# CONEXIONES MÚLTIPLES CLIENTES. HILOS

## Introducción a los chats

El desarrollo de interfaces visuales arrastrando componentes se facilita mediante la instalación del plugin **Windows Builder**.

Esta instalación suele llevarse a cabo en Eclipse a través de la opción *Help/Install new software…,* seleccionando la versión del IDE que se esté utilizando en el apartado *Work with:.* A continuación se seleccionarán todos aquellos componentes relacionados con Swing (filtros SWT y swing) y se aceptan los términos de la licencia. Después se reinicia Eclipse.

A continuación, y dentro de Eclipse, se selecciona el asistente de instalación (wizard) de Swing mediante una opción análoga a la siguiente: *File/New/WindowBuilder/Swing Designer/Application Windows*. Se selecciona la carpeta en donde está el código fuente (Source folder), el nombre del paquete (packet) y se proporciona un nombre de clase (Name). A partir de este momento, aparece una nueva pestaña llamada design, justo debajo de la pestaña de código.

## Creación de un chat TCP

Este chat está formado por los siguientes archivos fuente de Java:

* ServidorChat.java
* HiloServidor.java
* ClienteChat.java

Algunas de **características de este chat** son las siguientes:

* Varios clientes se conectan a un servidor
* Cada cliente es atendido por un hilo de ejecución
* Cada vez que un cliente envía un mensaje al servidor, éste se reenviará a todos los clientes conectados
* En el servidor se visualiza el número de clientes conectados así como la conversación mantenida entre ellos
* Con el botón Salir se finaliza el chat
* El servidor se mantiene a la escucha a través de un puerto concreto
* La oportuna adaptación del cliente permitirá la ejecución del programa en un servidor remoto

El detalle de cada uno de los archivos fuente de este chat se muestra a continuación:

**//ServidorChat.java:**

**import** java.awt.Color;

**import** java.awt.EventQueue;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.io.DataInputStream;

**import** java.io.DataOutputStream;

**import** java.io.IOException;

**import** java.net.ServerSocket;

**import** java.net.Socket;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTextArea;

**import** javax.swing.JTextField;

**public** **class** ServidorChat **extends** JFrame **implements** ActionListener {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**static** ServerSocket *servidor*;

**static** **final** **int** ***PUERTO***=44444; // Puerto en el que se escucha

**static** **int** *CONEXIONES* = 0; // Cuenta el número de conexiones

**static** **int** *ACTUALES* = 0;

**static** **int** *MAXIMO* = 10;

**static** JTextField *mensaje*= **new** JTextField("");

**static** JTextField *mensaje2*= **new** JTextField("");

**private** JScrollPane scrollpanel;

**static** JTextArea *textarea*;

JButton salir = **new** JButton("Salir");

**static** Socket *table*[]=**new** Socket[10]; /// Almacena los sockets de los clientes

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

*servidor* = **new** ServerSocket(***PUERTO***);

System.***out***.println("Servidor iniciado...");

ServidorChat pantalla = **new** ServidorChat();

pantalla.setBounds(0,0,540,400);

pantalla.setVisible(**true**);

*mensaje*.setText("NUMERO DE CONEXIONES ACTUALES");

///Se admiten hasta 10 Conexiones

**while** (*CONEXIONES* < *MAXIMO*) {

Socket s = **new** Socket();

s=*servidor*.accept(); /// Esperando al cliente

*table*[*CONEXIONES*]=s; //Guardamos el socket

*CONEXIONES* ++;

*ACTUALES* ++;

HiloServidor hilo=**new** HiloServidor(s);

hilo.start();

}// Fin while

// Cuando finaliza bucle cerrar servidor si no se ha cerrado antes

**if** (!*servidor*.isClosed()) {

*mensaje2*.setForeground(Color.***red***);

*mensaje2*.setText("Máximo número de conexiones establecidas" + *CONEXIONES*);

*servidor*.close();

}

System.***out***.println("Servidor finalizado...");

}

/\*\*

\* Create the application.

