Diseño Funcional **Comunicación Cliente-Servidor con Threads y Sockets**

Ejercicio 2

Integrantes Grupo 5:

* BIANCHI, JUAN ALBERTO
* BROCANI, AGUSTIN NICOLAS
* ESPINOSA, THOMAS GABRIEL
* PALACIOS, NERI MARTIN
* SANZ, ELISEO TOMÁS
* MY, TOMÁS

**pie de página**

**Sistema Cliente-Servidor de mensajes.**

### **Abstract**

Se simula un sistema donde varios clientes envían mensajes a un servidor. El servidor calcula la longitud de cada mensaje y responde. Los clientes activos son gestionados concurrentemente, y los demás esperan en una cola. La actividad continúa mientras existan clientes enviando mensajes al servidor.

### **Idea central**

* Un Cliente intenta conectarse al Servidor. El Servidor responde si se conecta, si espera o si rechaza la conexión.
* En el Servidor se implementan dos colas: una para clientes en espera y otra para clientes activos.
* Deben existir límites máximos de atención y espera en el Servidor.
* Una vez que el Cliente puede conectarse, envía un mensaje al Servidor. El Servidor calcula la longitud de dicho mensaje y se lo retorna al Cliente que lo envió.
* La finalización automática del servidor se da al desconectarse todos los Clientes. La misma puede realizarse luego de que al menos haya ingresado un Cliente.

### **Explicación de funcionamiento**

El servidor se inicia y queda escuchando conexiones entrantes.

Cuando se atiende al primer cliente, el servidor comienza a verificar si falta atender a otros. Esta verificación no se hace al inicio, sino que se activa una vez que se comienza a atender al primer cliente.

Todo cliente que se conecta pasa primero por la cola de espera del servidor, sin importar si hay lugar disponible o no.

Se definen dos límites importantes en el diseño:

* N: cantidad máxima de clientes que pueden ser atendidos simultáneamente.
* M: cantidad máxima de clientes que pueden esperar en la cola cuando no hay threads disponibles.

Cuando un cliente llega:

* Si hay lugares activos, el cliente será atendido.
* Si no hay lugares activos, pero hay lugar en la cola de espera, quedará en espera.
* Caso contrario, la conexión será rechazada.

Una vez que el servidor atiende a un cliente, este puedeenviar mensajes al servidor**,** que los procesa y responde.

El cliente permanece conectado hasta que el usuario lo cierre manualmente mediante un mensaje de “fin” o se rompe la conexión.

Al finalizar la conexión de un cliente:

* Se libera su lugar.
* El servidor acepta automáticamente a otro cliente que esté esperando.

El servidor sigue funcionando mientras haya clientes conectados o esperando.

Cuando todos los clientes se han desconectado y no hay más en la cola, el servidor se cierra automáticamente y libera todos los recursos.

### 

### **Diagrama**

