PRIMER PARCIAL HPC

MULTIPLICACION DE MATRICES TAMAÑO NXM CON CUDA

ALUMNO: JOSE WILSON CASTAÑO MARIN

PROFESOR: JHON OSORIO RIOS

DESARROLLAR LOS SIGUIENTES PUNTOS:

1. Crear un programa que implemente la técnica de tiling en el algoritmo de multiplicación de matrices y que soporte la multiplicación de matrices de cualquier tamaño.

Se realizó la implementación respectiva y se encuentra en el repositorio https://github.com/wilcasmar/hpcwilcas.

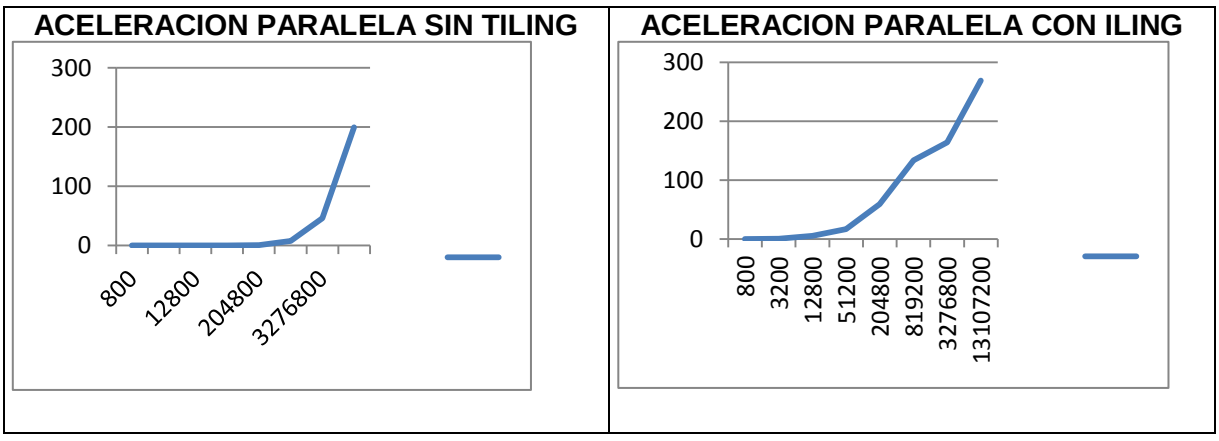
2. Verificar el correcto funcionamiento del algoritmo, tanto en su versión secuencial como paralela con tiling y sin tiling.

Se realizaron pruebas de escritorio y en la plataforma adaptada para pruebas del laboratorio sirius. (http://judge.utp.edu.co:3000)

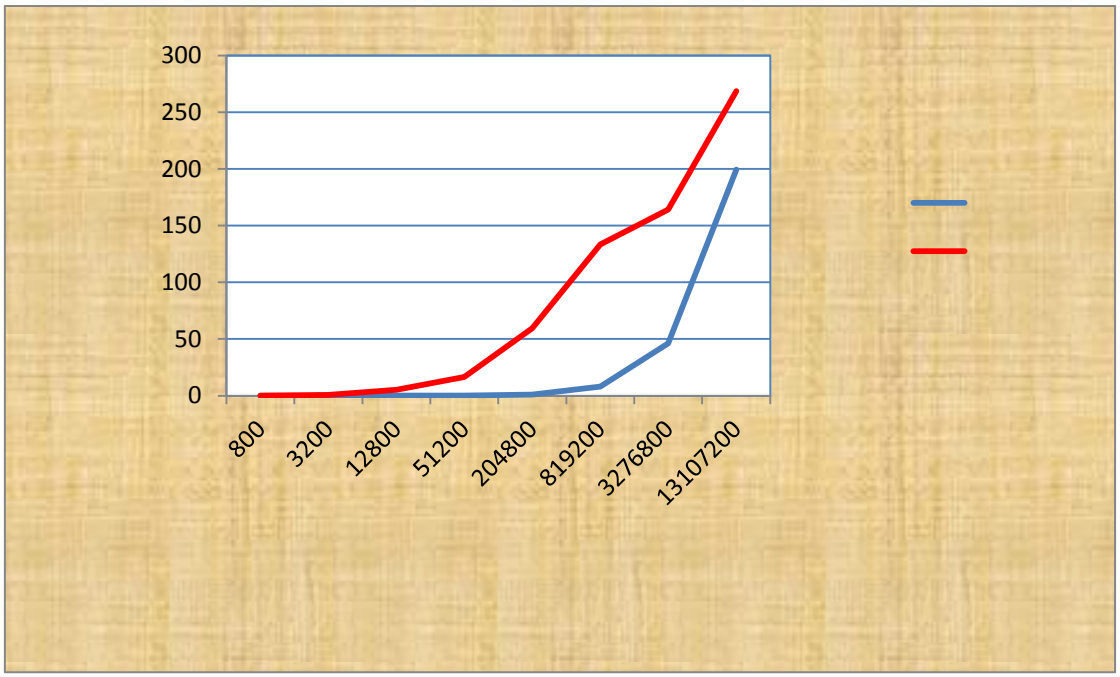
3. Tomar tiempos de ejecución para matrices de diferentes tamaños en las diferentes versiones del algoritmo. (Secuencial, paralelo sin tiling y paralelo con tiling).

Matriz A [n][m] B[m][o] usando Enteros						
tamaño matriz resultante	Matriz Dim C [n][m][o]	SECUENCIAL	PARALELO SIN TILING	PARALELO CON TILING	ACELRACION SIN TILING	ACELERACION CON TILING
800	20 X4X40	0,000011	0,105713	0,000474	0,000104055	0,023206751
3200	40X16X80	0,00028	0,097754	0,000475	0,002864333	0,589473684
12800	80 X32x160	0,002566	0,097498	0,000497	0,026318489	5,162977867
51200	160X64X320	0,011248	0,092418	0,00068	0,121707892	16,54117647
204800	320X128X640	0,088003	0,098177	0,001482	0,89637084	59,38124157
819200	640X256X1280	0,765438	0,096619	0,005739	7,922230617	133,374804
3276800	1280X512X2560	6,12594	0,13244	0,0373	46,25445485	164,2343164
13107200	2560X1024X5120	74,6621	0,374253	0,277837	199,49633	268,7262676

