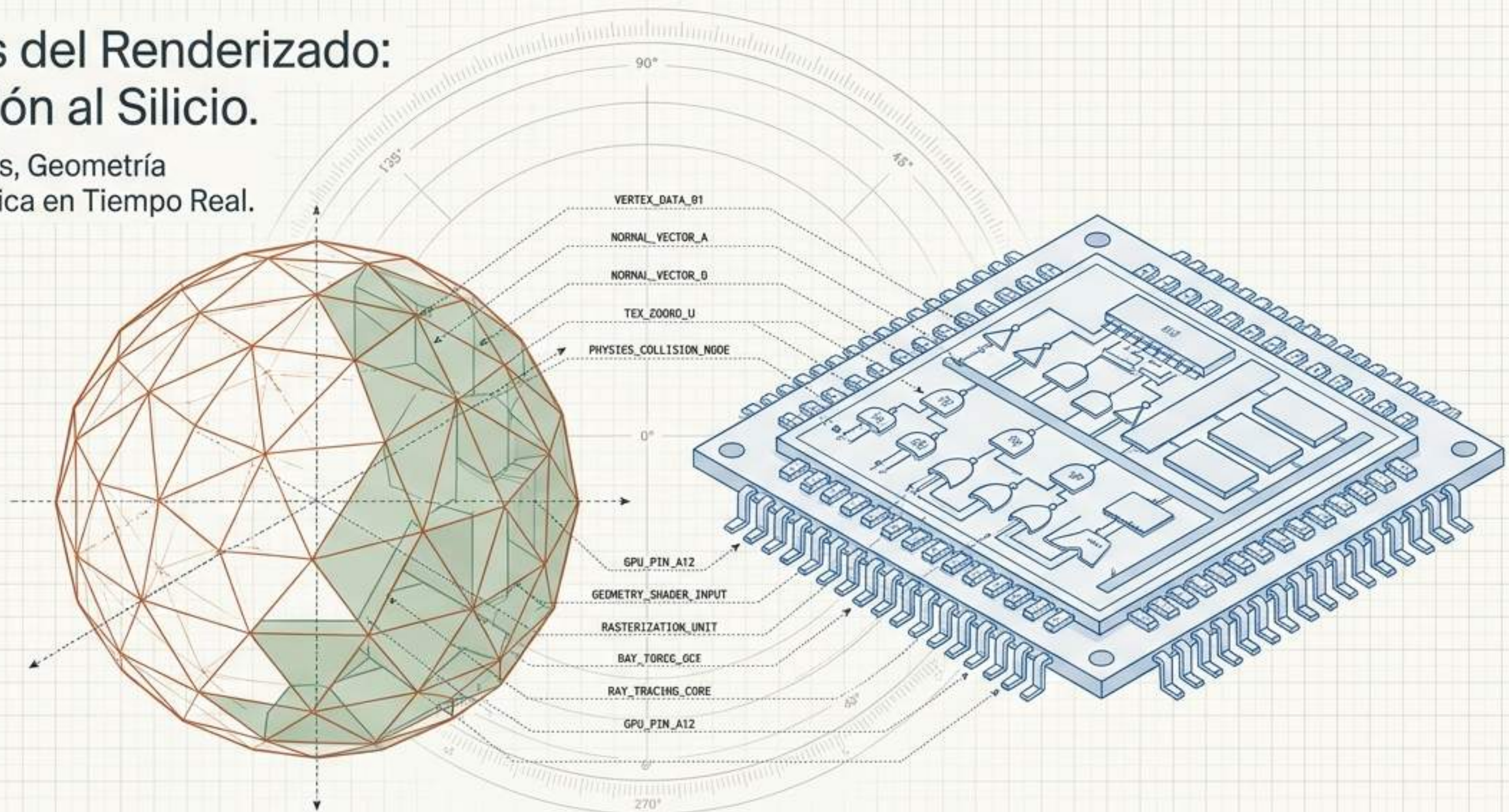


# La Simbiosis del Renderizado: De la Ecuación al Silicio.

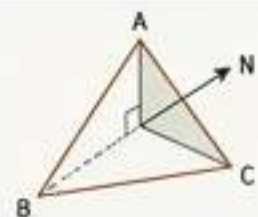
Arquitectura de GPUs, Geometría  
Computacional y Física en Tiempo Real.



## ECUACIÓN Y GEOMETRÍA (Lado Izquierdo)

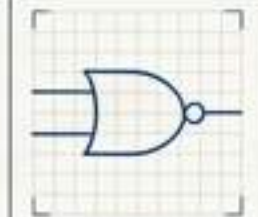
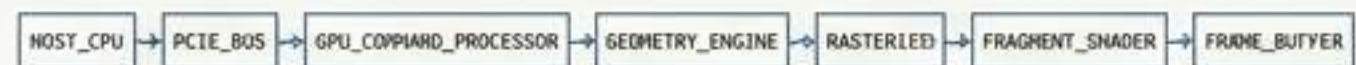
Representación matemática abstracta de objetos 3D mediante vértices, aristas y caras. Los volúmenes de colisión y vectores definen la física y la superficie.

$$V = \sum (v_i * w_i); N = \text{normalize}(\text{cross}(B-A, C-A))$$



## SILICIO Y ARQUITECTURA (Lado Derecho)

Implementación física en hardware de procesamiento gráfico. Los circuitos integrados y núcleos especializados ejecutan operaciones aritméticas en paralelo.

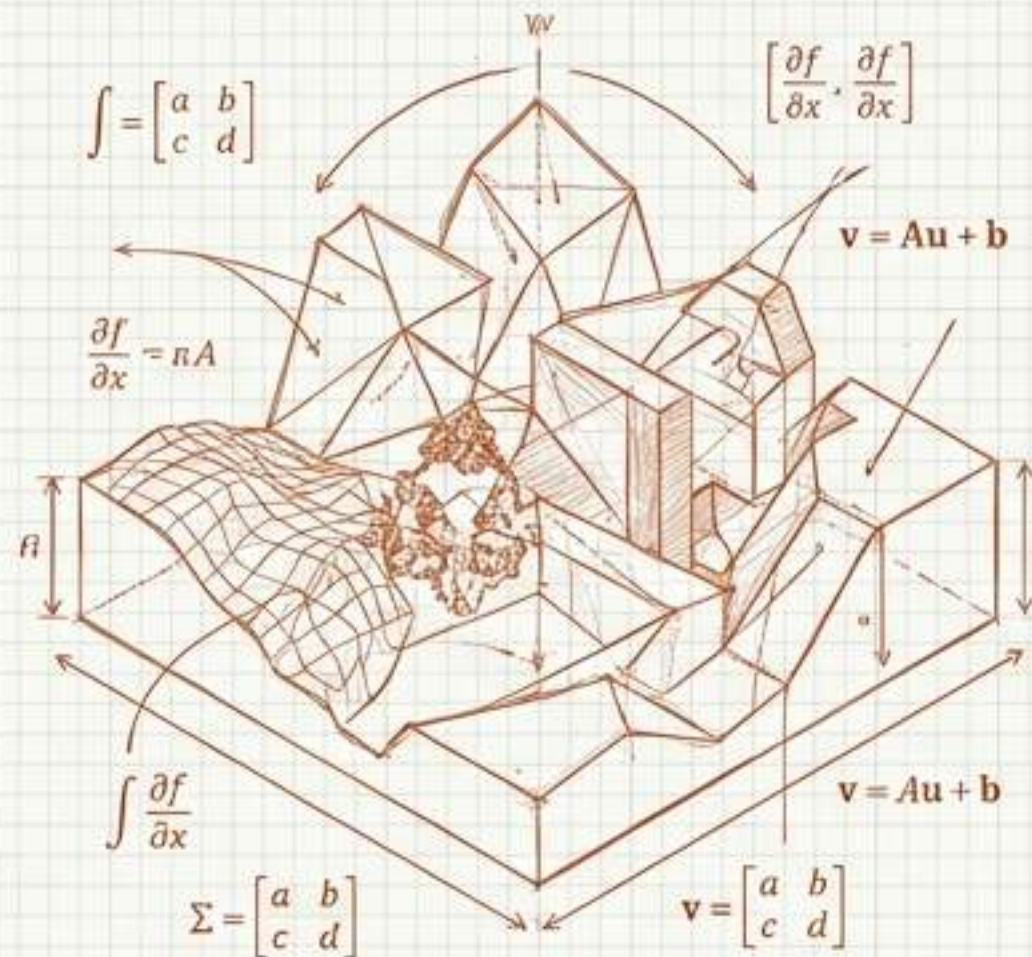


# Los mundos virtuales exigen un matrimonio perfecto entre la teoría matemática y el diseño de silicio.

La creación de gráficos a 60 FPS no es producto de la fuerza bruta.

