

19 de noviembre de 2024



*No. Practica: 04*  
*Nombre de la Practica:*  
*Comunicación cliente - servidor*

Instituto Politécnico Nacional.  
Escuela Superior de Cómputo.  
Licenciatura en ciencia de datos.

Nombre de la materia: Computo de alto Desempeño  
Grupo: 4AV1  
Profesor: Benjamín Cruz Torres

Nombre del equipo: x  
Integrantes del equipo:  
• López Méndez Emiliano

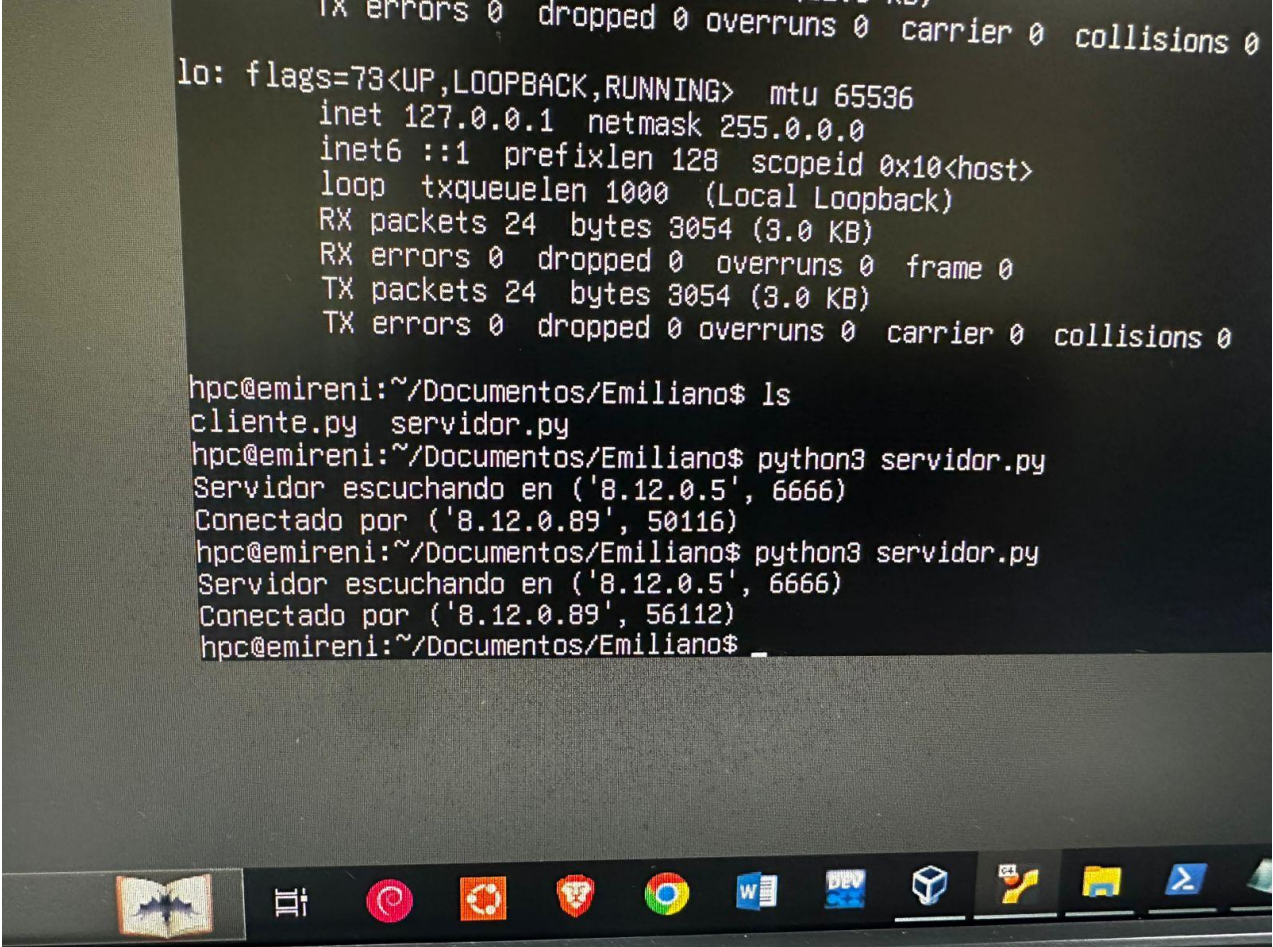
## Desarrollo de la Actividad

En esta práctica, se implementó un sistema cliente-servidor básico en Python utilizando sockets para la comunicación. Inicialmente, se configuraron los archivos `servidor.py` y `cliente.py` con el código base proporcionado. Posteriormente, se modificaron los códigos para permitir que el cliente y el servidor se ejecutaran en diferentes máquinas. Esto implicó cambiar las direcciones IP en los archivos y probar la comunicación remota, validando el envío y recepción de mensajes entre ambas partes. A través de este proceso, se verificó que el servidor pudiera recibir y responder correctamente los mensajes enviados por el cliente, incluso en un entorno distribuido.

Finalmente, se realizaron modificaciones adicionales para permitir que el servidor manejara múltiples clientes simultáneamente mediante hilos, garantizando que cada conexión se gestionara de manera independiente.

```
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 24 bytes 3054 (3.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 24 bytes 3054 (3.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

hpc@emireni:~/Documentos/Emiliano$ ls
cliente.py  servidor.py
hpc@emireni:~/Documentos/Emiliano$ python3 servidor.py
Servidor escuchando en ('8.12.0.5', 6666)
Conectado por ('8.12.0.89', 50116)
hpc@emireni:~/Documentos/Emiliano$ python3 servidor.py
Servidor escuchando en ('8.12.0.5', 6666)
Conectado por ('8.12.0.89', 56112)
hpc@emireni:~/Documentos/Emiliano$ _
