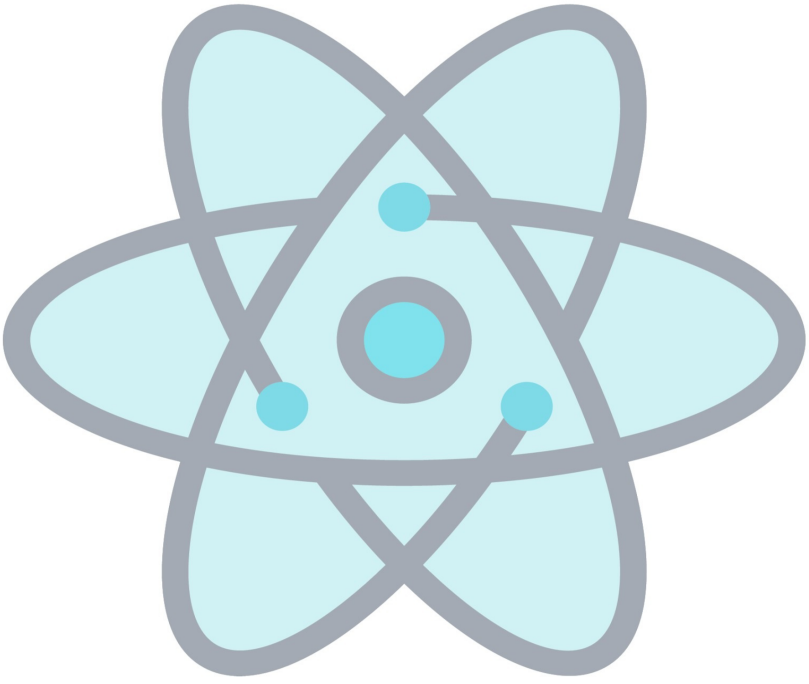
Códigos do Curso React para construção de Web Apps

Turma react8773



caelum

ensino e inovação

Sumário

Versão: 23.9.3

Capítulo 1

EXERCÍCIO: INÍCIO DE TUDO COM O VISUAL DO TWITTELUM.

1.1 PASSO A PASSO COM CÓDIGO

1. Há 17 arquivos a serem adicionados nas pastas css, js e raiz do projeto. Note que esses arquivos foram disponibilizados já prontos para você.

   Os 17 arquivos devem ser adicionados na seguinte estrutura de pastas:

   # novos arquivos do projeto

```
└─ pasta_raiz_do_projeto
   ├─ index.html
   ├─ css
   │  ├─ btn.css
   │  ├─ cabecalho.css
   │  ├─ container.css
   │  ├─ dashboard.css
   │  ├─ icon.css
   │  ├─ iconHeart.css
   │  ├─ navMenu.css
   │  ├─ notificacao.css
   │  ├─ novoTweet.css
   │  ├─ reset.css
   │  ├─ trendsArea.css
   │  ├─ tweet.css
   │  └─ widget.css
   └─ js
      └─ lib
         ├─ babel.js
         ├─ react-dom.js
         └─ react.js
```

CAPÍTULO 2

EXERCÍCIO: O TWITTELUM COM CÓDIGO DE VIEW IMPERATIVO USANDO DOM PURO.

2.1 PASSO A PASSO COM CÓDIGO

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

 # js/app.js

```
+const $tweetsArea = document.querySelector(".tweetsArea")
+const $formNovoTweet = document.querySelector(".novoTweet")
+const $textArea = document.querySelector('.novoTweet__editor')
+
+const listaTweets = [
+    "Tweet Novo 1",
+    "Tweet Novo 2"
+]
+
+render()
+
+setInterval(function(){
+    const novosTweetsDoServidor = [
+        "Tweet Servidor 1",
+        "Tweet Servidor 2"
+    ]
+    // spread operator ...
+    listaTweets.unshift(...novosTweetsDoServidor)
+
+    render()
+}, 5000)
+
+
+$formNovoTweet.addEventListener("submit", function poeTweet(evento) {
+    evento.preventDefault()
+
+    // Aqui seria um Request para mandar p/ servidor/. AJAX
+    const conteudoTweet = $textArea.value
+    listaTweets.unshift(conteudoTweet)
+
+    render()
+})
+
+
+function render() {
+    // Imperativo
+    // Ideal seria rodar um código para definir quem são os novos tweets
```

```
+    // const novosTweets = listaTweets
+
+    // $tweetsArea.innerHTML = ""
+
+    // for(const tweet of novosTweets) {
+    //     $tweetsArea.appendChild(
+    //         criaTweet(tweet)
+    //     )
+    // }
+
+    // Declarativo
+    for(const tweet of listaTweets) {
+        $tweetsArea.innerHTML += criaTweet(tweet)
+    }
+}
+
+function criaTweet(conteudoTweet) {
+
+    // Imperativo
+    // const $article = document.createElement('article')
+    // $article.classList.add("tweet")
+
+    // const $pConteudo = document.createElement('p')
+    // $pConteudo.classList.add("tweet__conteudo")
+    // $pConteudo.textContent = conteudoTweet
+
+    // $article.appendChild($pConteudo)
+
+    // return $article
+
+    // Declarativo
+    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"> ${ conteudoTweet } </p>
+            <footer class="tweet__footer">
+                <button class="btn btn--clean">
+                    <svg class="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
-1.632-8.018-4.128-1.79 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>
+    `
+}
+
```

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

 # index.html

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

<head>
    <!-- Código do <head> em index.html aqui. Omitido para facilitar a visualização. -->
</head>

<body>
    <!-- Código do <body> em index.html aqui. Omitido para facilitar a visualização. -->
+
+    <script type="module" src="js/app.js"></script>
</body>

</html>
```

Capítulo 3

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

3.1 PASSO A PASSO COM CÓDIGO

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

 # js/index.js

```
+// ESModules em navegadores antigos
+import { Tweet } from '/js/components/Tweet/index.js'
+
+const listaTweets = [
+    "Tweet 1",
+    "Tweet 2",
+    "Tweet 3"
+]
+
+const $listaTweets = listaTweets.map(conteudo => Tweet(conteudo))
+
+ReactDOM.render(
+    $listaTweets,
+    document.querySelector('.tweetsArea')
+)
```

2. Na pasta **`js`** , renomeie o arquivo **`app.js`** para **`appDOMPuro.js`** Faça também as seguintes modificações no código:

 # js/appDOMPuro.js

```
  const $tweetsArea = document.querySelector(".tweetsArea")
  const $formNovoTweet = document.querySelector(".novoTweet")
  const $textArea = document.querySelector('.novoTweet__editor')

  const listaTweets = [
      "Tweet Novo 1",
      "Tweet Novo 2"
  ]

 -render()
 +// render()

  setInterval(function(){
      const novosTweetsDoServidor = [
          "Tweet Servidor 1",
```

```
            "Tweet Servidor 2"
        ]
        // spread operator ...
        listaTweets.unshift(...novosTweetsDoServidor)

-       render()
+       // render()
    }, 5000)

    $formNovoTweet.addEventListener("submit", function poeTweet(evento) {
        evento.preventDefault()

        // Aqui seria um Request para mandar p/ servidor/. AJAX
        const conteudoTweet = $textArea.value
        listaTweets.unshift(conteudoTweet)

-       render()
+       // render()
    })

    function render() {
        // Imperativo
        // Ideal seria rodar um código para definir quem são os novos tweets
        // const novosTweets = listaTweets

        // $tweetsArea.innerHTML = ""

        // for(const tweet of novosTweets) {
        //     $tweetsArea.appendChild(
        //         criaTweet(tweet)
        //     )
        // }

        // Declarativo
        for(const tweet of listaTweets) {
            $tweetsArea.innerHTML += criaTweet(tweet)
        }
    }

    function criaTweet(conteudoTweet) {

        // Imperativo
        // const $article = document.createElement('article')
        // $article.classList.add("tweet")

        // const $pConteudo = document.createElement('p')
        // $pConteudo.classList.add("tweet__conteudo")
        // $pConteudo.textContent = conteudoTweet

        // $article.appendChild($pConteudo)

        // return $article

        // Declarativo
        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"> ${ conteudoTweet } </p>
                <footer class="tweet__footer">
```

```
                <button class="btn btn--clean">
                    <svg class="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
-1.632-8.018-4.128-1.79 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>
    `
}
```

3. No arquivo **`index.html`** na pasta **`raiz 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">
+
+   <link rel="stylesheet" href="js/components/Tweet/tweet.css">

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

<body>
   <!-- Código do <body> em index.html aqui. Omitido para facilitar a visualização. -->

-   <script type="module" src="js/app.js"></script>
+   <!-- TODO Usar módulos -->
+   <script src="js/lib/react.js"></script>
+   <script src="js/lib/react-dom.js"></script>
+
+   <!-- Ponto de entrada  -->
```

```
+    <script src="js/index.js"></script>
 </body>

 </html>
```

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

 # js/components/Tweet/index.js

```
+export function Tweet(conteudo) {
+    return (
+        React.createElement('article', { className: 'tweet' }, [
+            React.createElement('div', {className: 'tweet__cabecalho'},
+                "CONT HEADER"
+            ),
+            React.createElement('p', { className: 'tweet__conteudo'},
+                conteudo
+            ),
+            React.createElement('footer', {className: 'tweet__footer'},
+                "CONT Footer"
+            ),
+        ])
+    )
+}
+
+/*
+<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-reactro
ot="">#amet</a>
+            consectetur adipisicing <a href="/trends/#elit" data-reactroot="">#elit</a>. Adipisc
i 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" 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 0-6.227-1.632-8.0
18-4.128-1.79 2.496-4.71 4.129-8.017 4.129-5.45 0-9.868-4.418-9.868-9.868 0-.773.098-1.52.266-2.24
2C2.75 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 2.2
41">
+                    </path>
+                </g>
+            </svg>
+            0
+        </button>
+    </footer>
+</article>
+*/
```

5. Mova o arquivo **`tweet.css`** da pasta **`css`** para a pasta **`js/components/Tweet`** .

CAPÍTULO 4

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

4.1 PASSO A PASSO COM CÓDIGO

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

 # js/components/Tweet/index.js

```
 +import 'js/components/Tweet/tweet.css'
 +
  export function Tweet(conteudo) {
      return (
 -        React.createElement('article', { className: 'tweet' }, [
 -            React.createElement('div', {className: 'tweet__cabecalho'},
 -                "CONT HEADER"
 -            ),
 -            React.createElement('p', { className: 'tweet__conteudo'},
 -                conteudo
 -            ),
 -            React.createElement('footer', {className: 'tweet__footer'},
 -                "CONT Footer"
 -            ),
 -        ])
 +        <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">conteudo</p>
 +            <footer class="tweet__footer">
 +                <button class="btn btn--clean">
 +                    <svg class="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
.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-.773.098-1.52.
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>
      )
  }

  /*
  <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-reactro
ot="">#amet</a>
              consectetur adipisicing <a href="/trends/#elit" data-reactroot="">#elit</a>. Adipisc
i 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" 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 0-6.227-1.632-8.0
18-4.128-1.79 2.496-4.71 4.129-8.017 4.129-5.45 0-9.868-4.418-9.868-9.868 0-.773.098-1.52.266-2.24
2C2.75 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 2.2
41">
                      </path>
                  </g>
              </svg>
              0
          </button>
      </footer>
  </article>
  */
```

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

index.html

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

<head>
    <!-- Código do <head> em index.html aqui. Omitido para facilitar a visualização. -->
</head>

<body>
   <!-- Código do <body> em index.html aqui. Omitido para facilitar a visualização. -->
```

```
      <!-- TODO Usar módulos -->
      <script src="js/lib/react.js"></script>
      <script src="js/lib/react-dom.js"></script>
+     <script src="js/lib/babel6.js"></script>
+
+     <script type="text/babel" src="js/components/Tweet/index.js" data-plugins="transform-es2015-
modules-umd" data-presets="es2015, react"></script>

      <!-- Ponto de entrada  -->
-     <script src="js/index.js"></script>
+     <!-- jsx, ts, tsx, es2015 -->
+     <script type="text/babel" src="js/index.js" data-plugins="transform-es2015-modules-umd" data
-presets="es2015, react"></script>
  </body>

  </html>
```

3. Adicione o arquivo **`babel6.js`** na pasta **`js/lib`** . Este arquivo é um arquivo que foi disponibilizado já pronto para você.

CAPÍTULO 5

EXERCÍCIO: O PROCESSO DE BUILD – COMPILANDO MUITO MAIS QUE JSX. O COMEÇO DO BUILD COM NPM E REACT-SCRIPTS.

5.1 PASSO A PASSO COM CÓDIGO

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

 # package.json

