PROGRAMAÇÃO FRONT END II

Fabiano Berlinck Neumann

SAGAH SOLUÇÕES EDUCACIONAIS INTEGRADAS

Conhecendo o React

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Descrever o React.
- Explicar a função JSX.
- Produzir uma página HTML usando React e JSX.

Introdução

A demanda por profissionais no mercado de tecnologia vem crescendo ano a ano. As universidades nem sempre conseguem suprir as necessidades do mercado na mesma velocidade com que as vagas são abertas para algumas áreas, como gestão de projetos em TI, programação de sistemas *desktop* e Web, banco de dados, segurança da informação, cientistas de dados, programadores de jogos digitais, entre outras.

Desenvolver *software* tem sido cada vez menos complexo, mais popular e acessível, gerando a oportunidade da criação de franquias no Brasil e no exterior, que ensinam lógica, robótica e programação já para crianças.

Também é fato que grande parte das empresas nesse mercado aquecido exige certo nível de conhecimento técnico ou especialização — inclusive com habilidades comportamentais minimamente desenvolvidas — para preenchimento das melhores vagas com os melhores salários, independentemente da área de atuação. Por isso a necessidade do estudo aprofundado das tecnologias e ferramentas, sejam mais recentes ou legadas, além da constante atualização.

Neste capítulo, você estudará conceitos de uma das tecnologias mais recentes para o desenvolvimento de sistemas Web — mais especificamente na parte de *front end* —, conhecida como React, e da função JSX. Além disso, verá exemplos práticos com código utilizado para a produção de páginas HTML que usam React e JSX.

React

Como no caso de qualquer tecnologia ou linguagem a ser estuda pela primeira vez, é essencial buscar pelos conceitos básicos antes de partir para a aplicação ou o desenvolvimento. Essa prática visa a melhorar a curva de aprendizado e conhecer as características envolvidas, além de verificar se essa tecnologia atende de fato às necessidades para o desenvolvimento de um novo produto ou serviço. Com o React, isso não é diferente.

Segundo o *site* oficial, o React é uma biblioteca JavaScript utilizada para criação de interfaces de usuário em aplicações Web. Além disso, a biblioteca é declarativa, fazendo com que o código seja mais previsível e mais simples no que diz respeito à depuração (ou *debugging,* em inglês), que é o processo no qual o desenvolvedor realiza a análise ou testes por meio do acompanhamento do passo a passo da execução do programa, a fins de minimizar os erros (REACT..., 2019a).

Outra característica é que ele é baseado em componentes, sejam eles simples ou com controle de estado, que podem utilizar *plugins* externos para seu desenvolvimento. Além disso, uma vez entendido e utilizado adequadamente, pode ser renderizado tanto no *front end* — que é o lado do cliente ou navegador — quanto no *back end*, que é o lado do servidor.

O fato de o React ser declarativo torna a criação de interfaces de usuário menos complexa, onde podem existir *views* simples para cada estado da aplicação, e o próprio React atualizará e renderizará apenas os componentes que têm relação com os dados alterados ao longo do uso da aplicação. Já o fato de ser baseado em componentes permite ao desenvolvedor criar um componente que pode ser reutilizável em diversas partes da aplicação, além de os encapsular uns dentro de outros, facilitando o desenvolvimento e a posterior manutenção.

De acordo com o guia React, da W3Schools, o React é uma biblioteca JavaScript utilizada no desenvolvimento de interfaces de usuário e na criação de aplicações de página única, que permitem criar componentes de interface de usuário reutilizáveis (REACT..., 2019c). Além disso, para aprender React, é necessário conhecer HTML, CSS, DOM e JavaScript, que também devem ser estudados por quem deseja aprender a biblioteca.

