

# Arquivo Email GUI - Documentação Completa

---

## Objetivos e Funcionalidades

O `arquivo_email_gui.py` é uma ferramenta especializada para organização automática de arquivos, com foco principal em mensagens de email (`.eml`). O programa:

- Organiza automaticamente arquivos em uma estrutura cronológica de pastas (Ano/Ano-Mês)
- Extrai datas de cabeçalhos de emails para arquivos `.eml`
- Utiliza datas de modificação para outros tipos de arquivo
- Sanitiza nomes de arquivos removendo caracteres inválidos e prefixos desnecessários
- Resolve conflitos de nomes automaticamente
- Fornece feedback detalhado sobre o processo de arquivamento

## Modo de Usar

### 1. Execução do Programa:

- Execute o script `arquivo_email_gui.py` com Python
- Uma janela de diálogo será exibida solicitando a seleção da pasta a ser processada

### 2. Seleção da Pasta:

- Selecione a pasta que contém os arquivos a serem organizados
- O programa usará esta mesma pasta como raiz para criar a estrutura de arquivamento

### 3. Processamento:

- O programa processará automaticamente todos os arquivos na pasta selecionada
- Os arquivos serão movidos para subpastas no formato `Ano/Ano-Mês` baseado em suas datas
- Uma janela de resumo será exibida ao final do processamento

### 4. Verificação de Resultados:

- Após a conclusão, verifique a estrutura de pastas criada
- Se ocorreram erros, consulte os logs na pasta "ERROS" criada na raiz

## Detalhes Técnicos

### Estrutura de Arquivamento

- **Formato de Pastas:** `Ano/Ano-Mês` (exemplo: `2023/2023-05`)
- **Localização:** As pastas são criadas dentro da pasta selecionada pelo usuário

### Processamento de Datas

#### Para arquivos `.eml`:

1. Tenta extrair o cabeçalho `Date` do email

2. Utiliza múltiplos formatos de data para análise:
  - Formato RFC 5322 padrão com offset de timezone
  - Formatos alternativos com e sem dia da semana
3. Em caso de falha na extração, utiliza a data atual como fallback

### Para outros arquivos:

- Utiliza a data de modificação do arquivo obtida via `os.path.getmtime()`

## Tratamento de Nomes de Arquivo

### 1. Sanitização:

- Remove o prefixo "msg " (comum em emails exportados)
- Substitui caracteres inválidos (< > : " / \ | ? \*) por underscores
- Remove caracteres de controle (ASCII 0-31)
- Normaliza números no início do nome para remover zeros à esquerda

### 2. Truncamento:

- Limita o tamanho do nome para evitar exceder o limite de 255 caracteres
- Preserva a extensão original do arquivo

### 3. Resolução de Conflitos:

- Adiciona contadores incrementais (\_1, \_2, etc.) para nomes duplicados
- Em casos extremos, utiliza timestamp com precisão de microssegundos
- Implementa verificação recursiva após cada tentativa de resolução

## Sistema de Logging

- **Nível:** Configurado para registrar apenas erros (nível `ERROR`)
- **Localização:** Cria pasta "ERROS" na raiz selecionada
- **Formato:** Arquivos de log com timestamp único (`archive_failures_YYYYMMDDHHMMSS.log`)
- **Conteúdo:** Detalhes específicos sobre cada erro ocorrido durante o processamento

## Interface Gráfica

- **Tecnologia:** Implementada com `tkinter`
- **Componentes:**
  - Diálogo de seleção de pasta
  - Mensagens informativas no início e fim do processo
  - Janela de resumo com auto-fechamento ao final do processamento
  - Contadores de arquivos processados e erros encontrados

## Casos de Uso

### Organização de Backups de Email

Ideal para arquivos `.eml` exportados de clientes de email, mantendo a cronologia original baseada no cabeçalho "Date".

## Arquivamento de Documentos

Organiza documentos baseados em sua data de modificação, facilitando a localização por período.

## Consolidação de Backups

Unifica múltiplos backups em uma estrutura consistente, evitando duplicação através da resolução de conflitos.

## Migração entre Sistemas de Email

Prepara emails exportados para importação em novo sistema, mantendo a organização cronológica.

## Limitações e Considerações

- **Processamento de Subpastas:** O programa processa apenas arquivos no nível raiz da pasta selecionada
- **Movimentação vs. Cópia:** Utiliza `shutil.move()`, que remove o arquivo original da pasta de origem
- **Limites de Sistema de Arquivos:** Restrito ao limite de aproximadamente 255 caracteres para caminhos completos
- **Dependência de Formatos de Data:** A precisão da organização depende dos formatos de data nos cabeçalhos dos emails

## 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

- Faça um backup dos arquivos antes de executar o programa pela primeira vez
- Para arquivos `.eml`, verifique se os cabeçalhos de data estão em formatos padrão
- Evite nomes de arquivo extremamente longos para prevenir truncamento excessivo
- Verifique os logs de erro após o processamento para identificar possíveis problemas