

# Comparador de Conteúdo de Pastas

## (pastas\_diff.exe)

---

## 1. Objetivo

Este script Python tem como objetivo comparar o conteúdo (lista de arquivos) de duas pastas selecionadas pelo usuário, incluindo todas as suas subpastas. Ele identifica quais arquivos existem apenas na primeira pasta, quais existem apenas na segunda pasta e quantos arquivos são comuns a ambas, gerando um relatório detalhado das diferenças.

## 2. Funcionalidades Principais

- **Interface Gráfica para Seleção:** Utiliza Tkinter para permitir ao usuário selecionar facilmente as duas pastas que deseja comparar através de diálogos padrão do sistema.
- **Comparação Recursiva:** Analisa a pasta selecionada e **todas as suas subpastas** para listar os arquivos presentes.
- **Comparação Baseada em Caminho Relativo:** A comparação entre as pastas é feita usando os caminhos relativos dos arquivos a partir da raiz de cada pasta selecionada, garantindo uma comparação justa mesmo que as pastas raiz tenham nomes diferentes.
- **Ignora Arquivos Específicos:** Arquivos `.ffs_db` (usados pelo FreeFileSync) são ignorados durante a listagem e comparação (case-insensitive).
- **Geração de Relatório Detalhado:** Cria um arquivo de texto (`.txt`) contendo:
  - As pastas comparadas.
  - A data e hora da comparação.
  - O número total de arquivos encontrados em cada pasta (excluindo os ignorados).
  - Uma lista dos arquivos encontrados **exclusivamente** na primeira pasta.
  - Uma lista dos arquivos encontrados **exclusivamente** na segunda pasta.
  - Um resumo com a contagem de arquivos comuns e exclusivos de cada pasta.
- **Salvamento do Relatório:** O arquivo de relatório (`comparacao_pastas_*.txt`) é salvo automaticamente dentro da **primeira** pasta selecionada pelo usuário.
- **Registro de Erros:** Cria uma subpasta chamada `ERROS` dentro da **primeira** pasta selecionada. Erros que ocorram durante a listagem de arquivos ou o salvamento do relatório são registrados em um arquivo de log (`comparison_failures_*.log`) dentro desta pasta.
- **Feedback ao Usuário:**
  - Exibe mensagens informativas se o usuário cancelar a seleção de alguma pasta.
  - Ao final, exibe uma mensagem de confirmação indicando que a comparação foi concluída e onde o relatório foi salvo.
  - Exibe uma mensagem de erro se não for possível salvar o relatório, instruindo a verificar os logs.

## 3. Modo de Usar

1. **Execute o Script:** Certifique-se de ter o Python 3 instalado. Execute o script `pastas_diff.py` (por exemplo, clicando duas vezes nele ou rodando `python pastas_diff.py` no terminal).
2. **Execute o Programa:** Execute o arquivo `pastas_diff.exe` (ou o nome que o executável recebeu).

3. **Selecione a Primeira Pasta:** Uma janela de diálogo do sistema operacional será aberta com o título "Selecione a primeira pasta". Navegue até a primeira pasta que você deseja comparar e clique em "Selecionar pasta" (ou o botão equivalente).
4. **Selecione a Segunda Pasta:** Outra janela de diálogo aparecerá com o título "Selecione a segunda pasta". Navegue até a segunda pasta que você deseja comparar e clique em "Selecionar pasta".
5. **Aguarde a Comparação:** O script listará os arquivos em ambas as estruturas de pastas e realizará a comparação. O tempo de execução dependerá do número de arquivos em cada pasta. Mensagens de progresso podem aparecer no console onde o script foi executado.
6. **Verifique o Resultado:** Uma janela de mensagem aparecerá informando que a comparação foi concluída e indicando o nome e a localização do arquivo de relatório (que será salvo dentro da **primeira** pasta que você selecionou).
7. **Consulte o Relatório:** Navegue até a **primeira** pasta que você selecionou e abra o arquivo `.txt` gerado (ex: `comparacao_pastas_20240521153000.txt`) para ver a lista de arquivos exclusivos de cada pasta e o resumo.
8. **Consulte os Logs (se necessário):** Se a mensagem final indicou um erro ao salvar o relatório, ou se você suspeitar de problemas durante a listagem, navegue até a **primeira** pasta selecionada, abra a subpasta `ERROS` e procure pelo arquivo `.log` mais recente (ex: `comparison_failures_20240521153000.log`).

## 4. Especificações Técnicas

- **Tipo:** Executável para Windows (gerado a partir de Python 3.x).
- **Interface Gráfica (GUI):** Tkinter (módulo padrão) para seleção de pastas (`filedialog`) e exibição de mensagens (`messagebox`).
- **Dependências:** Nenhuma instalação adicional é necessária para executar o arquivo `.exe`, pois todas as dependências (como Python e bibliotecas necessárias) estão empacotadas nele.
- **Escopo da Comparação:** Recursiva (inclui subpastas).
- **Método de Comparação:** Baseado na presença/ausência de arquivos com o mesmo caminho relativo em ambas as pastas. **Não compara o conteúdo** dos arquivos.
- **Itens Ignorados:** Arquivos `.ffs_db` (case-insensitive).
- **Saída:**
  - Arquivo de relatório (`comparacao_pastas_*.txt`) salvo na **primeira** pasta selecionada, codificado em UTF-8.
  - Arquivo de log (`comparison_failures_*.log`) salvo na subpasta `ERROS` da **primeira** pasta selecionada, codificado em UTF-8 (registra apenas nível `ERROR`).
- **Logging:** Configurado para registrar apenas erros (`logging.ERROR`) que ocorram durante a listagem de arquivos ou salvamento do relatório.