

AULA 01 - REACT

- Criando um projeto

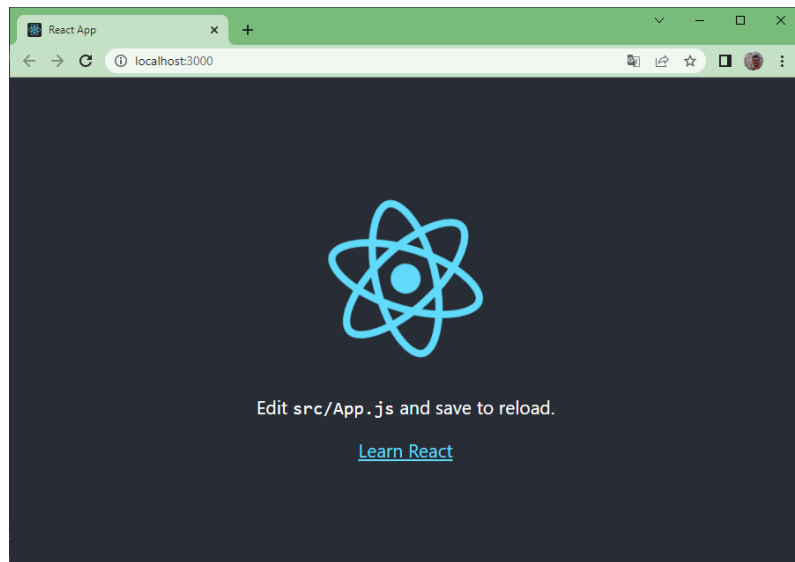
```
npx create-react-app seuprojeto
```

- Ir para a pasta do projeto

```
cd seuprojeto
```

- Para Iniciar a aplicação

```
npm start
```

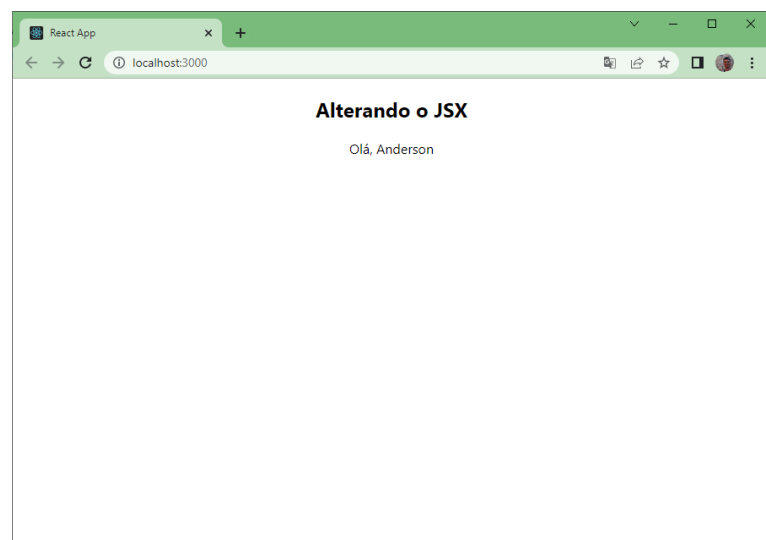


- Para encerrar a aplicação

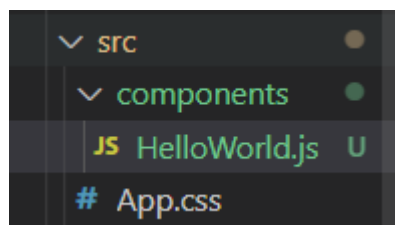
```
ctrl+c
```

- Alterando o arquivo **App.js**

```
JS App.js M X
exemplo > src > JS App.js > default
1  import './App.css';
2  const nome = 'Anderson'
3
4  function App() {
5    return (
6      <div className="App">
7        <h2>Alterando o JSX</h2>
8        <p>Olá, {nome}</p>
9      </div>
10   );
11 }
12
13 export default App;
```



- Criamos uma pasta para os componentes



```

exemplo > src > components > JS HelloWorld.js > ...
1  function HelloWorld(){
2      return(
3          <div>
4              <h1>Meu Primeiro Componente</h1>
5          </div>
6      )
7  }
8
9  export default HelloWorld

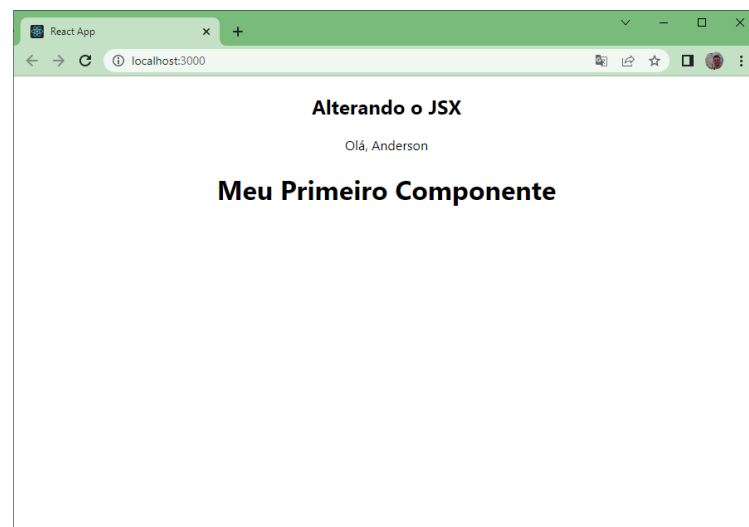
```