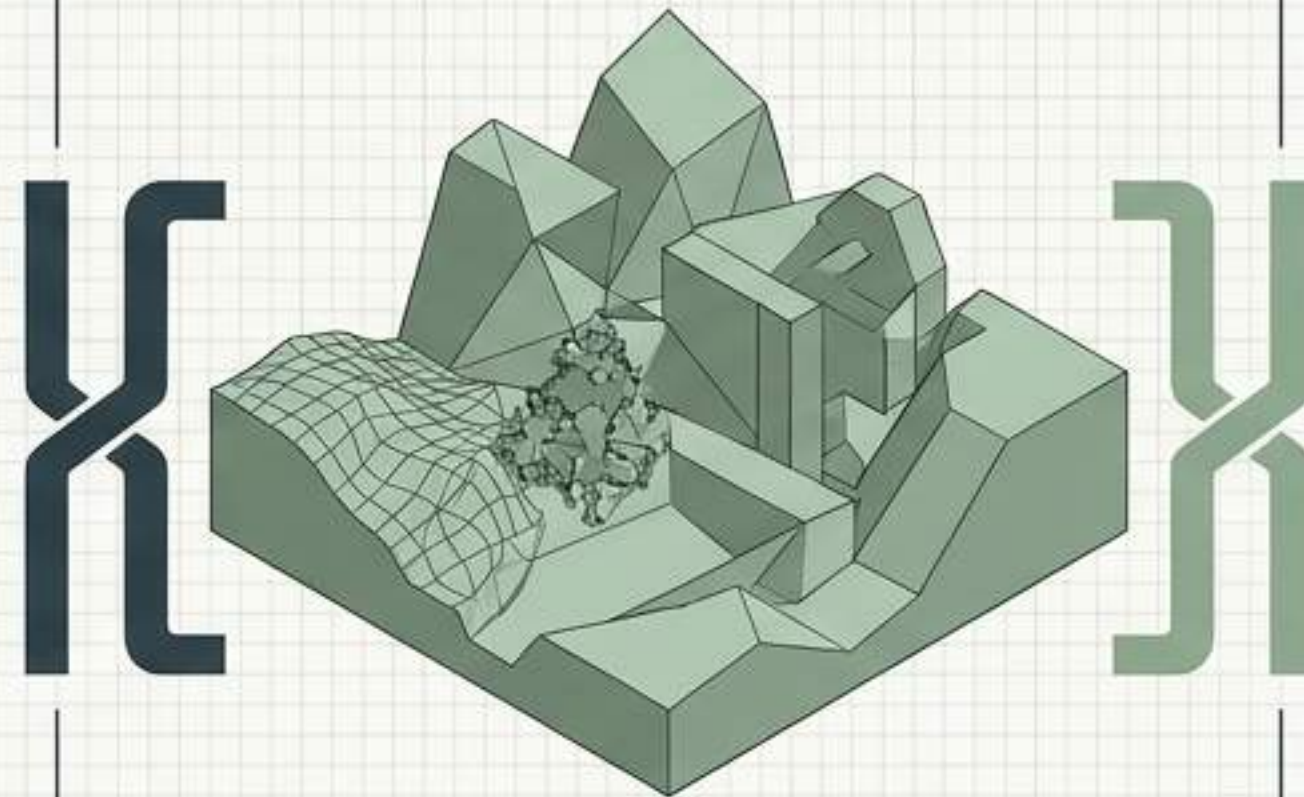
## El Modelo

Las matemáticas proveen las reglas físicas y estructurales



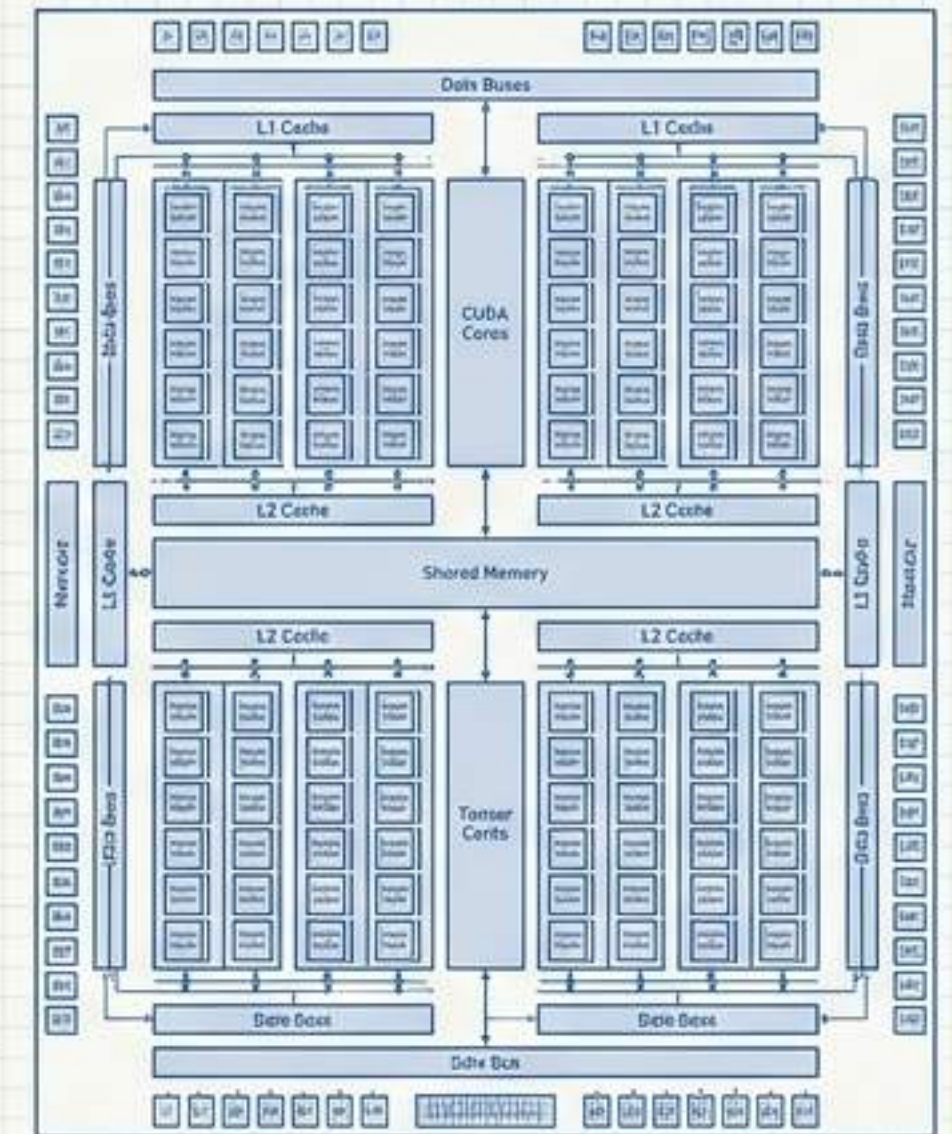
## La Traducción

El álgebra lineal convierte el modelo en un lenguaje calculable



## El Motor

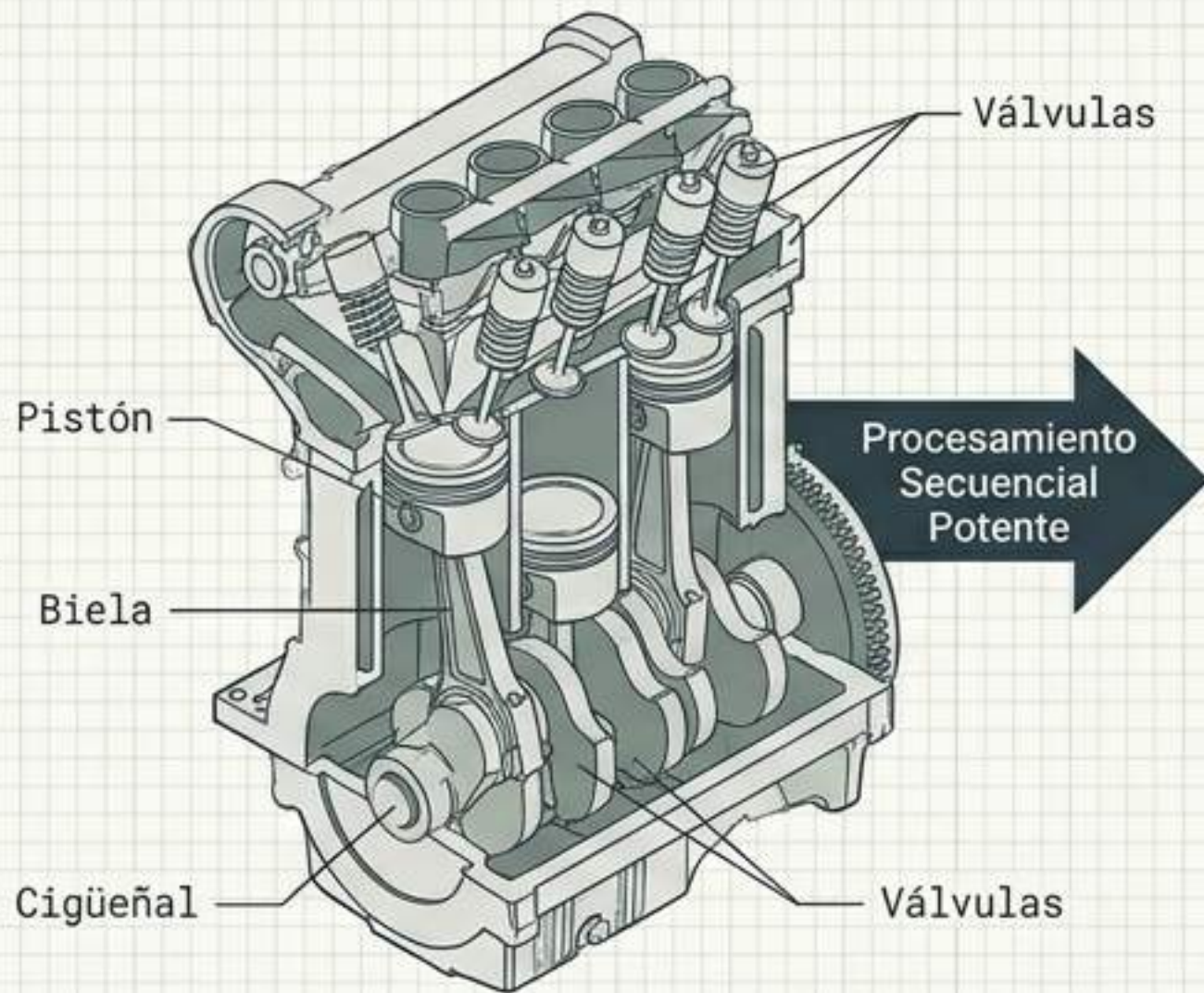
Las GPUs ofrecen silicio diseñado para operaciones matriciales paralelas



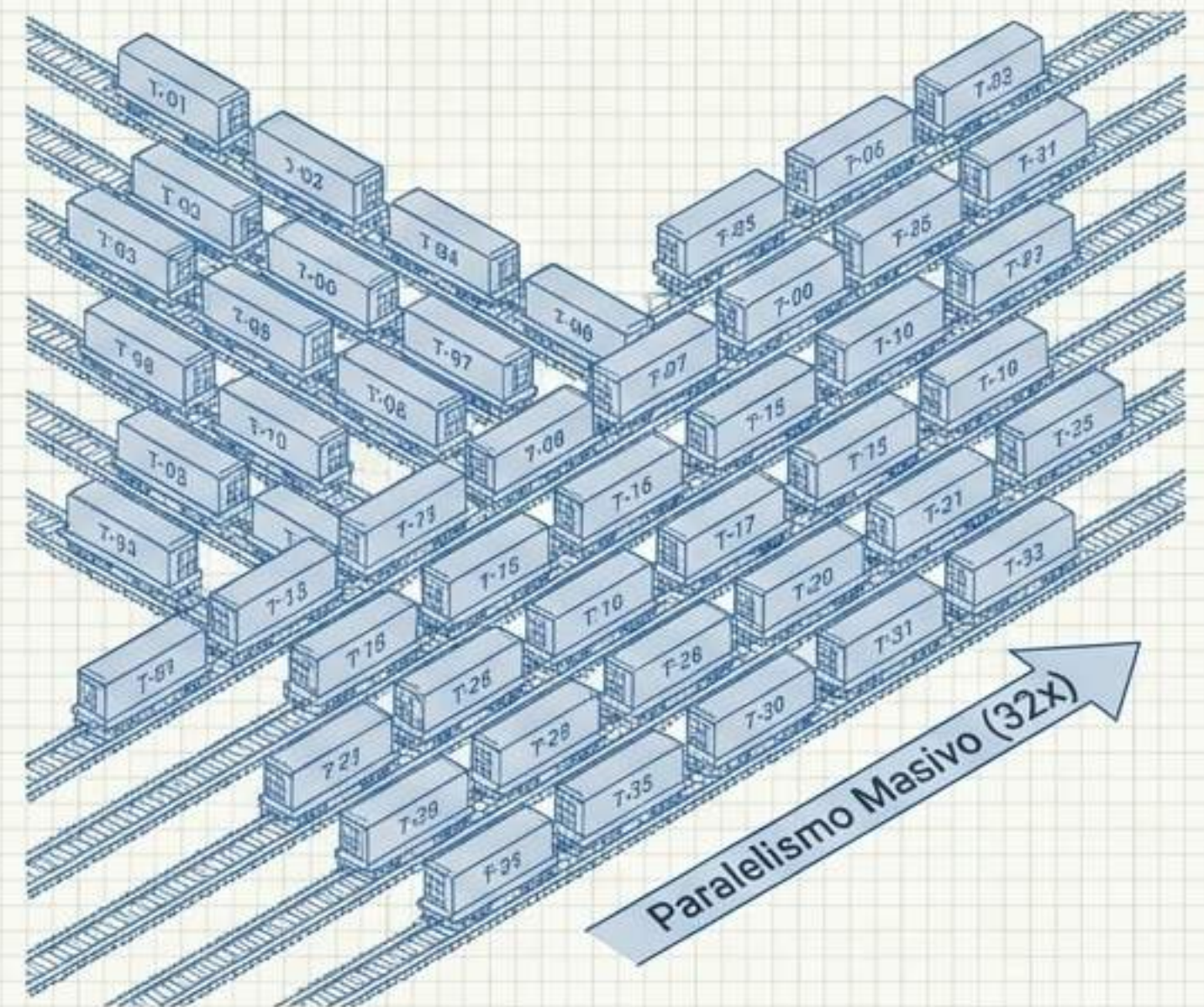
