

# ARQUITECTURA CLIENTE –SERVIDOR

## PROYECTO FINAL

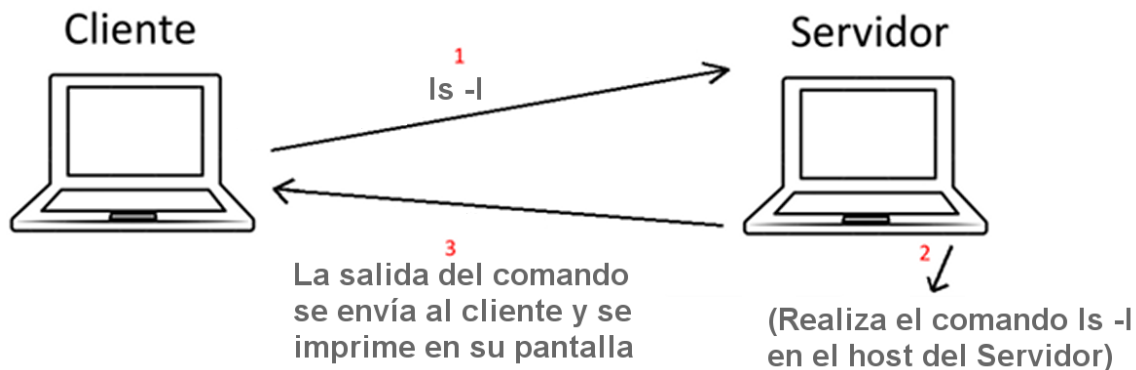
### “CLIENTE-SERVIDOR SSH”

El proyecto consiste en crear un Cliente-Servidor que ejecute comandos remotamente, como ocurre con un Cliente-Servidor SSH comercial o gratuito. El proyecto debe cumplir con los siguientes requerimientos:

- Estar desarrollado en una arquitectura cliente-servidor, con sockets TCP/IP y con conexión remota. Aunque los códigos los desarrolles en una misma máquina y lo pruebes localmente, pero la revisión será remota de host a host.
- Debe estar desarrollado en un ambiente Linux, en la distribución que gustes o en MacOS y programado en lenguaje C.
- No se aceptan proyectos programados en otros lenguajes.

#### EXPLICACIÓN DEL PROYECTO

- La comunicación entre el Cliente y el Servidor debe ser remota vía sockets TCP (Sockets Internet)
- El programa Servidor debe iniciarse en el host servidor, en el puerto que decidas, pero debe pasarse desde la línea de comandos.
- El programa Cliente debe iniciarse en el host cliente (pasando el dominio o IP del servidor y el puerto, ambos desde línea de comandos).
- El Servidor acepta la conexión.
- Una vez aceptada la conexión, el Cliente escribe el comando y lo envía por el socket al Servidor (Paso # 1 de la imagen). En este ejemplo se muestra el comando `ls -l` pero puede ser cualquier comando de Linux o MacOS.
- El Servidor recibe el comando y lo ejecuta en sistema local (Paso # 2 de la imagen). Puedes usar las siguientes funciones: `fork()` y la familia de funciones `exec()`.
- El Servidor debe devolver la salida al cliente (Paso # 3 de la imagen).



En la imagen anterior, el cliente ejecuta el comando `ls -l` pero no es una ejecución local en el cliente. El comando se aplica en el servidor y le envía el resultado al cliente.

Por último, con el comando:

salir (o exit) ↵

El cliente debe desconectarse del servidor.

#### MATERIAL DE REFERENCIA:

- Como tienen que ejecutar el comando por medio de las funciones `fork` y `exec()`, para recuperar la salida del comando **les ayudarán las funciones `pipe()` y `dup2()`**
- Este video de YouTube es muy bueno explicando la función `pipe()`  
<https://youtu.be/oxWxcYoJdM> **está en español**
- El uso de `dup2()`  
<https://baulderasec.wordpress.com/programando-2/programacion-con-linux/3-trabajando-con-los-archivos/acceso-de-bajo-nivel-a-archivos/dup-y-dup2/#:~:text=La%20llamada%20al%20sistema%20dup2,descriptor%20usar%20para%20la%20copia.&text=Esta%20funci%C3%B3n%20devuelve%20%2D1%20en,un%20duplicado%20del%20descriptor%20fildes%20.>
- Pueden apoyarse en cualquier libro o sitio que trate los sockets TCP/IP para C.
- Pueden apoyarse en el sitio de profesores  
<http://profesores.fi-b.unam.mx/carlos/acs/Tema-04-Sockets-INTERNET-C/>
- La documentación de las funciones de C para Linux, se encuentra en:  
<https://pubs.opengroup.org/onlinepubs/009696799/idx/index.html>

#### CONSIDERACIONES:

- El proyecto **es individual o en equipos de 2 integrantes máximo.**
- El proyecto debe funcionar para diferentes comandos con sus opciones, como `ls -l`, `ps -e -o columnas`, `pwd`, `date`, `cat archivo`, `whoami`, etc. Por facilidad, no aceptará comandos de salida dinámica como el comando `top`, por ejemplo.
- No limiten su servidor a que solamente pueda ejecutar los comandos que mencioné arriba. Puedo probar esos comandos con sus diferentes opciones.
- Proyecto en ambiente Linux o MacOS

- Programado en lenguaje C (**No se aceptan proyectos en otro lenguaje**).
- Sí está permitido el uso del IDE de tu preferencia.
- Código comentado.
- Entregar PDF con códigos fuente dentro del PDF, así como capturas de pantalla de, al menos, 2 pruebas.
- Además del PDF, entregar los archivos \*.c de los códigos fuente al correo [carlos@fi-b.unam.mx](mailto:carlos@fi-b.unam.mx). **Favor de no enviar ejecutables porque, por seguridad, el correo es bloqueado y no me llegará.**
- Enviar mensaje vía App de mensajería instantánea al 5517985738 para concertar una reunión de zoom de manera individual a cada alumno o equipo. Ahí revisaré el proyecto en ejecución **Para la revisión en Zoom deben abrir su cámara, por favor.**
- La revisión consiste en 2 pruebas: una prueba local donde Cliente y Servidor estén en tu host local (debes compartir pantalla en la sesión de Zoom) y otra prueba donde yo tomaré tus códigos fuentes que me enviaste por correo y los colocaré en 2 hosts para realizar la prueba remota (yo compartiré pantalla para que veas la prueba remota).
- **La entrega queda abierta ya sea un día entre semana, sábado o domingo. La fecha límite de entrega es el jueves 6 de junio de 2024 y el viernes 7 de junio subo calificaciones al sistema.**