

≡ 05

Evitando instrução duplicada

Tatiana aprendeu a lidar com possíveis erros em requisições e escreveu o seguinte esboço de código:

```
$.get('http://localhost:3000/frases', function(retorno) {  
  
    console.log('requisição efetuada com sucesso');  
    console.log('FIM');  
  
}).fail(function() {  
    console.log('requisição fracassou!');  
    console.log('FIM');  
});
```

Ela imprime uma mensagem de sucesso e uma mensagem de fracasso, contudo, veja que nos dois casos ela também imprime a mensagem "FIM" para indicar o fim da requisição.

Qual das opções abaixo altera o código de Tatiana para que exibe em um único lugar a mensagem `FIM` logo depois da mensagem de sucesso ou fracasso?

A

```
$.get('http://localhost:3000/frases', function(retorno) {  
  
    console.log('requisição efetuada com sucesso');  
  
}).fail(function() {  
  
    console.log('requisição fracassou!');  
  
})  
console.log('FIM');
```



B

```
$.get('http://localhost:3000/frases', function(retorno) {  
  
    console.log('requisição efetuada com sucesso');  
  
}).fail(function() {  
  
    console.log('requisição fracassou!');  
  
}).always(function() {  
    console.log('FIM');  
});
```



```
$.get('http://localhost:3000/frases', function(retorno) {  
  
    console.log('requisição efetuada com sucesso');
```



```
}).fail(function() {  
  
    console.log('requisição fracassou!');  
  
}).always(function() {  
    console.log('FIM');  
});
```

Isso mesmo!

A função passada para `always` será sempre chamada independente de a requisição foi bem ou mal sucedida.

PRÓXIMA ATIVIDADE