Universidade São Judas Tadeu

Gestão e Qualidade de Software – GQS

Lab 04 – Test-Driven Development (TDD)

Prof. Robson Calvetti

Aluno: Pedro Zorzete Brisolla de Campos

# Atividade 1 – Função maior(a, b)

Objetivo: Criar uma função que retorne o maior entre dois números.

## Implementação:

def maior(a, b):  
 return a if a > b else b

## Testes:

def test\_maior\_errado():  
 assert maior(5, 3) == 3 # Falha proposital  
  
def test\_maior\_forcado():  
 assert maior(5, 5) == 5 # Forçado  
  
def test\_maior\_correto():  
 assert maior(10, 7) == 10 # Correto

# Atividade 2 – Função inverter\_string(texto)

Objetivo: Criar uma função que devolve a string invertida.

## Implementação:

def inverter\_string(texto: str) -> str:  
 return texto[::-1]

## Testes:

def test\_inverter\_errado():  
 assert inverter\_string("abc") == "abc" # Falha proposital  
  
def test\_inverter\_forcado():  
 assert inverter\_string("x") == "x" # Forçado  
  
def test\_inverter\_correto():  
 assert inverter\_string("python") == "nohtyp" # Correto

# Atividade 3 – Função valida\_cpf(cpf)

Objetivo: Criar uma função que verifique se a string possui 11 dígitos numéricos.

## Implementação:

def valida\_cpf(cpf: str) -> bool:  
 return cpf.isdigit() and len(cpf) == 11

## Testes:

def test\_valida\_cpf\_errado():  
 assert valida\_cpf("12345678901") is False # Falha proposital  
  
def test\_valida\_cpf\_forcado():  
 assert valida\_cpf("00000000000") is True # Forçado  
  
def test\_valida\_cpf\_correto():  
 assert valida\_cpf("12345") is False # Correto