

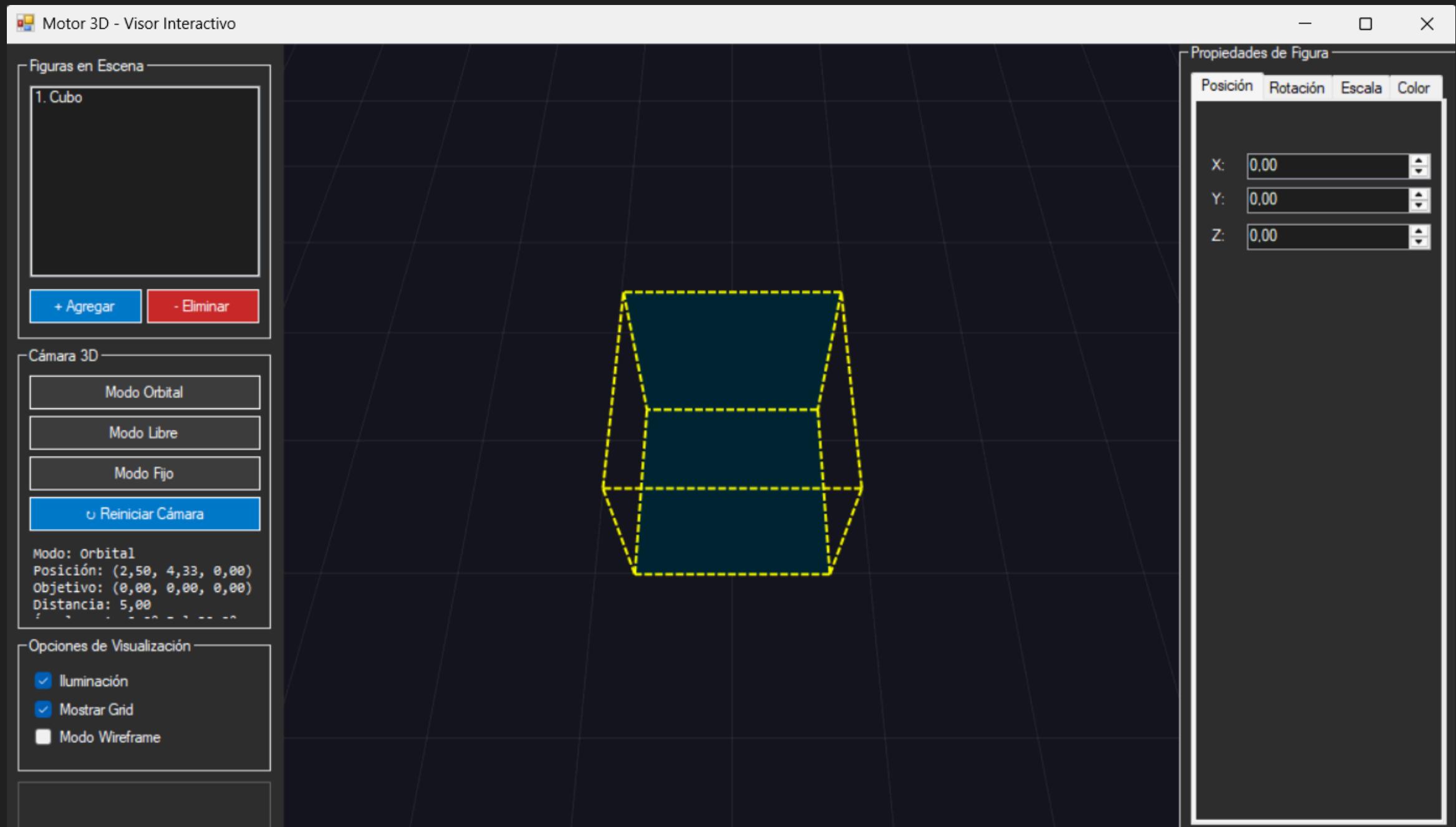
Proyecto Unidad 2

MOTOR GRÁFICO 3D



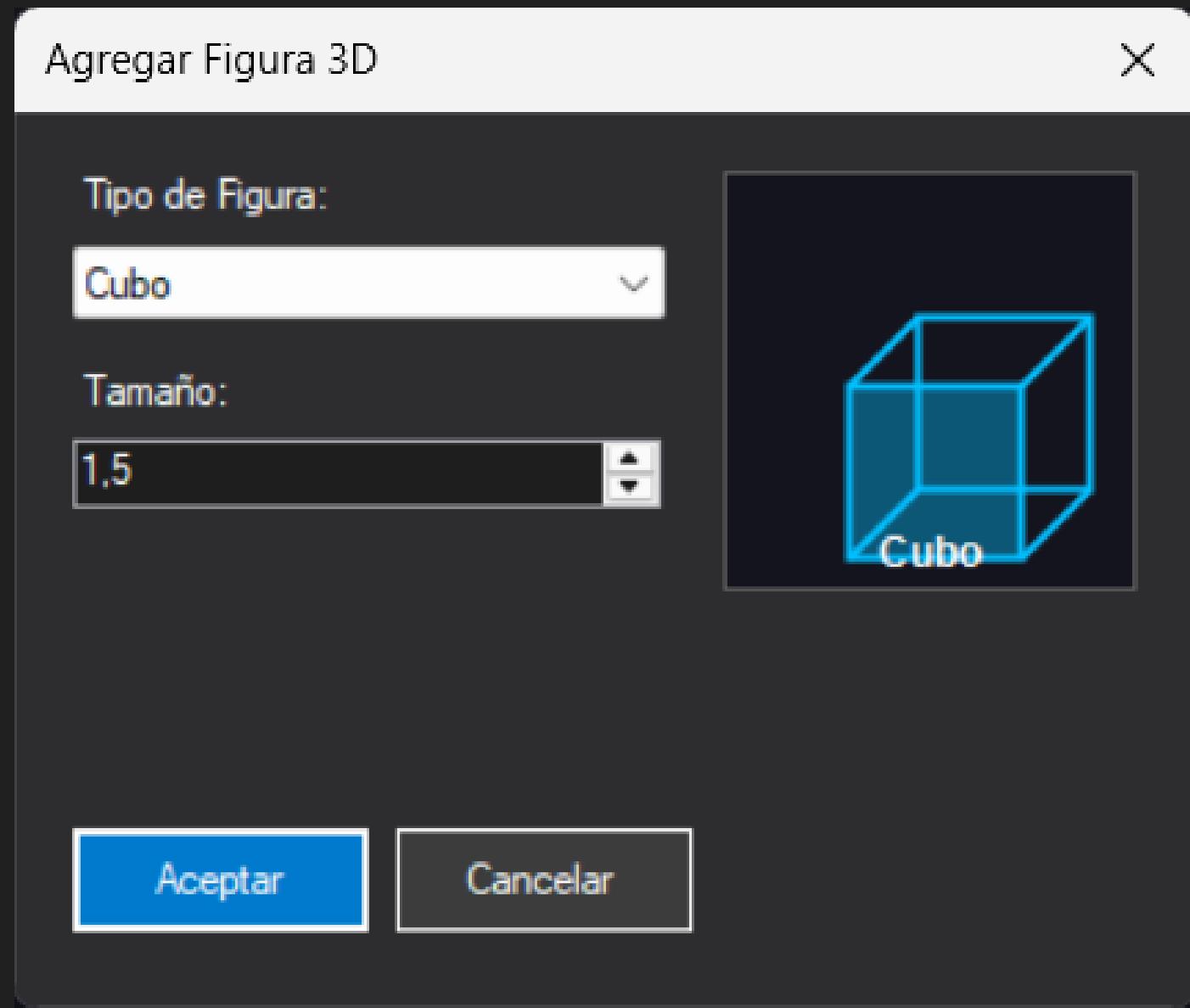
Computación Gráfica

DISEÑO DE LA INTERFAZ



La interfaz del sistema es clara, intuitiva y organizada, facilita la interacción del usuario con el entorno tridimensional. Presenta una distribución lógica de los elementos, separando las áreas de visualización, control y configuración. El uso de una paleta de colores oscuros mejora el contraste y la legibilidad de las figuras 3D.

DISEÑO DE LA INTERFAZ



Fue diseñado como una ventana modal de apoyo al formulario principal, con el objetivo de permitir la creación de nuevas figuras tridimensionales. La organización de los controles sigue una disposición vertical y secuencial, comenzando con la selección del tipo de figura mediante un ComboBox de lista cerrada. Se utilizaron etiquetas descriptivas ubicadas sobre cada control.

