

GUÍA N° 4 – MINICHAT

FACULTAD	CURSO	AMBIENTE
INGENIERÍA	SISTEMAS DISTRIBUIDOS	LABORATORIO DE DESARROLLO DE SOFTWARE 77C0203

ELABORADO POR	DAVID GÁLVEZ GÁLVEZ	APROBADO POR	ARTURO RIVERA
VERSIÓN	001	FECHA DE APROBACIÓN	01/03/2020

1. LOGRO GENERAL DE UNIDAD DE APRENDIZAJE

Al finalizar la sesión, el estudiante conocerá la importancia de trabajar con el entorno de comunicación, utilizando Netbeans como entorno de trabajo cliente servidor.

2. OBJETIVOS ESPECÍFICOS DE LA PRÁCTICA

- El estudiante ingresa a Netbeans
- El estudiante reconoce el entorno de comunicación.
- El estudiante aprender a crear una aplicación tipo Chat.

3. MATERIALES Y EQUIPOS

- Sistema Operativo Windows.
- Software Apache NetBeans IDE 19
- Software Java (TM) SE Development Kit 17 / 21

4. PAUTAS DE SEGURIDAD

Las computadoras y laptops deben de estar prendidas mientras se usan. Pero al terminar el laboratorio estas deben dejarse apagadas.

- En el laboratorio debe estar prendido el aire acondicionado para evitar sobrecalentamientos y averías, especialmente en épocas de altas temperaturas.

- Los estudiantes no pueden llevar alimentos que puedan derramar sobre los computadores.
- Computadoras, router, switch, puntos de acceso (caídas).
- Eléctricos, por contacto directo o indirecto, electricidad estática y por fenómeno térmico. Puede producir: electrocuciones y quemaduras.
- Procedimiento ante Corte de Energía Eléctrica
- No tocar el equipo eléctrico en el que se encuentra trabajando, puede que retorne la energía.
- Comunicarse con el Asistente de Operaciones de turno quien se comunicará con el Técnico.

5. FUNDAMENTO

Se trata de implementar un chat distribuido.

La aplicación cliente/servidor estará formada por un servidor que contiene en un objeto de información sobre los clientes conectados al chat, y el clientes que hablarán entre ellos.

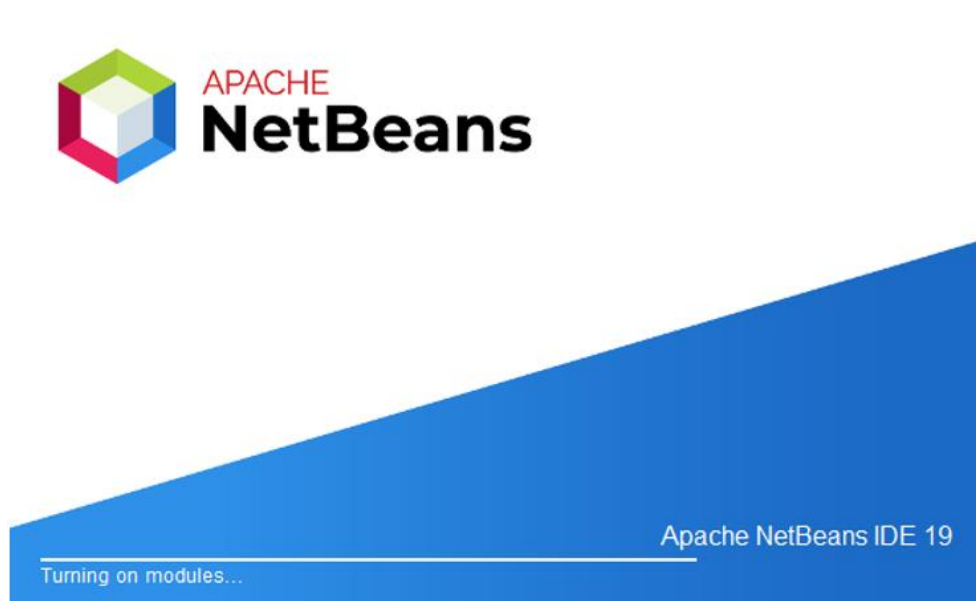
Cada vez que un cliente envíe un mensaje al servidor, éste lo enviará al resto de clientes.

6. PROCEDIMIENTO (DESARROLLO DE LA PRÁCTICA)

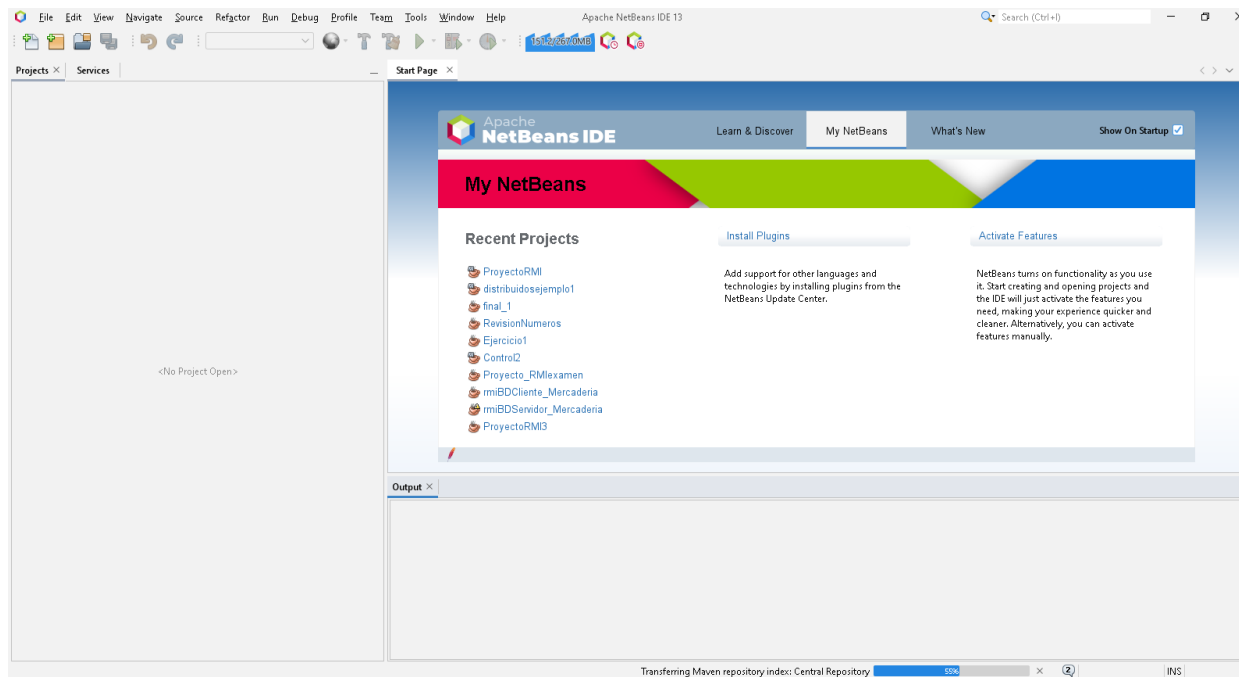
Comenzar con el entorno de trabajo Netbeans, corresponde buscar acceso directo en el escritorio:



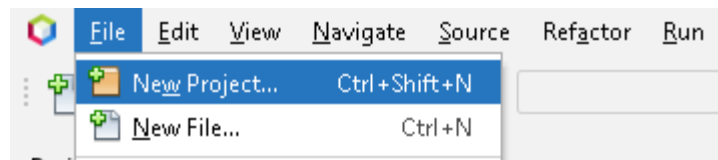
Aparece en pantalla una ventana de donde empiezan a cargar el software.