- Importando um componente

```

exemplo > src > JS App.js > default
1  import './App.css';
2  import HelloWorld from './components/HelloWorld';
3  const nome = 'Anderson'
4
5  function App() {
6      return (
7          <div className="App">
8              <h2>Alterando o JSX</h2>
9              <p>Olá, {nome}</p>
10             <HelloWorld />
11          </div>
12      );
13  }
14
15  export default App;

```



- Criando outro componente

```

exemplo > src > components > JS Frase.js > Frase
1 function Frase(){
2   return(
3     <div>
4       <p>Este é um componente com uma frase</p>
5     </div>
6   )
7 }
8 export default Frase

```

- Reutilizando um componente dentro de outro componente

```

exemplo > src > components > JS HelloWorld.js > default
1 import Frase from "./Frase"
2
3 function HelloWorld(){
4   return(
5     <div>
6       <h1>Meu Primeiro Componente</h1>
7       <Frase />
8     </div>
9   )
10 }
11
12 export default HelloWorld

```



Arquivos desta aula: https://github.com/ProfAndersonVanin/curso_react

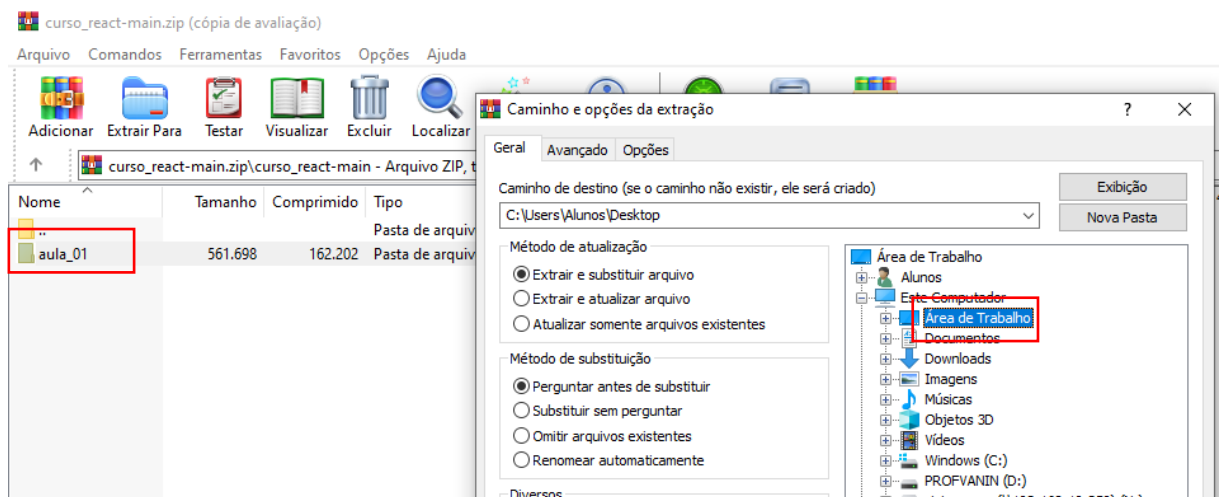
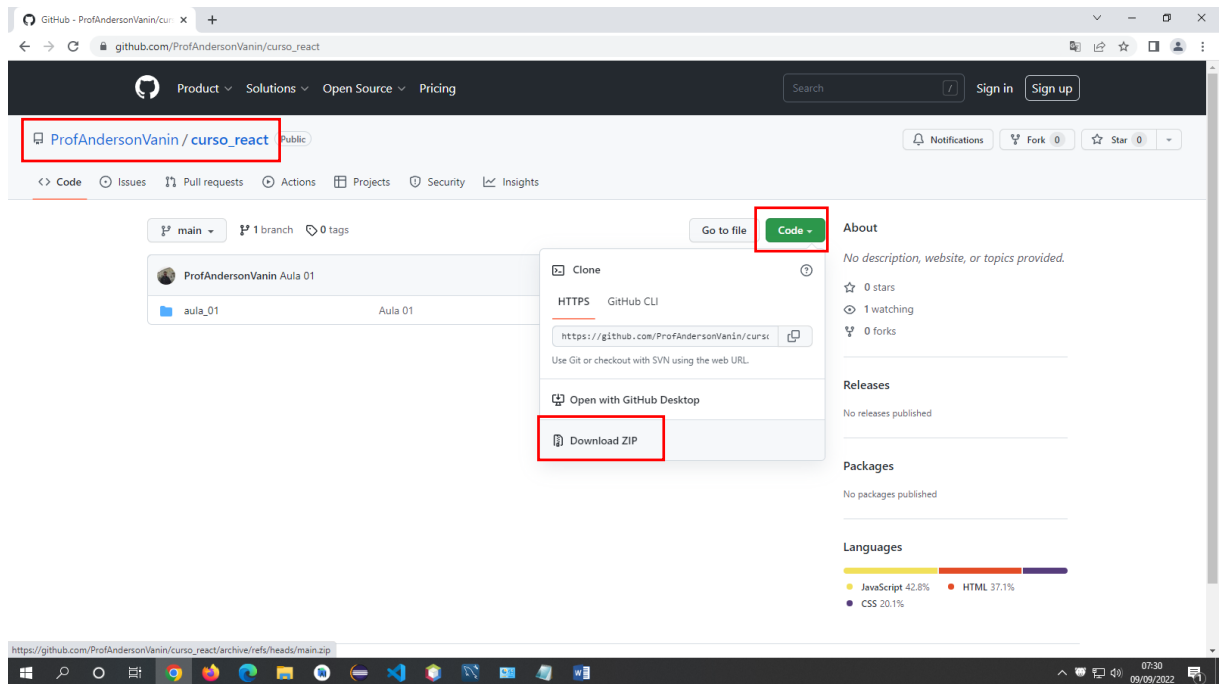
Pasta : aula_01