```
+{
+  "name": "twitelum",
+  "version": "0.1.0",
+  "private": true,
+  "dependencies": {
+    "react": "^16.10.2",
+    "react-dom": "^16.10.2",
+    "react-scripts": "3.2.0"
+  },
+  "scripts": {
+    "start": "react-scripts start",
+    "build": "react-scripts build",
+    "test": "react-scripts test",
+    "eject": "react-scripts eject"
+  },
+  "eslintConfig": {
+    "extends": "react-app"
+  },
+  "browserslist": {
+    "production": [
+      ">0.2%",
+      "not dead",
+      "not op_mini all"
+    ],
+    "development": [
+      "last 1 chrome version",
+      "last 1 firefox version",
+      "last 1 safari version"
+    ]
+  }
+}
```

2. Adicione o arquivo **`.gitignore`** na pasta **`raiz do projeto`** . Este arquivo é um arquivo que foi

disponibilizado já pronto para você.

Capítulo 6

EXERCÍCIO: O PROCESSO DE BUILD. TERMINANDO O BUILD COM NPM E REACT-SCRIPTS.

6.1 PASSO A PASSO COM CÓDIGO

1. Como visto em aula, existe um *bug* na biblioteca **chokidar** que é responsável por detectar alterações nos nossos arquivos e avisar o servidor de desenvolvimento do **react-scripts**. Por enquanto, para contornar esse *bug* iremos ativar o modo de *polling*, onde o chokidar irá periodicamente ficar olhando nossos arquivos para ver se algo mudou. Para isso alteraremos uma variável de ambiente criando o arquivo `.env` na pasta `raiz do projeto` com o seguinte código:

   # .env

   ```
   +CHOKIDAR_USEPOLLING=true
   ```

2. Mova o arquivo `index.html` para a pasta `public` . No momento este arquivo está na pasta `raiz 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">
   +   <!-- Apagar estilos e scripts e importá-los nos nossos módulos js-->
   -
   -   <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="js/components/Tweet/tweet.css">
      <title> Twittelum </title>
  </head>

  <body>
      <header class="cabecalho">
          <!-- Código do <header> em index.html aqui. Omitido para facilitar a visualização. -->
      </header>

      <div class="container">
         <!-- Código da <div class="container"> em index.html aqui. Omitido para facilitar a visua
lização. -->
      </div>

-     <!-- TODO Usar módulos -->
-     <script src="js/lib/react.js"></script>
-     <script src="js/lib/react-dom.js"></script>
-     <script src="js/lib/babel6.js"></script>
-
-     <script type="text/babel" src="js/components/Tweet/index.js" data-plugins="transform-es2015-
modules-umd" data-presets="es2015, react"></script>
-
-     <!-- Ponto de entrada  -->
-     <!-- jsx, ts, tsx, es2015 -->
-     <script type="text/babel" src="js/index.js" data-plugins="transform-es2015-modules-umd" data
presets="es2015, react"></script>
  </body>

  </html>
```

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

 # src/index.js

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

+import './css/reset.css'
+import './css/container.css'
+import './css/btn.css'
+import './css/icon.css'
+import './css/iconHeart.css'
+import './css/notificacao.css'
+
+import './css/cabecalho.css'
+import './css/navMenu.css'
+import './css/dashboard.css'
+import './css/widget.css'
+import './css/novoTweet.css'
+import './css/trendsArea.css'
+
 const listaTweets = [
     "Tweet 1",
    "Tweet 2",
    "Tweet 3"
 ]

 const $listaTweets = listaTweets.map(conteudo => Tweet(conteudo))
```

```
ReactDOM.render(
    $listaTweets,
    document.querySelector('.tweetsArea')
)
```

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

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

 # src/components/Tweet/Tweet.js

```
import 'js/components/Tweet/tweet.css'

export function Tweet(conteudo) {
    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">conteudo</p>
+           <p class="tweet__conteudo">{ conteudo }</p>
            <footer class="tweet__footer">
                <button class="btn btn--clean">
                    <svg class="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
.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-.773.098-1.52.
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>
    )
}
-
-/*
-<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-reactro
t="">#amet</a>
-             consectetur adipisicing <a href="/trends/#elit" data-reactroot="">#elit</a>. Adipisc
i 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" 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 0-6.227-1.632-8.0
8-4.128-1.79 2.496-4.71 4.129-8.017 4.129-5.45 0-9.868-4.418-9.868-9.868 0-.773.098-1.52.266-2.242
2.75 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. Há 4 arquivos a serem removidos da pasta js. Note que esses arquivos foram disponibilizados já prontos para você.

   Os 4 arquivos que devem ser removidos se encontram na seguinte estrutura de pastas:

   # arquivos removidos do projeto

   ```
   └─ pasta_raiz_do_projeto
      └─ js
         └─ lib
            ├─ babel.js
            ├─ babel6.js
            ├─ react-dom.js
            └─ react.js
   ```

7. Mova o arquivo **`appDOMPuro.js`** da pasta **`js`** para a pasta **`src`** .

8. Mova o arquivo **`btn.css`** da pasta **`css`** para a pasta **`src/css`** .

9. Mova o arquivo **`cabecalho.css`** da pasta **`css`** para a pasta **`src/css`** .

10. Mova o arquivo **`container.css`** da pasta **`css`** para a pasta **`src/css`** .

11. Mova o arquivo **`dashboard.css`** da pasta **`css`** para a pasta **`src/css`** .

12. Mova o arquivo **`icon.css`** da pasta **`css`** para a pasta **`src/css`** .

13. Mova o arquivo **`iconHeart.css`** da pasta **`css`** para a pasta **`src/css`**.

14. Mova o arquivo **`navMenu.css`** da pasta **`css`** para a pasta **`src/css`**.

15. Mova o arquivo **`notificacao.css`** da pasta **`css`** para a pasta **`src/css`**.

16. Mova o arquivo **`novoTweet.css`** da pasta **`css`** para a pasta **`src/css`**.

17. Mova o arquivo **`reset.css`** da pasta **`css`** para a pasta **`src/css`**.

18. Mova o arquivo **`trendsArea.css`** da pasta **`css`** para a pasta **`src/css`**.

19. Mova o arquivo **`widget.css`** da pasta **`css`** para a pasta **`src/css`**.

Capítulo 7

EXERCÍCIO: REACT E A COMPONENTIZAÇÃO. COMPONENTE TWEET COM JSX.

7.1 PASSO A PASSO COM CÓDIGO

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

 # src/components/Tweet/Tweet.js

```
 import React from 'react'

 import './tweet.css'

-export function Tweet(conteudo) {
+// Componente é uma função
+export function Tweet(props) {
+    // RETORNA um Elemento
     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>
+            <p className="tweet__conteudo">{ props.children }</p>
             <footer className="tweet__footer">
                 <button className="btn btn--clean">
                     <svg className="icon icon--small iconHeart" 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 0-6.227-1.632-8.018-4.128-1.79 2.496-4.71 4.129-8.017 4.129-5.45 0-9.868-4.418-9.868-9.868 0-.773.098-1.52.266-2.242C2.75 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
+                    { props.qtLikes }
```

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

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

 # src/index.js

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

 // ESModules em navegadores antigos
 import { Tweet } from './components/Tweet/Tweet.js'

 import './css/reset.css'
 import './css/container.css'
 import './css/btn.css'
 import './css/icon.css'
 import './css/iconHeart.css'
 import './css/notificacao.css'

 import './css/cabecalho.css'
 import './css/navMenu.css'
 import './css/dashboard.css'
 import './css/widget.css'
 import './css/novoTweet.css'
 import './css/trendsArea.css'

 const listaTweets = [
     "Tweet 1",
     "Tweet 2",
     "Tweet 3"
 ]

-const $listaTweets = listaTweets.map(conteudo => Tweet(conteudo))
+const $listaTweets = listaTweets.map(
+    conteudo => (
+        <Tweet qtLikes={2} >
+            {conteudo}
+        </Tweet>
+    )
+)

 ReactDOM.render(
     $listaTweets,
     document.querySelector('.tweetsArea')
 )
```

Capítulo 8

EXERCÍCIO: A PÁGINA INTEIRA SENDO CONTROLADA PELO REACT. O COMPONENTE HOME E A IDEIA DE SERVER SIDE RENDERING.

8.1 PASSO A PASSO COM CÓDIGO

1. 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">
    <!-- Apagar estilos e scripts -->
    <title> Twittelum </title>
</head>

<body>
    <!-- Recortar todo o contéudo do <body> para transferir para componente Home -->
-   <header class="cabecalho">
-       ... conteúdo do header ...
-   </header>

-   <div class="container">
-       ... conteúdo da div ...
-   </div>
</body>

</html>
```

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

 # src/pages/Home.js

```
+import React from 'react'
+
+export function Home() {
+    return (
+    <>
+        <header class="cabecalho">
```

```
+           ... conteúdo do header que estava em index.html...
+       </header>
+
+       <div class="container">
+           <div class="dashboard">
+              ... conteúdo do primeiro dashboard que estava em index.html...
+           </div>
+
+           <div class="dashboard dashboard__centro">
+               <div class="widget">
+                   <div class="tweetsArea">
+                       { listaTweets.map(conteudo => (
+                           <Tweet qtLikes={2} >
+                               {conteudo}
+                           </Tweet>
+                       )) }
+                   </div>
+               </div>
+           </div>
+       </div>
+   </>
+   )
+}
```

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

 # src/index.js

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

-// ESModules em navegadores antigos
-import { Tweet } from './components/Tweet/Tweet.js'
+import { Home } from './pages/Home.js'

 import './css/reset.css'
 import './css/container.css'
 import './css/btn.css'
 import './css/icon.css'
 import './css/iconHeart.css'
 import './css/notificacao.css'

 import './css/cabecalho.css'
 import './css/navMenu.css'
 import './css/dashboard.css'
 import './css/widget.css'
 import './css/novoTweet.css'
 import './css/trendsArea.css'

-const listaTweets = [
-    "Tweet 1",
-    "Tweet 2",
-    "Tweet 3"
-]
-
-const $listaTweets = listaTweets.map(
-    conteudo => (
-        <Tweet qtLikes={2} >
-            {conteudo}
-        </Tweet>
-    )
-)
```

```
-
-ReactDOM.render(
-    $listaTweets,
-    document.querySelector('.tweetsArea')
+ReactDOM.hydrate(
+    <Home/>,
+    document.querySelector('body')
 )
```

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

 # src/pages/Home.js

```
 import React from 'react'

+import { Tweet } from '../components/Tweet/Tweet.js'
+
+const listaTweets = [
+    "Tweet 1",
+    "Tweet 2",
+    "Tweet 3"
+]
+
+
 export function Home() {
     return (
     <>
         <header class="cabecalho">
            ... conteúdo do header que estava em index.html...
         </header>

         <div class="container">
             <div class="dashboard">
                ... conteúdo do primeiro dashboard que estava em index.html...
             </div>

             <div class="dashboard dashboard__centro">
                 <div class="widget">
                     <div class="tweetsArea">
+                        { listaTweets.map(conteudo => (
+                            <Tweet qtLikes={2} >
+                                {conteudo}
+                            </Tweet>
+                        )) }
                     </div>
                 </div>
             </div>
         </div>
     </>
     )
 }
```

CAPÍTULO 9

EXERCÍCIO: BOAS PRÁTICA – ELEMENTO RAIZ DO REACT, FRAGMENTS E CLASSNAME.

9.1 PASSO A PASSO COM CÓDIGO

1. Na pasta **`src`** , renomeie o arquivo **`appDOMPuro.js`** para **`___appDOMPuro___.js`** .

2. 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.js'

import './css/reset.css'
import './css/container.css'
import './css/btn.css'
import './css/icon.css'
import './css/iconHeart.css'
import './css/notificacao.css'

import './css/cabecalho.css'
import './css/navMenu.css'
import './css/dashboard.css'
import './css/widget.css'
import './css/novoTweet.css'
import './css/trendsArea.css'

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

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

 # src/pages/Home.js

```
import React from 'react'

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

```
const listaTweets = [
    "Tweet 1",
    "Tweet 2",
    "Tweet 3"
]

export function Home() {
    return (
-    <>
-        <header class="cabecalho">
-            <div class="cabecalho__container container">
-                <h1 class="cabecalho__logo">
+    <React.Fragment>
+        <header className="cabecalho">
+            <div className="cabecalho__container container">
+                <h1 className="cabecalho__logo">
                     <a href="/">Twitelum</a>
                 </h1>
-                <nav class="navMenu">
-                    <ul class="navMenu__lista">
-                        <li class="navMenu__item">
-                            <a class="navMenu__link" href="/">
+                <nav className="navMenu">
+                    <ul className="navMenu__lista">
+                        <li className="navMenu__item">
+                            <a className="navMenu__link" href="/">
                                 Bem vindo(a): <br />
                                 <strong> @usuario</strong>
                             </a>
                         </li>
-                        <li class="navMenu__item">
-                            <a class="navMenu__link" href="/">
+                        <li className="navMenu__item">
+                            <a className="navMenu__link" href="/">
                                 Página Inicial
                             </a>
                         </li>
-                        <li class="navMenu__item">
-                            <a class="navMenu__link" href="/hashtags">
+                        <li className="navMenu__item">
+                            <a className="navMenu__link" href="/hashtags">
                                 Hashtags
                             </a>
                         </li>
-                        <li class="navMenu__item">
-                            <a class="navMenu__link" href="/logout">
+                        <li className="navMenu__item">
+                            <a className="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
```

```
do?"></textarea>
 +        <div className="container">
 +            <div className="dashboard">
 +                <div className="widget">
 +                    <form className="novoTweet">
 +                        <div className="novoTweet__editorArea">
 +                            <span className="novoTweet__status">0/140</span>
 +                            <textarea className="novoTweet__editor" placeholder="O que está acon
tecendo?"></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">
 +               <div className="widget">
 +                   <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>
                 </div>
             </div>

 -           <div class="dashboard dashboard__centro">
 -               <div class="widget">
 -                   <div class="tweetsArea">
 +           <div className="dashboard dashboard__centro">
 +               <div className="widget">
 +                   <div className="tweetsArea">
                         { listaTweets.map(conteudo => (
                             <Tweet qtLikes={2} >
                                 {conteudo}
                             </Tweet>
                         )) }
                     </div>
                 </div>
             </div>
         </div>
 -   </>
 +   </React.Fragment>
     )
 }
```

