

# **Bootcamp: Desenvolvimento Front End**

## Desafio do módulo

	XXX	$\times$			$\times$	$\times \times \times$
Módulo 3	React					
	$\times$					

# **Objetivos**

Exercitar os seguintes conceitos trabalhados no Módulo:

- ✓ Criação de componentes com React.
- ✓ Utilização de functional components com hooks.

### Enunciado

Criar uma aplicação com React para simular a interface principal do Twitter (react-twitter).

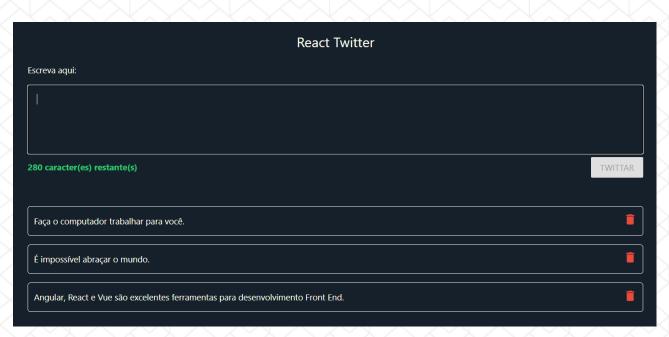
#### **Atividades**

Os alunos deverão desempenhar as seguintes atividades:

- Criar o projeto com o pacote create-react-app ou utilizar o projeto \_react-projetobase, já disponibilizado pelo professor no fórum do módulo. Esta última é a forma recomendada pelo professor.
- Utilizar, preferencialmente, Functional Components com Hooks. Você pode até utilizar Class Components, mas o questionário possui somente questões relacionadas aos Functional Components com Hooks.
- 3. Será fornecido pelo professor um backend que utiliza a biblioteca json-server.
- 4. O usuário pode criar tweets com, no máximo, 280 caracteres. O app deve permitir a inclusão de mais caracteres, mas deve ao mesmo tempo impedir a criação do

tweet. Isso deve ser demonstrado de forma visual para o usuário, conforme sugestões apresentadas nas imagens.

- A interface gráfica fica à critério do aluno. O foco da avaliação serão os conceitos relacionados a React Hooks.
- 6. Tecla de atalho para twittar, quando permitido: Ctrl + Enter.
- Ao clicar no ícone da lixeira em vermelho, o tweet correspondente deve ser imediatamente excluído.
- 8. Tanto a inserção quanto a exclusão devem ser refletidos no backend.
- 9. Após a inclusão de um tweet, faça com que o <textarea> fique com o foco, melhorando a experiência do usuário.
- 10. As imagens abaixo podem servir de orientação para o aluno.

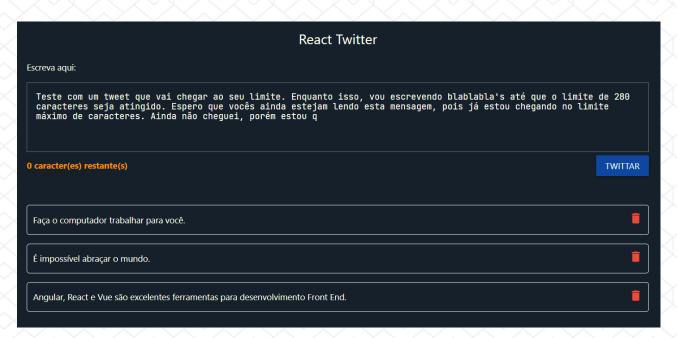


Estado inicial do app, indicando que não é possível twittar sem ao menos um caractere válido.





Enquanto houver até 10 caracteres restantes, o indicador permanece em verde.



Entre 9 e 0 caracteres restantes, o indicador fica em laranja.



Se o usuário ultrapassa o limite, a indicação continua em **vermelho** e o usuário fica impossibilitado de twittar.

Seguem algumas <u>dicas com base na implementação feita pelo professor</u>, que foi feita com a utilização de <u>React Hooks:</u>

- Lembre-se de que o objetivo dos desafios é sair da zona de conforto e pensar fora da caixa.
- 2. Utilize a biblioteca <u>uuid</u> para gerar id's únicos. Esse é o padrão do backend. Leia a documentação e utilize a função v4().
- 3. Sugiro utilizar o <u>axios</u> para as inclusões (POST) e exclusões (DELETE) no backend.
- 4. Minha divisão de componentes:
  - a. App.js, que comporta <Twitter />.
  - b. Twitter.js, que contém todo o estado e toda a lógica da aplicação.
  - c. Tweet.js, que contém os tweets e é usado por <Twitter />.
- 5. Utilizei, como estado:
  - a. O tweet atual do usuário, iniciando como " (string vazia).

- b. A lista de tweets, iniciando como [] (array vazio).
- 6. Em Twitter.js, criei a área de digitação como um <textarea>. No React, o comportamento é um pouco diferente do HTML nativo, e é bem semelhante a um input (com a utilização de value e onChange). Além disso, utilizei onKeyUp para monitorar o Ctrl + Enter. Para verificar se a tecla Ctrl está pressionada, pesquise por ctrlKey.
- 7. Para que o input receba o foco após a inclusão de um tweet, utilizei useEffect monitorando a lista de tweets. Um simples document.querySelector do <textarea> resolve o problema.
- 8. Não deixe que a aplicação permita a inclusão de um tweet que possui somente espaços em branco e/ou quebras de linha.
- 9. Lembre-se de refletir as persistências tanto no backend quanto no frontend.
- 10. Para excluir um tweet, utilize array.filter com o id do tweet.

### **Respostas Finais**

Os alunos deverão desenvolver a prática e, depois, responder às questões objetivas.