4. Realizar **GRÁFICAS** de aceleración donde se compare cada par de implementaciones.



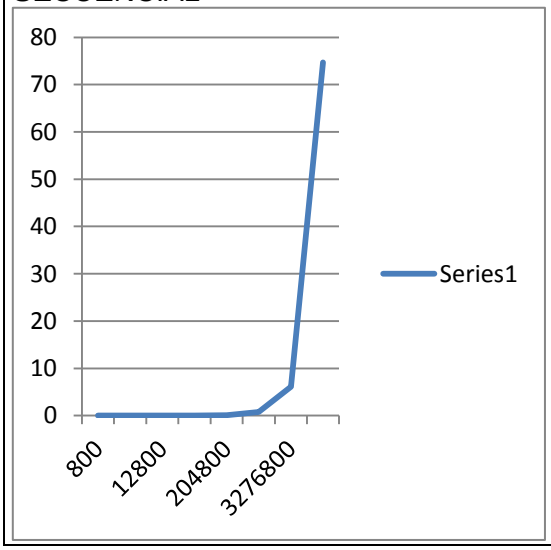
COMPARACION DE ACELERACIONES CON NUMEROS ENTEROS



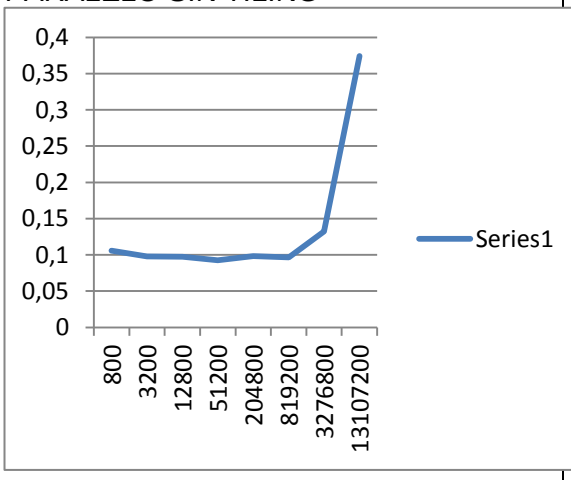
SIN TILING CON TILING

5. Realizar una **GRÁFICA** de tiempos de cada una de las implementaciones.

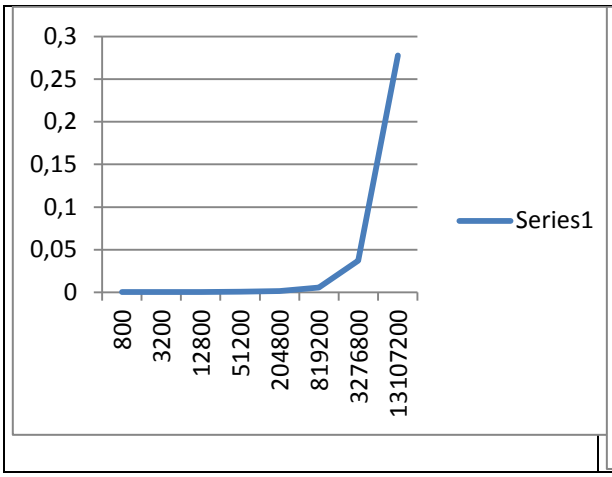
SECUENCIAL



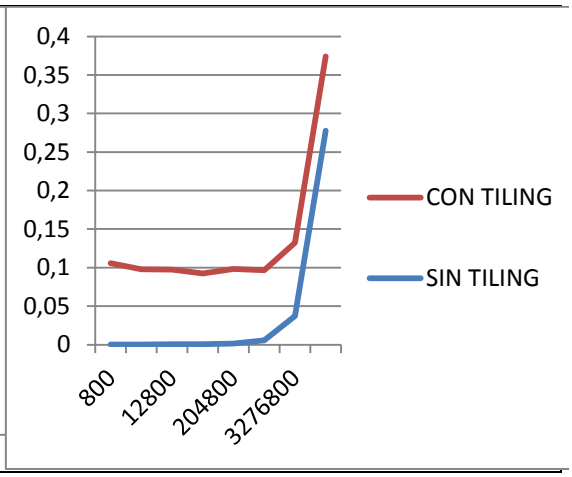
PARALELO SIN TILING



PARALELO CON TILING



PARALELO CON TILING VS SIN TILINING



6. Tomar tiempos de ejecución para diferentes tamaños de bloque (4x4, 16x16, 32x32), manteniendo el mismo número de datos, verificar el funcionamiento del algoritmo.

Matriz A [n][m] B[m][o] usando Enteros						
Bloques 4 - Ancho Tiles 4						
tamaño matriz resultante	Matriz Dim C [n][m][o]	SECUENCIAL	PARALELO SIN TILING	PARALELO CON TILING	ACELRACION SIN TILING	ACELERACION CON TILING
800	20 X4X40	1,30E-05	0,096112	0,000472	0,000135259	0,02754237
51200	160X64X320	0,01311	0,089248	0,000802	0,146894048	16,3466334
819200	640X256X1280	0,779958	0,105915	0,017704	7,363999434	44,0554677
3276800	1280X512X2560	5,89487	0,229106	0,149237	25,72988049	39,500057
13107200	2560X1024X5120	75,5396	1,262	1,17467	59,8570523	64,307082

Matriz A [n][m] B[m][o] usando Enteros						
Bloques 16 - Ancho Tiles 16						
tamaño matriz resultante	Matriz Dim C [n][m][o]	SECUENCIAL	PARALELO SIN TILING	PARALELO CON TILING	ACELRACION SIN TILING	ACELERACION CON TILING
800	20 X4X40	0,00001	0,117592	0,000455	0,0000850	0,02197802
51200	160X64X320	0,011965	0,087145	0,000633	0,137299902	18,9020537
819200	640X256X1280	0,76607	0,093465	0,005932	8,196330177	129,141942
3276800	1280X512X2560	6,50393	0,130954	0,038538	49,6657605	168,766672
13107200	2560X1024X5120	76,1612	0,38513	0,290221	197,7545244	262,424842

Matriz A [n][m] B[m][o] usando Enteros						
Bloques 32 - Ancho Tiles 32						
tamaño matriz resultante	Matriz Dim C [n][m][o]	SECUENCIAL	PARALELO SIN TILING	PARALELO CON TILING	ACELRACION SIN TILING	ACELERACION CON TILING
800	20 X4X40	0,000024	0,091488	0,000477	0,000262329	0,05031447
51200	160X64X320	0,016023	0,090813	0,000622	0,176439497	25,7604502
819200	640X256X1280	0,768682	0,098734	0,005718	7,785382948	134,431969
3276800	1280X512X2560	6,88455	0,135386	0,037238	50,8512697	184,879693
13107200	2560X1024X5120	74,5018	0,368588	0,278128	202,1275788	267,868751

BLOQUE 4 DE TILING 4 CON NUEROS DE PUNTO FLOTANTE

Matriz A [n][m] B[m][o] usando Enteros						
Bloques 4 - Ancho Tiles 4						
tamaño matriz resultante	Matriz Dim C [n][m][o]	SECUENCIAL	PARALELO SIN TILING	PARALELO CON TILING	ACELRACION SIN TILING	ACELERACION CON TILING
800	20 X4X40	2,50E-05	0,095162	0,00047	0,00026271	0,053191489
51200	160X64X320	0,010763	0,093817	0,000799	0,114723344	13,47058824
819200	640X256X1280	0,796586	0,110175	0,110175	7,230188337	7,230188337
3276800	1280X512X2560	6,56524	0,238077	0,238077	27,57612033	27,57612033
13107200	2560X1024X5120	76,9468	1,21667	1,21667	63,24377193	63,24377193

