

1. Atividades Práticas - Componentes e Props

Nesta atividade prática, vamos trabalhar com os conceitos de componentes e props do React.

1.1 Criando nosso Projeto

Vamos criar um projeto chamado my-blog, digitando o seguinte comando no terminal:

```
yarn create react-app my-blog
```

O processo leva alguns segundos e, logo após terminar, digite o seguinte comando para entrar no diretório recém criado do nosso projeto:

```
cd my-blog
```

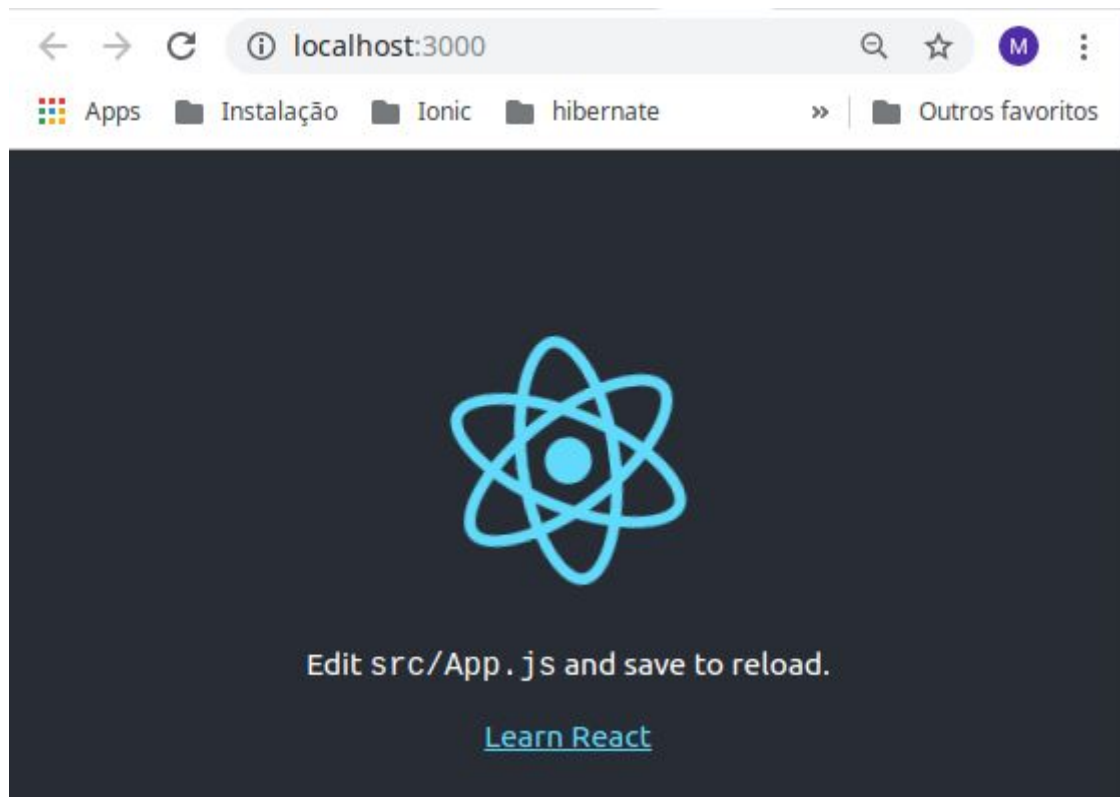
Você pode abrir a aplicação no editor Visual Studio Code. Para isso, dentro do diretório my-blog digite o seguinte comando:

```
code .
```

O Visual Studio Code deverá abrir com a pasta my-blog sendo acessada. Agora, vamos testar nossa aplicação. Digite a seguinte linha de comando:

```
yarn start
```

Ele irá rodar a aplicação em modo desenvolvimento em <http://localhost:3000>



A página será recarregada automaticamente se você fizer alterações no código.

Agora, vamos excluir os seguintes arquivos que são desnecessários para a nossa aplicação: App.js, App.css e App.test.js. Não se preocupe se a aplicação parar de funcionar neste momento.

Em seguida, crie um diretório chamado componentes na pasta src. Dentro do diretório src/componentes, crie um arquivo chamado Post.js. Adicione o seguinte código:

```
import React from 'react';

export default props =>
  <div>
    <img src={props.imagem} />
    <div>
      <h2>{props.autor}</h2>
      <h3>{props.titulo}</h3>
      <hr />
      <p>{props.texto}</p>
    </div>
  </div>
```

Observe que criamos um componente de função, onde vamos retornar dos dados de um post em nosso blog. Lembre que props contém os propriedades que foram definidas por quem chamou este componente. Ainda, devemos nos lembrar que um componente nunca deve modificar seus próprios props (props são somente leitura). Ficou definido que cada post deve mostrar informações referentes a uma imagem, nome do autor, título e texto.

