

Informe - Caso 2

Grupo 3

Brian Manuel Rivera
202015320

Pablo Alejandro Guatibonza
202014393

Juan Pablo Martínez
202012623

17 de Octubre, 2022

1. Descripción de estructuras de datos

El buffer TLB, la tabla de páginas y la RAM fueron representadas mediante clases en Java (TLB, TP y RAM) con sus respectivos atributos y métodos que permiten simular la administración de memoria. En primer lugar, el TLB es representado como una cola de Hashtables, en las cuales las parejas llave-valor son enteros que representan una dirección en memoria virtual y una dirección en memoria real, respectivamente. Este se actualiza cuando el thread que carga las referencias no encuentra la que está buscando ni en el TLB ni en la TP, dicho algoritmo utiliza el sistema FIFO revisando que a medida que inserta las hahtables no excede el tamaño involucrado.

En segundo lugar, la tabla de páginas (TP) y la RAM son representadas como arreglos de enteros. En la TP, cada posición representa una dirección de memoria virtual y el valor guardado en esa posición es el marco de página asociado a esa dirección. Por otro lado, en el arreglo de la RAM, cada posición representa un marco de página y el valor guardado en esa posición es una página de la memoria virtual. Ambas estructuras se actualizan cuando hay fallo de página. Primero, se revisa si hay RAM disponible, esto lo hace con un atributo llamado RAMfree que representa una cola con las posiciones de RAM disponibles, en caso afirmativo, se guarda el marco de página y la página en ambos arreglos y se desencola la posición de RAM que fue utilizada, en caso contrario, el thread que corre el algoritmo de envejecimiento se encarga de hallar la página menos referenciada para reemplazarla con la nueva referencia que se quiere cargar. Entonces, por su parte el algoritmo de envejecimiento utiliza una estructura denominada AggingStructure, en donde hay un arreglo de enteros donde cada posición tiene un Integer, donde se aprovechan los 32 bits que este contiene para representar los corrimientos con 1 o 0 según las páginas referenciadas, esto también tiene en cuenta que el primer bit tiene un 1 o 0 según si el número es negativo o positivo, y los 31 restantes definen el valor, por lo tanto para sacar la mejor página basta con obtener el entero más pequeño de ese arreglo y su posición significará la página virtual menos referenciada. Finalmente, cada vez que se realiza una referencia esta es agregada a una lista estática con los enteros referenciados, y cuando corre el algoritmo de envejecimiento este toma estos enteros referenciados para decidir en qué posiciones hace corrimiento con 0 o con 1.