As páginas únicas, ou SPA (do inglês *single page applications*), de acordo com Figueiredo (2018), são o ápice do crescimento do *front end* das aplicações nos últimos anos. SPA é uma técnica usada como base para criação de aplicações que rodam tanto nos navegadores quanto nos dispositivos móveis e *desktop*, tudo em uma única página, mudando apenas parte do conteúdo de acordo com a necessidade, como no caso do Gmail.

Em se tratando de componentes React, segundo o *site* oficial, eles implementam um método conhecido como `render()`, que recebe dados de entrada, gerando os que são exibidos na tela como retorno da função. Além disso, graças às suas propriedades, chamadas com `this.props`, esses dados de entrada podem ser acessados dentro do método `render()`.

Quanto à parte do controle de estados dos componentes, esses podem ser acessados via `this.state`, que controlam o estado dos dados. Quando esses dados do `state` são alterados, o código que havia sido renderizado anteriormente é atualizado por meio de nova invocação do método `render()`, chamado sempre que é necessário para atualizar os dados na tela.

Função JSX

De acordo com o *site* oficial, o React não requer o uso do JSX, ou seja, ele não é obrigatório para o desenvolvimento com a biblioteca. Contudo, apesar disso, muitas pessoas consideram o seu uso prático no sentido da sua ajuda visual quando se trabalha com interface de usuário dentro do trecho de código produzido em JavaScript, além de mostrar mensagens de erros e avisos úteis para o desenvolvedor (REACT..., 2019a).

De acordo com o guia React JSX, da W3Schools, JSX significa JavaScript XML, que permite ao desenvolvedor escrever elementos HTML dentro do trecho JavaScript de um componente React e adicioná-los dentro do DOM, sem a necessidade do uso de métodos como o `createElement()` ou `appendChild()` (REACT..., 2019b). Além disso, o JSX converte as *tags* HTML em elementos React.

Veja, a seguir, um exemplo de código utilizando JSX, com parte dele de uma página HTML e o código de um componente React, bem como o resultado do código apresentado na Figura 1.

Trecho de código da página HTML:

```
<body>
...
    <div id="exemplo-jsx"></div>
...
</body>
Código do compoente React com JSX:
class HelloJSX extends React.Componente {
    render() {
```

```
        return (
            <div>
                Olá {this.props.name}!
            </div>
        );
    }
}
ReactDOM.render(
    <HelloJSX name="JSX" />,
    document.getElementById('exemplo-jsx')
);
```

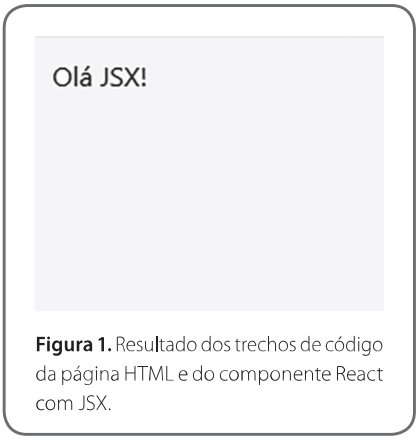

Figura 1. Resultado dos trechos de código da página HTML e do componente React com JSX.

Como pode ser observado no exemplo, o método `ReactDOM.render()` recebe dois parâmetros: o "elemento" React com a propriedade nome de valor JSX e a *tag* HTML que tem o identificador correspondente onde ele será inserido, que é obtida a partir do método `document.getElementById()`. Dessa forma, o método `render()` do componente da classe `HelloJSX` retorna a *tag* `div` dentro do trecho HTML com o texto "`Olá JSX!`", que teve parte substituída a partir da propriedade passada no primeiro parâmetro do método `ReactDOM.render()`: o `name="JSX"`.

