

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
Facultad de Ingeniería

Fundamentos de programación en
GPUs de propósito general con CUDA

LABORATORIO DE INTEL PARA LA ACADEMIA
Y CÓMPUTO DE ALTO DESEMPEÑO

Ariel Ulloa Trejo
Elba Karen Sáenz García
Laura Sandoval Montaña

Temario

- **Antecedentes**
- GPU – CUDA
 - GPGPU
 - GPU Genérico
 - Arquitectura CUDA
- Modelo de ejecución CUDA
 - Grids ,Bloques, Hilos
- Modelo de programación CUDA
 - Device y Host
 - Funciones CUDA
- Manejo de arreglos unidimensionales
- Manejo de arreglos bidimensionales
- Memoria Compartida



Manejo de Matrices

Manejo de Errores

- Vimos que las funciones pasadas regresan un valor *cudaError_t*.
- Éste es usado para detectar errores; si la ejecución fue exitosa, regresa un valor *cudaSuccess*. En otro caso, será el código del error.
- Para poder leer los códigos de los errores, usamos *cudaGetErrorString()*.
- *cudaGetLastError()* envía el último error. Si hubo uno antes de éste, no es reportado.

```
cudaError_t cudaGetLastError ( void )  
const char* cudaGetErrorString (cudaError_t error)  
error - code to convert to string
```

Manejo de Matrices

- Como los kernels son asíncronos, es necesario bloquear la ejecución hasta que el dispositivo haya terminado su trabajo. Para esto utilizamos la función *cudaDeviceSynchronize()*.



Errores comunes:

- Utilizar variables en un segmento de código donde no existen.
- La configuración de ejecución es inválida.
- Copiar memoria mayor a la alojada.
- El origen y destino de la copia está mal.

Manejo de Matrices

Ejemplo

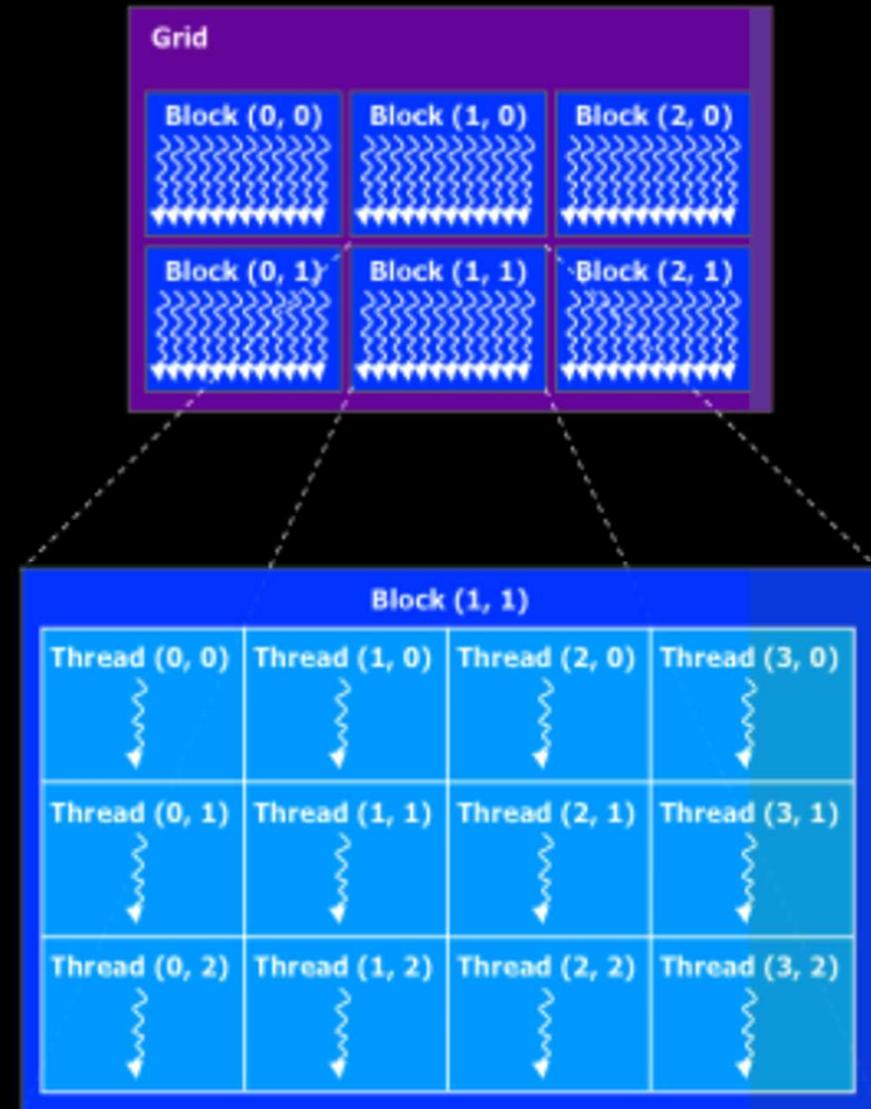
- Utilizar la biblioteca *Error.h* y generar y detectar errores en el código ejemplo.

