



EXTRAORDINARIA M9

Sólo debe haber una correcta, circúlala. Si te equivocas táchala y circula la correcta. Cada pregunta mal o no respondida resta 1/3. (7 puntos)

- | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| a | a | a | a | a | a | a | a | a | a | a | a | a |
| b | b | b | b | b | b | b | b | b | b | b | b | b |
| c | c | c | c | c | c | c | c | c | c | c | c | c |
| d | d | d | d | d | d | d | d | d | d | d | d | d |
- 1) Los sockets de java funcionan sobre protocolo:
a) TCP/IP
b) UDP/IP
c) Por TCP/IP o UDP/IP.
d) No usan el protocolo IP.
 - 2) Las clases datagrampacket y datagramsocket se usan en conexión por socket:
a) Indistintamente ambos TCP/IP o UDP/IP.
b) Sólo TCP/IP.
c) datagrampacket ambos y datagramsocket UDP
d) Ninguna es correcta.
 - 3) Las clases DataInputStream y DataOutputStream se usan en conexión por socket:
a) Indistintamente TCP/IP o UDP/IP.
b) Principalmente TCP/IP.
c) Principalmente UDP/IP.
d) Ninguna es correcta.
 - 4) En Java para que el cliente invoque al método remoto llamado "remoto" en el server, se declara en el RMI y:
a) Se llama con "RPC.remoto"
b) Se llama con "Server.remoto"
c) Se llama con "@RPC(remoto)"
d) Se llama con "remoto"
 - 5) En un servicio sin estado:
a) El servidor atiende peticiones sin aceptar o devolver parámetros.
b) El servidor no almacena ningún dato de cada petición.
c) El servidor atiende cada petición de manera independiente.
d) El servidor atiende peticiones sin establecer conexión directa con el cliente.
 - 6) ¿Qué ocurre si el cliente no conoce el puerto del servicio al solicitar la conexión al servidor?
a) Sólo podemos establecer una conexión sin estado.
b) Podemos establecer una conexión sin puerto destino.
c) Sólo podemos establecer una conexión por socket.
d) No podemos establecer una conexión.

- 7) En la invocación de métodos remotos, respecto a los sockets:
a) Son transparentes al programador.
b) Debo programar la conexión por sockets y TCP/IP.
c) Debo programar la conexión por sockets normalmente.
d) La invocación de métodos remotos no usa sockets.
- 8) ¿Cuántos puertos utiliza el servidor en una conexión por socket con un cliente?
a) Ninguno.
b) Uno.
c) Dos, uno para el servicio y otro para la transmisión.
d) Ninguna es correcta.
- 9) Si el PC A utiliza una invocación de métodos remotos en el PC B, la ejecución de ese método:
a) Se realiza en A.
b) Se realiza en B.
c) Se realiza según el formato de llamada.
d) Se ejecuta en la red.
- 10) Ventajas de utilizar un fichero para compartir información:
a) Más fácil de programar.
b) No estamos tan sujetos a errores de red.
c) Más fácil depurar errores.
d) Todas son correctas.
- 11) RMI permite a un objeto invocar métodos en un objeto que se ejecuta en otro JVM:
a) Sólo desde el mismo PC.
b) Sólo desde la misma red local.
c) Si, si hay conexión de red.
d) Ninguna es correcta.
- 12) Ventajas de usar comunicación por sockets frente a RMI es:
a) Permite el paso de parámetros.
b) El protocolo de comunicación es transparente al programador.
c) Es más fácil de programar.
d) Ninguna es correcta.
- 13) RMI usa por defecto el puerto:
a) 1099.
b) No hay puerto por defecto, debe indicarse.
c) Depende de si es cliente o servidor.
d) RMI no usa puerto.
- 14) Indica los pasos o hacer un esquema para que un cliente pueda invocar un método remoto en un servidor. (1 punto)

El servidor al arrancar registra, crea y publica una descripción de los métodos públicos para que estén a disposición de los clientes. El cliente localiza estos objetos remotos gracias a la descripción, entonces al ejecutar el código, invoca



los métodos remotos en el servidor, entonces el servidor ejecuta los métodos y devuelve al cliente el resultado de dichas operaciones, y el cliente por su parte acaba de ejecutar el código con dichos resultados.

15) Hacer un esquema, de los pasos de una comunicación por sockets.(1 punto)

El servidor abre un socket y empieza a escuchar en este -> El cliente abre un socket propio, y se conecta al socket en el servidor -> El servidor acepta la conexión, y empieza la transmisión de datos entre cliente y servidor (el cliente escribe datos en el servidor, este los lee y viceversa) -> Una vez finalizadas todas las operaciones, ambos cierran la conexión.

16) Explica la utilidad de los objetos “stub” en el cliente en los RMI. (1 punto)

Los objetos stub tienen la utilidad de permitir testear el cliente durante la fase de desarrollo, de modo que podamos validarlo antes de hacer el desarrollo en la parte de servidor.