



**Objetivo**

Implementar y probar un servidor TCP que atiende a un cliente a la vez y recibe mensajes de **5 bytes**, finalizando la sesión cuando llega la palabra **“FINAL”**. Desarrollar un cliente que envía **cinco** bloques “ABCDE” (5 bytes) y después “FINAL”.

**Entorno**

* VM Linux de prácticas.
* Archivos: tcp\_servidor1\_simple.py (servidor) y tcp\_cliente1\_simple.py (cliente).
* Pruebas locales con **localhost**.

**Procedimiento realizado**

1. Creación de los dos scripts en el PC (VS Code) y transferencia a la VM (WinSCP).
2. Ejecución del **servidor** en la VM, escuchando en un puerto TCP (por defecto 9999).
3. Ejecución del **cliente** en otra terminal de la VM, conectando a localhost y al mismo puerto.
4. El cliente envía **cinco** mensajes de 5 bytes (“ABCDE”) y, a continuación, **“FINAL”** para cerrar de forma ordenada.
5. El servidor, tras recibir “FINAL”, cierra la conexión con ese cliente y **vuelve a quedar a la espera** de nuevas conexiones.

**Resultados observados**

* En la consola del servidor aparece **“Conectado desde (127.0.0.1, puerto\_efímero)”**.
* Se muestran **cinco** líneas con **“Recibido bloque: ABCDE”**.
* A continuación se muestra **“Recibido FINAL → cierro sesión con cliente”**.
* El servidor vuelve a mostrar **“Esperando un cliente…”**, quedando listo para otra conexión.
* Al relanzar el cliente, el patrón se repite con el mismo comportamiento.

**Evidencias (capturas propuestas)**

* Servidor en estado **“Escuchando / Esperando un cliente”**.
* Secuencia de recepción en el servidor: **5× “Recibido bloque: ABCDE”** y **“Recibido FINAL → cierro sesión con cliente”**.
* Mensaje final de **“Esperando un cliente…”** tras cerrar la sesión.

**Conclusión**

Queda verificado el esquema **“tamaño prefijado”** en TCP: el servidor procesa bloques fijos de **5 bytes** y usa **“FINAL”** como señal de terminación, manteniéndose operativo para aceptar nuevos clientes después de cada sesión.