



| | | | |
|--|---|-----------|-----------|
| FGA 0238 - Testes de Software – Turma: | 02 | Semestre: | 2023.2 |
| Nome: | Raquel Temóteo Eucaria Pereira da Costa | Matrícula | 202045268 |
| Equipe: | 10 - Assertivos | | |

Atividade 3 – Desenvolver Testes de Unidade

1. Identificação do Projeto
Sistema de Gestão de Contratos (MEC-Energia)

2. Cobertura de testes

2.1 Suite de testes

```
test_subgroup.py 1, M X
tests > test_subgroup.py > TestContractEndpoint > test_throws_exception_when_suply_voltage_does_not_match_ranges
1  import pytest
2
3  ENDPOINT = '/api/contracts/'
4
5  from utils.subgroup_util import Subgroup
6
7  @pytest.mark.django_db
8  class TestContractEndpoint:
9      def setup_method(self):
10         self.contract_test_supply_voltage_1 = 250
11         self.contract_test_supply_voltage_2 = 100
12         self.contract_test_supply_voltage_3 = 40
13         self.contract_test_supply_voltage_4 = 70
14
15         def test_get_what_subgroup_contract_is_A1(self):
16             assert Subgroup.get_subgroup(self.contract_test_supply_voltage_1) == Subgroup.A1
17
18         def test_get_what_subgroup_contract_is_A2(self):
19             assert Subgroup.get_subgroup(self.contract_test_supply_voltage_2) == Subgroup.A2
20
21         def test_get_what_subgroup_contract_is_A3a(self):
22             assert Subgroup.get_subgroup(self.contract_test_supply_voltage_3) == Subgroup.A3A
23
24         def test_throws_exception_when_suply_voltage_does_not_match_ranges(self):
25             with pytest.raises(Exception) as e:
26                 Subgroup.get_subgroup(self.contract_test_supply_voltage_4)
27
28             assert 'Subgroup not found' in str(e.value)
```



2.2 Cobertura

```
===== test session starts =====
platform linux -- Python 3.10.5, pytest-7.4.3, pluggy-1.3.0
django: settings: mec_energia.settings (from ini)
rootdir: /home/dev/mec-energia-api
configfile: pytest.ini
testpaths: tests
plugins: metadata-3.0.0, order-1.1.0, env-0.8.1, django-4.5.2, cov-4.1.0, html-4.0.2
collected 82 items

tests/settings/test_settings.py ..... [ 8%]
tests/test_date.py .. [ 10%]
tests/test_subgroup.py .... [ 15%]
tests/contracts/contract/test_contract_dates.py .... [ 20%]
tests/contracts/contract/test_contract_endpoint.py ... [ 24%]
```

```
----- coverage: platform linux, python 3.10.5-final-0 -----
Name                                                                    Stmts   Miss Branch BrPart  Cover
-----
contracts/apps.py                                                         4        0      0      0    100%
contracts/migrations/0001_initial.py                                     7        0      0      0    100%
contracts/migrations/0002_contract_consumer_unit.py                     5        0      0      0    100%
contracts/migrations/0003_energybill_consumer_unit.py                   5        0      0      0    100%
...
utils/endpoints_util.py                                                  8        3      0      0     62%
utils/energy_bill_util.py                                               77       33     14      1     52%
utils/subgroup_util.py                                                  20        1     10      0     97%
utils/tariff_util.py                                                    9        8      4      0      8%
utils/user/authentication.py                                             16        7      2      0     50%
utils/user/user_type_util.py                                             18        4     10      4     71%
-----
TOTAL                                                                    337     113     38     14     66%
```

- Stmts - Total de Declarações
- Miss - Declarações não cobertas
- BrPart - Decisões condicionais (if, else, e loops)
- Cover - ((Stmts - Miss) / Stmts * 100%)

3. Método a ser testado

3.1 Link Fork

- https://gitlab.com/raqueleucaria/mec-energia-api/-/blob/develop/utils/subgroup_util.py?ref_type=heads

3.2 Sobre

- Nome: get_subgroup()
- Classe: Subgroup
- Propósito: Determinar a qual subgrupo determinada voltagem de fornecimento (em quilovolts) pertence. Ele percorre a lista de subgrupos definidos na classe, verifica as condições de faixa de voltagem de cada subgrupo e retorna o nome do subgrupo ao qual a voltagem fornecida pertence.