Matriz A [n][m] B[m][o] usando Enteros						
Bloques 16 - Ancho Tiles 16						
tamaño matriz resultante	Matriz Dim C [n][m][o]	SECUENCIAL	PARALELO SIN TILING	PARALELO CON TILING	ACELRACION SIN TILING	ACELERACION CON TILING
800	20 X4X40	2,50E-05	0,095209	0,000475	0,00026258	0,052631579
51200	160X64X320	0,010763	0,086672	0,000622	0,12418082	17,30385852
819200	640X256X1280	0,802478	0,092273	0,00615	8,696780207	130,4842276
3276800	1280X512X2560	6,56019	0,128203	0,04046	51,17033143	162,1401384
13107200	2560X1024X5120	78,1822	0,416158	0,306084	187,8666276	255,4272683

Matriz A [n][m] B[m][o] usando Enteros						
Bloques 32 - Ancho Tiles 32						
tamaño matriz resultante	Matrix Dim C [n][m][o]	SECUENCIAL	PARALELO SIN TILING	PARALELO CON TILING	ACELRACION SIN TILING	ACELERACION CON TILING
800	20 X4X40	0,000024	0,091488	0,000477	0,000262329	0,050314465
51200	160X64X320	0,016023	0,090813	0,000622	0,176439497	25,76045016
819200	640X256X1280	0,768682	0,098734	0,005718	7,785382948	134,4319692
3276800	1280X512X2560	5,88455	0,135386	0,037238	43,46498161	158,0254042
13107200	2560X1024X5120	74,5018	0,368588	0,278128	202,1275788	267,8687511

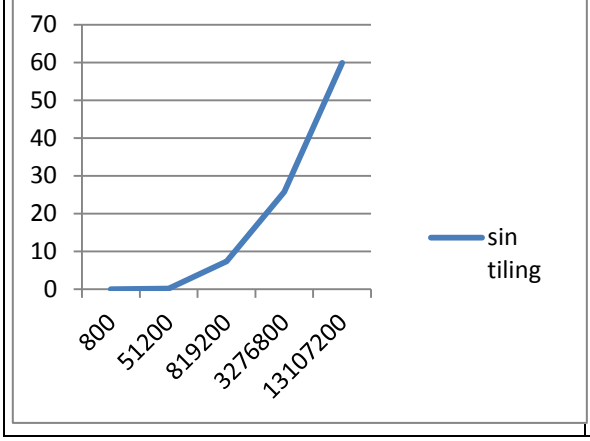
ACELERACION CON DIFERENTES TAMAÑOS DE BLOQUES Y ANCHO DE TILES

Matriz A [n][m] B[m][o] usando Enteros						
Bloques 4 - Tiles 32						
tamaño matriz resultante	Matriz Dim C [n][m][o]	SECUENCIAL	PARALELO SIN TILING	PARALELO CON TILING	ACELRACION SIN TILING	ACELERACION CON TILING
800	20 X4X40	2,50E-05	0,104437	0,000476	0,000239379	0,052521008
51200	160X64X320	0,014663	0,093082	0,000813	0,157527771	18,03567036
819200	640X256X1280	0,771372	0,10847	0,017659	7,111385637	43,68152217
3276800	1280X512X2560	6,27188	0,242251	0,149095	25,89000665	42,06633355
13107200	2560X1024X5120	74,5044	1,2684	1,17414	58,73888363	63,45444325

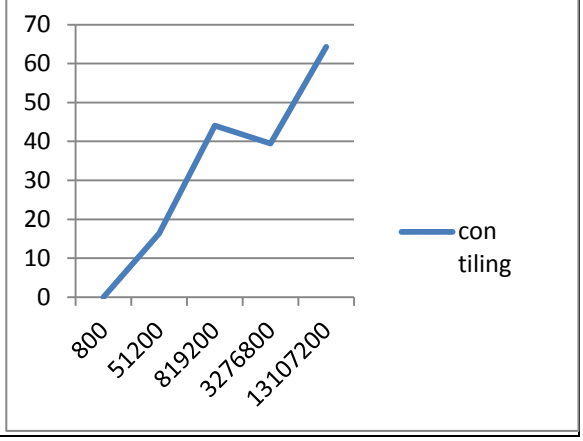
Matriz A [n][m] B[m][o] usando Enteros						
Bloques 16 - Ancho Tiles 4						
tamaño matriz resultante	Matriz Dim C [n][m][o]	SECUENCIAL	PARALELO SIN TILING	PARALELO CON TILING	ACELRACION SIN TILING	ACELERACION CON TILING
800	20 X4X40	1,10E-05	0,09637	0,000471	0,000114143	0,023354565
51200	160X64X320	0,015742	0,076793	0,000627	0,204992643	25,10685805
819200	640X256X1280	0,77621	0,09551	0,006112	8,127002408	126,9977094
3276800	1280X512X2560	6,3014	0,124621	0,03853	50,5645116	163,5452894
13107200	2560X1024X5120	74,3653	0,400662	0,289802	185,606072	256,6072698