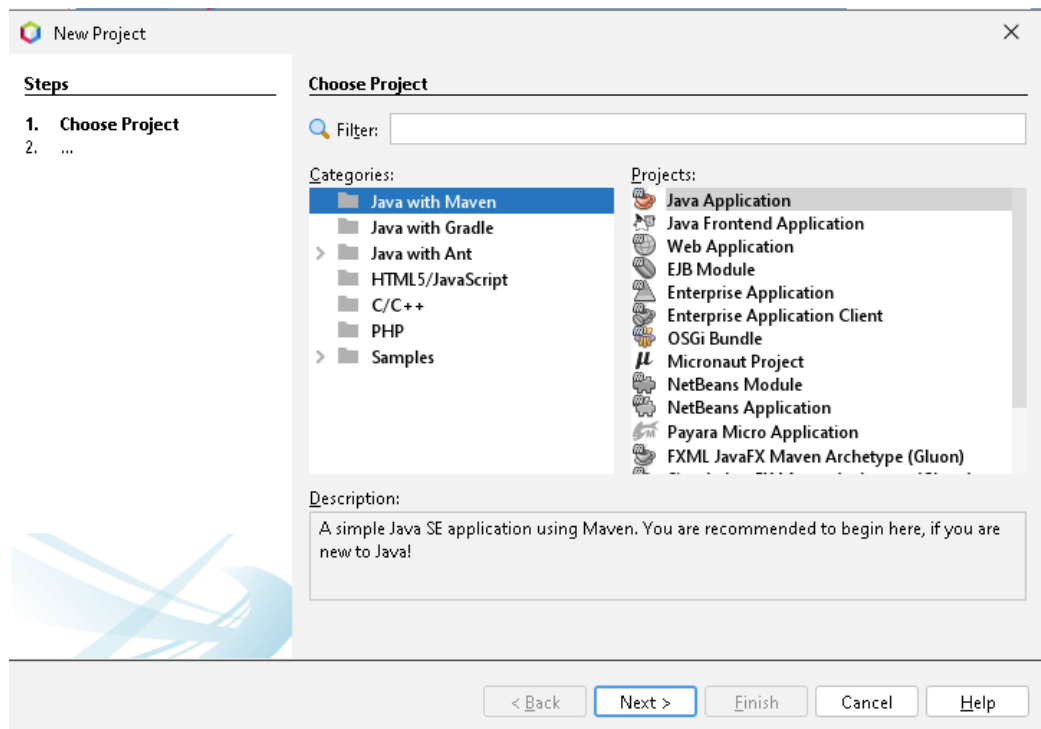
Ingresamos a NetBeans, nos aparece la siguiente pantalla, el entorno del Apache NetBeans IDE 19.



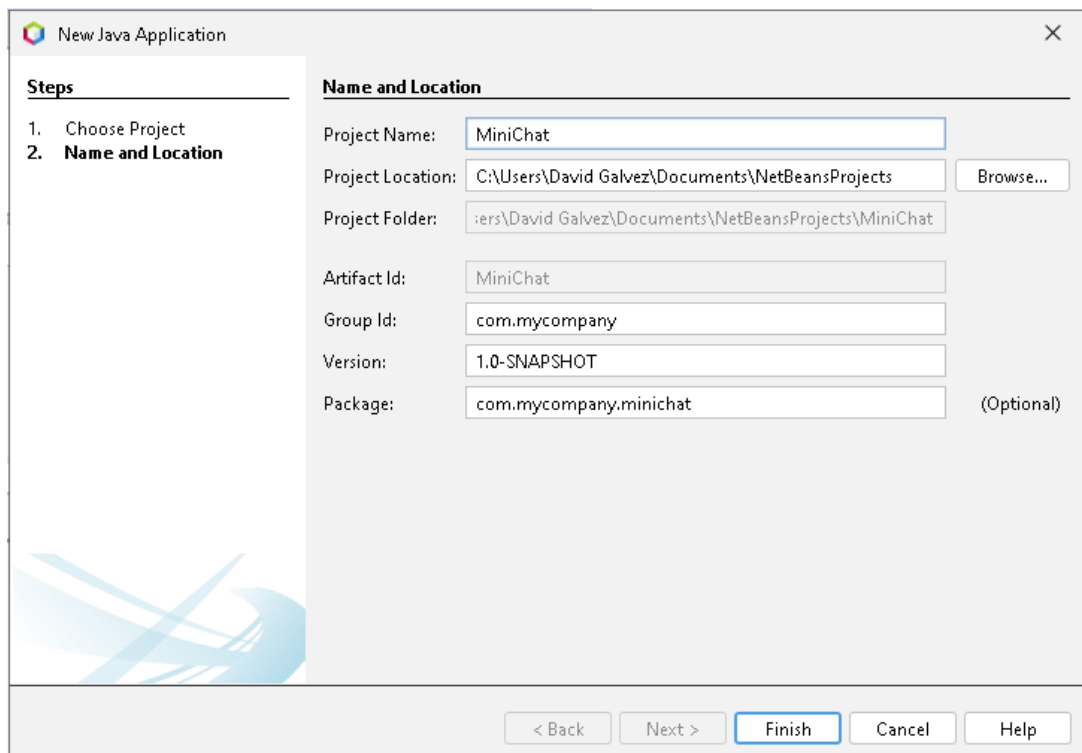
Para crear un nuevo proyecto, elegimos opción file, luego clic en NewProject.



