

FOTO GUIA SOBRE DESARROLLO DE APLICACIONES DISTRIBUIDAS CLIENTE SERVIDOR SOBRE TCP/IP USANDO TECNOLOGÍA RMI (REMOTE METHOD INVOCATION) DE JAVA Y UNA LIBRERÍA DE TERCEROS LLAMADA LIPE RMI QUE FACILITA EL PROCESO DE IMPLEMENTACIÓN DE RMI

- Crear una clase para transportar los datos del resultado del lmc 4 Crear una interfaz para invocar los métodos de forma remota 55. Construir el archivo distribuible (librerías, dependencia o componente JAR) 71. Crear un nuevo proyecto de tipo Java Application 82. Agregar la librería (componente o dependencia) LipeRMI 93. Crear la clase Servidor 104. Crear la clase CalculoRmilmcImplem 124. Agregar la librería del del proyecto que creamos anteriormente 125. Escribir el código de las clases. 141. Crear un nuevo proyecto de tipo Java Application 162. Agregar la librería LipeRMI 173. Agregar la librería del proyecto creado inicialmente 184. Crear la clase VentanaPrincipal. 205. Diseñar el formulario y programar los eventos de la GUI. 21 Ejecutar la aplicación Servidor 30 Ejecutar la aplicación Cliente 30 Iniciar la conexión desde el cliente hacia el servidor 31 Ingresar datos, solicitar realizar el cálculo de forma remota y mostrar el resultado. 31

Como actividad académica los estudiantes de forma individual deben:

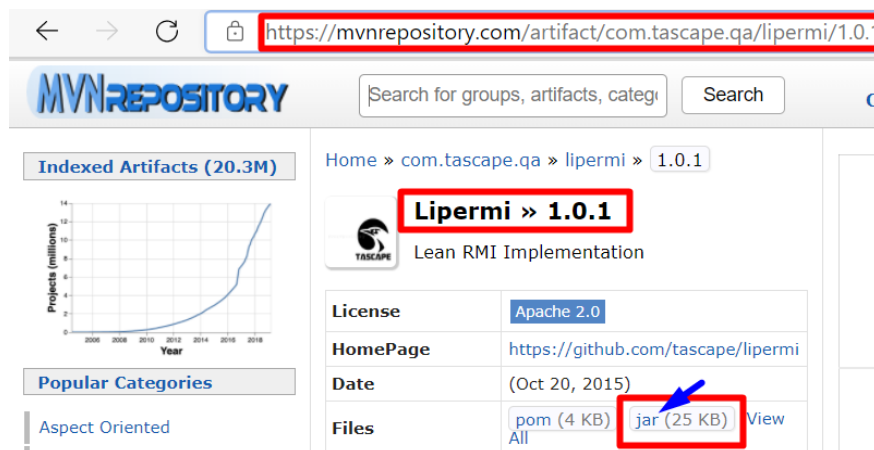
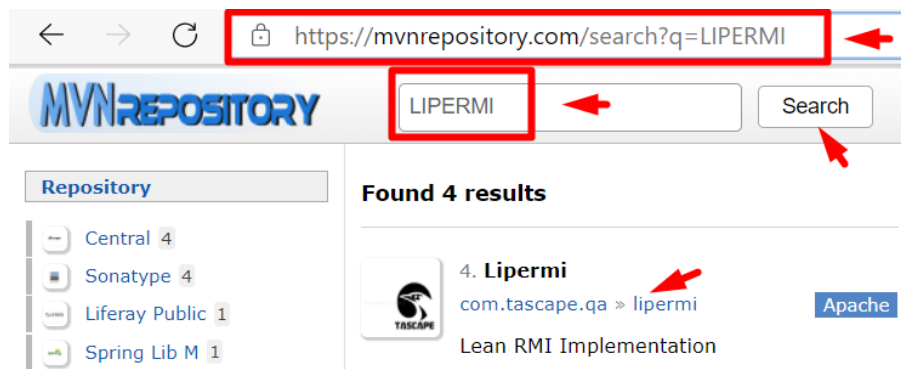
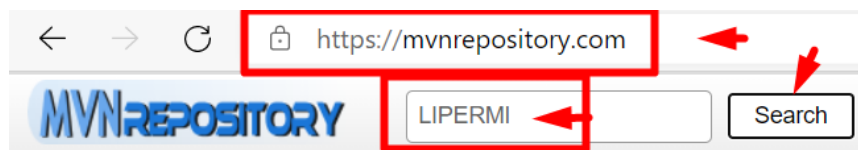
1. Realizar y documentar cada paso de esta Guía, entregando los fuentes de los 3 proyectos
2. Realizar el mismo ejemplo de la guía usando RMI Estándar de Java, sin usar la librería LipeRM, documentando cada paso del proceso y explicando las diferencias, similitudes con el uso de la LipeRmi.
3. Investigar si existen otras librerías (en Java, o entros lenguajes) que permitan desarrollar Sistemas Distribuidos usando RMI/RPC y entregar un ejemplo muy simple.

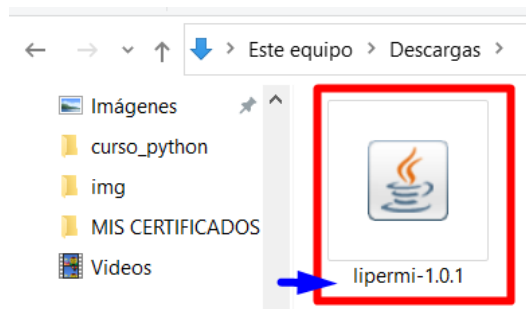


DESCARGAR, INSTALAR Y CONFIGURAR LAS HERRAMIENTAS NECESARIAS

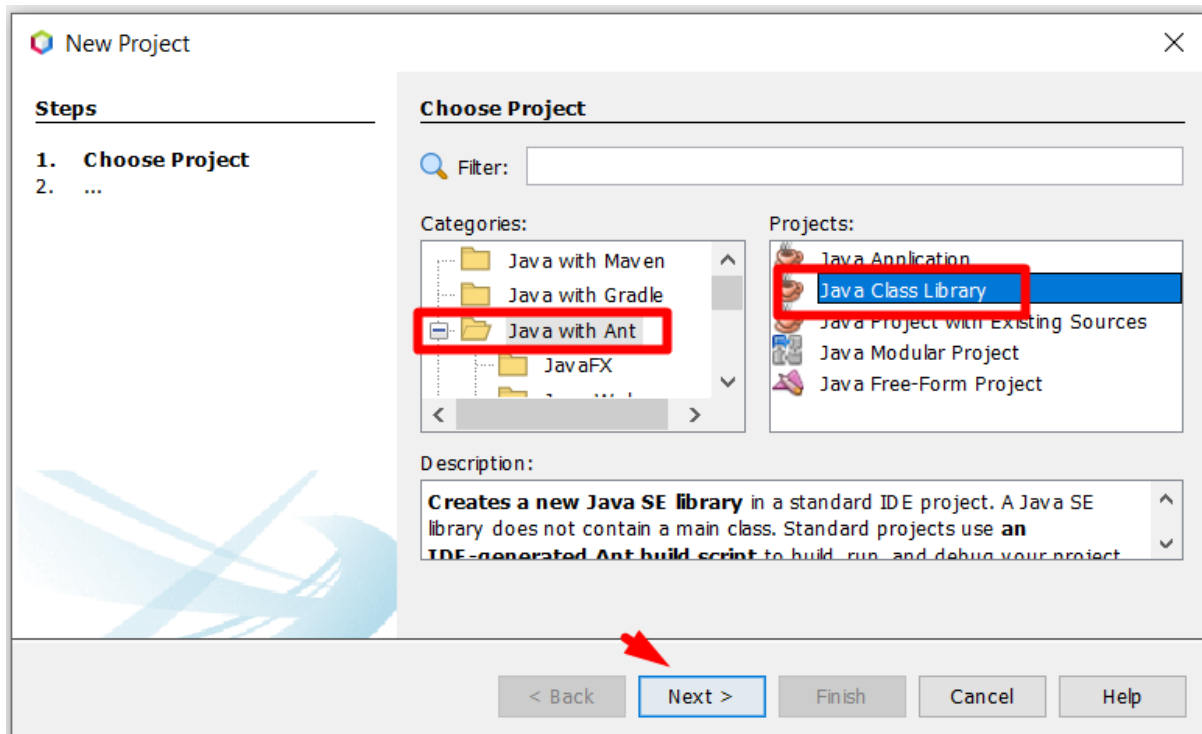
- a) Descargar, Instalar y configurar correctamente el JDK
 - i) <https://youtu.be/BoYRcKZbDb0>
- b) Descargar un IDE para Java, por ejemplo Netbeans IDE
 - i) <https://youtu.be/oT1cUI984zU>
- c) Descargar la librería LipeRMI <https://mvnrepository.com/>

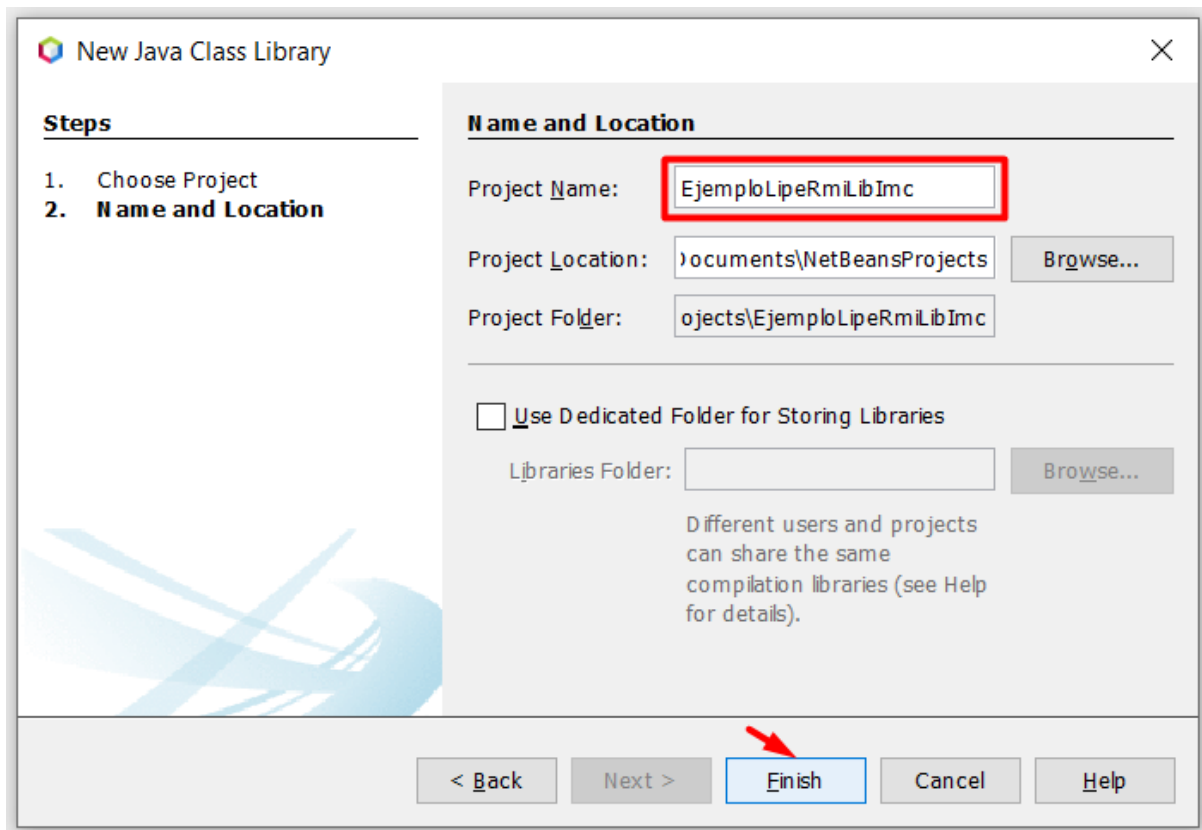
PASOS:



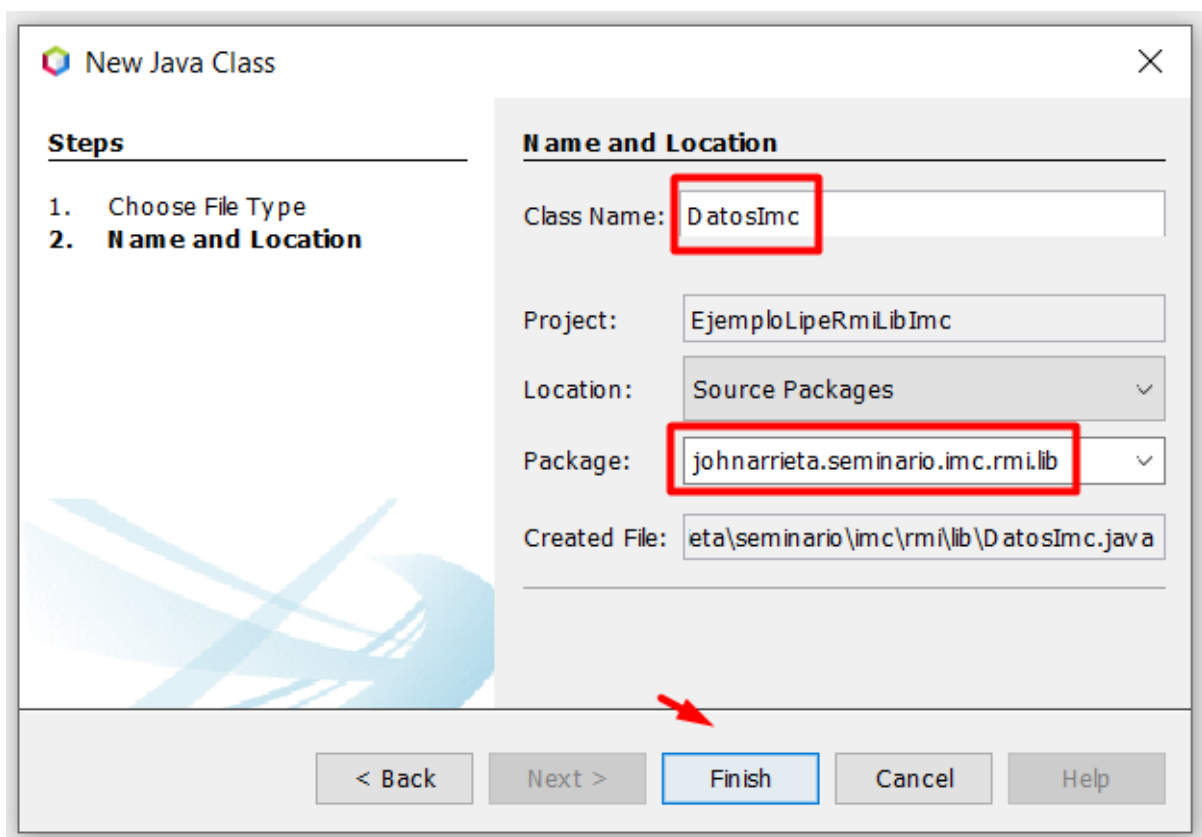


CREAR UN PROYECTO TIPO LIBRERÍA

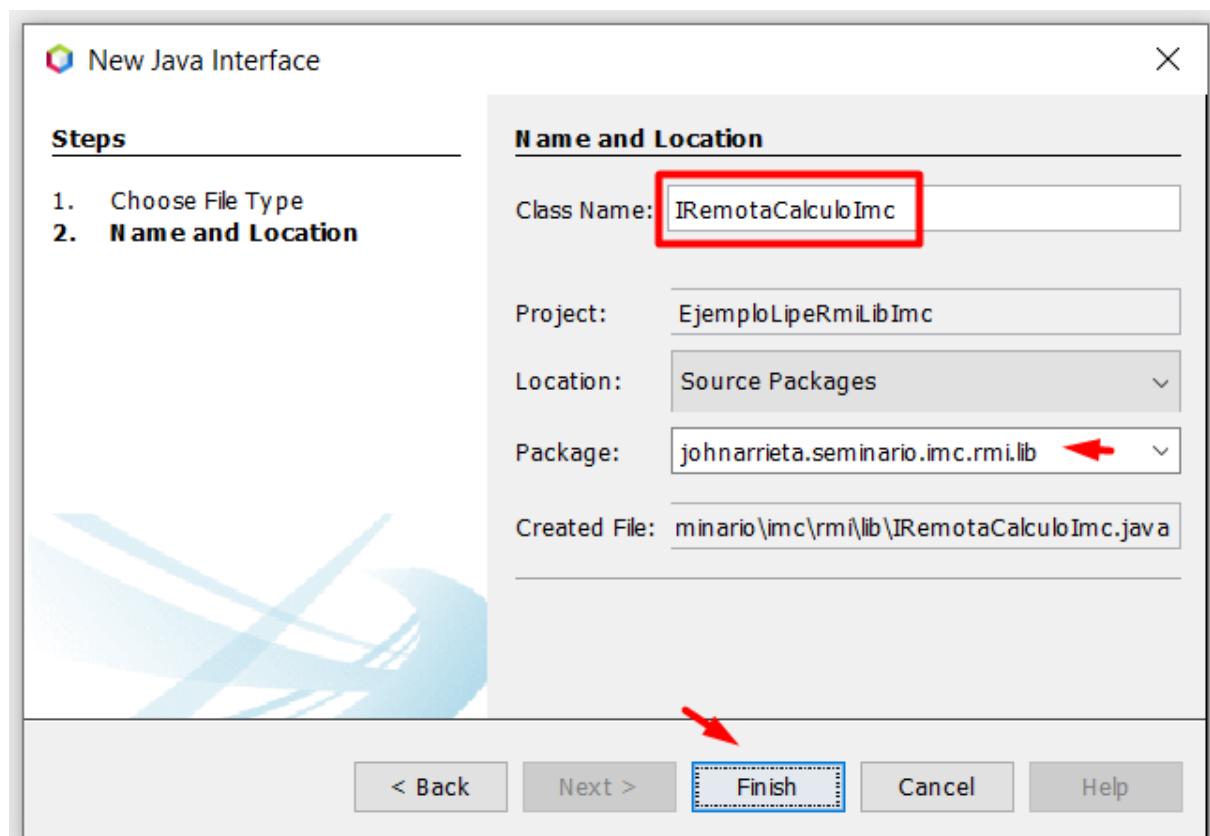
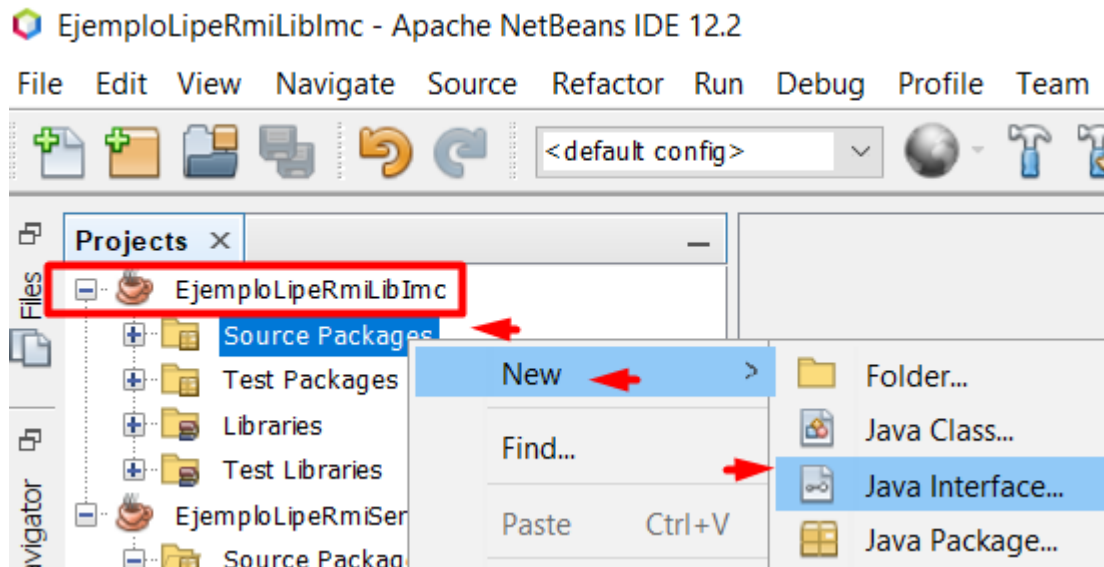


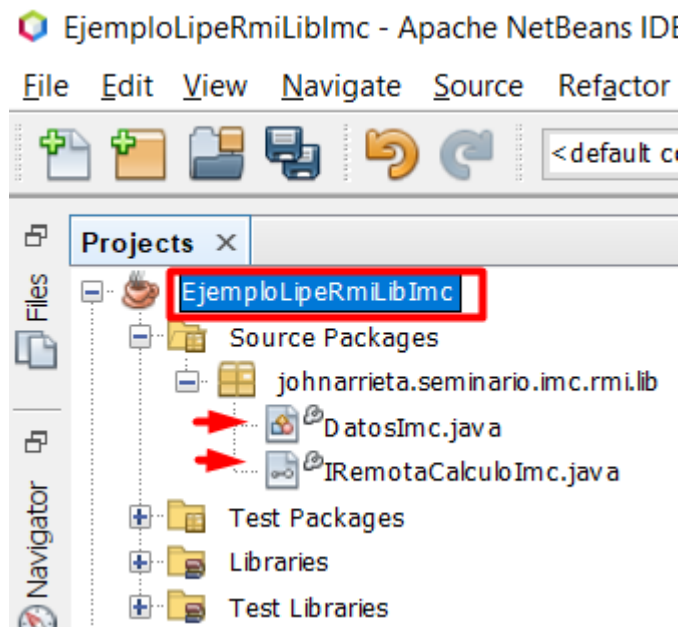


2) Crear una clase para transportar los datos del resultado del Imc



3) Crear una interfaz para invocar los métodos de forma remota





CÓDIGO DE LA CLASE DATOSIMC

```
package johnarrieta.seminario.imc.rmi.lib;

import java.io.Serializable;

/**
 *
 * @author JOHN CARLOS ARRIETA ARRIETA
 */
public class DatosImc implements Serializable{

    private float peso;
    private float altura;
    private float resultado;
    private String interpretación;

    public DatosImc() {
    }

    public DatosImc(float peso, float altura) {
        this.peso = peso;
        this.altura = altura;
    }

    public float getPeso() {
        return peso;
    }
}
```

```

    public void setPeso(float peso) {
        this.peso = peso;
    }

    public float getAltura() {
        return altura;
    }

    public void setAltura(float altura) {
        this.altura = altura;
    }

    public float getResultado() {
        return resultado;
    }

    public void setResultado(float resultado) {
        this.resultado = resultado;
    }

    public String getInterpretacion() {
        return interpretacion;
    }

    public void setInterpretacion(String interpretacion) {
        this.interpretacion = interpretacion;
    }
}

```

CÓDIGO DE LA INTERFAZ IREMOTACALCULOIMC

```

package johnarrieta.seminario.imc.rmi.lib;

/**
 *
 * @author JOHN CARLOS ARRIETA ARRIETA
 */
public interface IRemotaCalculoImc {

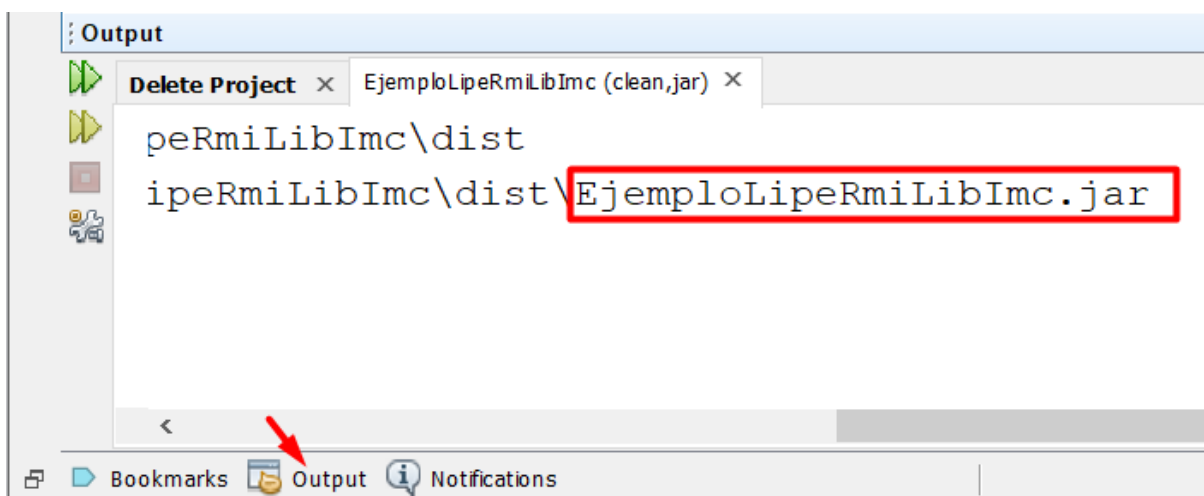
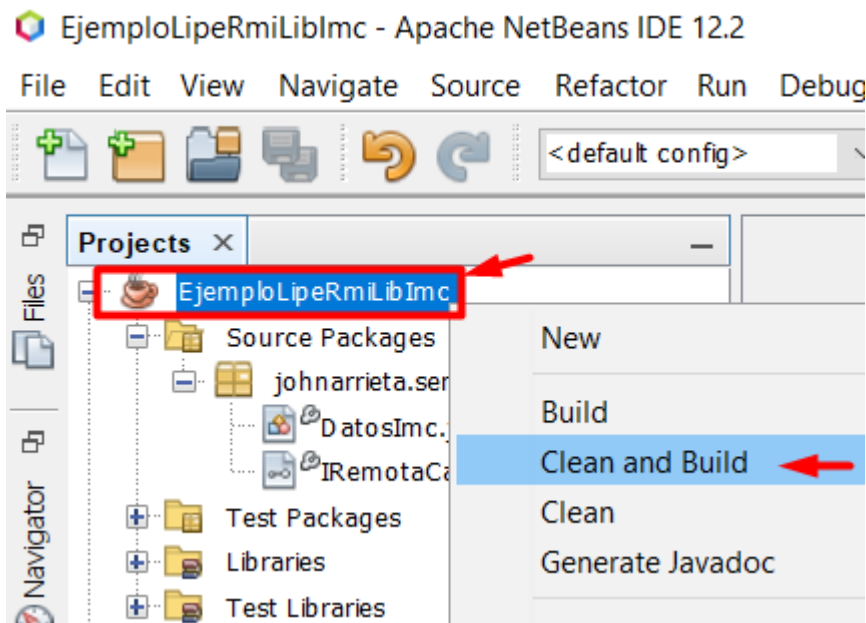
    public DatosImc calcularImc(DatosImc datos);
}