# El paradigma heterogéneo delega el paralelismo masivo a procesadores especializados.

**Host (CPU + System Memory)**  
Inicia aplicaciones.  
Gestiona lógica secuencial de alto nivel.

**Device (GPU + VRAM)**  
Ejecuta tareas computacionalmente intensivas.  
Renderiza millones de polígonos.



Bus PCIe



# SIMT y la arquitectura del cómputo paralelo en grupos indivisibles.

## Hilo (Thread):

La unidad mínima de ejecución.

## Warp:

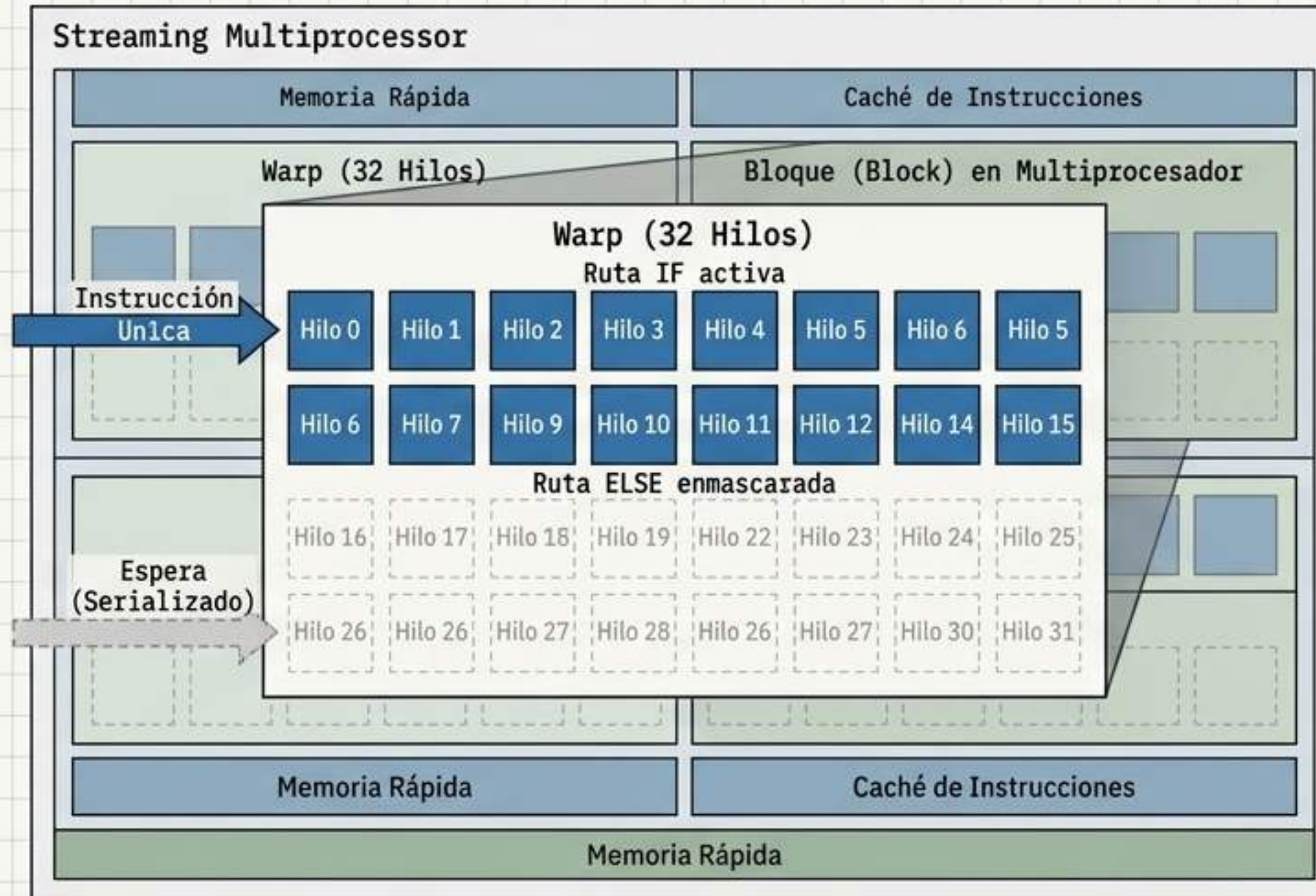
Grupo de 32 hilos que ejecutan la misma instrucción simultáneamente.