Agora, vamos criar o arquivo Blog.js em src/componentes:

```
import React from 'react';
import Post from './Post';

export default props =>
  <div>
    <Post

imagem="https://images.pexels.com/photos/3880017/pexels-photo-3880017.j
peg"

    autor="Márcio Santos"
    titulo="Escalando..."
    texto="Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Ut faucibus semper ex, et sagittis mauris lobortis eu. Nulla ex
ipsum, varius a mattis quis, ultricies non leo. Fusce aliquam lorem sit
amet ullamcorper pulvinar."
  />
  <Post

imagem="https://images.pexels.com/photos/2748019/pexels-photo-2748019.j
peg"

    autor="Rosana Marques"
    titulo="Lindo passeio"
    texto="Donec elit turpis, luctus at orci et, commodo laoreet
tortor. Donec et quam euismod, mollis mauris id, congue ipsum."
  />
  <Post

imagem="https://images.pexels.com/photos/3722888/pexels-photo-3722888.j
peg"

    autor="Beatriz Gomes"
    titulo="Muito Romântico"
```

```
        texto="Quisque eros ligula, tempus id ultrices sed,  
ultricies sed lorem. Maecenas turpis lectus, finibus a semper ac,  
vulputate id dui."  
    />  
</div>
```

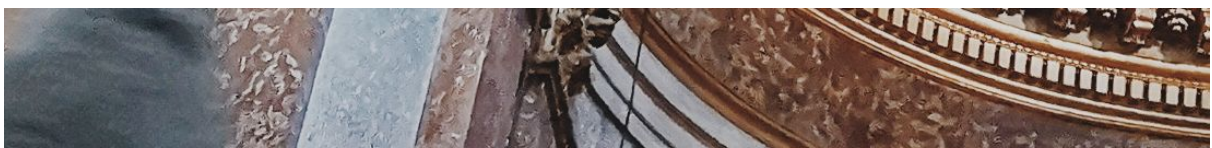
Observe que dentro do componente Blog, instanciamos 3 componentes Post, passando as propriedades imagem, autor, título e texto para este componente.

No index.js precisamos chamar este componente Blog.js:

```
import React from 'react';  
import ReactDOM from 'react-dom';  
import Blog from './componentes/Blog';  
  
const elemento = document.getElementById('root');  
ReactDOM.render(  
  <Blog />  
  , elemento );
```

Veja que renderizamos a tela, com o ReactDOM.render(), e instanciamos o componente Blog.

Veja o resultado no navegador.



Beatriz Gomes

Muito Romântico

Quisque eros ligula, tempus id ultrices sed, ultricies sed lorem. Maecenas turpis lectus, finibus a semper ac, vulputate id dui.

Observe que as imagens estão muito grandes, mas o blog está funcionando.

Agora, vamos ajustar o estilo desta aplicação usando CSS. CSS é a sigla para o termo em inglês Cascading Style Sheets que, traduzido para o português, significa Folha de Estilo em Cascatas. Ele é usado para estilizar elementos escritos em uma linguagem de marcação, tal como o HTML. Assim, o CSS separa o conteúdo da representação visual do site.

Uma forma de aplicar o CSS ao HTML é através da criação de classes CSS. Para isto, no documento CSS, basta colocar `.nome-da-classe-que-deseja`. Abaixo criamos a classe `título`:

```
.título{
  font-size: 3rem; /*tamanho da fonte do título*/
  color: #333; /*esta é a cor do texto*/
  text-align: center; /*o alinhamento do texto que carregar esta classe*/
}
```

Note que existe um ponto (.) antes do nome da classe. Isso faz com que o CSS entenda que se está criando uma classe para ser usada no HTML.