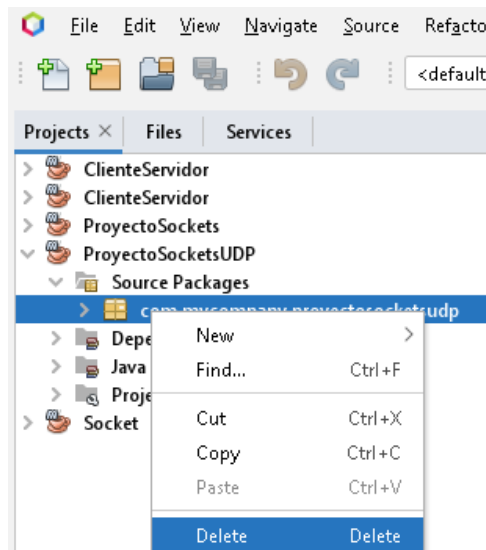
Elegimos en Categories la opción Java with Maven, luego en Projects elegimos Java Application, clic en el boton Next.



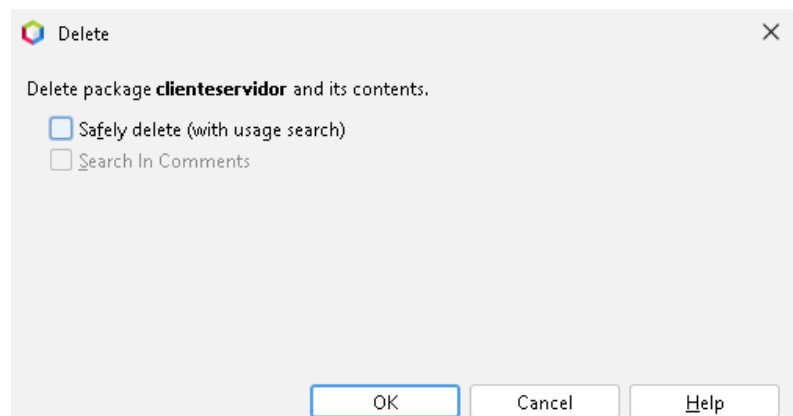
Colocamos un nombre al proyecto MiniChat, elegimos una carpeta en la cual colocar todo el proyecto en ProjectLocation, luego clic en botón Finish.



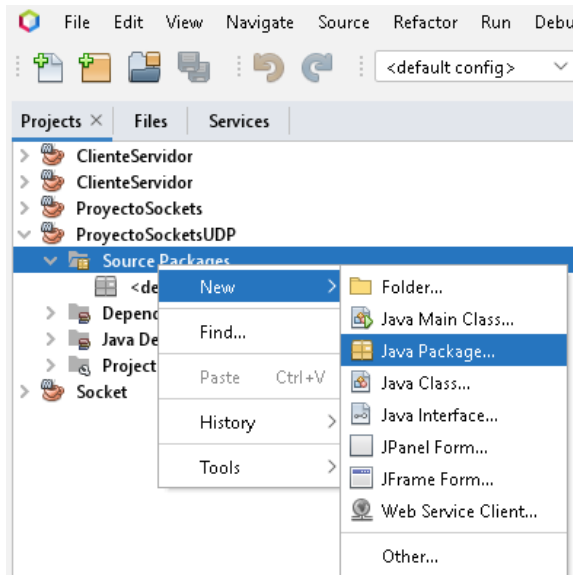
Eliminamos el paquete que se encuentra predeterminada, clic derecho en dicho paquete y luego clic en Delete.



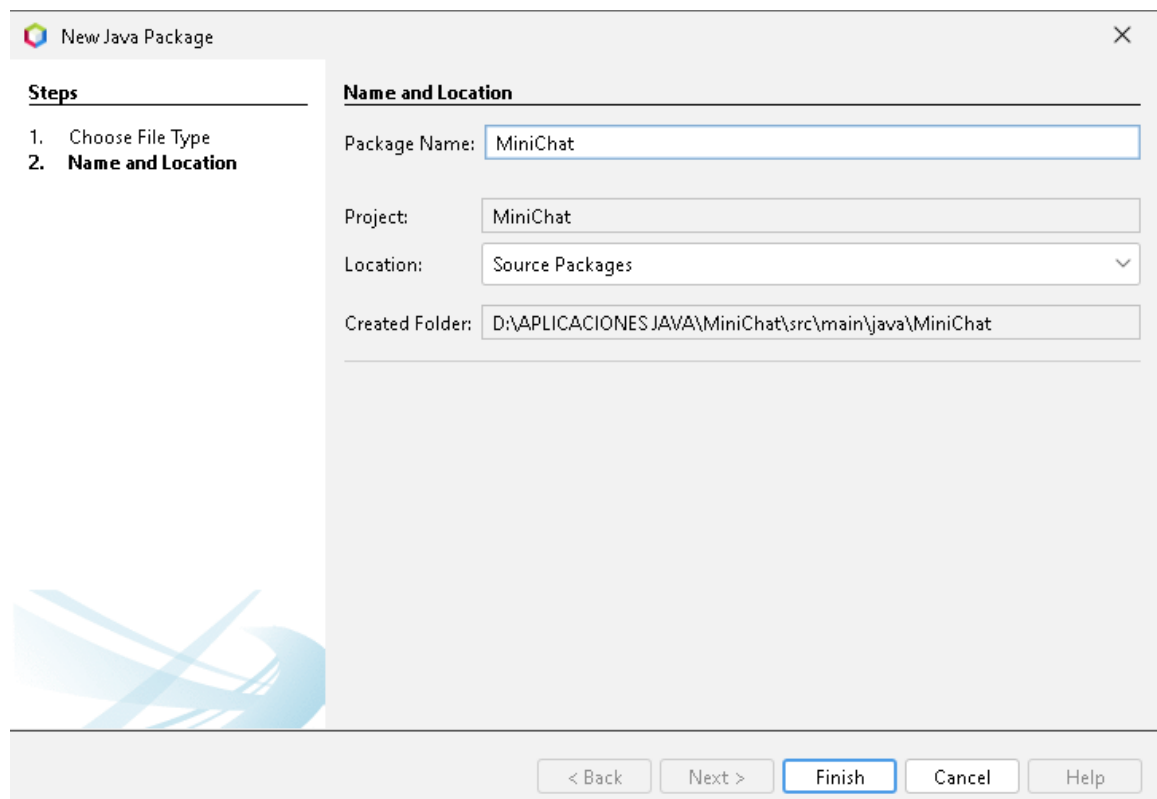
En la ventana Delete clic en el botón OK.



