

1. ACLARACIONES DEL PROGRAMA:

El programa se ejecuta corriendo la clase App.Java.

Las tablas y gráficas se harán en base a 6 casos establecidos en el documento.

- **Algoritmo para opción 1:**

- Recibo el tamaño de la página y el tamaño del entero por parametro
- Establezco una variable que corresponda a la página actual y un contador. Esta PA(página actual) iniciará en 0 y se incrementara en unidades cada vez que el contador exceda al tamaño de la página. Este contador se incrementa según el tamaño del entero(se incrementa de uno en uno si el tamaño del entero es 1, dos en dos si es 2,etc.).
- Cuando el contador exceda el tamaño de la página, se reinicia a 0 y se aumenta la página actual. Este contador corresponde al desplazamiento de la matriz.

2. ESTRUCTURAS DE DATOS:

- **ArrayList:** Se creo una lista para dar una representación a la memoria RAM real donde se cargan las páginas. Se actualiza cuando se cargan o descargan páginas de la memoria real. Esto consiste en añadir o borrar elementos de la ArrayList con base al algoritmo de reemplazo (envejecimiento o en la carga de páginas).
- **ConcurrentHashMap:** Esta estructura simula la tabla de páginas de la memoria, en esta se guardan las páginas con sus atributos y estados. Este hashmap se actualiza cada vez que se hace referencia a una página. La actualización consiste en obtener la página del hashmap, actualizarla y volverla a poner en el mapa. Ocurre lo mismo cuando el algoritmo de reemplazo (envejecimiento) carga o descarga páginas de la RAM (ArrayList).

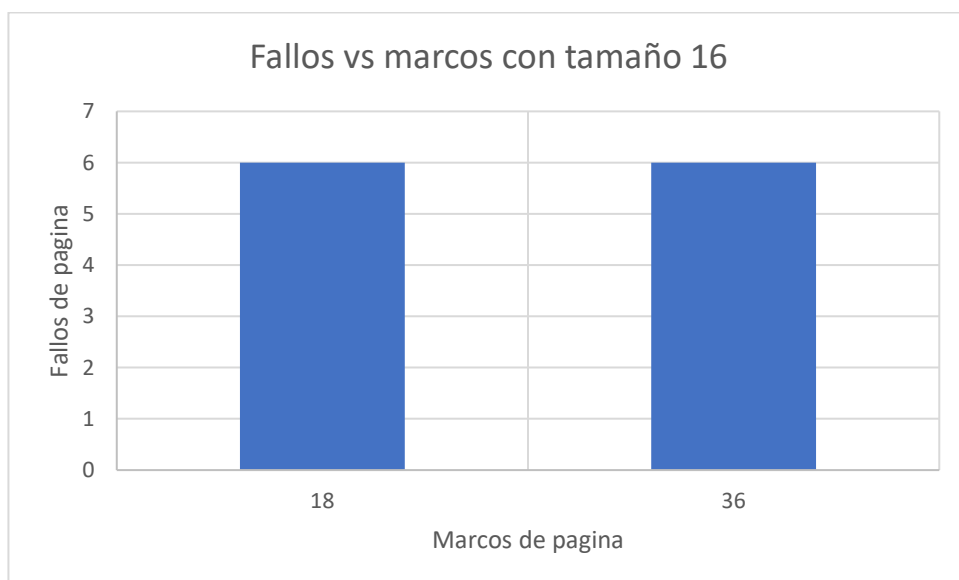
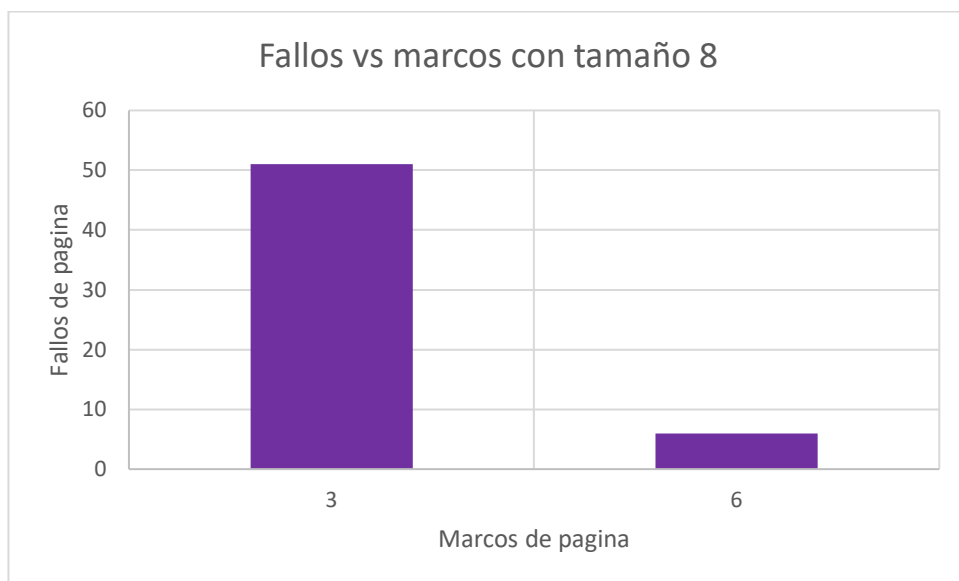
3. SINCRONIZACION:

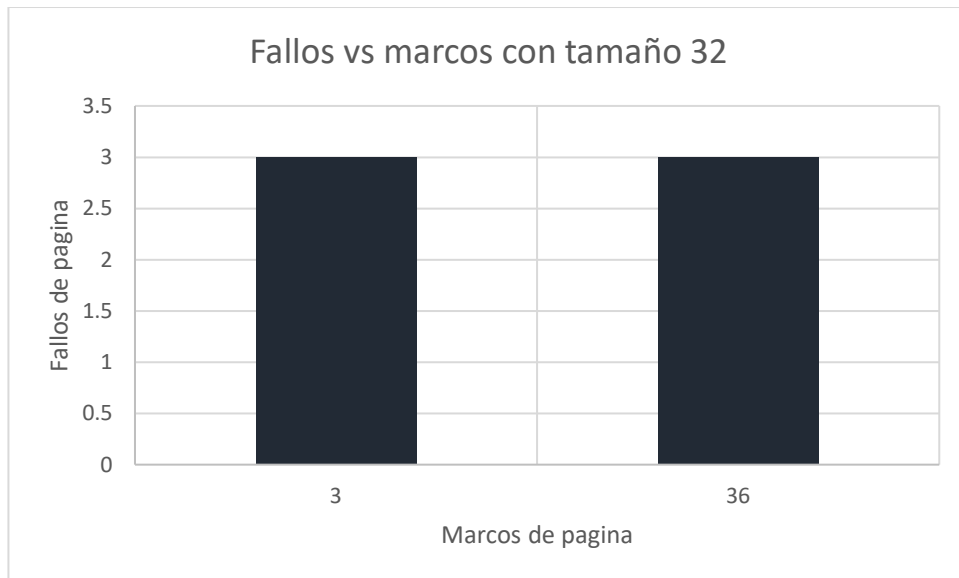
- Se utilizaron métodos sincronizados con exclusión mutua para evitar que se modifique la misma estructura de datos al mismo tiempo. Adicionalmente, se implementó un ConcurrentHashMap, mencionado previamente, el cual tiene la característica de que todos sus métodos básicos son de tipo de exclusión mutua. Con lo cual múltiples threads pueden

utilizar el HashMap al mismo tiempo, pero solo se excluyen mutuamente si ambos están utilizando el mismo elemento (llave, valor).

4. TABLA DE DATOS:

Escenario	Tamaño Pagina	Marcos de pagina	Fallos de pagina
1	8	3	51
2	8	6	6
3	16	18	6
4	16	36	6
5	32	3	3
6	32	36	3





5. INTERPRETACION Y CONCLUSION:

En general, los algoritmos y procedimientos propuestos para la opción 1 y opción 2 son pertinentes y predicen de manera adecuada la simulación. Para el caso de la opción 1, esto se evidencia en la correspondencia directa con los ejemplos presentes y que el algoritmo planteado no va a fallar independientemente de las condiciones (obviamente, dentro de lo razonable).

Los resultados obtenidos a partir de la tabla y grafica tienen sentido, en el cual se aprecian dos tendencias. A partir del código, nos dimos cuenta de que el número de páginas que son utilizadas para la simulación de paginación tienen un impacto. A pesar de que no existan gráficos en específico de este caso, nos dimos cuenta en los resultados de consola, que entre mayor sea el número de páginas utilizadas, mayor será el número de fallos de página y viceversa. Por último, el segundo ocurre cuando se tiene un número de páginas del proceso fijo, pero cambia la cantidad de marcos asignados. Entre más marcos pueda utilizar el proceso, menos fallos de página, mientras que, si este número es menor, muchos más fallos ocurrirán.