

Programación Concurrente 2022

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 adquisición de datos existen dos buffers y tres categorías de actores. Los buffers se denominan *Buffer Inicial* y *Buffer de validados* y tienen cada uno una capacidad de 100 datos (trabajan a pura pérdida, es decir si están llenos al venir un creador se descarta el dato). Los actores del sistema pueden ser Creadores de Datos, Revisores de Datos o Consumidores de Datos.

El ciclo de funcionamiento normal del sistema comienza con la creación de un dato por parte de un "Creador de Datos". Este proceso lleva un tiempo aleatorio en ms (no nulo, a elección del grupo); una vez creado es almacenado en el *Buffer Inicial*. En este buffer debe permanecer hasta que el total de "Revisores" hayan revisado el mismo, la revisión del dato lleva un tiempo en ms (no nulo a elección del grupo). Una vez que todos los "Revisores" hayan revisado el dato, el último "Revisor" guardará una copia del mismo en el *Buffer de Validados*. Los "Revisores" no pueden revisar más de una vez cada dato.

Los consumidores de datos son los encargados de eliminar los datos de ambos buffers, siempre y cuando ya hayan sido validados; la eliminación de un dato lleva un tiempo en ms (no nulo a elección del grupo).

Tips:

- ☐ Los datos tienen un atributo “reviews” el cual registra cuántos revisores han revisado el dato hasta el momento.
- ☐ Los datos deben ser revisados por todos los revisores del sistema para pasar al buffer de validados. Un revisor no puede revisar más de una vez cada dato.
- ☐ Una vez que el dato alcanzó la cantidad de revisiones, debe copiarse en el buffer de validados, sin borrarlo del buffer inicial. Este proceso es llevado a cabo por el último revisor que revisa el dato.
- ☐ Los Consumidores de datos, leen los datos del “buffer de validados” y los eliminan de ambos buffers.
- ☐ Al finalizar la ejecución es necesario verificar cuantos datos revisó cada revisor.
- ☐ Es necesario validar que cada revisor haya revisado solo una vez a cada dato.

El sistema debe contar con un LOG con fines estadísticos, el cual registre cada 2 segundos en un archivo:

- Cantidad de datos procesados.
- Ocupación del “Buffer Inicial”.
- Ocupación del “Buffer de Validados”.

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 buffer inicial vacío, un “Creador” genera un dato y lo deposita.
 - b) Luego, un “Creador” y un “Revisor” llegan al mismo tiempo al buffer. Mostrar qué pasa con cada uno.
- 3) Modelar e implementar un sistema con los dos buffers con la capacidad mencionada y 4 Creadores de datos, 2 revisores de datos y 2 consumidores de datos.
- 4) Las demoras del sistema (creación, revisión, consumo) deben configurarse de tal manera de poder procesar 1000 datos (desde creación hasta consumo) en un periodo mínimo de 5 segundos y máximo de 10 segundos.
- 5) El grupo debe poder explicar los motivos de los resultados obtenidos. Y los tiempos del sistema.
- 6) Debe hacer 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 25 de Abril - 23:59 hs

Fecha de defensa

Martes 26 de Abril