```

5. Construir el archivo distribuible (librerías, dependencia o componente JAR)



Reconocimiento – NoComercial – CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

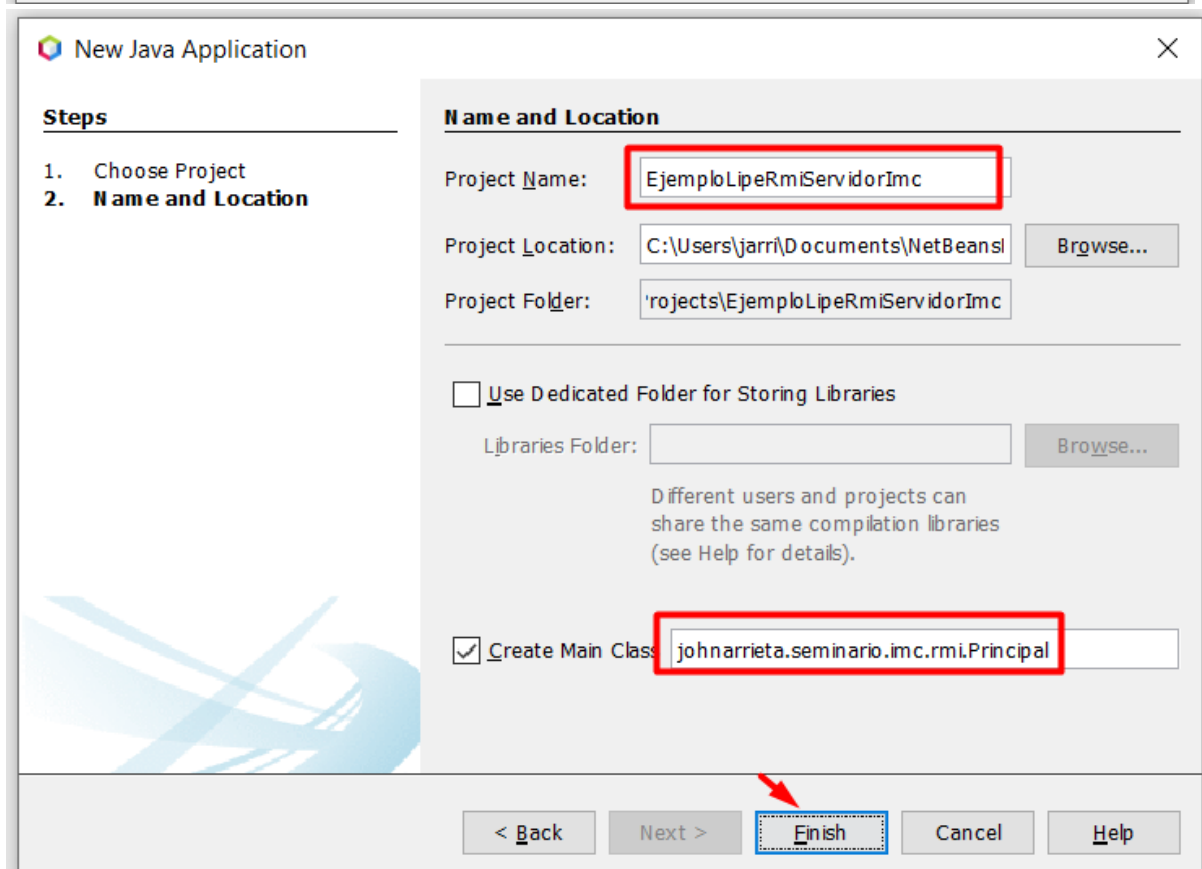
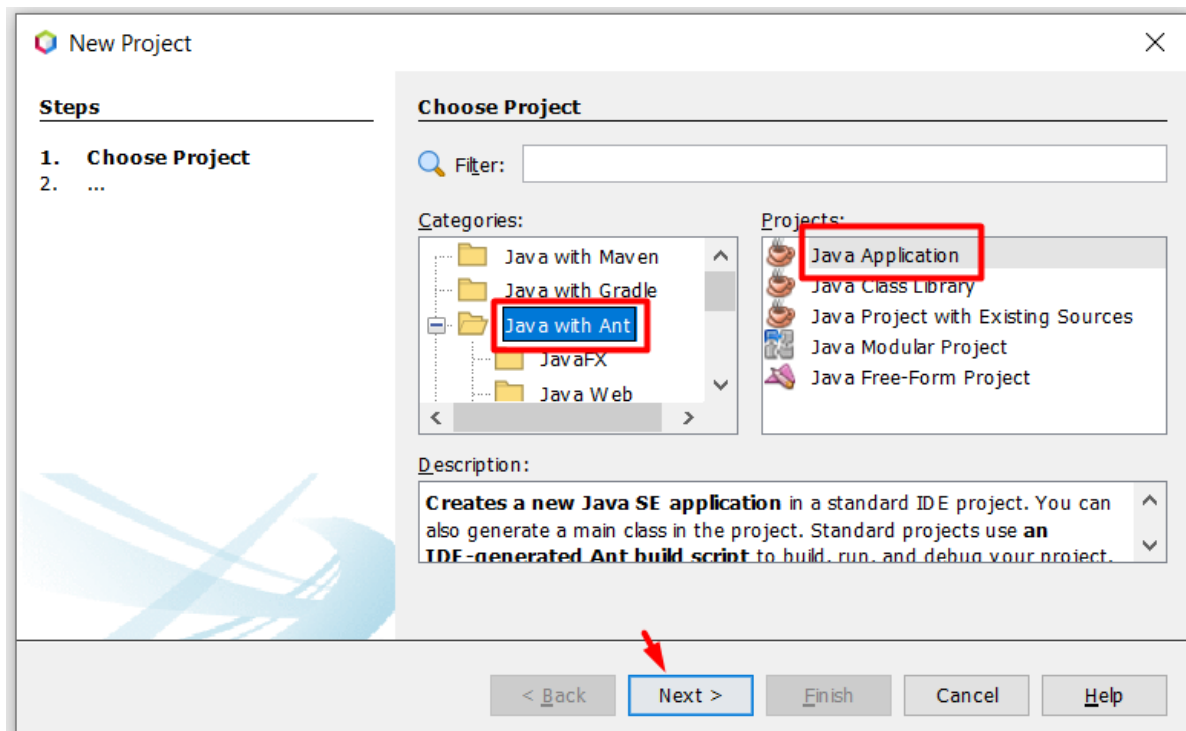


CREAR EL PROYECTO PARA EL SERVIDOR

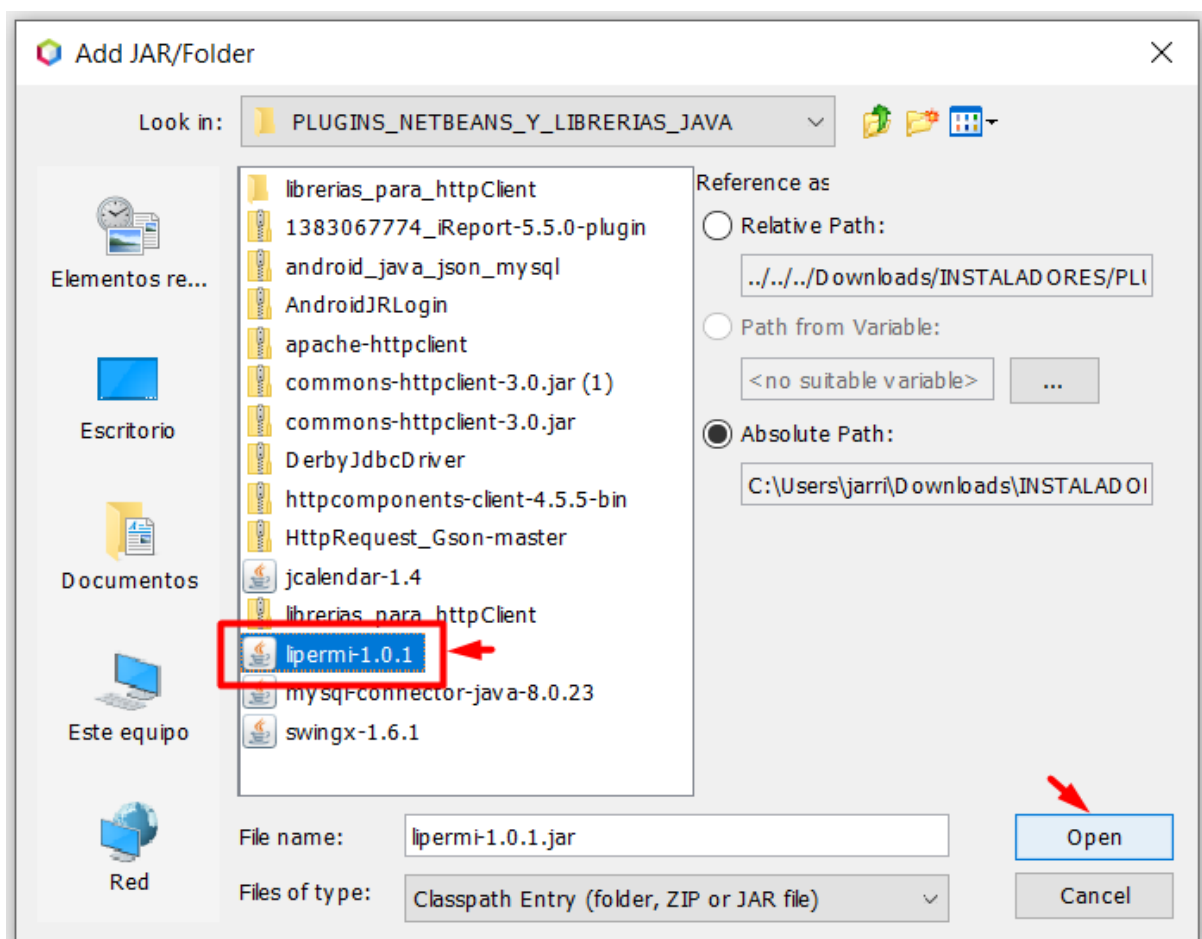
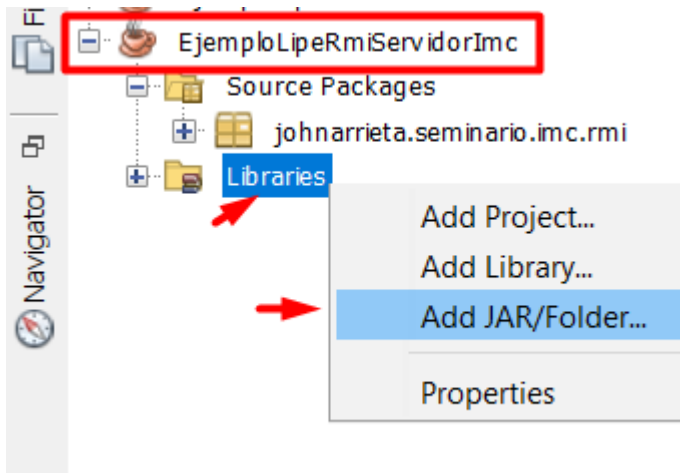
1. Crear un nuevo proyecto de tipo Java Application

Pasos:

1. Crear un nuevo proyecto en NB
2. Categoría Proyecto con Ant
3. Tipo Java Application
4. Nombre: EjemploLipeRmiServidorImc
5. Ubicación: Cualquier carpeta
6. Clase Principal: Tu_nombre.seminario.imc.rmi.Principal



