ALUMNO: **Fernando Alfonso Arana Salas**

MATRICULA: **A01272933**

Propósito

Evaluar tu habilidad para transformar objetos tridimensionales matemáticamente, mediante transformaciones homogéneas, tanto por procedimiento matemático, como por programación computacional de la solución.

Instrucciones

A partir de las matrices geométricas de traslación, escalado y rotación en 3D, deberás de realizar la siguiente actividad:

1. (30 pts) Tomando como base la teoría y códigos vistos en clase, implementa la función de rotación libre "rotate", que admite como parámetros: theta (ángulo de rotación), (x, y, z), eje de rotación a usar.

```
# Función de Rotación Libre
   def rotate(self, theta, a, b, c):

       if (b == 0 and c == 0):
           return self.rotate_x(theta)

       # Normalizar el vector de rotación
       norm = np.sqrt(a**2 + b**2 + c**2)
       a /= norm
       b /= norm
       c /= norm
       d = np.sqrt(b ** 2 + c ** 2)

       Rx = np.identity(4)
       Rx[1][1] = c / d
       Rx[1][2] = -1 * b / d
       Rx[2][1] = b / d
       Rx[2][2] = c / d

       Ry = np.identity(4)
       Ry[0][0] = d
       Ry[0][2] = a
       Ry[2][0] = -1 * a
       Ry[2][2] = d

       Rx_inv = np.identity(4)
       Rx_inv[1][1] = c / d
```

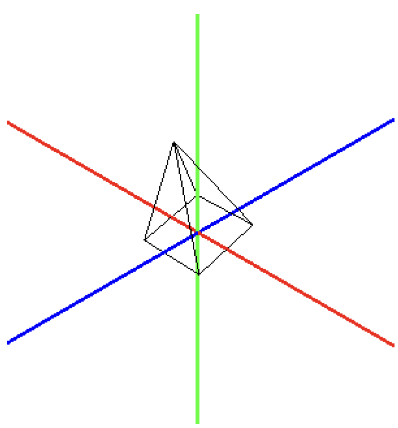
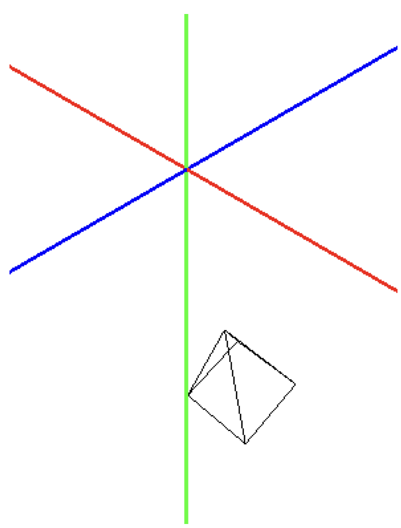
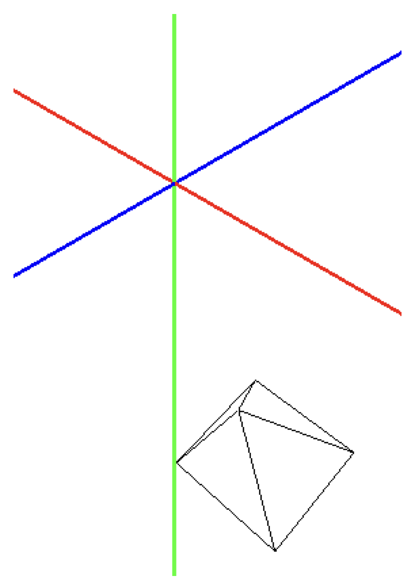
```
Rx_inv[1][2] = b / d
Rx_inv[2][1] = -1 * b / d
Rx_inv[2][2] = c / d

Ry_inv = np.identity(4)
Ry_inv[0][0] = d
Ry_inv[0][2] = -1 * a
Ry_inv[2][0] = a
Ry_inv[2][2] = d

RxRy = Ry @ Rx
RxRy_inv = Ry_inv @ Rx_inv
self.rotate_z(theta)
RxRy_inv @ self.A @ RxRy
```

