UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
Facultad de Ingeniería

# Introducción a Cómputo Paralelo con CUDA C/C++

## Laboratorio de Intel para la Academia y Cómputo de Alto Desempeño

Elaboran: Ariel Ulloa Trejo
Jaime Beltrán Rosales
Revisión: Ing. Laura Sandoval Montaño

# Temario

1. Antecedentes
2. El GPU
3. Modelo de programación CUDA
4. Manejo de matrices
5. Memoria compartida

*Ariel Ulloa Trejo*
*Jaime Beltrán Rosales*

# 4 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 *cudaSucces*. 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
```

*Ariel Ulloa Trejo*
*Jaime Beltrán Rosales*

# 4 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 *cudaDeviceSynchronice().*


**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.

*Ariel Ulloa Trejo*
*Jaime Beltrán Rosales*

# 4 Manejo de Matrices

**Ejemplo**

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

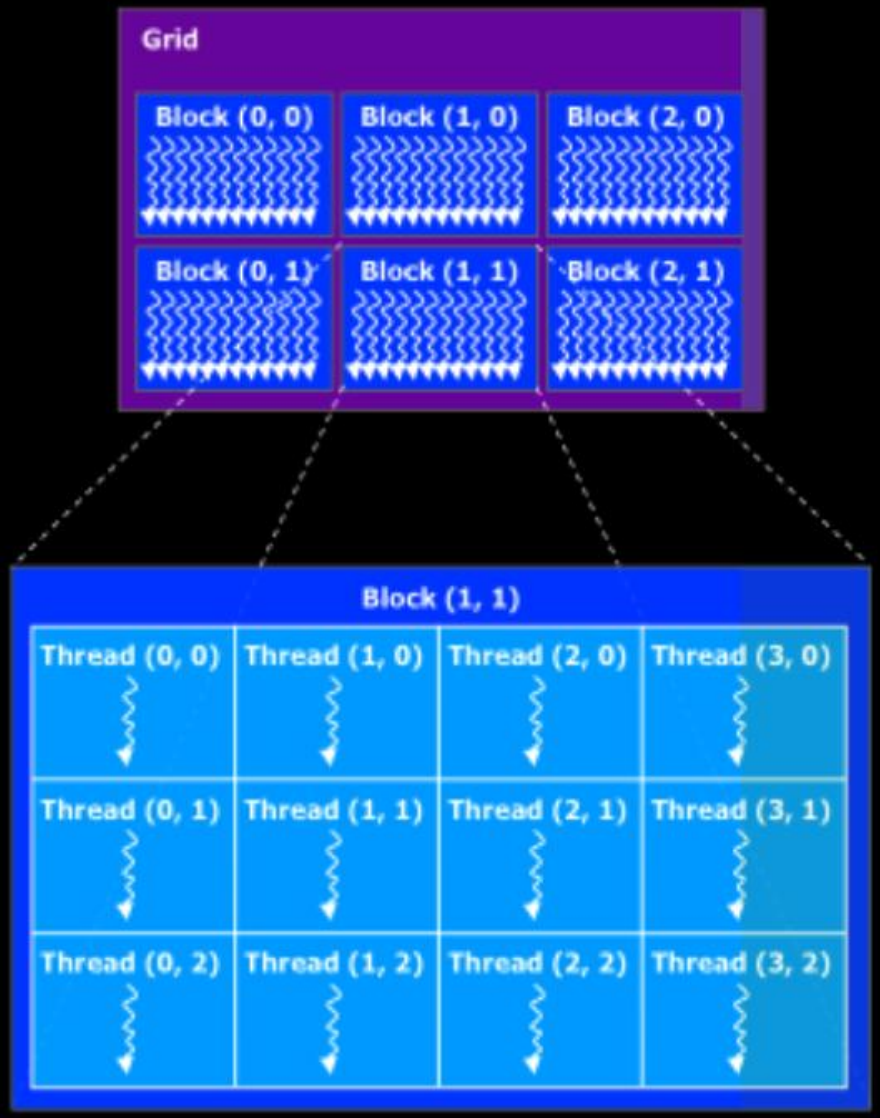
**05_errores_01.cu**

*Ariel Ulloa Trejo*
*Jaime Beltrán Rosales*

# 4 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).




*Ariel Ulloa Trejo*
*Jaime Beltrán Rosales*

# 4 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.***

*Ariel Ulloa Trejo*
*Jaime Beltrán Rosales*

# 4 Manejo de Matrices

**Ejemplo**

- Programa que suma dos matrices cuadradas.