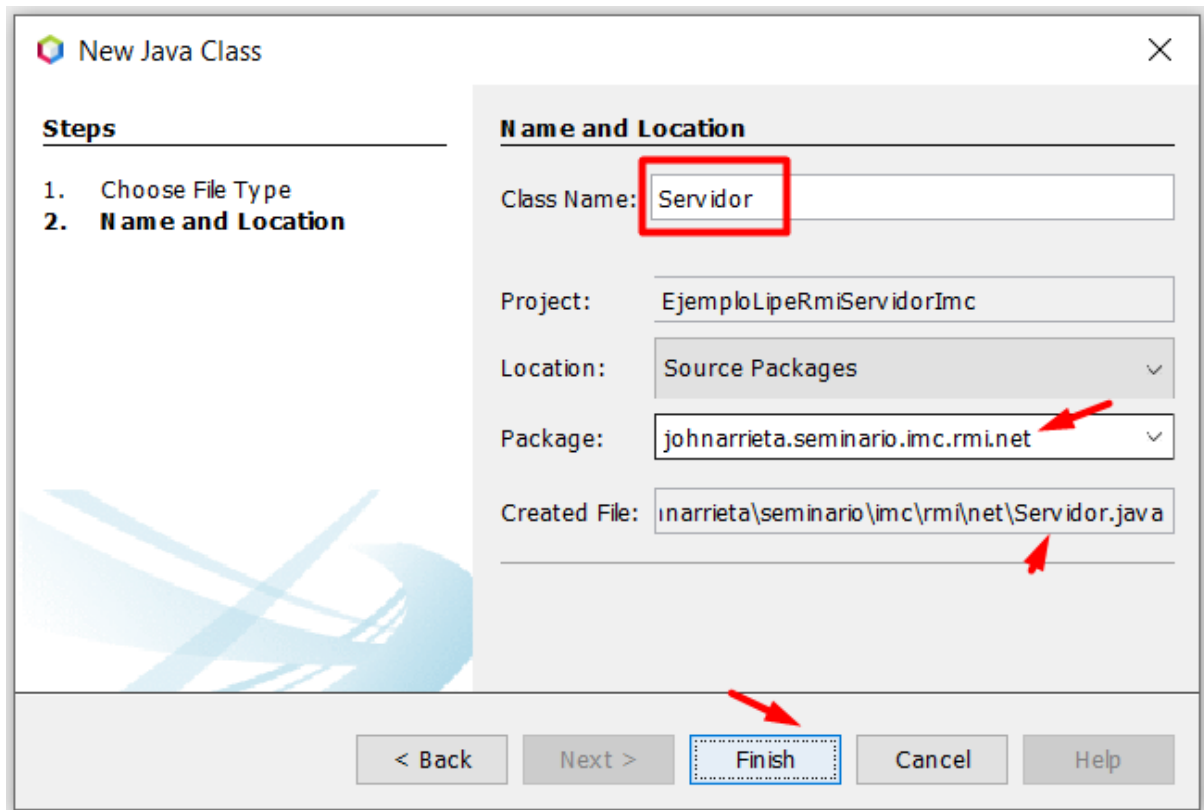
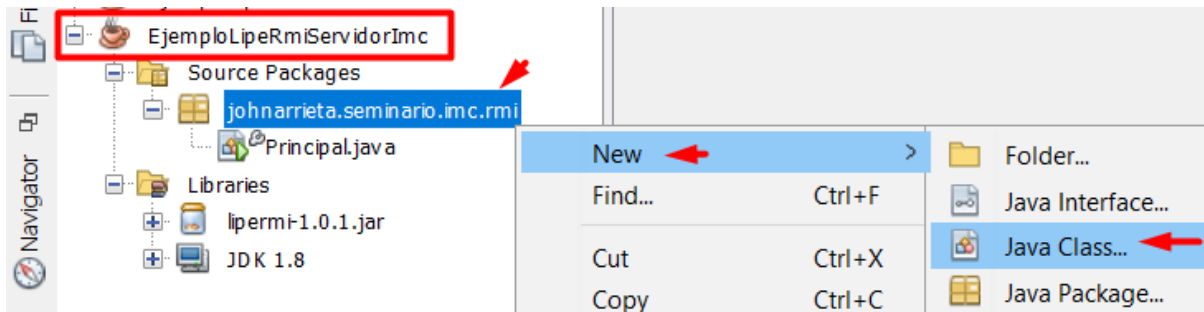
2. Agregar la librería (componente o dependencia) LipeRMI



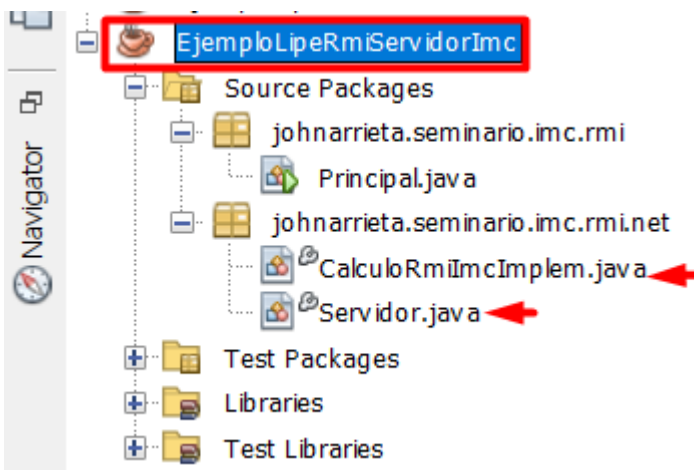
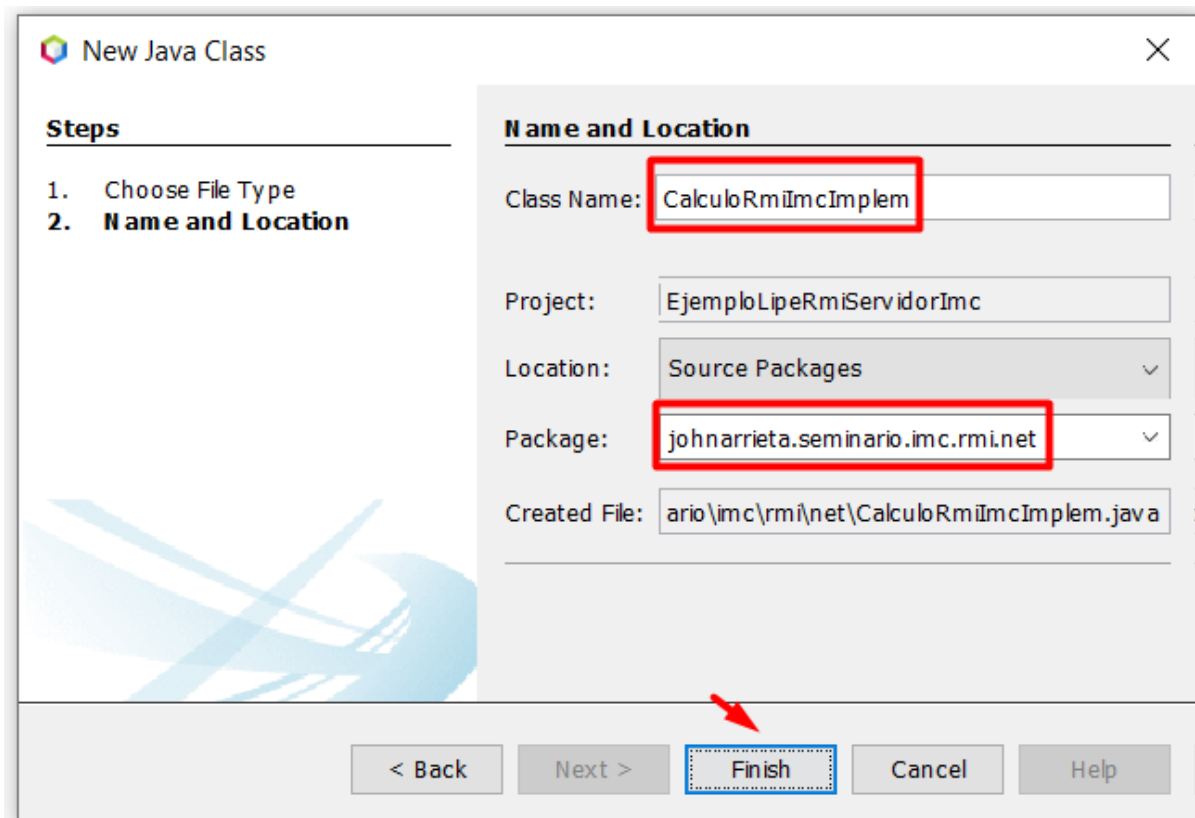
3. Crear la clase Servidor

GUIA DE DESARROLLO DE APLICACIONES DISTRIBUIDAS USANDO RMI

Autor: John Carlos Arrieta Arrieta



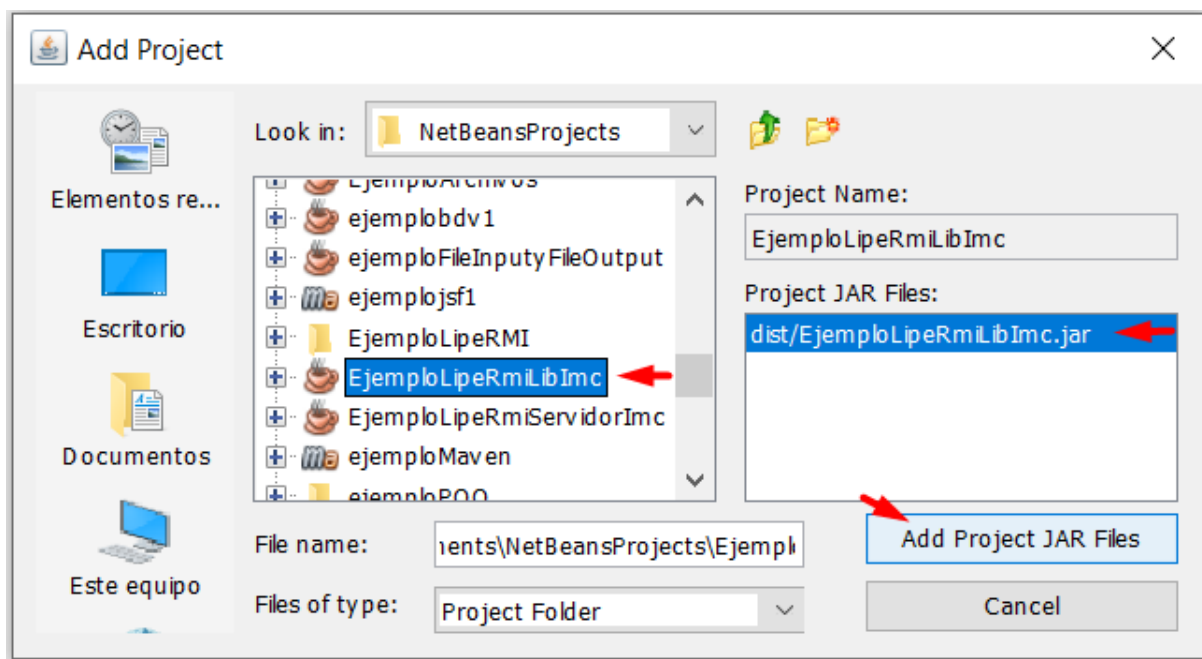
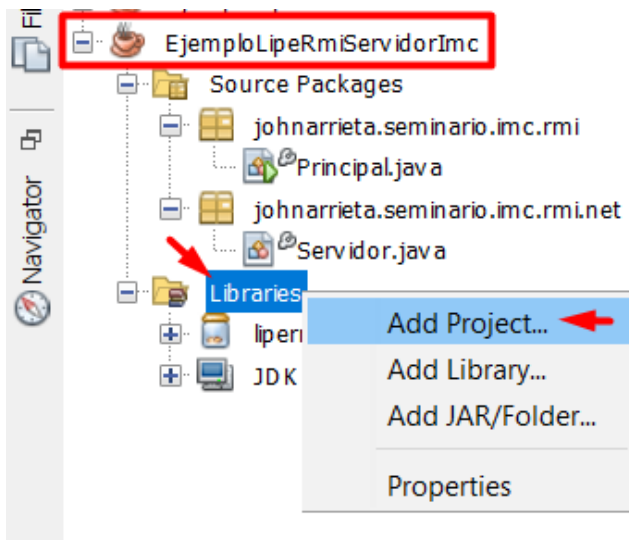
4. Crear la clase CalculoRmiImcImplem