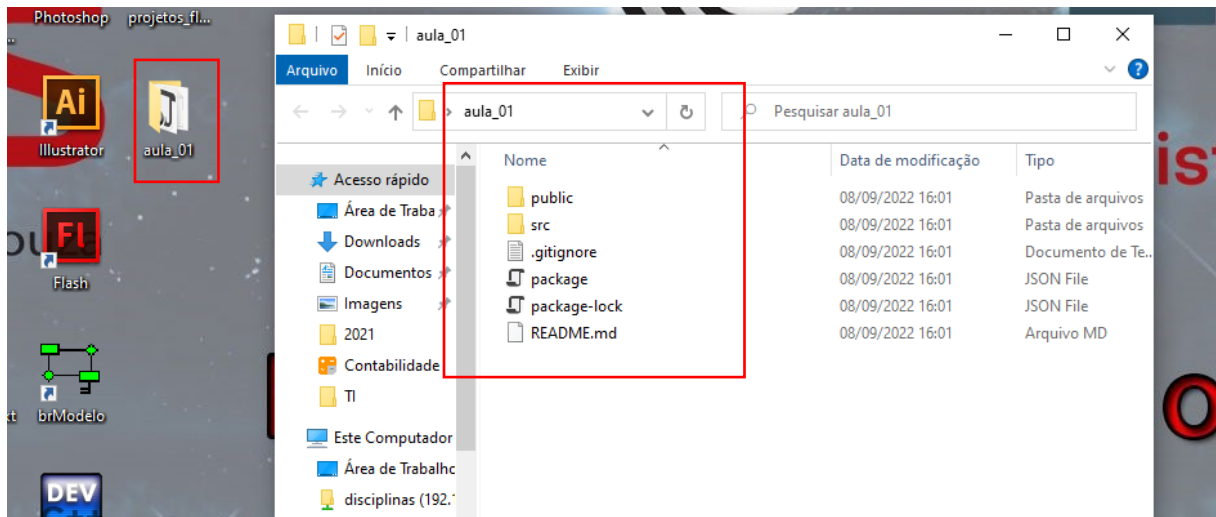
AULA 02 - REACT

- Baixando os arquivos do repositório:

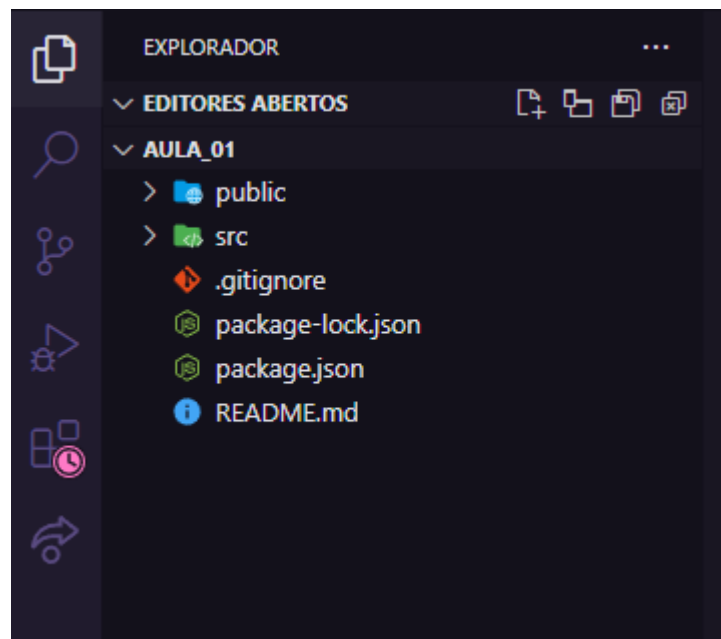
https://github.com/ProfAndersonVanin/curso_react

Você pode fazer o download de todo o repositório e extrair para um local específico os arquivos que estão dentro da pasta **aula_01**.





Agora abra a pasta no VSCODE



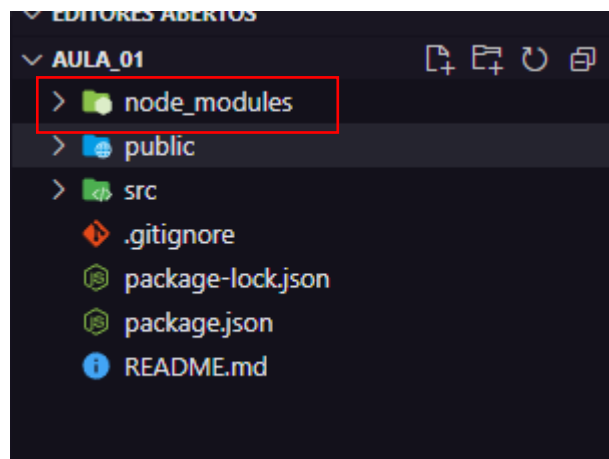
Para o react funcionar, é necessário que ele tenha todas as dependências corretamente instaladas. Abra um terminal e digite o comando:

```
npm install
```

```
PROBLEMAS  SAÍDA  CONSOLE DE DEPURAÇÃO  TERMINAL

Microsoft Windows [versão 10.0.19043.1165]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\Alunos\Desktop\aula_01>npm install
[██████████.....] - extract:color-name: sill extract ansi-styles
```



Após a instalação das dependências, teste a aplicação digitando o comando:

`npm start`



- Trabalhando com props

Vídeo #05

Volte ao arquivo App.js e deixe-o da seguinte forma:

```
src > JS App.js > [default]
1  import './App.css';
2  import HelloWorld from './components/HelloWorld';
3
4  function App() {
5    return (
6      <div className="App">
7        <HelloWorld />
8      </div>
9    );
10 }
11
12 export default App;
```

Crie um novo componente chamado DigaMeuNome.js

```
src > components > JS DigaMeuNome.js > [default]
1  function DigaMeuNome(){
2    return(
3
4    ~ ~
5  )
6  }
7  export default DigaMeuNome
```

Agora vamos trabalhar com as props


```
src > components > JS DigaMeuNome.js > DigaMeuNome
1  function DigaMeuNome(props){
2      return(
3          <div>
4              <p>Fala aí {props.nome}, suave na nave?</p>
5          </div>
6      )
7  }
8
9  export default DigaMeuNome
```

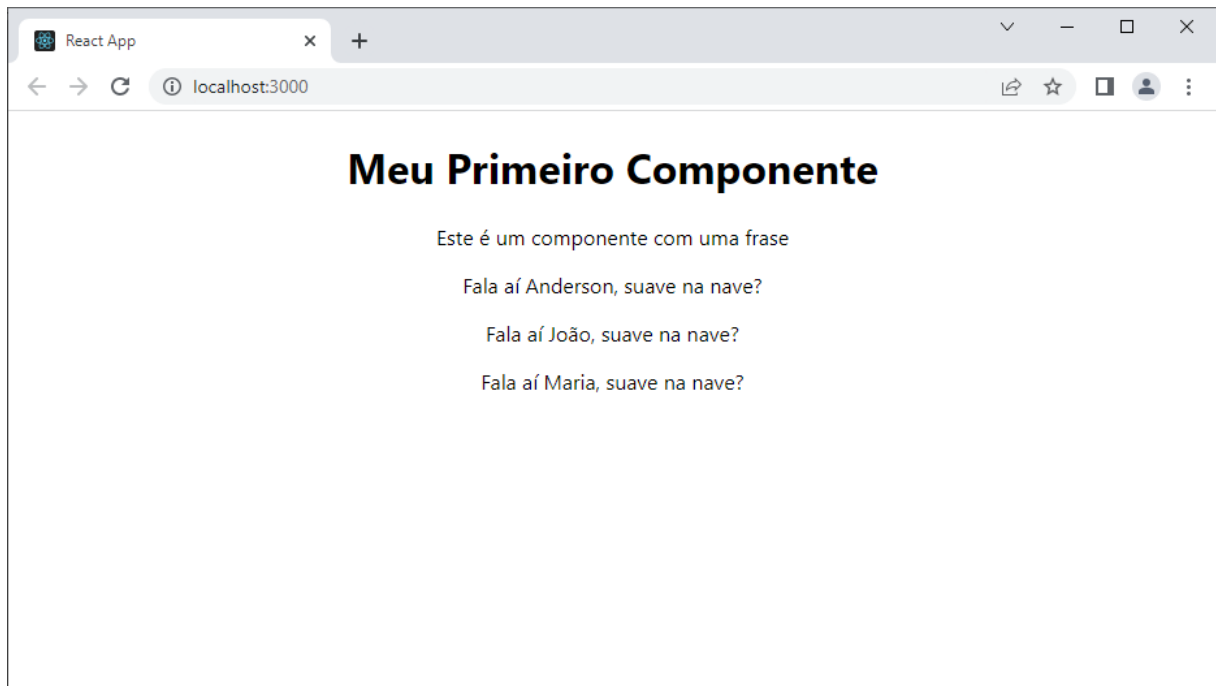
Agora vamos alterar o arquivo App.js, passando o valor de nome através de props

```
src > JS App.js > default
1  import './App.css';
2  import HelloWorld from './components/HelloWorld';
3  import DigaMeuNome from './components/DigaMeuNome';
4
5  function App() {
6      return (
7          <div className="App">
8              <HelloWorld />
9              <DigaMeuNome nome="Anderson" />
10         </div>
11     );
12 }
13
14 export default App;
```



Fazendo o reaproveitamento

```
src > JS App.js > App
1  import './App.css';
2  import HelloWorld from './components/HelloWorld';
3  import DigaMeuNome from './components/DigaMeuNome';
4
5  function App() {
6    return (
7      <div className="App">
8        <HelloWorld />
9        <DigaMeuNome nome="Anderson" />
10       <DigaMeuNome nome="João" />
11       <DigaMeuNome nome="Maria" />
12     </div>
13   );
14 }
15
16 export default App;
```

