UNIVERSIDAD DE NARIÑO
SISTEMAS DISTRIBUIDOS
GUIA SOCKET EN JAVA – CHAT /CLIENTE-SERVIDOR
PARTE I

PRESENTACIÓN

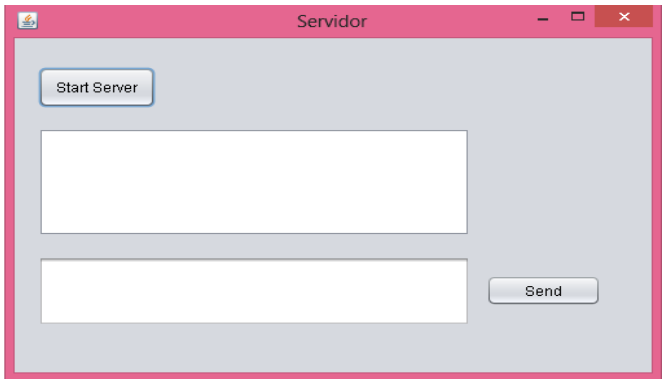
Los Socket son interfaces que facilitan la comunicación y el paso de mensajes entre aplicaciones Cliente/Servidor de una manera segura y usando protocolos TCP/IP o UDP.

El siguiente ejercicio muestra los pasos para implementar un Chat entre dos aplicaciones, una Cliente y otra Servidor usando una interfaz Socket en donde el servidor abre una conexión y el cliente solicita al servidor un servicio y comienza el intercambio de mensajes entre ambas aplicaciones. La aplicación esta implementada en Java.

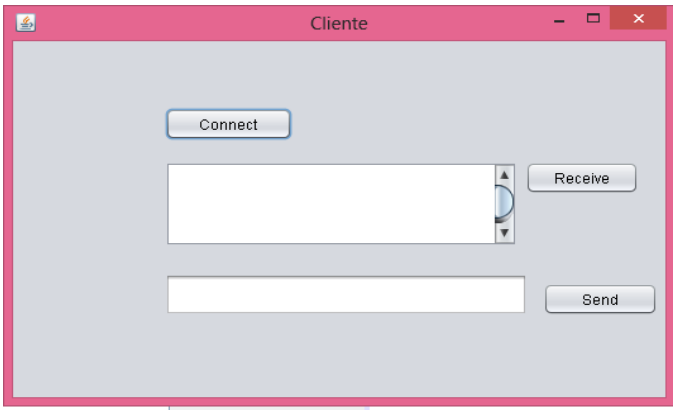
OBJETIVO:

Implementar un Chat entre una aplicación Cliente y una Servidor usando la Interfaz Socket en Java.

