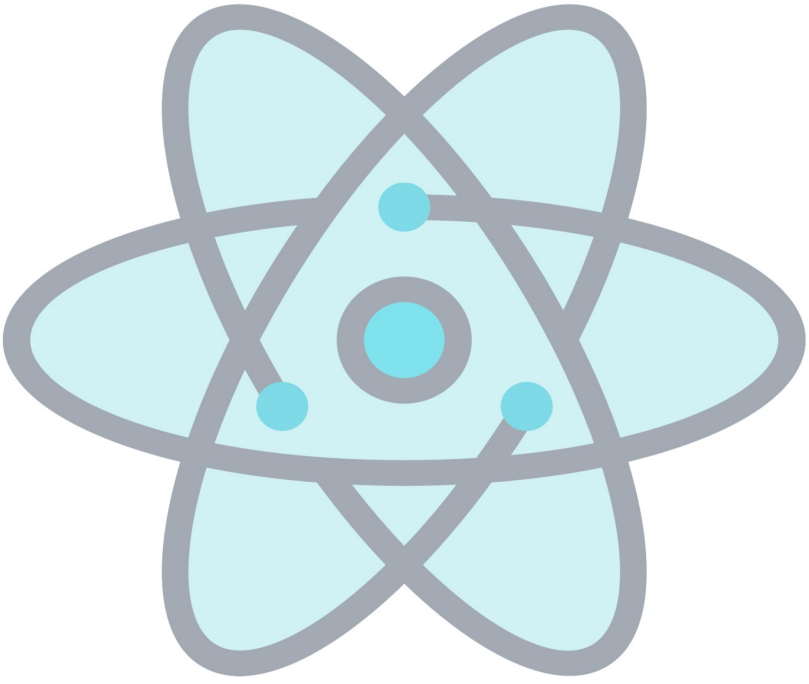
Códigos do Curso React para construção de Web Apps

Turma react8632



caelum

ensino e inovação

Sumário

Capítulo 1

EXERCÍCIO: O TWITTELUM COM CÓDIGO DE VIEW DECLARATIVO USANDO REACT PURO.

1.1 PASSO A PASSO COM CÓDIGO

1. Crie o arquivo **`twittelumReact.js`** na pasta **`js`** com o seguinte código:

 # js/twittelumReact.js

```
+const tweetsArea = document.querySelector('.tweetsArea')
+
+function criaTweet(conteudo) {
+    return React.createElement('article', {className: 'tweet'}, [
+        React.createElement('p', {className: 'tweet__conteudo'}, [
+            conteudo
+        ])
+    ])
+}
+
+const listaTweets = [
+    criaTweet('oi'),
+    criaTweet('tchau')
+]
+
+ReactDOM.render(listaTweets, tweetsArea)
```

2. No arquivo **`index.html`** na pasta **`raíz do projeto`** faça as seguintes alterações:

 # index.html

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="css/reset.css">
    <link rel="stylesheet" href="css/container.css">
    <link rel="stylesheet" href="css/btn.css">
    <link rel="stylesheet" href="css/icon.css">
    <link rel="stylesheet" href="css/iconHeart.css">
    <link rel="stylesheet" href="css/notificacao.css">

    <link rel="stylesheet" href="css/cabecalho.css">
```

```
        <link rel="stylesheet" href="css/navMenu.css">
        <link rel="stylesheet" href="css/dashboard.css">
        <link rel="stylesheet" href="css/widget.css">
        <link rel="stylesheet" href="css/novoTweet.css">
        <link rel="stylesheet" href="css/trendsArea.css">
        <link rel="stylesheet" href="css/tweet.css">

        <title> Twittelum </title>
    </head>

    <body>
        <header class="cabecalho">
            <div class="cabecalho__container container">
                <h1 class="cabecalho__logo">
                    <a href="/">Twitelum</a>
                </h1>
                <nav class=navMenu>
                    <ul class=navMenu__lista>
                        <li class=navMenu__item>
                            <a class=navMenu__link href="/">
                                Bem vindo(a): <br />
                                <strong> @alumna</strong>
                            </a>
                        </li>
                        <li class=navMenu__item>
                            <a class=navMenu__link href="/">
                                Página Inicial
                            </a>
                        </li>
                        <li class=navMenu__item>
                            <a class=navMenu__link href="/hashtags">
                                Hashtags
                            </a>
                        </li>
                        <li class=navMenu__item>
                            <a class=navMenu__link href="/logout">
                                Logout
                            </a>
                        </li>
                    </ul>
                </nav>
            </div>
        </header>

        <div class="container">

            <div class="dashboard">
                <div class="widget">
                    <form class="novoTweet">
                        <div class="novoTweet__editorArea">
                            <span class="novoTweet__status">0/140</span>
                            <textarea class="novoTweet__editor" placeholder="O que está acontecendo?
></textarea>
                        </div>
                        <button type="submit" class="novoTweet__envia">Tweetar</button>
                    </form>
                </div>
                <div class="widget">
                    <div class="trendsArea">
                        <h2 class="trendsArea__titulo widget__titulo">Trends Brasil</h2>
                        <ol class="trendsArea__lista">
                            <li><a href="/">#react</a></li>
                            <li><a href="/">#reactHooks</a></li>
```

```
                    </ol>
                </div>
            </div>
        </div>

        <div class="dashboard dashboard__centro">
            <div class="widget">
                <div class="tweetsArea">
                </div>
            </div>
        </div>

    </div>

-    <script src="js/twittelumDOMPuro.js"></script>
+    <script src="js/lib/react.js"></script>
+    <script src="js/lib/react-dom.js"></script>
+
+    <script src="js/twittelumReact.js"></script>
 </body>

 </html>
```

Capítulo 2

EXERCÍCIO: ORGANIZAÇÃO DO PROJETO.

2.1 PASSO A PASSO COM CÓDIGO

1. Crie o arquivo **`app.js`** na pasta **`js`** com o seguinte código:

   # js/app.js

   ```
   +import { Tweet } from './components/Tweet.js'
   +
   +const tweetsArea = document.querySelector('.tweetsArea')
   +
   +const listaTweets = [
   +    Tweet('oi'),
   +    Tweet('tchau')
   +]
   +ReactDOM.render(listaTweets, tweetsArea)
   ```

2. Crie o arquivo **`Tweet.js`** na pasta **`js/components`** com o seguinte código:

   # js/components/Tweet.js

   ```
   +
   +// Componente Tweet
   +export function Tweet(conteudo) {
   +    return React.createElement('article', {className: 'tweet'}, [
   +        React.createElement('div', {class: 'tweet__cabecalho'}, [
   +            "CABEACALHO TWEET"
   +        ]),
   +        React.createElement('p', {className: 'tweet__conteudo'}, [
   +            conteudo
   +        ]),
   +        React.createElement('footer', {class: 'tweet__footer'}, [
   +            "FOOTER TWEET"
   +        ]),
   +    ])
   +}
   ```

3. Remova o arquivo **`twittelumReact.js`** na pasta **`js`** .

4. No arquivo **`index.html`** na pasta **`raíz do projeto`** faça as seguintes alterações:

   # index.html

   ```
   <!DOCTYPE html>
   <html lang="pt-br">
   ```

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="css/reset.css">
    <link rel="stylesheet" href="css/container.css">
    <link rel="stylesheet" href="css/btn.css">
    <link rel="stylesheet" href="css/icon.css">
    <link rel="stylesheet" href="css/iconHeart.css">
    <link rel="stylesheet" href="css/notificacao.css">

    <link rel="stylesheet" href="css/cabecalho.css">
    <link rel="stylesheet" href="css/navMenu.css">
    <link rel="stylesheet" href="css/dashboard.css">
    <link rel="stylesheet" href="css/widget.css">
    <link rel="stylesheet" href="css/novoTweet.css">
    <link rel="stylesheet" href="css/trendsArea.css">
    <link rel="stylesheet" href="css/tweet.css">

    <title> Twittelum </title>
</head>

<body>
    <header class="cabecalho">
        <div class="cabecalho__container container">
            <h1 class="cabecalho__logo">
                <a href="/">Twitelum</a>
            </h1>
            <nav class=navMenu>
                <ul class=navMenu__lista>
                    <li class=navMenu__item>
                        <a class=navMenu__link href="/">
                            Bem vindo(a): <br />
                            <strong> @alumna</strong>
                        </a>
                    </li>
                    <li class=navMenu__item>
                        <a class=navMenu__link href="/">
                            Página Inicial
                        </a>
                    </li>
                    <li class=navMenu__item>
                        <a class=navMenu__link href="/hashtags">
                            Hashtags
                        </a>
                    </li>
                    <li class=navMenu__item>
                        <a class=navMenu__link href="/logout">
                            Logout
                        </a>
                    </li>
                </ul>
            </nav>
        </div>
    </header>

    <div class="container">

        <div class="dashboard">
            <div class="widget">
                <form class="novoTweet">
                    <div class="novoTweet__editorArea">
```

```
                        <span class="novoTweet__status">0/140</span>
                        <textarea class="novoTweet__editor" placeholder="O que está acontecendo?
></textarea>
                    </div>
                    <button type="submit" class="novoTweet__envia">Tweetar</button>
                </form>
            </div>
            <div class="widget">
                <div class="trendsArea">
                    <h2 class="trendsArea__titulo widget__titulo">Trends Brasil</h2>
                    <ol class="trendsArea__lista">
                        <li><a href="/">#react</a></li>
                        <li><a href="/">#reactHooks</a></li>
                    </ol>
                </div>
            </div>
        </div>

        <div class="dashboard dashboard__centro">
            <div class="widget">
                <div class="tweetsArea">
                </div>
            </div>
        </div>

    </div>

    <script src="js/lib/react.js"></script>
    <script src="js/lib/react-dom.js"></script>

-   <script src="js/twittelumReact.js"></script>
+
+   <script type="module" src="js/app.js"></script>
 </body>

 </html>
```

CAPÍTULO 3

EXERCÍCIO: CRIANDO ELEMENTOS REACT COM JSX E A COMPILAÇÃO COM BABEL EM TEMPO DE EXECUÇÃO.

3.1 PASSO A PASSO COM CÓDIGO

1. No arquivo **`app.js`** na pasta **`js`** faça as seguintes alterações:

 # js/app.js

```
 import { Tweet } from './components/Tweet.js'

 const tweetsArea = document.querySelector('.tweetsArea')

 const listaTweets = [
     Tweet('oi'),
     Tweet('tchau')
 ]
+
 ReactDOM.render(listaTweets, tweetsArea)
```

2. No arquivo **`Tweet.js`** na pasta **`js/components`** faça as seguintes alterações:

 # js/components/Tweet.js

```
-
 // Componente Tweet
 export function Tweet(conteudo) {
-    return React.createElement('article', {className: 'tweet'}, [
-        React.createElement('div', {class: 'tweet__cabecalho'}, [
-            "CABEACALHO TWEET"
-        ]),
-        React.createElement('p', {className: 'tweet__conteudo'}, [
-            conteudo
-        ]),
-        React.createElement('footer', {class: 'tweet__footer'}, [
-            "FOOTER TWEET"
-        ]),
-    ])
+    return <article class="tweet">
+        <div class="tweet__cabecalho">
+            <img class="tweet__fotoUsuario" src="https://placehold.it/50x50" alt="" />
+            <span class="tweet__nomeUsuario">Fulano de Tal</span>
+            <a href="/"><span class="tweet__userName">@usuario</span></a>
+        </div>
+        <p class="tweet__conteudo"><span>Lorem, ipsum dolor sit <a href="/trends/#amet" data-rea
```

```
ctroot="">#amet</a> consectetur adipisicing <a href="/trends/#elit" data-reactroot="">#elit</a>. A
dipisci ut cumque tempora? Quam velit vitae voluptatum tempora iste, mollitia, sa</span></p>
 +        <footer class="tweet__footer">
 +            <button class="btn btn--clean">
 +                <svg class="icon icon--small iconHeart" xmlns="http://www.w3.org/2000/svg" viewB
ox="0 0 47.5 47.5">
 +                    <defs>
 +                        <clipPath id="a">
 +                            <path d="M0 38h38V0H0v38z"></path>
 +                        </clipPath>
 +                    </defs>
 +                    <g clipPath="url(#a)" transform="matrix(1.25 0 0 -1.25 0 47.5)">
 +                        <path d="M36.885 25.166c0 5.45-4.418 9.868-9.867 9.868-3.308 0-6.227-1.6
32-8.018-4.128-1.79 2.496-4.71 4.129-8.017 4.129-5.45 0-9.868-4.418-9.868-9.868 0-.773.098-1.52.26
6-2.242C2.75 14.413 12.216 5.431 19 2.965c6.783 2.466 16.249 11.448 17.617 19.96.17.721.268 1.47.2
68 2.241"></path>
 +                    </g>
 +                </svg>
 +                0
 +            </button>
 +        </footer>
 +    </article>
  }
```

3. No arquivo **`index.html`** na pasta **`raíz do projeto`** faça as seguintes alterações:

 # index.html

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="css/reset.css">
    <link rel="stylesheet" href="css/container.css">
    <link rel="stylesheet" href="css/btn.css">
    <link rel="stylesheet" href="css/icon.css">
    <link rel="stylesheet" href="css/iconHeart.css">
    <link rel="stylesheet" href="css/notificacao.css">

    <link rel="stylesheet" href="css/cabecalho.css">
    <link rel="stylesheet" href="css/navMenu.css">
    <link rel="stylesheet" href="css/dashboard.css">
    <link rel="stylesheet" href="css/widget.css">
    <link rel="stylesheet" href="css/novoTweet.css">
    <link rel="stylesheet" href="css/trendsArea.css">
    <link rel="stylesheet" href="css/tweet.css">

    <title> Twittelum </title>
</head>

<body>
    <header class="cabecalho">
        <div class="cabecalho__container container">
            <h1 class="cabecalho__logo">
                <a href="/">Twitelum</a>
            </h1>
            <nav class=navMenu>
                <ul class=navMenu__lista>
                    <li class=navMenu__item>
```

```
                        <a class=navMenu__link href="/">
                            Bem vindo(a): <br />
                            <strong> @alumna</strong>
                        </a>
                    </li>
                    <li class=navMenu__item>
                        <a class=navMenu__link href="/">
                            Página Inicial
                        </a>
                    </li>
                    <li class=navMenu__item>
                        <a class=navMenu__link href="/hashtags">
                            Hashtags
                        </a>
                    </li>
                    <li class=navMenu__item>
                        <a class=navMenu__link href="/logout">
                            Logout
                        </a>
                    </li>
                </ul>
            </nav>
        </div>
    </header>

    <div class="container">

        <div class="dashboard">
            <div class="widget">
                <form class="novoTweet">
                    <div class="novoTweet__editorArea">
                        <span class="novoTweet__status">0/140</span>
                        <textarea class="novoTweet__editor" placeholder="O que está acontecendo?
></textarea>
                    </div>
                    <button type="submit" class="novoTweet__envia">Tweetar</button>
                </form>
            </div>
            <div class="widget">
                <div class="trendsArea">
                    <h2 class="trendsArea__titulo widget__titulo">Trends Brasil</h2>
                    <ol class="trendsArea__lista">
                        <li><a href="/">#react</a></li>
                        <li><a href="/">#reactHooks</a></li>
                    </ol>
                </div>
            </div>
        </div>

        <div class="dashboard dashboard__centro">
            <div class="widget">
                <div class="tweetsArea">
                </div>
            </div>
        </div>

    </div>

    <script src="js/lib/react.js"></script>
    <script src="js/lib/react-dom.js"></script>

+   <script src="js/lib/babel.js"></script>
```

```
 -    <script type="module" src="js/app.js"></script>
 +
 +    <script src="js/components/Tweet.js" data-plugins="transform-es2015-modules-umd" data-preset
="es2015, react"></script>
 +    <script src="js/app.js" data-plugins="transform-es2015-modules-umd" data-presets="es2015, re
act"></script>
  </body>

  </html>
```

CAPÍTULO 4

EXERCÍCIO: O PROCESSO DE BUILD – COMPILANDO MUITO MAIS QUE JSX.

4.1 PASSO A PASSO COM CÓDIGO

1. Crie o arquivo **`.env`** na pasta **`raíz do projeto`** com o seguinte código:

   # .env

   ```
   +CHOKIDAR_USEPOLLING=true
   ```

2. Mova o arquivo **`twittelumDOMPuro.js`** da pasta **`js`** para a pasta **`raíz do projeto`** e o renomeie para **`__twittelumDOMPuro.js`** .

3. Remova o arquivo **`Tweet.js`** na pasta **`js/components`** .

4. Crie o arquivo **`package.json`** na pasta **`raíz do projeto`** com o seguinte código:

   # package.json

   ```
   +{
   +  "name": "twittelum",
   +  "version": "0.1.0",
   +  "private": true,
   +  "dependencies": {
   +    "react": "^16.12.0",
   +    "react-dom": "^16.12.0",
   +    "react-scripts": "^3.3.0"
   +  },
   +  "scripts": {
   +    "start": "react-scripts start",
   +    "build": "react-scripts build"
   +  },
   +  "browserslist": {
   +    "production": [
   +      ">0.2%",
   +      "not dead",
   +      "not op_mini all"
   +    ],
   +    "development": [
   +      "last 1 chrome version",
   +      "last 1 firefox version",
   +      "last 1 safari version"
   +    ]
   +  }
   +}
   ```

5. Crie o arquivo **`Tweet.jsx`** na pasta **`src/components/Tweet`** com o seguinte código:

 # src/components/Tweet/Tweet.jsx

```
+import React from 'react'
+
+import './tweet.css'
+
+// Componente Tweet
+export function Tweet(conteudo) {
+    return (
+        <article className="tweet">
+            <div className="tweet__cabecalho">
+                <img className="tweet__fotoUsuario" src="https://placehold.it/50x50" alt="" />
+                <span className="tweet__nomeUsuario">Fulano de Tal</span>
+                <a href="/"><span className="tweet__userName">@usuario</span></a>
+            </div>
+            <p className="tweet__conteudo">{ conteudo }</p>
+            <footer className="tweet__footer">
+                <button className="btn btn--clean">
+                    <svg className="icon icon--small iconHeart" xmlns="http://www.w3.org/2000/sv
g" viewBox="0 0 47.5 47.5">
+                        <defs>
+                            <clipPath id="a">
+                                <path d="M0 38h38V0H0v38z"></path>
+                            </clipPath>
+                        </defs>
+                        <g clipPath="url(#a)" transform="matrix(1.25 0 0 -1.25 0 47.5)">
+                            <path d="M36.885 25.166c0 5.45-4.418 9.868-9.867 9.868-3.308 0-6.227
-1.632-8.018-4.128-1.79 2.496-4.71 4.129-8.017 4.129-5.45 0-9.868-4.418-9.868-9.868 0-.773.098-1.5
2.266-2.242C2.75 14.413 12.216 5.431 19 2.965c6.783 2.466 16.249 11.448 17.617 19.96.17.721.268 1.
47.268 2.241"></path>
+                        </g>
+                    </svg>
+                    0
+                </button>
+            </footer>
+        </article>
+    )
+}
```

6. Mova o arquivo **`app.js`** para a pasta **`src`** e o renomeie para **`index.js`** . No momento o arquivo está na pasta **`js`** . Faça também as seguintes modificações no código:

 # src/index.js

```
-import { Tweet } from './components/Tweet.js'
+import ReactDOM from 'react-dom'
+import { Tweet } from './components/Tweet/Tweet'

 const tweetsArea = document.querySelector('.tweetsArea')

 const listaTweets = [
     Tweet('oi'),
     Tweet('tchau')
 ]

 ReactDOM.render(listaTweets, tweetsArea)
```

7. No arquivo **.gitignore** na pasta **raíz do projeto** faça as seguintes alterações:

 # .gitignore

