**UD3. Boletín 3. Programación de comunicaciones en red**

1. Crea un Servidor que pueda devolver el día, mes, año u hora que un cliente pueda pedir. El servidor debe enviar un menú al cliente cuando se abra la conexión, el cliente debe escoger el dato que desea, y el servidor debe enviárselo. Una ver mostrado por la pantalla del cliente debe cerrarse la coexión.

import java.io.BufferedReader;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.InputStreamReader;

import java.net.Socket;

public class User {

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

Socket s = new Socket("localhost", 6666);

DataInputStream din = new DataInputStream(s.getInputStream());

DataOutputStream dout = new DataOutputStream(s.getOutputStream());

BufferedReader br = new BufferedReader(new InputStreamReader(System.***in***));

System.***out***.println("Server says: " + din.readUTF());

String str = br.readLine();

dout.writeUTF(str);

dout.flush();

String str2 = din.readUTF();

System.***out***.println("Server says: " + str2);

dout.close();

s.close();

}

}

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.net.ServerSocket;

import java.net.Socket;

import java.time.LocalDate;

import java.time.LocalTime;

public class Server {

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

ServerSocket ss = new ServerSocket(6666);

Socket s = ss.accept();

DataInputStream din = new DataInputStream(s.getInputStream());

DataOutputStream dout = new DataOutputStream(s.getOutputStream());

LocalTime hora = java.time.LocalTime.*now*();

LocalDate fecha = java.time.LocalDate.*now*();

dout.writeUTF("Que quieres mostrar, fecha o hora actual?(fecha -> 1 hora -> 2): ");

dout.flush();

String respuestaUser = din.readUTF();

if ("1".equals(respuestaUser)) {

dout.writeUTF(fecha.toString());

dout.flush();

}

if ("2".equals(respuestaUser)) {

dout.writeUTF(hora.toString());

dout.flush();

}

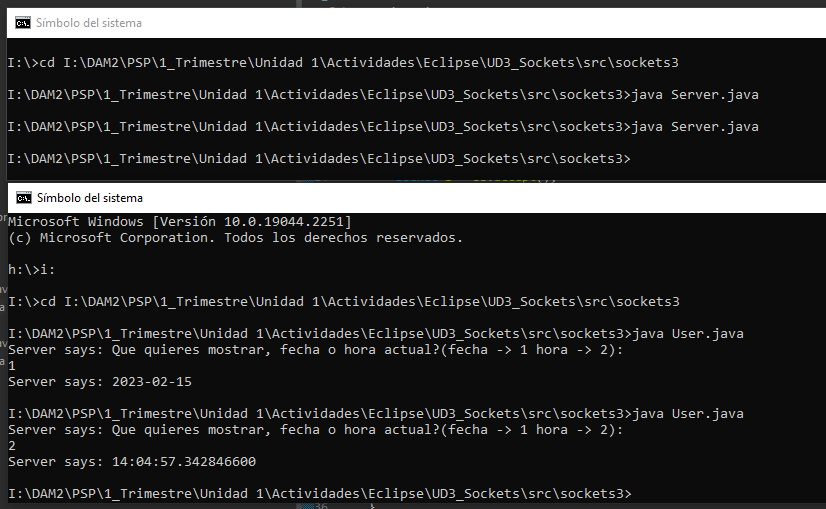
din.close();

s.close();

ss.close();

}

}



1. Crea un Servidor que pueda enviar a un cliente el contenido de un fichero .txt preexistente.

public class ServerFichero {

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

ServerSocket ss = new ServerSocket(6666);

Socket s = ss.accept();

DataInputStream din = new DataInputStream(s.getInputStream());

DataOutputStream dout = new DataOutputStream(s.getOutputStream());

String contenido = "";

FileReader fr = new FileReader("I:\\DAM2\\PSP\\1\_Trimestre\\Unidad 1\\Actividades\\Eclipse\\UD3\_Sockets\\leeme.txt");

BufferedReader br = new BufferedReader(fr);

do {

contenido = br.readLine();

} while (fr.read() != -1);

dout.writeUTF(contenido);

dout.flush();

din.close();

s.close();

ss.close();

}

}

public class UserRecibeFichero {

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

Socket s = new Socket("localhost", 6666);

DataInputStream din = new DataInputStream(s.getInputStream());

DataOutputStream dout = new DataOutputStream(s.getOutputStream());

String str2 = din.readUTF();

System.***out***.println("Server says: " + str2);

dout.close();

din.close();

s.close();

}

}

