

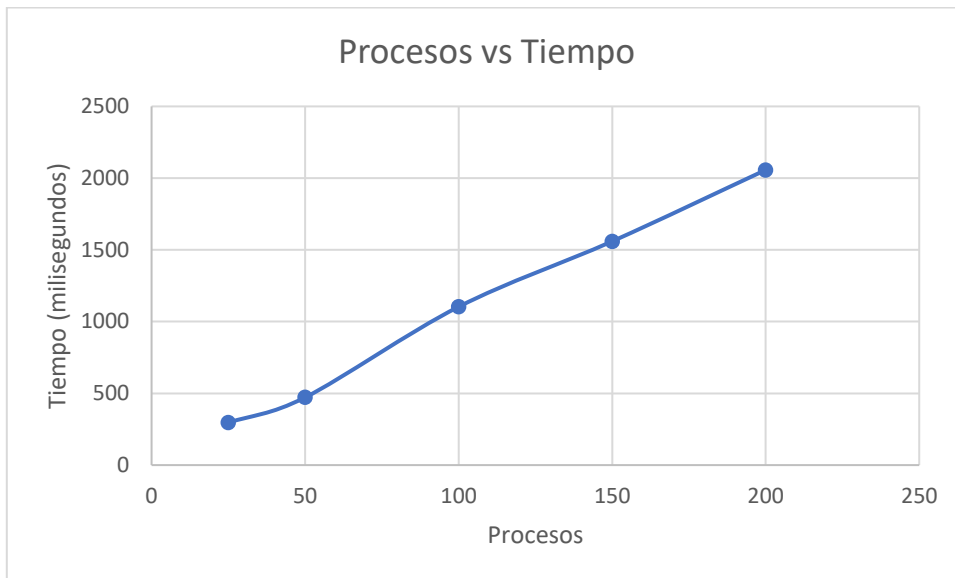
Algoritmos y estructuras de datos
Douglas Barrientos
Sección 10

HDT5

Adair Velásquez - 24596
Emilio Chen - 24841
Marcela Castillo - 24952

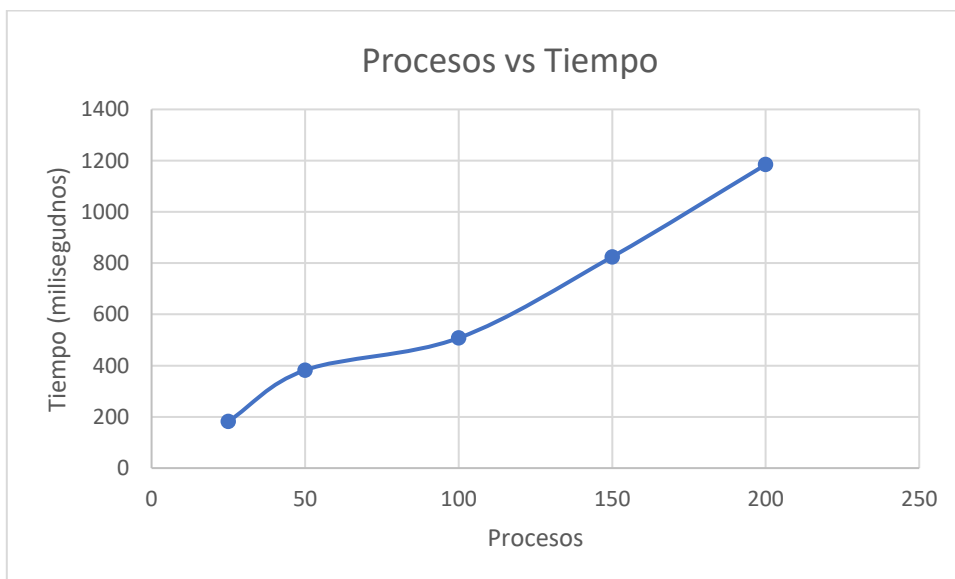
Link de github: <https://github.com/MarceC21/HDT5>

Grafica No. 1 Intervalos de 10



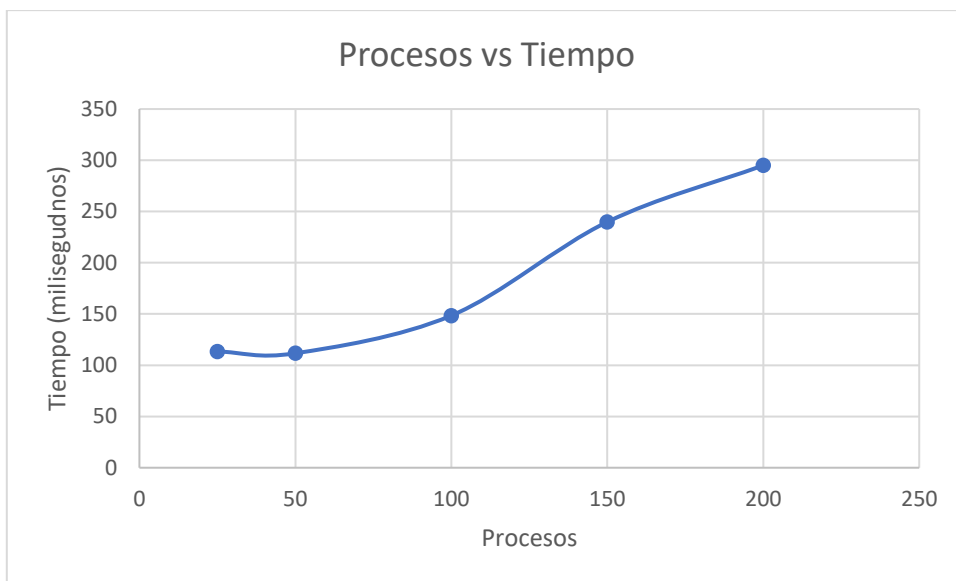
Desviación estándar: **736.41**

Grafica No.2 Intervalos de 5



Desviación estándar: **394.12**

Gráfica No. 3 Intervalos de 1



Desviación estándar: **81.98**

PARTE 2

i. incrementar la memoria a 200

No. Procesos	Resultado
1	Tarda mucho más
5	Tarda menos que antes
10	Tarda solo un poco mas

ii. luego con poner la memoria nuevamente a 100, pero tener un procesador más rápido)

No. Procesos	Resultado
1	Tarda casi un 50% menos
5	Tarda solo un poco menos
10	Tarda más que antes

iii. luego regrese a la velocidad normal procesador, pero emplee 2 procesadores

No. Procesos	Resultado
1	Tarda casi un 60 % menos
5	Tarda la mitad
10	Tarda solo un poco más

Analisis:

La diferencia en el tiempo de ejecución según el número de procesadores y el intervalo de llegada de procesos se debe a cómo se distribuye la carga de trabajo y a los efectos de congestión en el sistema.

Cuando los intervalos de llegada son pequeños (por ejemplo, 1 y 5), los procesos llegan rápidamente y se acumulan en la cola de espera. En este caso, contar con más procesadores permite distribuir mejor la carga, ejecutando múltiples procesos en paralelo y reduciendo significativamente el tiempo total de simulación. La mejora es notable porque el cuello de botella principal es la disponibilidad del CPU.

Sin embargo, cuando los intervalos de llegada son grandes (como 10), los procesos llegan más espaciados en el tiempo. Esto significa que incluso con un solo procesador, hay suficiente tiempo para ejecutar cada proceso antes de que llegue el siguiente. Como resultado, aunque agregar más procesadores sigue mejorando el rendimiento, la diferencia ya no es tan marcada, ya que el sistema no se congestiona tanto.