



Decorando o espaço

Módulo 3 - Aula 6 - HARD

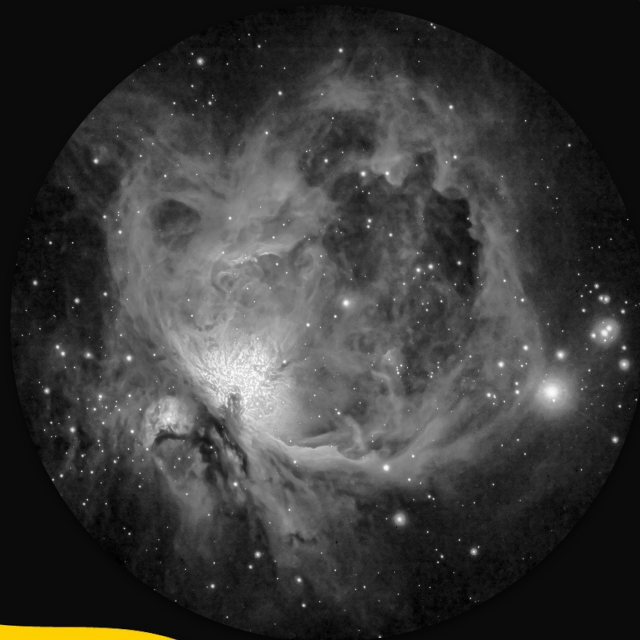


Photo by Aldebaran Son Unsplash



Todos os direitos reservados
©2022 Resilia Educação



Review

RECAP



Componentes de classes

Componentes de funções



A complex electronic circuit board, possibly a breadboard or a custom PCB, is shown against a dark background. The board is populated with numerous electronic components, including resistors, capacitors, and integrated circuits. A series of bright blue, glowing lines are drawn across the board, representing a signal path or a specific circuit configuration. The word "Props" is written in a bold, yellow, sans-serif font in the center of the image.

Props



TODO: Componente Tarefa.jsx

- ⇒ Crie o componente Tarefa.jsx
- ⇒ Dentro dele iremos:
 - Definir uma propriedade “titulo”
 - Conter uma check-box
 - Conter um label com o valor de “titulo” passado como prop
 - Quando clicarmos no label devemos marcar ou desmarcar o check-box

A photograph of a clothing rack against a light gray background. The rack is a dark metal bar with several hangers. From left to right, there are: a light green jacket, a tan jacket, a black leather jacket, a white ribbed sweater, and a light blue denim jacket. A white fedora hat with a black band is hanging on the right side of the rack. The text "Estilização no React" is overlaid in the center in a bold, yellow, sans-serif font.

Estilização no React



CSS Import

Podemos estilizar nossos componentes da mesma maneira que fazemos com o simples HTML:

- criamos nosso arquivo .css,
- usamos os seletores corretos
- e assim estilizamos.

CSS Import

A diferença está na importação da folha de estilo.

No HTML iríamos usar a tag `<link>` passando o atributo `href` com o caminho do arquivo.

No React iremos usar a palavra reservada **import** e passar o arquivo da folha de estilo.

CSS Import

```
import './caminho-da-folha-de-estilo'

function Button() {
  return (
    <button className="btn">Comprar</button>
  )
}
```

CSS Import - Desvantagem

O Webpack é responsável por agrupar os componentes no processo de build, onde as folhas de estilos também serão unidas. Imagine que dois ou mais componentes **usem o mesmo seletor**, é possível que haja **quebra de estilo**.

Quanto maior o projeto, maiores as chances de quebra na consistência de estilo

CSS Import - Build

Caso você queira entender melhor o processo de build do **Vite** é só rodar o comando:

```
npm run build
```


CSS Modules

Sua forma de estilizar é diferente. O CSS Modules garante que **não haverá conflitos de estilização** no seu projeto, pois ele **gera identificadores únicos** para os seus seletores.

Ou seja, a estilização pode impactar múltiplos elementos, mas apenas no contexto do componente.

CSS Modules

Quando comparado ao CSS Import há algumas diferenças. A primeira delas se dá pela extensão do arquivo CSS, agora ela passa a ser **.module.css**

Nesse arquivo a sintaxe permanece a do simples CSS, nada muda.

CSS Modules

A diferença está em como iremos dispor os seletores no componente.