## Bloque (Block):

Agrupación de Warps en el mismo multiprocesador que comparten memoria rápida.

## Grid:


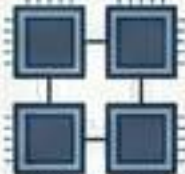
El conjunto total de bloques lanzados por un Kernel.



## Divergencia de Warp:

Las ramas condicionales (if/else) fuerzan a la GPU a serializar la ejecución, reduciendo la eficiencia a la mitad.

# El duelo arquitectónico: Comparativa de filosofías de diseño.

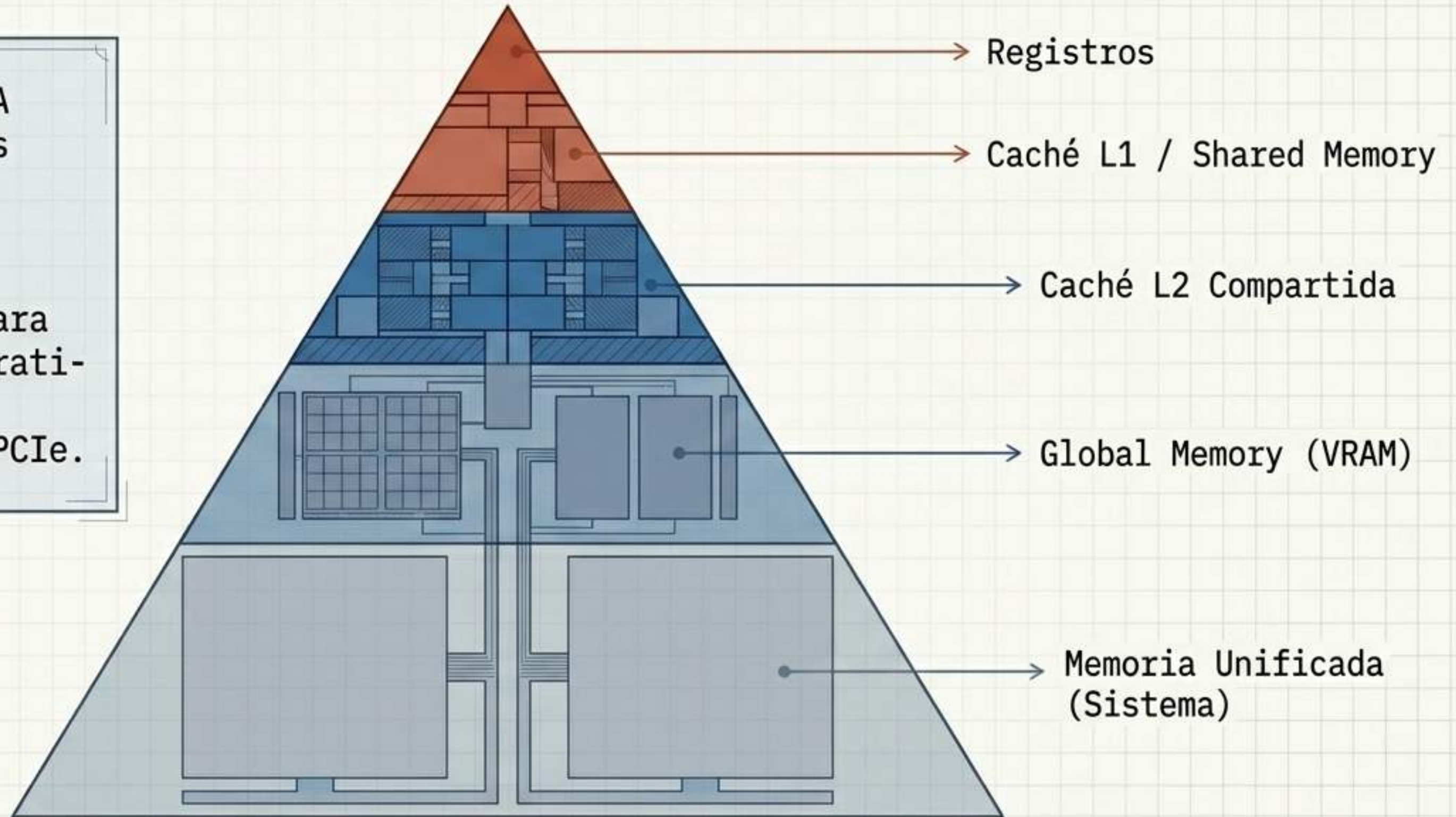
	NVIDIA (Ada Lovelace)	AMD (RDNA 3)
Unidad de Ejecución (ALU)	CUDA Core	Stream Processor
Unidad de Agrupamiento	Warp (32 hilos)	Wavefront (64 o 32 hilos)
Paradigma de Diseño	 Silicio Monolítico	 Chiplets Separados
Estrategia de Caché	Caché L2 Masiva Unificada	Infinity Cache en Módulos

