

DISCIPLINA:

**PROJETO DE SISTEMAS APLICADO AS MELHORES PRÁTICAS EM
QUALIDADE DE SOFTWARE E GOVERNANÇA DE TI**

**AULA 19 – TESTE INTEGRADO DE SOFTWARE (TESTE DE INTEGRAÇÃO)
EXERCÍCIOS**

PROFESSOR:

RENATO JARDIM PARDUCCI

PROFRENATO.PARDUCCI@FIAP.COM.BR

[Renato Parducci - YouTube](#)

**EXERCÍCIOS DE FIXAÇÃO
PROPOSTOS**
**(solucione e tire suas dúvidas com
o professor)**

1. DESAFIO

Crie os testes de Integração dos algoritmos especificados a seguir (crie as classes e os testes com Mockito):

```
//-----
// Calcula o valor de Pi usando serie com N termos
Função Pi(N : inteiro) : real
    declare soma      : real    // variáveis locais
           i,sinal    : inteiro

    soma ← 0.0
    sinal ← 1
    Para i ← 0 até N-1 Faça
        soma ← soma + sinal/(2.0*i+1.0)
        sinal ← -sinal
    Fim Para
    retorne 4*soma
Fim Função
//-----
Algoritmo ImprimePi. Testa função Pi.
Início
    declare A : inteiro

    Escreva("Numero de termos = ")
    Leia(A)
    Escreva("Valor de Pi = ",Pi(A)," com ",A," termos")
Fim
```

2. DESAFIO

Crie os testes de Integração dos subprogramas especificados a seguir (crie as classes e os testes com Mockito):

```
#include <iostream>
using namespace std;
//-----
float divisao(int dividendo, int divisor)
{ float quociente;

    quociente = 1.0 * dividendo / divisor;
    return quociente;
}
//-----
int main()
{ int a=3, b=4;

    cout << a << "/" << b << "="
          << divisao(a,b) << endl;
    cout << b << "/" << a << "="
          << divisao(b,a) << endl;
    return 0;
}
```

3. DESAFIO

Crie os testes de Integração dos subprogramas especificados a seguir (crie as classes e os testes com Mockito):

```
#include <iostream>
using namespace std;
//-----
float f(float x)                /* definicao da funcao f */
{
    return x * x - 3 * x + 2;
}
//-----
int main()                      /* programa principal */
{ float a,r;

    a = 3;
    /* utilizar a funcao numa expressao simples */
    r = f(a);                    /* ativacao de f */
    cout << "Valor de f(" << a << ")=" << r << endl;
    /* utilizar a funcao numa expressao mais elaborada */
    r = f(2*a) + 3 * f(1) + 2 * f(-1); /* ativacao de f */
    cout << r << endl;
    return 0;
}
```

4. DESAFIO

Tente criar um teste de integração para a API da SPTRANS de monitoração do tráfego da cidade de São Paulo ou outra API que você domine.