7. Realizar **GRÁFICAS** de aceleración y de tiempos del punto anterior.

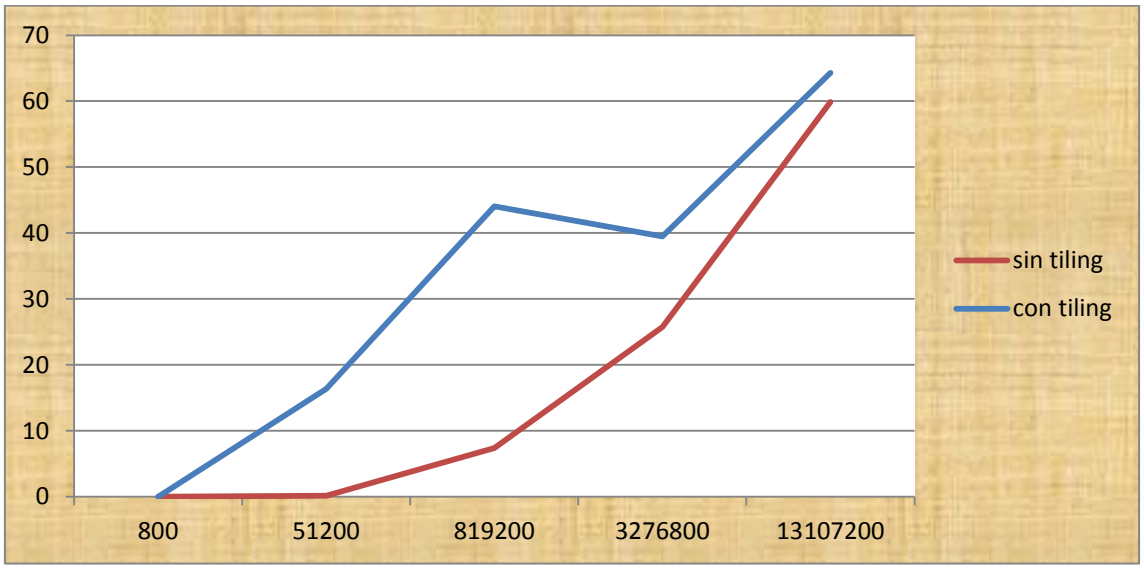
Aceleración bloques 4x4 sin tiling

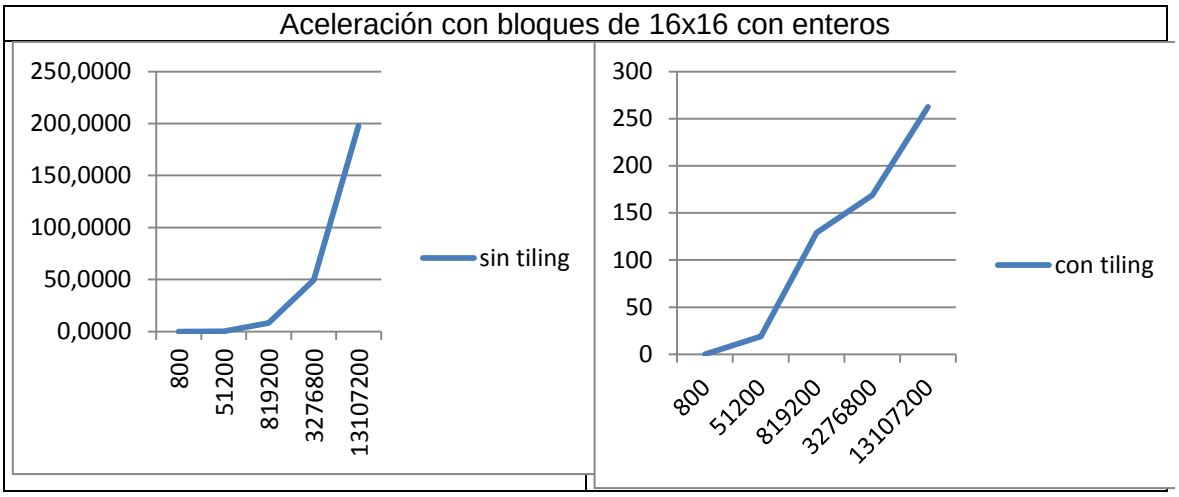


Aceleracion bloques 4x4 con tiling

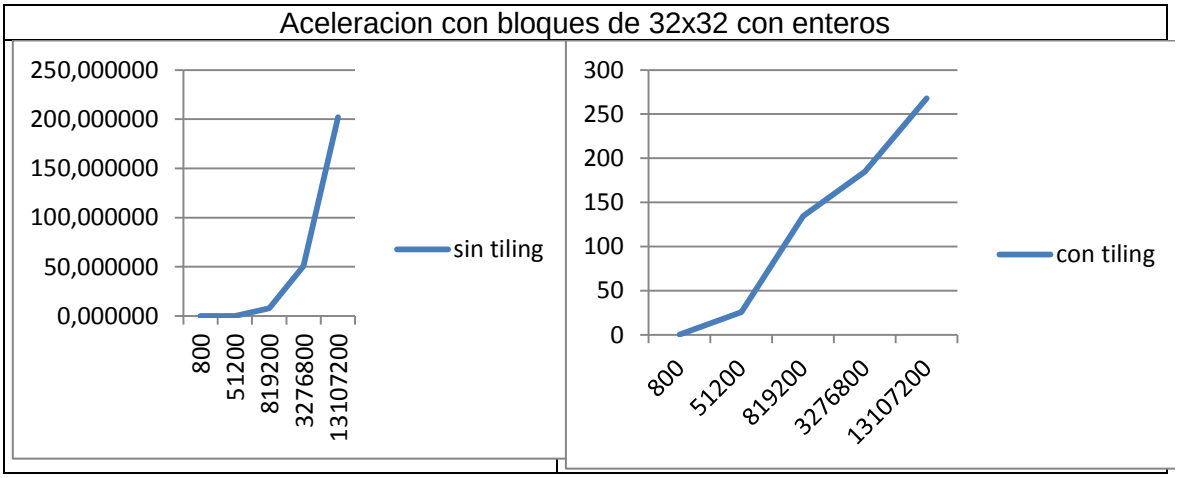


Comparacion aceleraciones

