

# TP 3 - Sockets

2025 - Programación sobre redes - 6AO



**Integrantes:** Tomas Valentin Muruchi, Sebastian Daniel Marcos

# Introducción

El presente trabajo práctico tiene como objetivo implementar una versión distribuida del clásico juego del **Ahorcado**, utilizando una arquitectura **cliente-servidor** basada en **sockets** como mecanismo de comunicación entre procesos (IPC). La propuesta consiste en desarrollar un servidor central capaz de gestionar partidas multijugador en red, mantener el estado de las palabras y coordinar la interacción entre los clientes, quienes se conectan de forma remota para jugar.

La dinámica del juego se mantiene fiel al formato tradicional: el servidor selecciona aleatoriamente una palabra secreta de una lista predefinida y los jugadores, por turnos, intentan adivinarla letra por letra o proponiendo la palabra completa. El servidor administra los turnos, cuenta los errores y mantiene un puntaje acumulado para cada participante.

Este trabajo busca aplicar y afianzar conceptos clave de la materia, tales como la **comunicación entre procesos (IPC)**, la **programación con sockets de red**, el **manejo de concurrencia** y la **definición de protocolos de comunicación**. Asimismo, uno de los principales desafíos del proyecto consiste en garantizar que el servidor pueda manejar múltiples jugadores concurrentes de manera eficiente, asegurando una comunicación fluida y confiable entre los clientes y el servidor para el correcto desarrollo de las partidas.

## Arquitectura del Sistema

El sistema se compone de un servidor y los clientes. El servidor hace uso de un threadpool y la lógica de hangman, definidos en archivos aparte. El servidor se compone de un thread principal para el manejo de partidas, un thread para la escucha de conexiones entrantes y, con el threadpool, 5 threads disponibles para el manejo de conexiones establecidas. Los clientes realizan una conexión inicial con el servidor y, posteriormente, reciben y escuchan mensajes de texto plano.

## Decisiones de Diseño

### Conexiones con los clientes

Las conexiones con clientes están limitadas a 5 a la vez. Se decidió utilizar un thread pool de 5 threads para manejar estas conexiones. Cada file descriptor de conexión se almacena en un vector "connections\_fd" para llevar la cuenta de las conexiones establecidas, así como para tener acceso fácil a todas las conexiones al momento de enviar mensajes a todos los clientes.

Desde el lado servidor, cada thread de conexión comienza esperando un nombre de jugador. Si la conexión no envía un nombre antes de comenzar la partida o envía un nombre

que ya está en uso, se desconecta. Luego de recibir un nombre, la conexión es añadida a otro vector "connections\_players" y la conexión entra definitivamente al juego.

Desde el lado cliente, un thread inicial se encarga de la conexión con el servidor. Una vez realizada la conexión, se delega a un thread "receiver" la recepción de mensajes y a un thread "sender" el envío de mensajes.

## Administración del servidor

Si la conexión entrante al servidor es la primera, se establece como administrador del servidor en una variable "admin\_connection". Puede comenzar la partida y volver a la espera de jugadores una vez que finalice.

## Juego

En el servidor, el thread principal chequea si se está esperando por jugadores, si hay una partida en curso o si la partida finalizó.

Si se están esperando jugadores, se le indica al thread de escucha de conexiones que permita nuevas conexiones entrantes. Luego, se espera que el administrador comience la partida con al menos 2 jugadores conectados.

Una vez comenzada la partida, el thread de escucha de conexiones desconecta cualquier intento de conexión nueva. El servidor muestra y repite el estado de la partida, los intentos restantes y el jugador del turno hasta que la partida llegue a su fin.

Una vez finalizada la partida, el servidor muestra la palabra, si hay ganador y el puntaje de cada cargador hasta el momento. Se espera que el administrador vuelva a la espera de jugadores nuevos.

## Implementación de la Comunicación

La comunicación entre cliente-servidor se realiza en texto plano. Cualquier texto enviado por el cliente se refleja en la terminal del servidor como un mensaje. Los clientes pueden realizar acciones mediante comandos precedidos por un caracter "/". El servidor envía mensajes para informar sobre el estado del juego a todos los jugadores conectados.

Durante sus turnos en la partida, los jugadores pueden escribir "/letra" o "/palabra" para intentar adivinar una letra o palabra, respectivamente. El administrador del servidor puede escribir "/jugar" durante la espera de jugadores para comenzar la partida, o "/lobby" al finalizar la partida para regresar a la espera de jugadores.