

# INFORMÁTICA PARA ENGENHARIA

## MATRIZES

Prof. Dr. Daniel Caetano

2019 - 1

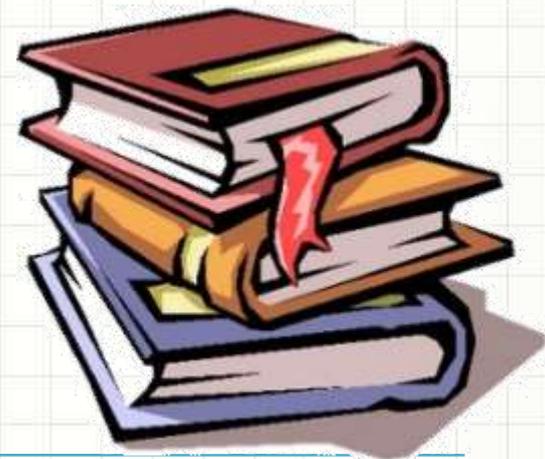
# Objetivos

- Compreender o uso das listas multidimensionais: as matrizes
- Transformar listas em matrizes numéricas
- Operações básicas com matrizes numéricas

- **Exercícios Aula 14**



# Material de Estudo



---

## Material

## Acesso ao Material

Notas de Aula e  
Apresentação

<http://www.caetano.eng.br/>  
(Informática para Engenharia – Aula 14)

Material Didático

Lógica de Programação, págs 151 a 172.

Biblioteca Virtual

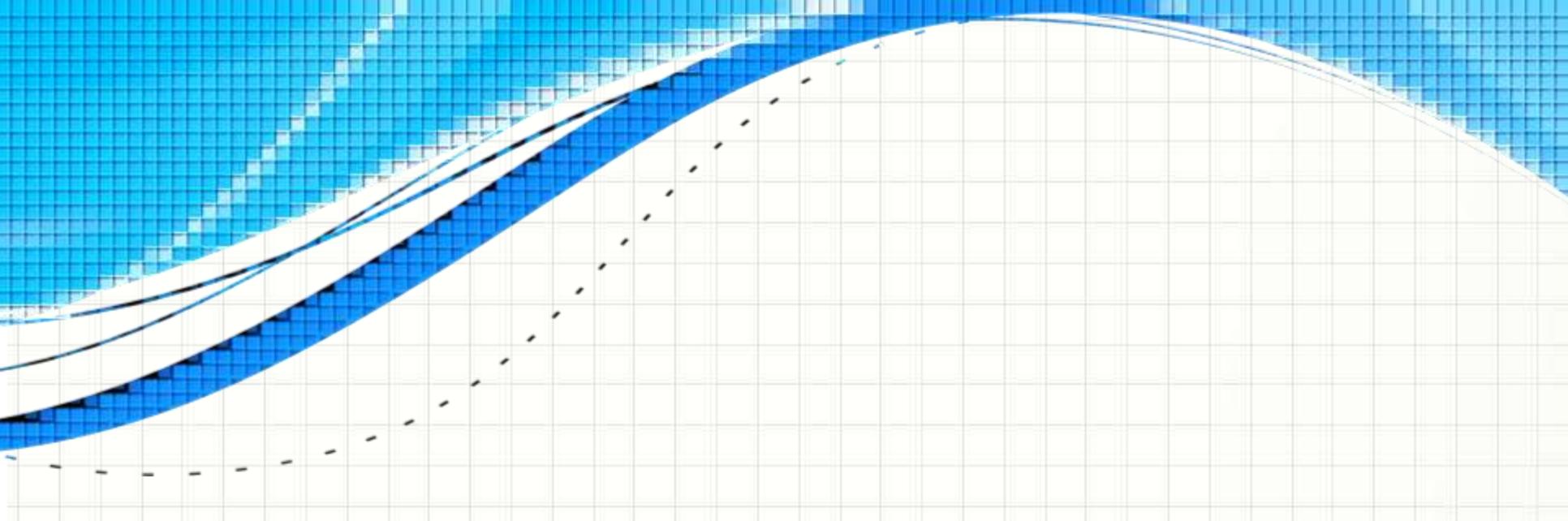
“Lógica de Programação – Fundamentos da  
Programação de Computadores”, págs 93 a 144.

