

## **Diseño**

- **Cliente:** El cliente, que hereda de Thread, posee un identificador, un arreglo de mensajes y una referencia al buffer. Este tiene la relación con mensaje, que es la manera con la cual se comunica con el buffer y además tiene una relación directa con el buffer para verificar la posibilidad de finalizar los thread de servidor.
- **Mensaje:** Los mensajes son objetos que tienen el mensaje por enviar, además del Cliente que lo envió.
- **Servidor:** El servidor, que también hereda de Thread, tiene un estado el cual indica si está activo o inactivo el servidor. Además, el servidor tiene un mensaje, el cual indica que mensaje está atendiendo y es el que posteriormente se modifica para dar respuesta al cliente.
- **Buffer:** Esta clase contiene una lista en la cual se guardan los mensajes, una lista donde se almacenan los servidores que están atendiendo las solicitudes y una capacidad máxima de mensajes. Este además es el que conecta los mensajes con los servidores a través de los métodos almacenar que almacena los mensajes, retirar, que retira y responde los mensajes y detenerServidor que detiene los servidores cuando no hay más mensajes. En esta clase también se encuentra el método main que inicializa la aplicación.

## **Funcionamiento**

- El main de la aplicación en un principio va a leer un archivo en el cual se encuentra la información de la cantidad de clientes, la cantidad de mensajes que envían los clientes, la cantidad de servidores que van a atender los mensajes y el tamaño del buffer el cual atenderá a los clientes y a los servidores.
- Al crear un cliente va a iniciar su ejecución la cual consiste en: Crear la cantidad de mensajes que entran como parámetro y guardarlas dentro del buffer. Para esto se ejecuta el método almacenar del buffer. El cual consiste en verificar si la cantidad de mensajes que hay dentro del buffer es diferente a el tamaño de este, en el caso que sea igual, pone a esperar a los clientes en la bolsa del buffer. De lo contrario, agrega el mensaje al buffer y pone a esperar al cliente hasta que obtenga una respuesta de un servidor. Cabe resaltar que ambas esperas son dentro de la bolsa del buffer.
- Seguido después de la creación y ejecución de los threads del cliente, se crean y ejecutan los threads de los servidores. Cabe resaltar que toca guardar todos los servidores en un arreglo del buffer para poder detenerlos en el caso de que no haya clientes esperando. Entonces al ejecutar el thread del servidor, este procederá a retirar los mensajes del buffer y a modificarlos para que tengan una respuesta diferente. Para esto, ejecuta el método retirar, el cual verifica si que cuando haya mensajes en el buffer, el servidor retira el mensaje de este y lo modifica para luego notificar al cliente de dicho mensaje que ya esta lista la

respuesta. Por ultimo, notifica a todos los clientes que estén en la bolsa de espera para almacenar un mensaje, para que lo intenten almacenar y ejecuten el método almacenar.

- Por ultimo, cuando el servidor notifica al cliente que ya hay una respuesta al mensaje, procederá a realizar el ultimo comando de la ejecución, el cual consiste en detener el servidor que lo está atendiendo. El método detener consiste en verificar si no hay mensajes en el buffer para que los servidores atiendan, los pone a dormir 50 milisegundos, seguido se vuelve a preguntar si hay mensajes, si no hay, detiene el servidor cambiando su estado en detenido.
- Cabe resaltar que el servidor después de retirar un mensaje hace yield para liberar espacio en el procesador y se pone en espera activa para atender otro mensaje.

### **Métodos sincronizados**

- Almacenar: El método almacenar se encuentra sincronizado en dos elementos, inicialmente, cuando el buffer está lleno, se sincroniza con respecto al cliente que envió el mensaje. Asimismo, al momento de ingresar un mensaje al buffer este se sincroniza con respecto al buffer para evitar que se ingresen dos o más mensajes al tiempo, pasando la capacidad del buffer.
- Retirar: El método se sincroniza con respecto al servidor que va a atender el mensaje, para evitar que se puedan retirar varios mensajes repetidos al tiempo y poder atender todos los mensajes.
- DetenerServidor: En este método se sincroniza con respecto al servidor, para evitar que todos los servidores realicen la comprobación al tiempo y realizar la notificación varias veces.