Sockets en java III

Queremos crear un server que cuando reciba conexión de un cliente cree un hilo y lo lance hasta que reciba la cadena salir:

Clase SimpleThread

```java
package sockets2LaPelicula;

public class SimpleThread extends Thread{
    public static final int FOR_EVER = -1;//Constante
    protected long delay;//variable
    protected int times;//Variable
    public SimpleThread (long delay, int times) {
        System.out.println("Constructor SimpleThread");
        this.delay = delay;
        this.times = times;
    }
    public void run() {
        try {
            for(int aux = times; (times>=0)||(aux == FOR_EVER); times--) {
                System.out.println("Mi Delay es:"+delay);
                sleep(delay);
            }
        }catch(Exception e) {
            System.out.println("Error.");
        }
    }


}
```

Clase Servidor

```java
package sockets2LaPelicula;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;

import sockets2LaPelicula.Cliente2;
```

```
public class Servidor2 {
    public static int PUERTO = 5000;//Indicamos el puerto del servidor
    public void ejecutarHastaCadenaSalir() {//Se ejecuta hasta recibir
cadena salir
        try {


        System.out.println("Lanzando Servidor...");
        ServerSocket serverSocket = new ServerSocket(Servidor2.PUERTO);
        Socket socket = serverSocket.accept();

        BufferedReader br = new BufferedReader(new
InputStreamReader(socket.getInputStream()));//Buffer entrada
        PrintWriter pw = new PrintWriter(new
OutputStreamWriter(socket.getOutputStream()));//Buffer salida

        String cadenaRecibida = "";
        while(!(cadenaRecibida = br.readLine()).equalsIgnoreCase("Salir"))
{//Mientras no se reciba salir ejecuta esto
            pw.println(cadenaRecibida);
            System.out.println("No se ha enviado salir");
            initialize();
            pw.flush();
        }
        System.out.println("Cerrando servidor.....");


        }catch(IOException e){
            e.printStackTrace();
        }
    }




    public static void main(String[] args) {
        Servidor2 server = new Servidor2();
        server.ejecutarHastaCadenaSalir();
    }
}

</code java>

===== Clase Cliente =====


<code>
package sockets2LaPelicula;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.net.Socket;
import java.util.Scanner;

import sockets2LaPelicula.Cliente2;
import sockets2LaPelicula.Servidor2;

public class Cliente2 {
    public void ejecutar() {
        try {


        System.out.print("Lanzando conexión....");

        Socket socket = new Socket ("127.0.0.1",
Servidor2.PUERTO);//Conectamos al servidor

        System.out.println("[OK]");


        BufferedReader br = new BufferedReader(new
InputStreamReader(socket.getInputStream()));//Bufer de lectura
        PrintWriter pw = new PrintWriter(new
OutputStreamWriter(socket.getOutputStream()));//Buffer de escritura

        String cadenaRecibida = br.readLine();
        System.out.println(cadenaRecibida);

        pw.println("cadena recibida" + cadenaRecibida);
        pw.flush();

        System.out.println("Fin del Cliente");

        }catch(IOException e){
            e.printStackTrace();
        }
    }




    public void ejecutarInfinito() {
        try {


        System.out.print("Lanzando conexión....");

        Socket socket = new Socket ("127.0.0.1",
Servidor2.PUERTO);//Conectamos al servidor

        System.out.println("[OK]");


        BufferedReader br = new BufferedReader(new
InputStreamReader(socket.getInputStream()));//Bufer de lectura
        PrintWriter pw = new PrintWriter(new
OutputStreamWriter(socket.getOutputStream()));//Buffer de escritura

        String lineaLeida ="";
        Scanner teclado = new Scanner(System.in);

        while(true) {
            lineaLeida = teclado.nextLine();
            pw.println(lineaLeida);
            pw.flush();
        }


        }catch(IOException e){
            e.printStackTrace();
        }
    }





    public static void main(String[] args) {
        Cliente2 client = new Cliente2();
        client.ejecutarInfinito();

    }
}
```