



**GESTÃO E QUALIDADE DE SOFTWARE - ATIVIDADE -
AULA 07**

Integrantes do grupo:

Gabriel Viegas Capecchi – 82213442

Gabriel Mariotti Higa – 822141216

Gabriela Dardis Rodrigues – 822141330

Igor Britto - 822141647

Maria Fernanda Mendes Tobias – 822137255

Renato Peduto Filho - 822126254

**São Paulo
2025**

Testes TDD (Test-Driven Development)

1. Verificar se número é par

```
def eh_par(n):  
    return n % 2 == 0  
  
import unittest  
  
class TestEhPar(unittest.TestCase):  
    def test_numeros_pares(self):  
        self.assertTrue(eh_par(2))  
        self.assertTrue(eh_par(0))  
  
    def test_numeros_impares(self):  
        self.assertFalse(eh_par(1))  
        self.assertFalse(eh_par(7))
```

2. Verificar se número é primo

```
def eh_primo(n):  
    if n <= 1:  
        return False  
    for i in range(2, int(n ** 0.5) + 1):  
        if n % i == 0:  
            return False  
    return True  
  
class TestEhPrimo(unittest.TestCase):  
    def test_numeros_primos(self):  
        self.assertTrue(eh_primo(2))  
        self.assertTrue(eh_primo(13))  
  
    def test_numeros_nao_primos(self):  
        self.assertFalse(eh_primo(1))  
        self.assertFalse(eh_primo(9))
```

3. Calcular média e verificar aprovação

```
def media_aprovacao(nota1, nota2):  
    media = (nota1 + nota2) / 2  
    return "Aprovado" if media >= 6 else "Reprovado"  
  
class TestMediaAprovacao(unittest.TestCase):  
    def test_aprovado(self):  
        self.assertEqual(media_aprovacao(7, 8), "Aprovado")  
  
    def test_reprovado(self):  
        self.assertEqual(media_aprovacao(4, 5), "Reprovado")
```

4. Verificar maioridade

```
def maior_idade(idade):  
    return idade >= 18  
  
class TestMaiorIdade(unittest.TestCase):  
    def test_maior_de_idade(self):  
        self.assertTrue(maior_idade(20))  
  
    def test_menor_de_idade(self):  
        self.assertFalse(maior_idade(16))
```

5. Verificar login

```
def login(usuario, senha):  
    return usuario == "admin" and senha == "1234"  
  
class TestLogin(unittest.TestCase):  
    def test_login_correto(self):  
        self.assertTrue(login("admin", "1234"))  
  
    def test_usuario_errado(self):  
        self.assertFalse(login("user", "1234"))  
  
    def test_senha_errada(self):  
        self.assertFalse(login("admin", "0000"))
```

6. Calcular fatorial

```
def fatorial(n):  
    if n < 0:  
        return None  
    resultado = 1  
    for i in range(1, n + 1):  
        resultado *= i  
    return resultado  
  
class TestFatorial(unittest.TestCase):  
    def test_fatorial_positivo(self):  
        self.assertEqual(fatorial(5), 120)  
  
    def test_fatorial_zero(self):  
        self.assertEqual(fatorial(0), 1)  
  
    def test_fatorial_negativo(self):  
        self.assertIsNone(fatorial(-3))
```

7. Somar dois números

```
def somar(a, b):  
    return a + b  
  
class TestSomar(unittest.TestCase):  
    def test_soma(self):  
        self.assertEqual(somar(2, 3), 5)  
        self.assertEqual(somar(-1, 1), 0)
```

8. Verificar se um número é múltiplo de outro

```
def eh_multiplo(a, b):  
    return a % b == 0  
  
class TestEhMultiplo(unittest.TestCase):  
    def test_multiplo(self):  
        self.assertTrue(eh_multiplo(10, 2))  
  
    def test_nao_multiplo(self):  
        self.assertFalse(eh_multiplo(7, 3))
```

Testes Ágeis - BDD (Behavior-Driven Development)

1. Verificar se número é par

Funcionalidade: Verificar se um número é par

Cenário: Número é par

Dado que eu tenho o número 8

Quando verificarmos se ele é par

Então o resultado deve ser verdadeiro

2. Verificar se número é primo

Funcionalidade: Verificar se um número é primo

Cenário: Número primo

Dado que eu tenho o número 7

Quando verificarmos se ele é primo

Então o resultado deve ser verdadeiro

3. Calcular média e verificar aprovação

Funcionalidade: Calcular média e verificar aprovação

Cenário: Aluno aprovado

Dado que o aluno tirou 7 e 8

Quando calcularmos a média

Então o aluno deve estar aprovado

4. Verificar maioridade

Funcionalidade: Verificar maioridade

Cenário: Pessoa maior de idade

Dado que a idade é 20

Quando verificarmos a maioridade

Então deve retornar verdadeiro

5. Verificar login

Funcionalidade: Verificar login

Cenário: Login com sucesso

Dado que o usuário é 'admin' e a senha é '1234'

Quando realizarmos o login

Então deve retornar verdadeiro

6. Calcular fatorial

Funcionalidade: Calcular fatorial

Cenário: Fatorial de 5

Dado que o número é 5

Quando eu calcular o fatorial

Então o resultado deve ser 120

7. Somar dois números

Funcionalidade: Somar dois números

Cenário: Soma de dois inteiros

Dado que os números são 3 e 5

Quando somarmos os dois

Então o resultado deve ser 8

8. Verificar múltiplo

Funcionalidade: Verificar se um número é múltiplo de outro

Cenário: Verificar múltiplo

Dado que os números são 10 e 2

Quando verificarmos se 10 é múltiplo de 2

Então o resultado deve ser verdadeiro