*Ariel Ulloa Trejo*
*Jaime Beltrán Rosales*

# 4 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)

*Ariel Ulloa Trejo*
*Jaime Beltrán Rosales*

- ***Solución: Para asignar los índices de cada elemento de la matriz a los hilos:***

```
int bx = blockIdx.x;
int by = blockidx.y;
int tx = threadIdx.x;
int ty = threadIdx.y;
int dy = blockDim.y;
int dx = blockDim.x;
d_a[ (by * dy + ty) * gridDim.x * blockDim.x + (bx * dx + tx) ];
```


```
d_a[ (0*3+0)*6+(0*3+0)] = a[0]
d_a[ (0*3+0)*6+(1*3+0)] = a[3]
d_a[ (1*3+0)*6+(0*3+0)] = a[18]
d_a[ (1*3+0)*6+(1*3+0)] = a[21]
```

**06_matrices_01.cu**

*Ariel Ulloa Trejo*
*Jaime Beltrán Rosales*

# 4 Manejo de Matrices

- ***Medir tiempos en el GPU***
- *Para medir los tiempos que toman las actividades en el GPU, se utilizan los siguientes métodos:*

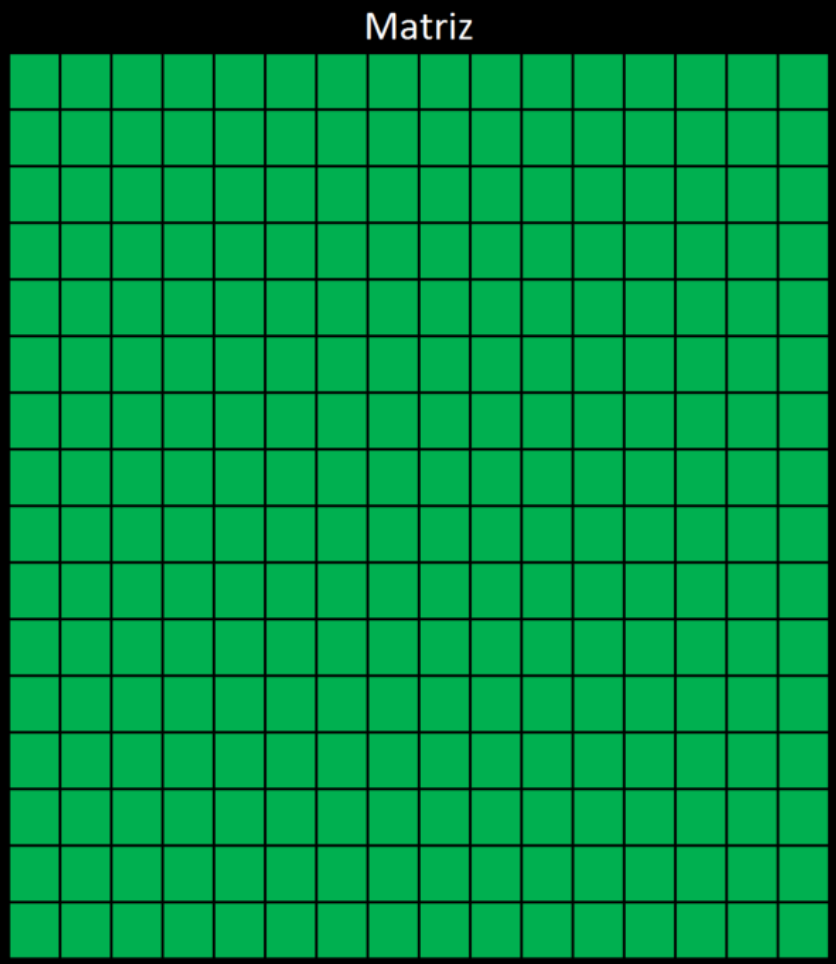
