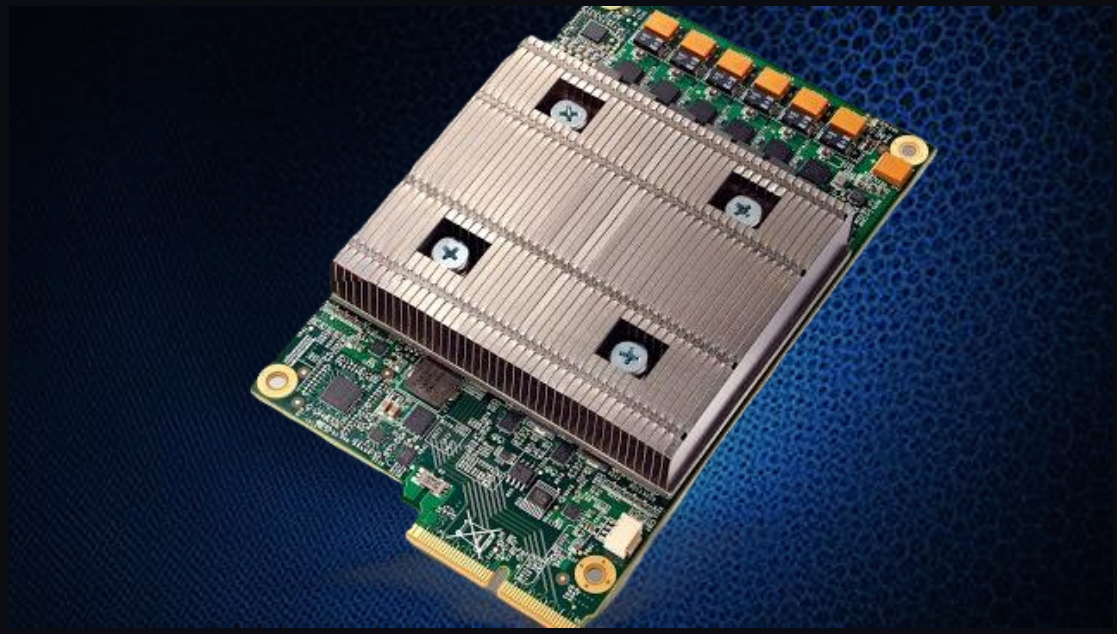
Tensor Processing Units



Universidad de Sonora - Arquitectura de Computadoras

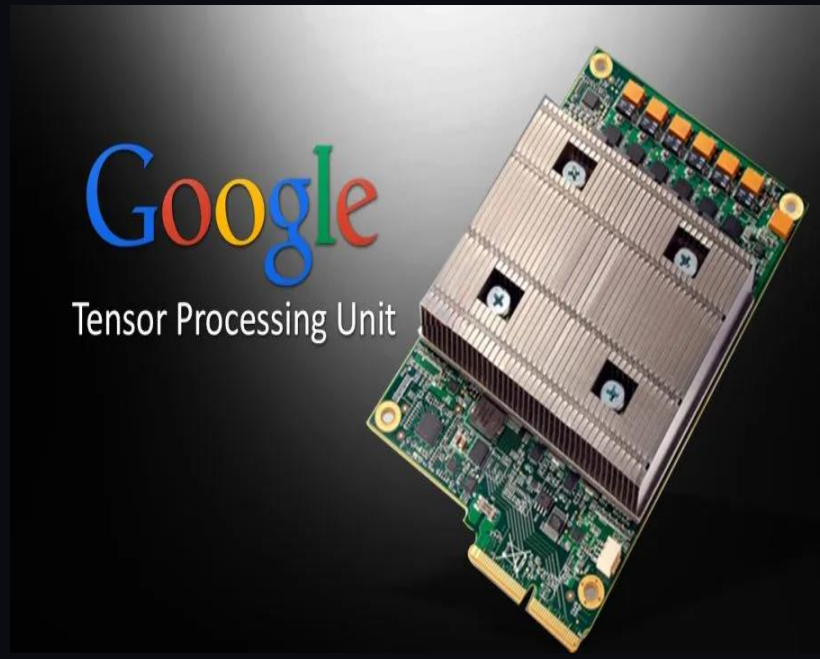
Jesús Beltrán, Rubén Romero, Sebastián Rodríguez

14 de Mayo del 2026

Definición: ¿Qué es una TPU?

