**Infraestructura Computacional**

**Caso #1**

**Luis Miguel Gómez Londoño – 201729597**

**Daniel Andrés Bernal - 201519654**

**Diseño y Funcionamiento**

Se desarrolló el programa para siguiera estrictamente los requerimientos estipulados en el documento del caso…

**Cliente**

El cliente en este caso posee un id, una referencia al buffer y además una lista de mensajes, como también un número de consultas (cuantos mensajes tiene). Lo único que hace el proceso del cliente es crear tantos mensajes como número de consultas y nada más… Por cada mensaje creado, se hace un intento de almacenarlo en el buffer…

**Mensaje**

El mensaje cuenta con un id, una referencia al buffer y al cliente a quien le pertenece… Además, contiene una variable entera de contenido inicializada en 0. Cuando el mensaje esta sin modificación alguna, esta es igual a 0. Cuando el mensaje ha sido modificado, su valor es modificado a 1 (a través del método *cambiar()*).

**Servidor**

El servidor cuenta únicamente con un id y con una referencia al buffer. El proceso del servidor es el siguiente: El servidor constantemente intenta hacer un retiro del buffer (con el método *buffer.retirar()*), luego se verifica que se haya podido hacer el retiro, y antes de hacerle cambio alguno, despierta los procesos que duermen dentro de él (ya que, si existe y se retiró, ya hay capacidad en el buffer disponible). Se realiza el cambio y se reinicia el proceso de retiro de mensajes. En caso que no hayan mensajes en el buffer (que esté vacío), se utiliza el método *yield()* para liberar el procesador que usan los threads de servidor. Si ya no hay mensajes restantes en el buffer para que procese, se sale del ciclo y el servidor muere.