Material Adicional

[https://www.ime.usp.br/~leo/mac2166/2017-1/introducao\\_vetores\\_python.html](https://www.ime.usp.br/~leo/mac2166/2017-1/introducao_vetores_python.html)  
<http://pythonize.org/tutorial-basico-numpy.html>

---

**LEMBRETE: CONSULTAR O “ANTES” DA AULA 15 NO SAVA!**



# **COMPREENDENDO A SITUAÇÃO PRÁTICA**

# Listas multidimensionais?

- Problema:
  - Leia o nome dos alunos e suas notas
  - Imprima o nome dos que estão acima da média
- Podemos resolver com duas listas separadas:

```
# Lê nomes e médias
```

```
NOMES = []
```

```
MEDIAS = []
```

```
for N in range(10) :
```

```
    NOMES.append(input("Digite o " + str(N+1) + "º nome: "))
```

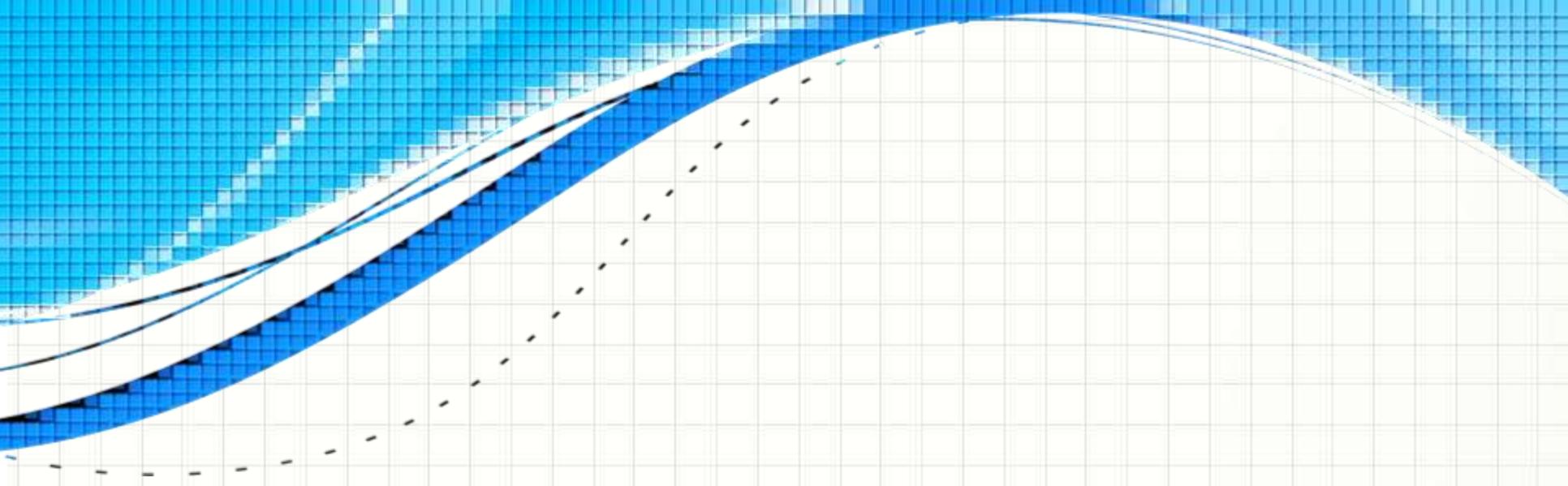
```
    MEDIAS.append(input("Digite a " + str(N+1) + "ª média: "))
```

```
for N in range(10) :
```

```
    if MEDIAS[N] >= 6.0 :
```

```
        print(NOMES[N])
```