Forma Servidor

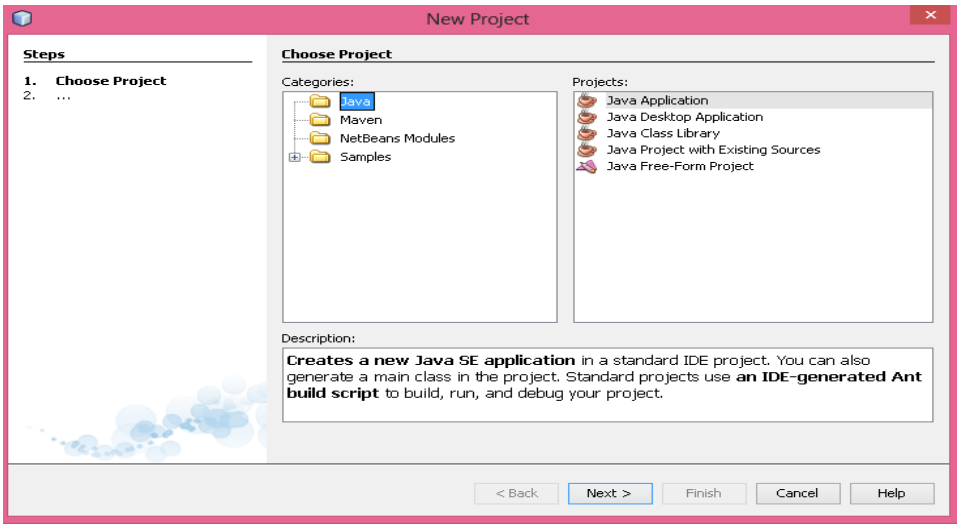


Forma Cliente

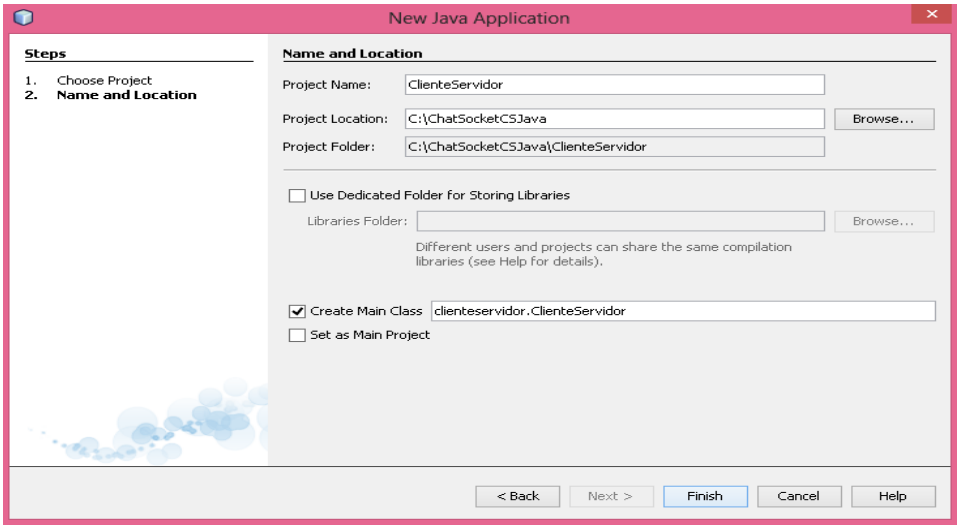


PASOS:

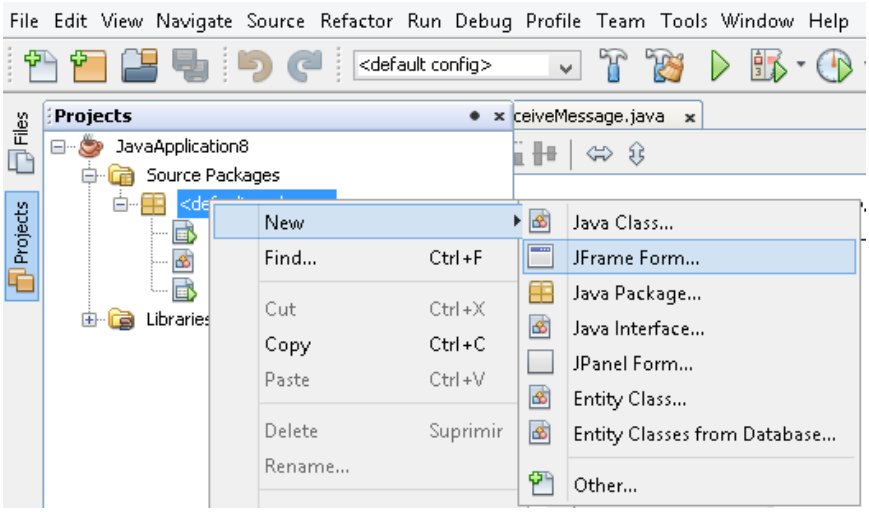
1. Abrir Netbeans y crear un nuevo Projecto Cliente/Servidor (Java Application)

