PROGRAMACIÓN DE SERVICIOS Y PROCESOS

UNIDAD 3

RETOS

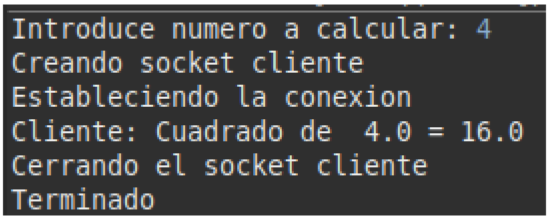
ADRIÁN ESCUDERO 2º DAM

EJERCICIO 1 - CLIENTE

```
package TEMA3.EJ1;

import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.InetSocketAddress;
import java.net.Socket;
import java.util.Scanner;

public class EJ1Cliente {
      public static void main (String[ ] args) throws InterruptedException {
            Scanner sc = new Scanner(System.in);
            System.out.print("Introduce numero a calcular: ");
            Double num = sc.nextDouble();
            try {
                  System.out.println("Creando socket cliente");
                  Socket clientSocket = new Socket();
                  System.out.println("Estableciendo la conexion");
                  InetSocketAddress addr = new InetSocketAddress("localhost",
6000);
                  clientSocket.connect(addr);
                  InputStream is = clientSocket.getInputStream();
                  OutputStream os = clientSocket.getOutputStream();
                  DataInputStream entrada = new DataInputStream(is);
                  DataOutputStream salida = new DataOutputStream(os);
                  salida.writeDouble(num);
                  Double cuadrado = entrada.readDouble();
                  System.out.println("Cliente: Cuadrado de  " + num + " = " +
cuadrado);
                  System.out.println("Cerrando el socket cliente");
                  clientSocket.close();
                  System.out.println("Terminado");
            } catch (IOException e) {
                  e.printStackTrace();
            }
      }
}
```

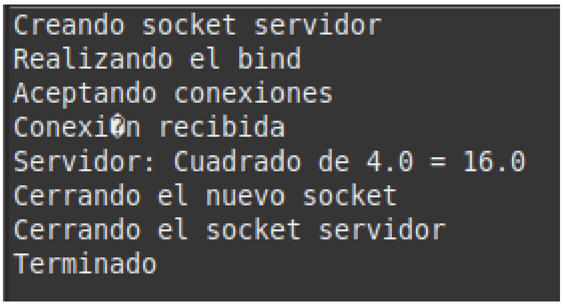


```
Introduce numero a calcular: 4
Creando socket cliente
Estableciendo la conexion
Cliente: Cuadrado de  4.0 = 16.0
Cerrando el socket cliente
Terminado
```

EJERCICIO 1 - SERVIDOR

```
package TEMA3.EJ1;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.InetSocketAddress;
import java.net.Socket;
import java.net.ServerSocket;

public class EJ1Servidor {
      public static void main (String[ ] args) {
            try {
                  System.out.println("Creando socket servidor");
                  ServerSocket serverSocket = new ServerSocket();
                  System.out.println("Realizando el bind");
                  InetSocketAddress addr = new InetSocketAddress("172.26.80.22",
6000) ;
                  serverSocket.bind(addr);
                  System.out.println("Aceptando conexiones");
                  Socket newSocket = serverSocket.accept();
                  System.out.println("Conexi�n recibida");
                  InputStream is = newSocket.getInputStream();
                  OutputStream os = newSocket.getOutputStream();
                  DataInputStream entrada = new DataInputStream(is);
                  DataOutputStream salida = new DataOutputStream(os);
                  Double num = entrada.readDouble();
                  Double cuadrado = num * num;
                  salida.writeDouble(cuadrado);
                  System.out.println("Servidor: Cuadrado de "+num+" =
"+cuadrado);
                  System.out.println("Cerrando el nuevo socket");
                  newSocket.close();
                  System.out.println("Cerrando el socket servidor");
                  serverSocket.close();
                  System.out.println("Terminado");
            } catch (IOException e) {
                  e.printStackTrace();
            }
      }
}
```



```
Creando socket servidor
Realizando el bind
Aceptando conexiones
Conexi�n recibida
Servidor: Cuadrado de 4.0 = 16.0
Cerrando el nuevo socket
Cerrando el socket servidor
Terminado
```