É fortemente recomendado que se use classes para estilizar seus elementos, lembre-se que IDs são usados para identificar apenas um elemento e não é ideal usá-los para estilização.

CSS Modules - Utilização

Imagine um arquivo CSS com essa classe tarefa

```
.tarefa {  
  width: max-content;  
  height: 40px;  
  padding: 0 5px;  
  
  display: flex;  
  align-items: center;  
  justify-content: space-between;  
  
  border: 1pt solid #111111;  
}
```


CSS Modules - Utilização

Ao chamarmos em nosso componente React, temos que nomear o arquivo `.module.css` como uma variável, comumente chamada de **styles**.

Após nomearmos, iremos definir os valores de atributos de classe (`className`) como sendo **styles.classeQueCriamos**
Todas as classes precisam seguir o padrão camelCase.

CSS Modules - Utilização

```
import React from "react";
import styles from "./tarefa.module.css";

function Tarefa({ titulo }) {
  return (
    <article className={styles.tarefa} role='task' aria-labelledby='title'>
      <label id='title' htmlFor='status'>
        {titulo}
      </label>
      <input id='status' type='checkbox' />
    </article>
  );
}

export default Tarefa;
```



Importação de Imagens, SVGs e Fontes



Importação de fontes

A importação das fontes permanece do jeito que estamos habituados com o código HTML puro.

Basta importar o CDN na tag `<head>` dentro do index.html do projeto e chamar a fonte no arquivo CSS.

Importação de fontes

Nada impede também de termos arquivos de fontes (.ttf, por exemplo) no nosso projeto e importá-las usando o **@font-face** do CSS.

Importação de imagens

Podemos importar imagens como faríamos em um arquivo HTML. Basta tê-la adicionada em uma pasta do projeto, ou ter sua URL na web e passar seu caminho para o atributo **src** da tag ****

Também podemos passar via atributo **url** do CSS

Importação de imagens - CSS

```
.cachorro {  
  width: 100px;  
  height: 50px;  
  background-image: url('./imagem/cachorro.jpg');  
  background-size: cover;  
}
```

Importação de imagens

Há apenas uma pequena diferença na importação quando temos um caminho de imagem no próprio projeto, devemos utilizar o método **require()**.

Como ele é um método devemos passá-lo dentro das **chaves { }**

Exemplo:

```
<img src={require('./imagem/cachorro.jpeg')} alt="Cachorro" />
```

É importante colocar o **./** no caminho, pois o Webpack irá substituir esses caminhos para um caminho final após o processo de build

Importação de imagens

Podemos importar imagens como uma **variável** direto para o componente.

Isso **facilita na manutenção**, caso o caminho seja alterado ou se precisarmos **importar múltiplas vezes**.

Não se preocupe! O Webpack vai gerar o caminho correto na build final.

Importação de imagens - Exemplo

Exemplo:

```
import cachorroImg from './imagem/cachorro.jpeg'

export default function Card() {
  return (
    <div className="card">
      <img src={cachorroImg} alt="cachorro" />
    </div>
  )
}
```

Importação de SVGs

SVGs são **gráficos vetoriais escalonáveis**, é uma linguagem XML (semelhante ao HTML) usada para descrever **desenhos e gráficos bidimensionais**.

Traz várias vantagens seu uso em aplicações web, já que ocupam muito pouco espaço quando comparados a imagens PNG, por exemplo.

Importação de SVGs

Os SVGs podem ser adicionados da mesma maneira que imagens, seguindo os passos anteriores.



Importação de SVGs

Mas também podem ser importados como componente.

Dessa maneira todo o código SVG é injetado direto no HTML, dando maior controle sobre o mesmo.

Para isso usamos **NomeDoSVG** na hora de importar

Importação de SVGs - Exemplo

Exemplo:

```
import { PhoneIcon } from './icons/phone-icon'
```

Importação de SVGs

Também podemos definir um SVG como um componente React.
Sendo o retorno o código SVG

OBS.: propriedades que tiverem hífen serão modificadas para camelCase
fill-rule vira **fillRule**.

Mão na Massa: Importando imagens

- ⇒ Criar uma pasta images dentro de src
- ⇒ Baixar imagens do site [Unsplash](https://unsplash.com)
- ⇒ Movê-las para a pasta images
- ⇒ Importá-las e exibir na aplicação

Dica: usá-las como variáveis



Atividade em pares

