



Universidade Federal do Rio Grande  
Centro de Ciências Computacionais



# Algoritmos e Estruturas de Dados I

## Arquivos

Profs. Drs. Cleo Billa, Rafael Penna e Thiago da Silveira

1º Semestre de 2020

# Roteiro

- Motivação
- Operações com Arquivos
- Arquivos de Texto
- Leitura de Arquivos de Texto
- Escrita em Arquivos de Texto

# Motivação

- Armazenamento de dados;
- Processamento posterior...

```
numNotas = 4
numAlunos = 40

for aluno in range(numAlunos):
    nome = input("Informe o nome do aluno: ")
    soma = 0
    for nota in range(numNotas):
        valor = float(input('Digite uma nota: '))
        soma += valor
    print('O aluno', nome, 'teve média ', soma/numNotas)
```

# Motivação

- Armazenamento de dados;
- Processamento posterior...

```
numNotas = 4
numAlunos = 40

nomes = []
medias = []

for aluno in range(numAlunos):
    nome = input("Informe o nome do aluno: ")
    nomes += [nome]
    soma = 0
    for nota in range(numNotas):
        valor = float(input('Digite uma nota: '))
        soma += valor
    medias += [soma/numNotas]

for i in range(numAlunos):
    print('O aluno', nomes[i], 'teve média ', medias[i])
```

# Motivação

- Armazenamento de dados;
- Processamento posterior...

Mas e se eu fechar meu programa?

```
numNotas = 4
numAlunos = 40

nomes = []
medias = []

for aluno in range(numAlunos):
    nome = input("Informe o nome do aluno: ")
    nomes += [nome]
    soma = 0
    for nota in range(numNotas):
        valor = float(input('Digite uma nota: '))
        soma += valor
    medias += [soma/numNotas]

for i in range(numAlunos):
    print('O aluno', nomes[i], 'teve média ', medias[i])
```

# Motivação

- Arquivos permitem o armazenamento **permanente** (**persistente**) de dados;
- É possível escrever programas que manipulam arquivos;
- Um programa pode manipular dados gerados por ele mesmo - ou não;
- Arquivos são armazenados em dispositivos de memória **secundária**;
- A memória secundária tem mais espaço de armazenamento que a principal.

# Arquivos e Operações

- Arquivos são conjuntos de bytes armazenados em dispositivo de memória secundária;
- Arquivos são manipulados, em um programa, como se fossem **variáveis** que implementam uma estrutura de dados;
- Arquivos podem ser classificados como dos tipos binário e **texto**.

# Arquivos e Operações

- Arquivos são conjuntos de bytes armazenados em dispositivo de memória secundária;
- Arquivos são manipulados, em um programa, como se fossem **variáveis** que implementam uma estrutura de dados;
- Arquivos podem ser classificados como dos tipos binário e **texto**.



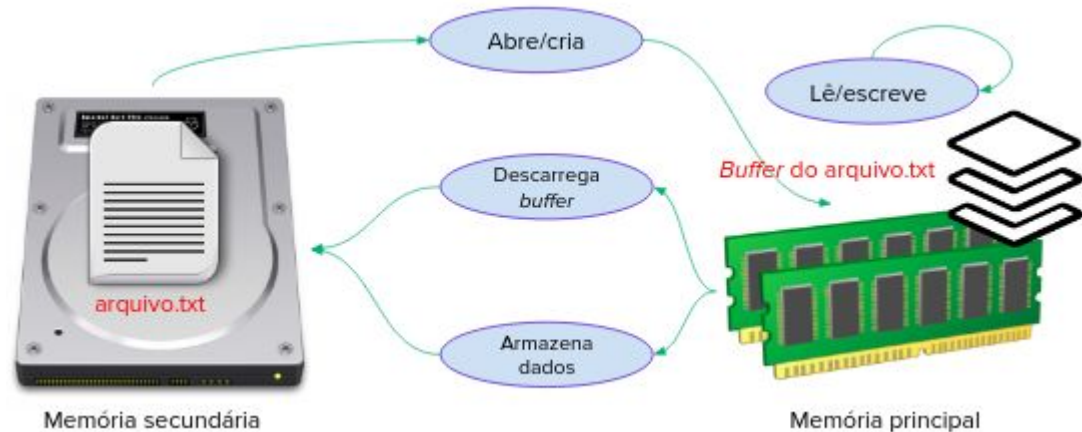
Nosso foco na disciplina!



