Juan Camilo Lara – 201424726

Maria del Rosario León - 201423755

Andrea Carolina Montoya - 201713889

Caso 1

Servidor - buffer

El servidor revisa si hay clientes en el buffer, si aún hay clientes accede de manera sincronizada a la lista de mensajes, saca un mensaje y lo porcesa. Una vez el mensaje haya sido procesado notifica a los clientes de manera sincronizada a través de notify().

Cliente – mensaje

El cliente crea un objeto tipo mensaje el cual genera un valor aleatorio para su contenido.

Cliente – buffer

Mientras que el numero de mensajes restantes sea mayor a cero se encarga de generarlos. Se envian al buffer de manera sincronizada para evitar que hayan dos threads manipulando la lista al mismo tiempo. Una vez el cliente termina de enviar los mensajes, sale del buffer.

Descripción general

El buffer contiene el main el cual lee el archivo que contiene la información del numero de servidores, clientes y mensajes a generar. Luego crea un objeto buffer y en orden crea los clientes y servidores y los pone a correr. Los clientes crean los mensajes según el número y el tipo que se le da por parámetro. Una vez generado el mensaje intenta enviarlo al buffer si este tiene espacio. La verificación del espacio en el buffer es un método sincronizado. El método de recibir en el buffer es sincronizado, lo que quiere decir que si dos cliente intentan meter un mensaje al mismo tiempo, uno de ellos no podra y quedara en espera activa.

Los servidores por su lado estan pendientes y mientras haya mensajes los recuperan de la lista del buffer. Este es un método sincronizado para que dos servidores no puedan sacar el mismo mensaje al tiempo. Una vez el mensaje es procesado, se notifica al cliente, es decir notify().