```
  .DS_Store
 +node_modules
```

8. Mova o arquivo **btn.css** da pasta **css** para a pasta **public/css** .
9. Mova o arquivo **cabecalho.css** da pasta **css** para a pasta **public/css** .
10. Mova o arquivo **container.css** da pasta **css** para a pasta **public/css** .
11. Mova o arquivo **dashboard.css** da pasta **css** para a pasta **public/css** .
12. Mova o arquivo **icon.css** da pasta **css** para a pasta **public/css** .
13. Mova o arquivo **iconHeart.css** da pasta **css** para a pasta **public/css** .
14. Mova o arquivo **navMenu.css** da pasta **css** para a pasta **public/css** .
15. Mova o arquivo **notificacao.css** da pasta **css** para a pasta **public/css** .
16. Mova o arquivo **novoTweet.css** da pasta **css** para a pasta **public/css** .
17. Mova o arquivo **reset.css** da pasta **css** para a pasta **public/css** .
18. Mova o arquivo **trendsArea.css** da pasta **css** para a pasta **public/css** .
19. Mova o arquivo **widget.css** da pasta **css** para a pasta **public/css** .
20. Mova o arquivo **index.html** para a pasta **public** . No momento este arquivo está na pasta **raíz do projeto** . Faça também as seguintes modificações no código:

 # public/index.html

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="css/reset.css">
    <link rel="stylesheet" href="css/container.css">
    <link rel="stylesheet" href="css/btn.css">
    <link rel="stylesheet" href="css/icon.css">
    <link rel="stylesheet" href="css/iconHeart.css">
    <link rel="stylesheet" href="css/notificacao.css">

    <link rel="stylesheet" href="css/cabecalho.css">
    <link rel="stylesheet" href="css/navMenu.css">
    <link rel="stylesheet" href="css/dashboard.css">
    <link rel="stylesheet" href="css/widget.css">
```

```
    <link rel="stylesheet" href="css/novoTweet.css">
    <link rel="stylesheet" href="css/trendsArea.css">
-   <link rel="stylesheet" href="css/tweet.css">
+   <!-- <link rel="stylesheet" href="css/tweet.css"> -->

    <title> Twittelum </title>
</head>

<body>
    <header class="cabecalho">
        <div class="cabecalho__container container">
            <h1 class="cabecalho__logo">
                <a href="/">Twitelum</a>
            </h1>
            <nav class=navMenu>
                <ul class=navMenu__lista>
                    <li class=navMenu__item>
                        <a class=navMenu__link href="/">
                            Bem vindo(a): <br />
                            <strong> @alumna</strong>
                        </a>
                    </li>
                    <li class=navMenu__item>
                        <a class=navMenu__link href="/">
                            Página Inicial
                        </a>
                    </li>
                    <li class=navMenu__item>
                        <a class=navMenu__link href="/hashtags">
                            Hashtags
                        </a>
                    </li>
                    <li class=navMenu__item>
                        <a class=navMenu__link href="/logout">
                            Logout
                        </a>
                    </li>
                </ul>
            </nav>
        </div>
    </header>

    <div class="container">

        <div class="dashboard">
            <div class="widget">
                <form class="novoTweet">
                    <div class="novoTweet__editorArea">
                        <span class="novoTweet__status">0/140</span>
                        <textarea class="novoTweet__editor" placeholder="O que está acontecendo?"
</textarea>
                    </div>
                    <button type="submit" class="novoTweet__envia">Tweetar</button>
                </form>
            </div>
            <div class="widget">
                <div class="trendsArea">
                    <h2 class="trendsArea__titulo widget__titulo">Trends Brasil</h2>
                    <ol class="trendsArea__lista">
                        <li><a href="/">#react</a></li>
                        <li><a href="/">#reactHooks</a></li>
                    </ol>
                </div>
```

```
            </div>
        </div>

        <div class="dashboard dashboard__centro">
            <div class="widget">
                <div class="tweetsArea">
+
                </div>
            </div>
        </div>

    </div>

-    <script src="js/lib/react.js"></script>
-    <script src="js/lib/react-dom.js"></script>
-
-    <script src="js/lib/babel.js"></script>
-
+    <!-- <script src="js/lib/react.js"></script>
+    <script src="js/lib/react-dom.js"></script> -->

-    <script src="js/components/Tweet.js" data-plugins="transform-es2015-modules-umd" data-presets
"es2015, react"></script>
-    <script src="js/app.js" data-plugins="transform-es2015-modules-umd" data-presets="es2015, rea
t"></script>
+    <!-- <script src="js/app.js" ></script> -->
 </body>

 </html>
```

21. Mova o arquivo **`tweet.css`** da pasta **`css`** para a pasta **`src/components/Tweet`** .

22. Há 3 arquivos a serem removidos da pasta js. Note que esses arquivos foram disponibilizados já prontos para você.

    Os 3 arquivos que devem ser removidos se encontram na seguinte estrutura de pastas:

    # arquivos removidos do projeto

```
└─ pasta_raiz_do_projeto
   └─ js
      └─ lib
         ├─ babel.js
         ├─ react-dom.js
         └─ react.js
```

CAPÍTULO 5

EXERCÍCIO: A PÁGINA INTEIRA SENDO CONTROLADA PELO REACT.

5.1 PASSO A PASSO COM CÓDIGO

1. Remova o arquivo **`__twittelumDOMPuro.js`** na pasta **`raíz do projeto`** .

2. No arquivo **`package.json`** na pasta **`raíz do projeto`** faça as seguintes alterações:

 # package.json

```
  {
    "name": "twittelum",
    "version": "0.1.0",
    "private": true,
    "dependencies": {
      "react": "^16.12.0",
      "react-dom": "^16.12.0",
      "react-scripts": "^3.3.0"
    },
    "scripts": {
      "start": "react-scripts start",
-     "build": "react-scripts build"
+     "_build:react": "react-scripts build",
+     "build": "npm run _build:react"
    },
    "browserslist": {
      "production": [
        ">0.2%",
        "not dead",
        "not op_mini all"
      ],
      "development": [
        "last 1 chrome version",
        "last 1 firefox version",
        "last 1 safari version"
      ]
    }
  }
```

3. No arquivo **`Tweet.jsx`** na pasta **`src/components/Tweet`** faça as seguintes alterações:

 # src/components/Tweet/Tweet.jsx

```
  import React from 'react'

  import './tweet.css'
```

```
-// Componente Tweet
-export function Tweet(conteudo) {
+function TweetCabecalho({ nomeCompletoUsuario, nomeUsuario }) {
     return (
-        <article className="tweet">
         <div className="tweet__cabecalho">
             <img className="tweet__fotoUsuario" src="https://placehold.it/50x50" alt="" />
-                <span className="tweet__nomeUsuario">Fulano de Tal</span>
-                <a href="/"><span className="tweet__userName">@usuario</span></a>
+            <span className="tweet__nomeUsuario">{ nomeCompletoUsuario }</span>
+            <a href="/"><span className="tweet__userName">@{ nomeUsuario }</span></a>
         </div>
-            <p className="tweet__conteudo">{ conteudo }</p>
+    )
+}
+
+function TweetFooter({ qtLikes }) {
+    return (
         <footer className="tweet__footer">
             <button className="btn btn--clean">
                 <svg className="icon icon--small iconHeart" xmlns="http://www.w3.org/2000/svg" v
iewBox="0 0 47.5 47.5">
                     <defs>
                         <clipPath id="a">
                             <path d="M0 38h38V0H0v38z"></path>
                         </clipPath>
                     </defs>
                     <g clipPath="url(#a)" transform="matrix(1.25 0 0 -1.25 0 47.5)">
                         <path d="M36.885 25.166c0 5.45-4.418 9.868-9.867 9.868-3.308 0-6.227-1.6
32-8.018-4.128-1.79 2.496-4.71 4.129-8.017 4.129-5.45 0-9.868-4.418-9.868-9.868 0-.773.098-1.52.26
6-2.242C2.75 14.413 12.216 5.431 19 2.965c6.783 2.466 16.249 11.448 17.617 19.96.17.721.268 1.47.2
68 2.241"></path>
                     </g>
                 </svg>
-                    0
+                { qtLikes }
             </button>
         </footer>
+    )
+}
+
+// Componente Tweet
+export function Tweet(props) {
+
+    const conteudo = props.children
+
+    return (
+        <article className="tweet">
+            <TweetCabecalho
+                nomeCompletoUsuario={props.nomeCompletoUsuario}
+                nomeUsuario={props.nomeUsuario}
+            />
+
+            <p className="tweet__conteudo">{ conteudo }</p>
+
+            <TweetFooter qtLikes={props.qtLikes} />
+
         </article>
     )
 }
```

4. No arquivo **`index.js`** na pasta **`src`** faça as seguintes alterações:

 # src/index.js

```
+import React from 'react'
 import ReactDOM from 'react-dom'
-import { Tweet } from './components/Tweet/Tweet'

-const tweetsArea = document.querySelector('.tweetsArea')
+import { Home } from './pages/Home/Home.jsx'

-const listaTweets = [
-    Tweet('oi'),
-    Tweet('tchau')
-]
-
-ReactDOM.render(listaTweets, tweetsArea)
+ReactDOM.render(
+    <Home />,
+    document.querySelector('body')
+)
```

5. Crie o arquivo **`Home.jsx`** na pasta **`src/pages/Home`** com o seguinte código:

 # src/pages/Home/Home.jsx

```
+import React from 'react'
+import { Tweet } from '../../components/Tweet/Tweet.jsx'
+
+export function Home() {
+    const infoTweet1 = {
+        conteudo: 'oi',
+        nomeCompletoUsuario: "Artur Diniz",
+        nomeUsuario: "artdiniz",
+        qtLikes: 0
+    }
+
+    const infoTweet2 = {
+        conteudo: 'tchau',
+        nomeCompletoUsuario: "Artur Adam",
+        nomeUsuario: "artadam",
+        qtLikes: 2
+    }
+
+    const listaTweets = [
+        React.createElement(Tweet, infoTweet1, [
+            "alo alo"
+        ]),
+        <Tweet { ...infoTweet2 } >
+            { infoTweet2.conteudo }
+        </Tweet>
+    ]
+
+    return (
+        <div>
+            <header class="cabecalho">
+                <div class="cabecalho__container container">
+                    <h1 class="cabecalho__logo">
+                        <a href="/">Twitelum</a>
+                    </h1>
+                    <nav class="navMenu">
+                        <ul class="navMenu__lista">
+                            <li class="navMenu__item">
```

```
+                                <a class="navMenu__link" href="/">
+                                    Bem vindo(a): <br />
+                                    <strong> @alumna</strong>
+                                </a>
+                            </li>
+                            <li class="navMenu__item">
+                                <a class="navMenu__link" href="/">
+                                    Página Inicial
+                                </a>
+                            </li>
+                            <li class="navMenu__item">
+                                <a class="navMenu__link" href="/hashtags">
+                                    Hashtags
+                                </a>
+                            </li>
+                            <li class="navMenu__item">
+                                <a class="navMenu__link" href="/logout">
+                                    Logout
+                                </a>
+                            </li>
+                        </ul>
+                    </nav>
+                </div>
+            </header>
+
+            <div class="container">
+                <div class="dashboard">
+                <div class="widget">
+                    <form class="novoTweet">
+                        <div class="novoTweet__editorArea">
+                            <span class="novoTweet__status">0/140</span>
+                            <textarea class="novoTweet__editor" placeholder="O que está acontece
ndo?"></textarea>
+                        </div>
+                        <button type="submit" class="novoTweet__envia">Tweetar</button>
+                    </form>
+                </div>
+                <div class="widget">
+                    <div class="trendsArea">
+                        <h2 class="trendsArea__titulo widget__titulo">Trends Brasil</h2>
+                        <ol class="trendsArea__lista">
+                            <li><a href="/">#react</a></li>
+                            <li><a href="/">#reactHooks</a></li>
+                        </ol>
+                    </div>
+                </div>
+            </div>
+
+                <div class="dashboard dashboard__centro">
+                    <div class="widget">
+                        <div class="tweetsArea">
+                            { listaTweets }
+                        </div>
+                    </div>
+                </div>
+            </div>
+        </div>
+    )
+}
```

6. Remova o arquivo **`tweet.html`** na pasta **`raíz do projeto`** .

7. No arquivo **`index.html`** na pasta **`public`** faça as seguintes alterações:

 # public/index.html

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="css/reset.css">
    <link rel="stylesheet" href="css/container.css">
    <link rel="stylesheet" href="css/btn.css">
    <link rel="stylesheet" href="css/icon.css">
    <link rel="stylesheet" href="css/iconHeart.css">
    <link rel="stylesheet" href="css/notificacao.css">

    <link rel="stylesheet" href="css/cabecalho.css">
    <link rel="stylesheet" href="css/navMenu.css">
    <link rel="stylesheet" href="css/dashboard.css">
    <link rel="stylesheet" href="css/widget.css">
    <link rel="stylesheet" href="css/novoTweet.css">
    <link rel="stylesheet" href="css/trendsArea.css">
    <!-- <link rel="stylesheet" href="css/tweet.css"> -->

    <title> Twittelum </title>
</head>

<body>
-    <header class="cabecalho">
-        <div class="cabecalho__container container">
-            <h1 class="cabecalho__logo">
-                <a href="/">Twitelum</a>
-            </h1>
-            <nav class=navMenu>
-                <ul class=navMenu__lista>
-                    <li class=navMenu__item>
-                        <a class=navMenu__link href="/">
-                            Bem vindo(a): <br />
-                            <strong> @alumna</strong>
-                        </a>
-                    </li>
-                    <li class=navMenu__item>
-                        <a class=navMenu__link href="/">
-                            Página Inicial
-                        </a>
-                    </li>
-                    <li class=navMenu__item>
-                        <a class=navMenu__link href="/hashtags">
-                            Hashtags
-                        </a>
-                    </li>
-                    <li class=navMenu__item>
-                        <a class=navMenu__link href="/logout">
-                            Logout
-                        </a>
-                    </li>
-                </ul>
-            </nav>
-        </div>
-    </header>
```

```
-    <div class="container">
-
-        <div class="dashboard">
-            <div class="widget">
-                <form class="novoTweet">
-                    <div class="novoTweet__editorArea">
-                        <span class="novoTweet__status">0/140</span>
-                        <textarea class="novoTweet__editor" placeholder="O que está acontecendo?
"></textarea>
-                    </div>
-                    <button type="submit" class="novoTweet__envia">Tweetar</button>
-                </form>
-            </div>
-            <div class="widget">
-                <div class="trendsArea">
-                    <h2 class="trendsArea__titulo widget__titulo">Trends Brasil</h2>
-                    <ol class="trendsArea__lista">
-                        <li><a href="/">#react</a></li>
-                        <li><a href="/">#reactHooks</a></li>
-                    </ol>
-                </div>
-            </div>
-        </div>
-
-        <div class="dashboard dashboard__centro">
-            <div class="widget">
-                <div class="tweetsArea">
-
-                </div>
-            </div>
-        </div>
-
-    </div>
-
-    <!-- <script src="js/lib/react.js"></script>
-    <script src="js/lib/react-dom.js"></script> -->
-
-    <!-- <script src="js/app.js" ></script> -->
 </body>

 </html>
```

8. Há 12 arquivos a serem adicionados na pasta build. Note que esses arquivos foram disponibilizados já prontos para você.

   Os 12 arquivos devem ser adicionados na seguinte estrutura de pastas:

   # novos arquivos do projeto

```
└─ pasta_raiz_do_projeto
   └─ build
      └─ css
         ├─ btn.css
         ├─ cabecalho.css
         ├─ container.css
         ├─ dashboard.css
         ├─ icon.css
         ├─ iconHeart.css
         ├─ navMenu.css
         ├─ notificacao.css
```

```
├─ novoTweet.css
├─ reset.css
├─ trendsArea.css
└─ widget.css
```

CAPÍTULO 6

EXERCÍCIO: COMPONENTIZAÇÃO DA HOME.

6.1 PASSO A PASSO COM CÓDIGO

1. Crie o arquivo **`Cabecalho.jsx`** na pasta **`src/components/Cabecalho`** com o seguinte código:

   # src/components/Cabecalho/Cabecalho.jsx

```
+import React from 'react';
+import './cabecalho.css';
+
+function Cabecalho(props) {
+    return (
+        <header className="cabecalho">
+            <div className="cabecalho__container container">
+                <h1 className="cabecalho__logo">
+                    <a href="/">Twitelum</a>
+                </h1>
+                { props.children }
+            </div>
+        </header>
+    )
+}
+
+export { Cabecalho }
```

2. Crie o arquivo **`Dashboard.jsx`** na pasta **`src/components/Dashboard`** com o seguinte código:

   # src/components/Dashboard/Dashboard.jsx

```
+import React from 'react'
+import './dashboard.css'
+
+function Dashboard(props) {
+    return (
+        <div className={`dashboard dashboard__${props.posicao}`}>
+            {props.children}
+        </div>
+    )
+}
+
+export { Dashboard }
```

3. Crie o arquivo **`NavMenu.jsx`** na pasta **`src/components/NavMenu`** com o seguinte código:

   # src/components/NavMenu/NavMenu.jsx

```
+import React from "react";
+import navMenuStyles from "./navMenu.module.css";
+
+function NavMenu(props) {
+  return (
+    <nav className={navMenuStyles.navMenu}>
+      <ul className={navMenuStyles.navMenu__lista}>
+        <li className={navMenuStyles.navMenu__item}>
+          <a className={navMenuStyles.navMenu__link} href="/">
+            Bem vindo(a): <br />
+            <strong>{props.usuario}</strong>
+          </a>
+        </li>
+        <li className={navMenuStyles.navMenu__item}>
+          <a className={navMenuStyles.navMenu__link} href="/">
+            Página Inicial
+          </a>
+        </li>
+        <li className={navMenuStyles.navMenu__item}>
+          <a className={navMenuStyles.navMenu__link} href="/hashtags">
+            Hashtags
+          </a>
+        </li>
+        <li className={navMenuStyles.navMenu__item}>
+          <a className={navMenuStyles.navMenu__link} href="/logout">
+            Logout
+          </a>
+        </li>
+      </ul>
+    </nav>
+  );
+}
+
+export { NavMenu }
```

4. Crie o arquivo **`TrendsArea.jsx`** na pasta **`src/components/TrendsArea`** com o seguinte código:

 # src/components/TrendsArea/TrendsArea.jsx

```
+import React from 'react'
+import './trendsArea.css'
+
+function TrendsArea() {
+    return (
+        <div className="trendsArea">
+            <h2 className="trendsArea__titulo widget__titulo">Trends Brasil</h2>
+            <ol className="trendsArea__lista">
+                <li><a href="/">#bagulhos</a></li>
+                <li><a href="/">#bagulheiros</a></li>
+            </ol>
+        </div>
+    )
+}
+
+export { TrendsArea }
```

5. No arquivo **`Tweet.jsx`** na pasta **`src/components/Tweet`** faça as seguintes alterações:

 # src/components/Tweet/Tweet.jsx

```
 import React from 'react'
```

```
 import './tweet.css'

-function TweetCabecalho({ nomeCompletoUsuario, nomeUsuario }) {
+export function TweetCabecalho({ nomeCompletoUsuario, nomeUsuario }) {
     return (
         <div className="tweet__cabecalho">
             <img className="tweet__fotoUsuario" src="https://placehold.it/50x50" alt="" />
             <span className="tweet__nomeUsuario">{ nomeCompletoUsuario }</span>
             <a href="/"><span className="tweet__userName">@{ nomeUsuario }</span></a>
         </div>
     )
 }

 function TweetFooter({ qtLikes }) {
     return (
         <footer className="tweet__footer">
             <button className="btn btn--clean">
                 <svg className="icon icon--small iconHeart" xmlns="http://www.w3.org/2000/svg" v
iewBox="0 0 47.5 47.5">
                     <defs>
                         <clipPath id="a">
                             <path d="M0 38h38V0H0v38z"></path>
                         </clipPath>
                     </defs>
                     <g clipPath="url(#a)" transform="matrix(1.25 0 0 -1.25 0 47.5)">
                         <path d="M36.885 25.166c0 5.45-4.418 9.868-9.867 9.868-3.308 0-6.227-1.6
32-8.018-4.128-1.79 2.496-4.71 4.129-8.017 4.129-5.45 0-9.868-4.418-9.868-9.868 0-.773.098-1.52.26
6-2.242C2.75 14.413 12.216 5.431 19 2.965c6.783 2.466 16.249 11.448 17.617 19.96.17.721.268 1.47.2
68 2.241"></path>
                     </g>
                 </svg>
                 { qtLikes }
             </button>
         </footer>
     )
 }

 // Componente Tweet
 export function Tweet(props) {

     const conteudo = props.children

     return (
         <article className="tweet">
             <TweetCabecalho
                 nomeCompletoUsuario={props.nomeCompletoUsuario}
                 nomeUsuario={props.nomeUsuario}
             />

             <p className="tweet__conteudo">{ conteudo }</p>

             <TweetFooter qtLikes={props.qtLikes} />

         </article>
     )
 }
```

6. Crie o arquivo **`Widget.jsx`** na pasta **`src/components/Widget`** com o seguinte código:

 # src/components/Widget/Widget.jsx

```
+import React from 'react'
+import './widget.css'
+
+function Widget(props) {
+    return (
+        <div className="widget">
+            { props.children }
+        </div>
+    )
+}
+
+export { Widget }
```

7. Crie o arquivo **`index.js`** na pasta **`src/components`** com o seguinte código:

 # src/components/index.js

```
+export { Cabecalho } from './Cabecalho/Cabecalho.jsx'
+export { NavMenu } from './NavMenu/NavMenu.jsx'
+export { Dashboard } from './Dashboard/Dashboard.jsx'
+export { Widget } from './Widget/Widget.jsx'
+export { TrendsArea } from './TrendsArea/TrendsArea.jsx'
+export { Tweet } from './Tweet/Tweet.jsx'
```

8. No arquivo **`index.js`** na pasta **`src`** faça as seguintes alterações:

 # src/index.js

