**Equipe 4 (React)**

|  |
| --- |
| Rafael Sacramento |
| Guilherme Cunha |
| Caio Carvalho |
| Pedro Costa |
| Marivaldo Júnior |

**Preparando o ambiente**

**Para** **criar uma aplicação react** é preciso possuir o **Node.js** na máquina

Baixar aqui: <https://nodejs.org/en/>

O Node.js é um executor/interpretador de código JavaScript, podendo ser facilmente executado via terminal. Se assegure que todas as informações da print estão selecionadas antes de clicar em Next.

Para ter certeza que a instalação foi feita corretamente, no terminal execute:

**node -v** (deve retornar a versão) e depois **npm -v** (retorna a versão do npm).

Recomendo fortemente a instalação do **YARN** que é um gerenciador de dependências, semelhante ao NPM porém mais rápido, a instalação é super simples. No terminal escreva: **npm i -g yarn**. Posteriormente rode o comando **yarn** para certificar-se que é válido.

**Ide recomendada:** vsCode

O VSCode possui diversas ferramentas para nos ajudar durante o desenvolvimento e dentre ela temos as extensões, que torna nosso Vscode ainda mais completo, dessas extensões recomendo o uso de:

1. Git Lens

2. ES7 React/Redux/Graphql snippets

3. ESLINT *(Se for usar o Eslint)*

4. Prettier

5. Prettier Eslint *(Se for usar o Eslint)*

6. Beautify

7. vscode-icons

**Para criar um projeto utilize:**

**yarn create react-app*****Nome\_Projeto*** (mais rápido)

ou

**npm init react-app** ***Nome\_Projeto***

Obs: O objeto será criado no local indicado pelo terminal.

**(Opcional)** Para auxiliar no aprendizado com boas práticas, recomendo a instalação do **ESLint**. Você pode desabilitar as regras que estiverem te atrapalhando.

**npm i -g eslint** (instala o ESlint global na máquina)

**eslint -v** (retorna a versão, para verficar se foi instalado corretamente)

Agora dentro do projeto (Recomendo usar o terminal do próprio vsCode com o projeto aberto nele) deve configurar o esLint pro projeto. É preciso configurar sempre que iniciar um projeto novo.

**eslint --init**

deve preencher o formulário de acordo com o que deseja pro seu projeto: Para esse vamos usar:

*1- To check syntax, find problems, and enforce code style*

*2-Javascript modules (import/export)*

*3-React*

*4-No*

*5-Browser*

*6-use a popular style guide*

*7-Airbnb*

*8-Javascript*

*9-Yes*

**Ct P ->Preferences:** Open settings JSON / Abrir configurações JSON

**adicionar esse regra:**

"editor.codeActionsOnSave": {

"source.fixAll.eslint":true

},

**copiar e colar no eslintrc.js (substituir o atual):**

**module.exports = {**

**env: {**

**browser: true,**

**es6: true,**

**},**

**extends: [**

**'plugin:react/recommended',**

**'airbnb',**

**],**

**globals: {**

**Atomics: 'readonly',**

**SharedArrayBuffer: 'readonly',**

**},**

**parserOptions: {**

**ecmaFeatures: {**

**jsx: true,**

**},**

**ecmaVersion: 2018,**

**sourceType: 'module',**

**},**

**plugins: [**

**'react',**

**],**

**rules: {**

**'react/jsx-filename-extension': [**

**'error',**

**{**

**extensions: ['.js', '.jsx'],**

**},**

**],**

**'react/prop-types': 0,**

**},**

**};**

**Conteúdo**

**JavaScript** (props, desestruturação, callbacks…)

**Principais Hooks do react** (useState, useContext e useEffect). Os Hooks permitem um código mais limpo e funcional, uma substitui a necessidade de criar classes para gerenciar estados.

**react-router-dom** (para utilização de rotas)

**react-bootstrap** (para agilizar a customização)

**axios** (para interagir com o bd)

**json server** (api fake para simular interações com o bd)

**react datepicker, date-fns, react-toastify, react-icons** (algumas dependências legais)

**Dependências**

yarn add react-router-dom

yarn add react-bootstrap bootstrap

yarn add axios

npm install - json-server

npm install react-icons --save

yarn add react-datepicker

yarn add date-fns

npm i react-toastify

import 'react-datepicker/dist/react-datepicker.css';

import 'bootstrap/dist/css/bootstrap.min.css';

import 'react-toastify/dist/ReactToastify.css';

**Iniciando a Aplicação:**

Abra um terminal no projeto e execute: json-server --watch db.json --port 3004

Se tudo ocorrer bem, abra outro terminal e inicia a aplicação com **Yarn Start** ou o equivalente com npm.

Se não ocorrer bem é possível que o json server não esteja presente, então utilize **npm install – json-server** para instalar ele globalmente e repita o processo anterior.