

# Renomeador de Arquivos EML (`renomear_eml.py`)

---

## 1. Objetivo

Este script Python, com interface gráfica para seleção de pasta, tem como objetivo principal renomear arquivos de e-mail no formato `.eml` localizados diretamente dentro de uma pasta selecionada. O novo nome é construído a partir de informações extraídas dos cabeçalhos do próprio e-mail (Data, Assunto, Remetente), facilitando a identificação e a ordenação cronológica dos arquivos pelo nome. Arquivos duplicados (que teriam o mesmo nome após a renomeação) ou arquivos que apresentam erros durante a leitura/processamento são movidos para subpastas específicas.

## 2. Funcionalidades Principais

- **Interface Gráfica para Seleção:** Utiliza Tkinter para permitir ao usuário selecionar facilmente a pasta que contém os arquivos `.eml` a serem renomeados.
- **Processamento de Arquivos `.eml`:** Analisa apenas arquivos com a extensão `.eml` (case-insensitive) diretamente na pasta selecionada.
- **Extração e Decodificação de Cabeçalhos:**
  - Lê os cabeçalhos `Date`, `Subject` e `From` do e-mail.
  - Utiliza a biblioteca `email` (com `policy.default`) para uma leitura robusta.
  - Tenta ler o arquivo como binário primeiro; se falhar, tenta como texto com encodings UTF-8 e Latin-1.
  - Decodifica corretamente cabeçalhos que podem estar codificados (ex: `=?utf-8?B?...?=`).
- **Análise e Formatação da Data:**
  - Analisa o cabeçalho `Date` usando `email.utils.parsedate_to_datetime`.
  - Converte datas com fuso horário para o fuso horário local do sistema (ou UTC como fallback se a conversão falhar).
  - Se a data do cabeçalho não puder ser analisada, utiliza a data de modificação do arquivo como fallback.
  - Se a data de modificação também falhar, usa a data/hora atual e registra um erro.
  - Formata a data resultante como `YYYY MM DD HHMMSS`.
- **Sanitização de Nomes:**
  - Limpa o Assunto e o Remetente extraídos para uso seguro em nomes de arquivo:
    - Substitui caracteres inválidos (`<`, `>`, `:`, `"`, `/`, `\`, `|`, `?`, `*`, caracteres de controle) por underscores (`_`).
    - Substitui múltiplos espaços ou underscores por um único underscore.
    - Remove underscores no início ou fim.
    - Trunca partes excessivamente longas (Assunto/Remetente) para evitar nomes de arquivo gigantescos (limite configurável `MAX_FILENAME_PART_LEN`).
    - Garante que as partes sanitizadas não fiquem vazias (usa "Desconhecido" ou "Invalido").
- **Formato do Novo Nome:** Constrói o novo nome no formato: `YYYY MM DD HHMMSS - AssuntoSanitizado - RemetenteSanitizado.eml`
- **Tratamento de Conflitos (Duplicatas):**
  - Se um arquivo com o novo nome proposto já existir na pasta principal:
    - Cria uma subpasta chamada `Duplicatas` (se não existir).

- Move o arquivo *original* que está sendo processado para a pasta **Duplicatas**.
- Se um arquivo com o mesmo nome já existir *dentro* da pasta **Duplicatas**, adiciona um sufixo numérico (**\_1**, **\_2**, ...) ao nome do arquivo movido para **Duplicatas**.
- **Tratamento de Erros de Leitura/Processamento:**
  - Se ocorrer um erro irrecuperável ao tentar ler ou processar os cabeçalhos de um arquivo **.eml**:
    - Cria uma subpasta chamada **Problemas** (se não existir).
    - Tenta obter a data de criação/modificação do arquivo como fallback para o nome.
    - Constrói um nome no formato **YYYY MM DD HHMMSS - ERRO\_LEITURA - NomeOriginalSanitizado.eml**.
    - Move o arquivo *original* problemático para a pasta **Problemas**.
    - Se um arquivo com o mesmo nome já existir *dentro* da pasta **Problemas**, adiciona um sufixo numérico (**\_1**, **\_2**, ...).
- **Feedback ao Usuário:**
  - Imprime o progresso e avisos/erros detalhados no console durante a execução.
  - Ao final, exibe uma janela pop-up (messagebox) com um resumo:
    - Número de arquivos renomeados com sucesso.
    - Número de arquivos movidos para **Duplicatas**.
    - Número de arquivos movidos para **Problemas**.
    - Número de arquivos/pastas ignorados.
    - Número total de erros encontrados.