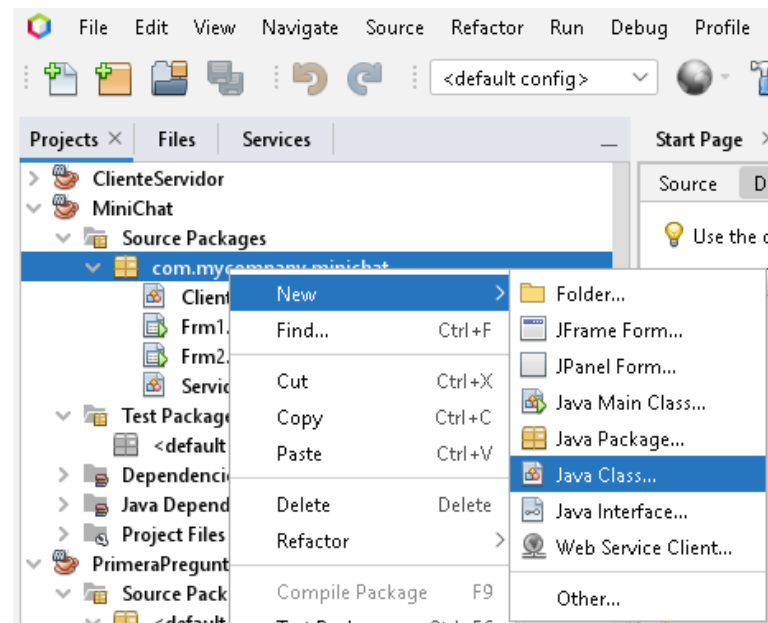
Creamos nuevos paquetes, clic derecho en Source Packages, clic en New, luego en Java Package.



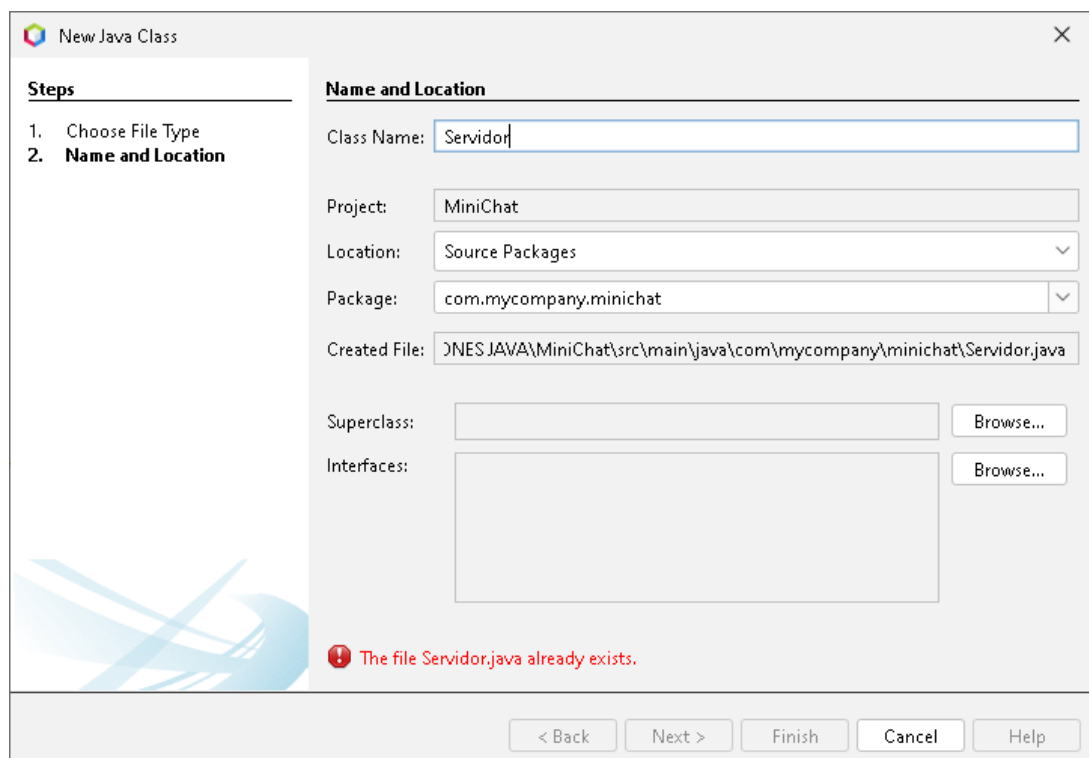
Colocamos de nombre MiniChat, clic en el botón Finish.



En el paquete MiniChat, clic derecho en New, luego la opción Java Class.



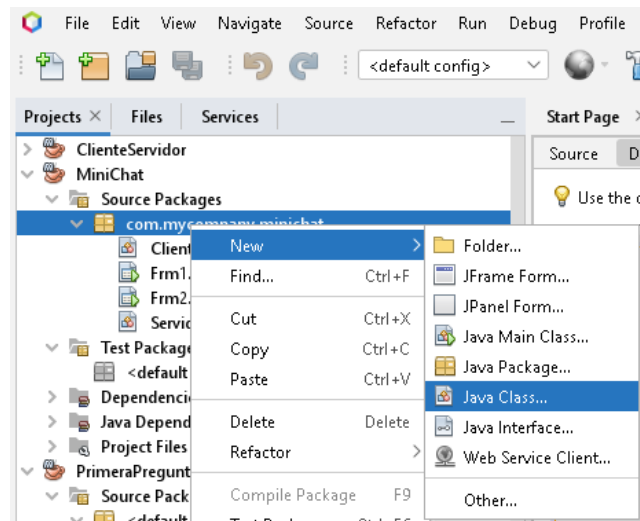
Colocamos el nombre Servidor, clic en el botón Finish.



