

React JS

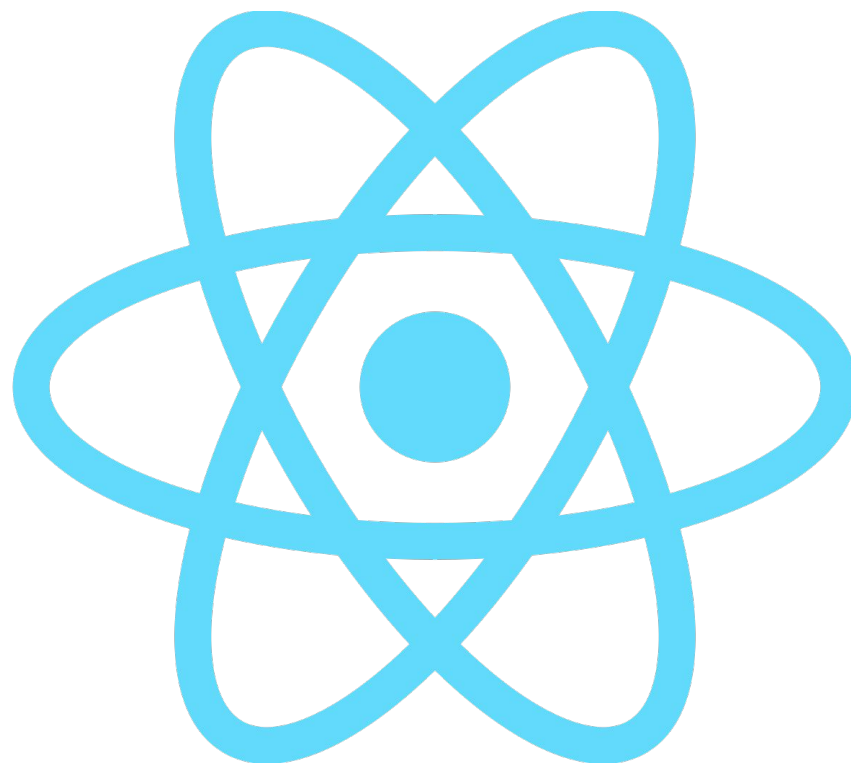
Aula 03

Instrutor: Rodolfo Moura

Intro

Revisão dos
comentários sobre
componentes do React

Comentários sobre
Javascript e Html



JSX

O que é?

Html ou Javascript?

User Interface

Transpilação

```
const element = <h1 className="title">Hello world!</h1>;
```

JSX

Após o transpiler:

```
const element = React.createElement(  
  "h1",  
  { className: "title" },  
  "Hello world!"  
)
```

O código fica pronto para ser exibido no DOM

JSX

Utilização de variáveis com ‘{ }’
dentro do código

Uso normal de funções

Inserção de valores em atributos de tags

Na utilização de vários elementos, envolver com
<div> </div>

JSX

Vantagens:

Muito parecido com o HTML

Integração entre o HTML e Javascript

Nosso projeto avisa quanto a variáveis/partes do código não utilizadas

Componentes

Servem para dividir a aplicação em partes

Componentes renderizam em JSX

Criamos arquivos de componentes

Usamos Export e Import para utilizá-los

Componentes

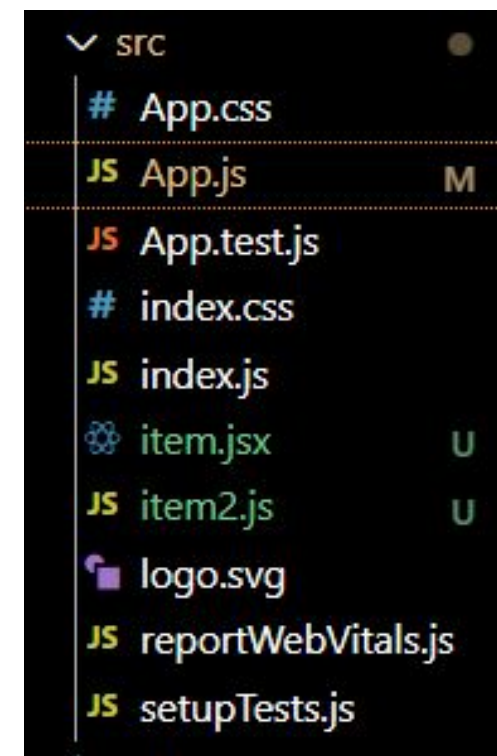
Criação de novos componentes

Geralmente na pasta SRC, podendo também

Optar pela criação de pasta apenas de componentes

Criamos um arquivo Javascript

Prestar atenção ao uso de CamelCase



Componentes (no arquivo do componente)

Exemplo com função para ser chamada

Lembrar do uso de parênteses

```
function Item2(){  
  return /
```

Abrimos as chaves para a inserção do código que ficaria em JSX

Exemplo simples com Return

Componentes (no arquivo do componente)

Observação:

Podemos chamar a função também através do uso de classes, nesse caso a sintaxe ficaria assim:

```
import React from 'react';
```

```
class Item3 extends React.Component {  
  render(){  
    return  
  }  
}
```

Componentes (no arquivo do componente)

Lembrar do uso da `<div>` para a inserção de mais comandos (+1 linha)

```
<div>  
|   <h1>Componente teste</h1>  
</div>
```

Para podermos chama-lo no nosso app temos que exportá-lo

```
export default Item2
```

Temos ele pronto para ser referenciado no nosso app!

Componentes (no arquivo do App)

Importamos o arquivo para o App.js

Lembrar de usar o nome exato

```
import Item2 from './item2'
```

Atentar ao aviso no Terminal

Para chama-lo usamos o uso de < />

```
<Item2/>
```

O componente aparecerá!

Componentes

Podemos fazer o uso de componentes dentro de outros componentes!

Usamos o **import** dentro do componente a ser usado e o “chamamos” (`< />`).

Criamos um reaproveitamento de componentes e uma “escada”

Atividade

...