Una TPU –abreviatura de unidad de procesamiento tensorial– es un tipo de chip informático optimizado para entrenar y servir ciertos tipos de modelos de IA. Más específicamente, las TPU son una forma de circuito integrado de aplicación específica, o ASIC. Un ASIC es cualquier tipo de chip diseñado para una tarea específica.

Google comenzó a desarrollar TPU en 2015 para su uso en sus propios proyectos de IA. A partir de 2018, los puso a disposición de otras empresas, principalmente ofreciendo instancias de servidores en la nube con tecnología TPU en Google

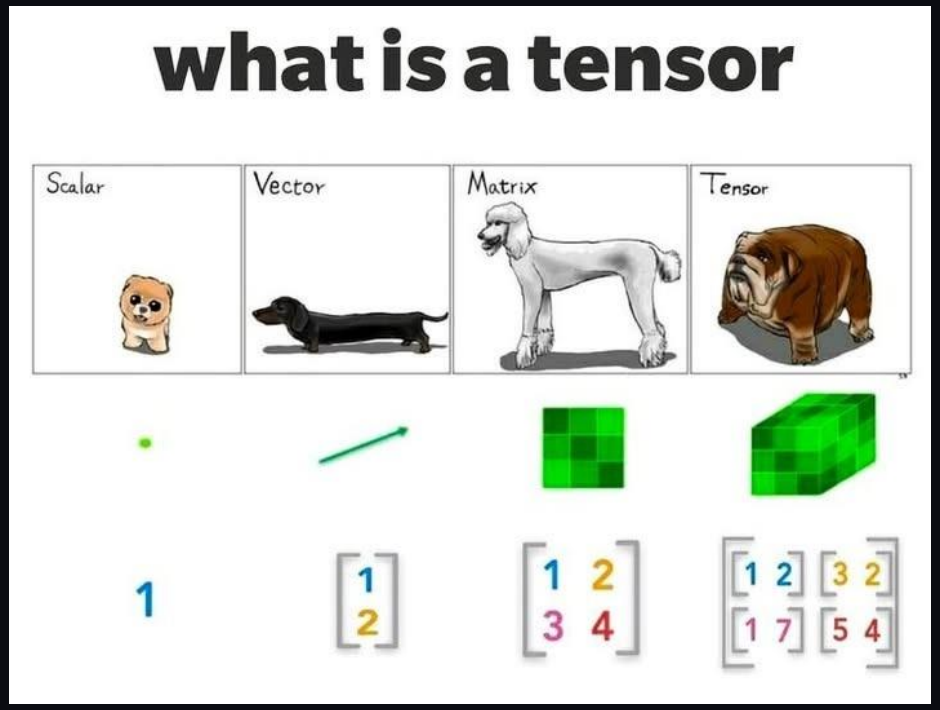


Propósito: ¿Por qué existen las TPU?

Un tensor es una forma de matriz, o número multidimensional, un elemento clave del procesamiento de IA que asigna y utiliza números de alta dimensión a los objetos para procesarlos.

