**PROJETO INTERDISCIPLINAR**

**TESTE DE SISTEMA**

class Ong:

def \_\_init\_\_(self, nome, descricao):

self.nome = nome

self.descricao = descricao

class AplicativoDoacoes:

def \_\_init\_\_(self):

self.ongs = {

"Alimentos": [

Ong("ONG Alimento Bom", "A ONG Alimento Bom arrecada alimentos não perecíveis."),

Ong("Cesta Solidária", "A Cesta Solidária distribui cestas básicas para famílias carentes.")

],

"Roupas": [

Ong("Roupa do Bem", "Roupa do Bem aceita doações de roupas usadas em bom estado."),

Ong("Vestir com Amor", "Vestir com Amor entrega roupas a pessoas em situação de vulnerabilidade.")

],

"Livros": [

Ong("Livros para Todos", "Livros para Todos doa livros para escolas públicas."),

Ong("Biblioteca Comunitária", "A Biblioteca Comunitária aceita doações de livros em bom estado.")

]

}

def selecionar\_doacao(self, tipo):

if tipo in self.ongs:

return self.ongs[tipo]

else:

return []

def mostrar\_descricao\_ong(self, ong):

return ong.descricao

def prazo\_entrega(self):

return "O prazo para entrega é de 5 dias úteis após a confirmação da doação."

# Função de teste do sistema

def teste\_sistema():

app = AplicativoDoacoes()

# Passo 1: Selecionar tipo de doação

tipo\_doacao = "Alimentos"

print(f"Selecionando tipo de doação: {tipo\_doacao}")

# Passo 2: Obter lista de ONGs

ongs\_disponiveis = app.selecionar\_doacao(tipo\_doacao)

assert len(ongs\_disponiveis) > 0, "Erro: Nenhuma ONG encontrada para o tipo de doação."

print(f"ONGs disponíveis para '{tipo\_doacao}':")

for ong in ongs\_disponiveis:

print(f"- {ong.nome}")

# Passo 3: Escolher uma ONG

ong\_escolhida = ongs\_disponiveis[0]

print(f"\nEscolhendo ONG: {ong\_escolhida.nome}")

# Passo 4: Mostrar descrição da ONG

descricao = app.mostrar\_descricao\_ong(ong\_escolhida)

print(f"Descrição da ONG: {descricao}")

# Passo 5: Confirmar escolha e mostrar prazo de entrega

print(app.prazo\_entrega())

print("\nTeste de sistema concluído com sucesso!")

# Executar o teste de sistema

if \_\_name\_\_ == "\_\_main\_\_":

teste\_sistema()

**teste de aceitação**

import unittest

# Simulação de classes e funções do aplicativo

class AplicativoDoacoes:

def \_\_init\_\_(self):

self.usuario\_logado = False

self.tipos\_de\_doacao = ["Alimentos", "Roupas", "Livros"]

self.ongs = {

"Alimentos": ["ONG Alimentos 1", "ONG Alimentos 2"],

"Roupas": ["ONG Roupas 1", "ONG Roupas 2"],

"Livros": ["ONG Livros 1", "ONG Livros 2"]

}

self.descricao\_ongs = {

"ONG Alimentos 1": "Descrição detalhada da ONG Alimentos 1",

"ONG Roupas 1": "Descrição detalhada da ONG Roupas 1",

"ONG Livros 1": "Descrição detalhada da ONG Livros 1"

}

self.prazo\_entrega = {

"ONG Alimentos 1": "7 dias",

"ONG Roupas 1": "5 dias",

"ONG Livros 1": "10 dias"

}

def login(self, usuario, senha):

# Simulação de login (simplificado)

if usuario == "teste" and senha == "1234":

self.usuario\_logado = True

return self.usuario\_logado

def selecionar\_tipo\_de\_doacao(self, tipo):

# Retorna a lista de ONGs disponíveis para o tipo de doação selecionado

return self.ongs.get(tipo, [])

def obter\_descricao\_ong(self, ong):

