

Examen 1

Nombre del Alumno 1: Cabrera Garibaldi Hernán Galileo

Nombre del Alumno 2: Maceda Nazario Luis Martín

Nombre del profesor: Carlos Alberto Román Zamitis

Clave de la Materia: 2946

Nombre de la Materia: Arquitectura Cliente Servidor

Semestre: 2021-1

Servidor

En este tramo se describe el código variante para el servidor, se optó por hacer una versión del código visto en clase

```
import java.net.*;
import java.io.*;
import java.util.Scanner;

public class Servidor02{
    public static void main(String a[]){
        ServerSocket serverSocket = null;
        Scanner sc = new Scanner(System.in);
        Socket socket = null;
        String peticion = "";
        String respuesta = "";
        String salir = "salir";
        String ls = "ls -l";
        String variable = "";
        String line = "";
        int i = 0;
        //System.out.println(a[0]);
        try{
            System.out.println("Escuchando por el puerto: "+a[0]);
            Integer x = Integer.valueOf(a[0]);
            serverSocket = new ServerSocket(x);
        }
        catch(IOException e){
            System.out.println("java.io.IOException generada");
            e.printStackTrace();
        }
    }
}
```

```

    }
    System.out.println("Esperando a que los clientes se conecten...");
    while(true){
        try{
            socket = serverSocket.accept();
            if (i == 0){
                System.out.println("Se conecto un cliente: " +
socket.getInetAddress().getHostName());
            }
            i = 1;
            DataInputStream entrada = new DataInputStream( socket.getInputStream());
            peticion = entrada.readUTF();
            System.out.println("El mensaje que me envio el cliente es: " + peticion);
            if(peticion.equals(salir)){
                break;
            }
            if(peticion.equals(ls)){
                String[] cmd = {"/bin/bash","-c","ls -l"};
                Process pb = Runtime.getRuntime().exec(cmd);
                BufferedReader input = new BufferedReader(new
InputStreamReader(pb.getInputStream()));
                while ((line = input.readLine()) != null) {
                    //System.out.println(line);
                    variable =variable + line+ "\n";
                }
            }
            DataOutputStream salida = new DataOutputStream( socket.getOutputStream());
            if (peticion.equals(ls)){
                salida.writeUTF(variable);
                System.out.println("Enviando Respuesta del ls -l");
                entrada.close();
                salida.close();
                socket = null;
            }else{
                System.out.println("Escribe Mensaje al cliente:");
                String mensaje = sc.nextLine();
                salida.writeUTF(mensaje);
                System.out.println("El mensaje que le envio al cliente es: " + mensaje);
                entrada.close();
                salida.close();
                socket = null;
            }
        }catch(IOException e){
            System.out.println("java.io.IOException generada");
            e.printStackTrace();
        }
    }
}

```

```
}  
}  
}
```

```
galigaribaldi@MacBook-Pro-de-Hernan Examen1 % javac Servidor02.java  
galigaribaldi@MacBook-Pro-de-Hernan Examen1 % java Servidor02 8000  
Escuchando por el puerto: 8000  
Esperando a que los clientes se conecten...  
█
```

Cliente

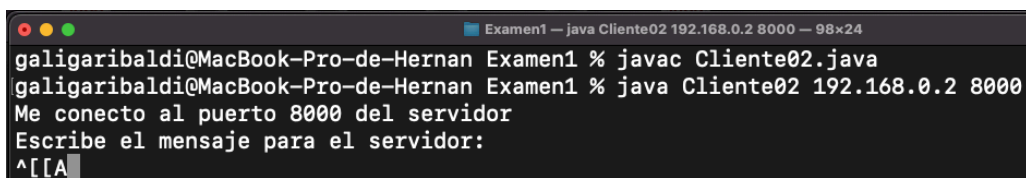
En este tramo se muestra el código correspondiente al cliente

```
import java.net.*;  
import java.io.*;  
import java.util.Scanner;  
  
public class Cliente02{  
    public static void main(String a[]){  
        Socket socket = null;  
        Scanner sc = new Scanner(System.in);  
        String peticion = "";  
        String respuesta = "";  
        String salir="salir";  
        int i=0;  
        while(true){  
            try{  
                Integer x = Integer.valueOf(a[1]);  
                socket = new Socket(a[0],x);  
                DataOutputStream dos = new DataOutputStream( socket.getOutputStream());  
                DataInputStream dis = new DataInputStream( socket.getInputStream() );  
                if (i == 0){  
                    System.out.println("Me conecto al puerto: "+ a[1]);  
                }  
            }  
            catch (Exception e){  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

