

Aplicação Doação de Cestas - Fluxo de operação

https://github.com/celsorv/cesta/blob/master/services/doacao_service.py

Hipoteticamente, digamos que uma cesta é composta de:

- 2 pacotes de arroz 5Kg
- 1 pacote de feijão 1Kg
- 1 frasco de óleo de soja
- 1 pacote de açúcar refinado
- 2 pacote de biscoito

Nota:

Para facilitar, digamos que esta igreja só tenha uma cesta para produzir/entregar no mês.

Pessoas acessam a aplicação e manifestam a intenção de doar (Doação Agendada):

João doou 1 pacote de arroz

Léa doou 1 pacote de arroz *(meta de arroz atingida baseado em Doações Agendadas)*

Yuri doou 1 pacote de açúcar *(meta de açúcar atingida baseado em Doações Agendadas)*

Cris doou 1 pacote de biscoito

Alan doou 1 frasco de óleo *(meta de óleo atingida baseado em Doações Agendadas)*

Bia doou 1 pacote de biscoito *(meta de biscoito atingida baseado em Doações Agendadas)*

Status da Meta de Cestas até o momento:

Meta de feijão não alcançada *(zero cestas completas baseado em Doações Agendadas)*

Pessoas comparecem à igreja para concretizar a doação (Doação Recebida)

Léa entregou 1 pacote de arroz

Yuri entregou 1 pacote de açúcar *(meta de açúcar atingida baseado em Doações Recebidas)*

João entregou 1 pacote de arroz *(meta de arroz atingida baseado em Doações Recebidas)*

Cris entregou 1 pacote de biscoito

Bia entregou 1 pacote de biscoito *(ignorado - validade inferior à data mínima exigida)*

Bete* entregou 1 pacote de biscoito *(meta de biscoito atingida baseado em Doações Recebidas)*

Igor* entregou 1 pacote de feijão *(meta de feijão atingida baseado em Doações Recebidas)*

* Doação Direta Recebida

Bete e Igor disseram não ter familiaridade com tecnologia e fizeram a entrega do alimento na igreja sem previamente registrarem na aplicação sua intenção de doar (Doação Agendada). Um voluntário se encarregou de registrar a entrada destes recebimentos na aplicação.

Intenção de doação desconsiderada:

Alan manifestou na aplicação sua intenção de doar 1 frasco de óleo de soja. Passados 11 dias ele ainda não fez a entrega da doação na igreja. A aplicação desconsiderou esta Doação Agendada.

Status da Meta de Cestas até o momento:

Meta de óleo de soja não alcançada *(zero cestas completas baseado em Agendadas/Recebidas)*

Uma pessoa acessa a aplicação, manifestam a intenção de doar e faz a entrega:

Arthur doou (Doação Agendada) 1 frasco de óleo de soja na aplicação e no dia seguinte fez a entrega (Doação Recebida) na igreja.

Status da Meta de Cestas até o momento:

Meta de óleo de soja atingida *(uma cesta completa baseado em Agendadas/Recebidas)*

Testes Automatizados

<https://github.com/celsorv/cesta/tree/master/tests>

Na estrutura da aplicação (GitHub), subpasta tests estão reunidos os testes divididos em 4 arquivos:

[] tests_login.py

Testes que se aplicam ao login de usuário

- test_login_dashboard

Garante que o acesso a url dashboard está levando à tela de login com redirecionamento para a url de dashboard.

- test_login_doacao

Garante que o acesso a url doacao está levando à tela de login com redirecionamento para a url de doação.

[] tests_forms.py

Testes que se aplicam aos formulários de entrada de dados.

- test_id_field_not_present

Garante que o campo id não está sendo disponibilizado na tela de entrada de dados.

- test_all_required_form_fields

Garante que os campos obrigatórios esperados estão presentes com esta condição.

[] test_models.py

Testes que se aplicam aos modelos de dados (models).

- test_create_new_doacao_agendada

Garante que é possível incluir um novo registro em doação agendada.

- test_quantity_must_be_greater_than_zero

Garante que uma quantidade zero não é aceita (uma exceção é lançada).

- test_status_must_be_in_list

Garante que o campo status aceite somente uma das opções previstas (uma exceção é lançada).

- test_non_unique_fields

Garante que seja totalmente possível incluir dois registros com dados idênticos (exceto a chave primária id – autoincremental)

- test_expiration_date_must_be_greater_than_or_equal_to_current

Garante que um produto doado tenha como validade no mínimo a data corrente.

[] test_views.py

Testes que se aplicam às views (código responsável por uma determinada rota).

- test_show_form_on_get

Garante que uma dada url esteja acessível e ao ser acessada, carrega o template (html) previsto.

- test_has_form_on_context

Garante que o formulário acessado pela view é aquele que fora previsto.

- test_show_form_with_errors

Garante que a view produza alertas de erro caso um formulário em branco seja submetido.

- test_save_new_doacao_agendada

Garante que a inclusão de uma nova doação agendada esteja funcionando.

- test_redirects_after_save

Garante que haja um redirecionamento para a página de agradecimento após uma doação ser submetida.

Para que seja possível fazer login para realizar os testes envolvendo acessos (get/post) às urls da aplicação, foi utilizado o recurso **Fixtures** que alimenta o banco de dados de testes a partir do conteúdo do arquivo data.json.

https://github.com/celsorv/cesta/blob/master/tests/tests_views.py

```
class DoacaoAgendadaCreateViewTest(TestCase):

    fixtures = ['data.json'] # insere dados no banco de dados de testes

    def setUp(self):
        self.doacao_id = 1
        self.data_post = {'produto': 1, 'quantidade': 1}

        # login de usuário antes de executar os testes
        self.login = self.client.login(email='gerencia@gmail.com', password='12345')
```

<https://github.com/celsorv/cesta/blob/master/data.json>

```
{
  "model": "users.user",
  "pk": 101,
  "fields": {
    "password":
"pbkdf2_sha256$320000$YVzA7q3GE2czSm4FD87nwg$ZRgtHxfsDZSK/hloZmFfqZBq0szZ4Osz381kcbyYrG
0=",
    "last_login": null,
    "is_superuser": false,
    "username": "gerencia",
    "first_name": "",
    "last_name": "",
    "email": "gerencia@gmail.com",
    "is_staff": false,
    "is_active": true,
    "date_joined": "2022-06-13T03:38:39Z",
    "administrador": true,
    "groups": [
      101
    ],
    "user_permissions": []
  }
},
```