Santiago Vela - 202026767

Verónica Escobar – 201922197

Caso 2: Memoria Virtual

Contenido

[Introducción 1](#_Toc100005176)

[Diagrama de solución 1](#_Toc100005177)

[Funcionamiento del Sistema (general) 1](#_Toc100005178)

[Funcionamiento opción 1 1](#_Toc100005179)

[Algoritmo de generación de referencias de pagina 1](#_Toc100005180)

[Estructuras de datos para la simulación del sistema de paginación 1](#_Toc100005181)

[Funcionamiento opción 2 1](#_Toc100005182)

[Sistema de sincronización 1](#_Toc100005183)

[Datos recopilados 1](#_Toc100005184)

[Graficas del comportamiento del sistema 1](#_Toc100005185)

[Graficas 1](#_Toc100005186)

[Interpretación de resultados 1](#_Toc100005187)

[Conclusiones generales 2](#_Toc100005188)

# Introducción

El objetivo de este caso fue el de crear una simulación de como el sistema operativo administra la memoria RAM de un dispositivo por medio de la creación de la memoria virtual y otros elementos tales como el sistema de paginación. Es por esto que el caso se enfoca principalmente en el Funcionamiento del sistema de paginación y su Funcionamiento. Para simular esto se escribió un programa en Java que simula el sistema de paginación usando el algoritmo de envejecimiento. En esta simulación se observa el comportamiento del sistema de paginación mientras que un proceso corre, en este caso la suma de dos matrices. En el transcurso de este proceso el sistema de paginación debe tomar decisiones de remplazo de página basados en el algoritmo de envejecimiento.

# Funcionamiento del Sistema (general)

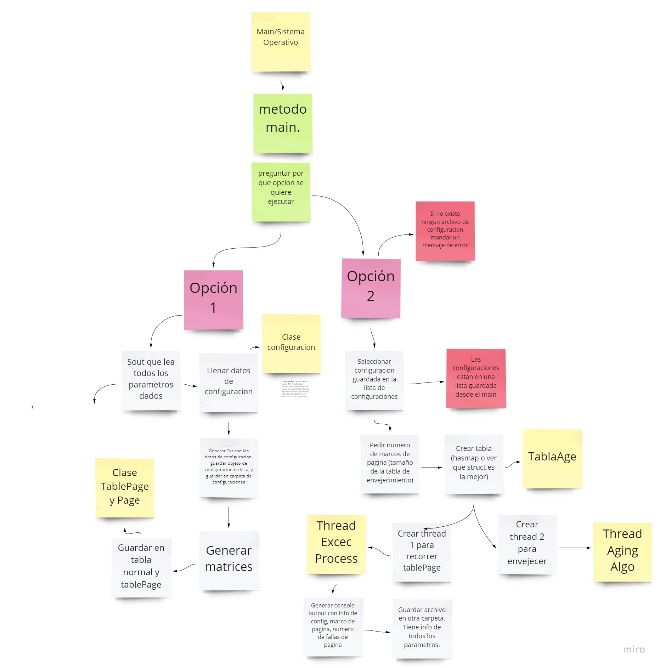


Ilustración 1. Diagrama de la solución

El diagrama anterior muestra la estructura general de la solución dada al caso y el flujo de decisiones para solucionar tanto la opción 1 como la opción 2. Aquí se ven presente algunas de las clases que fueron usadas en la implementación en java.

**Insertar diagrama de clases cuando terminemos**

En el diagrama se puede ver como Main es nuestra clase principal, sus responsabilidades constan de manejar todo lo que la interfaz de usuario, selección de opciones y la carga de archivos de configuración (que es donde se guardan los resultados de opción 1). Nuestra clase main se podrá ver como la implementación del sistema operativo.

La clase de Configuración es donde se encuentra la implementación de la opción 1, ya que es donde se genera la tabla de páginas y la tabla de referencias a dichas paginas dependiendo de los parámetros dados por el usuario (tamaño de página, tamaño del entero, tamaño de las matrices, recorrido). Una instancia de la clase configuración guarda el estado de la configuración dada por parte del usuario, generando las tablas y reportes necesarios para la opción 1. Adicionalmente, la clase Mein guarda un arreglo con las configuraciones preexistentes, para que sean de fácil acceso al ejecutar la opción 2. Clases como Pagina y ElementInfo son auxiliares a la clase de Configuración, y estas se usan para crear las diferentes tablas que produce esta clase. Internamente la clase Configuración crea varias tablas que facilitan la eventual creación de la tabla de páginas (que es la usada para crear los reportes) y para crear la tabla de referencias (usada en opción 2).

La clase de Opcion2 actúa como un administrador para todos los procesos que se tienen que llevar a cabo para similar el sistema de paginación en conjunto con el algoritmo de envejecimiento. Para cumplir con los requisitos de la opción 2 se usan 3 clases auxiliares: Thread1, Thread2 y Buffer. Son estas clases auxiliares las que implementan el algoritmo de envejecimiento y el sistema de paginación a detalle. La clase de Thread1 tiene una referencia a la tabla de referencias, donde se almacenan las referencias (en orden) hechas a cada página por el recorrido que fue previamente seleccionado en opción 1. El trabajo de thread 1 es recorrer la tabla de referencias, después intentar acceder al buffer para modificar la table del marco de página, donde se guardan las páginas accedidas y su “edad” para ingresar una nueva página, actualizar su edad (reseteando a 0, al ser la referencia más reciente), o generar un fallo de página para después swapear la página más reciente por la página referenciada actualmente. Adicionalmente, después de hacer esta acción el thread 1 se va a dormir por dos milisegundos. La clase Thread2 es responsable de ejecutar el algoritmo de envejecimiento, lo cual incurre acceder al buffer a la table del marco de página para incrementar por 1 la “edad” de cada referencia de página dentro del marco. Después de ejecutar el envejecimiento este thread se va a dormir por 1 milisegundo. Ambos threads correrán hasta que el thread 1 termine de recorrer la tabla de referencias. La clase del buffer, como antes mencionado, contiene la table del marco de página, la cual tiene las páginas referenciadas y su edad (que simboliza que tan reciente fue la última referencia a ellas dentro de la table de referencias). En esencia la clase del buffer esta sincronizada en relación a la table de marco de página y solo permite que un thread actúe sobre ella al tiempo, por medio de un método sincronizado sobre el cual expondremos en otra sección.

# Funcionamiento opción 1

## Algoritmo de generación de referencias de pagina

## Estructuras de datos para la simulación del sistema de paginación

# Funcionamiento opción 2

## Sistema de sincronización

## Datos recopilados

## Graficas del comportamiento del sistema

### Graficas

### Interpretación de resultados

# Conclusiones generales