## Insight:

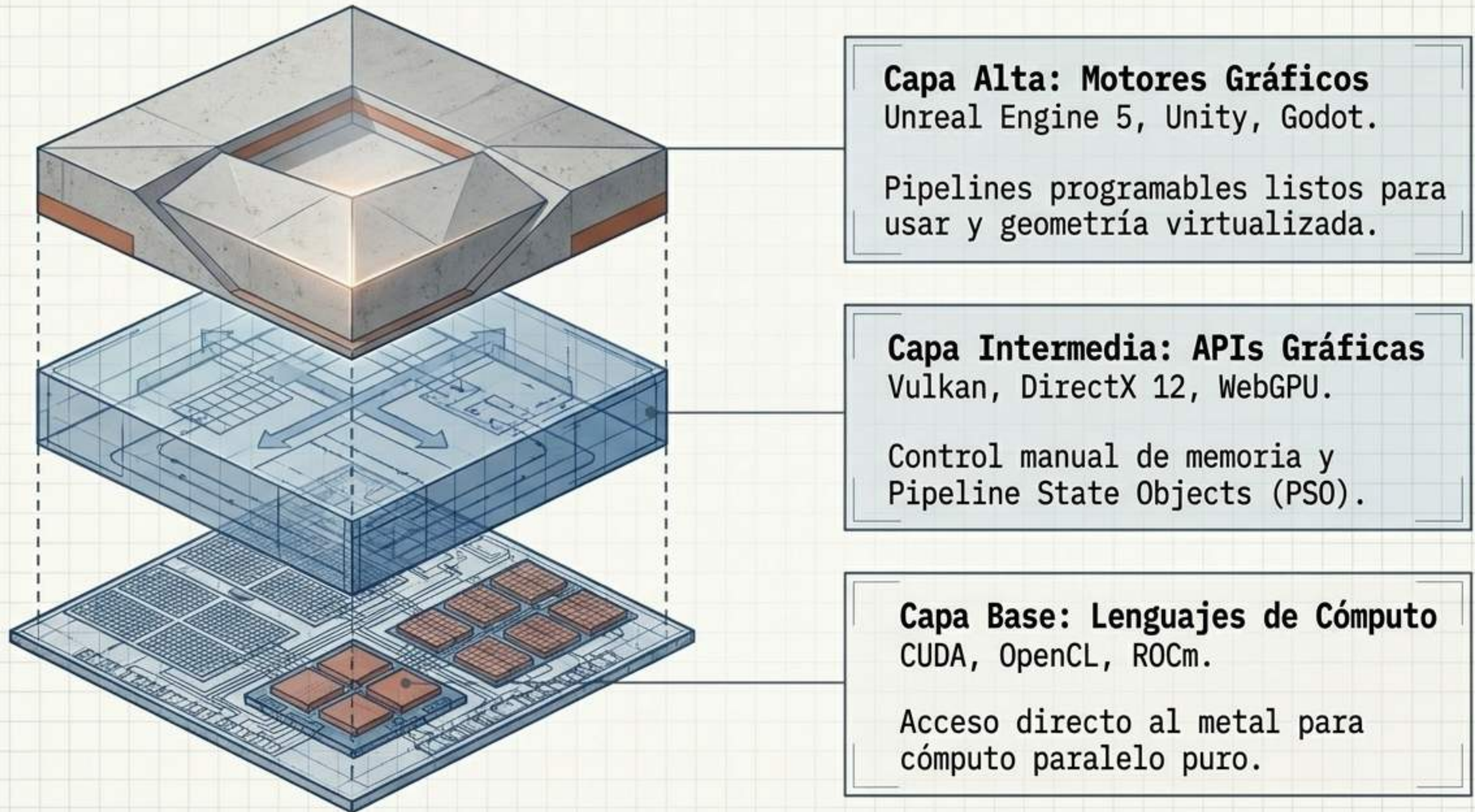
Más núcleos no siempre significa más velocidad. La optimización de la divergencia, el **ancho de banda** y la jerarquía de **caché** dictan el rendimiento real.

# La jerarquía de memoria dicta la velocidad de procesamiento.

El rendimiento AAA exige cargar datos manualmente en la Shared Memory ultrarrápida del microprocesador para operaciones cooperativas, ocultando la latencia del bus PCIe.

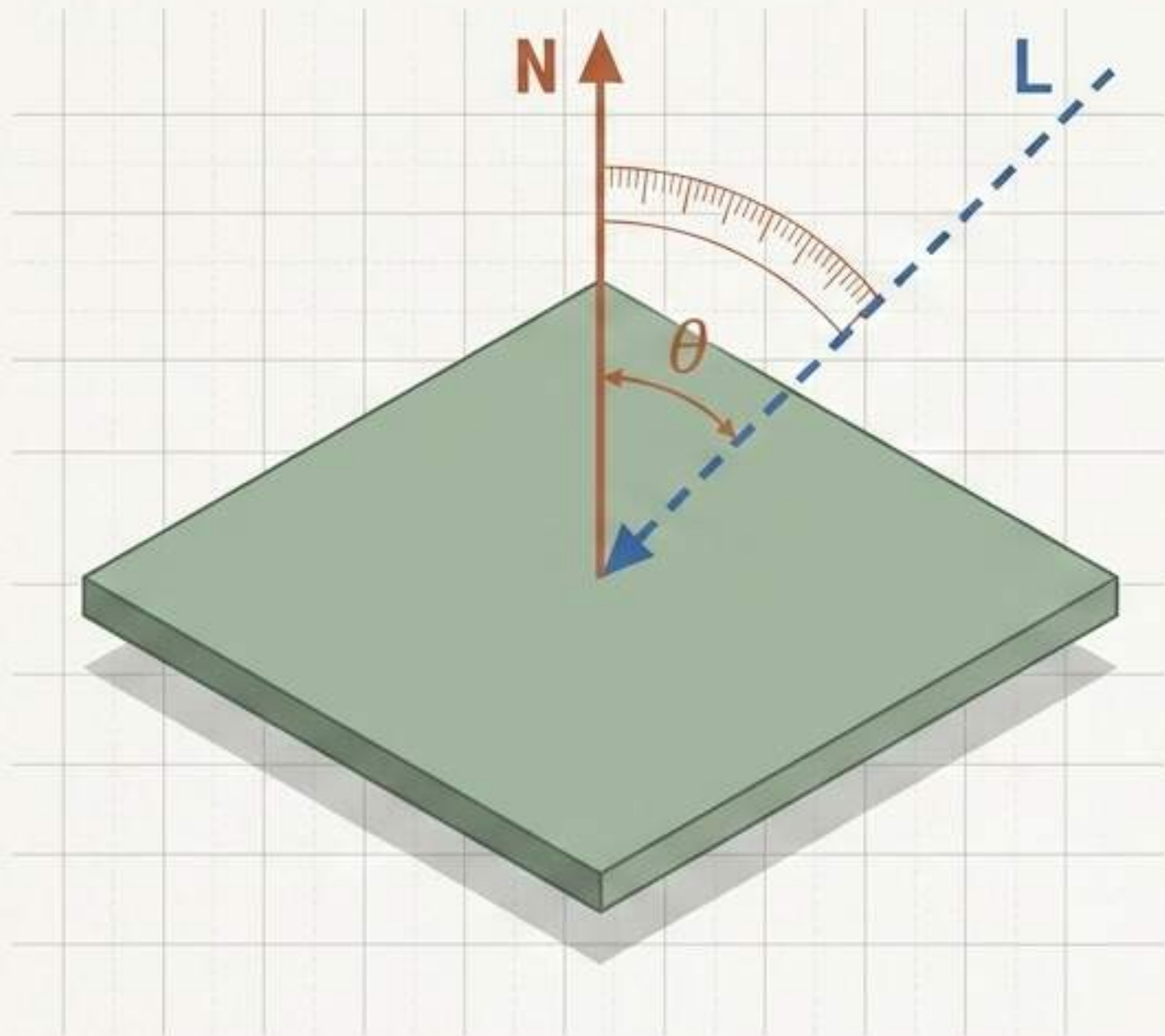


# Estratos de control: La pirámide de abstracción de software.



# El lenguaje de la luz: El Producto Interno en el Hardware

La iluminación de Lambert se calcula instantáneamente determinando el ángulo entre la normal de la superficie y la fuente de luz.



$$\mathbf{A} \cdot \mathbf{B} = x_1x_2 + y_1y_2 + z_1z_2 = |\mathbf{A}||\mathbf{B}| \cos(\theta)$$

Si  $\mathbf{N} \cdot \mathbf{L} = 1$ : Luz directa, brillo máximo.

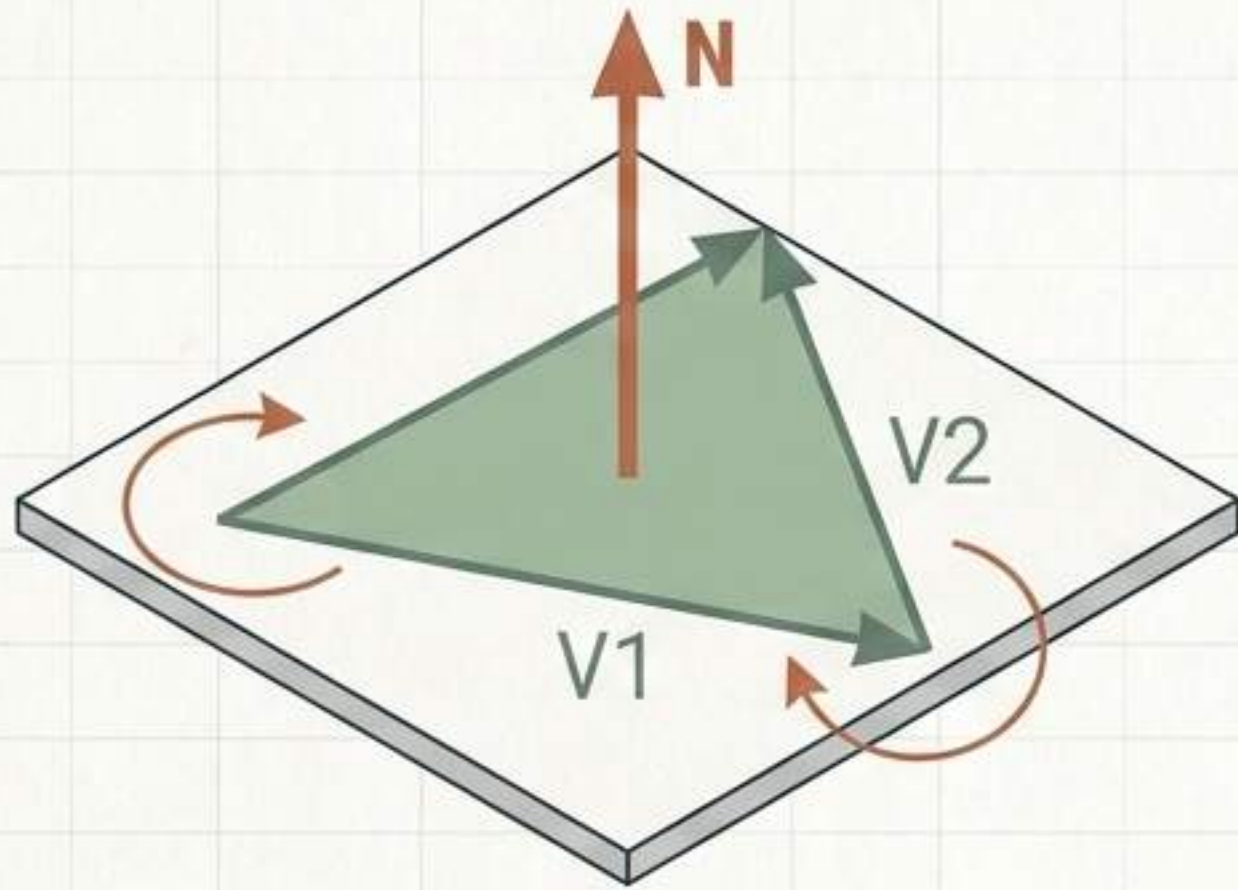
Si  $\mathbf{N} \cdot \mathbf{L} = 0$ : Luz rasante.