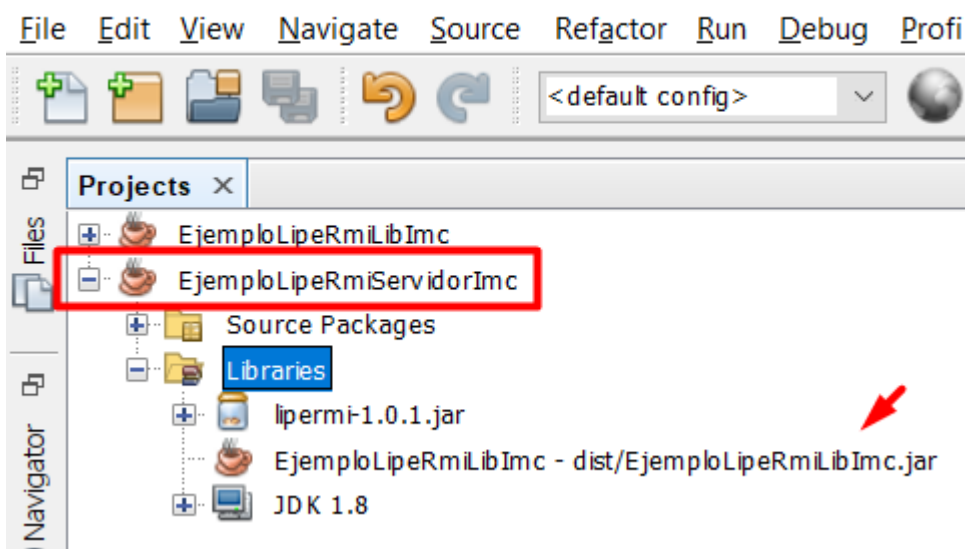
4. Agregar la librería del del proyecto que creamos anteriormente

GUIA DE DESARROLLO DE APLICACIONES DISTRIBUIDAS USANDO RMI

Autor: John Carlos Arrieta Arrieta



EjemploLipeRmiServidorImc - Apache NetBeans IDE 12.2



Reconocimiento – NoComercial – CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

5. Escribir el código de las clases.

Clase CalculoRmiImplem

```
package johnarrieta.seminario.imc.rmi.net;

import johnarrieta.seminario.imc.rmi.lib.DatosImc;
import johnarrieta.seminario.imc.rmi.lib.IRemotaCalculoImc;

/**
 *
 * @author JOHN CARLOS ARRIETA ARRIETA
 */
public class CalculoRmiImplem implements IRemotaCalculoImc {

    private DatosImc datos;

    public CalculoRmiImplem() {
    }

    /**
     public DatosImc calcularImc() {

    */
    @Override
    public DatosImc calcularImc(DatosImc datos) {

        float resultado = 0;
        if (datos.getPeso() <= 0 || datos.getAltura() <= 0) {
            datos.setInterpretacion("ERROR: El peso y la altura deben ser mayores que 0");
            return datos;
        } else {
            resultado = datos.getPeso() / (datos.getAltura() * datos.getAltura());
            datos.setResultado(resultado);
            if (resultado < 18.5) {
                datos.setInterpretacion("Debes consultar un Medico, tu peso es muy bajo");
            } else if (resultado >= 18.5 && resultado <= 24.9) {
                datos.setInterpretacion("Estas bien de peso");
            } else if (resultado > 24.9 && resultado <= 29.9) {
                datos.setInterpretacion("Debes bajar un poco de peso");
            } else {
                datos.setInterpretacion("Debes consultar un Medico, tu peso es muy alto");
            }
            return datos;
        }
    }
}
```


Código de la clase Servidor

```
package johnarrieta.seminario.imc.rmi.net;

import java.io.IOException;
import johnarrieta.seminario.imc.rmi.lib.IRemotaCalculoImc;
import net.sf.lipermi.exception.LipeRMIException;
import net.sf.lipermi.handler.CallHandler;
import net.sf.lipermi.net.Server;

/**
 *
 * @author JOHN CARLOS ARRIETA ARRIETA
 */
public class Servidor {

    private int puerto = 9007;
    private CallHandler invocador;
    private Server servidor;
    private CalculoRmiImcImplem calculoImc;
    private IRemotaCalculoImc calculoImcRemoto;

    public Servidor() {
        invocador = new CallHandler();
        servidor = new Server();
        calculoImc = new CalculoRmiImcImplem();
    }

    public void iniciar() throws Exception {
        try {
            invocador.registerGlobal(IRemotaCalculoImc.class, calculoImc);
            servidor.bind(puerto, invocador);
        } catch (LipeRMIException ex) {
            throw new Exception("Error: No es posible invocar metodos remotos");
        } catch (IOException ex) {
            throw new Exception("Error: I/O");
        }
    }

    public void detener() {
        servidor.close();
    }
}
```

Código de la clase Principal



Reconocimiento – NoComercial – CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.


```
package johnarrieta.seminario.imc.rmi;

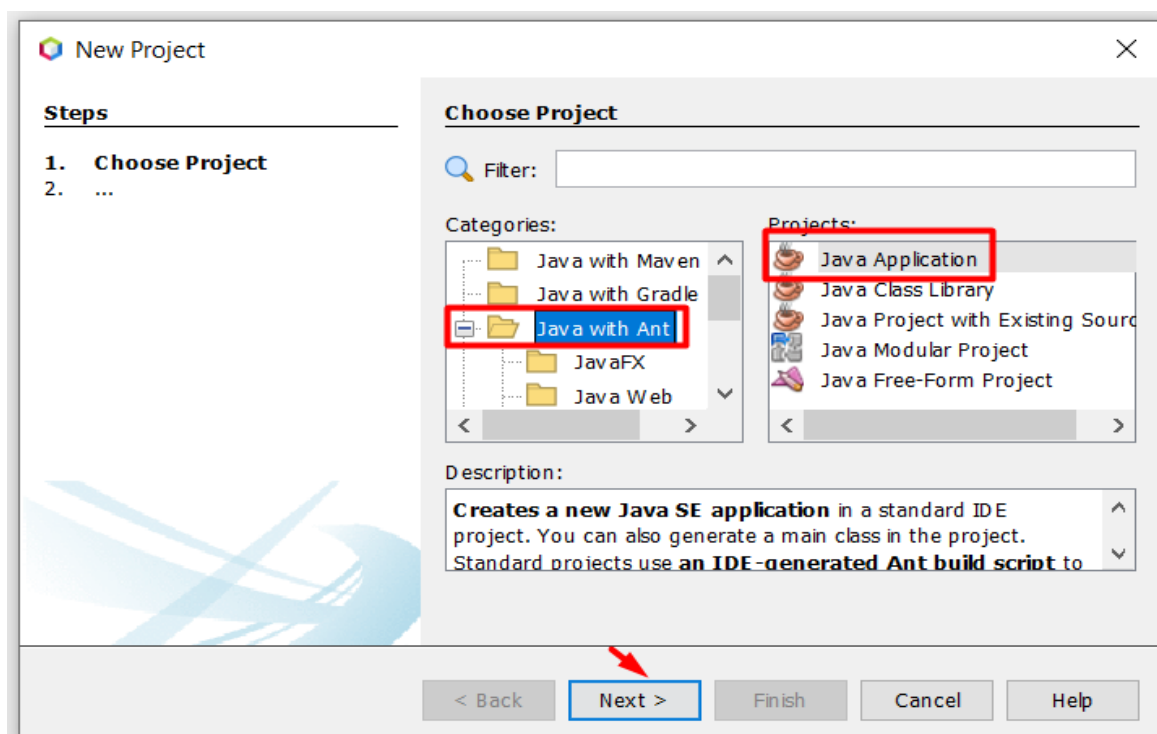
import johnarrieta.seminario.imc.rmi.net.Servidor;

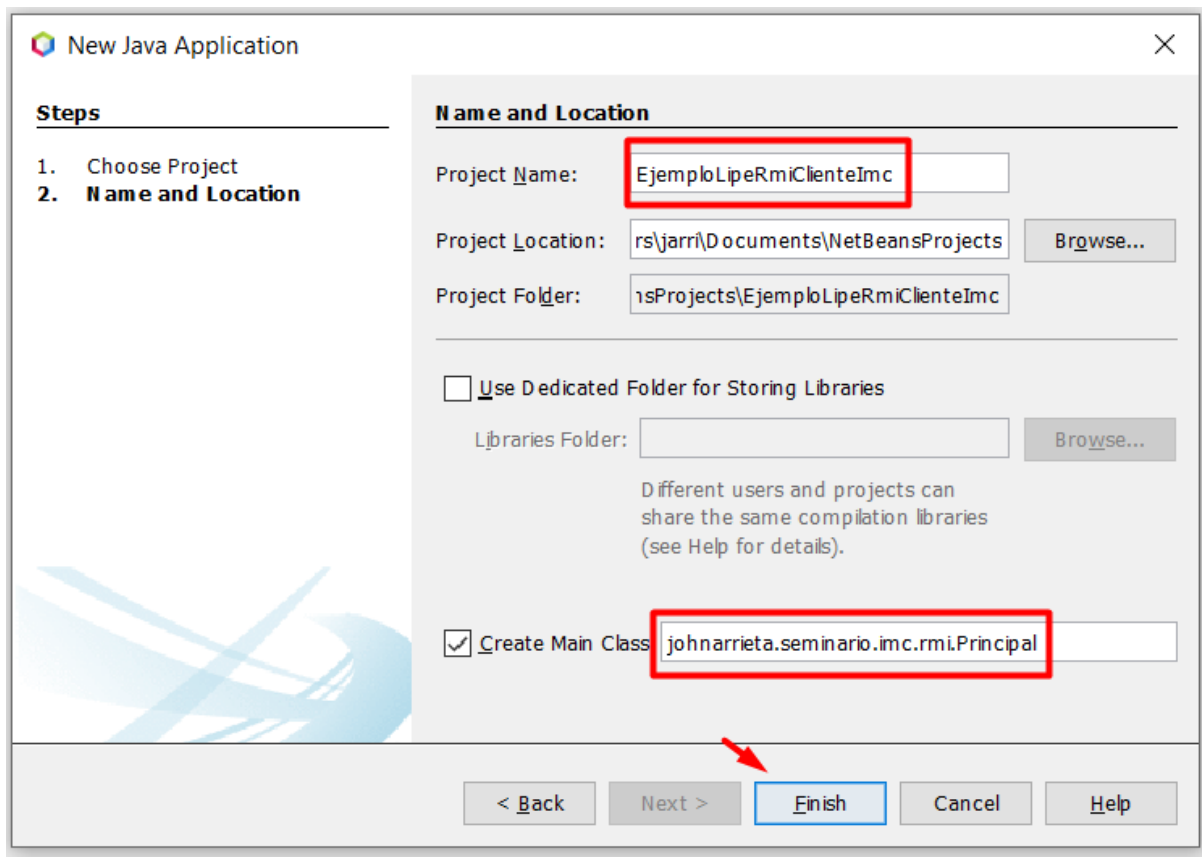
/**
 *
 * @author JOHN CARLOS ARRIETA ARRIETA
 */
public class Principal {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Servidor servicio = new Servidor();
        try {
            servicio.iniciar();
        } catch (Exception ex) {
            System.out.println(ex.getLocalizedMessage());
        }
    }
}
```