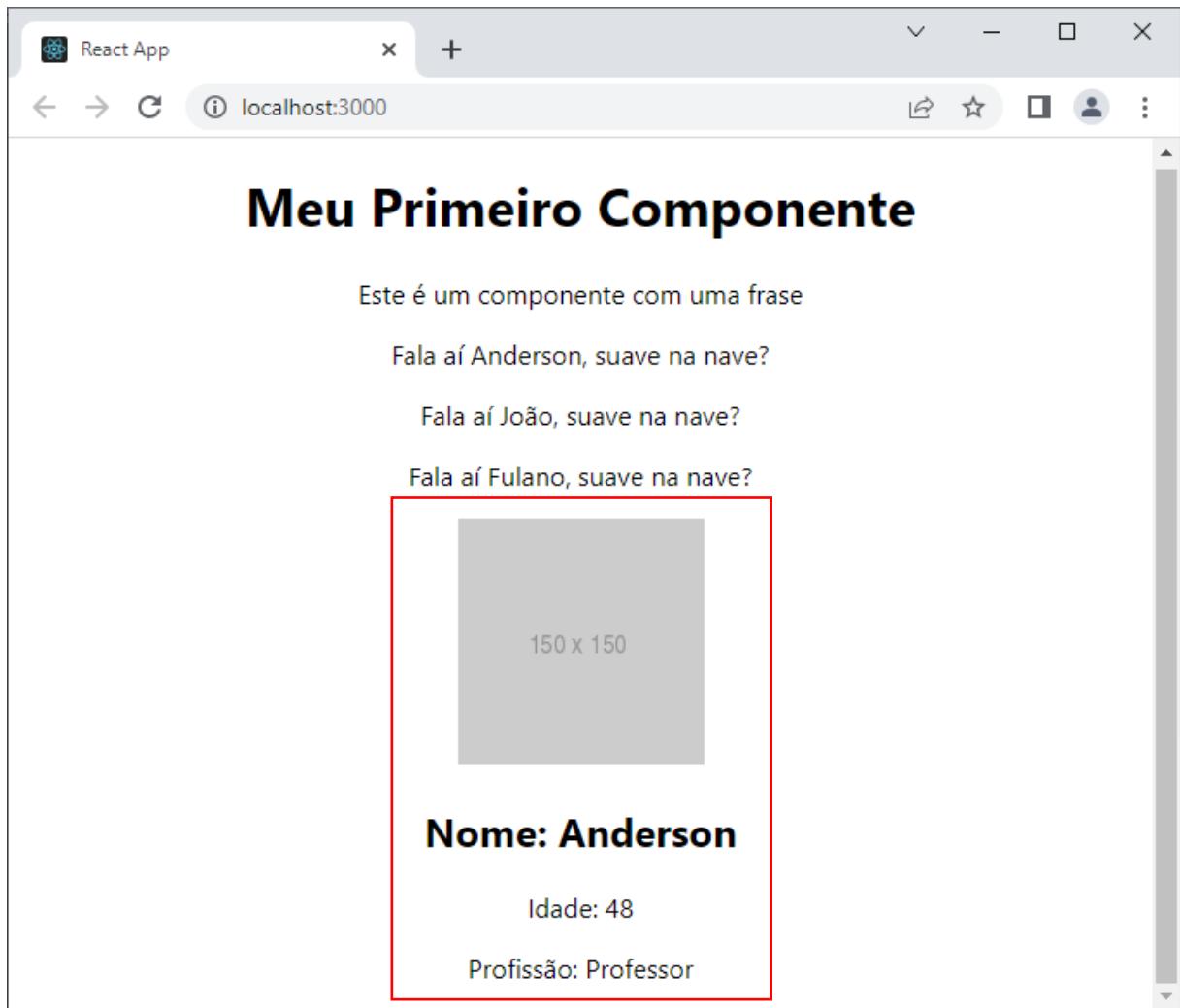


Vamos ver um exemplo de uso. Crie um componente chamado Pessoa.js

```
src > components > JS Pessoa.js > [🔗] default
1  function Pessoa(props){
2      return (
3          <div>
4              <img src={props.foto} alt={props.nome} />
5              <h2>Nome: {props.nome}</h2>
6              <p>Idade: {props.idade}</p>
7              <p>Profissão: {props.profissao}</p>
8          </div>
9      )
10 }
11
12 export default Pessoa
```

Agora altere o arquivo App.js

```
src > JS App.js > App
1 import './App.css';
2 import HelloWorld from './components/HelloWorld';
3 import DigaMeuNome from './components/DigaMeuNome';
4 import Pessoa from './components/Pessoa';
5
6 function App() {
7   const nome = "Fulano"
8   return (
9     <div className="App">
10       <HelloWorld />
11       <DigaMeuNome nome="Anderson" />
12       <DigaMeuNome nome="João" />
13       <DigaMeuNome nome={nome} />
14       <Pessoa
15         nome="Anderson"
16         idade="48"
17         profissao="Professor"
18         foto="https://via.placeholder.com/150"
19       />
20     </div>
21   );
22 }
23
24 export default App;
```



Uma outra maneira de passar vários valores pela propriedade props.

```
src > components > JS Pessoa.js > [e] default
1  function Pessoa({nome, idade, profissao, foto}){
2      return (
3          <div>
4              <img src={foto} alt={nome} />
5              <h2>Nome: {nome}</h2>
6              <p>Idade: {idade}</p>
7              <p>Profissão: {profissao}</p>
8          </div>
9      )
10 }
11
12 export default Pessoa
```

Pode-se omitir a propriedade props, visto já foi informado na função os parâmetros.

