

Relatório Geral dos Testes

Foram implementados testes para cobrir cinco áreas principais do projeto:

1. **Modelos:** Validação da criação e regras de negócio dos modelos Categoria, Despesa, e Depósito.
 2. **Views:** Testes de autenticação e verificação de status das views, como registrar, login e logout.
 3. **Formulários:** Testes de validação de formulários, garantindo a integridade e funcionalidade dos formulários de registro e login.
 4. **Integração:** Verificação do fluxo completo de registro, login e gestão de categorias.
 5. **Testes de Sistema com Selenium:** Automatização de testes usando Selenium para garantir a funcionalidade do sistema em nível de interface.
-

1. Testes de Modelos

Objetivo: Validar a criação e integridade dos dados dos modelos do sistema, assegurando que as regras de negócios sejam respeitadas.

- **CategoriaModelTest:**

- Verificou a criação de uma categoria com o nome e descrição corretos.
- Testou a possibilidade de uma descrição vazia, confirmando que o sistema aceita categorias sem descrição.

- **DespesaModelTest:**

- Verificou a criação correta de uma despesa com valores e categorias apropriados.
- Testou a regra de valor negativo para uma despesa, onde foi levantada uma exceção de `ValidationError`.
- Verificou se uma despesa pode ser criada com uma data futura.

- **DepositoModelTest:**

- Validou a criação de um depósito, garantindo que a descrição e o valor estão corretos.
 - Testou a regra de valor negativo para um depósito, também levantando uma exceção de `ValidationError`.
-

2. Testes de Views

Objetivo: Verificar o funcionamento correto das views de autenticação e gerenciamento de categorias.

- **AuthViewsTest:**

- Verificou o carregamento das páginas de registro e login com os templates corretos.
 - Testou o fluxo de login com credenciais corretas e verificou o redirecionamento para a página de lista de despesas.
 - Testou login com senha incorreta e confirmou que a view retorna a página de login sem redirecionar.
 - Verificou o comportamento do logout, garantindo que o usuário é desautenticado após o logout.
-

3. Testes de Formulários

Objetivo: Garantir que os formulários de registro e login estejam funcionando corretamente, tanto para entradas válidas quanto inválidas.

- **RegistroFormTest:**

- Verificou que o formulário de registro é validado corretamente com dados válidos (usuário, email e senhas iguais).
- Confirmou que o formulário falha para entradas inválidas (senhas que não coincidem).

- **LoginFormTest:**

- Validou o formulário de login com credenciais corretas.
 - Testou um login inválido e confirmou que o formulário retorna erro.
-

4. Testes de Integração

Objetivo: Garantir o fluxo completo de funcionalidades cruciais do sistema, como registro e login de usuários, além do gerenciamento de categorias.

- **AuthIntegrationTestCase:**

- Verificou o fluxo completo de registro e login do usuário, confirmando que o registro redireciona para o login e que o login redireciona para a lista de despesas.
- Testou o login com um usuário inexistente e garantiu que o sistema não permite.

- **CategoriaIntegrationTestCase:**

- Validou a criação e listagem de categorias. O teste garantiu que uma nova categoria é criada e listada corretamente na interface de gerenciamento de categorias.

5. Testes de Sistema com Selenium

Objetivo: Testar o sistema em nível de interface gráfica para garantir que o fluxo de autenticação funciona conforme o esperado quando acessado pelo navegador.

- **SistemaAuthTestCase:**

- Automatizou os testes de autenticação (registro, login e logout) usando o Selenium WebDriver com o Chrome. A interface foi testada em modo headless, sem abrir a janela do navegador, para garantir que a autenticação no sistema funciona corretamente em diferentes cenários.

Conclusão

Os testes implementados cobrem desde a criação e validação dos dados dos modelos até o fluxo completo de autenticação e gerenciamento de categorias. Além disso, foram realizados testes de formulário e automação de interface para assegurar a robustez do sistema tanto no back-end quanto na interface de usuário.

A estratégia de testes adotada garante a integridade das funcionalidades essenciais do sistema e facilita a identificação e correção de erros ao longo do ciclo de desenvolvimento.