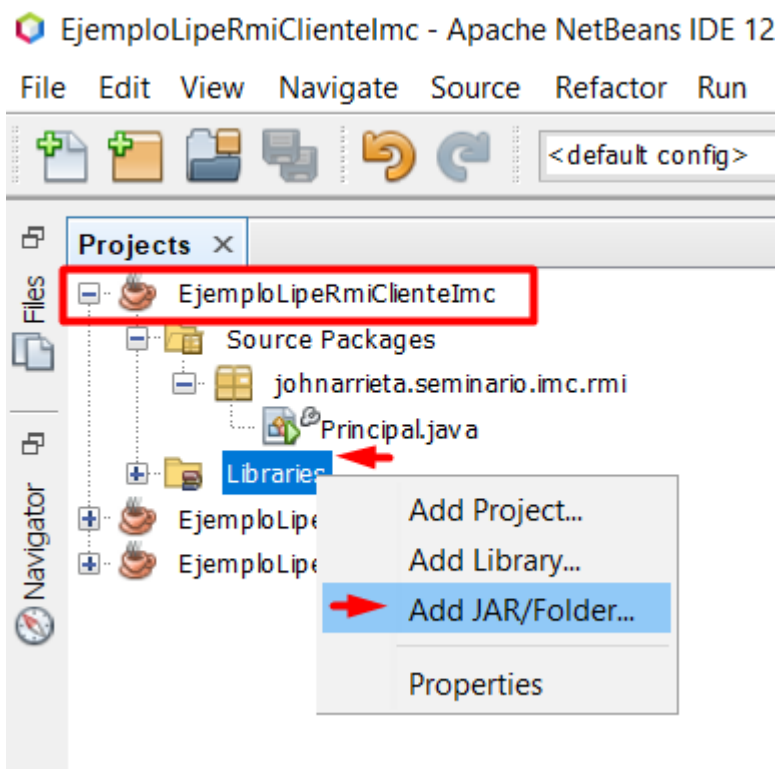
CREAR EL PROYECTO PARA LA APLICACIÓN CLIENTE

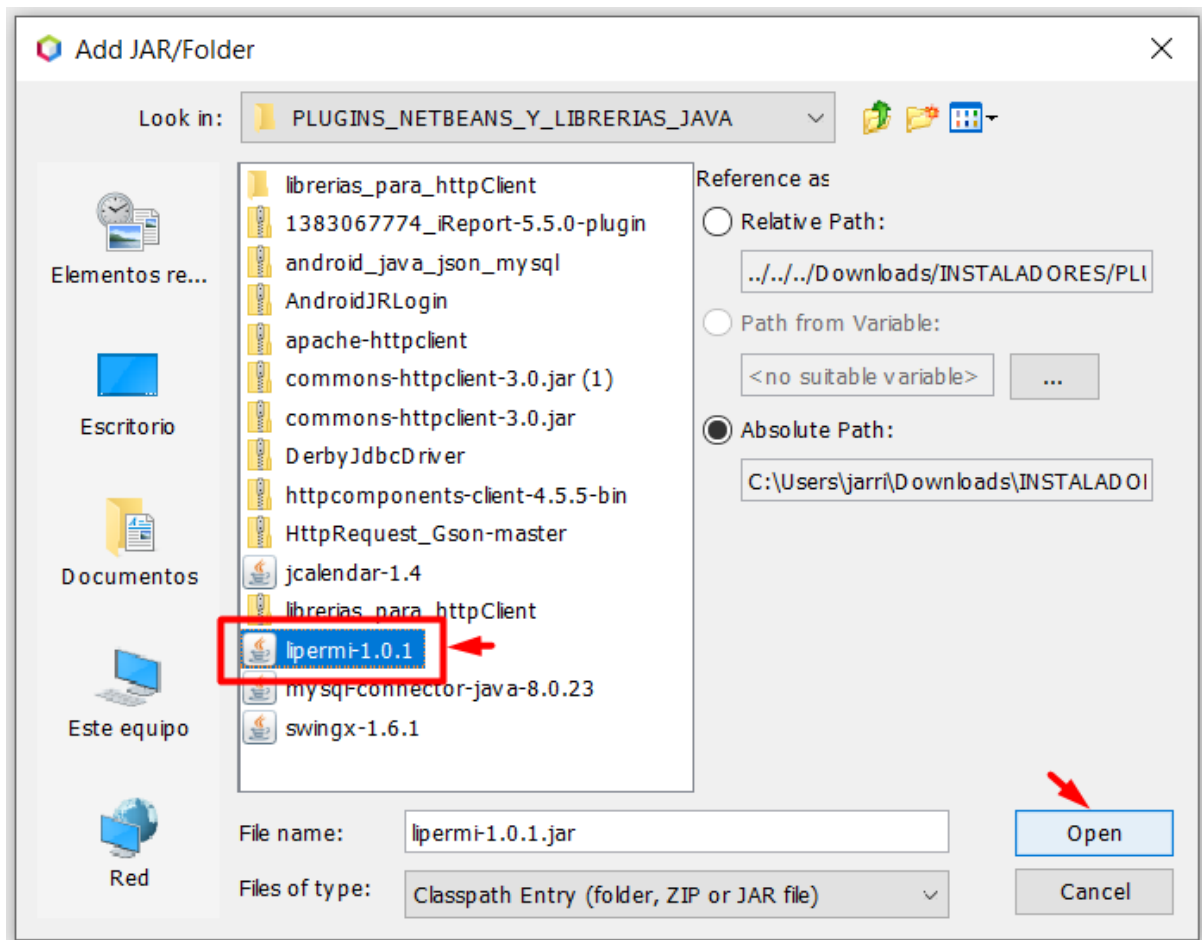
1. Crear un nuevo proyecto de tipo Java Application





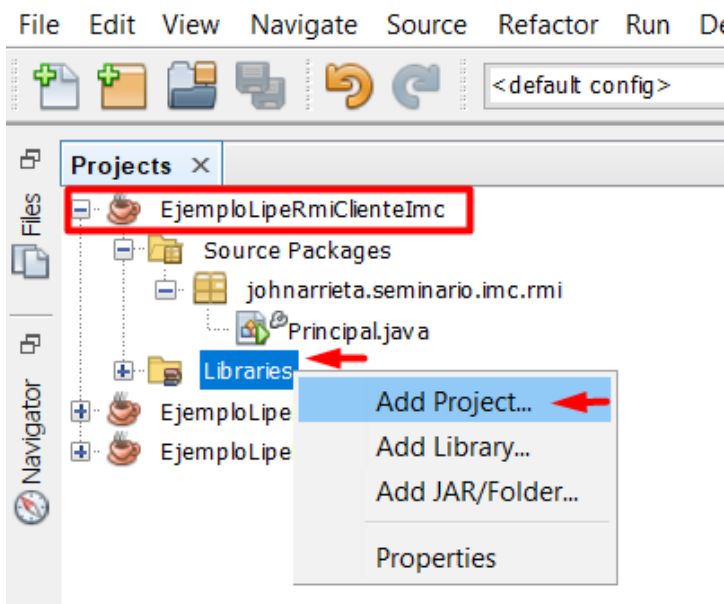
2. Agregar la librería LipeRMI

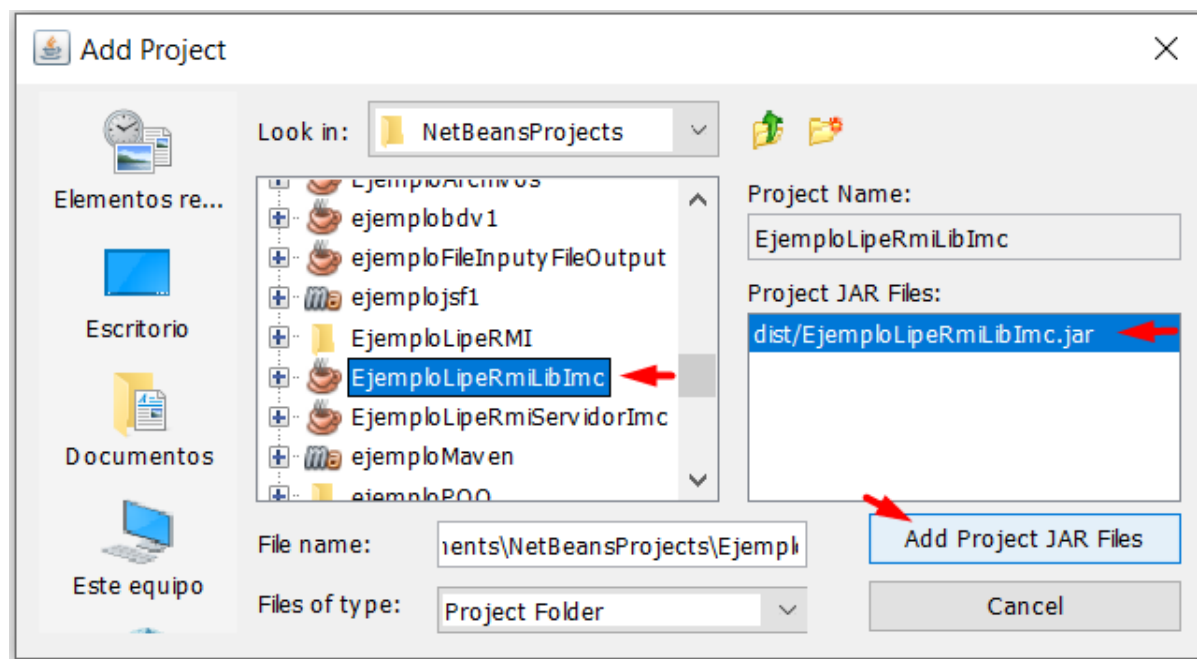




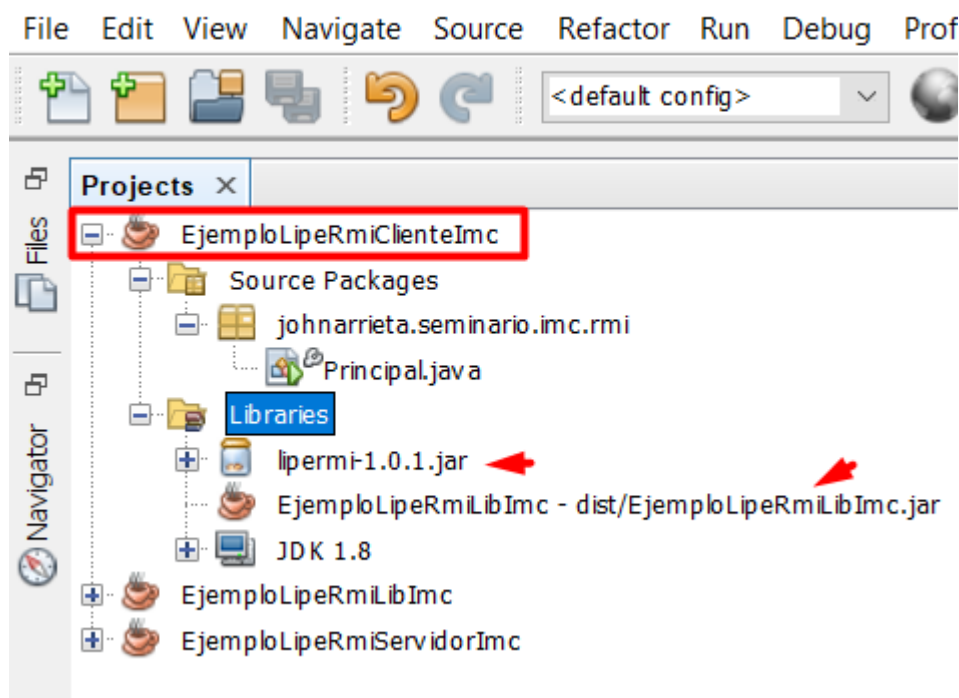
3. Agregar la librería del proyecto creado inicialmente

