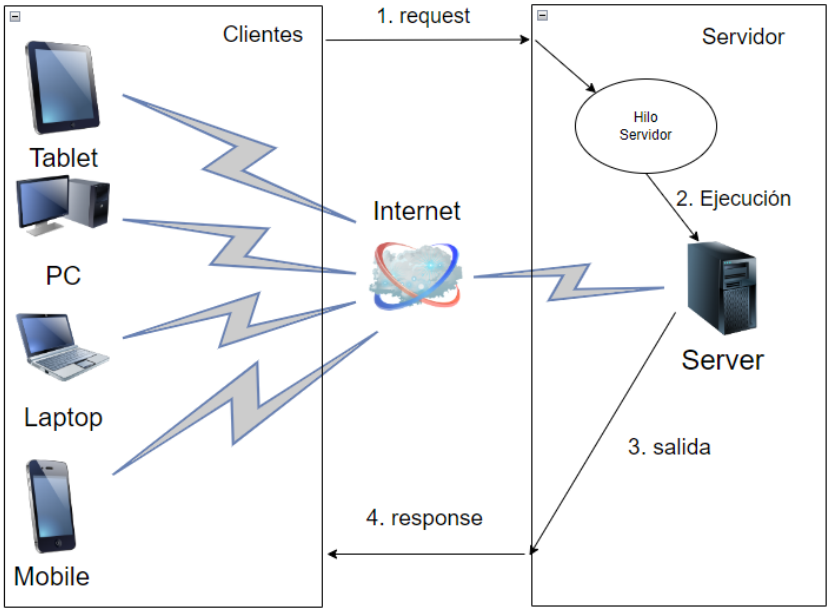
1. DESCRIPCIÓN DE LA APLICACIÓN

Se desea crear la aplicación ***REMOTE COMMANDER*** para la ejecución de comandos en remoto. La aplicación se programará en JAVA para poder ejecutarse desde cualquier tipo de entorno y tendrá una parte servidora multihilo llamada Remote Commander Server (RCS) que permitirá comunicaciones con sockets comunes y a través de sockets SSL.

La aplicación cliente se llamará Remote Commander Client (RCC) y permitirá administrar de forma remota el servidor que indiquemos como parámetro. El esquema de funcionamiento es muy sencillo. Los clientes se conectan al servidor utilizando un modo de conexión (normal o cifrado) y envían peticiones de comandos al servidor. El servidor, ejecuta esos comandos y devuelve la salida del comando ejecutado en el servidor:



REQUEST: Un RCC se conecta al RCS y, a través de un socket cliente, envía al servidor un comando de los siguientes:

LIST directorio_remoto
Lista el contenido de un directorio

SEND fichero_origen directorio_remoto
Envía un fichero del cliente al servidor

RECEIVE ruta_fichero_remoto fichero_origen
Envía un fichero del servidor al cliente

EXEC comando argumentos
El servidor ejecuta un comando del sistema operativo en el servidor y retorna al cliente la salida de dicho comando

RESPONSE: El RCS recibe el comando y lo ejecuta, devolviendo la respuesta al RCC:

El servidor, dependiendo del comando recibido, ejecutará las acciones necesarias para construir la respuesta al cliente.

Por ejemplo, si el comando recibido es:

LIST /home/ubuntu/misFicheros

El servidor comprobará si es válida la ruta y si lo es, recorrerá la carpeta devolviendo al cliente la lista de elementos que hay dentro de la carpeta.

Si el comando recibido por el servidor fue:

SEND imagen.png /home/ubuntu/misFicheros

El servidor comprobará que la ruta /home/ubuntu/misFicheros es un directorio válido y, si es valido, procederá a recibir el fichero imagen.png situado en la carpeta local del cliente. Una vez transferido el fichero, el cliente informará al usuario que ha recibido el fichero.

El caso sería análogo para el comando RECEIVE, solo que esta vez, será el servidor el que envíe el fichero al cliente.

Si el comando recibido por el servidor fue:

EXEC ip a

El servidor ejecutará el comando **ip a** y retornará al cliente un listado de sus direcciones ip disponibles (Comando del sistema operativo ip a).

2. PARÁMETROS DE LA LÍNEA DE COMANDOS

Para invocar la ejecución del servidor podrán especificarse los siguientes parámetros:

java rcserver <modo> <puerto> <max_clientes>

<***modo***> Para establecer si la aplicación funciona con sockets comunes o Sockets SSL se utilizán argumentos en la línea de comandos. Por ejemplo, para arrancar el servidor en modo SSL se puede especificar el parámetro modo:

```
java rcserver modo=SSL #arranca en modo SSL
java rcserver #arranca en modo normal
```

Los parámetros que el servidor puede recibir a través de la línea de comandos son:

<puerto>: número de puerto donde escucha el servidor. Ej: `puerto=8025`

```
java rcserver modo=SSL puerto=8025
```

<max_clientes> número máximo número de hilos que el servidor puede crear para clientes. Por ejemplo, para levantar el servidor en modo normal en el puerto 1721 y un máximo de 10 clientes:

```
java rcserver puerto=1721 max_clientes=10
```

Para la invocación del cliente se podrán especificar los siguientes parámetros:

java rclient <modo> <host> <puerto> <carpeta_cliente>

<modo>: Al igual que el servidor establecerá si se lanza un socket cliente común o uno SSL.

