

## Instrucciones de instalación, compilación y ejecución del proyecto de chat en C

Este documento explica cómo instalar las herramientas necesarias, preparar el entorno y compilar/ejecutar el servidor y los clientes del proyecto de chat en C.

### 1. Crear y preparar una máquina virtual

Puede utilizar cualquier sistema operativo basado en Linux, como Ubuntu Server 22.04 LTS. Si utiliza AWS Academy Learner Lab, EC2, VirtualBox o cualquier proveedor de nube, cree una instancia con las siguientes características:

- Sistema operativo: Ubuntu Server 22.04 LTS
- Tipo de máquina: t2.micro (o superior)
- Red: Debe permitir tráfico en el puerto TCP 22 (SSH) y 8080 (para el chat)
- Clave SSH: Asegúrese de tener acceso mediante un archivo .pem o claves SSH.

Recuerde crear una máquina virtual por cada componente del proyecto, es decir necesita una máquina virtual para el servidor, y una máquina virtual distinta para cada cliente.

### 2. Instalar herramientas necesarias

Una vez conectado a la máquina virtual, ejecute los siguientes comandos:

```
sudo apt update
sudo apt upgrade -y
sudo apt install gcc git -y
```

### 3. Crear carpeta de trabajo

Cree una carpeta para organizar los archivos:

```
mkdir proyecto-chat
cd proyecto-chat
```

### 4. Subir o crear los archivos

Puede copiar directamente los archivos desde su máquina local, o crearlos manualmente usando nano.

Para crear el servidor:

```
nano servidor_multicliente.c
(Pegar el código del servidor desde:
https://github.com/AndresSRZ/Proyecto\_Chat\_Telematica/blob/main/servidor.c)
```

Para crear el cliente:

*nano cliente.c*

(Pegar el código del cliente desde:

[https://github.com/AndresSRZ/Proyecto\\_Chat\\_Telematica/blob/main/cliente.c](https://github.com/AndresSRZ/Proyecto_Chat_Telematica/blob/main/cliente.c))

Recuerde modificar la IP en cliente.c:

```
inet_pton(AF_INET, "IP_DEL_SERVIDOR", &serv_addr.sin_addr);
```

Donde **IP\_DEL\_SERVIDOR** es la IP pública de **su** servidor.

5. Compilar el servidor

```
gcc servidor_multicliente.c -o servidor_multicliente -pthread
```

6. Ejecutar el servidor

Para ejecutar el servidor en primer plano:

```
./servidor_multicliente
```

Para ejecutarlo en segundo plano:

```
./servidor_multicliente &
```

7. Compilar el cliente

```
gcc cliente.c -o cliente -pthread
```

8. Ejecutar el cliente

```
./cliente
```

Se solicitará ingresar un nombre de usuario para el chat.

9. Salir del chat

Durante la ejecución, cualquier cliente puede salir del chat escribiendo:

```
exit o quit
```

Esto desconectará automáticamente al cliente del servidor.

10. Verificación de la persistencia de los datos

Ingresa a su máquina virtual donde tiene el servidor, y detenga el servidor con Control + C. Luego utilice el comando ls para ver que se haya creado un archivo .log, donde encontrará toda la actividad realizada en el chat. Compruebe esto con los comandos:

```
nano chat.log
```