```
 import React from 'react'
 import ReactDOM from 'react-dom'

 import { Home } from './pages/Home/Home.jsx'

 ReactDOM.render(
     <Home />,
-    document.querySelector('body')
+    document.querySelector('#raiz')
 )
```

9. No arquivo **`Home.jsx`** na pasta **`src/pages/Home`** faça as seguintes alterações:

 # src/pages/Home/Home.jsx

```
 import React from 'react'
-import { Tweet } from '../../components/Tweet/Tweet.jsx'
+import { Tweet, Cabecalho, NavMenu, Dashboard, Widget, TrendsArea } from '../../components/index
.js'

 export function Home() {
     const infoTweet1 = {
         conteudo: 'oi',
         nomeCompletoUsuario: "Artur Diniz",
         nomeUsuario: "artdiniz",
         qtLikes: 0
     }

     const infoTweet2 = {
         conteudo: 'tchau',
         nomeCompletoUsuario: "Artur Adam",
         nomeUsuario: "artadam",
```

```
        qtLikes: 2
    }

    const listaTweets = [
-       React.createElement(Tweet, infoTweet1, [
+       React.createElement(
+           Tweet,
+           {...infoTweet1, key: 1},
+           [
                "alo alo"
-       ]),
-       <Tweet { ...infoTweet2 } >
+           ]
+       ),
+       <Tweet { ...infoTweet2 } key="2">
            { infoTweet2.conteudo }
        </Tweet>
    ]

    return (
        <div>
-           <header class="cabecalho">
-               <div class="cabecalho__container container">
-                   <h1 class="cabecalho__logo">
-                       <a href="/">Twitelum</a>
-                   </h1>
-                   <nav class="navMenu">
-                       <ul class="navMenu__lista">
-                           <li class="navMenu__item">
-                               <a class="navMenu__link" href="/">
-                                   Bem vindo(a): <br />
-                                   <strong> @alumna</strong>
-                               </a>
-                           </li>
-                           <li class="navMenu__item">
-                               <a class="navMenu__link" href="/">
-                                   Página Inicial
-                               </a>
-                           </li>
-                           <li class="navMenu__item">
-                               <a class="navMenu__link" href="/hashtags">
-                                   Hashtags
-                               </a>
-                           </li>
-                           <li class="navMenu__item">
-                               <a class="navMenu__link" href="/logout">
-                                   Logout
-                               </a>
-                           </li>
-                       </ul>
-                   </nav>
-               </div>
-           </header>
+           <Cabecalho>
+              <NavMenu> </NavMenu>
+           </Cabecalho>

-           <div class="container">
-               <div class="dashboard">
-               <div class="widget">
-                   <form class="novoTweet">
-                       <div class="novoTweet__editorArea">
-                           <span class="novoTweet__status">0/140</span>
```

```
-                           <textarea class="novoTweet__editor" placeholder="O que está acontece
ndo?"></textarea>
+            <div className="container">
+                <Dashboard>
+                    <Widget>
+                        <form className="novoTweet">
+                            <div className="novoTweet__editorArea">
+                                <span className="novoTweet__status">0/140</span>
+                                <textarea className="novoTweet__editor" placeholder="O que está
acontecendo?"></textarea>
                             </div>
-                        <button type="submit" class="novoTweet__envia">Tweetar</button>
+                            <button type="submit" className="novoTweet__envia">Tweetar</button>
                         </form>
-                </div>
-                <div class="widget">
-                    <div class="trendsArea">
-                        <h2 class="trendsArea__titulo widget__titulo">Trends Brasil</h2>
-                        <ol class="trendsArea__lista">
-                            <li><a href="/">#react</a></li>
-                            <li><a href="/">#reactHooks</a></li>
-                        </ol>
-                    </div>
-                </div>
-            </div>
+                    </Widget>
+                    <Widget>
+                        <TrendsArea></TrendsArea>
+                    </Widget>
+                </Dashboard>

-                <div class="dashboard dashboard__centro">
-                    <div class="widget">
-                        <div class="tweetsArea">
+                <Dashboard posicao="centro">
+                    <Widget>
+                        <div className="tweetsArea">
+                            { listaTweets }
                         </div>
-                    </div>
-                </div>
+                    </Widget>
+                </Dashboard>
             </div>
         </div>
     )
 }
```

10. No arquivo **`index.html`** na pasta **`public`** faça as seguintes alterações:

 # public/index.html

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="css/reset.css">
    <link rel="stylesheet" href="css/container.css">
    <link rel="stylesheet" href="css/btn.css">
```

```
    <link rel="stylesheet" href="css/icon.css">
    <link rel="stylesheet" href="css/iconHeart.css">
    <link rel="stylesheet" href="css/notificacao.css">

    <link rel="stylesheet" href="css/cabecalho.css">
    <link rel="stylesheet" href="css/navMenu.css">
    <link rel="stylesheet" href="css/dashboard.css">
    <link rel="stylesheet" href="css/widget.css">
    <link rel="stylesheet" href="css/novoTweet.css">
    <link rel="stylesheet" href="css/trendsArea.css">
    <!-- <link rel="stylesheet" href="css/tweet.css"> -->

    <title> Twittelum </title>
</head>

<body>
+   <div id="raiz"></div>

</body>

</html>
```

11. Mova o arquivo **`cabecalho.css`** da pasta **`public/css`** para a pasta **`src/components/Cabecalho`**.

12. Mova o arquivo **`dashboard.css`** para a pasta **`src/components/Dashboard`**. No momento este arquivo está na pasta **`public/css`**. Faça também as seguintes modificações no código:

 # src/components/Dashboard/dashboard.css

```
/* dashboard */
.dashboard {
  float: left;
  width: 30%;
}
+
@media (max-width: 610px) {
  .dashboard {
    width: 100%;
  }
}
+
@media (min-width: 611px) {
  .dashboard__centro {
    width: calc(70% - 15px);
    margin-left: 15px;
    float: right;
  }
}
```

13. Mova o arquivo **`navMenu.css`** da pasta **`public/css`** para a pasta **`src/components/NavMenu`** e o renomeie para **`navMenu.module.css`**.

14. Mova o arquivo **`trendsArea.css`** da pasta **`public/css`** para a pasta **`src/components/TrendsArea`**.

15. Mova o arquivo **`widget.css`** da pasta **`public/css`** para a pasta **`src/components/Widget`**.

CAPÍTULO 7

EXERCÍCIO: ADICIONA TWEET – VALIDANDO O TWEET DIGITADO. OS EVENTOS E O ESTADO DOS COMPONENTES.

7.1 PASSO A PASSO COM CÓDIGO

1. No arquivo **`Home.jsx`** na pasta **`src/pages/Home`** faça as seguintes alterações:

 # src/pages/Home/Home.jsx

```
-import React from 'react'
-import { Tweet, Cabecalho, NavMenu, Dashboard, Widget, TrendsArea } from '../../components/index
.js'
+// Hook
+import React, { useState } from 'react'
+import {
+    Tweet,
+    Cabecalho,
+    NavMenu,
+    Dashboard,
+    Widget,
+    TrendsArea
+} from '../../components/index.js'

 export function Home() {
     const infoTweet1 = {
         conteudo: 'oi',
         nomeCompletoUsuario: "Artur Diniz",
         nomeUsuario: "artdiniz",
         qtLikes: 0
     }

     const infoTweet2 = {
         conteudo: 'tchau',
         nomeCompletoUsuario: "Artur Adam",
         nomeUsuario: "artadam",
         qtLikes: 2
     }

     const listaTweets = [
         React.createElement(
             Tweet,
             {...infoTweet1, key: 1},
             [
```

```
                "alo alo"
            ]
        ),
        <Tweet { ...infoTweet2 } key="2">
            { infoTweet2.conteudo }
        </Tweet>
    ]

+   const [ valorTamanhoTweetNovo, setTamanhoTweetNovo ] = useState(0)
+
+   function onChangeTextarea(evento) {
+       const novoTamanho = evento.target.value.length
+       setTamanhoTweetNovo(novoTamanho)
+   }
+
    return (
        <div>
            <Cabecalho>
               <NavMenu> </NavMenu>
            </Cabecalho>

            <div className="container">
                <Dashboard>
                    <Widget>
                        <form className="novoTweet">
                            <div className="novoTweet__editorArea">
-                               <span className="novoTweet__status">0/140</span>
-                               <textarea className="novoTweet__editor" placeholder="O que está
acontecendo?"></textarea>
+                               <span className="novoTweet__status">{ valorTamanhoTweetNovo }/14
0</span>
+                               <textarea
+                                   onChange={ onChangeTextarea }
+                                   className="novoTweet__editor"
+                                   placeholder="O que está acontecendo?"
+                               >
+                               </textarea>
                            </div>
                            <button type="submit" className="novoTweet__envia">Tweetar</button>
                        </form>
                    </Widget>
                    <Widget>
                        <TrendsArea></TrendsArea>
                    </Widget>
                </Dashboard>

                <Dashboard posicao="centro">
                    <Widget>
                        <div className="tweetsArea">
                            { listaTweets }
                        </div>
                    </Widget>
                </Dashboard>
            </div>
        </div>
    )
 }
```

2. No arquivo **`index.html`** na pasta **`public`** faça as seguintes alterações:

 # public/index.html

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="css/reset.css">
    <link rel="stylesheet" href="css/container.css">
    <link rel="stylesheet" href="css/btn.css">
    <link rel="stylesheet" href="css/icon.css">
    <link rel="stylesheet" href="css/iconHeart.css">
    <link rel="stylesheet" href="css/notificacao.css">
-
-    <link rel="stylesheet" href="css/cabecalho.css">
-    <link rel="stylesheet" href="css/navMenu.css">
-    <link rel="stylesheet" href="css/dashboard.css">
-    <link rel="stylesheet" href="css/widget.css">
    <link rel="stylesheet" href="css/novoTweet.css">
-    <link rel="stylesheet" href="css/trendsArea.css">
-    <!-- <link rel="stylesheet" href="css/tweet.css"> -->

    <title> Twittelum </title>
</head>

<body>
    <div id="raiz"></div>

</body>

</html>
```

CAPÍTULO 8

EXERCÍCIO: ADICIONANDO TWEET – MAIS EVENTOS COM HOOKS E LISTAS.

8.1 PASSO A PASSO COM CÓDIGO

1. No arquivo **`Home.jsx`** na pasta **`src/pages/Home`** faça as seguintes alterações:

 # src/pages/Home/Home.jsx

```
 // Hook
 import React, { useState } from 'react'
 import {
     Tweet,
     Cabecalho,
     NavMenu,
     Dashboard,
     Widget,
     TrendsArea
 } from '../../components/index.js'

-export function Home() {
-    const infoTweet1 = {
-        conteudo: 'oi',
+const listaInicialFake = [
+    {
+        conteudo: 'alo alo',
         nomeCompletoUsuario: "Artur Diniz",
         nomeUsuario: "artdiniz",
-        qtLikes: 0
+        qtLikes: 0,
+        id: 5
+    },
+    {
+        conteudo: 'vamo logar',
+        nomeCompletoUsuario: "Artur Adam",
+        nomeUsuario: "artadam",
+        qtLikes: 2,
+        id: 7
     }
+]

-    const infoTweet2 = {
-        conteudo: 'tchau',
+export function Home() {
+    const [ textoTweetNovo, setTextoTweetNovo ] = useState("")
+
+    const [ listaTweets, setListaTweets ] = useState(listaInicialFake)
+
+    function onSubmitFormulario(eventoSubmit) {
```

```
+        eventoSubmit.preventDefault()
+        const novoTweet = {
+            conteudo: textoTweetNovo,
             nomeCompletoUsuario: "Artur Adam",
             nomeUsuario: "artadam",
-        qtLikes: 2
+            qtLikes: 2,
+            id: 7
         }

-    const listaTweets = [
-        React.createElement(
-            Tweet,
-            {...infoTweet1, key: 1},
-            [
-                "alo alo"
-            ]
-        ),
-        <Tweet { ...infoTweet2 } key="2">
-            { infoTweet2.conteudo }
-        </Tweet>
-    ]
+        setListaTweets( [ novoTweet, ...listaTweets ] )
+    }

-    const [ valorTamanhoTweetNovo, setTamanhoTweetNovo ] = useState(0)

     function onChangeTextarea(evento) {
-        const novoTamanho = evento.target.value.length
-        setTamanhoTweetNovo(novoTamanho)
+        const novoTweet = evento.target.value
+        setTextoTweetNovo(novoTweet)
     }

+    const isInvalido = textoTweetNovo.length > 140
+
+    const classeStatusTweet = (isInvalido)
+        ? "novoTweet__status novoTweet__status--invalido"
+        : "novoTweet__status"
+
     return (
         <div>
             <Cabecalho>
                <NavMenu> </NavMenu>
             </Cabecalho>

             <div className="container">
                 <Dashboard>
                     <Widget>
-                        <form className="novoTweet">
+                        <form onSubmit={ onSubmitFormulario } className="novoTweet">
                             <div className="novoTweet__editorArea">
-                                <span className="novoTweet__status">{ valorTamanhoTweetNovo }/14
0</span>
+                                <span className={ classeStatusTweet }>
+                                    { textoTweetNovo.length }/140
+                                </span>
                                 <textarea
+                                    id="novoTweet"
                                     onChange={ onChangeTextarea }
                                     className="novoTweet__editor"
                                     placeholder="O que está acontecendo?"
                                 >
```

```
                                </textarea>
                            </div>
 -                          <button type="submit" className="novoTweet__envia">Tweetar</button>
 +                          <button disabled={ isInvalido } type="submit" className="novoTweet__
envia">Tweetar</button>
                        </form>
                    </Widget>
                    <Widget>
                        <TrendsArea></TrendsArea>
                    </Widget>
                </Dashboard>

                <Dashboard posicao="centro">
                    <Widget>
                        <div className="tweetsArea">
 -                          { listaTweets }
 +
 +                          {listaTweets.map(infoTweet =>
 +                              <Tweet { ...infoTweet } key={infoTweet.id}>
 +                                  { infoTweet.conteudo }
 +                              </Tweet>
 +                          )}
 +
                        </div>
                    </Widget>
                </Dashboard>
            </div>
        </div>
    )
  }
```

Capítulo 9

EXERCÍCIO: ROTEAMENTO PARA PÁGINA DE LOGIN – USANDO LIBS EXTERNAS E O ROTEAMENTO.

9.1 PASSO A PASSO COM CÓDIGO

1. No arquivo **`package.json`** na pasta **`raíz do projeto`** faça as seguintes alterações:

 # package.json

```
 {
   "name": "twittelum",
   "version": "0.1.0",
   "private": true,
   "dependencies": {
     "react": "^16.12.0",
     "react-dom": "^16.12.0",
+    "react-router-dom": "^5.1.2",
     "react-scripts": "^3.3.0"
   },
   "scripts": {
     "start": "react-scripts start",
     "_build:react": "react-scripts build",
     "build": "npm run _build:react"
   },
   "browserslist": {
     "production": [
       ">0.2%",
       "not dead",
       "not op_mini all"
     ],
     "development": [
       "last 1 chrome version",
       "last 1 firefox version",
       "last 1 safari version"
     ]
   }
 }
```

2. Crie o arquivo **`App.jsx`** na pasta **`src`** com o seguinte código:

 # src/App.jsx

```
+import React from 'react'
+
+import { Home } from './pages/Home/Home.jsx'
```

```
+import { Login } from './pages/Login/Login.jsx'
+import { BrowserRouter as Router, Route, Switch } from 'react-router-dom'
+
+export function App() {
+    return (
+        <Router>
+            <Switch>
+                <Route path="/" exact={true}>
+                    <Home />
+                </Route>
+                <Route path="/login">
+                    <Login />
+                </Route>
+            </Switch>
+        </Router>
+    )
+}
```

3. No arquivo **`Cabecalho.jsx`** na pasta **`src/components/Cabecalho`** faça as seguintes alterações:

 # src/components/Cabecalho/Cabecalho.jsx

```
 import React from 'react';
 import './cabecalho.css';

 function Cabecalho(props) {
     return (
         <header className="cabecalho">
             <div className="cabecalho__container container">
                 <h1 classNamе="cabecalho__logo">
                     <a href="/">Twitelum</a>
                 </h1>
                 { props.children }
+
             </div>
         </header>
     )
 }

 export { Cabecalho }
```

4. No arquivo **`NavMenu.jsx`** na pasta **`src/components/NavMenu`** faça as seguintes alterações:

 # src/components/NavMenu/NavMenu.jsx

```
 import React from "react";
 import navMenuStyles from "./navMenu.module.css";

+import { Link } from 'react-router-dom'
+
 function NavMenu(props) {
   return (
     <nav classNamе={navMenuStyles.navMenu}>
       <ul classNamе={navMenuStyles.navMenu__lista}>
         <li classNamе={navMenuStyles.navMenu__item}>
           <a classNamе={navMenuStyles.navMenu__link} href="/">
             Bem vindo(a): <br />
             <strong>{props.usuario}</strong>
           </a>
         </li>
         <li classNamе={navMenuStyles.navMenu__item}>
```

```
-           <a className={navMenuStyles.navMenu__link} href="/">
+           <Link className={navMenuStyles.navMenu__link} to="/">
              Página Inicial
-           </a>
+           </Link>
          </li>
          <li className={navMenuStyles.navMenu__item}>
            <a className={navMenuStyles.navMenu__link} href="/hashtags">
              Hashtags
            </a>
          </li>
          <li className={navMenuStyles.navMenu__item}>
            <a className={navMenuStyles.navMenu__link} href="/logout">
              Logout
            </a>
          </li>
        </ul>
      </nav>
    );
  }

  export { NavMenu }
```

5. No arquivo **`index.js`** na pasta **`src`** faça as seguintes alterações:

 # src/index.js

```
  import React from 'react'
  import ReactDOM from 'react-dom'

- import { Home } from './pages/Home/Home.jsx'
+ import { App } from './App.jsx'

  ReactDOM.render(
-     <Home />,
+     <App />,
      document.querySelector('#raiz')
  )
```

6. Crie o arquivo **`Login.jsx`** na pasta **`src/pages/Login`** com o seguinte código:

 # src/pages/Login/Login.jsx

```
+ import React, { Component, Fragment } from 'react'
+ import { Cabecalho } from '../../components/Cabecalho/Cabecalho.jsx'
+ import { Widget } from '../../components/Widget/Widget.jsx'
+
+ import './loginPage.css'
+
+ function Login() {
+     return (
+         <Fragment>
+             <Cabecalho />
+             <div className="loginPage">
+                 <div className="container">
+                     <Widget>
+                         <h2 className="loginPage__title">Seja bem vindo!</h2>
+                         <form className="loginPage__form" action="/">
+                             <div className="loginPage__inputWrap">
+                                 <label className="loginPage__label" htmlFor="login">Login</label
>
```

```
 +                                <input className="loginPage__input" type="text" id="login" name=
"login"/>
 +                            </div>
 +                            <div className="loginPage__inputWrap">
 +                                <label className="loginPage__label" htmlFor="senha">Senha</label
>
 +                                <input className="loginPage__input" type="password" id="senha" n
ame="senha"/>
 +                            </div>
 +                            <div className="loginPage__errorBox">
 +                                Mensagem de erro!
 +                            </div>
 +                            <div className="loginPage__inputWrap">
 +                                <button className="loginPage__btnLogin" type="submit">
 +                                    Logar
 +                                </button>
 +                            </div>
 +                        </form>
 +                    </Widget>
 +                </div>
 +            </div>
 +        </Fragment>
 +    )
 +}
 +
 +
 +export {Login}
```

7. Adicione o arquivo **`loginPage.css`** na pasta **`src/pages/Login`** . Este arquivo é um arquivo que foi disponibilizado já pronto para você.

Capítulo 10

EXERCÍCIO: LOGIN - LÓGICA DE NEGÓCIO VS LÓGICA DE VIEW. VARIÁVEIS DE AMBIENTE NO PROCESSO DE BUILD.

10.1 PASSO A PASSO COM CÓDIGO

1. No arquivo **`.env`** na pasta **`raíz do projeto`** faça as seguintes alterações:

 # .env

```
  CHOKIDAR_USEPOLLING=true
 +REACT_APP_URL_API=https://twitelum-api.herokuapp.com
```

2. No arquivo **`NavMenu.jsx`** na pasta **`src/components/NavMenu`** faça as seguintes alterações:

 # src/components/NavMenu/NavMenu.jsx

