

Proyecto I

Desarrollar una aplicación de chat en Python que permita la comunicación en tiempo real entre múltiples usuarios. La aplicación debe aprovechar el procesamiento en paralelo mediante el uso de hilos (threads) o procesos, para manejar eficientemente múltiples conexiones y mensajes simultáneos.

Requisitos:

1. Servidor de Chat
 - a. Debe aceptar conexiones entrantes de múltiples clientes simultáneamente.
 - b. Gestionará el envío y recepción de mensajes entre los clientes conectados.
 - c. Distribuirá los mensajes entrantes al destinatario correcto o a todos los usuarios en el caso de mensajes de difusión.
2. Cliente de Chat
 - a. Permitirá a los usuarios conectarse al servidor y enviar/recibir mensajes en tiempo real.
 - b. Tendrá una interfaz para ingresar mensajes y visualizar las conversaciones.
3. Procesamiento en Paralelo
 - a. Utilizar hilos (módulo ``threading``) o procesos (módulo ``multiprocessing``) para manejar cada conexión de cliente de forma independiente.
 - b. Asegurar que el servidor pueda atender nuevas conexiones y mensajes sin bloquearse.
4. Comunicación en Red
 - a. Implementar protocolos de comunicación utilizando sockets
 - b. Utilizar protocolos TCP para conexiones fiables.
5. Sincronización y Seguridad
 - a. Manejar adecuadamente la sincronización entre hilos o procesos para evitar race condition
 - b. Utilizar mecanismos de bloqueo (locks) cuando sea necesario.
6. Escalabilidad y Rendimiento
 - a. Optimizar el uso de recursos del sistema para manejar un gran número de conexiones simultáneas.
 - b. Evaluar el rendimiento y ajustar la arquitectura (hilos vs. procesos) según las necesidades.

Consideraciones Adicionales

- Manejo de Errores: Implementar manejo de excepciones para conexiones interrumpidas y otros errores de E/S.
- Extensibilidad: Diseñar el sistema de manera modular para facilitar futuras ampliaciones, como salas de chat privadas, autenticación de usuarios, o transferencia de archivos.
- Interfaz de Usuario Gráfica