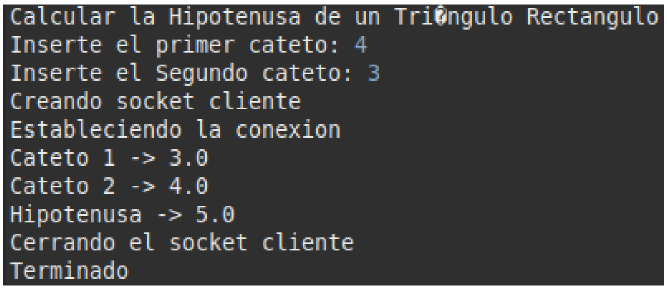
EJERCICIO 2 - CLIENTE

```
package TEMA3.EJ2;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.InetSocketAddress;
import java.net.Socket;
import java.util.Scanner;

public class EJ2Cliente {
      public static void main (String[ ] args) throws InterruptedException {
            Scanner sc = new Scanner(System.in);
            System.out.println("Calcular la Hipotenusa de un Triangulo 
Rectangulo");
            System.out.print("Inserte el primer cateto: ");
            Double cat1 = sc.nextDouble();
            System.out.print("Inserte el Segundo cateto: ");
            Double cat2 = sc.nextDouble();
            try {
                  System.out.println("Creando socket cliente");
                  Socket clientSocket = new Socket();
                  System.out.println("Estableciendo la conexion");
                  InetSocketAddress addr = new InetSocketAddress("172.26.80.23", 
6000);
                  clientSocket.connect(addr);

                  InputStream is = clientSocket.getInputStream();
                  OutputStream os = clientSocket.getOutputStream();
                  DataInputStream dis = new DataInputStream(is);
                  DataOutputStream dos = new DataOutputStream(os);
                  dos.writeDouble(cat1);
                  dos.writeDouble(cat2);
                  Double hipotenusa = dis.readDouble();
                  if (cat1 < cat2) {
                        System.out.println("Cateto 1 -> " + cat1);
                        System.out.println("Cateto 2 -> " + cat2);
                  } else {
                        System.out.println("Cateto 1 -> " + cat2);
                        System.out.println("Cateto 2 -> " + cat1);
                  }
                  System.out.println("Hipotenusa -> " + hipotenusa);

                  System.out.println("Cerrando el socket cliente");
                  clientSocket.close();
                  System.out.println("Terminado");
            } catch (IOException e) {
                  e.printStackTrace();
            }
            sc.close();
      }
}
```

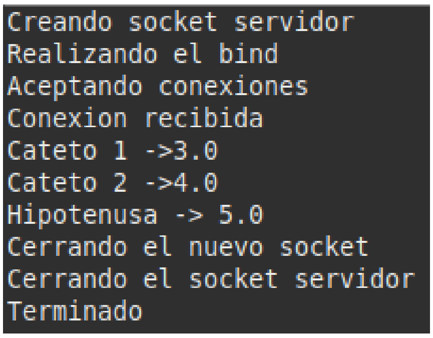


```
Calcular la Hipotenusa de un Tri�ngulo Rectangulo
Inserte el primer cateto: 4
Inserte el Segundo cateto: 3
Creando socket cliente
Estableciendo la conexion
Cateto 1 -> 3.0
Cateto 2 -> 4.0
Hipotenusa -> 5.0
Cerrando el socket cliente
Terminado
```