05_errores_01.cu

Manejo de Matrices

Más de grids, bloques e hilos:

- CUDA permite la ejecución de *grids* compuestos por bloques hasta de 3 dimensiones, y a su vez, cada bloque compuesto por hilos en 3D.
- Para hacerlo, utilizamos `dim3` (para encapsular datos multidimensionales).



Manejo de Matrices

- *Kernel<<<dim3(Ax, Ay, Az), dim3(Bx,By,Bz)>>>()*
- *Kernel<<<dim3(10, 32), dim3(10, 10)>>>()*
- ***En caso de que una dimensión no sea especificada, es reemplazada automáticamente por 1.***

Manejo de Matrices

Ejemplo

- Programa que suma dos matrices cuadradas.

Manejo de Matrices

Ejercicio:

- Dividir una matriz de 6x6 en 4 bloques y sumar a cada elemento el primero de cada bloque. Utilizar la configuración:
 - dim3 bloques(2, 2)
 - dim3 hilos(3, 3)

Elementos de la Matriz

0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17
18	19	20	21	22	23
24	25	26	27	28	29
30	31	32	33	34	35

Ejemplos:

$$0 = 0 + 0$$

$$7 = 7 + 0$$

$$16 = 16 + 3$$

$$33 = 33 + 21$$

Grid

bloque (0,0)

bloque (1,0)

(0,0)	(1,0)	(2,0)	(0,0)	(1,0)	(2,0)
(0,1)	(1,1)	(2,1)	(0,1)	(1,1)	(2,1)
(0,2)	(1,2)	(2,2)	(0,2)	(1,2)	(2,2)
(0,0)	(1,0)	(2,0)	(0,0)	(1,0)	(2,0)
(0,1)	(1,1)	(2,1)	(0,1)	(1,1)	(2,1)
(0,2)	(1,2)	(2,2)	(0,2)	(1,2)	(2,2)

bloque (0,1)

bloque (1,1)

Manejo de Matrices

- *Medir tiempos en el GPU*
- *Para medir los tiempos que toman las actividades en el GPU, se utilizan los siguientes métodos:*
 - *cudaEventCreate()* *Crea las variables*
 - *cudaEventRecord()* *Realiza el inicio y parada*
 - *cudaEventDestroy()* *Libera variables*
 - *cudaEventElapsedTime()* *Obtiene dif. de tiempos*

GpuTimer.h

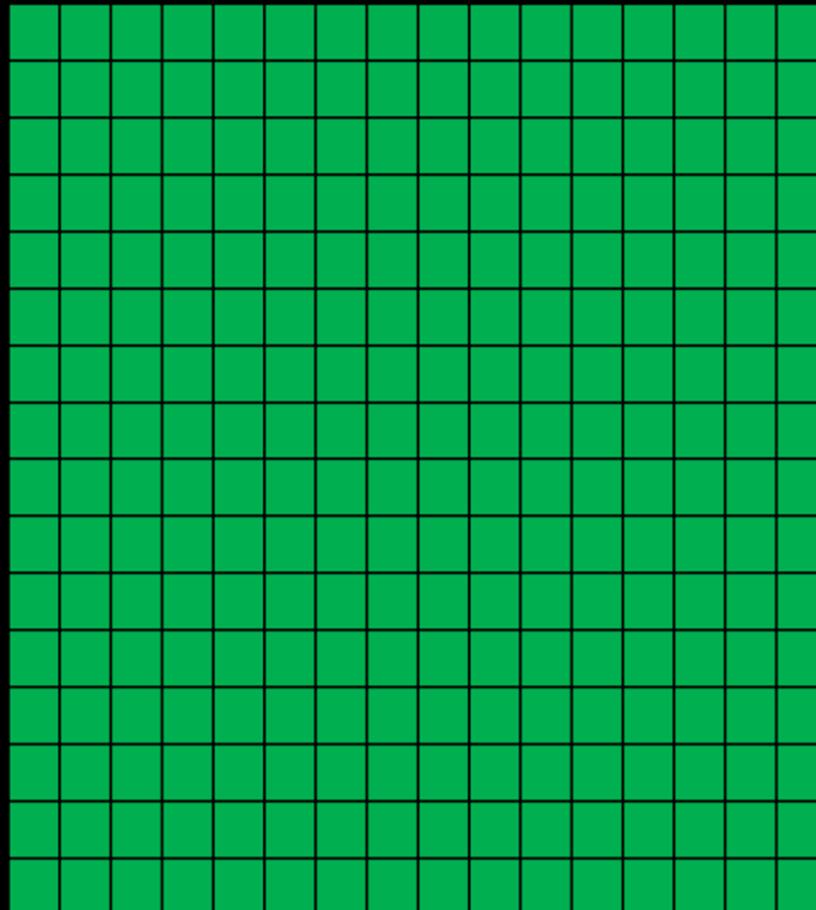


*Ariel Ulloa - Laura
Sandoval-Karen Sáenz*

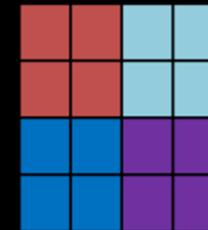
Ejercicio

- Hacer un programa genérico que sume dos matrices cuadradas. Utilizar manejo de errores y medir tiempo con diferentes configuraciones de ejecución

