

transformaciones a vectores - E

November 24, 2024

1 Matriz para Letra A

2 Trabajo realizado por: Jesus Enrique Lugo ramirez

3 Graficación Computacional

3.1 Profesora: Hazem Álvarez Rodríguez

3.1.1 Clase del 20 de noviembre de 2024

```
[1]: import numpy as np
import matplotlib.pyplot as plt

# Definimos la matriz E con los puntos de la letra "E" corregida
E = np.array([
    (0, 0, 1), (3, 0, 1), (3, 1, 1), (1, 1, 1), (1, 2, 1),
    (2, 2, 1), (2, 3, 1), (1, 3, 1), (1, 4, 1), (3, 4, 1),
    (3, 5, 1), (0, 5, 1), (0, 0, 1) # Cerramos la forma de la letra E
])

# Definir matrices de transformación
Ic = np.eye(3)

# Matriz de Reflexión con respecto al eje x
Refx = np.array([[1., 0, 0], [0, -1., 0], [0., 0., 1.]])

# Matriz de Reflexión con respecto al eje y
Refy = np.array([[-1., 0, 0], [0, 1., 0], [0., 0., 1.]])

# Matriz de Rotación
theta = np.pi / 3 # Ángulo de rotación deseado
R = np.array([
    [np.cos(theta), np.sin(theta), 0.],
    [-np.sin(theta), np.cos(theta), 0.],
    [0., 0., 1.]
])

# Matriz de cambio de escala
```

```

s = 2 # Escalar
S = np.array([
    [s, 0, 0.],
    [0., s, 0.],
    [0., 0., 1.]
])

# Matriz de deformación horizontal/vertical
h = -1
v = 2
D = np.array([
    [1., h, 0.],
    [v, 1., 0.],
    [0., 0., 1.]
])

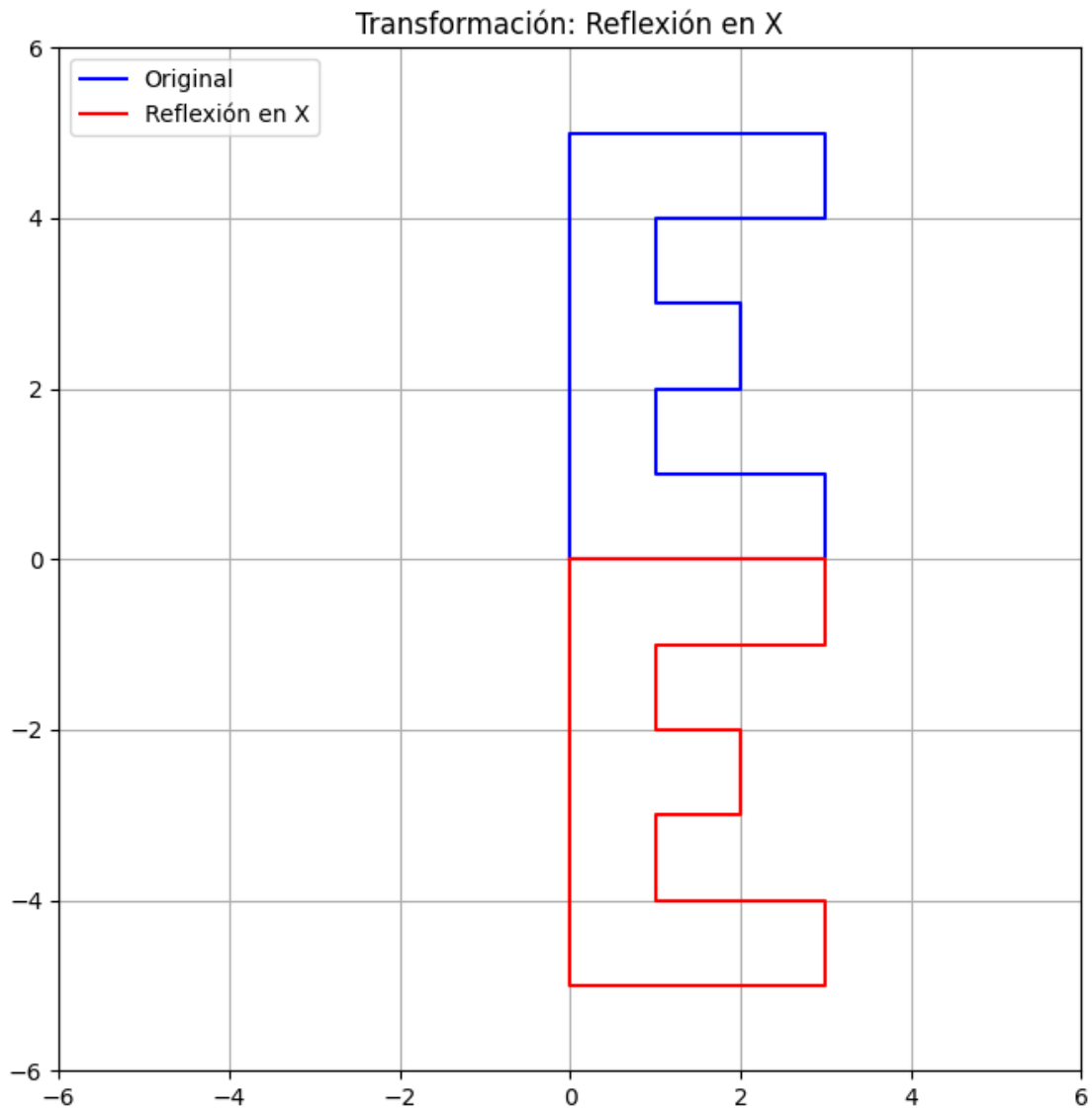
# Matriz de traslación
tx = -3
ty = -5
T = np.array([
    [1., 0, tx],
    [0, 1., ty],
    [0., 0., 1.]
])

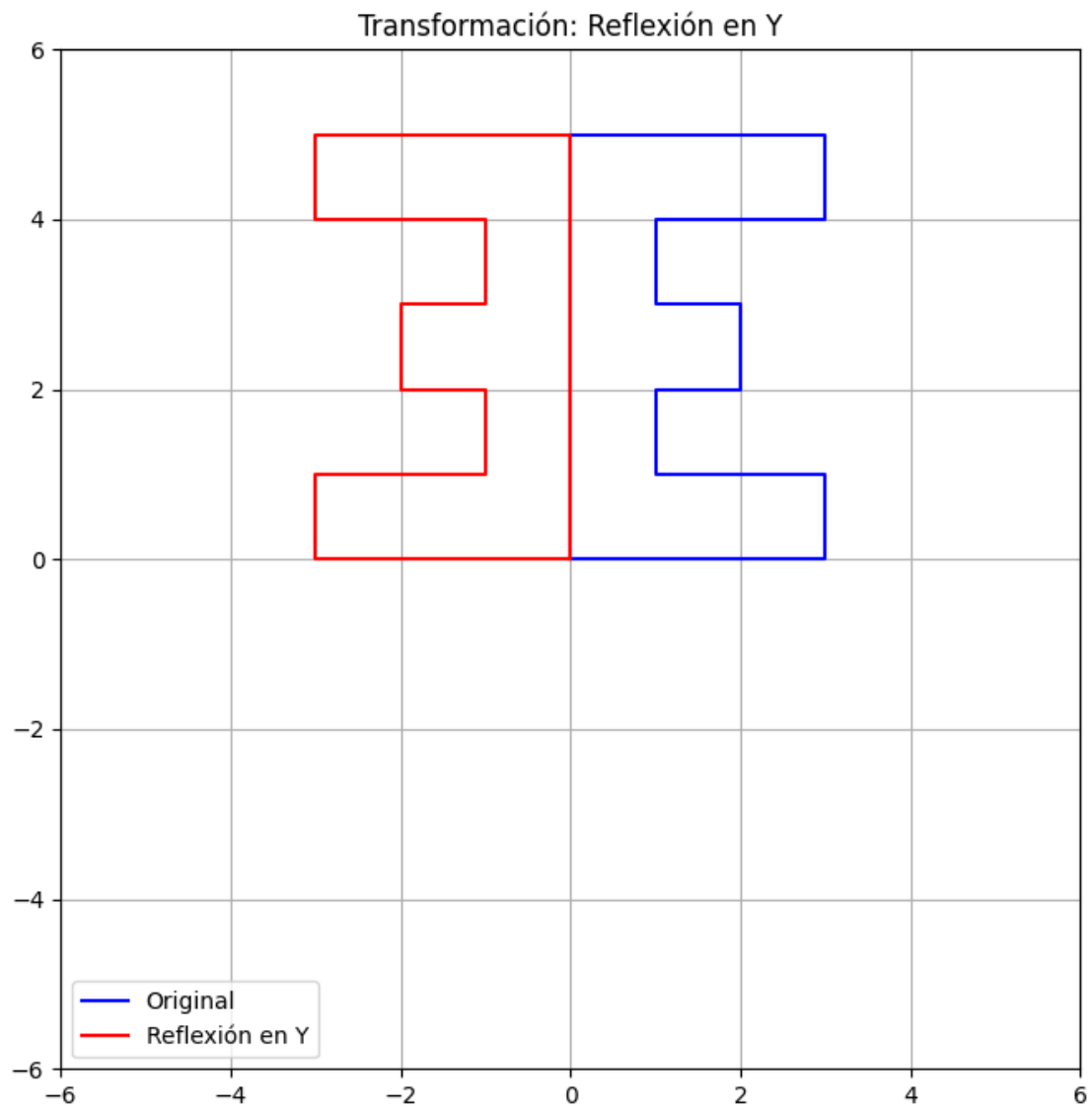