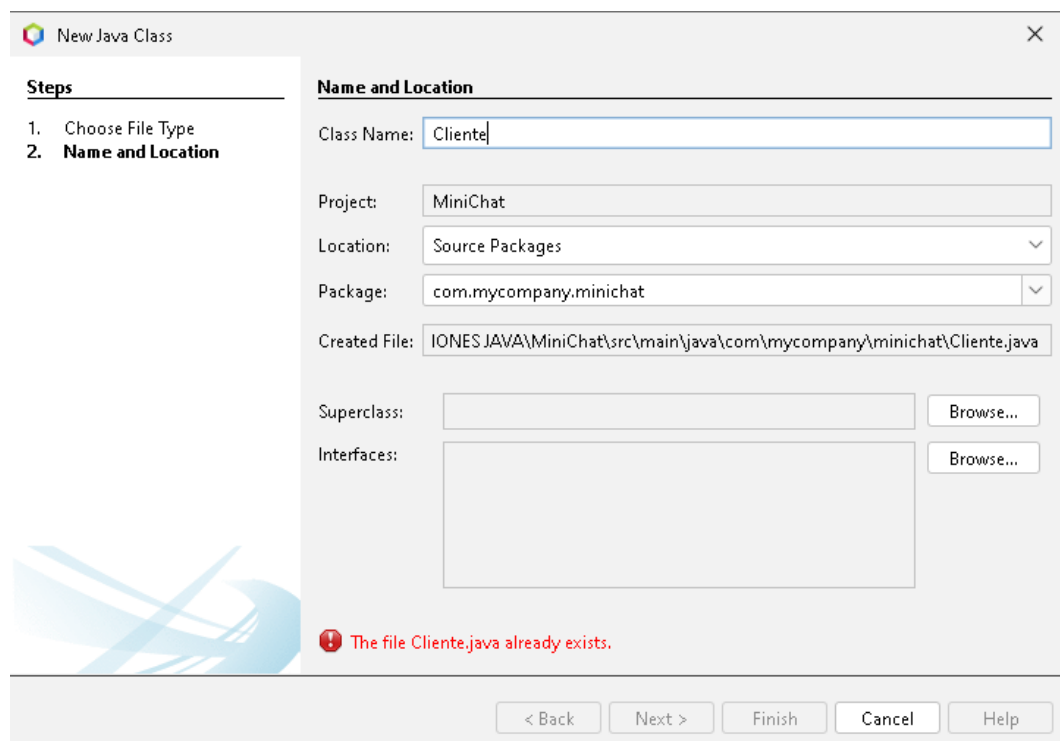
Colocar el siguiente código:

```
1 package com.mycompany.minichat;
2
3 import java.io.DataInputStream;
4 import java.io.DataOutputStream;
5 import java.io.IOException;
6 import java.net.ServerSocket;
7 import java.net.Socket;
8 import java.util.Observable;
9 import java.util.logging.Level;
10 import java.util.logging.Logger;
11
12 public class Servidor extends Observable implements Runnable {
13     private int puerto;
14
15     public Servidor (int puerto){
16         this.puerto=puerto;
17     }
18
19
20     @Override
21     public void run() {
22
23         ServerSocket servidor=null;
24         Socket sc=null;
25         DataInputStream in;
26         DataOutputStream out;
27
28         try{
29             servidor= new ServerSocket(puerto);
30             System.out.println("Servidor Iniciado");
31
32
33             while(true) {
34                 sc=servidor.accept();
35                 System.out.println("Cliente Conectado");
36                 in=new DataInputStream(sc.getInputStream());
37
38                 String mensaje=in.readUTF();
39                 System.out.println(mensaje);
40
41                 this.setChanged();
42                 this.notifyObservers(mensaje);
43                 this.clearChanged();
44
45                 sc.close();
46                 System.out.println("Cliente está desconectado");
47             }
48
49         } catch(IOException ex) {
50             Logger.getLogger(Servidor.class.getName()).log(Level.SEVERE, null, ex);
51         }
52     }
53 }
54
55 }
56
```

En el paquete Socket, clic derecho en New, luego la opción Java Class.



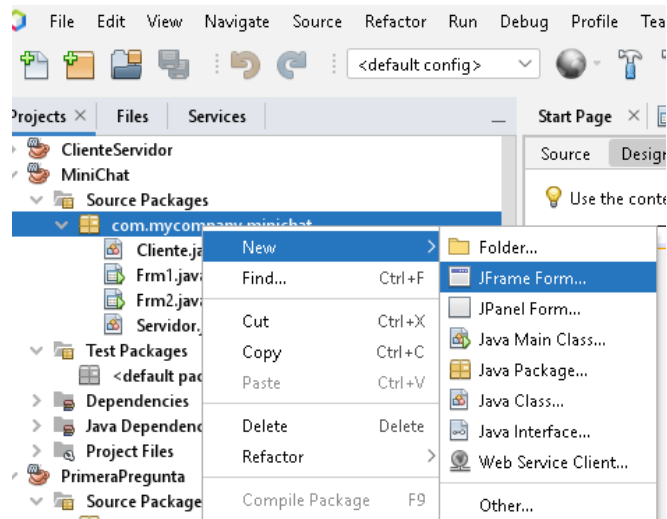
Colocamos el nombre Cliente, clic en el botón Finish.



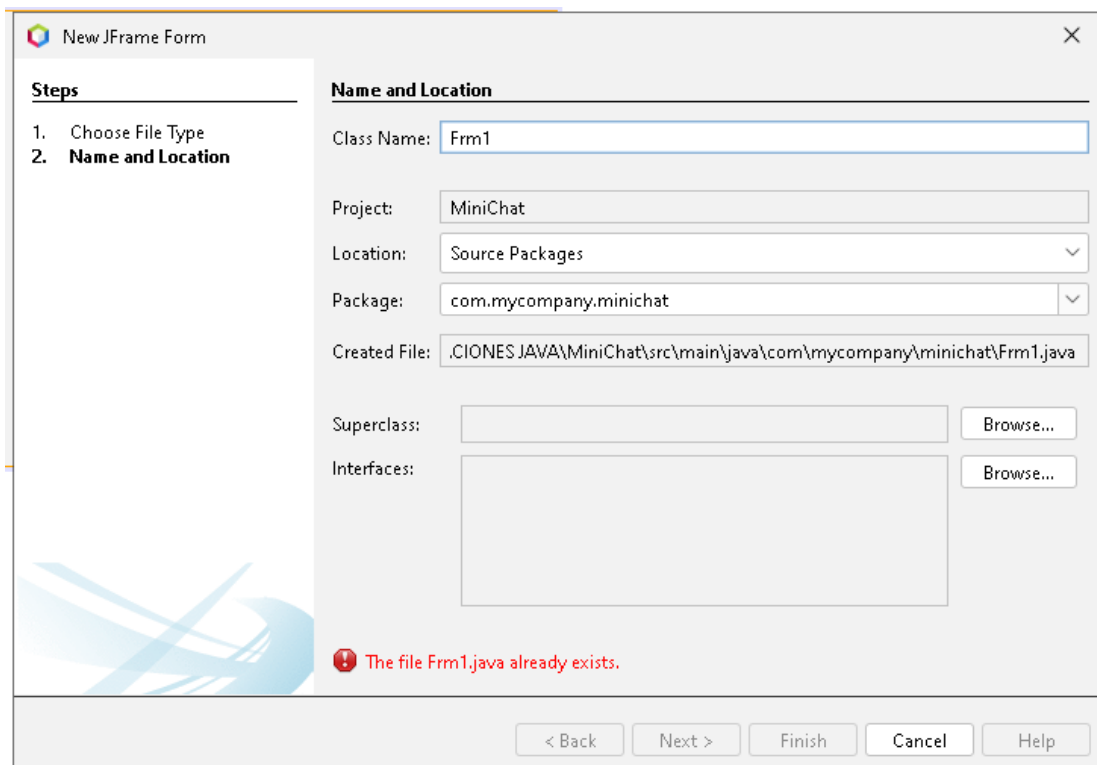
Colocar el siguiente código:

```
1 package com.mycompany.minichat;
2
3 import java.io.DataInputStream;
4 import java.io.DataOutputStream;
5 import java.io.IOException;
6 import java.net.Socket;
7 import java.util.logging.Level;
8 import java.util.logging.Logger;
9
10 public class Cliente implements Runnable {
11     private int puerto;
12     private String mensaje;
13
14     public Cliente (int puerto, String mensaje)
15     {
16         this.puerto=puerto;
17         this.mensaje=mensaje;
18     }
19
20     @Override
21     public void run() {
22         final String HOST="127.0.0.1";
23         DataInputStream in;
24         DataOutputStream out;
25
26         try{
27             Socket sc=new Socket (HOST,puerto);
28             out=new DataOutputStream(sc.getOutputStream());
29             out.writeUTF(mensaje);
30             sc.close();
31         }catch(IOException ex){
32             Logger.getLogger(Servidor.class.getName()).log(Level.SEVERE,null,ex);
33         }
34
35         throw new UnsupportedOperationException("Aún, No soportado");
36
37     }
38 }
```

En el paquete MiniChat, clic derecho en New, luego la opción JFrame Form



Colocamos el nombre Frm1, clic en el botón Finish.



