Yeisson Steven Tarazona Bernal

ys.tarazona

201617841

Caso 1

* Cliente – Mensaje:

El cliente crea el número de mensajes indicado en el archivo de configuración, le asigna un numero aleatorio como contenido de cada mensaje. Luego de creado un mensaje, el cliente hace un llamado al método enviar() de la clase Mensaje, donde se realiza la comunicación del mensaje con el buffer y el cliente se queda dormido sobre el mensaje, haciendo uso de un wait(), donde queda a la espera de que el servidor lo despierte. Una vez creados, enviados y respondidos todos los mensajes, el cliente crea un mensaje extra, desde el cual hace un llamado al método fin() de mensaje, el cual le comunica al buffer que el cliente ha terminado de enviar mensajes.

* Mensaje – Buffer:

Como se mencionó anteriormente, la clase Mensaje contiene el método enviar(), en este el mensaje envía la referencia de sí mismo al buffer, haciendo uso del método enviarMsg() de la clase Buffer, en este el buffer revisa constantemente si el número de mensajes guardados no ha superado la capacidad, si la supera, se duerme hasta que se haga un llamado al método leerMsg() el cual, si hay mensajes en el buffer, saca, retorna un mensaje y hace un notify(), así una vez haya espacio en el buffer se guarda el mensaje.

* Servidor – Buffer:

El servidor le pide constantemente al buffer el número de clientes activos, si aún hay clientes, llama al método leerMsg() del buffer, hasta que este le retorne un mensaje, con el cual hace un llamado al método responder() de Mensaje, luego hace un notify() sobre el mismo mensaje, el cual despierta al cliente dormido que se encontraba esperando una respuesta de parte del servidor. Una vez el contador de clientes del Buffer llega a cero, el servidor termina su ejecución.

* Servidor – Mensaje:

Como se mencionó anteriormente, el servidor al obtener un mensaje, hace un llamado al método responder() de la clase Mensaje, el cual le suma 1 a su contenido, para así luego notificar al cliente.