

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computación
Lenguajes de Programación - IC4700
I Semestre de 2025

Estudiantes:

- Sebastián Calvo Hernández - 2022099320
- Isaac Gamboa Ureña - 2022437592

Profesor:

- Bryan Hernández Sibaja

Sistema de Mensajería en C++ (Programación Imperativa)

Este proyecto implementa un sistema de mensajería bastante simple utilizando el paradigma imperativo en C++, con comunicación basada en sockets TCP entre un servidor y múltiples clientes.

Enlace al repositorio de GitHub: <https://github.com/SebasCH04/cpp-messaging-system/tree/main>

Pasos de instalación del programa

Requisitos:

- Sistema operativo: **Linux**
- Compilador: **g++**
- Terminal con soporte para **colores ANSI** (para distinguir mensajes enviados y recibidos)

Instalación:

1. **Clonar o descargar** el repositorio.
2. Asegúrate de tener instalado **g++**. Si no lo tienes, se instala con:

```
sudo apt update  
sudo apt install g++
```

3. **Compilar el servidor:**

```
g++ servidor/main.cpp -o servidor.out
```

4. **Compilar los clientes**, dependiendo de cuál desees usar:

```
g++ cliente/main.cpp -o cliente.out
g++ cliente2/main.cpp -o cliente2.out
g++ cliente3/main.cpp -o cliente3.out
```

Manual de usuario

Configuración previa:

Cada cliente tiene un archivo `config.txt` dentro de su carpeta correspondiente (`cliente/`, `cliente2/`, etc.) con el siguiente formato:

```
PUERTO=5001
```

Puedes cambiar el número de puerto para que cada cliente tenga un puerto diferente y pueda recibir mensajes correctamente.

Ejecución:

1. Ejecutar el servidor:

```
./servidor.out
```

Esto iniciará el servidor escuchando en el puerto `9000`.

2. Ejecutar los clientes, en **terminales separadas**, según el cliente que quieras usar:

```
./cliente.out
./cliente2.out
./cliente3.out
```

3. Al iniciar un cliente:

- Se te pedirá ingresar tu nombre de usuario.
- El cliente obtiene automáticamente tu IP y el puerto desde `config.txt`.
- Luego, se conecta con el servidor y se registra el usuario.

4. Después del registro:

- Puedes escribir el nombre del destinatario y luego el mensaje que le quieres enviar.
 - Los mensajes **enviados** se mostrarán en color **verde**.
 - Los mensajes **recibidos** se mostrarán en color **azul**.

Arquitectura lógica utilizada

Este sistema implementa una arquitectura **cliente-servidor** basada en TCP/IP.

Servidor:

- Se ejecuta una única vez.
- Escucha conexiones en el puerto **9000**.
- Mantiene una **tabla de usuarios registrados** (nombre, IP, puerto).
- Recibe:
 - Mensajes de **registro**: "**usuario|IP|puerto**"
 - Mensajes para **reenviar**: "**destinatario|remitente: mensaje**"
- El servidor reenvía el mensaje directamente al cliente destinatario usando su IP y puerto.

Clientes:

- Cada cliente:
 - Lee su puerto desde **config.txt**
 - Obtiene su IP automáticamente
 - Se registra al servidor al iniciar
- Luego:
 - Un proceso hijo escucha mensajes entrantes en su puerto (utilizando **fork()**)
 - El proceso principal permite enviar mensajes a otros usuarios

Explicación del funcionamiento:

1. Inicio del servidor:

El servidor inicia y escucha en el puerto 9000 por nuevas conexiones.

2. Registro de cliente:

Cada cliente, al ejecutarse:

- Solicita el nombre de usuario.
- Obtiene la IP automáticamente.
- Lee el puerto desde su archivo **config.txt**.
- Se conecta al servidor y se registra.

3. Comunicación entre clientes:

- El usuario ingresa el nombre del destinatario y el mensaje.
- El cliente formatea el mensaje y lo envía al servidor.
- El servidor busca al destinatario en su tabla y reenvía el mensaje directamente al cliente correcto.
- El cliente receptor muestra el mensaje en su terminal (en azul).

4. Interfaz limpia y colorida:

- Verde: mensaje enviado
- Azul: mensaje recibido