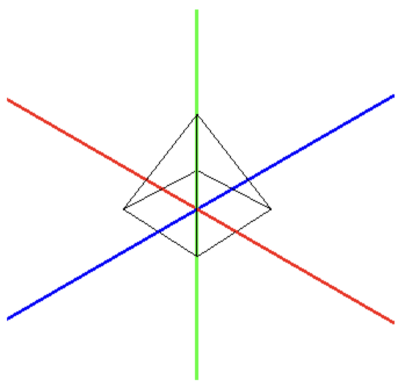
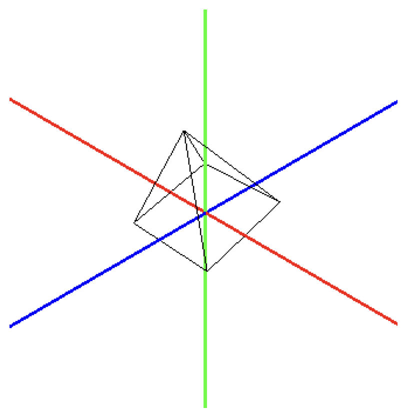
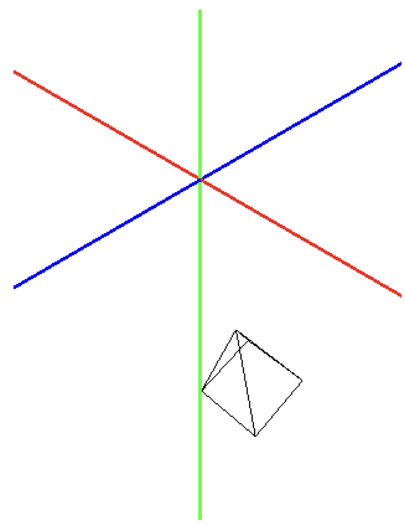
2. Usando la clase "pirámide" vista en clase, aplicar las siguientes operaciones en la secuencia que se indica; realiza una captura de pantalla del resultado obtenido como evidencia de cada uno de los casos planteados:
    1. (25 pts)

       1: rotación libre con theta = 15 grados y eje de rotación = (0.5, 0.5, 0.0); 2: traslación = (3.0, - 5.0, 1.0); 3: escalado = (1.5, 1.0, 1.5)

1: rotación libre con theta = 15 grados y eje de rotación = (0.5, 0.5, 0.0)	**2: traslación = (3.0, - 5.0, 1.0)**	**3: escalado = (1.5, 1.0, 1.5)**
		

2. (25 pts)

   1: escalado = (1.5, 1.0, 1.5); 2: rotación libre con theta = 15 grados y eje de rotación = (0.5, 0.5, 0.0); 3: traslación = (3.0, - 5.0, 1.0)

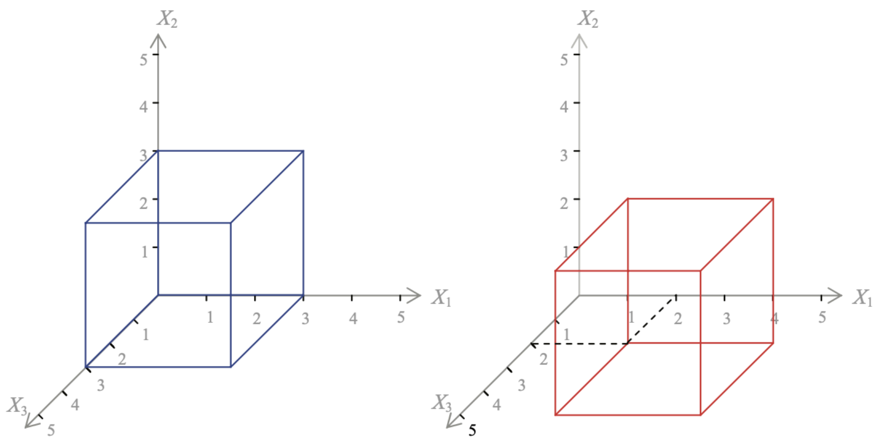
1: escalado = (1.5, 1.0, 1.5)	**2: rotación libre con theta = 15 grados y eje de rotación = (0.5, 0.5, 0.0)**	**3: traslación = (3.0, - 5.0, 1.0)**
		

