



# EXAMEN SOCKETS TCP/IP

**MARIO JIMÉNEZ MARSET**

**ÍNDICE**

1. ENUNCIADO – OBJETIVOS .....	3
2. DESARROLLO – PROCEDIMIENTOS.....	3

## 1. ENUNCIADO – OBJETIVOS

En este examen se pedía crear una aplicación cliente – servidor, el cual contuviese sockets de protocolo TCP/IP. El servidor proporciona a los clientes la dirección IP y un mensaje de bienvenida o, en su defecto, la fecha y hora del servidor (el cliente elige que petición elige a través de un pequeño menú que proporciona el servidor).

## 2. DESARROLLO – PROCEDIMIENTOS

Se muestra el código del programa comentado.

Código Clase Cliente:

```
package examen;
import java.io.*;
import java.net.*;
import java.util.*;

public class Cliente {
    public static void main(String[] args) {
        //Clase Cliente TCP/IP
        //se crean las variables a utilizar en el programa
        String direccion="127.0.0.1";
        int numeroPuerto=5000;
        DataInputStream in;
        DataOutputStream out;
        Scanner entrada=new Scanner(System.in);
        try {
            //dentro de un bloque try-catch, se crea el Socket que recoge la dirección IP y el
            número de puerto
            Socket sc = new Socket(direccion, numeroPuerto);
            //con este método se establece un tiempo de espera: si no llega una respuesta,
            saldrá un mensaje de error
            sc.setSoTimeout(5000);
            //se inicializan los inputs que recogen y envían la información
            in = new DataInputStream(sc.getInputStream());
            out = new DataOutputStream(sc.getOutputStream());
            //se lee la petición recogida del servidor
            String peticion=in.readUTF();
            System.out.println("La peticion es "+peticion);
            //dependiendo de cuál sea la petición, se mostrará en pantalla una información
            recibida u otra
            if(peticion.equalsIgnoreCase("IP")) {
                String leerBienvenida = in.readUTF();
                System.out.println("El mensaje de bienvenida es: "+leerBienvenida);
                String leerIpServidor = in.readUTF();
                System.out.println("La IP del servidor es: "+leerIpServidor);
            }
            else{
                String fechaHoraRecibida=in.readUTF();
                System.out.println("La fecha y hora del servidor es: "+fechaHoraRecibida);
            }
            //se cierran el Scanner y el Socket
            entrada.close();
            sc.close();
        } catch (IOException e) {
```

```

        System.out.println(e.getMessage());
    }
}
}

```

Código Clase Servidor:

```

package examen;
import java.io.*;
import java.net.*;
import java.text.SimpleDateFormat;
import java.util.*;

public class Servidor {
    public static void main(String[] args) {
        //Clase Servidor TCP/IP
        //se crean las variables a utilizar en el programa
        ServerSocket servidor=null;
        Socket sc=null;
        DataInputStream in;
        DataOutputStream out;
        Scanner entrada=new Scanner(System.in);
        int numeroPuerto = 5000;
        try {
            //se crea el socket servidor que tendrá por parámetro el puerto
            servidor = new ServerSocket(numeroPuerto);
            System.out.println("Servidor iniciado");
            //se crea un bucle infinito
            while (true) {
                //con el método accept, se espera a que algún cliente se conecte
                sc = servidor.accept();
                System.out.println("Cliente conectado");
                //se inicializan los inputs que recogen y envían información
                in = new DataInputStream(sc.getInputStream());
                out = new DataOutputStream(sc.getOutputStream());
                //con un método de la clase InetAddress, se recoge cual es la IP del servidor
                InetAddress direccion=InetAddress.getLocalHost();
                //se envía la petición
                System.out.println("Cual es tu petición\n1-IP\n2-Fecha Hora");
                String peticion=entrada.nextLine();
                out.writeUTF(peticion);
                //dependiendo de cual sea la petición, se mandará la información de un
                //condicional u otro
                if(peticion.equalsIgnoreCase("IP")) {
                    out.writeUTF("Bienvenido al Servidor");
                    out.writeUTF("La direccion IP del servidor es
"+direccion.getHostAddress());
                }else if(peticion.equalsIgnoreCase("Fecha Hora")) {
                    //se crea la clase que muestra la fecha del servidor actual
                    Date dateServidor = new Date();
                    System.out.println("Un cliente ha mandado una peticion de fecha y
hora");
                    Date dateCliente = new Date();
                    String formato=new SimpleDateFormat("HH:mm:ss:SS
dd/MM/yyyy").format(dateCliente);
                    String mensaje = new String(formato);
                    //se envía el mensaje que muestra fecha y hora del servidor
                    out.writeUTF(mensaje);
                }
            }
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
    }
}

```

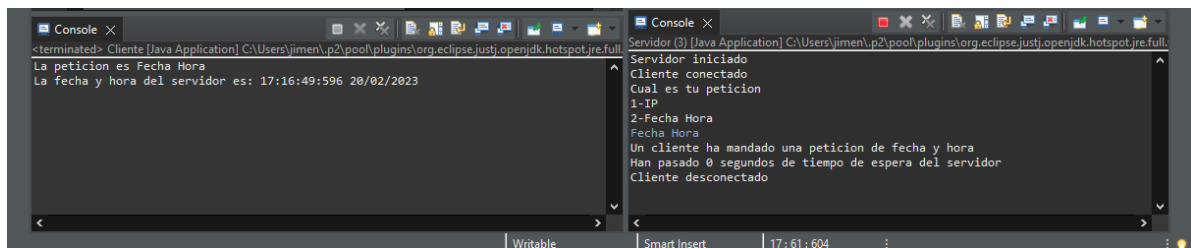
```

        long horaServidor = dateServidor.getTime();
        long horaCliente = dateCliente.getTime();
        long diferencia = horaCliente - horaServidor;
        System.out.println("Han pasado " + diferencia / 1000 + " segundos de
tiempo de espera del servidor");
    }
    //se cierra el socket y scanner
    sc.close();
    entrada.close();
    System.out.println("Cliente desconectado");
}
} catch (IOException e) {
    System.out.println(e.getMessage());
}
}
}
}

```

### RESULTADOS CAPTURAS CONSOLA:

Opción 1:



Opción 2:

