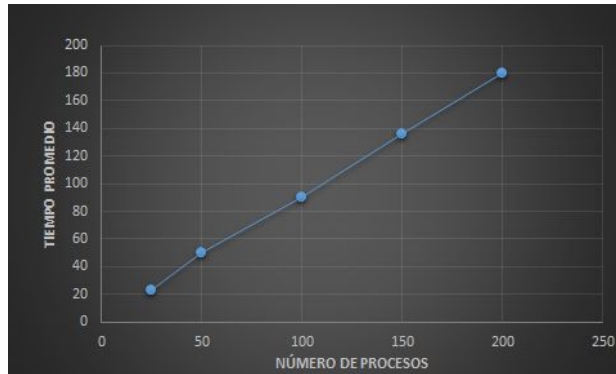


Hoja de Trabajo #5

a. Procesos normales (Intervalo=10/Memoria=100/Capacidad=1)

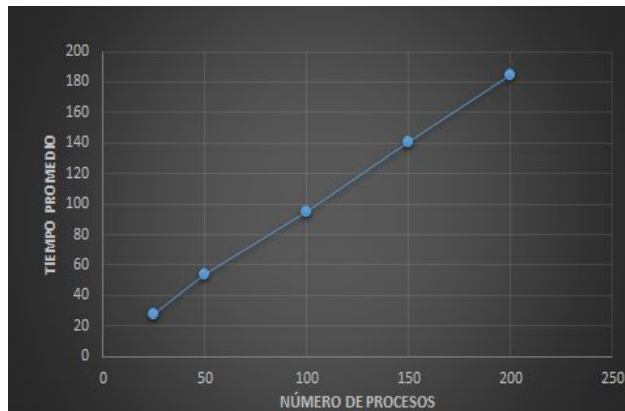
Gráfica #1 Simulación en intervalos de 1



# de Procesos	Tiempo Promedio	Desviación Estándar
25	23.232159	11.648551
50	49.794781	23.497957
100	90.700136	43.870230
150	136.348197	71.898795
200	179.798661	96.809080

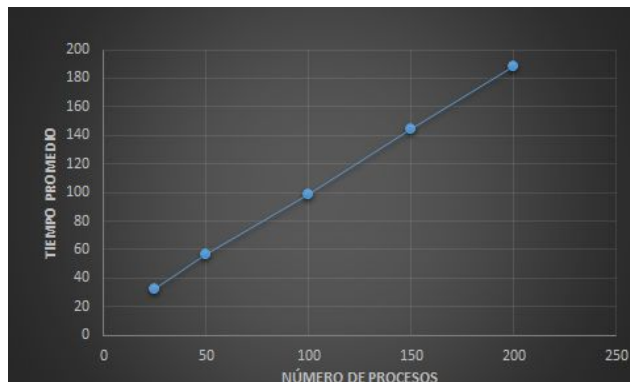
b. Acelerando procesos (Intervalo=5 y 1 /Memoria=100/Capacidad=1)

Gráfica #2 Simulación en intervalos de 5



# de Procesos	Tiempo Promedio	Desviación Estándar
25	28.056145	13.218466
50	53.764057	25.484020
100	95.111735	47.243407
150	140.722987	75.593394
200	184.520997	100.642577

Gráfica #3 Simulación en intervalos de 1

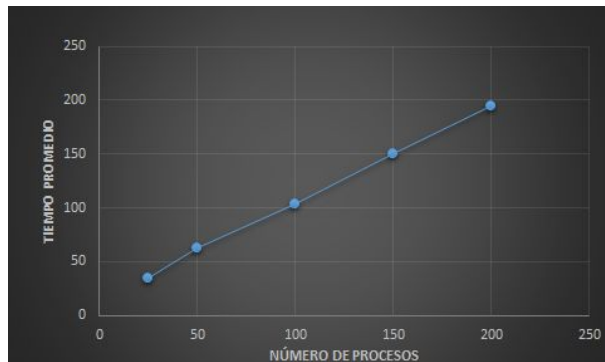


# de Procesos	Tiempo Promedio	Desviación Estándar
25	32.059229	14.518825
50	56.886145	27.680356
100	98.579680	50.120818
150	144.222820	78.594780
200	188.298866	103.738537

c. Reduciendo Tiempo Promedio

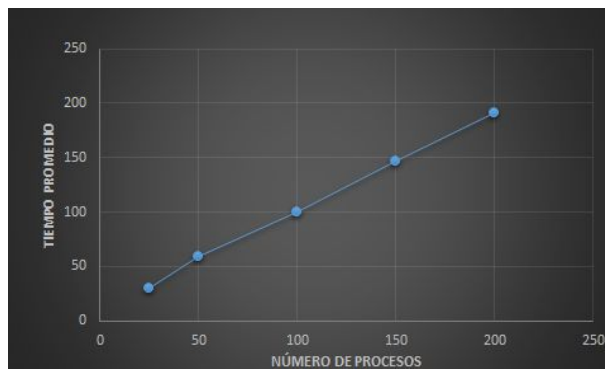
- Memoria Ram=200 , 3 instrucciones por tiempo, capacidad=1