Matriz

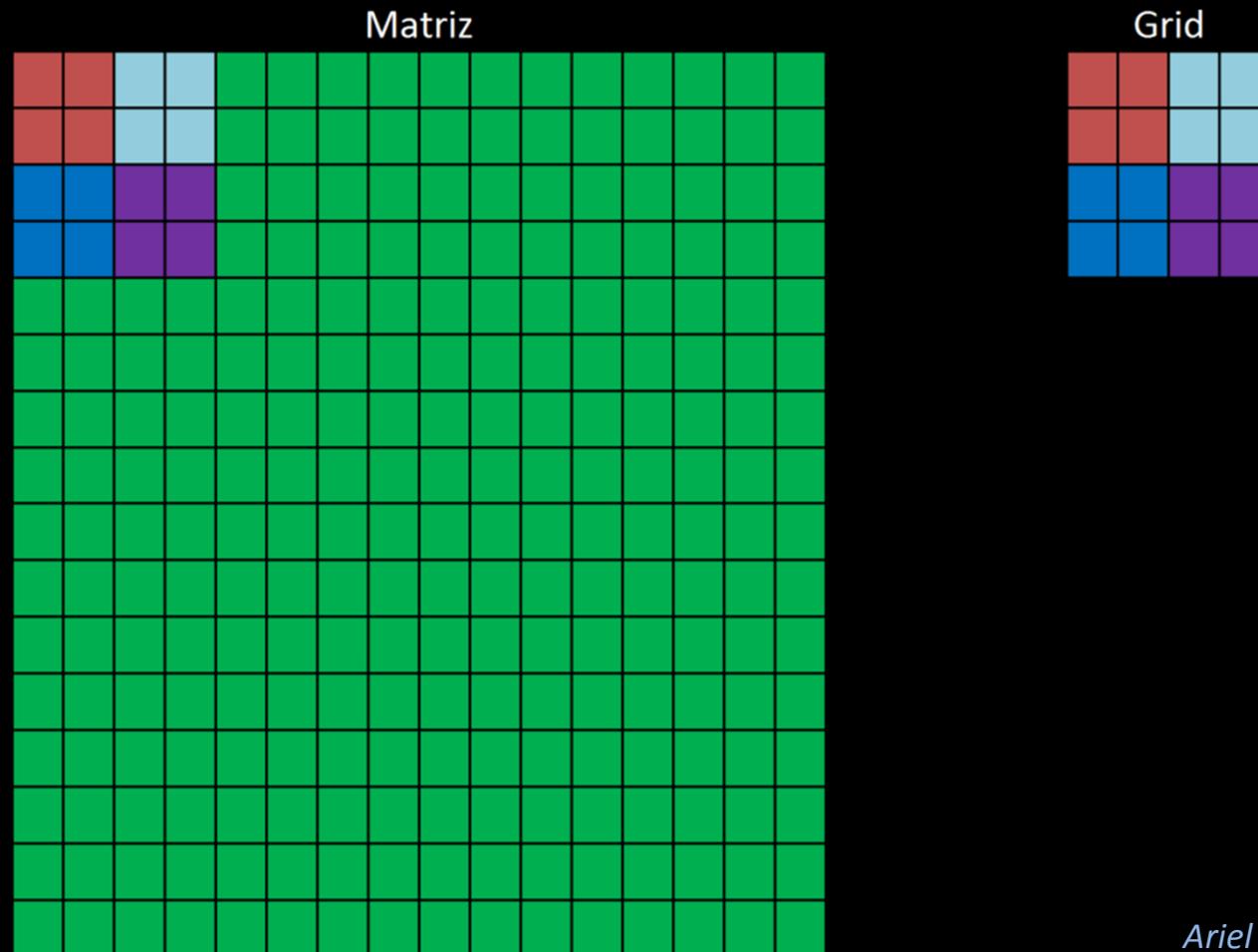


Grid



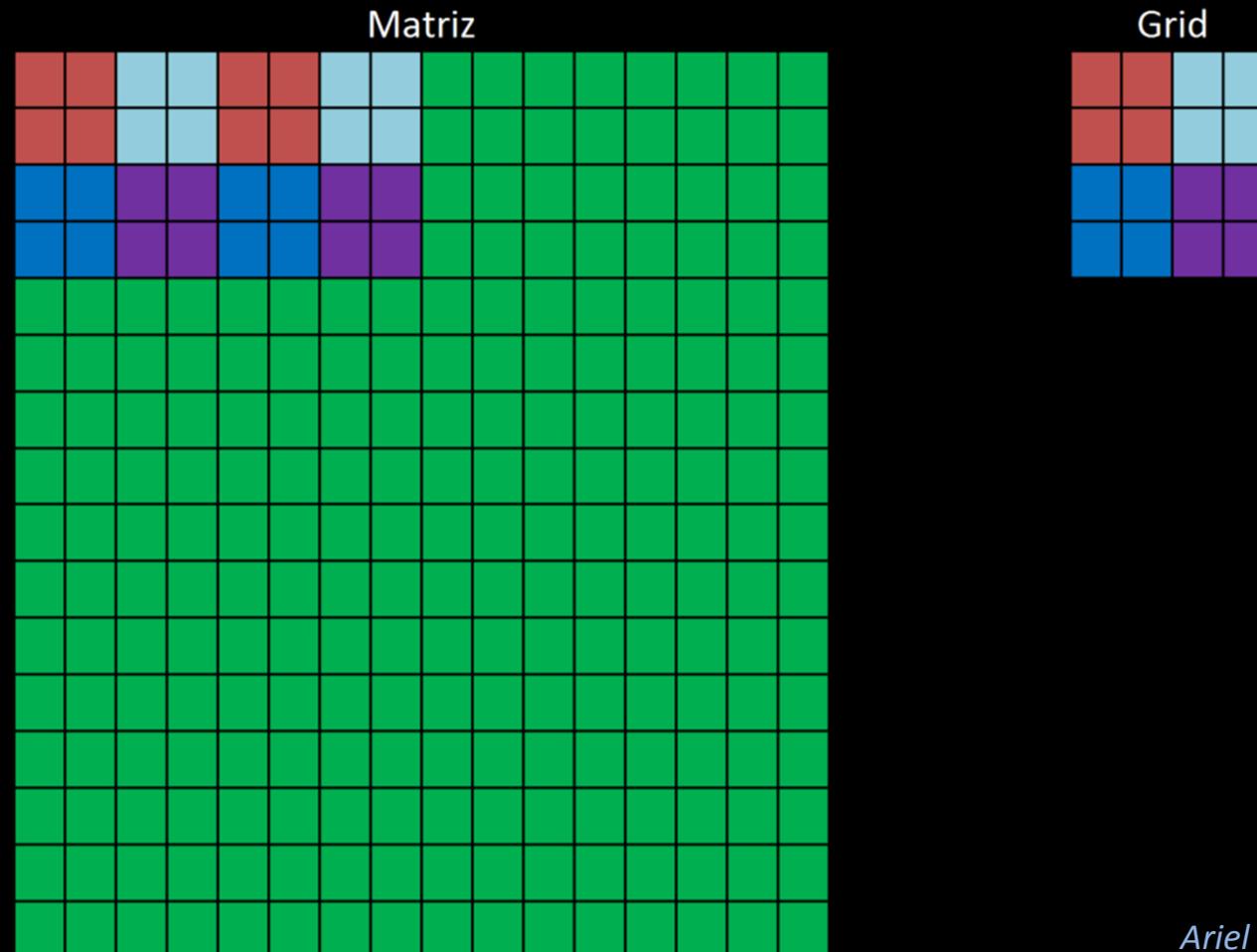
Ejercicio

- Hacer un programa genérico que sume dos matrices cuadradas. Utilizar manejo de errores y medir tiempo con diferentes configuraciones de ejecución



