



# Módulo 3

## Sesión N° 2



### ACTIVIDAD:



## Resolución de Sistemas de Ecuaciones y Aplicaciones de Transformaciones 2D

- Objetivo: Resolver sistemas de ecuaciones lineales utilizando técnicas de álgebra matricial (mediante funciones como `np.linalg.solve` y `np.linalg.lstsq`). Modelar y visualizar transformaciones lineales en 2D (como rotaciones y escalados) aplicándolas sobre un conjunto de puntos para evidenciar el efecto de dichas transformaciones.



### Instrucciones:

#### 1. Resolución de Sistemas de Ecuaciones Lineales:

##### ○ Datos Fijos:

- Utilizar una matriz  $A$  de  $3 \times 3$  y un vector  $b$  de 3 elementos, por ejemplo:

```
A = np.array([[3, -1, 2],  
             [1, 2, 1],  
             [2, 1, 3]], dtype=float)  
b = np.array([5, 6, 7], dtype=float)
```

##### ○ Implementación:

- Resolver el sistema  $Ax = b$  empleando `np.linalg.solve(A, b)`.
- En caso de encontrarse con sistemas sobre- o sub-determinados, se deberá emplear `np.linalg.lstsq(A, b, rcond=None)` para hallar la solución en el sentido de mínimos cuadrados.



#### 2. Aplicación de Transformaciones Matriciales en 2D:

##### ○ Parámetros Fijos:

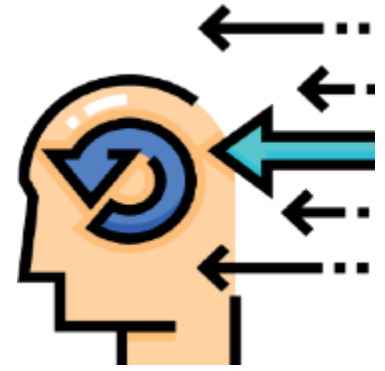
- Ángulo de rotación:  $45^\circ$  (convertir a radianes usando `np.radians(45)`).
- Factor de escalado: 1.5.

##### ○ Matriz de Transformación Compuesta:

- Crear la matriz de rotación:

$$R(45^\circ) = \begin{pmatrix} \cos(45^\circ) & -\sin(45^\circ) \\ \sin(45^\circ) & \cos(45^\circ) \end{pmatrix}$$

- Crear la matriz de escalado:



$$S = \begin{pmatrix} 1.5 & 0 \\ 0 & 1.5 \end{pmatrix}$$

- La matriz compuesta es:

$$T = S \cdot R$$

- Conjunto de Puntos:

- Utilizar el siguiente conjunto de puntos fijos:

```
puntos = np.array([[1, 2],  
                  [3, 1],  
                  [2, 4],  
                  [4, 3],  
                  [0, 0],  
                  [3, 3]], dtype=float)
```

- Implementación:

- Aplicar la transformación T a cada punto.
- Visualizar, mediante Matplotlib, los puntos originales y los transformados, utilizando gráficos que conecten cada punto original con su correspondiente punto transformado.



### 3. Documentación y Comentarios:

- Cada función o bloque del código debe incluir docstrings y comentarios que expliquen claramente su propósito, parámetros y retorno.
- Se debe incluir un breve documento (archivo README) que resuma la estructura del proyecto y justifique las decisiones de diseño adoptadas en la solución.
- Se debe presentar la salida en consola con la solución del sistema de ecuaciones (los valores obtenidos para x).
- Incluir capturas de pantalla del código ejecutado y del gráfico que evidencie la transformación 2D.

### 4. Entrega:

- Tiempo estimado Desarrollo: 100 minutos.
- Formato de Ejecución: Grupal.

