



Habilitação Profissional Técnica em Informática para Internet

Sintaxe ReactJS



O que são componentes React

Componentes react são tags ou componentes html, com funcionalidades.

- Componentes comuns:
 - Tags HTML exemplo: `<div>` , `<link>`, `<input>`, etc...
 - Possui propriedades já definidas como `name`, `alt`, `src`, etc...
 - E eventos, como `onmouseover`, `onload`, `onclick`, `onchange`, etc...
- Componentes customizados:
 - Tags com nomes livres: `<meu-componente>` , `<nome>` , etc...
 - Sem propriedades pré definidas, liberdade para criar novas : `<meu-componente prop="OK">`
 - Eventos precisam ser declarados manualmente caso necessário



Fragments

Fragmentos: usados para agrupar mas de um elementos dentro de um componente customizado.

Podendo ser em forma de tag simples:

```
<></>
```

ou completa, sendo necessário a versão completa quando precisar declarar a chave única:

```
<Fragment key={item.id}></Fragment>
```



Condicionais booleanos: *if()*, *else*

Assim como no JavaScript, podemos utilizar lógicas condicionais para carregar conteúdos a medida que precisarmos, podemos utilizar chaves para mais de uma linha, ou sem chaves para informações de uma linha apenas:

```
if (condição booleana) {  
  let variavel = true;  
  return true;  
}
```

```
if (condição booleana)  
  return true;
```

Também podemos utilizar a versão de apenas uma linha, simplificando o retorno:

```
let variavel = condição booleana ? "Retorno verdadeiro" : "Retorno Falso";  
{condição booleana ? (<div>verdadeiro</div>) : (<div>falso</div>)}
```



Loops de repetição: *map()*, *forEach()*

Muitas vezes precisamos repetir a informação repetidas vezes pela tela, podemos fazer isso utilizando as várias formas de repetição do Javascript.

```
let meuArray = ["informacao1", "informacao2"];
meuArray.forEach(function (valor, indice) {
  console.log("Valor do array " + valor + " indice " + indice )
})
```

Como na maioria dos casos vamos precisar para exibir informações na tela, utilizaremos o ".map" para mapear o nosso array, alterando a cada indice presente em um formato de componentes reconhecido pelo react:

```
meuArray.map(function (valor, indice) {
  return ( <div>{indice} {valor}</div> );
});
return meuArray
```



Declaração de estados: *useState()*

Utilizamos o *useState()* para instanciar uma nova variável que será reconhecida pelo react. Assim que ela for atualizada automaticamente em toda a página é possível visualizar as mudanças. A sintaxe do *useState* é esta:

```
const [idade, definirIdade] = useState(28);
```

Onde utilizamos a desestruturação de um array em javascript para pegar o retorno do método *useState()* para definir [*nome da variável a ser atualizada* , *método que atualiza a variável*].

Como parâmetro da função *useState()* podemos colocar o valor inicial para quando esta variável for inicializada. Este parâmetro não é obrigatório, mas é uma boa prática para ajudar a identificar o tipo desta variável.

Por conta da forma como o React inicializa a variável, ela *não pode ser alterada diretamente*, apenas utilizando o método descrito. Ou ela perderá as propriedades reativas dela. Por isso definimos como uma constante para auxiliar a IDE a entender isso.

Manipular array com o *useState()*

	Evitar (modifica o array)	Preferir (retorna um novo array)
adicionar	<code>push</code> , <code>unshift</code>	<code>.concat</code> , <code>[...arr]</code>
remover	<code>pop</code> , <code>shift</code> , <code>splice</code>	<code>.filter</code> , <code>.slice</code>
substituir	<code>splice</code> , <code>arr[i] = ... atribuição</code>	<code>map</code>
ordenar	<code>reverse</code> , <code>sort</code>	copiar o array anteriormente

Embora existam vários métodos para manipular arrays e objetos no JS, somente alguns métodos podem ser utilizados no React. São aqueles que ao invés de manipular o array, retorna um array novo. A tabela acima demonstra quais podem ser utilizados



Links

- Components: <https://react.dev/reference/react-dom/components>
- Fragments: <https://react.dev/reference/react/Fragment>
- UseState: <https://react.dev/reference/react/useState>
 - <https://react.dev/learn/updating-arrays-in-state#removing-from-an-array>