| | |
|---|---|
| • *cudaEventCreate()* | *Crea las variables* |
| • *cudaEventRecord()* | *Realiza el inicio y parada* |
| • *cudaEventDestroy()* | *Libera variables* |
| • *cudaEventElapsedTime()* | *Obtiene dif. de tiempos* |

**GpuTimer.h**

*Ariel Ulloa Trejo*
*Jaime Beltrán Rosales*

## Ejercicio

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

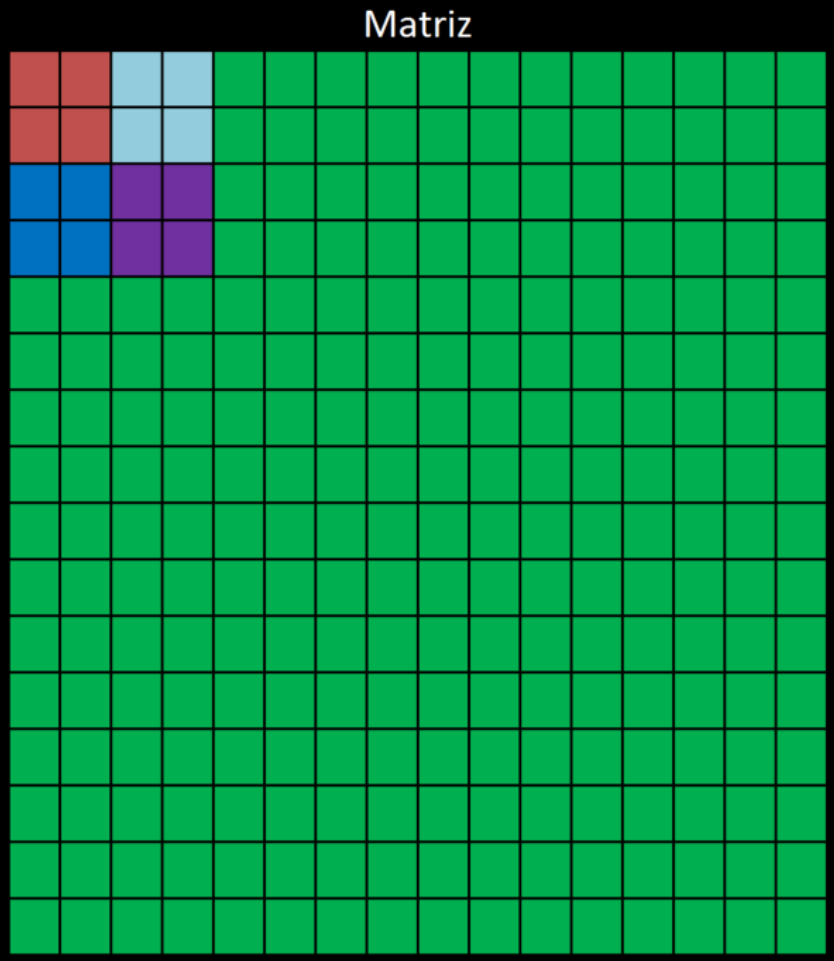




*Ariel Ulloa Trejo*
