

**TECNOLÓGICO NACIONAL DE MÉXICO**  
**INSTITUTO TECNOLÓGICO DE CUAUTLA**

---

**INGENIERÍA EN SISTEMAS**

**GRAFICACIÓN**

**SEMESTRE ENERO JUNIO-2026**

DOCENTE: CABALLERO ALFARO ARÍSTIDES

**UNIDAD 1**

**MARTÍNEZ GALLARDO SANTIAGO**

4TO SEMESTRE

GRUPO: 1

25 DE FEBRERO DEL 2026

Este nuevo script representa un salto cualitativo en el uso de la programación para generar geometría sagrada, específicamente el patrón base de la **Flor de la Vida**. A diferencia del código anterior que construía un solo objeto punto por punto, este utiliza operadores de alto nivel de Blender para generar múltiples círculos interconectados con una precisión matemática absoluta. El proceso se divide en una fase de limpieza, una configuración geométrica y una ejecución cíclica que posiciona cada elemento en el espacio 3D.

El código comienza definiendo una función de utilidad llamada `limpiar_lienzo`, la cual se encarga de seleccionar y eliminar todos los objetos presentes en la escena. Esto es fundamental para asegurar que, cada vez que ejecutes el script, el resultado sea limpio y no se encimen geometrías de ejecuciones previas. Tras limpiar la escena, el script establece los parámetros fundamentales de la composición: un radio ( $R$ ) de 3 unidades y una división de 6 lados. La variable salto se calcula dividiendo 360 grados entre los lados, lo que resulta en un intervalo de **60°**. Este valor es la clave geométrica, ya que en la Flor de la Vida, cada círculo periférico se sitúa exactamente a un radio de distancia del centro y desplazado por ángulos de 60°.

La construcción visual inicia con la creación de un círculo central ubicado en las coordenadas origen  $(0, 0, 0)$  mediante el comando `bpy.ops.mesh.primitive_circle_add`. Posteriormente, el script entra en un ciclo `for` que recorre la circunferencia completa (de 0 a 360 grados) en pasos de 60. Dentro de este bucle, ocurre la transformación matemática más importante: la conversión de grados a radianes mediante `math.radians`. Esta conversión es necesaria para que las funciones trigonométricas de la librería `math` de Python puedan procesar los datos. Utilizando nuevamente las fórmulas de posición  $posX = R \cdot \cos(rad)$  y  $posY = R \cdot \sin(rad)$ , el algoritmo calcula los seis puntos exactos que rodean al círculo central.

En cada una de estas seis posiciones calculadas, el script ordena a Blender añadir un nuevo círculo con el mismo radio original. El resultado final, como se observa en tu captura de pantalla de Blender 5.0.1, es una estructura perfectamente simétrica compuesta por **siete círculos**: uno central y seis periféricos que se intersectan en el origen. Esta disposición no solo crea una estética visual compleja, sino que demuestra cómo la repetición de instrucciones simples bajo una lógica matemática puede generar patrones geométricos de gran sofisticación en segundos.