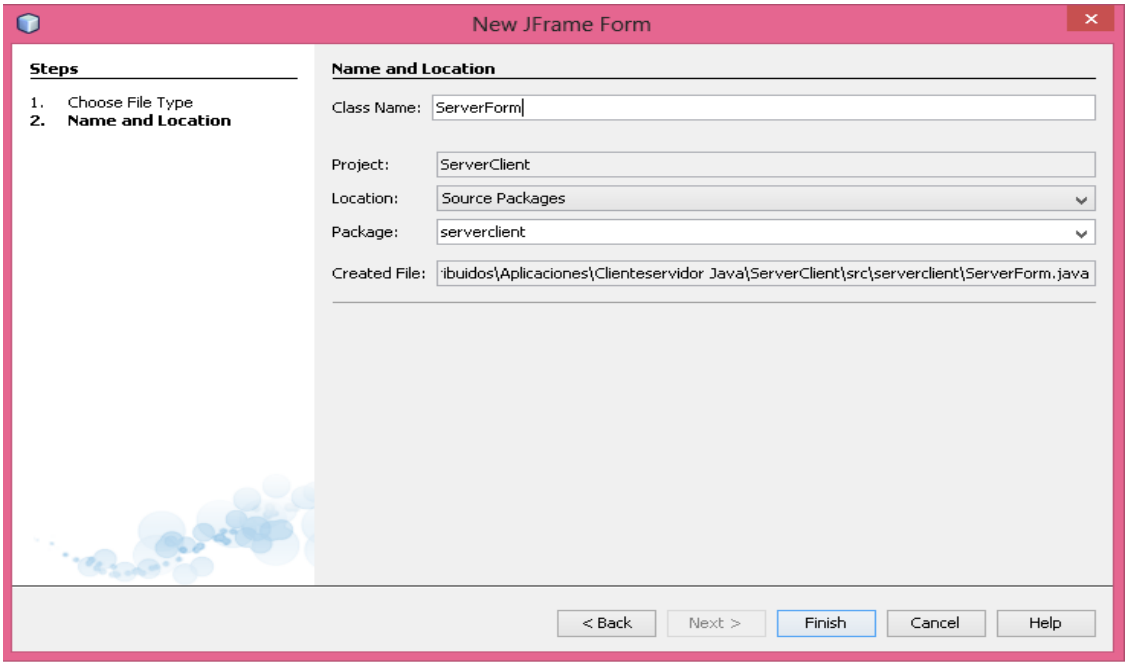


El nombre del Project puede ser ClienteServidor. Se debe tener precaución en asignar una carpeta para guardar el proyecto.