A seguir, observe o código do componente React para o mesmo exemplo, que utiliza o mesmo trecho de código HTML e gera exatamente o mesmo resultado apresentado na Figura 1.

```
class HelloJSX extends React.Componente {
    render() {
        return React.createElement(
            "div",
            null,
            "Ol\xE1 ",
            this.props.name,
            "!"
        );
    }
}
ReactDOM.render(
    React.createElement(HelloJSX, {name: "JSX"}),
    document.getElementById('exemplo-jsx')
);
```

Como é possível verificar no método `render()` da classe `HelloJSX`, o código é alterado a partir da palavra `return`, que é equivalente ao código com JSX do exemplo anterior, mas agora utilizando o método `React.createElement()`. Além disso, muda a forma de passagem do primeiro parâmetro do método `ReactDOM.render()`, que agora também recebe o método `React.createElement()`.

Página HTML usando React e JSX

Agora que você já está familiarizado(a) com alguns componentes do React e pôde observar como eles são apresentados dentro de uma *tag* HTML, é hora de analisar um exemplo mais completo em uma página HTML que utiliza o que poderia ser um grande componente, mas está separado em componentes menores.

No exemplo a seguir, apresentado no tutorial da documentação do React, há uma página HTML com a *tag* em que são inseridos os componentes React e o código dos componentes utilizando JSX. Essa página simula a parte do comentário de um fórum, contendo uma imagem, o nome de quem escreveu, o texto de comentário e a data em que foi realizado (adaptado de ADICIONE..., 2019):

```
...
<body>
    <div id="root"></div>
    <!-- Scripts para carregar o React -->
    <script src="https://unpkg.com/react@16/umd/react.develop-
ment.js" crossorigin></script>
    <script src="https://unpkg.com/react-dom@16/umd/react-dom.
development.js" crossorigin></script>
    <!-- Script para carregar os componentes React -->
    <script src="index.js"></script>
</body>
...
```

Nesse trecho do código do arquivo HTML, que pode ser salvo como `index.html`, é criada uma `div` identificada como `root`, para que seja possível acessar a *tag* e inserir os componentes React. Além disso, são inseridas duas *tags* `script` para carregar o React e uma para referenciar e carregar os componentes do arquivo informado dentro do atributo `src` — nesse caso, código do arquivo `index.js`.

```
function formatDate(date) {
    return date.toLocaleDateString();
}
function Avatar(props) {
    return (
        <img
            className="Avatar"
            src={props.user.avatarUrl}
            alt={props.user.name}
        />
    );
}
function UserInfo(props) {
    return (
        <div className="UserInfo">
            <Avatar user={props.user} />
          <div className="UserInfo-name">{props.user.name}</div>
        </div>
    );
```

```
}
function Comment(props) {
    return (
        <div className="Comment">
            <UserInfo user={props.author} />
            <div className="Comment-text">{props.text}</div>
            <div className="Comment-date">
                {formatDate(props.date)}
            </div>
        </div>
    );
}
const comment = {
    date: new Date(),
    text: 'Exemplo de código com React e JSX',
    author: {
        name: 'Professor Sagah',
        avatarUrl: 'https://linkdafotonoservidor.com',
    },
};
ReactDOM.render(
    <Comment
        date={comment.date}
        text={comment.text}
        author={comment.author}
    />,
    document.getElementById('root')
);
```

Quando o método `ReactDOM.render()` é chamado, o componente `Comment` é renderizado com as propriedades `date`, `text` e `author` dentro da *tag* `div`, que contém o identificador `root` do arquivo `index.html`. O valor atribuído às propriedades do componente `Comment`, por sua vez, é carregado a partir da constante `comment`. E, para que aconteça a renderização desse componente, a função propositalmente nomeada `Comment()` é chamada e recebe as propriedades `date`, `text` e `author`, passadas dentro do componente no método `React.render()`.

Para que os elementos de retorno da função `Comment()`, filhas da *tag* `div`, sejam renderizados, é necessário carregar o componente `UserInfo`, que contém a propriedade `user` e seu valor atribuído a partir da propriedade `author` do componente `Comment`. Além disso, um dos elementos HTML retornados pela função `Comment()` tem seu valor atribuído pela propriedade `text` da constante, e o outro realiza a chamada da função `formatDate()`, que recebe o valor da propriedade `date` da constante `comment`.

