

¿Qué son los sockets en Java?

En Java, los sockets son una forma de permitir la comunicación entre procesos que se ejecutan en diferentes máquinas a través de la red. Utilizamos las clases `Socket` y `ServerSocket` del paquete `java.net` para implementar esta comunicación.

Creación de un servidor utilizando sockets en Java

Ejemplo básico de cómo crear un servidor utilizando sockets en Java:

```
java
Copiar código
import java.io.*;
import java.net.*;

public class ServidorSocket {
    public static void main(String[] args) {
        try {
            ServerSocket serverSocket = new ServerSocket(1234); // Puerto de
escucha
            System.out.println("Servidor esperando conexiones...");

            Socket socket = serverSocket.accept(); // Espera a que un cliente
se conecte
            System.out.println("Cliente conectado desde: " +
socket.getInetAddress().getHostName());

            // Flujo de entrada del cliente
            BufferedReader entrada = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
            String mensaje = entrada.readLine();
            System.out.println("Mensaje recibido del cliente: " + mensaje);

            // Cerrar flujos y sockets
            entrada.close();
            socket.close();
            serverSocket.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Creación de un cliente utilizando sockets en Java

Ejemplo básico de cómo crear un cliente utilizando sockets en Java:

```
java
Copiar código
import java.io.*;
import java.net.*;

public class ClienteSocket {
```

```

    public static void main(String[] args) {
        try {
            Socket socket = new Socket("localhost", 1234); // Conexión al
servidor
            System.out.println("Conectado al servidor...");

            // Flujo de salida al servidor
            PrintWriter salida = new PrintWriter(socket.getOutputStream(),
true);
            salida.println("Hola, servidor!");

            // Cerrar flujos y socket
            salida.close();
            socket.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

Funcionamiento

1. El servidor espera a que un cliente se conecte en un puerto específico utilizando `ServerSocket`.
2. Cuando un cliente se conecta, se crea un nuevo `Socket` para la comunicación con ese cliente.
3. El servidor puede leer y escribir datos a través de los flujos de entrada y salida del socket.
4. El cliente se conecta al servidor utilizando la dirección IP del servidor y el puerto específico.
5. Una vez conectado, el cliente puede enviar y recibir datos utilizando flujos de entrada y salida del socket.

Consideraciones adicionales

- Los sockets en Java pueden trabajar tanto con TCP (Transmission Control Protocol) como con UDP (User Datagram Protocol).
- Es importante manejar adecuadamente las excepciones que pueden ocurrir durante la comunicación, como la pérdida de conexión o los errores de entrada/salida.