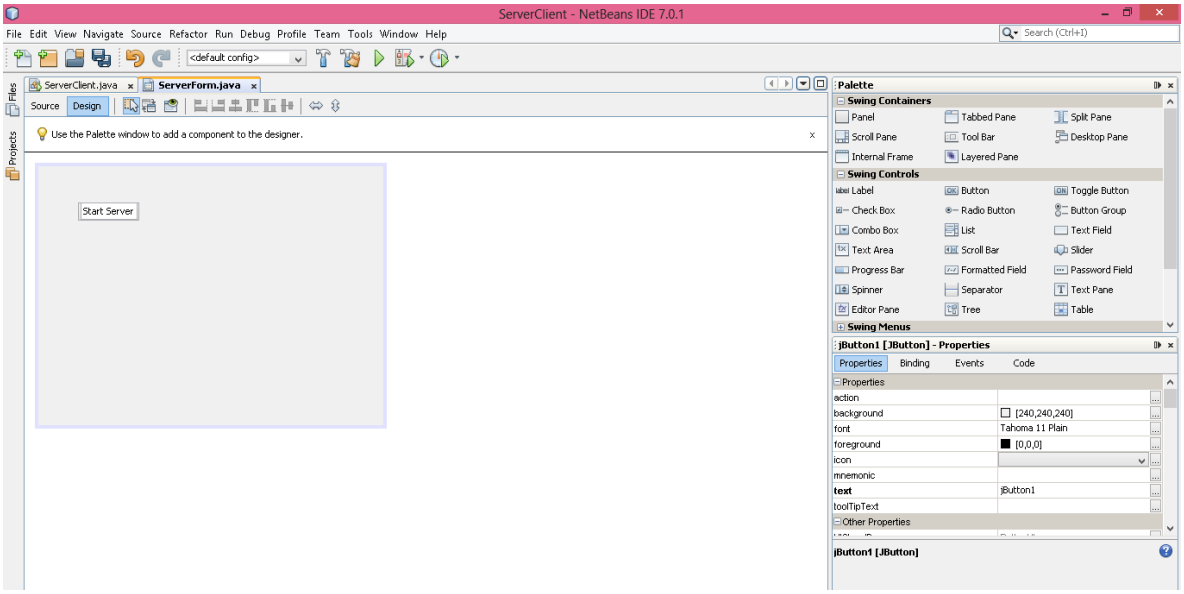


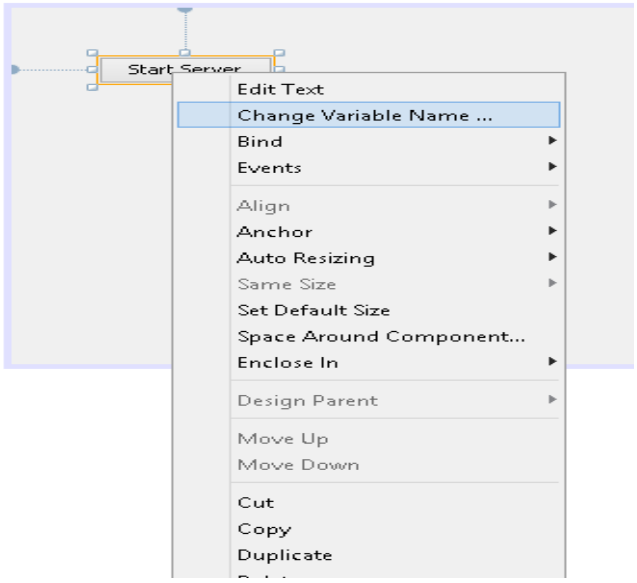
2. **Crear una nueva JFrame con el nombre de ServerForm en Source Package dando click derecho en <default package>**

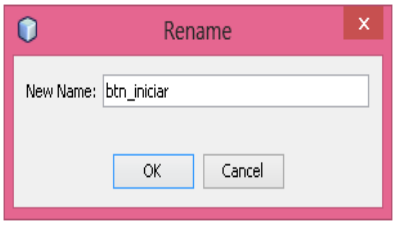




3. **En la forma ServerForm colocamos un Boton al cual le cambiamos el rotulo por Start Server y le cambiamos en propiedades el name: btn_iniciar o dando click derecho sobre el botón y en Change Variable Name**



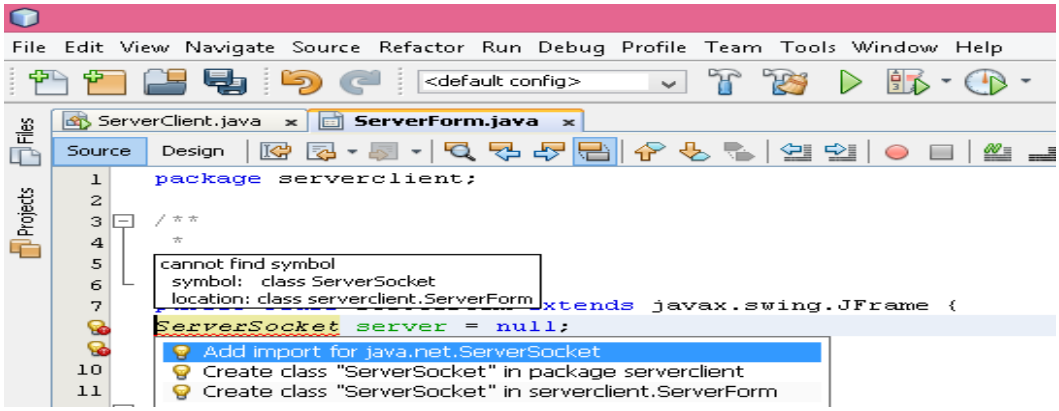




4. **En la pestaña source entramos al código de la clase e inicializamos las variables de tipo socket tanto del cliente como del servidor con null así:**
   Socket client = null;
   ServerSocket server = null;

Para lo anterior es necesario importar las clases del paquete *java.net*. Estas clases proporcionan comunicación de red independiente del sistema. Entre ellas están: *SocketServer y Socket* que usan el protocolo TCP para comunicarse a través de la red.