4. 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">
     <!-- Apagar estilos e scripts -->
     <title> Twittelum </title>
 +   <script analytics></script>
```

```
+    <script analytics></script>
 </head>

 <body>
-
+    <div id="rootReact"></div>

 </body>

 </html>
```

Capítulo 10

EXERCÍCIO: MAIS COMPONENTES. QUEBRANDO A HOME EM COMPONENTES.

10.1 PASSO A PASSO COM CÓDIGO

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

 # src/components/Cabecalho/Cabecalho.js

```
+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.js`** na pasta **`src/components/Dashboard`** com o seguinte código:

 # src/components/Dashboard/Dashboard.js

```
+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.js`** na pasta **`src/components/NavMenu`** com o seguinte código:

 # src/components/NavMenu/NavMenu.js

```
+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.js`** na pasta **`src/components/TrendsArea`** com o seguinte código:

 # src/components/TrendsArea/TrendsArea.js

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

 # src/components/Widget/Widget.js

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

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

 # src/components/index.js

```
+export * from './Cabecalho/Cabecalho.js'
+export * from './Dashboard/Dashboard.js'
+export * from './NavMenu/NavMenu.js'
+export * from './TrendsArea/TrendsArea.js'
+export * from './Widget/Widget.js'
+export * from './Tweet/Tweet.js'
```

7. 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.js'
-
 import './css/reset.css'
 import './css/container.css'
 import './css/btn.css'
 import './css/icon.css'
 import './css/iconHeart.css'
 import './css/notificacao.css'

-import './css/cabecalho.css'
-import './css/navMenu.css'
-import './css/dashboard.css'
-import './css/widget.css'
-import './css/novoTweet.css'
-import './css/trendsArea.css'
+import { Home } from './pages/Home.js'

 ReactDOM.render(
     <Home/>,
     document.querySelector('#rootReact')
 )
```

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

 # src/pages/Home.js

```
 import React from 'react'

-import { Tweet } from '../components/Tweet/Tweet.js'
+import '../css/novoTweet.css'
+
+import {
+    Cabecalho,
+    NavMenu,
+    Dashboard,
+    Widget,
+    TrendsArea,
+    Tweet
+} from '../components/index.js'

 const listaTweets = [
     "Tweet 1",
     "Tweet 2",
     "Tweet 3"
 ]

-
 export function Home() {
     return (
     <React.Fragment>
-        <header className="cabecalho">
-            <div className="cabecalho__container container">
-                <h1 className="cabecalho__logo">
-                    <a href="/">Twitelum</a>
-                </h1>
-                <nav className="navMenu">
-                    <ul className="navMenu__lista">
-                        <li className="navMenu__item">
-                            <a className="navMenu__link" href="/">
-                                Bem vindo(a): <br />
-                                <strong> @usuario</strong>
-                            </a>
-                        </li>
-                        <li className="navMenu__item">
-                            <a className="navMenu__link" href="/">
-                                Página Inicial
-                            </a>
-                        </li>
-                        <li className="navMenu__item">
-                            <a className="navMenu__link" href="/hashtags">
-                                Hashtags
-                            </a>
-                        </li>
-                        <li className="navMenu__item">
-                            <a className="navMenu__link" href="/logout">
-                                Logout
-                            </a>
-                        </li>
-                    </ul>
-                </nav>
-            </div>
-        </header>
+        <Cabecalho>
+            <NavMenu usuario="@artdiniz"></NavMenu>
+        </Cabecalho>

         <div className="container">
-            <div className="dashboard">
-                <div className="widget">
```

```
+            <Dashboard>
+                <Widget>
                     <form className="novoTweet">
                         <div className="novoTweet__editorArea">
                             <span className="novoTweet__status">0/140</span>
                             <textarea className="novoTweet__editor" placeholder="O que está acon
tecendo?"></textarea>
                         </div>
                         <button type="submit" className="novoTweet__envia">Tweetar</button>
                     </form>
-                </div>
-                <div className="widget">
-                    <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>
-                </div>
-            </div>
+                </Widget>
+                <Widget>
+                    <TrendsArea></TrendsArea>
+                </Widget>
+            </Dashboard>

-            <div className="dashboard dashboard__centro">
-                <div className="widget">
+            <Dashboard posicao="centro">
+                <Widget>
                     <div className="tweetsArea">
                         { listaTweets.map(conteudo => (
                             <Tweet qtLikes={2} >
                                 {conteudo}
                             </Tweet>
                         )) }
                     </div>
-                </div>
-            </div>
+                </Widget>
+            </Dashboard>
         </div>
     </React.Fragment>
     )
 }
```

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

10. Há 4 arquivos a serem adicionados na pasta src. Note que esses arquivos foram disponibilizados já prontos para você.

    Os 4 arquivos devem ser adicionados na seguinte estrutura de pastas:

    # novos arquivos do projeto

```
└─ pasta_raiz_do_projeto
   └─ src
      └─ components
         ├─ Dashboard
```

```
│  └─ dashboard.css
├─ NavMenu
│  └─ navMenu.module.css
├─ TrendsArea
│  └─ trendsArea.css
└─ Widget
   └─ widget.css
```

Capítulo 11

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

11.1 PASSO A PASSO COM CÓDIGO

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

 # src/pages/Home.js

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

 import '../css/novoTweet.css'

 import {
     Cabecalho,
     NavMenu,
     Dashboard,
     Widget,
     TrendsArea,
     Tweet
 } from '../components/index.js'

 const listaTweets = [
     "Tweet 1",
     "Tweet 2",
     "Tweet 3"
 ]

 export function Home() {
+
+    const [ textoTweet, setTextoTweet ] = useState("")
+
+    function onTextareaChange(evento) {
+        const $textArea = evento.target
+        setTextoTweet($textArea.value)
+    }
+
+    const isTweetInvalido = textoTweet.length > 140
+    const classeStatus = "novoTweet__status " +  (isTweetInvalido ? "novoTweet__status--invalido
: "")
+
     return (
```

```
      <React.Fragment>
          <Cabecalho>
              <NavMenu usuario="@artdiniz"></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 acon
ecendo?"></textarea>
 +                            <span className={ classeStatus }>{ textoTweet.length }/140</span>
 +                            <textarea className="novoTweet__editor" placeholder="O que está acon
tecendo?" onChange={ onTextareaChange }></textarea>
                          </div>
 -                        <button type="submit" className="novoTweet__envia">Tweetar</button>
 +                        <button disabled={ isTweetInvalido }  type="submit" className="novoTweet
__envia">Tweetar</button>
                      </form>
                  </Widget>
                  <Widget>
                      <TrendsArea></TrendsArea>
                  </Widget>
              </Dashboard>

              <Dashboard posicao="centro">
                  <Widget>
                      <div className="tweetsArea">
                          { listaTweets.map(conteudo => (
                              <Tweet qtLikes={2} >
                                  {conteudo}
                              </Tweet>
                          )) }
                      </div>
                  </Widget>
              </Dashboard>
          </div>
      </React.Fragment>
      )
  }
```

Capítulo 12

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

12.1 PASSO A PASSO COM CÓDIGO

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

 # src/pages/Home.js

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

 import '../css/novoTweet.css'

 import {
     Cabecalho,
     NavMenu,
     Dashboard,
     Widget,
     TrendsArea,
     Tweet
 } from '../components/index.js'

-const listaTweets = [
-    "Tweet 1",
-    "Tweet 2",
-    "Tweet 3"
-]
-
 export function Home() {

     const [ textoTweet, setTextoTweet ] = useState("")
+    const [ listaTweets, setListaTweets ] = useState([])

     function onTextareaChange(evento) {
         const $textArea = evento.target
         setTextoTweet($textArea.value)
     }

+    function onFormSubmit(evento) {
+        evento.preventDefault()
+        setListaTweets([ textoTweet , ...listaTweets])
+    }
+
     const isTweetInvalido = textoTweet.length > 140
     const classeStatus = "novoTweet__status " +  (isTweetInvalido ? "novoTweet__status--invalido
: "")

     return (
     <React.Fragment>
```

```
          <Cabecalho>
              <NavMenu usuario="@artdiniz"></NavMenu>
          </Cabecalho>

          <div className="container">
              <Dashboard>
                  <Widget>
 -                    <form className="novoTweet">
 +                    <form className="novoTweet" onSubmit={ onFormSubmit }>
                          <div className="novoTweet__editorArea">
                              <span className={ classeStatus }>{ textoTweet.length }/140</span>
                              <textarea className="novoTweet__editor" placeholder="O que está acon
tecendo?" onChange={ onTextareaChange }></textarea>
                          </div>
                          <button disabled={ isTweetInvalido }  type="submit" className="novoTweet
__envia">Tweetar</button>
                      </form>
                  </Widget>
                  <Widget>
                      <TrendsArea></TrendsArea>
                  </Widget>
              </Dashboard>

              <Dashboard posicao="centro">
                  <Widget>
                      <div className="tweetsArea">
                          { listaTweets.map(conteudo => (
                              <Tweet qtLikes={2} >
                                  {conteudo}
                              </Tweet>
                          )) }
                      </div>
                  </Widget>
              </Dashboard>
          </div>
      </React.Fragment>
      )
  }
```

CAPÍTULO 13

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

13.1 PASSO A PASSO COM CÓDIGO

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

 # package.json

```
 {
   "name": "twitelum",
   "version": "0.1.0",
   "private": true,
   "dependencies": {
     "react": "^16.10.2",
     "react-dom": "^16.10.2",
+    "react-router-dom": "^5.1.2",
     "react-scripts": "3.2.0"
   },
   "scripts": {
     "start": "react-scripts start",
     "build": "react-scripts build",
     "test": "react-scripts test",
     "eject": "react-scripts eject"
   },
   "eslintConfig": {
     "extends": "react-app"
   },
   "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.js`** na pasta **`src`** com o seguinte código:

 # src/App.js

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

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

 # src/components/NavMenu/NavMenu.js

```
 import React from "react";
+
+import { Link } from 'react-router-dom'
+
 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="/">
+          <Link className={navMenuStyles.navMenu__link} to="/login">
             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 }
```

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 './css/reset.css'
 import './css/container.css'
 import './css/btn.css'
 import './css/icon.css'
 import './css/iconHeart.css'
 import './css/notificacao.css'

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

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

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

 # src/pages/LoginPage/LoginPage.js

```
+import React, { Component, Fragment } from 'react'
+import { Cabecalho } from '../../components/Cabecalho/Cabecalho.js'
+import { Widget } from '../../components/Widget/Widget.js'
+
+import './loginPage.css'
+
+class LoginPage extends Component {
+    render() {
+        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</l
abel>
+                                    <input className="loginPage__input" type="text" id="login" n
ame="login"/>
+                                </div>
+                                <div className="loginPage__inputWrap">
+                                    <label className="loginPage__label" htmlFor="senha">Senha</l
abel>
+                                    <input className="loginPage__input" type="password" id="senh
a" name="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 {LoginPage}
```

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

Capítulo 14

EXERCÍCIO: VALIDAÇÃO DO FORMULÁRIO DE LOGIN - MAIS ESTADO COM HOOKS E CONDIÇÕES NO JSX.

14.1 PASSO A PASSO COM CÓDIGO

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

 # src/pages/LoginPage/LoginPage.js

```
-import React, { Component, Fragment } from 'react'
+import React, { useState, useRef, Fragment } from 'react'
 import { Cabecalho } from '../../components/Cabecalho/Cabecalho.js'
 import { Widget } from '../../components/Widget/Widget.js'

 import './loginPage.css'

-class LoginPage extends Component {
-    render() {
+function LoginPage() {
+
+    const [isValido, setIsValido] = useState(true)
+
+    const $inputLogin = useRef(null)
+    const $inputSenha = useRef(null)
+
+    function onFormSubmit(evento) {
+        evento.preventDefault()
+
+        const usuario = $inputLogin.current.value
+        const senha = $inputSenha.current.value
+
+        const isValidoSubmit = usuario.length !== 0 && senha.length !== 0
+
+        setIsValido(isValidoSubmit)
+    }
+
     return (
         <Fragment>
             <Cabecalho />
             <div className="loginPage">
                 <div className="container">
                     <Widget>
                         <h2 className="loginPage__title">Seja bem vindo!</h2>
-                            <form className="loginPage__form" action="/">
+                        <form className="loginPage__form" action="/" onSubmit={ onFormSubmit }>
                             <div className="loginPage__inputWrap">
```

```
                                <label className="loginPage__label" htmlFor="login">Login</label
 -                                  <input className="loginPage__input" type="text" id="login" n
me="login"/>
 +                                  <input ref={$inputLogin} className="loginPage__input" type="text
 id="login" name="login"/>
                                </div>
                                <div className="loginPage__inputWrap">
                                <label className="loginPage__label" htmlFor="senha">Senha</label
 -                                  <input className="loginPage__input" type="password" id="senh
a" name="senha"/>
 +                                  <input ref={$inputSenha} className="loginPage__input" type="pass
word" id="senha" name="senha"/>
 +                                </div>
 +                                {
 +                                    !isValido
 +                                        ? <div className="loginPage__errorBox">
 +                                            Senha ou usuário vazio
                                          </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 {LoginPage}
```

Capítulo 15

EXERCÍCIO: VALIDAÇÃO DE TIPOS EM PROPS E ESTADO. PROPTYPES E HOOKS CUSTOMIZADOS.