EJERCICIO 2 - SERVIDOR

```
package TEMA3.EJ2;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.InetSocketAddress;
import java.net.Socket;
import java.net.ServerSocket;

public class EJ2Servidor {
      public static void main (String[ ] args) {
            try {
                  System.out.println("Creando socket servidor");
                  ServerSocket serverSocket = new ServerSocket();
                  System.out.println("Realizando el bind");
                  InetSocketAddress addr = new
InetSocketAddress("172.26.80.22" , 6000) ;
                  serverSocket.bind(addr);
                  System.out.println("Aceptando conexiones");
                  Socket newSocket = serverSocket.accept();
                  System.out.println("Conexion recibida");
                  InputStream is = newSocket.getInputStream();
                  OutputStream os = newSocket.getOutputStream();
                  DataInputStream dis = new DataInputStream(is);
                  DataOutputStream dos = new DataOutputStream(os);
                  Double cat1 = dis.readDouble();
                  Double cat2 = dis.readDouble();
                  Double hipotenusa = Math.sqrt((Math.pow(cat1, 2) +
Math.pow(cat2, 2)));
                  dos.writeDouble(hipotenusa);
                  if (cat1 < cat2) {
                        System.out.println("Cateto 1 ->" + cat1);
                        System.out.println("Cateto 2 ->" + cat2);
                  } else {
                        System.out.println("Cateto 1 ->" + cat2);
                        System.out.println("Cateto 2 ->" + cat1);
                  }
                  System.out.println("Hipotenusa -> " + hipotenusa);
                  System.out.println("Cerrando el nuevo socket");
                  newSocket.close();
                  System.out.println("Cerrando el socket servidor");
                  serverSocket.close();
                  System.out.println("Terminado");
            } catch (IOException e) {
                  e.printStackTrace();
            }
      }
}
```



```
Creando socket servidor
Realizando el bind
Aceptando conexiones
Conexion recibida
Cateto 1 ->3.0
Cateto 2 ->4.0
Hipotenusa -> 5.0
Cerrando el nuevo socket
Cerrando el socket servidor
Terminado
```

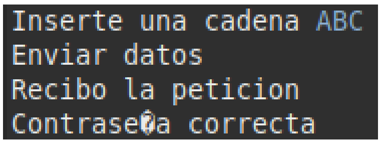
EJERCICIO 3 - CLIENTE

```
package TEMA3.EJ3;

import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;

public class EJ3Cliente {

	public static void main(String[] args) throws IOException {
		Scanner sc = new Scanner(System.in);
		byte[] buffer = new byte[1024];
		System.out.print("Inserte una cadena ");
		String txt = sc.nextLine();
		InetAddress server = InetAddress.getByName("172.26.80.23");
		DatagramSocket socket = new DatagramSocket();
		buffer = txt.getBytes();
		DatagramPacket packet = new DatagramPacket(buffer, buffer.length,
server, 6000);
		System.out.println("Enviar datos");
		socket.send(packet);
		DatagramPacket data = new DatagramPacket(buffer, buffer.length);
		socket.receive(data);
		System.out.println("Recibo la peticion");
		int largo = txt.length();
		int total = 0;
		for (int i = 0; i < largo; i++) {
			char c = txt.charAt(i);
			total+= (int)c;
		}
		txt = new String(data.getData());
		if (total == Integer.parseInt(txt)) {
			System.out.println("Contraseña correcta");
		}
		socket.close();
		sc.close();
	}
}
```



```
Inserte una cadena ABC
Enviar datos
Recibo la peticion
Contrase�a correcta
```

