

React JS

★ AULA 05 ★

★ PROFESSOR
MAROMO ★

React JS

Props

Propriedade em React



Agenda

- Introdução a propriedades
- Aplicando propriedades
- Criando um aplicativo de exemplo



Props (Conceito)

"Props" é a abreviação de propriedades em React. Elas são um meio de passar dados de componentes pais para componentes filhos. Em essência, as props em React são semelhantes aos argumentos de uma função em JavaScript puro. Elas são imutáveis, o que significa que um componente filho não pode alterar o valor das props recebidas do pai.



Importância

A principal importância das props está em permitir a comunicação e a passagem de dados entre componentes. Elas também auxiliam na reutilização de componentes, uma vez que você pode passar diferentes dados para o mesmo componente, permitindo que ele se comporte de maneira diferente com base nesses dados.



Como usar props

Para usar props em React, você as define no componente pai e as utiliza no componente filho. As props são definidas com um atributo e um valor, e então passadas para o componente filho. No componente filho, elas podem ser acessadas através do objeto props.

Em resumo, as props são uma parte essencial do React, ajudando a criar componentes mais dinâmicos e reutilizáveis.

Mecânica do Props

Funcionamento:

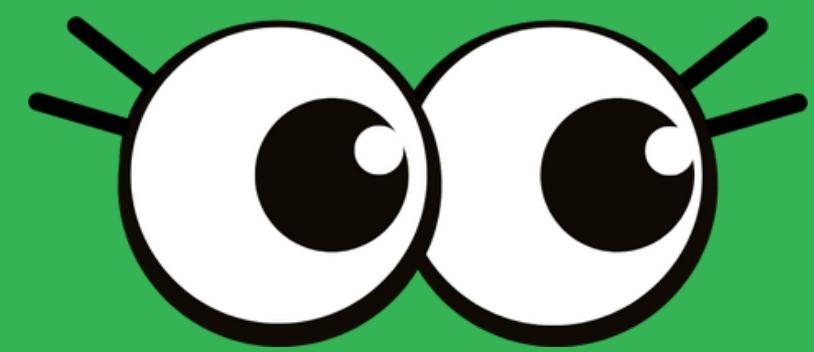
- 1. Criamos nosso componente definindo onde as propriedades recebidas vão aparecer.**
- 2. Importamos o componente e passamos argumentos (valores) para ele.**



Como que eu
uso props ???

Vamos a um exemplo prático





Nossa aplicação é uma tela que exibe uma carta para cada signo, como mostra a Figura.

BARALHO DOS SIGNOS

Aquário



21/01 - 19/02

Peixes



20/02 - 20/03

Áries



21/03 - 20/04

Touro



21/04 - 21/05

Gêmeos



22/05 - 21/06

Câncer



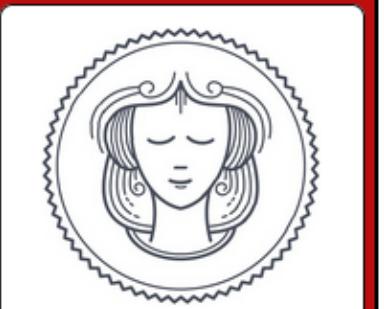
21/06 - 23/07

Leão



24/07 - 23/08

Virgem



24/08 - 23/09

Libra



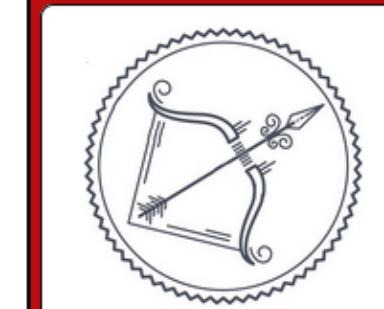
24/09 - 23/10

Escorpião



24/10 - 22/11

Sagitário



23/11 - 21/12

Capricórnio



22/12 - 20/01

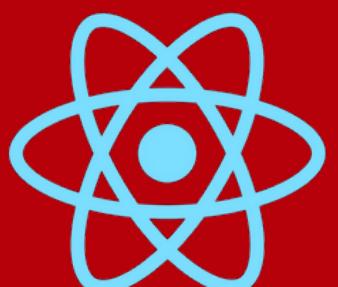
Mão na massa

Zodiac Signs

Laboratório



Desenvolvimento de uma aplicação SPA usando o React JS. Nome da aplicação **sample05**.



