

---

## Guía de laboratorio 1: Álgebra Matricial con Python.

### 1. Introducción

El álgebra matricial constituye una herramienta fundamental en la ingeniería y las ciencias aplicadas, ya que permite modelar y resolver problemas complejos de manera sistemática. Operaciones como la suma, la multiplicación, el cálculo de determinantes y la obtención de inversas son esenciales en el análisis de sistemas lineales, circuitos eléctricos, transformaciones geométricas, mecánica estructural y procesamiento de datos.

### 2. Objetivos

#### 2.1 Objetivo General

Aplicar los conceptos de álgebra matricial en la resolución de problemas computacionales mediante Python, fortaleciendo la comprensión teórica y la capacidad de modelar situaciones reales en ingeniería.

#### 2.2 Objetivos Específicos

- Implementar operaciones básicas de matrices (suma y multiplicación) utilizando Python y verificar los resultados manualmente.
- Calcular determinantes de matrices para analizar propiedades de sistemas lineales y condiciones de solución.
- Obtener la matriz inversa y comprobar su validez mediante la multiplicación con la matriz original.
- Relacionar las operaciones matriciales con aplicaciones reales en ingeniería.
- Desarrollar habilidades de programación en Python orientadas a la manipulación matricial con NumPy.

### 3. Materiales y Recursos

- Computadora personal o laboratorio de cómputo.
- Python 3.10+ instalado (preferible en Anaconda).
- Librería NumPy instalada (pip install numpy).
- Editor de código (Sublime Text, Visual Studio Code o Jupyter Notebook).
- Opcional (crear un ambiente virtual)

### 4. Procedimiento General

- Abrir el entorno de programación (Jupyter Notebook, VSCode o Sublime Text).
- Importar la librería NumPy.
- Definir matrices en Python como arreglos bidimensionales.
- Ejecutar las operaciones matriciales solicitadas (suma, multiplicación, determinante, inversa).
- Verificar manualmente los resultados en al menos un caso por operación.
- Relacionar los resultados obtenidos con aplicaciones prácticas.

## 5. Desarrollo de ejercicios

### 5.1 Suma de matrices

Se hace elemento a elemento:

$$[a_{ij}] + [b_{ij}] = [a_{ij} + b_{ij}].$$

```
1. import numpy as np
2.
3. # =====
4. # 1. ALGEBRA MATRICIAL: SUMA DE MATRICES
5. # =====
6.
7. # Ejercicio 1: Suma de matrices 2x2
8. A1 = np.array([[1, 2],
9.                [3, 4]])
10. B1 = np.array([[5, 6],
11.                [7, 8]])
12. print("Ejercicio 1:\n", A1 + B1, "\n")
13.
14. # Ejercicio 2: Suma de matrices 3x3
15. A2 = np.array([[2, 0, 1],
16.                [1, 3, 2],
17.                [0, 4, 5]])
18. B2 = np.array([[1, 2, 3],
19.                [4, 0, 6],
20.                [7, 8, 9]])
21. print("Ejercicio 2:\n", A2 + B2, "\n")
22.
23. # Ejercicio 3: Suma de matrices 2x3
24. A3 = np.array([[1, -2, 3],
25.                [4, 5, -6]])
26. B3 = np.array([[7, 8, -9],
27.                [-1, 2, 3]])
28. print("Ejercicio 3:\n", A3 + B3, "\n")
29.
30. # Ejercicio 4: Suma de matrices 3x2
31. A4 = np.array([[2, 4],
32.                [6, 8],
33.                [10, 12]])
34. B4 = np.array([[1, 1],
35.                [1, 1],
36.                [1, 1]])
37. print("Ejercicio 4:\n", A4 + B4, "\n")
```