# Arquivos e Operações

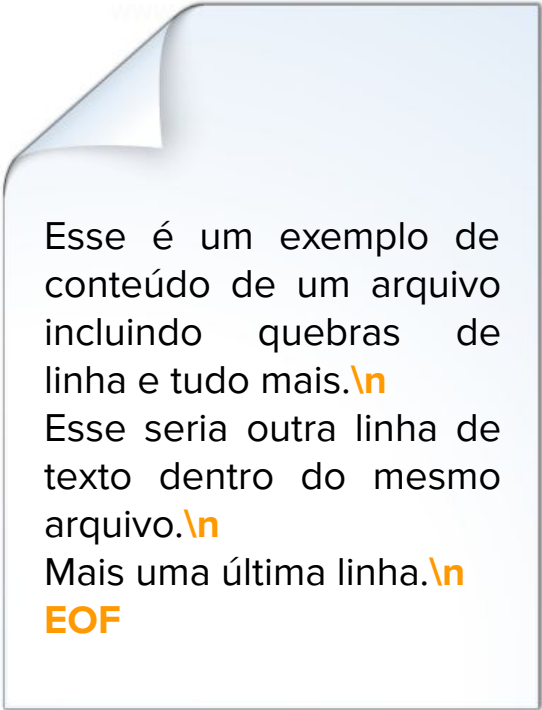
- A manipulação de arquivos acontece na **memória principal**;

- Abertura
- Leitura
- Escrita
- Fechamento



# Arquivos de Texto

- Características e caracteres de controle
  - Elemento/posição corrente
    - Linha e/ou caractere
  - Marcador de final de linha
    - Dependente de sistema operacional (“\n”, “\r”, “\r\n”)
    - Convencionado “\n” no Python
  - Marcador de final de arquivo
    - Caractere EOF



Esse é um exemplo de conteúdo de um arquivo incluindo quebras de linha e tudo mais.\n  
Esse seria outra linha de texto dentro do mesmo arquivo.\n  
Mais uma última linha.\n  
EOF

# Abertura de Arquivos de Texto

- Para que se possa manipular um arquivo, precisamos utilizar uma primeira operação de **abertura de arquivo**.
- A função `var = open(filename[, mode])` retorna um "objeto" do tipo arquivo que permite manipular o arquivo desejado.
  - O parâmetro `filename` indica o caminho para o arquivo:
    - `open("Arquivo.txt")`
    - `open("C:\\Users\\Aluno\\Desktop\\Arquivo.txt")`

Depois de um `var = open(filename)` deve haver um `var.close()`

# Abertura de Arquivos de Texto

- A função `open(filename[, mode])` retorna um "objeto" do tipo arquivo que permite manipular o arquivo desejado.
  - O parâmetro `mode` indica o modo de abertura para o arquivo:
    - “r” para **leitura (valor padrão)**
      - Se o arquivo não existir, ocorrerá um erro de leitura
    - “w” para **escrita**
      - Se o arquivo já existir, este irá ser sobrescrito
    - “a” para **escrita no modo *append***
      - Se o arquivo já existir, este ser complementado ao final

# Abertura de Arquivos de Texto

- A função `open(filename[, mode])` retorna um "objeto" do tipo arquivo que permite manipular o arquivo desejado.
  - O parâmetro `mode` indica o modo de abertura para o arquivo:
    - `"r+"` para **leitura e escrita**
      - Se o arquivo não existir, não ocorrerá um erro de leitura
    - `"w+"` para **leitura e escrita**
      - Se o arquivo já existir, este irá ser sobrescrito
    - `"a+"` para **leitura e escrita no modo *append***
      - Se o arquivo já existir, este ser complementado ao final

# Operações com Arquivos de Texto

- Para manipulação de arquivos no modo texto, usamos:
  - `read([size])`: lê `size` dados de uma só vez. Retorna `""` ao fim do arquivo;
  - `readline()`: lê uma linha do arquivo (até o próximo `"\n"`). Retorna `""` ao fim do arquivo;
  - `readlines()`: lê todas as linhas do arquivo e retorna uma lista com o conteúdo.

# Operações com Arquivos de Texto

- Para manipulação de arquivos no modo texto, usamos:
  - `write(string)`: escreve `string` no arquivo e retorna o número de caracteres escritos;
  - `tell()`: indica qual a posição corrente da leitura/escrita;
  - `seek(offset[, whence])`: altera a posição corrente da leitura/escrita adicionando um deslocamento (`offset`) relativo a alguma referência (`whence`)
    - `whence = 0` indica o início do arquivo como a referência
    - `whence = 1` indica a posição corrente do arquivo como a referência
    - `whence = 2` indica o fim do arquivo como a referência

# Motivação de novo...

- Armazenamento de dados;
- Processamento posterior...