React JS

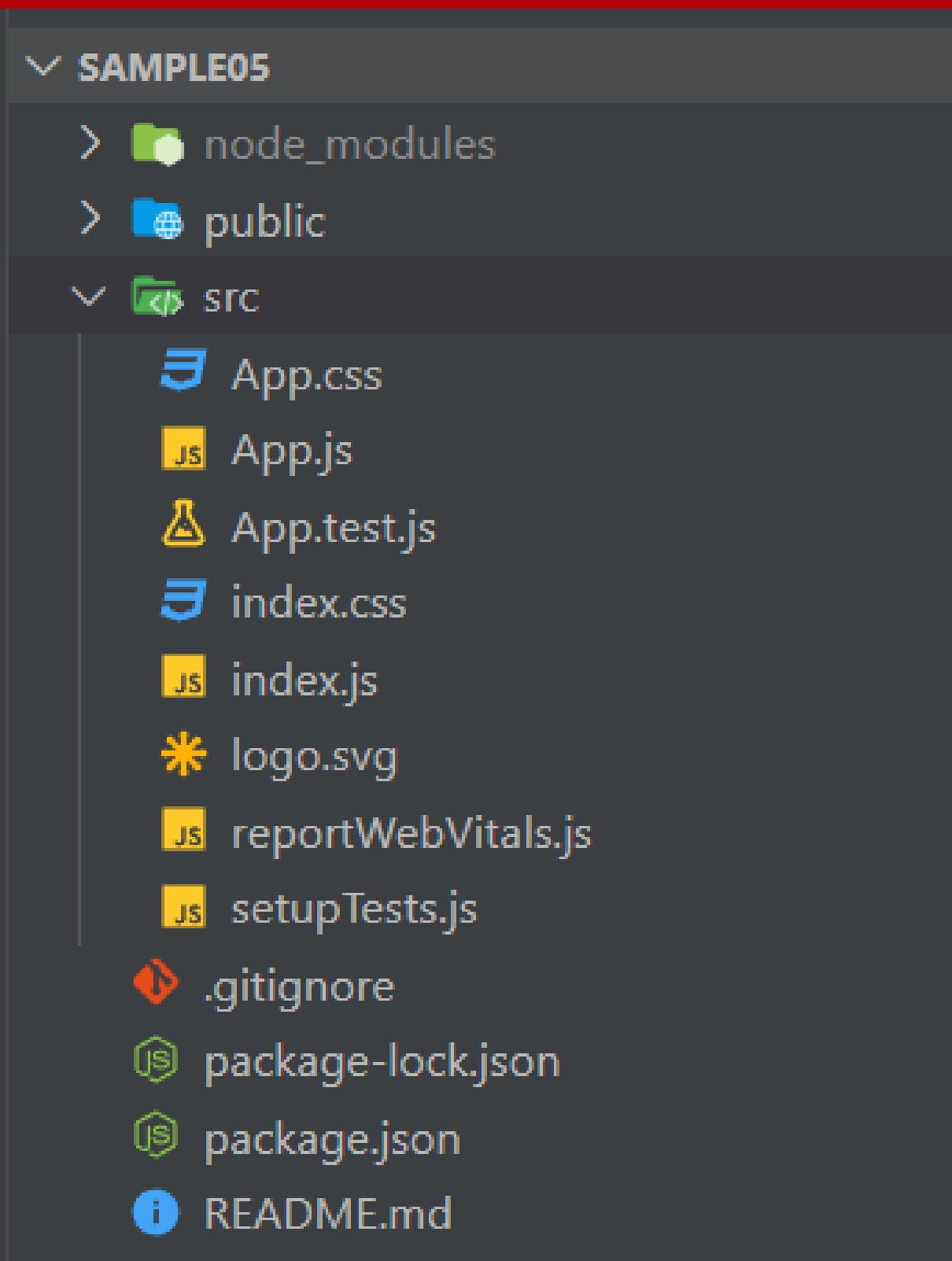


Passo 1: Configuração do ambiente de desenvolvimento

1. Execute o comando `npx create-react-app sample05` para criar um novo projeto React chamado "sample05". Aguarde até que a criação do projeto seja concluída.
2. O processo anterior, de criação do projeto, pode levar alguns instantes enquanto create-react-app monta todo o **boilerplate** mínimo que uma aplicação React para a web precisa ter.
3. Entre no diretório [cd sample05] e digite [code .] para abrir o VS Code.



