

- 1.- Ejemplo 1**
- 2.- Ejemplo 2**
- 3.- Ejemplo 3**
- 4.- Ejemplo 4**
- 5.- Ejemplo 5**
- 6.- Ejemplo 6**

1.- Ejemplo 1

servidor.java

```
import java.io.* ;
import java.net.* ;
class Servidor {
static final int Puerto=2000;
public Servidor( ) {
try
{
ServerSocket sServidor = new ServerSocket(Puerto);
System.out.println("Escucho el puerto " + Puerto );
for ( int nCli = 0; nCli < 3; nCli++)
{
Socket sCliente = sServidor.accept();
System.out.println ("Sirvo al cliente " + nCli);
DataOutputStream flujo_salida= new DataOutputStream (sCliente.getOutputStream());
flujo_salida.writeUTF( "Hola cliente " + nCli );
sCliente.close();
}
System.out.println("Se han atendido los clientes");
}
catch( Exception e )
{
System.out.println ( e.getMessage() );
} }
public static void main( String[] arg ) {
new Servidor();
}
}
```

2.- Ejemplo 2

Servidor.java

```
import java.io.* ;
import java.net.* ;
class Servidor {
static final int Puerto=2000;
public Servidor( ) {
try
{
ServerSocket sServidor = new ServerSocket(Puerto);
System.out.println("Escucho el puerto " + Puerto );
for ( int nCli = 0; nCli < 3; nCli++) {
Socket sCliente = sServidor.accept();
System.out.println ("Sirvo al cliente " + nCli);
DataOutputStream flujo_salida= new DataOutputStream (sCliente.getOutputStream());
flujo_salida.writeUTF( "Hola cliente " + nCli );
sCliente.close(); }
}
```

```

System.out.println("Se han atendido los clientes");
}
catch( Exception e )
{
System.out.println
( e.getMessage() );
}
}
public static void main( String[] arg )
{
new Servidor();
}
}

```

3.- Ejemplo 3

Servidor.java

```

import java.io.* ;
import java.net.* ;
class Servidor extends Thread{
Socket skCliente;
static final int Puerto=2000;
public Servidor(Socket sCliente)
{
skCliente=sCliente;
}
public static void main( String[] arg )
{
try{
// Inicio el servidor en el puerto
ServerSocket skServidor = new ServerSocket(Puerto);
System.out.println("Escucho el puerto " + Puerto );
while(true){
// Se conecta un cliente
Socket skCliente = skServidor.accept();
System.out.println("Cliente conectado");
// Atiendo al cliente mediante un thread
new Servidor(skCliente).start();
} }
catch (Exception e)
{;}
}
public void run(){
try {
// Creo los flujos de entrada y salida
DataInputStream flujo_entrada = new DataInputStream(skCliente.getInputStream());
DataOutputStream flujo_salida= new DataOutputStream(skCliente.getOutputStream());
// ATENDER PETICIÓN DEL CLIENTE
flujo_salida.writeUTF("Se ha conectado el cliente de forma correcta");
// Se cierra la conexión
skCliente.close();
}
}
}

```

```

System.out.println("Cliente desconectado");
}
catch( Exception e )
{
System.out.println( e.getMessage() );
} } }

```

Lógicamente, el funcionamiento del cliente no cambia ya que la concurrencia la realiza el servidor.

4.- Ejemplo 4

Cliente.java

```

import java.io.*;
import java.net.*;
class Cliente {
static final String HOST = "localhost";
static final int Puerto=2000;
public Cliente( ) {
try{
Socket sCliente = new Socket( HOST , Puerto );
// Creo los flujos de entrada y salida
DataInputStream flujo_entrada = new DataInputStream(skCliente.getInputStream());
DataOutputStream flujo_salida= new DataOutputStream(skCliente.getOutputStream());
// TAREAS QUE REALIZA EL CLIENTE
String datos=flujo_entrada.readUTF();
System.out.println(datos);
sCliente.close(); }
catch( Exception e )
{
System.out.println( e.getMessage() );
} }
public static void main( String[] arg )
{
new Cliente();
} }

```

5.- Ejemplo 5

Servidor.java

```

import java.io.* ;
import java.net.* ;
import java.util.Date;
class Servidor {
static final int Puerto=2000;
public Servidor( ) {
try {
// Inicio el servidor en el puerto
ServerSocket sServidor = new ServerSocket(Puerto);
System.out.println("Escucho el puerto " + Puerto );
// Se conecta un cliente Socket sCliente = sServidor.accept();

```

```

// Crea objeto
System.out.println("Cliente conectado");
// Creo los flujos de entrada y salida
DataInputStream flujo_entrada = new DataInputStream( sCliente.getInputStream());
DataOutputStream flujo_salida= new DataOutputStream(sCliente.getOutputStream());
// CUERPO DEL ALGORITMO
long tiempo1=(new Date()).getTime();
flujo_salida.writeUTF(Long.toString(tiempo1));
// Se cierra la conexión
sCliente.close();
System.out.println("Cliente desconectado");
}
catch( Exception e ) {
System.out.println( e.getMessage() );
} }
public static void main( String[] arg ) {
new Servidor();
}
}

```

6.- Ejemplo 6

```

Cliente.java
import java.io.*;
import java.net.*;
import java.util.Date;
class Cliente {
    static final String HOST = "localhost";
    static final int Puerto=2000;
    public Cliente( ) {
        String datos=new String();
        String num_cliente=new String();
        // para leer del teclado
        BufferedReader reader=new BufferedReader(new InputStreamReader(System.in));
        try{
            // Me conecto al puerto
            Socket sCliente = new Socket( HOST , Puerto );
            // Creo los flujos de entrada y salida
            DataInputStream flujo_entrada = new DataInputStream(sCliente.getInputStream());
            DataOutputStream flujo_salida= new DataOutputStream(sCliente.getOutputStream());
            // CUERPO DEL ALGORITMO
            datos=flujo_entrada.readUTF();
            long tiempo1=Long.valueOf(datos);
            long tiempo2=(new Date()).getTime();
            System.out.println("\n El tiempo es:"+(tiempo2-tiempo1));
            // Se cierra la conexión
            sCliente.close();
        }
        catch ( Exception e ) {
            System.out.println( e.getMessage() );
        }
    }
}

```

```
public static void main( String[] arg ) {  
    new Cliente ();  
}  
}
```