\*/

**public** ServidorChat() {

**super** ("VENTANA DEL SERVIDOR DE CHAT");

getContentPane().setLayout(**null**);

*mensaje*.setBounds (10,10, 400,30);

getContentPane().add(*mensaje*);

*mensaje*.setEditable(**false**);

*mensaje2*.setBounds(10,348,400,30);

getContentPane().add(*mensaje2*);

*mensaje2*.setEditable(**false**);

*textarea* = **new** JTextArea();

scrollpanel = **new** JScrollPane(*textarea*);

scrollpanel.setBounds(10,50,500,300);

getContentPane().add(scrollpanel);

salir.setBounds(420,10, 100,30);

getContentPane().add(salir);

*textarea*.setEditable(**false**);

salir.addActionListener(**this**);

setDefaultCloseOperation (JFrame.***DO\_NOTHING\_ON\_CLOSE***);

}

/////// Cuando pulsamos el botón click

@Override

**public** **void** actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

**if** (e.getSource() == salir)

**try** {

*servidor*.close();

System.*exit*(0);

} **catch** (IOException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

}

}

**//HiloServidor.java**

**import** java.io.DataInputStream;

**import** java.io.DataOutputStream;

**import** java.io.IOException;

**import** java.net.Socket;

**public** **class** HiloServidor **extends** Thread {

DataInputStream fentrada;

Socket socket = **null**;

**public** HiloServidor (Socket s) {

socket = s;

**try** {

//ServidorChat.mensaje2.setText("**TODO** OK");

fentrada = **new** DataInputStream (socket.getInputStream());

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

System.***out***.println("ERROR E/S");

e.printStackTrace();

}

}

**public** **void** run() {

ServidorChat.*mensaje*.setText("NUMERO DE CONEXIONES ACTUALES: "+ServidorChat.*ACTUALES*);

// Nada más conectarse el cleiente le envío todos los mensajes

String texto= ServidorChat.*textarea*.getText();

EnviarMensajes(texto);

**while** (**true**) {

String cadena="";

**try** {

cadena=fentrada.readUTF(); // Leo lo que el cliente escribe

**if** (cadena.trim().equals("\*")) {

ServidorChat.*ACTUALES*--;

ServidorChat.*mensaje*.setText("NUMERO DE CONEXIONES ACTUALES" + ServidorChat.*ACTUALES*);

**break**;// Salir del while

}

ServidorChat.*textarea*.append(cadena + "\n");

texto = ServidorChat.*textarea*.getText();

EnviarMensajes(texto); /// Envio texto a todos los clientes

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}// fin While

}

// ENVIA LOS MENSAJES DEL TEXTAREA A LOS CLIENTES DEL CHAT

**private** **void** EnviarMensajes (String texto) {

**int** i;

**for** (i = 0; i< ServidorChat.*CONEXIONES*; i++) {

Socket s1= ServidorChat.*table*[i];// .tabla[i];

**try** {

DataOutputStream fsalida = **new** DataOutputStream(s1.getOutputStream());

fsalida.writeUTF(texto);

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

} // Fin for

} // Fin Enviar mensajes

}

}

**//ClienteChat.java**

**import** java.awt.ScrollPane;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.io.DataInputStream;

**import** java.io.DataOutputStream;

**import** java.io.IOException;

**import** java.net.Socket;

**import** java.net.UnknownHostException;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JOptionPane;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTextArea;

**import** javax.swing.JTextField;

**public** **class** ClienteChat **extends** JFrame **implements** ActionListener {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

Socket socket = **null**;

// Streams

DataInputStream fentrada; // para leer mensajes de todos

DataOutputStream fsalida; // para escribir sus mensajes

String nombre;

**static** JTextField *mensaje*=**new** JTextField();

**private** JScrollPane scrollpane1;

**static** JTextArea *textarea1*;

JButton boton = **new** JButton ("Enviar");

JButton desconectar = **new** JButton ("Salir");