## 5.2 Multiplicación de matrices

Se realiza fila x columna

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 4 & 4 \\ 10 & 8 \end{bmatrix}$$

```
import numpy as np

# Ejercicio 1: Multiplicación de matrices 2x2
A1 = np.array([[1, 2],
               [3, 4]])
B1 = np.array([[2, 0],
               [1, 2]])
print("Ejercicio 1:\n", np.dot(A1, B1), "\n")

# Ejercicio 2: Multiplicación de matrices 2x3 * 3x2 = 2x2
A2 = np.array([[1, 2, 3],
               [4, 5, 6]])
B2 = np.array([[7, 8],
               [9, 10],
               [11, 12]])
print("Ejercicio 2:\n", np.dot(A2, B2), "\n")

# Ejercicio 3: Multiplicación de matrices 3x3
A3 = np.array([[2, 0, 1],
               [1, 3, 2],
               [0, 4, 5]])
B3 = np.array([[1, 2, 3],
               [4, 0, 6],
               [7, 8, 9]])
print("Ejercicio 3:\n", np.dot(A3, B3), "\n")

# Ejercicio 4: Multiplicación de matrices 3x2 * 2x3 = 3x3
A4 = np.array([[1, 2],
               [3, 4],
               [5, 6]])
B4 = np.array([[7, 8, 9],
               [10, 11, 12]])
print("Ejercicio 4:\n", np.dot(A4, B4), "\n")
```

### 5.3 Determinantes

Calcular el determinante de la matriz:

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

Verificar manualmente que:  $\det(M) = ad - bc$

NOTA:

Escalar que indica el “volumen” y si la matriz es invertible.

Si el determinante = 0 → no tiene inversa.

```
6 import numpy as np
7
8 # Ejercicio 1: Determinante de una matriz 2x2
9 A1 = np.array([[4, 2],
10               [3, 1]])
11 det1 = np.linalg.det(A1)
12 print("Ejercicio 1 - Determinante:\n", det1, "\n")
13
14 # Ejercicio 2: Determinante de una matriz 3x3
15 A2 = np.array([[1, 2, 3],
16               [0, 1, 4],
17               [5, 6, 0]])
18 det2 = np.linalg.det(A2)
19 print("Ejercicio 2 - Determinante:\n", det2, "\n")
20
21 # Ejercicio 3: Determinante de una matriz 4x4
22 A3 = np.array([[2, 0, 1, 3],
23               [1, 2, 0, 4],
24               [3, 1, 2, 5],
25               [0, 2, 3, 1]])
26 det3 = np.linalg.det(A3)
27 print("Ejercicio 3 - Determinante:\n", det3, "\n")
28
29 # Ejercicio 4: Determinante de una matriz 3x3 triangular
30 A4 = np.array([[2, 1, 0],
31               [0, 3, 4],
32               [0, 0, 5]])
33 det4 = np.linalg.det(A4)
34 print("Ejercicio 4 - Determinante:\n", det4, "\n")
```

#### 5.4 Matriz Inversa

Calcular la inversa de:

$$N = \begin{bmatrix} 2 & 1 \\ 5 & 3 \end{bmatrix}$$

y comprobar que:  $N \cdot N^{-1} = I$

```
4. import numpy as np
5.
6. # Ejercicio 1: Matriz inversa de una 2x2
7. A1 = np.array([[4, 7],
8.               [2, 6]])
9. inv1 = np.linalg.inv(A1)
10. print("Ejercicio 1 - Inversa de A1:\n", inv1, "\n")
11.
12. # Ejercicio 2: Matriz inversa de una 3x3
13. A2 = np.array([[1, 2, 3],
14.               [0, 1, 4],
15.               [5, 6, 0]])
16. inv2 = np.linalg.inv(A2)
17. print("Ejercicio 2 - Inversa de A2:\n", inv2, "\n")
18.
19. # Ejercicio 3: Matriz inversa de una 4x4
20. A3 = np.array([[1, 2, 0, 1],
21.               [0, 1, 3, 2],
22.               [2, 0, 1, 1],
23.               [1, 1, 0, 1]])
24. inv3 = np.linalg.inv(A3)
25. print("Ejercicio 3 - Inversa de A3:\n", inv3, "\n")
26.
27. # Ejercicio 4: Matriz inversa de una triangular superior 3x3
28. A4 = np.array([[2, 1, 1],
29.               [0, 3, 2],
30.               [0, 0, 4]])
31. inv4 = np.linalg.inv(A4)
32. print("Ejercicio 4 - Inversa de A4:\n", inv4, "\n")
```