

## Propuesta Formativa Obligatoria

### TP: Implementación de un Chat Básico Cliente-Servidor con Sockets y Base de Datos

#### Objetivo:

Aprender a configurar un servidor de sockets en Python que reciba mensajes de clientes, los almacene en una base de datos y envíe confirmaciones, aplicando buenas prácticas de modularización y manejo de errores.

Utilizar los comentarios para explicar tus configuraciones en el servidor.

#### Servidor:

Creá un socket que escuche en localhost:5000.

Usá funciones separadas para:

- Inicializar el socket.
- Aceptar conexiones y recibir mensajes.
- Guardar cada mensaje en una DB SQLite con los campos: id, contenido, fecha\_envio, ip\_cliente.
- Manejar errores (puerto ocupado, DB no accesible).
- Responde al cliente con: "Mensaje recibido: <timestamp>".

#### Cliente:

- Debe tener la capacidad de conectarse al servidor y enviar múltiples mensajes hasta que el usuario escriba éxito.
- Mostrá la respuesta del servidor para cada mensaje.

#### Algunas recomendaciones:

- Usá el módulo sqlite3 para la base de datos.
- Comentá cada sección clave (ej: # Configuración del socket TCP/IP).
- Ejecutá pruebas locales (primero servidor, luego cliente en otra terminal).

#### Entrega:

- La fecha de apertura de la actividad es el 22/4 , finalizando el 29/4.
- Sube el enlace del repositorio de código con la solución propuesta (Github / Bitbucket).
- En caso contrario, sube un archivo comprimido (.zip, .rar) con la solución.