Si  $\mathbf{N} \cdot \mathbf{L} < 0$ : Superficie en la sombra.

## Hardware:

Ejecutado millones de veces por segundo vía instrucciones FMA (Fused Multiply-Add).

# Estructura y orientación: Definiendo la cara visible.



## Cálculo de Normales

El Producto Vectorial de dos aristas nos da la orientación exacta en el espacio 3D.

$$\mathbf{N} = \mathbf{V1} \times \mathbf{V2} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ v1x & v1y & v1z \\ v2x & v2y & v2z \end{vmatrix}$$

## Backface Culling

Evaluar el giro determina instantáneamente si una cara mira hacia la cámara. Descartar triángulos ocultos ahorra hasta un 50% de los ciclos de GPU.

# La cuarta dimensión unifica las transformaciones

## El Problema:

Rotar y escalar usaba multiplicaciones matriciales puras. Trasladar requería sumas separadas, rompiendo el pipeline de hardware.

## La Solución:

Añadir una coordenada ficticia "w" (Coordenadas Homogéneas) convierte la traslación en una transformación afín lineal.

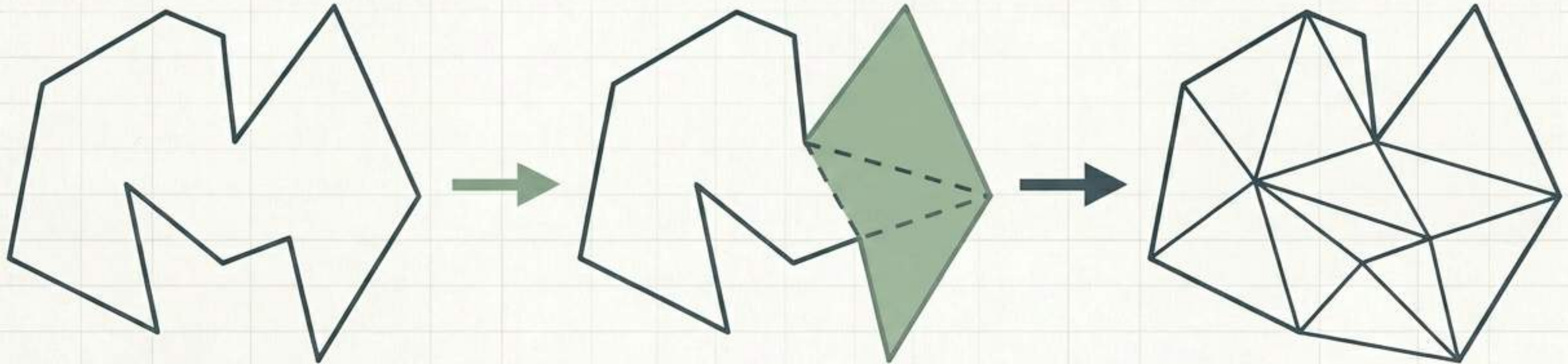
## Resultado:

Las GPUs procesan matrices masivas, acumulando todas las operaciones en un flujo único.

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \rightarrow \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix}$$

# Triangulación: El idioma nativo de la rasterización.

---



## ¿Por qué Triángulos?

1. Planaridad garantizada (3 puntos siempre forman un plano).
2. Convexidad absoluta.
3. Interpolación baricéntrica perfecta.

## Ear Clipping:

Busca triángulos en los bordes ("orejas") convexos y vacíos, recortándolos iterativamente.

## Delaunay:

Maximiza ángulos mínimos para evitar triángulos astillados (*slivers*) en mallas físicas.

# El truco del espacio elipsoide simplifica la física.

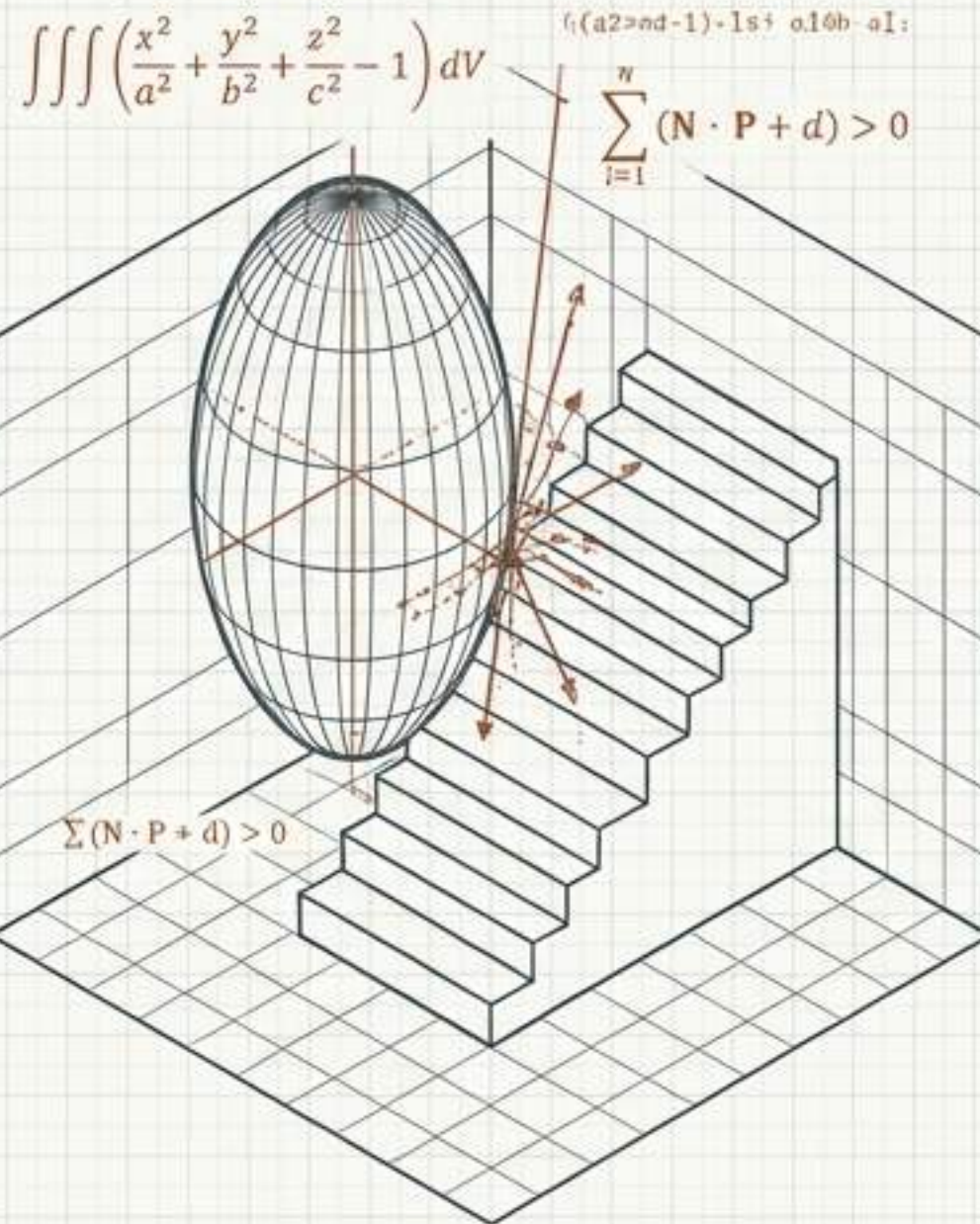
## Problema: Colisiones Complejas

Calcular la intersección de un elipsoide libre contra infinitos polígonos es matemáticamente inestable y costoso.

