

Recapitulando

DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

O ***useState*** nos permite criar estados em um componente criado a partir de uma função, assim como o “*state*” presente em componentes criados a partir de classes. A inicialização do “*useState*” retorna uma variável e um método para atualizá-la.

Importante destacar que a cada atualização de uma variável com `useState`, seu valor será atualizado também na view, ou seja, é feita uma nova renderização do componente.

```
const [text, setText] = useState('Senai')
```

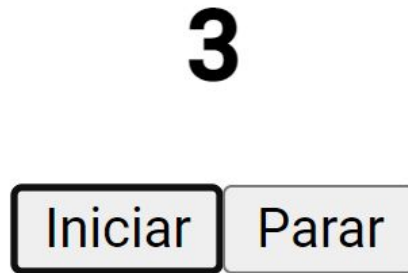
O ***useEffect*** é um Hook que serve para lidar com os efeitos. Podemos usá-los como os lifeCycles `componentDidMount`, `componentDidUpdate` e `componentWillUnmount` ou para executar determinada ação após uma alteração em alguma variável, para isso passamos o array de dependências com a variável de controle.

```
useEffect(() => {  
  console.log('Lorem ipsum.')})
```

Exercício 1 - Cronômetro (15min)

Construir um cronômetro em segundos utilizando Hooks, adicionar um botão para iniciar a contagem e um segundo botão para parada da contagem mas sem ação.

Para adicionar um valor a cada segundo utilize o método [setInterval\(\)](#).





useRef



DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

O Hook **useRef** é uma função que retorna um objeto ref mutável, que é um objeto JavaScript simples, no qual a propriedade **.current** desse objeto é inicializada com o argumento passado (initialValue).

O objeto retornado persistirá durante todo o ciclo de vida do componente. E com este hook podemos manter uma variável mutável e acessar imperativamente um componente filho.

```
const refInput = useRef(initialValue)
```

Declarando useRef

```
const countRef = useRef(0);
```

Variável

Valor inicial de
"countRef"

Exemplo 1

```
1  import { useRef } from 'react';
2
3  export default function ContagemRef() {
4
5      const countRef = useRef(0);
6
7      let value = 0;
8
9      const contar = () => {
10         countRef.current ++;
11         value += 1;
12         console.log(`Ref: ${countRef.current}`);
13         console.log(`Value: ${value}`)
14     };
15
16     return <button onClick={contar}>Clickar!</button>;
17 }
```


Qual será a saída no console do código anterior?

Pelo **useRef** persistir o valor durante todo o ciclo de vida do componente, isso significa que o componente não perderá o valor da variável declarada por meio de **useRef** caso haja uma atualização da visualização.

Exemplo 2

```
1  import './style.css';
2  import { useRef } from 'react';
3
4  export default function Case2Ref() {
5
6      const profissaoRef = useRef();
7
8      function analisaTecla(e) {
9          if (e.key === 'Enter') {
10              profissaoRef.current.focus();
11          }
12      }
13
14      return (
15          <div>
16              <input type='text' onKeyUp={analisaTecla} placeholder='Nome'></input>
17              <input ref={profissaoRef} type='text' placeholder='Profissão'></input>
18              <button>Enviar</button>
19          </div>
20      )
21  }
```

O código anterior mostra que a variável retornada pelo useRef combinado com um componente HTML, funcionará como uma declaração do tipo `document.getElementById()` que é muito utilizado no Javascript. A manipulação por meio deste comando não é muito utilizado no React justamente pelas possibilidades e simplificação com o useRef.

Exemplo 3

```
1  import React, { useRef, useState, useEffect } from "react";
2
3  export default function Case3Ref() {
4    const [count, setCount] = useState(0);
5    const prevCountRef = useRef();
6
7    useEffect(() => {
8      prevCountRef.current = count;
9    });
10
11    const prevCount = prevCountRef.current;
12
13    return (
14      <>
15        <h3>
16          Valor do contador agora: {count} <br />
17          Valor do contador antes: {prevCount}
18        </h3>
19        <button onClick={() => setCount(count + 1)}>Somar um ao contador</button>
20      </>
21    );
22  }
```

Exercício 2 - Cronômetro (10min)

Com o código do exercício 1, defina uma ação de parar a contagem do tempo iniciada com o `setInterval()`.

Esse método, `setInterval`, pode ser parado com a declaração `clearInterval(id)`.

Para adicionar um valor a cada segundo utilize o método [setInterval\(\)](#).

3

Iniciar

Parar

Classname Dinâmica

DEVinHouse

Parcerias para desenvolver a sua carreira


SENAI

<LAB365>

className é o atributo de um componente para atribuir classes css a ele, seu uso é simples e existe a possibilidade de adicionar um conteúdo dinâmico, como alteração de cor de um botão quando ele está ativo.

className - Uso

```
16   const botaoAtivo = 1;  
17  
18   return (  
19     <div className='div-padrao'>  
20       <button className={botaoAtivo ? 'btn-ativo' : 'btn-inativo'}>Enviar</button>  
21     </div>  
22   )
```