- Alterando CSS

Dentro da pasta components crie um arquivo chamado Frase.module.css

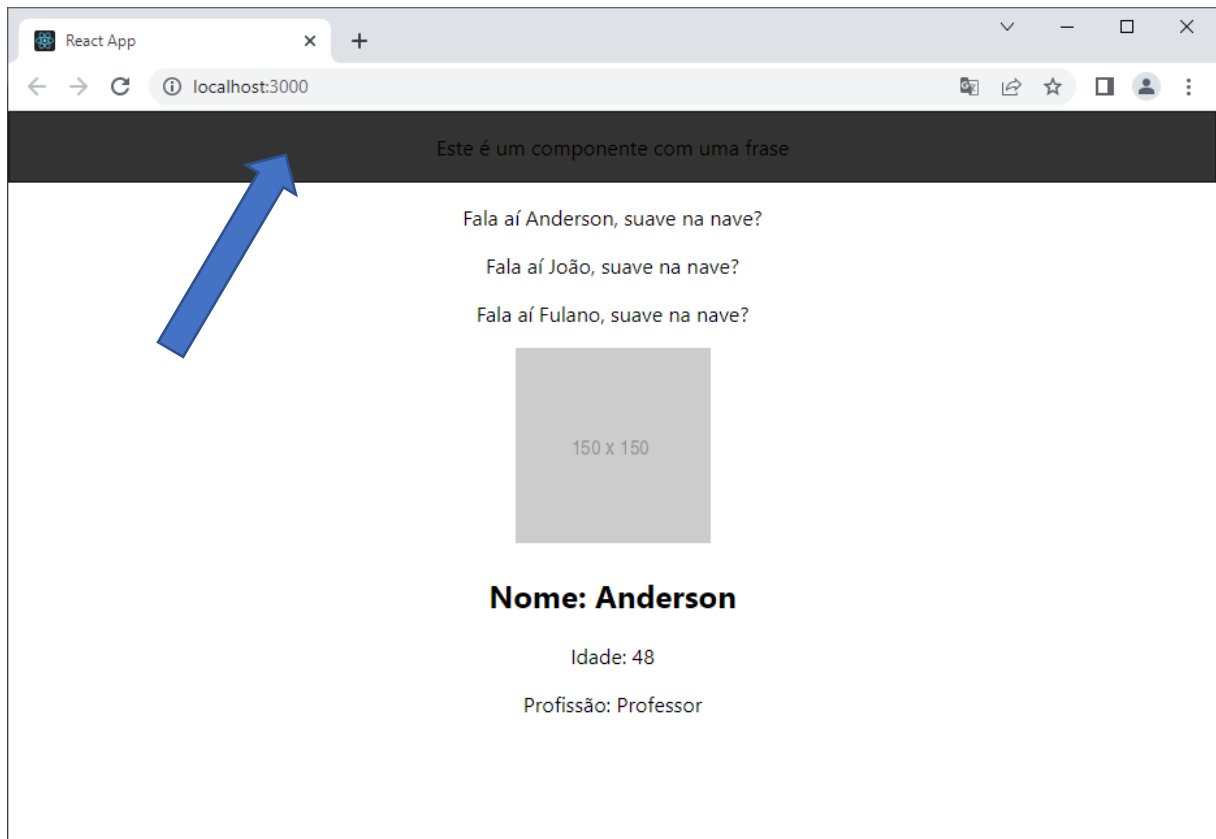
```
src > components > Frase.module.css > .fraseContent
1  /* No React não é possível criar classes em CSS que tenham como separador
2  do nome um traço - . por exemplo: .classe-qualquer
3  Para juntar dois nomes, você pode usar o underline ou CamelCase
4  Por exemplo: .classeQualquer ou .classe_qualquer
5  */
6
7  .fraseContainer{
8    background-color: #333;
9    border: 1px solid #111;
10 }
11
12 .fraseContent{
13   color: #fff;
14   background-color: #333;
15   margin: 0;
16 }
```

Agora vamos aplicar estes estilos ao componente Frase.js

```
src > components > Frase.js > default
1  import styles from './Frase.module.css'
2  // styles vai funcionar da mesma forma que a propriedade props
3
4  function Frase(){
5    return(
6      <div className={styles.fraseContainer}>
7        <p>Este é um componente com uma frase</p>
8      </div>
9    )
10 }
11 export default Frase
```

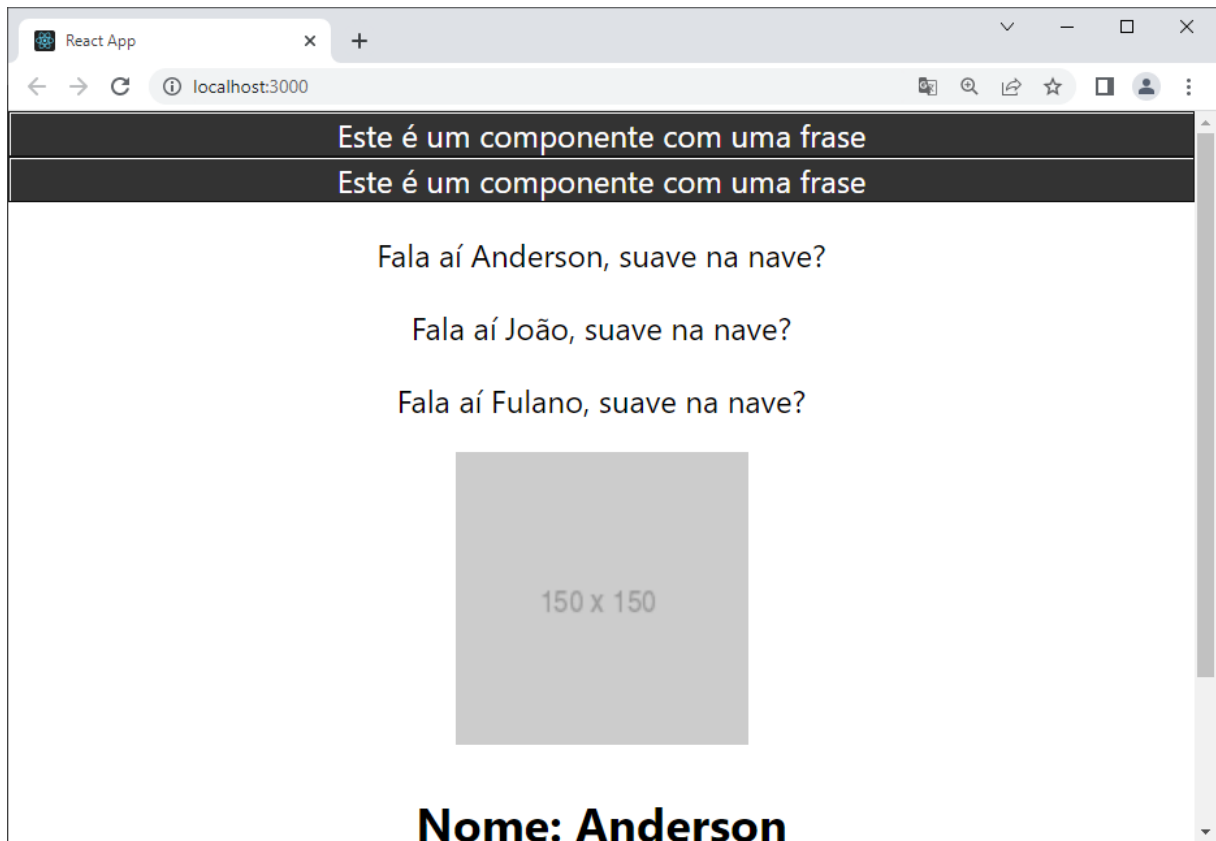
NOTA: Não se esqueça de importar o componente Frase.js no App.js para que ele apareça.

```
1  import './App.css';
2
3  import DigaMeuNome from './components/DigaMeuNome';
4  import Pessoa from './components/Pessoa';
5  import Frase from './components/Frase';
6
7  function App() {
8    const nome = "Fulano"
9    return (
10      <div className="App">
11        <Frase />
12        <DigaMeuNome nome="Anderson"/>
13        <DigaMeuNome nome="João"/>
14        <DigaMeuNome nome={nome}/>
15        <Pessoa
16          nome="Anderson"
17          idade="48"
18          profissao="Professor"
19          foto="https://via.placeholder.com/150"
20        />
21      </div>
22    );
23  }
24
25  export default App;
```

