Documentação Técnica - Agregador de Dados Excel

1. Visão Geral do Projeto

Este projeto consiste na construção de um agregador de dados em Microsoft Excel, projetado para facilitar a entrada, validação, armazenamento e análise de informações. A solução visa ser robusta, com controle de entradas validadas, menus de navegação intuitivos e funcionalidades extras, mantendo uma interface amigável e prática para o usuário final.

2. Tecnologias Utilizadas

- **Microsoft Excel:** Plataforma principal para a interface do usuário, armazenamento de dados e visualização.
- **Python (openpyxl):** Utilizado para a criação programática da estrutura inicial do arquivo Excel (.xlsx), incluindo a criação de planilhas, definição de cabeçalhos, formatação básica, validações de dados e hiperlinks de navegação. Esta abordagem permite a automação da configuração inicial do arquivo, garantindo consistência e reduzindo erros manuais.
- **VBA (Visual Basic for Applications):** Linguagem de programação integrada ao Excel, utilizada para implementar funcionalidades avançadas de automação, como a transferência de dados da planilha de entrada para o banco de dados, limpeza de formulários e outras interações dinâmicas que não são suportadas diretamente pelo `openpyxl`.

3. Estrutura do Arquivo Excel (`Agregador_de_Dados.xlsx`)

O arquivo é composto pelas seguintes planilhas, cada uma com um propósito específico:

- **Menu Principal:**
  - **Propósito:** Dashboard de navegação central.
  - **Conteúdo:** Título principal, links de navegação para as outras planilhas (`Entrada de Dados`, `Banco de Dados`, `Relatórios`, `Configurações`).
  - **Implementação:** Criada e formatada via `openpyxl`. Hiperlinks configurados para navegação entre as planilhas.
- **Entrada de Dados:**
  - **Propósito:** Formulário para inserção de novos registros.
  - **Conteúdo:** Colunas para `Data`, `Categoria`, `Descrição`, `Valor`, `Status`.
  - **Implementação:**
    - **Layout:** Definido via `openpyxl` com cabeçalhos e formatação básica.
    - **Validação de Dados:** Implementada via `openpyxl` para:
      - `Data`: Tipo `date`, `lessThanOrEqual` a `TODAY()`, com mensagem de erro personalizada.
      - `Categoria`: Tipo `list`, com `formula1` referenciando a lista de categorias na planilha `Configurações`.
      - `Valor`: Tipo `decimal`, `greaterThanOrEqual` a `0`, com mensagem de erro personalizada.
      - `Status`: Tipo `list`, com `formula1` referenciando a lista de status na planilha `Configurações`.
    - **Botões (VBA):** Placeholder para botões "Adicionar Dados" e "Limpar Formulário", que serão associados a macros VBA.
    - **Link de Retorno:** Hiperlink "Voltar ao Menu Principal" para facilitar a navegação.

- **Banco de Dados:**
    - **Propósito:** Armazenamento centralizado e estruturado de todos os dados brutos.
    - **Conteúdo:** Colunas para `ID`, `Data`, `Categoria`, `Descrição`, `Valor`, `Status`, `Data Inserção`.
    - **Implementação:** Criada e formatada via `openpyxl`. Projetada para ser uma tabela simples, sem formatação excessiva, para facilitar a manipulação e o uso em Tabelas Dinâmicas.
    - **Link de Retorno:** Hiperlink "Voltar ao Menu Principal".
- **Relatórios:**
    - **Propósito:** Área para visualização e análise dos dados agregados.
    - **Conteúdo:** Placeholders para "Área para Dashboards e Gráficos" e "Área para Tabela Dinâmica".
    - **Implementação:** Criada e formatada via `openpyxl`. A criação de gráficos e tabelas dinâmicas interativas é um processo manual no Excel ou via VBA, não diretamente suportado pelo `openpyxl`.
    - **Link de Retorno:** Hiperlink "Voltar ao Menu Principal".
- **Configurações:**
    - **Propósito:** Gerenciamento de listas de validação e parâmetros globais.
    - **Conteúdo:** Listas de `Categorias` e `Status` pré-preenchidas com exemplos.
    - **Implementação:** Criada e formatada via `openpyxl`. As listas são referenciadas pelas validações de dados na planilha `Entrada de Dados`.
    - **Link de Retorno:** Hiperlink "Voltar ao Menu Principal".

4. Código VBA (`vba_code.md`)

O arquivo `vba_code.md` contém o código VBA a ser inserido no Editor VBA do Excel. As macros principais são:

- `AdicionarDados()`:

    - **Função:** Transfere os dados da linha de entrada da planilha `Entrada de Dados` para a próxima linha vazia na planilha `Banco de Dados`.
    - **Lógica:**
        - Identifica a próxima `ID` disponível no `Banco de Dados`.
        - Valida se os campos obrigatórios (`Data`, `Categoria`, `Valor`) na `Entrada de Dados` estão preenchidos.
        - Copia os valores das células `A5:E5` da `Entrada de Dados` para as colunas correspondentes no `Banco de Dados`.
        - Adiciona a `Data Inserção` automaticamente.
        - Formata as colunas de data no `Banco de Dados`.
        - Exibe uma mensagem de sucesso.
        - Chama a macro `LimparFormulario`.

- `LimparFormulario()`:
    - **Função:** Limpa o conteúdo das células de entrada na planilha `Entrada de Dados`.
    - **Lógica:** Limpa o intervalo `A5:E5`.

5. Formatação Condicional (Manual)

A formatação condicional para realçar campos obrigatórios ou dados inválidos deve ser aplicada manualmente no Excel, conforme as instruções detalhadas no `vba_code.md`. Exemplos de regras incluem:

- Realçar células vazias em campos obrigatórios.
- Realçar valores negativos (se o campo `Valor` for sempre positivo).
- Realçar datas futuras (se o campo `Data` não puder ser futuro).

6. Considerações de Desenvolvimento

- **Automação vs. Manual:** A criação da estrutura base é automatizada via Python, enquanto as funcionalidades interativas avançadas (VBA, Tabelas Dinâmicas,

Gráficos) requerem configuração manual ou via VBA.

- **Manutenção:** Alterações na estrutura das planilhas (adição/remoção de colunas) podem exigir ajustes no script Python e no código VBA.
- **Segurança:** Ao compartilhar o arquivo, instrua os usuários a habilitar as macros para o funcionamento completo da ferramenta.

7. Próximos Passos

- **Testes:** Realizar testes abrangentes de todas as funcionalidades.
- **Otimização:** Avaliar o desempenho e otimizar o código VBA, se necessário.
- **Expansão:** Considerar a adição de novas funcionalidades, como exportação de dados, relatórios mais complexos ou integração com outras fontes de dados.