Gráfica #4 Simulación en intervalos de 1



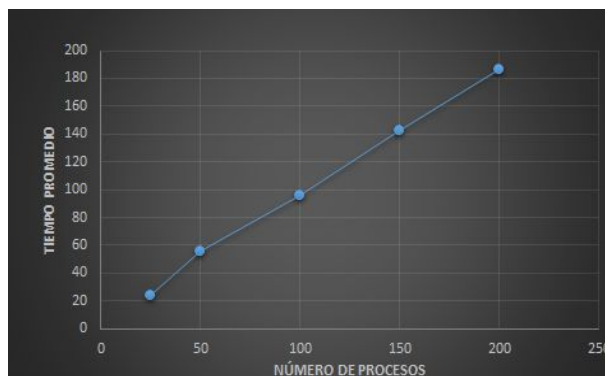
# de Procesos	Tiempo Promedio	Desviación Estándar
25	34.099229	15.946198
50	63.052811	28.746478
100	103.746347	49.518749
150	150.640597	77.702672
200	194.660533	103.091136

Gráfica #5 Simulación en intervalos de 5



# de Procesos	Tiempo Promedio	Desviación Estándar
25	29.602812	14.449546
50	59.470724	28.298884
100	100.205068	47.166482
150	147.102987	74.872792
200	190.882664	100.108107

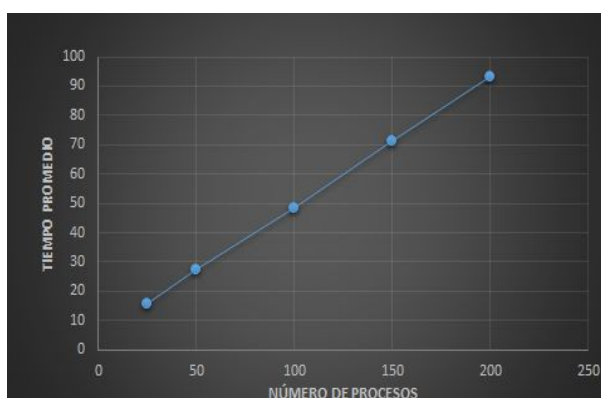
Gráfica #6 Simulación en intervalos de 10



# de Procesos	Tiempo Promedio	Desviación Estándar
25	23.578825	12.241026
50	55.301448	27.461178
100	95.646803	44.512306
150	142.668197	71.442358
200	186.128661	96.465652

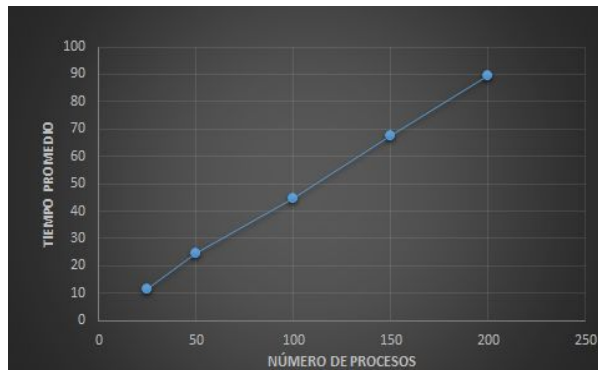
- Memoria Ram=100 , 6 instrucciones por tiempo, capacidad=1

Gráfica #7 Simulación en intervalos de 1



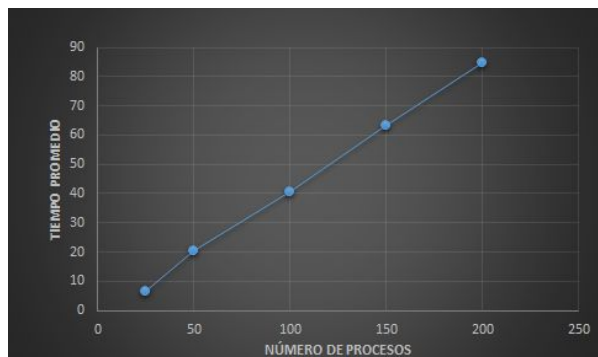
# de Procesos	Tiempo Promedio	Desviación Estándar
25	15.472562	6.936337
50	27.436145	14.317257
100	48.341347	25.202880
150	71.077264	39.379180
200	93.086366	51.801994

