
Teste Unitário

O que é?

- Um teste unitário basicamente é o teste da menor parte testável de um programa.
- Se você programa em uma linguagem que suporte paradigma funcional por exemplo, a menor parte testável do seu código deve ser uma função. Então um teste unitário seria o teste de qualquer função. No caso de orientação a objetos seria o teste de um método de seu objeto.

Métodos

- **setup():** Método chamado antes de cada método de teste.
- **tearDown():** – Método chamado depois de cada método de teste


Asserts

| Método | Descrição | Teste passa se |
|---------------------------------|--------------------------------|--------------------------|
| <code>assertEquals(a,b)</code> | Compara dois valores | <code>a.equals(b)</code> |
| <code>assertFalse(a)</code> | Avalia uma expressão booleana | <code>a == false</code> |
| <code>assertTrue(a)</code> | | <code>a == true</code> |
| <code>assertNotNull(a)</code> | Compara uma variável com nulo | <code>a != null</code> |
| <code>assertNull(a)</code> | | <code>a == null</code> |
| <code>assertNotSame(a,b)</code> | Compara dois objetos | <code>a == b</code> |
| <code>assertSame(a,b)</code> | | <code>a != b</code> |
| <code>fail()</code> | Causa uma falha no teste atual | |

Exemplo

```
class TestMath{
    var math: Math? = null

    @Before
    fun setUp() {
        math = Math()
    }

     @Test
    fun testSum() {
        val number1 = 5
        val number2 = 5
        assertEquals( expected: 10, math!!.sum(number1 = number1, number2 = number2))
    }

    @After
    fun tearDown() {
        math = null
    }
}
```

Vamos praticar!