

Teste de Mutação

O que o Code Coverage não mostra, os Mutants descobrem.

Eu

GitHub - [lucasleandro1204](#)

Linkedin - [lucasleandro1204](#)

Twitter - [lucasleandro412](#)



O que é Code Coverage

E por que 100% de cobertura não é realmente tão útil assim

```
1 const someOddFunction = (a: Number, b: Number): Boolean => a > b;  
2  
3 test('someOddFunction', () => {  
4   expect(someOddFunction(50, 10)).to.be.true;  
5   expect(someOddFunction(10, 50)).to.be.false;  
6 });
```

```
1 const someOddFunction = (a: Number, b: Number): Boolean => a > b;  
2  
3 test('someOddFunction', () => {  
4   expect(someOddFunction(50, 10)).to.be.true;  
5   expect(someOddFunction(10, 50)).to.be.false;  
6 });
```

```
1 const someOddFunction = (a: Number, b: Number): Boolean => a > b;  
2  
3 test('someOddFunction', () => {  
4   expect(someOddFunction(50, 10)).to.be.true;  
5   expect(someOddFunction(10, 50)).to.be.false;  
6 });
```

```
1 const someOddFunction = (a: Number, b: Number): Boolean => a > b;  
2  
3 test('someOddFunction', () => {  
4   expect(someOddFunction(50, 10)).to.be.true;  
5   expect(someOddFunction(10, 50)).to.be.false;  
6 });
```

The Introdução

1. Ve se tudo passa
2. Executa a mutação
3. Ve se tudo quebra

**Como matar algo que eu
não conheço 101**

```
1 $ npm install left-pad --save
```

↓ weekly downloads

2,584,275



JavaScript joke™



Aritmética

```
1 // original
2 const sum = (a: Number, b: Number): Boolean => a + b;
3
4 // mutated
5 const sum = (a: Number, b: Number): Boolean => a - b;
```

Atribuição

```
1 // original
2 let a = 1;
3 let b = 2;
4
5 a += b;
6
7 // mutated
8 let a = 1;
9 let b = 2;
10
11 a -= b;
```

Condicional

```
1 // original
2 if (a > b) {
3     // do something
4 }
5
6 // mutated
7 if (true) {
8     // do something
9 }
10
11
12 if (false) {
13     // do something
14 }
```


Bloco

```
1 // original
2 const someOddFunction = (a: Number, b: Number): Any => {
3   if (a > b) {
4     return 'not allowed';
5   }
6
7   if (a == b) {
8     return true;
9   }
10
11   return 100;
12 };
13
14 // mutated
15 const someOddFunction = (a: Number, b: Number): Any => {
16   //
17 };
```

Quando que ele entra no desenvolvimento?

Alguns sistemas

- **Stryker** (JS, C#, Scala)
- **Pit** (Java)
- **Infection** (PHP)cosmic-ray
- **Cosmic ray** (Python)

Teste de mutação é bem simples:

Mutantes (bugs, falhas, erros, modificações) são inseridos automaticamente no seu código, depois os seus testes são executados.

Se o seu teste falhar, o mutante foi morto.
Se o seu teste passar, o mutante sobreviveu.

A qualidade do seu teste é proporcional a porcentagem de mutantes mortos.

FIM