

PROF. MARCOS CALDAS
DESENVOLVIMENTO
DE SISTEMAS

PROF. MARCOS CALDAS
DESENVOLVIMENTO
DE SISTEMAS

QUALIDADE E TESTE DE SOFTWARE

Prof. Marcos Caldas

PROF. MARCOS CALDAS
DESENVOLVIMENTO
DE SISTEMAS

PROF. MARCOS CALDAS
DESENVOLVIMENTO
DE SISTEMAS

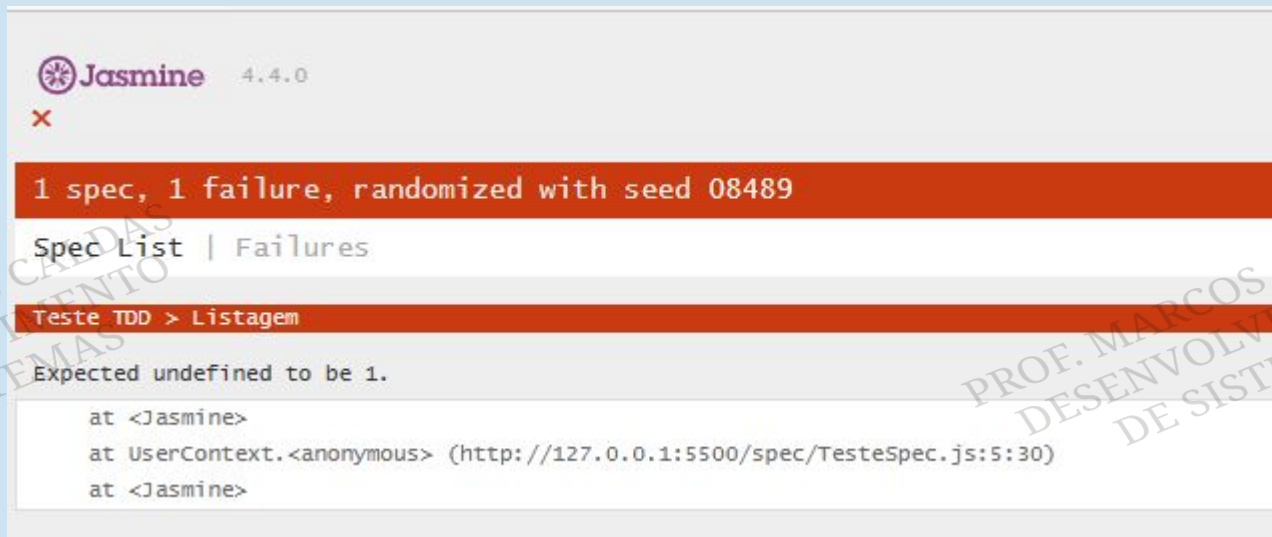
Fizz buzz é um jogo de palavras em grupo para as crianças aprenderem sobre divisão.

Os jogadores se revezam para contar de forma incremental, substituindo qualquer número divisível por três pela palavra "fizz" e qualquer número divisível por cinco pela palavra "buzz".

Se o número é divisível por 3 e por 5 = fizz buzz.

```
JS TesteSpec.js X
spec > JS TesteSpec.js
1 // pasta spec é onde fica o teste automatizado
2
3 describe("Teste TDD",function(){
4     it("Listagem", function(){
5         expect(inteiro(1)).toBe(1);
6     });
7 });
```

```
src > JS Teste.js > ...
1 //src é onde fica o código fonte
2
3 function inteiro(inteiro){
4
5     };
```




src > JS Teste.js > ...

