

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

El script de Python ejecutado en Blender 5.0.1 representa un ejercicio fundamental de geometría procedimental, donde la estructura de un objeto no se modela a mano, sino que se define mediante algoritmos matemáticos y la API bpy. El flujo de trabajo inicia con una fase de preparación crítica en la que el script toma control total de la escena de Blender; mediante las operaciones de selección y borrado global, se eliminan cubos, cámaras o luces preexistentes, estableciendo un "lienzo en blanco" digital que garantiza que los datos generados posteriormente sean los únicos presentes en el entorno de trabajo. Este paso es esencial en el desarrollo de herramientas de automatización para evitar la acumulación de objetos huérfanos en la base de datos del proyecto.

La arquitectura del código se centraliza en la función `crear_poligono_2d`, la cual opera bajo una jerarquía de datos específica: primero crea un contenedor de malla (Mesh), que es la estructura de datos pura, y luego lo envuelve en un objeto de escena (Object), que es la entidad que posee propiedades de transformación como posición y rotación. Una vez establecido este contenedor, el script aplica principios de trigonometría clásica para posicionar cada uno de los seis vértices del hexágono. Utilizando un bucle que recorre el número de lados, se calcula el ángulo de cada punto distribuyendo equitativamente los 360 grados (2π radianes) de una circunferencia. Al multiplicar el coseno y el seno de estos ángulos por el radio definido de 5 unidades, el algoritmo obtiene las coordenadas exactas en el plano cartesiano X e Y , mientras mantiene el eje Z en cero absoluto para asegurar que la figura sea perfectamente plana.

La construcción de la topología es el paso final y más ingenioso del script. No basta con colocar los puntos en el espacio; es necesario indicarles cómo conectarse para formar una silueta. El código genera una lista de "aristas" (edges) que vinculan el índice de un vértice con el siguiente (i con $i+1$). Para resolver el problema del cierre de la figura, se emplea el operador matemático de módulo $\%$, el cual permite que, al llegar al último vértice, el sistema "dé la vuelta" y lo conecte automáticamente con el primero, creando un ciclo cerrado infinito. El resultado visible en tu captura de pantalla es un hexágono de alta precisión técnica, mostrado en modo de estructura de alambre, que sirve como base perfecta para procesos más complejos como la generación de patrones arquitectónicos o la creación de interfaces de usuario tridimensionales dentro de Blender.