15.1 PASSO A PASSO COM CÓDIGO

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

 # package.json

```
 {
   "name": "twitelum",
   "version": "0.1.0",
   "private": true,
   "dependencies": {
+    "prop-types": "^15.7.2",
     "react": "^16.10.2",
     "react-dom": "^16.10.2",
     "react-router-dom": "^5.1.2",
     "react-scripts": "3.2.0"
   },
   "scripts": {
     "start": "react-scripts start",
     "build": "react-scripts build",
     "test": "react-scripts test",
     "eject": "react-scripts eject"
   },
   "eslintConfig": {
     "extends": "react-app"
   },
   "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.js`** na pasta **`src/components/Tweet`** faça as seguintes alterações:

 # src/components/Tweet/Tweet.js

```
 import React from 'react'
+import PropTypes from 'prop-types'

 import './tweet.css'

-// Componente é uma função
+Tweet.propTypes = {
+    qtLikes: PropTypes.number
+}
+
 export function Tweet(props) {
     // RETORNA um Elemento
     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">{ props.children }</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-.773.098-1.52.
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>
                     { props.qtLikes }
                 </button>
             </footer>
         </article>
     )
 }
```

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

 # src/pages/Home.js

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

import '../css/novoTweet.css'

import {
    Cabecalho,
    NavMenu,
    Dashboard,
    Widget,
    TrendsArea,
```

```
    Tweet
} from '../components/index.js'

export function Home() {

    const [ textoTweet, setTextoTweet ] = useState("")
    const [ listaTweets, setListaTweets ] = useState([])

    function onTextareaChange(evento) {
        const $textArea = evento.target
        setTextoTweet($textArea.value)
    }

    function onFormSubmit(evento) {
        evento.preventDefault()
        setListaTweets([ textoTweet , ...listaTweets])
    }

    const isTweetInvalido = textoTweet.length > 140
    const classeStatus = "novoTweet__status " +  (isTweetInvalido ? "novoTweet__status--invalido
: "")

    return (
    <React.Fragment>
        <Cabecalho>
            <NavMenu usuario="@artdiniz"></NavMenu>
        </Cabecalho>

        <div className="container">
            <Dashboard>
                <Widget>
                    <form className="novoTweet" onSubmit={ onFormSubmit }>
                        <div className="novoTweet__editorArea">
                            <span className={ classeStatus }>{ textoTweet.length }/140</span>
                            <textarea className="novoTweet__editor" placeholder="O que está acon
tecendo?" onChange={ onTextareaChange }></textarea>
                        </div>
                        <button disabled={ isTweetInvalido }  type="submit" className="novoTweet
__envia">Tweetar</button>
                    </form>
                </Widget>
                <Widget>
                    <TrendsArea></TrendsArea>
                </Widget>
            </Dashboard>

            <Dashboard posicao="centro">
                <Widget>
                    <div className="tweetsArea">
                        { listaTweets.map(conteudo => (
-                           <Tweet qtLikes={2} >
+                           <Tweet qtLikes={ "oi" } >
                                {conteudo}
                            </Tweet>
                        )) }
                    </div>
                </Widget>
            </Dashboard>
        </div>
    </React.Fragment>
    )
}
```

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

 # src/pages/LoginPage/LoginPage.js

```
 import React, { useState, useRef, Fragment } from 'react'
 import { Cabecalho } from '../../components/Cabecalho/Cabecalho.js'
 import { Widget } from '../../components/Widget/Widget.js'

 import './loginPage.css'

-function LoginPage() {

-    const [isValido, setIsValido] = useState(true)
+// Custom hooks
+// High Order Function
+function useStateBooleano(valorInicial) {
+    const [ valorDaVariavel, setValorDaVariavelReact] = useState(valorInicial)
+
+    const setValorDaVariavel = function(valorNovo) {
+        if(typeof valorNovo !== "boolean") {
+            throw Error("Tipo inválido: " + typeof valorNovo)
+        }
+
+        setValorDaVariavelReact(valorNovo)
+    }
+
+    return [
+        valorDaVariavel,
+        setValorDaVariavel
+    ]
+}
+
+function LoginPage() {
+    const [isValido, setIsValido] = useStateBooleano(true)

     const $inputLogin = useRef(null)
     const $inputSenha = useRef(null)

     function onFormSubmit(evento) {
         evento.preventDefault()

         const usuario = $inputLogin.current.value
         const senha = $inputSenha.current.value

         const isValidoSubmit = usuario.length !== 0 && senha.length !== 0

         setIsValido(isValidoSubmit)
     }

     return (
         <Fragment>
             <Cabecalho />
             <div className="loginPage">
                 <div className="container">
                     <Widget>
                         <h2 className="loginPage__title">Seja bem vindo!</h2>
                         <form className="loginPage__form" action="/" onSubmit={ onFormSubmit }>
                             <div className="loginPage__inputWrap">
                                 <label className="loginPage__label" htmlFor="login">Login</label

                                 <input ref={$inputLogin} className="loginPage__input" type="text
id="login" name="login"/>
```

```
                                </div>
                                <div className="loginPage__inputWrap">
                                    <label className="loginPage__label" htmlFor="senha">Senha</label

                                    <input ref={$inputSenha} className="loginPage__input" type="pass
word" id="senha" name="senha"/>
                                </div>
                                {
                                    !isValido
                                        ? <div className="loginPage__errorBox">
                                            Senha ou usuário vazio
                                          </div>
                                        : ''
                                }

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

    export {LoginPage}
```

Capítulo 16

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

16.1 PASSO A PASSO COM CÓDIGO

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

 # .env

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

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

 # src/model/services/LoginService.js

```
+const URL_API = process.env.REACT_APP_URL_API
+
+export async function logar(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")
+}
```

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

 # src/pages/LoginPage/LoginPage.js

```
 import React, { useState, useRef, Fragment } from 'react'
 import { Cabecalho } from '../../components/Cabecalho/Cabecalho.js'
 import { Widget } from '../../components/Widget/Widget.js'

 import './loginPage.css'

+import * as LoginService from '../../model/services/LoginService.js'

 // Custom hooks
 // High Order Function
 function useStateBooleano(valorInicial) {
     const [ valorDaVariavel, setValorDaVariavelReact] = useState(valorInicial)

     const setValorDaVariavel = function(valorNovo) {
         if(typeof valorNovo !== "boolean") {
             throw Error("Tipo inválido: " + typeof valorNovo)
         }

         setValorDaVariavelReact(valorNovo)
     }

     return [
         valorDaVariavel,
         setValorDaVariavel
     ]
 }

 function LoginPage() {
     const [isValido, setIsValido] = useStateBooleano(true)

     const $inputLogin = useRef(null)
     const $inputSenha = useRef(null)

     function onFormSubmit(evento) {
         evento.preventDefault()

         const usuario = $inputLogin.current.value
         const senha = $inputSenha.current.value

         const isValidoSubmit = usuario.length !== 0 && senha.length !== 0

         setIsValido(isValidoSubmit)
+
+        if(isValidoSubmit) {
+            LoginService.logar(usuario, senha)
+                .catch(error => setIsValido(false))
+        }
+
     }

     return (
         <Fragment>
```

```
            <Cabecalho />
            <div className="loginPage">
                <div className="container">
                    <Widget>
                        <h2 className="loginPage__title">Seja bem vindo!</h2>
                        <form className="loginPage__form" action="/" onSubmit={ onFormSubmit }>
                            <div className="loginPage__inputWrap">
                                <label className="loginPage__label" htmlFor="login">Login</label

                                <input ref={$inputLogin} className="loginPage__input" type="text
 id="login" name="login"/>
                            </div>
                            <div className="loginPage__inputWrap">
                                <label className="loginPage__label" htmlFor="senha">Senha</label

                                <input ref={$inputSenha} className="loginPage__input" type="pass
word" id="senha" name="senha"/>
                            </div>
                            {
                                !isValido
                                    ? <div className="loginPage__errorBox">
-                                       Senha ou usuário vazio
+                                       Senha ou usuário inválido
                                      </div>
                                    : ''
                            }

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

 export {LoginPage}
```

CAPÍTULO 17

EXERCÍCIO: LOGIN - AUTENTICAÇÃO PARA ACESSO ÀS PÁGINAS. SOLUÇÃO DUPLICANDO A LÓGICA EM CADA PÁGINA.

17.1 PASSO A PASSO COM CÓDIGO

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

 # src/App.js

```
 import React from 'react'

-import { BrowserRouter as Router, Switch, Route } from 'react-router-dom'
+import { BrowserRouter as Router, Switch, Route, Redirect} from 'react-router-dom'

 import { Home } from './pages/Home.js'
 import { LoginPage } from './pages/LoginPage/LoginPage.js'

 export function App() {
-
     return (
         <Router>
             <Switch>
                 <Route path="/" component={ Home } exact={true}/>
                 <Route path="/login" component={ LoginPage } />
             </Switch>
         </Router>
     )
 }
```

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

 # src/model/services/LoginService.js

```
 const URL_API = process.env.REACT_APP_URL_API

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

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

 # src/pages/Home.js

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

 import '../css/novoTweet.css'

+import * as LoginService from '../model/services/LoginService.js'
+
 import {
     Cabecalho,
     NavMenu,
     Dashboard,
     Widget,
     TrendsArea,
     Tweet
 } from '../components/index.js'

-export function Home() {
+import { Redirect } from 'react-router-dom'

+export function Home() {
     const [ textoTweet, setTextoTweet ] = useState("")
     const [ listaTweets, setListaTweets ] = useState([])

+    const isAutenticado = LoginService.isAutenticado()
+
     function onTextareaChange(evento) {
         const $textArea = evento.target
         setTextoTweet($textArea.value)
     }

     function onFormSubmit(evento) {
         evento.preventDefault()
         setListaTweets([ textoTweet , ...listaTweets])
```

```
      }

      const isTweetInvalido = textoTweet.length > 140
      const classeStatus = "novoTweet__status " +  (isTweetInvalido ? "novoTweet__status--invalido
 : "")

 -    return (
 +    const $pagina = (
          <React.Fragment>
              <Cabecalho>
                  <NavMenu usuario="@artdiniz"></NavMenu>
              </Cabecalho>

              <div className="container">
                  <Dashboard>
                      <Widget>
                          <form className="novoTweet" onSubmit={ onFormSubmit }>
                              <div className="novoTweet__editorArea">
                                  <span className={ classeStatus }>{ textoTweet.length }/140</span

                                  <textarea className="novoTweet__editor" placeholder="O que está
acontecendo?" onChange={ onTextareaChange }></textarea>
                              </div>
                              <button disabled={ isTweetInvalido }  type="submit" className="novoT
weet__envia">Tweetar</button>
                          </form>
                      </Widget>
                      <Widget>
                          <TrendsArea></TrendsArea>
                      </Widget>
                  </Dashboard>

                  <Dashboard posicao="centro">
                      <Widget>
                          <div className="tweetsArea">
                              { listaTweets.map(conteudo => (
                                  <Tweet qtLikes={ "oi" } >
                                      {conteudo}
                                  </Tweet>
                              )) }
                          </div>
                      </Widget>
                  </Dashboard>
              </div>
          </React.Fragment>
      )
 +
 +    return (
 +        <React.Fragment>
 +            {isAutenticado
 +                ? $pagina
 +                : <Redirect to="/login" />
 +            }
 +        </React.Fragment>
 +    )
  }
```

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

 # src/pages/LoginPage/LoginPage.js

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

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

 import './loginPage.css'

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

+import { Redirect } from 'react-router-dom'
+
 // Custom hooks
 // High Order Function
 function useStateBooleano(valorInicial) {
     const [ valorDaVariavel, setValorDaVariavelReact] = useState(valorInicial)

     const setValorDaVariavel = function(valorNovo) {
         if(typeof valorNovo !== "boolean") {
             throw Error("Tipo inválido: " + typeof valorNovo)
         }

         setValorDaVariavelReact(valorNovo)
     }

     return [
         valorDaVariavel,
         setValorDaVariavel
     ]
 }

-function LoginPage() {
+function LoginPage(props) {
     const [isValido, setIsValido] = useStateBooleano(true)

+    const isAutenticado = LoginService.isAutenticado()
+
     const $inputLogin = useRef(null)
     const $inputSenha = useRef(null)

     function onFormSubmit(evento) {
         evento.preventDefault()

         const usuario = $inputLogin.current.value
         const senha = $inputSenha.current.value

         const isValidoSubmit = usuario.length !== 0 && senha.length !== 0

         setIsValido(isValidoSubmit)

         if(isValidoSubmit) {
             LoginService.logar(usuario, senha)
+                .then(() => props.history.push("/"))
                 .catch(error => setIsValido(false))
         }

     }

-    return (
-        <Fragment>
+    const $pagina = (
+        <React.Fragment>
             <Cabecalho />
             <div className="loginPage">
                 <div className="container">
                     <Widget>
```

