Universidad Nacional de Córdoba  
Facultad de Ciencias Exactas, Físicas y Naturales

****

Programación Concurrente

2022

|  |  |
| --- | --- |
| Alumnos: | Matrícula: |
| * Arnolfo, Emanuel | 41.602.041 |
|  |  |
| * Raya Plasencia, Matias | 40.089.058 |
|  |  |

# 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).

# Desarrollo

Como primera instancia en el desarrollo del trabajo práctico, pensamos y luego realizamos el diagrama de clases, para así tener una mejor perspectiva del trabajo a realizar. Una vez que comenzamos a trabajar, nos dimos cuenta que teníamos que mejorar algunas cuestiones sobre este diagrama para que se adaptara mejor a como íbamos desarrollando el código del programa.

Una vez que tuvimos el diagrama de clases y el código en una etapa más desarrollada, procedimos a realizar el diagrama de secuencia, en el cual comenzamos a debatir como debería ser la implementación. Ahí debatimos como encarar el problema y analizamos distintas posibilidades sobre como implementarlo.

Una vez que tuvimos ambos diagramas, avanzamos sobre la finalización del código. Posteriormente analizamos distintas alternativas para solucionar los distintos problemas de concurrencia, y de sincronización, que se nos fueron presentando mientras estábamos finalizándolo.

Para todo el desarrollo del código decidimos utilizar la herramienta Git y GitHub, para así poder ir realizando un gestionado de versiones y nos pareció una manera práctica y sencilla de que todos tuviésemos el código de manera espontánea y en la última versión trabajada. Además, realizamos un cronograma con Trello en el cuál íbamos tildando las partes que íbamos completando del código.

# Conclusiones

Una vez finalizado el trabajo práctico, realizamos una reunión vía meet para analizar el código y poner en común la forma en la que cada uno fue analizando la forma de resolución. Al realizar la actividad propuesta nos encontramos con el inconveniente de como poder crear los distintos datos para así posteriormente almacenarlos en su respectivo buffer. Una vez solucionado este inconveniente pasamos a ver como podíamos acceder a los buffers desde los distintos actores, para ello se creo una clase buffer y de esa manera poder trabajar con ambos buffers en simultaneo eliminando esa complicación.

Analizando lo que se obtuvo en el archivo LOG, pudimos observar que en no se llegan a igualar los tamaños de los buffers ya que se poseen más creadores que consumidores y se tiene la demora que se produce por los revisores.