



DESENVOLVIMENTO FULL STACK

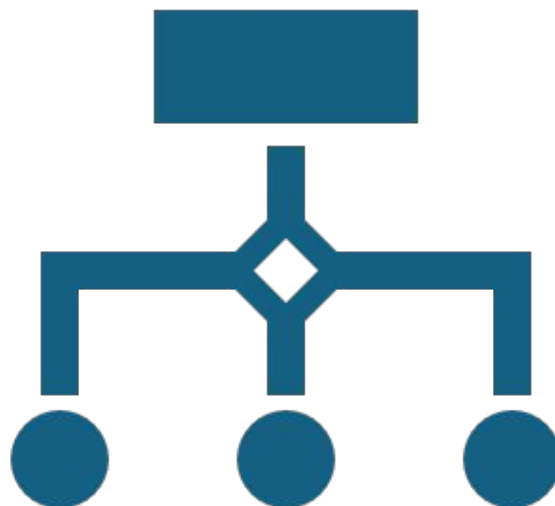
React - Componentes

Caio Eduardo do Prado Ireno

caio.ireno@faculdadeimpacta.com.br

O QUE É COMPONENTES EM REACT?

Os componentes são um dos principais conceitos do React. Eles são a base sobre a qual você constrói interfaces de usuário (UI), o que os torna o lugar perfeito para começar sua jornada no React!



Componentes: blocos de construção da interface do usuário

Na Web, o **HTML** nos permite criar documentos estruturados ricos com seu conjunto integrado de tags como `<h1>`

```
<article>
```

```
<h1>My First Component</h1>
```

```
<ol>
```

```
<li>Carro</li>
```

```
<li>Moto</li>
```

```
<li>Bike</li>
```

```
</ol>
```

```
</article>
```

Esta marcação representa um **artigo**, seu **título** e um índice (abreviado) como uma **lista ordenada**. Tais marcações, combinada com CSS e JavaScript para interatividade, está por trás de cada barra lateral, modal, menu suspenso...

O React permite combinar sua marcação, CSS e JavaScript em "componentes" personalizados, elementos de interface do usuário reutilizáveis para seu aplicativo.



DICAS

Conforme seu projeto se expande, você perceberá que muitos dos seus designs podem ser criados reutilizando componentes já desenvolvidos, o que agiliza significativamente o processo de desenvolvimento.

Por exemplo, o índice mencionado acima pode ser facilmente integrado a qualquer tela do projeto! Além disso, é possível começar seu trabalho utilizando milhares de componentes compartilhados pela comunidade de código aberto do React, como os oferecidos pelas bibliotecas **Chakra UI** e **Material UI**.

Definindo um componente

O React prioriza a criação de interfaces interativas de forma simples e eficiente, utilizando uma base familiar: **JavaScript**.

Um componente React nada mais é do que uma **função JavaScript** que combina **lógica e estrutura**, permitindo que você incorpore **elementos de marcação** diretamente no código de maneira intuitiva.