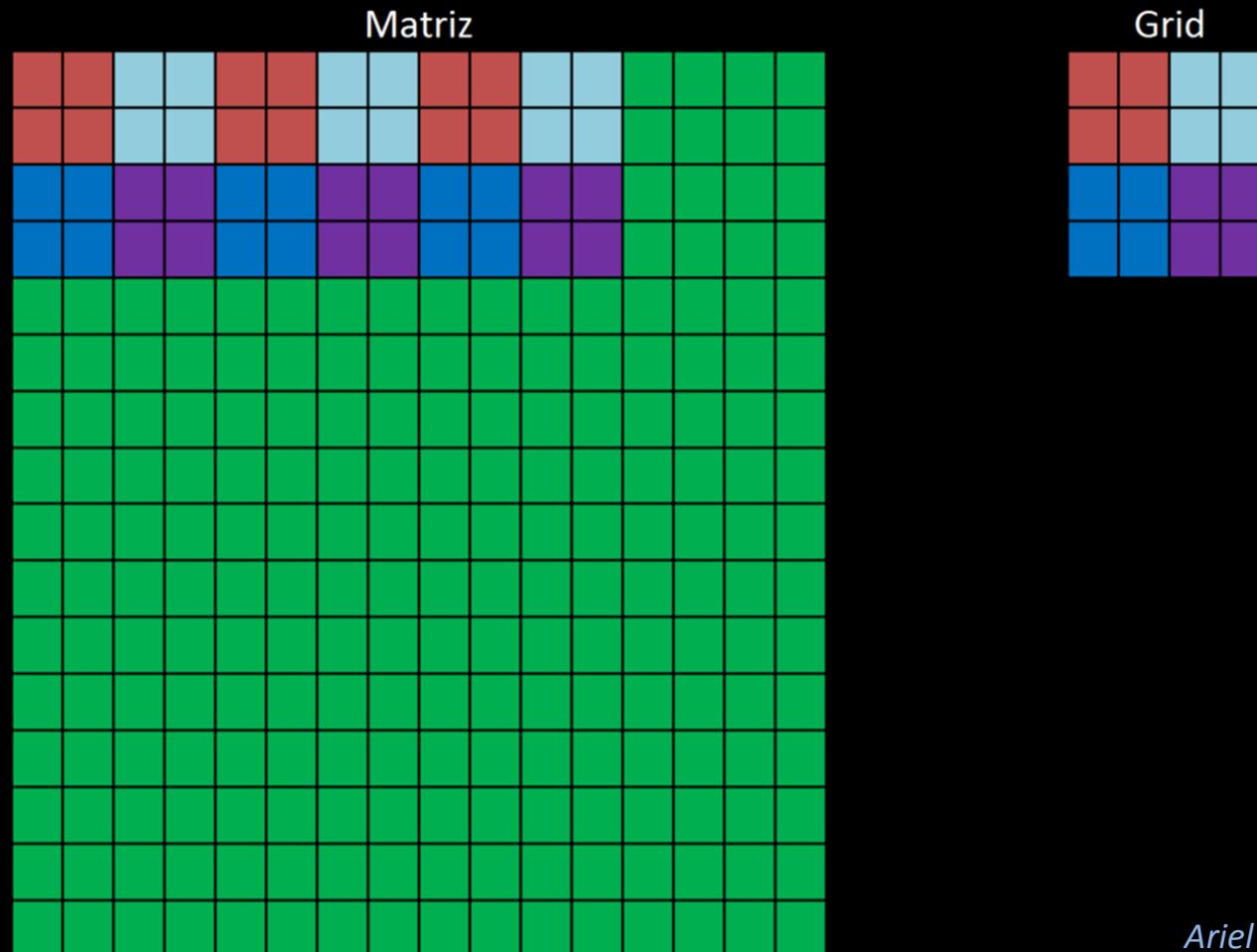
Ejercicio

- Hacer un programa genérico que sume dos matrices cuadradas. Utilizar manejo de errores y medir tiempo con diferentes configuraciones de ejecución



