

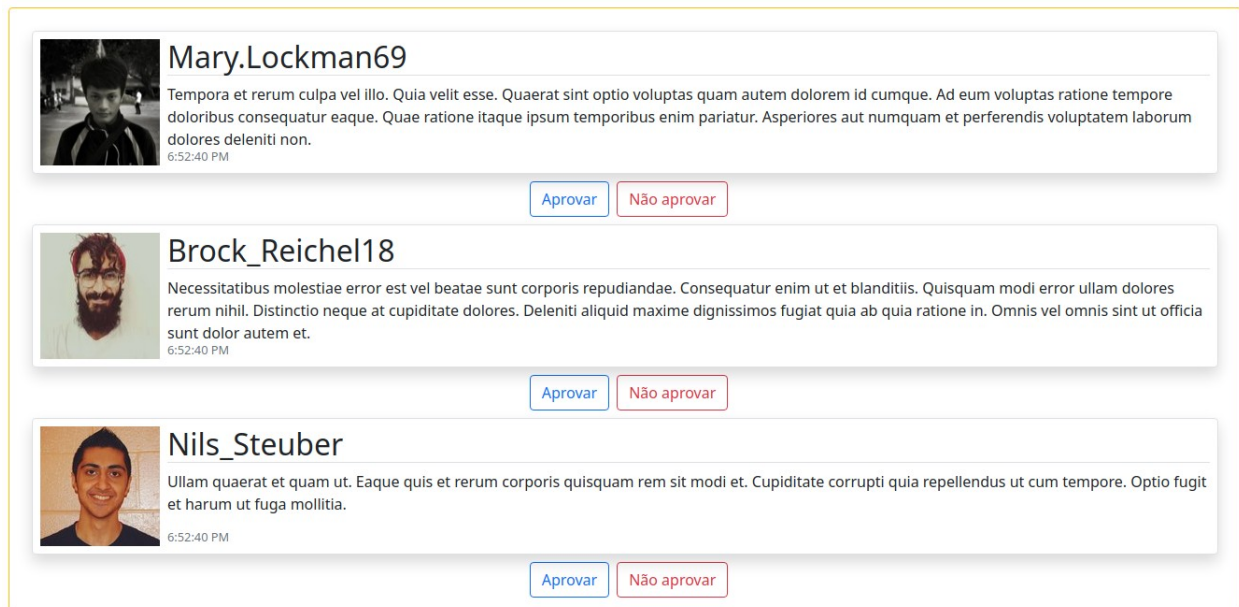
# ReactJS

## Componentes - Props

### Exercícios

1. Crie uma aplicação ReactJS que exibe uma lista de três comentários feitos por usuários de uma rede social. Ela pode ser parecida com o que exibe a Figura 1.1.

Figura 1.1



As características desejadas são as seguintes.

1.1 Cada comentário tem uma foto do usuário, seu nome, data e hora de realização e texto.

1.2 Cada comentário deve ser exibido como um “cartão”.

1.3 Cada comentário pode ser aprovado ou reprovado.

1.3 A aplicação deve possuir os seguintes componentes ReactJS.

- ListaComentarios - exibe conteúdo genérico, especificado por meio de sua propriedade **children**. Seu conteúdo deverá ser uma coleção de três comentários.

- Cartao - exibe conteúdo genérico, especificado por meio de sua propriedade **children**. O conteúdo de um cartão deve ser um comentário e botões para feedback.
- Comentario. Representa um comentário. Suas características lhe são entregues via **props**.
- Feedback. Um componente que exibe dois botões para o usuário dar o seu feedback.

## Solução

1. (Criando a aplicação e instalando dependências) Crie a aplicação com

```
npx create-react-app lista-de-comentarios
```

A seguir, navegue até o diretório que abriga seu conteúdo com

```
cd lista-de-comentarios
```

Apague todo o conteúdo da pasta **src** (somente dela) e crie um arquivo chamado **index.js**. Seu conteúdo inicial aparece no Bloco de Código 1.1.

