Uso de ChatGPT e IA Generativa:

En este práctico NO se permite el uso de estas herramientas.

Práctico 3 - Acceso a la memoria de la GPU

El objetivo de este práctico es poner en práctica conceptos relacionados con el acceso a los distintos espacios de memoria de la GPU, en especial la memoria global y la memoria compartida. También se podrá ver cómo la elección de los parámetros de la grilla puede afectar el desempeño.

Los tiempos de ejecución deben tomarse como el promedio de 10 ejecuciones (presentando únicamente el promedio y desviación estándar en el informe). Para obtener este dato fácilmente es conveniente utilizar la herramienta NsightSystems de la siguiente forma:

```
sbatch lanzar.sh nsys profile --stats true ./programa
```

Ejercicio 1	Memoria global

Nota:

Uno de los conceptos clave para resolver los ejercicios es el de acceso coalesced, es decir, el mecanismo mediante el cual se divide una operación de memoria de los threads de un warp en cierto número de transacciones. La regla para unificar los accesos es sencilla y podría escribirse de la siguiente manera: **los accesos a memoria simultáneos de un warp son agrupados en la menor cantidad de transacciones de 32B necesaria para satisfacer todos los accesos[a].**

[a] Válido para dispositivos de la generación 6.0 en adelante o 5.x sin caché L1 para accesos a memoria global.

Construir un kernel que reciba una matriz de enteros alojada en memoria global y devuelva la matriz transpuesta. Reserve dos espacios de memoria distintos para las matrices de entrada y salida. En este ejercicio el kernel no debe utilizar la memoria compartida (todas las lecturas y escrituras deben realizarse en memoria global). La grilla debe ser bidimensional y los bloques también deben ser bidimensionales.

1. Ejecute el kernel con un tamaño de bloque de 32×32 y analice el patrón de acceso a memoria global de cada warp que se da en las lecturas y escrituras. Mida el tiempo de ejecución del kernel.

2. Modifique el tamaño de bloque para reducir los accesos no-coalesced. Justifique adecuadamente la elección de tamaño de bloque. Ejecute el kernel nuevamente y compare el tiempo de ejecución con el caso anterior.

Ejercicio 2	Memoria compartida

En este ejercicio se seguirá una estrategia distinta para favorecer el acceso coalesced en la lectura y escritura para el kernel que traspone la matriz. Esta consiste en usar una copia del tile a trasponer en memoria compartida de forma de realizar los accesos que serían ineficientes sobre el memoria compartida en lugar de sobre la memoria global.

1. Vuelva a configurar la grilla para utilizar bloques de tamaño 32×32. Reserve un espacio en memoria compartida de tamaño igual al del bloque y utilice este espacio para evitar los accesos no-coalesced a la memoria global (es decir, realizar los accesos que serían no-coalesced sobre este espacio en lugar de la memoria global). Compare el desempeño del kernel con las versiones del ejercicio anterior.

2. Analice el patrón de lectura y escritura en la memoria compartida y determine cuando ocurren conflictos de bancos. Solucione los conflictos de bancos agregando una columna *dummy* al final del tile de memoria compartida. Compruebe el efecto de esta modificación comparando los tiempos de ejecución con la parte anterior y explique por qué esto soluciona los conflictos de bancos.

Entrega

- Se debe entregar un informe en PDF con la solución de los ejercicios que contenga, **como máximo, 4 páginas** (sin contar índices, carátulas, figuras, etc.).
- Entregar un archivo comprimido con el código fuente y los scripts necesarios para ejecutar un caso de prueba de forma sencilla.