Passo 2: Estrutura do projeto React

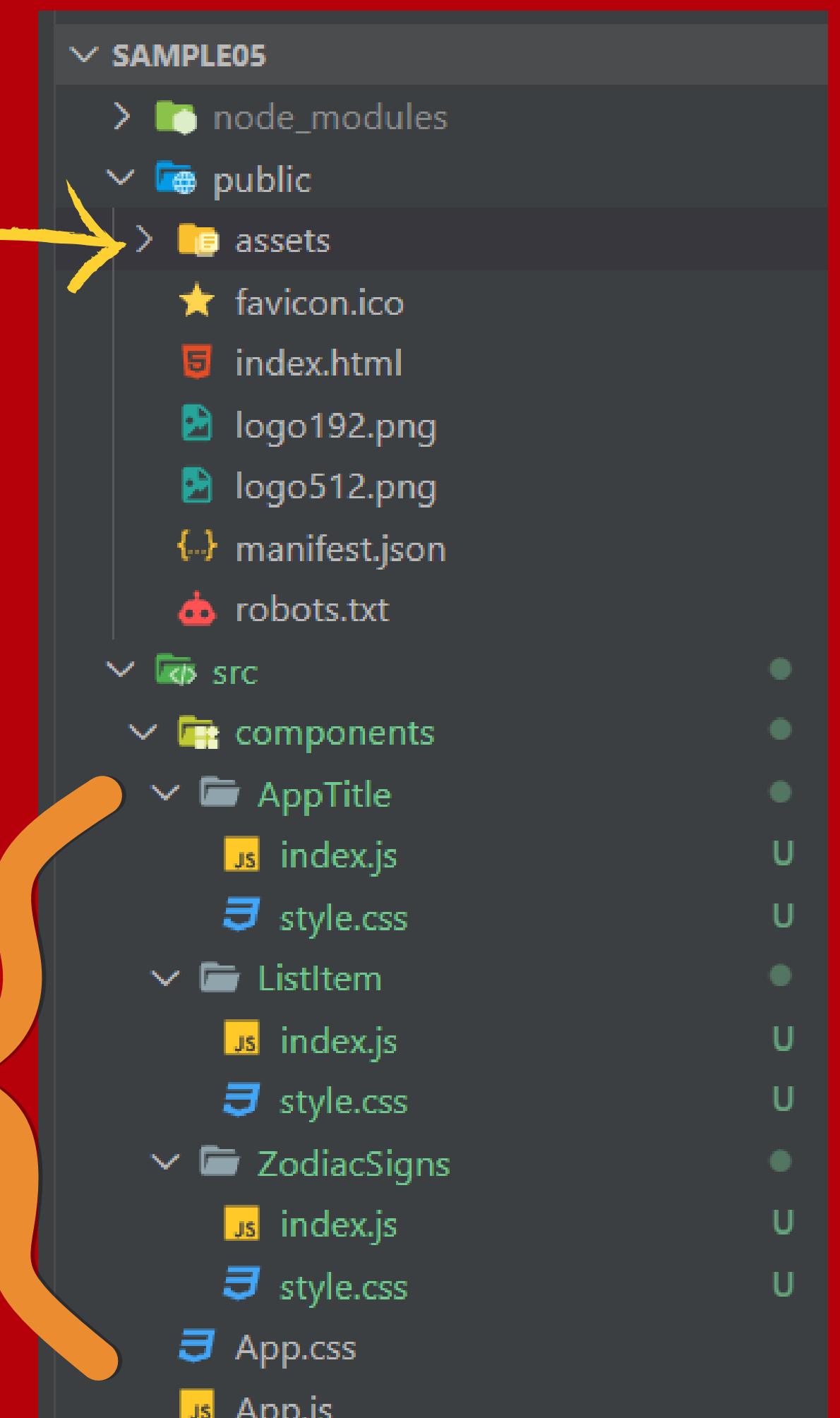


Passo 3

Com os conhecimentos adquiridos nas aulas anteriores, defina a estrutura do projeto, como apresenta a figura:

