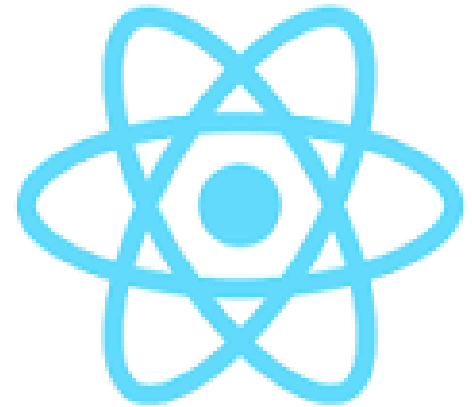


Programação Web



Aula 02 - ReactJS: JSX

Profa. Me. Janaine C. S. Arantes

Nessa Aula (02)

- Introdução React
- SPAs
 - Vantagens
 - SPA x Página Web Tradicional
- JSX
- React com JSX
- Exemplos

Introdução React

- React é uma biblioteca JavaScript para a construção de interfaces de usuário, criada pelo Facebook em 2013.
- Objetivo: tornar o desenvolvimento de Single Page Applications (SPAs) mais fácil.
- React cria um DOM virtual (VDOM) em memória.
- O VDOM é um conceito de programação em uma representação ideal, ou "virtual", da interface do usuário, que é mantida em memória e sincronizada com o DOM "real" por uma biblioteca como o ReactDOM. Esse processo é chamado de reconciliação.
- Versão atual: V17.0.1 (2021).

SPAs

- São aplicações desenvolvidas em JavaScript, que rodam quase inteiras no lado do cliente (browser).
- Assim que o usuário aceita o site, a aplicação fica armazenada do lado do cliente em forma de templates (pedaços de HTML).
- As aplicações feitas em SPA fazem uma transição entre templates, que estão dentro do browser, e só fazem requisições no servidor para buscar dados brutos de conteúdo, que são enviados, normalmente, em JSON.
- O Google foi o primeiro nessa tecnologia, com o Gmail.