1 //src é onde fica o código fonte

2

3 ✓ function inteiro(inteiro){

4  return 1; // solução mínima para o teste

5 };




4.4.0

1 spec, 0 failures, randomized with seed 53441

Teste TDD

• Listagem

```
JS Teste.js JS TesteSpec.js X
spec > JS TesteSpec.js > ...
1 // pasta spec é onde fica o teste automatizado
2
3 describe("Teste TDD",function(){
4     it("Listagem", function(){
5         expect(inteiro(1)).toBe(1);
6         expect(inteiro(2)).toBe(2);
7     });
8 });
```

 **Jasmine** 4.4.0

X

1 spec, 1 failure, randomized with seed 28745

Spec List	Failures
Teste TDD > Listagem	
Expected 1 to be 2.	
at <Jasmine>	
at UserContext.<anonymous> (http://127.0.0.1:5500/spec/TesteSpec.js:6:30)	
at <Jasmine>	

src > JS Teste.js > ...

```
1 //src é onde fica o código fonte
2
3 function inteiro(inteiro){
4     //return 1; // solução mínima para o teste
5
6     return inteiro; // generalizando e passando o próprio inteiro
7 };
```

 Jasmine 4.4.0

1 spec, 0 failures, randomized with seed 40744

Teste TDD
• Listagem

spec > **js** TesteSpec.js > ...

```
1 // pasta spec é onde fica o teste automatizado
2
3 describe("Teste TDD",function(){
4     it("Listagem", function(){
5         expect(inteiro(1)).toBe(1);
6         expect(inteiro(2)).toBe(2);
7         expect(inteiro(3)).toBe("fizz");
8     });
9 });
```

 **Jasmine** 4.4.0

×

1 spec, 1 failure, randomized with seed 13513

Spec List | Failures

Teste TDD > Listagem

Expected 3 to be 'fizz'.

at <Jasmine>

at UserContext.<anonymous> (http://127.0.0.1:5500/spec/TesteSpec.js:7:30)

at <Jasmine>

```
src > JS Teste.js > ...
1 //src é onde fica o código fonte
2
3 function inteiro(inteiro){
4     //return 1; // solução mínima para o teste
5
6     if(inteiro == 3){
7         return "fizz"
8     }
9
10    return inteiro; // generalizando e passando o próprio inteiro
11
12    };
```



Jasmine

4.4.0

1 spec, 0 failures, randomized with seed 68975

Teste TDD
• Listagem

spec > **Js** TesteSpec.js > ...

```
1  // pasta spec é onde fica o teste automatizado
2
3      describe("Teste TDD",function(){
4          it("Listagem", function(){
5              expect(inteiro(1)).toBe(1);
6              expect(inteiro(2)).toBe(2);
7              expect(inteiro(3)).toBe("fizz");
8              expect(inteiro(4)).toBe(4);
9          });
10     });
```

O teste com o 4 irá passar normalmente.

```
spec > js TesteSpec.js > ...
1 // pasta spec é onde fica o teste automatizado
2
3 describe("Teste TDD",function(){
4     it("Listagem", function(){
5         expect(inteiro(1)).toBe(1);
6         expect(inteiro(2)).toBe(2);
7         expect(inteiro(3)).toBe("fizz");
8         expect(inteiro(4)).toBe(4);
9         expect(inteiro(5)).toBe("buzz");
10    });
11 });
```

 Jasmine 4.4.0

×

1 spec, 1 failure, randomized with seed 67551

Spec List | Failures

Teste TDD > Listagem

Expected 5 to be 'buzz'.


at <Jasmine>

at UserContext.<anonymous> (http://127.0.0.1:5500/spec/TesteSpec.js:9:30)

at <Jasmine>

src > JS Teste.js > ...

```
1 //src é onde fica o código fonte
2
3 function inteiro(inteiro){
4     //return 1; // solução mínima para o teste
5
6     if(inteiro == 3){
7         return "fizz"
8     }
9     if(inteiro == 5){
10        return "buzz"
11    }
12
13    return inteiro; // generalizando e passando o próprio inteiro
14
15 };
```



```
spec > js TesteSpec.js > ...  
1 // pasta spec é onde fica o teste automatizado  
2  
3 describe("Teste TDD",function(){  
4   it("Listagem", function(){  
5     expect(inteiro(1)).toBe(1);  
6     expect(inteiro(2)).toBe(2);  
7     expect(inteiro(3)).toBe("fizz");  
8     expect(inteiro(4)).toBe(4);  
9     expect(inteiro(5)).toBe("buzz");  
10    expect(inteiro(6)).toBe("fizz");  
11  });  
12 });
```

 **Jasmine** 4.4.0



1 spec, 1 failure, randomized with seed 34325

Spec List | Failures

Teste TDD > Listagem

Expected 6 to be 'fizz'.

at <Jasmine>

at UserContext.<anonymous> (http://127.0.0.1:5500/spec/TesteSpec.js:10:30)

at <Jasmine>

src > JS Teste.js > ...