***Socket*: Implementa un extremo de la conexión TCP.**
***ServerSocket*: Se encarga de implementar el extremo Servidor de la conexión en la que se esperarán las conexiones de los clientes.**



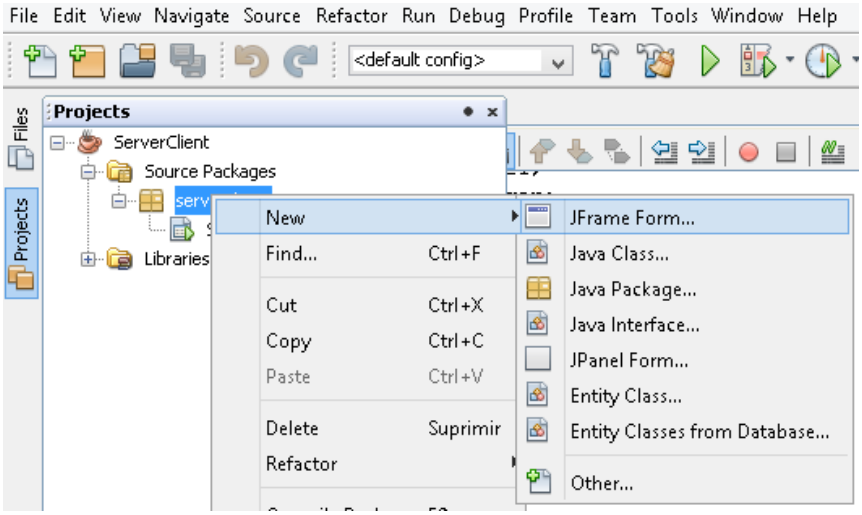

```
1  package serverclient;
2
3  import java.net.ServerSocket;
4  import java.net.Socket;
5
6  /**
7   *
8   * @author Usuario
9   */
10 public class ServerForm extends javax.swing.JFrame {
11 ServerSocket server = null;
12 Socket client = null;
13
```

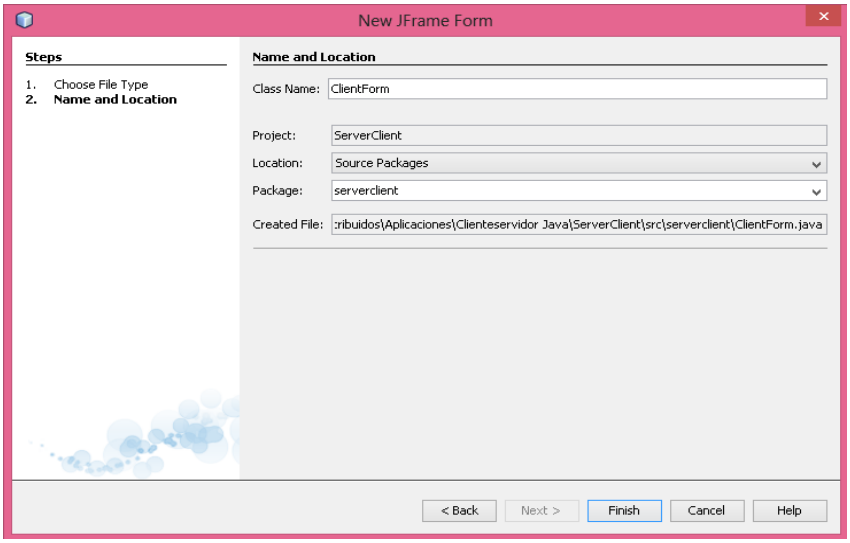
5. **Entrar en modo Design para programar el Botón de Start Server. Para ello damos Click Derecho sobre el Boton + Events + Actión + Action Performed, e inicializamos las variables por donde escucha el servidor y el puerto. Se deben controlar las excepciones con Try y Catch. Se debe importar la clase *javax.swing.JOptionPane*.**

```
private void btn_iniciarActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            server = new ServerSocket(6969);
            client = server.accept();
            JOptionPane.showMessageDialog(null,"Cliente Aceptado");
        } catch (IOException ex) {
            JOptionPane.showMessageDialog(null,"Cliente no Disponible");
            Logger.getLogger(ServerForm.class.getName()).log(Level.SEVERE, null, ex);
        }
}
```