# Graficar transformaciones aplicadas a la matriz E
def graficar_transformacion(E, transformacion, titulo):
    Ex, Ey = [], []
    for row in E:
        output_row = transformacion.dot(row) # Aplicar transformación
        x, y, _ = output_row
        Ex.append(x)
        Ey.append(y)

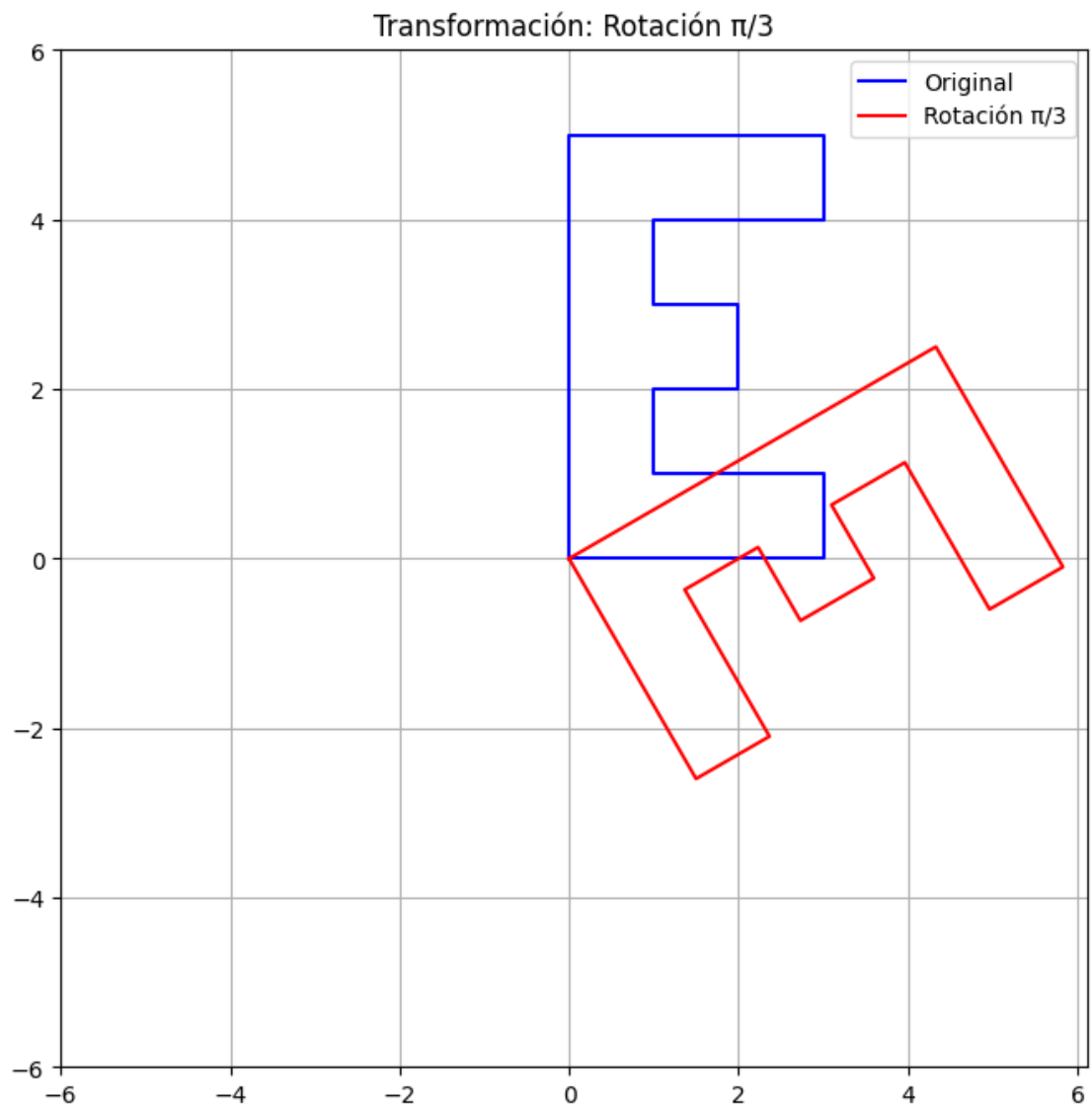
    # Graficar letra original y transformada
    plt.figure(figsize=(8, 8))
    plt.plot(E[:, 0], E[:, 1], color="blue", label="Original")
    plt.plot(Ex, Ey, color="red", label=titulo)
    ax = plt.gca()
    ax.set_xticks(np.arange(-6, 8, 2))
    ax.set_yticks(np.arange(-6, 8, 2))
    ax.set_aspect('equal', adjustable='box')
    plt.title(f"Transformación: {titulo}")
    plt.grid()
    plt.legend()
    plt.show()

