

Programación Concurrente 2023

Trabajo Práctico N° 1

Condiciones

- El trabajo es grupal, de 4-5 alumnos (3 es una excepción)
- La defensa del trabajo es con el grupo completo
- La evaluación es individual (hay una calificación particular para cada integrante)
- Solo se corrigen los trabajos que hayan sido subidos al aula virtual (LEV)
- Los problemas de concurrencia deben estar correctamente resueltos y explicados.
- El trabajo debe implementarse en lenguaje Java.
- Se evaluará la utilización de objetos y colecciones, como así también la explicación de los conceptos relacionados a la programación concurrente.

Enunciado

En un sistema de gestión de imágenes, existe una funcionalidad que se encarga de mejorar la calidad de los archivos obtenidos, previo a la entrega final al cliente. Esta funcionalidad ha quedado obsoleta, por lo tanto se solicita realizarlo nuevamente, teniendo en cuenta los siguientes lineamientos de diseño.

Dicha funcionalidad se divide en cuatro procesos.

El **primer proceso** se encarga de cargar las imágenes en un contenedor, en su estado original. Existen **dos hilos** que ejecutan este proceso, demorando un tiempo aleatorio en ms para obtener la imagen y depositarla en el contenedor. Ambos hilos dejan las imágenes en el mismo contenedor.

El **segundo proceso** tiene por objetivo mejorar la iluminación de cada imagen. En este caso **tres hilos** ejecutan este proceso, demorando un tiempo aleatorio en ms para mejorar la iluminación de cada imagen. Cada hilo debe tomar una imagen aleatoria del contenedor por vez, y no puede tomar más de una vez la misma imagen. En el tiempo que demora en mejorar la imagen no debe bloquear otros hilos que quieran tomar otras imágenes para mejorar. Cada imagen debe ser mejorada todos los hilos (los 3 del presente proceso) para que el siguiente proceso pueda tomarla.

El **tercer proceso** debe ajustar las imágenes al tamaño final solicitado. Este proceso tiene que ejecutarse después que la imagen ya fue mejorada. **Tres hilos** son los encargados de ejecutar este proceso, tomando cada uno una imagen aleatoria del contenedor por vez para ajustarla, y demorando un tiempo aleatorio en ms. En el tiempo que demora en ajustar la imagen no debe bloquear otros hilos que quieran tomar otras imágenes para ajustar. Una imagen solo puede ser ajustada una sola vez.

El **cuarto proceso** toma las imágenes ya mejoradas y ajustadas, les hace una copia en el contenedor final, y luego las elimina del contenedor inicial. **Dos hilos** ejecutan este proceso, y demoran un tiempo en ms en realizar su trabajo. Cada hilo toma de a una imagen por vez de manera aleatoria.

Consideraciones:

- ☐ Los objetos imagen tienen un atributo "improvements" el cual registra cuántos hilos han tomado el archivo para mejorarlo (proceso 2).
- ☐ Cuando se menciona un tiempo aleatorio en ms (milisegundos), el mismo debe ser no nulo y a elección del grupo.
- ☐ Una imagen no puede pasar a ser ajustada sin antes haber sido mejorada por los 3 hilos del segundo proceso.
- ☐ Al finalizar la ejecución es necesario verificar cuantas imágenes ajustó cada hilo del tercer proceso.
- ☐ Al finalizar la ejecución es necesario verificar cuantas imágenes movió del contenedor inicial hacia el contenedor final, cada hilo del cuarto proceso.

El sistema debe contar con un LOG con fines estadísticos, el cual registre cada 500 milisegundos en un archivo:

- Cantidad de imágenes insertadas en el contenedor.
- Cantidad de imágenes mejoradas completamente.
- Cantidad de imágenes ajustadas.
- Cantidad de imágenes que han finalizado el último proceso.
- El estado de cada hilo del sistema.

Ejercicios

- 1) Hacer un diagrama de clases que modele el sistema de datos con TODOS los actores y partes.
- 2) Hacer un solo diagrama de secuencias que muestre la siguiente interacción:
 - a) Con el contenedor inicial vacío, la inserción de un archivo.
 - b) La interacción de un hilo del segundo proceso al intentar acceder al archivo para la mejora.
 - c) La interacción de un hilo del tercer proceso cuando accede a un archivo para ajustar el tamaño de la imagen.
- 3) Las demoras del sistema en sus cuatro procesos deben configurarse de tal manera de poder procesar 100 imágenes (desde la inserción en el primer contenedor hasta que son movidas al contenedor final) en un periodo mínimo de 10 segundos y máximo de 20 segundos.

- 4) Hacer un análisis analítico detallado de los tiempos que el programa demora. Luego contrastarlo con múltiples ejecuciones obteniendo las conclusiones pertinentes.
- 5) El grupo debe poder explicar los motivos de los resultados obtenidos. Y los tiempos del sistema.
- 6) Debe haber una clase Main que al correrla, inicie los hilos.

Entregar:

- a) 1 archivo de imagen con el diagrama de clases en buena calidad.
- b) 1 archivo de imagen con el diagrama de secuencias en buena calidad.
- c) El código fuente Java (proyecto) de la resolución del ejercicio. El proyecto debe incluir librerías y extensiones necesarias, y debe poder correr en cualquier máquina independientemente del sistema operativo o IDE utilizada.
- d) 1 log y la explicación de los resultados.
- e) **Un informe obligatorio**, en formato pdf, con estructura formal (carátula, integrantes, desarrollo, etc.), donde se detalle:
 - i) Todo el trabajo realizado.
 - ii) Decisiones de diseño tomadas y su justificación (tiempos, etc)
 - iii) Las conclusiones obtenidas en base a los resultados
 - iv) Pueden incluir cualquier otra explicación que crean pertinente para el trabajo

Subir al LEV el trabajo **TODOS** los participantes del grupo.

Fecha de entrega

Hasta Lunes 24 de Abril - 23:59 hs

Fecha de defensa

Martes 25 de Abril