*Jaime Beltrán Rosales*

## Ejercicio

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

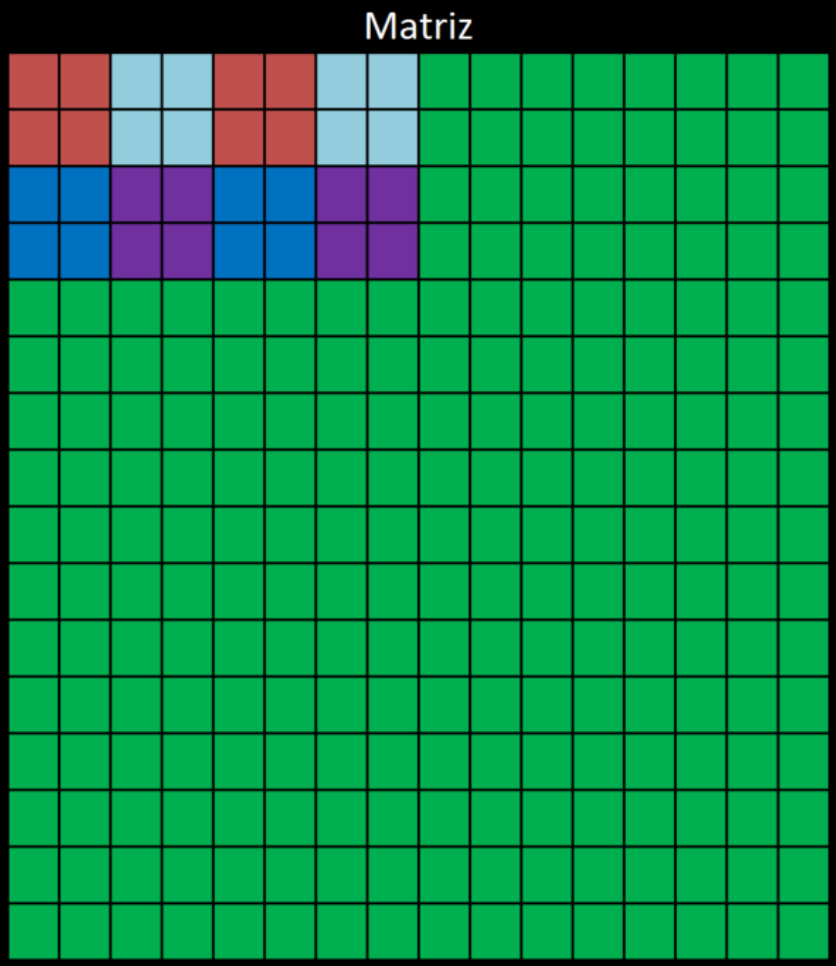




*Ariel Ulloa Trejo*
*Jaime Beltrán Rosales*

## Ejercicio

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

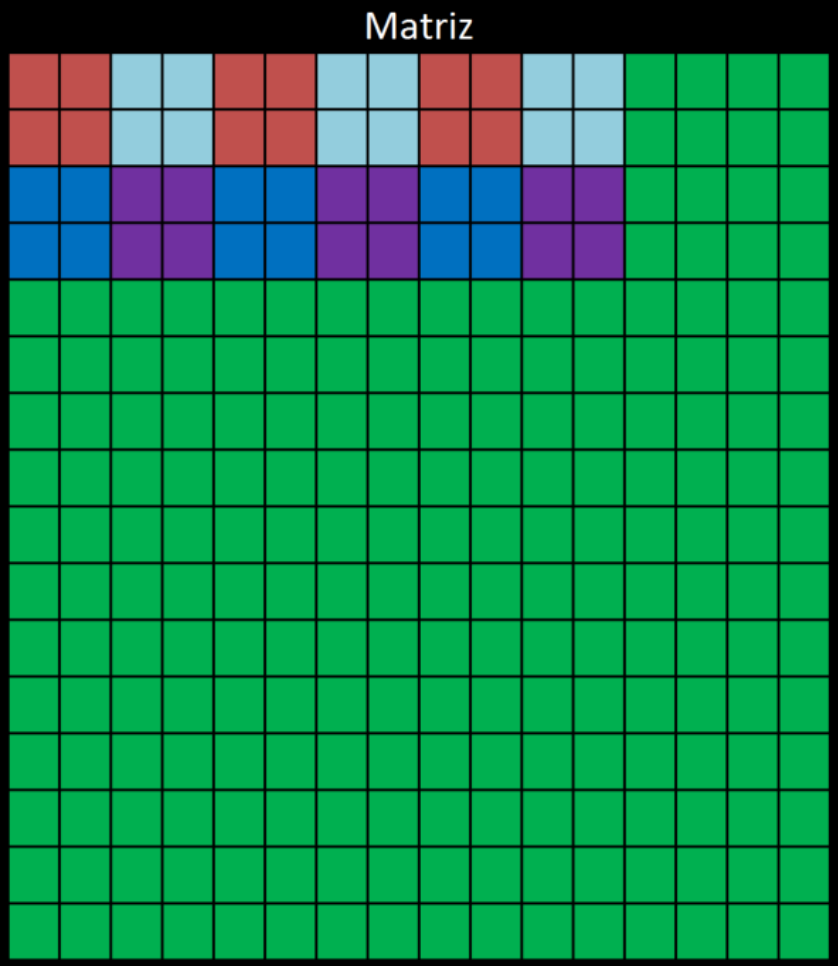


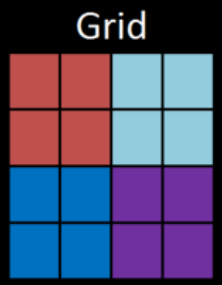


*Ariel Ulloa Trejo*
*Jaime Beltrán Rosales*