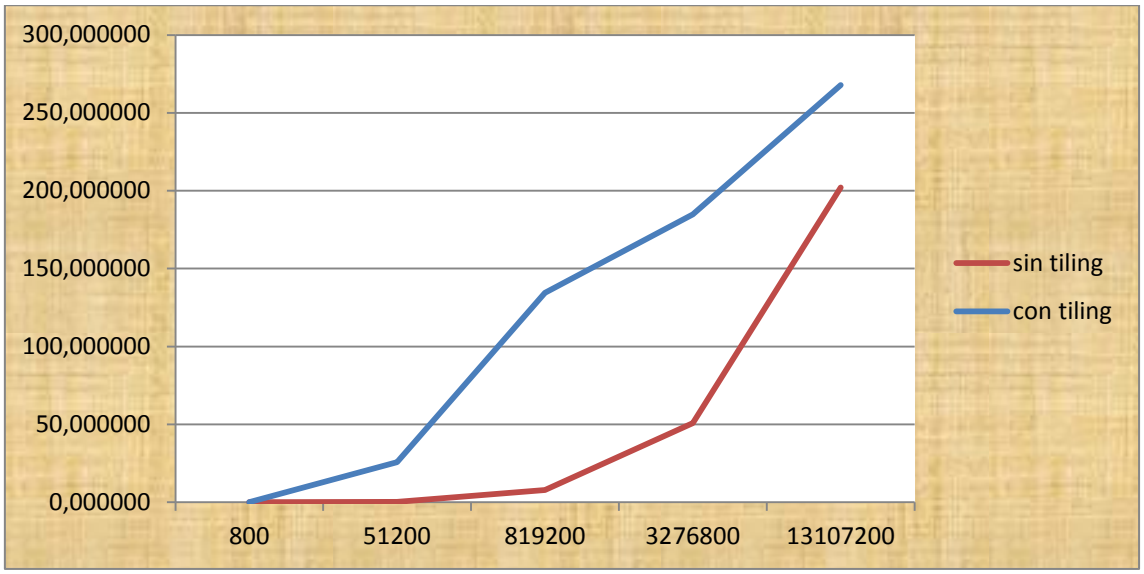




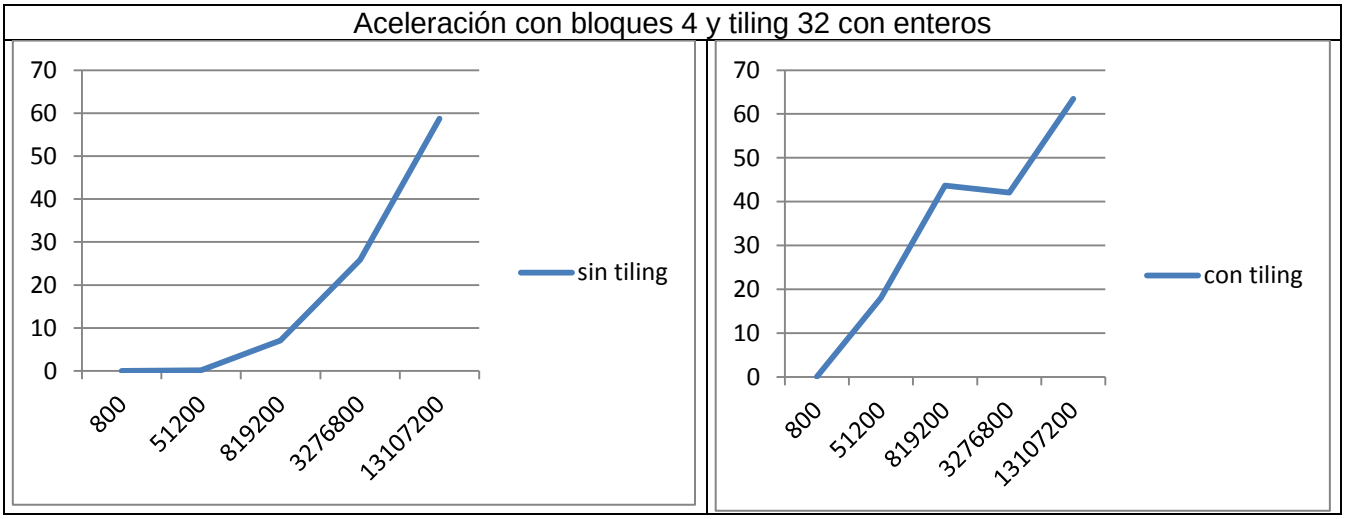
Comparacion de aceleraciones



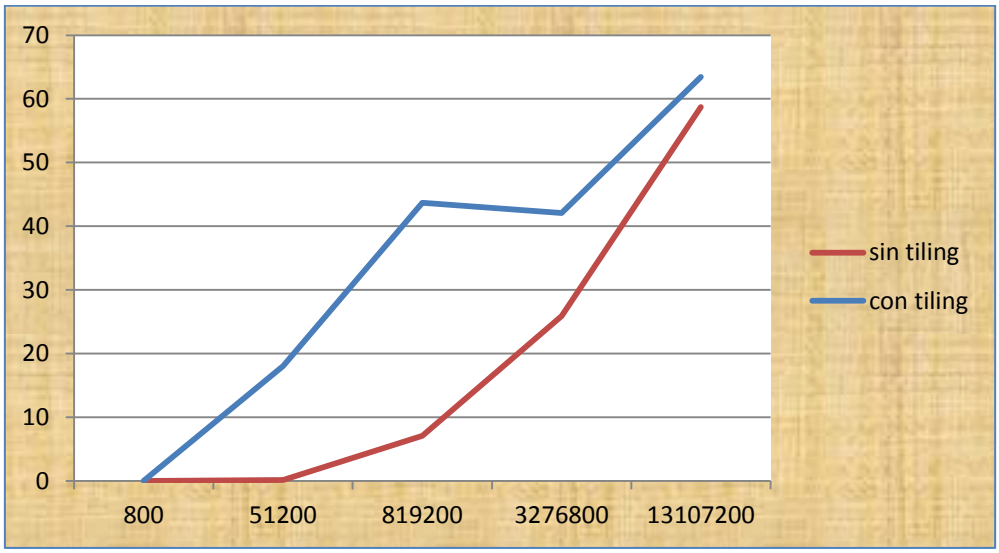
Comparación de aceleraciones con bloques 32x32



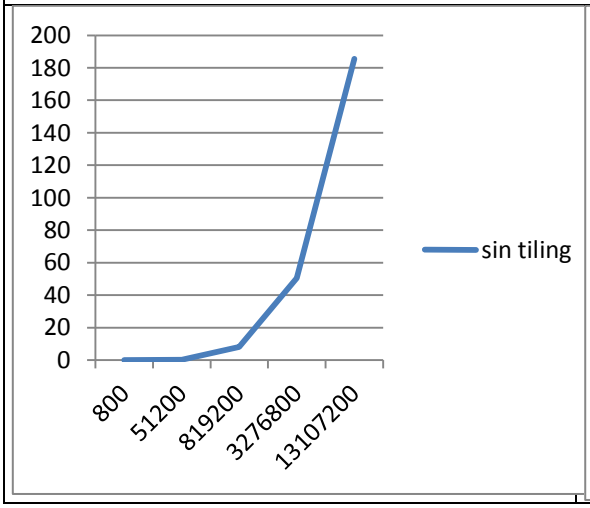
Combinacion con cambio de diferentes tamaños de bloques y tiling