En el formulario Frm1 colocamos 3 elementos: un TextField, un TextArea y un Button

Diagrama de un formulario Frm1. El formulario contiene tres elementos: un TextField (campo de texto), un TextArea (área de texto) y un Button (botón). El TextField y el TextArea están apilados verticalmente a la izquierda, y el Button está a la derecha del TextArea. El botón contiene el texto "Enviar".

Colocar el siguiente código:

```
1 package com.mycompany.minichat;
2
3 import java.util.Observable;
4 import java.util.Observer;
5
6 public class Frm1 extends javax.swing.JFrame implements Observer{
7
8     public Frm1() {
9         initComponents();
10        this.getRootPane().setDefaultButton(this.jButton1);
11        Servidor s=new Servidor(5000);
12        s.addObserver(this);
13
14        Thread t = new Thread (s);
15        t.start();
16    }
17
18    @SuppressWarnings("unchecked")
19
20    Generated Code
21
22
23
24
25
26 private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
27     // TODO add your handling code here:
28 }
29
30
31 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
32     String mensaje="1: "+this.jTextField1.getText()+"\n";
33
34     this.jTextArea1.append(mensaje);
35
36     Cliente c=new Cliente (6000,mensaje);
37     Thread t = new Thread(c);
38     t.start();
39 }
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
```

```

88
89
90
91
92
93
94
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135

```

```

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)

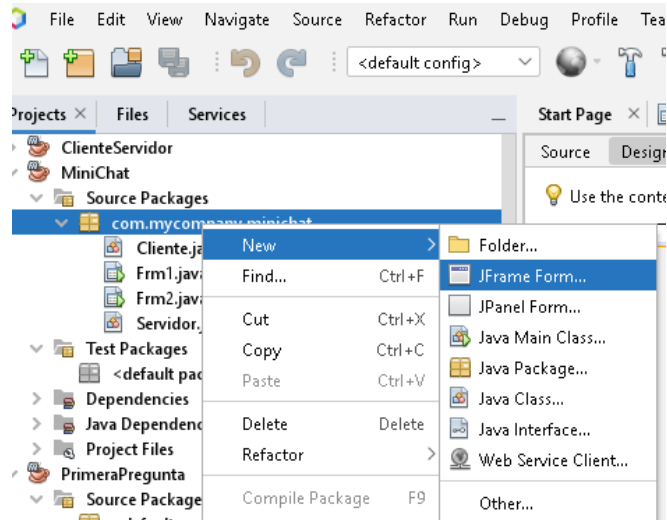
    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new Frm1().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTextArea jTextArea1;
private javax.swing.JTextField jTextField1;
// End of variables declaration

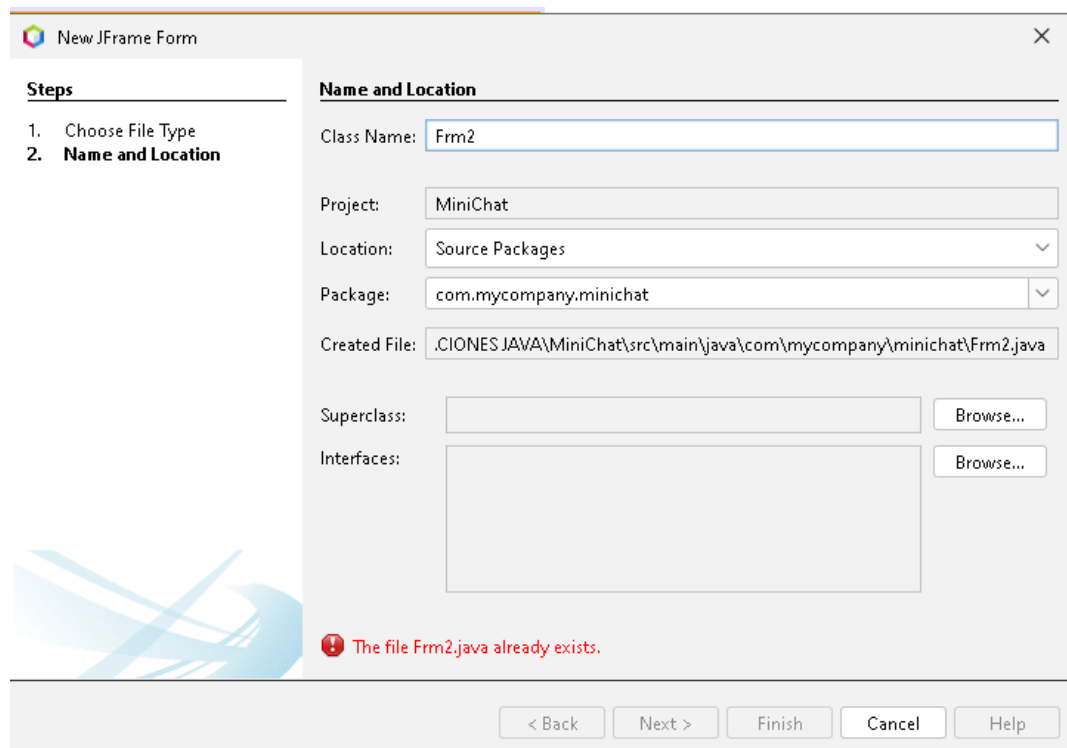
@Override
public void update(Observable o, Object arg) {
    this.jTextArea1.append((String) arg);
}

```

En el paquete MiniChat, clic derecho en New, luego la opción JFrame Form



Colocamos el nombre Frm2, clic en el botón Finish.



En el formulario Frm2 colocamos 3 elementos: un TextField, un TextArea y un Button

Diagrama de un formulario Frm2. El formulario contiene tres elementos: un TextField (campo de texto), un TextArea (área de texto) y un Button (botón). El TextField y el TextArea están apilados verticalmente a la izquierda, y el Button está a la derecha del TextArea. El Button tiene el texto "Enviar".

