

## Programa de Ingeniería de Sistemas Laboratorio de Sistemas Operativos Proyecto Segundo Parcial-2-2025 Grupo A

**Fecha:** 22 de octubre de 2025

### Planteamiento del problema

Se deberá desarrollar dos programas en C que implementen la funcionalidad de un chat simple entre usuarios conectados a la red, usando sockets con dominio AFT\_INET. Se deberá permitir la conexión simultánea de varios clientes (máximo 5), los cuales estarán conectados a un servidor central que se encargará de reenviar los mensajes.

### Funcionamiento del cliente

El cliente deberá ingresar por entrada estándar (o como argumento de línea de comandos) el login (nombre de usuario) con el cual se registrará en el chat. Cuando un cliente se conecte al servidor de manera exitosa, el cliente deberá enviar un mensaje al servidor, informando el nombre de usuario deseado. El servidor deberá responder con un mensaje de confirmación, o de error si el nombre de usuario está siendo usado.

Posteriormente, el cliente tendrá que ejecutar dos tareas (usando hilos): una tarea recibirá líneas de texto por entrada estándar, y las enviará al servidor, quien las replicará a los demás clientes conectados. La otra tarea del cliente deberá esperar por mensajes enviados por el servidor, y los mostrará por la salida estándar (la pantalla).

### Comandos del cliente

El usuario puede enviar mensajes o comandos. El cliente recibe líneas de texto (mensajes) de la entrada estándar y las envía al servidor. Si una línea comienza con el carácter "/" se considerará un comando. El comando por implementar será el siguiente:

/exit: Envía este mensaje al servidor y cierra la conexión. Es necesario enviar el mensaje, ya que el servidor debe informar a los demás clientes que se ha cerrado una conexión. El cliente antes de cerrar debe almacenar todos los mensajes del chat en ./chats/chats.txt.

## Funcionamiento del servidor

El proceso servidor deberá recibir una conexión del cliente, verificar que el nombre de usuario especificado no se encuentre usado en el momento y crear una nueva tarea (hilo) que atenderá esta conexión.

El hilo se encargará de recibir mensajes del cliente conectado y reenviarlo a los demás clientes que también se encuentren conectados para los mensajes públicos.

Cuando un cliente envía el comando /exit al servidor, se deberá enviar un mensaje a los demás clientes conectados.

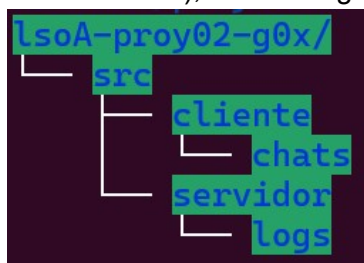
Debido a que cada cliente se gestiona mediante un hilo, se deberá tener una estructura de datos central que permita llevar la cuenta de los clientes conectados (los descriptores obtenidos en la llamada accept). Esta estructura de datos es un recurso compartido que deberá ser protegido mediante algún mecanismo de sincronización.

El servidor repetirá el proceso de recibir y atender solicitudes hasta que se reciba la señal SIGTERM. Una vez recibida esta señal deberá almacenar en ./logs/log-servidor.txt, la fecha y hora de inicio y final de la ejecución del servidor. En este momento deberá informar a todos los clientes conectados y cerrar todas las conexiones.

Nota: La solución entregada debe basarse en los códigos fuente del Taller 06.

## ¿Qué se debe entregar?

En una carpeta denominada **IsoA-proy02-g0X**, (donde X corresponde al número de grupo asignado en este curso), crear la siguiente estructura de carpetas:



En las subcarpetas cliente/ y servidor/, deben incluir los archivos .c, .h y el archivo Makefile. En la subcarpeta cliente/chats se deben almacenar el archivo chats.txt y en la subcarpeta servidor/logs se debe almacenar el archivo logs-servidor.txt. Antes de enviar los archivos fuente de la solución, limpiar la carpeta src/cliente y src/servidor mediante el comando "make clean". Comprimir la carpeta y subir el archivo comprimido (en formato .rar o .zip) mediante el enlace que se defina a la plataforma Univirtual.

**Fecha de entrega:** 29 de octubre de 2025, hasta las 9:00 am.