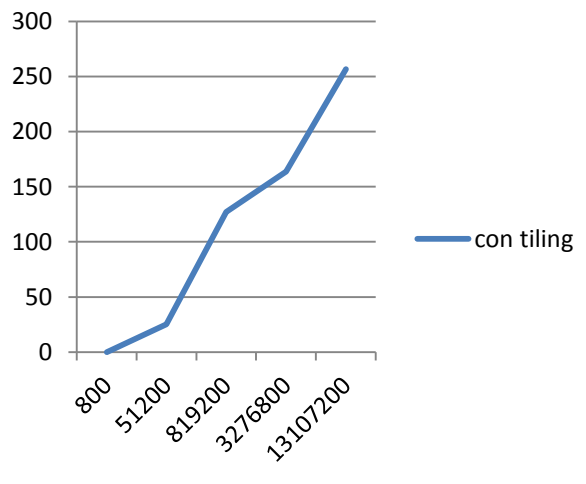


Comparacion de aceleraciones con cambio de bloque y de tiling

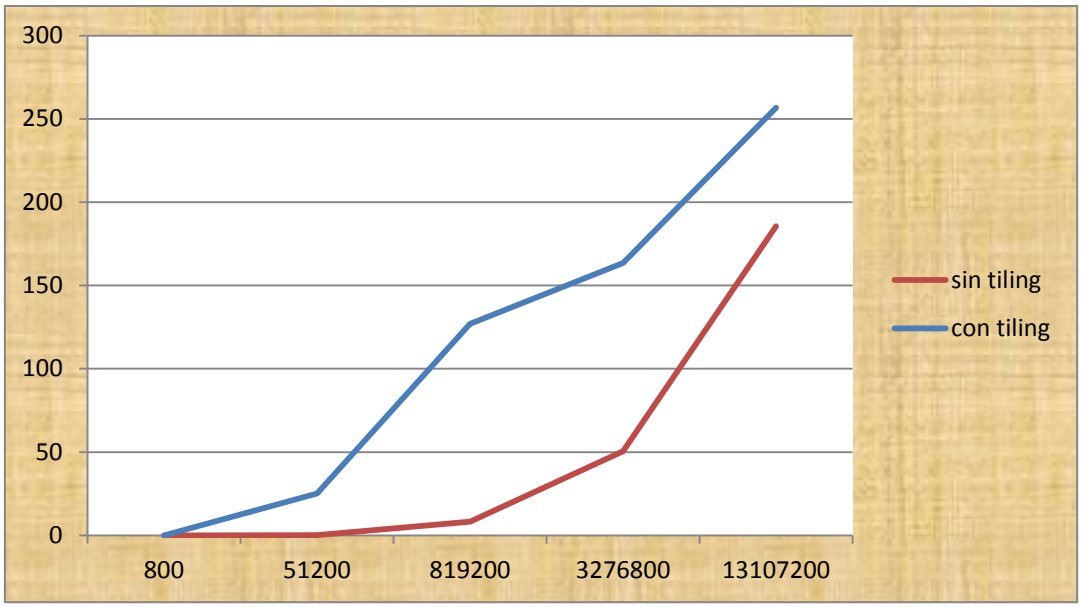


ACELERACION CON BLOQUES 16 Y ANCHO DEL TILE 4



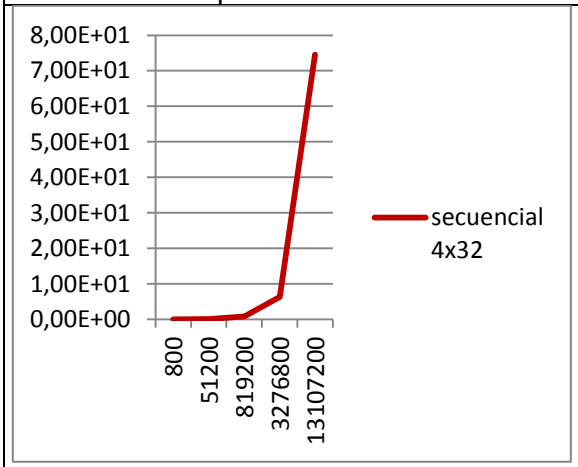


COMPARACION DE ACELERACIONES BLOQUES 16 – ANCHO TILING 4

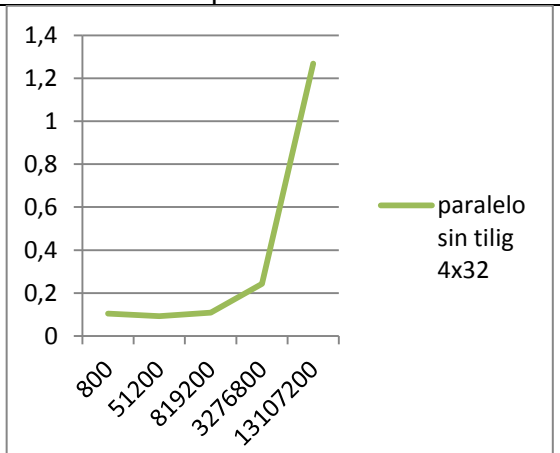


Graficas de tiempos con diferentes tamaños de Bloque y tiling con enteros.

