

## Preguntas

### 1. ¿Que es un puerto?

Un puerto es un identificador de virtualización que define un punto final de servicio o identificador de sesión.

Se pueden explicar como compuertas de datos que permiten al software de aplicación dentro de la red usar recursos sin interferencia. Estos puertos se usan por protocolos y se suelen asociar a una dirección IP para establecer comunicaciones

### 2. ¿Cómo están formados los endpoints?

Un endpoint está compuesto por una dirección de IP y un puerto, que se utilizan para ingresar dentro de una red de comunicación.

### 3. ¿Que es un socket?

Un socket es una interfaz input/output de datos que permite que se comuniquen los procesos entre si, mediante uno o distintos sistemas unidos mediante una red. A su vez, los sockets sirven de metodo de comunicacion entre programas cliente/servidor en una red, siendo el punto final en una conexion.

### 4. ¿A qué capa del modelo TPC/IP pertenecen los sockets? ¿Porque?

A la capa de transporte ya que no dependen de una red fisica que se este utilizando para transportar datos. Esto es asi ya que TCP/IP es orientado a la conexion y trabaja con puertos logicos y junto con la capa de red forma los sockets.

### 5. ¿Cómo funciona el modelo cliente-servidor con TCP/IP Sockets?

basicamente al usar sockets serian como un puente que comunica a los clientes con el servidor trasladando los paquetes de informacion en ambas direcciones. Usan flujos de entrada y salidas y hilos o observadores que estan pendientes de los cambios de informacion o de acciones en el puerto (como recibir un paquete con data).

### 6. ¿Cuales son las causas comunes por la que la conexión entre cliente/servidor falle?

- Puerto del Servidor Bloqueado
- Intercambio de datos con clientes Offline (por parte del servidor)
- Endpoint en uso
- Direcciones de host o puertos invalidos para conectarse
- Servidor deshabilitado

### 7. Diferencias entre sockets UDP y TCP

Resumen Diferencias:

UDP es mas rapido y eficiente, se suele usar para archivos de audio y video. TCP es mas rustico pero es mas fiable, prioriza garantizar la entrada de paquetes en el mismo orden.

Explicacion de diferencias:

- TCP está orientado a la conexión, mientras que UDP es un protocolo sin conexión.
- Tanto TCP como UDP pueden comprobar si hay errores, pero sólo TCP puede corregir el error ya que tiene control de congestión y de flujo.
- TCP es altamente confiable para transferir datos útiles ya que toma el acuse de recibo de la información enviada. Y vuelve a enviar los paquetes perdidos si los hay. Mientras que en el caso de UDP, si el paquete se pierde, el paquete llega como corrupto. Por lo tanto, UDP es un protocolo poco fiable.
- TCP es más lento en comparación con UDP, ya que TCP establece la conexión antes de transmitir los datos y garantiza la entrega adecuada de los paquetes. Por otro lado, UDP no reconoce si los datos transmitidos son recibidos o no.
- El tamaño de head de UDP es de 8 bytes, y el de TCP es más de 20 bytes.

#### 8. Diferencia entre sync & async sockets?

La misma diferencia que metodos sync y async, dicho simple seria linealidad versus paralelismo, especifico del socket se habla de comunicaciones bloqueantes por parte de la espera de respuesta (SINCRONICO) o comunicaciones sin blockers, con un orden desestructurado para emision/recepcion de datos.