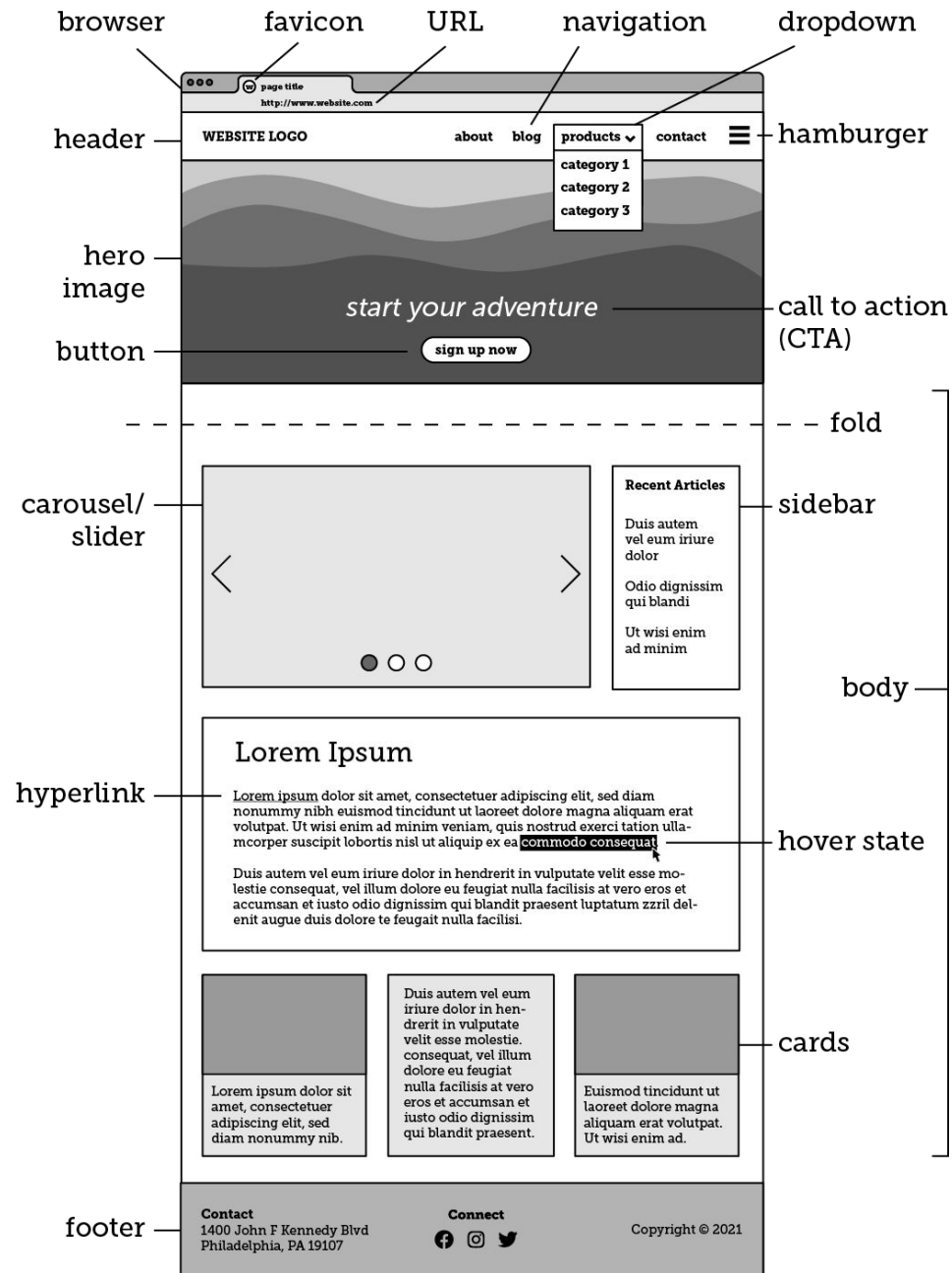
Vamos pensar em um projeto...

O que uma página web tem?

O ideal é dividir o aplicativo em pequenos componentes para facilitar a manutenção do mesmo.



Iremos trabalhar durante as aulas com componentes funcionais.



Vamos iniciar um novo projeto

Vamos rodar: `npm create vite@latest .`

Para esse projeto vamos utilizar: **React** e **Typescript**

```
> npx
> create-vite .

✓ Package name: ... projeto-frontend
✓ Select a framework: » React
✓ Select a variant: » TypeScript
```



O que é TypeScript?

"TypeScript é JavaScript com sintaxe para tipos."

TypeScript é uma linguagem de programação fortemente tipada que se baseia em JavaScript, oferecendo melhores ferramentas em qualquer escala."