**boolean** repetir = **true**;

**public** **static** **void** main(String[] args) {

**int** puerto=44444;

String nombre = JOptionPane.*showInputDialog*("Introduce tu nombre o Nick");

Socket s= **null**;

// Cliente y servidor se ejecutan en la máquina local

**try** {

s=**new** Socket ("localhost", puerto);

} **catch** (UnknownHostException e) {

// **TODO** Auto-generated catch block

JOptionPane.*showMessageDialog*(**null**, "IMPOSIBLE CONECTAR CON EL SERVIDOR\n" + e.getMessage(),"<<MENSAJE DE ERROR: 1>>",JOptionPane.***ERROR\_MESSAGE***);;

System.*exit*(0);

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**if** (!nombre.trim().equals("")) {

ClienteChat cliente = **new** ClienteChat(s,nombre);

cliente.setBounds(0,0,540,500);

cliente.setVisible(**true**);

cliente.ejecutar();

} **else**{

System.***out***.println("El nombre está vacío,....");

}

}

**public** **void** run() {

ServidorChat.*mensaje*.setText("NUMERO DE CONEXIONES ACTUALES: "+ ServidorChat.*ACTUALES*);

// Nada más conectarse el cleiente le envío todos los mensajes

String texto= ServidorChat.*textarea*.getText();

EnviarMensajes(texto);

}

// ENVIA LOS MENSAJES DEL TEXTAREA A LOS CLIENTES DEL CHAT

**private** **void** EnviarMensajes (String texto) {

**int** i;

**for** (i = 0; i< ServidorChat.*CONEXIONES*; i++) {

Socket s1= ServidorChat.*table*[i];// .tabla[i];

**try** {

DataOutputStream fsalida = **new** DataOutputStream(s1.getOutputStream());

fsalida.writeUTF(texto);

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

} // Fin for

} // Fin Enviar mensajes

}

/\*\*

\* Creamos el constructor

\*/

**public** ClienteChat (Socket s, String nombre) {

**super** ("CONEXIÓN DEL CLIENTE CHAT: " + nombre);

setLayout (**null**);

*mensaje*.setBounds(10, 10, 400, 30);add(*mensaje*);

*textarea1* = **new** JTextArea();

scrollpane1 = **new** JScrollPane(*textarea1*);

scrollpane1.setBounds(10,50,400,300);add(scrollpane1);

boton.setBounds(420,10,100,30);add(boton);

desconectar.setBounds(420,50,100,30);add(desconectar);

*textarea1*.setEditable(**false**);

boton.addActionListener(**this**);

desconectar.addActionListener(**this**);

setDefaultCloseOperation(JFrame.***DO\_NOTHING\_ON\_CLOSE***);

socket = s;

**this**.nombre=nombre;

////////

**try** {

fentrada = **new** DataInputStream(socket.getInputStream());

fsalida = **new** DataOutputStream(socket.getOutputStream());

String texto = " > Entra en el Chat ..." + nombre;

fsalida.writeUTF(texto); //escribe mensaje de entrada

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

////// Boton click

@Override

**public** **void** actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

**if** (e.getSource() == boton) {

String texto = nombre + "> "+ *mensaje*.getText();

**try** {

*mensaje*.setText("");

fsalida.writeUTF(texto);

} **catch** (IOException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

}

**if** (e.getSource() == desconectar) {

String texto = " > Abandona el Chat ..."+nombre;

**try** {

fsalida.writeUTF(texto);

fsalida.writeUTF("\*");

repetir = **false**; // Para salir del bucle

} **catch** (IOException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

}

}

**public** **void** ejecutar () {

String texto = "";

**while** (repetir) {

**try** {

texto = fentrada.readUTF();

*textarea1*.setText(texto);

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

JOptionPane.*showMessageDialog*(**null**, "IMPOSIBLE CONECTAR CON EL SERVIDOR\n" + e.getMessage(),"<<MENSAJE DE ERROR: 2>>",JOptionPane.***ERROR\_MESSAGE***);;

repetir = **false**; // Salir del bucle

}

} // while

**try** {

socket.close();