FUNCIONALIDADES CLAVE

- Figuras: cubo, esfera, cilindro, cono, pirámide, toroide.
- Transformaciones: translación, rotación, escalamiento en los 3 ejes (X,Y,Z)
- Propiedades visuales: cambio de color + iluminación básica
- Interacción y cambios en tiempo real de las figuras.

01

BASE MATEMÁTICA

01/ BASE MATEMÁTICA

Para que el motor funcione, se implementan operaciones fundamentales de álgebra lineal. Los vectores 3D permiten representar posiciones y direcciones, y se usan en normalización, producto punto (ángulos/iluminación) y producto cruz (normales).

$$\mathbf{v} = (x, y, z)$$

Las transformaciones 3D se modelan con matrices 4×4 y coordenadas homogéneas, porque así se pueden combinar rotación, escala y traslación mediante multiplicación matricial en un solo pipeline de cálculo.

$$\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



01/

BASE MATEMÁTICA

Rotación

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

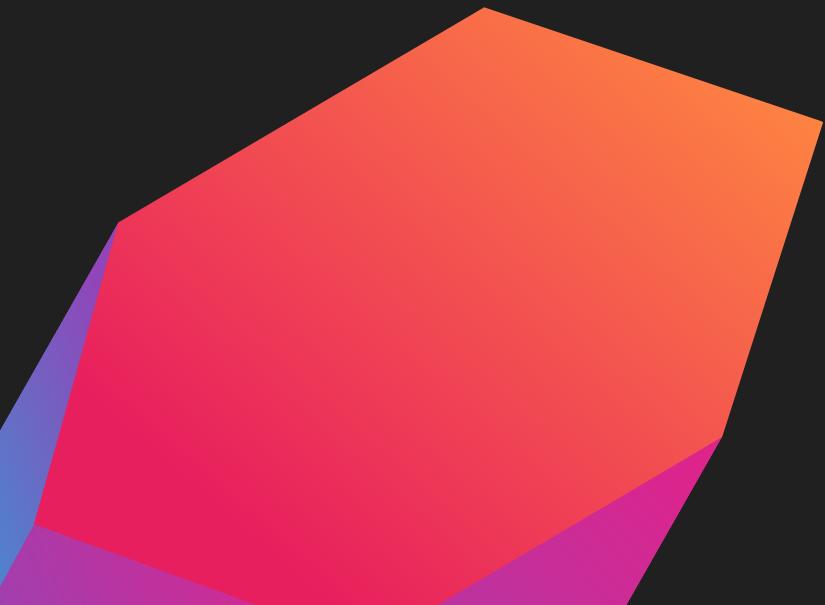
$$\begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & 0 & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Traslación