3. (20 pts) Redacta en tus palabras una reflexión alrededor de los operadores geométricos de traslación, rotación y escalado 3D vistos en clase, abordando aspectos como: la facilidad o complejidad que implica manipular un objeto 3D; uso de operaciones matriciales para la implementación de los operadores; guías proporcionadas para la comprensión del tema; conclusiones personales

Reflexión sobre Translación

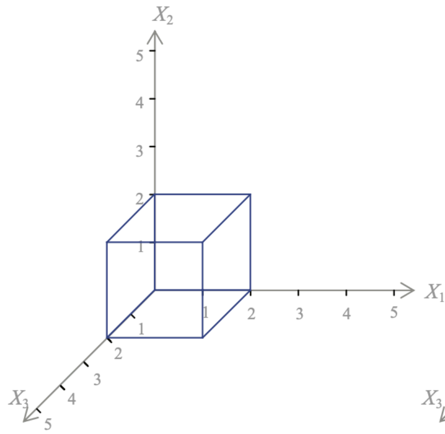
Facilidad	**Complejidad**	**Uso de Operaciónes Matriciales**

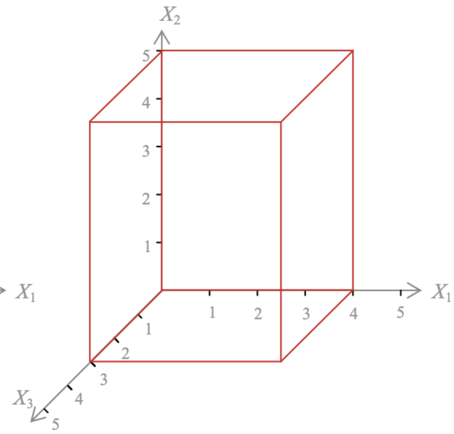
La translación en el espacio tridimensional resulta relativamente sencilla de comprender y aplicar. La operación implica desplazar un objeto a lo largo de los ejes x, y, y z, lo que puede visualizarse como mover el objeto en el espacio.	Aunque es fácil conceptualmente, la complejidad radica en la coordinación precisa del desplazamiento en cada eje. Además, la realización manual puede volverse más complicada a medida que se añaden objetos o se combinan diferentes operaciones geométricas.	$\cdot\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ d_1 & d_2 & d_3 & 1 \end{bmatrix}$