```
  import React from "react";
  import navMenuStyles from "./navMenu.module.css";

  import { Link } from 'react-router-dom'

  function NavMenu(props) {
    return (
      <nav className={navMenuStyles.navMenu}>
        <ul className={navMenuStyles.navMenu__lista}>
          <li className={navMenuStyles.navMenu__item}>
            <a className={navMenuStyles.navMenu__link} href="/">
              Bem vindo(a): <br />
              <strong>{props.usuario}</strong>
            </a>
          </li>
          <li className={navMenuStyles.navMenu__item}>
            <Link className={navMenuStyles.navMenu__link} to="/">
              Página Inicial
            </Link>
          </li>
          <li className={navMenuStyles.navMenu__item}>
            <a className={navMenuStyles.navMenu__link} href="/hashtags">
              Hashtags
            </a>
          </li>
          <li className={navMenuStyles.navMenu__item}>
```

```
            <a className={navMenuStyles.navMenu__link} href="/logout">
              Logout
+             {/* TODO roteamento e deslogamento */}
            </a>
          </li>
        </ul>
      </nav>
    );
  }

  export { NavMenu }
```

3. Crie o arquivo **`AutenticarService.js`** na pasta **`src/model/services`** com o seguinte código:

 # src/model/services/AutenticarService.js

```
+// React Scripts
+//  Webpack
+const URL_API = process.env.REACT_APP_URL_API
+
+export function isAutenticado() {
+    // Login inocente
+    return localStorage.getItem('TOKEN') !== null
+}
+
+export async function autenticar(usuario, senha) {
+    const response = await fetch(URL_API + '/login', {
+        method: "POST",
+        headers: {
+            "Content-Type": "application/json"
+        },
+        body: JSON.stringify({ login: usuario, senha})
+    })
+
+    if(!response.ok) {
+        const erroServidor = await (response.json())
+        const erroJS = Error(erroServidor.message)
+        erroJS.status = response.status
+        throw erroJS
+    }
+
+
+    const loginInfoServidor = await response.json()
+
+    const tokenLogin = loginInfoServidor.token
+
+
+    if(tokenLogin) {
+        localStorage.setItem('TOKEN', tokenLogin)
+        return tokenLogin
+    }
+
+    throw new Error("Token não encontrado")
+}
```

4. No arquivo **`Login.jsx`** na pasta **`src/pages/Login`** faça as seguintes alterações:

 # src/pages/Login/Login.jsx

```
-import React, { Component, Fragment } from 'react'
+import React, { Fragment, useState } from 'react'
```

```
 import { Cabecalho } from '../../components/Cabecalho/Cabecalho.jsx'
 import { Widget } from '../../components/Widget/Widget.jsx'

+
+import { Redirect } from 'react-router-dom'
+import * as AutenticarService from '../../model/services/AutenticarService.js'
+
 import './loginPage.css'

 function Login() {
-    return (
+
+    const [ msgErro, setMsgErro ] = useState("")
+
+    const [ isLogado, setIsLogado ] = useState(
+        AutenticarService.isAutenticado()
+    )
+
+    function onSubmitForm(eventoSubmit) {
+        eventoSubmit.preventDefault()
+        // TODO Revisando validacao
+
+        const usuario = eventoSubmit.target.elements.login.value
+        const senha = eventoSubmit.target.elements.senha.value
+
+        AutenticarService.autenticar(usuario, senha)
+            .then(() => {
+                setIsLogado(true)
+            })
+            .catch((erro) => {
+                setMsgErro(erro.message)
+            })
+
+    }
+
+    const pagina = (
         <Fragment>
             <Cabecalho />
             <div className="loginPage">
                 <div className="container">
                     <Widget>
                         <h2 className="loginPage__title">Seja bem vindo!</h2>
-                        <form className="loginPage__form" action="/">
+                        <form onSubmit={ onSubmitForm } className="loginPage__form" action="/">
                             <div className="loginPage__inputWrap">
                                 <label className="loginPage__label" htmlFor="login">Login</label
>
                                 <input className="loginPage__input" type="text" id="login" name=
"login"/>
                             </div>
                             <div className="loginPage__inputWrap">
                                 <label className="loginPage__label" htmlFor="senha">Senha</label
>
                                 <input className="loginPage__input" type="password" id="senha" n
ame="senha"/>
                             </div>
-                            <div className="loginPage__errorBox">
-                                Mensagem de erro!
+                            {(msgErro.length > 0)
+                                ? <div className="loginPage__errorBox">
+                                    { msgErro }
+                                  </div>
+                                : ''
```

```
+                            }
                             <div className="loginPage__inputWrap">
                                 <button className="loginPage__btnLogin" type="submit">
                                     Logar
                                 </button>
                             </div>
                         </form>
                     </Widget>
                 </div>
             </div>
         </Fragment>
     )
+
+    return (
+        (!isLogado)
+            ? pagina
+            : <Redirect to="/" />
+    )
 }

export {Login}
```

CAPÍTULO 11

EXERCÍCIO: LOGIN - AUTENTICAÇÃO PARA ACESSO ÀS PÁGINAS. ABSTRAINDO LÓGICA DE VIEW.

11.1 PASSO A PASSO COM CÓDIGO

1. Crie o arquivo **`ComponenteAutenticado.jsx`** na pasta **`src/components/ComponenteAutenticado`** com o seguinte código:

 # src/components/ComponenteAutenticado/ComponenteAutenticado.jsx

```
+import React from 'react'
+import { Redirect } from 'react-router-dom'
+
+export function ComponenteAutenticado(props) {
+
+    const { children: pagina, redirecionarPara, podeExibir} = props
+
+    return (
+        (podeExibir)
+            ? pagina
+            : <Redirect to={redirecionarPara} />
+    )
+}
```

2. No arquivo **`index.js`** na pasta **`src/components`** faça as seguintes alterações:

 # src/components/index.js

```
 export { Cabecalho } from './Cabecalho/Cabecalho.jsx'
 export { NavMenu } from './NavMenu/NavMenu.jsx'
 export { Dashboard } from './Dashboard/Dashboard.jsx'
 export { Widget } from './Widget/Widget.jsx'
 export { TrendsArea } from './TrendsArea/TrendsArea.jsx'
 export { Tweet } from './Tweet/Tweet.jsx'
+
+export { ComponenteAutenticado } from './ComponenteAutenticado/ComponenteAutenticado.jsx'
```

3. No arquivo **`Home.jsx`** na pasta **`src/pages/Home`** faça as seguintes alterações:

 # src/pages/Home/Home.jsx

```
 // Hook
 import React, { useState } from 'react'
 import {
```

```
     Tweet,
     Cabecalho,
     NavMenu,
     Dashboard,
     Widget,
-    TrendsArea 
+    TrendsArea,
+    ComponenteAutenticado
+
 } from '../../components/index.js'

+import * as AutenticarService from '../../model/services/AutenticarService.js'
+
 const listaInicialFake = [
     {
         conteudo: 'alo alo',
         nomeCompletoUsuario: "Artur Diniz",
         nomeUsuario: "artdiniz",
         qtLikes: 0,
         id: 5
     },
     {
         conteudo: 'vamo logar',
         nomeCompletoUsuario: "Artur Adam",
         nomeUsuario: "artadam",
         qtLikes: 2,
         id: 7
     }
 ]

-export function Home() {
+function HomeSemAutenticacao() {
     const [ textoTweetNovo, setTextoTweetNovo ] = useState("")

     const [ listaTweets, setListaTweets ] = useState(listaInicialFake)

     function onSubmitFormulario(eventoSubmit) {
         eventoSubmit.preventDefault()
         const novoTweet = {
             conteudo: textoTweetNovo,
             nomeCompletoUsuario: "Artur Adam",
             nomeUsuario: "artadam",
             qtLikes: 2,
             id: 7
         }

         setListaTweets( [ novoTweet, ...listaTweets ] )
     }

     function onChangeTextarea(evento) {
         const novoTweet = evento.target.value
         setTextoTweetNovo(novoTweet)
     }

     const isInvalido = textoTweetNovo.length > 140

     const classeStatusTweet = (isInvalido)
         ? "novoTweet__status novoTweet__status--invalido"
         : "novoTweet__status"

     return (
         <div>
             <Cabecalho>
```

```
                <NavMenu> </NavMenu>
            </Cabecalho>

            <div className="container">
                <Dashboard>
                    <Widget>
                        <form onSubmit={ onSubmitFormulario } className="novoTweet">
                            <div className="novoTweet__editorArea">
                                <span className={ classeStatusTweet }>
                                    { textoTweetNovo.length }/140
                                </span>
                                <textarea
                                    id="novoTweet"
                                    onChange={ onChangeTextarea }
                                    className="novoTweet__editor"
                                    placeholder="O que está acontecendo?"
                                >
                                </textarea>
                            </div>
                            <button disabled={ isInvalido } type="submit" className="novoTweet__
envia">Tweetar</button>
                        </form>
                    </Widget>
                    <Widget>
                        <TrendsArea></TrendsArea>
                    </Widget>
                </Dashboard>

                <Dashboard posicao="centro">
                    <Widget>
                        <div className="tweetsArea">

                            {listaTweets.map(infoTweet =>
                                <Tweet { ...infoTweet } key={infoTweet.id}>
                                    { infoTweet.conteudo }
                                </Tweet>
                            )}

                        </div>
                    </Widget>
                </Dashboard>
            </div>
        </div>
    )
 }
+
+
+export function Home(props) {
+    const isLogado = AutenticarService.isAutenticado()
+
+    return (
+        <ComponenteAutenticado podeExibir={isLogado} redirecionarPara="/login" >
+            <HomeSemAutenticacao {...props}/>
+        </ComponenteAutenticado>
+    )
+}
```

4. No arquivo **`Login.jsx`** na pasta **`src/pages/Login`** faça as seguintes alterações:

 # src/pages/Login/Login.jsx

```
 import React, { Fragment, useState } from 'react'
```

```
  import { Cabecalho } from '../../components/Cabecalho/Cabecalho.jsx'
  import { Widget } from '../../components/Widget/Widget.jsx'
 +import { ComponenteAutenticado } from '../../components/ComponenteAutenticado/ComponenteAutentic
ado.jsx'

 -
 -import { Redirect } from 'react-router-dom'
  import * as AutenticarService from '../../model/services/AutenticarService.js'

  import './loginPage.css'

  function Login() {

      const [ msgErro, setMsgErro ] = useState("")

      const [ isLogado, setIsLogado ] = useState(
          AutenticarService.isAutenticado()
      )

      function onSubmitForm(eventoSubmit) {
          eventoSubmit.preventDefault()
          // TODO Revisando validacao

          const usuario = eventoSubmit.target.elements.login.value
          const senha = eventoSubmit.target.elements.senha.value

          AutenticarService.autenticar(usuario, senha)
              .then(() => {
                  setIsLogado(true)
              })
              .catch((erro) => {
                  setMsgErro(erro.message)
              })

      }

      const pagina = (
          <Fragment>
              <Cabecalho />
              <div className="loginPage">
                  <div className="container">
                      <Widget>
                          <h2 className="loginPage__title">Seja bem vindo!</h2>
                          <form onSubmit={ onSubmitForm } className="loginPage__form" action="/">
                              <div className="loginPage__inputWrap">
                                  <label className="loginPage__label" htmlFor="login">Login</label
>
                                  <input className="loginPage__input" type="text" id="login" name=
"login"/>
                              </div>
                              <div className="loginPage__inputWrap">
                                  <label className="loginPage__label" htmlFor="senha">Senha</label
>
                                  <input className="loginPage__input" type="password" id="senha" n
ame="senha"/>
                              </div>
                              {(msgErro.length > 0)
                                  ? <div className="loginPage__errorBox">
                                      { msgErro }
                                    </div>
                                  : ''
                              }
                              <div className="loginPage__inputWrap">
```

```
                                <button className="loginPage__btnLogin" type="submit">
                                    Logar
                                </button>
                            </div>
                        </form>
                    </Widget>
                </div>
            </div>
        </Fragment>
    )

    return (
-       (!isLogado)
-           ? pagina
-           : <Redirect to="/" />
+       <ComponenteAutenticado podeExibir={!isLogado} redirecionarPara="/" >
+           { pagina }
+       </ComponenteAutenticado>
    )
 }

-
 export {Login}
```

CAPÍTULO 12

EXERCÍCIO: COMPONENTIZANDO O FORMULÁRIO DE ADICIONAR TWEET. COMPARTILHAMENTO DE ESTADO ENTRE COMPONENTES COM PROPS DE CALLBACK.

12.1 PASSO A PASSO COM CÓDIGO

1. Crie o arquivo **`Form.jsx`** na pasta **`src/components/Form`** com o seguinte código:

 # src/components/Form/Form.jsx

```
+import React, { useState } from 'react'
+
+import './novoTweet.css'
+
+export function Form(props) {
+    const [ textoTweetNovo, setTextoTweetNovo ] = useState("")
+
+    const { adicionaTweet } = props
+
+    function onSubmitFormulario(eventoSubmit) {
+        eventoSubmit.preventDefault()
+        adicionaTweet(textoTweetNovo)
+    }
+
+    function onChangeTextarea(evento) {
+        const novoTweet = evento.target.value
+        setTextoTweetNovo(novoTweet)
+    }
+
+    const isInvalido = textoTweetNovo.length > 140
+
+    const classeStatusTweet = (isInvalido)
+        ? "novoTweet__status novoTweet__status--invalido"
+        : "novoTweet__status"
+
+    return (
+        <form onSubmit={ onSubmitFormulario } className="novoTweet">
+            <div className="novoTweet__editorArea">
+                <span className={ classeStatusTweet }>
+                    { textoTweetNovo.length }/140
+                </span>
+                <textarea
```

```
+                    id="novoTweet"
+                    onChange={ onChangeTextarea }
+                    className="novoTweet__editor"
+                    placeholder="O que está acontecendo?"
+                >
+                </textarea>
+            </div>
+            <button disabled={ isInvalido } type="submit" className="novoTweet__envia">Tweetar</
button>
+        </form>
+    )
+}
```

2. No arquivo **`index.js`** na pasta **`src/components`** faça as seguintes alterações:

 # src/components/index.js

```
 export { Cabecalho } from './Cabecalho/Cabecalho.jsx'
 export { NavMenu } from './NavMenu/NavMenu.jsx'
 export { Dashboard } from './Dashboard/Dashboard.jsx'
 export { Widget } from './Widget/Widget.jsx'
 export { TrendsArea } from './TrendsArea/TrendsArea.jsx'
 export { Tweet } from './Tweet/Tweet.jsx'
-
 export { ComponenteAutenticado } from './ComponenteAutenticado/ComponenteAutenticado.jsx'
+export { Form } from './Form/Form.jsx'
```

3. No arquivo **`AutenticarService.js`** na pasta **`src/model/services`** faça as seguintes alterações:

 # src/model/services/AutenticarService.js

```
 // React Scripts
 //  Webpack
 const URL_API = process.env.REACT_APP_URL_API

+
+export function getToken() {
+    // Login inocente
+    return localStorage.getItem('TOKEN')
+}
+
 export function isAutenticado() {
     // Login inocente
     return localStorage.getItem('TOKEN') !== null
 }

 export async function autenticar(usuario, senha) {
     const response = await fetch(URL_API + '/login', {
         method: "POST",
         headers: {
             "Content-Type": "application/json"
         },
         body: JSON.stringify({ login: usuario, senha})
     })

     if(!response.ok) {
         const erroServidor = await (response.json())
         const erroJS = Error(erroServidor.message)
         erroJS.status = response.status
         throw erroJS
     }
```

```
        const loginInfoServidor = await response.json()

        const tokenLogin = loginInfoServidor.token

        if(tokenLogin) {
            localStorage.setItem('TOKEN', tokenLogin)
            return tokenLogin
        }

        throw new Error("Token não encontrado")
    }
```

4. Crie o arquivo **`TweetsService.js`** na pasta **`src/model/services`** com o seguinte código:

 # src/model/services/TweetsService.js

```
+import * as AutenticarService  from './AutenticarService.js'
+
+const TWEETS_URL = "https://twitelum-api.herokuapp.com/tweets"
+
+export const carrega = () =>
+    fetch(`${TWEETS_URL}?X-AUTH-TOKEN=${AutenticarService.getToken()}`)
+    .then(
+        response => response.json()
+    )
+
+export const adiciona = conteudo =>
+    fetch(
+        `${TWEETS_URL}?X-AUTH-TOKEN=${AutenticarService.getToken()}`
+        ,{
+            method: "POST",
+            headers: {
+                "Content-type": "application/json"
+            },
+            body: JSON.stringify({
+                conteudo
+            })
+        }
+    )
+    .then(respostaDoServer => {
+        return respostaDoServer.json()
+    })
+
+export const remove = idTweetQueVaiSerRemovido =>
+    fetch(
+        `https://twitelum-api.herokuapp.com/tweets/${idTweetQueVaiSerRemovido}?X-AUTH-TOKEN=${Au
tenticarService.getToken()}`
+        ,{ method: "DELETE" }
+    )
+    .then(data => data.json())
+
+export const like = idDoTweet =>
+    fetch(
+        `https://twitelum-api.herokuapp.com/tweets/${idDoTweet}/like?X-AUTH-TOKEN=${AutenticarSe
rvice.getToken()}`
+        ,{ method: "POST" }
+    )
+    .then(response => response.json())
```

5. No arquivo **`Home.jsx`** na pasta **`src/pages/Home`** faça as seguintes alterações:

 # src/pages/Home/Home.jsx

```
 // Hook
 import React, { useState } from 'react'
 import {
     Tweet,
     Cabecalho,
     NavMenu,
     Dashboard,
     Widget,
     TrendsArea,
-    ComponenteAutenticado
-
+    ComponenteAutenticado,
+    Form
 } from '../../components/index.js'

 import * as AutenticarService from '../../model/services/AutenticarService.js'
-
-const listaInicialFake = [
-    {
-        conteudo: 'alo alo',
-        nomeCompletoUsuario: "Artur Diniz",
-        nomeUsuario: "artdiniz",
-        qtLikes: 0,
-        id: 5
-    },
-    {
-        conteudo: 'vamo logar',
-        nomeCompletoUsuario: "Artur Adam",
-        nomeUsuario: "artadam",
-        qtLikes: 2,
-        id: 7
+import * as TweetsService from '../../model/services/TweetsService.js'
+
+function converteTweet(tweet) {
+    return {
+        nomeUsuario: tweet.usuario.login,
+        nomeCompletoUsuario: tweet.usuario.nome,
+        qtLikes: tweet.totalLikes,
+        conteudo: tweet.conteudo
     }
-]
-
-function HomeSemAutenticacao() {
-    const [ textoTweetNovo, setTextoTweetNovo ] = useState("")
-
-    const [ listaTweets, setListaTweets ] = useState(listaInicialFake)
-
-    function onSubmitFormulario(eventoSubmit) {
-        eventoSubmit.preventDefault()
-        const novoTweet = {
-            conteudo: textoTweetNovo,
-            nomeCompletoUsuario: "Artur Adam",
-            nomeUsuario: "artadam",
-            qtLikes: 2,
-            id: 7
-        }
-
-        setListaTweets( [ novoTweet, ...listaTweets ] )
-    }
-
```

```
+}

-    function onChangeTextarea(evento) {
-        const novoTweet = evento.target.value
-        setTextoTweetNovo(novoTweet)
+// TODO usar o Componente Autenticado
+export function HomeSemAutenticacao() {
+    const [ listaTweets, setListaTweets ] = useState([])
+
+    TweetsService.carrega()
+        .then(listaServidor => {
+            setListaTweets([
+                ...listaServidor.map(converteTweet),
+                ...listaTweets
+            ])
+        })
+
+    // Função Closure
+    // Prop de callback
+    // Adicionar Tweets no servidor
+    function adicionaTweet(textoTweetNovo) {
+        TweetsService.adiciona(textoTweetNovo)
+            .then(novoTweet => {
+                setListaTweets([
+                    converteTweet(novoTweet),
+                    ...listaTweets
+                ])
+            })
     }

-    const isInvalido = textoTweetNovo.length > 140
-
-    const classeStatusTweet = (isInvalido)
-        ? "novoTweet__status novoTweet__status--invalido"
-        : "novoTweet__status"
-
     return (
         <div>
             <Cabecalho>
-               <NavMenu> </NavMenu>
+               <NavMenu usuario="artadam"> </NavMenu>
             </Cabecalho>

             <div className="container">
                 <Dashboard>
                     <Widget>
-                        <form onSubmit={ onSubmitFormulario } className="novoTweet">
-                            <div className="novoTweet__editorArea">
-                                <span className={ classeStatusTweet }>
-                                    { textoTweetNovo.length }/140
-                                </span>
-                                <textarea
-                                    id="novoTweet"
-                                    onChange={ onChangeTextarea }
-                                    className="novoTweet__editor"
-                                    placeholder="O que está acontecendo?"
-                                >
-                                </textarea>
-                            </div>
-                            <button disabled={ isInvalido } type="submit" className="novoTweet__
envia">Tweetar</button>
-                        </form>
+                        <Form adicionaTweet={ adicionaTweet }/>
```

```
                    </Widget>
                    <Widget>
                        <TrendsArea></TrendsArea>
                    </Widget>
                </Dashboard>

                <Dashboard posicao="centro">
                    <Widget>
                        <div className="tweetsArea">

                            {listaTweets.map(infoTweet =>
                                <Tweet { ...infoTweet } key={infoTweet.id}>
                                    { infoTweet.conteudo }
                                </Tweet>
                            )}

                        </div>
                    </Widget>
                </Dashboard>
            </div>
        </div>
    )
}

export function Home(props) {
    const isLogado = AutenticarService.isAutenticado()

    return (
        <ComponenteAutenticado podeExibir={isLogado} redirecionarPara="/login" >
            <HomeSemAutenticacao {...props}/>
        </ComponenteAutenticado>
    )
}
```

6. No arquivo **`index.html`** na pasta **`public`** faça as seguintes alterações:

 # public/index.html

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="css/reset.css">
    <link rel="stylesheet" href="css/container.css">
    <link rel="stylesheet" href="css/btn.css">
    <link rel="stylesheet" href="css/icon.css">
    <link rel="stylesheet" href="css/iconHeart.css">
    <link rel="stylesheet" href="css/notificacao.css">
-   <link rel="stylesheet" href="css/novoTweet.css">

    <title> Twittelum </title>
</head>

<body>
    <div id="raiz"></div>

</body>

</html>
```

7. Mova o arquivo **`novoTweet.css`** da pasta **`public/css`** para a pasta **`src/components/Form`** .

CAPÍTULO 13

EXERCÍCIO: RESOLVENDO TWEETS QUE CARREGAM REPETIDAMENTE E INFINITAMENTE COM USEEFFECT.