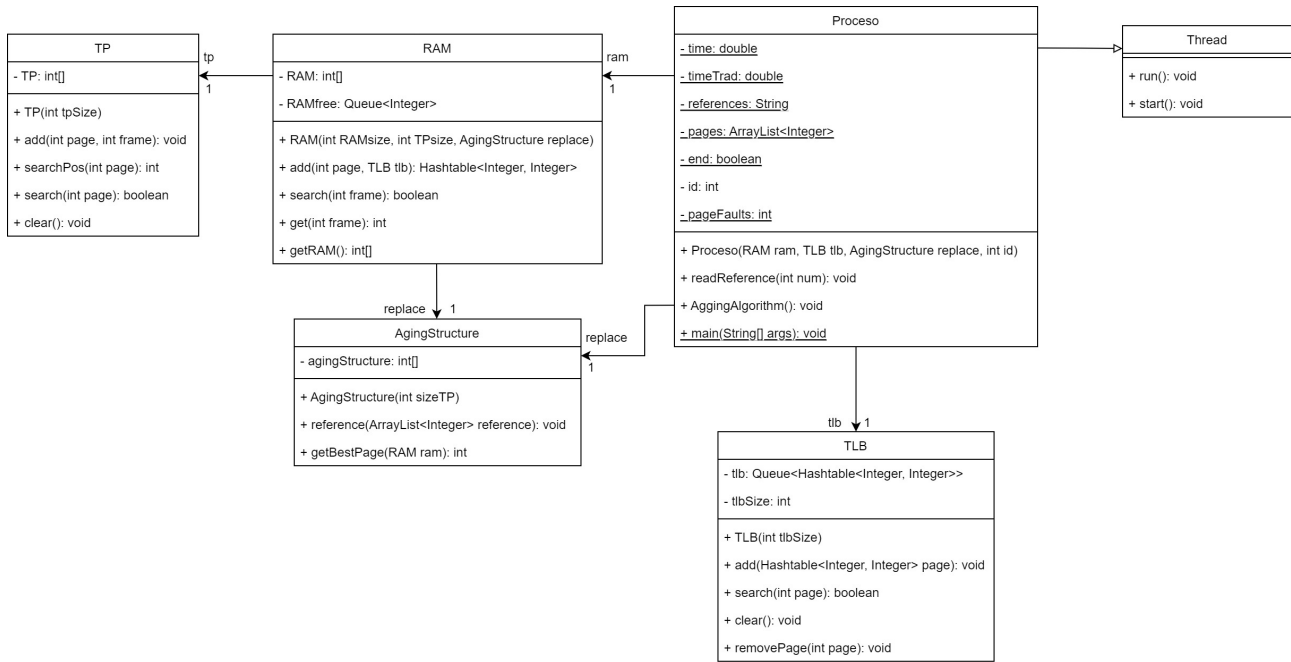


Figura 1: Diagrama de clases del sistema

2. Sincronización

Es necesario usar sincronización al momento de actualizar la lista con las páginas que ya se han cargado, también al momento de actualizar la estructura que maneja el algortimo de envejecimiento. Esto debido a que ambos threads comparten la información guardada en ambas estructuras para ejecutar sus respectivos procesos. Se utiliza la exclusión mutua como esquema de sincronización con ayuda de la palabra reservada *synchronized* disponible en Java. Por otro lado, también hay sincronización al momento de decidir qué página será intercambiada de la RAM, esto es relevante debido a que no debería ser posible ejecutar el algoritmo de envejecimiento mientras se decide la página menos referenciada, para no decidir con datos incompletos.

3. Datos recopilados

Se presentan los datos obtenidos al ejecutar el programa respecto al tiempo de carga de páginas en memoria y al tiempo de resolución de direcciones virtuales. Asimismo, se muestran las gráficas asociadas a estos resultados que ilustran el comportamiento del sistema.

3.1. Carga de datos en un proceso con localidad alta

Marcos de página \ Tamaño de TLB	Tamaño de TLB			
	8	16	32	64
8	2490019177	2520016664	2480016432	2540016780
16	1960020723	1970013865	1990013645	1960013415
32	1160018324	1240011619	1220010523	1230010021

Cuadro 1: Tiempos en nanosegundos de carga de datos en proceso con localidad alta