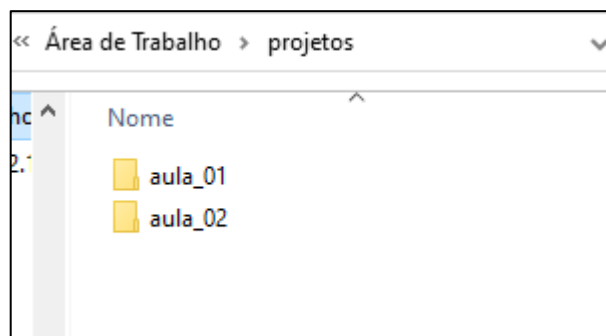


Continue aplicando os estilos criados em Frase.js

```
src > components > JS Frase.js > default
1  import styles from './Frase.module.css'
2  // styles vai funcionar da mesma forma que a propriedade props
3
4  function Frase(){
5    return(
6      <div className={styles.fraseContainer}>
7        <p className={styles.fraseContent}>Este é um componente com uma frase</p>
8      </div>
9    )
10 }
11 export default Frase
```

- Suba seu projeto para o Github. Abra o diretório contendo os arquivos das aulas



Abra a pasta do projeto e clique com o botão direito do mouse selecionando:

Git bash here

Digite os comandos (um de cada vez!)

git config --global user.name "Seu nome"

git config --global user.email seu-email@example.com

git init

git status

git add .

git commit -m "Aula_02"

Digite no bash git o comando:

git remote add origin https://github.com/ProfAndersonVanin/curso_react.git

```
Alunos@Lab03-Prof MINGW64 ~/Desktop/aula_02 (master)
$ git remote add origin https://github.com/ProfAndersonVanin/curso_react.git
```

Digite:

git branch -M main

git push -u origin main

AULA 03 - REACT

- Fragments

- Os **React Fragments** permite a criação de um componente sem elemento pai;
- O propósito é **descomplicar os nós do DOM**;
- A sintaxe é **<>** e **</>**, não há um nome para a tag;
- Criamos no próprio JSX;