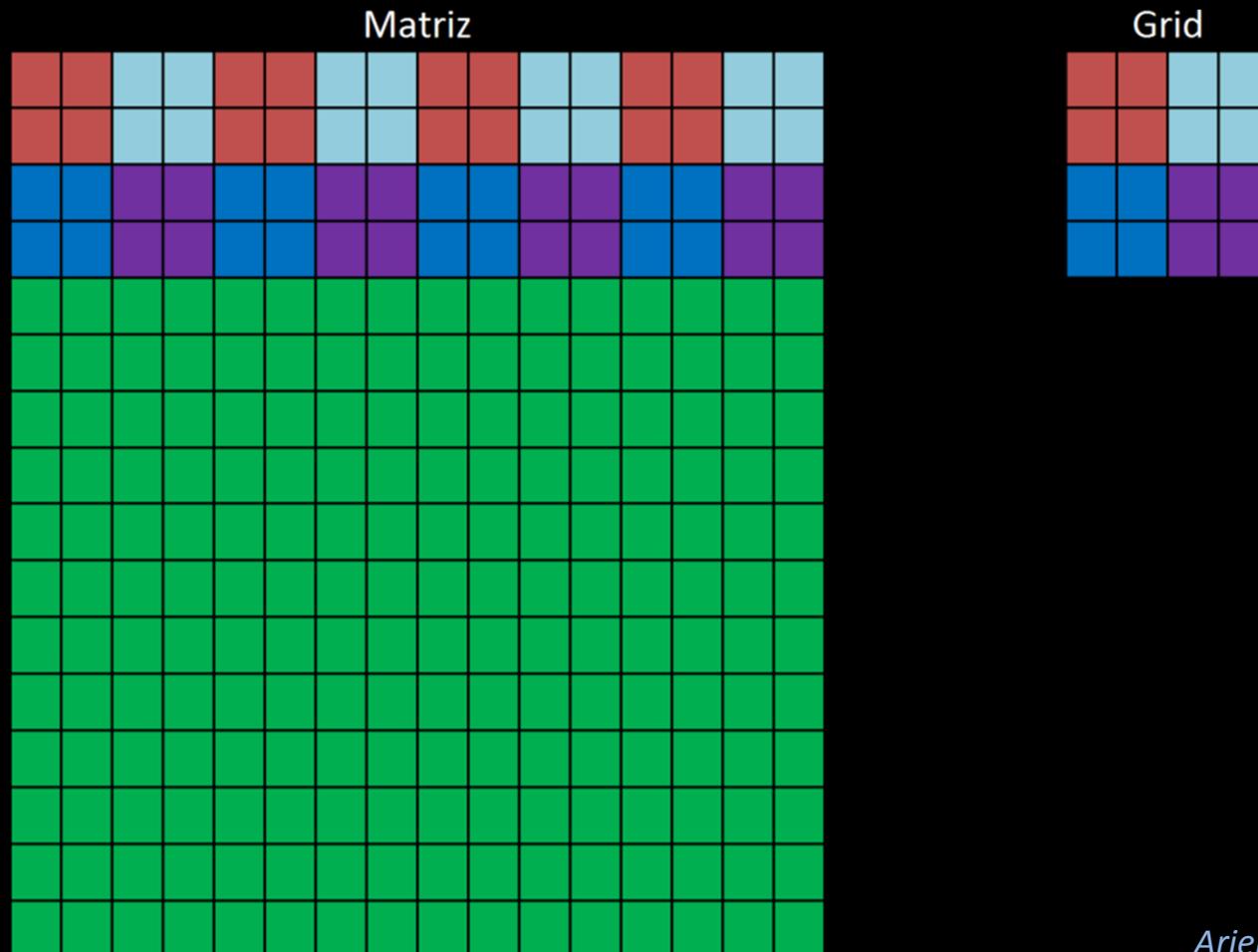
Ejercicio

- Hacer un programa genérico que sume dos matrices cuadradas. Utilizar manejo de errores y medir tiempo con diferentes configuraciones de ejecución



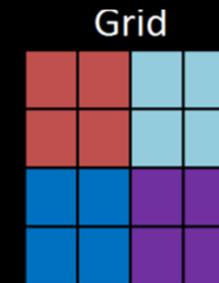
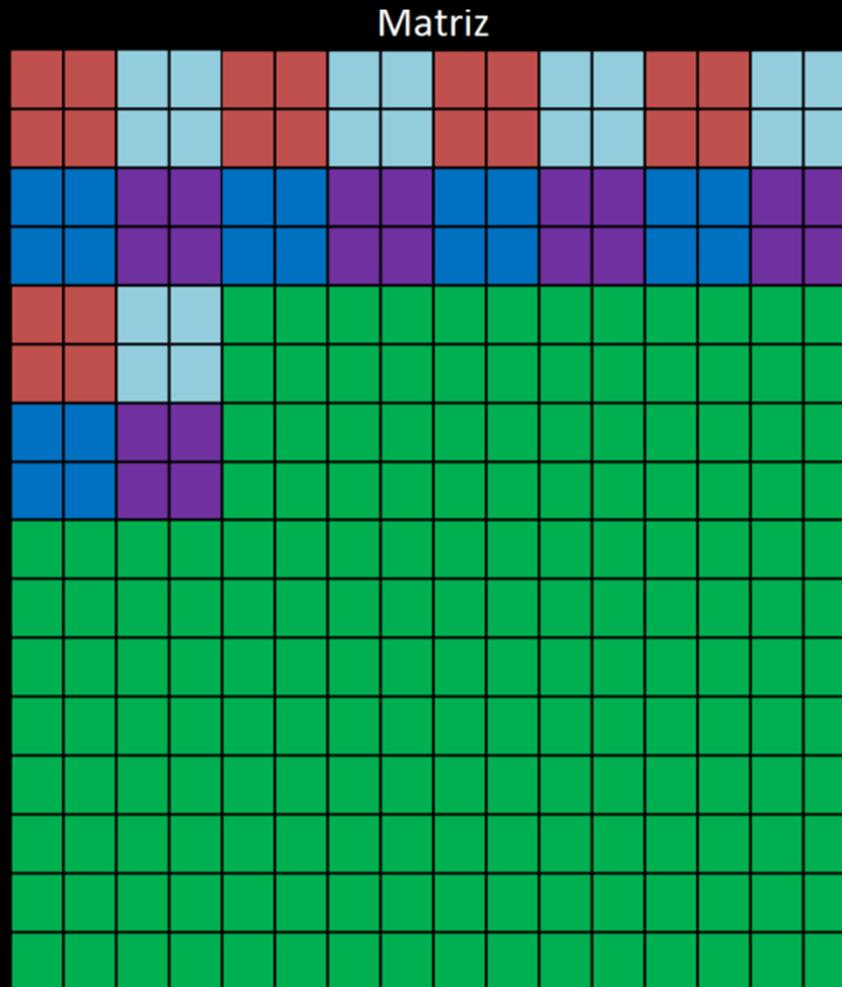
Ejercicio

- Hacer un programa genérico que sume dos matrices cuadradas. Utilizar manejo de errores y medir tiempo con diferentes configuraciones de ejecución