13.1 PASSO A PASSO COM CÓDIGO

1. No arquivo **`Home.jsx`** na pasta **`src/pages/Home`** faça as seguintes alterações:

 # src/pages/Home/Home.jsx

```
 // Hook
-import React, { useState } from 'react'
+import React, { useState, useEffect} from 'react'
 import {
     Tweet,
     Cabecalho,
     NavMenu,
     Dashboard,
     Widget,
     TrendsArea,
     ComponenteAutenticado,
     Form
 } from '../../components/index.js'

 import * as AutenticarService from '../../model/services/AutenticarService.js'
 import * as TweetsService from '../../model/services/TweetsService.js'

 function converteTweet(tweet) {
     return {
         nomeUsuario: tweet.usuario.login,
         nomeCompletoUsuario: tweet.usuario.nome,
         qtLikes: tweet.totalLikes,
         conteudo: tweet.conteudo
     }
 }

 // TODO usar o Componente Autenticado
 export function HomeSemAutenticacao() {
     const [ listaTweets, setListaTweets ] = useState([])

+    useEffect(() => {
         TweetsService.carrega()
             .then(listaServidor => {
                 setListaTweets([
                     ...listaServidor.map(converteTweet),
                     ...listaTweets
                 ])
```

```
            })
+   }, [])

    // Função Closure
    // Prop de callback
    // Adicionar Tweets no servidor
    function adicionaTweet(textoTweetNovo) {
        TweetsService.adiciona(textoTweetNovo)
            .then(novoTweet => {
                setListaTweets([
                    converteTweet(novoTweet),
                    ...listaTweets
                ])
            })
    }

    return (
        <div>
            <Cabecalho>
               <NavMenu usuario="artadam"> </NavMenu>
            </Cabecalho>

            <div className="container">
                <Dashboard>
                    <Widget>
                        <Form adicionaTweet={ adicionaTweet }/>
                    </Widget>
                    <Widget>
                        <TrendsArea></TrendsArea>
                    </Widget>
                </Dashboard>

                <Dashboard posicao="centro">
                    <Widget>
                        <div className="tweetsArea">

                            {listaTweets.map(infoTweet =>
                                <Tweet { ...infoTweet } key={infoTweet.id}>
                                    { infoTweet.conteudo }
                                </Tweet>
                            )}

                        </div>
                    </Widget>
                </Dashboard>
            </div>
        </div>
    )
}

export function Home(props) {
    const isLogado = AutenticarService.isAutenticado()

    return (
        <ComponenteAutenticado podeExibir={isLogado} redirecionarPara="/login" >
            <HomeSemAutenticacao {...props}/>
        </ComponenteAutenticado>
    )
}
```

CAPÍTULO 14

EXERCÍCIO: COMPARTILHAMENTO DE ESTADO ENTRE COMPONENTES COM A CONTEXT API.

14.1 PASSO A PASSO COM CÓDIGO

1. No arquivo **`App.jsx`** na pasta **`src`** faça as seguintes alterações:

 # src/App.jsx

```
 import React from 'react'

 import { Home } from './pages/Home/Home.jsx'
 import { Login } from './pages/Login/Login.jsx'
 import { BrowserRouter as Router, Route, Switch } from 'react-router-dom'

+import { Notificacao } from './components/Notificacao/Notificacao.jsx'
+
 export function App() {
     return (
+    <Notificacao>
         <Router>
             <Switch>
                 <Route path="/" exact={true}>
                     <Home />
                 </Route>
                 <Route path="/login">
                     <Login />
                 </Route>
             </Switch>
         </Router>
+    </Notificacao>
     )
 }
```

2. No arquivo **`Form.jsx`** na pasta **`src/components/Form`** faça as seguintes alterações:

 # src/components/Form/Form.jsx

```
 import React, { useState } from 'react'

 import './novoTweet.css'

 export function Form(props) {
-    const [ textoTweetNovo, setTextoTweetNovo ] = useState("")
-
```

```
      const { adicionaTweet } = props

+
+     const [ textoTweetNovo, setTextoTweetNovo ] = useState("")
+
      function onSubmitFormulario(eventoSubmit) {
          eventoSubmit.preventDefault()
          adicionaTweet(textoTweetNovo)
      }

      function onChangeTextarea(evento) {
          const novoTweet = evento.target.value
          setTextoTweetNovo(novoTweet)
      }

      const isInvalido = textoTweetNovo.length > 140

      const classeStatusTweet = (isInvalido)
          ? "novoTweet__status novoTweet__status--invalido"
          : "novoTweet__status"

      return (
          <form onSubmit={ onSubmitFormulario } className="novoTweet">
              <div className="novoTweet__editorArea">
                  <span className={ classeStatusTweet }>
                      { textoTweetNovo.length }/140
                  </span>
                  <textarea
                      id="novoTweet"
                      onChange={ onChangeTextarea }
                      className="novoTweet__editor"
                      placeholder="O que está acontecendo?"
                  >
                  </textarea>
              </div>
              <button disabled={ isInvalido } type="submit" className="novoTweet__envia">Tweetar</
button>
          </form>
      )
  }
```

3. Crie o arquivo **`Notificacao.jsx`** na pasta **`src/components/Notificacao`** com o seguinte código:

 # src/components/Notificacao/Notificacao.jsx

```
+import React, { useState, createContext } from 'react'
+
+export const Contexto = createContext({
+     setMsg: () => {}
+})
+
+export function Notificacao(props) {
+     const conteudo = props.children
+
+     const [ msg, setMsg ] = useState("")
+
+     return (
+         <Contexto.Provider
+             value={ {setMsg: setMsg}  }
+         >
```

```
+            { conteudo }
+            {msg.length > 0
+                ? <div className="notificacaoMsg"> { msg } </div>
+                : ''
+            }
+        </Contexto.Provider>
+    )
+}
```

4. No arquivo **`Home.jsx`** na pasta **`src/pages/Home`** faça as seguintes alterações:

 # src/pages/Home/Home.jsx

```
 // Hook
-import React, { useState, useEffect} from 'react'
+import React, { useState, useEffect, useContext} from 'react'
 import {
     Tweet,
     Cabecalho,
     NavMenu,
     Dashboard,
     Widget,
     TrendsArea,
     ComponenteAutenticado,
     Form
 } from '../../components/index.js'

 import * as AutenticarService from '../../model/services/AutenticarService.js'
 import * as TweetsService from '../../model/services/TweetsService.js'

+import { Contexto as NotificacaoContexto } from '../../components/Notificacao/Notificacao.jsx'
+
 function converteTweet(tweet) {
     return {
         nomeUsuario: tweet.usuario.login,
         nomeCompletoUsuario: tweet.usuario.nome,
         qtLikes: tweet.totalLikes,
         conteudo: tweet.conteudo
     }
 }

 // TODO usar o Componente Autenticado
 export function HomeSemAutenticacao() {
     const [ listaTweets, setListaTweets ] = useState([])

+    const { setMsg } = useContext(NotificacaoContexto)
+
     useEffect(() => {
         TweetsService.carrega()
             .then(listaServidor => {
                 setListaTweets([
                     ...listaServidor.map(converteTweet),
                     ...listaTweets
                 ])
             })
     }, [])

+    useEffect(() => {
+        setTimeout(() => {
+            setMsg("")
+        }, 5000)
+    })
```

```
+
     // Função Closure
     // Prop de callback
     // Adicionar Tweets no servidor
     function adicionaTweet(textoTweetNovo) {
         TweetsService.adiciona(textoTweetNovo)
             .then(novoTweet => {
                 setListaTweets([
                     converteTweet(novoTweet),
                     ...listaTweets
                 ])
             })
     }

     return (
         <div>
             <Cabecalho>
                <NavMenu usuario="artadam"> </NavMenu>
             </Cabecalho>

             <div className="container">
                 <Dashboard>
                     <Widget>
                         <Form adicionaTweet={ adicionaTweet }/>
                     </Widget>
                     <Widget>
                         <TrendsArea></TrendsArea>
                     </Widget>
                 </Dashboard>

                 <Dashboard posicao="centro">
                     <Widget>
                         <div className="tweetsArea">

                             {listaTweets.map(infoTweet =>
                                 <Tweet { ...infoTweet } key={infoTweet.id}>
                                     { infoTweet.conteudo }
                                 </Tweet>
                             )}

                         </div>
                     </Widget>
                 </Dashboard>
             </div>
         </div>
     )
 }

 export function Home(props) {
     const isLogado = AutenticarService.isAutenticado()

     return (
         <ComponenteAutenticado podeExibir={isLogado} redirecionarPara="/login" >
             <HomeSemAutenticacao {...props}/>
         </ComponenteAutenticado>
     )
 }
```

5. No arquivo **`Login.jsx`** na pasta **`src/pages/Login`** faça as seguintes alterações:

 # src/pages/Login/Login.jsx

```
 -import React, { Fragment, useState } from 'react'
 +import React, { Fragment, useState, useContext } from 'react'
  import { Cabecalho } from '../../components/Cabecalho/Cabecalho.jsx'
  import { Widget } from '../../components/Widget/Widget.jsx'
  import { ComponenteAutenticado } from '../../components/ComponenteAutenticado/ComponenteAutentic
ado.jsx'

  import * as AutenticarService from '../../model/services/AutenticarService.js'

 +import { Contexto as NotificacaoContexto } from '../../components/Notificacao/Notificacao.jsx'
 +
  import './loginPage.css'

  function Login() {

      const [ msgErro, setMsgErro ] = useState("")

      const [ isLogado, setIsLogado ] = useState(
          AutenticarService.isAutenticado()
      )

 +    const { setMsg } = useContext(NotificacaoContexto)
 +
      function onSubmitForm(eventoSubmit) {
          eventoSubmit.preventDefault()
          // TODO Revisando validacao

          const usuario = eventoSubmit.target.elements.login.value
          const senha = eventoSubmit.target.elements.senha.value

          AutenticarService.autenticar(usuario, senha)
              .then(() => {
                  setIsLogado(true)
 +                setMsg("Logado com sucesso")
              })
              .catch((erro) => {
                  setMsgErro(erro.message)
              })

      }

      const pagina = (
          <Fragment>
              <Cabecalho />
              <div className="loginPage">
                  <div className="container">
                      <Widget>
                          <h2 className="loginPage__title">Seja bem vindo!</h2>
                          <form onSubmit={ onSubmitForm } className="loginPage__form" action="/">
                              <div className="loginPage__inputWrap">
                                  <label className="loginPage__label" htmlFor="login">Login</label
>
                                  <input className="loginPage__input" type="text" id="login" name=
"login"/>
                              </div>
                              <div className="loginPage__inputWrap">
                                  <label className="loginPage__label" htmlFor="senha">Senha</label
>
                                  <input className="loginPage__input" type="password" id="senha" n
ame="senha"/>
                              </div>
                              {(msgErro.length > 0)
                                  ? <div className="loginPage__errorBox">
```

```
                                    { msgErro }
                                </div>
                            : ''
                        }
                        <div className="loginPage__inputWrap">
                            <button className="loginPage__btnLogin" type="submit">
                                Logar
                            </button>
                        </div>
                    </form>
                </Widget>
            </div>
        </div>
    </Fragment>
)

return (
    <ComponenteAutenticado podeExibir={!isLogado} redirecionarPara="/" >
        { pagina }
    </ComponenteAutenticado>
)
}

export {Login}
```

CAPÍTULO 15

EXERCÍCIO: EXTRA - VALIDAÇÃO DO FORMULÁRIO DE LOGIN - O HOOK USEREF E INTEGRACAO COM VALIDACAO DO DOM.

15.1 PASSO A PASSO COM CÓDIGO

1. No arquivo **`Home.jsx`** na pasta **`src/pages/Home`** faça as seguintes alterações:

 # src/pages/Home/Home.jsx

```
// Hook
import React, { useState, useEffect, useContext} from 'react'
import {
    Tweet,
    Cabecalho,
    NavMenu,
    Dashboard,
    Widget,
    TrendsArea,
    ComponenteAutenticado,
    Form
} from '../../components/index.js'

import * as AutenticarService from '../../model/services/AutenticarService.js'
import * as TweetsService from '../../model/services/TweetsService.js'

import { Contexto as NotificacaoContexto } from '../../components/Notificacao/Notificacao.jsx'

function converteTweet(tweet) {
    return {
        nomeUsuario: tweet.usuario.login,
        nomeCompletoUsuario: tweet.usuario.nome,
        qtLikes: tweet.totalLikes,
        conteudo: tweet.conteudo
    }
}

-// TODO usar o Componente Autenticado
 export function HomeSemAutenticacao() {
    const [ listaTweets, setListaTweets ] = useState([])

    const { setMsg } = useContext(NotificacaoContexto)

    useEffect(() => {
```

```
        TweetsService.carrega()
            .then(listaServidor => {
                setListaTweets([
                    ...listaServidor.map(converteTweet),
                    ...listaTweets
                ])
            })
    }, [])

    useEffect(() => {
        setTimeout(() => {
            setMsg("")
        }, 5000)
    })

    // Função Closure
    // Prop de callback
    // Adicionar Tweets no servidor
    function adicionaTweet(textoTweetNovo) {
        TweetsService.adiciona(textoTweetNovo)
            .then(novoTweet => {
                setListaTweets([
                    converteTweet(novoTweet),
                    ...listaTweets
                ])
            })
    }

    return (
        <div>
            <Cabecalho>
               <NavMenu usuario="artadam"> </NavMenu>
            </Cabecalho>

            <div className="container">
                <Dashboard>
                    <Widget>
                        <Form adicionaTweet={ adicionaTweet }/>
                    </Widget>
                    <Widget>
                        <TrendsArea></TrendsArea>
                    </Widget>
                </Dashboard>

                <Dashboard posicao="centro">
                    <Widget>
                        <div className="tweetsArea">

                            {listaTweets.map(infoTweet =>
                                <Tweet { ...infoTweet } key={infoTweet.id}>
                                    { infoTweet.conteudo }
                                </Tweet>
                            )}

                        </div>
                    </Widget>
                </Dashboard>
            </div>
        </div>
    )
}

export function Home(props) {
```

```
    const isLogado = AutenticarService.isAutenticado()

    return (
        <ComponenteAutenticado podeExibir={isLogado} redirecionarPara="/login" >
            <HomeSemAutenticacao {...props}/>
        </ComponenteAutenticado>
    )
}
```

2. No arquivo **`Login.jsx`** na pasta **`src/pages/Login`** faça as seguintes alterações:

 # src/pages/Login/Login.jsx

```
-import React, { Fragment, useState, useContext } from 'react'
+import React, { Fragment, useState, useContext, useRef } from 'react'
 import { Cabecalho } from '../../components/Cabecalho/Cabecalho.jsx'
 import { Widget } from '../../components/Widget/Widget.jsx'
 import { ComponenteAutenticado } from '../../components/ComponenteAutenticado/ComponenteAutentic
ado.jsx'

 import * as AutenticarService from '../../model/services/AutenticarService.js'

 import { Contexto as NotificacaoContexto } from '../../components/Notificacao/Notificacao.jsx'

 import './loginPage.css'

 function Login() {

     const [ msgErro, setMsgErro ] = useState("")

     const [ isLogado, setIsLogado ] = useState(
         AutenticarService.isAutenticado()
     )

     const { setMsg } = useContext(NotificacaoContexto)

+    const refInputUsuarioDOM = useRef()
+    const refInputSenhaDOM = useRef()
+
     function onSubmitForm(eventoSubmit) {
         eventoSubmit.preventDefault()
-        // TODO Revisando validacao
+        setMsgErro("")
+
+        const $campoUsuario = refInputUsuarioDOM.current
+        const $campoSenha = refInputSenhaDOM.current

-        const usuario = eventoSubmit.target.elements.login.value
-        const senha = eventoSubmit.target.elements.senha.value
+        if(!$campoUsuario.validity.valid) {
+            setMsgErro('Usuário inválido. Min 3 carácteres.')
+        } else if(!$campoSenha.validity.valid) {
+            setMsgErro('Senha inválida. Min 6 carácteres.')
+        } else {
+            const usuario = $campoUsuario.value
+            const senha = $campoSenha.value

             AutenticarService.autenticar(usuario, senha)
                 .then(() => {
                     setIsLogado(true)
                     setMsg("Logado com sucesso")
                 })
```

```
                .catch((erro) => {
                    setMsgErro(erro.message)
                })
 -
 +       }
      }

      const pagina = (
          <Fragment>
              <Cabecalho />
              <div className="loginPage">
                  <div className="container">
                      <Widget>
                          <h2 className="loginPage__title">Seja bem vindo!</h2>
 -                        <form onSubmit={ onSubmitForm } className="loginPage__form" action="/">
 +                        <form noValidate onSubmit={ onSubmitForm } className="loginPage__form" a
ction="/">
                              <div className="loginPage__inputWrap">
                                  <label className="loginPage__label" htmlFor="login">Login</label
>
 -                                <input className="loginPage__input" type="text" id="login" name=
"login"/>
 +                                <input className="loginPage__input" type="text" id="login" name=
"login" ref={refInputUsuarioDOM} required minLength={3}/>
                              </div>
                              <div className="loginPage__inputWrap">
                                  <label className="loginPage__label" htmlFor="senha">Senha</label
>
 -                                <input className="loginPage__input" type="password" id="senha" n
ame="senha"/>
 +                                <input className="loginPage__input" type="password" id="senha" n
ame="senha" ref={refInputSenhaDOM} required minLength={6}/>
                              </div>
                              {(msgErro.length > 0)
                                  ? <div className="loginPage__errorBox">
                                      { msgErro }
                                    </div>
                                  : ''
                              }
                              <div className="loginPage__inputWrap">
                                  <button className="loginPage__btnLogin" type="submit">
                                      Logar
                                  </button>
                              </div>
                          </form>
                      </Widget>
                  </div>
              </div>
          </Fragment>
      )

      return (
          <ComponenteAutenticado podeExibir={!isLogado} redirecionarPara="/" >
              { pagina }
          </ComponenteAutenticado>
      )
  }

  export {Login}
```

CAPÍTULO 16

EXERCÍCIO: EXTRA - PÁGINA DE LOGOUT

16.1 PASSO A PASSO COM CÓDIGO

1. No arquivo **`App.jsx`** na pasta **`src`** faça as seguintes alterações:

 # src/App.jsx

```
 import React from 'react'

 import { Home } from './pages/Home/Home.jsx'
 import { Login } from './pages/Login/Login.jsx'
+import { Logout } from './pages/Logout/Logout.jsx'
 import { BrowserRouter as Router, Route, Switch } from 'react-router-dom'

 import { Notificacao } from './components/Notificacao/Notificacao.jsx'

 export function App() {
     return (
     <Notificacao>
         <Router>
             <Switch>
                 <Route path="/" exact={true}>
                     <Home />
                 </Route>
                 <Route path="/login">
                     <Login />
                 </Route>
+                <Route path="/logout">
+                    <Logout />
+                </Route>
             </Switch>
         </Router>
     </Notificacao>
     )
 }
```

2. No arquivo **`NavMenu.jsx`** na pasta **`src/components/NavMenu`** faça as seguintes alterações:

 # src/components/NavMenu/NavMenu.jsx

```
 import React from "react";
 import navMenuStyles from "./navMenu.module.css";

 import { Link } from 'react-router-dom'

 function NavMenu(props) {
```

```
    return (
      <nav className={navMenuStyles.navMenu}>
        <ul className={navMenuStyles.navMenu__lista}>
          <li className={navMenuStyles.navMenu__item}>
            <a className={navMenuStyles.navMenu__link} href="/">
              Bem vindo(a): <br />
              <strong>{props.usuario}</strong>
            </a>
          </li>
          <li className={navMenuStyles.navMenu__item}>
            <Link className={navMenuStyles.navMenu__link} to="/">
              Página Inicial
            </Link>
          </li>
          <li className={navMenuStyles.navMenu__item}>
            <a className={navMenuStyles.navMenu__link} href="/hashtags">
              Hashtags
            </a>
          </li>
          <li className={navMenuStyles.navMenu__item}>
-           <a className={navMenuStyles.navMenu__link} href="/logout">
+           <Link className={navMenuStyles.navMenu__link} to="/logout">
              Logout
-             {/* TODO roteamento e deslogamento */}
-           </a>
+           </Link>
          </li>
        </ul>
      </nav>
    );
  }

  export { NavMenu }
```

3. No arquivo **`AutenticarService.js`** na pasta **`src/model/services`** faça as seguintes alterações:

 # src/model/services/AutenticarService.js

```
  // React Scripts
  //  Webpack
  const URL_API = process.env.REACT_APP_URL_API

  export function getToken() {
      // Login inocente
      return localStorage.getItem('TOKEN')
  }

  export function isAutenticado() {
      // Login inocente
      return localStorage.getItem('TOKEN') !== null
  }

 +export async function desautenticar() {
 +    localStorage.removeItem('TOKEN')
 +}
 +
  export async function autenticar(usuario, senha) {
      const response = await fetch(URL_API + '/login', {
          method: "POST",
          headers: {
              "Content-Type": "application/json"
          },
```

```
            body: JSON.stringify({ login: usuario, senha})
        })

        if(!response.ok) {
            const erroServidor = await (response.json())
            const erroJS = Error(erroServidor.message)
            erroJS.status = response.status
            throw erroJS
        }

        const loginInfoServidor = await response.json()

        const tokenLogin = loginInfoServidor.token

        if(tokenLogin) {
            localStorage.setItem('TOKEN', tokenLogin)
            return tokenLogin
        }

        throw new Error("Token não encontrado")
    }
```

4. Crie o arquivo **`Logout.jsx`** na pasta **`src/pages/Logout`** com o seguinte código:

 # src/pages/Logout/Logout.jsx

```
+import React from "react"
+
+import * as AutenticarService from '../../model/services/AutenticarService.js'
+import { Redirect } from "react-router-dom"
+
+export function Logout() {
+
+    AutenticarService.desautenticar()
+
+    return (
+        <Redirect to="/login" />
+    )
+}
```

CAPÍTULO 17

EXERCÍCIO: PRÉ REDUX - O MODAL DE TWEETS – IMPLEMENTANDO O VISUAL.