**Não há um jeito mais “limpo”?**



# **LISTAS DE LISTAS**

**TAMBÉM CONHECIDAS COMO MATRIZES**

# Listas de Listas

- Podemos ter uma lista de listas
  - Cada elemento da lista... É outra lista

## Console

```
In [1]: matriz = [ [ 0, 1, 2 ],  
                  [ 3, 4, 5 ],  
                  [ 6, 7, 8 ] ]
```

```
In [2]: print( matriz )
```

**O que  
aconteceu?**

# Listas de Listas

- Podemos imprimir uma linha da lista

Console

```
In [3]: print( matriz[0] )
```

**O que  
aconteceu?**

# Listas de Listas

- Podemos imprimir um elemento da lista

Console

```
In [4]: print( matriz[0][2] )
```

**O que  
aconteceu?**

# Listas de Listas

- Podemos imprimir um elemento da lista

Console

```
In [5]: print( matriz[0,2] )
```

**O que  
aconteceu?**

# Listas de Listas

- Podemos percorrer as linhas da matriz

Console

```
In [6]: for X in matriz:  
        print( X )
```

**O que  
aconteceu?**

# Listas de Listas

- Podemos percorrer todos os elementos da matriz

## Console

```
In [7]: for linha in matriz:  
        for coluna in linha:  
            print( coluna )
```

**O que  
aconteceu?**

# Listas de Listas

- Melhorando o visual...
  - Teste essa variação!

## Console

```
In [8]: for linha in matriz:  
        for coluna in linha:  
            print( coluna )  
        print("----")
```

**O que  
aconteceu?**

# Exemplo

- Crie a agenda abaixo e faça com que ela seja impressa formatada:

Nome: Daniel

Telefone: 11-5555-1234

Data Nasc.: 10/02/1973

-----

aula14ex01.py

```
# Imprime agenda
```

```
agenda = [ ["Daniel", "11-5555-1234", "10/02/1973"],  
            ["João", "11-5555-2345", "02/10/1980"],  
            ["Alberto", "11-5555-3456", "11/01/1987"]]
```

# Exemplo

- Crie a agenda abaixo e faça com que ela seja impressa formatada:

Nome: Daniel  
Telefone: 11-5555-1234  
Data Nasc.: 10/02/1973  
-----

aula14ex01.py

```
# Imprime agenda

agenda = [ ["Daniel", "11-5555-1234", "10/02/1973"],
            ["João", "11-5555-2345", "02/10/1980"],
            ["Alberto", "11-5555-3456", "11/01/1987"]]

for contato in agenda:
    print("{:>12} {}".format("Nome:", contato[0]))
    print("{:>12} {}".format("Telefone:", contato[1]))
    print("{:>12} {}".format("Data Nasc.:", contato[2]))
    print("-----")
```

**Não precisa indicar os tipos!**

# Exemplo

- Cadastro de Lista de Notas
  - Perguntar nome do aluno
  - Perguntar nota
  - Quando nome vazio for digitado, finalizar
  - Ao final, imprimir
    - a média
    - A lista de alunos com nota acima da média
      - Incluir nome e média

# Exemplo

- Cadastro de Lista de Notas
  - Nome, nota... Para com nome vazio
  - Média e imprimir alunos acima da média

## aula14ex02.py

```
# Cadastro de notas de alunos
TURMA = []
while True:
    NOME = input("Nome: ")
    if len(NOME) == 0 :
        break
    NOTA = float(input("Nota: "))
    TURMA.append([NOME, NOTA])
```

## aula14ex02.py (cont)

```
TOTAL = 0
for ALUNO in TURMA :
    TOTAL = TOTAL + ALUNO[1]
MEDIA = TOTAL / len(TURMA)
print("Média: {:>2.1f}".format(MEDIA))
for ALUNO in TURMA:
    if ALUNO[1] > MEDIA :
        print(ALUNO[0], " - ", ALUNO[1])
```