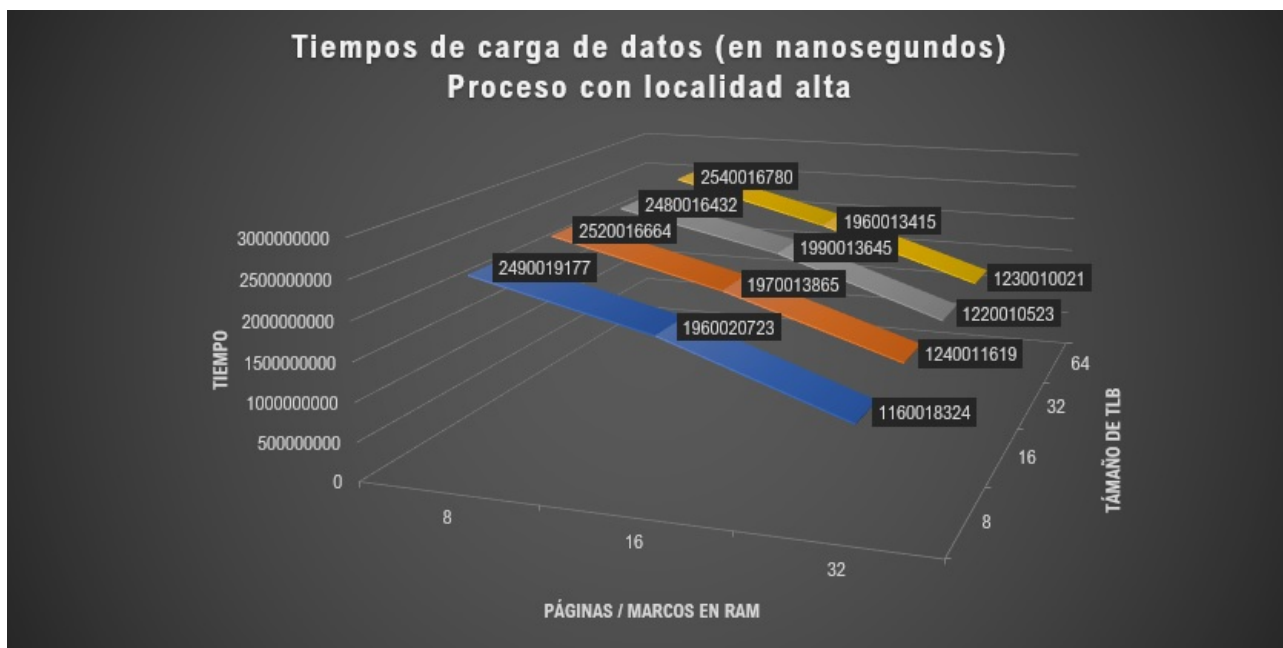


Figura 2: Tiempos de carga de datos (localidad alta)

3.2. Carga de datos en un proceso con localidad baja

Marcos de página	Tamaño de TLB	8	16	32	64
8		6600040972	6630040530	6670040734	6680040792
16		5300037044	5240033672	5250032722	5260032556
32		2750029226	2680026160	2740024212	2770022818

Cuadro 2: Tiempos en nanosegundos de carga de datos en proceso con localidad baja