## Ejercicio

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

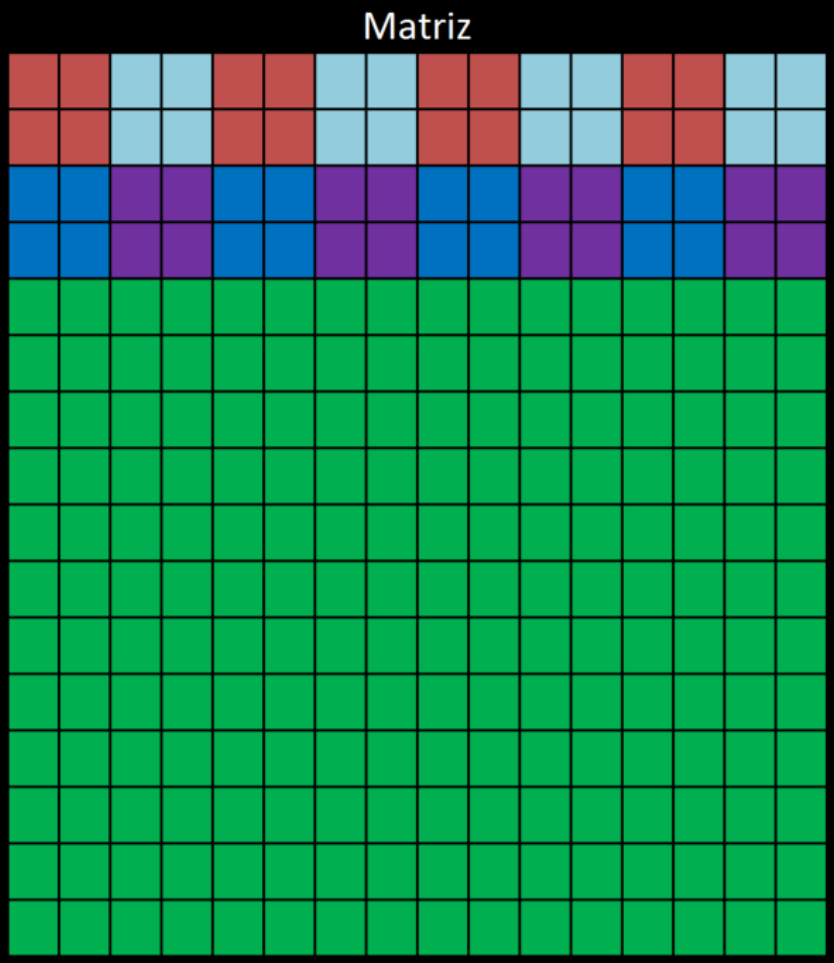



*Ariel Ulloa Trejo*
*Jaime Beltrán Rosales*

## Ejercicio

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

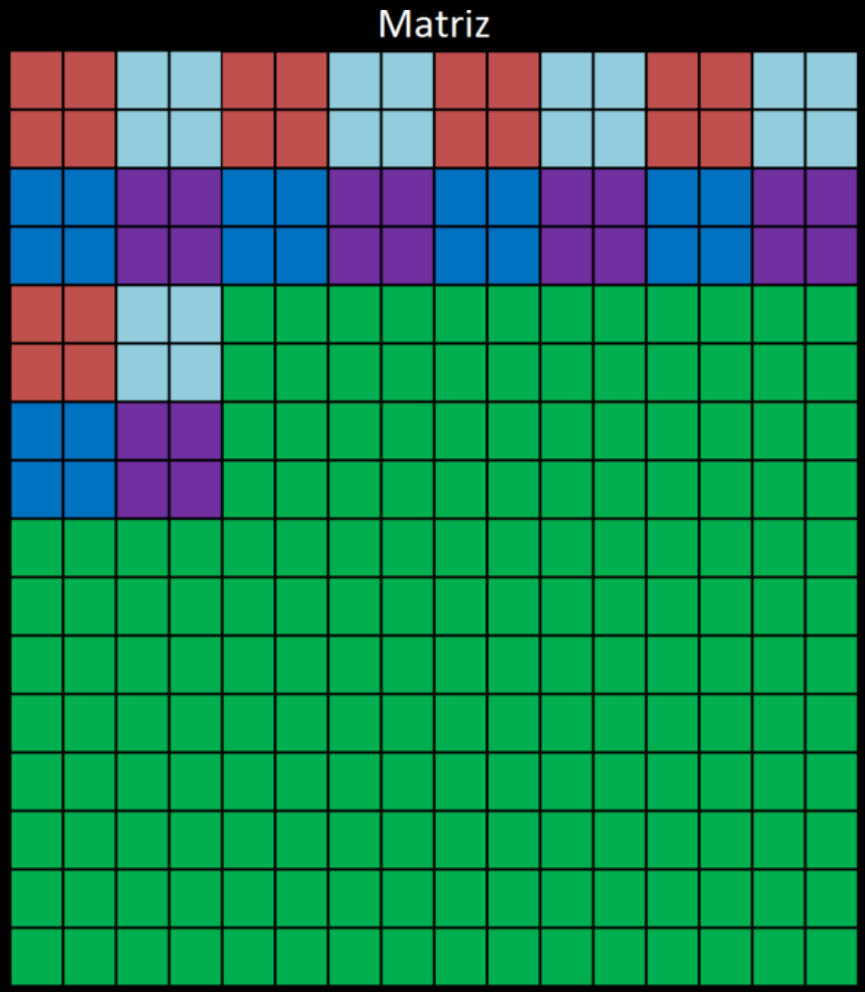


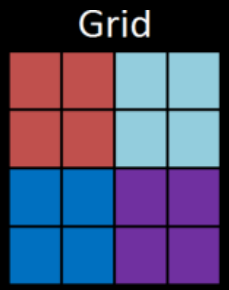


*Ariel Ulloa Trejo*
*Jaime Beltrán Rosales*

## Ejercicio

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

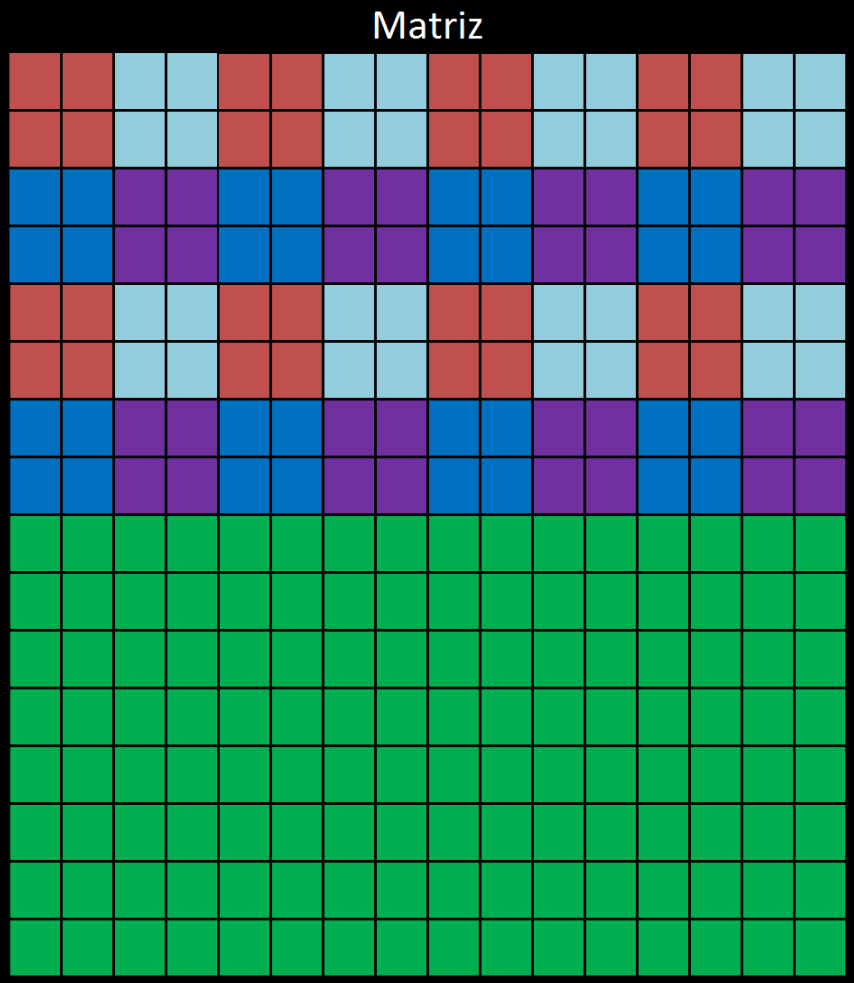


