



INSTITUTO FEDERAL

Mato Grosso do Sul

Matcher (comparadores) – parte 2

- Verifica se uma exceção é lançada por um método.
- Não realiza a validação em detalhe o tipo da exceção lançada, apenas certifica que um erro ocorreu na execução da função.
- Deve ser utilizada quando deseja apenas certificar que um erro ocorreu, sem se preocupar com detalhes como tipo ou mensagem de erro.

toThrow

```
describe('Suíte de testes do toThrow', function () {  
  var comErro = function () {  
    return numero * 10;  
  };  
  var semErro = function (numero) {  
    return numero * 10;  
  };  
  it('deve validar o uso do matcher toThrow', function () {  
    expect(comErro).toThrow();  
    expect(semErro).toThrow();  
  });  
});
```

toThrowError

- Verifica se uma exceção é lançada por um método.
- Valida o tipo da exceção lançada.
- Valida a mensagem de erro contida na exceção.
- Suporta expressão regular na validação da mensagem de erro da exceção.
- Deve ser utilizada para maior controle do erro lançado.

toThrowError

```
describe('Suíte de testes do toThrowError', function(){
  var calcularDobro = function(numero){
    if(numero <= 0){
      throw new TypeError('O número deve ser maior que 0.');
```

Falha manual (Fail)

- Permite interromper um teste lançando um erro.
- O Jasmine possui a função 'fail' para falhar manualmente um teste.
- Utilizamos a falha manual para certificar que uma operação não desejada não seja executada.

Falha manual (Fail)

```
describe('Testa a função fail',function(){
  var operacao = function(deveExecutar, callBack){
    if(deveExecutar){
      callBack();
    }
  };
  it('não deve executar a função de callBack',function(){
    operacao(false, function(){
      fail('Função de callback foi executada');
    });
  });
});
```

Executando código antes e depois do teste/suíte

- O Jasmine permite executar códigos antes e depois de um teste / suíte com o uso de funções especiais
- Com o uso dessas funções repetições de códigos podem ser evitadas
- Permite executar um mesmo código antes e/ou depois de cada teste / suíte

beforeEach (antes do teste)

- Função JavaScript global do Jasmine que é executada antes de cada teste
- Por ser executada antes de cada teste, serve para inicializar ou reiniciar um status
- Pode também executar uma ação antes de cada teste

beforeEach (antes do teste)

```
describe('Suíte de testes do beforeEach',function(){  
  var contador = 0;  
  beforeEach(function(){  
    contador++;  
  });  
  it('deve incrementar o contador para 1',function(){  
    expect(contador).toEqual(1);  
  });  
  it('deve incrementar o contador para 2',function(){  
    expect(contador).toEqual(2);  
  });  
});
```

afterEach (depois do teste)

- Função JavaScript global do Jasmine que é executada depois de cada teste
- Por ser executada depois de cada teste, serve para reiniciar um status
- Pode também executar uma ação depois de cada teste

```
describe('Suíte de testes do afterEach',function(){
  var contador = 0;
  beforeEach(function(){
    contador++;
  });
  afterEach(function(){
    contador = 0;
  });
  it('deve incrementar o contador para 1',function(){
    expect(contador).toEqual(1);
  });
  it('deve reinicializar o contador para 1',function(){
    expect(contador).toEqual(1);
  });
});
```

beforeAll (executar uma única vez antes)

- Função JavaScript global do Jasmine que é executada uma única vez antes da execução dos testes
- Por ser executada antes de todos os testes serve para inicializar um status, criar objetos

```
describe('Suíte de testes do beforeAll',function(){  
  var contador = 0;  
  beforeAll(function(){  
    contador = 10;  
  });  
  beforeEach(function(){  
    contador ++;  
  });  
  it('deve exibir o contador como 11',function(){  
    expect(contador).toEqual(11);  
  });  
  it('deve reinicializar o contador para 1',function(){  
    expect(contador).toEqual(12);  
  });  
});
```

afterAll (executar uma única vez depois)

- Função JavaScript global do Jasmine que é executada uma única vez depois da execução dos testes
- Por ser executada depois de todos os testes serve para limpar algum status
- Na prática não é muito utilizado
- Difícil testar, pois é somente depois que as expects terminam

```
describe('Suíte de testes do beforeAll',function(){
  var contador = 0;
  beforeAll(function(){
    contador = 10;
  });
  afterAll(function(){
    contador =0;
  });
  it('deve exibir o contador como 10',function(){
    expect(contador).toEqual(10);
  });
  it('deve manter o contador para 10',function(){
    expect(contador).toEqual(10);
  });
});
```