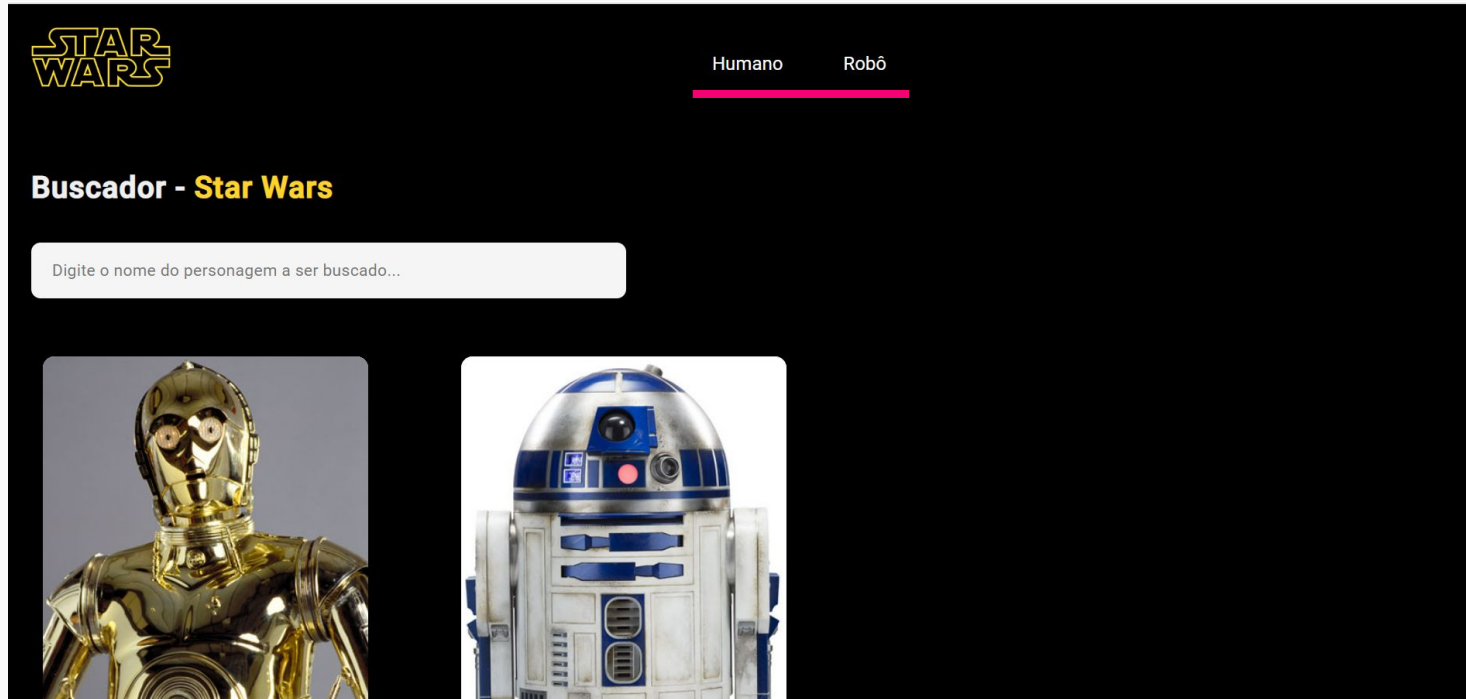
Este modelo de vinculação de classe funciona como um if. Se a condição passar, adiciona o primeiro item ao className, caso a condição não seja validada o segundo item será adicionado.

Exercício 3 - Star Wars (20min)

Com o código do Star Wars, adicione um campo para escolher se o usuário deseja buscar por nome ou tipo de personagem.

Considere tipo a “origem” do personagem, se humano ou robô.

Exercício 3 - Star Wars (20min)



Ref: Buscando por robô.

Bibliografia

- Documentação oficial sobre Hooks - <https://pt-br.reactjs.org/docs/hooks-intro.html>
- Ciclos de vida de um componente react - <https://blog.cod3r.com.br/ciclos-de-vida-em-react/>
- Repo com os códigos das aulas - <https://github.com/yanestevesufjf/clamed-react>

AVALIAÇÃO DOCENTE

O que você está achando das minhas aulas neste conteúdo?

[Clique aqui](#) ou escaneie o QRCode ao lado para avaliar minha aula.

Sinta-se à vontade para fornecer uma avaliação sempre que achar necessário.





DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!



<LAB365>

AGENDA

- Informações do template
- Exemplo com texto (Light mode)
- Exemplo com texto (Dark mode)
- Exemplo com imagens (Light mode)
- Exemplo com imagens (Dark mode)

INFORMAÇÕES DO TEMPLATE

- Título da Apresentação:
 - Fonte: Ubuntu Bold
 - Formato: Maiúsculo
 - Tamanho: 34
 - Cor: Branco
- Título do Slide:
 - Fonte: Ubuntu Bold
 - Formato: Maiúsculo
 - Tamanho: 22
 - Cor: Branco
- Parágrafos:
 - Fonte: Open Sans Normal
 - Tamanho: 14 a 18
 - Cores: Branco (Dark Mode) ou Preto (Light Mode)
- Marcadores de tópicos:
 - Formatos: Símbolos ou Alfanuméricos
 - Cor: Laranja
- Padrão de Cores:
 - Cinza - #868584
 - Preto - #1C1C19
 - Branco - #FAFAFA
 - Laranja - #F08305
 - Rosa - #c71d81
 - Azul - #0e1d8e

EXEMPLO COM TEXTO (LIGHT MODE)

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

EXEMPLO COM TEXTO (DARK MODE)

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

INTERVALO DE AULA

DEV!

Finalizamos o nosso primeiro período de hoje. Que tal descansar um pouco?!

Nos vemos em 20 minutos.

Início: 20:20

Retorno: 20:40



EXEMPLO COM IMAGENS (LIGHT MODE)

Título

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.



Título

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

EXEMPLO COM IMAGENS (DARK MODE)

Título

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.



Título

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.