

**Universidad del Valle de Guatemala**  
**Facultad de ingeniería**



**Corto 5**  
**Computación Paralela**

**Mariana David Sosa 201055**

**Guatemala 30 de agosto del 2023**

## Corriendo Secuencial

```
PS C:\Users\Mariana\Documents\Universidad\Semestre8\paralela\corto\corto> cd "c:\Users\Mariana\Documents\Universidad\Semestre8\paralela\corto\corto\" ; if ($?) { g++ secuencial.
cpp -o secuencial } ; if ($?) { .\secuencial }
Ingrese la cantidad de numeros aleatorios a generar: 100000
[EXIT050] Se han generado y escrito numeros aleatorios en el archivo randomResults/random_secuencial.txt
[EXIT050] Los numeros se han clasificado y escrito en el archivo sortedResults/sorted_secuencial.txt
Tiempo de ejecución: 20.0848 segundos
```

## Corriendo Paralelo

```
Tiempo de ejecución: 20.0848 segundos
PScd "c:\Users\Mariana\Documents\Universidad\Semestre8\paralela\corto\corto\" ; if ($?) { g++ paralelo.cpp -o paralelo } ; if ($?) { .\paralelo }
Por favor, ingrese la cantidad de numeros aleatorios a generar: 10
[EXIT050] Se han generado y escrito numeros aleatorios en el archivo randomResults/random_paralelo.txt
[EXIT050] Los numeros se han clasificado y escrito en el archivo sortedResults/sorted_paralelo.txt
Tiempo de ejecución: 6.3e-06 segundos
PS C:\Users\Mariana\Documents\Universidad\Semestre8\paralela\corto\corto> █
```

**¡Observación! Las capturas de pantalla fueron tomados luego de los cálculos por lo que no coincidieron con los de la tabla**

## Resultados

Resultados de pruebas			
Cantidad de numeros aleatorios generados	Tiempo de secuencial (segundos)	Tiempo de paralelo (segundos)	SpeedUp (segundos)
10	0.0292134	4.09e-05	0.7127
100	0.0281204	5.2e-06	1,30E-06
1000	0.0376263	0.0005283	1,32E-04
10000	0.296464	0.0049593	1,24
100000	20.296.465	0.0512434	0,0126

SpeedUp Formula:

Speed-up =  $1 / (F / x + (1-F))$  Tiempo de ejecución con mejora = tiempo de ejecución sin mejora \* F / x + tiempo de ejecución sin mejora \* (1-F)

## Discusión

Los resultados de las pruebas muestran claramente la efectividad del enfoque paralelo en la mejora del rendimiento en comparación con la ejecución secuencial. A medida que aumenta la cantidad de números aleatorios generados, el speedup se vuelve más evidente. Para 10 números aleatorios, el speedup es modesto pero aún presente, mientras que para 1000 y 10000 números aleatorios, el programa paralelo logra una mejora significativa en el rendimiento. Sin embargo, en el caso de 100000 números aleatorios, el speedup disminuye notablemente debido a la creciente complejidad de administrar un gran número de tareas paralelas. Estos resultados subrayan la importancia de considerar la naturaleza de la tarea y la cantidad de recursos disponibles al decidir si la programación paralela es la solución adecuada para un problema específico.