PASSO A PASSO PARA PREPARAR O AMBIENTE

1 - ETAPA - CRIAR DIRETÓRIOS DE TRABALHO

Abrir um terminal do seu sistema operacional e executar:

mkdir reactjs reactjs/aulas
cd reactjs/aulas

2 - ETAPA - CRIAR DIRETÓRIO PRINCIPAL

Ainda no terminal crie:

mkdir parteA
cd parteA

Abra o VS CODE com o comando:
code .

3 - ETAPA - INICIALIZAR O package.json

Dentro do VS CODE abra o terminal e inicialize o arquivo com o comando:

npm init -y

O arquivo criado após a execução do comando, é responsável por conter as informações principais do projeto. Exemplo, name, version, license, dependecies e etc.

4 - ETAPA - INSTALANDO DEPENDÊNCIAS

Para instalar o **React**, execute o comando:

npm add react

Após a execução do comando será criada uma pasta **node_modules** contendo todos os scripts necessários para o funcionamento do REACT. Note que, sempre ao incluir uma nova dependência, esta aparecerá no arquivo package.json
Para instalar o React-dom, responsável por possibilitar que o React trabalhe com elementos da árvore DOM.

npm add react-dom

5 - ETAPA - ESTRUTURAR OS DIRETÓRIOS

Dentro do diretório parteA:

Criar **src**: Código principal da Aplicação

Criar **public**: Arquivos públicos para acesso externo

6 - ETAPA - CRIAR ARQUIVOS

Criar **public/index.html**

Em seguida, crie no arquivo o código html padrão.

Criar **src/index.jsx**

Crie o seguinte código dentro de **src/index.jsx**:

```
import React from 'react';

function App() {
    return <h1> Hello World </h1>
}
```

7 - ETAPA - INSTALAÇÃO DO BABEL

O babel é responsável em geral pela conversão do código React para que navegadores mais modernos entendam a codificação.
Obs.: o **-D** indica que o babel será instalado em dependências de desenvolvimento.

```
npm add @babel/core @babel/cli @babel/preset-env -D
```

Como dinâmica de estudo pesquise o que cada uma dessas dependências faz.

8 - ETAPA - CONFIGURAR BABEL

Criar babel.config.js

No arquivo criado, escreva:

```
module.exports = {
    presets:[
        '@babel/preset-env',
        '@babel/preset-typescript',
        ['@babel/preset-react', {
            runtime: 'automatic'
        }]
    ]
}
```

9 - ETAPA - INSTALAR E CONFIGURAR O WEBPACK

```
npm add webpack webpack-cli webpack-dev-server -D
npm add babel-loader -D
```

Como dinâmica de estudo pesquise o que cada uma dessas dependências faz.

Criar webpack.config.js

No arquivo criado, escreva:

```
const path = require('path')

module.exports = {
   entry: path.resolve(_dirname, 'src', index.js),
   output: {
     path.resolve(_dirname, 'dist'),
     filename: 'bundle.js'
   },
   resolve: {
     rules: [
        {
            test: /\.jsx$/,
            exclude: /node_modules/,
            use: 'babel-loader',
        }
     ]
   }
}
```

10 - ETAPA - PREPARAÇÃO FINAL

No arquivo **public/index.html** crie a seguinte estrutura dentro da body:

```
<div id="root"></div>
```

Será dentro dessa estrutura que toda nossa aplicação será criada.

Na pasta **src/index.jsx,** altere o código para:

```
import {render} from 'react-dom';
import {App} from './App'

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

Crie o seguinte arquivo **src/App.jsx,** e dentro dele digite:

```
export function App() {
   return <h1> Hello World </h1>
}
```

REFERÊNCIAS:

Creating a React App… From Scratch.