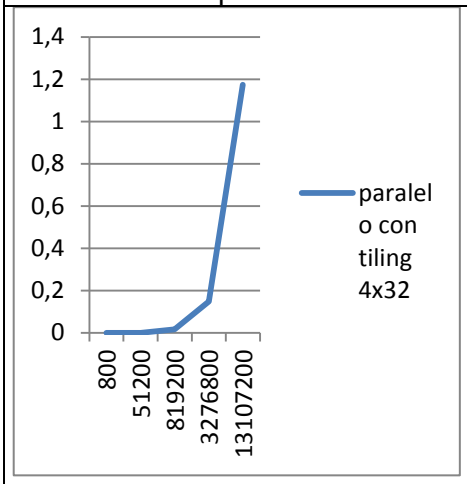
Grafica de tiempos 4x32



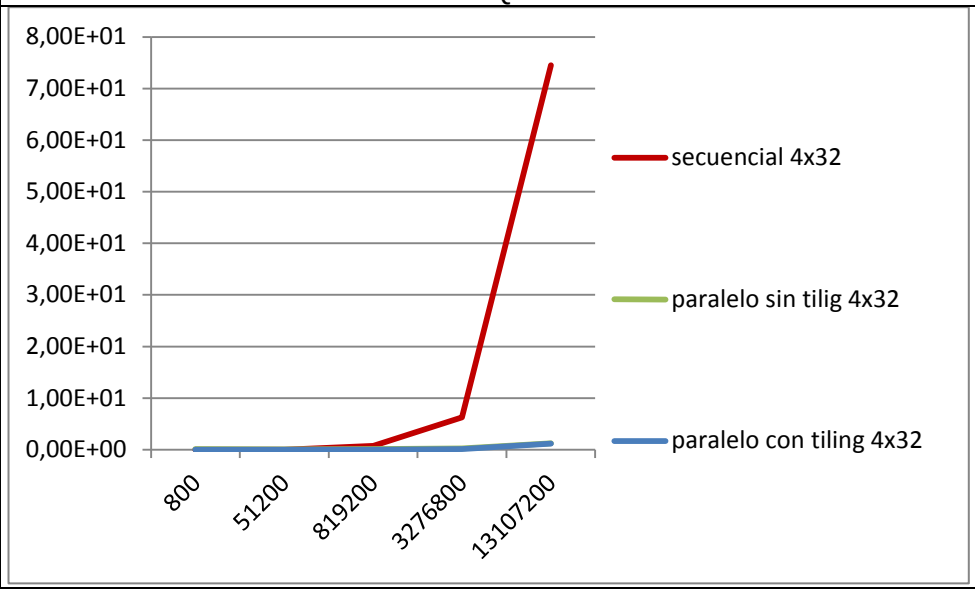
Grafica de tiempos 4x32



Grafica de tiempos 4x32



COMPARACION DE TIEMPOS BLOQUES 4 TILING 32



Grafica de tiempos de 16x4 con enteros

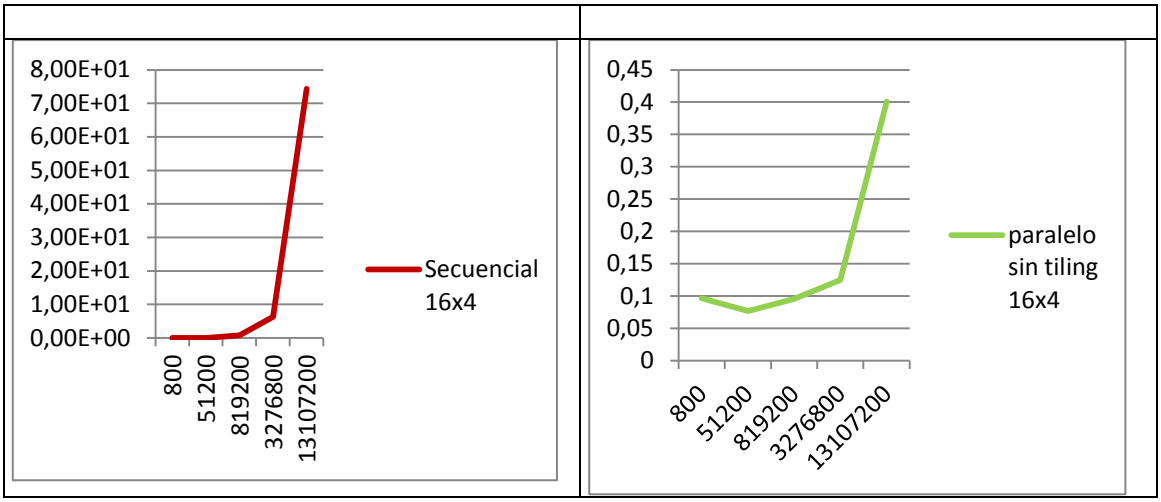
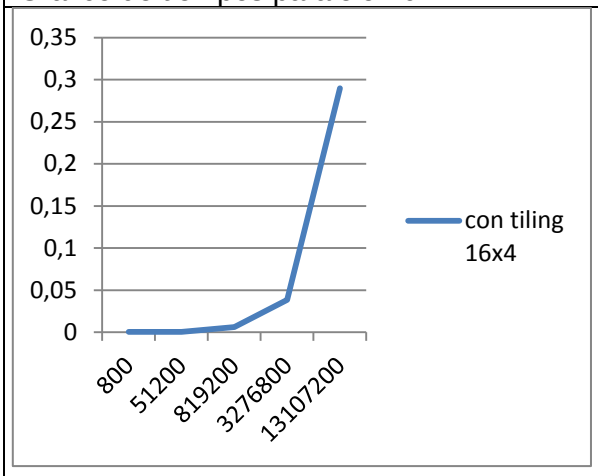
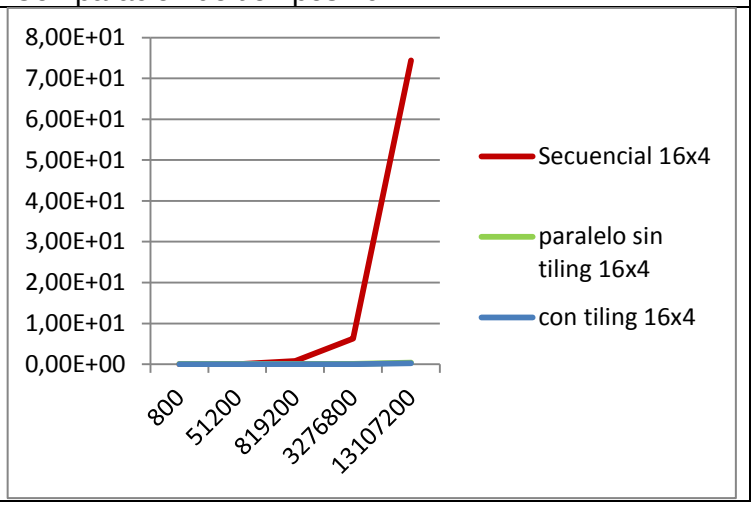
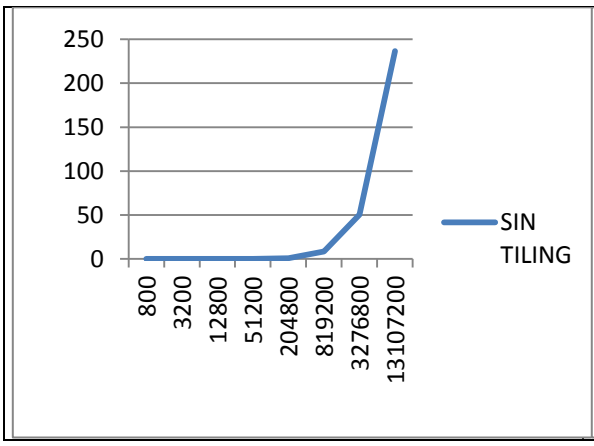


Gráfico de tiempos paralelo 16x4	Comparación de tiempos 16x4
	

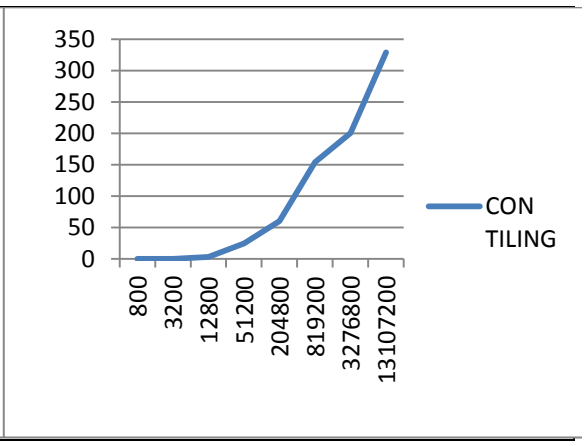
8. Realizar la implementación del algoritmo pero con números en punto flotante, realizar el mismo estudio para esta implementación.

Matriz A [n][m] B[m][o] usando Números con punto flotante						
tamaño matriz resultante	Matriz Dim C [n][m][o]	SECUENCIAL	PARALELO SIN TILING	PARALELO CON TILING	ACELRACION SIN TILING	ACELERACION CON TILING
800	20 X4X40	0,000024	0,109484	0,000474	0,00021921	0,050632911
3200	40X16X80	0,000137	0,097	0,000484	0,001412371	0,283057851
12800	80 X32x160	0,002642	0,096731	0,00084	0,027312857	3,145238095
51200	160X64X320	0,015482	0,091475	0,000626	0,169248429	24,73162939
204800	320X128X640	0,08379	0,093597	0,001396	0,895221001	60,02148997
819200	640X256X1280	0,789373	0,094846	0,005117	8,322680978	154,2648036
3276800	1280X512X2560	6,33678	0,126018	0,031572	50,28472123	200,7088559
13107200	2560X1024X5120	75,9336	0,3208029	0,23068	236,6986084	329,1728802