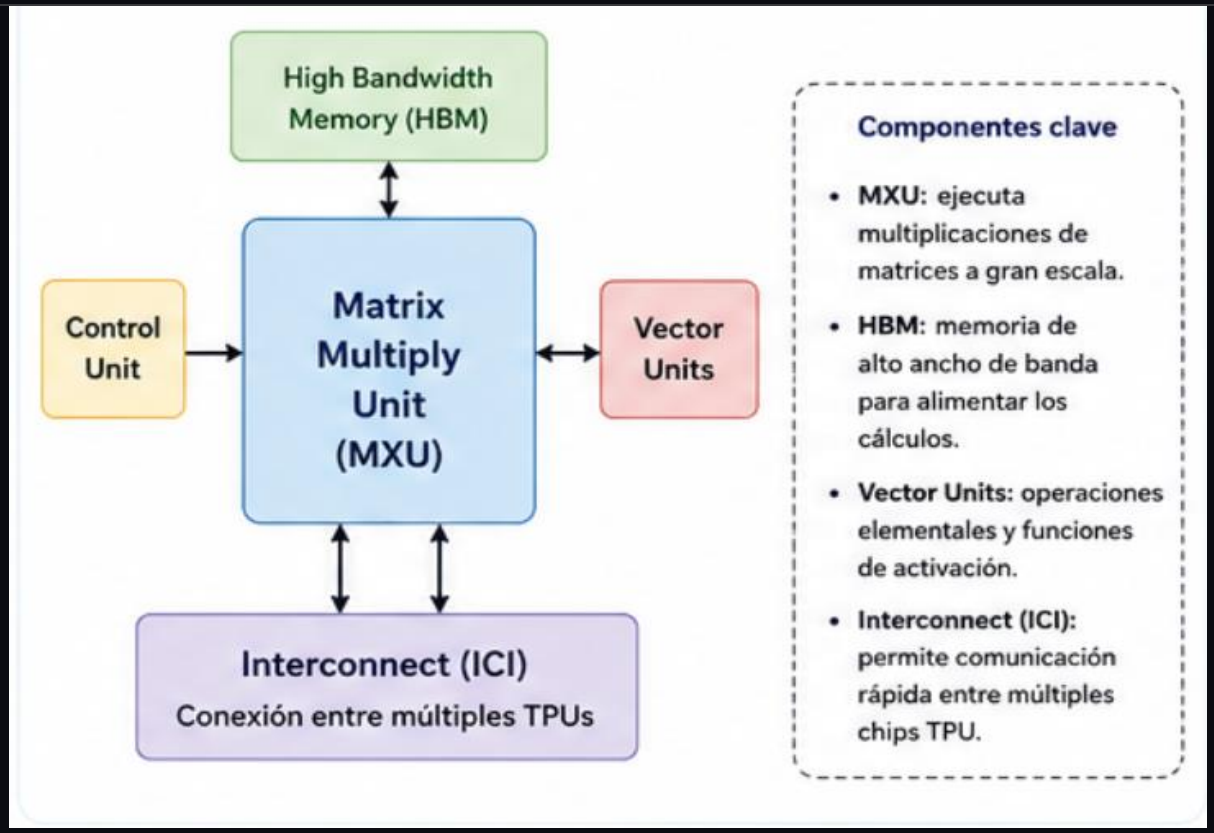
Las TPU utilizan un diseño que coloca los datos y parámetros de un modelo de IA en una matriz y luego los procesa en paralelo.

Este enfoque es beneficioso para las cargas de trabajo de IA que utilizan aprendizaje profundo o aprendizaje por refuerzo – los métodos que impulsan la mayoría de los principales modelos de lenguaje grandes (LLM) disponibles en la actualidad.



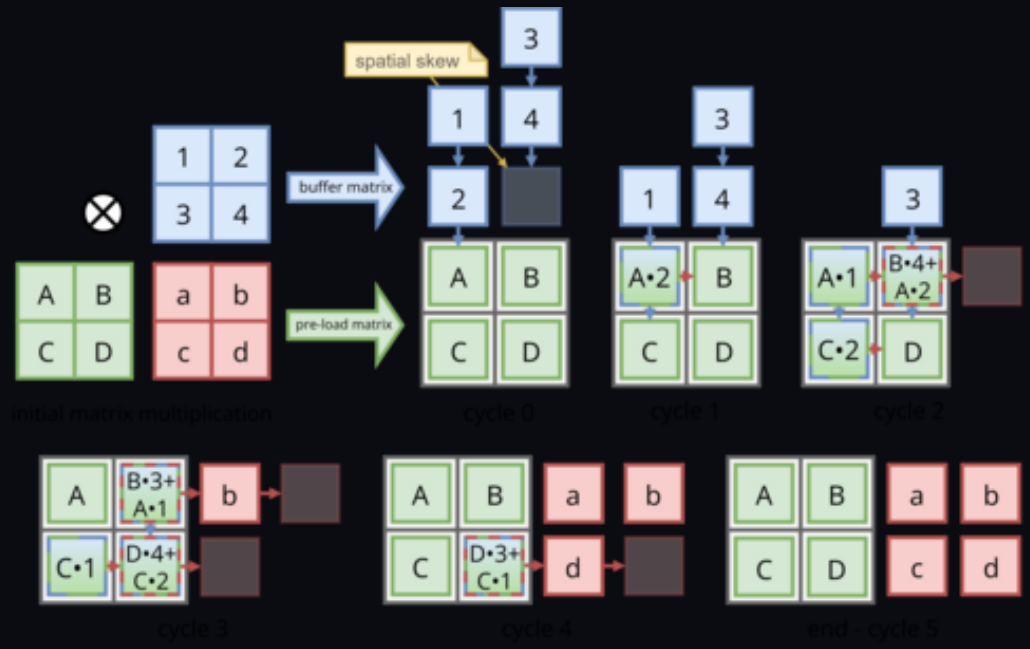
Arquitectura: ¿Cómo funciona una TPU?