<host>: dirección del servidor al que se va a conectar

<puerto>: número de puerto del servidor al que se va a conectar

<carpeta_cliente>: carpeta donde se almacenan los ficheros transferidos desde el servidor y desde dónde enviará ficheros al servidor.

Por ejemplo, para conectar el cliente al puerto 1721 del servidor 192.168.1.33 en modo SSL en la carpeta c:\flt:

```
java rclient modo=SSL host=192.168.1.33 puerto=1721 carpeta_cliente=c:\lft
```

Cada grupo puede añadir los parámetros que considere necesarios tanto en el cliente como en el servidor, siempre y cuando estén justificados para el funcionamiento correcto de la aplicación.

3. DESARROLLO Y DESPLIEGUE

El trabajo se realizará en equipos de 4 alumnos. Cualquier equipo con un número diferente de miembros, deberá ser autorizado previamente. Cada equipo diseñará y establecerá el protocolo de capa de aplicación por el cual cliente y servidor efectúan el intercambio de peticiones y respuestas. *Cada equipo deberá elegir un nombre para su grupo y publicar los componentes de este en el foro de la asignatura del campus virtual.*

El sistema se implantará en la cuenta del laboratorio de redes 2 de la plataforma de amazon web services (aws), utilizando las direcciones IP privadas de, al menos, las siguientes instancias dentro de una VPC:

- Una máquina windows como cliente
- Una máquina ubuntu como cliente
- Una máquina ubuntu como servidor

Además, se utilizará un cliente externo que se conecte a la IP pública del servidor.

Los alumnos indicarán en la memoria del trabajo las instancias que se han utilizado para la implantación del sistema.

4. REQUISITOS

a) Tanto servidor como cliente reportarán errores a un fichero de log "errores.log" que se almacenará en el directorio donde se ubiquen los binarios. En este log se almacenarán tanto los errores por falta de permisos a la hora de acceder a ficheros, como errores de sintaxis a la hora de aceptar comandos (por ejemplo, poner GET sin nombre de fichero) y errores a la hora de gestionar las conexiones.

b) Cada comando procesado tanto por el cliente como por el servidor se reportarán a un fichero llamado acciones.log

c) Se deberán tratar las excepciones necesarias para que tanto el servidor como el cliente, se mantengan activos sin cerrarse abruptamente.

d) Se deberán incluir capturas de wireshark de la interacción entre cliente y servidor, incluyendo capturas descifradas para el modo SSL.

e) El trabajo incluirá el diseño de las las pruebas unitarias: se incluirán capturas de pantalla de las pruebas realizadas para probar el funcionamiento de la aplicación.

5. PRESENTACIÓN ORAL

Los alumnos prepararán un vídeo con una demostración del funcionamiento de la aplicación sobre la plataforma AWS tal y como se especifica en el punto 3. La defensa del trabajo tendrá una duración de entre 4 y 6 minutos por miembro del equipo [TOTAL 20 minutos] e incluirá explicaciones sobre las decisiones de diseño de la aplicación y su arquitectura. Deberán participar en el video todos los componentes del grupo.

6. ENTREGA

Se entregará en una carpeta comprimida en zip “Nombre_Equipo.zip” que contenga:

- Código fuente de la aplicación.
- Una memoria en pdf que contendrá explicaciones relativas al diseño de la aplicación y desarrollo, el diseño del protocolo de capa de aplicación y capturas de pantalla del funcionamiento de la aplicación con las pruebas unitarias diseñadas. En esta memoria se incluirá el nombre de los componentes del grupo y el usuario de aws sobre el que se ha desplegado la aplicación.
- Un documento de texto con un enlace al vídeo de la presentación.
- Las capturas de wireshark

7. CRITERIOS DE CALIFICACIÓN

Cada miembro del grupo tendrá dos calificaciones:

- La defensa oral: Cada miembro puede tener una calificación distinta
- Desarrollo del trabajo: Todos los miembros comparten nota

Criterios de calificación del desarrollo del trabajo:

Puntos	Criterio
0,5 punto	El servidor y el cliente leen correctamente los parámetros de la línea de comandos
1 punto	La aplicación implementa correctamente el comando LIST.
3 puntos	La aplicación implementa correctamente los comandos SEND y RECEIVE (se envían tanto ficheros de texto como binarios).
2 puntos	La aplicación implementa correctamente el comando EXEC.
0,5 puntos	La aplicación reporta al log de errores los eventos sucedidos en la comunicación y a log de acciones los comandos procesados por cliente y servidor.
0,5 puntos	Se tratan las excepciones de forma coherente
2 puntos	Los comandos se implementan correctamente usando el modo SSL
0,5 puntos	Se incluyen capturas de wireshark para ilustrar el funcionamiento de cliente y servidor

Criterios de calificación de la defensa oral:

Puntos	Criterio
2 puntos (equipo)	Organización y estructura de la presentación. Se ha seguido una estructura clara y coherente, incluyendo una introducción, un desarrollo y una conclusión.
2 puntos (equipo)	Creatividad y originalidad de la presentación. Se ha utilizado algún recurso creativo para mantener la atención de la exposición
2 puntos (individual)	Claridad de la comunicación oral. El alumno ha hablado con claridad, usando un tono de voz adecuado, pronunciando correctamente las palabras y usando un vocabulario técnico apropiado.
2 puntos (individual)	Habilidades de presentación: tales como la postura, la gestión del tiempo y el uso de apoyos visuales.
2 puntos (individual)	Claridad y la calidad del contenido presentado. El alumno ha presentado información relevante y suficiente para su parte.