```

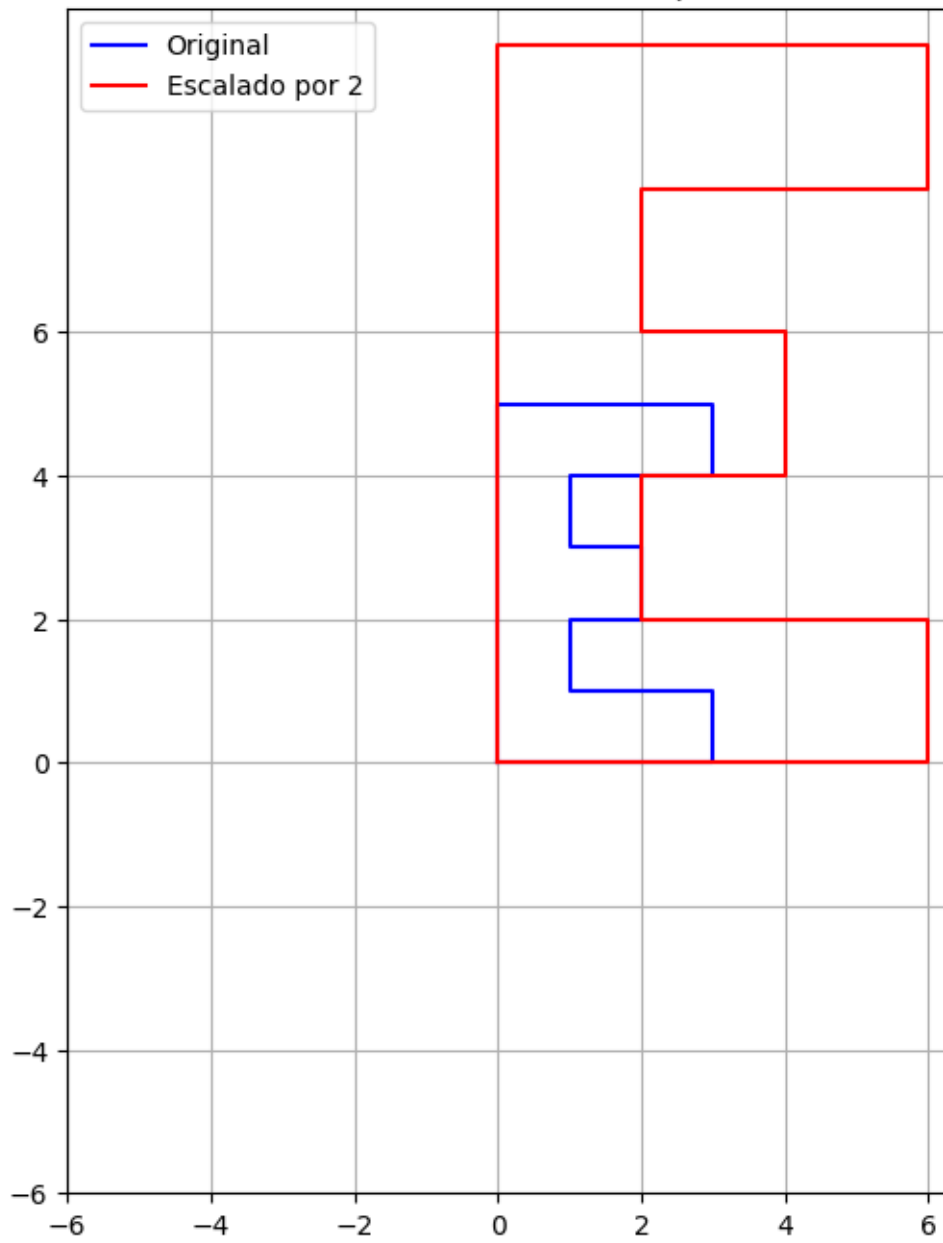
```
# Aplicar y graficar cada transformación
graficar_transformacion(E, Refx, "Reflexión en X")
graficar_transformacion(E, Refy, "Reflexión en Y")
graficar_transformacion(E, R, "Rotación /3")
graficar_transformacion(E, S, "Escalado por 2")
graficar_transformacion(E, D, "Deformación (h=-1, v=2)")
graficar_transformacion(E, T, "Traslación (-3, -5)")
```



