

# Informe de Comparación y análisis

## Integrantes:

Sergio Rojas 202273619-4

Martin Ferrera 202273552-k

## Especificaciones de los equipos de prueba:

PC1:

Procesador: Ryzen 5 8500G (6 Núcleos y 12 Hilos) @ 3.60GHz

RAM: 32GB

SO: Windows 11

Arquitectura: 64 bits

PC2:

Procesador: 11th Gen Intel(R) Core(TM) i5-11320H (4 Núcleos y 8 Hilos) @ 3.20GHz

RAM: 8GB

SO: Windows 11

Arquitectura: 64 bits

## Análisis

1. ¿Cuál de las dos implementaciones tuvo un mejor rendimiento en términos de tiempo de ejecución? ¿A qué crees que se debe esto?

En nuestro caso nos topamos con que en el computador que tiene una CPU con menos hilos el Fork fue más rápido, mientras que en el computador que contaba con una CPU de más hilos el uso de Threads fue más rápido. Podemos atribuir esto a la mayor cantidad de hilos de una CPU sobre otra (un 50% más)

## **2. ¿Se observó alguna diferencia significativa en el uso de recursos del sistema (CPU, RAM) entre ambas soluciones?**

Se notó un gran aumento de uso de Memoria en la técnica de Forks + Mensajería (alrededor de 4 a 6 veces más de RAM dependiendo del computador), esto puede ser debido a la necesidad de implementar los pipes para la comunicación entre procesos

## **3. ¿En qué escenarios consideras más adecuado el uso de procesos (forks) frente a hilos (threads)?**

La primera limitante tendría que ser el hardware y el presupuesto, si tenemos pocos hilos hay que priorizar a forks, ya que es más barato comprar memoria frente a un cambio de la CPU

## **4. ¿Qué estrategias de optimización aplicarías al programa con menor rendimiento para que iguale o supere la eficiencia del programa mejor evaluado?**

Tendremos que tomar cuenta cuál es el que tiene peor rendimiento:

- Si fork tiene peor rendimiento:
  - Implementar memoria compartida para disminuir la carga sobre la RAM
  - Limitar los procesos a los núcleos físicos, no a los lógicos
- Si Threads tiene peor rendimiento:
  - Ajustar la afinidad de los hilos
  - Modificar la localidad de los datos para que los registros que se carguen en caché sean más eficientes al no tener que cargar de más (transponer la matriz y que los bloques de memoria que vayan a la caché contengan solo información útil (Gracias Algoco))