# Exemplo

- Some todos os valores da matriz:

$$X = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

`aula14ex03.py`

```
# Soma elementos da matriz  
M = [[1,2],[3,4]]
```

# Exemplo

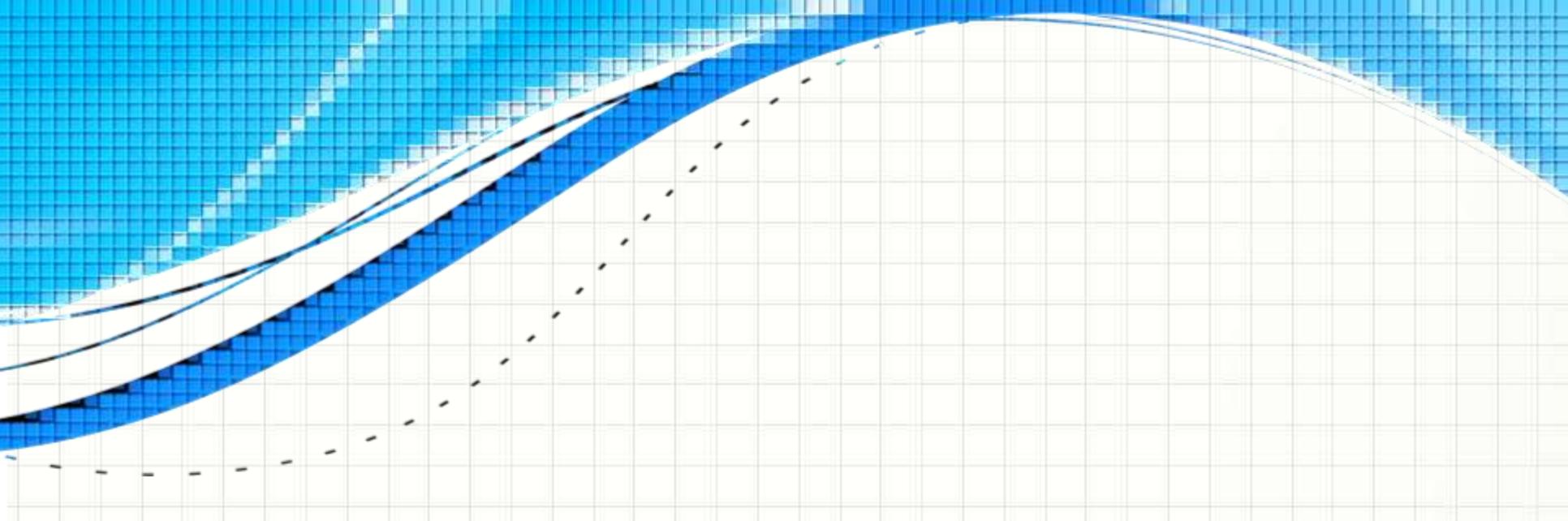
- Some todos os valores da matriz:

$$X = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

aula14ex03.py

```
# Soma elementos da matriz
M = [[1,2],[3,4]]

SOMA = 0
for LINHA in M:
    for VALOR in LINHA:
        SOMA = SOMA + VALOR
print("A soma é:", SOMA)
```



# **MATRIZES**

# **USANDO NUMPY**

# Contas com Matrizes “Lista”

- Matrizes padrão não permitem contas diretas

– Observe:

$$X = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad Y = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} \quad Z = \begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix}$$

$$Z = X + Y$$

aula14ex04.py

```
# Operações com matrizes
x = [[1,2] , [3,4]]
y = [[5,6] , [7,8]]
z = x + y
print(z)
```

# Matrizes com Numpy

- Numpy: biblioteca para operação com matrizes

– Observe:

$$X = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad Y = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} \quad Z = \begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix}$$

$$Z = X + Y$$

**aula14ex05.py**

```
# Operações com matrizes com Numpy
import numpy as np
x = np.array([[1,2] , [3,4]])
y = np.array([[5,6] , [7,8]])
z = x + y
print(z)
```