```
                        <h2 className="loginPage__title">Seja bem vindo!</h2>
                        <form className="loginPage__form" action="/" onSubmit={ onFormSubmit }>
                            <div className="loginPage__inputWrap">
                                <label className="loginPage__label" htmlFor="login">Login</label

                                <input ref={$inputLogin} className="loginPage__input" type="text
 id="login" name="login"/>
                            </div>
                            <div className="loginPage__inputWrap">
                                <label className="loginPage__label" htmlFor="senha">Senha</label

                                <input ref={$inputSenha} className="loginPage__input" type="pass
word" id="senha" name="senha"/>
                            </div>
                            {
                                !isValido
                                    ? <div className="loginPage__errorBox">
                                        Senha ou usuário inválido
                                    </div>
                                    : ''
                            }

                            <div className="loginPage__inputWrap">
                                <button className="loginPage__btnLogin" type="submit">
                                    Logar
                                </button>
                            </div>
                        </form>
                    </Widget>
                </div>
            </div>
+       </React.Fragment>
+   )
+
+   return (
+       <Fragment>
+           {!isAutenticado
+               ? $pagina
+               : <Redirect to="/" />
+           }
        </Fragment>
    )
 }

 export {LoginPage}
```

CAPÍTULO 18

EXERCÍCIO: LOGIN - AUTENTICAÇÃO PARA ACESSO ÀS PÁGINAS. ABSTRAINDO LÓGICA DE VIEW EM HIGH ORDER COMPONENTS.

18.1 PASSO A PASSO COM CÓDIGO

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

 # src/App.js

```
  import React from 'react'

 -import { BrowserRouter as Router, Switch, Route, Redirect} from 'react-router-dom'
 +import { BrowserRouter as Router, Switch, Route } from 'react-router-dom'

  import { Home } from './pages/Home.js'
  import { LoginPage } from './pages/LoginPage/LoginPage.js'

 +import * as LoginService from './model/services/LoginService.js'
 +
  export function App() {
      return (
          <Router>
              <Switch>
 -                <Route path="/" component={ Home } exact={true}/>
 -                <Route path="/login" component={ LoginPage } />
 +                <Route path="/" exact={true} render={(routerProps) => (
 +                    <Home {...routerProps} isAcessoPermitido={ LoginService.isAutenticado() } re
directPermissaoNegada="/login" />
 +                )} />
 +                <Route path="/login" render={(routerProps) => (
 +                    <LoginPage {...routerProps} isAcessoPermitido={ !LoginService.isAutenticado(
) } redirectPermissaoNegada="/" />
 +                )} />
              </Switch>
          </Router>
      )
  }
```

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

 # src/pages/Home.js

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

 import '../css/novoTweet.css'

-import * as LoginService from '../model/services/LoginService.js'
-
 import {
     Cabecalho,
     NavMenu,
     Dashboard,
     Widget,
     TrendsArea,
     Tweet
 } from '../components/index.js'

-import { Redirect } from 'react-router-dom'
+import { withPermissao } from  './withPermissao.js'

-export function Home() {
+function HomeSemAutenticacao() {
     const [ textoTweet, setTextoTweet ] = useState("")
     const [ listaTweets, setListaTweets ] = useState([])

-    const isAutenticado = LoginService.isAutenticado()
-
     function onTextareaChange(evento) {
         const $textArea = evento.target
         setTextoTweet($textArea.value)
     }

     function onFormSubmit(evento) {
         evento.preventDefault()
         setListaTweets([ textoTweet , ...listaTweets])
     }

     const isTweetInvalido = textoTweet.length > 140
     const classeStatus = "novoTweet__status " +  (isTweetInvalido ? "novoTweet__status--invalido
 : "")

-    const $pagina = (
+    return (
         <React.Fragment>
             <Cabecalho>
                 <NavMenu usuario="@artdiniz"></NavMenu>
             </Cabecalho>

             <div className="container">
                 <Dashboard>
                     <Widget>
                         <form className="novoTweet" onSubmit={ onFormSubmit }>
                             <div className="novoTweet__editorArea">
                                 <span className={ classeStatus }>{ textoTweet.length }/140</span

                                 <textarea className="novoTweet__editor" placeholder="O que está
acontecendo?" onChange={ onTextareaChange }></textarea>
                             </div>
                             <button disabled={ isTweetInvalido }  type="submit" className="novoT
weet__envia">Tweetar</button>
                         </form>
                     </Widget>
                     <Widget>
                         <TrendsArea></TrendsArea>
                     </Widget>
```

```
                </Dashboard>

                <Dashboard posicao="centro">
                    <Widget>
                        <div className="tweetsArea">
                            { listaTweets.map(conteudo => (
                                <Tweet qtLikes={ "oi" } >
                                    {conteudo}
                                </Tweet>
                            )) }
                        </div>
                    </Widget>
                </Dashboard>
            </div>
        </React.Fragment>
    )
-
-    return (
-        <React.Fragment>
-            {isAutenticado
-                ? $pagina
-                : <Redirect to="/login" />
-            }
-        </React.Fragment>
-    )
 }
+
+export const Home = withPermissao(HomeSemAutenticacao)
```

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

 # src/pages/LoginPage/LoginPage.js

```
-import React, { useState, useRef, Fragment } from 'react'
+import React, { useState, useRef } from 'react'
 import { Cabecalho } from '../../components/Cabecalho/Cabecalho.js'
 import { Widget } from '../../components/Widget/Widget.js'

 import './loginPage.css'

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

-import { Redirect } from 'react-router-dom'
+import { withPermissao } from  '../withPermissao.js'

 // Custom hooks
 // High Order Function
 function useStateBooleano(valorInicial) {
     const [ valorDaVariavel, setValorDaVariavelReact] = useState(valorInicial)

     const setValorDaVariavel = function(valorNovo) {
         if(typeof valorNovo !== "boolean") {
             throw Error("Tipo inválido: " + typeof valorNovo)
         }

         setValorDaVariavelReact(valorNovo)
     }

     return [
         valorDaVariavel,
         setValorDaVariavel
     ]
```

```
 }

-function LoginPage(props) {
+function LoginPageSemAutenticacao(props) {
     const [isValido, setIsValido] = useStateBooleano(true)

-    const isAutenticado = LoginService.isAutenticado()
-
     const $inputLogin = useRef(null)
     const $inputSenha = useRef(null)

     function onFormSubmit(evento) {
         evento.preventDefault()

         const usuario = $inputLogin.current.value
         const senha = $inputSenha.current.value

         const isValidoSubmit = usuario.length !== 0 && senha.length !== 0

         setIsValido(isValidoSubmit)

         if(isValidoSubmit) {
             LoginService.logar(usuario, senha)
                 .then(() => props.history.push("/"))
                 .catch(error => setIsValido(false))
         }

     }

-    const $pagina = (
+    return (
         <React.Fragment>
             <Cabecalho />
             <div className="loginPage">
                 <div className="container">
                     <Widget>
                         <h2 className="loginPage__title">Seja bem vindo!</h2>
                         <form className="loginPage__form" action="/" onSubmit={ onFormSubmit }>
                             <div className="loginPage__inputWrap">
                                 <label className="loginPage__label" htmlFor="login">Login</label

                                 <input ref={$inputLogin} className="loginPage__input" type="text
id="login" name="login"/>
                             </div>
                             <div className="loginPage__inputWrap">
                                 <label className="loginPage__label" htmlFor="senha">Senha</label

                                 <input ref={$inputSenha} className="loginPage__input" type="pass
word" id="senha" name="senha"/>
                             </div>
                             {
                                 !isValido
                                     ? <div className="loginPage__errorBox">
                                         Senha ou usuário inválido
                                     </div>
                                     : ''
                             }

                             <div className="loginPage__inputWrap">
                                 <button className="loginPage__btnLogin" type="submit">
                                     Logar
                                 </button>
                             </div>
```

```
                        </form>
                    </Widget>
                </div>
            </div>
        </React.Fragment>
    )
-
-    return (
-        <Fragment>
-            {!isAutenticado
-                ? $pagina
-                : <Redirect to="/" />
-            }
-        </Fragment>
-    )
 }

-export {LoginPage}
+export const LoginPage = withPermissao(LoginPageSemAutenticacao)
```

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

 # src/pages/withPermissao.js

```
+import React from 'react'
+
+import { Redirect } from 'react-router-dom'
+
+export function withPermissao(Componente) {
+    return function ComponentePaginaComPermissao(props) {
+        const {children, isAcessoPermitido, redirectPermissaoNegada, ...outrasProps} = props
+
+        if(isAcessoPermitido){
+            return (
+                <Componente {...outrasProps} >
+                    { props.children }
+                </Componente>
+            )
+        } else {
+            return <Redirect to={ redirectPermissaoNegada } />
+        }
+    }
+
+}
```

Capítulo 19

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

19.1 PASSO A PASSO COM CÓDIGO

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

 # src/components/FormNovoTweet/FormNovoTweet.js

```
 +import React, { useState } from 'react'
 +
 +import './novoTweet.css'
 +
 +export function FormNovoTweet(props) {
 +    const [ textoTweet, setTextoTweet ] = useState("")
 +
 +    function onTextareaChange(evento) {
 +        const $textArea = evento.target
 +        setTextoTweet($textArea.value)
 +    }
 +
 +    function onFormSubmit(evento) {
 +        evento.preventDefault()
 +        // Aqui tem que adicionar um tweet. Mas é no estado de algum componente parente, não aqu
i.
 +        props.onNovoTweet(textoTweet)
 +    }
 +
 +    const isTweetInvalido = textoTweet.length > 140
 +    const classeStatus = "novoTweet__status " +  (isTweetInvalido ? "novoTweet__status--invalido
 : "")
 +
 +    return (
 +        <form className="novoTweet" onSubmit={ onFormSubmit }>
 +            <div className="novoTweet__editorArea">
 +                <span className={ classeStatus }>{ textoTweet.length }/140</span>
 +                <textarea className="novoTweet__editor" placeholder="O que está acontecendo?" on
Change={ onTextareaChange }></textarea>
 +            </div>
```

```
 +            <button disabled={ isTweetInvalido }  type="submit" className="novoTweet__envia">Twe
etar</button>
 +        </form>
 +    )
 +}
```

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

 # src/pages/Home.js

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

 -import '../css/novoTweet.css'
 -
  import {
      Cabecalho,
      NavMenu,
      Dashboard,
      Widget,
      TrendsArea,
      Tweet
  } from '../components/index.js'

  import { withPermissao } from  './withPermissao.js'

 +import { FormNovoTweet } from '../components/FormNovoTweet/FormNovoTweet.js'
 +
 +function AreaNovoTweet(props) {
 +    return (
 +        <React.Fragment>
 +            <p> Coisas novas em cima </p>
 +            <br />
 +            <br />
 +            <FormNovoTweet onNovoTweet={props.onNovoTweet} />
 +            <br />
 +            <br />
 +            <p> Coisas novas embaixo </p>
 +        </React.Fragment>
 +    )
 +}
 +
  function HomeSemAutenticacao() {
 -    const [ textoTweet, setTextoTweet ] = useState("")
      const [ listaTweets, setListaTweets ] = useState([])

 -    function onTextareaChange(evento) {
 -        const $textArea = evento.target
 -        setTextoTweet($textArea.value)
 -    }
 -
 -    function onFormSubmit(evento) {
 -        evento.preventDefault()
 -        setListaTweets([ textoTweet , ...listaTweets])
 +    function adicionaTweet(novoTweet) {
 +        setListaTweets([ novoTweet , ...listaTweets])
      }

 -    const isTweetInvalido = textoTweet.length > 140
 -    const classeStatus = "novoTweet__status " + (isTweetInvalido ? "novoTweet__status--invalido
" : "")
 -
      return (
```

```
            <React.Fragment>
                <Cabecalho>
                    <NavMenu usuario="@artdiniz"></NavMenu>
                </Cabecalho>

                <div className="container">
                    <Dashboard>
                        <Widget>
-                           <form className="novoTweet" onSubmit={ onFormSubmit }>
-                               <div className="novoTweet__editorArea">
-                                   <span className={ classeStatus }>{ textoTweet.length }/140</span

-                                   <textarea  className="novoTweet__editor" placeholder="O que está
contecendo?" onChange={ onTextareaChange }></textarea>
-                               </div>
-                               <button disabled={ isTweetInvalido }  type="submit" className="novoT
weet__envia">Tweetar</button>
-                           </form>
+                          <AreaNovoTweet onNovoTweet={adicionaTweet} />
                        </Widget>
                        <Widget>
                            <TrendsArea></TrendsArea>
                        </Widget>
                    </Dashboard>

                    <Dashboard posicao="centro">
                        <Widget>
                            <div className="tweetsArea">
                                { listaTweets.map(conteudo => (
                                    <Tweet qtLikes={ "oi" } >
                                        {conteudo}
                                    </Tweet>
                                )) }
                            </div>
                        </Widget>
                    </Dashboard>
                </div>
            </React.Fragment>
        )
    }

    export const Home = withPermissao(HomeSemAutenticacao)
```

3. Adicione o arquivo **`novoTweet.css`** na pasta **`src/components/FormNovoTweet`**. Este arquivo é um arquivo que foi disponibilizado já pronto para você.

Capítulo 20

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

20.1 PASSO A PASSO COM CÓDIGO

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

 # src/App.js