17.1 PASSO A PASSO COM CÓDIGO

1. Crie o arquivo **`Modal.jsx`** na pasta **`src/components/Modal`** com o seguinte código:

 # src/components/Modal/Modal.jsx

```
+import React from "react"
+import PropTypes from "prop-types"
+
+import "./modal.css"
+import { Widget } from "../index.js"
+
+export function Modal(props) {
+    function handleBlackAreaClick(infosDoEvento) {
+        const isModalTag = infosDoEvento.target.classList.contains('modal')
+        if (isModalTag) props.onFechando && props.onFechando()
+    }
+
+    return (
+        <div
+            onClick={handleBlackAreaClick}
+            className='modal modalActive'
+        >
+            <div>
+                <Widget>{ props.children }</Widget>
+            </div>
+        </div>
+    )
+}
+
+Modal.propTypes = {
+    onFechando: PropTypes.func.isRequired,
+    children: PropTypes.element.isRequired
+}
```

2. No arquivo **`Tweet.jsx`** na pasta **`src/components/Tweet`** faça as seguintes alterações:

 # src/components/Tweet/Tweet.jsx

```
import React from 'react'

import './tweet.css'

export function TweetCabecalho({ nomeCompletoUsuario, nomeUsuario }) {
    return (
```

```
            <div className="tweet__cabecalho">
                <img className="tweet__fotoUsuario" src="https://placehold.it/50x50" alt="" />
                <span className="tweet__nomeUsuario">{ nomeCompletoUsuario }</span>
                <a href="/"><span className="tweet__userName">@{ nomeUsuario }</span></a>
            </div>
        )
    }

    function TweetFooter({ qtLikes }) {
        return (
            <footer className="tweet__footer">
                <button className="btn btn--clean">
                    <svg className="icon icon--small iconHeart" xmlns="http://www.w3.org/2000/svg" v
iewBox="0 0 47.5 47.5">
                        <defs>
                            <clipPath id="a">
                                <path d="M0 38h38V0H0v38z"></path>
                            </clipPath>
                        </defs>
                        <g clipPath="url(#a)" transform="matrix(1.25 0 0 -1.25 0 47.5)">
                            <path d="M36.885 25.166c0 5.45-4.418 9.868-9.867 9.868-3.308 0-6.227-1.6
32-8.018-4.128-1.79 2.496-4.71 4.129-8.017 4.129-5.45 0-9.868-4.418-9.868-9.868 0-.773.098-1.52.26
6-2.242C2.75 14.413 12.216 5.431 19 2.965c6.783 2.466 16.249 11.448 17.617 19.96.17.721.268 1.47.2
68 2.241"></path>
                        </g>
                    </svg>
                    { qtLikes }
                </button>
            </footer>
        )
    }

    // Componente Tweet
    export function Tweet(props) {

        const conteudo = props.children

        return (
            <article className="tweet">
                <TweetCabecalho
                    nomeCompletoUsuario={props.nomeCompletoUsuario}
                    nomeUsuario={props.nomeUsuario}
                />

-               <p className="tweet__conteudo">{ conteudo }</p>
+               <p onClick={props.onConteudoClicado} className="tweet__conteudo">{ conteudo }</p>

                <TweetFooter qtLikes={props.qtLikes} />

            </article>
        )
    }
```

3. No arquivo **`index.js`** na pasta **`src/components`** faça as seguintes alterações:

 # src/components/index.js

```
export { Cabecalho } from './Cabecalho/Cabecalho.jsx'
export { NavMenu } from './NavMenu/NavMenu.jsx'
export { Dashboard } from './Dashboard/Dashboard.jsx'
export { Widget } from './Widget/Widget.jsx'
export { TrendsArea } from './TrendsArea/TrendsArea.jsx'
```

```
 export { Tweet } from './Tweet/Tweet.jsx'
 export { ComponenteAutenticado } from './ComponenteAutenticado/ComponenteAutenticado.jsx'
 export { Form } from './Form/Form.jsx'
+export { Modal } from './Modal/Modal.jsx'
```

4. No arquivo **`Home.jsx`** na pasta **`src/pages/Home`** faça as seguintes alterações:

 # src/pages/Home/Home.jsx

```
 // Hook
 import React, { useState, useEffect, useContext} from 'react'
 import {
     Tweet,
     Cabecalho,
     NavMenu,
     Dashboard,
     Widget,
     TrendsArea,
     ComponenteAutenticado,
-    Form
+    Form,
+    Modal
 } from '../../components/index.js'

 import * as AutenticarService from '../../model/services/AutenticarService.js'
 import * as TweetsService from '../../model/services/TweetsService.js'

 import { Contexto as NotificacaoContexto } from '../../components/Notificacao/Notificacao.jsx'

 function converteTweet(tweet) {
     return {
         nomeUsuario: tweet.usuario.login,
         nomeCompletoUsuario: tweet.usuario.nome,
         qtLikes: tweet.totalLikes,
-        conteudo: tweet.conteudo
+        conteudo: tweet.conteudo,
+        id: tweet._id
     }
 }

 export function HomeSemAutenticacao() {
     const [ listaTweets, setListaTweets ] = useState([])

     const { setMsg } = useContext(NotificacaoContexto)

     useEffect(() => {
         TweetsService.carrega()
             .then(listaServidor => {
                 setListaTweets([
                     ...listaServidor.map(converteTweet),
                     ...listaTweets
                 ])
             })
     }, [])

     useEffect(() => {
         setTimeout(() => {
             setMsg("")
         }, 5000)
     })

-    // Função Closure
```

```
-     // Prop de callback
-     // Adicionar Tweets no servidor
+     const [tweetModal, setTweetModal] = useState(null)
+
+     function abreModal(tweet) {
+         setTweetModal(tweet)
+     }
+
+     function fechaModal() {
+         setTweetModal(null)
+     }
+
      function adicionaTweet(textoTweetNovo) {
          TweetsService.adiciona(textoTweetNovo)
              .then(novoTweet => {
                  setListaTweets([
                      converteTweet(novoTweet),
                      ...listaTweets
                  ])
              })
      }

      return (
          <div>
              <Cabecalho>
                 <NavMenu usuario="artadam"> </NavMenu>
              </Cabecalho>

              <div className="container">
                  <Dashboard>
                      <Widget>
                          <Form adicionaTweet={ adicionaTweet }/>
                      </Widget>
                      <Widget>
                          <TrendsArea></TrendsArea>
                      </Widget>
                  </Dashboard>

                  <Dashboard posicao="centro">
                      <Widget>
                          <div className="tweetsArea">

                              {listaTweets.map(infoTweet =>
-                                 <Tweet { ...infoTweet } key={infoTweet.id}>
+                                 <Tweet { ...infoTweet } key={infoTweet.id} onConteudoClicado={()
=> abreModal(infoTweet)}>
                                      { infoTweet.conteudo }
                                  </Tweet>
                              )}

                          </div>
                      </Widget>
                  </Dashboard>
              </div>
+
+             {tweetModal !== null
+                 ? (
+                     <Modal onFechando={fechaModal}>
+                         <Tweet {...tweetModal}>
+                             {tweetModal.conteudo}
+                         </Tweet>
+                     </Modal>
+                 )
```

```
+                : ''
+            }
+
          </div>
      )
  }

  export function Home(props) {
      const isLogado = AutenticarService.isAutenticado()

      return (
          <ComponenteAutenticado podeExibir={isLogado} redirecionarPara="/login" >
              <HomeSemAutenticacao {...props}/>
          </ComponenteAutenticado>
      )
  }
```

5. Adicione o arquivo **`modal.css`** na pasta **`src/components/Modal`** . Este arquivo é um arquivo que foi disponibilizado já pronto para você.

Capítulo 18

EXERCÍCIO: PRÉ REDUX - CURTINDO TWEETS. O PROBLEMA DA SINCRONIZAÇÃO DE ESTADOS.

18.1 PASSO A PASSO COM CÓDIGO

1. No arquivo **`Tweet.jsx`** na pasta **`src/components/Tweet`** faça as seguintes alterações:

 # src/components/Tweet/Tweet.jsx

```
  import React from 'react'

  import './tweet.css'

  export function TweetCabecalho({ nomeCompletoUsuario, nomeUsuario }) {
      return (
          <div className="tweet__cabecalho">
              <img className="tweet__fotoUsuario" src="https://placehold.it/50x50" alt="" />
              <span className="tweet__nomeUsuario">{ nomeCompletoUsuario }</span>
              <a href="/"><span className="tweet__userName">@{ nomeUsuario }</span></a>
          </div>
      )
  }

 -function TweetFooter({ qtLikes }) {
 +function TweetFooter({ qtLikes, likeado, onLike }) {
      return (
          <footer className="tweet__footer">
 -            <button className="btn btn--clean">
 -                <svg className="icon icon--small iconHeart" xmlns="http://www.w3.org/2000/svg" v
iewBox="0 0 47.5 47.5">
 +            <button className="btn btn--clean" onClick={ onLike }>
 +                <svg className={'icon icon--small iconHeart' + (likeado ? ' iconHeart--active' :
 '')} xmlns="http://www.w3.org/2000/svg" viewBox="0 0 47.5 47.5">
                      <defs>
                          <clipPath id="a">
                              <path d="M0 38h38V0H0v38z"></path>
                          </clipPath>
                      </defs>
                      <g clipPath="url(#a)" transform="matrix(1.25 0 0 -1.25 0 47.5)">
                          <path d="M36.885 25.166c0 5.45-4.418 9.868-9.867 9.868-3.308 0-6.227-1.6
32-8.018-4.128-1.79 2.496-4.71 4.129-8.017 4.129-5.45 0-9.868-4.418-9.868-9.868 0-.773.098-1.52.26
6-2.242C2.75 14.413 12.216 5.431 19 2.965c6.783 2.466 16.249 11.448 17.617 19.96.17.721.268 1.47.2
68 2.241"></path>
                      </g>
                  </svg>
                  { qtLikes }
```

```
            </button>
        </footer>
    )
}

// Componente Tweet
export function Tweet(props) {

    const conteudo = props.children

    return (
        <article className="tweet">
            <TweetCabecalho
                nomeCompletoUsuario={props.nomeCompletoUsuario}
                nomeUsuario={props.nomeUsuario}
            />

            <p onClick={props.onConteudoClicado} className="tweet__conteudo">{ conteudo }</p>

-           <TweetFooter qtLikes={props.qtLikes} />
+           {/*Problema: prop drilling, passando o onLike para o TweetFooter */}
+           <TweetFooter qtLikes={props.qtLikes} likeado={props.likeado} onLike= { props.onLike
} />

        </article>
    )
}
```

2. No arquivo **`Home.jsx`** na pasta **`src/pages/Home`** faça as seguintes alterações:

 # src/pages/Home/Home.jsx

```
// Hook
import React, { useState, useEffect, useContext} from 'react'
import {
    Tweet,
    Cabecalho,
    NavMenu,
    Dashboard,
    Widget,
    TrendsArea,
    ComponenteAutenticado,
    Form,
    Modal
} from '../../components/index.js'

import * as AutenticarService from '../../model/services/AutenticarService.js'
import * as TweetsService from '../../model/services/TweetsService.js'

import { Contexto as NotificacaoContexto } from '../../components/Notificacao/Notificacao.jsx'

function converteTweet(tweet) {
    return {
        nomeUsuario: tweet.usuario.login,
        nomeCompletoUsuario: tweet.usuario.nome,
        qtLikes: tweet.totalLikes,
        conteudo: tweet.conteudo,
+       likeado: false,
        id: tweet._id
    }
}
```

```
 export function HomeSemAutenticacao() {
     const [ listaTweets, setListaTweets ] = useState([])

     const { setMsg } = useContext(NotificacaoContexto)

     useEffect(() => {
         TweetsService.carrega()
             .then(listaServidor => {
                 setListaTweets([
                     ...listaServidor.map(converteTweet),
                     ...listaTweets
                 ])
             })
     }, [])

     useEffect(() => {
         setTimeout(() => {
             setMsg("")
         }, 5000)
     })

     const [tweetModal, setTweetModal] = useState(null)

     function abreModal(tweet) {
         setTweetModal(tweet)
     }

     function fechaModal() {
         setTweetModal(null)
     }

+    // Problema: Lógica de gerenciamento de estado no Componente
+    function dahLike(idTweetLikeado) {
+        const tweetLikeado = listaTweets.find(({id}) => id === idTweetLikeado)
+
+        if (tweetLikeado.likeado) {
+            tweetLikeado.likeado = false
+            tweetLikeado.qtLikes = tweetLikeado.qtLikes - 1
+        } else {
+            tweetLikeado.likeado = true
+            tweetLikeado.qtLikes = tweetLikeado.qtLikes + 1
+        }
+
+        setListaTweets([...listaTweets])
+    }
+
     function adicionaTweet(textoTweetNovo) {
         TweetsService.adiciona(textoTweetNovo)
             .then(novoTweet => {
                 setListaTweets([
                     converteTweet(novoTweet),
                     ...listaTweets
                 ])
             })
     }

     return (
         <div>
             <Cabecalho>
                <NavMenu usuario="artadam"> </NavMenu>
             </Cabecalho>

             <div className="container">
```

```
                <Dashboard>
                    <Widget>
                        <Form adicionaTweet={ adicionaTweet }/>
                    </Widget>
                    <Widget>
                        <TrendsArea></TrendsArea>
                    </Widget>
                </Dashboard>

                <Dashboard posicao="centro">
                    <Widget>
                        <div className="tweetsArea">

                            {listaTweets.map(infoTweet =>
-                               <Tweet { ...infoTweet } key={infoTweet.id} onConteudoClicado={() => abreModal(infoTweet)}>
+                               <Tweet { ...infoTweet } key={infoTweet.id} onConteudoClicado={() => abreModal(infoTweet)} onLike={() => dahLike(infoTweet.id)}>
                                    { infoTweet.conteudo }
                                </Tweet>
                            )}

                        </div>
                    </Widget>
                </Dashboard>
            </div>

            {tweetModal !== null
                ? (
                    <Modal onFechando={fechaModal}>
-                       <Tweet {...tweetModal}>
+                       <Tweet {...tweetModal} onLike={() => dahLike(tweetModal.id)}>
                            {tweetModal.conteudo}
                        </Tweet>
                    </Modal>
                )
                : ''
            }

        </div>
    )
}

export function Home(props) {
    const isLogado = AutenticarService.isAutenticado()

    return (
        <ComponenteAutenticado podeExibir={isLogado} redirecionarPara="/login" >
            <HomeSemAutenticacao {...props}/>
        </ComponenteAutenticado>
    )
}
```

Capítulo 19

EXERCÍCIO: GERENCIAMENTO DE ESTADO COM REDUX. A BASE.

19.1 PASSO A PASSO COM CÓDIGO

1. No arquivo **`package.json`** na pasta **`raíz do projeto`** faça as seguintes alterações:

 # package.json

```
  {
    "name": "twittelum",
    "version": "0.1.0",
    "private": true,
    "dependencies": {
      "react": "^16.12.0",
      "react-dom": "^16.12.0",
      "react-router-dom": "^5.1.2",
-     "react-scripts": "^3.3.0"
+     "react-scripts": "^3.3.0",
+     "redux": "^4.0.5"
    },
    "scripts": {
      "start": "react-scripts start",
      "_build:react": "react-scripts build",
      "build": "npm run _build:react"
    },
    "browserslist": {
      "production": [
        ">0.2%",
        "not dead",
        "not op_mini all"
      ],
      "development": [
        "last 1 chrome version",
        "last 1 firefox version",
        "last 1 safari version"
      ]
    }
  }
```

2. No arquivo **`Tweet.jsx`** na pasta **`src/components/Tweet`** faça as seguintes alterações:

 # src/components/Tweet/Tweet.jsx

```
  import React from 'react'

  import './tweet.css'
```

```
 +import { store } from '../../store.js'
 +
  export function TweetCabecalho({ nomeCompletoUsuario, nomeUsuario }) {
      return (
          <div className="tweet__cabecalho">
              <img className="tweet__fotoUsuario" src="https://placehold.it/50x50" alt="" />
              <span className="tweet__nomeUsuario">{ nomeCompletoUsuario }</span>
              <a href="/"><span className="tweet__userName">@{ nomeUsuario }</span></a>
          </div>
      )
  }

 -function TweetFooter({ qtLikes, likeado, onLike }) {
 +function TweetFooter({ qtLikes, likeado, onLike}) {
      return (
          <footer className="tweet__footer">
              <button className="btn btn--clean" onClick={ onLike }>
                  <svg className={'icon icon--small iconHeart' + (likeado ? ' iconHeart--active' :
 '')} xmlns="http://www.w3.org/2000/svg" viewBox="0 0 47.5 47.5">
                      <defs>
                          <clipPath id="a">
                              <path d="M0 38h38V0H0v38z"></path>
                          </clipPath>
                      </defs>
                      <g clipPath="url(#a)" transform="matrix(1.25 0 0 -1.25 0 47.5)">
                          <path d="M36.885 25.166c0 5.45-4.418 9.868-9.867 9.868-3.308 0-6.227-1.6
32-8.018-4.128-1.79 2.496-4.71 4.129-8.017 4.129-5.45 0-9.868-4.418-9.868-9.868 0-.773.098-1.52.26
6-2.242C2.75 14.413 12.216 5.431 19 2.965c6.783 2.466 16.249 11.448 17.617 19.96.17.721.268 1.47.2
68 2.241"></path>
                      </g>
                  </svg>
                  { qtLikes }
              </button>
          </footer>
      )
  }

  // Componente Tweet
  export function Tweet(props) {

      const conteudo = props.children

 +    function dahLike() {
 +        store.dispatch({type: "LIKE", id: props.id})
 +    }
 +
      return (
          <article className="tweet">
              <TweetCabecalho
                  nomeCompletoUsuario={props.nomeCompletoUsuario}
                  nomeUsuario={props.nomeUsuario}
              />

              <p onClick={props.onConteudoClicado} className="tweet__conteudo">{ conteudo }</p>

              {/*Problema: prop drilling, passando o onLike para o TweetFooter */}
 -            <TweetFooter qtLikes={props.qtLikes} likeado={props.likeado} onLike= {props.onLike
 } />
 +            <TweetFooter qtLikes={props.qtLikes} likeado={props.likeado} onLike={ dahLike }/>

          </article>
      )
  }
```