EjemploLipeRmiClienteImc - Apache NetBeans IDE 12.2



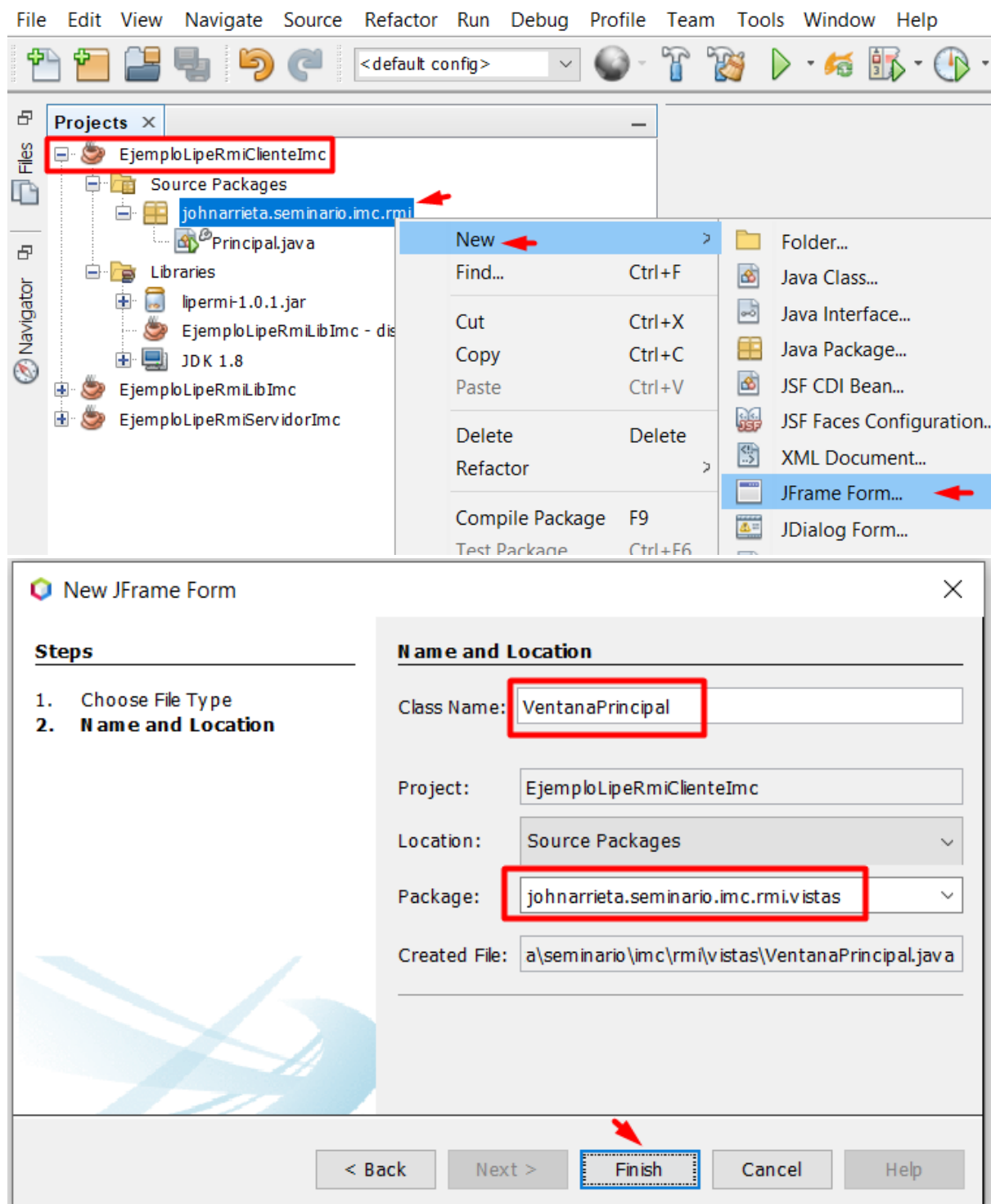


EjemploLipeRmiClienteImc - Apache NetBeans IDE 12.2

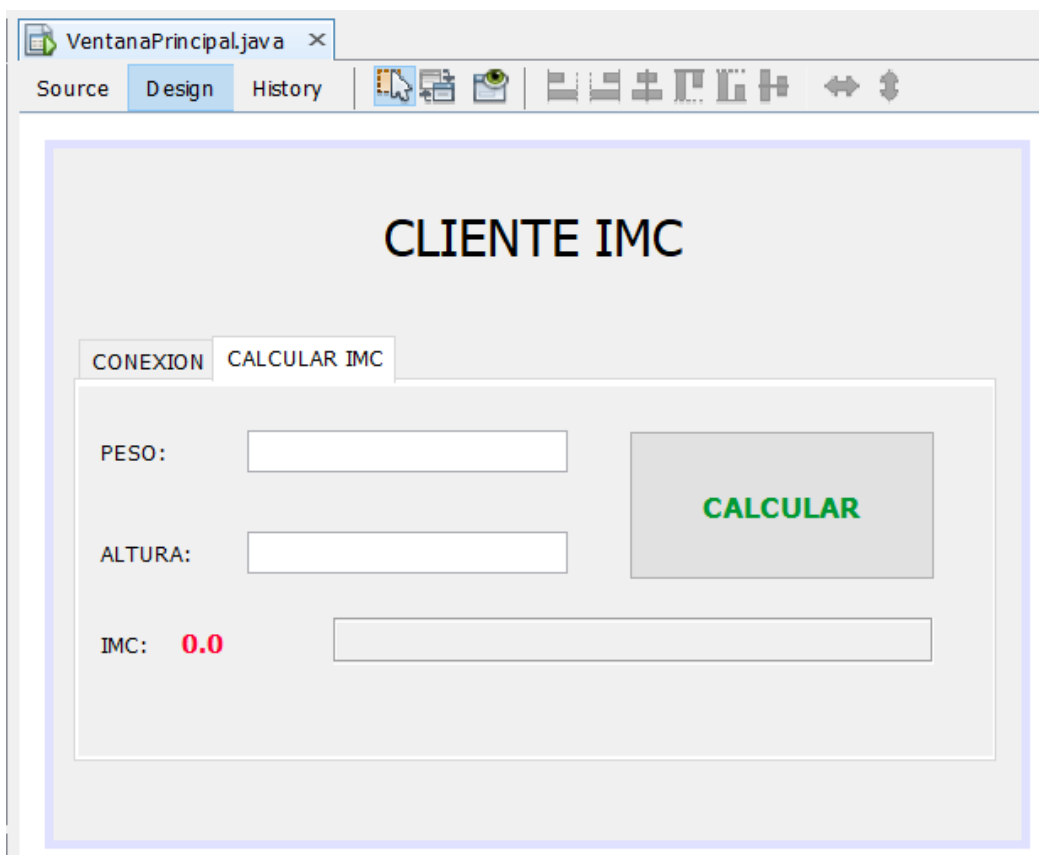


4. Crear la clase VentanaPrincipal.

EjemploLipeRmiClienteImc - Apache NetBeans IDE 12.2



5. Diseñar el formulario y programar los eventos de la GUI.



GUIA DE DESARROLLO DE APLICACIONES DISTRIBUIDAS USANDO RMI

Autor: John Carlos Arrieta Arrieta

Código de los dos principales manejadores (Oyentes de Eventos) más importantes de esta GUI

```
private void btnIniciarActionPerformed(java.awt.event.ActionEvent evt) {  
228  
229     try {  
230         if (btnIniciar.getText().equalsIgnoreCase("Conectar")) {  
231  
232             puerto = Integer.parseInt(campoPuertoServidor.getText());  
233             ipServidor = campoIPServidor.getText();  
234             invocadorRemoto = new CallHandler();  
235             cliente = new Client(ipServidor, puerto, invocadorRemoto);  
236             calculoImcRemoto = (IRemotaCalculoImc) cliente.getGlobal(IRemotaCalculoImc.class);  
237             btnIniciar.setText("Desconectar");  
238             btnIniciar.setForeground(Color.RED);  
239             txtEstado.setText("Conectado");  
240             txtEstado.setForeground(Color.GREEN);  
241         } else if (btnIniciar.getText().equalsIgnoreCase("Desconectar")) {  
242             cliente.close();  
243             btnIniciar.setText("Conectar");  
244             txtEstado.setText("Desconectado");  
245             btnIniciar.setForeground(Color.GREEN);  
246             txtEstado.setForeground(Color.RED);  
247         }  
248     } catch (IOException ex) {  
249         System.out.println("ERROR AL CONECTAR");  
250         ex.printStackTrace();  
251     }  
252 }
```

```
private void btnIniciarIActionPerformed(java.awt.event.ActionEvent evt) {  
255  
256     float peso = Float.parseFloat(campoPeso.getText());  
257     float altura = Float.parseFloat(campoAltura.getText());  
258     Thread hilo = new Thread() {  
259         @Override  
260         public void run() {  
261             try {  
262                 System.out.println("Peso: " + peso);  
263                 System.out.println("Altura: " + altura);  
264                 DatosImc datos = new DatosImc();  
265                 datos.setAltura(altura);  
266                 datos.setPeso(peso);  
267                 System.out.println("Enviados los datos\nEsperando respuesta");  
268                 datos = calculoImcRemoto.calcularImc(datos);  
269                 System.out.println("IMC: " + datos.getResultado() + "\nMensaje: " + datos.getInterpretacion());  
270                 txtResultado.setText(datos.getResultado() + "");  
271                 txtMensaje.setText(datos.getInterpretacion());  
272             } catch (Exception ex) {  
273                 JOptionPane.showMessageDialog(VentanaPrincipal.this, "ERROR con el cliente " + ex.getMessage());  
274                 System.out.println("ERROR con el cliente " + ex.getMessage());  
275                 ex.printStackTrace();  
276             }  
277         }  
278     };  
279     hilo.start();  
280 }
```

Código completo de la clase Ventana Principal



Reconocimiento – NoComercial – CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.


```

package johnarrieta.seminario.imc.rmi.vistas;

import java.awt.Color;
import java.io.IOException;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

import johnarrieta.seminario.imc.rmi.lib.*;

import net.sf.lipermi.handler.CallHandler;
import net.sf.lipermi.net.Client;

/**
 * @author JOHN CARLOS ARRIETA ARRIETA
 */
public class VentanaPrincipal extends javax.swing.JFrame {

    /**
     * Creates new form VentanaPrincipal
     */
    CallHandler invocadorRemoto;
    String ipServidor = "localhost";
    int puerto = 9007;
    IRemotaCalculoImc calculoImcRemoto;
    Client cliente;

    public VentanaPrincipal() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jPanel1 = new javax.swing.JPanel();
        campoIPServidor = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        campoPuertoServidor = new javax.swing.JTextField();
        btnIniciar = new javax.swing.JButton();
        jLabel4 = new javax.swing.JLabel();
        txtEstado = new javax.swing.JLabel();
        jPanel3 = new javax.swing.JPanel();
        jLabel5 = new javax.swing.JLabel();
        campoPeso = new javax.swing.JTextField();
        jLabel6 = new javax.swing.JLabel();
        campoAltura = new javax.swing.JTextField();
        btnIniciar1 = new javax.swing.JButton();
        jLabel7 = new javax.swing.JLabel();
        txtResultado = new javax.swing.JLabel();
        txtMensaje = new javax.swing.JLabel();
    }

```

```

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jLabel1.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel1.setText("CLIENTE IMC");

jLabel2.setText("DIRECCIÓN IP: ");

campoIPServidor.setText("localhost");

jLabel3.setText("PUERTO DE RED:");

campoPuertoServidor.setText("9007");

btnIniciar.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
btnIniciar.setForeground(new java.awt.Color(0, 153, 51));
btnIniciar.setText("Conectar");
btnIniciar.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnIniciarActionPerformed(evt);
    }
});

jLabel4.setText("ESTADO: ");

txtEstado.setForeground(new java.awt.Color(255, 0, 51));
txtEstado.setText("Desconectado");

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 131, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(10, 10, 10)
                    .addComponent(campoIPServidor, javax.swing.GroupLayout.PREFERRED_SIZE, 152, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(10, 10, 10)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 131, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(10, 10, 10)
                        .addComponent(campoPuertoServidor)
                        .addGap(10, 10, 10)
                        .addGroup(jPanel1Layout.createSequentialGroup()
                            .addComponent(jLabel4)
                            .addGap(10, 10, 10)
                            .addComponent(txtEstado, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addGap(10, 10, 10)
                            .addComponent(btnIniciar, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED_SIZE, 145, javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGap(10, 10, 10)
                            .addGap(137, 137, 137)
                        )
                    )
                .addGap(10, 10, 10)
            )
        )
        .addGap(10, 10, 10)
    );
jPanel1Layout.setVerticalGroup(

```

```

        jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(21, 21, 21)

            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel2)
                .addComponent(campoIPServidor, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel3)
                .addComponent(campoPuertoServidor, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(18, 18, 18)

            .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel4)
                .addComponent(txtEstado))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 19,
Short.MAX_VALUE)
                .addComponent(btnIniciar)
                .addGap(20, 20, 20))
        );

        jTabbedPane1.addTab("CONEXION", jPanel1);

        jLabel15.setText("PESO:");

        jLabel16.setText("ALTURA:");

        btnIniciar1.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
        btnIniciar1.setForeground(new java.awt.Color(0, 153, 51));
        btnIniciar1.setText("CALCULAR");
        btnIniciar1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnIniciar1ActionPerformed(evt);
            }
        });

        jLabel17.setText("IMC: ");

        txtResultado.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
        txtResultado.setForeground(new java.awt.Color(255, 0, 51));
        txtResultado.setText("0.0");

        txtMensaje.setBorder(javax.swing.BorderFactory.createTitledBorder(""));

        javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
        jPanel3.setLayout(jPanel3Layout);
        jPanel3Layout.setHorizontalGroup(
            jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel3Layout.createSequentialGroup()
                    .addGap(21, 21, 21)
                    .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(jPanel3Layout.createSequentialGroup()
                            .addGap(21, 21, 21)
                            .addContainerGap()

            .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel3Layout.createSequentialGroup()
                    .addGap(21, 21, 21)
                    .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(jPanel3Layout.createSequentialGroup()
                            .addGap(21, 21, 21)
                            .addContainerGap()

```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
    .addComponent(txtResultado, javax.swing.GroupLayout.PREFERRED_SIZE, 66,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(txtMensaje, javax.swing.GroupLayout.PREFERRED_SIZE, 285,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addGap(0, 0, Short.MAX_VALUE))
    .addGroup(jPanel3Layout.createSequentialGroup())

    .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addComponent(jLabel6, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(campoAltura, javax.swing.GroupLayout.PREFERRED_SIZE,
152, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 66,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(campoPeso, javax.swing.GroupLayout.PREFERRED_SIZE,
152, javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap(29, 29, 29)
        .addComponent(btnIniciar1, javax.swing.GroupLayout.PREFERRED_SIZE, 145,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap(28, 28, 28))
    );
    jPanel3Layout.setVerticalGroup(
        jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addGap(21, 21, 21)

    .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addGroup(jPanel3Layout.createSequentialGroup()

    .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel5)
        .addComponent(campoPeso, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(28, 28, 28)

    .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel6)
        .addComponent(campoAltura, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(21, 21, 21))
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addComponent(btnIniciar1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addGap(18, 18, 18)))

    .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel7)
        .addComponent(txtResultado, javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE)
```



```

        .addComponent(txtMensaje, javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addContainerGap(43, Short.MAX_VALUE))
    );

    jTabbedPane1.addTab("CALCULAR IMC", jPanel13);