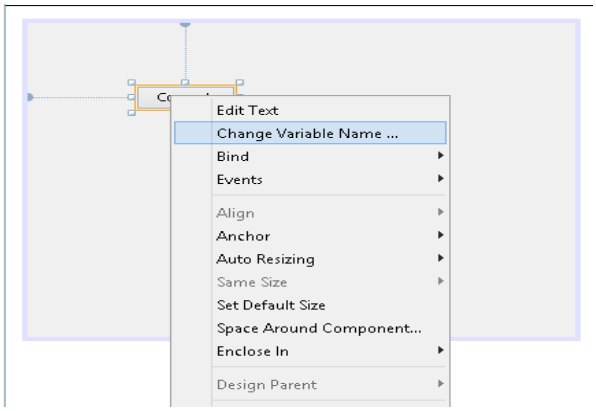
Terminamos en este punto inicial de crear la conexión del servidor el cual se encarga de escuchar.

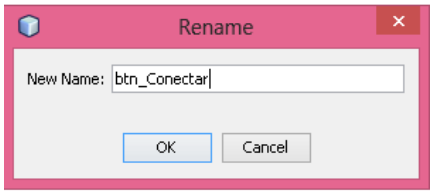
6. **Ahora iniciamos creando la aplicación cliente, para ello creamos una nueva Jframe en Project + <default Package> Click derecho + New + JframeForm y le asignamos el nombre ClientForm.**





7. **En la ClientForm colocamos un Boton al cual le cambiamos el rotulo por Connect y en name le asignamos btn_Conectar**





8. **En la pestaña Source entramos a la clase ClientForm e iniciamos la variable** *server con null* **e importamos la clase** *import java.net.Socket* **así***:*

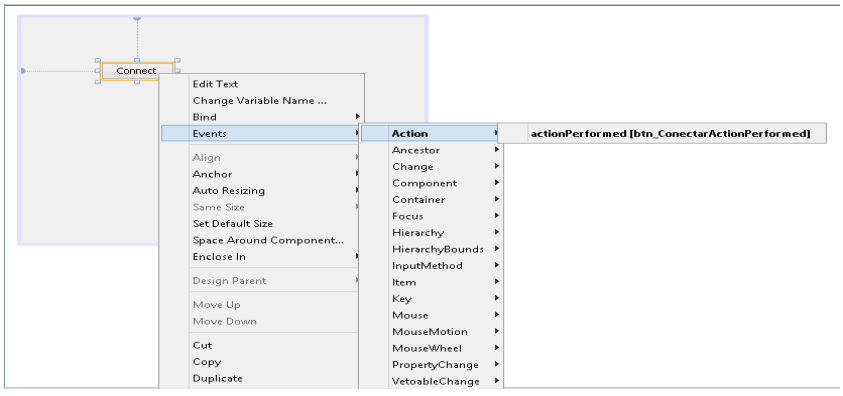
```
import java.net.Socket;


public class ClientForm extends javax.swing.JFrame {

    Socket server = null;
```

9. **En modo de diseño programamos el Boton Connect dando click derecho sobre el botón + Events + Actión+ Actión Performed.**

En esta clase programamos el socket del cliente para que conozca la dirección Ip del Servidor y el puerto por el cual escucha. De igual manera manejamos las excepciones con el Try y el Catch.