3. No arquivo **`Home.jsx`** na pasta **`src/pages/Home`** faça as seguintes alterações:

 # src/pages/Home/Home.jsx

```
 // Hook
 import React, { useState, useEffect, useContext} from 'react'
 import {
     Tweet,
     Cabecalho,
     NavMenu,
     Dashboard,
     Widget,
     TrendsArea,
     ComponenteAutenticado,
     Form,
     Modal
 } from '../../components/index.js'

 import * as AutenticarService from '../../model/services/AutenticarService.js'
 import * as TweetsService from '../../model/services/TweetsService.js'

 import { Contexto as NotificacaoContexto } from '../../components/Notificacao/Notificacao.jsx'

-function converteTweet(tweet) {
-    return {
-        nomeUsuario: tweet.usuario.login,
-        nomeCompletoUsuario: tweet.usuario.nome,
-        qtLikes: tweet.totalLikes,
-        conteudo: tweet.conteudo,
-        likeado: false,
-        id: tweet._id
-    }
-}
+import { store } from '../../store.js'

 export function HomeSemAutenticacao() {
     const [ listaTweets, setListaTweets ] = useState([])

     const { setMsg } = useContext(NotificacaoContexto)

+    store.subscribe(() => {
+        setListaTweets(store.getState().listaTweets)
+    })
+
     useEffect(() => {
         TweetsService.carrega()
             .then(listaServidor => {
-                setListaTweets([
-                    ...listaServidor.map(converteTweet),
-                    ...listaTweets
-                ])
+                store.dispatch({
+                    type: "LISTA",
+                    lista: listaServidor
+                })
             })
     }, [])

     useEffect(() => {
         setTimeout(() => {
             setMsg("")
         }, 5000)
```

```
    })

    const [tweetModal, setTweetModal] = useState(null)

    function abreModal(tweet) {
        setTweetModal(tweet)
    }

    function fechaModal() {
        setTweetModal(null)
    }

-   // Problema: Lógica de gerenciamento de estado no Componente
-   function dahLike(idTweetLikeado) {
-       const tweetLikeado = listaTweets.find(({id}) => id === idTweetLikeado)
-
-       if (tweetLikeado.likeado) {
-           tweetLikeado.likeado = false
-           tweetLikeado.qtLikes = tweetLikeado.qtLikes - 1
-       } else {
-           tweetLikeado.likeado = true
-           tweetLikeado.qtLikes = tweetLikeado.qtLikes + 1
-       }
-
-       setListaTweets([...listaTweets])
-   }
-
    function adicionaTweet(textoTweetNovo) {
        TweetsService.adiciona(textoTweetNovo)
            .then(novoTweet => {
-               setListaTweets([
-                   converteTweet(novoTweet),
-                   ...listaTweets
-               ])
+               store.dispatch({
+                   type: "ADICIONA",
+                   tweet: novoTweet
+               })
            })
    }

    return (
        <div>
            <Cabecalho>
               <NavMenu usuario="artadam"> </NavMenu>
            </Cabecalho>

            <div className="container">
                <Dashboard>
                    <Widget>
                        <Form adicionaTweet={ adicionaTweet }/>
                    </Widget>
                    <Widget>
                        <TrendsArea></TrendsArea>
                    </Widget>
                </Dashboard>

                <Dashboard posicao="centro">
                    <Widget>
                        <div className="tweetsArea">

                            {listaTweets.map(infoTweet =>
-                               <Tweet { ...infoTweet } key={infoTweet.id} onConteudoClicado={()
```

```
=> abreModal(infoTweet)} onLike={() => dahLike(infoTweet.id)}>
+                                   <Tweet { ...infoTweet } key={infoTweet.id} onConteudoClicado={()
=> abreModal(infoTweet)} >
                                       { infoTweet.conteudo }
                                   </Tweet>
                               )}

                           </div>
                       </Widget>
                   </Dashboard>
               </div>

               {tweetModal !== null
                   ? (
                       <Modal onFechando={fechaModal}>
-                          <Tweet {...tweetModal} onLike={() => dahLike(tweetModal.id)}>
+                          <Tweet {...tweetModal} >
                               {tweetModal.conteudo}
                           </Tweet>
                       </Modal>
                   )
                   : ''
               }

           </div>
       )
   }

   export function Home(props) {
       const isLogado = AutenticarService.isAutenticado()

       return (
           <ComponenteAutenticado podeExibir={isLogado} redirecionarPara="/login" >
               <HomeSemAutenticacao {...props}/>
           </ComponenteAutenticado>
       )
   }
```

4. Crie o arquivo **`store.js`** na pasta **`src`** com o seguinte código:

 # src/store.js

```
+import { createStore } from 'redux'
+
+const ESTADO_INICIAL = { listaTweets: [] }
+
+function converteTweet(tweet) {
+    return {
+        nomeUsuario: tweet.usuario.login,
+        nomeCompletoUsuario: tweet.usuario.nome,
+        qtLikes: tweet.totalLikes,
+        conteudo: tweet.conteudo,
+        likeado: false,
+        id: tweet._id
+    }
+}
+
+export const store = createStore(function reducer(estado = ESTADO_INICIAL, acao) {
+
+    if (acao.type === "LISTA") {
+        return {
+            listaTweets: acao.lista.map(converteTweet)
```

```
+        }
+    }
+
+    if (acao.type === "LIKE") {
+        const tweetLikeado = estado.listaTweets.find(({id}) => id === acao.id)
+
+        if (tweetLikeado.likeado) {
+            tweetLikeado.likeado = false
+            tweetLikeado.qtLikes = tweetLikeado.qtLikes - 1
+        } else {
+            tweetLikeado.likeado = true
+            tweetLikeado.qtLikes = tweetLikeado.qtLikes + 1
+        }
+
+        return {
+            listaTweets: [...estado.listaTweets]
+        }
+    }
+
+    if (acao.type === "ADICIONA") {
+        return {
+            listaTweets: [
+                converteTweet(acao.tweet),
+                ...estado.listaTweets
+            ]
+        }
+    }
+
+    return estado
+})
```

CAPÍTULO 20

EXERCÍCIO: ORGANIZANDO NOSSO CÓDIGO REDUX – EVITANDO DUPLICAÇÃO NA CRIAÇÃO DE ACTIONS COM ACTION CREATORS.

20.1 PASSO A PASSO COM CÓDIGO

1. No arquivo **`Tweet.jsx`** na pasta **`src/components/Tweet`** faça as seguintes alterações:

 # src/components/Tweet/Tweet.jsx

```
 import React from 'react'

 import './tweet.css'

-import { store } from '../../store.js'
+import { store, criaAcaoLike } from '../../store.js'

 export function TweetCabecalho({ nomeCompletoUsuario, nomeUsuario }) {
     return (
         <div className="tweet__cabecalho">
             <img className="tweet__fotoUsuario" src="https://placehold.it/50x50" alt="" />
             <span className="tweet__nomeUsuario">{ nomeCompletoUsuario }</span>
             <a href="/"><span className="tweet__userName">@{ nomeUsuario }</span></a>
         </div>
     )
 }

 function TweetFooter({ qtLikes, likeado, onLike}) {
     return (
         <footer className="tweet__footer">
             <button className="btn btn--clean" onClick={ onLike }>
                 <svg className={'icon icon--small iconHeart' + (likeado ? ' iconHeart--active' :
''')} xmlns="http://www.w3.org/2000/svg" viewBox="0 0 47.5 47.5">
                     <defs>
                         <clipPath id="a">
                             <path d="M0 38h38V0H0v38z"></path>
                         </clipPath>
                     </defs>
                     <g clipPath="url(#a)" transform="matrix(1.25 0 0 -1.25 0 47.5)">
                         <path d="M36.885 25.166c0 5.45-4.418 9.868-9.867 9.868-3.308 0-6.227-1.6
32-8.018-4.128-1.79 2.496-4.71 4.129-8.017 4.129-5.45 0-9.868-4.418-9.868-9.868 0-.773.098-1.52.26
6-2.242C2.75 14.413 12.216 5.431 19 2.965c6.783 2.466 16.249 11.448 17.617 19.96.17.721.268 1.47.2
68 2.241"></path>
                     </g>
```

```
                </svg>
                { qtLikes }
            </button>
        </footer>
    )
}

// Componente Tweet
export function Tweet(props) {

    const conteudo = props.children

    function dahLike() {
-       store.dispatch({type: "LIKE", id: props.id})
+       store.dispatch(criaAcaoLike(props.id))
    }

    return (
        <article className="tweet">
            <TweetCabecalho
                nomeCompletoUsuario={props.nomeCompletoUsuario}
                nomeUsuario={props.nomeUsuario}
            />

            <p onClick={props.onConteudoClicado} className="tweet__conteudo">{ conteudo }</p>

            {/*Problema: prop drilling, passando o onLike para o TweetFooter */}
            <TweetFooter qtLikes={props.qtLikes} likeado={props.likeado} onLike={ dahLike }/>

        </article>
    )
}
```

2. No arquivo **`Home.jsx`** na pasta **`src/pages/Home`** faça as seguintes alterações:

 # src/pages/Home/Home.jsx

```
// Hook
import React, { useState, useEffect, useContext} from 'react'
import {
    Tweet,
    Cabecalho,
    NavMenu,
    Dashboard,
    Widget,
    TrendsArea,
    ComponenteAutenticado,
    Form,
    Modal
} from '../../components/index.js'

import * as AutenticarService from '../../model/services/AutenticarService.js'
import * as TweetsService from '../../model/services/TweetsService.js'

import { Contexto as NotificacaoContexto } from '../../components/Notificacao/Notificacao.jsx'

-import { store } from '../../store.js'
+import { store, criaAcaoAdiciona, criaAcaoLista } from '../../store.js'

export function HomeSemAutenticacao() {
    const [ listaTweets, setListaTweets ] = useState([])
```

```
    const { setMsg } = useContext(NotificacaoContexto)

    store.subscribe(() => {
        setListaTweets(store.getState().listaTweets)
    })

    useEffect(() => {
        TweetsService.carrega()
            .then(listaServidor => {
-               store.dispatch({
-                   type: "LISTA",
-                   lista: listaServidor
-               })
+               store.dispatch(criaAcaoLista(listaServidor))
            })
    }, [])

    useEffect(() => {
        setTimeout(() => {
            setMsg("")
        }, 5000)
    })

    const [tweetModal, setTweetModal] = useState(null)

    function abreModal(tweet) {
        setTweetModal(tweet)
    }

    function fechaModal() {
        setTweetModal(null)
    }

    function adicionaTweet(textoTweetNovo) {
        TweetsService.adiciona(textoTweetNovo)
            .then(novoTweet => {
-               store.dispatch({
-                   type: "ADICIONA",
-                   tweet: novoTweet
-               })
+               store.dispatch(criaAcaoAdiciona(novoTweet))
            })
    }

    return (
        <div>
            <Cabecalho>
               <NavMenu usuario="artadam"> </NavMenu>
            </Cabecalho>

            <div className="container">
                <Dashboard>
                    <Widget>
                        <Form adicionaTweet={ adicionaTweet }/>
                    </Widget>
                    <Widget>
                        <TrendsArea></TrendsArea>
                    </Widget>
                </Dashboard>

                <Dashboard posicao="centro">
                    <Widget>
                        <div className="tweetsArea">
```

```
                                {listaTweets.map(infoTweet =>
                                    <Tweet { ...infoTweet } key={infoTweet.id} onConteudoClicado={()
=> abreModal(infoTweet)} >
                                        { infoTweet.conteudo }
                                    </Tweet>
                                )}

                            </div>
                        </Widget>
                    </Dashboard>
                </div>

                {tweetModal !== null
                    ? (
                        <Modal onFechando={fechaModal}>
                            <Tweet {...tweetModal} >
                                {tweetModal.conteudo}
                            </Tweet>
                        </Modal>
                    )
                    : ''
                }

            </div>
        )
    }

    export function Home(props) {
        const isLogado = AutenticarService.isAutenticado()

        return (
            <ComponenteAutenticado podeExibir={isLogado} redirecionarPara="/login" >
                <HomeSemAutenticacao {...props}/>
            </ComponenteAutenticado>
        )
    }
```

3. No arquivo **`store.js`** na pasta **`src`** faça as seguintes alterações:

 # src/store.js

```
    import { createStore } from 'redux'

    const ESTADO_INICIAL = { listaTweets: [] }

    function converteTweet(tweet) {
        return {
            nomeUsuario: tweet.usuario.login,
            nomeCompletoUsuario: tweet.usuario.nome,
            qtLikes: tweet.totalLikes,
            conteudo: tweet.conteudo,
            likeado: false,
            id: tweet._id
        }
    }

    export const store = createStore(function reducer(estado = ESTADO_INICIAL, acao) {

        if (acao.type === "LISTA") {
            return {
                listaTweets: acao.lista.map(converteTweet)
```

```
        }
    }

    if (acao.type === "LIKE") {
        const tweetLikeado = estado.listaTweets.find(({id}) => id === acao.id)

        if (tweetLikeado.likeado) {
            tweetLikeado.likeado = false
            tweetLikeado.qtLikes = tweetLikeado.qtLikes - 1
        } else {
            tweetLikeado.likeado = true
            tweetLikeado.qtLikes = tweetLikeado.qtLikes + 1
        }

        return {
            listaTweets: [...estado.listaTweets]
        }
    }

    if (acao.type === "ADICIONA") {
        return {
            listaTweets: [
                converteTweet(acao.tweet),
                ...estado.listaTweets
            ]
        }
    }

    return estado
})
+
+
+// Action Creators
+export const criaAcaoLista = (tweets) => {
+    return {
+        type: "LISTA",
+        lista: tweets
+    }
+}
+
+export const criaAcaoLike = (id) => {
+    return {
+        type: "LIKE",
+        id: id
+    }
+}
+
+export const criaAcaoAdiciona = (tweet) => {
+    return {
+        type: "ADICIONA",
+        tweet: tweet
+    }
+}
```

CAPÍTULO 21

EXERCÍCIO: ORGANIZANDO NOSSO CÓDIGO REDUX – AÇÕES ASSÍNCRONAS COM MIDDLEWARE DE THUNK PARA O REDUX.

21.1 PASSO A PASSO COM CÓDIGO

1. No arquivo **`package.json`** na pasta **`raíz do projeto`** faça as seguintes alterações:

 # package.json

```
 {
   "name": "twittelum",
   "version": "0.1.0",
   "private": true,
   "dependencies": {
     "react": "^16.12.0",
     "react-dom": "^16.12.0",
     "react-router-dom": "^5.1.2",
     "react-scripts": "^3.3.0",
-    "redux": "^4.0.5"
+    "redux": "^4.0.5",
+    "redux-thunk": "^2.3.0"
   },
   "scripts": {
     "start": "react-scripts start",
     "_build:react": "react-scripts build",
     "build": "npm run _build:react"
   },
   "browserslist": {
     "production": [
       ">0.2%",
       "not dead",
       "not op_mini all"
     ],
     "development": [
       "last 1 chrome version",
       "last 1 firefox version",
       "last 1 safari version"
     ]
   }
 }
```

2. No arquivo **`Home.jsx`** na pasta **`src/pages/Home`** faça as seguintes alterações:

 # src/pages/Home/Home.jsx

```
 // Hook
 import React, { useState, useEffect, useContext} from 'react'
 import {
     Tweet,
     Cabecalho,
     NavMenu,
     Dashboard,
     Widget,
     TrendsArea,
     ComponenteAutenticado,
     Form,
     Modal
 } from '../../components/index.js'

 import * as AutenticarService from '../../model/services/AutenticarService.js'
-import * as TweetsService from '../../model/services/TweetsService.js'

 import { Contexto as NotificacaoContexto } from '../../components/Notificacao/Notificacao.jsx'

-import { store, criaAcaoAdiciona, criaAcaoLista } from '../../store.js'
+import { store, criaAcaoAdicionarServidor, criaAcaoCarregarServidor } from '../../store.js'

 export function HomeSemAutenticacao() {
     const [ listaTweets, setListaTweets ] = useState([])

     const { setMsg } = useContext(NotificacaoContexto)

     store.subscribe(() => {
         setListaTweets(store.getState().listaTweets)
     })

     useEffect(() => {
-        TweetsService.carrega()
-            .then(listaServidor => {
-                store.dispatch(criaAcaoLista(listaServidor))
-            })
+        store.dispatch(criaAcaoCarregarServidor())
     }, [])

     useEffect(() => {
         setTimeout(() => {
             setMsg("")
         }, 5000)
     })

     const [tweetModal, setTweetModal] = useState(null)

     function abreModal(tweet) {
         setTweetModal(tweet)
     }

     function fechaModal() {
         setTweetModal(null)
     }

     function adicionaTweet(textoTweetNovo) {
-        TweetsService.adiciona(textoTweetNovo)
-            .then(novoTweet => {
-                store.dispatch(criaAcaoAdiciona(novoTweet))
-            })
```

```
+         store.dispatch(criaAcaoAdicionarServidor(textoTweetNovo))
      }

      return (
          <div>
              <Cabecalho>
                 <NavMenu usuario="artadam"> </NavMenu>
              </Cabecalho>

              <div className="container">
                  <Dashboard>
                      <Widget>
                          <Form adicionaTweet={ adicionaTweet }/>
                      </Widget>
                      <Widget>
                          <TrendsArea></TrendsArea>
                      </Widget>
                  </Dashboard>

                  <Dashboard posicao="centro">
                      <Widget>
                          <div className="tweetsArea">

                              {listaTweets.map(infoTweet =>
                                  <Tweet { ...infoTweet } key={infoTweet.id} onConteudoClicado={()
=> abreModal(infoTweet)} >
                                      { infoTweet.conteudo }
                                  </Tweet>
                              )}

                          </div>
                      </Widget>
                  </Dashboard>
              </div>

              {tweetModal !== null
                  ? (
                      <Modal onFechando={fechaModal}>
                          <Tweet {...tweetModal} >
                              {tweetModal.conteudo}
                          </Tweet>
                      </Modal>
                  )
                  : ''
              }

          </div>
      )
  }

  export function Home(props) {
      const isLogado = AutenticarService.isAutenticado()

      return (
          <ComponenteAutenticado podeExibir={isLogado} redirecionarPara="/login" >
              <HomeSemAutenticacao {...props}/>
          </ComponenteAutenticado>
      )
  }
```

3. No arquivo **`store.js`** na pasta **`src`** faça as seguintes alterações:

 # src/store.js

```
-import { createStore } from 'redux'
+import { createStore, applyMiddleware } from 'redux'
+
+import thunkMiddleware from 'redux-thunk'
+
+import * as TweetsService from './model/services/TweetsService.js'

 const ESTADO_INICIAL = { listaTweets: [] }

 function converteTweet(tweet) {
     return {
         nomeUsuario: tweet.usuario.login,
         nomeCompletoUsuario: tweet.usuario.nome,
         qtLikes: tweet.totalLikes,
         conteudo: tweet.conteudo,
         likeado: false,
         id: tweet._id
     }
 }

-export const store = createStore(function reducer(estado = ESTADO_INICIAL, acao) {
+export const store = createStore(
+    reducer,
+    applyMiddleware(
+        thunkMiddleware
+    )
+)
+
+function reducer(estado = ESTADO_INICIAL, acao) {

     if (acao.type === "LISTA") {
         return {
             listaTweets: acao.lista.map(converteTweet)
         }
     }

     if (acao.type === "LIKE") {
         const tweetLikeado = estado.listaTweets.find(({id}) => id === acao.id)

         if (tweetLikeado.likeado) {
             tweetLikeado.likeado = false
             tweetLikeado.qtLikes = tweetLikeado.qtLikes - 1
         } else {
             tweetLikeado.likeado = true
             tweetLikeado.qtLikes = tweetLikeado.qtLikes + 1
         }

         return {
             listaTweets: [...estado.listaTweets]
         }
     }

     if (acao.type === "ADICIONA") {
         return {
             listaTweets: [
                 converteTweet(acao.tweet),
                 ...estado.listaTweets
             ]
         }
     }
```

```
     return estado
-})
+}

 // Action Creators
 export const criaAcaoLista = (tweets) => {
     return {
         type: "LISTA",
         lista: tweets
     }
 }

 export const criaAcaoLike = (id) => {
     return {
         type: "LIKE",
         id: id
     }
 }

 export const criaAcaoAdiciona = (tweet) => {
     return {
         type: "ADICIONA",
         tweet: tweet
     }
 }
+
+//Thunk Action Creator
+export const criaAcaoAdicionarServidor = (textoTweetNovo) => {
+    return (dispatch) => {
+        TweetsService
+            .adiciona(textoTweetNovo)
+            .then(novoTweet => {
+                dispatch(criaAcaoAdiciona(novoTweet))
+            })
+    }
+}
+
+//Thunk Action Creator
+export const criaAcaoCarregarServidor = () => {
+    return (dispatch) => {
+        TweetsService
+            .carrega()
+            .then((listaServidor) => {
+                dispatch(criaAcaoLista(listaServidor))
+            })
+    }
+}
```

CAPÍTULO 22

EXERCÍCIO: ORGANIZANDO NOSSO CÓDIGO REDUX – SEPARAÇÃO DE RESPONSABILDIADES E MODULARIZAÇÃO DO CÓDIGO REDUX COM O PADRÃO DUCKS.

