

Documentação do Projeto de Arquivamento e Comparação de Arquivos

Este projeto consiste em um conjunto de ferramentas para organização, arquivamento e comparação de arquivos, com foco especial em mensagens de email e documentos.

Ferramentas Disponíveis

1. Arquivo Email GUI (`arquivo_email_gui.py`)

Ferramenta com interface gráfica para organização de emails e outros arquivos em estrutura cronológica. [Ver documentação detalhada.](#)

2. Arquivo Email (`arquivo_email.py`)

Objetivos e Funcionalidades

- Versão de linha de comando para organização de arquivos `.eml` e outros tipos
- Organiza arquivos em estrutura cronológica (Ano/Ano-Mês)
- Extrai datas de cabeçalhos de emails para arquivos `.eml`
- Utiliza datas de modificação para outros tipos de arquivo

Diferenças da versão GUI

- Não possui interface gráfica
- Configurado para uso via linha de comando ou importação em outros scripts
- Mantém a mesma lógica de processamento de arquivos

3. Arquivo Raiz (`arquivo_raiz.py`)

Objetivos e Funcionalidades

- Processa arquivos na pasta raiz e suas subpastas
- Move arquivos de subpastas para a raiz, sanitizando e renomeando conforme necessário
- Opção para remover pastas vazias após a movimentação dos arquivos
- Mantém contadores de arquivos processados, renomeados, movidos e erros

Características Específicas

- Método `process_files_in_root()` para processar arquivos em toda a estrutura
- Método `remove_empty_folders()` para limpar pastas vazias após processamento
- Suporte a pastas excluídas que não serão processadas

4. Arquivo Subpastas (`arquivo_subpastas.py`)

Objetivos e Funcionalidades

- Versão especializada para processamento de arquivos em subpastas
- Mantém a estrutura de pastas original, mas organiza os arquivos dentro delas
- Implementa lógica avançada para tratamento de datas em emails
- Contabiliza pastas criadas durante o processamento

Características Específicas

- Método `move_file_to_archive()` aprimorado para decisão entre mover, renomear ou ignorar
- Tratamento mais robusto de datas em emails com múltiplos formatos
- Contadores para pastas criadas durante o processamento

5. Comparador de Pastas (`pastas_diff.py`)

Objetivos e Funcionalidades

- Compara o conteúdo de duas pastas e gera relatório detalhado das diferenças
- Identifica arquivos presentes em apenas uma das pastas
- Detecta arquivos com mesmo nome mas conteúdo diferente
- Salva relatório de comparação em formato texto

Características Específicas

- Interface para seleção das duas pastas a serem comparadas
- Método `compare_folders()` para análise detalhada das diferenças
- Sistema de logging para registro de erros durante a comparação
- Método `save_report()` para salvar o relatório em arquivo de texto

Detalhes Técnicos Comuns

Sistema de Logging

- Todos os scripts utilizam o módulo `logging` para registro de erros
- Logs são salvos em pasta "ERROS" dentro da pasta principal de processamento
- Arquivos de log nomeados com timestamp para identificação única

Tratamento de Nomes de Arquivo

- Sanitização para remover caracteres inválidos e prefixos desnecessários
- Truncamento para respeitar limites de tamanho de caminho
- Resolução de conflitos para evitar sobrescrita de arquivos

Processamento de Datas

- Extração de datas de cabeçalhos de email para arquivos `.eml`
- Uso de data de modificação para outros tipos de arquivo
- Múltiplas estratégias de fallback para garantir processamento mesmo com formatos não padrão

Requisitos Técnicos

- Python 3.6 ou superior

- Bibliotecas padrão: `os`, `shutil`, `email`, `logging`, `re`, `datetime`, `tkinter`
- Não requer instalação de pacotes externos

Dicas de Uso

- Escolha a ferramenta adequada para sua necessidade específica:
 - `arquivo_email_gui.py` para organização simples com interface gráfica
 - `arquivo_raiz.py` para consolidar arquivos de subpastas na raiz
 - `arquivo_subpastas.py` para manter a estrutura de pastas original
 - `pastas_diff.py` para comparar o conteúdo de duas pastas
- Faça backup dos arquivos antes de executar qualquer ferramenta de movimentação
- Verifique os logs após o processamento para identificar possíveis problemas