Por sua vez, para que aconteça a renderização dos elementos e componentes do retorno da função `UserInfo()`, é necessário carregar o componente `Avatar` que, assim como o componente `UserInfo`, tem a propriedade chamada de `user`. Já para carregar o componente `Avatar`, é necessária a função `Avatar()`, que retorna uma imagem por meio da *tag* HTML `img` com seus atributos `src` e `alt`, com as suas respectivas propriedades.

Configurando o ambiente

Uma vez vistos os conceitos e exemplos de código em React, é hora de configurar o ambiente de desenvolvimento para começar a praticar.

Para a sua utilização, deve-se configurar um ambiente React na máquina do desenvolvedor, instalando o npm e o Node.js, simultaneamente à instalação do Node.js, obtido no seu *site* oficial. Uma vez instalados, é possível instalar o React utilizando o seguinte comando no prompt de comando ou no terminal:

```
npm install -g create-react-app
```

Dessa forma, o gerenciador de pacotes instalará o React na máquina do desenvolvedor, permitindo que ele possa rodar o outro comando, que iniciará o projeto na pasta em que o desenvolvedor estiver acessando pelo terminal, apresentado a seguir:

```
npx create-react-app nomedaaplicacao
```

Será, então, criada uma pasta com o projeto, sendo necessário navegar até a pasta do projeto pelo terminal com o comando:

```
cd nomedaaplicacao
```

Feito isso, bastar rodar o seguinte comando para executar a sua aplicação React, que deve abrir o seu navegador no endereço `localhost:3000`. Caso não aconteça, esse endereço pode ser digitado no navegador de preferência do desenvolvedor:

```
npm start
```

Link

Em caso de dúvidas, é possível acessar o guia oficial do React para encontrar documentação, dicas de programação e tutoriais, por meio do *link* a seguir.

https://qrgo.page.link/fGaFe

Referências

ADICIONE o React em Um Minuto. *React/Facebook Open Source*, [*S. l.*], 2019. Disponível em: https://pt-br.reactjs.org/docs/add-react-to-a-website.html. Acesso em: 17 out. 2019.

FIGUEIREDO, E. O que é uma SPA: Single-Page Application? *School of Net*, Indaiatuba, 26 jul. 2018. Disponível em: https://blog.schoolofnet.com/o-que-e-uma-spa-single-page-application/. Acesso em: 17 out. 2019.

REACT – Uma biblioteca JavaScript para criar interfaces de usuário. *React/Facebook Open Source*, [*S. l.*], 2019a. Disponível em: https://pt-br.reactjs.org/. Acesso em: 17 out. 2019.

REACT JSX. *W3Schools*, Sandnes, 2019b. Disponível em: https://www.w3schools.com/react/react_jsx.asp. Acesso em: 17 out. 2019.

REACT Tutorial. *W3Schools*, Sandnes, 2019c. Disponível em: https://www.w3schools.com/react/default.asp. Acesso em: 17 out. 2019.

Leituras recomendadas

JSX In Depth. *React/Facebook Open Source*, [*S. l.*], 2019a. Disponível em: https://pt-br.reactjs.org/docs/jsx-in-depth.html. Acesso em: 17 out. 2019.

REACT. *Font Awesome*, Bentonville; Boston; Joplin; Seattle; Vergennes, [201-?]. Disponível em: https://fontawesome.com/how-to-use/on-the-web/using-with/react. Acesso em: 17 out. 2019.

TUTORIAL: Introdução ao React. *React/Facebook Open Source*, [*S. l.*], 2019a. Disponível em: https://pt-br.reactjs.org/tutorial/tutorial.html. Acesso em: 17 out. 2019.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:

SOLUÇÕES EDUCACIONAIS INTEGRADAS