3.3 Código



subgroup_util.py 1.29 KiB

Alterar

Substituir

Excluir



```
1 class Subgroup:
2     AS = 'AS'
3     A4 = 'A4'
4     A3A = 'A3a'
5     A3 = 'A3'
6     A2 = 'A2'
7     A1 = 'A1'
8
9     subgroups = [
10         {
11             "name": AS,
12             "min": 0,
13             "max": 2.3,
14         },
15         {
16             "name": A4,
17             "min": 2.3,
18             "max": 25,
19         },
20         {
21             "name": A3A,
22             "min": 30,
23             "max": 44,
24         },
25         {
26             "name": A3,
27             "min": 69,
28             "max": 69,
29         },
30         {
31             "name": A2,
32             "min": 88,
33             "max": 138,
34         },
35         {
36             "name": A1,
37             "min": 238,
38             "max": None,
39         }
40     ]
41
42
43 def get_subgroup(supply_voltage_in_kv: float):
44     for subgroup in Subgroup.subgroups:
45         if supply_voltage_in_kv >= subgroup['min']:
46             if not subgroup['max']:
47                 return subgroup['name']
48
49             if supply_voltage_in_kv < subgroup['max']:
50                 return subgroup['name']
51
52             if subgroup['min'] == subgroup['max'] and subgroup['min'] == supply_voltage_in_kv:
53                 return subgroup['name']
54
55     raise Exception('Subgroup not found for this supply voltage')
56
57 def get_all_subgroups():
58     return Subgroup.subgroups
```

```
43 def get_subgroup(supply_voltage_in_kv: float):
44     for subgroup in Subgroup.subgroups:
45         if supply_voltage_in_kv >= subgroup['min']:
46             if not subgroup['max']:
47                 return subgroup['name']
48
49             if supply_voltage_in_kv < subgroup['max']:
50                 return subgroup['name']
51
52             if subgroup['min'] == subgroup['max'] and subgroup['min'] == supply_voltage_in_kv:
53                 return subgroup['name']
54
55     raise Exception('Subgroup not found for this supply voltage')
56
57 def get_all_subgroups():
58     return Subgroup.subgroups
```



4. Classe de Teste

3.1 Link Fork

- https://gitlab.com/raqueleucaria/mec-energia-api/-/blob/develop/tests/test_subgroup.py?ref_type=heads

3.2 Código

```

test_subgroup.py 1.02 KiB
[Alterar] [Substituir] [Excluir] [Ícone] [Ícone] [Ícone]

1 import pytest
2
3 ENDPOINT = '/api/contracts/'
4
5 from utils.subgroup_util import Subgroup
6
7 @pytest.mark.django_db
8 class TestContractEndpoint:
9     def setup_method(self):
10         self.contract_test_supply_voltage_1 = 250
11         self.contract_test_supply_voltage_2 = 100
12         self.contract_test_supply_voltage_3 = 40
13         self.contract_test_supply_voltage_4 = 70
14
15     def test_get_what_subgroup_contract_is_A1(self):
16         assert Subgroup.get_subgroup(self.contract_test_supply_voltage_1) == Subgroup.A1
17
18     def test_get_what_subgroup_contract_is_A2(self):
19         assert Subgroup.get_subgroup(self.contract_test_supply_voltage_2) == Subgroup.A2
20
21     def test_get_what_subgroup_contract_is_A3a(self):
22         assert Subgroup.get_subgroup(self.contract_test_supply_voltage_3) == Subgroup.A3A
23
24     def test_throws_exception_when_supply_voltage_does_not_match_ranges(self):
25         with pytest.raises(Exception) as e:
26             Subgroup.get_subgroup(self.contract_test_supply_voltage_4)
27
28         assert 'Subgroup not found' in str(e.value)