```

    }
    i = 1;
    System.out.println("Escribe el mensaje para el servidor:");
    peticion = sc.nextLine();
    dos.writeUTF(peticion);
    if (peticion.equals(salir)){
        break;
    }
    else{
        System.out.println("El mensaje enviado: " + peticion);
        respuesta = dis.readUTF();
        System.out.println("El mensaje que me envio el servidor es: " +
respuesta);
        dos.close();
        dis.close();
        socket.close();
    }
}
catch(IOException e){
    System.out.println("java.io.IOException generada");
    e.printStackTrace();
}
}
}
}

```



```

Examen1 — java Cliente02 192.168.0.2 8000 — 98x24
galigaribaldi@MacBook-Pro-de-Hernan Examen1 % javac Cliente02.java
galigaribaldi@MacBook-Pro-de-Hernan Examen1 % java Cliente02 192.168.0.2 8000
Me conecto al puerto 8000 del servidor
Escribe el mensaje para el servidor:
^[[A

```

Ejecutando comando ls -l

En esta parte se ejecuta el comando ls -l desde el programa de Java

```
Examen1 - zsh - 78x24
Escribe el mensaje para el servidor:
Hola
El mensaje enviado: Hola
El mensaje que me envio el servidor es: Hola
Escribe el mensaje para el servidor:
ls -l
El mensaje enviado: ls -l
El mensaje que me envio el servidor es: total 88
drwxr-xr-x@ 5 galigaribaldi staff 160 Dec 9 22:37 Capturas
-rw-r--r-- 1 galigaribaldi staff 1695 Dec 9 22:12 Cliente01A.class
-rw-r--r-- 1 galigaribaldi staff 1123 Dec 9 20:28 Cliente01A.java
-rw-r--r-- 1 galigaribaldi staff 1921 Dec 9 22:36 Cliente02.class
-rw-r--r-- 1 galigaribaldi staff 1227 Dec 9 22:25 Cliente02.java
-rw-r--r-- 1 galigaribaldi staff 3915 Dec 9 22:35 Examen 1.md
-rw-r--r-- 1 galigaribaldi staff 2149 Dec 9 22:16 Servidor01A.class
-rw-r--r-- 1 galigaribaldi staff 1382 Dec 9 22:13 Servidor01A.java
-rw-r--r-- 1 galigaribaldi staff 2851 Dec 9 22:35 Servidor02.class
-rw-r--r-- 1 galigaribaldi staff 2285 Dec 9 22:27 Servidor02.java
-rw-r--r-- 1 galigaribaldi staff 1087 Dec 9 21:16 pruebas.class
-rw-r--r-- 1 galigaribaldi staff 483 Dec 9 21:15 pruebas.java

Escribe el mensaje para el servidor:
salir
galigaribaldi@MacBook-Pro-de-Hernan Examen1 %
```

```
Examen1 - zsh - 78x24
galigaribaldi@MacBook-Pro-de-Hernan Examen1 % java Servidor02 8000
Escuchando por el puerto: 8000
Esperando a que los clientes se conecten...
Se conecto un cliente: 192.168.0.2
El mensaje que me envio el cliente es: Hola
Escribe Mensaje al cliente:
^C
galigaribaldi@MacBook-Pro-de-Hernan Examen1 % java Servid
or02 8000
Escuchando por el puerto: 8000
Esperando a que los clientes se conecten...
Se conecto un cliente: 192.168.0.2
El mensaje que me envio el cliente es: Hola
Escribe Mensaje al cliente:
Hola
El mensaje que le envio al cliente es: Hola
El mensaje que me envio el cliente es: ls -l
Enviando Respuesta del ls -l
salir
El mensaje que me envio el cliente es: salir
galigaribaldi@MacBook-Pro-de-Hernan Examen1 % salir
zsh: command not found: salir
galigaribaldi@MacBook-Pro-de-Hernan Examen1 %
```