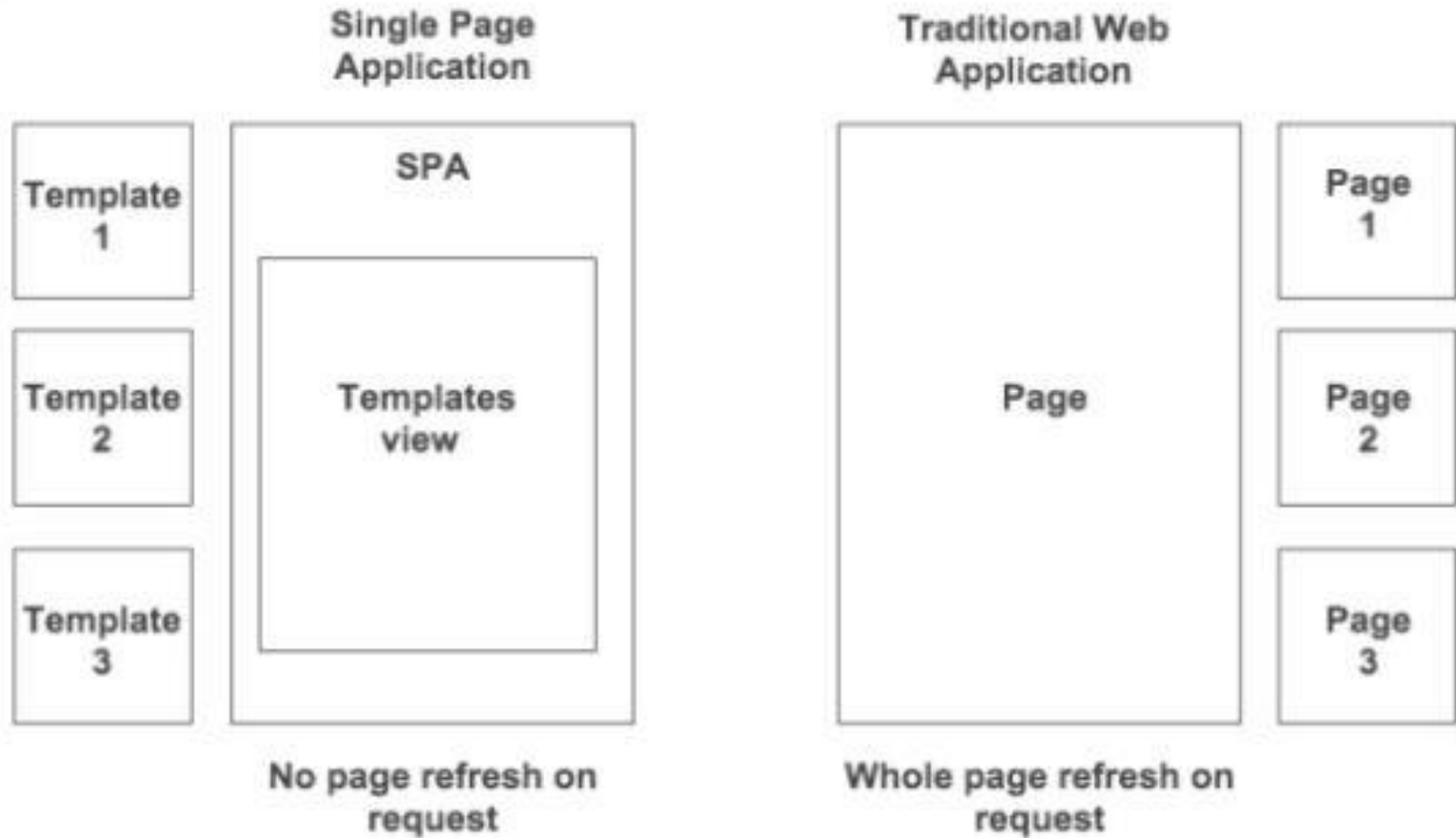
Vantagens

- Melhor experiência do usuário.
- Melhor performance.
- Menor carga de trabalho no servidor.
- Não há reloads nem carregamento de página inteira, apenas trechos da página são mudados, dependendo da ação do usuário, fazendo uma conexão e transferência do servidor para o cliente muito mais leve e rápida.

Vantagens

- O servidor envia somente os dados do processo em formato JSON para o navegador do usuário.
- Com o padrão SPA considere um hidden e show de partes do HTML, formando uma página apresentável para o cliente.
- O servidor recebe a requisição, faz o seu processamento, e monta um JSON como resposta.
- Serviço com menos processo.
- Tráfego mais leve entre o cliente e o servidor.
- Tudo fica mais rápido.

SPA x Página Web Tradicional



JSX

JSX

- JSX significa JavaScript eXtension, ou seja, uma extensão de sintaxe para o JavaScript.
- JSX: JavaScript + XML.
- Permite escrever elementos HTML em JavaScript.
- Converte tags HTML em elementos React.
- Não é obrigatório usar JSX para escrever aplicações React.

Babel

- Compila JSX em código JavaScript.
- Transpilador de JavaScript, que traduz o código escrito em ES6+ para ES5, para que funcione nos browsers, que não implementam certas funcionalidades do ES6+.

ReactJS com JSX

```
<head>
  <!-- 3 CDNs - React, React DOM e Babel -->
  <script src="https://unpkg.com/react@17/umd/react.development.js"></script>
  <script src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
  <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>
</head>

<!-- Container DOM ao HTML -->
<div id = "root"></div>

<!-- Utilização do JSX -->
<script type="text/babel">
  // Código React.
</script>
```

<CTRL> + <SHIFT> + P
Selecionar: LiveServer - Open with Live Server

Exemplo 01

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Exemplo 01</title>

    <!-- 3 CDNs - React, React DOM e Babel -->
    <script src="https://unpkg.com/react@17/umd/react.development.js"></script>
    <script src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
    <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>
  </head>
  <body>
    <div id="root"></div>

