

CUESTIONARIO DE SOCKETS

1. ¿Qué es un socket en Java?

Un socket es un punto final de una conexión bidireccional entre dos programas que se comunican a través de una red. En Java, permite enviar y recibir datos entre cliente y servidor.

2. ¿Qué clases principales se utilizan para manejar sockets en Java?
Las clases principales son:

- Socket (para el cliente)
- ServerSocket (para el servidor)

3. ¿Qué paquete de Java contiene las clases para trabajar con sockets?
El paquete java.net.

4. ¿Qué es un puerto en el contexto de los sockets?
Un puerto es un número que identifica de forma única un proceso o servicio específico dentro de un host, permitiendo que varios servicios puedan ejecutarse simultáneamente.

5. ¿Cómo se crea un socket cliente en Java?
Usando la clase Socket, por ejemplo:

```
Socket socket = new Socket("localhost", 8080);
```

6. ¿Cómo se crea un socket servidor en Java?
Usando la clase ServerSocket, por ejemplo:

```
ServerSocket server = new ServerSocket(8080);
```

```
Socket client = server.accept();
```

7. ¿Qué método se utiliza para aceptar conexiones en el servidor?
El método accept() de la clase ServerSocket.

8. ¿Qué métodos se utilizan para enviar y recibir datos por un socket?

- Para enviar datos: `getOutputStream()`
- Para recibir datos: `getInputStream()`

9. ¿Qué sucede si el servidor no está disponible cuando el cliente intenta conectarse?

Se lanza una excepción `IOException` indicando que la conexión no pudo establecerse.

10. ¿Qué es el modelo cliente-servidor en el uso de sockets?
Es un modelo donde el servidor espera conexiones y atiende solicitudes, mientras que el cliente inicia la comunicación para enviar o recibir información.

11. ¿Qué tipo de comunicación ofrecen los sockets en Java por defecto?
Comunicación orientada a conexión, es decir, basada en TCP (Transmission Control Protocol).

12. ¿Cómo se puede cerrar un socket correctamente?
Llamando al método `close()` del socket y de sus flujos de entrada/salida, por ejemplo:

```
socket.close();
```

13. ¿Cuál es la diferencia entre TCP y UDP en el contexto de sockets?

- **TCP:** Conexión confiable y orientada a flujo (usa `Socket` y `ServerSocket`).
- **UDP:** Conexión no orientada, más rápida pero sin control de errores (usa `DatagramSocket` y `DatagramPacket`).

14. ¿Qué es un `DatagramSocket`?
Es una clase usada para implementar comunicación basada en UDP, donde los datos se envían en paquetes independientes llamados datagramas.

15. ¿Qué ocurre si no se cierra un socket después de usarlo?
Puede provocar fugas de recursos, bloqueos de puerto y un consumo innecesario de memoria o conexiones en el sistema.