**CASO I**

**Manejo de la concurrencia**

**Diseño y funcionamiento del sistema**

En el desarrollo del sistema se incluyeron las clases Cliente y ServidorThread que extienden de la clase Thread. La clase Cliente tiene el número de consultas a realizar tiene como atributo de tipo entero y, al igual que la clase ServidorThread, un atributo estático con referencia al Buffer en el que el cliente almacena los mensajes y el servidor los retiras. De esta manera, la clase Buffer modela el objeto mediante el cuál se establece la comunicación entre los clientes y servidores de la aplicación.

La clase Buffer posee como atributos el tamaño, una lista con los mensajes almacenados, el número de clientes que restan por atender y los objetos auxiliares lleno y vacío sobre los cuales se realiza la sincronización del almacenamiento y retiro de mensajes. Dentro de esta clase se implementan los métodos almacenar y retirar mensaje, y retirar cliente. El primer método mencionado anteriormente, permite que un cliente almacene un mensaje en el Buffer, en caso de que este se encuentre lleno realiza una espera pasiva mediante el método *wait()* sobre el objeto lleno hasta ser notificado por uno de los servidores. El método *retirar(),* al ser invocado por el servidor, permite que este responda un mensaje de los clientes. En caso de que el Buffer se encuentre vacío realiza una espera activa, cediendo el procesador después de cada intento a partir del método *yield().* En cada intento se verifica si existe un mensaje en el Buffer o si el número de clientes que quedan por ser atendidos es cero. Por un lado, de cumplirse el primer caso el servidor remueve un mensaje del Buffer y notifica al cliente que espera almacenar un mensaje. Adicionalmente, responde el mensaje y notifica mediante el método *notify()* al cliente que se encuentra en espera pasiva sobre el objeto mensaje. Por otro lado, si se cumple el segundo caso, el servidor termina la espera activa y retorna *null* y termina su ejecución. El funcionamiento descrito anteriormente puede resumirse a:



Finalmente, se replica este comportamiento para un número dado de clientes en el sistema y un número dado de servidores. Así, las consultas de un mismo cliente pueden ser atendidas por distintos servidores. El número de servidores, clientes, consultas por cliente y el tamaño del Buffer se carga del archivo almacenado en la dirección “./data/data.txt”, el formato utilizado para la carga de datos corresponde a:

./data/data.txt

numClientes = 5

numServidores = 2

numMensajesCliente = 10

bufferTam = 5