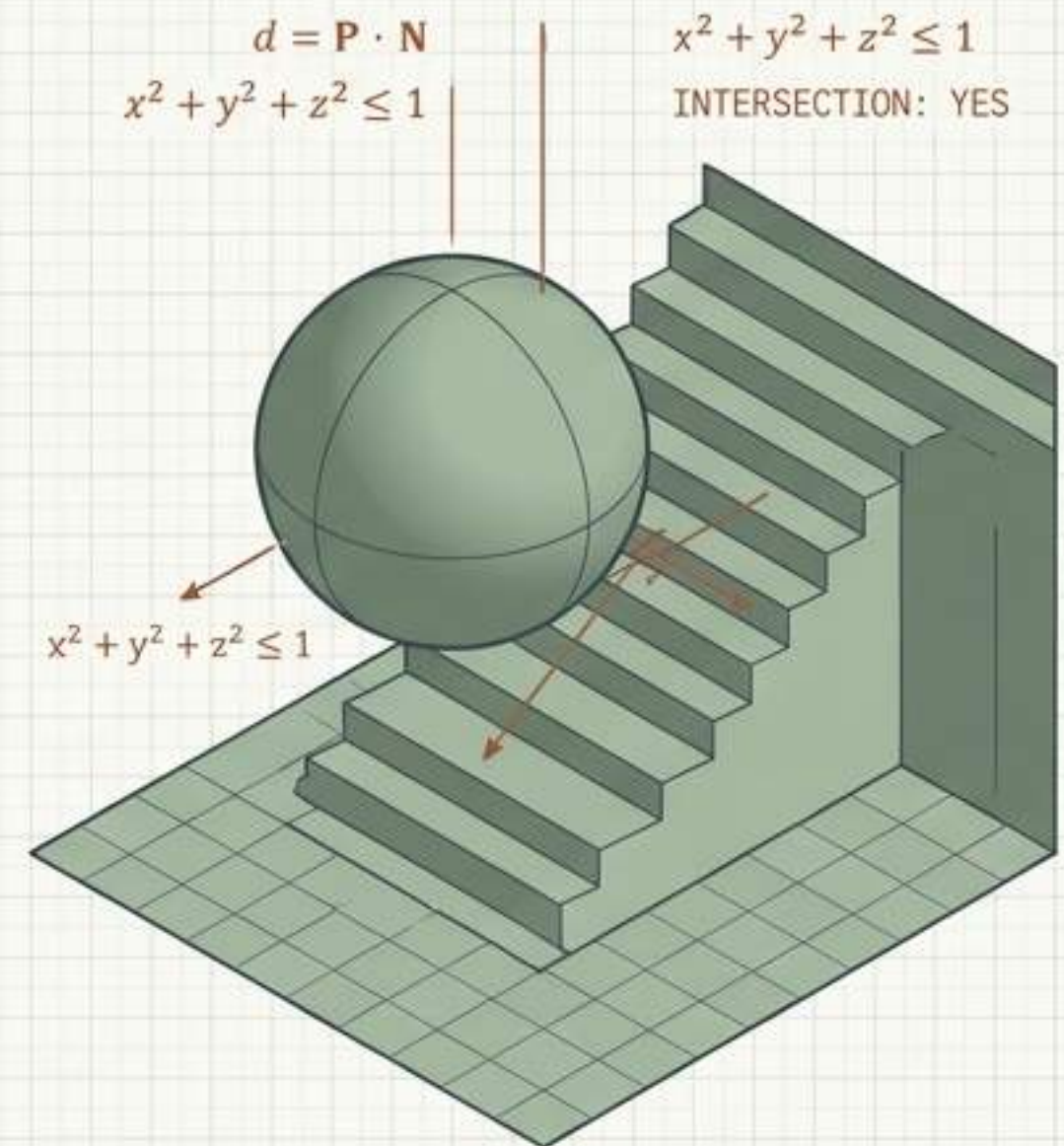
## Solución: Espacio Elipsoide

En lugar de deformar al jugador, deformamos el mundo.  
Escalar el espacio transforma al jugador en una Esfera Unitaria (Radio 1).  
Computar la intersección de un plano con una esfera unitaria es matemáticamente trivial.

PROBLEMA: ESPACIO FÍSICO



SOLUCIÓN: ESPACIO ELIPSOIDE



# Mecánica de intersección: Validación Punto-Triángulo.

## 1. Intersección con el Plano

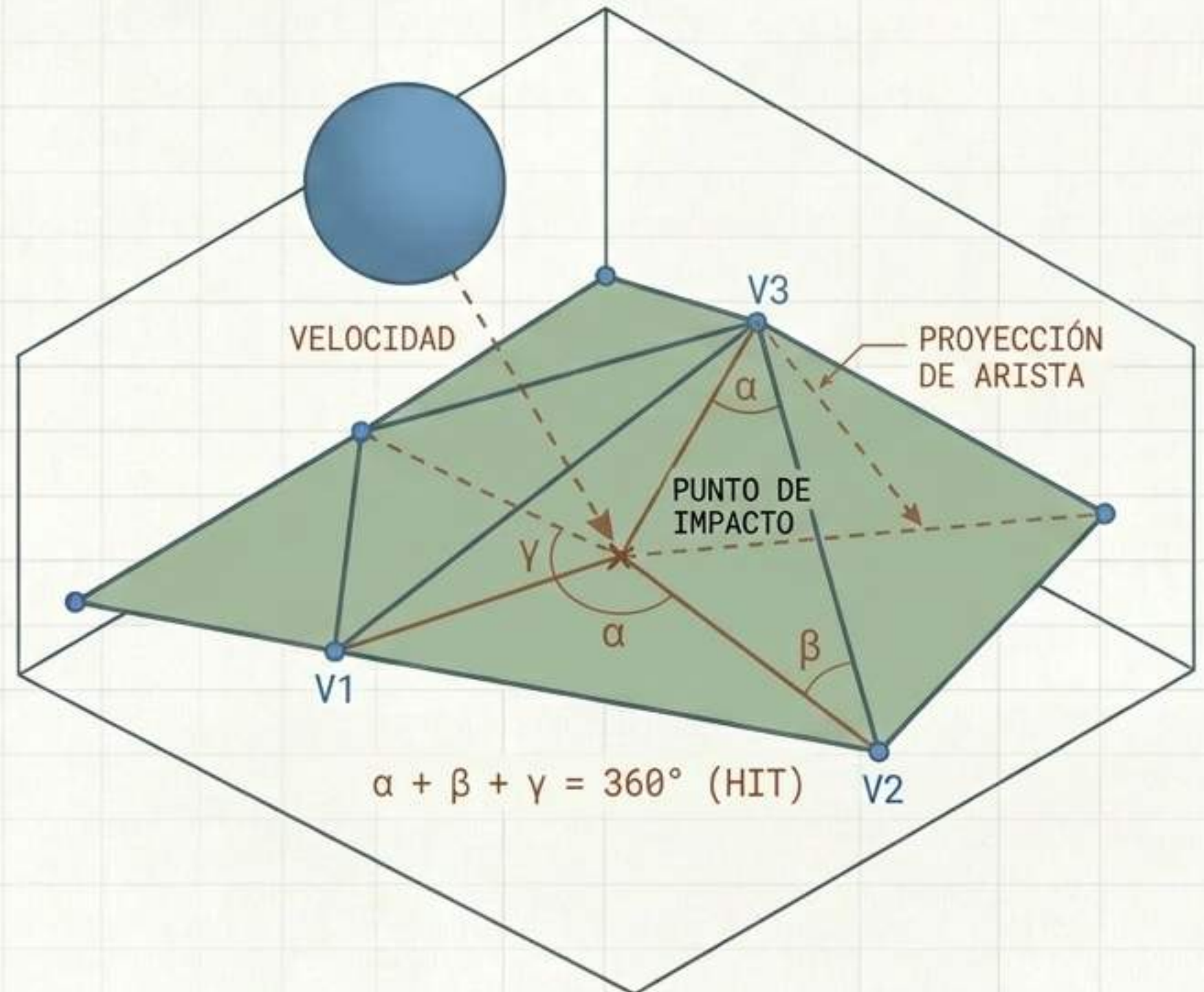
Lanzar un rayo a lo largo de la velocidad para encontrar la coordenada exacta de cruce con el plano infinito.