```
  import React from 'react'

  import { BrowserRouter as Router, Switch, Route } from 'react-router-dom'

  import { Home } from './pages/Home.js'
  import { LoginPage } from './pages/LoginPage/LoginPage.js'

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

 +import { NotificacaoProvider } from './components/NotificacaoProvider/NotificacaoProvider.js'
 +
  export function App() {
      return (
 +    <NotificacaoProvider>
          <Router>
              <Switch>
                  <Route path="/" exact={true} render={(routerProps) => (
                      <Home {...routerProps} isAcessoPermitido={ LoginService.isAutenticado() } re
directPermissaoNegada="/login" />
                  )} />
                  <Route path="/login" render={(routerProps) => (
                      <LoginPage {...routerProps} isAcessoPermitido={ !LoginService.isAutenticado(
) } redirectPermissaoNegada="/" />
                  )} />
              </Switch>
          </Router>
 +    </NotificacaoProvider>
      )
  }
```

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

 # src/components/NotificacaoProvider/NotificacaoProvider.js

```
 +import React, { createContext, useState, useEffect } from 'react'
```

```
+
+export const NotificacaoContexto = createContext({
+    notificar: () => {}
+})
+
+export function NotificacaoProvider(props) {
+
+    const [msg, setMsg] = useState("")
+
+    useEffect(() => {
+        setTimeout(() => {
+            setMsg("")
+        }, 2000)
+    })
+
+    return (
+        <NotificacaoContexto.Provider value={{notificar: setMsg}}>
+            { props.children }
+
+            {msg
+                ? <div class="notificacaoMsg"> { msg } </div>
+                : ''
+            }
+        </NotificacaoContexto.Provider>
+    )
+}
```

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

 # src/pages/LoginPage/LoginPage.js

```
-import React, { useState, useRef } from 'react'
+import React, { useState, useRef, useContext } from 'react'
 import { Cabecalho } from '../../components/Cabecalho/Cabecalho.js'
 import { Widget } from '../../components/Widget/Widget.js'

+import { NotificacaoContexto } from '../../components/NotificacaoProvider/NotificacaoProvider.js

+
 import './loginPage.css'

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

 import { withPermissao } from  '../withPermissao.js'

 // Custom hooks
 // High Order Function
 function useStateBooleano(valorInicial) {
     const [ valorDaVariavel, setValorDaVariavelReact] = useState(valorInicial)

     const setValorDaVariavel = function(valorNovo) {
         if(typeof valorNovo !== "boolean") {
             throw Error("Tipo inválido: " + typeof valorNovo)
         }

         setValorDaVariavelReact(valorNovo)
     }

     return [
         valorDaVariavel,
         setValorDaVariavel
     ]
```

```
 }

 function LoginPageSemAutenticacao(props) {
     const [isValido, setIsValido] = useStateBooleano(true)

     const $inputLogin = useRef(null)
     const $inputSenha = useRef(null)

+    const contexto = useContext(NotificacaoContexto)
+
     function onFormSubmit(evento) {
         evento.preventDefault()

         const usuario = $inputLogin.current.value
         const senha = $inputSenha.current.value

         const isValidoSubmit = usuario.length !== 0 && senha.length !== 0

         setIsValido(isValidoSubmit)

         if(isValidoSubmit) {
             LoginService.logar(usuario, senha)
-                .then(() => props.history.push("/"))
+                .then(() => {
+                    contexto.notificar("Logado com sucesso")
+                    props.history.push("/")
+                })
                 .catch(error => setIsValido(false))
         }

     }

     return (
         <React.Fragment>
             <Cabecalho />
             <div className="loginPage">
                 <div className="container">
                     <Widget>
                         <h2 className="loginPage__title">Seja bem vindo!</h2>
                         <form className="loginPage__form" action="/" onSubmit={ onFormSubmit }>
                             <div className="loginPage__inputWrap">
                                 <label className="loginPage__label" htmlFor="login">Login</label

                                 <input ref={$inputLogin} className="loginPage__input" type="text
id="login" name="login"/>
                             </div>
                             <div className="loginPage__inputWrap">
                                 <label className="loginPage__label" htmlFor="senha">Senha</label

                                 <input ref={$inputSenha} className="loginPage__input" type="pass
word" id="senha" name="senha"/>
                             </div>
                             {
                                 !isValido
                                     ? <div className="loginPage__errorBox">
                                         Senha ou usuário inválido
                                     </div>
                                     : ''
                             }

                             <div className="loginPage__inputWrap">
                                 <button className="loginPage__btnLogin" type="submit">
                                     Logar
```

```
                                </button>
                            </div>
                        </form>
                    </Widget>
                </div>
            </div>
        </React.Fragment>
    )
}

export const LoginPage = withPermissao(LoginPageSemAutenticacao)
```

Capítulo 21

EXERCÍCIO: TWEETS COM INFORMAÇÕES DE VERDADE. SALVANDO E CARREGANDO TWEETS DO SERVIDOR.

21.1 PASSO A PASSO COM CÓDIGO

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

 # src/components/Tweet/Tweet.js

```
  import React from 'react'
  import PropTypes from 'prop-types'

  import './tweet.css'

  Tweet.propTypes = {
      qtLikes: PropTypes.number
  }

  export function Tweet(props) {
      // RETORNA um Elemento
      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>
 +                <img className="tweet__fotoUsuario" src={props.usuario.foto} alt="" />
 +                <span className="tweet__nomeUsuario">{props.usuario.nome} {props.usuario.sobreno
me}</span>
 +                <a href="/"><span className="tweet__userName">@{props.usuario.login}</span></a>
              </div>
              <p className="tweet__conteudo">{ props.children }</p>
              <footer className="tweet__footer">
                  <button className="btn btn--clean">
 -                    <svg className="icon icon--small iconHeart" xmlns="http://www.w3.org/2000/sv
" viewBox="0 0 47.5 47.5">
 +                    <svg className={"icon icon--small iconHeart " + (props.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
```

```
.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-.773.098-1.52.
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>
-                       { props.qtLikes }
+                       { props.totalLikes }
                    </button>
                </footer>
            </article>
        )
    }
```

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

 # src/model/services/LoginService.js

```
 const URL_API = process.env.REACT_APP_URL_API

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

 export async function logar(usuario, senha) {
+    if(TOKEN === null) {
         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) {
+            TOKEN = tokenLogin
             localStorage.setItem('TOKEN', tokenLogin)
             return tokenLogin
         }

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

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

 # src/model/services/TweetsService.js

```
+import * as LoginService from './LoginService.js'
+
+const TWEETS_URL = "https://twitelum-api.herokuapp.com/tweets"
+
+export const carrega = () =>
+    fetch(`${TWEETS_URL}?X-AUTH-TOKEN=${LoginService.getToken()}`)
+    .then(
+        response => response.json()
+    )
+
+export const adiciona = conteudo =>
+    fetch(
+        `${TWEETS_URL}?X-AUTH-TOKEN=${LoginService.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/${idTweetQueVaiSerRemovido}?X-AUTH-TOKEN=${Lo
ginService.getToken()}`
+        ,{ method: "DELETE" }
+    )
+    .then(data => data.json())
+
+export const like = idDoTweet =>
+    fetch(
+        `https://twitelum-api.herokuapp.com/tweets/${idDoTweet}/like?X-AUTH-TOKEN=${LoginService
.getToken()}`
+        ,{ method: "POST" }
+    )
+    .then(response => response.json())
```

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

 # src/pages/Home.js

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

 import {
     Cabecalho,
     NavMenu,
     Dashboard,
     Widget,
     TrendsArea,
```

```
     Tweet
 } from '../components/index.js'

 import { withPermissao } from  './withPermissao.js'

 import { FormNovoTweet } from '../components/FormNovoTweet/FormNovoTweet.js'

+import * as TweetsService from '../model/services/TweetsService.js'
+
 function AreaNovoTweet(props) {
     return (
         <React.Fragment>
             <p> Coisas novas em cima </p>
             <br />
             <br />
             <FormNovoTweet onNovoTweet={props.onNovoTweet} />
             <br />
             <br />
             <p> Coisas novas embaixo </p>
         </React.Fragment>
     )
 }

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

     function adicionaTweet(novoTweet) {
-        setListaTweets([ novoTweet , ...listaTweets])
+        TweetsService.adiciona(novoTweet)
+            .then((novoTweetInfo) => {
+                setListaTweets([ novoTweetInfo , ...listaTweets])
+            })
     }

+    useEffect(() => {
+        TweetsService.carrega()
+            .then((tweets) => {
+                setListaTweets(tweets)
+            })
+    }, [])
+
     return (
         <React.Fragment>
             <Cabecalho>
                 <NavMenu usuario="@artdiniz"></NavMenu>
             </Cabecalho>

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

                 <Dashboard posicao="centro">
                     <Widget>
                         <div className="tweetsArea">
-                            { listaTweets.map(conteudo => (
-                                <Tweet qtLikes={ "oi" } >
-                                    {conteudo}
```

```
+                           { listaTweets.map(tweetInfo => (
+                               <Tweet {...tweetInfo} >
+                                   {tweetInfo.conteudo}
                                </Tweet>
                            )) }
                        </div>
                    </Widget>
                </Dashboard>
            </div>
        </React.Fragment>
    )
}

export const Home = withPermissao(HomeSemAutenticacao)
```

Capítulo 22

EXERCÍCIO: REMOVENDO TWEETS – MAIS UMA PROP DE CALLBACK.

22.1 PASSO A PASSO COM CÓDIGO

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

 # src/components/Tweet/Tweet.js

```
  import React from 'react'
  import PropTypes from 'prop-types'

  import './tweet.css'

  Tweet.propTypes = {
      qtLikes: PropTypes.number
  }

  export function Tweet(props) {
      // RETORNA um Elemento
      return (
          <article className="tweet">
              <div className="tweet__cabecalho">
                  <img className="tweet__fotoUsuario" src={props.usuario.foto} alt="" />
                  <span className="tweet__nomeUsuario">{props.usuario.nome} {props.usuario.sobreno
me}</span>
                  <a href="/"><span className="tweet__userName">@{props.usuario.login}</span></a>
              </div>
              <p className="tweet__conteudo">{ props.children }</p>
              <footer className="tweet__footer">
                  <button className="btn btn--clean">
                      <svg className={"icon icon--small iconHeart " + (props.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
.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-.773.098-1.52.
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>
                      { props.totalLikes }
                  </button>
```

```
+                {
+                    props.removivel &&
+                    <button onClick={props.removeHandler} className="btn btn--blue btn--remove">
+                        X
+                    </button>
+                }
            </footer>
        </article>
    )
}
```

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

 # src/pages/Home.js

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

import {
    Cabecalho,
    NavMenu,
    Dashboard,
    Widget,
    TrendsArea,
    Tweet
} from '../components/index.js'

import { withPermissao } from  './withPermissao.js'

import { FormNovoTweet } from '../components/FormNovoTweet/FormNovoTweet.js'

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

function AreaNovoTweet(props) {
    return (
        <React.Fragment>
            <p> Coisas novas em cima </p>
            <br />
            <br />
            <FormNovoTweet onNovoTweet={props.onNovoTweet} />
            <br />
            <br />
            <p> Coisas novas embaixo </p>
        </React.Fragment>
    )
}

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

    function adicionaTweet(novoTweet) {
        TweetsService.adiciona(novoTweet)
            .then((novoTweetInfo) => {
                setListaTweets([ novoTweetInfo , ...listaTweets])
            })
    }

+   function removeTweet(id) {
+       TweetsService
+           .remove(id)
+           .then(() => {
+               setListaTweets(listaTweets.filter(({_id}) => id !== _id))
+           })
```

```
+    }
+
     useEffect(() => {
         TweetsService.carrega()
             .then((tweets) => {
                 setListaTweets(tweets)
             })
     }, [])

     return (
         <React.Fragment>
             <Cabecalho>
                 <NavMenu usuario="@artdiniz"></NavMenu>
             </Cabecalho>

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

                 <Dashboard posicao="centro">
                     <Widget>
                         <div className="tweetsArea">
                             { listaTweets.map(tweetInfo => (
-                                <Tweet {...tweetInfo} >
+                                <Tweet {...tweetInfo} removeHandler={() => removeTweet(tweetInfo
._id)}>
                                     {tweetInfo.conteudo}
                                 </Tweet>
                             )) }
                         </div>
                     </Widget>
                 </Dashboard>
             </div>
         </React.Fragment>
     )
 }

 export const Home = withPermissao(HomeSemAutenticacao)
```

CAPÍTULO 23

EXERCÍCIO: O MODAL DE TWEETS – IMPLEMENTANDO O VISUAL.