Dentro de `scr/componentes`, crie um arquivo chamado `Blog.css` com o seguinte estilo:

```
@import
url('https://fonts.googleapis.com/css?family=Roboto:400,700&display=swap');

* {
  box-sizing: border-box;
}

.post {
  font-family: 'Roboto', Arial, Helvetica, sans-serif;
  border-radius: 10px;
  background-color: #f3f3f3;
  padding: 25px;
  display: flex;
  max-width: 960px;
  margin: 50px auto;
}

.post-imagem {
  width: 200px;
  max-width: 300px;
  margin-right: 50px;
  border-radius: 10px;
}

.post-autor {
  margin: 0;
  font-size: 24px;
}
```

```

}

.post-titulo {
  margin: 0;
  margin-bottom: 15px;
  font-weight: normal;
  font-size: 20px;
}

.post hr {
  max-width: 200px;
  margin-left: 0;
}

```

Para adicionar classe de CSS aos componentes do React, passe a string para a prop `className`, por exemplo:

```
<h1 className="titulo">Menu</h1>
```

Dessa forma, em seguida, vamos aplicar este estilo definido no componente `Post.js`:


```

import React from 'react';
import './Blog.css';

export default props =>
  <div className="post">
    <img alt="" src={props.imagem} className="post-imagem" />
    <div>
      <h2 className="post-autor">{props.autor}</h2>
      <h3 className="post-titulo">{props.titulo}</h3>
      <hr/>
      <p>{props.texto}</p>
    </div>
  </div>


```

Veja o resultado no navegador:




Márcio Santos
Escalando...

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut faucibus semper ex, et sagittis mauris lobortis eu. Nulla ex ipsum, varius a mattis quis, ultricies non leo. Fusce aliquam lorem sit amet ullamcorper pulvinar.



Rosana Marques
Lindo passeio

Donec elit turpis, luctus at orci et, commodo laoreet tortor. Donec et quam euismod, mollis mauris id, congue ipsum.



Beatriz Gomes
Muito Romântico

Quisque eros ligula, tempus id ultrices sed, ultricies sed lorem. Maecenas turpis lectus, finibus a semper ac, vulputate id dui.

Funcionou!!!

Agora, vamos modificar nossa aplicação. Vamos considerar que temos um array com todos os 3 posts de nosso blog vindos de uma fonte externa, por exemplo. Neste caso, modifique index.js para conter este array:

```
import React from 'react';
import ReactDOM from 'react-dom';
import Blog from '../componentes/Blog';

const myPosts = [
  {
    imagem:
      "https://images.pexels.com/photos/3880017/pexels-photo-3880017.jpeg",
    autor: "Márcio Santos",
    titulo: "Escalando...",
    texto: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut faucibus semper ex, et sagittis mauris lobortis eu. Nulla ex ipsum, varius a mattis quis, ultricies non leo. Fusce aliquam lorem sit amet ullamcorper pulvinar."
  },
  {
    imagem:
      "https://images.pexels.com/photos/2748019/pexels-photo-2748019.jpeg",
    autor: "Rosana Marques",
    titulo: "Lindo passeio",
```

```

      texto: "Donec elit turpis, luctus at orci et, commodo laoreet
tortor. Donec et quam euismod, mollis mauris id, congue ipsum."
    },
    {
      imagem:
"https://images.pexels.com/photos/3722888/pexels-photo-3722888.jpeg",
      autor: "Beatriz Gomes",
      titulo: "Muito Romântico",
      texto: "Quisque eros ligula, tempus id ultrices sed, ultricies sed
lorem. Maecenas turpis lectus, finibus a semper ac, vulputate id dui."
    }
  ]

const elemento = document.getElementById('root');
ReactDOM.render(
  <Blog posts={myPosts}/>
  , elemento );

```

Veja que criamos um array de objetos chamado myPost. Este array é passado para o componente Blog através da props chamada posts.

Logo, em Blog.js, vamos alterar o código da seguinte maneira:

```

import React from 'react';
import Post from './Post';

export default props =>
  <div>
    {
      props.posts.map(post =>
        <Post
          imagem={post.imagem}
          autor={post.autor}
          titulo={post.titulo}
          texto={post.texto}
        />
      )
    }
  </div>

```


Veja que agora, ao invés de termos 3 posts fixos no nosso blog, usamos o método `map()` do Javascript. Cabe lembrar que o método `map()` chama uma função *callback* recebida por parâmetro para cada elemento do Array original, em ordem, e constrói um novo array com base nos retornos de cada chamada.

Veja no navegador que esta estratégia funcionou!



Márcio Santos

Escalando...

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut faucibus semper ex, et sagittis mauris lobortis eu. Nulla ex ipsum, varius a mattis quis, ultricies non leo. Fusce aliquam lorem sit amet ullamcorper pulvinar.



Rosana Marques

Lindo passeio

Donec elit turpis, luctus at orci et, commodo laoreet tortor. Donec et quam euismod, mollis mauris id, congue ipsum.



Beatriz Gomes

Muito Romântico

Quisque eros ligula, tempus id ultrices sed, ultricies sed lorem. Maecenas turpis lectus, finibus a semper ac, vulputate id dui.