# PRIMEIROS PASSOS

1. No console do VS Code, digitar: “npx create-react-app my-app”
   1. Embora a instalação crie um arquivo “index.js”, não é nele que desenvolvemos nossa aplicação; mas sim no arquivo “app.js”, que também foi criado.
2. Depois de instalado, passamos o comando “cd my-app”.
3. O próximo comando é “npm start”.

# LIMPANDO O AMBIENTE

1. A ferramenta instala algumas coisas de que talvez não precisemos.
2. Em “**app.js**”, podemos apagar todo o código do **return**, pois não será aquela a nossa aplicação.

No mesmo arquivo, podemos apagar as aplicações que não estivermos usando. Ex:  
 import logo from './logo.svg';

import './App.css';

1. Podemos apagar os arquivos (se não formos usar):
   1. “App.css”
   2. “App.test.js”
   3. “logo.svg”
   4. “serviceworker.js”
   5. “setupTest.js”
2. Porém, eu apaguei e deu merda.

# DESENVOLVIMENTO

1. O código é escrito de maneira muito semelhante ao HTML, inclusive usando as mesmas tags.
2. Para imprimir no navegador, todas as tags devem estar dentro de uma tag-pai. Tags-irmãs não podem ser impressas sozinhas.
   1. Ou seja, hierarquicamente é obrigatório ter uma tag que encapsule todo o código. Pode ser uma simples div.
3. Para não ficar tudo bagunçado, criamos arquivos “.jsx” nos quais estabelecemos uma classe com algum elemento da página. Depois, importamos essas classes no “App.js”.
   1. E importamos essa classe no código entre <>.
   2. No arquivo **.jsx**, é importante importar o **React** e o {**Component**}.
   3. Também precisamos estender a classe criada e estender o **Component**
   4. Para fins de organização, organizamos os componentes numa pasta de mesmo nome.
4. Exemplo de arquivo **.jsx**:  
   import React, { Component } from 'react';

export class FormularioCadastro extends Component {

render(){return(

<form>

<input type="text" placeholder="Título" />

<textarea placeholder="Escreva sua nota..."></textarea>

<button>Criar Nota</button>

</form >

);

}

}

1. Podemos criar um arquivo .css normal e importar no documento .jsx.
   1. Ex: import "./nota.css"
   2. Como "class" é uma palavra reservada do JS, usamos "**className**" para desempenhar o papel de rotular o elemento com uma classe.
2. Lembrar de colocar o “bind()” para evitar erros.
   1. Ex:

|  |
| --- |
| constructor() {  super();  this.titulo = "";  }    handleMudancaTitulo(evento) {  this.titulo = evento.target.value;  }  render() {  return (  <form className="form-cadastro">  <input  type="text"  placeholder="Título"  className="form-cadastro\_input"  onChange={this.handleMudancaTitulo.**bind(this)**}  />  <textarea  rows={15}  placeholder="Escreva sua nota..."  className="form-cadastro\_input"  />  <button className="form-cadastro\_input form-cadastro\_submit">  Criar Nota  </button>  </form>  );  }  } |

# IMPORTANDO SVG

1. Pra ficar mais fácil de usar repetidamente no código, podemos importar como um componente no início do documento
   1. Ex:  
      import deleteSVG from "../../assets/icones/icone-delete.svg";
   2. E depois usar dentro de uma tag img:  
      <img src={deleteSVG}></img>
2. Também podemos importar como um ReactComponent.
   1. Ex:  
      import {ReactComponente as DeleteSVG} from "../../assets/icones/icone-delete.svg";
   2. E importar o componente diretamente no código:  
      <DeleteSVG/>
      1. O ReactComponent precisa começar com letra maiúscula.