### Bloco de Código 1.1

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

const App = () => {

  return (
    <div>
      </div>
  )
}

ReactDOM.render(
  <App />,
  document.querySelector('#root')
)
```

A aplicação terá as seguintes dependências

- Bootstrap
- faker (para gerar conteúdos como fotos, textos, nomes, datas aleatoriamente)

Instale-as com

`npm install bootstrap`

e

`npm install faker`

Importe ambas no arquivo `index.js`, como mostra o Bloco de Código 1.2.

Bloco de Código 1.2

```
import React from 'react'
import ReactDOM from 'react-dom'
import 'bootstrap/dist/css/bootstrap.min.css'
import faker from 'faker'
const App = () => {
  ...
```

**2 (O componente Comentario)** O componente Comentario se encarrega de exibir os dados (nome, texto, data e foto) que caracterizam um comentário. Veja a sua definição no Bloco de Código 2.1. Para criá-lo, crie um arquivo chamado `Comentario.js` na pasta **src**.

### Bloco de Código 2.1

```
import React from 'react'

export default function Comentario({nome, texto, data, foto}) {
  return (
    // flex para filhos serem dispostos na horizontal, flex-direction é row por padrão
    <div className="d-flex">
      <img src={foto} />
      {/* margem à esquerda, zero padding e ajuste de position para que os filhos se
      posicionem de maneira "absoluta" */}
      <div className="ms-2 p-0 position-relative">
        <h2 className="border-bottom">{nome}</h2>
        <p>{texto}</p>
        {/* zero unidades de medida a partir de baixo e do começo (esquerda) */}
        <p style={{fontSize: '0.8rem'}} className="text-muted position-absolute bottom-0
start-0 m-0">{data}</p>
      </div>
    </div>
  )
}
```

**3 (O componente Cartao)** O componente cartão é bastante simples. Ele apenas define detalhes como borda e sombra. O conteúdo que exibe é aquele recebido por meio da propriedade **children**. Veja o Bloco de Código 3.1. A sua definição deve ser feita em um arquivo chamado **Cartao.js** que deve ser criado na pasta **src**.

### Bloco de Código 3.1

```
import React from 'react'

export default function Cartao(props) {
  return (
    <div className={estilos.principal}>
      {props.children}
    </div>
  )
}

const estilos = {
  principal: 'card border rounded m-2 p-2 shadow'
}
```

**4 (O componente ListaComentarios)** O componente ListaComentarios é semelhante ao componente Cartao. Ele exibirá o conteúdo que receber na propriedade **children**. Seu conteúdo será uma coleção de cartões. Veja o Bloco de Código 4.1.

Bloco de Código 4.1

```
import React from 'react'

export default function ListaComentarios({children}) {
  return (
    <div className={estilos.principal}>
      {children}
    </div>
  )
}

const estilos = {
  principal: 'container border border-warning rounded my-3 p-3'
}
```

**5 (O componente Feedback)** O componente Feedback será construído de maneira bastante genérica. Ele possui dois botões cujos textos exibidos e funções associadas podem ser especificadas via **props**. Veja o Bloco de Código 5.1.

Bloco de Código 5.1

```
import React from 'react'

export default function Feedback({funcaoOK, funcaoNOK, textoOK, textoNOK}) {
  return (
    <div className="d-flex">
      <button className="mx-2 btn btn-outline-primary"
onClick={funcaoOK}>{textoOK}</button>
      <button className="btn btn-outline-danger"
onClick={funcaoNOK}>{textoNOK}</button>
    </div>
  )
}
```

**6 (O componente principal)** O componente principal, definido no arquivo **index.js** tem como componente raiz um **ListaComentarios**. Ele especifica a coleção de cartões a serem exibidos, cada qual responsável por exibir um componente **Comentario** e um componente **Feedback**.

- Comece importando os componentes como no Bloco de Código 6.1.

Bloco de Código 6.1

