***Easy***

***Finance***

***Controller***

*Aplicativo de controle financeiro pessoal.*

***Requisitos de Sistema***

***Por: Luciano José Kratzer***

**1. Planejamento e Estruturação**

* **1.1 Definição do Escopo:**
  + Defina exatamente quais funcionalidades o aplicativo terá.
  + Priorize as funcionalidades, do essencial ao opcional.
* **1.2 Escolha do Nome do Projeto:**
  + O nome do projeto será **"Easy Finance Controller"**.
* **1.3 Estruturação do Código:**
  + Organize os diretórios do projeto (por exemplo, src/, assets/, docs/).
  + Defina a estrutura básica dos arquivos (separação da interface, lógica de negócios, manipulação de dados, etc.).

**2. Configuração Inicial**

* **2.1 Ambiente de Desenvolvimento:**
  + Configure o ambiente com as bibliotecas necessárias (Tkinter, matplotlib para gráficos, pandas para manipulação de dados, psycopg2 para conectar ao PostgreSQL).
* **2.2 Configuração do Banco de Dados (PostgreSQL):**
  + Instale e configure o PostgreSQL.
  + Crie o banco de dados e as tabelas necessárias para armazenar transações, categorias, investimentos, etc.
  + Configure o PGAdmin para gerenciar o banco de dados.
* **2.3 Interface Gráfica Básica:**
  + Crie a janela principal do aplicativo usando Tkinter.
  + Adicione uma barra de menu com opções básicas (Arquivo, Configurações, Ajuda).

**3. Implementação das Funcionalidades Básicas**

* **3.1 Lançamento de Gastos e Receitas:**
  + Crie formulários para lançar gastos e receitas com campos como descrição, data, valor, categoria.
  + Implemente a funcionalidade de salvar esses dados no banco de dados PostgreSQL.
* **3.2 Visualização de Transações:**
  + Crie uma tabela ou lista que exiba as transações lançadas, carregando os dados do PostgreSQL.
  + Adicione opções para editar ou excluir transações.

**4. Adição de Funcionalidades Avançadas**

* **4.1 Categorias de Despesas e Receitas:**
  + Implemente a criação e edição de categorias para melhor organização das transações.
* **4.2 Relatórios e Gráficos:**
  + Adicione a geração de relatórios mensais/anuais.
  + Implemente gráficos de barras, pizza, ou linhas para visualizar receitas, despesas e saldo ao longo do tempo.
* **4.3 Alertas Financeiros:**
  + Configure alertas para quando os gastos em uma categoria ou período ultrapassarem um limite definido.
* **4.4 Pesquisa e Filtro de Transações:**
  + Implemente um sistema de busca por descrição, data, categoria.
  + Adicione filtros para exibir apenas transações de um tipo específico.

**5. Funcionalidades Específicas**

* **5.1 Registro de Investimentos:**
  + Crie uma seção para registrar e acompanhar investimentos.
  + Implemente cálculos para ganhos e perdas.
* **5.2 Histórico de Pix:**
  + Adicione uma funcionalidade específica para registrar transferências via Pix, com descrição, data, valor e destinatário.
* **5.3 Saldo Atual:**
  + Calcule e exiba o saldo atual com base nas receitas e despesas registradas.
* **5.4 Configurações de Recorrência:**
  + Adicione a possibilidade de registrar receitas e despesas recorrentes automaticamente.

**6. Finalização e Melhoria**

* **6.1 Exportação de Dados:**
  + Implemente a opção de exportar dados para CSV, PDF, etc.
* **6.2 Backup e Restauração:**
  + Adicione funcionalidades para fazer backup e restaurar dados do aplicativo.
* **6.3 Testes e Correções:**
  + Realize testes funcionais e de usabilidade.
  + Corrija possíveis bugs e otimize o desempenho.

**7. Documentação e Lançamento**

* **7.1 Documentação do Código:**
  + Documente o código com comentários e crie uma documentação para o usuário.
* **7.2 Lançamento do Aplicativo:**
  + Prepare a versão final do aplicativo.
  + Disponibilize o projeto em um repositório, como GitHub, ou distribua como um instalador.

**8. Manutenção e Atualizações**

* **8.1 Coleta de Feedback:**
  + Recolha feedback dos usuários para futuras melhorias.
* **8.2 Atualizações e Novas Funcionalidades:**
  + Planeje e implemente novas funcionalidades ou melhorias baseadas no feedback.