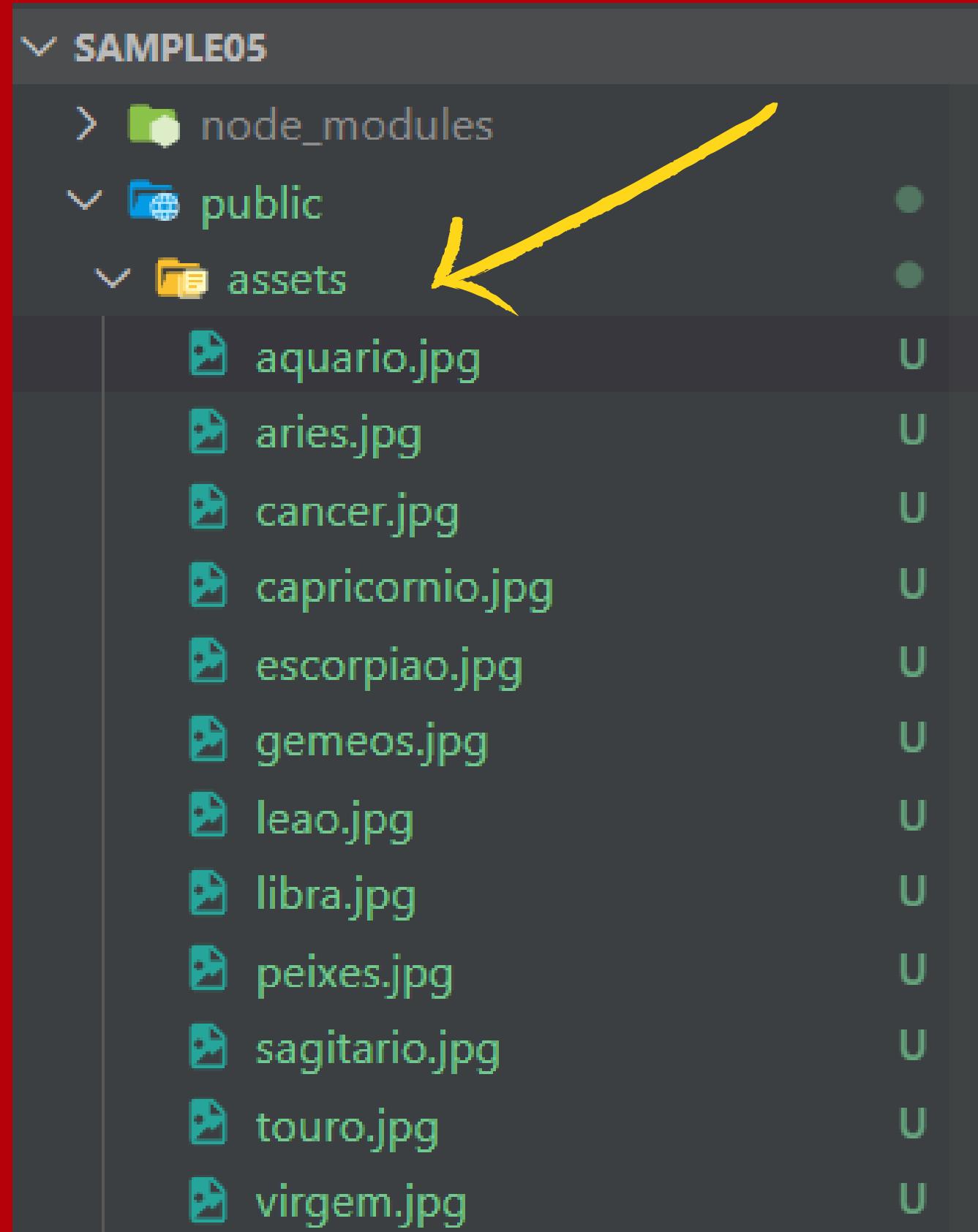
components

assets



**Passo 4 - Copie as imagens fornecidas pelo professor,
e coloque na pasta assets.**

Nota





Divisão da aplicação

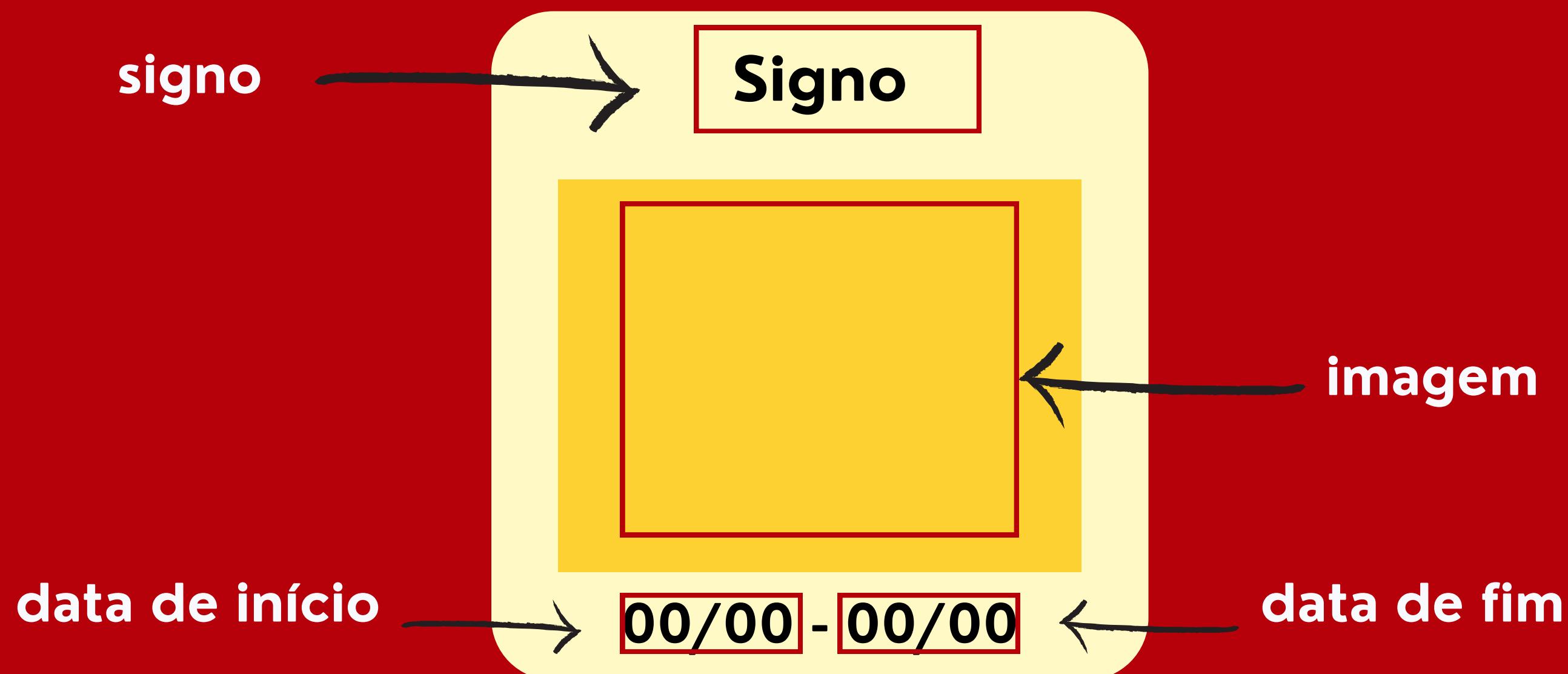
Nossa aplicação está dividida em:

1. **App.js** => componente pai da nossa aplicação.
2. **ZodiacSigns/index.js** => a tela que tem uma barra de título e a lista de cartas.
3. **AppTitle/index.js** => a barra de título.
4. **ListItem/index.js** => o componente (reutilizado) representando cada carta.
5. **style.css** => cada componente possui em suas respectivas pastas este arquivo responsável por estilizar o componente.
6. **public/assets** => as imagens que vamos utilizar no nosso projeto.