1. Recepción de datos: La TPU recibe los datos de entrada del modelo de IA
2. Carga en memoria HBM: Los datos y los pesos de la red neuronal se almacenan temporalmente en la memoria rápida HBM (High Bandwidth Memory).
3. Envío a la Matrix Multiply Unit (MXU): La unidad de control dirige los datos hacia la MXU, que es el núcleo principal de procesamiento.
4. Comunicación entre TPUs: Si el modelo es muy grande, varias TPUs se conectan mediante el sistema Interconnect
5. Generación del resultado: Finalmente, la TPU produce la salida del modelo



Detalles: Arreglos 'Systolicos'

1. Se define a un arreglo systólico como una red de unidades de procesamiento pequeñas que pasan datos de una celda a otra de forma rítmica.
2. Diseño permite multiplicar matrices de manera muy eficiente.
3. Se puede pensar como una línea de producción en donde cada estación hace una parte del trabajo y pasa el resultado a la siguiente.
4. Implementación:
   - Datos entran como filas y columnas.
   - Cada celda multiplica y acumula.
   - Los resultados se mueven por arreglo.
   - Se reutilizan los datos constantemente sin tener que ir a memoria.



Evolución: Cambios a través de generaciones

- **2015 – TPU v1:** Inferencia de redes neuronales.
- **2017 – TPU v2:** Entrenamiento a escala. (Inter-Chip)
- **2018 – TPU v3:** Rendimiento con refrigeración líquida.
- **2021 – TPU v4:** Conmutación de Circuitos Ópticos.
- **2022, 2023 – TPU v5p y v5e:** División por caso de uso. (p = entrenamiento, e = inferencia)
- **2024 – TPU v6 Trillium:** Cuadriplica el rendimiento matricial. (Pasa de 128x128 a 256x256, eficiencia energética mejora 67%)
- **2025 – TPU v7 Ironwood:** Era de la inferencia con FP8.

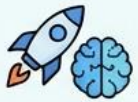
Use Case	Best Choice	Rationale
Cost-optimized inference	TPU v5e	Lowest cost per inference query
Large-scale training (>1000 chips)	TPU v5p or Ironwood	3D torus + OCS enables massive pods
Medium training jobs (256 chips)	TPU v6e Trillium	Best perf/watt, 4.7× compute vs v5e
Memory-bound models (>70B params)	Ironwood	192GB HBM enables larger batch sizes
Long-context inference (>100K tokens)	Ironwood	HBM capacity supports massive KV caches
Embedding-heavy workloads	TPU v5p or Ironwood	SparseCore + large HBM

Comparativa: CPUs vs GPUs vs TPUs

Característica	CPU	GPU	TPU
Propósito	General	Paralelismo gráfico y cómputo	IA / machine learning
Flexibilidad	Muy alta	Alta	Menor, más especializada
Mejor uso	Lógica general, prototipos, I/O	Entrenamiento IA, gráficos, cómputo paralelo	Modelos grandes con muchas operaciones matriciales
Ecosistema	Universal	Muy amplio	Más ligado a Google Cloud/JAX/PyTorch/Tensor Flow
Ventaja principal	Versatilidad	Paralelismo flexible	Eficiencia para tensores y matrices