23.1 PASSO A PASSO COM CÓDIGO

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

 # src/components/Modal/Modal.js

```
+import React from "react"
+import PropTypes from "prop-types"
+
+import "./modal.css"
+import { Widget } from "../Widget/Widget.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 ' + (props.isAberto ? 'modalActive' : '')}
+        >
+            <div>
+                <Widget>{props.isAberto && props.children()}</Widget>
+            </div>
+        </div>
+    )
+}
+
+Modal.propTypes = {
+    isAberto: PropTypes.bool.isRequired,
+    onFechando: PropTypes.func.isRequired,
+    children: PropTypes.func.isRequired
+}
```

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

 # src/components/Tweet/Tweet.js

```
import React from 'react'
import PropTypes from 'prop-types'

import './tweet.css'
```

```
  Tweet.propTypes = {
      qtLikes: PropTypes.number
  }

  export function Tweet(props) {
      // RETORNA um Elemento
      return (
          <article className="tweet">
              <div className="tweet__cabecalho">
                  <img className="tweet__fotoUsuario" src={props.usuario.foto} alt="" />
                  <span className="tweet__nomeUsuario">{props.usuario.nome} {props.usuario.sobreno
me}</span>
                  <a href="/"><span className="tweet__userName">@{props.usuario.login}</span></a>
              </div>
 -            <p ~~className="tweet__conteudo">{ props.children }</p~~
 +            <p onClick={props.onConteudoClicado} className="tweet__conteudo">{ props.children }<
p>
              <footer className="tweet__footer">
                  <button className="btn btn--clean">
                      <svg className={"icon icon--small iconHeart " + (props.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
.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-.773.098-1.52.
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>
                      { props.totalLikes }
                  </button>
                  {
                      props.removivel &&
                      <button onClick={props.removeHandler} className="btn btn--blue btn--remove">
                          X
                      </button>
                  }
              </footer>
          </article>
      )
  }
```

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

 # src/pages/Home.js

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

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

```
 import { withPermissao } from  './withPermissao.js'

 import { FormNovoTweet } from '../components/FormNovoTweet/FormNovoTweet.js'
+import { Modal } from "../components/Modal/Modal.js"

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

 function AreaNovoTweet(props) {
     return (
         <React.Fragment>
             <p> Coisas novas em cima </p>
             <br />
             <br />
             <FormNovoTweet onNovoTweet={props.onNovoTweet} />
             <br />
             <br />
             <p> Coisas novas embaixo </p>
         </React.Fragment>
     )
 }

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

     function adicionaTweet(novoTweet) {
         TweetsService.adiciona(novoTweet)
             .then((novoTweetInfo) => {
                 setListaTweets([ novoTweetInfo , ...listaTweets])
             })
     }

     function removeTweet(id) {
         TweetsService
             .remove(id)
             .then(() => {
                 setListaTweets(listaTweets.filter(({_id}) => id !== _id))
             })
     }

+    const [tweetModal, setTweetModal] = useState(null)
+
+    function abreModal(tweet) {
+        setTweetModal(tweet)
+    }
+
+    function fechaModal() {
+        setTweetModal(null)
+    }
+
     useEffect(() => {
         TweetsService.carrega()
             .then((tweets) => {
                 setListaTweets(tweets)
             })
     }, [])

     return (
         <React.Fragment>
             <Cabecalho>
                 <NavMenu usuario="@artdiniz"></NavMenu>
             </Cabecalho>
```

```
            <div className="container">
                <Dashboard>
                    <Widget>
                       <AreaNovoTweet onNovoTweet={adicionaTweet} />
                    </Widget>
                    <Widget>
                        <TrendsArea></TrendsArea>
                    </Widget>
                </Dashboard>

                <Dashboard posicao="centro">
                    <Widget>
                        <div className="tweetsArea">
                            { listaTweets.map(tweetInfo => (
-                               <Tweet {...tweetInfo} removeHandler={() => removeTweet(tweetInfo
._id)}>
+                               <Tweet {...tweetInfo} removeHandler={() => removeTweet(tweetInfo
._id)} onConteudoClicado={() => abreModal(tweetInfo)}>
                                    {tweetInfo.conteudo}
                                </Tweet>
                            )) }
                        </div>
                    </Widget>
                </Dashboard>
            </div>
+
+           <Modal isAberto={tweetModal !== null} onFechando={fechaModal}>
+               {() => (
+                   <Tweet {...tweetModal}>
+                       {tweetModal.conteudo}
+                   </Tweet>
+               )}
+           </Modal>
        </React.Fragment>
    )
}

export const Home = withPermissao(HomeSemAutenticacao)
```

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

Capítulo 24

EXERCÍCIO: CURTINDO TWEETS. O PROBLEMA DA SINCRONIZAÇÃO DE ESTADOS.

24.1 PASSO A PASSO COM CÓDIGO

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

 # src/components/Tweet/Tweet.js

```
 -import React from 'react'
 +import React, { useState } from 'react'
  import PropTypes from 'prop-types'

  import './tweet.css'

  Tweet.propTypes = {
      qtLikes: PropTypes.number
  }

  export function Tweet(props) {
      // RETORNA um Elemento
 +
      return (
          <article className="tweet">
              <div className="tweet__cabecalho">
                  <img className="tweet__fotoUsuario" src={props.usuario.foto} alt="" />
                  <span className="tweet__nomeUsuario">{props.usuario.nome} {props.usuario.sobreno
me}</span>
                  <a href="/"><span className="tweet__userName">@{props.usuario.login}</span></a>
              </div>
              <p onClick={props.onConteudoClicado} className="tweet__conteudo">{ props.children }<
p>
              <footer className="tweet__footer">
 -                <button className="btn btn--clean">
 +                <button className="btn btn--clean" onClick={props.onLike}>
                      <svg className={"icon icon--small iconHeart " + (props.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
.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-.773.098-1.52.
```

```
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>
                        { props.totalLikes }
                    </button>
                    {
                        props.removivel &&
                        <button onClick={props.removeHandler} className="btn btn--blue btn--remove">
                            X
                        </button>
                    }
                </footer>
            </article>
        )
    }
```

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

 # src/pages/Home.js

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

import {
    Cabecalho,
    NavMenu,
    Dashboard,
    Widget,
    TrendsArea,
    Tweet
} from '../components/index.js'

import { withPermissao } from  './withPermissao.js'

import { FormNovoTweet } from '../components/FormNovoTweet/FormNovoTweet.js'
import { Modal } from "../components/Modal/Modal.js"

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

function AreaNovoTweet(props) {
    return (
        <React.Fragment>
            <p> Coisas novas em cima </p>
            <br />
            <br />
            <FormNovoTweet onNovoTweet={props.onNovoTweet} />
            <br />
            <br />
            <p> Coisas novas embaixo </p>
        </React.Fragment>
    )
}

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

    function adicionaTweet(novoTweet) {
        TweetsService.adiciona(novoTweet)
            .then((novoTweetInfo) => {
                setListaTweets([ novoTweetInfo , ...listaTweets])
            })
```

```
      }

+     const [tweetModal, setTweetModal] = useState(null)
+
      function removeTweet(id) {
          TweetsService
              .remove(id)
              .then(() => {
                  setListaTweets(listaTweets.filter(({_id}) => id !== _id))
+                 setTweetModal(null)
              })
      }

-     const [tweetModal, setTweetModal] = useState(null)
-
      function abreModal(tweet) {
          setTweetModal(tweet)
      }

      function fechaModal() {
          setTweetModal(null)
      }

      useEffect(() => {
          TweetsService.carrega()
              .then((tweets) => {
                  setListaTweets(tweets)
              })
      }, [])

+     function dahLike(id) {
+         const tweetLikeado = listaTweets.find(({_id}) => id === _id)
+
+         tweetLikeado.likeado = true
+         tweetLikeado.totalLikes = tweetLikeado.totalLikes + 1
+
+         setListaTweets([...listaTweets])
+     }
+
      return (
          <React.Fragment>
              <Cabecalho>
                  <NavMenu usuario="@artdiniz"></NavMenu>
              </Cabecalho>

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

                  <Dashboard posicao="centro">
                      <Widget>
                          <div className="tweetsArea">
                              { listaTweets.map(tweetInfo => (
-                                 <Tweet {...tweetInfo} removeHandler={() => removeTweet(tweetInfo
._id)} onConteudoClicado={() => abreModal(tweetInfo)}>
+                                 <Tweet {...tweetInfo} onLike={() => dahLike(tweetInfo._id)} remo
veHandler={() => removeTweet(tweetInfo._id)} onConteudoClicado={() => abreModal(tweetInfo)}>
```

```
                                    {tweetInfo.conteudo}
                                </Tweet>
                            )) }
                        </div>
                    </Widget>
                </Dashboard>
            </div>

            <Modal isAberto={tweetModal !== null} onFechando={fechaModal}>
                {() => (
-                   <Tweet {...tweetModal}>
+                   <Tweet {...tweetModal} onLike={() => dahLike(tweetModal._id)} removeHandler=
{() => removeTweet(tweetModal._id)}>
                        {tweetModal.conteudo}
                    </Tweet>
                )}
            </Modal>
        </React.Fragment>
    )
}

export const Home = withPermissao(HomeSemAutenticacao)
```

CAPÍTULO 25

EXERCÍCIO: GERENCIAMENTO DE ESTADO COM REDUX. A BASE.

25.1 PASSO A PASSO COM CÓDIGO

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

 # package.json

```
 {
   "name": "twitelum",
   "version": "0.1.0",
   "private": true,
   "dependencies": {
     "prop-types": "^15.7.2",
     "react": "^16.10.2",
     "react-dom": "^16.10.2",
     "react-router-dom": "^5.1.2",
-    "react-scripts": "3.2.0"
+    "react-scripts": "3.2.0",
+    "redux": "^4.0.4"
   },
   "scripts": {
     "start": "react-scripts start",
     "build": "react-scripts build",
     "test": "react-scripts test",
     "eject": "react-scripts eject"
   },
   "eslintConfig": {
     "extends": "react-app"
   },
   "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.js`** na pasta **`src/pages`** faça as seguintes alterações:

 # src/pages/Home.js

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

 import {
     Cabecalho,
     NavMenu,
     Dashboard,
     Widget,
     TrendsArea,
     Tweet
 } from '../components/index.js'

 import { withPermissao } from  './withPermissao.js'

 import { FormNovoTweet } from '../components/FormNovoTweet/FormNovoTweet.js'
 import { Modal } from "../components/Modal/Modal.js"

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

+import { store } from '../store.js'
+
 function AreaNovoTweet(props) {
     return (
         <React.Fragment>
             <p> Coisas novas em cima </p>
             <br />
             <br />
             <FormNovoTweet onNovoTweet={props.onNovoTweet} />
             <br />
             <br />
             <p> Coisas novas embaixo </p>
         </React.Fragment>
     )
 }

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

-    function adicionaTweet(novoTweet) {
-        TweetsService.adiciona(novoTweet)
+    store.subscribe(() => {
+        setListaTweets(store.getState().listaTweets)
+    })
+
+    useEffect(() => {
+        TweetsService.carrega()
+            .then((tweets) => {
+                store.dispatch({
+                    type: "LISTA",
+                    lista: tweets
+                })
+            })
+    }, [])
+
+    function adicionaTweet(textoNovoTweet) {
+        TweetsService.adiciona(textoNovoTweet)
             .then((novoTweetInfo) => {
                 setListaTweets([ novoTweetInfo , ...listaTweets])
             })
     }

     const [tweetModal, setTweetModal] = useState(null)

     function removeTweet(id) {
```

```
            TweetsService
                .remove(id)
                .then(() => {
                    setListaTweets(listaTweets.filter(({_id}) => id !== _id))
                    setTweetModal(null)
                })
        }

        function abreModal(tweet) {
            setTweetModal(tweet)
        }

        function fechaModal() {
            setTweetModal(null)
        }

-       useEffect(() => {
-           TweetsService.carrega()
-               .then((tweets) => {
-                   setListaTweets(tweets)
-               })
-       }, [])
-
        function dahLike(id) {
-           const tweetLikeado = listaTweets.find(({_id}) => id === _id)
-
-           tweetLikeado.likeado = true
-           tweetLikeado.totalLikes = tweetLikeado.totalLikes + 1
-
-           setListaTweets([...listaTweets])
+         store.dispatch({type: "LIKE", id: id})
        }

        return (
            <React.Fragment>
                <Cabecalho>
                    <NavMenu usuario="@artdiniz"></NavMenu>
                </Cabecalho>

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

                    <Dashboard posicao="centro">
                        <Widget>
                            <div className="tweetsArea">
                                { listaTweets.map(tweetInfo => (
                                    <Tweet {...tweetInfo} onLike={() => dahLike(tweetInfo._id)} remo
veHandler={() => removeTweet(tweetInfo._id)} onConteudoClicado={() => abreModal(tweetInfo)}>
                                        {tweetInfo.conteudo}
                                    </Tweet>
                                )) }
                            </div>
                        </Widget>
                    </Dashboard>
                </div>
```

```
                <Modal isAberto={tweetModal !== null} onFechando={fechaModal}>
                    {() => (
                        <Tweet {...tweetModal} onLike={() => dahLike(tweetModal._id)} removeHandler=
{() => removeTweet(tweetModal._id)}>
                            {tweetModal.conteudo}
                        </Tweet>
                    )}
                </Modal>
            </React.Fragment>
        )
    }

    export const Home = withPermissao(HomeSemAutenticacao)
```

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

 # src/store.js