Mas e se eu fechar meu programa?

```
numNotas = 4
numAlunos = 40

nomes = []
medias = []

for aluno in range(numAlunos):
    nome = input("Informe o nome do aluno: ")
    nomes += [nome]
    soma = 0
    for nota in range(numNotas):
        valor = float(input('Digite uma nota: '))
        soma += valor
    medias += [soma/numNotas]

for i in range(numAlunos):
    print('O aluno', nomes[i], 'teve média ', medias[i])
```



# Escrita de Arquivos de Texto

```
numNotas = 4
numAlunos = 40

arquivo = open("notas-algoritmos-2019.txt", "w")

for aluno in range(numAlunos):
    nome = input("Informe o nome do aluno: ")
    soma = 0
    for nota in range(numNotas):
        valor = float(input('Digite uma nota: '))
        soma += valor
    media = soma/numNotas
    minhastring = nome+"- "+str(media)+"\n"
    arquivo.write(minhastring)
arquivo.close()
```

# Escrita de Arquivos de Texto

```
numNotas = 4
numAlunos = 40

arquivo = open("notas-algoritmos-2019.txt", "w")

for aluno in range(numAlunos):
    nome = input("Informe o nome do aluno: ")
    soma = 0
    for nota in range(numNotas):
        valor = float(input('Digite uma nota: '))
        soma += valor
    media = soma/numNotas
    minhastring = nome+"- "+str(media)+"\n"
    arquivo.write(minhastring)
arquivo.close()
```

```
numNotas = 4
numAlunos = 40

arquivo = open("notas-algoritmos-2019.txt", "w")

for aluno in range(numAlunos):
    nome = input("Informe o nome do aluno: ")
    soma = 0
    minhastring = nome+"- "
    for nota in range(numNotas):
        valor = float(input('Digite uma nota: '))
        minhastring += str(valor)+"- "
        soma += valor
    media = soma/numNotas
    minhastring += str(media)+"\n"
    arquivo.write(minhastring)
arquivo.close()
```

# Leitura de Arquivos de Texto

notas-algoritmos-2019.txt

```
Fulano de Tal-9.0-6.0-7.0-8.5-7.625
Cicrano de Tal-3.0-9.0-9.0-9.0-7.5
Beltrano de Tal-6.0-7.0-8.0-9.0-7.5
Fulano de Tal-9.0-6.0-7.0-8.5-7.625
Cicrano de Tal-3.0-9.0-9.0-9.0-7.5
Beltrano de Tal-6.0-7.0-8.0-9.0-7.5
```

```
arquivo = open("notas-algoritmos-2019.txt", "r")
infos = arquivo.readlines()
arquivo.close()

for i in range(len(infos)):
    print(infos[i])
```

# Leitura de Arquivos de Texto

## notas-algoritmos-2019.txt

```
Fulano de Tal-9.0-6.0-7.0-8.5-7.625
Cicrano de Tal-3.0-9.0-9.0-9.0-7.5
Beltrano de Tal-6.0-7.0-8.0-9.0-7.5
Fulano de Tal-9.0-6.0-7.0-8.5-7.625
Cicrano de Tal-3.0-9.0-9.0-9.0-7.5
Beltrano de Tal-6.0-7.0-8.0-9.0-7.5
```

```
arquivo = open("notas-algoritmos-2019.txt", "r")
infos = arquivo.readlines()
arquivo.close()

for i in range(len(infos)):
    print(infos[i])
```

```
nomes = []
medias = []

arquivo = open("notas-algoritmos-2019.txt", "r")

info = ' '
contador = 0
auxiliar = ''
while info != '':
    info = arquivo.read(1)
    if info != '-':
        if info == '\n' or info == '':
            medias += [auxiliar]
            contador = 0
            auxiliar = ''
        else:
            auxiliar += info
    else:
        if contador == 0:
            nomes += [auxiliar]
        contador += 1
        auxiliar = ''

arquivo.close()

for i in range(len(nomes)):
    print(nomes[i], medias[i])
```



Universidade Federal do Rio Grande  
Centro de Ciências Computacionais



# Algoritmos e Estruturas de Dados I

## Arquivos

Profs. Drs. Cleo Billa, Rafael Penna e Thiago da Silveira

1º Semestre de 2020