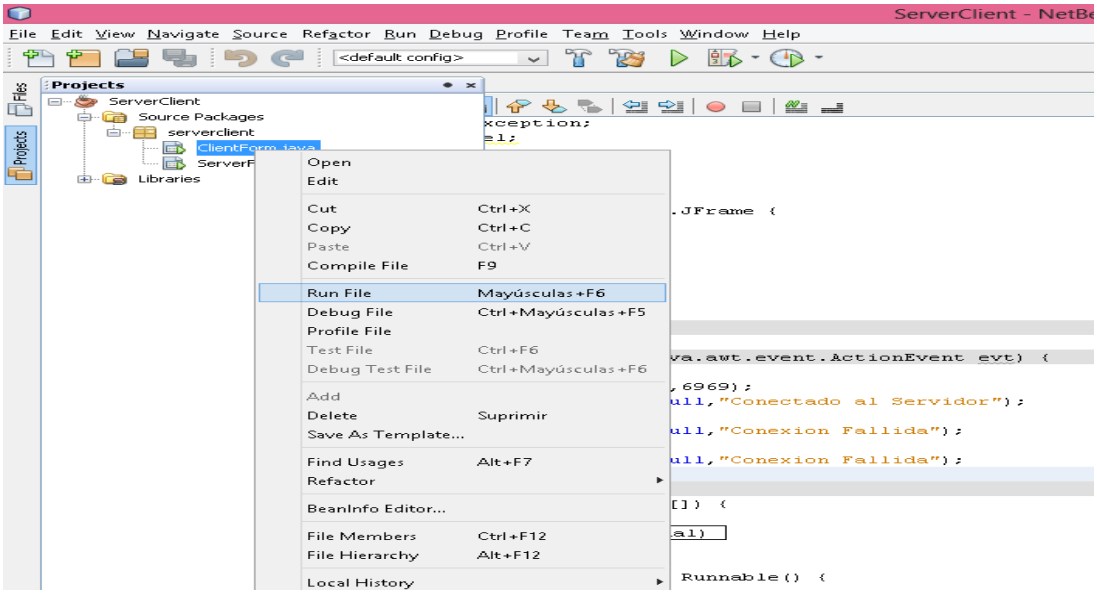


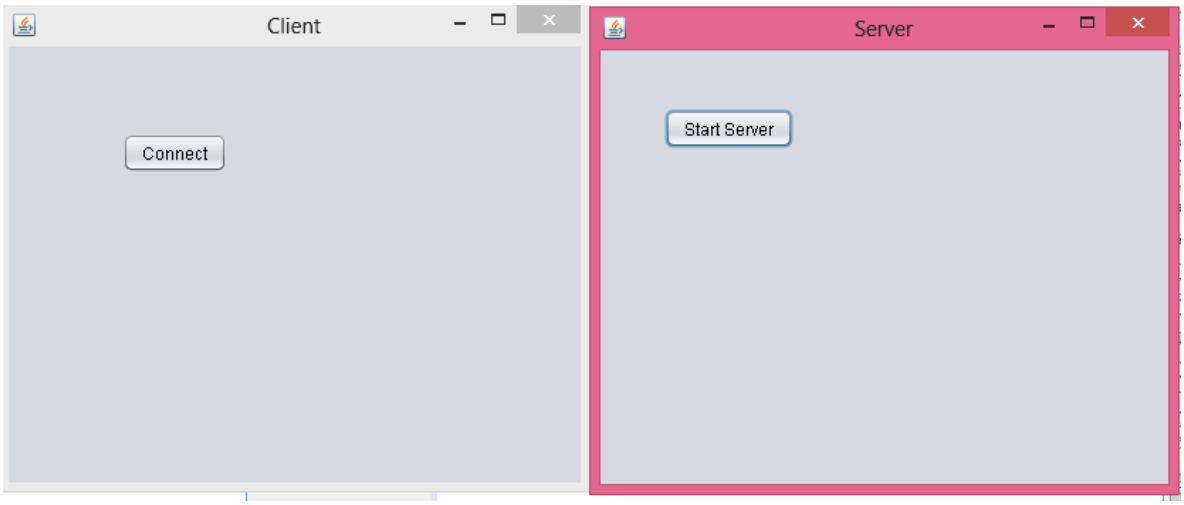
Se debe importar la clase *import javax.swing.JOptionPane;*

```
private void btn_ConectarActionPerformed(java.awt.event.ActionEvent evt) {
        try {
            server = new Socket("127.0.0.1",6969);
            JOptionPane.showMessageDialog(null,"Conectado al Servidor");
        } catch (UnknownHostException ex) {
            JOptionPane.showMessageDialog(null,"Conexion Fallida");
        } catch (IOException ex) {
            JOptionPane.showMessageDialog(null,"Conexion Fallida");
        }
}
```

Hasta este punto hemos terminado de implementar las conexiones entre cliente y servidor y para ello corremos los dos programas para probar la comunicación.

10. **Damos clic derecho sobre la clase *ServerForm y Run,* lo mismo que sobre la *Clase ClientForm y Run***




Client
Connect
Server
Start Server