```

5. Tabela de decisões/condições

Tabela 1: Decisões e Condições

| ID | Decisão (linha) | Condição | Situação para Verdadeiro | Situação para Falso |
|-----|--------------------|---|---|---|
| CD1 | 45 | supply_voltage_in_kv | Variável for maior ou igual ao subgroup['min'] | Variável for menor ao subgroup['min'] |
| CD2 | 46 | supply_voltage_in_kv | Variável não possui o subgroup['max'] | Variável possui o subgroup['max'] |
| CD3 | 49 | supply_voltage_in_kv | Variável for menor que o subgroup['max'] | Variável for maior ou igual ao subgroup['max'] |
| CD4 | 52 | subgroup['min'] == subgroup['max'] | Variável for igual ao subgroup['max'] | Variável for diferente do subgroup['max'] |
| CD5 | 52 | subgroup['min'] == supply_voltage_in_kv | Variável for igual ao supply_voltage_in_kv | Variável for diferente do supply_voltage_in_kv |
| CD6 | 55 | supply_voltage_in_kv | Variável for menor que o subgroup['min'] | Variável for maior ou igual ao subgroup['min'] |



6. Tabelas verdade, pares de independência e combinações de condições MC/DC

Tabela 2: Tabela Verdade para Decisão da linha 52

| ID | subgroup['min'] == subgroup['max'] | subgroup['min'] == supply_voltage_in_kv | Resultado |
|----|------------------------------------|---|-----------|
| 1 | V | V | V |
| 2 | V | F | F |
| 3 | F | V | F |
| 4 | F | F | F |

Nº de casos de teste = Nº de condições + 1 = 2 + 1 = **3 casos de teste**

Pares de independência para cada condição:

- subgroup['min'] == subgroup['max'] : Teste 1 e 3
- subgroup['min'] == supply_voltage_in_kv: Teste 1 e 2

Combinações obtidas a partir dos pares de independência:

- (CB1) subgroup['min'] == subgroup['max'] **and** subgroup['min'] == supply_voltage_in_kv **V**
- (CB2) subgroup['min'] == subgroup['max'] **and** subgroup['min'] != supply_voltage_in_kv **F**
- (CB3) subgroup['min'] != subgroup['max'] **and** subgroup['min'] == supply_voltage_in_kv **F**

7. Especificação dos Casos de Testes



Tabela 3: Casos de Testes

| ID | Entrada | Saída Esperada | Cobertura (Condição + Situação V ou F) |
|-----|--|---|--|
| CT1 | (-1) supply_voltage_in_kv menor que o subgroup['min'] outros casos irrelevantes | 'Subgroup not found for this supply voltage' | CD6V, CD1F |
| CT2 | (69) supply_voltage_in_kv igual ao subgroup['min'] supply_voltage_in_kv possui subgroup['max'] supply_voltage_in_kv == subgroup['max'] subgroup['min'] == subgroup['max'] and subgroup['min'] == supply_voltage_in_kv | subgroup['name'] = A3 | CD1V, CD4V, CD5V, CD2F, CD3F, CD6F |
| CT3 | (230) supply_voltage_in_kv igual ao subgroup['min'] supply_voltage_in_kv não possui subgroup['max'] supply_voltage_in_kv < subgroup['max'] subgroup['min'] != subgroup['max'] and subgroup['min'] == supply_voltage_in_kv | subgroup['name'] = A1 | CD1V, CD2V, CD3V, CD5V, CD4F, CD6F |
| CT4 | (70) supply_voltage_in_kv maior ao subgroup['min'] supply_voltage_in_kv possui subgroup['max'] supply_voltage_in_kv > subgroup['max'] subgroup['min'] == subgroup['max'] and subgroup['min'] != supply_voltage_in_kv | 'Subgroup not found for this supply voltage' | CD1V, CD4V, CD2F, CD3F,, CD5F, CD6F |