Componente ListItem

Vamos aplicar o conceito de **props** no componente **ListItem** definindo onde as propriedades recebidas serão exibidas, como mostra a ilustração da Figura.





Passo 5: Componente AppTitle

```
1 import React from 'react';
2 import './style.css';
3
4 function AppTitle() {
5     return (
6         <div className="box">
7             <p className="texto">Baralho dos Signos</p>
8         </div>
9     )
10 }
11 export default AppTitle
```



Passo 6: Estilo do AppTitle style.css

```
.box {  
    background-color: #6d0c0c;  
    align-items: center;  
    padding: 10px;  
}  
  
.texto {  
    font-size: 32px;  
    font-weight: 600;  
    color: #fff;  
    text-transform: uppercase;  
    text-align: center;  
}
```

Componente: ListItem



```
import React from 'react';
import './style.css';
function ListItem(props) {
  return (
    <div className="boxSigno">
      <div className="nomeSigno">
        {props.signo}
      </div>
      <img className="imagemSigno" src={props.imagem}
           alt={props.signo} />
      <div className="periodoNascimento">
        {props.dataInicio} - {props.dataFim}
      </div>
    </div>
  )
}
export default ListItem
```



Explicando o Listitem

As {props.símbolo}, {props.imagem}, {props.dataInício} e {props.dataFim} são referências a propriedades que são esperadas para serem passadas a este componente quando for usado. O valor dessas propriedades é então renderizado dentro dos elementos div e img.

`<div className="boxSigno">...</div>` - Este é o elemento raiz do componente, que possui a classe CSS boxSigno. Esta classe define o estilo do elemento e seus filhos.

`` - Este é um elemento de imagem que usa duas propriedades: imagem para o atributo src da imagem e símbolo para o atributo alt da imagem.

Assim, ListItem é um componente que espera receber as propriedades símbolo, imagem, dataInício e dataFim, e renderiza essas propriedades em um layout específico. Esse tipo de componente é útil para listas de itens onde cada item tem a mesma estrutura de layout, mas diferentes conteúdos.

Passo 8: Definindo a aparência do ListItem, arquivo: style.css

```
.boxSigno {  
background-color: #856957;  
color: #fff;  
margin: 25px 10px 0;  
padding: 5px 0px;  
display: flex;  
flex-direction: column;  
align-items: center;  
justify-content: space-between;  
box-sizing: border-box;  
border-radius: 5px;  
border: 1px solid #734136;  
width: 150px;  
height: 200px;  
}
```

```
.imagemSigno {  
width: 140px;  
height: 100px;  
background-color: #fff;  
object-fit: contain;  
border-radius: 2px;  
border: 1px solid #734136;  
box-sizing: border-box;  
margin-bottom: 10px;  
padding: 8px;  
flex: 1;  
}  
.nomeSigno {  
font-weight: 600;  
font-size: 18px;  
margin-bottom: 10px;  
}  
.periodoNascimento {  
font-size: 14px;  
}
```



Passo 9: Passando propriedades para o componente ZodiacSigns

```
1 import React from 'react';
2 import AppTitle from '../AppTitle';
3 import ListItem from '../ListItem';
4 import './style.css';
5 function ZodiacSigns() {
6     return (
7         <div>
8             <AppTitle />
9             <div className="lista">
10                 <ListItem
11                     signo="Aquário" dataInicio="21/01"
12                     dataFim="19/02" imagem="assets/aquario.jpg" />
13             </div>
14     </div>
15 }
```

Linha 1 a 13



Passo 9: Passando propriedades para o componente ZodiacSigns

```
14 <ListItem  
15   signo="Peixes" dataInicio="20/02"  
16   dataFim="20/03" imagem="assets/peixes.jpg" />  
17  
18 <ListItem  
19   signo="Áries" dataInicio="21/03"  
20   dataFim="20/04" imagem="assets/aries.jpg" />  
21  
22 <ListItem  
23   signo="Touro" dataInicio="21/04"  
24   dataFim="21/05" imagem="assets/touro.jpg" />  
25  
26 <ListItem  
27   signo="Gêmeos" dataInicio=" 22/05"  
28   dataFim="21/06" imagem="assets/gemeos.jpg" />  
29
```