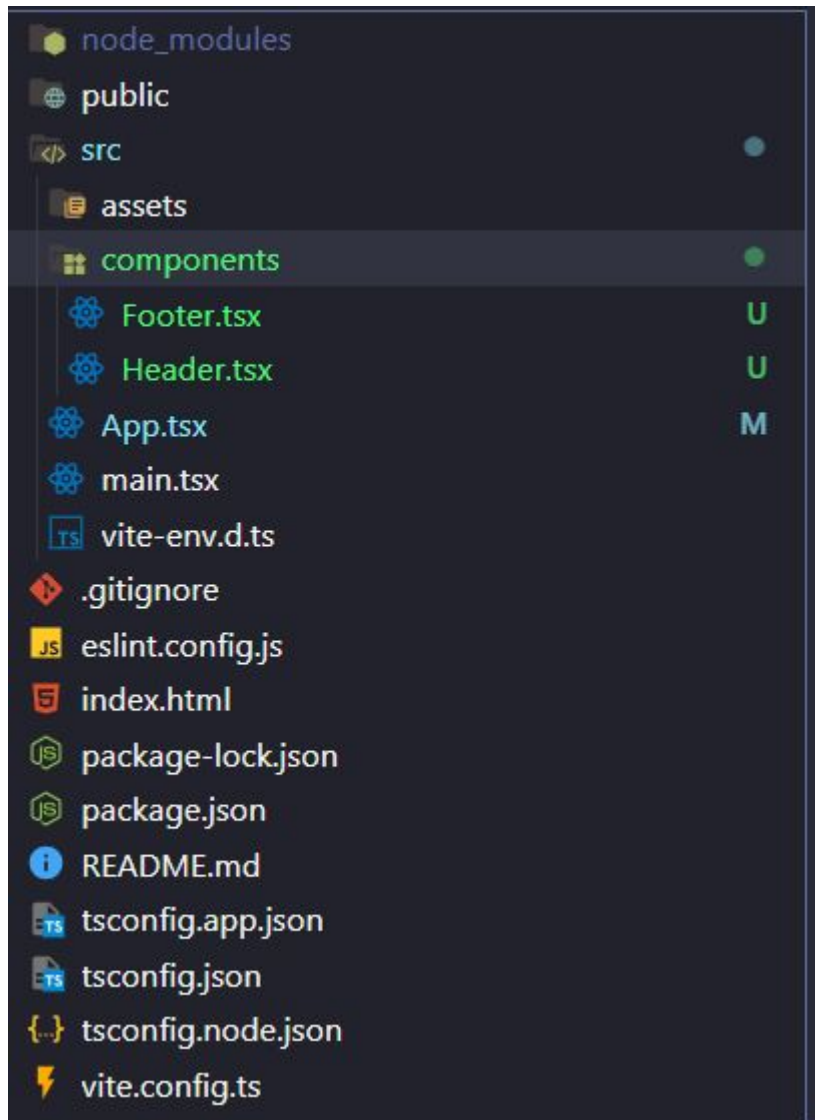
<https://www.typescriptlang.org/pt/>



A principal diferença entre **JavaScript (JS)** e **TypeScript (TS)** é que TS é um superset de JS que adiciona **tipagem estática** e **recursos avançados**.

Isso permite detectar erros em tempo de desenvolvimento, enquanto JS é mais flexível, porém menos seguro, pois é dinamicamente tipado.

Nosso projeto



Header.tsx

```
import React from "react";

const Header: React.FC = () => {
  return (
    <header
      style={{
        backgroundColor: "#f9f4f1",
        padding: "10px",
      }}
    >
      <h1>Header</h1>
    </header>
  );
};

export default Header;
```

Footer.tsx

```
import React from "react";

const Footer: React.FC = () => {
  return (
    <footer
      style={{
        backgroundColor: "#f8f9fa",
        padding: "10px",
        textAlign: "center",
      }}
    >
      <p>© 2025 Meu Footer</p>
    </footer>
  );
};

export default Footer;
```

Componentes

Um componente deve sempre retornar algo. O retorno pode ser qualquer tipo de dado aceitado pelo JSX (string, array, um elemento JSX, null e etc).

Um componente deve sempre retornar um elemento único no return. Caso você deseje retornar mais de um elemento, envolva os mesmos em uma div ou dentro do `<React.Fragment></React.Fragment>` ou `<></>`

Não existe limite para a composição de componentes, eles podem ser desde componentes gerais como Header e Footer, até micro componentes como Input e Button.



Obrigado!