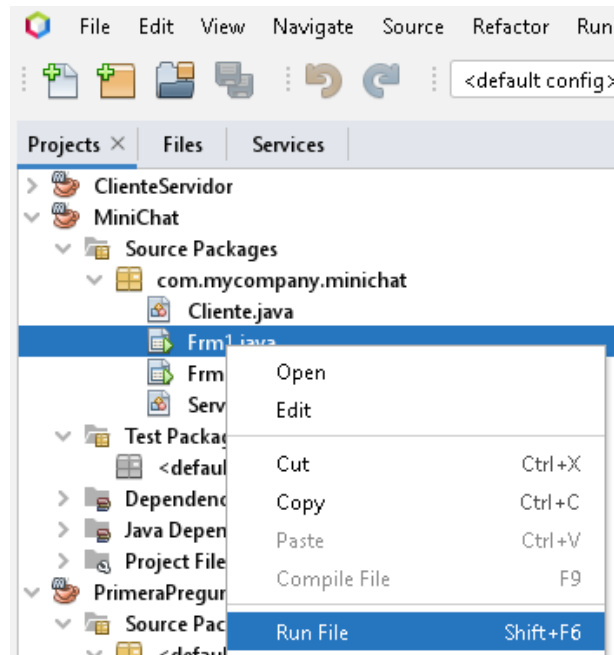
Colocar el siguiente código:

```

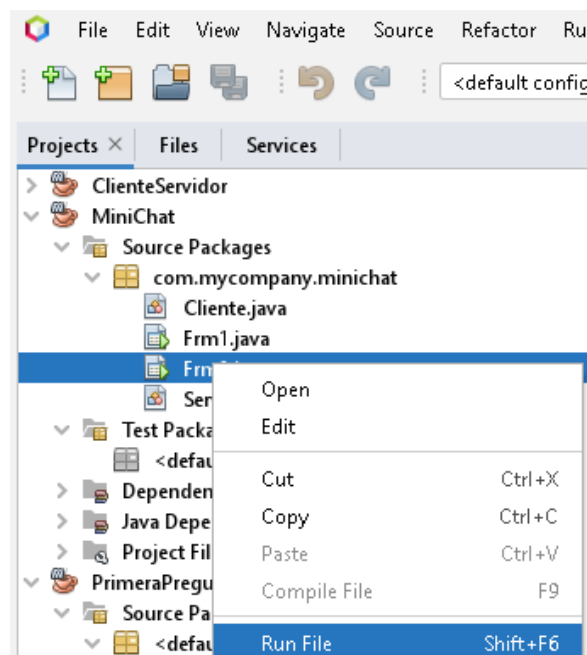
1
2 package com.mycompany.minichat;
3
4 import java.util.Observable;
5 import java.util.Observer;
6
7
8 public class Frm2 extends javax.swing.JFrame implements Observer {
9
10
11     public Frm2() {
12         initComponents();
13         this.getRootPane().setDefaultButton(this.jButton1);
14         Servidor s=new Servidor(6000);
15         s.addObserver(this);
16
17         Thread t =new Thread (s);
18         t.start();
19     }
20
21
22
23     @SuppressWarnings("unchecked")
24     Generated Code
25
26
27
28     private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
29         String mensaje = "2: "+this.jTextField1.getText() + "\n";
30         this.jTextArea1.append(mensaje);
31         Cliente c=new Cliente (5000,mensaje);
32         Thread t=new Thread(c);
33         t.start();
34     }
35
36
37
38     /**
39      * @param args the command line arguments
40      */
41
42     public static void main(String args[]) {
43         /* Set the Nimbus look and feel */
44         Look and feel setting code (optional)
45
46         /* Create and display the form */
47         java.awt.EventQueue.invokeLater(new Runnable() {
48             public void run() {
49                 new Frm2().setVisible(true);
50             }
51         });
52     }
53
54     // Variables declaration - do not modify
55     private javax.swing.JButton jButton1;
56     private javax.swing.JScrollPane jScrollPane1;
57     private javax.swing.JTextArea jTextArea1;
58     private javax.swing.JTextField jTextField1;
59     // End of variables declaration
60
61
62     @Override
63     public void update(Observable o, Object arg) {
64         this.jTextArea1.append((String) arg);
65     }
66 }

```

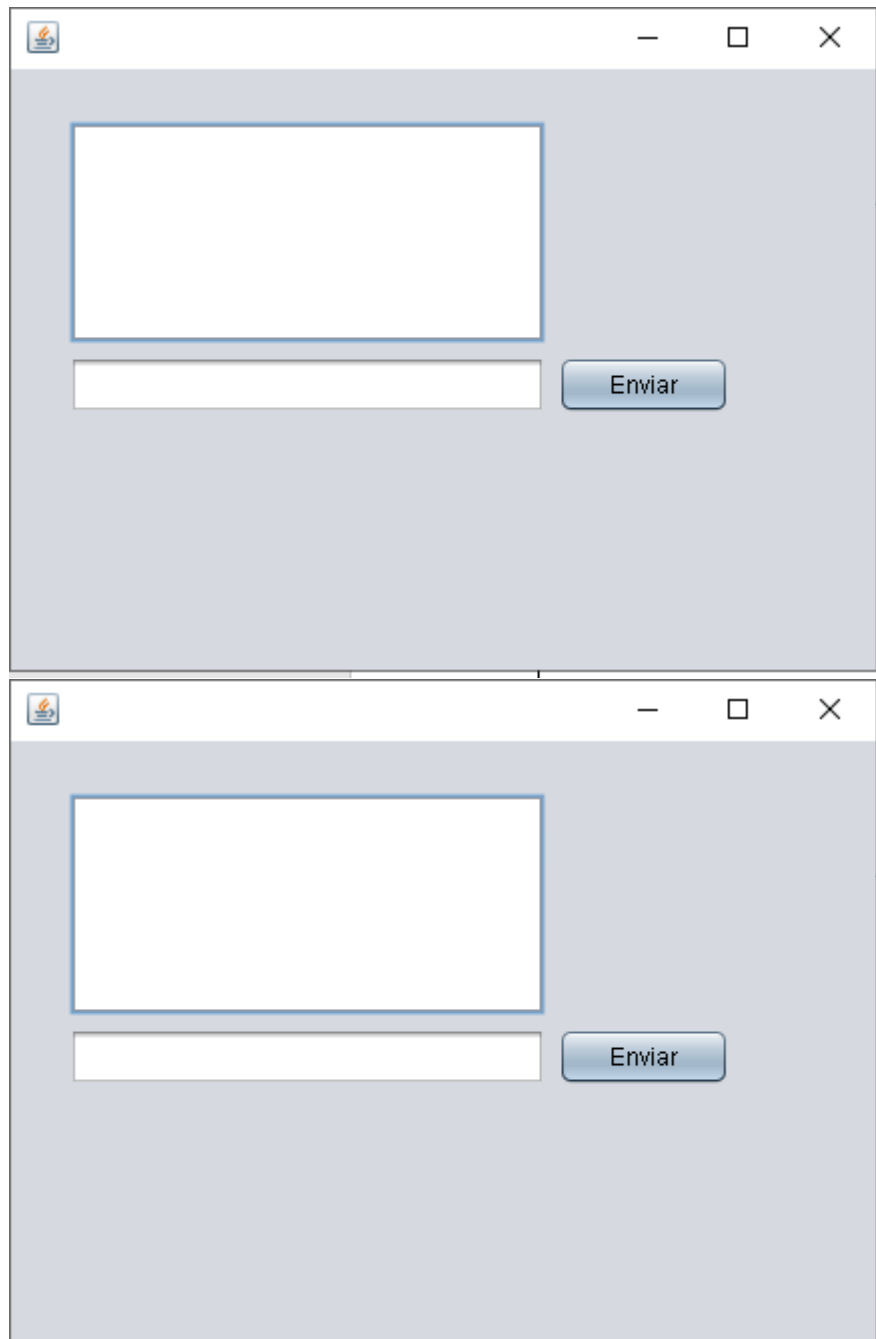
Ejecutamos el formulario frm1, vamos al formulario frm1, clic derecho, luego en Run File.