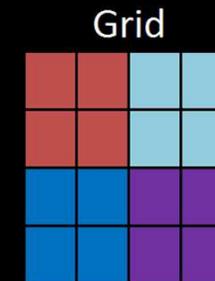
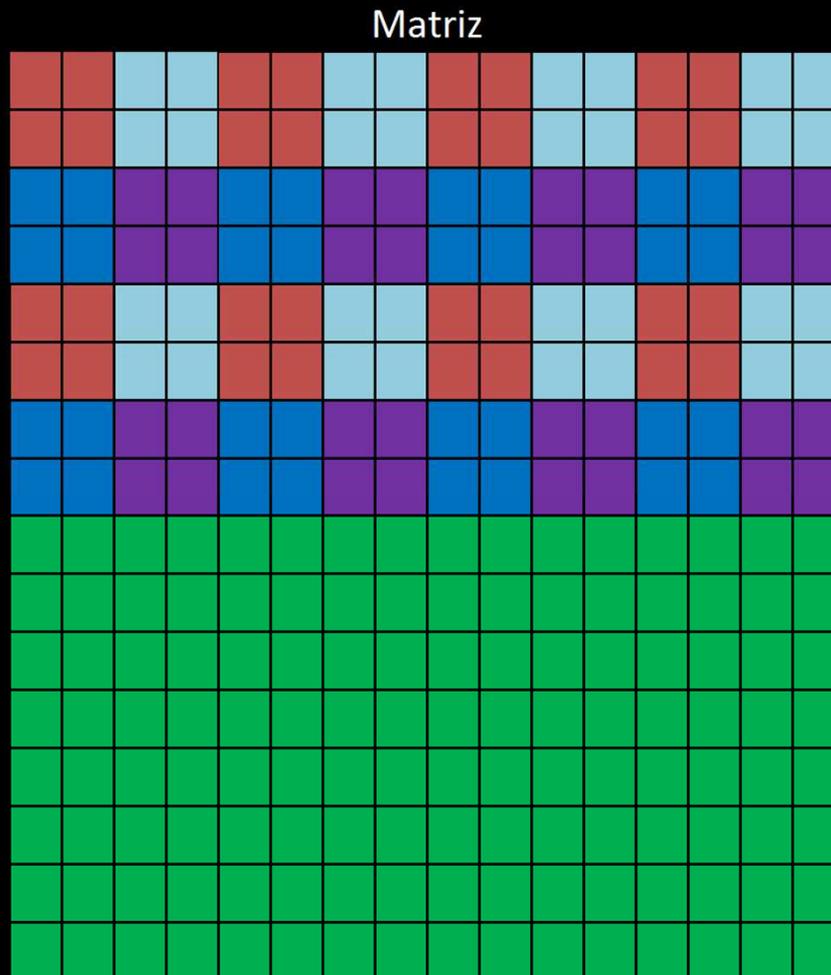
Ejercicio

- Hacer un programa genérico que sume dos matrices cuadradas. Utilizar manejo de errores y medir tiempo con diferentes configuraciones de ejecución



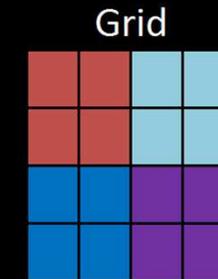
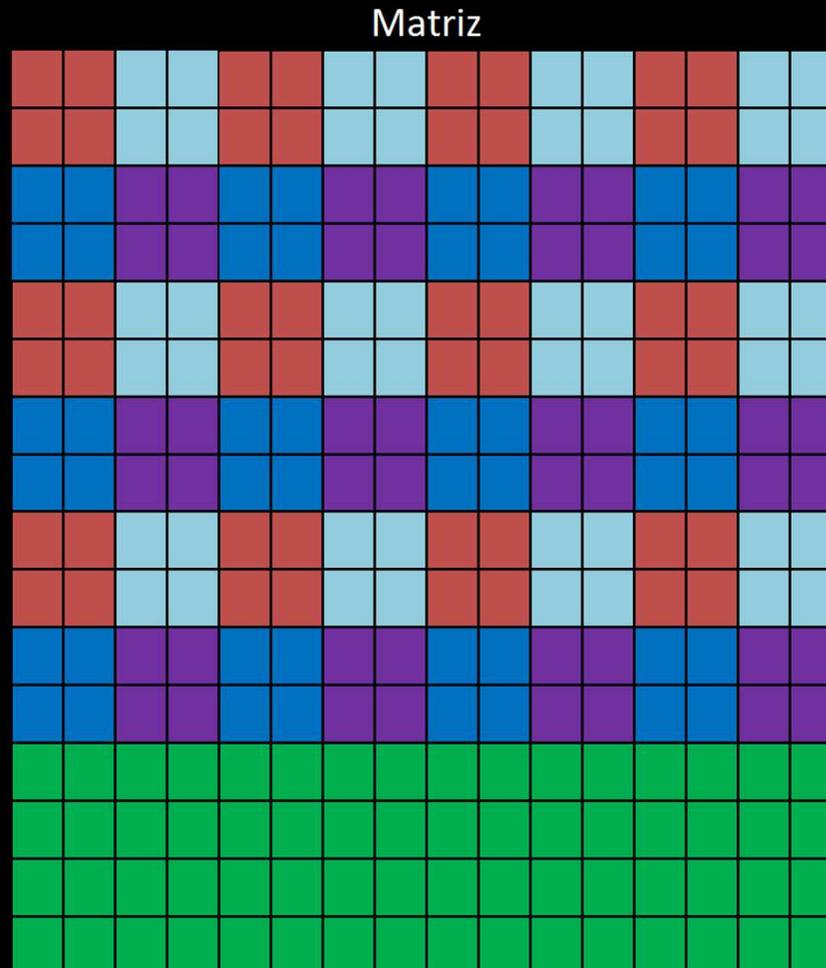
Ejercicio