### 3. Modo de Usar

1. **Execute o Script:** Certifique-se de ter o Python 3 instalado. Execute o script **renomear\_eml.py** (por exemplo, clicando duas vezes nele ou rodando **python renomear\_eml.py** no terminal).
2. **Selecione a Pasta:** Uma janela de diálogo do sistema operacional será aberta. Navegue até a pasta que contém os arquivos **.eml** que você deseja renomear e clique em "Selecionar pasta" (ou o botão equivalente).
3. **Aguarde o Processamento:** O script começará a analisar cada arquivo **.eml** na pasta selecionada. O progresso, incluindo quais arquivos estão sendo processados, renomeados, movidos ou apresentando erros, será exibido no console/terminal onde o script foi iniciado.
4. **Verifique o Resumo:** Ao final do processo, uma janela pop-up aparecerá com um resumo das operações realizadas (quantos arquivos foram renomeados, movidos para **Duplicatas**, movidos para **Problemas**, ignorados e quantos erros ocorreram). Clique em "OK" para fechar a janela.
5. **Consulte os Resultados:**
  - Verifique os arquivos na pasta original; muitos devem ter sido renomeados para o formato **YYYY MM DD HHMMSS - Assunto - Remetente.eml**.
  - Se a subpasta **Duplicatas** foi criada, verifique seu conteúdo. Ela conterá arquivos que teriam o mesmo nome de outros após a renomeação.
  - Se a subpasta **Problemas** foi criada, verifique seu conteúdo. Ela conterá arquivos que não puderam ser lidos ou processados corretamente. O nome desses arquivos incluirá "ERRO\_LEITURA" e uma data de fallback.
  - Consulte o console/terminal para obter detalhes sobre avisos ou erros específicos que ocorreram durante o processo.

### 4. Especificações Técnicas

- **Linguagem:** Python 3.x
- **Interface Gráfica (GUI):** Tkinter (módulo padrão) para seleção de pasta (`filedialog`) e mensagem de resumo (`messagebox`).
- **Dependências:** Utiliza apenas módulos padrão do Python: `os`, `tkinter`, `email` (incluindo `policy`, `header`, `utils`), `datetime`, `re`, `shutil`, `sys`.
- **Escopo:** Processa apenas arquivos `.eml` (case-insensitive) localizados **diretamente** na pasta selecionada. Não processa subpastas.
- **Codificação de Leitura:** Tenta ler `.eml` como binário (com `policy.default`), depois como texto UTF-8 (ignorando erros), e como texto Latin-1 (ignorando erros).
- **Formato do Nome:** `YYYY MM DD HHMMSS - AssuntoSanitizado - RemetenteSanitizado.eml`
- **Sanitização:** Remove/substitui caracteres inválidos (`<>:"/\\|?*\\x00-\\x1f`), normaliza espaços/underscores, trunca partes longas (`MAX_FILENAME_PART_LEN = 60`).
- **Subpastas Criadas:** `Duplicatas` (para conflitos de nome), `Problemas` (para erros de leitura/processamento).
- **Saída:** Arquivos renomeados na pasta original, arquivos movidos para `Duplicatas` ou `Problemas`, mensagens de progresso/erro no console, janela de resumo final.
- **Logging:** Não utiliza a biblioteca `logging`. Erros e avisos são impressos diretamente no console (`stdout/stderr`).