# Retorna a descrição da ONG selecionada

return self.descricao\_ongs.get(ong, "Descrição não disponível")

def obter\_prazo\_entrega(self, ong):

# Retorna o prazo de entrega da doação para a ONG selecionada

return self.prazo\_entrega.get(ong, "Prazo não disponível")

# Classe de testes unitários e de aceitação

class TestAplicativoDeDoacoes(unittest.TestCase):

def setUp(self):

# Configuração inicial para cada teste

self.app = AplicativoDoacoes()

self.usuario = "teste"

self.senha = "1234"

# Verifica se o login foi bem-sucedido

self.assertTrue(self.app.login(self.usuario, self.senha), "Falha no login do usuário")

def test\_selecao\_tipo\_de\_doacao(self):

# Cenário 1: Verifica a seleção do tipo de doação

tipo\_de\_doacao = "Alimentos"

ongs\_disponiveis = self.app.selecionar\_tipo\_de\_doacao(tipo\_de\_doacao)

self.assertGreater(len(ongs\_disponiveis), 0, f"Nenhuma ONG disponível para {tipo\_de\_doacao}")

print(f"ONGs disponíveis para {tipo\_de\_doacao}: {ongs\_disponiveis}")

def test\_selecao\_ong(self):

# Cenário 2: Verifica a seleção de uma ONG e se a descrição está correta

tipo\_de\_doacao = "Roupas"

ongs\_disponiveis = self.app.selecionar\_tipo\_de\_doacao(tipo\_de\_doacao)

ong\_escolhida = ongs\_disponiveis[0]

descricao = self.app.obter\_descricao\_ong(ong\_escolhida)

self.assertIn(ong\_escolhida, self.app.descricao\_ongs, "ONG não encontrada")

self.assertEqual(descricao, self.app.obter\_descricao\_ong(ong\_escolhida), "Descrição incorreta")

print(f"ONG selecionada: {ong\_escolhida}, Descrição: {descricao}")

def test\_prazo\_entrega(self):

# Cenário 3: Verifica o prazo de entrega da doação

ong\_selecionada = "ONG Alimentos 1"

prazo = self.app.obter\_prazo\_entrega(ong\_selecionada)

self.assertEqual(prazo, "7 dias", "Prazo de entrega incorreto")

print(f"Prazo de entrega para {ong\_selecionada}: {prazo}")

def test\_interface\_acessibilidade(self):

# Cenário 4: Simula a navegação no aplicativo e verifica a acessibilidade das páginas

paginas\_acessiveis = ["Página Inicial", "Página de Doação", "Página de ONGs"]

for pagina in paginas\_acessiveis:

self.assertTrue(self.app.usuario\_logado, f"Usuário não conseguiu acessar {pagina}")

print("Todas as páginas estão acessíveis.")

def test\_fluxo\_completo\_doacao(self):

# Cenário Completo: Testa o fluxo completo desde a escolha da doação até a entrega

print("Iniciando o teste do fluxo completo de doação...")

tipo\_de\_doacao = "Livros"

ongs\_disponiveis = self.app.selecionar\_tipo\_de\_doacao(tipo\_de\_doacao)

self.assertGreater(len(ongs\_disponiveis), 0, "Nenhuma ONG disponível")

ong\_escolhida = ongs\_disponiveis[0]

descricao = self.app.obter\_descricao\_ong(ong\_escolhida)

self.assertIn(ong\_escolhida, self.app.descricao\_ongs, "ONG não encontrada na lista")

prazo = self.app.obter\_prazo\_entrega(ong\_escolhida)

self.assertNotEqual(prazo, "Prazo não disponível", "Prazo de entrega não encontrado")

print(f"Fluxo completo: Tipo de Doação: {tipo\_de\_doacao}, ONG Escolhida: {ong\_escolhida}, Prazo: {prazo}")

# Executando os testes

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()