Vamos criar um novo componente chamado List.js

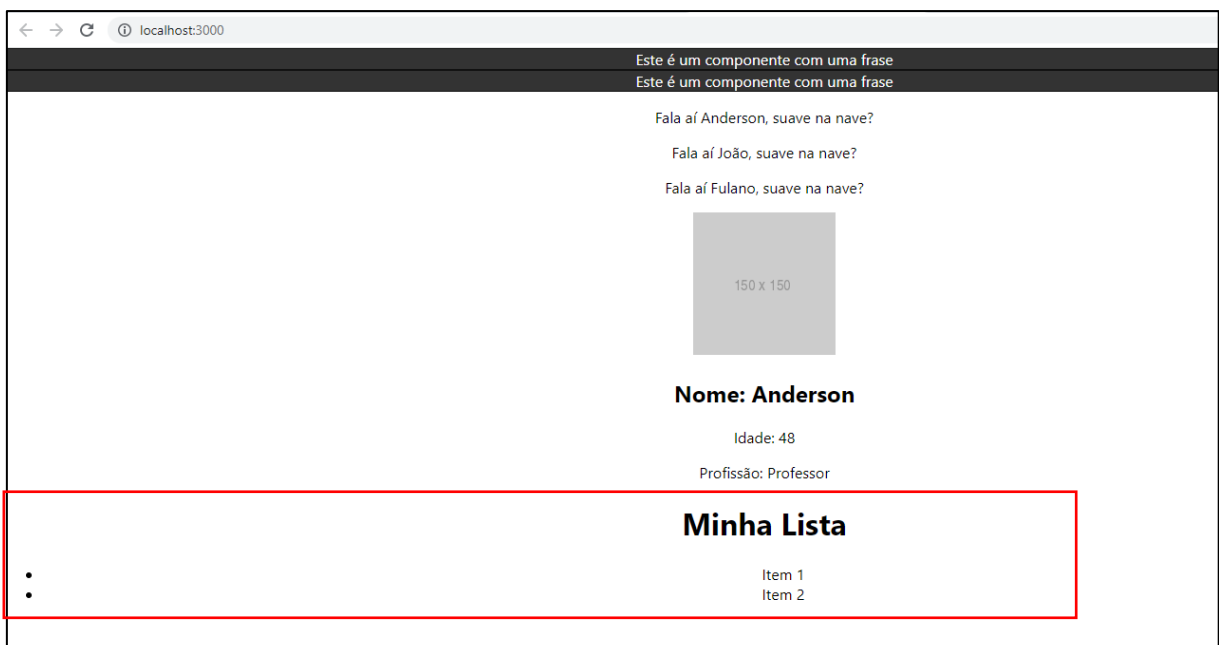
```
aula_03 > src > components > js List.js > ...  
1  function List(){  
2      return(  
3          <>  
4              <h1>Minha Lista</h1>  
5              <ul>  
6                  <li>Item 1</li>  
7                  <li>Item 2</li>  
8              </ul>  
9          </>  
10     )  
11 }  
12  
13 export default List
```

Importe este componente no App.js

```

aula_03 > src > App.js > ...
1   import './App.css';
2
3   import DigameuNome from './components/DigameuNome';
4   import Pessoa from './components/Pessoa';
5   import Frase from './components/Frase';
6   import List from './components/List';
7
8   function App() {
9     const nome = "Fulano"
10    return (
11      <div className="App">
12        <Frase />
13        <Frase />
14        <DigameuNome nome="Anderson"/>
15        <DigameuNome nome="João"/>
16        <DigameuNome nome={nome}/>
17        <Pessoa
18          nome="Anderson"
19          idade="48"
20          profissao="Professor"
21          foto="https://via.placeholder.com/150"
22        />
23        <List />
24      </div>
25    );
26  }
27
28  export default App;

```



```

    <div>...</div>
    <div>...</div>
    <h1>Minha Lista</h1>
    <ul>
      <li>...</li>
      <li>...</li>
    </ul>
  </div>
</div>

```

Cada item da lista pode ser passado dinamicamente através de **props**.



Crie um componente chamado **Item.js**

```

aula_03 > src > components > Js Item.js > ...
1  function Item(props){
2    return(
3      <>
4        <li>{props.marca}</li>
5      </>
6    )
7  }
8
9  export default Item

```

Importe este componente em List.js

aula_03 > src > components >  List.js >  List

```
1  import Item from "../Item"
2
3  function List(){
4    return(
5      <>
6        <h1>Minha Lista</h1>
7        <ul>
8          <Item marca="FIAT"/>
9          <Item marca="VW"/>
10        </ul>
11      </>
12    )
13  }
14
15  export default List
```



- Avançando em props

- Podemos **definir tipos para as props**, realizando uma espécie de validação;
- Definimos em um objeto chamado `propTypes` no próprio componente;
- E ainda há a possibilidade de **definir um valor padrão**;
- Neste caso utilizamos o objeto **`defaultProps`**;

Vamos adicionar mais um props aos itens de nossa lista:

```

aula_03 > src > components > JS Item.js > ...
1  function Item(props){
2      return(
3          <>
4              <li>{props.marca} - {props.ano}</li>
5          </>
6      )
7  }
8
9  export default Item

```

Em ano, queremos passar um valor numérico, então faça a seguinte alteração em List.js

```


aula_03 > src > components > JS List.js > ...
1  import Item from "../Item"
2
3  function List(){
4      return(
5          <>
6              <h1>Minha Lista</h1>
7              <ul>
8                  <Item marca="FIAT" ano={1952}/>
9                  <Item marca="VW" ano={1999}/>
10             </ul>
11          </>
12      )
13  }
14
15  export default List

```

Para passar um valor numérico ao invés das “ ” utilizaremos as { }. Isso faz com que o valor seja interpretado como numérico ao invés de string.

Imagine que no componente List.js não seja enviado a marca ou o ano para o componente Item.js. Podemos utilizar uma classe interna do react que atribui um valor padrão no caso da falta de valores que devem ser passados para um props.

Altere o componente Item.js

```
aula_03 > src > components >  Item.js > ...  
1  import PropTypes from 'prop-types'  
2  
3  function Item(props){  
4    return(  
5      <>  
6        <li>{props.marca} - {props.ano}</li>  
7      </>  
8    )  
9  }  
10  
11  Item.propTypes = {  
12    marca: PropTypes.string.isRequired,  
13    ano: PropTypes.number  
14  }  
15  
16  Item.defaultProps = {  
17    marca: 'Faltou a marca',  
18    ano: 0,  
19  }  
20  
21  export default Item
```

Altere o componente List.js

```

aula_03 > src > components > JS List.js > ...
1  import Item from "../Item"
2
3  function List(){
4      return(
5          <>
6              <h1>Minha Lista</h1>
7              <ul>
8                  <Item marca="FIAT" ano={1952}/>
9                  <Item marca="VW" ano={1999}/>
10                 <Item />
11             </ul>
12         </>
13     )
14 }
15
16 export default List

```

Veja o resultado no Browser com a validação de informações que não foram passadas ao componente Item.js

