    </BrowserRouter> , document.getElementById('root'));
// If you want your app to work offline and load faster, you can change
// unregister() to register() below. Note this comes with some pitfalls.
// Learn more about service workers: https://bit.ly/CRA-PWA
serviceWorker.unregister();
```

Na página index.js criarmos a rota que manda você para a página hello. E vamos criar o sistema de rotas, primeiro importaremos o browserRouter, route e switch do react-router-dom, depois modificamos nosso código para receber as rotas. Importamos as páginas exportadas anteriormente de cada diretório e depois montamos as rotas e damos um caminho ou URL para cada uma ficando desta forma:

https://blog.fcamara.com.br/criando-sistemas-de-rotas-com-react/

RECODE pro

Usando react-router-dom para criar navegação entre paginas

Agora nossa página inicial deve conter um botão “próxima página”. Ficando da seguinte forma:

Ao clicar no botão será direcionado para a nova página hello

http://localhost:3000/hello

HELLO WORLD!!

Parabéns! Agora você está pronto para criar todo um sistema de navegação por rotas

https://blog.fcamara.com.br/criando-sistemas-de-rotas-com-react/

Flávio Mota