```



Comunicación al servidor



```
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 8.12.0.5 netmask 255.255.255.0 broadcast 8.12.0.255
    inet6 fe80::a00:27ff:fef8:255d prefixlen 64 scopeid 0x20
    ether 08:00:27:f8:25:5d txqueuelen 1000 (Ethernet)
    RX packets 5590 bytes 455131 (455.1 KB)
    RX errors 0 dropped 706 overruns 0 frame 0
    TX packets 153 bytes 22356 (22.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 24 bytes 3054 (3.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 24 bytes 3054 (3.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

hpc@emireni:~/Documentos/Emiliano$ ls
cliente.py servidor.py
hpc@emireni:~/Documentos/Emiliano$ python3 servidor.py
Servidor escuchando en ('8.12.0.5', 6666)
Conectado por ('8.12.0.89', 50116)
hpc@emireni:~/Documentos/Emiliano$ python3 servidor.py
Servidor escuchando en ('8.12.0.5', 6666)
Conectado por ('8.12.0.89', 56112)
hpc@emireni:~/Documentos/Emiliano$ python3 servidor.py
Servidor escuchando en ('8.12.0.5', 6666)
Conectado por ('8.12.0.89', 53104)
hpc@emireni:~/Documentos/Emiliano$ python3 servidor.py
Servidor escuchando en ('8.12.0.5', 6666)
Conectado por ('8.12.0.89', 42836)
Mensaje recibido del cliente: Hola mi amor 2.0
```

Yo siendo el servidor y clientes mandando mensajes, ya no pudimos hacer mas clientes por falta de tiempo, y porque ha muchos no les funcionaba sus códigos, pero comprobamos que si se podía eventualmente ya la final de la clase.

## Conclusiones

En esta práctica se aprendió cómo funciona la comunicación entre un cliente y un servidor utilizando sockets en Python. Al principio, fue sencillo entender cómo enviar y recibir mensajes entre ambos, pero al modificar los códigos para que funcionaran en diferentes máquinas, me di cuenta de la importancia de configurar bien las direcciones IP y el puerto. Ya que, si un número está mal o un punto, no se conectará para nada al servidor.