

Arquitetura JavaScript (React I)

CAPÍTULO 2 – INTRODUÇÃO AO REACT

PROF. RAPHAEL GOMIDE

React I

CAPÍTULO 2 – INTRODUÇÃO AO REACT

PROF. RAPHAEL GOMIDE

Nesta aula



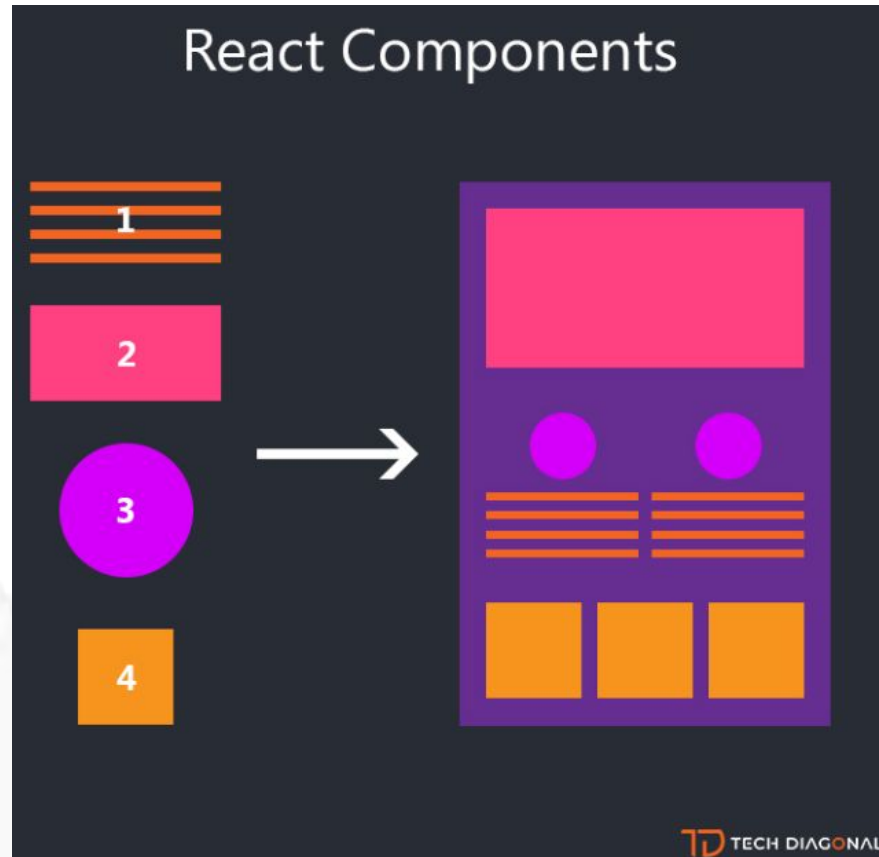
- ❑ Introdução ao React.
- ❑ Criação do projeto **react-hello**.

O que é o React?



- “A JavaScript **library** for building user interfaces”.
- Declarativo:
 - Componentes reativos com JSX.
 - **Mais foco** no **estado** do app e **regras de negócio**;
 - **Menos foco** em manipulação do **DOM manual**.
 - Manipulação do DOM performática (Virtual DOM).
- Baseado em componentes:
 - Alto grau de **reutilização de código**.

O que é o React?



[Fonte](#)