# Matrizes com Numpy

- Numpy: pode converter listas em matrizes

– Observe:  $X = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$      $Y = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$      $Z = \begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix}$

$$Z = X + Y$$

aula14ex06.py

```
# Operações com matrizes com Numpy
import numpy as np
x = [[1,2] , [3,4]]
y = [[5,6] , [7,8]]
i = np.array(x)
j = np.array(y)
z = i + j
print(z)
```

# Matrizes com Numpy

- Operações mais complexas

– Observe:

$$X = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad Y = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} \quad Z = \begin{bmatrix} 7 & 10 \\ 13 & 16 \end{bmatrix}$$

$$Z = 2 * X + Y$$

aula14ex07.py

```
# Operações com matrizes com Numpy
import numpy as np
x = np.array([[1,2] , [3,4]])
y = np.array([[5,6] , [7,8]])
z = 2*x + y
print(z)
```

# Operações do Numpy

- Há alguns comandos prontos no Numpy
  - Matriz transposta:  $M^t$

## Console

```
In [9]: import numpy as np
In [10]: matriz = np.array([[1,2,3], [4,5,6], [7,8,9]])
In [11]: print(matriz)
In [12]: matriz.transpose()
```

# Operações do Numpy

- Há alguns comandos prontos no Numpy
  - Soma dos elementos da matriz:

Console

```
In [13]: matriz.sum()
```

# Operações do Numpy

- Há alguns comandos prontos no Numpy
  - Mínimo e máximo da matriz:

## Console

```
In [14]: matriz.min()
```

```
In [15]: matriz.max()
```

# Operações do Numpy

- Há alguns comandos prontos no Numpy
  - Determinante da matriz: `det(M)`

## Console

```
In [16]: np.linalg.det(matriz)
```

```
In [17]: matriz2 = np.array([[10, 0], [0, 2]])
```

```
In [18]: np.linalg.det(matriz2)
```