22.1 PASSO A PASSO COM CÓDIGO

1. No arquivo **`Tweet.jsx`** na pasta **`src/components/Tweet`** faça as seguintes alterações:

 # src/components/Tweet/Tweet.jsx

```
 import React from 'react'

 import './tweet.css'

-import { store, criaAcaoLike } from '../../store.js'
+import { store } from '../../store.js'
+import { criaAcaoLike } from '../../ducks/listaTweets.js'

 export function TweetCabecalho({ nomeCompletoUsuario, nomeUsuario }) {
     return (
         <div className="tweet__cabecalho">
             <img className="tweet__fotoUsuario" src="https://placehold.it/50x50" alt="" />
             <span className="tweet__nomeUsuario">{ nomeCompletoUsuario }</span>
             <a href="/"><span className="tweet__userName">@{ nomeUsuario }</span></a>
         </div>
     )
 }

 function TweetFooter({ qtLikes, likeado, onLike}) {
     return (
         <footer className="tweet__footer">
             <button className="btn btn--clean" onClick={ onLike }>
                 <svg className={'icon icon--small iconHeart' + (likeado ? ' iconHeart--active' :
''')} xmlns="http://www.w3.org/2000/svg" viewBox="0 0 47.5 47.5">
                     <defs>
                         <clipPath id="a">
                             <path d="M0 38h38V0H0v38z"></path>
                         </clipPath>
                     </defs>
                     <g clipPath="url(#a)" transform="matrix(1.25 0 0 -1.25 0 47.5)">
                         <path d="M36.885 25.166c0 5.45-4.418 9.868-9.867 9.868-3.308 0-6.227-1.6
32-8.018-4.128-1.79 2.496-4.71 4.129-8.017 4.129-5.45 0-9.868-4.418-9.868-9.868 0-.773.098-1.52.26
```

```
6-2.242C2.75 14.413 12.216 5.431 19 2.965c6.783 2.466 16.249 11.448 17.617 19.96.17.721.268 1.47.2
68 2.241"></path>
                    </g>
                </svg>
                { qtLikes }
            </button>
        </footer>
    )
}

// Componente Tweet
export function Tweet(props) {

    const conteudo = props.children

    function dahLike() {
        store.dispatch(criaAcaoLike(props.id))
    }

    return (
        <article className="tweet">
            <TweetCabecalho
                nomeCompletoUsuario={props.nomeCompletoUsuario}
                nomeUsuario={props.nomeUsuario}
            />

            <p onClick={props.onConteudoClicado} className="tweet__conteudo">{ conteudo }</p>

            {/*Problema: prop drilling, passando o onLike para o TweetFooter */}
            <TweetFooter qtLikes={props.qtLikes} likeado={props.likeado} onLike={ dahLike }/>

        </article>
    )
}
```

2. Crie o arquivo **`listaTweets.js`** na pasta **`src/ducks`** com o seguinte código:

 # src/ducks/listaTweets.js

```
+import * as TweetsService from '../model/services/TweetsService.js'
+
+const LISTA_TWEETS_INICIAL = []
+
+function converteTweet(tweet) {
+    return {
+        nomeUsuario: tweet.usuario.login,
+        nomeCompletoUsuario: tweet.usuario.nome,
+        qtLikes: tweet.totalLikes,
+        conteudo: tweet.conteudo,
+        likeado: false,
+        id: tweet._id
+    }
+}
+
+export function listaTweetsReducer(listaTweets = LISTA_TWEETS_INICIAL, acao) {
+
+    if (acao.type === "LISTA") {
+        return acao.lista.map(converteTweet)
+    }
+
+    if (acao.type === "LIKE") {
+        const tweetLikeado = listaTweets.find(({id}) => id === acao.id)
```

```
+
+        if (tweetLikeado.likeado) {
+            tweetLikeado.likeado = false
+            tweetLikeado.qtLikes = tweetLikeado.qtLikes - 1
+        } else {
+            tweetLikeado.likeado = true
+            tweetLikeado.qtLikes = tweetLikeado.qtLikes + 1
+        }
+
+        return [...listaTweets]
+    }
+
+    if (acao.type === "ADICIONA") {
+        return [
+            converteTweet(acao.tweet),
+            ...listaTweets
+        ]
+    }
+
+    return listaTweets
+}
+
+
+// Action Creators
+export const criaAcaoLista = (tweets) => {
+    return {
+        type: "LISTA",
+        lista: tweets
+    }
+}
+
+export const criaAcaoLike = (id) => {
+    return {
+        type: "LIKE",
+        id: id
+    }
+}
+
+export const criaAcaoAdiciona = (tweet) => {
+    return {
+        type: "ADICIONA",
+        tweet: tweet
+    }
+}
+
+//Thunk Action Creator
+export const criaAcaoAdicionarServidor = (textoTweetNovo) => {
+    return (dispatch) => {
+        TweetsService
+            .adiciona(textoTweetNovo)
+            .then(novoTweet => {
+                dispatch(criaAcaoAdiciona(novoTweet))
+            })
+    }
+}
+
+//Thunk Action Creator
+export const criaAcaoCarregarServidor = () => {
+    return (dispatch) => {
+        TweetsService
+            .carrega()
+            .then((listaServidor) => {
+                dispatch(criaAcaoLista(listaServidor))
```

```
+            })
+    }
+}
```

3. No arquivo **`Home.jsx`** na pasta **`src/pages/Home`** faça as seguintes alterações:

 # src/pages/Home/Home.jsx

```
 // Hook
 import React, { useState, useEffect, useContext} from 'react'
 import {
     Tweet,
     Cabecalho,
     NavMenu,
     Dashboard,
     Widget,
     TrendsArea,
     ComponenteAutenticado,
     Form,
     Modal
 } from '../../components/index.js'

 import * as AutenticarService from '../../model/services/AutenticarService.js'

 import { Contexto as NotificacaoContexto } from '../../components/Notificacao/Notificacao.jsx'

-import { store, criaAcaoAdicionarServidor, criaAcaoCarregarServidor } from '../../store.js'
+import { store } from  '../../store.js'
+
+import { criaAcaoAdicionarServidor, criaAcaoCarregarServidor } from '../../ducks/listaTweets.js'

 export function HomeSemAutenticacao() {
     const [ listaTweets, setListaTweets ] = useState([])

     const { setMsg } = useContext(NotificacaoContexto)

     store.subscribe(() => {
         setListaTweets(store.getState().listaTweets)
     })

     useEffect(() => {
         store.dispatch(criaAcaoCarregarServidor())
     }, [])

     useEffect(() => {
         setTimeout(() => {
             setMsg("")
         }, 5000)
     })

     const [tweetModal, setTweetModal] = useState(null)

     function abreModal(tweet) {
         setTweetModal(tweet)
     }

     function fechaModal() {
         setTweetModal(null)
     }

     function adicionaTweet(textoTweetNovo) {
         store.dispatch(criaAcaoAdicionarServidor(textoTweetNovo))
```

```
    }

    return (
        <div>
            <Cabecalho>
               <NavMenu usuario="artadam"> </NavMenu>
            </Cabecalho>

            <div className="container">
                <Dashboard>
                    <Widget>
                        <Form adicionaTweet={ adicionaTweet }/>
                    </Widget>
                    <Widget>
                        <TrendsArea></TrendsArea>
                    </Widget>
                </Dashboard>

                <Dashboard posicao="centro">
                    <Widget>
                        <div className="tweetsArea">

                            {listaTweets.map(infoTweet =>
                                <Tweet { ...infoTweet } key={infoTweet.id} onConteudoClicado={()
=> abreModal(infoTweet)} >
                                    { infoTweet.conteudo }
                                </Tweet>
                            )}

                        </div>
                    </Widget>
                </Dashboard>
            </div>

            {tweetModal !== null
                ? (
                    <Modal onFechando={fechaModal}>
                        <Tweet {...tweetModal} >
                            {tweetModal.conteudo}
                        </Tweet>
                    </Modal>
                )
                : ''
            }

        </div>
    )
}

export function Home(props) {
    const isLogado = AutenticarService.isAutenticado()

    return (
        <ComponenteAutenticado podeExibir={isLogado} redirecionarPara="/login" >
            <HomeSemAutenticacao {...props}/>
        </ComponenteAutenticado>
    )
}
```

4. No arquivo **`store.js`** na pasta **`src`** faça as seguintes alterações:

 # src/store.js

```
-import { createStore, applyMiddleware } from 'redux'
-
+import { createStore, applyMiddleware, combineReducers} from 'redux'
 import thunkMiddleware from 'redux-thunk'

-import * as TweetsService from './model/services/TweetsService.js'
-
-const ESTADO_INICIAL = { listaTweets: [] }
-
-function converteTweet(tweet) {
-    return {
-        nomeUsuario: tweet.usuario.login,
-        nomeCompletoUsuario: tweet.usuario.nome,
-        qtLikes: tweet.totalLikes,
-        conteudo: tweet.conteudo,
-        likeado: false,
-        id: tweet._id
-    }
-}
+import { listaTweetsReducer } from './ducks/listaTweets.js'

 export const store = createStore(
-    reducer,
+    combineReducers({
+        listaTweets: listaTweetsReducer
+    }),
     applyMiddleware(
         thunkMiddleware
     )
 )
-
-function reducer(estado = ESTADO_INICIAL, acao) {
-
-    if (acao.type === "LISTA") {
-        return {
-            listaTweets: acao.lista.map(converteTweet)
-        }
-    }
-
-    if (acao.type === "LIKE") {
-        const tweetLikeado = estado.listaTweets.find(({id}) => id === acao.id)
-
-        if (tweetLikeado.likeado) {
-            tweetLikeado.likeado = false
-            tweetLikeado.qtLikes = tweetLikeado.qtLikes - 1
-        } else {
-            tweetLikeado.likeado = true
-            tweetLikeado.qtLikes = tweetLikeado.qtLikes + 1
-        }
-
-        return {
-            listaTweets: [...estado.listaTweets]
-        }
-    }
-
-    if (acao.type === "ADICIONA") {
-        return {
-            listaTweets: [
-                converteTweet(acao.tweet),
-                ...estado.listaTweets
-            ]
-        }
-    }
```

```
-
-     return estado
-}
-
-
-// Action Creators
-export const criaAcaoLista = (tweets) => {
-     return {
-         type: "LISTA",
-         lista: tweets
-     }
-}
-
-export const criaAcaoLike = (id) => {
-     return {
-         type: "LIKE",
-         id: id
-     }
-}
-
-export const criaAcaoAdiciona = (tweet) => {
-     return {
-         type: "ADICIONA",
-         tweet: tweet
-     }
-}
-
-//Thunk Action Creator
-export const criaAcaoAdicionarServidor = (textoTweetNovo) => {
-     return (dispatch) => {
-         TweetsService
-             .adiciona(textoTweetNovo)
-             .then(novoTweet => {
-                 dispatch(criaAcaoAdiciona(novoTweet))
-             })
-     }
-}
-
-//Thunk Action Creator
-export const criaAcaoCarregarServidor = () => {
-     return (dispatch) => {
-         TweetsService
-             .carrega()
-             .then((listaServidor) => {
-                 dispatch(criaAcaoLista(listaServidor))
-             })
-     }
-}
```

CAPÍTULO 23

EXERCÍCIO: ORGANIZANDO NOSSO CÓDIGO REDUX – COMPARTILHANDO STORE COMO CONTEXTO.

23.1 PASSO A PASSO COM CÓDIGO

1. No arquivo **`App.jsx`** na pasta **`src`** faça as seguintes alterações:

 # src/App.jsx

```
 import React from 'react'

 import { Home } from './pages/Home/Home.jsx'
 import { Login } from './pages/Login/Login.jsx'
 import { Logout } from './pages/Logout/Logout.jsx'
 import { BrowserRouter as Router, Route, Switch } from 'react-router-dom'

-import { Notificacao } from './components/Notificacao/Notificacao.jsx'
+import {
+    Notificacao,
+    ReduxStoreProvider
+} from './components/index.js'

 export function App() {
     return (
+    <ReduxStoreProvider>
         <Notificacao>
             <Router>
                 <Switch>
                     <Route path="/" exact={true}>
                         <Home />
                     </Route>
                     <Route path="/login">
                         <Login />
                     </Route>
                     <Route path="/logout">
                         <Logout />
                     </Route>
                 </Switch>
             </Router>
         </Notificacao>
+    </ReduxStoreProvider>
     )
 }
```

2. Crie o arquivo **`ReduxStoreProvider.jsx`** na pasta **`src/components/ReduxStoreProvider`** com

o seguinte código:

 # src/components/ReduxStoreProvider/ReduxStoreProvider.jsx

```
+import React, { createContext } from 'react'
+import { createStore, applyMiddleware, combineReducers} from 'redux'
+import thunkMiddleware from 'redux-thunk'
+
+import { listaTweetsReducer } from '../../ducks/listaTweets.js'
+
+export const StoreContexto = createContext({store: null})
+
+export function ReduxStoreProvider(props) {
+    const store = createStore(
+        combineReducers({
+            listaTweets: listaTweetsReducer
+        }),
+        applyMiddleware(
+            thunkMiddleware
+        )
+    )
+
+    return (
+        <StoreContexto.Provider value={ {store: store} }>
+            { props.children }
+        </StoreContexto.Provider>
+    )
+}
```

3. No arquivo **`Tweet.jsx`** na pasta **`src/components/Tweet`** faça as seguintes alterações:

 # src/components/Tweet/Tweet.jsx

```
-import React from 'react'
+import React, { useContext } from 'react'

 import './tweet.css'

-import { store } from '../../store.js'
 import { criaAcaoLike } from '../../ducks/listaTweets.js'

+import { StoreContexto } from  '../../components/index.js'
+
 export function TweetCabecalho({ nomeCompletoUsuario, nomeUsuario }) {
     return (
         <div className="tweet__cabecalho">
             <img className="tweet__fotoUsuario" src="https://placehold.it/50x50" alt="" />
             <span className="tweet__nomeUsuario">{ nomeCompletoUsuario }</span>
             <a href="/"><span className="tweet__userName">@{ nomeUsuario }</span></a>
         </div>
     )
 }

 function TweetFooter({ qtLikes, likeado, onLike}) {
     return (
         <footer className="tweet__footer">
             <button className="btn btn--clean" onClick={ onLike }>
                 <svg className={'icon icon--small iconHeart' + (likeado ? ' iconHeart--active' :
''')} xmlns="http://www.w3.org/2000/svg" viewBox="0 0 47.5 47.5">
                     <defs>
```

```
                        <clipPath id="a">
                            <path d="M0 38h38V0H0v38z"></path>
                        </clipPath>
                    </defs>
                    <g clipPath="url(#a)" transform="matrix(1.25 0 0 -1.25 0 47.5)">
                        <path d="M36.885 25.166c0 5.45-4.418 9.868-9.867 9.868-3.308 0-6.227-1.6
32-8.018-4.128-1.79 2.496-4.71 4.129-8.017 4.129-5.45 0-9.868-4.418-9.868-9.868 0-.773.098-1.52.26
6-2.242C2.75 14.413 12.216 5.431 19 2.965c6.783 2.466 16.249 11.448 17.617 19.96.17.721.268 1.47.2
68 2.241"></path>
                    </g>
                </svg>
                { qtLikes }
            </button>
        </footer>
    )
}

// Componente Tweet
export function Tweet(props) {

+    const { store } = useContext(StoreContexto)
+
    const conteudo = props.children

    function dahLike() {
        store.dispatch(criaAcaoLike(props.id))
    }

    return (
        <article className="tweet">
            <TweetCabecalho
                nomeCompletoUsuario={props.nomeCompletoUsuario}
                nomeUsuario={props.nomeUsuario}
            />

            <p onClick={props.onConteudoClicado} className="tweet__conteudo">{ conteudo }</p>

            {/*Problema: prop drilling, passando o onLike para o TweetFooter */}
            <TweetFooter qtLikes={props.qtLikes} likeado={props.likeado} onLike={ dahLike }/>

        </article>
    )
}
```

4. No arquivo **`index.js`** na pasta **`src/components`** faça as seguintes alterações:

 # src/components/index.js

```
export { Cabecalho } from './Cabecalho/Cabecalho.jsx'
export { NavMenu } from './NavMenu/NavMenu.jsx'
export { Dashboard } from './Dashboard/Dashboard.jsx'
export { Widget } from './Widget/Widget.jsx'
export { TrendsArea } from './TrendsArea/TrendsArea.jsx'
export { Tweet } from './Tweet/Tweet.jsx'
export { ComponenteAutenticado } from './ComponenteAutenticado/ComponenteAutenticado.jsx'
export { Form } from './Form/Form.jsx'
export { Modal } from './Modal/Modal.jsx'
+
+export { Notificacao } from './Notificacao/Notificacao.jsx'
+export * from './ReduxStoreProvider/ReduxStoreProvider.jsx'
```

5. No arquivo **`Home.jsx`** na pasta **`src/pages/Home`** faça as seguintes alterações:

 # src/pages/Home/Home.jsx

```
 // Hook
-import React, { useState, useEffect, useContext} from 'react'
+import React, { useState, useEffect, useContext } from 'react'
 import {
     Tweet,
     Cabecalho,
     NavMenu,
     Dashboard,
     Widget,
     TrendsArea,
     ComponenteAutenticado,
     Form,
     Modal
 } from '../../components/index.js'

 import * as AutenticarService from '../../model/services/AutenticarService.js'

 import { Contexto as NotificacaoContexto } from '../../components/Notificacao/Notificacao.jsx'

-import { store } from  '../../store.js'
+import { StoreContexto } from  '../../components/index.js'

 import { criaAcaoAdicionarServidor, criaAcaoCarregarServidor } from '../../ducks/listaTweets.js'

 export function HomeSemAutenticacao() {
     const [ listaTweets, setListaTweets ] = useState([])

     const { setMsg } = useContext(NotificacaoContexto)

+    const { store } = useContext(StoreContexto)
+
     store.subscribe(() => {
         setListaTweets(store.getState().listaTweets)
     })

     useEffect(() => {
         store.dispatch(criaAcaoCarregarServidor())
     }, [])

     useEffect(() => {
         setTimeout(() => {
             setMsg("")
         }, 5000)
     })

     const [tweetModal, setTweetModal] = useState(null)

     function abreModal(tweet) {
         setTweetModal(tweet)
     }

     function fechaModal() {
         setTweetModal(null)
     }

     function adicionaTweet(textoTweetNovo) {
         store.dispatch(criaAcaoAdicionarServidor(textoTweetNovo))
     }
```

```
    return (
        <div>
            <Cabecalho>
               <NavMenu usuario="artadam"> </NavMenu>
            </Cabecalho>

            <div className="container">
                <Dashboard>
                    <Widget>
                        <Form adicionaTweet={ adicionaTweet }/>
                    </Widget>
                    <Widget>
                        <TrendsArea></TrendsArea>
                    </Widget>
                </Dashboard>

                <Dashboard posicao="centro">
                    <Widget>
                        <div className="tweetsArea">

                            {listaTweets.map(infoTweet =>
                                <Tweet { ...infoTweet } key={infoTweet.id} onConteudoClicado={()
=> abreModal(infoTweet)} >
                                    { infoTweet.conteudo }
                                </Tweet>
                            )}

                        </div>
                    </Widget>
                </Dashboard>
            </div>

            {tweetModal !== null
                ? (
                    <Modal onFechando={fechaModal}>
                        <Tweet {...tweetModal} >
                            {tweetModal.conteudo}
                        </Tweet>
                    </Modal>
                )
                : ''
            }

        </div>
    )
}

export function Home(props) {
    const isLogado = AutenticarService.isAutenticado()

    return (
        <ComponenteAutenticado podeExibir={isLogado} redirecionarPara="/login" >
            <HomeSemAutenticacao {...props}/>
        </ComponenteAutenticado>
    )
}
```

6. Remova o arquivo **`store.js`** na pasta **`src`** .