

## Programación Avanzada

### Juan Manuel Fernández Peña Curso 2013

A continuación se muestra un par de enchufes del tipo TCP. Para cada uno se tiene una clase generadora y otra con el enchufe propiamente dicho. Aunque se pueden reunir en una sola y a veces reunir en un solo método toda la actividad de cada enchufe, resulta conveniente separar funciones para poder utilizarlos en aplicaciones más complejas.

#### Enchufe servidor

```
import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.ServerSocket;
import java.net.Socket;

public class EnchufeServidor {
    private int port = 1234;
    private ServerSocket recibe;
    private Socket serv;
    private BufferedReader entra;
    private DataOutputStream sale;

    public EnchufeServidor(){
        try{
            recibe = new ServerSocket(port);
            System.out.println("Conectado a "+recibe.getInetAddress());
        }catch(IOException io){
            //falta
        }
    }

    public boolean conecta(){
        // Este método representa una espera por una solicitud; una vez recibida
        // se establece contacto por medio de dos streams
        boolean resp = false;
        try{
            serv = recibe.accept();
            entra = new BufferedReader(new
InputStreamReader(serv.getInputStream()));
            sale = new DataOutputStream(serv.getOutputStream());
            resp = true;
        }catch(IOException io){
            //falta
            System.err.println("Falla al conectar");
            serv=null;
        }
        return resp;
    }

    public String recibe(){
        String rec = "";
        try{
            rec = entra.readLine();
            System.out.println("Recibi: "+rec+"|");
        }catch(IOException io){
            //falta
            System.err.println("Falla al recibir");
        }
        return rec;
    }
}
```

```

}

public boolean envía(String men){
    boolean resp = false;
    try{
        sale.writeBytes(men+"\n");
        sale.flush();
        resp = true;
        System.out.println("Envie: "+men+"|");
    }catch(IOException io){
        //falta
        System.err.println("Falla al enviar");
    }
    return resp;
}
}

```

```

public class LanzaServidor {
    private EnchufeServidor miServidor;

    public LanzaServidor(){
        miServidor = new EnchufeServidor();
    }

    public void enEspera(){
        int nc = 0;
        String lin ="vacío";
        while (miServidor.conecta()){
            nc++;
            System.out.println("Inicia una conversación "+nc);
            while (!lin.equals("FIN")){

                lin = miServidor.recibe();
                System.out.println("Llegó "+lin);
                //aquí podría ir una acción previa a responder
                miServidor.envía("Recibido "+lin);
            }
            System.out.println("Termina la conversación "+nc);
            lin = "vacío";
        }
    }

    public static void main(String[] args) {
        System.out.println("Inicia lanzamiento de enchufe servidor");
        LanzaServidor lan = new LanzaServidor();
        lan.enEspera();
    }
}

```

## Enchufe Cliente

```
import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.Socket;

public class EnchufeCliente {

    //elementos para enlace
    private Socket cliente;
    private DataOutputStream sale;
    private BufferedReader entra;

    // valores por omisión
    private String host="localhost";
    private int port = 1234;

    public EnchufeCliente(String hh, int pp){
        host = hh;
        port = pp;
    }

    // Los métodos que siguen dejan pendiente la atención a IOException
    public boolean conecta()throws IOException{
        boolean resp=false;
        cliente = new Socket(host, port);
        sale = new DataOutputStream(cliente.getOutputStream());
        entra = new BufferedReader(new InputStreamReader(cliente.getInputStream()));
        resp = true;
        return resp;
    }

    public String dameMensaje()throws IOException{
        String res ="###";
        System.out.println("Voy a recibir: "+res);
        res = entra.readLine();
        System.out.println("Cliente recibe: "+res+"|");
        return res;
    }

    public boolean envíaMensaje(String mm)throws IOException{
        boolean resp = false;
        sale.writeBytes(mm+"\n");
        resp = true;
        System.out.println("Cliente envió: "+mm+"|");
        return resp;
    }

    public boolean cierra()throws IOException{
        boolean resp = false;
        entra.close();
        sale.close();
        cliente.close();
        resp = true;
        return resp;
    }
}
```

```

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Comunicate {
    EnchufeCliente enlace;

    public Comunicate(){
        enlace = new EnchufeCliente("localhost", 1234);
    }

    public void platica(){
        String linsal = "";
        String lin="";

        try{
            BufferedReader consola = new BufferedReader(new
InputStreamReader(System.in));
            enlace.conecta();
            System.out.print("Deme mensaje:");
            System.out.flush();
            lin = consola.readLine();
            while (!lin.equals("FINIS")){
                enlace.envíaMensaje(lin);
                linsal="$$$";
                linsal=enlace.dameMensaje();
                System.out.println("Recibimos respuesta: "+linsal);
                System.out.print("Deme mensaje:");
                System.out.flush();
                if (!lin.equals("FIN"))
                    lin = consola.readLine();
                else lin ="FINIS";
            }
        }catch (IOException ie){
            //falta
            System.out.println("Hubo un fallo en la conversación "+ie);
            ie.printStackTrace();
        }
    }

    public static void main(String[] args) {
        System.out.println("Comienza plática con proceso");
        Comunicate comu = new Comunicate();
        comu.platica();
        System.out.println("Termina plática entre procesos");
    }
}

```