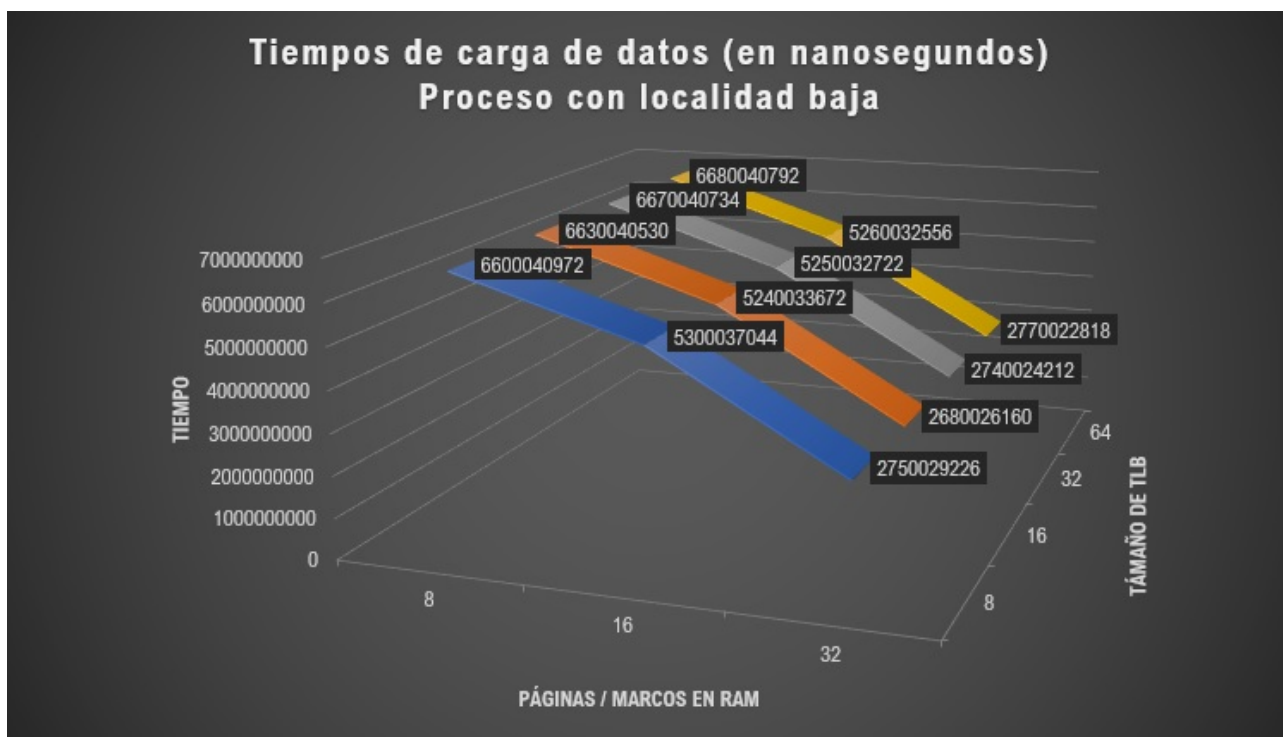


Figura 3: Tiempos de carga de datos (localidad baja)

3.3. Resolución de direcciones en un proceso con localidad alta

Marcos de página	Tamaño de TLB			
	8	16	32	64
8	18908	16722	16780	16896
16	18926	13864	13324	13590
32	18278	11464	10502	9702

Cuadro 3: Tiempos en nanosegundos de traducción de direcciones virtuales en proceso con localidad alta

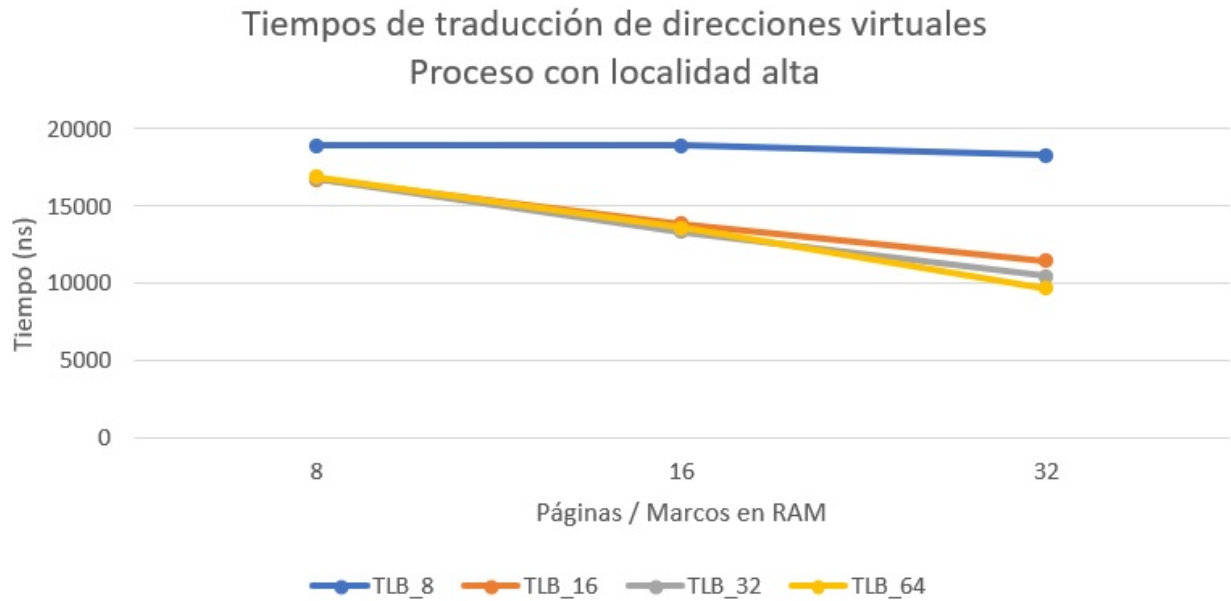


Figura 4: Tiempos de resolución de direcciones (localidad alta)

3.4. Resolución de direcciones en un proceso con localidad baja

Marcos de página	Tamaño de TLB			
	8	16	32	64
8	40942	40029	40037	39979
16	37273	34261	32659	32903
32	29341	26181	24303	22467

Cuadro 4: Tiempos en nanosegundos de traducción de direcciones virtuales en proceso con localidad baja

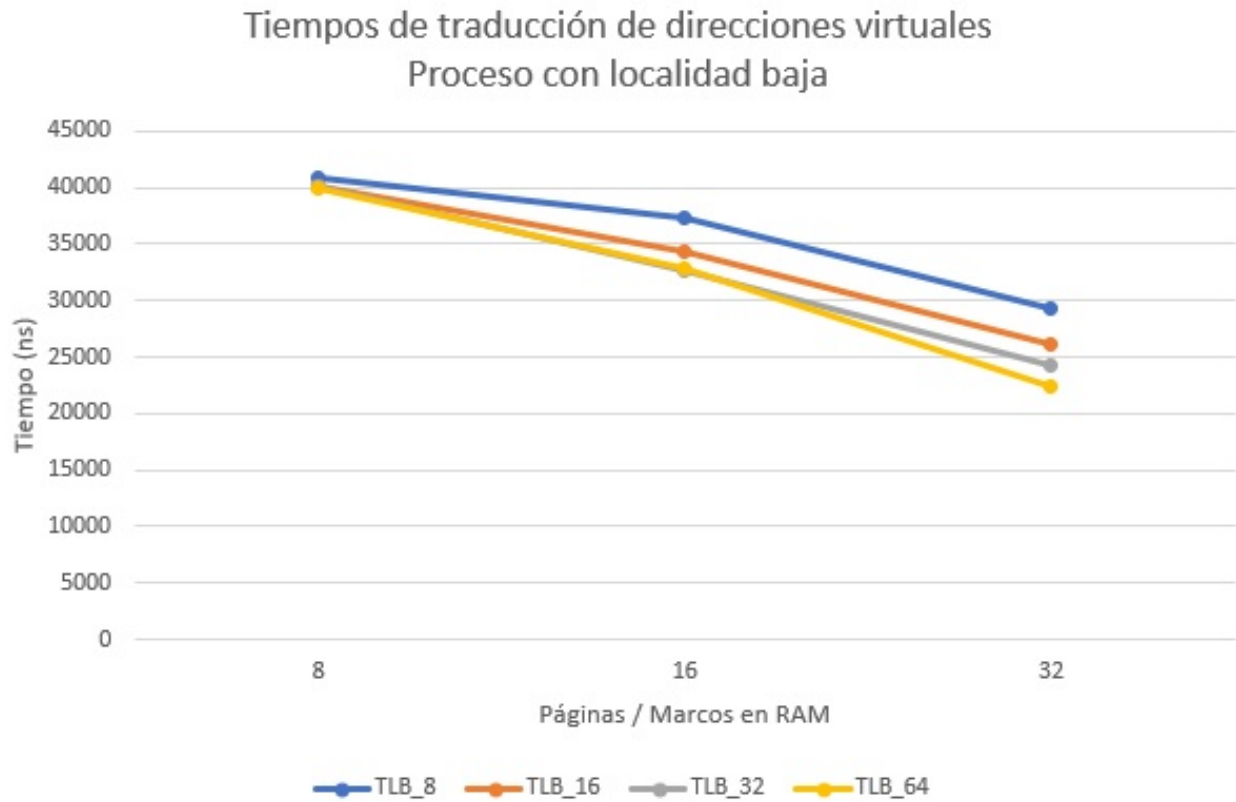


Figura 5: Tiempos de resolución de direcciones (localidad baja)

4. Interpretación de resultados

Los resultados obtenidos tienen sentido, ya que, al tener un tamaño mayor en la TLB, el número de veces que se necesita consultar una referencia en la tabla de páginas es menor, de esta manera, el tiempo que tarda traduciendo y por ende cargando y accediendo a todas las referencias se reduce. La reducción en tiempo también se ve reflejada al comparar los procesos con localidad alta y baja, en el primer caso se cumple el principio de localidad y el número de fallos de página se reduce, en consecuencia, el tiempo de carga de datos y de resolución de direcciones disminuye considerablemente. Por otro lado, la cantidad de marcos de página disponibles en RAM es relevante ya que de allí depende la cantidad de fallos de página que sufrirá el proceso que los referencia, como se puede apreciar en las gráficas estas muestran un comportamiento decreciente según la cantidad de marcos de página en RAM aumentan.