



Universidade Federal do Ceará Campus Quixadá

QXD0099 - Desenvolvimento de Software para Persistência

Lista 2 - Processamento de Arquivos e Armazenamento Compactado

Objetivo: Criar um programa que leia e processe arquivos de texto, escreva os resultados em um novo arquivo e compacte o arquivo de saída em um ZIP.

Descrição: Você deverá criar um programa Python que realiza as seguintes etapas:

1. **Leitura de Arquivos:** Leia vários arquivos de texto em um diretório, processando o conteúdo de cada um.
2. **Processamento dos Dados:** Remova quaisquer linhas em branco e faça uma contagem do número de palavras e caracteres em cada arquivo.
3. **Escrita em um Arquivo Consolidado:** Grave os resultados em um arquivo consolidado, listando o nome de cada arquivo e as estatísticas correspondentes (número de palavras e caracteres).
4. **Compactação em Arquivo ZIP:** Após processar e consolidar os dados, compacte o arquivo consolidado em um arquivo ZIP.

Instruções:

1. Crie um diretório chamado `textos` e adicione 3 arquivos de texto com conteúdo variado para o teste (por exemplo: `arquivo1.txt`, `arquivo2.txt`, `arquivo3.txt`).
2. Escreva o código Python que segue o fluxo descrito.

Explicação do Código:

- **Leitura dos Arquivos:** O código lê todos os arquivos `.txt` da pasta `textos`.
- **Processamento:** Remover linhas em branco, calcular a quantidade de palavras e caracteres.
- **Escrita Consolidada:** Escreve o nome do arquivo e as estatísticas no arquivo `consolidado.txt`.
- **Compactação ZIP:** Compacta o arquivo `consolidado.txt` no arquivo `saida.zip`.

Bônus (Opcional):

- Ajuste o código para incluir uma opção para listar todos os arquivos do ZIP.

- Adicione tratamento de exceções para arquivos não encontrados ou diretórios vazios.
-

Reflexão:

- Como o uso de compactação ajuda no armazenamento de arquivos processados?
- O que acontece se tentar compactar um arquivo inexistente?