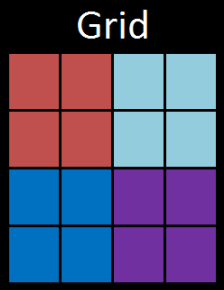




*Ariel Ulloa Trejo*
*Jaime Beltrán Rosales*

## Ejercicio

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

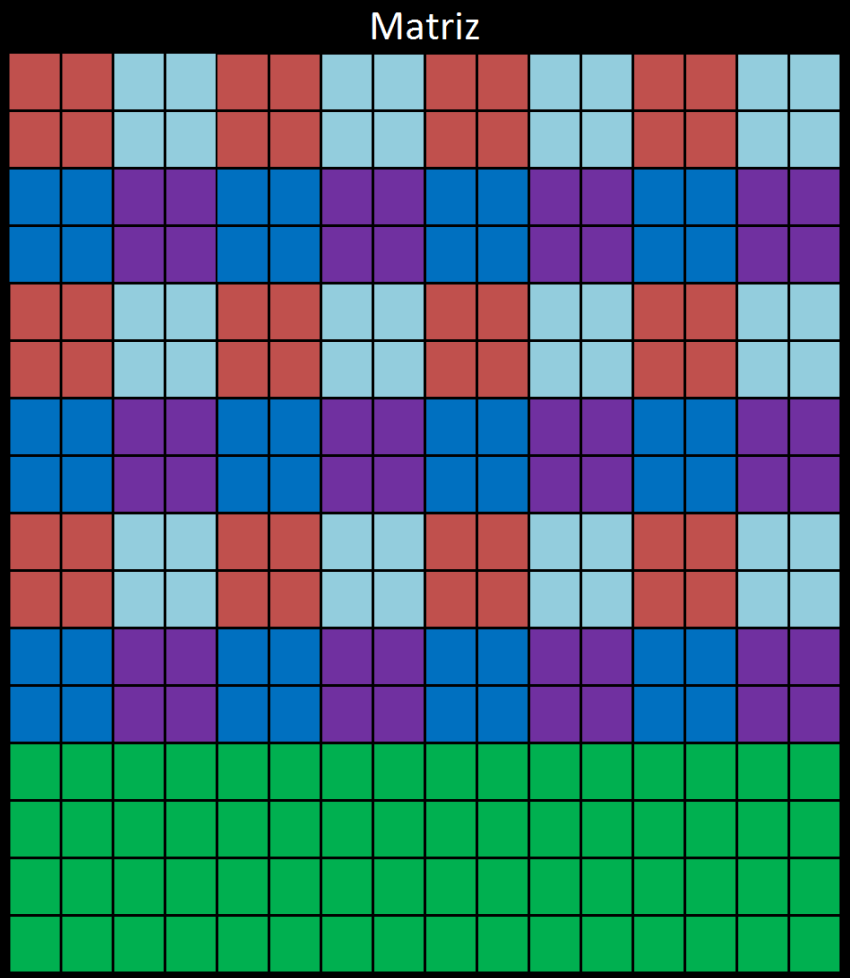


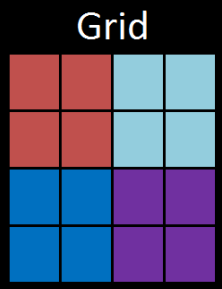




*Ariel Ulloa Trejo*
*Jaime Beltrán Rosales*

## Ejercicio

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

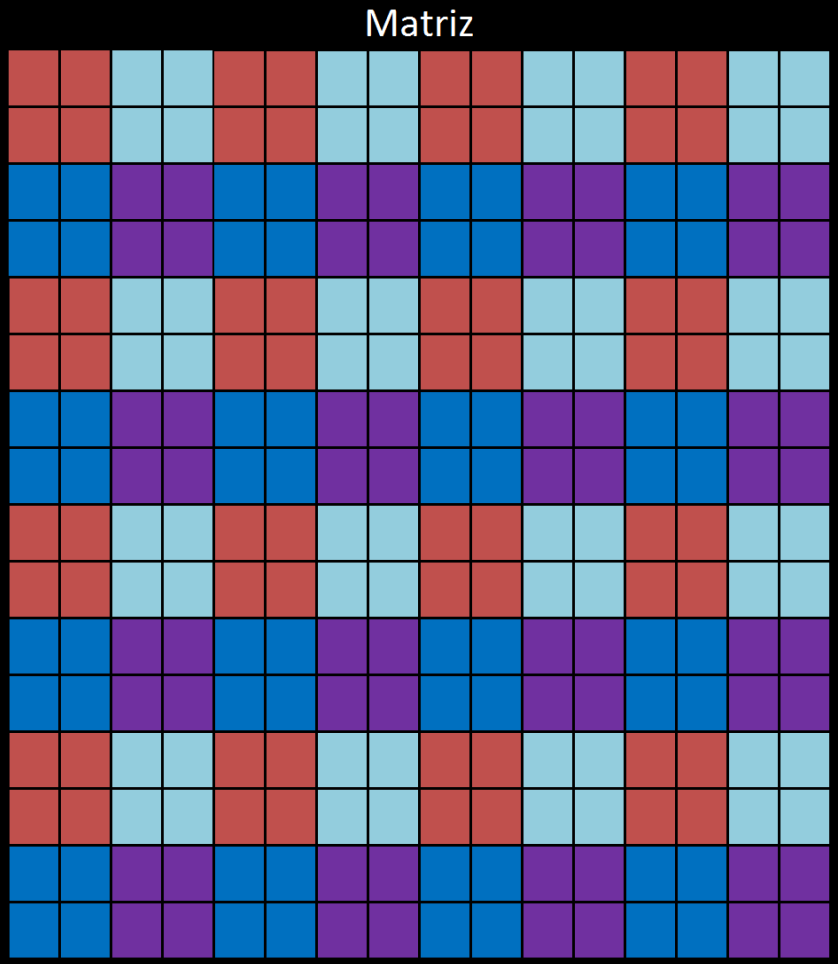


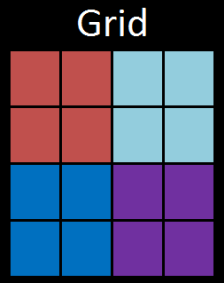




*Ariel Ulloa Trejo*
*Jaime Beltrán Rosales*

## Ejercicio

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




*Ariel Ulloa Trejo*
*Jaime Beltrán Rosales*

# 4 Manejo de Matrices

**Ejemplo**

- Programa que transpone matrices pequeñas

*Ariel Ulloa Trejo*
*Jaime Beltrán Rosales*

# 4 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 Trejo*
*Jaime Beltrán Rosales*