

apunres-resumen

SERVIDOR

```
public class Servidor {

    public static void main(String[] args) throws IOException {

        ServerSocket socketEscucha = null;

        try{

            socketEscucha = new ServerSocket(9876);
            System.out.println("Arrancando el servidor");

            Numero numero = new Numero();

            while (true){
                try {
                    Socket conexion = socketEscucha.accept();

                    Peticion hilo = new Peticion(conexion, numero);

                    hilo.start();
                } catch (IOException e){
                    e.printStackTrace();
                    throw e;
                }
            }
        } catch (IOException e){
            e.printStackTrace();
            throw e;
        } finally {
            try{
                if (socketEscucha != null){
                    socketEscucha.close();
                }
            } catch (IOException e){
                e.printStackTrace();
            }
        }
    }
}
```

PETICION

```
public class Peticion extends Thread{

    private Socket socket;
```

```

//Clase serializable para el paso de objetos
private Numero numero;

public Peticion(Socket socket, Numero numero){
    this.socket = socket;
    this.numero = numero;
}

@Override
public void run(){

    try {
        escuchar();
    } catch (IOException e) {
        throw new RuntimeException(e);
    } catch (ClassNotFoundException e) {
        throw new RuntimeException(e);
    }
}

public void escuchar() throws IOException, ClassNotFoundException {

    ObjectInputStream ois = null;
    ObjectOutputStream oos = null;

    System.out.println("Conexion con el cliente exitosa");

    //Para leer la peticion del cliente
    ois = new ObjectInputStream(socket.getInputStream());

    //Para enviar datos al cliente
    oos = new ObjectOutputStream(socket.getOutputStream());

    boolean salir = false;
    while (!salir){

        //Escucho la peticion del cliente
        String peticionCliente = (String) ois.readObject();

        int numeroCliente = Integer.parseInt(peticionCliente);
        String respuesta = numero.adevinaNumero(numeroCliente);

        oos.writeObject(respuesta);
        oos.writeObject("fin_peticion");
        oos.flush();

    }

    //Cerramos
    oos.close();
    ois.close();
    socket.close();
}
}

```

CLIENTE

```
public class Cliente {

    Socket socket = null;

    //Para leer los datos del servidor
    ObjectInputStream ois = null;

    //Para enviar peticiones al servidor
    ObjectOutputStream oos = null;

    Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) throws IOException, ClassNotFoundException {

        Cliente c = new Cliente();
        c.lanzarCliente();

    }

    public void lanzarCliente() throws IOException, ClassNotFoundException {

        InetAddress direccion = new InetAddress("localhost", 9876);

        socket = new Socket();

        socket.connect(direccion);

        System.out.println("Cliente --> Servidor conectado correctamente");

        //Para hablar al servidor
        oos = new ObjectOutputStream( socket.getOutputStream() );

        //Recogemos la info del servidor
        ois = new ObjectInputStream( socket.getInputStream() );

        boolean salir = false;
        while(!salir){

            //Leo la peticion del cliente
            System.out.print("El numero a adivinar es: ");
            String peticionCliente = scanner.nextLine();

            //Se la envio al servidor
            oos.writeObject(peticionCliente);

            //Escucho la respuesta del servidor
            boolean leerRespuesta = true;
            while(leerRespuesta){

                String respuesta = (String) ois.readObject();

                if( respuesta.contains("acertado") ){

                    leerRespuesta = false;
                    salir = true;
                    System.out.println(respuesta);

                }

            }

        }

    }

}
```

```
    } else if (respuesta.equalsIgnoreCase("fin_peticion")) {  
  
        leerRespuesta = false;  
  
    } else {  
        System.out.println(respuesta);  
    }  
}  
  
}  
  
oos.close();  
ois.close();  
socket.close();  
}  
}
```