**Caso #2**

El caso nos pide realizar el control de fallos de página para dos recorridos de matrices, la opción 1 se da cuando se quiere recorrer la matriz de izquierda a derecha y la opción 2 cuando se quiere recorrer la matriz de derecha a izquierda.

Para la opción 1 no se consideró necesario utilizar ningún tipo de estructura que nos permitiese guardar los valores que contenía cada una de las matrices, por lo tanto, se considero que la cantidad de fallos de página ya que disponíamos del espacio en ram para guardar las 3 matrices por completo se definía como: 3\*(número columnas \* número filas); por lo tanto, simplemente se concentró en generar el documento solicitado de manera correcta.

Para la opción dos se debieron usar varias clases de apoyo además del uso de varias estructuras que proveía Java y una implementación de una tabla HashLinearProbing la cual nos permitía manejar los valores aleatorios de las matrices y sus posibles colisiones dentro de la tabla hash para así obtener un rápido tiempo de comprobación de si una página esta cargada en ram o no, además se utilizaron un arreglo de tamaño fijo que nos permitía conocer el estado de una página requerida y por último un ArrayList que nos permitía almacenar los valores que se requerían para hacer el recorrido de las una matrices. Por otro lado la opción dos requería que se usaran Threads para poder llevar el estado y ejecutar el algoritmo de envejecimiento de las páginas que habían cargado en memoria ram, por lo cuál fue necesario que se utilizara tanto el mecanismo de sincronización sobre la tabla hash que mencionamos anteriormente, la cuál modelaba las páginas de memoria ram, ya que si sincronizábamos los métodos podíamos era obtener un rendimiento menos óptimo de los algoritmos y la actualización de los estados que sería casi como si se estuviesen ejecutando de manera secuencial, por lo tanto y ya que era solo un objeto el que se debía sincronizar se tomó la decisión de que era mejor solo sincronizar sobre el objeto en vez del método.

Por último, el archivo generado por la opción 1 se guarda en la misma carpeta docs en la que se encuentra el presente documento, para poderse ejecutar y comprobar cada una de las opciones, además de que la opción 2 implementada nos permite ver el paso a paso del algoritmo de envejecimiento.

**Diagramas y resultados:**

Gráfico

Descripción generada automáticamente

Gráfico

Descripción generada automáticamente

**Conclusiones:**

Según los resultados obtenidos consideramos que tienen sentido ya que entre menos marcos de página mayor será el número de fallos y reemplazos que se tendrán que hacer, además entre más reemplazos la probabilidad de tener que reemplazar elementos que se usen de manera recurrente para usar otros que se requieran en el momento es más alta, incluso con el algoritmo de la página menos usada recientemente.