

## Práctica 5: Mensajes

- 1- Se desea implementar un servidor que provea el siguiente comportamiento: debe aceptar la conexión de un cliente y luego generar un número pseudoaleatorio entero en el rango  $[0,10]$ . El cliente debe intentar adivinar el número que ha generado el servidor. Para ello envía sucesivamente mensajes conteniendo un número. Cada mensaje es contestado por el servidor indicando si el cliente acertó o no. Cuando un cliente acierta, el servidor cierra la conexión y queda disponible para aceptar un nuevo cliente.
- 2- Modificar la solución anterior de manera tal que un servidor pueda procesar los pedidos de varios clientes.
- 3- Se desea implementar un servidor que recibe mensajes de dos clientes distintos y muestra por pantalla la concatenación de un par de mensajes (uno de cada cliente). Cada cliente continuamente envía un mensaje y espera una cantidad aleatoria de tiempo. El servidor recibe los mensajes de los clientes y por cada par de mensajes recibidos muestra por pantalla la concatenación de los mensajes recibidos.
- 4- Modificar la solución anterior de manera tal de permitir que el servidor pueda aceptar conexiones de varios pares de clientes concurrentemente.
- 5- Considerar dos procesos que interactúan de la siguiente manera. El primer proceso envía mensajes al segundo (alternando envío de mensajes y espera). El segundo proceso recibe mensajes y los muestra por pantalla (alterna también con períodos de espera). Suponer que el segundo proceso es más lento que el primero y que se requiere que en todo momento la cantidad de mensajes enviados por el primer proceso y que no han sido procesados por el segundo no supere una cantidad  $N$  de mensajes fijada a priori. Dar una implementación a este problema.
- 6- Se desea implementar un servidor que funciona de la siguiente manera. Recibe mensajes numéricos de dos clientes. Cada vez que recibe un mensaje del cliente 1 lo muestra por pantalla multiplicado por el valor contenido en el último mensaje recibido del cliente 2 (considerar que el servidor muestra 0 cuando no ha recibido mensajes del cliente 2).
- 7- Modificar la solución anterior de manera tal que el servidor no procese ningún mensaje del cliente 1 hasta tanto no haya recibido un mensaje del cliente 2.