- Hacer un programa genérico que sume dos matrices cuadradas. Utilizar manejo de errores y medir tiempo con diferentes configuraciones de ejecución



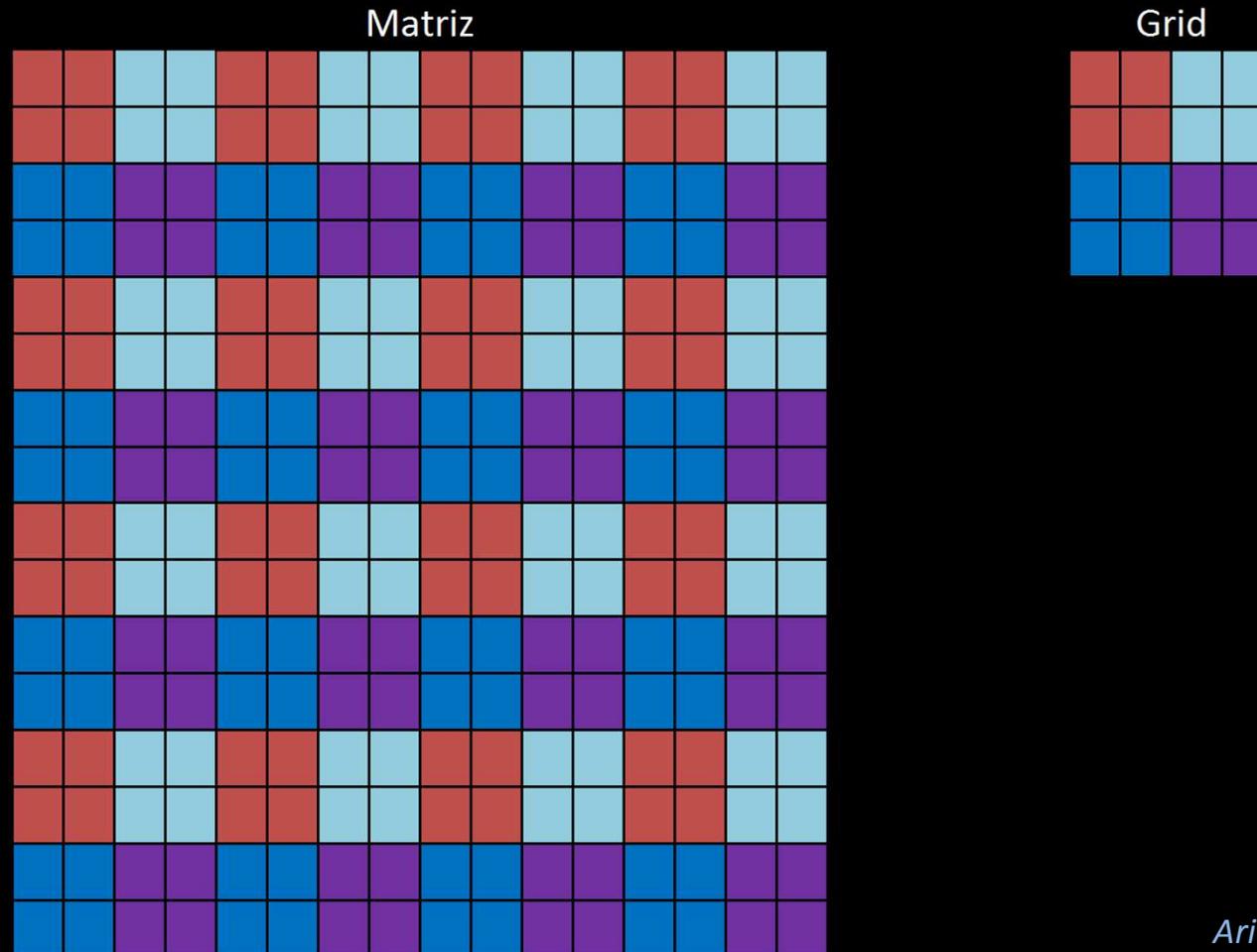
Ejercicio

- Hacer un programa genérico que sume dos matrices cuadradas. Utilizar manejo de errores y medir tiempo con diferentes configuraciones de ejecución



Ejercicio

- Hacer un programa genérico que sume dos matrices cuadradas. Utilizar manejo de errores y medir tiempo con diferentes configuraciones de ejecución



4 Manejo de Matrices

Ejemplo

- Programa que transpone matrices pequeñas

Manejo de Matrices

Ejercicio

- Hacer un programa genérico que realice la transpuesta de una matriz. Utilizar manejo de errores y medir tiempo con diferentes configuraciones de ejecución

*Ariel Ulloa - Laura
Sandoval-Karen Sáenz*