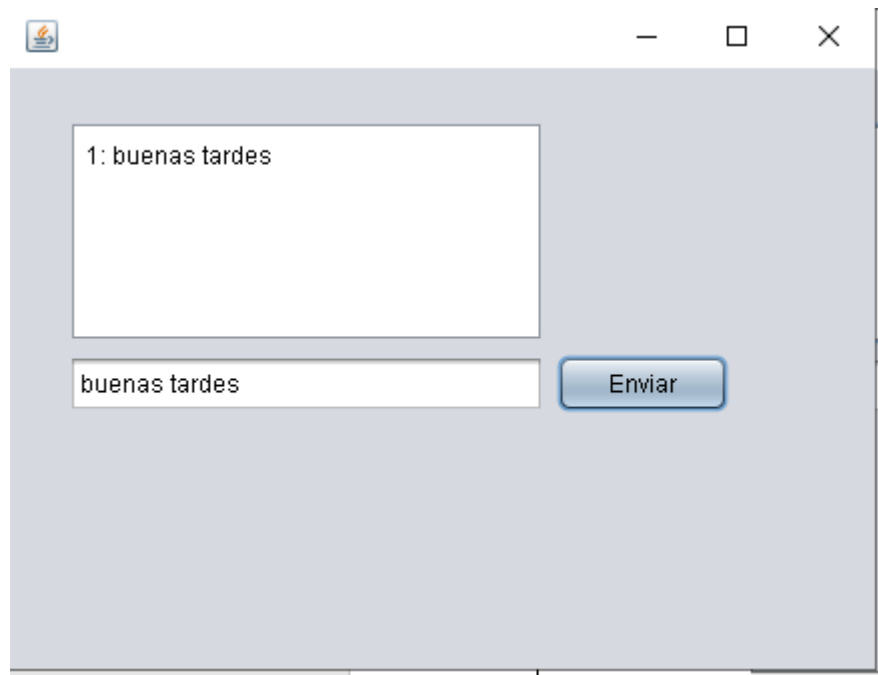
Ejecutamos el formulario frm2, vamos al formulario frm2, clic derecho, luego en Run File.



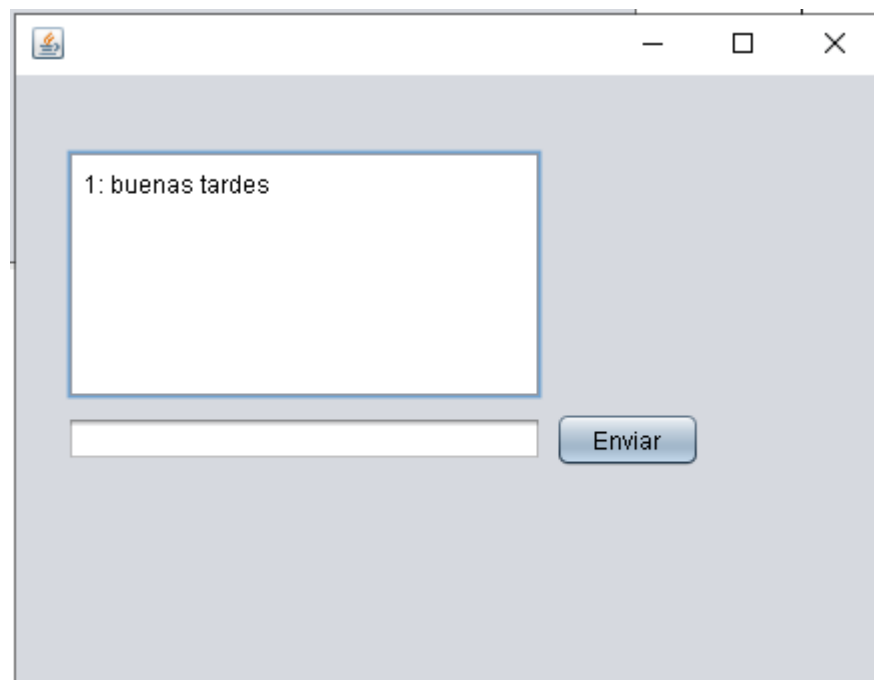
Verificamos que aparezca el mensaje “Servidor Levantado”



Ingresamos texto en el primer formulario, luego colocar texto y presionar en envia. Verificar c
en el formulario 2 aparezca el mensaje.



A screenshot of a web application window. The window has a title bar with a small icon on the left and standard minimize, maximize, and close buttons on the right. The main content area has a light gray background. It contains a large white text input field with the text "1: buenas tardes". Below this field is a smaller white text input field containing the text "buenas tardes". To the right of the smaller input field is a blue button with the text "Enviar".



A second screenshot of the same web application window. The large text input field still contains "1: buenas tardes". The smaller text input field below it is now empty. The "Enviar" button remains to the right of the empty input field.

7. ENTREGABLES

1. Colocar títulos a ambos formularios donde al primer formulario indique “Chat origen” y el segundo formulario indique “Chat destino”
2. Colocar un nuevo formulario con los elementos indicados y que los tres formularios puedan enviarse mensajes.

8. FUENTES DE INFORMACIÓN COMPLEMENTARIA

- López Fuentes, Francisco de Asís (2015). Sistemas Distribuidos. Editorial Macro. 1ra. edición.
- Pérez, Jesús & De Miguel, Pedro & Pérez, Fernando & García, Félix, Claudio Alejandro (2001). Sistemas Operativos Una Visión Aplicada. 1ra edición.