Caso 1

Se hicieron 4 clases para el funcionamiento del caso: Cliente (Thread), Servidor (Thread), Mensaje y Buffer (donde se encuentra el main).

El cliente genera un número fijo de mensajes que va a mandar al servidor para procesar. En este caso, la consulta se hace directamente desde el cliente hacia el buffer con un mensaje de parámetro. El buffer lo recibe y lo intenta almacenar en un arreglo de mensajes que es de donde el servidor va a cogerlos para modificarlos. Si no puede almacenar un mensaje porque el buffer está lleno, el thread se duerme sobre un nuevo objeto (llamado lleno) hasta que el servidor atienda un nuevo mensaje y le notifique que ya quitó uno del buffer. Este objeto entra a una sección crítica para asegurar exclusión mutua: solamente puede entrar un thread a la vez. Una vez se despierte, se tiene que hacer la verificación otra vez (de preguntar si el buffer está lleno) para ver si puede entrar. Dentro del buffer, este objeto se debe sincronizar con una sección crítica para verificar que solamente 1 entre a la vez (synchronized(this)).

El servidor, por su lado, está siempre intentado procesar mensajes del buffer. Si entra al buffer y no encuentra ninguno para procesar, se duerme hasta que un cliente almacene su mensaje en el buffer y lo despierte. Para procesar un mensaje el servidor accede al buffer y elimina el que se encuentra en la primera posición ([0]), que se encuentra dentro de una sección crítica dentro del buffer (synchronized(this)). Luego el servidor modifica el contenido del mensaje y le agrega uno para verificar que sí haya servido.

La clase mensaje tiene un contenido (en este caso un int) y un identificador.