## 2. Prueba de Inclusión

Trazar vectores desde el impacto hacia los vértices. Si la suma de los ángulos interiores es  $360^\circ$ , es un 'hit' directo.

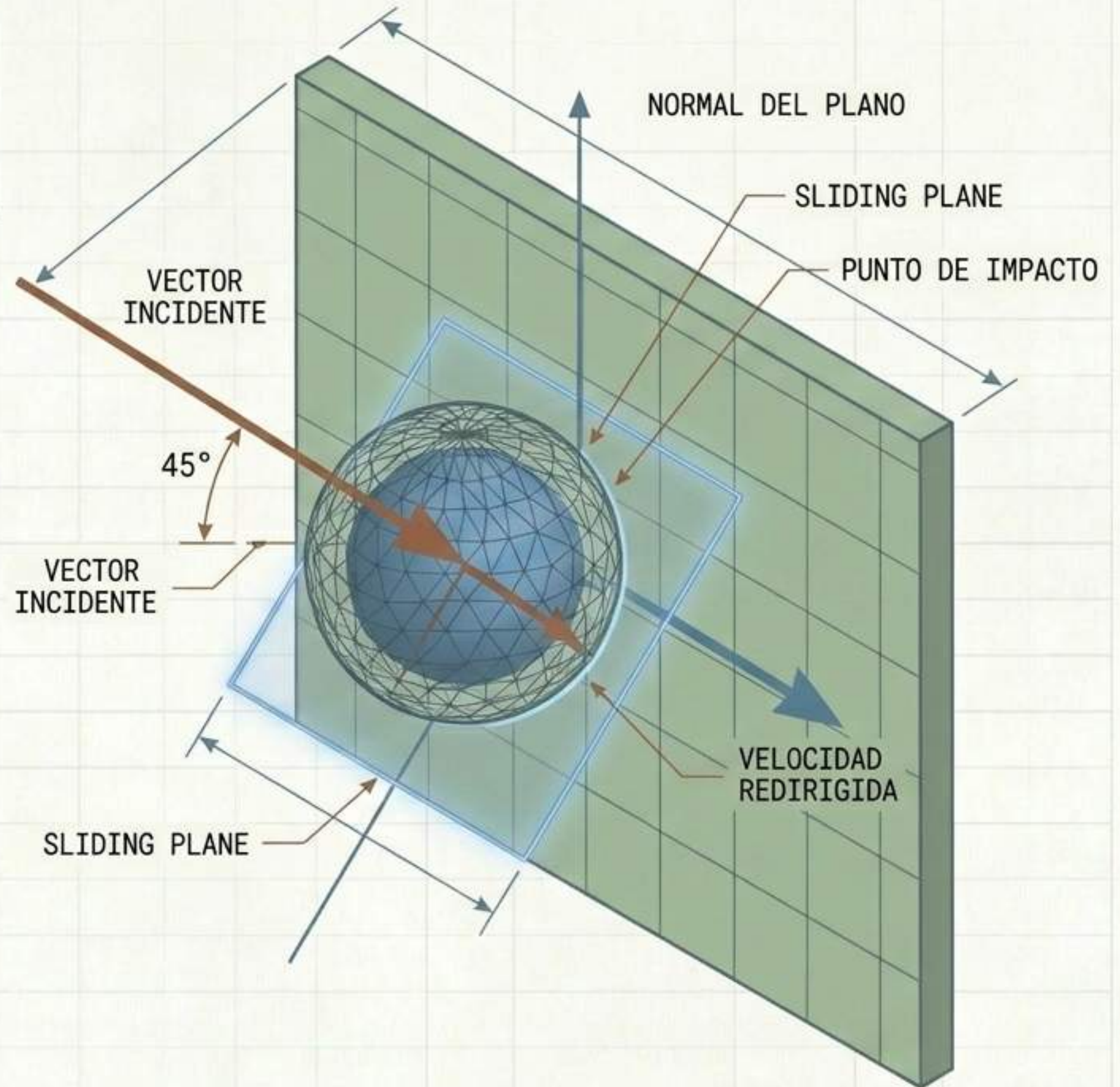
## 3. Impacto de Arista

Si falla, proyectar hacia los bordes para detectar colisiones en esquinas.

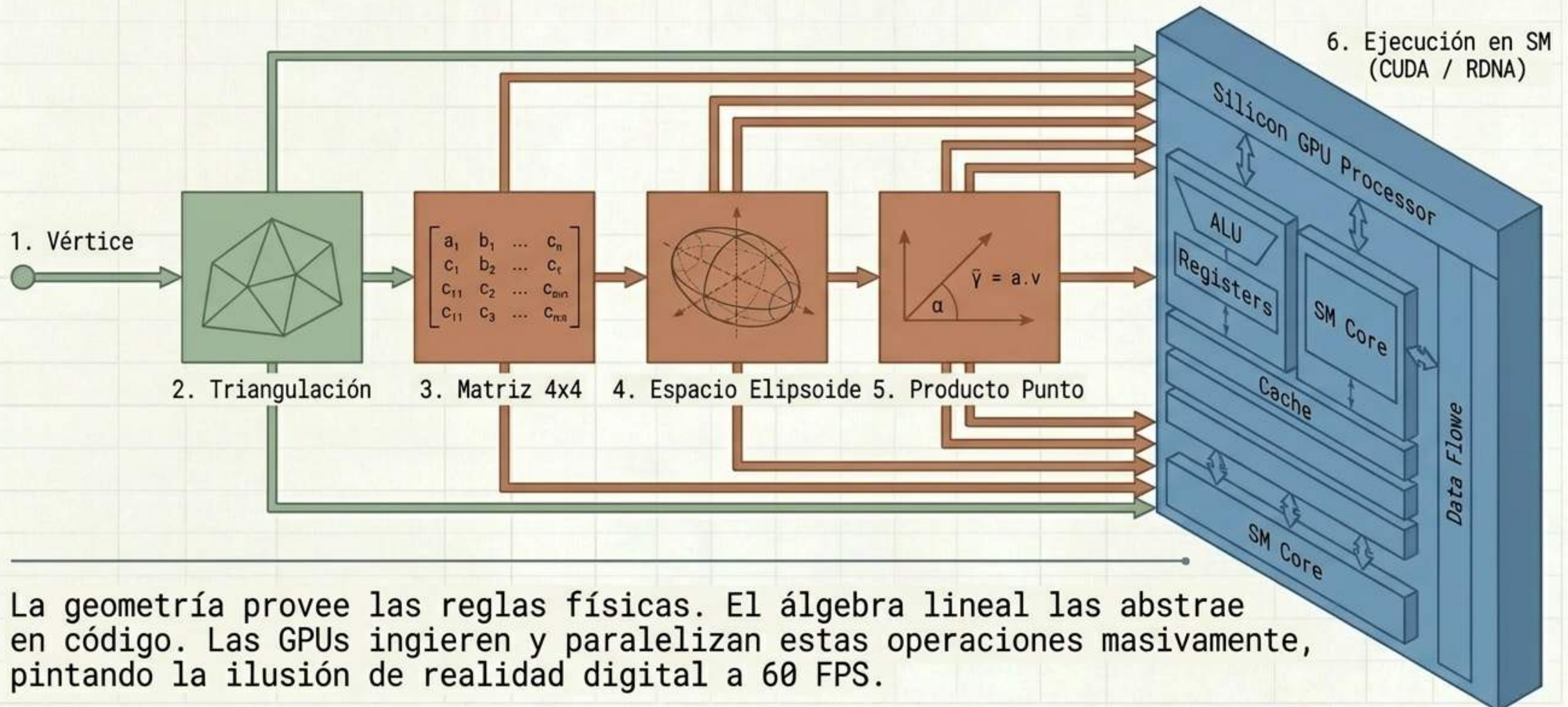


# Respuesta Recursiva y Planos de Deslizamiento.

- En lugar de una detención brusca, el motor calcula el *Sliding Plane* (el plano tangente exacto en el punto de impacto).
- El vector de velocidad original se proyecta sobre este plano tangente.
- La GPU redirige el movimiento sobrante en paralelo al muro, permitiendo a los cuerpos deslizarse fluidamente a lo largo de los obstáculos matemáticos.



# El ciclo simbiótico: Geometría destilada en silicio.



La geometría provee las reglas físicas. El álgebra lineal las abstrae en código. Las GPUs ingieren y paralelizan estas operaciones masivamente, pintando la ilusión de realidad digital a 60 FPS.