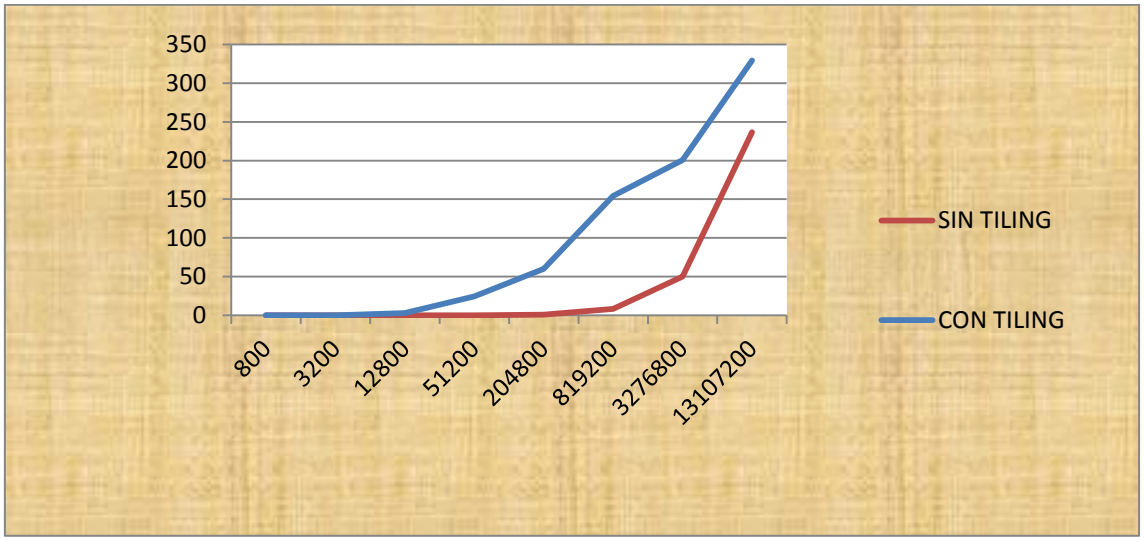
ACELERACION SIN TILING



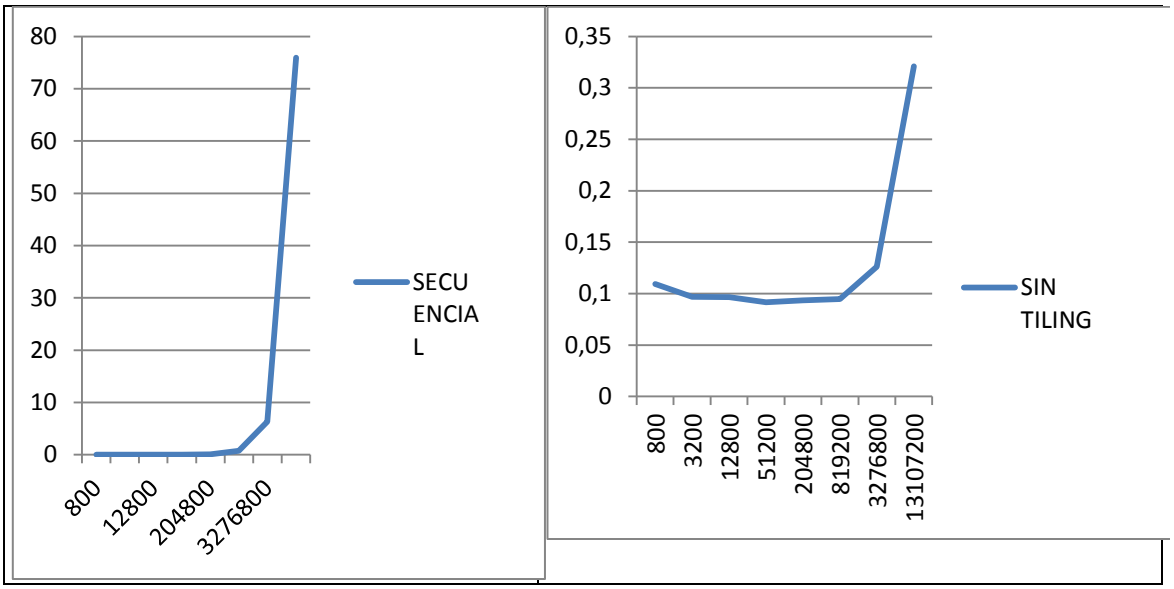
ACELERACION CON TILING

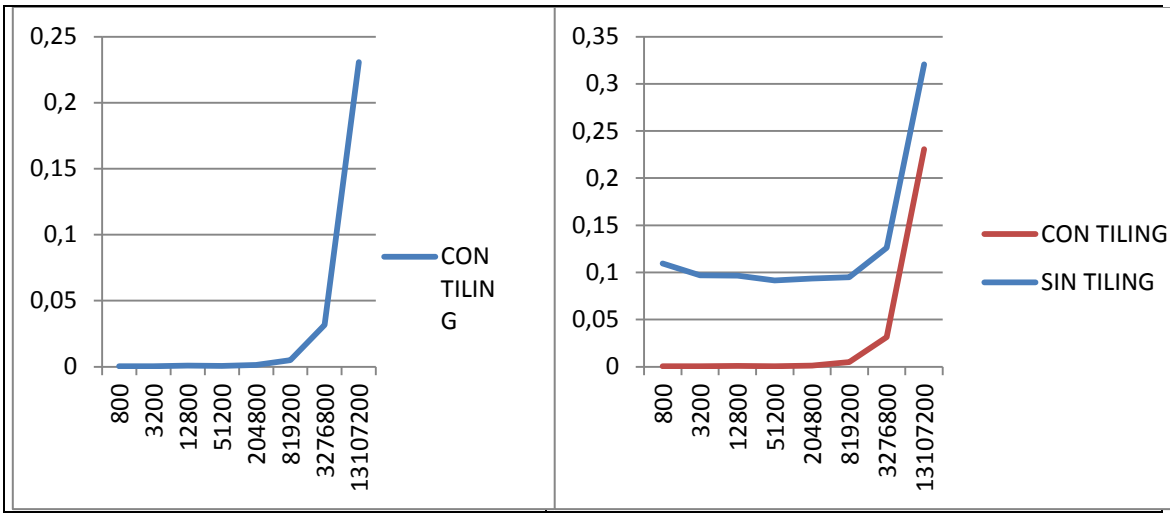


COMPARACION DE ACELERACIONES CON NUMEROS DE PUNTO FLOTANTE



GRÁFICA de tiempos CON NUMEROS DE PUNTO FLOTANTE.





9. Realizar un capítulo de **CONCLUSIONES.** Donde expliquen los resultados obtenidos gráfica.

Conclusiones:

a) Se puede observar claramente que para multiplicación y operaciones entre las matrices cuando el programa apenas inicia con las operaciones básicas la velocidad es mayor en CPU, pero cuando aumentamos el trabajo y cambiamos de tamaño las matrices ya el rendimiento es mejor en GPU el paralelismo hace se agilicen los procesos.
b) Es notoria la ganancia de la aceleración cuando trabajamos con tiles, este tipo de memoria y su manejo hace que la aceleración sea mayor y por ende tendremos un mejor rendimiento.
c) En cuanto a la implementación con números de tipo entero y números punto flotante, la diferencia en tiempos de ejecución no es muy notoria.
d) Se destaca el comportamiento y el beneficio de usar memoria compartida, la diferencia al usar el algoritmo implementando Tiles con los demás algoritmos secuencia y paralelo normal es de destacarse reduciendo los tiempos de ejecución para matrices grandes.
e) La aceleración observada en los dos casos que el algoritmo es implementado paralelamente es mucho mayor utilizando Tiles.
f) Al realizar cambios en el tamaño del Bloque y el Ancho del TILE se evidencia que a medida que se aumenta el tamaño para el tamaño del Bloque y el Tile el funcionamiento es mucho mejor, optimizando los tiempos de ejecución, siempre y cuando los valores de Bloque fuera igual al Tile.
g) El tiempo de ejecución que toma el algoritmo cuando el tamaño del Bloque es el mayor (32) y el tamaño del Tile es igual a 4, se puede notar que el tiempo de ejecución para el algoritmo con Tiling es mucho mejor.

.