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED_SIZE, 437,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(27, 27, 27)
            .addComponent(jLabel11)
            .addGap(32, 32, 32)
            .addComponent(jTabbedPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 203,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(37, Short.MAX_VALUE))
        );

    pack();
} // </editor-fold>

```

```

private void btnIniciarActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        if (btnIniciar.getText().equalsIgnoreCase("Conectar")) {
            puerto = Integer.parseInt(campoPuertoServidor.getText());
            ipServidor = campoIPServidor.getText();
            invocadorRemoto = new CallHandler();
            cliente = new Client(ipServidor, puerto, invocadorRemoto);
            calculoImcRemoto = (IRemotaCalculoImc)
                cliente.getGlobal(IRemotaCalculoImc.class);
            btnIniciar.setText("Desconectar");
            btnIniciar.setForeground(Color.RED);
            txtEstado.setText("Conectado");
            txtEstado.setForeground(Color.GREEN);
        } else if (btnIniciar.getText().equalsIgnoreCase("Desconectar")) {
            cliente.close();
            btnIniciar.setText("Conectar");
            txtEstado.setText("Desconectado");
            btnIniciar.setForeground(Color.GREEN);
            txtEstado.setForeground(Color.RED);
        }
    } catch (IOException ex) {
        System.out.println("ERROR AL CONECTAR");
        ex.printStackTrace();
    }
}

```

```
private void btnIniciarActionPerformed(java.awt.event.ActionEvent evt) {
    float peso = Float.parseFloat(campoPeso.getText());
    float altura = Float.parseFloat(campoAltura.getText());
    Thread hilo = new Thread() {
        @Override
        public void run() {
            try {
                System.out.println("Peso: " + peso);
                System.out.println("Altura: " + altura);
                DatosImc datos = new DatosImc();
                datos.setAltura(altura);
                datos.setPeso(peso);
                System.out.println("Enviados los datos\nEsperando
                                respuesta");

                datos = calculoImcRemoto.calcularImc(datos);
                System.out.println("IMC: " + datos.getResultado()+
                                "\Mensaje: " + datos.getInterpretacion());
                txtResultado.setText(datos.getResultado() + "");
                txtMensaje.setText(datos.getInterpretacion());
            } catch (Exception ex) {
                JOptionPane.showMessageDialog(VentanaPrincipal.this,
                    "ERROR con el cliente " + ex.getMessage());
                System.out.println("ERROR con el cliente " +
                    ex.getMessage());
                ex.printStackTrace();
            }
        }
    };
    hilo.start();
}
```

```
public JLabel getTxtEstado() {
    return txtEstado;
}

public JButton getBtnIniciar() {
    return btnIniciar;
}
```

```
// Variables declaration - do not modify
private javax.swing.JButton btnIniciar;
private javax.swing.JButton btnIniciar1;
private javax.swing.JTextField campoAltura;
private javax.swing.JTextField campoIPServidor;
private javax.swing.JTextField campoPeso;
```

```
private javax.swing.JTextField campoPuertoServidor;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel3;
private javax.swing.JTabbedPane jTabbedPane1;
private javax.swing.JLabel txtEstado;
private javax.swing.JLabel txtMensaje;
private javax.swing.JLabel txtResultado;
// End of variables declaration
}
```

Código de la clase Principal

```
package johnarrieta.seminario.imc.rmi;

import johnarrieta.seminario.imc.rmi.vistas.VentanaPrincipal;

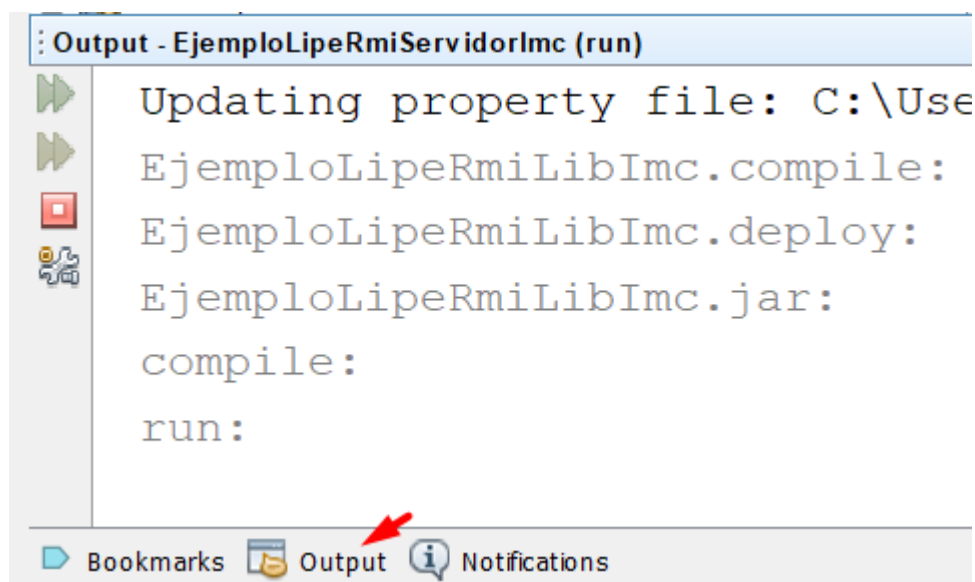
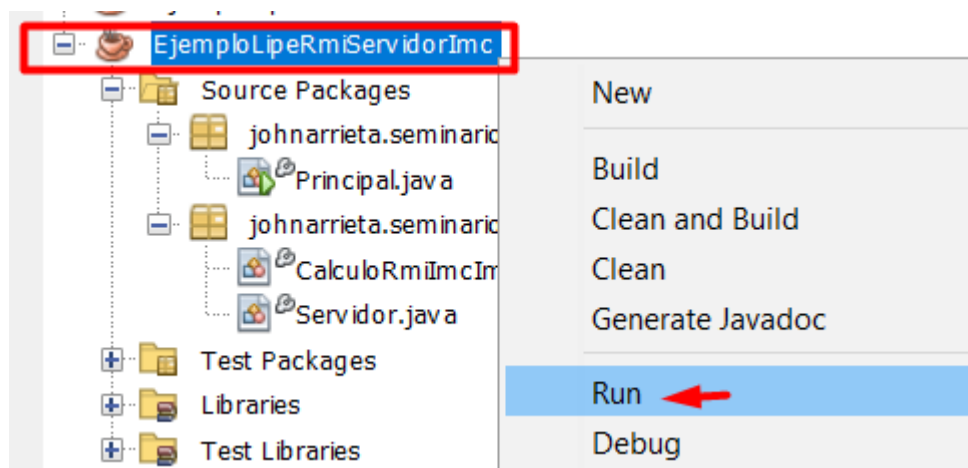
/**
 *
 * @author jarri
 */
public class Principal {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        VentanaPrincipal v = new VentanaPrincipal();
        v.setLocationRelativeTo(null);
        v.setVisible(true);
    }
}
```

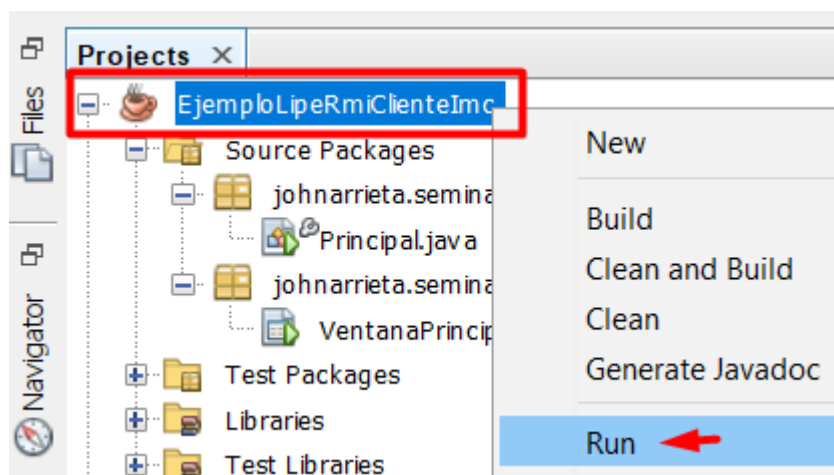
Probar el sistemas distribuido cliente servidor

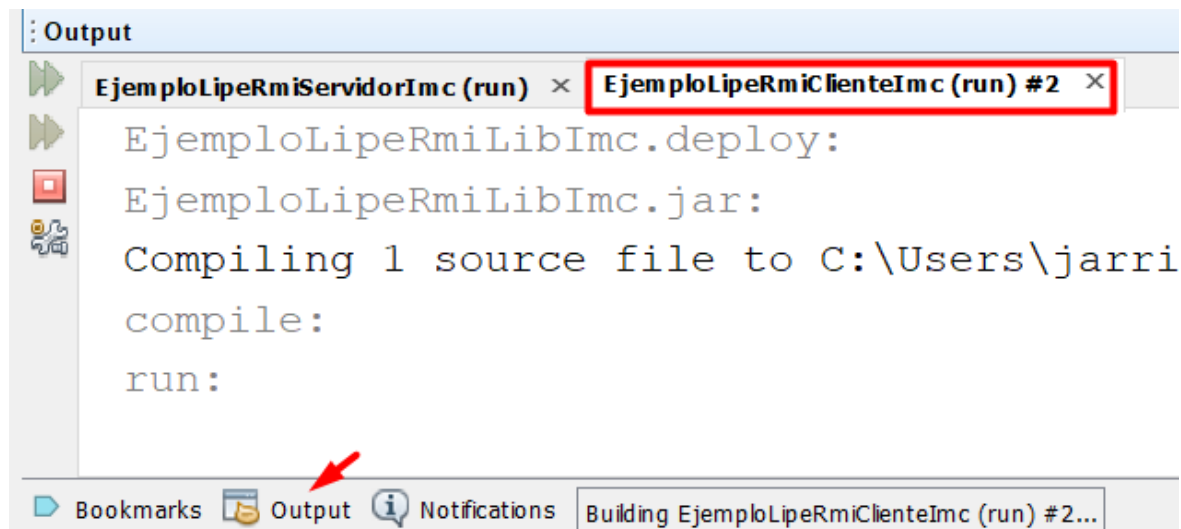


1) Ejecutar la aplicación Servidor

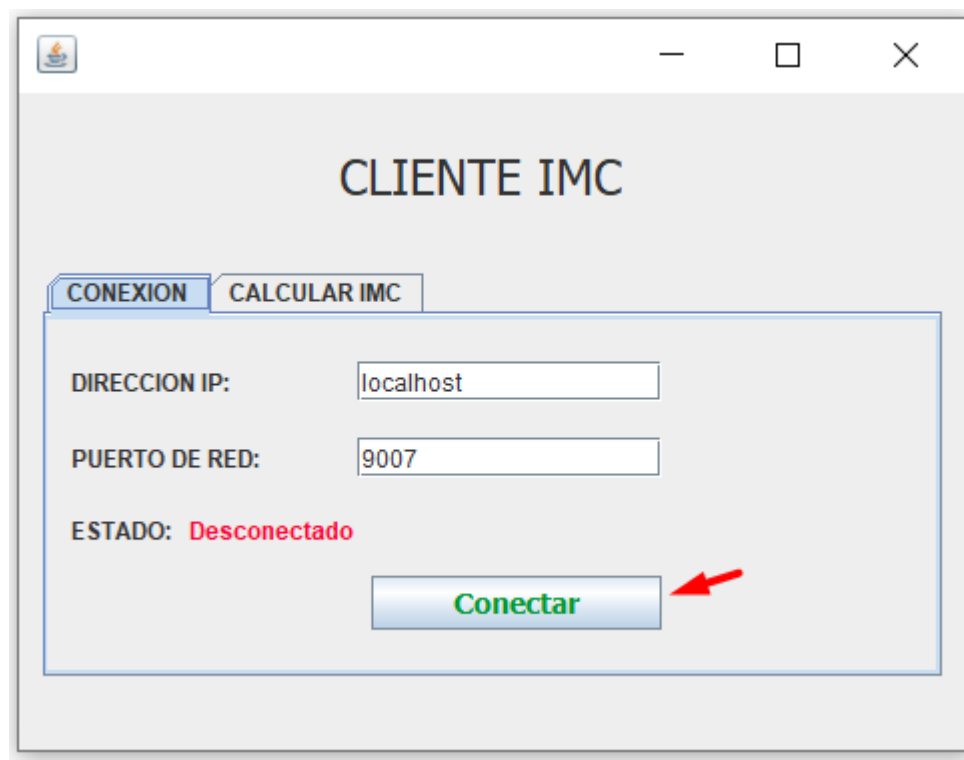


2) Ejecutar la aplicación Cliente





3) Iniciar la conexión desde el cliente hacia el servidor



4) Ingresar datos, solicitar realizar el cálculo de forma remota y mostrar el resultado.

CLIENTE IMC

CONEXION CALCULAR IMC

PESO: 78

ALTURA: 1.7

IMC: 26.9896... Debes bajar un poco de peso

CALCULAR