1 //src é onde fica o código fonte

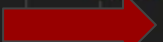
2

3 function inteiro(inteiro){

4 //return 1; // solução mínima para o teste

5

6 //if(inteiro == 3){

7  if(inteiro %3 == 0){ // generalizando

8 return "fizz"

9 }

10 if(inteiro == 5){

11 return "buzz"

12 }


13

14 return inteiro; // generalizando e passando o próprio inteiro

15

16 };

```
spec > js TesteSpec.js > ...  
1 // pasta spec é onde fica o teste automatizado  
2  
3 describe("Teste TDD",function(){  
4   it("Listagem", function(){  
5     expect(inteiro(1)).toBe(1);  
6     expect(inteiro(2)).toBe(2);  
7     expect(inteiro(3)).toBe("fizz");  
8     expect(inteiro(4)).toBe(4);  
9     expect(inteiro(5)).toBe("buzz");  
10    expect(inteiro(6)).toBe("fizz");  
11    expect(inteiro(10)).toBe("buzz");  
12  });  
13 });  
14
```

 **Jasmine** 4.4.0

✖

1 spec, 1 failure, randomized with seed 70978

Spec List | Failures


Teste TDD > Listagem

Expected 10 to be 'buzz'.

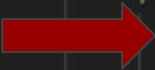
at <Jasmine>

at UserContext.<anonymous> (http://127.0.0.1:5500/spec/TesteSpec.js:11:31)

at <Jasmine>

src >  Teste.js > ...

```
1 //src é onde fica o código fonte
2
3 function inteiro(inteiro){
4     //return 1; // solução mínima para o teste
5
6     //if(inteiro == 3){
7         if(inteiro %3 == 0){ // generalizando
8             return "fizz"
9         }
10        //if(inteiro == 5){
11            if(inteiro %5 == 0){ // generalizando
12                return "buzz"
13            }
14
15        return inteiro; // generalizando e passando o próprio inteiro
16    }
17 }
```



```
spec > JS TesteSpec.js > ...
1 // pasta spec é onde fica o teste automatizado
2
3 describe("Teste TDD",function(){
4     it("Listagem", function(){
5         expect(inteiro(1)).toBe(1);
6         expect(inteiro(2)).toBe(2);
7         expect(inteiro(3)).toBe("fizz");
8         expect(inteiro(4)).toBe(4);
9         expect(inteiro(5)).toBe("buzz");
10        expect(inteiro(6)).toBe("fizz");
11        expect(inteiro(10)).toBe("buzz");
12        expect(inteiro(15)).toBe("fizzbuzz");
13    });
14 });
```

 **Jasmine** 4.4.0

 1 spec, 1 failure, randomized with seed 86633

Spec List | Failures

Teste TDD > Listagem

Expected 'fizz' to be 'fizzbuzz'.

- at <Jasmine>
- at UserContext.<anonymous> (http://127.0.0.1:5500/spec/TesteSpec.js:12:31)
- at <Jasmine>

src > JS Teste.js > ...

```
1 //src é onde fica o código fonte
2
3 function inteiro(inteiro){
4     //return 1; // solução mínima para o teste
5
6     if(inteiro % 3 == 0 && inteiro % 5 == 0){
7         return "fizzbuzz"
8     }
9
10    //if(inteiro == 3){
11        if(inteiro % 3 == 0){ // generalizando
12            return "fizz"
13        }
14        //if(inteiro == 5){
15            if(inteiro % 5 == 0){ // generalizando
16                return "buzz"
17            }
18
19        return inteiro; // generalizando e passando o próprio inteiro
20
21    };
```