Caso 1: Informe

Nombres integrantes:

* Jaime Esteban Alfonso, 202116525
* Sebastián casanova Ospina, 202115116
* Paola Andrea Campiño, 202020785

# Diseño:

Como solución se ha planteado la necesidad de generar 3 clases:

* Buzon: es la clase de ser un intermedio entre los procesos, pues esos son encargados se almacenar una cierta cantidad de mensajes.
* Consola: La consola es la clase encargada de recibir los parámetros de entrada, el número de mensajes, al igual que el tamaño de los buzones intermedios y buzones extremos.
* Procesot: La clase encargada de hacer los cambios/Transformaciones en los mensajes

En cuanto a relaciones se podría visualizar de la siguiente manera:

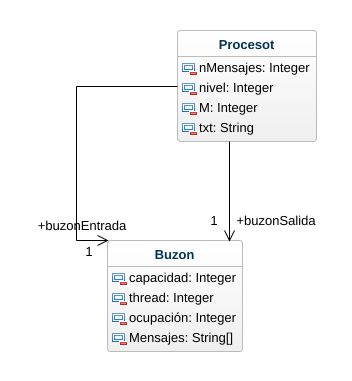


Diagrama [1]: Un proceso tiene un buzón de entrada y salida, por los cuales los mensajes van a pasar.

Debido a esto se ha establecido que la clase procesot va a ser quien herede de Thread de manera que cada procedimiento sea una Thread, pues de esta forma se está garantizado que cada procedimiento esté ocurriendo de manera concurrente.

Así mismo se ha garantizado que el proceso inicial de enviar es semiactivo haciendo uso del método en Buzon llamado poneri() donde se hace uso yield() para lograr esta espera semiactivo, en donde se garantiza que el proceso no pase por un estado de dormido. Esto también pasa en el proceso final esto se puede ver con el método quitarf() en donde se hace uso el método yield con el fin de garantizar este tipo de espera.

Adicionalmente es importante resaltar el método quitar () y poner () presente en la clase Buzón que tienen como objetivo comunicar procesos y buzones intermedio se presentan con una espera pasiva, esto se puede ver con un while (ocupación == capacidad) wait(), esto quiere decir que el proceso debe pasar por un estado de dormido.

# Validación:

Con el fin de evaluar el programa se estableció los siguientes casos para establecer que los resultados llegaran a ser correctos.

## Verificación 1:

Lo primero que hay que tener en cuenta es que todo string generado debe finalizar con FIN, FIN, FIN ya que estos mensajes deben haber pasado por al menos 3 procesos.

## Verificación 2:

El Thread por el que el mensaje se está transformando no puede variar por lo que el mensaje final no puede tener variaciones en el segundo digito de la transformación:

M1T11T21T31FIN, FIN, FIN

## Verificación 3:

Los niveles de transformación por los que pasa el mensaje deben ir ascendente.

## Verificación 4:

La cantidad de mensajes a transformar debe ser igual a la cantidad de subconjuntos presentados en consola.

## Caso 1:

### Entradas: M= 3 , tamaño buzón intermedio = 1, tamaño buzón extremo=3

Salida Esperada [posible resultado]: M1T11T21T31FIN, FIN, FIN, M2T12T22T32FIN, FIN, FIN, M3T12T22T32FIN, FIN, FIN

Cabe resaltar que el orden del mensaje es irrelevante al igual que la Thread por la que el mensaje va a pasar.