

# Recapitulando

DEVinHouse

Parcerias para desenvolver a sua carreira

**SENAI**

<LAB365>

# Styled Components

Trata-se de uma biblioteca (lib) que nos permite escrever códigos CSS dentro do JavaScript. Dessa maneira não precisamos mais ficar importando nossos arquivos .css em nossas páginas e se um dia precisarmos utilizar esse mesmo componente em outro projeto, uma das maneiras seria basicamente copiar o arquivo .js.

# Styled Components - Instalação

```
npm install styled-components --save
```

# Styled Components - Uso

```
1  import styled from "styled-components";
2
3  export const MeuComponentePersonalizado = styled.div`
4    display: flex;
5    flex-direction: column;
6    max-width: 240px;
7    padding: 20px;
8    margin: 0 auto;
9  `;
10
11 export const Titulo = styled.h4`
12   font-size: 24px;
13   color: #202020;
14 `;
15
16 export const SubTitulo = styled.h6`
17   font-size: 16px;
18   color: #aaa;
19   font-weight: 400;
20 `;
```

# Styled Components - Uso

```
1  import React from 'react';
2  import { MeuComponentePersonalizado, Titulo, SubTitulo } from './styles';
3
4  export default function StyledComponentExample() {
5
6      return (
7          <MeuComponentePersonalizado>
8              <Titulo>Olá, bem vindo</Titulo>
9              <SubTitulo>Informe seu nome</SubTitulo>
10          </MeuComponentePersonalizado>
11      )
12  }
```

# Estilos Dinâmicos

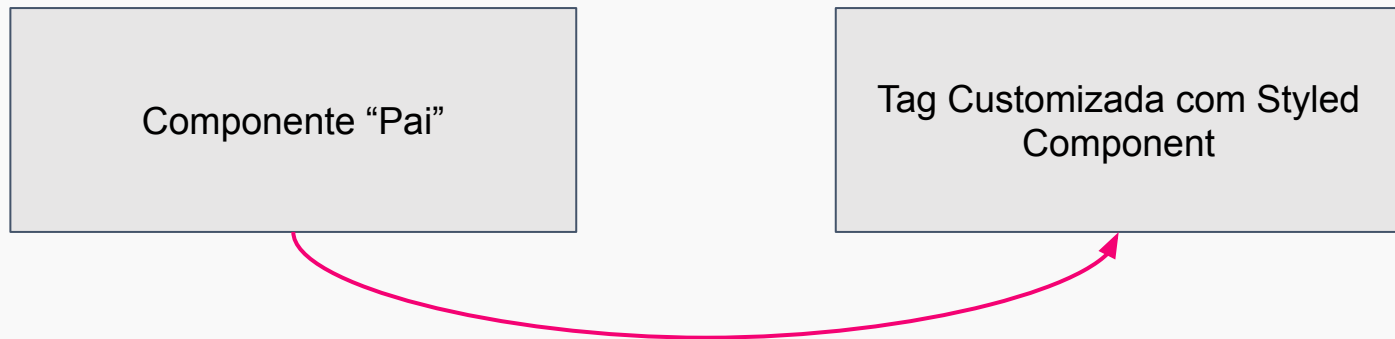
DEVinHouse

Parcerias para desenvolver a sua carreira

**SENAI**

<LAB365>

# Styled Components - Fluxo das props



O componente, que faz a referência da tag customizada, injeta as propriedades na tag customizada criada com Styled Component.

# Estilos Dinâmicos - styled components

```
5  export default function StyledComponentExample() {
6
7    const [nome, setNome] = useState()
8
9    return (
10      <MeuComponentePersonalizado>
11        <Titulo>Olá, bem vindo</Titulo>
12        <SubTitulo>Informe seu nome</SubTitulo>
13        <input value={nome} onChange={(e) => setNome(e.target.value)}/>
14        <Button valid={nome}>Enviar</Button>
15      </MeuComponentePersonalizado>
16    )
17  }
```



# Estilos Dinâmicos - styled components

```
26  export const Button = styled.button`  
27      margin-top: 30px;  
28      height: 28px;  
29      width: 120px;  
30      border-radius: 3px;  
31      border: none;  
32      color: white;  
33      cursor: pointer;  
34      background-color: ${props => props.valid ? 'green' : 'red'}  
35  `
```

## Exercício 1 - Menu (20min)

Construa um menu que mantenha visualmente o estado da página aberta, a estilização aplicada deve ser de aumentar o peso da fonte e aplicar um sublinhado a tag selecionada.

Home

Produtos

Carrinho

Minha Conta

# Composition

DEVinHouse

Parcerias para desenvolver a sua carreira

**SENAI**

<LAB365>

Composition, ou composição de componentes, é um padrão poderoso para tornar seus componentes mais reutilizáveis. No React, podemos tornar os componentes mais genéricos aceitando props, que são para os componentes do React o que os parâmetros são para as funções.

