Entrega do Projeto da PI

02/10/2024

Teste unitários e integração

Testes Unitários

1. Teste de Seleção de Tipo de Doação

import unittest

class DoacaoApp:

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

self.tipo\_doacao\_selecionado = None

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

tipos\_validos = ['alimentos', 'roupas', 'livros']

if tipo in tipos\_validos:

self.tipo\_doacao\_selecionado = tipo

else:

raise ValueError(f'Tipo de doação inválido: {tipo}')

class DoacaoAppTests(unittest.TestCase):

def setUp(self):

self.app = DoacaoApp()

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

tipo\_doacao = 'alimentos'

self.app.selecionar\_tipo\_doacao(tipo\_doacao)

self.assertEqual(self.app.tipo\_doacao\_selecionado, tipo\_doacao)

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

tipo\_doacao = 'brinquedos'

with self.assertRaises(ValueError):

self.app.selecionar\_tipo\_doacao(tipo\_doacao)

unittest.TextTestRunner().run(unittest.TestLoader().loadTestsFromTestCase(DoacaoAppTests))

2. Teste de Filtragem de ONGs

import unittest

class DoacaoApp:

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

self.tipo\_doacao\_selecionado = None

self.ongs = [

{'nome': 'ONG A', 'tipos\_de\_doacao\_aceitos': ['alimentos', 'roupas']},

{'nome': 'ONG B', 'tipos\_de\_doacao\_aceitos': ['livros', 'roupas']},

{'nome': 'ONG C', 'tipos\_de\_doacao\_aceitos': ['alimentos']},

{'nome': 'ONG D', 'tipos\_de\_doacao\_aceitos': ['roupas', 'brinquedos']},

]

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

tipos\_validos = ['alimentos', 'roupas', 'livros']

if tipo in tipos\_validos:

self.tipo\_doacao\_selecionado = tipo

else:

raise ValueError(f'Tipo de doação inválido: {tipo}')

def filtrar\_ongs\_por\_doacao(self, tipo\_doacao):

return [ong for ong in self.ongs if tipo\_doacao in ong['tipos\_de\_doacao\_aceitos']]

class DoacaoAppTests(unittest.TestCase):

def test\_filtragem\_ongs(self):

app = DoacaoApp()

tipo\_doacao = 'roupas'

app.selecionar\_tipo\_doacao(tipo\_doacao)

ongs = app.filtrar\_ongs\_por\_doacao(tipo\_doacao)

for ong in ongs:

self.assertIn(tipo\_doacao, ong['tipos\_de\_doacao\_aceitos'])

unittest.TextTestRunner().run(unittest.TestLoader().loadTestsFromTestCase(DoacaoAppTests))

3. Teste de Exibição de Descrição da ONG

import unittest

class DoacaoApp:

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

self.tipo\_doacao\_selecionado = None

self.ongs = [

{'nome': 'ONG A', 'tipos\_de\_doacao\_aceitos': ['alimentos', 'roupas']},

{'nome': 'ONG B', 'tipos\_de\_doacao\_aceitos': ['livros', 'roupas']},

{'nome': 'ONG C', 'tipos\_de\_doacao\_aceitos': ['alimentos']},

{'nome': 'ONG D', 'tipos\_de\_doacao\_aceitos': ['roupas', 'brinquedos']},

]

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

tipos\_validos = ['alimentos', 'roupas', 'livros']

if tipo in tipos\_validos:

self.tipo\_doacao\_selecionado = tipo

else:

raise ValueError(f'Tipo de doação inválido: {tipo}')

def filtrar\_ongs\_por\_doacao(self, tipo\_doacao):

return [ong for ong in self.ongs if tipo\_doacao in ong['tipos\_de\_doacao\_aceitos']]

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

return ong['descricao']

class DoacaoAppTests(unittest.TestCase):

def test\_exibir\_descricao\_ong(self):

app = DoacaoApp()

ong = {

'nome': 'ONG Exemplo',

'descricao': 'Esta ONG aceita doações de alimentos e roupas.'

}

descricao = app.exibir\_descricao\_ong(ong)

self.assertEqual(descricao, 'Esta ONG aceita doações de alimentos e roupas.')

unittest.TextTestRunner().run(unittest.TestLoader().loadTestsFromTestCase(DoacaoAppTests))

4. Teste de Definição de Prazo de Entrega

import unittest

class DoacaoApp:

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

self.tipo\_doacao\_selecionado = None

self.ongs = [

{'nome': 'ONG A', 'tipos\_de\_doacao\_aceitos': ['alimentos', 'roupas']},

{'nome': 'ONG B', 'tipos\_de\_doacao\_aceitos': ['livros', 'roupas']},

{'nome': 'ONG C', 'tipos\_de\_doacao\_aceitos': ['alimentos']},

{'nome': 'ONG D', 'tipos\_de\_doacao\_aceitos': ['roupas', 'brinquedos']},

]