EJERCICIO 3 - SERVIDOR

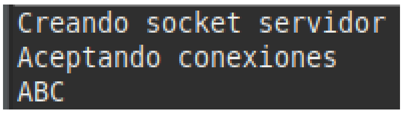
```java
package TEMA3.EJ3;

import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;

public class EJ3Servidor {
      public static void main(String[] args) throws IOException {

            byte[] buffer = new byte[1024];
            System.out.println("Creando socket servidor");
            DatagramSocket socket = new DatagramSocket(6000);
            DatagramPacket packet = new DatagramPacket(buffer, buffer.length);
            socket.receive(packet);
            System.out.println("Aceptando conexiones");
            String txt = new String(packet.getData());
            System.out.println(txt);
            int puertoCliente = packet.getPort();
            InetAddress direccion = packet.getAddress();
            int largo = txt.length();
            int total = 0;
            for (int i = 0; i < largo; i++) {
                  char c = txt.charAt(i);
                  total+= (int)c;
            }
            buffer = Integer.toString(total).getBytes();
            DatagramPacket paquete = new DatagramPacket(buffer, buffer.length, 
direccion, puertoCliente);
            socket.send(paquete);
            socket.close();

      }
}
```



```
Creando socket servidor
Aceptando conexiones
ABC
```

EJERCICIO 4 - CLIENTE

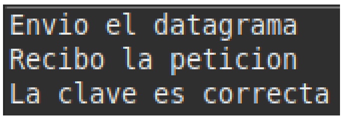
```
package TEMA3.EJ4;
import java.io.IOException;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;


public class EJ4Cliente {

      public static void main(String[] args) throws IOException {
            byte[] buffer = new byte[1024];
            FileReader fr = null;
            BufferedReader br = null;
            FileWriter fw = null;
            File fichero = new File("password.txt");
            fr = new FileReader(fichero);
            br = new BufferedReader(fr);
            String txt = br.readLine();
            InetAddress server = InetAddress.getByName("172.26.80.23");
            DatagramSocket socket = new DatagramSocket();
            buffer = txt.getBytes();
            DatagramPacket packet = new DatagramPacket(buffer, buffer.length,
server, 6000);
            System.out.println("Envio el datagrama");
            socket.send(packet);
            DatagramPacket data = new DatagramPacket(buffer, buffer.length);
            socket.receive(data);
            System.out.println("Recibo la peticion");
            int largo = txt.length();
            int total = 0;
            for (int i = 0; i < largo; i++) {
                  char c = txt.charAt(i);
                  total+= (int)c;
            }
            txt = new String(data.getData());
            fw = new FileWriter(fichero,true);
            if (total == Integer.parseInt(txt)) {
                  System.out.println("La clave es correcta");
                  fw.write("\n"+txt);
            } else {
                  System.out.println("Clave incorrecta");
                  fw.write("\nSe ha intentado acceder con una clave erronea");
            }
            socket.close();
            br.close();
            fr.close();
            fw.close();

      }

}
```



```
Envio el datagrama
Recibo la peticion
La clave es correcta
```

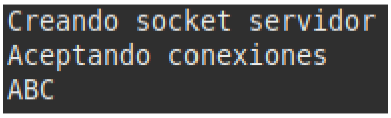
EJERCICIO 4 - SERVIDOR

```
package TEMA3.EJ4;
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;


public class EJ4Servidor {
      public static void main(String[] args) throws IOException {

            byte[] buffer = new byte[1024];
            System.out.println("Creando socket servidor");
            DatagramSocket socket = new DatagramSocket(6000);
            DatagramPacket packet = new DatagramPacket(buffer, buffer.length);
            socket.receive(packet);
            System.out.println("Aceptando conexiones");
            String txt = new String(packet.getData());
            System.out.println(txt);
            int puertoCliente = packet.getPort();
            InetAddress direccion = packet.getAddress();
            int largo = txt.length();
            int total = 0;
            for (int i = 0; i < largo; i++) {
                  char c = txt.charAt(i);
                  total+= (int)c;
            }
            buffer = Integer.toString(total).getBytes();
            DatagramPacket data = new DatagramPacket(buffer, buffer.length,
direccion, puertoCliente);
            socket.send(data);
            socket.close();

      }
}
```



```
Creando socket servidor
Aceptando conexiones
ABC
```