Proyecto #3: CUDA

**Contenido**

[Anexos: 3](#_Toc135881989)

[Anexo 1: Seguimiento de guía: 3](#_Toc135881990)

# Anexos:

## Anexo 1: Seguimiento de guía:

1.a Cálculo de la fórmula para crear el gloID en el kernel

A picture containing text, screenshot, font

Description automatically generated

Se usó una fórmula genérica para distinguir a cada hilo individualmente basada en desplazar el identificador del hilo dentro del bloque por la cantidad de hilos presentes en los bloques que preceden al bloque actual en el orden. Es mejor usar esta fórmula porque es lo mínimo que se necesita para distinguir a cada hilo individualmente. Existen fórmulas más complicadas que toman en cuenta la dimensionalidad de la cuadrícula, pero no es necesario usarlas dado que el kernel opera linealmente.

1.b Hace falta la liberación de memoria al final del programa.

A screenshot of a computer code

Description automatically generated with low confidence

2 Incorpore medición de tiempo de la llamada al kernel mediante el uso de CUDA events.

Siguiendo la documentación oficial:

A screenshot of a computer screen

Description automatically generated with low confidence

A screen shot of a computer code

Description automatically generated with low confidence

A picture containing text, screenshot, font

Description automatically generated

3 Podemos ver que en el kernel se calcula xCoord y también yCoord. Explique en sus palabras que se está realizando en esas operaciones y porque se calcula de tal forma.

La librería pgm transformó a la imagen de un vector de dos dimensiones en un vector de solo una dimensión al colocar todos los pixeles de corrido al partir la imagen en filas y colocando una fila seguida de la otra. Entendiendo esto, podemos revertir esa misma transformación al dividir el identificador del hilo por el ancho de la imagen y usando el residuo como la coordenada en horizontal y al cociente como la vertical. Finalmente, se traslada la transformación de un rango entre 0 a 1, a un rango entre -0.5 y 0.5 al restar la mitad de del ancho y del alto de las coordenadas horizontal y vertical, respectivamente; con el objetivo de que el origen del plano (0, 0) sea el centro de la imagen.

# Referencias

Harris, M. (25 de 5 de 2023). *How to Implement Performance Metrics in CUDA C/C++ | NVIDIA Technical Blog*. Obtenido de NVIDIA Developer: https://developer.nvidia.com/blog/how-implement-performance-metrics-cuda-cc/