VENTAJAS Y DESVENTAJAS DE UTILIZAR CHIPS TPU (TENSOR PROCESSING UNIT)

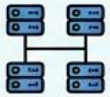
VENTAJAS



1. Alto Rendimiento (IA/AA)
Optimizado específicamente para cargas de trabajo de Inteligencia Artificial y Aprendizaje Automático

2. Velocidad de Entrenamiento Excepcional
Acelera drásticamente el entrenamiento de modelos grandes (Deep Learning)

3. Alta Eficiencia Energética (TOPS/W)
Consume menos energía por operación que las CPUs/GPUs para tareas de IA



4. Escalabilidad (Google Cloud)
Fácilmente escalable en Google Cloud Platform para proyectos masivos.

DESVENTAJAS

1. Costo Elevado de Hardware/Servicio
El costo inicial y operativo (GCP) puede ser muy alto

2. Complejidad de Implementación
Requiere conocimientos especializados e integración específica con frameworks de Google

3. Ecosistema de Software Limitado (Bloqueo)
Dependencia fuerte de TensorFlow/JAX; menos versatilidad que GPUs generales

4. Menos Versatilidad General
No adecuado para cargas de trabajo que no sean de IA (gráficos, bases de datos)

5. Disponibilidad Restringida
Principalmente accesible a través de Google Cloud, no disponible ampliamente para compra.

Para generar la imagen anterior se utilizó:

Gemini 3.1 Flash Image (Nano Banana 2). Es un modelo multimodal de última generación diseñado para la generación de imágenes de alta fidelidad y edición rápida.

- Fue entrenado con TPUs
- También utiliza TPUs para generar imágenes dentro de un rango de 2 a 5 segundos
- Se gastaron aproximadamente 0.051 horas-watt para generarla
- Con un costo en pesos de alrededor de 10 milésimas de centavo

Proyectos que los han utilizado

- Entrenamiento de Grandes Modelos de Lenguaje (LLMs), puesto que embonan *perfectamente* con la arquitectura de *Transformers*.
- Google Search y Google Photos
- AlphaGo y AlphaZero
- Investigación Genómica y Plegamiento de Proteínas
- Sistemas de Recomendación

Cuándo NO usarlos

1. Renderizado de Gráficos y Video

2. Desarrollo Prototípico y Experimentación Rápida

- Las TPU requieren que el código sea muy específico (especialmente en la gestión de memoria y tipos de datos). Las GPUs pueden utilizarse con código que no está perfectamente optimizado.
- Por lo tanto, para probar nuevas ideas o prototipos, con tan sólo configurar las TPUs se perdería el beneficio de velocidad.

Cuándo NO usarlos

3. Modelos Pequeños o con Datos Limitados

- Las TPU tienen una "latencia de arranque" y están diseñadas para procesar lotes (*batches*) de datos muy grandes.
- Si el modelo es pequeño o tiene pocos datos, el tiempo que se tarda en cargar los datos en la memoria de alta velocidad es mayor a simplemente correrlo en una GPU.

4. Cargas de Trabajo de Propósito General (Minería, Criptografía, Bases de Datos)

- Muchos algoritmos de minería o de búsqueda en bases de datos requieren operaciones lógicas, por lo tanto no se pueden correr en TPUs.

Conclusión: Un ASIC que usamos a diario sin darnos cuenta

Los avances en la computación que hemos visto en los últimos años han sido posibles gracias a los TPUs. Muchos proyectos , como los mencionados anteriormente, han dado lugar a los servicios que más utilizamos hoy en día, sin siquiera hablar solamente de LLMs.

A pesar de que los TPUs son propiedad de Google y no podemos aún saber cómo es que funcionan exactamente, la explicación proporcionada por la misma compañía elucida un método bastante eficiente y específico que abrió las puertas al procesamiento de datos masivo.

Es por eso que sacrificar flexibilidad por eficiencia incrementa enormemente los beneficios y la accesibilidad de los servicios que más utilizamos, ya que no hay optimización para un procesador de propósito general que se le pueda comparar a la velocidad que un TPU logra.