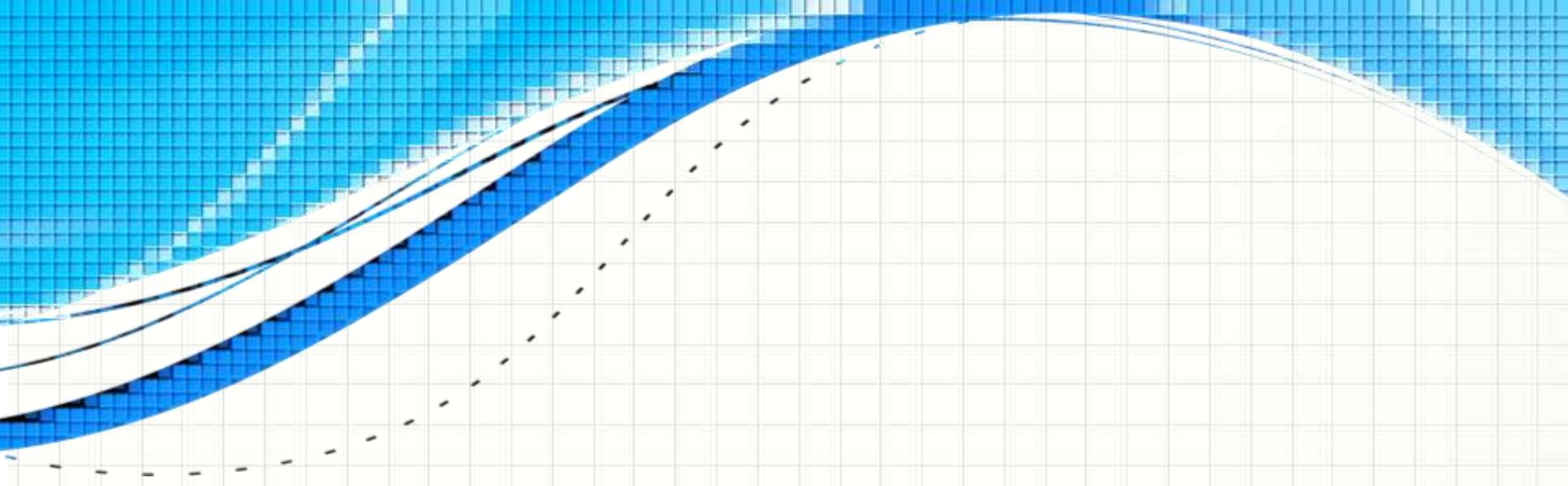
# Operações do Numpy

- Há alguns comandos prontos no Numpy
  - Inversa da matriz:  $M^{-1}$

## Console

```
In [19]: matriz2
```

```
In [20]: np.linalg.inv(matriz2)
```



**EXERCÍCIO**

# Exercício

- Faça um programa que calcule:

$$X = \begin{bmatrix} 10 & 5 \\ 7 & 3 \end{bmatrix} \quad Y = \begin{bmatrix} 1 & 22 \\ 7 & 14 \end{bmatrix}$$

Inversa



$$\det((2 * X - 3 * Y^t)^{-1})$$

Transposta



# Exercício

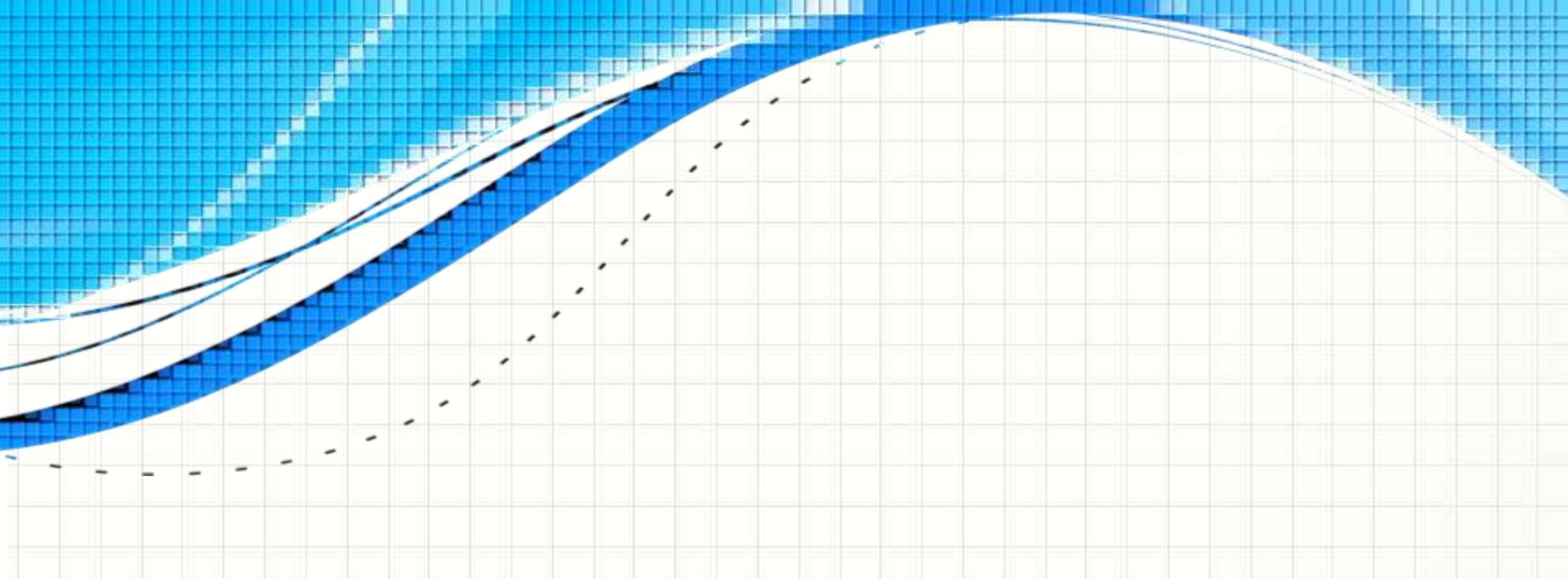
- Faça um programa que calcule:

$$X = \begin{bmatrix} 10 & 5 \\ 7 & 3 \end{bmatrix} \quad Y = \begin{bmatrix} 1 & 22 \\ 7 & 14 \end{bmatrix} \quad \det((2 * X - 3 * Y^t)^{-1})$$

aula14ex08.py

```
# Calcula expressão com matrizes
import numpy as np

X = np.array([[10,5],[7,3]])
Y = np.array([[1,22],[7,14]])
Z = np.linalg.det(np.linalg.inv(2*X - 3*Y.transpose()))
print(Z)
```



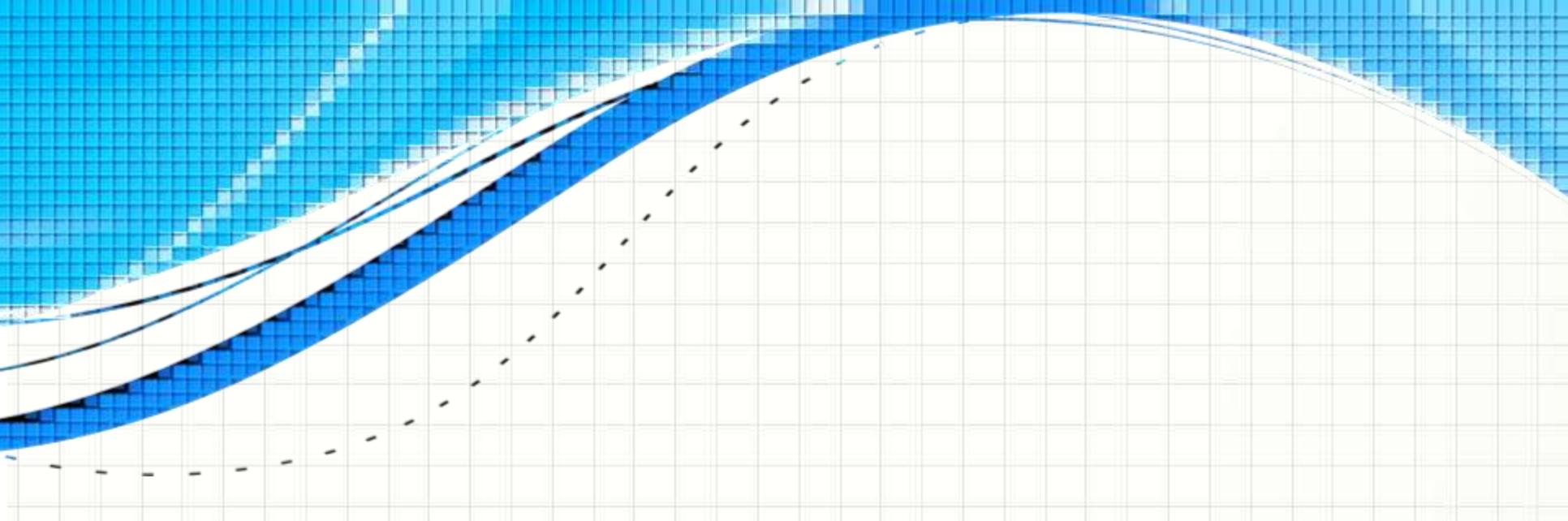
**CONCLUSÕES**

# Resumo

- Matrizes
  - Listas de listas...
  - Matrizes numéricas: numpy
- Muitas operações úteis prontas
  - Por isso engenheiros gostam tanto do Python!

**SAVA15!**

- Estruturas de dados avançadas!
- Para que mais servem as listas?
  - Filas e pilhas!



**PERGUNTAS?**

# Exercício para Casa

- Faça um programa que calcule o determinante da matriz 3x3 abaixo:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

- Faça um programa que some todos os elementos da matriz acima, sem usar **sum()**
- Desafio: Faça um programa que leia os 9 valores de uma matriz 3x3 e calcule seu determinante