Linha 14 a 29



Passo 9: Passando propriedades para o componente ZodiacSigns

```
30 <ListItem  
31   signo="Câncer" dataInicio="21/06"  
32   dataFim="23/07" imagem="assets/cancer.jpg" />  
33  
34 <ListItem  
35   signo="Leão" dataInicio="24/07"  
36   dataFim="23/08" imagem="assets/leao.jpg" />  
37  
38 <ListItem  
39   signo="Virgem" dataInicio="24/08"  
40   dataFim="23/09" imagem="assets/virgem.jpg" />  
41  
42 <ListItem  
43   signo="Libra" dataInicio="24/09"  
44   dataFim="23/10" imagem="assets/libra.jpg" />  
45
```

Linha 30 a 45



Passo 9: Passando propriedades para o componente ZodiacSigns

```
46   <ListItem
47     signo="Escorpião" dataInicio="24/10"
48     dataFim="22/11" imagem="assets/escorpiao.jpg" />
49
50   <ListItem
51     signo="Sagitário" dataInicio="23/11"
52     dataFim="21/12" imagem="assets/sagitario.jpg" />
53
54   <ListItem
55     signo="Capricórnio" dataInicio="22/12"
56     dataFim="20/01" imagem="assets/capricornio.jpg" />
57
58   </div>
59 </div>
60 );
61 }
62 export default ZodiacSigns;
```

Linha 46 a 62



Passo 10: Estilo do ZodiacSigns

style.css

```
1 .lista {  
2     display: flex;  
3     flex-wrap: wrap;  
4     justify-content: center;  
5     width: 1024px;  
6     margin: 0 auto;  
7     box-sizing: border-box;  
8 }
```

Configurações finais

Arquivos que não serão comentados

São eles:

- App.js
- App.css

App.js

```
import ZodiacSigns from './components/ZodiacSigns';
import './App.css';

function App() {
  return (
    <ZodiacSigns />
  );
}

export default App;
```

App.css

```
1 .App {  
2     text-align: center;  
3 }
```

BARALHO DOS SIGNOS

Aquário



21/01 - 19/02

Peixes



20/02 - 20/03

Áries



21/03 - 20/04

Touro



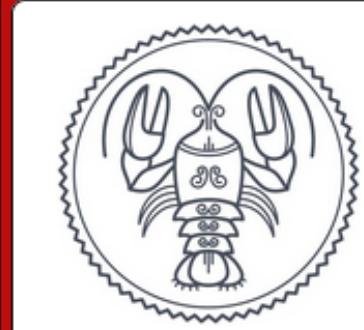
21/04 - 21/05

Gêmeos



22/05 - 21/06

Câncer



21/06 - 23/07

Leão



24/07 - 23/08

Virgem



24/08 - 23/09

Libra



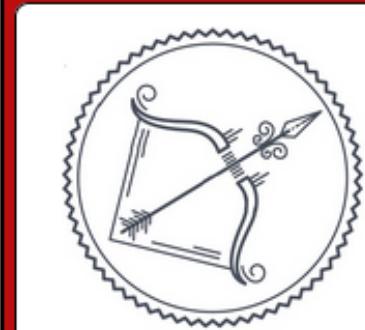
24/09 - 23/10

Escorpião



24/10 - 22/11

Sagitário



23/11 - 21/12

Capricórnio



22/12 - 20/01

Resultado da Execução



Resumo

Como vimos, "props" (abreviação de propriedades) são um aspecto fundamental da programação em React, facilitando a passagem de dados entre componentes e tornando-os reutilizáveis e dinâmicos.

Elas permitem que os componentes sejam configuráveis e promovem o encapsulamento e a separação de responsabilidades.

Ao manter as props imutáveis, o React garante previsibilidade e consistência na renderização dos componentes, resultando em interfaces de usuário mais estáveis e de fácil manutenção.

O tempo é o melhor professor,
mas infelizmente acaba matando
todos os seus alunos



FIM