$$\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Escalamiento

$$\begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



02

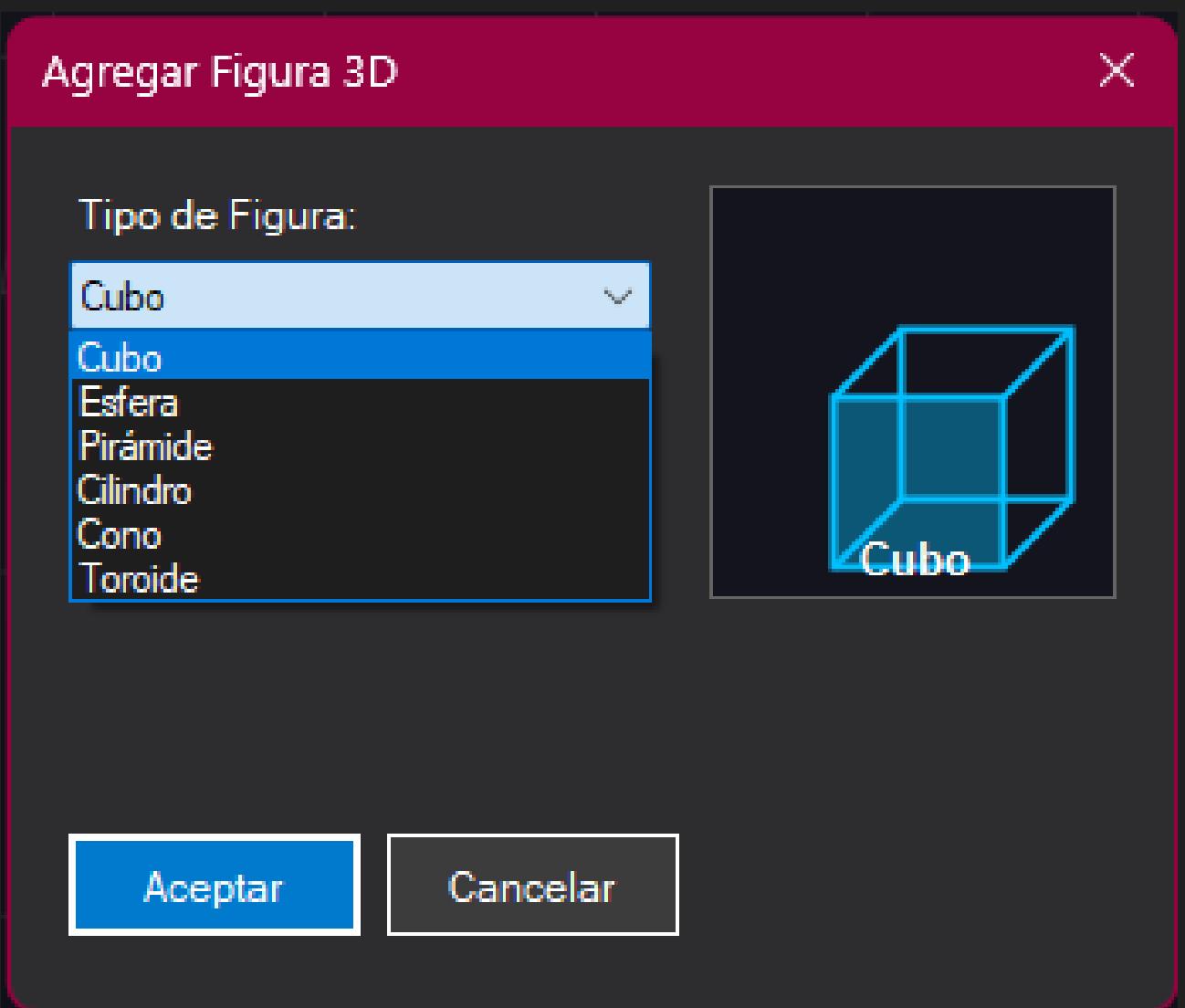
GENERACIÓN DE FIGURAS

02/ GENERACIÓN DE FIGURAS



La geometría se crea con una “Figuras3DFactory” que retorna objetos con sus vértices, caras y propiedades listas para renderizar.

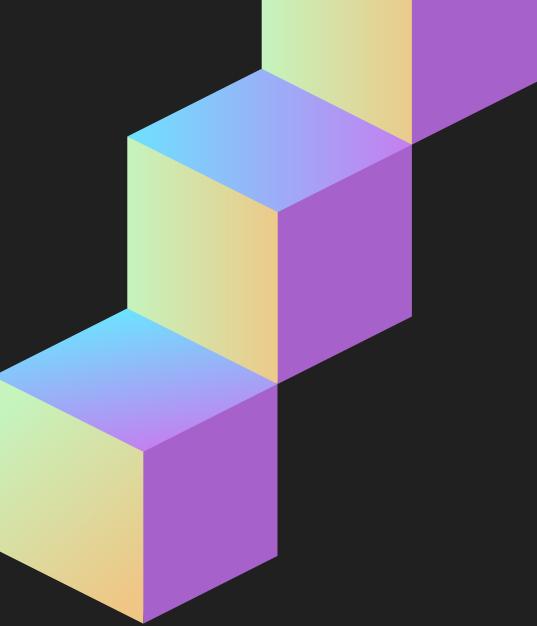
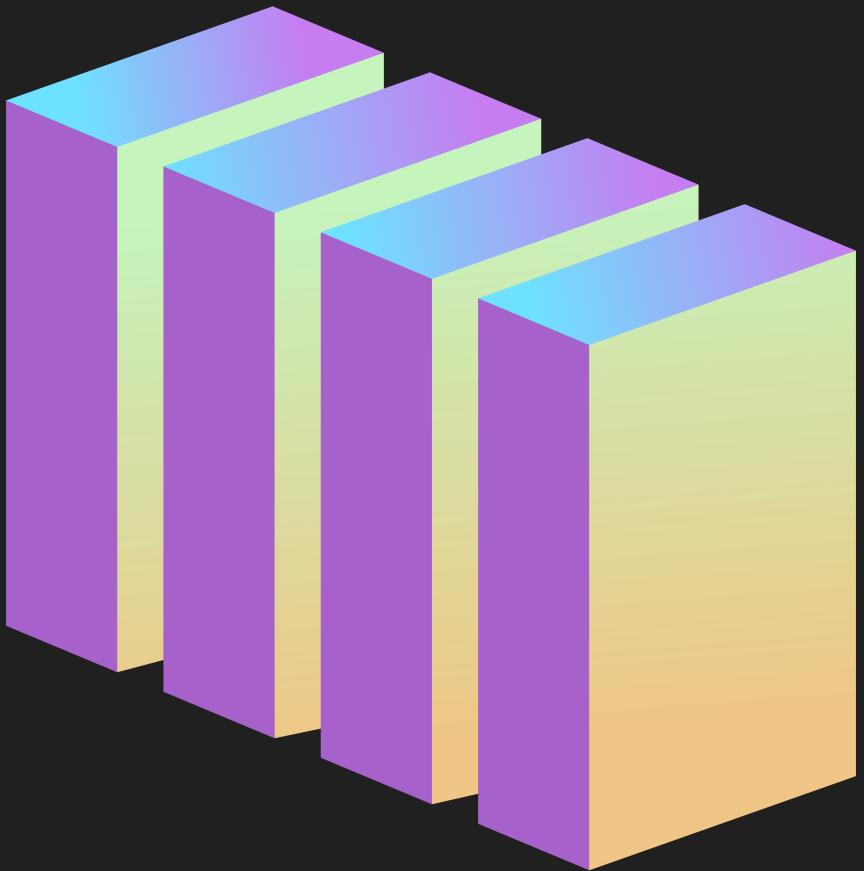
Se incluyen primitivas como cubo, esfera, cilindro, cono, pirámide y toroide, variando la complejidad según la cantidad de segmentos/subdivisiones usadas.