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

tipos\_validos = ['alimentos', 'roupas', 'livros']

if tipo in tipos\_validos:

self.tipo\_doacao\_selecionado = tipo

else:

raise ValueError(f'Tipo de doação inválido: {tipo}')

def filtrar\_ongs\_por\_doacao(self, tipo\_doacao):

return [ong for ong in self.ongs if tipo\_doacao in ong['tipos\_de\_doacao\_aceitos']]

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

return ong['descricao']

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

return ong.get('prazo\_entrega', 'Prazo não definido')

class DoacaoAppTests(unittest.TestCase):

def test\_prazo\_entrega(self):

app = DoacaoApp()

ong = {'nome': 'ONG Exemplo', 'prazo\_entrega': '5 dias úteis'}

prazo = app.definir\_prazo\_entrega(ong)

self.assertEqual(prazo, '5 dias úteis')

unittest.TextTestRunner().run(unittest.TestLoader().loadTestsFromTestCase(DoacaoAppTests))

Testes de Integração

1. Teste de Integração entre Seleção de Doação e Filtragem de ONGs

import unittest

class DoacaoApp:

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

self.tipo\_doacao\_selecionado = None

self.ongs = [

{'nome': 'ONG A', 'tipos\_de\_doacao\_aceitos': ['alimentos', 'roupas']},

{'nome': 'ONG B', 'tipos\_de\_doacao\_aceitos': ['livros', 'roupas']},

{'nome': 'ONG C', 'tipos\_de\_doacao\_aceitos': ['alimentos']},

{'nome': 'ONG D', 'tipos\_de\_doacao\_aceitos': ['livros', 'roupas', 'brinquedos']},

]

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

tipos\_validos = ['alimentos', 'roupas', 'livros']

if tipo in tipos\_validos:

self.tipo\_doacao\_selecionado = tipo

else:

raise ValueError(f'Tipo de doação inválido: {tipo}')

def filtrar\_ongs\_por\_doacao(self, tipo\_doacao):

return [ong for ong in self.ongs if tipo\_doacao in ong['tipos\_de\_doacao\_aceitos']]

class DoacaoAppIntegrationTests(unittest.TestCase):

def test\_integracao\_selecao\_doacao\_e\_filtragem\_ongs(self):

app = DoacaoApp()

tipo\_doacao = 'livros'

app.selecionar\_tipo\_doacao(tipo\_doacao)

ongs = app.filtrar\_ongs\_por\_doacao(tipo\_doacao)

self.assertTrue(all(tipo\_doacao in ong['tipos\_de\_doacao\_aceitos'] for ong in ongs))

unittest.TextTestRunner().run(unittest.TestLoader().loadTestsFromTestCase(DoacaoAppIntegrationTests))

2. Teste de Integração entre Escolha da ONG e Definição de Prazo

import unittest

class DoacaoApp:

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

self.tipo\_doacao\_selecionado = None

self.ong\_selecionada = None

self.ongs = [

{'nome': 'ONG A', 'tipos\_de\_doacao\_aceitos': ['alimentos', 'roupas'], 'prazo\_entrega': '5 dias úteis'},

{'nome': 'ONG B', 'tipos\_de\_doacao\_aceitos': ['livros', 'roupas'], 'prazo\_entrega': '3 dias úteis'},

{'nome': 'ONG C', 'tipos\_de\_doacao\_aceitos': ['alimentos'], 'prazo\_entrega': '4 dias úteis'},

{'nome': 'ONG D', 'tipos\_de\_doacao\_aceitos': ['alimentos', 'brinquedos'], 'prazo\_entrega': '7 dias úteis'},

]

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

**tipos\_validos = ['alimentos', 'roupas', 'livros']**

**if tipo in tipos\_validos:**

**self.tipo\_doacao\_selecionado = tipo**

**else:**

**raise ValueError(f'Tipo de doação inválido: {tipo}')**

**def filtrar\_ongs\_por\_doacao(self, tipo\_doacao):**

**return [ong for ong in self.ongs if tipo\_doacao in ong['tipos\_de\_doacao\_aceitos']]**

**def selecionar\_ong(self, ong):**

**if ong in self.ongs:**

**self.ong\_selecionada = ong**

**else:**

**raise ValueError("ONG não encontrada.")**

**def definir\_prazo\_entrega(self, ong):**

**return ong.get('prazo\_entrega', 'Prazo não definido')**

**class DoacaoAppIntegrationTests(unittest.TestCase):**

**def test\_integracao\_escolha\_ong\_e\_prazo\_entrega(self):**

**app = DoacaoApp()**

**tipo\_doacao = 'alimentos'**

**app.selecionar\_tipo\_doacao(tipo\_doacao)**

**ong = {'nome': 'ONG D', 'tipos\_de\_doacao\_aceitos': ['alimentos', 'brinquedos'], 'prazo\_entrega': '7 dias úteis'}**

**app.selecionar\_ong(ong)**

**prazo = app.definir\_prazo\_entrega(ong)**

**self.assertEqual(prazo, '7 dias úteis')**

**unittest.TextTestRunner().run(unittest.TestLoader().loadTestsFromTestCase(DoacaoAppIntegrationTests))**