8. Implementação dos Casos de Teste

```
test_subgrupo2.py 1.13 KiB
[Alterar] [Substituir] [Excluir] [Icones]

1 import pytest
2
3 ENDPOINT = '/api/contracts/'
4
5 from utils.subgroup_util import Subgroup
6
7 @pytest.mark.django_db
8 class TestContractEndpoint:
9     def setup_method(self):
10         self.contract_test_supply_voltage_1 = -1 # CT1
11         self.contract_test_supply_voltage_2 = 69 # CT2
12         self.contract_test_supply_voltage_3 = 230 # CT3
13         self.contract_test_supply_voltage_4 = 70 # CT4
14
15     def test_throws_exception_when_suply_voltage_below_minimum(self):
16         with pytest.raises(Exception) as e:
17             Subgroup.get_subgroup(self.contract_test_supply_voltage_1)
18         assert 'Subgroup not found' in str(e.value)
19
20     def test_get_group_minimum_and_maximum_equal(self):
21         assert Subgroup.get_subgroup(self.contract_test_supply_voltage_2) == Subgroup.A3
22
23     def test_get_group_without_maximum_voltage(self):
24         assert Subgroup.get_subgroup(self.contract_test_supply_voltage_3) == Subgroup.A1
25
26     def test_throws_exception_when_suply_voltage_does_not_match_ranges(self):
27         with pytest.raises(Exception) as e:
28             Subgroup.get_subgroup(self.contract_test_supply_voltage_4)
29         assert 'Subgroup not found' in str(e.value)
```

https://gitlab.com/raqueleucaria/mec-energia-api/-/blob/develop/tests/test_subgrupo2.py?ref_type=heads

9. Análises e Resultados

```
===== test session starts =====
platform linux -- Python 3.10.5, pytest-7.4.3, pluggy-1.3.0
django: settings: mec_energia.settings (from ini)
rootdir: /home/dev/mec-energia-api
configfile: pytest.ini
testpaths: tests
plugins: metadata-3.0.0, order-1.1.0, env-0.8.1, django-4.5.2, cov-4.1.0, html-4.0.2
collected 86 items

tests/settings/test_settings.py ..... [ 8%]
tests/test_date.py .. [ 10%]
tests/test_subgroup.py .... [ 15%]
tests/test_subgrupo2.py .... [ 19%]
tests/contracts/contract/test_contract_dates.py .... [ 24%]
tests/energy_data/test_data.py ..... [ 29%]
utils/subgroup_util.py ..... [ 34%]
..... [ 39%]
```

Em relação aos casos de teste (CTs) previamente executados, conforme indicado na imagem no item 2.2, observou-se um aumento de 3% na taxa de sucesso dos testes



realizados, enquanto a cobertura de testes permaneceu inalterada. E a cobertura se manteve a mesma.

10. Pull Request

adicionando novo caso de teste para subgrupo_util

Aberto Raquel Eucaria solicitou mesclagem de [raqueleucaria/mec-energi...](#) em [develop](#) agora

Visão geral 0 Commits - Pipelines 0 Alterações -

0 0

Pipeline de solicitação de mesclagem #1077434782 executando

Pipeline de solicitação de mesclagem executando para `d656b9af`

8 A aprovação é opcional

Pronto para mesclar por membros que podem escrever na ramificação de destino.

- 1 commit e 1 merge commit será adicionado à `develop` (squashes 1 commit).
- A ramificação de origem não será excluída.

https://gitlab.com/lappis-unb/projects/mec-energia/mec-energia-api/-/merge_requests/63

11. Links

Fork do projeto: <https://gitlab.com/raqueleucaria/mec-energia-api>

Commit da

implementação: https://gitlab.com/raqueleucaria/mec-energia-api/-/compare/develop..develop?from_project_id=39291800

Pull Request:

https://gitlab.com/lappis-unb/projects/mec-energia/mec-energia-api/-/merge_requests/63