System.*exit*(0);

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

} // Ejecutar

}

## Creación de un chat UDP

Aunque en el siguiente enlace:

<https://www.daniweb.com/programming/software-development/threads/392710/basic-udp-chat-system>

aparece el código Java de tanto el servidor como del cliente de un chat UDP, a continuación se incluye el código del cliente (ChatUDP.java) de un chat de este tipo:

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.io.IOException;

**import** java.net.InetAddress;

**import** java.net.MulticastSocket;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JOptionPane;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTextArea;

**import** javax.swing.JTextField;

**import** java.net.\*;

**public** **class** ChatUDP **extends** JFrame **implements** ActionListener, Runnable

{

**static** JTextField *mensaje*=**new** JTextField();

**private** JScrollPane scrollpane1;

**static** JTextArea *textarea1*;

JButton boton = **new** JButton ("Enviar");

JButton desconectar = **new** JButton ("Salir");

**static** String *nombre*;

**static** **byte**[] *buf* = **new** **byte**[1000];

**static** InetAddress *grupo* = **null**;

**static** **int** *Puerto* = 12345;

**static** **boolean** *repetir*=**true**;

**static** MulticastSocket *ms*;

**public** ChatUDP (String nombre) {

**super** ("CONEXIÓN DEL CLIENTE CHAT: " + nombre);

setLayout (**null**);

*mensaje*.setBounds(10, 10, 400, 30);add(*mensaje*);

*textarea1* = **new** JTextArea();

scrollpane1 = **new** JScrollPane(*textarea1*);

scrollpane1.setBounds(10,50,400,300);add(scrollpane1);

boton.setBounds(420,10,100,30);add(boton);

desconectar.setBounds(420,50,100,30);add(desconectar);

*textarea1*.setEditable(**false**);

boton.addActionListener(**this**);

desconectar.addActionListener(**this**);

setDefaultCloseOperation(JFrame.***DO\_NOTHING\_ON\_CLOSE***);

}

**public** **static** **void** main(String[] args) **throws** IOException

{

*nombre* = JOptionPane.*showInputDialog*("Introduce tu nombre o Nick");

*ms* =**new** MulticastSocket (*Puerto*);

*grupo*= InetAddress.*getByName*("225.0.0.1");

*ms*.joinGroup(*grupo*);

**if** (!*nombre*.trim().equals("")) {

ChatUDP server = **new** ChatUDP(*nombre*);

server.setBounds(0,0,540,400);

server.setVisible(**true**);

**new** Thread (server).start();

}

**else** {

System.***out***.println("El nombre esta vacio");

}

} // main

**public** **void** actionPerformed(ActionEvent e)

{

**if**(e.getActionCommand()=="Enviar")

{

**try** {

String texto = *nombre* + " >> " +*mensaje*.getText();

DatagramPacket paquete=**new** DatagramPacket(texto.getBytes(),texto.length(),*grupo*,*Puerto*);

*ms*.send(paquete);

*mensaje*.setText("");

} **catch** (IOException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

}

**if**(e.getActionCommand()=="Salir")

{

**try** {

String texto = *nombre* + " >> " +"Abandona el chat";

DatagramPacket paquete=**new** DatagramPacket(texto.getBytes(),texto.length(),*grupo*,*Puerto*);

*ms*.send(paquete);

*ms*.close();

*repetir*=**false**;

System.***out***.println("Abandona el chat:"+*nombre*);

System.*exit*(0);

} **catch** (IOException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

}

}

@Override

**public** **void** run() {

**while** (*repetir*) {

**try** {

DatagramPacket p = **new** DatagramPacket(*buf*, *buf*.length);

*ms*.receive(p);

String texto=**new** String (p.getData(), 0, p.getLength());

*textarea1*.append(texto+"\n");

}

**catch** (SocketException s) {System.***out***.println(s.getMessage()); }

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

}

}

}