# Entrega 1

## Controler

El Controler es quien inicia la secuencia, definiendo los parámetros de tamaño de buffer, cantidad de servidores y cantidad de clientes. De la misma forma, se encarga tanto de la inicialización de los hilos de operación de clientes y servidores, como del buffer común que utilizan tanto clientes como servidores.

## Buffer

Esta clase es la encargada de almacenar, con un espacio limitado, los mensajes en espera de respuesta junto con las respuestas a cada una de las solicitudes presentadas. En términos de sincronización, esta clase tiene dos métodos sincronizados *leer()* y *escribir()*, que permiten la apropiación del objeto de manera no determinística por parte de un lector o un escritor por igual. No obstante, se tiene que cumplir que sólo un lector o escritor puede estar realizando cambios con respecto a un mensaje, esto permite que no se generen inconsistencias en la información presentada. Asimismo, se tienen restricciones para que una solicitud de cliente y un servidor puedan operar dentro del buffer y solicitar los servicios respectivos. Para el primero se tiene que cumplir que el buffer no esté lleno; para el segundo debe existir por lo menos un mensaje en el buffer para ser procesado. Si las condiciones para cada actor no se cumplen, el hilo de operación libera el *lock* que tienen sobre el objeto y se queda esperando por una notificación. En términos del problema, estas notificaciones se emiten cuando hay espacio disponible en el buffer para los clientes, o cuando hay mensajes existentes dentro del sistema para los clientes.

Otra de las características del buffer es que almacena la cantidad limitada de mensajes en un arreglo, para el cual se diseñó un esquema de atención FIFO basado en colas, con el fin de administrar la referenciación de los mensajes adecuadamente, omitiendo el riesgo de que se pueda editar la respuesta a un mensaje que no corresponde, o se eliminen las referencias del arreglo.

## Cliente

En primer lugar, se puede resaltar que el cliente se declara como un hilo de operación que se va a estar ejecutando en paralelo con otros clientes y con los servidores. Dicha clase solo tiene interacción con un mensaje, sobre el cual especifica el contenido para ser procesado por los servidores. Por otra parte, se encarga de mostrar al usuario la respuesta que obtuvo para su pregunta al sistema.

## Servidor

En primera instancia, el servidor está directamente relacionado con el buffer y es el encargado de responder los mensajes solicitados al buffer. Al igual que el cliente, este es declarado como un hilo de operación que se ejecuta en paralelo con clientes y otros servidores. Se tiene que una de las características de concurrencia a resaltar de la clase es que, cuando ha obtenido un mensaje y ha terminado de procesar la solicitud, esa instancia del servidor cede el procesador al sistema para que sea empleado por otro hilo de operación.

## Mensaje

Esta clase se considera una de las más importantes, pues es el objeto en cuestión sobre el cual se hacen las ediciones de valores. En primer lugar, se poseen métodos sincronizados que hacen la modificación de la respuesta y la solicitud, para cada llamado que se hace en las otras clases ya mencionadas. Lo anterior supone que solo un servidor o cliente puede estar modificando los atributos del objeto. Por otra parte, el mensaje se encarga de retornar la respuesta cuando esta no sea nula, lo cual implica que se tiene una espera activa del objeto hasta que esto ocurra; cediendo el procesador a otro hilo de operación que lo pueda necesitar.

## Main (view)

En esta clase, para reducir el acoplamiento, se llama una clase auxiliar que genera todos lo threads para cada uno de los objetos. La clase, para generar los objetos, llama a su vez otra clase que es la encargada de leer del archivo de propiedades.

Para iniciar la ejecución se corre la clase view.java en la carpeta View.

Para configurar el programa, se modifica el archivo config.propierties. Es conocido que por las diferentes variables %PATH% de los equipos, la ruta al archivo sea diferente, por esto, en caso de fallar, el programa imprime la ruta correcta donde se debe poner el archivo.