03

ARQUITECTURA

03/ ARQUITECTURA



El motor está organizado en tres bloques principales:

1. Motor3D: contiene operaciones matemáticas, transformaciones y proyección.
2. Figuras3DFactory: genera las figuras mediante listas
3. Camara3D: controla la vista y la proyección según el modo de cámara.

Figura

Transformaciones

Proyección

Rasterización



04

PIPELINE GRÁFICO

A large, semi-transparent 3D geometric model is positioned on the left side of the slide. It features a purple rectangular base, a yellow square middle section with a purple diamond-shaped cutout, and a pink and purple pyramid-like top. A smaller, similar geometric model is also visible in the bottom right corner.

04/ PIPELINE GRÁFICO

El pipeline transforma figuras en coordenadas locales a imágenes 2D en pantalla.

Etapas:

- Espacio objeto: definición de geometría local (vértices, caras).
- Espacio mundo: transformación a la escena global (R,T,E).
- Espacio vista (cámara): Se transforma la escena según la posición y orientación de la cámara.
- Espacio clip: Se aplica la proyección perspectiva, convirtiendo coordenadas 3D en coordenadas homogéneas.

05

ILUMINACIÓN

05/ ILUMINACIÓN

La iluminación trabaja con el ángulo que existe entre:

- N = normal de la cara (hacia dónde “apunta” la cara)
- L = dirección de la luz

Y ese ángulo se calcula con producto punto.