Reflexión sobre Escalado

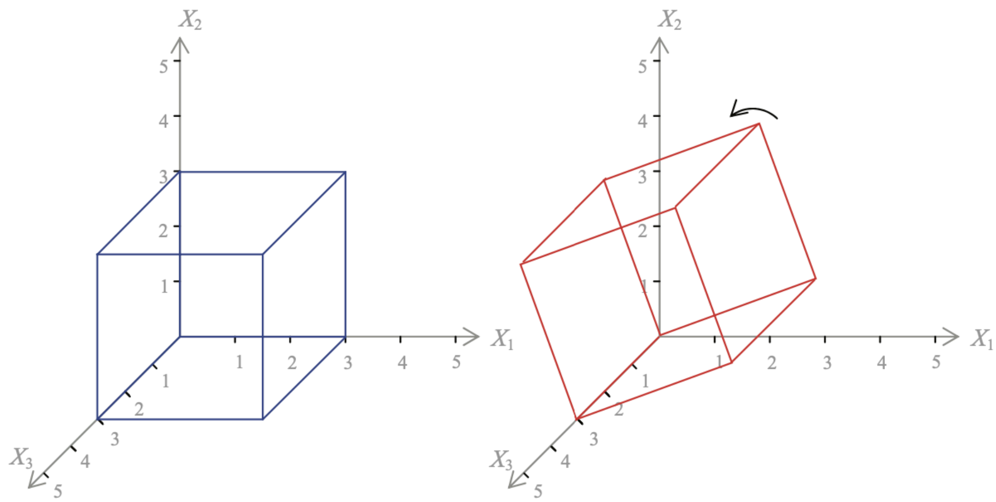
Facilidad	**Complejidad**	**Uso de Operaciones Matriciales**
El escalado 3D implica cambiar las dimensiones de un objeto en el espacio tridimensional. Conceptualmente, es fácil entender cómo el objeto se agranda o reduce en función de factores de escala en cada dimensión.	La complejidad surge al mantener la proporcionalidad y evitar deformaciones no deseadas. Coordinar el escalado de diferentes partes de un objeto puede requerir cuidado para no afectar su apariencia general.	$\begin{bmatrix} s_1 & 0 & 0 & 0 \\ 0 & s_2 & 0 & 0 \\ 0 & 0 & s_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$





Reflexión sobre Rotación Eje X_1, X_2, X_3

Facilidad	**Complejidad**	**Uso de Operaciónes Matriciales**
La rotación en 3D implica girar un objeto alrededor de un eje. Aunque el concepto es comprensible, coordinar la rotación en múltiples ejes puede volverse desafiante.	La complejidad aumenta al realizar rotaciones múltiples y al tratar con diferentes sistemas de coordenadas. Mantener la orientación deseada puede requerir un análisis cuidadoso.	**Eje X_1** $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ **Eje X_2** $\begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ **Eje X_3** $\begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$



Reflexión sobre Rotación Libre (Eje Arbitrario)

Facilidad	**Complejidad**	**Uso de Operaciones Matriciales**
La rotación libre alrededor de un eje arbitrario introduce un grado adicional de complejidad. Aunque conceptualmente es comprensible, coordinar la rotación alrededor de un eje no alineado con los ejes estándar puede resultar más desafiante para visualizar y ejecutar.	La complejidad aumenta debido a la necesidad de normalizar el vector de rotación y calcular la matriz de rotación correspondiente. La introducción de componentes adicionales para gestionar la rotación alrededor del eje arbitrario puede aumentar la probabilidad de errores y dificultades en la implementación.	$IDA = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -a_1 & -a_2 & -a_3 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_1 & -\sin\alpha_1 & 0 \\ 0 & \sin\alpha_1 & \cos\alpha_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\alpha_2 & -\sin\alpha_2 & 0 & 0 \\ \sin\alpha_2 & \cos\alpha_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $REG = \begin{bmatrix} \cos\alpha_2 & \sin\alpha_2 & 0 & 0 \\ -\sin\alpha_2 & \cos\alpha_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_1 & \sin\alpha_1 & 0 \\ 0 & -\sin\alpha_1 & \cos\alpha_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ a_1 & a_2 & a_3 & 1 \end{bmatrix}$

Conclusión

En conclusión, la manipulación de objetos 3D a través de translación, escalado y rotación implica un equilibrio entre la comprensión conceptual y la coordinación precisa de operaciones. El uso de operaciones matriciales nos ofrece una forma poderosa y eficiente de implementar estas transformaciones, facilitando la programación de aplicaciones gráficas y simulaciones tridimensionales. La complejidad aumenta con la combinación de múltiples operaciones, pero la representación matricial proporciona una herramienta valiosa para gestionar estas transformaciones de manera efectiva. Es útil y valioso aprender dichas operaciones matemáticas para conocer el funcionamiento de bajo nivel de programas como Unity o Unreal.