    <script type="text/babel">
      const meuElemento = <h1>Olá React!</h1> ;
      ReactDOM.render(meuElemento, document.getElementById('root'));
    </script>
  </body>
</html>
```

A função ReactDOM.render() recebe dois argumentos:

- Código HTML + Elemento HTML.

O seu propósito é mostrar o código HTML dentro do elemento HTML, previamente especificado.

Exemplo 02

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Exemplo 02</title>
    <!-- 3 CDNs - React, React DOM e Babel -->
    <script src="https://unpkg.com/react@17/umd/react.development.js"></script>
    <script src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
    <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>
  </head>
  <body>
    <div id="root"></div>

    <script type="text/babel">
      const root = document.getElementById('root');
      let nome = "Janaine Arantes";
      ReactDOM.render(<h3>O meu nome é: {nome}</h3>, root);
    </script>
  </body>
</html>
```

Exemplo 03

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Exemplo 03</title>
    <!-- 3 CDNs - React, React DOM e Babel -->
    <script src="https://unpkg.com/react@17/umd/react.development.js"></script>
    <script src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
    <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>
  </head>
  <body>
    <div id="root"></div>

    <script type="text/babel">
      // Expressões em JSX.
      const meuElemento = <h1>React é {5 + 5} vezes melhor com JSX.</h1>;
      ReactDOM.render(meuElemento, document.getElementById('root'));
    </script>
  </body>
</html>
```

Exemplo 04

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Exemplo 04</title>
    <!-- 3 CDNs - React, React DOM e Babel -->
    <script src="https://unpkg.com/react@17/umd/react.development.js"></script>
    <script src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
    <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>
  </head>
  <body>
    <div id="root"></div>

    <script type="text/babel">
      /* Para escrever múltiplas linhas de HTML, como por exemplo
         uma lista, coloque o HTML dentro de parenteses.
      */
      const meuElemento = (
        <ul>
          <li>Maçãs</li>
          <li>Bananas</li>
          <li>Laranjas</li>
        </ul>
      );
      ReactDOM.render(meuElemento, document.getElementById('root'));
    </script>
  </body>
</html>
```

Exemplo 05

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Exemplo 05</title>

    <!-- 3 CDNs - React, React DOM e Babel -->
    <script src="https://unpkg.com/react@17/umd/react.development.js"></script>
    <script src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
    <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>
  </head>
  <body>
    <div id="root"></div>

    <script type="text/babel">
      /* Se você quiser escrever dois cabeçalhos, você deve
         colocá-los dentro de um elemento pai, como o div.
      */
      const meuElemento = (
        <div>
          <h1>Eu sou um cabeçalho.</h1>
          <h1>Eu também sou um cabeçalho.</h1>
        </div>
      );
      ReactDOM.render(meuElemento, document.getElementById('root'));
    </script>
  </body>
</html>
```

Exemplo 06

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Exemplo 06</title>
    <!-- 3 CDNs - React, React DOM e Babel -->
    <script src="https://unpkg.com/react@17/umd/react.development.js"></script>
    <script src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>
    <script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>
  </head>
  <body>
    <div id="root"></div>
    <script type="text/babel">
      // Elementos HTML devem ser fechados corretamente.
      const meuElemento = <input type="text" />;
      ReactDOM.render(meuElemento, document.getElementById('root'));
    </script>
  </body>
</html>
```

Referências

- REACTJS.ORG. **React**, c2020. Página inicial. Disponível em: <<https://pt-br.reactjs.org/>>. Acesso em: 22 de fev. de 2021.
- STEFANOV, Stoyan. **Primeiros Passos com React: Construindo Aplicações Web**. São Paulo: Novatec, 2016.
- W3SCHOOLS.COM. **React Tutorial**, c1999-2020. Disponível em: <<https://www.w3schools.com/react/default.asp>>. Acesso em: 22 de fev. de 2021.