

# Lista de exercícios 3: Persistência de Dados

Gabriel Rodrigues Caldas de Aquino

28/05/2025

## 1 Leitura Básica de Arquivo

**Descrição:** Crie um programa em Python que abra um arquivo chamado "dados.txt" (modo leitura), imprima todo o seu conteúdo na tela e depois feche o arquivo.

## 2 Contador de Linhas

**Descrição:** Escreva um programa que conte quantas linhas existem em um arquivo chamado "nomes.txt". O programa deve abrir o arquivo, contar as linhas, imprimir o total e fechar o arquivo.

## 3 Gravação Simples

**Descrição:** Crie um programa que grave três nomes (um em cada linha) em um arquivo chamado "alunos.txt". Use o modo de escrita ('w') e não esqueça de fechar o arquivo.

## 4 Adição de Conteúdo

**Descrição:** Modifique o exercício anterior para adicionar mais dois nomes ao final do arquivo "alunos.txt" (em vez de sobrescrever). Use o modo de adição ('a').

## 5 Tratamento de Erros

**Descrição:** Escreva um programa que tente abrir um arquivo "notas.txt" para leitura. Se o arquivo não existir, o programa deve imprimir uma mensagem amigável (como "Arquivo não encontrado") em vez de mostrar o erro do Python. Use try-except.

## 6 Manipulação de Arquivo CSV

**Descrição:** Crie um programa que leia um arquivo *dados.csv* (no formato Nome,Idade,Cidade) e gere um novo arquivo *maiores-de-idade.txt* contendo apenas as pessoas com idade maior ou igual a 18 anos.

- Exemplo de entrada (*dados.csv*):

```
João,17,Caxias
Maria,19,Maricá
Pedro,20,Niterói
Ana,16,Belém
```

- Saída esperada (*maiores-de-idade.txt*):

```
Maria,19,Maricá
Pedro,20,Niterói
```

## 7 Busca em Arquivo

**Descrição:** Crie um programa que permita ao usuário buscar um nome em um arquivo *contatos.txt*, onde cada linha tem o formato *Nome:Telefone*. Se o nome existir, exiba o telefone; caso contrário, informe que o contato não foi encontrado

- Exemplo de entrada (*contatos.txt*):

```
João:21987654321
Maria:21912345678
Pedro:21955554444
```

- Exemplo de execução:

```
Digite o nome a buscar: Maria
Telefone: 21912345678
```

## 8 Log de Execução

**Descrição:** Crie um programa que registre em um arquivo *log.txt* todas as vezes que ele for executado, adicionando uma nova linha com a data e hora.

- Formato:

```
Programa executado em: 2025-05-29 14:30:00
Programa executado em: 2025-05-30 09:15:00
```

**Dica:** Use *import datetime* e *datetime.now()* para obter a data/hora atual

## 9 Soma de Dados Numéricos

Crie um programa que leia um arquivo *valores.txt* (formato: *NodeID,Valor*) e calcule:

- A soma total dos valores.
- O maior valor da lista.
- Grave os resultados em um novo arquivo *estatisticas.txt*.
- Exemplo de entrada (*valores.txt*):

Node1,15.5

Node2,20.0

Node3,10.2

- Saída esperada (*estatisticas.txt*):

Soma total: 45.7

Maior valor: 20.0

## 10 Operações com Matrizes a partir de Arquivos

**Objetivo:** Ler duas matrizes de arquivos *.txt*, realizar operações básicas (soma, subtração) e salvar o resultado em um novo arquivo.

- Utilize dois arquivos (*matrizA.txt* e *matrizB.txt*) no formato onde cada linha representa uma linha da matriz, com valores separados por vírgula:

1 2 3

4 5 6

7 8 9

Operações Requeridas:

- Ler as matrizes dos arquivos
- Verificar se pode ser realizada a operação
- Realizar soma ou subtração
- Salvar o resultado em *resultado.txt*