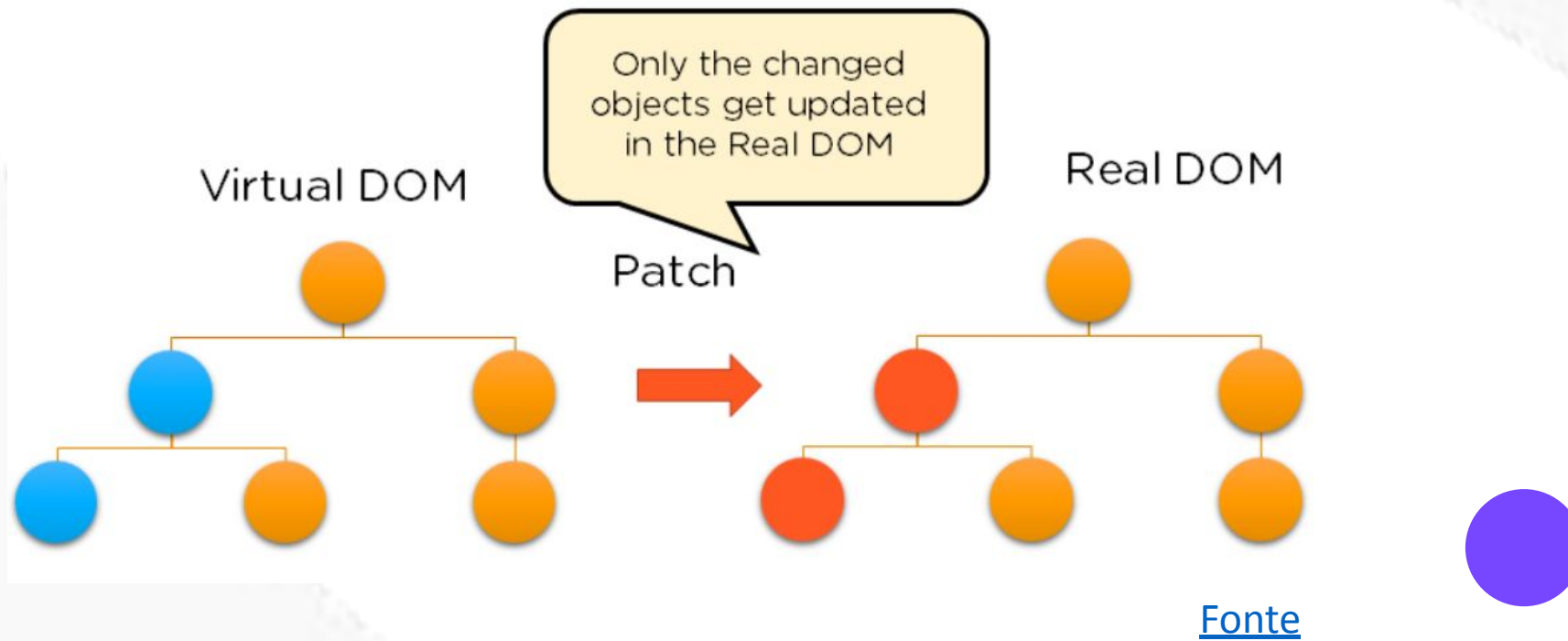
Virtual DOM



- Manipulação performática do DOM.
- A manipulação manual do DOM é considerada uma operação **cara (lenta)**.
- O React só modifica o DOM nos locais que foram realmente alterados.
- Esse processo é mais conhecido como [Reconciliation](#).



Virtual DOM



Termos importantes



- Componente:
 - **Função** que agrupa determinado comportamento.
 - Em regra, realiza processamento e retorna dados renderizados (HTML + CSS).
 - Se bem escritos, podem ser reaproveitados.
 - Uma aplicação React é geralmente composta por diversos **componentes**.
- JSX:
 - JavaScript XML.
 - Utilizada pelo React para a escrita da renderização de **componentes**.



Termos importantes



- Sintaxe de importação e exportação:
 - Palavras-chave **import** e **export**.
 - Faz parte do JavaScript Moderno (ES6+).
 - Utilizadas para definirmos o que utilizar (**import**) e o que fornecer (**export**) durante a escrita do código-fonte.
- props:
 - **Propriedades** de componentes. Consideradas “somente leitura”.
 - Semelhantes aos atributos de tags HTML.
 - Existe uma **prop** especial – *children*.
 - Utilizados para o **envio** e **recebimento** de dados entre **componentes**.

Termos importantes



- Closure:
 - **Função** implementada dentro do **escopo** de outra **função**.
 - Permite o **acesso** ao **escopo externo**.
 - Muito utilizado pelo React na criação de **componentes**, através da criação de funções para lidar com ocorrência de eventos (cliques, digitação etc.).
- State:
 - Representa o **estado** de componentes.
 - **Estado**: dado que pode ser modificado com o tempo.
 - Exemplos: dados de formulários, Back End etc.

Termos importantes



- Hooks:
 - Estrutura do React que pode ser vinculada (hooked) a componentes.
 - Internamente, são implementadas como **funções**.
 - Atuam diretamente na reatividade do React.
 - **Convenção**: começar com o prefixo “**use**”.
 - Principais hooks: **useState** e **useEffect**.
 - É possível criar nossos próprios hooks.



Termos importantes



- useState:
 - Hook para lidar com **estado**.
 - Devolve um valor e função modificadora (setter).
 - Sempre que o setter é invocado, uma nova renderização do componente é agendada e pode ocorrer a qualquer momento.
 - Isso é feito de forma assíncrona.



Termos importantes



- useEffect:
 - Hook para lidar com “efeitos colaterais” (**side effects**).
 - Palavra-chave: **sincronização**.
 - Estado do app x estado do mundo.
 - Permite, por exemplo:
 - Manipular o DOM manualmente.
 - Monitorar eventos do navegador.
 - Trabalhar com o Back End.

Termos no código



- [Link no CodeSandBox.](#)

```
1 // Importação do hook useState
2 import { useEffect, useState } from "react";
3
4 /**
5  * Declaração do componente
6  * O export indica que ele
7  * poderá ser utilizado externamente
8  */
9 export default function App() {
10   // CSS in JS
11   const { containerStyle } = styles;
12
13   // Exemplo de utilização de useEffect
14   useEffect(() => {
15     document.title = "react-counter";
16   }, []);
17
18   // JSX
19   return (
20     <div style={containerStyle}>
21       <h1>react-counter</h1>
22
23       {/* Utilização de outro componente */}
24       <Counter />
25     </div>
26   );
27 }
```

Termos no código



- [Link no CodeSandbox.](#)

```
29 // Outro componente
30 function Counter() {
31   // Exemplo de utilização de useState
32   const [count, setCount] = useState(0);
33
34   // Closure
35   function handleDecrement() {
36     setCount((currentCount) => currentCount - 1);
37   }
38
39   // Closure
40   function handleIncrement() {
41     setCount(count + 1);
42   }
43
44   // JSX com props
45   return (
46     <div>
47       <button onClick={handleDecrement}>-</button>
48       <input type="text" value={count} readOnly />
49       <button onClick={handleIncrement}>+</button>
50     </div>
51   );
52 }
```

react-counter

A screenshot of a web application titled 'react-counter'. It features a text input field with the value '0'. To the left of the input is a button with a minus sign ('-'), and to the right is a button with a plus sign ('+').

Prática



- Acompanhe o professor:
 - Criação do projeto **react-hello**.

Próxima aula



- Construção de componentes mais robustos.
- Criação do projeto **react-countries**.