Gráfica #8 Simulación en intervalos de 5



# de Procesos	Tiempo Promedio	Desviación Estándar
25	11.542746	5.702415
50	24.367391	12.171354
100	44.873401	22.333443
150	67.577432	36.380464
200	89.308497	48.705894

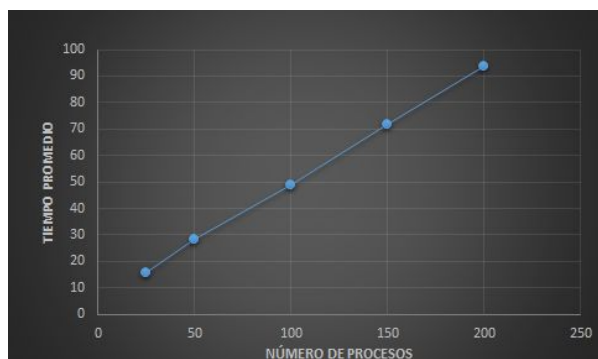
Gráfica #9 Simulación en intervalos de 10



# de Procesos	Tiempo Promedio	Desviación Estándar
25	6.588974	4.186393
50	20.626224	10.523284
100	40.538469	19.081756
150	63.202641	32.758351
200	84.586161	44.914367

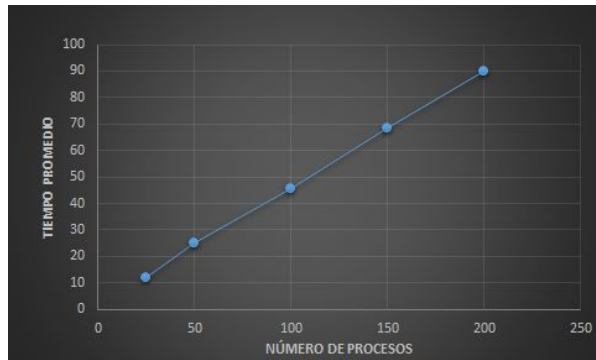
- Memoria Ram=100 , 3 instrucciones por tiempo, capacidad=2

Gráfica #10 Simulación en intervalos de 1



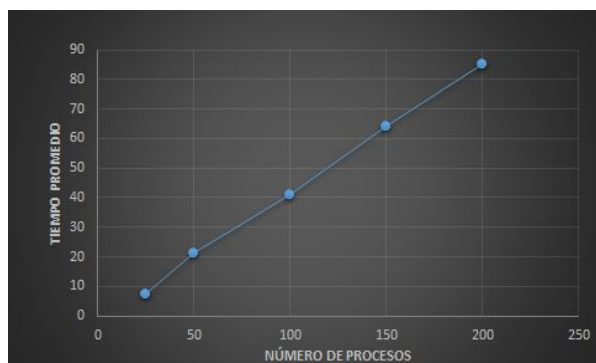
# de Procesos	Tiempo Promedio	Desviación Estándar
25	15.705616	6.975594
50	28.182344	13.575007
100	49.000268	24.667371
150	71.817296	38.921932
200	93.800078	51.509900

Gráfica #11 Simulación en intervalos de 5



# de Procesos	Tiempo Promedio	Desviación Estándar
25	11.891115	5.879793
50	25.080649	11.696832
100	45.548009	21.865143
150	68.322036	35.952152
200	90.027056	48.434975

Gráfica #12 Simulación en intervalos de 10



# de Procesos	Tiempo Promedio	Desviación Estándar
25	7.464010	4.638495
50	21.333056	10.253400
100	41.139035	18.836287
150	63.952961	32.375238
200	85.310778	44.674494

Conclusión:

Al buscar reducir el tiempo promedio el método menos eficiente, al ver las gráficas y tablas, fue el de solamente incrementar la memoria RAM. Por otro lado al comparar los resultados cuando se dejó la memoria RAM en 100 pero se modificó para que se ejecutarán 6 instrucciones por tiempo y cuando se dejó la memoria en 100 y en 3 instrucciones por ejecución pero con dos procesadores, se obtuvieron resultados muy similares y cercanos unos de los otros. Sin embargo al ver detalladamente los gráficos se puede concluir que el menor tiempo de ejecución se obtuvo al tener una memoria RAM= 100 y ejecutar 6 instrucciones por unidad de tiempo. Al analizar también los cortos tiempos al ejecutar con 2 procesadores se infiere que si se usaran 6 instrucciones por tiempo y 2 procesadores, los tiempos promedio serían significativamente menores. También se concluye que en las 3 formas de ejecutar el programa el menor tiempo se obtuvo al hacerlo en intervalos de 10 por lo que la mejor estrategia para reducir los tiempos sería utilizar intervalos grandes para no saturar al procesador y obtener un procesador de alta velocidad, que pueda procesar una mayor cantidad de instrucciones por unidad de tiempo de la forma más eficiente posible, junto a una RAM que soporte estas características.