

Universidade de Brasília - UnB
Faculdade do Gama - FGA



Plano de Testes

Djorkaeff Alexandre Vilela Pereira - 160026822

Gabriela Medeiros da Silva - 160121817

Ignacio Sanabria Alonso de Caso - 190055511

João Lucas Sousa Reis - 160009758

Larissa Siqueira Sales - 160130883

Lucas Gomes Silva - 160133505

Brasília - DF

2019

Sumário

Introdução	2
Abordagem de Testes	3
Requisitos Funcionais	3
Requisitos Não Funcionais	3
Detalhamento da Abordagem de Testes	3
Ambiente de Teste	4
Programação dos Testes	5
Casos de Teste	6

1. Introdução

O foco do exercício é identificar os defeitos de um produto de software desenvolvido e testado pelos alunos para a funcionalidade de inclusão de novas funções transacionais de Entradas Externas (EEs), Consultas Externas (CEs) e Saídas externas (SEs) de uma contagem de pontos de função.

Nessa fase 3 do TBL, será utilizado o framework do Django para front-end, com HTML e JavaScript, enquanto para back-end será utilizado a linguagem Python.

2. Abordagem de Testes

2.1. Requisitos Funcionais

- **RF001** - O sistema deve permitir a inclusão de função transacional.
- **RF002** - O sistema deve gerar um relatório de funções transacionais cadastradas.

2.2. Requisitos Não Funcionais

- **NF001** - O sistema deve ter combos boxes ou pop ups.
- **NF002** - O sistema deve emitir mensagens acusando erros de preenchimento de acordo com cada campo obrigatório não preenchido.

2.3. Detalhamento da Abordagem de Testes

Tipo do teste:	Funcional
Subtipo do teste:	Requisitos
Objetivo do teste:	Testar a funcionalidade de incluir uma função transacional.
Requisitos que motivaram esse teste:	RF001 - O sistema deve permitir a inclusão de função transacional.

Tipo do teste:	Funcional
Subtipo do teste:	Requisitos
Objetivo do teste:	Testar a funcionalidade de criar um relatório das funções cadastradas.
Requisitos que motivaram esse teste:	RF002 - O sistema deve gerar um relatório de funções transacionais cadastradas.

Tipo do teste:	Não Funcional
Subtipo do teste:	Requisitos

Objetivo do teste:	Testar se a funcionalidade de abrir um popup está sendo carregada com as informações corretas.
Requisitos que motivaram esse teste:	NF001 - O sistema deve ter combos boxes ou pop ups.

Tipo do teste:	Não Funcional
Subtipo do teste:	Requisitos
Objetivo do teste:	Testar se a funcionalidade de gerar uma mensagem de erro está de acordo com o que o campo exige.
Requisitos que motivaram esse teste:	NF002 - O sistema deve emitir mensagens acusando erros de preenchimento de acordo com cada campo obrigatório não preenchido.

2.4. Ferramentas

As ferramentas que serão utilizadas neste projeto são:

- Django, v2.2.2
- Coverage.py, v4.5.3

3. Ambiente de Teste

Os testes unitários serão desenvolvidos individualmente, mantendo um volume baixo de dados e estarão presentes em toda a aplicação, incluindo *model* e *view*.

Os computadores que serão utilizados para o teste deverão ter o sistema operacional *Linux* e as ferramentas *Django* e *Coverage.py* devidamente instalados.

Todo o sistema será desenvolvido utilizando a linguagem *python*

4. Programação dos Testes

O sistema desenvolvido deverá:

- Permitir que o usuário inclua uma função transacional;
- Conter o nome do contador que cadastrou a função transacional;
- Conter a data de cadastramento da função;
- Incluir as informações no Banco de Dados no momento que a funcionalidade for efetivada;
- Gerar um relatório contendo todas as funções cadastradas e suas respectivas informações;
- Organizar o relatório por tipo de função e apresentar o total de pontos (soma) de cada classe de função transacional;
- Permitir que o relatório possa ser filtrado por tipo e complexidade da função.