

Universidad del Valle de Guatemala  
18 Avenida 11-95 Guatemala  
Facultad de Ingeniería  
Departamento de Computación



# Investigación para el proyecto chat

## **Integrantes:**

Diego Alexander Hernández Silvestre, 21270

Mario Antonio Guerra Morales, 21008

Linda Ines Jimenez Vides, 21169

## **Curso:**

Sistemas Operativos

## **Sección:**

Sección 20

Guatemala, 28 de mayo de 2024

Github: <https://github.com/Dahernandezsilve/ChatClienteServidor.git>

## **Tecnologías utilizadas y documentación**

### **a. Pthreads**

Dentro de la aplicación de chat se utilizaron en diversas partes pthreads, los cuales se describen a continuación:

- En la parte del cliente se utilizaron principalmente para manejar múltiples clientes en el servidor.
  - Cuando un cliente se registra correctamente se crea un nuevo hilo para manejar las solicitudes de ese cliente individualmente, permitiendo así que el servidor maneje múltiples conexiones simultáneas sin bloquearse
  - Se utiliza mutex para proteger el acceso a la lista de clientes. Esto se usa para evitar race conditions cuando varios hilos intentan modificar la misma estructura de datos como agregar clientes.
  - Al finalizar el servicio de un cliente, cada hilo asociado se termina para liberar los recursos asignados al hilos y que el programa continúe ejecutándose.
- En la parte del servidor se utilizan principalmente:
  - Un hilo se encarga de recibir mensajes del servidor del chat. Este hilo espera en un bucle por eventos de recepción de mensajes y procesa estos mensajes según sea necesario.
  - Se utiliza una cola para almacenar temporalmente las respuestas recibidas del servidor antes de ser procesadas. Esto permite que el hilo dedicado a recibir mensajes y el hilo principal puedan trabajar de manera asíncrona, ya que el hilo principal puede leer de esta cola mientras el hilo dedicado a recibir mensajes, recibe aún más.
  - Se utiliza mutex y condicionales para sincronizar el acceso a la cola de respuesta y garantizar que solo un hilo pueda acceder a ella a la vez. Esto evita race conditions y asegura que los hilos puedan comunicarse segura y eficientemente.
  - Al finalizar se termina la ejecución de ambos hilos antes de cerrar el socket y salir del programa.

### **b. Sockets**

Los sockets se utilizan principalmente para establecer conexiones de red y comunicarse con clientes y servidores, detallados más a continuación:

- Se crea un socket para establecer una conexión de red, el cual es utilizado para escuchar conexiones entrantes de los clientes.

- Después de crear el socket se configura la dirección del servidor como bind, el cual asigna un número de puerto al socket creado anteriormente, permitiendo que el servidor escuche ese puerto.
- Se utiliza listen en el servidor para que este comience a escuchar conexiones entrantes en el socket especificado, permitiendo que el servidor acepte múltiples conexiones simultáneas.
- Cuando un cliente intenta conectarse al servidor, se usa un accept para aceptar la conexión y devuelve un nuevo socket específico para esa conexión. Este nuevo socket se utiliza para comunicarse con el cliente individualmente.
- También se utilizan sockets para enviar y recibir datos entre el servidor y los clientes.
- Finalmente, cuando un cliente se desconecta o se completa su interacción, el socket de esa conexión se cierra.

También del lado del cliente se tuvo estos usos:

- El socket se crea en la función principal, el cual se utiliza para una conexión TCP con el servidor del chat.
- Se establece una conexión al servidor del chat especificado por la IP y el puerto proporcionados como argumentos de línea de comando. También se envían solicitudes al servidor utilizando ese socket, y a la vez recibe respuestas de este.
- Se utiliza también para enviar mensajes o comandos al servidor según la función a utilizar según sea necesario.
- Al final el socket se cierra después de que todas las operaciones necesarias se hayan completado.

## Referencias

The Single UNIX . (1997).  
<https://pubs.opengroup.org/onlinepubs/007908775/xsh/pthread.h.html>

socket(2) - Linux manual page. (2023).  
<https://man7.org/linux/man-pages/man2/socket.2.html>