Composição de componentes é o nome para passar componentes como adereços (“props”) para outros componentes, criando assim novos componentes com outros componentes.

Portanto, composição nada mais é que enviar componentes por meio das chamadas “props”. Podemos explicar a composição a partir de sites de notícias que mantêm matérias protegidas a partir de um número de visitas no mês. Os dados da matéria podem vir de um servidor e este dado é renderizado em um componente genérico para o usuário, o fluxo acaba sendo um retorno de tags HTML e esse retorno é passado a um componente genérico da notícia.

# Composition

```
export default function Site() {  
  return (  
    <SiteLayout>  
      <HeaderComponent></HeaderComponent>  
      {/* Componente filho carregado e envio uma propriedade content*/}  
      <ContentComponent content=  
        // O valor da propriedade content, ao invés de ser uma variável  
        // passa a ser um conteúdo HTML  
        <div>  
          <h2>Título da Primeira Notícia</h2>  
          <p>Lorem ipsum ...</p>  
        </div>  
      } />  
      <FooterComponent></FooterComponent>  
    </SiteLayout>  
  )  
}
```

# Composition

```
export default function ContentComponent({ content }) {  
  return (  
    <>  
      {content}  
    </>  
  )  
}
```



## Exercício 2 - Composition (20min)

Construir uma **página de configurações (C1)** com um checkbox ou qualquer outro elemento de controle de estado (podendo ser menu), que seja responsável pelo que será mostrado no **componente filho (C2)**. Caso a **variável de controle seja 0**, envie por props um texto com **“Nenhuma informação a mostrar”** e caso a **variável seja 1**, injete um outro **componente (C3)** por composição no **“C2”**.

# Formulários

DEVinHouse

Parcerias para desenvolver a sua carreira

**SENAI**

<LAB365>

Diferente do Angular e Vue.js, o React não possui sistema de data binding de formulários integrados ao template, ele é fortemente baseado em JavaScript/Typescript, no entanto podemos simular isto atribuindo eventos de onChange a campos de texto, checkbox, radio buttons, etc, e então alterar o estado do componente com o novo valor.

```
const [form, setForm] = useState({
  email: '',
  senha: ''
})

function updateForm(key, value) {
  let newData = { ...form };
  newData[key] = value;
  setForm(newData)
}

function enviarForm(event) {
  console.log('-- formulario --')
  console.log(form)
  event.preventDefault();
}
```

# Forms

```
<div className='login-page'>
  <form onSubmit={enviarForm}>
    <fieldset>
      <label>E-mail</label>
      <input type='email' value={form.email} onChange={e => updateForm('email', e.target.value)} />
    </fieldset>

    <fieldset>
      <label>Senha</label>
      <input type='password' value={form.senha} onChange={e => updateForm('senha', e.target.value)} />
    </fieldset>

    <input type='submit' value='Entrar' />
  </form>
</div>
```

# Validação de formulário

Por ser fortemente baseado em Javascript, a validação de formulário do React será basicamente a utilizada no HTML+JS puro. Mas com a ajuda do hooks podemos simplificar alguns processos, como mostrar mensagens de erro.

# Validação de formulário

```
const [status, setStatus] = useState({
  type: '',
  mensagem: ''
});

function enviarForm(event) {
  if (!validate()) return;
  event.preventDefault();
}

function validate() {
  if (!form.email) return setStatus({ type: 'error', mensagem: 'Erro: Necessário preencher o campo e-mail!' });
  if (!form.senha) return setStatus({ type: 'error', mensagem: 'Erro: Necessário preencher o campo senha!' });
  return true;
}
```

# Validação de formulário

```
<div className='login-page'>
  <form onSubmit={enviarForm}>
    <fieldset>
      <label>E-mail</label>
      <input type='email' required value={form.email} onChange={e => updateForm('email', e.target.value)} />
    </fieldset>

    <fieldset>
      <label>Senha</label>
      <input type='password' required min={6} value={form.senha} onChange={e => updateForm('senha', e.target.value)} />
    </fieldset>

    {status.type === 'error' ? <p style={{ color: "#ff0000" }}>{status.mensagem}</p> : ""}

    <input type='submit' value='Entrar' />
  </form>
</div>
```



## Exercício 3 - Login (20min)

Construir dois componentes (páginas), uma para a home e outra de login. Adicionar um formulário de login que redirecione o usuário para a rota /home quando o e-mail for válido e a senha for maior que 8 caracteres.

- Construindo formulários com formik - <https://medium.com/reactbrasil/formik-construa-formul%C3%A1rios-em-react-sem-l%C3%A1grimas-a80c52887882>
- Validação de formulários com react-hook-form - <https://www.devmedia.com.br/validando-formularios-com-react-hook-forms/42903>
- Exemplo de validação com criação de hook <https://dev.to/jucian0/reactjs-formularios-55mo>
- 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.