

Actividad 3.7 PSP

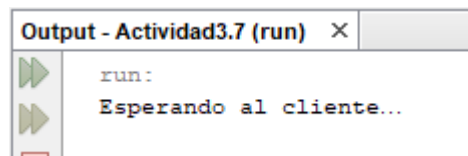
- Código Cliente:

```
public static void main(String[] args) {
    String Host = "localhost";
    int Puerto = 6000; //puerto remoto
    System.out.println("PROGRAMA CLIENTE INICIADO");
    try {
        //Creo el socket
        Socket cliente = new Socket(Host, Puerto);
        Scanner sc = new Scanner(System.in);
        int numero1;
        //Pido numero
        System.out.println("Introduzca un numero: ");
        numero1 = sc.nextInt();
        //Flujo de Salida para objetos
        Numeros num = new Numeros(numero1);
        ObjectOutputStream flSal = new
ObjectOutputStream(cliente.getOutputStream());
        flSal.writeObject(num);
        //Flujo de entrada para objetos
        ObjectInputStream flEnt = new ObjectInputStream(cliente.getInputStream());
        Numeros numRec = (Numeros) flEnt.readObject();
        System.out.println("Recibido: " + numRec.getNumero() + ": el cuadrado es -> " +
numRec.getCuadrado() + " y el cubo es -> " + numRec.getCubo());
        //CERRAR STREAMS Y SOCKETS
        flSal.close();
        flEnt.close();
        cliente.close();
    } catch (IOException e) {
        System.out.println("El servidor no está conectado");
        e.printStackTrace();
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
}
```

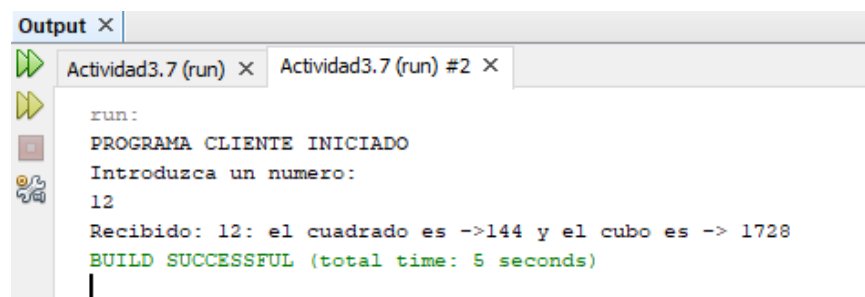
- Código Servidor:

```
public static void main(String[] args) throws IOException, ClassNotFoundException {
    int numeroPuerto = 6000; // Puerto
    // Creo el servidor
    ServerSocket servidor = new ServerSocket(numeroPuerto);
    System.out.println("Esperando al cliente...");
    // Creo el socket cliente
    Socket cliente = servidor.accept();
    // Se prepara un flujo de entrada para objetos
    ObjectInputStream flEntrada = new ObjectInputStream(cliente.getInputStream());
    Numeros numRec = (Numeros) flEntrada.readObject();
    // Elevo al cuadrado lo introducido por el cliente
    numRec.setCuadrado((long) Math.pow(numRec.getNumero(), 2));
    numRec.setCubo((long) Math.pow(numRec.getNumero(), 3));
    // Y se lo envío al cliente
    ObjectOutputStream flSalida = new
    ObjectOutputStream(cliente.getOutputStream());
    flSalida.writeObject(numRec);
    // CERRAR STREAMS Y SOCKETS
    flEntrada.close();
    flSalida.close();
    cliente.close();
    servidor.close();
}
```

- Ejecuto 1º el servidor:



- Ejecuto luego el cliente:



- Y así se finaliza el servidor:

