



Departamento de Programación

Facultad de Informática
Universidad Nacional del Comahue



Programación Concurrente

Examen Parcial - 07/11/2025

Nombre y Apellido: _____ Carrera: LCC / LSI / PI

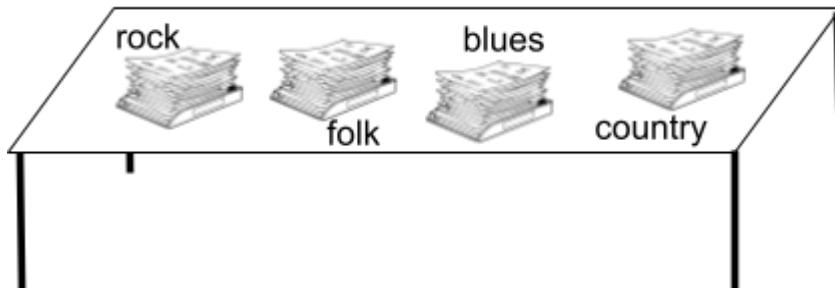
Legajo: _____ Número de hojas entregadas (sin enunciado) _____

- Implemente cada ejercicio utilizando el mecanismo de sincronización que considere adecuado (semáforos y monitores/locks y condiciones).
- Se puede utilizar “sout(...)”, en reemplazo de “System.out.println(...)”
- Las soluciones planteadas deben estar libres de **deadlock**, y **espera activa**
- Las soluciones deben contemplar las propiedades de **seguridad y viveza**.

Problema 1: Bob Dylan

Bob Dylan es un músico que dedica su vida a componer canciones. Cada pocos segundos es capaz de componer una nueva canción, la cual puede ser de distintos géneros musicales: **rock**, **folk**, **country** o **blues**. Él utiliza una regla general en la que todas las canciones que sean del mismo género musical deben tener la misma duración (duracionR, duracionF, duracionC y duracionB).

Bob prefiere escribir sus canciones en papel, y cada vez que termina de componer una nueva canción la coloca sobre una mesa. Como es muy organizado con sus producciones las va colocando separadas en bandejas según su género musical.



Bob tiene un grupo de productores que se encargan de agregar canciones a los nuevos álbumes y un manager que se encarga de promocionar cada nuevo álbum. La primera acción al crear un nuevo álbum es asignarle un **nombre**. Esta tarea le corresponde al manager.

Cada productor va tomando canciones de la mesa, y las va agregando al nuevo álbum, siempre que sea posible. Cada álbum tiene una **duración** que no puede ser superada en ningún caso (duracionAlbum), sin importar el número de canciones que haya en él. Si no se supera esta duración, la canción puede ser agregada al álbum.

Cada álbum puede tener mezcla de géneros.

Cada Productor se ocupa de un género en particular, aunque puede haber más de un productor por género. Por ejemplo, puede haber 2 productores para rock, 1 productor para blues, 3 productores para folk y 1 productor para country.

Cuando un productor no puede agregar una canción al álbum se considera que el álbum está terminado y debe avisar al manager de Bob para que se encargue de promocionarlo.



Una vez promocionado el álbum terminado, el mánager dedica un tiempo para pensar el nombre que le va a poner al próximo álbum que va a crearse. Una vez tenga el nombre decidido, los productores pueden continuar con la tarea de agregar nuevas canciones a este álbum y el mánager espera a que este se complete. Inicialmente, no hay ningún álbum creado.

Observaciones:

- No es necesario implementar las canciones, solo se requiere saber la cantidad que hay disponibles de cada género. Considere un hilo “GeneradorCanciones” que tiene un método generarCancion(...) que devuelve el género de la nueva canción
- El primer productor que no pueda colocar una canción en el álbum, le avisará al manager para que lo promocione, sin importar si hay otra canción disponible que pueda llenar la duración que queda.

Problema 2: Buffer oscilante

Algunos sistemas de buffering permiten que las operaciones de insertar y extraer se realicen en simultáneo. Debido a ello, no se puede asumir que la ejecución simultánea de dos operaciones sobre un recurso de tipo cola mantenga el estado consistente. Por lo tanto los accesos sobre dicho recurso deben ser **excluyentes**.

¿Qué ocurre si un buffer estuviera formado por dos colas independientes? Pareciera que existe la posibilidad de insertar en una y extraer de la otra de forma simultánea. Pero es necesario asegurar que el buffer, en su conjunto, respeta una política FIFO.

Para ello se debe (oscilar / cambiar) entre una cola u otra en momentos clave. Veamos un escenario de ejecución:

- a. Inicialmente las dos colas están vacías. Una de ellas etiquetada para insertar y la otra para extraer. Si un hilo quiere extraer datos no puede hacerlo porque no hay ningún dato (Más precisamente porque no hay datos en la cola de extraer).



- b. Si un hilo quiere insertar datos debe hacerlo en la cola que tiene la etiqueta de inserción.

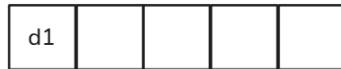
Se inserta el primer elemento d1 en la cola de inserción





Se oscilan las colas, dado que la cola de extracción está vacía. Es decir, se deben cambiar las etiquetas de insertar y extraer para posibilitar la simultaneidad entre una extracción y una inserción.

Cola 1: EXTRAER



Cola 2: INSERTAR



- c. Supongamos que llegan dos hilos que quieren insertar datos (d2 y d3), tendrán que insertar en la cola etiquetada para ello y lo tendrán que hacer en exclusión mutua:

Cola 1: EXTRAER



Cola 2: INSERTAR



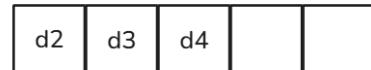
- d. Si ahora un hilo quiere insertar y otro extraer, pueden realizar la operación de forma simultánea. Finalmente, como no quedan elementos en la cola de extracción se debe volver a oscilar:

Se inserta d4

Cola 1: EXTRAER

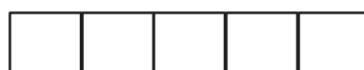


Cola 2: INSERTAR



Se extrae d1 y se oscilan las colas

Cola 1: INSERTAR



Cola 2: EXTRAER



Como puede observarse la situación permite que de nuevo puedan ejecutarse de forma simultánea una operación de inserción y una de extracción. La oscilación SOLO OCURRE cuando la cola de extracción queda vacía.

Se debe implementar un sistema de buffering como el indicado, considerando *hilos insertores* e *hilos extractores*, y el recurso compartido. Considere **tamaño ilimitado**