```
+import { createStore } from 'redux'
+
+const ESTADO_INICIAL = { listaTweets: [] }
+
+export const store = createStore(function reducer(estado = ESTADO_INICIAL, acao) {
+
+    if (acao.type === "LISTA") {
+        return {
+            listaTweets: acao.lista
+        }
+    }
+
+    if (acao.type === "LIKE") {
+        const tweetLikeado = estado.listaTweets.find(({_id}) => acao.id === _id)
+
+        if(tweetLikeado) {
+            tweetLikeado.likeado = true
+            tweetLikeado.totalLikes = tweetLikeado.totalLikes + 1
+
+            return {
+                listaTweets: [...estado.listaTweets]
+            }
+        }
+    }
+
+    return estado
+})
```

CAPÍTULO 26

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

26.1 PASSO A PASSO COM CÓDIGO

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

 # src/pages/Home.js

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

import {
    Cabecalho,
    NavMenu,
    Dashboard,
    Widget,
    TrendsArea,
    Tweet
} from '../components/index.js'

import { withPermissao } from  './withPermissao.js'

import { FormNovoTweet } from '../components/FormNovoTweet/FormNovoTweet.js'
import { Modal } from "../components/Modal/Modal.js"

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

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

function AreaNovoTweet(props) {
    return (
        <React.Fragment>
            <p> Coisas novas em cima </p>
            <br />
            <br />
            <FormNovoTweet onNovoTweet={props.onNovoTweet} />
            <br />
            <br />
            <p> Coisas novas embaixo </p>
        </React.Fragment>
    )
}
```

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

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

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

     function adicionaTweet(textoNovoTweet) {
         TweetsService.adiciona(textoNovoTweet)
             .then((novoTweetInfo) => {
                 setListaTweets([ novoTweetInfo , ...listaTweets])
             })
     }

     const [tweetModal, setTweetModal] = useState(null)

     function removeTweet(id) {
         TweetsService
             .remove(id)
             .then(() => {
                 setListaTweets(listaTweets.filter(({_id}) => id !== _id))
                 setTweetModal(null)
             })
     }

     function abreModal(tweet) {
         setTweetModal(tweet)
     }

     function fechaModal() {
         setTweetModal(null)
     }

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

     return (
         <React.Fragment>
             <Cabecalho>
                 <NavMenu usuario="@artdiniz"></NavMenu>
             </Cabecalho>

             <div className="container">
                 <Dashboard>
                     <Widget>
                        <AreaNovoTweet onNovoTweet={adicionaTweet} />
                     </Widget>
                     <Widget>
                         <TrendsArea></TrendsArea>
```

```
                    </Widget>
                </Dashboard>

                <Dashboard posicao="centro">
                    <Widget>
                        <div className="tweetsArea">
                            { listaTweets.map(tweetInfo => (
                                <Tweet {...tweetInfo} onLike={() => dahLike(tweetInfo._id)} remo
veHandler={() => removeTweet(tweetInfo._id)} onConteudoClicado={() => abreModal(tweetInfo)}>
                                    {tweetInfo.conteudo}
                                </Tweet>
                            )) }
                        </div>
                    </Widget>
                </Dashboard>
            </div>

            <Modal isAberto={tweetModal !== null} onFechando={fechaModal}>
                {() => (
                    <Tweet {...tweetModal} onLike={() => dahLike(tweetModal._id)} removeHandler=
{() => removeTweet(tweetModal._id)}>
                        {tweetModal.conteudo}
                    </Tweet>
                )}
            </Modal>
        </React.Fragment>
    )
}

export const Home = withPermissao(HomeSemAutenticacao)
```

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

 # src/store.js

```
import { createStore } from 'redux'

const ESTADO_INICIAL = { listaTweets: [] }

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

    if (acao.type === "LISTA") {
        return {
-           listaTweets: acao.lista
+           listaTweets: acao.payload.lista
        }
    }

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

        if(tweetLikeado) {
            tweetLikeado.likeado = true
            tweetLikeado.totalLikes = tweetLikeado.totalLikes + 1

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

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

CAPÍTULO 27

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

27.1 PASSO A PASSO COM CÓDIGO

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

 # package.json

```
 {
   "name": "twitelum",
   "version": "0.1.0",
   "private": true,
   "dependencies": {
     "prop-types": "^15.7.2",
     "react": "^16.10.2",
     "react-dom": "^16.10.2",
     "react-router-dom": "^5.1.2",
     "react-scripts": "3.2.0",
-    "redux": "^4.0.4"
+    "redux": "^4.0.4",
+    "redux-thunk": "^2.3.0"
   },
   "scripts": {
     "start": "react-scripts start",
     "build": "react-scripts build",
     "test": "react-scripts test",
     "eject": "react-scripts eject"
   },
   "eslintConfig": {
     "extends": "react-app"
   },
   "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.js`** na pasta **`src/pages`** faça as seguintes alterações:

 # src/pages/Home.js

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

import {
    Cabecalho,
    NavMenu,
    Dashboard,
    Widget,
    TrendsArea,
    Tweet
} from '../components/index.js'

import { withPermissao } from  './withPermissao.js'

import { FormNovoTweet } from '../components/FormNovoTweet/FormNovoTweet.js'
import { Modal } from "../components/Modal/Modal.js"

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

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

function AreaNovoTweet(props) {
    return (
        <React.Fragment>
            <p> Coisas novas em cima </p>
            <br />
            <br />
            <FormNovoTweet onNovoTweet={props.onNovoTweet} />
            <br />
            <br />
            <p> Coisas novas embaixo </p>
        </React.Fragment>
    )
}

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

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

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

    function adicionaTweet(textoNovoTweet) {
        TweetsService.adiciona(textoNovoTweet)
            .then((novoTweetInfo) => {
                setListaTweets([ novoTweetInfo , ...listaTweets])
            })
```

```
    }

    const [tweetModal, setTweetModal] = useState(null)

    function removeTweet(id) {
        TweetsService
            .remove(id)
            .then(() => {
                setListaTweets(listaTweets.filter(({_id}) => id !== _id))
                setTweetModal(null)
            })
    }

    function abreModal(tweet) {
        setTweetModal(tweet)
    }

    function fechaModal() {
        setTweetModal(null)
    }

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

    return (
        <React.Fragment>
            <Cabecalho>
                <NavMenu usuario="@artdiniz"></NavMenu>
            </Cabecalho>

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

                <Dashboard posicao="centro">
                    <Widget>
                        <div className="tweetsArea">
                            { listaTweets.map(tweetInfo => (
                                <Tweet {...tweetInfo} onLike={() => dahLike(tweetInfo._id)} remo
veHandler={() => removeTweet(tweetInfo._id)} onConteudoClicado={() => abreModal(tweetInfo)}>
                                    {tweetInfo.conteudo}
                                </Tweet>
                            )) }
                        </div>
                    </Widget>
                </Dashboard>
            </div>

            <Modal isAberto={tweetModal !== null} onFechando={fechaModal}>
                {() => (
                    <Tweet {...tweetModal} onLike={() => dahLike(tweetModal._id)} removeHandler=
{() => removeTweet(tweetModal._id)}>
                        {tweetModal.conteudo}
                    </Tweet>
                )}
            </Modal>
```

```
            </React.Fragment>
        )
    }

    export const Home = withPermissao(HomeSemAutenticacao)
```

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: [] }

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

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

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

        if(tweetLikeado) {
            tweetLikeado.likeado = true
            tweetLikeado.totalLikes = tweetLikeado.totalLikes + 1

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

    return estado
-})
+}
+
+export const store = createStore(
+    reducer,
+    applyMiddleware(
+        thunkMiddleware
+    )
+)

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

```
 export const criaAcaoLike = (id) => {
+    // objeto
     return {
         type: "LIKE",
         payload: {
             id: id
         }
     }
 }
+
+//Thunk Action Creator
+export const criaAcaoCarrega = () => {
+    return (dispatch) => {
+        TweetsService
+            .carrega()
+            .then((tweets) => {
+                dispatch(criaAcaoLista(tweets))
+            })
+    }
+}
```

CAPÍTULO 28

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

28.1 PASSO A PASSO COM CÓDIGO

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

 # src/ducks/listaTweets/listaTweets.js

```
+
+import * as TweetsService from '../../model/services/TweetsService.js'
+
+const LISTA_INICIAL = []
+
+const LISTA = "listaTweets/LISTA"
+const LIKE = "listaTweets/LIKE"
+
+// ducks
+export function listaTweetsReducer(estadoListaTweets = LISTA_INICIAL, acao) {
+     if (acao.type === LISTA) {
+        return acao.payload.lista
+    }
+
+    if (acao.type === LIKE) {
+        const tweetLikeado = estadoListaTweets.find(({_id}) => acao.payload.id === _id)
+
+        if(tweetLikeado) {
+            tweetLikeado.likeado = true
+            tweetLikeado.totalLikes = tweetLikeado.totalLikes + 1
+
+            return [...estadoListaTweets]
+        }
+    }
+
+    return estadoListaTweets
+}
+
+// Action Creators
+export const criaAcaoLista = (tweets) => {
+    return {
+        type: LISTA,
```

```
+        payload: {
+            lista: tweets
+        }
+    }
+}
+
+export const criaAcaoLike = (id) => {
+    // objeto
+    return {
+        type: LIKE,
+        payload: {
+            id: id
+        }
+    }
+}
+
+// Thunk Action Creator
+export const criaAcaoCarrega = () => {
+    return (dispatch) => {
+        TweetsService
+            .carrega()
+            .then((tweets) => {
+                dispatch(criaAcaoLista(tweets))
+            })
+    }
+}
```

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

 # src/pages/Home.js

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

 import {
     Cabecalho,
     NavMenu,
     Dashboard,
     Widget,
     TrendsArea,
     Tweet
 } from '../components/index.js'

 import { withPermissao } from  './withPermissao.js'

 import { FormNovoTweet } from '../components/FormNovoTweet/FormNovoTweet.js'
 import { Modal } from "../components/Modal/Modal.js"

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

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

 function AreaNovoTweet(props) {
     return (
         <React.Fragment>
             <p> Coisas novas em cima </p>
             <br />
             <br />
             <FormNovoTweet onNovoTweet={props.onNovoTweet} />
             <br />
```

```
            <br />
            <p> Coisas novas embaixo </p>
        </React.Fragment>
    )
}

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

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

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

    function adicionaTweet(textoNovoTweet) {
        TweetsService.adiciona(textoNovoTweet)
            .then((novoTweetInfo) => {
                setListaTweets([ novoTweetInfo , ...listaTweets])
            })
    }

    const [tweetModal, setTweetModal] = useState(null)

    function removeTweet(id) {
        TweetsService
            .remove(id)
            .then(() => {
                setListaTweets(listaTweets.filter(({_id}) => id !== _id))
                setTweetModal(null)
            })
    }

    function abreModal(tweet) {
        setTweetModal(tweet)
    }

    function fechaModal() {
        setTweetModal(null)
    }

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

    return (
        <React.Fragment>
            <Cabecalho>
                <NavMenu usuario="@artdiniz"></NavMenu>
            </Cabecalho>

            <div className="container">
                <Dashboard>
                    <Widget>
                       <AreaNovoTweet onNovoTweet={adicionaTweet} />
                    </Widget>
                    <Widget>
                        <TrendsArea></TrendsArea>
                    </Widget>
                </Dashboard>
```

```
                    <Dashboard posicao="centro">
                        <Widget>
                            <div className="tweetsArea">
                                { listaTweets.map(tweetInfo => (
                                    <Tweet {...tweetInfo} onLike={() => dahLike(tweetInfo._id)} remo
veHandler={() => removeTweet(tweetInfo._id)} onConteudoClicado={() => abreModal(tweetInfo)}>
                                        {tweetInfo.conteudo}
                                    </Tweet>
                                )) }
                            </div>
                        </Widget>
                    </Dashboard>
                </div>

                <Modal isAberto={tweetModal !== null} onFechando={fechaModal}>
                    {() => (
                        <Tweet {...tweetModal} onLike={() => dahLike(tweetModal._id)} removeHandler=
{() => removeTweet(tweetModal._id)}>
                            {tweetModal.conteudo}
                        </Tweet>
                    )}
                </Modal>
            </React.Fragment>
        )
    }

    export const Home = withPermissao(HomeSemAutenticacao)
```

3. 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 reducer(estado = ESTADO_INICIAL, acao) {
-
-    if (acao.type === "LISTA") {
-        return {
-            listaTweets: acao.payload.lista
-        }
-    }
-
-    if (acao.type === "LIKE") {
-        const tweetLikeado = estado.listaTweets.find(({_id}) => acao.payload.id === _id)
-
-        if(tweetLikeado) {
-            tweetLikeado.likeado = true
-            tweetLikeado.totalLikes = tweetLikeado.totalLikes + 1
-
-            return {
-                listaTweets: [...estado.listaTweets]
-            }
-        }
-    }
-
```

```
-    return estado
-}
+import { listaTweetsReducer } from './ducks/listaTweets/listaTweets.js'

 export const store = createStore(
-    reducer,
+    combineReducers({
+        listaTweets: listaTweetsReducer
+    }),
     applyMiddleware(
         thunkMiddleware
     )
 )
-
-
-// Action Creators
-export const criaAcaoLista = (tweets) => {
-    return {
-        type: "LISTA",
-        payload: {
-            lista: tweets
-        }
-    }
-}
-
-export const criaAcaoLike = (id) => {
-    // objeto
-    return {
-        type: "LIKE",
-        payload: {
-            id: id
-        }
-    }
-}
-
-//Thunk Action Creator
-export const criaAcaoCarrega = () => {
-    return (dispatch) => {
-        TweetsService
-            .carrega()
-            .then((tweets) => {
-                dispatch(criaAcaoLista(tweets))
-            })
-    }
-}
```