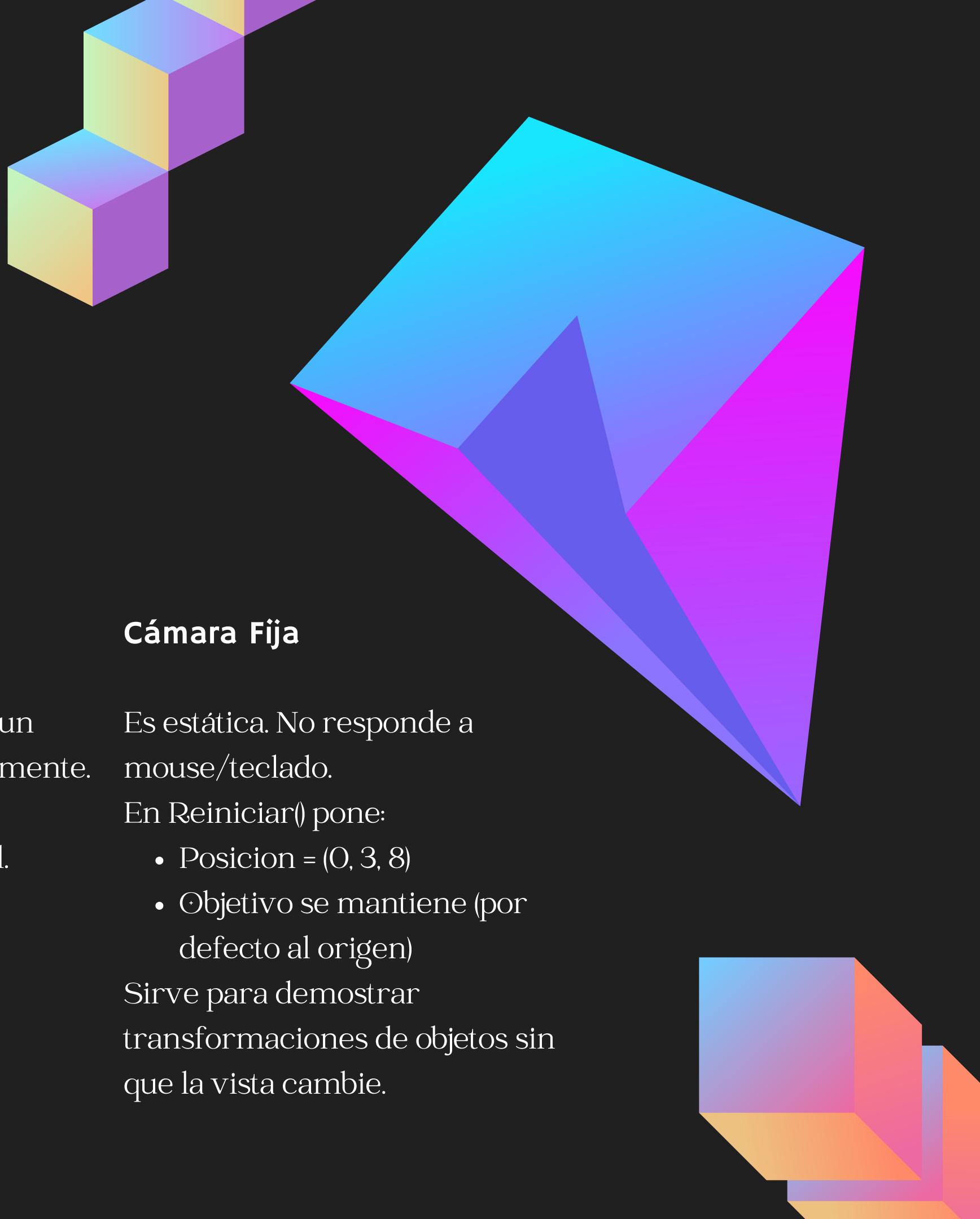
En este trabajo se implementó el modelo de iluminación Phong Simplificado que trabaja con 2 componentes (sin componente especular):

- Componente ambiental (IA): es una aproximación de la luz indirecta (rebotes de luz en el entorno). Ilumina igual a todas las caras sin importar el ángulo.
- Componente difusa (Ley de Lambert – Id): depende de qué tan alineados estén la normal y la luz. Modela la “luz directa” que llega desde una fuente y pega en la superficie.



06
CÁMARA

06/ CÁMARA



Una Cámara tiene las siguientes propiedades:

- Posición (Posicion): dónde está la cámara en el espacio.
- Objetivo (Objetivo): a qué punto está mirando.
- Arriba (Arriba): qué dirección considera “arriba” (sirve para que no se voltee la imagen).

Cámara Orbital (la típica de modeladores 3D)

La cámara gira alrededor de un punto fijo: el objetivo (normalmente el origen).

Variables clave

- Distancia: qué tan lejos estás del objetivo.
- AnguloAzimutal: giro horizontal (izquierda/derecha).
- AnguloPolar: giro vertical (arriba/abajo).

Cámara Libre (tipo FPS)

La cámara se comporta como un jugador: se mueve y mira libremente.

Variables clave

- Desviación: giro horizontal.
- Tono: giro vertical.
- SensMouse: sensibilidad.

Cámara Fija

Es estática. No responde a mouse/teclado.

En Reiniciar() pone:

- Posicion = (0, 3, 8)
- Objeto se mantiene (por defecto al origen)

Sirve para demostrar transformaciones de objetos sin que la vista cambie.



Universidad de las Fuerzas Armadas “ESPE”



Computación Gráfica