```
import React from 'react'
import ReactDOM from 'react-dom'
import Comentario from './Comentario'
import Cartao from './Cartao'
import Feedback from './Feedback'
import ListaComentarios from './ListaComentarios'
import 'bootstrap/dist/css/bootstrap.min.css'
import faker from 'faker'
const App = () => {
  ...
```

- O componente **ListaComentarios** tem a ele aplicada uma classe do tipo **container** do Bootstrap, o que visa tratar aspectos de responsividade. Assim, seu conteúdo utilizará as conhecidas classes **row** e **col-\*** deste framework. Veja o Bloco de Código 6.2. Nele, começamos a definição do corpo do componente principal. Teremos três linhas, cada qual com o objetivo de exibir um dos comentários.

## Bloco de Código 6.2

```
...
const App = () => {
  return (
    <ListaComentarios>
      <div className="row">
        <div className="col-12">

          </div>
        </div>
        <div className="row">
          <div className="col-12">

            </div>
          </div>
          <div className="row">
            <div className="col-12">

              </div>
            </div>
          </div>
        </div>
      </ListaComentarios>
    )
  }
  ...
}
```

- O componente Feedback é idêntico para todos os comentários. Por isso, definiremos seus textos e funções uma única vez. O próprio componente será definido uma única vez e reutilizado. Veja o Bloco de Código 6.3.

### Bloco de Código 6.3

```
...
const App = () => {
  const funcaoOK = () => alert('Comentário aprovado!')
  const funcaoNOK = () => alert('Comentário não aprovado!')
  const textoOK = 'Aprovar'
  const textoNOK = 'Não aprovar'
  const feedbackComponent = <Feedback funcaoOK={funcaoOK}
funcaoNOK={funcaoNOK} textoOK={textoOK} textoNOK={textoNOK} />
  return (
    <ListaComentarios>
  )
}
```

- A seguir, definimos um componente Cartao que será filho da primeira linha. Ele abriga um Comentario e um Feedback. Repare no uso das funções **faker** ao especificar as **propriedades** do Comentario. Veja o Bloco de Código 6.4.

### Bloco de Código 6.4

```
...
const App = () => {
  return (
    <ListaComentarios>
      <div className="row">
        <div className="col-12">
          <Cartao>
            <Comentario foto={faker.image.avatar()} nome={faker.internet.userName()}
data={new Date(faker.time.recent()).toLocaleTimeString()} texto={faker.lorem.sentences()} />
            <div className="d-flex justify-content-center">
              {feedbackComponent}
            </div>
          </Cartao>
        </div>
      </div>
    <div className="row">

```

- O conteúdo das duas linhas restantes é exatamente o mesmo: um Comentario e um Feedback, ambos filhos do mesmo Cartao. Veja o Bloco de Código 6.5.



## Bloco de Código 6.5

```
...
<ListaComentarios>
  <div className="row">
    <div className="col-12">
      <Cartao>
        <Comentario foto={faker.image.avatar()} nome={faker.internet.userName()}
data={new Date(faker.time.recent()).toLocaleTimeString()} texto={faker.lorem.sentences()} />
        <div className="d-flex justify-content-center">
          {feedbackComponent}
        </div>
      </Cartao>
    </div>
  </div>
  <div className="row">
    <div className="col-12">
      <Cartao>
        <Comentario foto={faker.image.avatar()} nome={faker.internet.userName()}
data={new Date(faker.time.recent()).toLocaleTimeString()} texto={faker.lorem.sentences()} />
        <div className="d-flex justify-content-center">
          {feedbackComponent}
        </div>
      </Cartao>
    </div>
  </div>
  <div className="row">
    <div className="col-12">
      <Cartao>
        <Comentario foto={faker.image.avatar()} nome={faker.internet.userName()}
data={new Date(faker.time.recent()).toLocaleTimeString()} texto={faker.lorem.sentences()} />
        <div className="d-flex justify-content-center">
          {feedbackComponent}
        </div>
      </Cartao>
    </div>
  </div>
</ListaComentarios>
```