1. Introdução

1.1 Objetivo do Documento

Este plano de testes tem como objetivo definir a estratégia e o escopo das atividades de teste para o projeto "Meu Bolso", uma aplicação desenvolvida para o controle financeiro pessoal. Ele especifica os tipos de testes a serem realizados, os critérios de aceitação, os recursos necessários e o cronograma de execução para garantir que o sistema esteja livre de erros críticos e atenda aos requisitos de desempenho e segurança.

1.2 Escopo

O escopo dos testes abrange todas as funcionalidades do sistema "Meu Bolso". As principais áreas abordadas incluem:

- **Login Seguro:** Verificação de funcionalidades de autenticação, validação de credenciais e segurança.
- **Registro de Depósitos e Despesas:** Testes de adição, edição, exclusão e consistência de registros financeiros.
- **Visualização de Lista de Entradas Financeiras:** Testes que asseguram a exibição correta dos dados financeiros inseridos.
- **Gráfico Estatístico:** Testes relacionados à geração, exibição e precisão dos gráficos estatísticos baseados nas transações financeiras registradas.

2. Itens de Teste

As áreas de funcionalidade que serão testadas incluem:

- **Login Seguro:**
    - Criação de contas;
    - Autenticação de usuários (login/logout);
    - Segurança de senhas (armazenamento seguro e validação).
- **Registro de Depósitos e Despesas:**
    - Adição, edição e exclusão de depósitos;
    - Adição, edição e exclusão de despesas;
    - Validação de integridade e consistência dos dados.
- **Visualização de Lista:**
    - Exibição de todas as transações em uma lista detalhada;

    - Paginação e filtros;
    - Atualização automática da lista após inserção ou exclusão de dados.
- **Gráficos Estatísticos:**
    - Geração de gráficos financeiros precisos;
    - Verificação da exatidão dos dados exibidos nos gráficos;
    - Responsividade da exibição gráfica em diferentes dispositivos.

3. Abordagem de Testes

3.1 Tipos de Testes

Testes de Unidade

Os testes de unidade têm como objetivo validar que funções e métodos individuais do sistema estão funcionando conforme o esperado. Cada teste irá isolar pequenas unidades de código e avaliar sua precisão. As áreas incluem:

- **Testes de Funções de Registro:**
    - Verificar se as funções que registram depósitos e despesas tratam corretamente diferentes tipos de entrada (números, strings, valores nulos, etc.).
    - Testar cenários de borda, como registro de valores negativos ou grandes.
- **Testes de Funções de Autenticação:**
    - Testar a criação de novos usuários, validação de senhas e recuperação de senhas.
    - Garantir que a autenticação falhe para credenciais inválidas ou incorretas.
- **Testes de Estatísticas:**
    - Avaliar as funções responsáveis por calcular totais de receitas e despesas, assegurando que os cálculos matemáticos estejam corretos.
    - Validar a geração de dados para os gráficos estatísticos (por exemplo, se os dados são agrupados corretamente por categoria ou período).

Testes de Integração

Os testes de integração visam garantir que os módulos individuais do sistema funcionem corretamente em conjunto.

- **Integração do Módulo de Login com o Registro de Dados:**
    - Após um login bem-sucedido, verificar se o sistema permite o registro de dados sem falhas.
    - Validar que as credenciais do usuário autenticado estão sendo usadas corretamente para associar os dados ao perfil.
- **Integração entre Registro de Dados e Visualização:**
    - Testar se as transações registradas aparecem corretamente na interface de visualização.
    - Garantir que a interface gráfica seja atualizada automaticamente após operações de inserção, edição e exclusão.
- **Integração com o Banco de Dados:**
    - Verificar se os dados são armazenados corretamente no banco de dados SQLite.
    - Testar a recuperação precisa dos dados para exibição e relatórios.

Testes de Sistema

Os testes de sistema visam validar o comportamento geral do software em um ambiente completo, simulando o uso real do sistema.

- **Testes Funcionais Completos:**
    - Avaliar todas as funcionalidades principais do sistema, incluindo login, cadastro de transações e visualização de relatórios.
    - Garantir que todas as interações entre módulos estejam funcionando corretamente.
- **Testes de Usabilidade:**
    - Testar a interface do usuário, avaliando se a navegação e o design são intuitivos.
    - Observar a facilidade de uso e a experiência do usuário ao realizar operações comuns, como adicionar transações ou visualizar gráficos.
- **Testes de Segurança:**

    - Garantir que dados sensíveis (como senhas e informações financeiras) estejam protegidos contra acessos não autorizados.
    - Testar a resistência a ataques como SQL Injection e Cross-Site Scripting (XSS).
- **Testes de Performance:**
    - Testar a resposta do sistema sob diferentes níveis de carga (múltiplos usuários simultâneos).
    - Medir o tempo de resposta de operações críticas (login, criação de transações, geração de gráficos), garantindo que esteja abaixo de 5 segundos para operações principais.

4. Critérios de Aceitação

- **Login Seguro:** O usuário deve conseguir criar uma conta e fazer login com credenciais válidas. Credenciais inválidas devem ser rejeitadas corretamente.
- **Registro de Depósitos e Despesas:** Todos os registros devem ser consistentes e salvos corretamente no banco de dados. Edições e exclusões devem refletir imediatamente na interface.
- **Visualização de Lista:** Todos os registros devem ser exibidos corretamente na lista, e filtros e paginação devem funcionar sem erros.
- **Gráficos Estatísticos:** Os gráficos devem refletir com precisão os dados financeiros inseridos, com uma exibição clara e responsiva.

5. Recursos Necessários

- **Ferramentas de Teste:**
    - PyTest para testes de unidade e integração.
    - Selenium ou Cypress para testes de interface e usabilidade.
    - Ferramentas de análise de segurança, como OWASP ZAP.
- **Ambiente de Testes:**
    - Servidor local com banco de dados SQLite.
    - Ambiente de desenvolvimento isolado com a aplicação configurada.