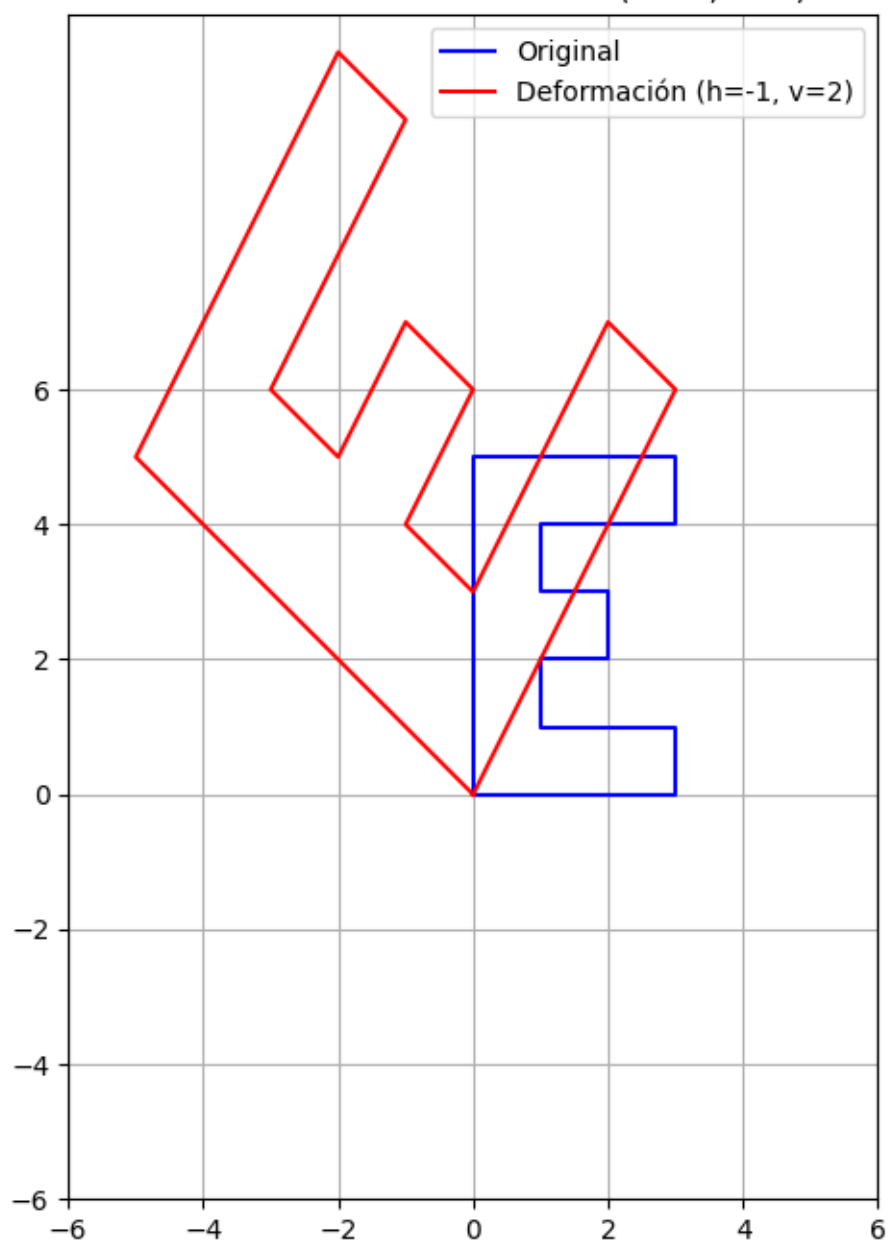


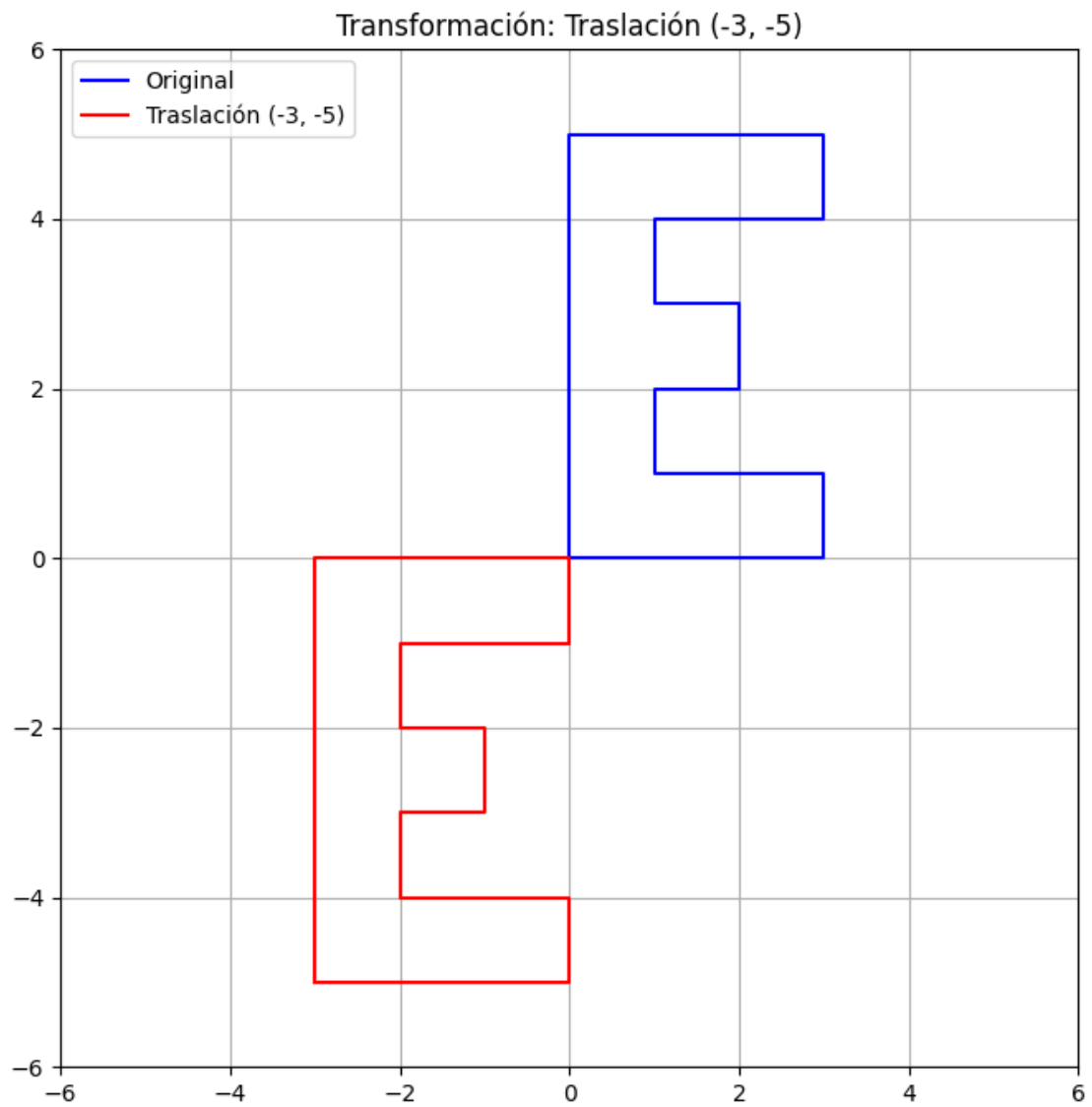


Transformación: Escalado por 2



Transformación: Deformación ($h=-1$, $v=2$)





[]: